Rechnernetze und Anwendungen

SS 97

Prof. Dr. Christian Siemers,
Institut für Informatik
der
Technischen Universität Clausthal

Inhaltsverzeichnis:

KAPITEL 0: EINLEITUNG 1

0.1 Einführung in die Thematik 1

0.2 Gebräuchliche Abkürzungen 1

0.3 Literatur 2

KAPITEL 1: ÜBERSICHT 3

1.1 Klassifizierung von Rechnersystemen und -netzen 3

1.2 Architektur von Kommunikationssystemen 6

1.3 Routing und Flußkontrolle 8

1.4 Leistungsbewertung 9

1.5 Netzwerkmanagement 10

KAPITEL 2: GRUNDLEGENDE BEGRIFFE 11

2.1 Entfernung 11

2.2 Anwendungen 12

2.3 Übertragungstechnik 15

2.4 Übertragungswege 18

2.5 Topologie 19

2.6 Kopplung von Rechnernetzen 21

KAPITEL 3: TECHNISCHE GRUNDLAGEN 24

3.1 Bemerkungen zur Informationstheorie 24

3.2 Zerlegung analoger Funktionen in Oberwellen 25

3.3 Physikalische Grundlagen 27

3.4 Übertragung von Informationen 29

3.5 Informationsübertragungsmodell 30

3.6 Zur Ausnutzung von Kanälen 31

KAPITEL 4: LEITUNGSCODIERUNG 32

4.1 Grundsätzliches zur Codierungstechnik 32

4.2 Betriebsarten 33

4.3 Übertragungsprozeduren 34

4.4 Übertragungscodes 38

4.5 Gruppencodierung 42

KAPITEL 5: SERIELLE UND PARALLELE SCHNITTSTELLEN FÜR LOKALE KONZEPTE 47

5.1 Die serielle Schnittstelle RS-232C (V.24/V.28) 47

5.2 Die Weiterentwicklung zur RS422/423/485 55

5.3 Parallele I/O-Systeme: IEEE-488 bzw. IEC-625 57

KAPITEL 6: FELDBUSSYSTEME UND INDUSTRIELLE DATENKOMMUNIKATION 65

6.1 Gesamtübersicht zur Einordnung der Bussysteme 65

6.2 Netzwerkhierarchien bei CIM 66

6.3 Beispiele genormter Feld- und Sensor-Aktor-Busse 72

KAPITEL 7: LEITUNGSPROTOKOLLE 83

7.1 Einfache Leitungsprotokolle 83

7.2 BSC-Leitungsprotokolle 88

7.3 Das HDLC-LAPB-Leitungsprotokoll 90

KAPITEL 8: MEDIENZUGANGSKONTROLLE 93

8.1 Zuteilungsstrategien 94

8.2 Zufallsstrategien 98

KAPITEL 9: VERMITTLUNGSPROTOKOLLE 102

9.1 Einleitung 102

9.2 Die Funktionen und Protokollelemente in X.25 102

9.3 Das Internet-Protokoll 108

KAPITEL 10: TRANSPORTPROTOKOLLE 114

10.1 Die OSI-Umgebung 114

10.2 Transportprotokolle des ISO/OSI-Basisreferenzmodell 115

10.3 Transportprotokolle im Internet 117

KAPITEL 11: ANWENDUNGSORIENTIERTE SCHICHTEN 119

11.1 Die Kommunikationssteuerungsschicht 119

11.2 Die Darstellungsschicht 120

11.3 Anwendungsdienstelemente 120

11.4 Anwendungsdienste 121

KAPITEL 12: ISDN UND ATM-NETZE 122

12.1 ISDN 122

12.2 Breitband-ISDN 124

12.3 ATM 124

KAPITEL 13: NETZWERKMANAGEMENT 129

13.1 Ziele des Netzwerkmanagements 129

13.2 Aufgaben des Netzwerkadministrators 129

13.3 Managementansätze 131

13.4 Betriebsmittelverwaltung in ATM 132

KAPITEL 14: BEWERTUNG VON RECHNERNETZEN 134

KAPITEL 15: ROUTING-ALGORITHMEN 135

15.1 Einleitung 135

15.2 Begriffe zum Routing **135**

15.3 Klassifikation der Wegeermittlungsverfahren **136**

15.4 Nicht adaptive Wegeermittlung **137**

15.5 Adaptive Wegeermittlung **137**

KAPITEL 16: LAN, MAN UND WAN IM ÜBERBLICK **139**

STICHWORTVERZEICHNIS: **140**

Kapitel 0: Einleitung

0.1 Einführung in die Thematik

Netze werden für eine Vielzahl von Aufgaben eingesetzt; hier ist insbsondere die Kommunikation zwischen Rechnern auf sehr unterschiedliche Entfernungen hin zu nennen, was konsequenterweise auch zu verschiedenen Netzen führt. Andere Netze dienen 'lediglich' der einfachen Informationsverteilung (wie etwa Telefonnetze).

In dieser Vorlesung stehen die Netze zwecks Rechnerkommunikation im Vordergrund der Betrachtung; die Trennung z.B. vom öffentlichen Telefonnetz wird jedoch mit der Einführung von ISDN (Integrated Services Digital Network) nahezu aufgehoben. Rechnerkommunikation dient dabei in abnehmendem Maß der Kommunikation zwischen Terminals oder Satellitenrechnern und einem Zentralrechnern; in den Vordergrund schieben sich balancierte Kommunikationen zwischen gleichberechtigten Rechnern im Sinn eines verteilten Systems.

Diese Vorlesung befaßt sich mit den Grundbegriffen, Grundlagen und Architekturen von Rechnernetzen, ohne auf eventuelle spezielle Produkte einzugehen. Zu diesem Zweck werden nacheinander die Grundbegriffe, anschließend die technischen Grundlagen und die verschiedenen Protokollebenen behandelt. Im Anschluß daran folgt das Netzwerkmanagement.

Rechnerkommunikation kann auch auf Punkt-zu-Punkt-Kommunikationen, Feldbussystemen und Meßgerätebussen basieren. Aus diesem Grund werden diese Kommunikationsformen ebenfalls in einigen Kapiteln behandelt.

Der Abschluß dieses Curriculums bilden die vergleichende Darstellung von LAN, MAN und WAN, ISDN und ATM.

0.2 Gebräuchliche Abkürzungen

ISO	International Standard Organisation
OSI	Open System Interconnect
CCITT	Comité Consultatif Internationale de Télégraphique et Téléfonique
LAN	Local Area Network
MAN	Metropolitan Area Network
WAN	Wide Area Network
ISDN	Integrated Services Digital Network
ATM	Asynchronous Transfer Mode

TCP/IP	Transfer Control Program/Internet Protocol
FDDI	Fibre Distributed Data Interconnect
DCN	Data Communication Network
DÜE	Datenübertragungseinheit
DCE	Data Communication Equipment
DEE	Datenendeinheit
DTE	Data Teminal Equipment

0.3 Literatur

[1] Kowalk, W.P.; Burke, M.: Rechnernetze. Teubner-Verlag, Stuttgart, 1994. ISBN 3-519-02141-2

[2] Kauffels, F.-J.: Rechnernetzwerksystemarchitekturen und Datenkommunikation. Reihe Informatik, Band 54, BI Wissenschaftsverlag, Mannheim, 1991. ISBN 3-411-14763-6

[3] Hennekeuser, J.; Peter, G.: Rechnerkommunikation für Anwender. Springer-Verlag Berlin Heidelberg New York, 1994

[4] Preuß, L.; Musa, H.: Computerschnittstellen. Carl Hanser Verlag München Wien, 1991

[5] Schnell, G.: Bussysteme in der Automatisierungstechnik. Vieweg-Verlag, Braunschweig, 1994. ISBN 3-528-06569-9

Kapitel 1: Übersicht

Literatur: [1] Kowalk, Burke: Rechnernetze, Kapitel 1

Die Informatik befaßt sich mit der systmatischen algorithmischen Verarbeitung von **Information**. Unter Information werden dabei die **Zustände** der beachteten realen oder abstrakten Systeme verstanden. Die Beschreibung dieser Zustände erfolgt mittels **Symbolen**.

Die Informatik bedient sich dabei also intensiver **Modellbildungstechniken**, sie ist im Grunde genommen eine solche Technik. Ein dabei gewonnenes Symbol wird als **Datum** oder - falls an einen Transport gedacht wird - als **Nachricht** bezeichnet. Dieses Datum/Nachricht wird während der informatorischen Verarbeitung **gespeichert**, **umgewandelt** und **transportiert**.

Die **Speicherung** und das Auffinden (der Wiedererhalt) von Informationen wird dabei bei kleinen Gesamtdatenmengen durch die einfache Benennung der Daten realisiert. Bei größeren Datenmengen hingegen, insbesondere falls die Daten mit vielen Attributen ausgestattet sind, die für Suchkriterien genutzt werden sollen, werden Datenbanksysteme für die Speicherung und den Wiedererhalt benutzt.

Die **Umwandlung** von Informationen wird mittels Algorithmen (= Berechnungsvorschriften) durchgeführt. Diese Berechnung bezieht sich lediglich auf die symbolische Darstellung, die semantische Interpretation bleibt davon ausgeschlossen.

Der **Transport** von Information, d.h. Nachrichten, ist bei verteilten Systemen notwendig, um die symbolischen Darstellungen an die entsprechenden Instanzen zu verteilen.

Diese Vorlesung befaßt sich mit dem Transport von Nachrichten und den damit zusammenhängenden Methoden und Techniken.

1.1 Klassifizierung von Rechnersystemen und -netzen

Zunächst müssen Rechnernetze in ihrer Gesamtheit von anderen Informatiksystemen unterschieden werden; dies dient insbesondere dazu, die weiteren Methoden, die in diesem Curriculum besprochen werden sollen, auf diese beschränken zu können. Innerhalb der Oberklasse Rechnernetze werden dann verschiedene Klassen von Netzwerken definiert, die speziell die Funktionsweise charakterisieren.

Ein einzelnes Gerät mit Prozessor, Speicher und E/A-Geräten wird als **Rechner** bezeichnet. Existiert in diesem System nur eine Speicherhierarchie, jedoch mehrere Prozessoren, spricht man von einem **Mehrprozessorsystem**. Diese Prozessoren müssen im Sinn eines Mehrprozessorsystems natürlich frei programmierbar sein, d.h. ein

Rechner, der für spezielle Aufgaben Prozessoren einsetzt z.B. für einen Grafikcontroller gilt in diesem Sinn nicht als Mehrprozessorsystem.

Mehrprozessorsysteme werden wiederum als Rechner bezeichnet, falls die Anzahl der Prozessoren für die Funktionalität ohne Belang ist.

Ein **Rechnerverbund** besteht aus einer Anzahl von Rechnern, die als einheitliches System erscheinen, also an einer gemeinsamen Aufgabe arbeiten (können). Dazu ist eine Kommunikation zwischen den Rechnern notwendig, deren Intensität aus den jeweiligen Aufgaben hervorgeht. Ist die Kommunikationsmöglichkeit allerdings nur zum beliebigen Informationsaustausch gegeben, ohne daß ein gemeinsames Betriebsziel erkennbar ist, wird dies als **Rechnernetz** bezeichnet. Die Einteilung eines Zusammenschlusses von Rechnern in Rechnerverbund und Rechnernetz ist dabei fließend und willkürlich, da auch aus dem puren Informationsaustausch je nach Aufgabenstellung ein Rechnerverbund erhältlich ist.

Bei Rechnernetze steht also die Kommunikation zwischen den einzelnen Rechnern im Vordergrund, ohne daß die Rechner ihren Status als Einzelsystem verlieren. Bei Bedarf werden Informationen an andere übermittelt oder von anderen erhalten. Für die Lösung von Aufgabenstellungen ist der Zustand der anderen Rechner unerheblich.

Eine besondere Stellung nehmen die sogenannten **Client-Server-Systeme** ein. Sie bestehen aus einem System von Rechnern mit einem ausgezeichneten Rechner, dem Server, der an zentraler Stelle mit z.B. großer Plattenkapazität oder speziellen Ausgabesystemen (Farblaserdrucker, Netzwerkzugänge nach außerhalb etc.) eine gesonderte Stellung einnimmt. Der **Server** kann im Extremfall auch die einzige Station im Netzwerk sein, der über ein Festplattensystem verfügt. Die **Clients** werden in diesem Fall mit den notwendigen Bootinformationen und dem Betriebssystem vom Server versorgt; im weniger extremen Fall können Programme wie Compilersysteme, Textverarbeitung (TEX) auf dem Server zum Aufruf bereitstehen.

Client-Server-Systeme werden zur Kategorie Rechnernetze zugeschlagen, da ihr Hauptzweck nicht der gemeinsamen Aufgabenbearbeitung, sondern dem Ressourcensharing dient. Die komplette Unabhängigkeit der einzelnen Rechner im Netz voneinander ist in Richtung Server natürlich nicht mehr gegeben.

Abb. 1-1 faßt die Klassifizierung der Rechnersysteme zusammen.

Rechnernetze selbst können weiter klassifiziert werden, indem man ihre Ausdehnung bzw. bestimmte technische Merkmale zu Hilfe nimmt. Die Ausdehnung führt dabei zu 3 (bis 4, je nach Auffassung) Kategorien, die wie nachfolgend klassifiziert werden können. Wichtig ist dabei jedoch die Erkenntnis, daß diese Klassifizierung recht willkürlich ist (oder zumindestens erscheint) und durch weitere Erkenntnisse noch untermauert werden sollte:

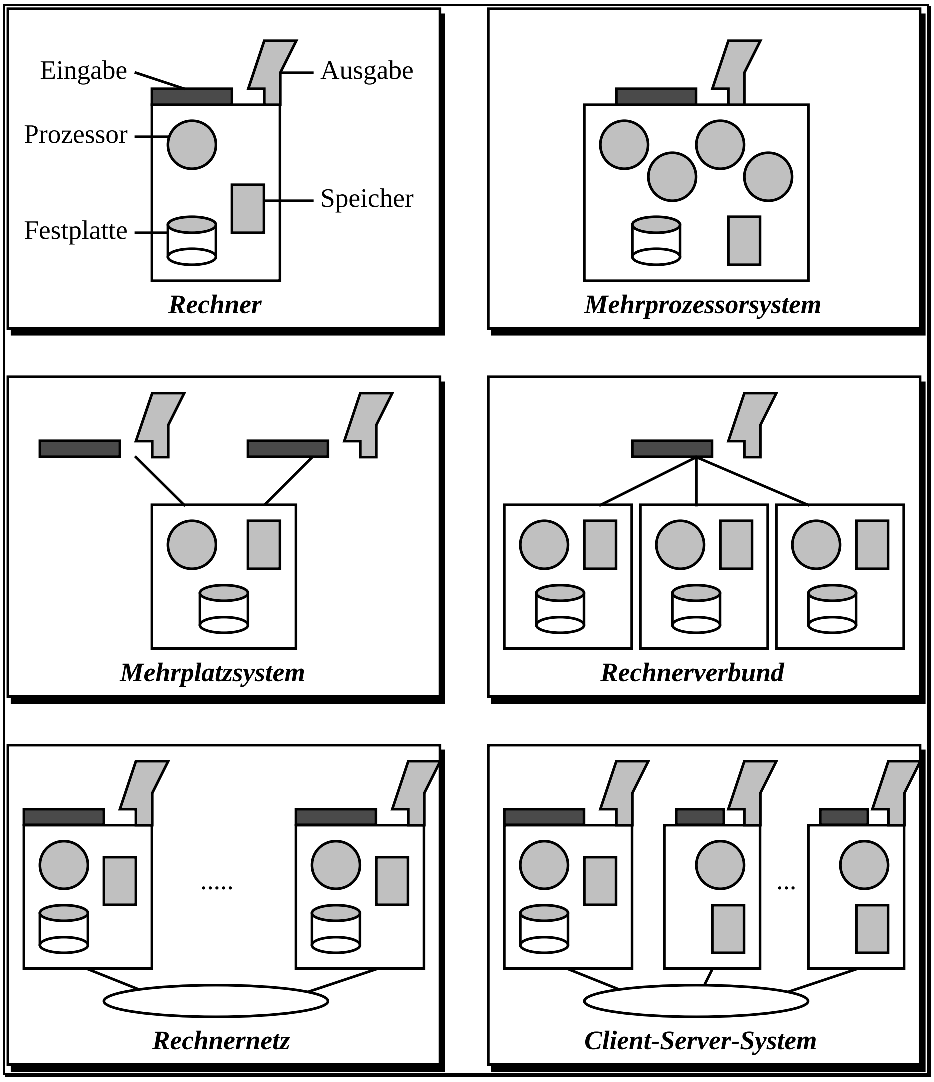

Abb. 1-1: Klassifizierung der Rechnersysteme

- Für einen Abstand weniger Meter spricht man von **Controller Area Network** (CAN). Dieser Begriff entstammt mehr dem Steuerungs- und Regelungsbau und wird in der Klassifizierung der Rechnernetze gewöhnlich nicht genutzt.
- Rechnernetze mit einem maximalen Abstand von wenigen 100 Meter zwischen den Rechnern werden als **Local Area Network** (LAN) bezeichnet. Diese Netzwerkdimension ist häufig auch an rechtliche Formen gebunden, etwa innerhalb eines privaten Geländes.
- Rechnernetze mit einer maximalen Ausdehnung von einigen 10 Kilometern, die also auf eine Region beschränkt sind, werden als **Metropolitan Area Networks** (MAN) bezeichnet.
- Rechnernetze, die eine derartige Region überschreiten bis hin zu weltweiten Rechnernetzen, werden dementsprechend als **Wide Area Networks** (WAN) bezeichnet.

Parallel zur Entfernungsklassifizierung können technische Merkmale zur weiteren Klassifizierung genutzt werden. Elektrische, leitungsgebundene Übertragung läßt sich dabei - je nach Leitungsstandard - für CAN und LAN einsetzen. Spezielle nicht-leitungsgebundene Verfahren (elektromagnetische Strahlung findet aufgrund der

Störungsanfälligkeit nur lokal Zugang. Lichtwellenleiter werden für Hochgeschwindigkeits-LANs, MANs und spezielle Einsätze im WAN-Bereich genutzt. Ihre Hauptvorteile sind geringe Dämfung mit höchsten Übertragungsraten, der wesentliche Nachteil liegt in der speziellen Verlegung, da elektrische Leitungen meist bereits vorhanden sind. Im allgemeinen Bereich der WANs wird häufig auf öffentliche Übertragungsnetze zurückgegriffen, z.B. Datex-P (X.25) etc.

1.2 Architektur von Kommunikationssystemen

Ein offenes Kommunikationssystem kann als komplexes System angesehen werden, das keineswegs mehr von nur einer Person alleine realisiert werden kann. Dies allein erfordert grundsätzliche Gedanken zur Strukturierung dieses Systems. Eine strikte Modularisierung der Software, wie sie generell empfohlen wird, bringt weitere Forderungen in Richtung Strukturierung: Die Definition der Architektur von Kommunikationssystemen ist zwingende Voraussetzung für eine 'offene' Implementierung.

Eine Standardarchitektur wurde 1978 von der ISO als Open System Interconnection (OSI) entworfen. Diese Standardarchitektur wurde derart entworfen, daß sie möglichst für alle Anwendungsfelder nutzbar war, mit zwei (negativen) Konsequenzen: Einzelne Schichten können durchaus leer sein, und die Implementierung eines Gesamtprotokolls kann sehr laufzeitintensiv sein.

Ein Kommunikationssystem wird in **Layer** oder **Schichten** eingeteilt. Jeder Schicht wird dabei eine Menge von Funktionen zugeordnet, wobei den höheren Schichten alle Funktionen der niedrigeren zur Verfügung stehen, selbst jedoch weitere zur Verfügung stellen können (Abb. 1-2).

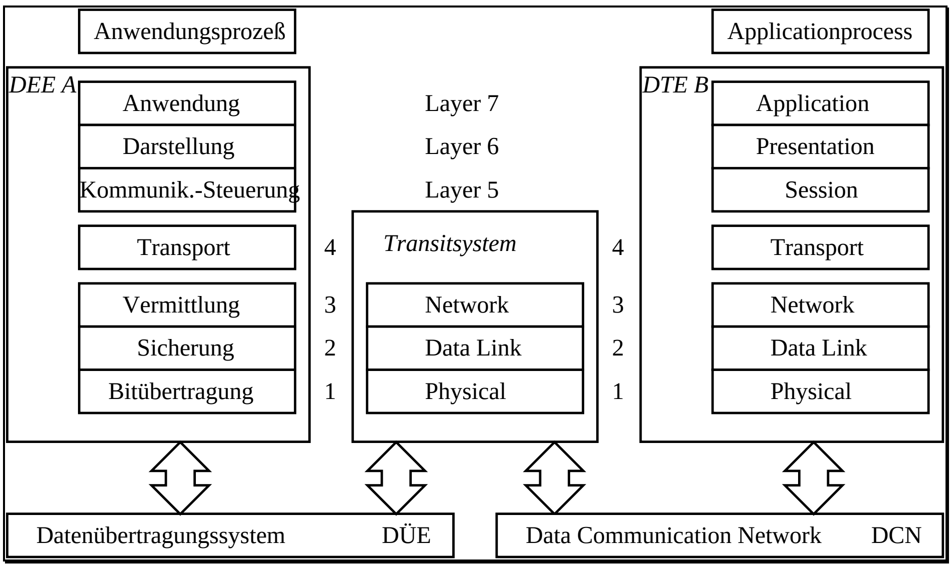

Abb. 1-2: OSI-Basisreferenzmodell

Andererseits soll die eigentliche Applikation keinesfalls den vollen Kanon aller Funktionsaufrufe und Dienste sehen, da dies zu einer unerwünschten Komplexität führen würde. Schicht 7 faßt also die gesamten Funktionen zusammen und bietet dem Anwendungsprozeß nur wenige Dienstprimitive (service primitives) an. Im übrigen sollte diese strikte Trennung der Schichten voneinander auch in den Layern 1 - 6 durchgeführt sein: Ein Schicht n kennt im Idealfall nur die Schichten n-1 und n+1. Die Praxis zeigt dabei aber einen tieferen Durchgriff, als dies in der Theorie gewünscht ist.

Die 7 Schichten werden grob zu zwei Teilen zusammengefaßt, die

- Anwendungsprotokolle (Application Protocols) und
- die Übertragungsprotokolle (Transport Protocols)

Zu den **Anwendungsprotokollen** gehören die **Anwendungsschicht**, die **Darstellungsschicht** und die **Kommunikationssteuerungsschicht**.

Die **Anwendungsschicht** besteht aus Protokollen, die die Kommunikation zwischen Anwendungsprozessen in Verbindung mit den Verwaltungsfunktionen, die die Anwendungsprozesse unterstützen, durchzuführen. Zu diesem Zweck werden anwendungsorientierte Grunddienste z.B. für File Transfer, Electronic Mail, Remote Job Entry, Virtual Terminal und damit zusammenhängende Darstellungskonventionen z.B. für die Festlegung der Struktur von auszutauschenden Dokumenten sowie Verwaltungsfunktionen definiert. Diese Protokolle gliedern sich in drei Grundtypen:

- Struktur-Verwaltungsprotokolle für die Kommunikation der Verwaltungsinstanzen globaler Art (gesamte Systemsicht)
- System-Protokolle für die Kommunikation der Verwaltungsinstanzen lokaler Art (Knotensicht: Welche Mittel stehen hier zur Verfügung)
- In-Line-Verwaltungsprotokolle für die Verwaltung der Betriebsmittel im Umfeld der Kommunikationsverbindung: Zugangskontrolle, Deadlock-Vermeidung.

Im Klartext: Viele Dienste der Anwendungsschicht stellen eine Erweiterung des Betriebssystems in Richtung der offenen Kommunikation dar. Dies bewirkt u.a. auch, daß viele Aufgaben des Netzwerkmanagements nicht von denen der Schicht 7 zu trennen sind.

Die **Darstellungsschicht** hat die Aufgabe, die Daten, die ausgetauscht werden, entsprechend den Vereinbarungen zu codieren bzw. zu interpretieren. Dies bedeutet natürlich auch die Vereinbarung einer gemeinsamen Semantik. Die dazu einzig zur Verfügung stehende offizielle Norm lautet **ASN.1** (Abstract Syntax Notation Numer One) für den Wertebereich und die Bedeutung der Daten, die konkrete Codierung von Daten in Bytes wird dabei durch **BER** (Basic Encoding Rules) festgelegt. Häufig werden jedoch auch herstellerspezifische Definitionen in Protokollen eingesetzt.

Die **Kommunikationssteuerungsschicht** hat die Aufgabe, die geregelte Kommunikation zwischen zwei Teilnehmern sicherzustellen. Dies umfaßt z.B. die online-Synchronisation oder das Einfügen von Synchronisationspunkten, die die jederzeitige Wiederaufnahme einer Kommunikation gestatten.

Die **Übertragungsprotokolle** umfassen die **Transportschicht**, die **Vermittlungsschicht**, die **Datensicherungsschicht** und die **Bitübertragungsschicht**.

Die **Transportschicht** hat die Aufgabe, den höheren Schichten und damit der Applikation eine einheitliche Zugangsschnittstelle zum Netz zur Verfügung zu stellen. Sie baut End-To-End-Verbindungen auf bzw. ab, kann dabei zwischen verschiedenen Übertragungswegen auswählen, kann auch mehrere virtuelle Verbindungen gleichzeitig unterhalten, eine physikalische Verbindung also zwischen mehreren Anwendungen multiplexen.

Die **Vernittlungsschicht** bestimmt einen Weg durch ein grundsätzlich ausgewähltes Netz von Vermittlungsstationen zu dem gewünschten Ziel. Die Zerlegung von zu großen Datenpaketen zur Übertragung fällt ggf. ebenfalls in diesen Aufgabenbereich. Hierbei wird eine Numerierung der erhaltenen Pakete eingefügt, um die richtige Reihenfolge wiederherstellen zu können.

Die **Datensicherungsschicht** garantiert den zuverlässigen Transport von Information zwischen zwei benachbarten (d.h. durch die gleiche Leitung miteinander verbundenen) Rechnern oder Vermittlungsstationen. Dies geschieht per Kontrollinformationen in einem zusätzlichen Rahmen um den bisherigen Datenblock. Zur Datensicherungsschicht gehört auch die Zugangskontrolle zum Medium (**Media Access Control**), was insbesondere bei LANs eine erhebliche Rolle spielt und aus diesem Grund auch häufig als Layer 2a bezeichnet wird.

Die **Bitübertragungsschicht** schließlich ist für die Übertragung der einzelnen Bits zwischen zwei Rechnern/Vermittlungsstationen am gleichen Link verantwortlich. Hierzu bedarf es weitreichender Festlegungen z.B. des Mediums, der Darstellung, der Stecker, der elektrischen Größen usw.

1.3 Routing und Flußkontrolle

Die Aufgaben der **Routing**-Verfahren, auch Verfahren zur **Leitwegbestimmung** genannt, ist die Auswahl eines geeigneten Weges für die Nachrichten, die zwischen zwei Rechnern ausgetauscht werden sollen. Diese Auswahl ist in der ISO/OSI-Schicht 3 (Network, Vermittlung) angesiedelt. Grundvoraussetzung ist natürlich die Auswahlmöglichkeit zwischen verschiedenen Wegen, was in LANs kaum gegeben sein dürfte.

Die Verhältnisse ändern sich in WANs, wo z.B. bei Nutzung eines X.25-Netzes die Nachricht häufig in Teilpakete eingeteilt wird, die durchaus differierende Wege laufen können. Routing-Verfahren werden sehr wichtig, wenn neben der gesicherten Übertragung die gute ('balancierte') Auslastung von Teilnetzen bei erhöhtem

Verkehrsaufkommen für die Übertragungsdauer entscheidend wird. Die Teilziele der Leitwegsbestimmung können dabei durchaus in Konkurrenz zueinander stehen, ihre Gewichtung ist den einzelnen Algorithmen überlassen: Gleichmäßige Auslastung der Leitungen, kurze Wartezeiten auf den Knotenrechnern, geringe Anzahl der zu durchlaufenden Knoten usw. können wesentliche Kriterien sein.

Routing-Verfahren werden in adaptive und nicht-adaptive unterteilt. **Adaptive Routingverfahren** sind dabei in der Lage, entsprechend dem Bedarf (Belastungsverhältnisse von Leitungen, Ausfall von Knoten oder Verbindungen) den Weg der Nachrichten zwischen zwei Rechnern zu verändern. Dabei werden drei Arten von adaptiven Routingalgorithmen unterschieden, die sich darin voneinander abgrenzen, wie sie die Informationen zur Bestimmung der 'optimalen' Leitwege zwischen zwei Rechnern erhalten:

Die **globalen Algorithmen** erhalten diese Informationen aus den verschiedenen Teilnetzen. Die **lokalen Algorithmen** sind hingegen komplett auf den einzelnen Teilknoten und ermöglichen somit an jeder Stelle eine lokale Auswahl - das globale Optimum kann hingegen verpaßt werden. Eine Kombination beider Algorithmen wird durch einen ständigen Informationsaustausch zwischen den Knoten erreicht, so daß nicht nur die Teilnetze in sich oder die Knoten für sich ein Optimum finden, sondern die Gesamtheit. Der Preis hierfür ist ein hoher Kommunikationsaufwand.

Nicht-adaptive Routingalgorithmen beruhen auf Messungen und Schätzungen des zu erwartenden Datenverkehrs. Die Schätzungen sind dabei mit Rücksicht auf die Topologie des Netzes, die Übertragungsgeschwindigkeiten, die Warteschlangen und das mittlere Datenaufkommen vorzunehmen. Nachteil dieser Algorithmen sind weniger in der Schätzmethode zu sehen - schließlich sind die Steuerparameter durchaus nachstellbar - sondern vielmehr in der Unflexibilität bei nicht-vorhersehbaren Ereignissen (Ausfall eines Knotens): Diese erfordern ein Stillegen des Netzes und eine Neuparametrisierung!

Während also adaptive Verfahren flexibler bei hohem Kommunikationsaufwand zur Steuerung sind, zeichnen sich nicht-adaptive Verfahren durch eine Robustheit wie einen geringen Managementaufwand aus, gepaart eben mit der geringen Flexibilität.

Das Problem der **Flußkontrolle**, welche bei Rechnern insbesondere unterschiedlicher Leistungsstärke wichtig werden kann, betrifft gegenüber dem Routing die Endrechner selbst, ist aber ansonsten verwandt. Bei einer schneller sendenden Quelle (**Source**) kann es zu einer Überflutung der Senke (**Drain**) kommen, so daß die Rechner in eine Kommunikation zur Vermeidung dieser Situation treten müssen, die als Flußkontrolle bezeichnet wird. Diese Flußkontrolle wird als End-To-End-Transportprotokoll in Schicht 4 durchgeführt und bedeutet die Einführung eines **Rückkanals**. Leitungsvermittelnde WANs werden hiervon weniger betroffen, da sie durch eine Online-Kopplung eine automatische Synchronisierung erzeugen. Speicherübermittelnde Systeme wie X.25 hingegen benötigen wirksame Instrumente zur Flußkontrolle.

1.4 Leistungsbewertung

Rechnernetze sind als technische Systeme in ihrer Leistung zu bewerten. Hierzu gehören Kosten, Nettoübertragungsrate, Durchsatz, mittlere und maximale Antwortzeit, Fehlerwahrscheinlichkeit, Verfügbarkeit usw., deren Kenntnis oder zumindestens Abschätzung für den Netzwerkmanager von entscheidender Bedeutung sind.

Werte werden durch Messung wie durch Modellbildung und/oder Simulation erhalten. Die als wesentlich erachteten Größen sind überdies von der jeweiligen Applikation abhängig: Echtzeitnetzwerke benötigen beispielsweise Aussagen über maximale Gesamtübertragungszeiten, während eine Maximierung eines Gesamtdurchsatzes hier von untergeordnetem Interesse sein wird.

Simulationen können an dieser Stelle wertvolle Erkenntnisse über gegenseitige Beeinflußungen liefern. LANs, die auf statistischen Zugriffsverfahren beruhen, sind nicht-deterministisch, was die maximale Übertragungszeit einer Nachricht betrifft. Hier können Wahrscheinlichkeiten erhalten werden, die je nach Verkehrsbelastung des Netzes Abschätzungen über die Einsetzbarkeit geben.

1.5 Netzwerkmanagement

Netzwerke sind als komplexe, verteilte Systeme aufzufassen, die aufgrund von fehlerhafter Soft- oder Hardware sowie durch äußere Einflüße (Beschädigung, EMV) ständig zu überwachen und zu steuern sind. Diese Überwachung und Steuerung sowie andere Aspekte wie Rekonfiguration, Wartung und Neuinstallation werden unter dem Stichwort **Netzwerkmanagement** zusammengefaßt.

Das Netzwerkmanagement wird zumeist zweistufig aufgebaut: Ein **Knotenmanagement** ist eng mit der Anwendungsschicht 7 verbunden; hier werden Informationen aus den darunterliegenden Schichten 1 bis 6 zusammengetragen und zu einer lokalen Ressourcenverwaltung hinzugefügt.

Das eigentliche Netzwerkmanagement nutzt diese lokal vorhandenen Daten, um in Zusammenarbeit mit Datenbankinformationen (MIB, Management Information Base) das gesamte Netz zu steuern. Weiterhin können Laufzeitmessungen durch spezielle Datenpakete ausgelöst werden. Als Resultat werden dann Informationen an die einzelnen Knoten z.B. zum Routing gesendet. Das zugehörige Kommunikationsprotokoll wird dem normalen Netzprotokoll hinzugefügt.

Kapitel 2: Grundlegende Begriffe

Literatur: [1] Kowalk, Burke: Rechnernetze, Kapitel 2

Die Nomenklatur von Rechnernetzen erweitert die übliche Begriffswelt der Informatik um eine Reihe von Begriffen und Definitionen, deren Verständnis und Wissen für das weitere Vorgehen zwingend notwendig ist. Aus diesem Grund werden in diesem Kapitel alle wichtigen Grundbegriffe des Gebiets um Netze und Rechnerkommunikation zusammengefaßt.

2.1 Entfernung

Rechnernetze alleine stehen für eine Kommunikationsmöglichkeit im Rahmen der technischen Gegebenheiten. Weit mehr als diese pure Übertragung von Daten ohne Bewegung von materiellen Datenträgern ermöglichen sie die Zusammenfassung der ‘Rechenkraft’ der Teilnehmer am Netz, aus globaler Sicht zu einem einzigen rechnenden Gebilde.

Im Unterschied zu einem einzigen Rechner, der innerhalb eines Gehäuses alle zentralen und peripheren Einheiten vereint, müssen die Teilnehmer an einem Netz nicht an einem Ort vereint sein; ihre räumliche Trennung ist nicht nur möglich, sondern der Regelfall bei einem Netz. Rechtliche und physikalische Gründe ergeben nun für ein verteiltes Netz von Rechnern sinnvolle Einteilungen, die mit der Entfernung in ursächlichem Zusammenhang stehen.

Die Menge aller Abstände d_{ij} der Rechner i und j wird zu einer unscharf definierten Metrik für Netze zusammengefaßt, da die Handhabung aller Abstände ein unnötige Komplexität bedeuten würde. Diese Menge ließe die Bestimmung eines maximalen Abstands zu, der bei manchen Netzdefinitionen auch vorgeschrieben ist. In der Regel werden sogenannte Ausreißer zur Bestimmung fortgelassen, falls nicht technische Gründe dagegen sprechen: Man spricht dann von einer typischen Entfernung.

Diese typischen oder maximalen Entfernung werden in drei Klassen eingeteilt, die auch rechtlichen Aspekten Rechnung tragen (beispielweise Trägerschaft der Kommunikationsübertragung: Privat oder öffentlich-rechtlich?). Diese Klassen lauten in der üblichen Nomenklatur:

- Entfernungen ≤ 1000 m: Local Area Networks, LAN
- Entfernungen ≤ 10000 m: Metropolitan Area Networks, MAN
- Entfernungen > 10000 m: Wide Area Networks, WAN

2.2 Anwendungen

Neben der Unterscheidung nach Entfernungen, das ein recht formales Kriterium darstellt, kann ebenfalls eine Differenzierung nach Anwendungen durchgeführt werden. Dies führt in sehr viel verstärkterem Maß zum Einsatzzweck von Rechnerkommunikationen, wobei andererseits zu bemerken ist, daß eine eindeutige Zuordnung eines Netzes zu einem Anwendungsgebiet ebenso unmöglich wie sinnlos ist.

Die Angabe von Anwendungen für Netze ist vielmehr als Merkmalsangabe zu verstehen; diese Merkmale können bei Planungen den Netzen zugrundegelegt werden, ebenso können die Hauptmerkmale eines Netzbetriebs zeitlich variieren.

2.2.1 Lastverbund

Ziel: Die gleichmäßige Auslastung verschiedener Ressourcen.
Methode: Aufteilung stoßweise anfallender Lasten auf verschiedene Rechner.

Jeder einzelne Rechner eines Netzes sollte möglichst lange arbeiten, um in einer großen Anzahl von Rechnern die Ressourcen optimal zu nutzen. Diese Forderung resultiert daraus, daß ein Rechner im Zustand ‘IDLE’ (nicht ausgefallen, aber zur Zeit ohne Aufgabe) eine Investition ohne Nutzung darstellt, und gilt für alle technischen Anlagen. Resultierend daraus müssen Aufgaben auf andere Rechner auslagerbar sein, falls im Netz Berechnungskapazitäten zur Verfügung stehen und auf einzelnen Rechnern stoßweise hohe Kapazitätsansprüche auftreten. Die Forderung lautet also, möglichst homogene Netze für einen Lastverbund zur Verfügung zu stellen, damit mehr als ein Rechner ein Programm ausführen kann.

Ist diese Forderung zumindestens teilweise erfüllt (teil-homogene Netze), kann ein ***Betriebsmittel-Pool*** eingerichtet werden, über den ein ***Lastmanager*** verfügt und so die einzelnen Tasks auf die Rechner im Netz aufteilt. Dadurch werden Leerlaufzeiten bei Rechnern (und Nutzern) verringert sowie ggf. andere Ressourcen wie Speicher und Drucker besser ausgelastet.

2.2.2 Leistungsverbund

Ziel: Verringerte Antwortzeiten.
Methode: Aufteilung einer Aufgabe in Teilaufgaben.

Falls es gelingt, eine (komplexe) Aufgabe in unabhängige Teilaufgaben zu zerlegen, deren Ergebniszusammenfügung quasi am Schluß ohne größeren Aufwand geschehen kann, können diese Teilaufgaben auf verschiedene Rechner im Netz verteilt werden und führen dann zu einer stark verringerten Antwortzeit des gesamten Systems auf diese Aufgabe.

Dies ist das Ziel des Leistungsverbunds von Rechnern via Netzwerk. Ist die Teilbarkeitsvorraussetzung gegeben, dann kann eine automatische Prozeßsteuerung einsetzen, die die einzelnen Tasks verteilt und via Protokoll anschließend wieder zusammenführt. Wichtiger ist in diesem Fall die automatische Trennung von Teilaufgaben, was ein wichtiges Forschungsgebiet darstellt: Welche Granularität der Teiltasks ist erhaltbar (möglichst unter Beibehaltung der ‘klassischen’ Programmiersprachen, um lediglich Neucompilierung durchführen zu müssen), welche ist optimal zur Durchführung (unter Berücksichtigung des ‘Kommunikationsoverheads’)?

Rechner, die unter diesen Bedingungen gemeinsam an einem Ziel arbeiten, werden als ***Rechnerverbund*** oder ***verteiltes System*** bezeichnet.

2.2.3 Kommunikationsverbund

Ziel: Übertragung von Daten, insbesondere Nachrichten, an verschiedene, räumlich getrennte Stellen.

Methode: Erstellung eines Briefdienstes (Electronic Mail).

Die Benutzung eines Netzes als Nachrichtendienst stellt eine der ersten und wichtigsten Anwendungen dar, obwohl Nachrichtendienste kaum als Begründung für die Einführung eines LAN ausreichen dürften. Für solche Briefdienste werden entsprechende Services eingerichtet, die den Anwendern das Versenden einer Nachricht ermöglichen, ohne daß dieser den Aufenthaltsort des Adressaten wissen muß. Netzübergänge z.B. zwischen LAN und WAN werden integriert. Voraussetzung ist lediglich, daß die Teilnehmer einen Rechner als ‘Personal Computer’ z.B. mit Zugangskennung betreiben.

Die technischen Bedingungen, die an ein solches System gestellt werden müssen, lauten: Editor, Benutzeragent(-Programm), der ein Adreßregister zur Verfügung stellt, Nachrichten (asynchron) empfängt, speichert und anzeigt, eventuelle Verschlüsselungsverfahren bei vertraulichen Nachrichten sowie ein Netzanschluß. Für Nachrichtendienste mit Netzübergängen müssen ggf. Protokollwandler vorhanden sein.

2.2.4 Datenverbund

Ziel: Bessere Auslastung von Harddisks, erhöhte Sicherheit und Verfügbarkeit.

Methode: Speicherung von Daten an verschiedenen Stellen.

Datenverbunde treten häufig in Zusammenhang mit verteilten Datenbanken auf, bei denen zusammenhängende Datenbestände durchaus auf verschiedenen, räumlich dislozierten Rechnern gehalten werden, auf. Die Haltung eines Teilbestands auf einem Rechner ist genau dann sinnvoll, wenn die Hauptnutzung auf diesem stattfindet. Unter diesen Umständen besteht die wesentliche Aufgabe des Datenbanksystems in der Konsistenzhaltung der Daten sowie dem Management des lokalen und nicht-lokalen Zugriffs.

Eine weitere Aufgabe des Datenverbunds außerhalb der Welt der verteilten Datenbanken ist in dem aufkommenden RAID-Standard zu sehen (RAID: Redundant Arrays of Inexpensive Disks). Bei diesem Standard ist vorgesehen, die Datenspeicherung im allgemeinen in redundanter Weise auf mehreren, auch verteilten Speichermedien, die ihrerseits nicht besonders hohen Ausfallsicherheiten (und Kosten!) unterliegen, zu managen, um durch die Redundanz eines hohe Sicherheit vor Datenverlust zu bekommen. Dieser RAID-Standard ist zur Zeit in Diskussion bzw. Normung.

2.2.5 Wartungsverbund

Ziel: Schnellere und billigere Wartung verschiedener Rechner.
Methode: Zentrale Störungserkennung und -behebung.

Im Wartungsverbund, für den ggf. spezielle Formen von Netzen ausschließlich eingesetzt werden (Wählnetz mit Telefon und Modem), werden Ferndiagnosen, Updates per Netzübertragung ebenso durchgeführt wie die Einsatzsteuerung und -unterstützung von Wartungspersonal.

Im ersteren Fall wird ein Serviceprozessor zusätzlich zum Einsatzfall eines Rechners (oder Geräts) eingesetzt, der über ein Netzwerk mit einer Servicezentrale verbunden ist und ggf. aktiviert wird. Der Einsatz dieser Systeme spart insbesondere Personalkosten, da beispielsweise bei Updates kein persönliches Erscheinen notwendig ist.

In anderweitig genutzten Netzen kann über Netzwerkmanagementprozesse der Zustand der Rechner ständig aufgezeichnet und in einer genormten Struktur (MIB, Management Information Base) gespeichert werden. Diese Informationen können dann im Servicefall wesentliche Hilfe darstellen, insbesondere kann bei Netzübertragung das Servicepersonal entsprechend an den Einsatzort gelenkt werden.

2.2.6 Funktionsverbund

Ziel: Bereitstellung spezieller Funktionen an verschiedenen Stellen.
Methode: Verteilung spezieller Aufgaben auf spezielle Rechner (Array-Prozessoren, Vektorrechner, Transputer, Supercomputer etc.).

Für bestimmte Rechenaufgaben werden auch heutzutage Spezialrechner benutzt, da sie diese Aufgaben in vergleichsweise geringer Zeit lösen können. Zu diesem Zweck müssen die Aufgaben so codiert werden, daß der Spezialrechner sie lösen kann (Crosscompiler).
Andere Aufgaben können beispielsweise in der Nutzung spezieller Hardware (Farbdrucker, große Plotter etc.) liegen, die dann einen Resourcenmanager am Netz besitzen.

2.2.7 Kapazitätsverbund

Ziel: Ausnutzung sämtlicher zur Verfügung stehender Rechenkapazität.

Methode: Versendung von Aufgaben an möglichst viele verschiedene Rechner.

Der Kapazitätsverbund steht in engem Zusammenhang mit dem Lastverbund, den verteilten Systemen und dem Datenverbund. Er stellt letztendlich einen Oberbegriff aus Betreiber- oder Benutzersicht dar, denn es steht weniger eine Aufgabe im Vordergrund, mehr die komplette Nutzung aller Ressourcen am Netz. Dies kann sich sowohl auf Software- und Hardwareresourcen als auch Rechenkapazitäten beziehen.

2.3 Übertragungstechnik

Die Kommunikation zwischen Komponenten benötigt Verbindungen, welche in geeigneter Weise Signale übertragen können. Geeignet heißt in diesem Fall, daß ein Interface existiert, um die übertragenen Signale den Komponenten in interpretierbarer Form zur Verfügung steht.

Zu diesem Zweck wurden verschiedene Technologien zur Übertragung entwickelt, die sich sowohl auf physikalische Technik als auch Protokolle beziehen. Die ersten Techniken bestanden in vorhandenen elektrischen Leitungen (Telefonkabel, twisted pair, Koaxialkabel etc.) zwischen jeweils zwei Rechnern (Point-To-Point-Verbindungen). Diese Verbindungen benötigten kaum Protokolle, waren daher sehr einfach handzuhaben und resultierten in den Modem-Übertragungen (Modulator-Demodulator, RS-232 als Verbindung zwischen Modem und Rechner).

Der Wunsch nach Multipoint-Verbindungen ließ neue Verfahren aufkommen: Ethernet für LANs verbreitete sich ab den 70er Jahren, Satelliten sorgten für die interkontinentale Kommunikation mit großen Geschwindigkeiten, in den 80er Jahren starteten dann die Glasfasernetze. Die heutige physikalische Grenze für die Bruttokommunikation liegt bei etwa 2 Giga-Bit pro Sekunde; dies bedeutet 500 ps pro Bit!

2.3.1 Vermittlungsnetze, Rundsendesysteme

Ein **Vermittlungsnetz** baut für jede Kommunikationssitzung eine eigene Verbindung zwischen zwei Teilnehmern auf, wobei nun die Verbindung als Point-To-Point-Verbindung angesehen wird. Für diesen Zweck können andere Netze, z.B. für Telefon genutzt werden; die Störung durch andere Teilnehmer ist, da die beiden exklusiv sind, praktisch sehr gering.

Bei Rundsendesystemen steht meist ein leistungsfähiger Kanal für viele Teilnehmer zur Verfügung. Der Zugang erfolgt durch ständiges Mithören am Kanal und Benutzung gleichzeitig (frequency multiplex) oder in Zeitscheiben (time multiplex, time division). Dieses Verfahren wird Multi-Point-Kommunikation, auch Multi-

Point-Access genannt. Das genaue Zugriffsverfahren wird später z.B. für Ethernet und Token-basierte Verfahren erklärt.

2.3.2 Leitungsvermittlung, Speichervermittlung

Bei der ***Leitungsvermittlung*** (circuit switching) wird zwischen den Endteilnehmern eine exklusive, reale oder virtuelle Leitung geschaltet, die ausschließlich sie benutzen dürfen. Auf diesem Prinzip beruht beispielsweise das Telefonnetz, die Teilnehmer sind in der Regel gegenüber Störungen geschützt. Das Netz ist dabei nicht in der Lage, Zwischenspeicherungen oder Taktumsetzungen vorzunehmen. Aus diesem Grund müssen die Teilnehmer vorher eine Synchronisation ihrer Übertragungsraten (neben den weiteren Werten zur Übertragung wie das Zeichenformat) vornehmen. Wichtigste Vertreter dieser Vermittlungsform sind das Telefonnetz, ISDN und Datex-L.

Die ***Speichervermittlung*** schaltet nur zwischen zwei Relais-Stationen eine echte physikalische Leitung, in der Relais-Station erfolgt eine gewisse Zwischenspeicherung. Die zu übertragenden Daten werden in Pakete fester Länge aufgeteilt und als solche verschickt, und zwar von Relaisstation zu Relaisstation. Der Weg für verschiedene Pakete kann dabei unterschiedlich lang und schnell sein, zudem können durch Überlast Pakete verloren gehen, so daß eine Fluß- und Fehlerkontrolle sowie Behebung unerläßlich ist. Vertreter dieser Gattung: Datex-P (X.25).

2.3.3 Verbindungslose und verbindungsorientierte Kommunikation

Muß sich ein Teilnehmer vor der Übertragung der Informationen beim Empfänger anmelden, so spricht man von einer ***verbindungsorientierten***, anderenfalls von einer ***verbindungslosen Kommunikation***. Diese Konzepte sind auf logische Basis angesiedelt, verbindungslose wie verbindungsorientierte Kommunikationen lassen sich auf Leitungs- oder Speichervermittlungen anwenden.

Die zu senden Datenpakete einer verbindungslosen Kommunikation werden ***Datagramme*** genannt.

2.3.4 Übertragungsmedien

Übertragungsmedien und -technik hängen stark voneinander ab: Die benutzten Signale sind meist elektrischer oder optischer Natur, die frequenz-, amplituden- oder phasenmoduliert sind (Optik: nur Amplituden). In seltenen Ausnahmefällen wird beispielsweise Ultraschall genutzt. Die Änderungsgeschwindigkeit dieser Modulation ist entscheidend für die Übertragungsrate, weniger die Ausbreitungsgeschwindigkeit, die für elektrische/optische Verfahren im wesentlichen gleich der Lichtgeschwindigkeit ist (bis auf einen Medieneffekt).

Die Medien im einzelnen:

- ***Verdrillte Kupferkabel*** (twisted pair) gibt es in geschirmter (STP, shielded TP) wie ungeschirmter Form (UTP). Diese Leiter können trotz ihrer Einfachheit bis zu 100 Mbit/s übertragen (bei 100 m Entfernung). Die Übertragungsreichweite und -geschwindigkeit ist im wesentlichen durch Störungseinflüße (elektromagnetische Induktion) und die Abflachung der elektrischen Impulse durch endliche Kapazitäten und Induktivitäten begrenzt.
- ***Koaxialkabel*** bestehen aus einem zentralen Leiter, der von einem peripheren Leiter abgeschirmt wird. Eine elektromagnetische Induktion wirkt auf beide Leiter gemeinsam und hebt sich gegenseitig auf, während die elektromagnetische Abstrahlung durch die Bauform sehr gering bleibt. Aus letzterem folgt eine geringe Dämpfung, so daß diese Kabel für größere Entfernungen bei hohen bis höchsten Frequenzen geeignet sind. Schwierig bleibt einzig die Einkopplung.
- ***Lichtwellenleiter*** dienen der Übertragung optischer Signale und sind für höchste Frequenzen wie Reichweiten geeignet. Die Dämpfung beträgt ca 1,5 dB/km, so daß kaum optische Verstärker eingesetzt werden müssen. Die Einkopplung von Signalen, die häufig aus modulierten Laserdioden stammen, erfolgt so, daß der Lichtstrahl entweder durch Totalreflexion im LWL gehalten wird (Stufenindexfaser) oder durch ständige Brechung zum Leitungsinneren (Gradientenfaser). Die Begrenzung der Übertragungsrate liegt in dem Verwaschen der Lichtimpulse, wobei das Licht amplitudenmoduliert wird: Strahlen, die vielfach reflektiert werden, haben einen längeren Weg!
- ***Richtfunktverbindungen*** benötigen quasi-optische Sicht und werden zur Überbrückung unwegsamen Geländes genutzt.
- ***Satelliten*** werden zur Überbrückung von Kontinenten oder als Rundsendesystem eingesetzt. Die Laufzeit zu geostationären Satelliten und zurück beträgt 200 ms, was zu erheblichen Rückmeldeverzögerungen führen kann.

2.3.5 Basisband- und Breitbandnetze

Eine ***Basisbandübertragung*** nutzt das gesamte Übertragungsspektrum (als Frequenzspektrum gesehen) nur zur Übertragung einer einzigen Kommunikation; die Kommunikation für andere Verbindungen erfolgt zu einem anderen Zeitpunkt, so daß für Basisbandnetze, die mehr als eine Verbindung aufnehmen, der Zeitmultiplex zur wesentlichen Modulationsart wird.

Bei den Zeitmultiplexverfahren werden noch synchrone Verfahren (gleiche Zeitscheibengröße für alle) sowie asynchrone unterschieden: Letztere erlauben – in gewissen Grenzen – die Variation der Zeitscheiben für optimalere Übertragung.

Breitbandnetze teilen das Frequenzspektrum eines Leiters in einzelne Kanäle auf, die sich durch verschiedene Frequenzlagen unterscheiden. Diese Technik ist noch sehr teuer, da höchste Anforderungen an die Interfaces gestellt werden.

2.4 Übertragungswege

Die Wege zur Übertragung, auch bei nichtleitungsgebundener Form, sind nahezu immer gesetzlich sanktioniert: Es gibt private oder öffentliche Anbieter, teilweise Monopole, die als Netzbetreiber den Auftrag zur Verfügbarmachung von Netzleitungen haben. International werden diese Netze durch bilaterale oder multilaterale Abkommen sowie die internationalen Standards geregelt.

2.4.1 Öffentliche Netze

Ein Netz wird als öffentlich bezeichnet, wenn im Prinzip jeder zu einer angemessenen Gebühr Nutzer dieses werden kann. Öffentliche Netze werden in der Regel von der Regierung kontrolliert und haben die Verpflichtung zum Angebot an jedermann.

Die Konsequenz daraus ist, daß ein öffentlicher Netzanbieter nicht nur die Pflicht zum Angebot hat, sondern auch das Recht zum alleinigen Angebot; ein LAN beispielsweise ist auch durch die Beschränkung auf privates Gelände gekennzeichnet. Wird nun für ein solches Netz öffentlicher Grund überquert, muß dies (mit Ausnahmen) durch den öffentlichen Anbieter geschehen.

2.4.2 Private Netze

Bei Nutzung ausschließlich privater Leitungen, die auch von öffentlichen Anbietern zur alleinigen Nutzung gemietet sein können, spricht man von privaten Netzen. Beispiele: Telefonnebenstellenanlagen, LAN, Buchungsnetze.

2.4.3 Elektrische Netze

Die elektrischen Netze sind quasi die natürlichste Form der Vernetzung: Die Darstellung der Informationen in Rechnern erfolgt auf elektrischem Wege, so daß das Interface zur Übertragung entsprechend wenig Aufwand erfordert. Zudem stellt die elektrische Übertragung einen sehr guten Kompromiß zwischen nahezu höchster Bitrate, Verlust, Anschlußfähigkeit und Kosten dar.

2.4.4 Lichtwellenleiter

Für Lichtwellenleiter gibt es prinzipiell kaum noch Probleme mit Bandbreiten, da die Übertragung in LWLs eine sehr große Bandbreite zur Verfügung stellen kann. Die Problematik liegt jedoch in den Interfaceeinrichtungen, die die Übertragungen auch gestalten müssen.

2.4.5 Satellitennetze

Satelliten werden im WAN-Bereich zur kontinentalen und interkontinentalen Kommunikation benutzt. Die langen Signallaufzeiten (Abstand Erde-Satellit mindestens 36.000 km) sind diese Verbindungen weniger zur interaktiven Kommunikation geeignet; im Bereich von WAN-Übertragungen müssen ggf. spezielle Satellitenprotokolle zum Einsatz kommen, um bei zu kurzen Rückmeldezeiten (-fristen laut Protokoll) nicht einen Abbruch der Verbindung zu signalisieren, obwohl nur eine lange Signallaufzeit vorliegt.

2.4.6 Funknetze

Werden elektromagnetische Wellen als Träger für Daten benutzt, spricht man von ***Funknetzen***. Dies Form wurde jahrelang zurückgedrängt (Unzuverlässigkeit), erlebt zur Zeit allerdings in Form der ***Wireless-LANs*** (WLAN) eine gewisse Rückkehr. Die Einsatzgebiete solcher Funknetze sind begrenzt, so z.B. bei temporären Netzen (zwecks Schulung) oder in unzugänglichen Gebieten.

2.5 Topologie

Eine weitere Unterscheidungsmöglichkeit für Netze besteht in der ***Topologie***. Diese Topologie ist zwar für den späteren Nutzer quasi transparent, da er sich in seinen Netzaktivitäten nicht danach richten muß, jedoch sind Performance, Zuverlässigkeit und weitere Werte stark von der Netztopologie abhängig.

Abb. 2.5-1 gibt einen kurzen Überblick über die wichtigsten Topologien; diese treten in der Realität natürlich auch in gemischter Form auf.

2.5.1 Busnetze

Sind alle Rechner am Netz an einem Strang angeschlossen, dann wird dies als Busnetz bezeichnet. Der Strang ist dabei ein einziger elektrischer Leiter (oder ein paralleles Leiterwerk mit logisch voneinander getrennten Funktionen, z.B. zur parallelen Datenübertragung), so daß aus physikalischer Sicht der Bus einem Rundsendesystem entspricht: Jeder Teilnehmer hört ständig am Bus, was passiert.

Aus logischer Sicht kann diesem Bus jedoch eine andere Struktur aufgeprägt werden, die dann z.B. für geordnete Kommunikation sorgt. Ethernet läßt alle mithören, eine sendewillige Station ist für ihren geordneten Sendezugriff selbst verantwortlich. Im Tokenbus kreist eine Sendeberechtigung (Token), während in älteren Systemen (Aloha) ein Master eine Zeitscheibeneinteilung vornimmt: Auf diese Weise wird aus dem System gleichberechtigter Teilnehmer ständig einer ausgezeichnet.

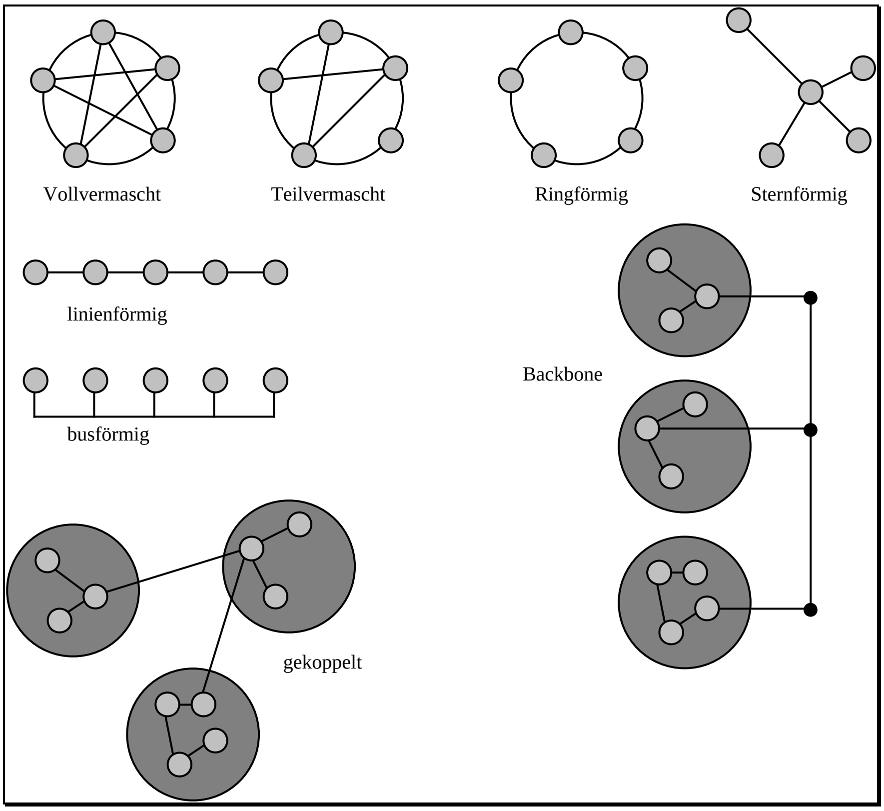

Abb. 2-1: Gebräuchliche Netztopologien

Eine Erweiterung des einfachen Buskonzepts wurde mit FDDI eingeführt: Der *Doppelbus DBDQ (Double Bus Double Queue).* In diesem Verfahren kreisen zwei Token in zwei Bussystemen, so daß eine bessere Auslastung und vor allem – durch selbständige Fehlerbehebung – eine wesentlich verbesserte Ausfallsicherheit die Folgen sind. FDDI zählt allerdings zu den Ringen, die durch die Schließung der offenen Busenden erhalten werden:

2.5.2 Ringnetze

Der bereits erwähnte Zusammenschluß der Busenden führt zu einem Ring, bei dem nunmehr kein Rundesendesystem mehr vorliegt, sondern nur in eine Richtung gesendet, aus der anderen empfangen wird. Technisch gesehen entspricht der Ring einer Reihe von Punkt-zu-Punkt-Verbindungen. Die Sendeberechtigung wird zumeist auch in einem Token- oder Doppel-Tokenverfahren weitergegeben.

2.5.3 Sternnetze

In Sternnetzen wird die Äquivalenz aller Busteilnehmer aufgegeben: Ein Rechner wird als zentraler Rechner im Stern ausgezeichnet. Dies geschieht im übrigen häufig auch bei den anderen Netztopologien, nur zumeist auf höheren Ebenen: Die Einrichtung eines zentralen Servers betrifft nicht die eigentliche Netzübertragung.

Der Nachteil eines Sternnetzes ist leicht detektiert: Der zentrale Rechner muß hochzuverlässig sein, ansonsten bricht das gesamte Netz sofort zusammen. Andererseits sind Ausfälle an den peripheren Rechnern sofort zu detektieren, da der zentrale Rechner ein zentrales Logbuch führen kann. Beispiele für sternförmige Netze sind das Telefonnetz und ARC-Netze.

Die Verkabelung von an sich bus- oder ringförmigen Netzen durch sternförmige Kabel mit speziellen Netzzugängen hat sich aus Kostengründen ebenfalls bewährt. Solche zentralen Netzverteiler, die nicht mit einem zentralen Rechner verwechselt werden sollten, werden ***Hubs*** genannt. Die Weiterentwicklung dieser Hubs, die in der Lage sind, mehrere Verbindung zur gleichen Zeit zu schalten (ein Vorgang, der bei einem busorientierten Netzwerk eigentlich nicht möglich ist), werden als ***Switch*** bezeichnet. Diese zentralen Verteiler können die durch die Verkabelung entstandenen Kapazitäten in sehr gutem Maß ausnutzen, müssen jedoch eine präzise Netzkenntnis haben (insbesondere für Weiterschaltungen) und die Netzpakete teilweise auswerten!

2.5.4 Vollvermaschte Netze

Der Sinn eines ***vollvermaschten Netzes***, bei dem jeder Rechner mit jedem anderen durch eine direkte Verbindung verbunden ist, liegt in der möglichst schnellen Kommunikation. Dies macht bei Rechnern im Leistungs- Last- oder Kapazitätsverbund besonders Sinn, falls sehr große Kommunikationsmengen zu erwarten sind.

Die einzelnen Verbindungen, deren Anzahl mit

$$0{,}5 * n * (n-1)$$

eingeht, werden als Punkt-zu-Punkt-Verbindungen geschaltet. Mit anderen Worten: Der Aufwand im Netz und in den Rechnern steigt quadratisch an.

2.5.5 Gekoppelte Netze

Die Kopplung (beliebiger) Rechnernetze miteinander wird ***gekoppeltes Netz*** (internet) genannt. Die Kopplung selbst kann dabei sehr aufwendig sein, so daß im nächsten Abschnitt kurz darauf eingegangen wird.

2.5.6 Backbone-Netze

Eine spezielle Form der Kopplung liegt vor, wenn durch das Koppel-Netz lediglich die Zusammenfügung der bisherigen Teilnetze zu einem größeren, das aber logisch einem einzigen entspricht, erreicht werden soll. Dies geschieht meist durch schnelle Übertragungen und wird ***Backbone-Netz*** genannt.

2.6 Kopplung von Rechnernetzen

Bei der Kopplung von Rechnernetzen miteinander werden hierfür zumeist wiederum Rechner eingesetzt, um die umfangreichen Transformationen entsprechend zu bearbeiten. Ein solches Gerät wird gemäß ISO allgemein ***Relais*** (relay) genannt. Für diese Relais wird je nach Aufgabenbereich ein weitere Klassifizierung betrieben, die sich an den nicht behandelten Protokollschichten orientiert.

Allgemein gehören zu den Aufgaben eines Relais die Bearbeitung von:

- Adressierung und Namensvereinbarungen
- Wegewahl (routing) und Flußkontrolle
- Informationsgrößen, wie Paketzähler oder Fenstergrößen
- Umsetzung von Paketgrößen
- Umsetzung zwischen verbindungsorientierten und verbindungslosen Diensten
- Fehlerkontrolle
- Zeitüberwachung
- Weiterleitung von Unterbrechungen
- Senden von Statusberichten und Abrechnungen
- Kontrolle des Nutzerzugriffs
- Verbindungsauf- und -abbau

2.6.1 Repeater

Repeater können als Verstärker angesehen werden: Die bearbeitete Schicht ist die Schicht 1, die Bitsignale eines Netzes werden in Bitsignale des anderen umgesetzt. Dies kann sich auch auf die Kopplung gleichartiger Netze beziehen, die aus technischen Gründen voneinander getrennt sein müssen (etwa Reichweite).

Alle weiteren Schichten müssen zwischen den Netzen vollkommen identisch sein, ein Umsetzung erfolgt nicht.

2.6.2 Bridge

Bridges müssen die Blöcke (frames, Rahmen) des einen Netzes in die des zweiten Netzes und umgekehrt umsetzen können. Die Umsetzung bezieht sich also auf Schicht 2, die Bearbeitung der Adressierung sowie der Blockfehlersumme ist notwendig. Das bereits erwähnte Konzept der Switches stellt eine spezielle Form der Bridges dar.

2.6.3 Router

Router müssen die Pakete eines Netzes in solche eine anderen Netzes umsetzen. Neben der Anpassung der Adressen betrifft dies auch das Paketformat, das in unterschiedlichen Netzen auch unterschiedlich sein kann. Fehlermeldungen seitens der Router können die Adressierung, sollten jedoch nicht die Paketformate betreffen; letzteres ist Aufgabe der Endteilnehmer.

2.6.4 Gateway

Gateways verbinden Netze, deren Protokolle sich bereits in den Anwendungsschichten unterscheiden. Hierzu können diverse Postdienste (Electronic Mail) oder Dateiübertragungen gezählt werden, ebenso kann die Darstellung der Zeichen variieren (ASCII kontra EBCDIC). Der Aufwand in Gateways ist also entsprechend hoch.

Kapitel 3: Technische Grundlagen

Literatur: [1] Kowalk, W.P.; Burke, M.: Rechnernetze, Kapitel 4

3.1 Bemerkungen zur Informationstheorie

Die Informationstheorie, durch Shannon im Jahre 1950 als **Kommunikationstheorie** begründet, betrachtet den Informationszustand der miteinander verbundenen Rechner als unterschiedlich: Der sendende Rechner weiß mehr als der empfangene, wobei durch die Übermittlung der Nachricht dies zumindestens gemindert wird.

Dieser unterschiedliche Kenntnisstand wird **Entropie** genannt. Dieser Begriff stammt ursprünglich aus der Physik (Thermodynamik) und beschreibt dort den Ordnungszustand eines statistischen Systems, wobei eine niedrigere Entropie einem geordneterem System entspricht. (Anmerkung: Ein in sich abgeschlossenes physikalisches System vermag seine Entropie nur zu steigern, es wird ‘ungeordneter’.)

Der (informatorischen) Entropie H wird ein Maß für die Informationsmenge, die ein Empfänger benötigt, um sein Unwissen zu beseitigen, zugeordnet (differentiell oder absolut). Der Empfänger fragt hierzu, der Sender antwortet mit Ja/Nein. Da die einzelnen Zeichen, aus denen eine Nachricht besteht, unterschiedlich häufig sind, muß dies in Form einer Gewichtung (p_i) beachtet werden. Damit kann die Entropie H mittels

$$H = -\sum_i p_i \bullet \log_2 p_i$$

definiert werden, wobei diese Definition der ***ideellen Entropie*** entspricht. Die ***wirkliche Entropie*** kann nur mit Angabe des tatsächlichen Verfahrens errechnet werden und ergibt numerische Werte $\geq H_{ideell}$. Die Bemühung, einen Code zu finden, der der ideellen Entropie entspricht, führt (theoretisch) zu optimalen Codes, da alle auftretenden Fälle ihrer Gewichtung nach codiert werden.

Es sollte an dieser Stelle nicht unerwähnt bleiben, daß der Entropiebegriff ca 1958 durch von Neumann geprägt wurde – mehr als spontane Begriffsbildung denn als T ransformation aus anderen Naturwissenschaften, hier der Physik. Andererseits sind gewisse Verwandtschaften unverkennbar, so daß sich die Begriffsbildung durchgesetzt hat.

Die theoretischen Codes bleiben für eine praktische Realisierung nur in Form der Angabe einer Obergrenze interessant, da sie sich in idealer Form nur fallbezogen definieren lassen. Rechner und Netze sind aber für den allgemeinen Fall gebaut, so daß ein großer Aufwand für wenig Nutzen entstehen würde. Oberstes Prinzip einer Rechnerkommunikation bleibt die Offenheit, d.h. die Verwendung offener Codes bei eindeutiger Zuordnung einer Codierung zu Bitgruppen (siehe Kapitel 4), sowie die Systematik, die weltweit kommuniziert werden kann.

Intensive Bemühungen seitens der Codierungstheorie sind für Übertragungsratenreduktion beispielsweise bei Bildkommunikation (verlustbehaftete Kompression bei hohen Datenraten) und bei allgemeiner Datenübertragung (verlustlose Kompression) sowie in der Verschlüsselung ein aktuelles Forschungsthema.

3.2 Zerlegung analoger Funktionen in Oberwellen

Um eine beliebige analoge Funktion für einen digitalen Rechner heutiger Bauart (auf elektrischer Basis) zugänglich zu machen, werden zwei grundsätzliche Funktionen benötigt:

- Die analoge Funktion muß als elektrische Signal zur Verfügung stehen oder mit Hilfe einer Wechselwirkung (Sensor) und einer eindeutigen Abbildung in dieses umsetzbar sein
- Die analoge Funktion muß digitalisierbar sein, d.h., die Wertemenge des Funktionsbereichs, die i.a. ein Intervall der reellen Zahlen darstellt, wird auf eine Menge von diskreten Werten abgebildet (Diskretisierung), die ihrerseits in Form von rechnerlesbaren Zeichen dargestellt werden (Codierung).

Umgekehrt werden bei Kommunikationen zwischen Rechnern digitale Signale mit Hilfe von analogen Funktionen übertragen. Die Sendezuordnung ist dabei eindeutig definierbar, bei der Übertragung jedoch kommt es zu Verlusten, so daß empfangsseitig die Zuordnung nicht mehr einfach möglich ist. Um diese Effekte transparent zu machen, wird die Sendefunktion als eindeutige Abbildung (Funktion im mathematischen Sinn)

$$f : T \rightarrow R$$

mit T als Intervall der reellen Zahlen (Zeitfunktion) aufgefaßt. Dieser Darstellung im Zeitbereich entspricht eineindeutig eine Darstellung im Frequenzbereich. Dies kann mit Hilfe der ***Fouriertransformation*** mathematisch formuliert werden.

Die Fouriertransformation basiert auf der Darstellung einer (stetigen, z.T. auch unstetigen mit endlichen Sprungstellen) Funktion durch die Überlagerung von orthogonalen Funktionen trigonometrischer Natur (Sinus- und Cosinus):

$$f(t) = \int_{\omega \in R+} (A_\omega \bullet \sin(\omega t) + B_\omega \bullet \cos(\omega t)) d\omega$$

Für periodische Funktionen wird das Integral zur (unendlichen) Summe über diskrete Frequenzen ω. Die Interpretation des erhaltenen Spektrums wird in diesem Fall besonders anschaulich: Es existiert (neben einem eventuellen Gleichspannungsanteil in B_0) eine unterste Frequenz mit einem Koeffizienten A_ω oder B_ω ungleich 0. Diese Frequenz wird ***Grundschwingung*** genannt, alle anderen sind ganzzahlige Vielfache dieser Grundfrequenz, sogenannte *Oberwellen*.

Die Bestimmung der Koeffizienten A_ω und B_ω wird Fourieranalyse genannt; dies geschieht im mathematischen Sinne durch Berechnung von bestimmten Integralen über die Taktperiode (bei unperiodischen Funktionen bis ins Unendliche), im algorithmischen Sinn z.B. durch die Fast-Fourier-Transformation für digitale Rechenanlagen. Das Ergebnis sind dabei zwei Amplitudenfunktionen A und B in Abhängigkeit von ω, oder – unter Definition einer Phasenbeziehung – eine Amplituden- und Phasenfunktion von ω. Diese Darstellung wird ***Frequenzspektrum*** genannt. Wird anstelle der Amplitude die Energie (proportional dem Amplitudenquadrat) über der Frequenz aufgetragen, spricht man von einem ***Frequenzenergiespektrum***.

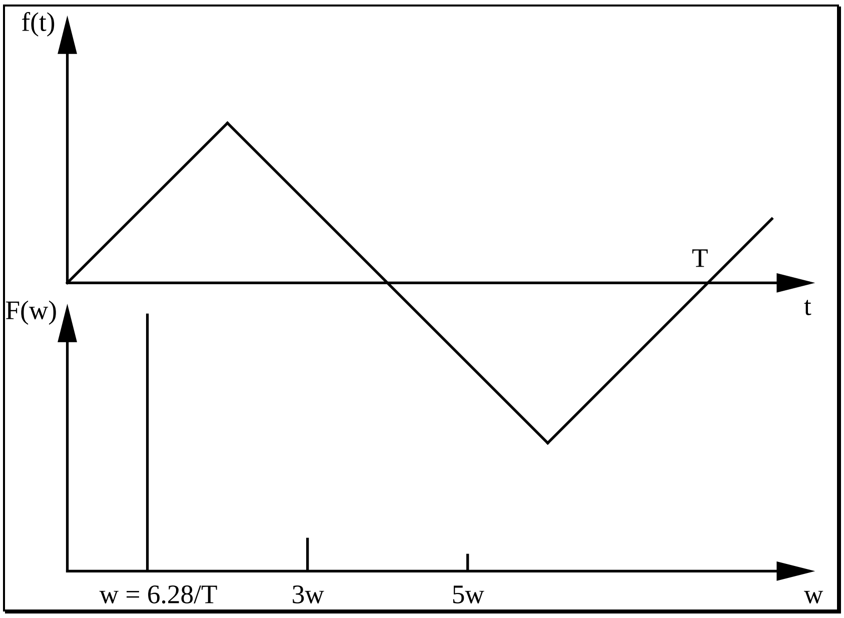

Abb. 3-1: Frequenzspektrum einer Dreiecksschwingung

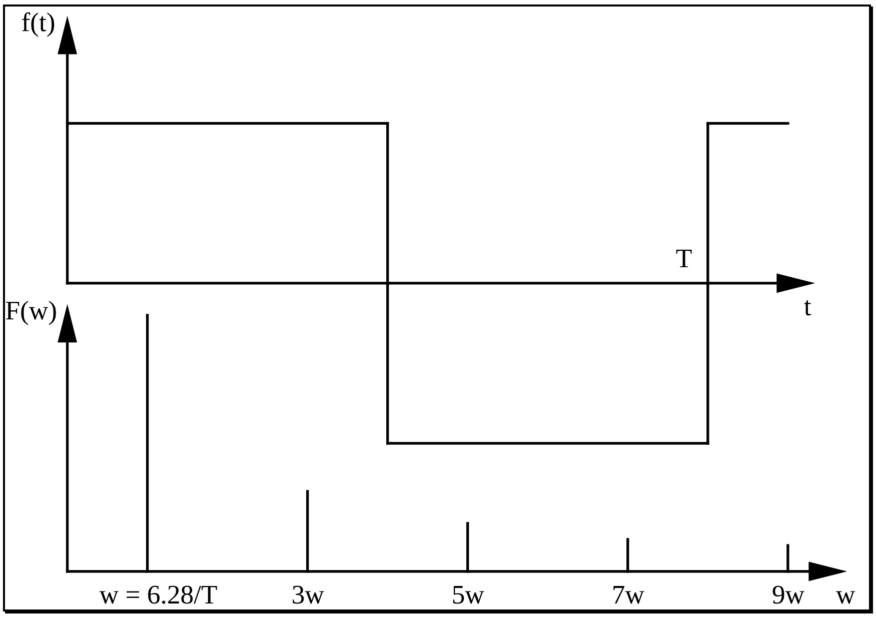

Abb. 3-2: Frequenzspektrum eines Rechtecks

Die übertragende Leitung bildet nun ein Signalverarbeitungssystem mit frequenzabhängigen Komponenten, d.h., die Übertragungsfähigkeit und damit die Dämpfung hängt von der Frequenz des Signals ab. Mit anderen Worten: Hochfrequente Teile des Signals werden ggf. gedämpfter übertragen als die niederfrequenten, das Signal wird verfälscht. Die Übertragungsstrecke hat modellhaft folgendes Ersatzschaltbild:

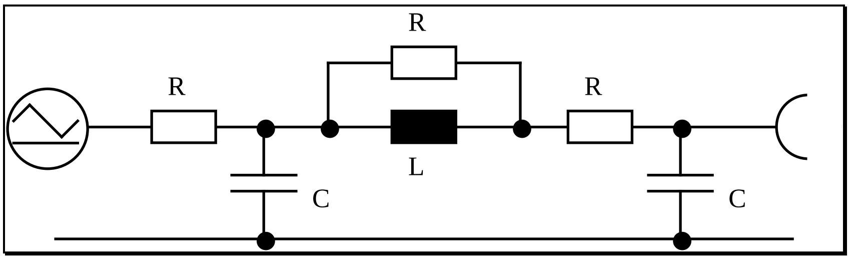

Abb. 3-3: Ersatzschaltbild einer Übertragungsstrecke

Zur Ermittlung einer optimalen Signalform müssen jedoch nicht nur Faktoren wie Signifikanz, Übertragungsfunktion und möglichst geringe Wechselwirkung durch gegenseitige Beeinflußung, sondern auch Störunanfälligkeit einbezogen werden.

3.3 Physikalische Grundlagen

3.3.1 Elektrische Systeme

Wie bereits dargestellt, besteht ein elektrisches Übertragungssystem aus aktiven Elementen (Spannungs- oder Stromquellen), die gesteuert werden und passiven Elementen wie Widerstände, Kondensatoren und Spulen; die modellhafte Vorstellung der reinen Übertragungsleitung geht dabei nur von R, C und L aus.

Die Übertragung eines einzelnen Signals, beispielsweise eines Bits, besteht aus elektrischer Sicht in der Umladung von Kapazitäten mit Hilfe von Strömen, die durch Widerstände und Spulen fließen. Bestimmend für die Umladezeit ist – bei Abwesenheit von Induktivitäten – die Zeitkonstante

$$\tau = R \bullet C.$$

Diese Werte können innerhalb der ICs mit R = 50 Ω .. 100 kΩ bei $C \leq 100$ pF angenommen werden, so daß mit τ = 5 ns .. 10 µs gerechnet werden kann; mit einer Umladezeit $\approx 2 \bullet \tau$ ergeben sich so maximale Übertragungsfrequenzen von 10 kHz und 100 Mhz, die durch Leitungseffekte weiter eingeschränkt werden.

Die einzelnen Kabelarten für eine elektrische Übertragung sind:

- Einfache Leitung (nebeneinander, unverdrillt)
- Verdrilltes Leitungspaar (twisted pair) mit oder ohne gemeinsame Abschirmung
- Flachbandkabel
- Mehraderkabel, z.B. mit gemeinsamer Abschirmung
- Koaxialkabel

Von diesen Kabelarten sind für hochfrequente Übertragungen oder weite Strecken insbesondere die Twisted Pair Verdrahtungen und die Koaxialkabel geeignet. Die zeichnen sich durch geringe Strahlung (und damit Übersprechen, insbesondere bei Koaxialkabeln) bei moderaten Kapazitäten und Widerständen aus; Twisted Pair ist insbesondere häufig vorhanden und damit benutzbar.

3.3.2 Lichtwellenleiter

Die (gebundene) Nachrichtenübertragung mittels (quasi-)optischer Wechselwirkung beruht auf dem Einsatz von Lichtwellenleitern, in denen sich die elektromagnetische Welle (Licht im sichtbaren oder nahen infraroten Bereich) nahezu ungedämpft ausbreiten kann. Da das Funktionsprinzip aller kommerziell eingesetzten Computer auf elektrischer Basis beruht, muß ein aufwendigeres Interfacing zwischen Rechner und Netz integriert werden. Dies besteht im Kern aus LEDs (Light Emitting Diodes) oder Halbleiterlasern sendeseitig sowie Phototransistoren oder -dioden empfangsseitig.

Das Prinzip der Lichtwellenleiter (LWL, fibre optics) beruht auf einer permanenten Brechung oder auf der Totalreflexion, je nach Ausführung des LWL. Der Brechungsindex, aufgetragen über den Querschnitt des LWL, ist für die jeweilige Form verantwortlich:

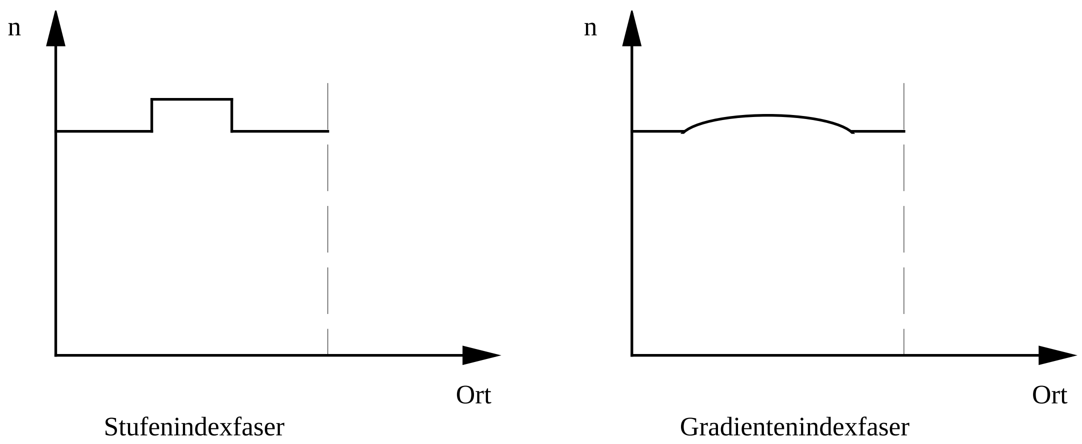

Abb. 3-4: Lichtwellenleiterarten

In der Stufenindexfaser werden Lichtstrahlen, die in einem bestimmten Winkel (Akzeptanzwinkel, Sinus des Winkel heißt Apertur), auf den LWL auftreffen, durch permanente Totalreflexion im Kernbereich des Leiters gehalten. Dieser Unterschied im Brechungsindex zwischen Kern- und Mantelbereich wird z.B. durch mechanisches Ziehen erhalten, ein relativ preiswertes Verfahren. Der Nachteil dieser Faser liegt in den unterschiedlichen optischen Wegen, die die einzelnen Strahlen durchlaufen müssen. Dadurch wird die maximale Bitfolgerate begrenzt.

Die Gradientenfaser bricht alle Lichtstrahlen innerhalb des Kernbereichs ständig zur Mitte hin; Hierdurch erfolgt die Bündelung, die Lichtstrahlenwege sind jedoch nicht so unterschiedlich, da in Bereichen niedrigeren Brechungsindex' die Lichtgeschwindigkeit größer ist: Der optische Weg bleibt überwiegend für alle Strahlen gleich. Gradientenfaser werden durch gezielte Dotierung erhalten.

LWLs zeigen dennoch Verluste durch Streuung unterschiedlicher Herkunft; diese Streuungen werden an Inhomogenitäten wie Kristalliten, Blasen, und durch grundsätzliche physikalische Prinzipien hervorgerufen und lassen sich auf etwa 0,3 dB/km begrenzen.

3.4 Übertragung von Informationen

Die Darstellung von Informationen in elektrischer (und optischer) Form kann auf mehrere Arten geschehen, die im folgenden zusammengefaßt sind.

Die Informationen können auf elektrischer Basis durch

- Spannungsbereiche,
- eingeprägte Ströme oder
- Modulierungen einer Grundschwingung

dargestellt werden. Im optischen Bereich wird zur Zeit lediglich eine Informationsdarstellung durch Impulsfolgen, die mit der Soannungs- oder Stromsteuerung vergleichbar sind, genutzt.

Die ***spannungsgesteuerten Schnittstellen***, beispielsweise RS-232 oder RS-422 (siehe Kapitel 5), definieren für '1' und '0' Spannungsbereiche, die signifikant voneinander unterschieden werden können. Aus dem empfangenen Signalwert kann dann der Zeichenwert abgeleitet werden.

Als Beispiel für eine ***stromgesteuerte Schnittstelle*** gilt die 20mA-Schleife. Hier wird der Zeichenwert '1' durch einen aufgeprägten Strom von 4 .. 20 mA übertragen, '0' entsprechend durch keinen Strom. Die Übertragung ist bereits frühzeitig definiert worden, kann nur mit geringen Baudraten angewendet werden, besitzt dann aber eine hohe Störfestigkeit insbesondere in industriellen Anwendungen!

Die frequenzabhängigen Schnittstellen werden im allgemeinen Modulationsverfahren genannt. Hier wird eine Grundschwingung, die durch Amplitude, Frequenz und Phasenlage bestimmt ist, zeichenabhängig so verändert, daß der Signalwert eindeutig in einen Zeichenwert zu interpretieren ist. Entsprechend den veränderlichen Werten werden dabei ***Amplitudenmodulation*** (AM), ***Frequenzmodulation*** (FM) und ***Phasenmodulation*** (PM) unterschieden.

Bevor analoge Eingangssignale in Form von digitalen Zeichenwerten übertragen werden können, muß eine Quantisierung und Codierung erfolgen - dies war bereits in Abschnitt 3.2 erwähnt worden. Diese ***AD-Wandlung*** unterliegt technischen Randbedingungen:

- Die Anzahl der Quantisierungsstufen muß festgelegt werden; dies wird üblicherweise in Form von 2er-Potenzen geschehen, so daß 256 Stufen eine Codierung in 8 Bits benötigen, diese aber vollständig ausnutzen.
- Die Abtastrate muß so festgelegt werden, daß die höchste darzustellende (und zu übertragende) Frequenz (siehe Frequenzanalyse) digitalisiert werden kann. Nach dem ***Nyquist***-Kriterium muß dazu die Abtastfrequenz mindestens um den Faktor 2 größer sein (besser wäre Faktor 5, man nutzt beim Telefonieren ca 2,5). Abtastrate * Anzahl der Codierbits ergibt die Nettobitrate für die Übertragung.
- Bei der AD-Wandlung sind zusätzliche Bedingungen zu beachten: Das analoge Signal darf sich während der Digitalisierung nicht ändern (Sample&Hold-Schaltung), und die höchste digitalisierbare Frequenz darf im Eingang nicht überschritten werden (Tiefpaßfilter), soll es nicht zu Fehlern kommen.

3.5 Informationsübertragungsmodell

Eine der zentralen Aufgaben für den Empfänger ist die Detektierung von Zustandsänderungen am Eingang, um ein Zeichen oder eine Zeichenkette zu erkennen. Diese Verfahren werden im nächsten Kapitel näher betrachtet.

Weiterhin muß der Empfänger erkennen können, ob das empfangene Zeichen für ihn bestimmt ist, oder ob es sich um eine Störung oder Falschsendung handelt:

In der ***asynchronen Übertragung*** wird ein Zeichen als Ganzes übertragen, eingeschlossen in Start- und Stoppbits und versehen mit einem entsprechenden Protokoll. Dies funktioniert nur mit Punkt-zu-Punkt-Verbindungen, da es zeichenbasiert ist. Die Asynchronität der Übertragung bezieht sich hierbei auf Layer 2 (Data Link); in Kapitel 4 werden synchrone und asynchrone Übertragungen im Detail dargestellt.

Die ***synchrone Übertragung*** nutzt spezielle Start- und Stoppzeichen und umgibt damit die Nutzinformation innerhalb dieser Blockung. Während also die einzelnen Zeichen in einer asynchronen oder synchronen Prozedur übertragbar bleiben, synchronisieren die Start- und Stoppzeichen die Nutzinformation für den Empfänger.

Bei der ***Paketübertragung*** werden zusätzliche Informationen wie Paketnummer, Empfänger usw. mit übertragen. Diese Übertragungsform wird beispielsweise genutzt, um größere Datenmengen zu strukturieren und durch die zusätzlichen Informationen anschließend wieder zu synchronisieren. Innerhalb der Pakete erfolgt wieder die Synchronisation durch die o.g. Verfahren.

3.6 Zur Ausnutzung von Kanälen

Eine weitere Frage, die in dem Betrieb einer Kommunikation durchaus sehr wichtig werden kann, besteht in der Ausnutzung der Kanalkapazität. Die theoretisch berechenbare oder technisch nutzbare Kapazität wird in der Regel nicht durch eine einzige Kommunikation ausgenutzt; liegen hier erhebliche Investitionen oder Betriebskosten vor, ist der Betreiber aus wirtschaftlichen Gründen geneigt, diese Kapazität besser zu nutzen, indem er die Ressourcen mit anderen teilt.

Die Mehrfachnutzung der Bandbreite eines Kommunikationskanals kann entweder durch ***Konzentrierung*** oder durch ***Multiplexen*** geschehen. Die ***Konzentrierung*** findet sich häufig in Paketübertragungsnetzen wie X.25 (Datex-P). Ein Kommunikationskanal bleibt ungenutzt, sofern keine zu übertragenden Zeichen vorliegen. Im Fall einer Datenübertragung wird das Paket mit entsprechender Adressierung versehen und erreicht so den Empfänger. Dieses Verfahren, auch in LANs (Ethernet, Token Ring) implementiert, eignet sich besonders für ungleichmäßige und unregelmäßige Kommunikationen.

Beim ***Multiplexen*** werden Teile der gesamten Kanalkapazität den einzelnen Kommunikationen zugeordnet. Dies kann durch Frequenzmultiplexen oder Zeitmultiplexen geschehen; ersteres ordnet verschiedene Frequenzbereiche den einzelnen Kanälen zu und wird mit modulierten Übertragungen angewendet.

Das *Zeitmultiplexen* kann durch die synchrone Zuteilung von Zeitschlitzen an die einzelnen Kanäle geschehen. Diese Technik wird insbesondere beim ISDN genutzt, wo die Zeitsynchronisation durch geeignete Steuerzeichen erfolgt. Asynchrone Zeitmultiplexverfahren teilen die Übertragungskapazität zwar in Zeitschlitze auf, ordnen diese jedoch nicht fest zu; leere Schlitze werden gekennzeichnet, und jeder Sender darf einen solchen Zeitschlitz füllen. Diese Technik ist hochmodern und wird beim ATM (Asynchronous Transfer Mode) im Breitband-ISDN genutzt (werden).

Kapitel 4: Leitungscodierung

Literatur: [1] Kowalk, W.P.; Burke, M.: Rechnernetze, Kapitel 4

Die Übertragung von Informationen erfolgt generell über Leitungen oder elektromagnetische Ausbreitung im Vakuum. Das physikalische Prinzip entstammt dabei meist der Elektrizität, in zunehmendem Maße auch der Optik; während elektrische Wechselwirkungen leitungsgebunden wie leitungslos genutzt werden können, beschränkt sich die optische Übertragung auf Lichtwellenleiter und nur in Ausnahmefällen (über kurze Distanzen) auf die Ausbreitung in Vakum/Luft.

Zur Nachrichtenübertragung muß der Sender die Zeichendarstellung der Nachricht in ein (oder eine Reihe von) physikalisches(n) Signal(en) umwandeln, welches der Empfänger als zugehöriges Zeichen interpretiert. Diese physikalische Darstellung wird als ***Leitungscodierung*** bezeichnet.

4.1 Grundsätzliches zur Codierungstechnik

Die Informationen in der Informatik können in der Regel als binäre digitale Werte (d.h. zweiwertig mit diskreten Wertebereichsdarstellungen) aufgefaßt werden; die Verfahren, die von einer Nachricht hierhin führen, sind hinlänglich bekannt, allerdings nicht Gegenstand dieser Vorlesung.

Die Darstellung erfolgt dann in sogenannten Bits (als Abkürzung für Binary Digit), wobei mehrere Bits zu einem **Byte**, in der Nachrichtentechnik häufig als **Oktett** bezeichnet (8 Bits), oder auch **Wörtern** mit z.B. 16 oder 32 Bit Länge zusammengefaßt werden können. Die Bits können die (logischen) Werte 0 und 1 annehmen und werden entsprechend bezeichnet (0-Bit, 1-Bit).

Bei einer Anzahl von n Bits kann diese Bitanhäufung 2^n-Werte annehmen und somit codieren. Dementsprechend sind in Bytes 256 Werte, in 16-Bit-Wörtern 65536 und in 32-Bit-Wörtern 4294967296 darstellbar; die übliche Codierung eines Alphabets kommt mit 7-Bit bzw. bei Sonderzeichen mit 8-Bit aus. Der z.T. nicht mehr existierende Telexdienst nutzte beispielsweise nur 5 Bits, um Zeit zu sparen, und hatte Umschaltzeichen für Buchstaben und Ziffern.

Reichen die codierbaren Zeichen nicht aus, um alle Sonderfälle wie beispielsweise eine Kommandoübergabe, die nicht als Nachricht zu interpretieren ist, übertragen zu können, werden gewöhnlich ***ESCAPE***-Zeichen (z.B. 0x1B als *Data Link Escape*) definiert, die das eine oder mehrere Zeichen als besonders zu interpretieren darstellen. Dieses Verfahren läßt sich besonders an Übertragungen zwischen Rechnern und Druckern beobachten, wo die ESCAPE-Sequenzen Befehle an den Drucker zur Formatierung codieren, während die reine Nachricht eben auszudrucken ist. Während die Codierung des Alphabets einigermaßen standardisiert ist (ASCII 7-Bit, die 8-Bit-Erweiterung ist mehrfach erfolgt), ist bei den ESCAPE-Sequenzen dies in weiter Ferne!

Kann eine Nachrichtenübermittlung jede beliebige Zeichenfolge übertragen (also ohne Beschränkung z.B. auf die darstellbaren Zeichen), spricht man von ***transparenter*** Datenübertragung.

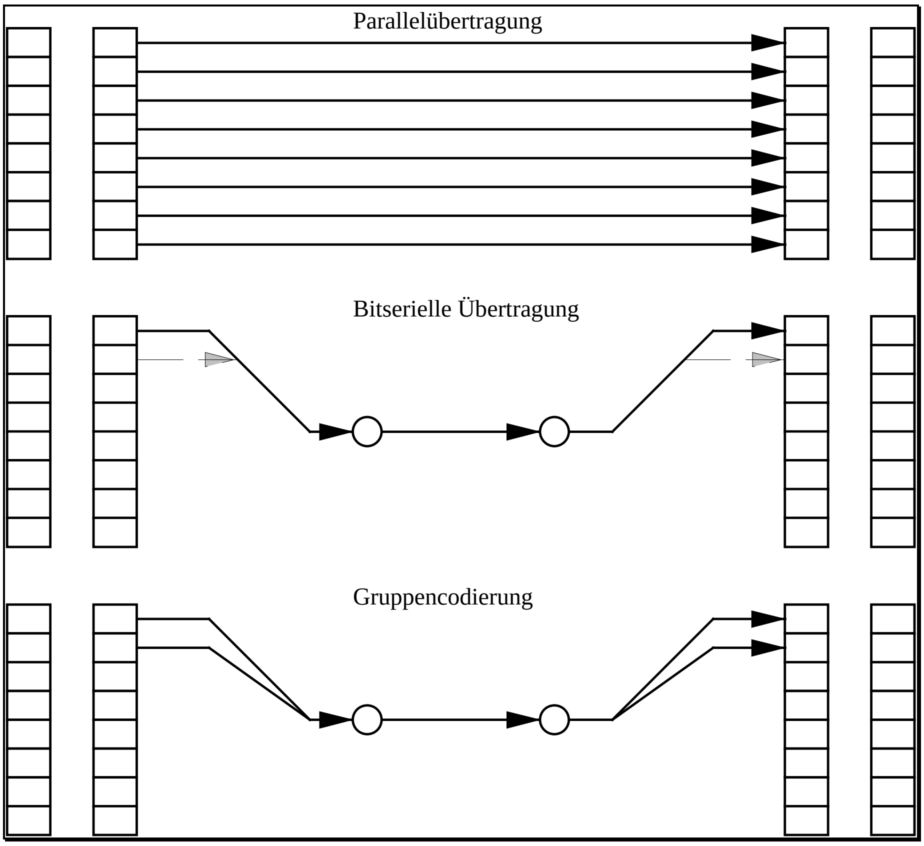

Abb. 4-1: Übertragungsverfahren

Zur Übertragung selbst müssen die Zeichen in geeignete Einheiten zerlegt werden (im Zweifelsfall in Bits, dann spricht man von ***bit-serieller*** Übertragung); diese Einheiten werden dann in physikalische Signale umgewandelt, am Empfänger rückgewandelt und wiederum zusammengesetzt. Bei Übertragung mehrerer Signalwerte gleichzeitig spricht man von ***paralleler*** Übertragung (Beispiel: IEEE-488, byte-seriell aber bit-parallel). Hierfür sind naturgemäß mehrere Leitungen notwendig, so daß durch diesen Aufwand bedingt die Übertragungsentfernung nur gering bleiben wird (ca 10 m). Wird jedoch mehr als ein Bit in einem Signalwert codiert, so nennt man dies ***Gruppencodierung*** (Beispiel: V.42-Norm für DFÜ).

4.2 Betriebsarten

Die reine Übertragung von Informationen von einem Ort zu einem anderen wird (unabhängig von Quittierungen) **Simplexbetrieb** (Richtungsbetrieb, Abkürzung sx) genannt. Dies kommt einem Verteildienst wie Rundfunksendungen gleich. Das Aussenden von einfachen Quittierungen kann dabei durch Rücksendesignale auf Bitbasis geschehen; komplexe Quittierungen sind bei Simplexbetrieb hiermit jedoch kaum hantierbar. Bei

Nachrichten mit sehr langer Laufzeit (Satelliten in geostationärer Erdumlaufbahn) wird auf derartige Nachrichten verzichtet.

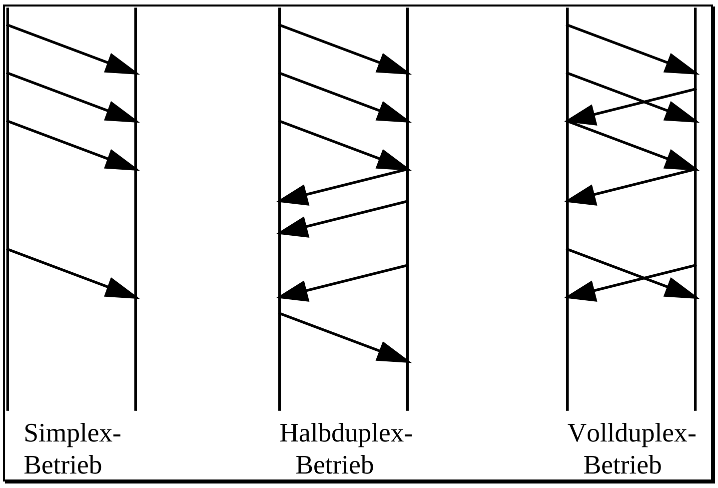

Abb. 4-2: Betriebsarten von Übertragungseinrichtungen

Läßt sich die Übertragungsrichtung in dem Sinne ändern, daß zu einem Zeitpunkt nur eine Richtung zulässig ist, im Prinzip aber beide Richtungen möglich sind, wird von ***Halbduplex*** (Wechselbetrieb, Abkürzung hx) gesprochen. Eine Wechselsprechanlage beispielsweise entspricht diesem Prinzip, auch in der Computerwelt wird in Netzen wie Ethernet (CSMA/CD-Verfahren, siehe Kapitel 7) diese Betriebsart genutzt.

Der ***Vollduplexbetrieb*** (Gegenbetrieb, dx) wird durch die Sendemöglichkeit beider Stationen zur gleichen Zeit charakterisiert. Dies wird z.B. durch verschiedene physikalische Signalträger (Leitungen) oder Übertragungstrennung mittels verschiedener Modulationen erreicht und ermöglicht den kontinuierlichen Datenaustausch einschließlich echtzeitfähiger Quittierungen und Fehlermeldungen.

4.3 Übertragungsprozeduren

Während nunmehr die Aussendung von Daten bereits teilweise geklärt ist (es gibt noch eine Vielzahl von Fragen, die die nächsten Abschnitte wie Kapitel füllen werden), muß noch größere Sorge für die Synchronisierung von Empfänger und Sender getragen werden. Dies liegt insbesondere daran, daß dem Empfänger die Sendezeitpunkte nicht bekannt sind (von wenigen, sehr extremen Ausnahmen einmal abgesehen). Der Empfänger muß daher ständig (im wahrsten Sinne des Wortes) auf seiner Empfangsleitung mithören, sprich den Pegel abtasten, um Änderungen detektieren und darauf reagieren zu können. Dies liefert ihm zwei Informationen: Eine Sendung beginnt (Startzeichen), und er bekommt die übermittelte Information.

Grundsätzlich kann zunächst kein Empfänger zu jedem Zeitpunkt an der Leitung mithören; die Gründe zum Ausfall sind z.T. trivial, da das Gerät beispielsweise ausgeschaltet ist, die Energieversorgung unterbrochen wurde, ein Kabelbruch vorliegt oder aber der Empfänger von der Informationsflut überrollt wurde. Aus diesem Grund werden ***Synchronisationszeichen*** ausgetauscht, um den notwendigen Gleichlauf herzustellen. Diese Synchronisationszeichen enthalten keine Fehlerkennungen o.ä., sondern stellen lediglich bei beiden Teilnehmern den gleichen Zustand her, um anschließend auf gesicherter Basis weiter miteinander kommunizieren zu können.

Die Art der Synchronisation unterscheidet die verschiedenen Verfahren zur Informationsübertragung. Die Normung findet im ISO/OSI-Basisreferenzmodell im Rahmen der Bitübertragungs- und Sicherungsschicht statt, wo von ***Übertragungsprozeduren*** gesprochen wird. Es werden synchrone und asynchrone Übertragungsprozeduren unterschieden, wobei erstere nochmals in bit- und zeichensynchrone Verfahren unterteilt ist.

4.3.1 Synchrone und asynchrone Übertragung in der Bitübertragungsschicht

Die ***synchrone Übertragungstechnik*** (bezogen auf die Bitübertragungsschicht, Physical Layer) garantiert einen Gleichlauf zwischen Sender und Empfänger über praktisch beliebig lange Zeiten. Dies kann in Ausnahmefällen über hochgenaue Uhren geschehen oder durch eine ständige ***Synchronisierung***. Diese Synchronisierung besteht dabei aus der Definition eines Synchronisationsmasters (fast immer der Sender, ggf. eine zentrale Taktfunktion), der dem anderen Partner mitteilt, wann das nächste Bit (bzw. Zeichen bei Bitparallelität) gesendet wird. Die Aussendung kann dabei über eine eigenständige Leitung oder Taktrückgewinnung aus dem im physikalischen Signal codierten Takt geschehen.

Die Konsequenzen eines solchen Verfahrens sind:

- Lange Laufzeiten ohne aufwendige Resynchronisationsverfahren auf Blockbasis sind möglich
- Die Geschwindigkeit läßt sich gegenüber asynchronen Verfahren wesentlich erhöhen (häufig sind Interfacebausteine für Taktverhältnisse 16 : 1 (synchron : asynchron) ausgelegt).

Asynchrone Übertragungsprozeduren sind grundsätzlich zeichenorientiert, d.h. eine bestimmte Anzahl von Bits (5 - 8) werden als Basiseinheit übertragen. Zwischen Sender und Empfänger muß eine Synchronisation auf den Start gegeben werden, da dieser zu einem beliebigen Zeitpunkt stattfinden kann, die anschließenden Bits werden nicht synchronisiert, d.h. der Zeitpunkt der Bitabtastung muß generell vereinbart werden (Bitfolgerate).

Die Synchronisierung geschieht durch Definition eines Ruhepegels und Ausgabe eines Startbits, das von diesem Ruhepegel abweicht. Der Empfänger reagiert auf die Flanke, die durch diesen Wechsel entsteht. Im einzelnen gehen Sender/Empfänger wie folgt vor:

1. Der Sender hält auf der Leitung den sogenannten Ruhepegel (Signalwert 1), der vorteilhafterweise von einem Wert ohne Kabelanschluß abweicht, um ggf. Kabelbrüche detektieren zu können.

2. Soll ein Zeichengesendet werden, dann geschieht dies durch das sogenannte Startbit, das von dem Ruhewert abweicht (Signalwert 0). Dieses Startbit kann in seiner Länge von den Nutzdatenbits abweichen, muß aber nicht. Der Empfänger muß die Datenleitung ständig abhören, um das Startbit zu detektieren, ohne er reagiert auf die (negative) Flanke.
3. Der Sender gibt die vorher vereinbarte Zahl von Nutzdatenbits in seinem lokalen Takt auf der Datenleitung aus; der Empfänger leitet aus der Detektierung des Startbits und der vereinbarten Bitfolgerate die Zeitpunkte der Datenbitabfragen her und übernimmt den Bitwert der Datenleitung zu diesen Zeitpunkten.
4. Den Datenbits kann ein zusätzliches Paritätsbit folgen (nicht in Abb. 4-3).
5. Der Abschluß wird durch ein Stoppbit signalisiert, das den gleichen Wert wie der Ruhepegel hat und somit vom Startbit zu unterscheiden ist. Dieses Stoppbit kann länger als die Datenbits sein (üblich sind 1, 1.5 oder 2 Bits), unmittelbar folgend kann das nächste Startbit gesendet werden.

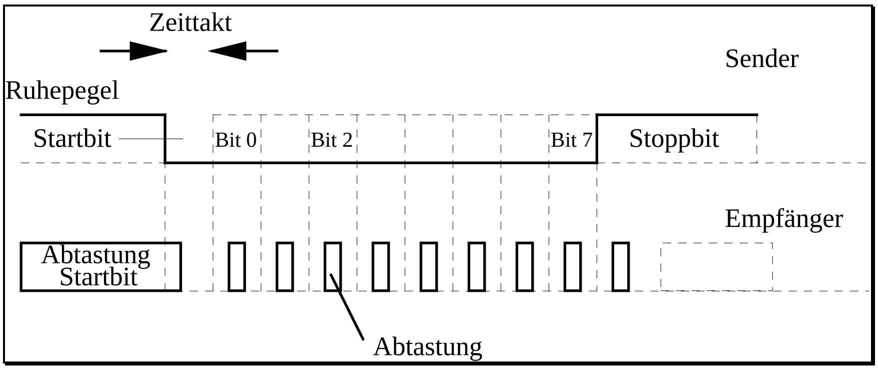

Abb. 4-3: Asynchrone Übertragung

Für eine korrekte Übertragung müssen demnach die verschiedenen Pegel, die Länge des Startbits, die Bitfolgerate, die Anzahl der Datenbits, eventuelle Paritätbits und die Mindestlänge des Stoppbits vereinbart werden. Die Asynchronität bezieht sich hiermit auf den Abstand zweier beliebiger Zeichen, während die Bits sehr synchron (aber nicht synchronisiert) übertragen werden.

4.3.2 Synchrone Übertragungsprozeduren in der Sicherungsschicht

Eine weitere Synchronisierung besteht in der Aufteilung der zu übertragenden Nachricht in ***Übertragungsblöcke*** (transmission blocks). Ein solcher Block kann in der Regel jederzeit (also asynchron) gesendet werden und wird zur Kenntlichmachung durch ***Steuerzeichen*** (control character) eingeleitet bzw. begrenzt.

Zur Übertragung werden zunächst einige (2 bis 8) SYN-Zeichen auf die Leitung gelegt; diese dienen dem Empfänger zur eigentlichen (Block-)Synchronisierung, während die Bit-Synchronisierung weiterhin auf der Übertragung des Takts basiert. Der eigentliche Block beginnt mit dem ***STX***-Zeichen (Start-of-Text, ASCII-Wert 2), es folgen die Daten (in dem jeweiligen Format) und am Ende das ***ETX***-Zeichen (End-of-Text, ASCII-Wert 3). Anschließend werden in der Regel Prüfsummenbytes geschickt, die aus den übertragenen Zeichen berechenbar sind und der Fehlerkontrolle dienen (***BCC***, Block Check Character, i.d.R. 2-4 Bytes).

....	SYN	SYN	SYN	STX	Daten	....	ETX	BCC	BCC

Abb. 4-4: Aufbau eines Übertragungsblocks

Die Blöcke sind meist in der Länge begrenzt, häufig wird die Länge exakt festgelegt oder mit den ersten Zeichen übertragen. Die Begrenzung hat pragmatische Gründe (Blöcke müssen beim Empfänger zwischengespeichert werden), technische (der Gleichlauf zwischen Sender und Empfänger kann mit Sicherheit nur eine gewisse Zeit aufrechterhalten werden) sowie theoretische Gründe (optimale Länge für Fehlersicherung durch eine bestimmte Anzahl von Fehlerzeichen). Um mit dieser Begrenzung Informationen beliebiger Länge übertragen zu können, muß die Nachricht ggf. in mehrere Blöcke unterteilt werden. Man spricht dann von ***Fragmentierung*** sowie ***Füllung*** bei zu kleinen Blöcken. Das ETX-Zeichen wird bei Teilblöcken, die nicht der letzte Block sind, in diesem Fall gewöhnlich durch das ETB-Zeichen ersetzt.

In zeichenorientierten Übertragungen kann die Übertragung von Zeichen (beispielsweise aus Grafiken), die Steuerzeichen entsprechen, wiederum durch Escape-Zeichen, die vor das zu übertragende Zeichen gestellt werden, erreicht werden. Ein solches Escape-Zeichen (DLE, ASCII-Wert 27) verhindert die Interpretation des nachfolgenden als Steuerzeichen und wird beim Empfänger einfach wieder entfernt. Diese Technik wird ***Codeerweiterungstechnik*** (character stuffing) genannt.

Bit-orientierte Übertragungen, die also keine Blockung einer festgelegten Bitanzahl zu Zeichen kennen und somit mit einer beliebigen Anzahl (und Folge) von Bits ablaufen, wird mit einer anderen Technik gearbeitet, dem ***Bitstopfen*** (bit stuffing). Auf eine Folge von 5 Bits (beispielsweise so im HDLC-Protokoll (high level data link Protocol) implementiert) von Nutzdaten, die alle den Wert ‘1’ besitzen, folgt eine zusätzlich ‘eingemischte’ ‘0’, die beim Empfänger wiederum herausgefiltert wird. Wird diese Folge nun verletzt, indem eine Folge von 6 ‘1’-Bits gesendet wurde, handelt es sich um ein Steuerzeichen (z.B. flag ‘01111110’), das entsprechend zu interpretieren ist.

	Synchron		Asynchron	
Synchronisierung durch	Takt		Start/Stoppbit	Layer 1
Minimale Länge	1 Bit		1 Zeichen	
	Bit-synchron	Zeichen-synchron		
Synchronisierung durch	Bitfolgenverletzung	Spezielle Zeichen		Layer 2
Längeneinheit	Bit	Zeichen		

Tab. 4-1: Übertragungsprozeduren Layer 1 und 2

Tabelle 4-1 zeigt in der Zusammenfassung die verschiedenen Übertragunsprozeduren.

4.4 Übertragungscodes

Die Darstellung der Daten wurde bereits als ***Zeichenwerte*** bezeichnet. Für ein Bit werden üblicherweise die Zeichenwerte ‘0’ und ‘1’ als annehmbare Werte definiert, die ihrerseits zur Speicherung oder Datenübertragung in physikalischen Signalen, den *Signalwerten* dargestellt werden. Die Wechselwirkung und die Form der Darstellung sind dabei für die Logik sehr nebensächlich, solange die Interpretation eindeutig ist. Für einfache, asynchrone Übertragungsverfahren wie RS232 (jetzige Version : RS232E, auch mit V.24/V.28 bezeichnet) werden Spannungsbereiche zur Darstellung definiert: -3 bis -15 V bedeutet MARK, eine ‘1’, +3 bis +15V SPACE, eine ‘0’. Für die wesentlich sicherere Übertragung nach RS422/485 werden Differenzpegel zwischen zwei Leitungen für 1 und 0 angegeben. Die Abbildung zwischen Zeichen- und Signalwert wird als ***Code*** bezeichnet.

Die Abbildung zwischen Signal- und Zeichenwert wird im allgemeinen (aber nicht immer!) als bijektiv (eineindeutig) gefordert, um eine sofortige Interpretation ohne Kenntnis etwaiger Randbedingungen (wie Vorgeschichte) durchführen zu können. Im folgenden werden einige Codierungen betrachtet:

4.4.1 Eigenschaften von Übertragungscodes

Die Eigenschaften der einzelnen Übertragungscodes können nach ihren verschiedenen Eigenschaften klassifiziert werden, zu denen sowohl elektrische wie informationstechnische gehören:

- Gleichstromanteil, Störabstand
- Ausnutzung der Kanalkapazität
- Anzahl der verschiedenen physikalischen Signalwerte
- Fähigkeit zur Taktrückgewinnung
- Anzahl gemeinsam codierter Zeichen
- Fähigkeit zur Resynchronisation
- Redundanz

Die elektrischen Werte, die bei anderen Signaldarstellungen wie Optik beispielsweise entfallen können, beinhalten (exemplarisch) den ***Gleichspannungsanteil*** und den ***Störabstand***. Der Gleichspannungsanteil eines Codes soll im Weitverkehr möglichst Null sein, um Übertrager einsetzen zu können. Diese einfache Forderung führt jedoch sofort zur Unterscheidung zwischen Daten- und Telekommunikationswelten und entsprechenden Interfaces zwischen beiden. Der Störabstand zwischen den einzelnen physikalischen Werten soll möglichst groß sein, um eine eindeutige Unterscheidung auch bei überlagerten Störungen zu ermöglichen; andererseits führen große Spannungsabstände zwischen den zulässigen Signalwerten zu langen ‘Umladezeiten’ und damit zu verringerten Übertragungsraten.

Die ***Kanalkapazität*** wird sehr häufig ebenfalls durch elektrische Größen beeinflußt; wichtig sind die maximale und minimale zu übertragende Frequenz bei gegebener Dämpfung (oder umgekehrt), wobei mit ***Dämpfung*** die Schwächung des Signalpegels gegenüber dem Nullwert oder dem Rauschen gemeint ist: Unterhalb einer Grenze kann nicht mehr detektiert werden, ob es sich hier um ein gewolltes Signal handelt oder nicht. Die Angabe der Kanalkapazität bezieht sich dabei auf die komplette Strecke, Übertragungsweg wie Sender/ Empfänger.

Die Anzahl der physikalischen Signalwerte kann in Zusammenhang mit dem Gleichspannungsanteil (möglichst 0) und weiteren Effekten wie Echounterdrückung gesehen werden. Das hier behandelte Grundprinzip der digitalen (digitus lat. = Finger) Zeichen/Signalwertzuordnung bleibt hiervon unberührt, lediglich die Zweiwertigkeit kann auf physikalischer Seite verloren gehen. Ein Beispiel hierfür ist der quasi-ternäre Code, der für 0 den Spannungswert 0, für 1 alternierend + und - Spannung bietet. Eine echte Mehrwertigkeit von Signalen ist ebenfalls möglich, führt aber zu geringen Signalstörabständen und kann sehr aufwendig sein.

Die ***Taktrückgewinnung*** erlaubt durch Codierung des Sendetakts in die Signalwerte die Synchronisation auf Bitbasis. Ist es auch notwendig, eine weitere Synchronisation auf Blockbasis einzuführen, geschieht dies durch spezielle Zeichen (***Rahmenbildung***), beispielsweise Synchronisationszeichen oder auch die bewußte Verletzung einer Codierung.

Wird mehr als ein Zeichen in einem Signalwert codiert, so wurde dies als ***Gruppencodierung*** bezeichnet. Anwendungen hierfür sind im ISDN und FDDI zu finden, wobei beachtet werden sollte, daß bei einer Gruppencodierung nur ganzzahlige Vielfache einer Gruppe codierbar sind.

Die Darstellung einer Gruppencodierung geschieht häufig durch Angabe der Eingangs- und Ausgangsanzahl der Signale sowie ihrer Wertigkeit. Die abkürzenden Buchstaben dafür sind B (binär), T (ternär) und Q (quartär), die Anzahl der Signal wird durch Ziffern angegeben: 4B2Q wäre die Darstellung von 4 binären Zeichenwerten durch 2 quartäre Signalwerte.

Die ***Resynchronisation*** geschieht zumeist auf Rahmenbasis und dient unter anderem der Fehlerbegrenzung beispielweise durch einen ‘Seiteneinstieg’ des Empfängers.

Redundante Codes dienen der Fehlererkennung. Je nach Aufwand für die Redundanz (die zusätzlichen Informationen müssen ja mit übertragen werden, führen also zu einer Minderung der Nettoübertragungsrate) kann man hierbei zwischen erkennden und korrigierenden Codes unterscheiden; bei erkennenden Codes, die zumeist angewendet werden, wird bei Fehlern erneut übertragen.
Im folgenden werden einige Codierung besprochen, wobei die elektrischen Eigenschaften im wesentlichen nicht angesprochen werden, sondern mehr die informationstechnischen.

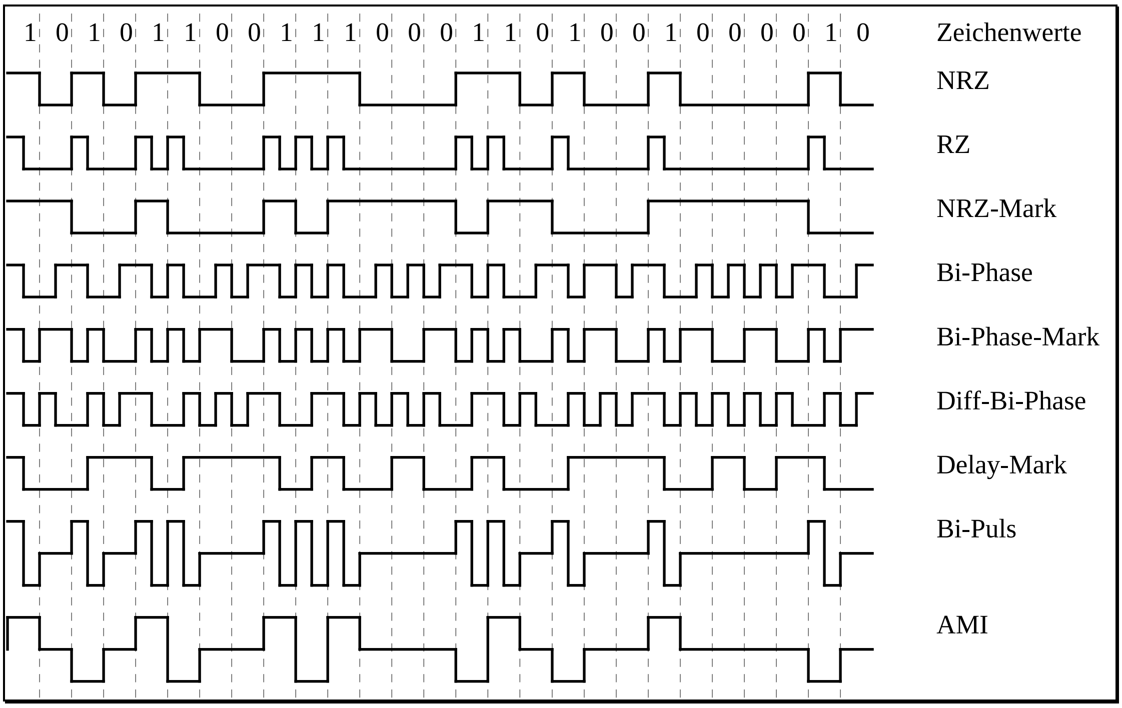

Abb. 4-5: Signalwerte verschiedener Codes

Für die Darstellung im folgenden wird für die positive Spannung das '+'-Zeichen, für die negative das '-'-Zeichen und für den Nullwert des Signal ein '=' vereinbart.

4.4.2 Non-Return-to-Zero (NRZ)

Diese einfachste Form der Codierung weist jedem Bit eindeutig einen Signalpegel zu; dies kann z.B. durch Zuweisung der '='-Spannung an '0' sowie der +-Spannung an '1' geschehen, wobei meistens nicht bestimmte Werte, sondern Spannungsbereiche mit einer Trennung voneinander zugewiesen werden.

NRZ-Codes gestatten keine Taktrückgewinnung, die eine minimale Folge von Taktflanken nicht gewährt werden kann. Der Gleichstromanteil kann u.U. sehr stark sein. Fehler wirken sich nur auf einzelne Bits aus.

4.4.3 Return-to-Zero (RZ)

In dieser Form der Codierung wird eine 1 als Folge der Signalzustände '+' und '=' für jeweils eine halbe Taktperiode, eine 0 lediglich als '=' dargestellt. Der Gleichstromanteil ist dadurch nur halb so groß wie bei NRZ (und gleicher Bitanzahl), dafür jedoch die maximal zu übertragende Frequenz doppelt so hoch. Eine Taktrückgewinnung ist hier ebenfalls nicht möglich, da bei längeren 0-Folgen keinerlei Wechselsignal zur Verfügung steht.

4.4.4 Non-Return-to-Zero-Mark (NRZ-Mark)

Die einzelnen Bits werden nicht mehr durch Signale, sondern durch Signalwechsel dargestellt, wobei eine ‘1’ einen Wechsel zum jeweils anderen Signalzustand (‘+’ oder ‘=’), eine 0 jedoch keinen Wechsel vollzieht. Durch diese Codierung entsteht der Eindruck eines sehr ruhigen Codes, er entspricht im Frequenzverhalten dem NRZ-Code, auch beim Gleichstromanteil kann dieser beliebig groß werden. Eine Taktrückgewinnung ist ebenfalls nicht möglich.

4.4.5 Bi-Phase

Bi-Phase stellt jedes Bit in einem Wechselsignal dar: Eine +/=-Signalfolge stellt eine ‘1’, eine =/+-Signalfolge entsprechend eine ‘0’ dar. Dies bedeutet, daß die Frequenz sehr hoch ist, jedoch jederzeit ein Takt zurückgewonnen werden kann (PLL-Schaltung). Die Unterscheidung zwischen ‘1’ und ‘0’ kann allerdings nur dann mit Sicherheit getroffen werden, wenn eine 1-0-Kombination auftritt (größerer Schaltungsaufwand). Der Gleichspannungsanteil beträgt die Hälfte der ‘+’-Spannung.

4.4.6 Bi-Phase-Mark (Manchester)

Am Anfang einer Taktperiode wird in jedem Fall ein Pegelwechsel vollzogen, in der Taktmitte jedoch nur bei ‘1’-Bits. Dieses Verfahren sichert ebenfalls die Taktrückgewinnung, liefert sichere Signalstartzeitpunkte für die 0 und darüber für die 1 und ist besonders für die magnetische Aufzeichnung (Harddisk) geeignet.

4.4.7 Differential Bi-Phase

Bei der Differential-Bi-Phase wird in jedem Fall ein Signalwechsel in der Mitte des Takts vollzogen, am Anfang jedoch nur bei einer ‘0’. Die Eigenschaften unterscheiden sich damit nicht von denen der Bi-Phase-Mark-Codierung.

4.4.8 Delay-Mark

In der Mitte einer ‘1’ wird ein Signalwechsel vollzogen, am Anfang aller Bitzustände jedoch fortgelassen; dies würde zu einem ähnlichen Code wie NRZ führen; um zusätzlich eine Taktrückgewinnung zu ermöglichen, wird bei Folgen von ‘0’ mit mehr als 2 Bits bei jedem eingeschlossenen ‘0’-Bit am Taktanfang ein Wechsel eingesetzt.

Die Folge dieser Codierung ist ein recht ‘ruhiger’ Code, d.h. es sind keine besonders hochfrequenten Anteile enthalten, wobei trotzdem eine Taktrückgewinnung möglich ist. Der Nachteil liegt in der zusätzlichen Auswertung der Phasenlage eines Wechsels, wobei eine eindeutige Phasenlage nur an ‘101’ oder ‘1001’-Bitfolgen auszumachen ist. Dadurch erhöht sich der Aufwand im Empfänger, und das Verfahren ist fehleranfällig.

4.4.9 Bi-Puls (Dipuls)

Die Einführung eines quasi-ternären Signals, das durch seine Definition keinen Gleichspannungsanteil mehr enthält, führt zu weiteren Codierungen, wobei bei der Bi-Puls-Codierung für eine '1' ein +/- -Impuls, für eine '0' ein '='-Signal gesendet wird. Hierdurch ist einerseits die maximale Frequenz recht hoch, der Takt aber nicht rückgewinnbar. Die Bedeutung dieser Codierung ist entsprechend sehr gering.

4.4.10 Alternating Mark Insertion (AMI)

Die einfache AMI-Codierung entspricht der NRZ-Codierung, wobei jedoch alternierend eine '1' durch das '+' oder '-'-Signal dargestellt wird, die '0' wie bisher durch das '='-Signal. Die Folge dieses quasi-ternären Codes ist kein Gleichspannungsanteil, allerdings auch keine Taktrückgewinnung.

In einer speziellen Ausführung mit Einfügung einer Gruppencodierung und teilweise Verletzung der einfachen Inversionsregeln wurde AMI jedoch als G.703 durch die CCITT zur Standardcodierung für die 2-MBit/s-S0-Kanäle im ISDN. Diese Codierung wird in den Abschnitten 4.5.3 und 4.5.4 behandelt.

4.4.11 Wertung der Codierungsformen

Viele der Codes, die in diesem Subkapitel behandelt wurden, sind mehr von theoretischem Interesse oder gelten als Grundlage zur Weiterentwicklung. Im täglichen Gebrauch sind besonders die NRZ-Codierung (RS232E), die Manchester-Codierung (Bi-Phase-Mark für magnetische Aufzeichnung) und AMI als G.703 zu finden.

4.5 Gruppencodierung

Unter Gruppencodierung wird die Abbildung einer Menge von Zeichen auf eine Menge von Signalen verstanden, wobei die Elemente der Signalmenge ihrerseits aus mehr als einem Signalzustand bestehen. Die Abbildung ist hierbei im Sinne von Relationen zu sehen, d.h. es gibt Dupel von Zeichen/Signalgruppen, die erlaubt sind, wobei einem Zeichen durchaus mehrere Signalgruppen, umgekehrt jedoch jedem Signal exakt ein Zeichen zugeordnet wird. Die Codierung kann in deterministischen endlichen Automaten dargestellt werden; da die Sendecodierung offenbar zu mehreren Signalgruppen führen kann, die Empfangsdecodierung jedoch zu eindeutigen Zeichen, ist der Empfangsautomat in der Regel einfacher ausgeführt.

Die Darstellung der Codierung geschieht durch Benennung der Anzahl der Eingangswerte (und deren Wertigkeit, z.B. B für binär) sowie der Ausgangswerte. Diese Benennung ist natürlich nur teilqualifizierend, so daß eine genaue Angabe der Codierung in Form von Abbildungsvorschriften unerläßlich ist.

4.5.1 Paired Selected Ternary (PST, 2B2T)

Bei PST wird jeweils einer Gruppe von 2 Binärwerten eine Gruppe von 2 Ternärwerten zugeordnet: Es liegt also ein 2B2T-Code vor. Dies bedeutet die Zuordnung von 4 Eingangswerten auf maximal 9 Signalgruppen, von denen nur 6 ausgewählt werden. Die Zuordnungstabelle hat folgende Gestalt:

Eingangssignalgruppe	Gleichspannungsanteil < 0	Gleichspannungsanteil > 0
00	-+	-+
01	=+	=-
10	+=	-=
11	+-	+-

Tabelle 4-2: 2B2T-Codierung PST

Die Form der Codierung (am Sender!) unterscheidet also zwischen einem Gleichspannungsanteil < 0 und >0, so daß entsprechende Codes ausgewählt werden können, um den Gleichspannungsanteil nahezu bei 0 zu halten. Die Detektierung des jeweiligen Gleichspannungsanteils kann durch einfaches Mitzählen der ‘01’- und ‘10’-Eingangsgruppen bei den jeweiligen Zuständen.

4.5.2 4 Binary, 3 Ternary (4B3T)

Die 4B3T-Codierung, der häufig bei digitalen Fernmeldeanlagen eingesetzt wird, werden insgesamt 16 Eingangsgruppen (4B) auf 27 mögliche Ausgangswerte (3T) umgesetzt. Die dabei entstehenden redundaten Signalgruppen werden wiederum für den Ausgleich des Gleichspannungsanteils zu 0 genutzt, können aber auch bestimmte Fehlerinformationen enthalten. Der 4B3T-Code wird gemäß Tabelle 4-3 codiert.

Die Vorteile einer solchen Codierung liegen im übrigen nicht nur in der Reduzierung des Gleichspannungsanteils auf 0 und der Möglichkeit zur Taktrückgewinnung, sondern auch in der Reduzierung der maximalen Übertragungsrate um 25% (3 Zustände anstatt 4). Weitere Vorteile liegen in speziellen adaptiven Filteralgorithmen wie Echounterdrückung, die ebenfalls auf dem 3T-Code rechnen können. Für die Datenwelt mit ihrer Zweiwertigkeit ergeben sich aber Gewöhnungsprobleme, da beispielweise ICs, die die Codierung/Decodierung übernehmen sollen, dreiwertig ausgelegt sein müssen!

Eingangscode	Gleichsp. < 0	Gleichsp. > 0	Eingangscode	Gleichsp. < 0	Gleichsp. > 0
0000	=-+	=-+	1000	=+-	=+-
0001	-+=	-+=	1001	+-=	+-=
0010	-=+	-=+	1010	+=-	+=-
0011	+-+	-+-	1011	+==	-==
0100	=++	=--	1100	+=+	-=-
0101	=+=	=-=	1101	++=	--=
0110	==+	==-	1110	++-	--+
0111	-++	+--	1111	+++	---

Tabelle 4-3: 4B3T-Codierung

Auch beim 4B3T-Code kann der momentane Gleichspannungsanteil durch Mitzählen der ‘ungleichgewichtigen’ Signalzustände mit entsprechender Wertigkeit (0 bis 3) bestimmt werden.

4.5.3 AMI-Codierung

Die im Abschnitt 4.4.10 vorgestellte AMI-Codierung erfüllt bereits die Bedingung, im Durchschnitt keinen Gleichspannungsanteil zu liefern und somit für Übertragersysteme geeignet zu sein. Die zweite wichtige Forderung der Taktrückgewinnung kann aber mit diesem einfachen Verfahren nicht erfüllt werden. Hierzu wird ein ***Compatible High Density Binary*** genannter Code (CHDBn) eingeführt, der nach einer Folge von n (>1) ‘0’-Bits die n+1-’0’ als ‘1’ mit verkehrtem, d.h. nicht alterniertem Signalpegel aussendet. Der Code wird dabei recht komplex, da z.B. für n = 3 eine Wertetabelle mit Gruppen zu 5 Bits und zusätzlichem Zustand (der vorigen ‘1’-Codierung) auszufüllen ist. Aus diesem Grund sei es bei der verbalen Beschreibung belassen.

Erschwerend kommt beim CHDBn-Code hinzu, daß eine Folge von sehr vielen ‘0’-Bits der Gleichspannungsanteil wieder (zeitlich befristet, aber in der Übertragung sehr wohl bemerkbar) sehr groß werden kann, da die alternierende Regel ständig verletzt wird. Daher wird, falls eine mehrfache einfache Verletzung vorliegen würde, anstelle der einfachen Codierung des vierten Bits (bei CHDB3) die gesamte Gruppe der 4 ‘0’-Bits als ‘+==+’ oder ‘-==-’ codiert, ansonsten alternierend zu den ‘1’-Codierungen. Die Eindeutigkeit der Zuordnung ist dabei durch die Verletzung der Alternierungsregel nicht nach n, sondern nach n-1-Bits gegeben.

Folgendes Beispiel möge eine gewisse Klarheit zur Codierung geben:

Zeichen	0101	1100	0010	1101	0000	0000	1000	0000	1011	0101
AMI	=+=-	+-==	==+=	-+=-	====	====	+===	====	-=+-	=+=-
CHDB3	=+=-	+-==	=-+=	-+=-	===-	===-	+===	+===	-=+-	=+=-
	=+=-	+-==	=-+=	-+=-	+==+	-==-	+-==	-===	+=-+	=-=+

Tabelle 4-4: Beispiel zur CHDB3-Codierung

4.5.4 Der Code HDB3 als Beispiel für eine Statemachine

Eine andere Variante, mit HDB3 (High Density Bipolarcode of Order 3) bezeichnet, bildet die Grundlage für die ISDN-Übertragungssysteme bei 2,048 Mbit/s, 8,448 Mbit/s und 34,368 Mbit/s (G.703). In dieser Variante, wiederum auf AMI basierend, werden die 0-Bits wie folgt codiert:

- Vier aufeinanderfolgende ‘0’-Bits der Binärfolge werden bei
  - gerader Anzahl von ‘1’ seit der letzten Verletzung durch ‘100V’ ersetzt,
  - ungerader Anzahl von ‘1’ seit der letzten Verletzung durch ‘000V’ ersetzt.

‘V’ bedeutet dabei die Verletzung der AMI-Regel, jeden von Null abweichenden Pegel zu alternieren; auf diese Weise kann die Vierergruppe von Nullen entdeckt werden. Treten nun mehrere Vierergruppen hintereinander auf, dann wird jede nach obiger Regel codiert, unabhängig, wieviele Vorgängergruppen existieren, wobei die ‘1’ und ‘V’-Pegel von Gruppe zu Gruppe alternieren; dies ergibt sich aus der Anzahl Null von ‘1’ zwischen diesen beiden Gruppen, die Null gilt als gerade Zahl. Beispiel: ‘0000 0000’ wird bei gerader Anzahl von ‘1’ vorher zu ‘100V 100V’ codiert.

Zur Codierung des HDB3-Codes müssen jeweils 4 Bits (B0 .. B3) in ein Schieberegister eingelesen und dann drei Fälle unterschieden werden:

1. Bit B0 ist eine ‘1’: Es folgt eine normale AMI-Codierung, in das Schieberegister wird das nächste Bit eingelesen, B0 wird herausgeschoben. Dieser Fall wird mit ‘H’ codiert
2. Bit B0 ist eine ‘0’, jedoch sind nicht alle Bits B1 bis B3 ‘0’. Jetzt wird eine normale ‘0’ ausgegeben, das nächste Bit eingelesen. Dieser Fall wird mit ‘L’ beschrieben.
3. Die Bits B0 .. B3 sind alle ‘0’, so daß der Ersetzungsfall auftritt. Jetzt wird ‘100V’ oder ‘000V’ auf die Leitung gegeben, je nach Anzahl der ‘1’ vorher. Dieser Fall sei mit ‘E’ codiert.

H, L und E können mit Hilfe eine Schieberegisters mit anschließender Logik sehr einfach dargestellt werden. In PAL-Assemblernotation lautet dies:

```
H = B0;
L = /B0 * B1 + /B0 * B2 + /B0 * B3;
E = /B0 * /B1 * /B2 * /B3;
```

{H, L, E} bilden nun das Eingangsalphabet eines (getakteten) Moore-Automaten, während mit {-, 0, +} das Ausgangsalphabet definiert sei:

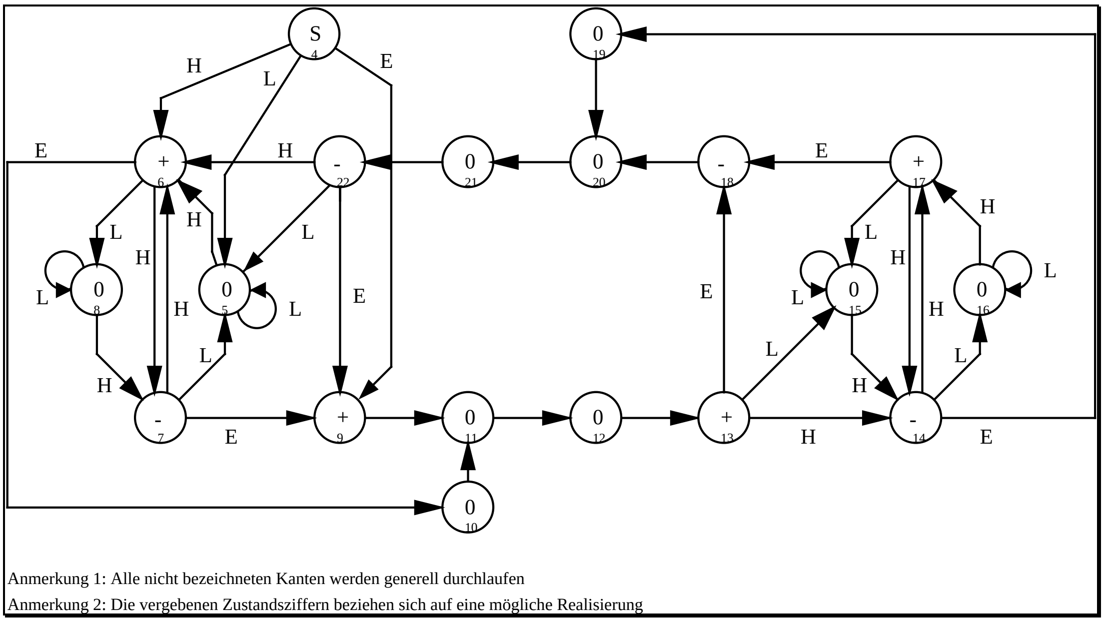

Abb. 4-6: Zustandsdiagramm für HDB3, Sender

Dieses Zustandsdiagramm kann nun leicht in eine textuelle Formulierung umgesetzt werden; dies ist beispielsweise in der Datei ST_HDB3S.S16 geschehen, während ein wesentlich einfacherer Empfangsautomat in den Dateien ST_HDB3E.S16 und ST_HDB3E.PAL beschrieben wird.

Kapitel 5: Serielle und parallele Schnittstellen für lokale Konzepte

Literatur: [4] Preuß, L.; Musa, H.: Computerschnittstellen. Carl Hanser Verlag München Wien, 1991

Die Vorlesung über Netze und Rechnerkommunikation wird an dieser Stelle – den Fluß der Ausführung für die Netze betreffend – durch Hervorhebung des Themas der seriellen und parallelen Schnittstellen und damit einhergehend der reinen Rechnerkommunikation unterbrochen. Dies hat besondere Gründe, da nunmehr die grundlegenden Eigenschaften zum Verständnis von Netzübertragungen geschaffen sind. Rechner-Rechner-Kommunikationen auf Basis von Punkt-zu-Punkt-Verbindungen sowie Feldbuskonzepte (Kapitel 6) sind teilweise bereits sehr lange in Benutzung, teilweise auch gerade in Entwicklung oder Definition. Gemeinsam ist ihnen die gegenüber Netzen wesentlich einfachere Handhabung bei verringerter Leistungsfähigkeit und damit eine universelle Einsetzbarkeit.

In diesem Kapitel werden zwei serielle Kommunikationen sowie eine parallele Schnittstelle vorgestellt: RS-232, RS-485 und IEEE-488.

5.1 Die serielle Schnittstelle RS-232C (V.24/V.28)

5.1.1 Allgemeines

Die Schnittstelle RS-232C (Recommended Standard) entspricht einer bestimmten Form der V.24/V.28-Definition der CCITT. Die Feinheiten sind dabei eher unwichtig, wichtig ist aber der Umfang und die Verwendung dieser Standardschnittstelle.

RS-232C (C bedeutet quasi die Revisionnummer) wurde für die asynchrone Punkt-zu-Punkt-Verbindung zwischen Datenendeinrichtungen (DEE) und Datenübertragungseinrichtungen (DÜE, DTE) definiert: Es ist also eine unsymmetrische Definition, die heute ein bißchen mißachtet wird: Unter Umgehung der exakten Definition von RS-232C können symmetrische Punkt-zu-Punkt-Verbindungen aufgebaut werden; dies ist ein Quasi-Standard für Peripheriegeräte mit niedrigen Übertragungsraten.

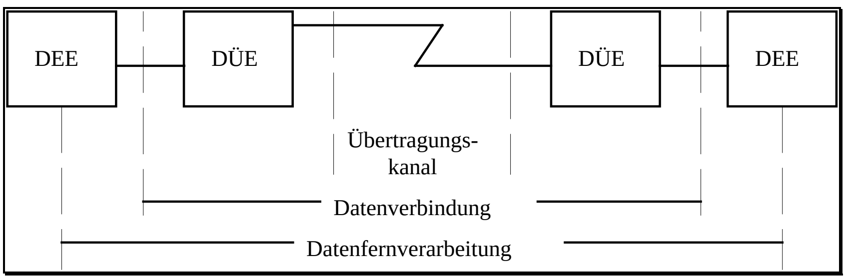

Abb. 5-1: Frühe Form der DFÜ

5.1.2 Die Leitungsdefinition der RS-232C

Abb. 5-2 zeigt diese Teilausrüstung der V.24-Schnittstellen mit Belegung am 25poligen Submin-D-Stecker (ISO 2110) und am 9poligen Submin-D-Stecker (PC/AT-Norm):

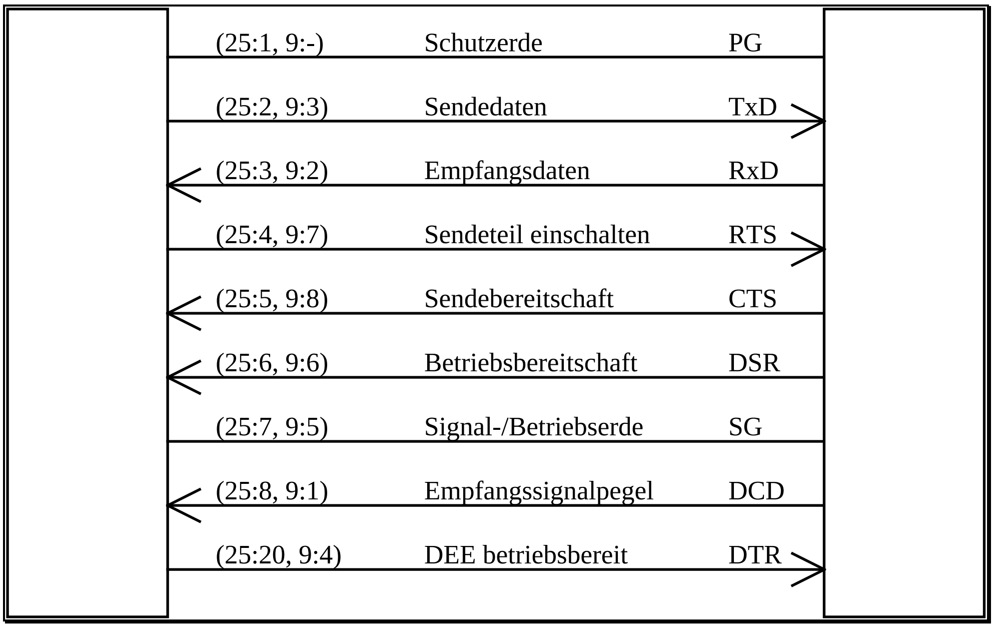

Abb. 5-2: Teilausrüstung V.24-Schnittstelle

Betriebserde und Rückleiter:

- **PG (Protective Ground, Schutzerde)**

  Die Schutzerde wird mit dem Gehäuse und dem Schutzleiter der miteinander gekoppelten Übertragungseinrichtungen DEE und DÜE verbunden. Im 9poligen System entfällt dieser Pin, nicht aber die Funktion durch die Kabelschirmung.

- **SG (Signal Ground, Betriebserde)**

  Die Betriebserde liegt auf einem Pegel von 0 Volt und dient für alle Signale als gemeinsamer Bezugspunkt. Sie muß von der Schutzerde getrennt sein!

Datenleitungen:

- **TxD (Transmit Data, Sendedaten)**

  Über diese Leitung werden der DÜE die zu übertragenden Daten als bitserieller, digitaler Datenstrom zugeführt. Die Zuführung ist zumeist asynchron, also ohne zusätzliche Taktleitung, das empfangende Gerät synchronisiert seinen Empfang selbständig. Diese Datenübertragung ist nur möglich, wenn sich die Steuerleitungen RTS und DTR sowie die Meldeleitungen CTS und DSR im EIN-Zustand befinden.

- **RxD (Receive Data, Empfangsdaten)**

Die DEE empfängt auf gleiche Art die empfangenen Daten aus der DÜE.

Steuerleitungen:

- **RTS (Request To Send, Sendeteil einschalten)**
  Diese Steuerleitung veranlaßt das Modem, das Sendeteil einzuschalten und in diesem Zustand zu verbleiben, solange RTS den Pegel EIN behält. Zum RTS-Signal zugehörig ist die Meldeleitung CTS, so daß dieses Leitungspaar mit gegenseitiger Kopplung zur direkten Rechner-Rechner-Anbindung genutzt werden kann.

- **DTR (Data Terminal Ready, DEE betriebsbereit)**
  Der Zustand EIN auf dieser Leitung signalisiert der DÜE, daß die DEE eingeschaltet und empfangsbereit ist, falls Daten an den Rechner gesendet werden sollen.

Meldeleitungen

- **CTS (Clear To Send, Sendebereitschaft)**
  meldet die DÜE via EIN-Zustand auf dieser Leitung an die DEE. RTS/CTS bilden, wie bereits erwähnt ein Leitungspaar.

- **DSR (Data Set Ready, Betriebsbereitschaft)**
  ist eine durchaus vergleichbare Meldeleitung, mit der aber die Verbindung zum Übertragungskanal und die Betriebsbereitschaft (ON-Line) durch den EIN-Zustand signalisiert werden soll.

- **DCD (Data Channel Received Line Signal Detector, Empfangssignalpegel)**
  zeigt den Empfangssignalpegel innerhalb gewisser Grenzen durch den EIN-Zustand an. Werden zwei Rechner ohne Übertragungskanal miteinander gekoppelt, dann soll DCD sofort den Zustand EIN annehmen oder mit dem RTS bzw. DTR-Signal der gegenüberliegenden Seite gekoppelt werden, um sofort den Zustand des verbundenen Kabels anzuzeigen.

Es muß an dieser Stelle nochmals auf die unsymmetrische Definition der V.24-Schnittstelle hingewiesen werden. Die Leitungsbezeichnungen sind für DEE und DÜE identisch, nur bedeutet TxD für die DEE, daß wirklich Daten gesendet werden, während für die DÜE auf dieser Leitung Daten von der DEE empfangen werden. Dies ergibt gewöhnlich Probleme bei der Kopplung zwischen Computer und einem anderen Gerät gemäß RS232, da beide Geräte die "Norm" erfüllen können, ohne daß es sofort zusammenpassen muß. Auf diese Problematik wird in Kürze eingegangen.

5.1.3 Elektrische Signalpegel

Die Definition der elektrischen Signalpegel beinhaltet die Spannungswerte für die logischen Werte '0' und '1' sowie Festlegungen zu den Geschwindigkeiten, Anstiegs/Abfallraten der Spannungswerte usw.

Gemäß V.28 haben die Spannungswerte auf den Datenleitungen folgende Definition:

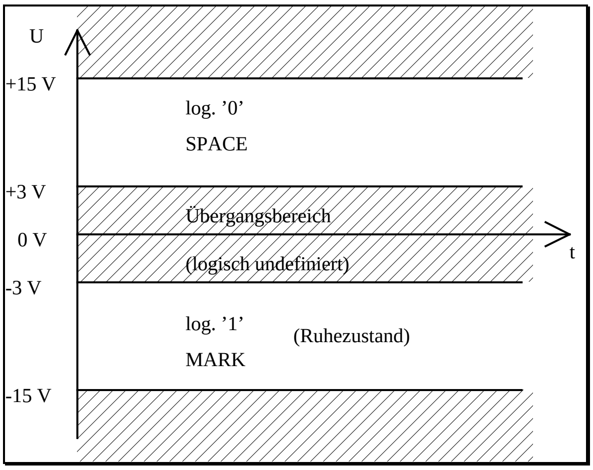

Abb. 5-3: Logikdefinition für Datenleitungen

Ist die Spannung auf einem der Datenleitungen (TxD, RxD) also gegenüber der Betriebserde (SG) um 3 bis 15 Volt (Empfang) bzw. 5 bis 15 Volt (Sendebetrieb) im Betrag unterschiedlich und

- **negativ**, so liegt eine logische '1' auf der Leitung; dies wird auch *MARK* genannt und stellt den *Ruhezustand* dar: hingegen
- **positiv**, so wird eine logische '0' repräsentiert; dieser Zustand heißt auch *SPACE*.

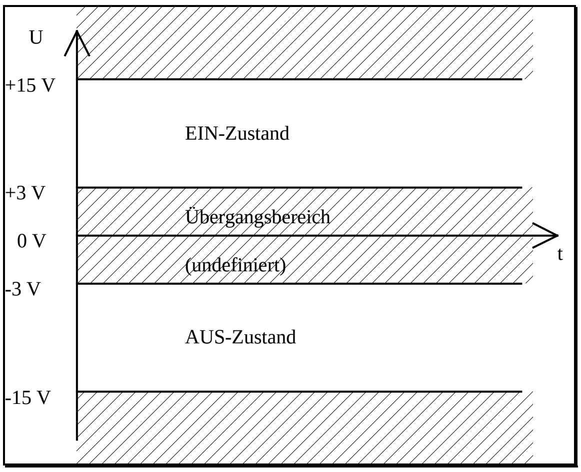

Abb. 5-4: Logikdefinition für Steuer- und Meldeleitungen

Für Signal- und Meldeleitungen gibt es eine eigene Definition: Ist die Spannung einer Signal- oder Meldeleitung im Betrag 3 bzw. 5 bis 15 Volt größer als die Betriebserde (SG) und

- negativ, so herrscht der AUS-Zustand,
- positiv, so herrscht den EIN-Zustand.

Abb. 5-4 stellt dies grafisch dar.

Mit den logischen und physikalischen Pegeln ist noch nichts über die minimale Anstiegsrate (in V/µs), Welligkeit (in % vom Vollausschlag) usw. ausgesagt worden, und in der Tat definieren V.24/V.28 sowie RS232C, RS232D und RS232E keine minimalen bzw. maximalen Werte zu diesen Größen, allerdings zur maximalen Anstiegsrate mit 30 V/µs. Abhilfe schafft an dieser Stelle erst der mit RS232 kompatible neue (1991) Standard EIA RS562, der z.B. eine Mindestanstiegsrate von 4 V/µs und eine maximale Welligkeit von 5% (bezogen auf den Spannungshub V_{High} - V_{Low}) zuläßt.

5.1.4 Synchronisation und Übertragungsgeschwindigkeit

RS-232C ist ein asynchrones Verfahren (siehe 4.3), das also ohne spezielle Synchronisationsleitungen auskommt. Daraus ergibt sich ein Synchronisationsrahmen mit Start- und Stoppbit, die neben einem eventuellem Paritätsbit zu den (üblicherweise 8) Datenbits hinzukommen. Zu den Schnittstellenvereinbarungen zweier gekoppelter Geräte gehören demnach

- Anzahl der Bits pro gesendetem Zeichen
- Übertragungsgeschwindigkeit (Wechselfrequenz) der Bits
- keine spezielle Vereinbarungen wie z.B. spätestens nach 5 * logisch '0' einmal logisch '1' automatisch dazuzugeben

Der Zeichenrahmen besteht seinerseits aus zusätzlichen Bits, die also keine Nutzinformationen mehr enthalten. Diese Bits sind

- 1 Start-Bit, das im Wechsel zum Ruhezustand steht (logisch '0'),
- kein oder 1 Paritätsbit, dessen Wert definierbar von den Werten der Datenbits abhängt, und
- 1, 1.5 oder 2 Stop-Bits, die den Zustand logisch '1' innehaben und sofort vom nächsten Start-Bit gefolgt sein können.

Das Paritätsbit wird, wenn es vereinbart wurde, wie folgt bestimmt:

Paritätskontrolle	Datenwort	Paritätsbit	
gerade (even)	10011000	1	(zusammen 4 * '1')

	11110101	0	(zusammen 6 * ’1’)
ungerade (odd)	10011000	0	(zusammen 3 * ’1’)
	11110101	1	(zusammen 7 * ’1’)

Das Paritätsbit ergänzt den Datenstrom also so, daß eine gerade oder ungerade Anzahl von Bits, je nach Wunsch, entsteht und dient so der *Fehlererkennung* (nicht der *automatischen Fehlerbehebung*).

Der vollständige Bitstrom, der übertragen wird, hat also die Gestalt wie in Abb. 5-5 angegeben:

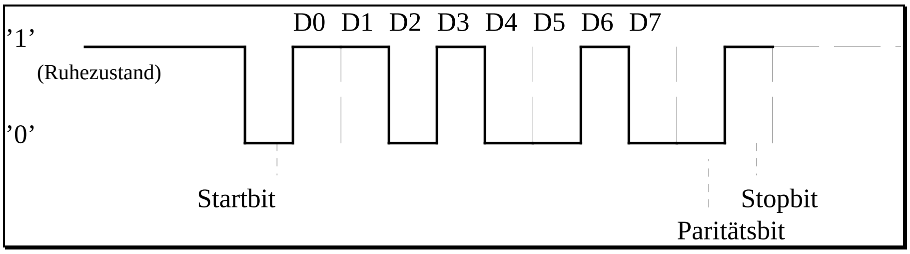

Abb. 5-5: Bitstrom im asynchronen Verfahren (1 Start-, 1 Stopbit, gerade Parität, 8 Datenbits)

Im asynchronen Verfahren ist die maximale Geschwindigkeit der RS232-Schnittstelle auf 20 kBit/s, auch 20 kBaud genannt begrenzt (RS562: 64 kBit/s). Übliche Übetragungsgeschwindigkeit sind 110 Baud, 300 Baud, 600 Baud, 1.2 kBaud, 2.4 kBaud, 4.8 kBaud, 9.6 kBaud und 19.2 kBaud. In Computern gelegentlich auffindbare höhere Bitraten von 38.4, 57.6 oder 115.2 kBaud werden gewissermaßen "auf eigene Gefahr" betrieben.

Bei 19200 Bit/s hat ein Bit eine Sendedauer von 52 µs. Der Wirkungsgrad einer bitseriellen Übertragung mit 1 Startbit, 1 Paritätsbit und 1 Stopbit bei 8 Datenbits beträgt 8/11 = 72,3 % maximal, für die Übertragung von 1 MByte Nutzdaten werden rund 9,5 Minuten Zeit benötigt.

5.1.5 Rechner-Rechner-Kopplung mittels RS-232

Die RS-232-Schnittstelle hat sich aufgrund ihrer Universalität und trotz ihrer "Un"definiertheit im Computerbereich erfolgreich durchgesetzt. Rechner-Peripherie- und Rechner-Rechner-Kopplungen werden gerne mit Hilfe einer solchen seriellen Schnittstellen durchgeführt.

Die Verdrahtung der Schnittstellen miteinander erweist sich dabei häufig als Problem, dessen Ursache in der ursprünglichen Definition liegt. Eine möglichst problemfreie Verbindungsart ist im folgenden angegeben.

Das Mehrdraht-Handshake-Verfahren (häufig auch fälschlicherweise als Nullmodem-Verfahren bezeichnet, siehe Abb. 5-8) besteht aus der gegenseitigen Verbindung der Paare RTS/CTS und DTR/DSR. Der empfangende Rechner kann den Zeichenstrom durch seine Handshake-Leitungen steuern, indem er bei drohendem Überlauf RTS und/oder DTR auf AUS setzt und somit die Übertragung stoppt.

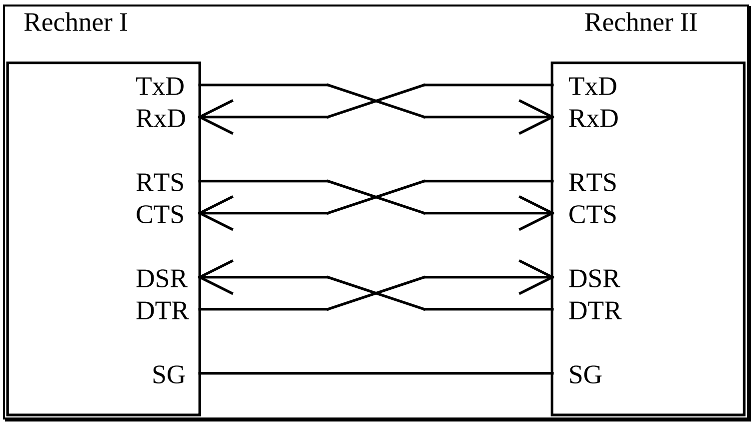

Abb. 5-6: Verkabelung im Mehrdraht-Handshake-Verfahren

In Abb. 5-7 wird das andere Extrem dargestellt, die sogenannte Drei-Draht-Leitung. Verbindungen an den Rechner zwischen RTS/CTS und DTR/DSR sind ggf. notwendig.

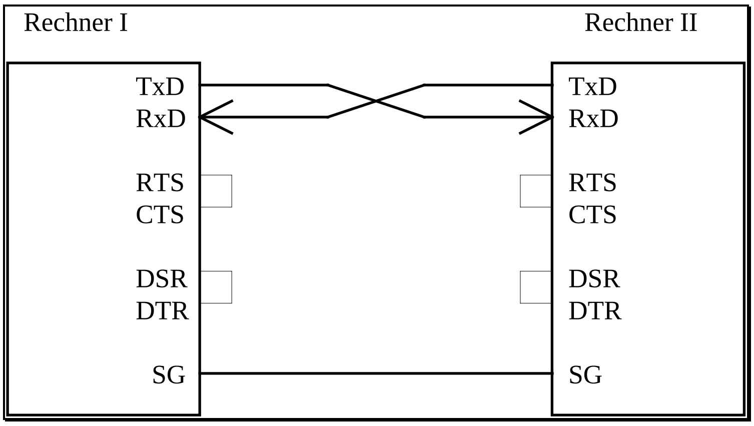

Abb. 5-7: Drei-Draht-Leitung mit Software-Handshake

Hardwaremäßig sehen beide Rechner ständige Empfangsbereitschaft, so daß, falls der Zeichenstrom gesteuert werden soll oder muß, in diesem Fall ein zusätzliches Softwareprotokoll (Software-Handshake) eingeführt werden muß. Diese Verfahren können vom Empfänger aus steuern (Beispiel: **XON/XOFF-Protokoll**), oder aber auch vom Sender aus (**ETX/ACK-Protokoll**).

Das bekannteste dieser Verfahren ist zeichen-orientiert und wird vom Empfänger aus gesteuert, wenn dessen Puffer droht, überzulaufen. In diesem Fall kann er im gesamten Zeichenstrom ein vereinbartes Zeichen (XOFF, ASCII-Bezeichnung DC3, Wert 13h, "CTRL-S") an den Sender schicken, und dieser muß bis zum Eintreffen des Freigabezeichens (XON, ASCII-Bezeichnung DC1, Wert 11h, "CTRL-Q") die weiteren Sendungen von Zeichen stoppen. Dieses Verfahren simuliert auf einfache Art die Hardware-Handshake-Leitungen, muß aber gewöhnlich in der Software ausgeführt sein.

Das andere Verfahren ist weniger sicher, da hier eine Unterbrechung nicht vorgesehen ist. Die Kommunikation arbeitet auf Blockebene, sie wird durch den Empfänger mittels des Zeichens ACK (Wert 06h, "CTRL-F") eingeleitet, der Sender sendet dann einen (von ihm bestimmten) Block von Zeichen ohne Unterbrechungsmöglichkeit und beendet diesen Block mit dem Steuerzeichen ETX (Wert 03h, "CTRL-C").

Sender und Empfänger müssen für dieses Verfahren abgestimmte Blockgrößen besitzen; setzt man die Blockgröße auf 1 Zeichen, dann wird das Verfahren zwar sicher, aber der Übertragungs-Overhead gegenüber den anderen Verfahren sehr groß.

Abschließend zeigt Abb. 5-8 die eigentliche Nullmodemverkabelung, die das Vorhandensein der Modems gegenseitig simuliert und damit die sicherste Form der Verbindungsherstellung sein dürfte.

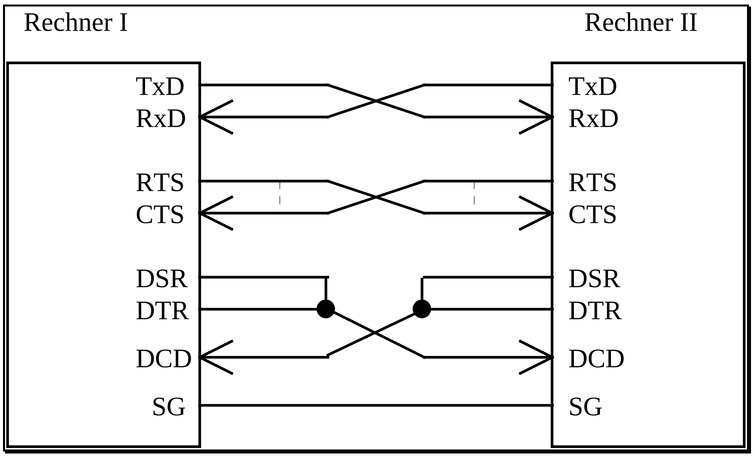

Abb. 5-8: Verkabelung im Nullmodem-Verfahren

Die RTS-CTS-Verbindungen können dabei eventuell aufgetrennt werden und durch Brücken am Rechner selbst die entsprechenden Pegel zur Kommunikation liefert (punktierte Linien).

Analoge ICs, die die Spannungen zwischen TTL-kompatiblen Werten (+5 Volt oder +3.3 Volt) und den RS-232-Werten wandeln können, sind z.B.

MAX230 bis MAX253
MAX3222 bis MAX3243

Das Betreiben der Schnittstelle, also das Erkennen des Zeichenrahmens, die Speicherung der Bits sowie Fehlererkennungen kann beispielsweise über die digitalen Schnittstellenbausteine

USART 8251
UART 8250
UART 16450

erfolgen.

5.2 Die Weiterentwicklung zur RS422/423/485

Im PC-Bereich ist die RS-232-Schnittstelle sehr weit verbreitet, für den professionellen Bereich mit dem Zwang zu sichereren und schnelleren Verbindungen wurden Weiterentwicklungen durchgeführt, deren Vorteile in Abb. 5-9 dargestellt sind:

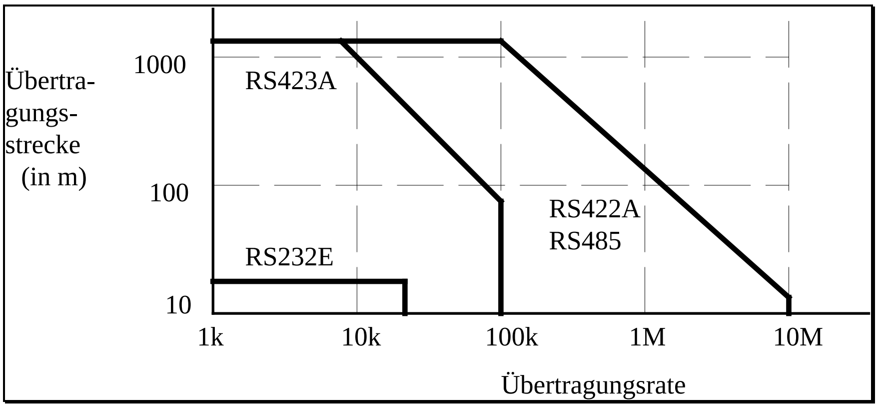

Abb. 5-9: Vergleich serieller Schnittstellen

Zu beachten ist in Abb. 5-9 der doppelt logarithmische Maßstab!

Tabelle 5-1 stellt einige wesentliche Eigenschaften der seriellen Schnittstellen gegenüber:

	RS232C/E	RS562	RS423A	RS422A	RS485
Übertragungsart	Asymmetr.	Asymmetr.	Asymmetr.	Symmetrisch	Symmetrisch
Anzahl Treiber	1	1	1	1	32
Empfänger	1	1	10	10	32
max. Kabellänge	15/3 m	nicht spez.	1200 m	1200 m	1200 m
max. Datenrate	20 kBit/s	64 kBit/s	100 kBit/s	10 Mbit/s	10 MBit/s
Treiberausgangssignal					
ohne Last	+/- 15 V	+/- 13.2 V	+/- 6 V	+/- 5 V	+/- 5 V
mit Last	+/- 5 V	+/- 3.7 V	+/- 3.6 V	+/- 2 V	+/- 1.5 V
Treiberlast	3 .. 7 k Ω	3 .. 7 k Ω	450 Ω	100 Ω	54 Ω
Eingangsspannung	+/- 15 V	+/- 15 V	+/- 12 V	+/- 7 V	-7 ... +12 V
Empfindlichkeit	+/- 3V	+/- 3 V	+/- 200 mV	+/- 200 mV	+/- 200 mV

Tabelle 5-1: Schnittstellenvergleich

5.2.1 Diskussion der Schnittstellen RS422, RS423 und RS485

RS422A, RS423A und RS485 kennen nur noch die Sende- und Empfangsleitungen sowie die Betriebserde, so daß bei einer Datenübertragung immer Softwareprotokolle zur Datenflußregelung zum Einsatz kommen müssen.

Unterschiedlich sind für alle drei Schnittstellen die elektrischen Übertragungsparameter. RS423A benutzt ähnlich RS232E eine asymmetrische Spannung von +/- 3.6 bis +/- 6 Volt an einer Signalleitung gegen Masse. RS422A und RS485 senden beide eine Spannungsdifferenz auf zwei Signalader pro Richtung. Diese Spannungsdifferenz beträgt +/- 1.5 V bis +/- 5 Volt, je nach Last. Da das Signal als Spannungsdifferenz aufgebaut ist, bewirkt eine elektromagnetische Störung auf einem Kabel nichts, da die induzierte Spannung ja auf beide Kabel gleichermaßen vorhanden ist, die Spannungsdifferenz also im wesentlichen unberührt bleibt.

RS422A und RS485 unterscheiden sich in der Anzahl der anschließbaren Sender/Empfänger: RS422A wird als Point_to_Multipoint-Verbindung bezeichnet, mit RS485 wird der Übergang zwischen den Point_to_Point-Verbindung klassischer Art zu den Netzwerken geschaffen, bei denen ein spezielles Zugangsverfahren zum Übertragungsmedium Kabel geschaffen werden muß, damit nicht alle Sender (hier 32) gleichzeitig senden.

RS485 ist die Grundlage für Feldbussysteme, hat seinen Haupteinsatzpunkt also in der Prozeßsteuerung. Als elektrischer Schnittstellenbaustein für RS422A und RS485 wird z.B. der Transceiverbaustein SN 75176A eingesetzt. Alle hier genannten Verfahren können weiterhin mit den UART/USART-Bausteinen betrieben werden, da die Asynchronität erhalten bleibt und sich lediglich die elektrischen Parameter ändern. Der UART 16450 beispielsweise kann Datenraten bis 115 kBaud verarbeiten. Schnellere Bausteine auch mit Einbeziehung von weiteren Protokollschichten sind für die einzelnen Buskonzepte vorhanden.

5.2.2 Die seriellen Schnittstellen im OSI-Basisreferenzmodell

Die Entwicklung der ersten seriellen Schnittstellen erfolgte weit vor den Bestrebungen zur Vereinheitlichung der Kommunikationsprotokolle. Aus diesem Grund lassen sich die älteren Schnittstellen wie die Weiterentwicklungen nur bedingt auf die Layer im OSI-Modell abbilden.

Für die einzelnen Normen gelten folgende Zuordnungen:

RS232/EIA562:

- Bitübertragungsschicht ist definiert, teilweise mit Lücken.
- Layer 2: Media Access Control ist für den Hardwarezugriff definiert; beim Software-gesteuerten Zugriff (3-Draht-Leitung) muß dies in der Applikation erfolgen.
- Layer 3 und 4 entfallen für Point-To-Point-Verbindungen
- Layer 5 bis 7 werden in den Applikationen definiert.

RS422/485:

- Layer 1 ist definiert.
- Layer 2: RS422 hat getrennte Leitungen für Senden und Empfang, ansonsten keine Steuerleitungen, d.h., MAC muß per Software als Flußsteuerung definiert werden. RS485 besitzt 1 oder 2 Leitungspaare, vor

allem jedoch maximal 32 Sender, so daß nicht nur die Flußkontrolle, sondern auch die Zugriffskontrolle durch höhere Normen (z.B. Feldbus!) oder per Software erfolgen muß.

- Layer 3 bis 7 sind undefiniert.

5.3 Parallele I/O-Systeme: IEEE-488 bzw. IEC-625

5.3.0 Einführung in die Thematik

Der IEEE-488-Bus (IEC-625, General Purpose Interface Bus GPIB) entstand aus den Gedanken, einen universellen Meßgeräteanschlußbus zu schaffen. Meßgeräteaufbauten waren bereits Anfang der 70er Jahre eine Tatsache, nur der Anschluß untereinander sowie zum Computer war ungenormt und somit eine Spezialität für jeden Aufbau.

1965 erstmalig von Hewlett-Packard vorgestellt (als HP-IB oder GP-IB), wurde dieser Interfacebus 1975 erstmalig genormt. Die verschiedenen Normungen unterscheiden sich letztendlich nur im Steckerformat, nicht in der Anzahl der Leitungen oder in der Funktionalität.

Die Grundidee des IEEE-488-Bus' ist eine Bit-parallele, Byte-serielle Übertragung mit entsprechenden Handshake- und Kommandoleitungen bzw. -signalen. Im folgenden wird diese Funktionalität im einzelnen erklärt.

5.3.1 Hardware des IEEE-488-Bus

Der Bus besteht aus insgesamt 16 Signalleitungen, die in drei Gruppen unterteilt werden:

- Datenbus (8 Bit)
- Data Byte Transfer Control Bus (Übergabesteuerbus, Handshake Bus)
- General Interface Management Bus (Steuerbus)

Abb. 5-10 zeigt den Aufbau des IEEE-488-Bus im Detail. In dieser Abbildung wird nicht näher zwischen dem eigentlichen **Gerät** und dem **Interface** zum Bus unterschieden, obwohl dieser Unterschied gelegentlich wichtig ist: Das **Gerät** tauscht lediglich Daten mit anderen Geräten aus, hat also nur Zugang zum Bus, wenn dieser nicht Kommandos sendet, sondern Daten. Das **Interface** ist für den ordnungsgemäßen Zugang zum Bus zuständig und wertet zu diesem Zweck die Kommandos am Bus aus, während Daten lediglich "durchgeschleift" werden.

Physikalisch besitzen alle Bussignale aktiv low definierte TTL-Pegel, also die Darstellung des Signals '0' durch einen Pegel von >2.0 Volt und von '1' durch < 0.8 Volt (am Empfänger). Ein Buchstabe 'E' am Datenbus (ASCII-Wert 45h) wird somit (scheinbar) als 0BAh entsprechend 10111010b am Bus gesendet, wobei natürlich der

Empfänger diese Signale wieder invertiert. Die Leitungen sind mit Tristate- (High-Speed bis 1 MByte/s Übertragungsgeschwindigkeit) oder als Open-Collector-(bis 200 kByte/s) Treiberlogik ausgestattet. Der Sinn solcher Treiberelemente wird in Abschnitt 5.3.2 näher erläutert.

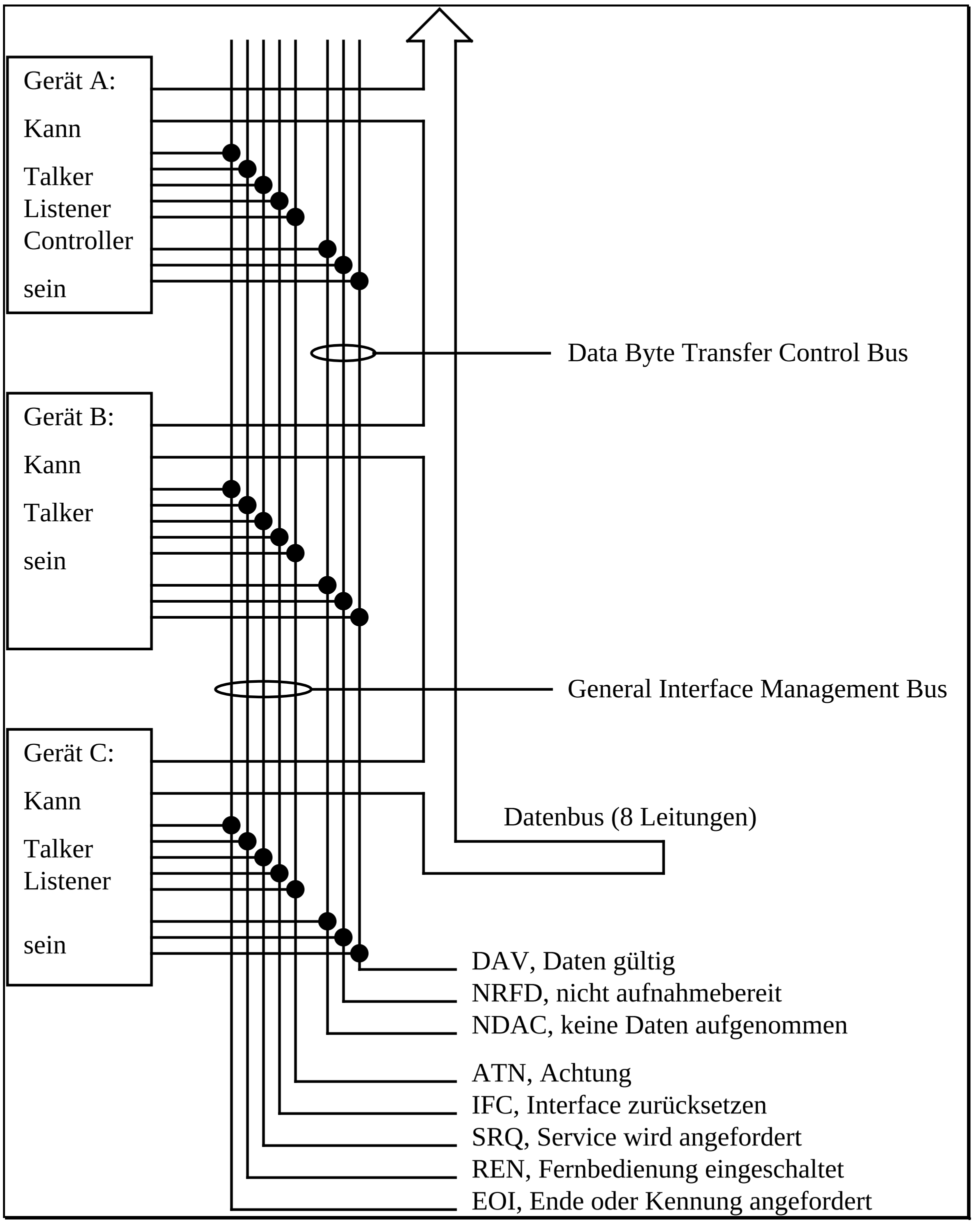

Abb. 5-10: IEEE-488-Bus-Struktur

Der 8-Bit breite Datenbus, der bidirektional ausgeführt ist, wird mit DIO1 bis DIO8 (Data Input/Output 1 .. 8) bezeichnet und kann ein freies Format übertragen, also ASCII-Werte, Binärformat usw. Die Formatentscheidung obliegt der Anwendung! Das niederwertigste Bit wird dabei auf Leitung DIO1 übertragen.

Der Datenbus kann generell Daten oder Kommandos führen, die Entscheidung darüber wird in der Leitung ATN (Attention, Achtung).

Die übrigen Signale im Detail:

Der *Data Byte Transfer Control Bus* besteht aus drei Leitungen, deren Funktion im nächsten Abschnitt genauer erläutert wird:

DAV - Data Valid, Daten gültig

Ein aktiv Low-Pegel auf dieser Leitung signalisiert, daß es sich bei den Pegeln auf dem Datenbus um eine gültige Information handelt.

NRFD - Not Ready For Data, nicht aufnahmebereit

Dieses Signal (aktiv low) teilt dem Sender (Talker) mit, daß mindestens einer der Empfänger (Listener) momentan keine Daten entgegennehmen kann.

NDAC - Not Data ACcepted, Daten nicht übernommen

Mindestens einer der Zuhörer hat das angebotene Datum noch nicht angenommen (ebenfalls aktiv low).

Die *General Interface Management Bus* Leitungen dienen dem generellen Management des IEEE-488-Bus'. Dazu ist folgendes anzumerken:

Geräte am Interfacebus können grundsätzlich drei Funktionen ausführen: Listener, Talker und Controller. Während (fast) beliebig viele Listener und Talker am Bus sein können, ist natürlich nur ein Controller zu einem Zeitpunkt erlaubt. Die Rechte des Controllers könn en übertragen werden.

Die Übertragung von Daten muß am Bus generell durch Kontrollsignale begonnen werden. Dieses Verfahren wird durch den Controller eingeleitet und beendet. Kontrollsignale werden über den Datenbus transferiert, wobei zur Unterscheidung von normalen Daten die Leitung ATN aktiv (low) gesetzt sein muß.

Der Controller teilt den Teilnehmern am Bus mit, was er will. Dazu werden auf den Datenleitungen DIO1 bis DIO5 die Geräteadresse (also eine von 32 möglichen) am Bus gelegt; Bit 7 und 6 bezeichnen, ob der Befehl als Talker- (10), Listener-(01)Befehl, als IEEE-Bus-Befehl (00) oder als sogenannter Sekundärbefehl (11) zu verstehen ist. Der Controller richtet also am Bus z.B. eine Reihe von Listener und einen Talker ein.

Der Schluß des Datentransfers (mit ATN = high) wird durch die spezielle Leitung EOI eingeleitet, die beim Datentransfer ENDE bedeutet. Anschließend werden die Befehle (mit ATN = low) Unlisten (0111111b) und/oder Untalk (1011111b) gegeben, die alle weiteren Aktivitäten der Teilnehmer beenden. Diese Befehle, die quasi die Geräteadresse 32 beinhalten, begrenzen die Anzahl der Teilnehmer auf 31.

Die Management-Leitungen im einzelnen:

ATN - ATteNtion, Achtung

dient der Umschaltung zwischen Gerät (Datenübertragungen) und Interface (Befehle). ATN = low (aktiv) bedeutet, daß nunmehr Adreß- und Steuerinformationen über den Datenbus kommen, die das Interface betreffen. ATN = high (inaktiv) selektiert den Datenbus als Datenträger für das Gerät.

EOI - End Or Identify, Ende oder Kennung angefordert

besitzt zwei Bedeutungen, die von ATN abhängen. Für ATN = high, also im Datentransfermodus, zeigt die gesetzte (aktiv low) Leitung EOI den Transfer des letzten Bytes an. Für ATN = low, im Kommandomodus, fordert der Controller die angeschlossenen Geräte zu einer Parallelabfrage auf.

REN - Remote ENable, Fernbedienung eingeschaltet

schaltet die Fernbedienung der angeschlossenen Geräte ein oder aus. REN low bedeutet, daß die Fernbedienung eingeschaltet ist, REN = high läßt die Geräte in die lokale Handbedienung zurückkehren.

SRQ - Service ReQuest, Bedienung angefordert

stellt eine Interruptleitungen des IEEE-488-Bus' dar. Geräte, die nicht die Controllerfunktion innehaben, aber eine Bedienung erwarten, können über den aktiv low Pegel auf dieser Leitung eine Abfrage durch den Controller erzwingen.

IFC - InterFace Clear, Interfaces in den Grundzustand

ist aktiv low für mindestens 100 Mikrosekunden und stellt eine Art Resetleitung dar, die nur vom Controller gesetzt werden kann. Alle angeschlossenen Geräte müssen dann augenblicklich in ihren Grundzustand zurückkehren.

5.3.2 Handshake-Verfahren 2- und 3-Draht

5.3.2.1 Die Buslogik

Die Daten- und Steuerbustreiber können u.a. in Open-Collector-Technik ausgeführt sein, deren Verfahren sehr einfach in Abb. 5-11 dargestellt ist:

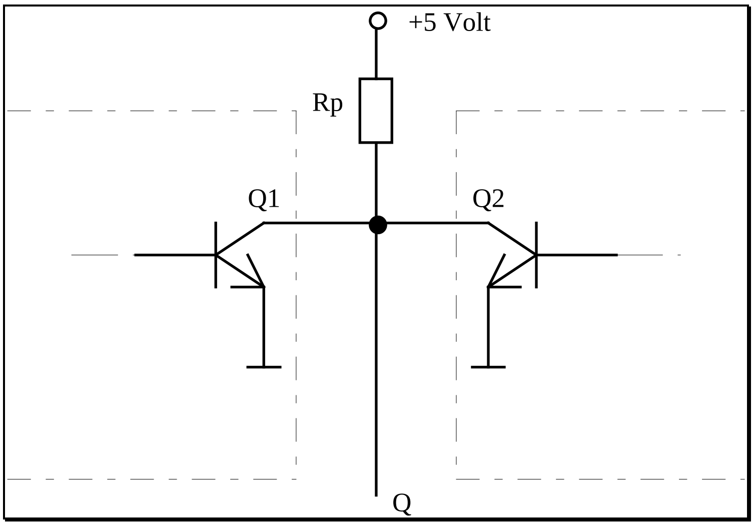

Abb. 5-11: Open-Collector-Ausgänge

Open Collector-Treiber bestehen also aus npn-Transistoren mit offenem Collector, der extern über einen Widerstand Rp auf 5 Volt Potential gezogen wird. Schaltet der Transistor durch, fallen die 5 Volt über dem Widerstand ab, die Leitung Q führt den logischen Level 0. Sperrt der Transistor, dann liegt die Leitung auf Potential High (gemäß TTL-Pegeldefinition).

Liegen nun mehr als ein Ausgang in Open-Collector-Technologie auf einer Summenleitung, dann genügt das Durchschalten eines Transistors, um die gesamte Leitung auf den Pegel low zu senken; insbesondere kommt es zu keinerlei Konflikten, da ja nur das Collectorpotential der anderen Transistoren auf Low-Pegel abgesenkt wird, was aber zu keinen Kurzschlüssen führt. Bei positiver Logik nennt man dies Wired-AND, da gemäß logischer UND-Verknüpfung vieler Eingänge das Ausgangssignal auf Low liegt, falls nur einer dieser Eingänge den logischen Wert low hat.

Da der IEEE-488-Bus negierte Logik (aktiv low) führt, nennt man das Verfahren in diesem Fall **Wired-OR**.

Die sich aus Rp und allen Transistoren ergebende RC-Kombination ist letztlich verantwortlich für die maximale Übertragungsgeschwindigkeit.

5.3.2.2 Das Zweidraht-Handshake-Verfahren

Für den am Bus einfachsten Fall, ein Talker, ein Listener, läßt sich ein Handshake-Verfahren mittels zweier Verbindungen konstruieren, das in jedem Fall eine sichere Verbindung gewährleistet. Dazu werden die Leitungen DAV und NDAC ausgewählt, so daß das Verfahren wie folgt abläuft:

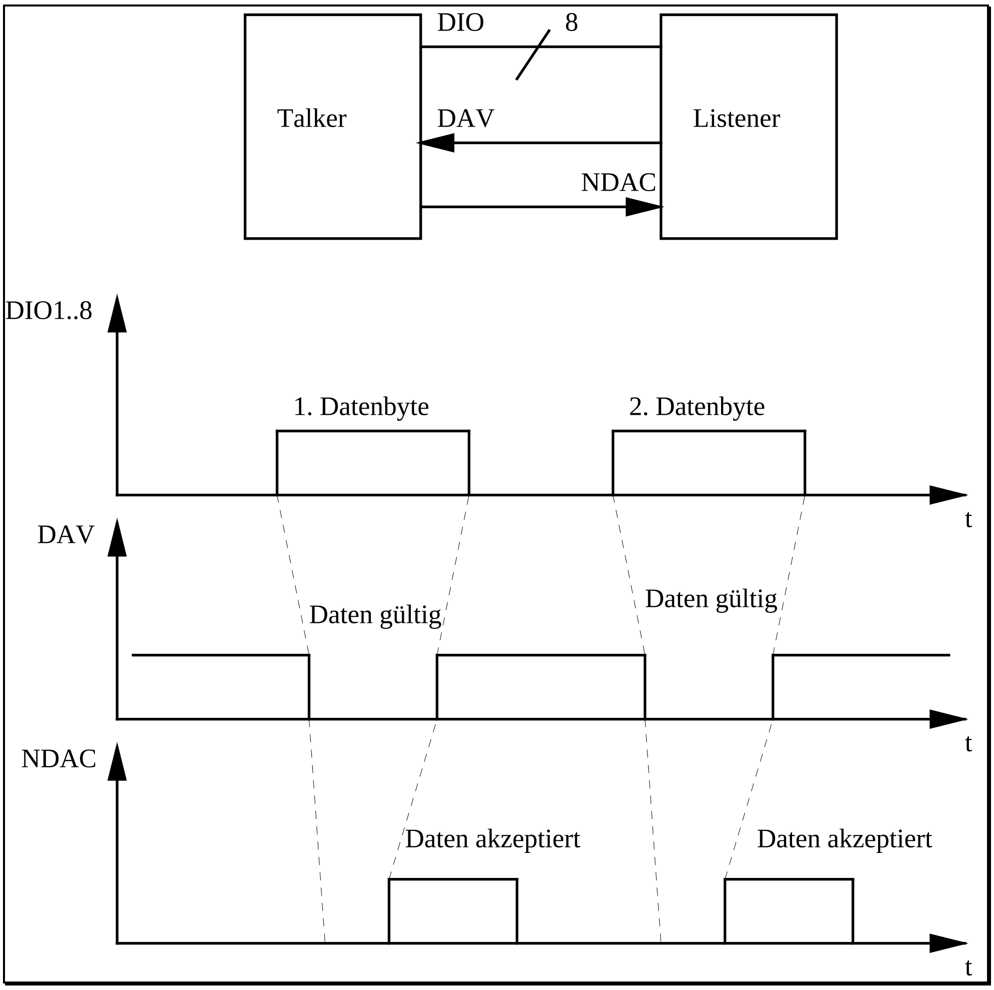

Abb. 5-12: Zweidraht-Handshake

Dem Listener wird mittels des Signals DAV (aktiv low) signalisiert, daß gültige Daten am Datenbus vorliegen. Er hat dann prinzipiell beliebig Zeit, die Daten zu akzeptieren und signalisiert dies mittels NDAC inaktiv (high), worauf der Talker seine Signale DAV inaktiv stellt und ein neues Datenbyte auf den Bus bringt. Nachdem der Talker DAV inaktiv gestellt hat, setzt der Listener NDAC wiederum aktiv low, um beim nächsten Datenbyte nicht sofort den Anschein zu erwecken, daß sofort das Datenbyte akzeptiert worden wäre.

Dieses Verfahren ist auf einen Talker und einen Listener beschränkt, bei mehreren Zuhörern kann es bei verschiedenen Verarbeitungsgeschwindigkeiten ab dem 2. Datenbyte zu Problemen kommen, wie im folgenden gezeigt werden wird.

Der Talker setzt seine Signale wie bisher, nur die Listener zeigen sehr unterschiedliche Verhaltensweisen:

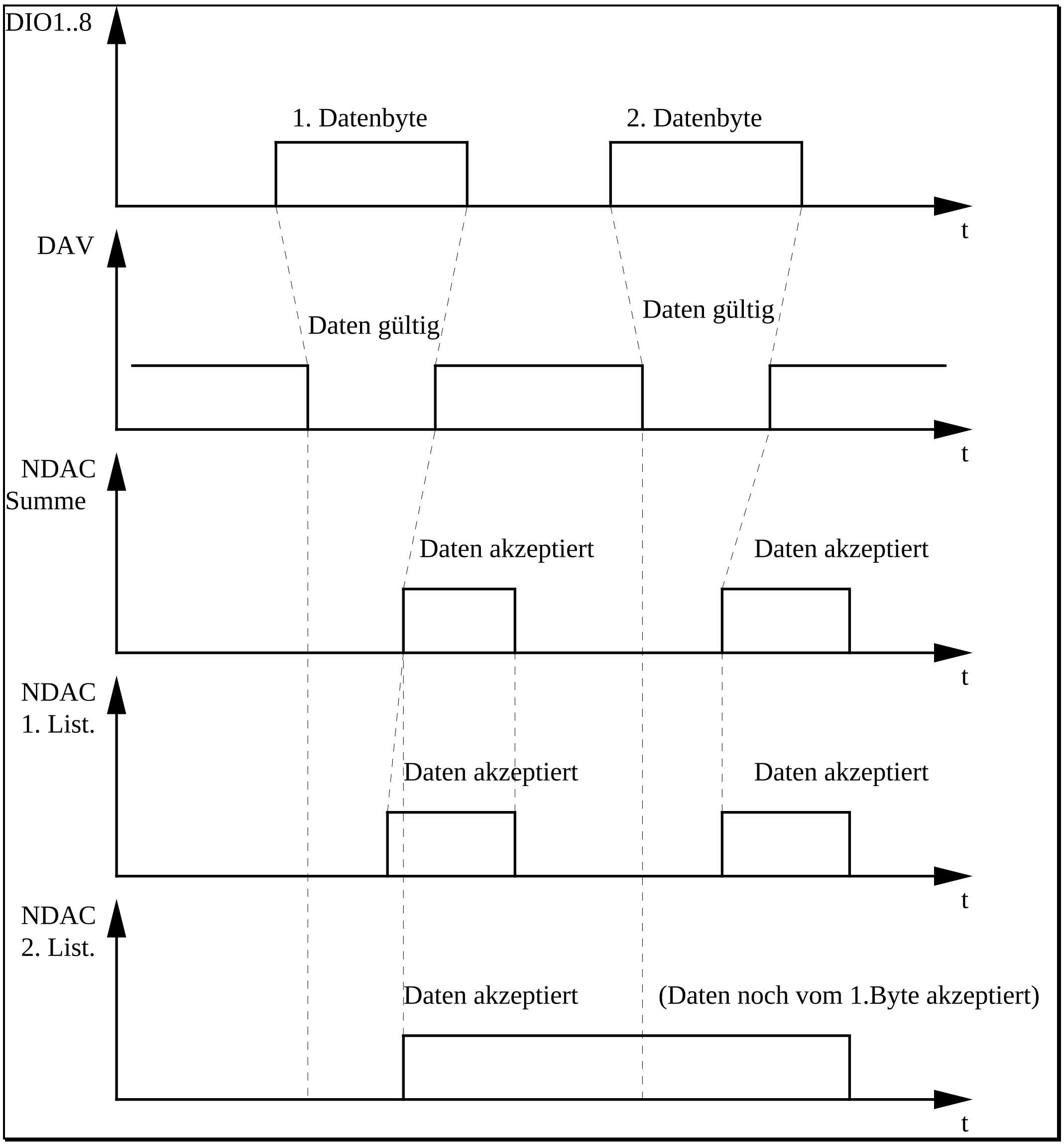

Abb. 5-13: Probleme bei Zweidrahthandshake

Der Talker hat alles richtig gemacht: Sein neues Datenbyte kommt genau dann auf den Bus, wenn das alte akzeptiert ist. DAV wird aktiv gesetzt, wenn NDAC wieder aktiv ist. Dies ist jedoch bereits der Fall, wenn eines der beiden NDAC-Signale aktiv low ist, und so kommt es zum Konflikt, wenn sich eines der Geräte wie in diesem Beispiel das 2. nicht sehr beeilt. Aus diesem Grund hat man eine zusätzliche Leitung, Not Ready For Data eingeführt und damit ein Drei-Draht-Handshake:

5.3.2.3 Drei-Draht-Handshake

Folgendes Flußdiagramm macht den Kontrollsignalfluß im Drei-Draht-Handshake-Verfahren deutlich, wobei die Listener-Signale immer die Wired-OR-Verknüpfung aller am Bus aktiven Listener darstellt:

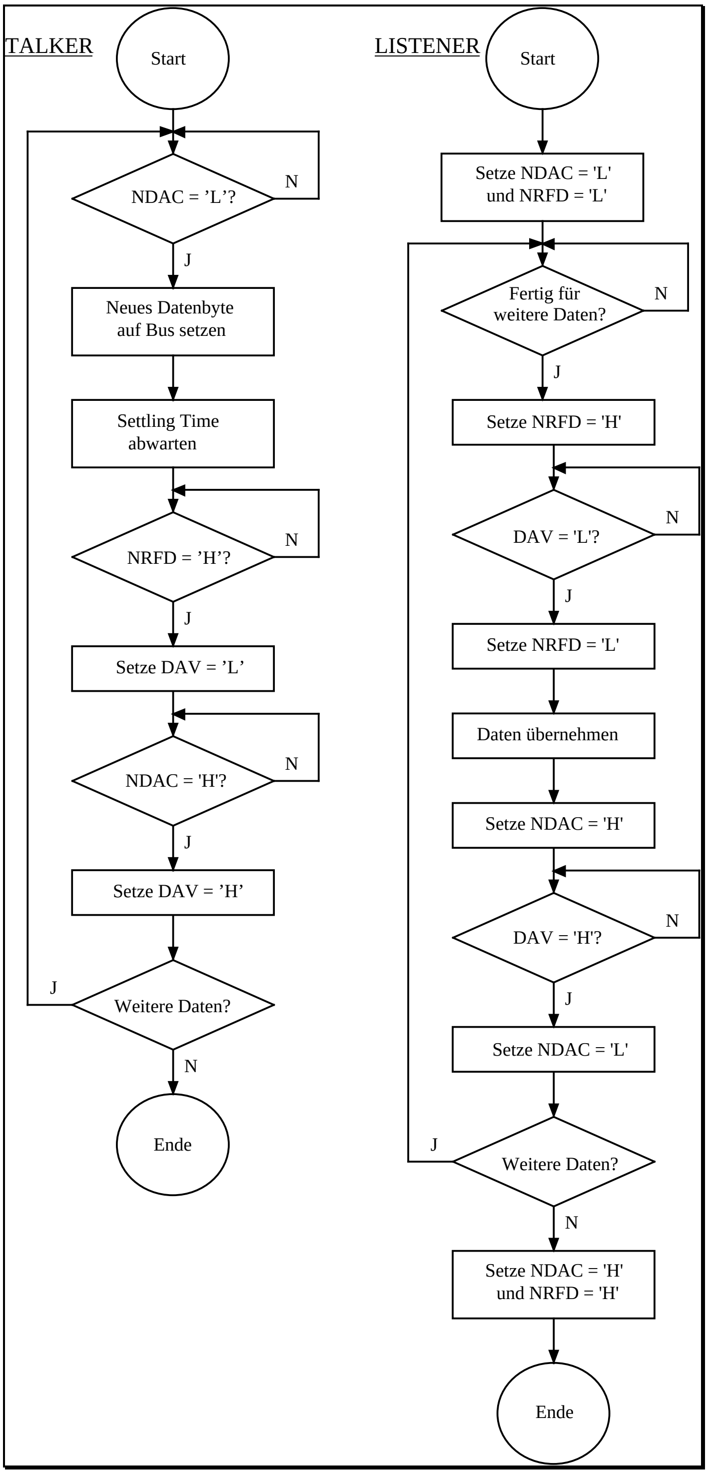

Abb. 5-14: Drei-Draht-Handshake

Der IEEE-488-Bus ist durch seine Signale selbststeuernd, stellt also den Prototyp eines asynchronen Busses mit Datenflußsteuerung auf Byte-Ebene dar. Die Busbedienung kann durch spezielle Prozessoren (NEC7210) und Treiberbausteine (MC3448) geschehen.

Kapitel 6: Feldbussysteme und industrielle Datenkommunikation

Der zweite Teil des Exkurses in die industrielle Datenkommunikation befaßt sich nun mit den Feldbussystemen und den darauf definierten Kommunikationen. Hierzu muß zunächst die spezielle Anwendung von Bussystemen in der Automatisierungstechnik in den Gesamtzusammenhang der Netze gestellt werden, bevor auf die speziellen Eigenheiten der Feldbussysteme eingegangen werden kann.

Literatur:

[5] Schnell, G.: Bussysteme in der Automatisierungstechnik. Vieweg-Verlag, Braunschweig, 1994. ISBN 3-528-06569-9

6.1 Gesamtübersicht zur Einordnung der Bussysteme

Eine bisher noch nicht dargestellte Klassifizierung der Bussysteme unterscheidet zunächst zwischen bitserieller und bitparalleler Übertragung, weiterhin zwischen Punkt-zu-Punkt-, Punkt-zu-Multipunkt-, unbalancierten Multipunkt-zu-Multipunkt- und balancierten Multipunkt-zu-Multipunkt-Verbindungen. Sie ist in Abb. 6-1 dargestellt.

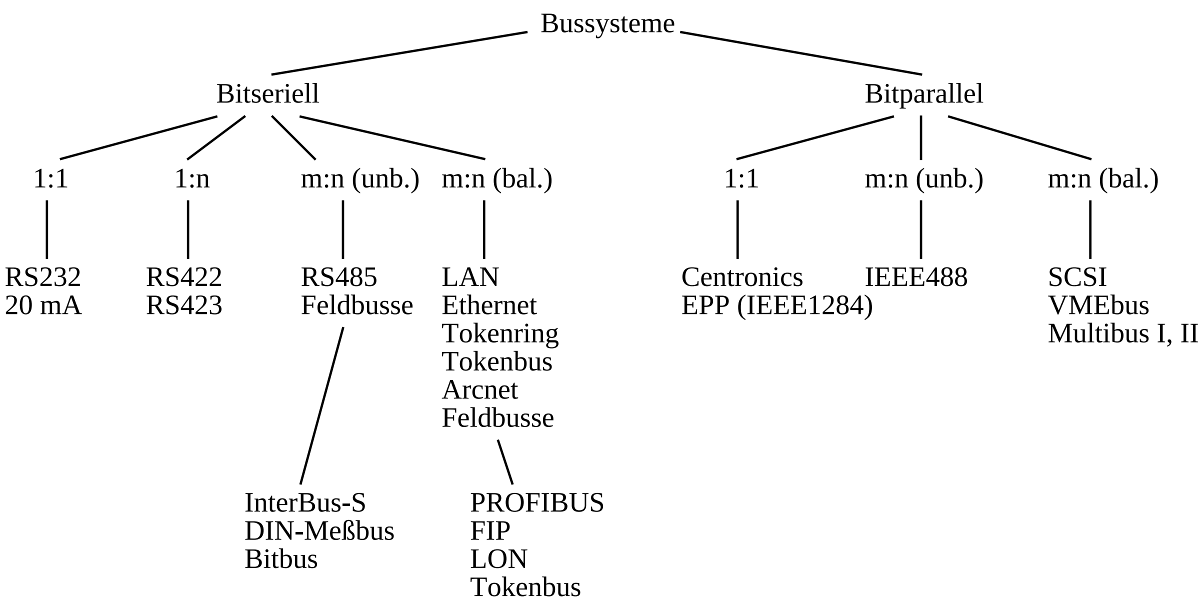

Abb. 6-1: Gesamtübersicht über Bussysteme

Die Feldbussysteme werden damit sehr eindeutig als bitserielle Multipunkt-zu-Multipunkt-Verbindungen klassifiziert.

6.2 Netzwerkhierarchien bei CIM

Die Wandlungen der Kommunikation in der Automatisierungstechnik können insbesondere am Beispiel der chemischen Industrie (1995) gezeigt werden, da dieser Industriezweig bereits seit Jahrzehnten eine automatische Überwachung seiner Produktionsprozesse integriert hat, andererseits aber besondere Anforderungen stellt.

Meßaufgaben wurden und werden in der Regel von analogen Sensoren durchgeführt, die ein physikalisches Signal in ein analoges elektrisches Signal umwandeln. Dieses Signal wird - entsprechend verstärkt - über eine analoge Leitung im Feld zu einem Rangierverteiler geführt, dort entsprechend umgesetzt, bis ein AD-Wandler in einem Rechnersystem die Daten in den digitalen Teil einnimmt. Da an einem Prozeß und dem entsprechenden Behälter durchaus 200 Sensoren/Aktoren beteiligt sind, ergeben sich entsprechend 200 Leitungspaare. Resultierend hieraus sind Zuverlässigkeits-, Wartungs- und Umstellungsprobleme, von den Problemen der Ungenauigkeit in der Übertragung (durch analoge Werte) einmal angesehen.

Die nächste Stufe war die Digitalisierung in der Nähe der Sensoren: Die Übertragungsungenauigkeiten entfallen, die Werte werden digitalisiert durch eine 20 mA-Stromschleife zur Weiterverarbeitung gesendet. Hierdurch sind noch keine Vorteile beim Anlagenbau, Umstellung und Wartung zu erwarten, der Aufwand bleibt identisch. Der digitalen Übertragung mittels der Stromschleife kann jedoch ein Wechselsignal überlagert werden, das zur Kommunikation mit dem Meßgerät dienen kann: Statusmeldungen, ggf. Einstellungskommandos etc. lassen sich übertragen. Der Verkabelungsaufwand bleibt jedoch vorhanden, ebenso die Wartung und der Umstellungsaufwand, da die Meßgeräte nur ein elektrisch kompatibles Interface besitzen, keineswegs jedoch eine genormte Befehlsschnittstelle.

In den kommenden Jahren wird der Einsatz von Feldbussystemen nun vorangetrieben werden. Man erwartet seitens der Industrie folgende Vorteile:

- Der Verkabelungsaufwand sinkt drastisch, da ein Bussystem eine Vielzahl von Teilnehmern (z.B. 32) erreicht.
- Die Wartung und der Umbau werden durch derartige Bussysteme unterstützt: Vorteilhaft wäre hier eine genormte Befehlsschnittstelle für alle Meßgeräte.

Warum dauert die Umstellung auf Feldbussysteme zumindestens in bestimmten Industriebereichen so lang? Die Antwort ist einfach: Die Anforderungen an ein Bussystem, das im rauhen industriellen Feld eingesetzt wird, sind sehr vielfältig, gleichwohl in den einzelnen Bereichen in unterschiedlicher Gewichtung:

- Echtzeitfähigkeit, deterministische Zugriffsverfahren
- hohe Übertragungsgeschwindigkeit, insbesondere bei Berücksichtigung der Anzahl von Teilnehmern am Bus, die ggf. Meldungen absetzen wollen
- Zuverlässigkeit und Fehlertoleranz

- geringe Störempfindlichkeit gegen elektromagnetische Einstrahlung (EMV)
- ‘Eigensicherheit’ in explosionsgefährdeten Bereichen
- Topologie so gestaltet, daß ein flächendeckendes Netz möglich wird
- Flexibilität im Sinne eines möglichen Umbaus etc.
- Wirtschaftlichkeit

Die Gesamtheit der Forderungen ist nicht erfüllbar, da einige bereits technisch widersprüchlich sind (Echtzeit, hohe Übertragungsraten und Fehlertoleranz), andere sich wirtschaftlich ausschließen. Die Lösung besteht aus einem hierarchischen Netzmodell, das je nach Ausprägung eingesetzt wird. Die in Abb. 6-2 gezeigten Hierarchieebenen sollten in diesem Zusammenhang nicht mit dem bekannten Layermodell nach ISO/OSI verwechselt werden: Hier sind Firmenhierarchien gemeint!

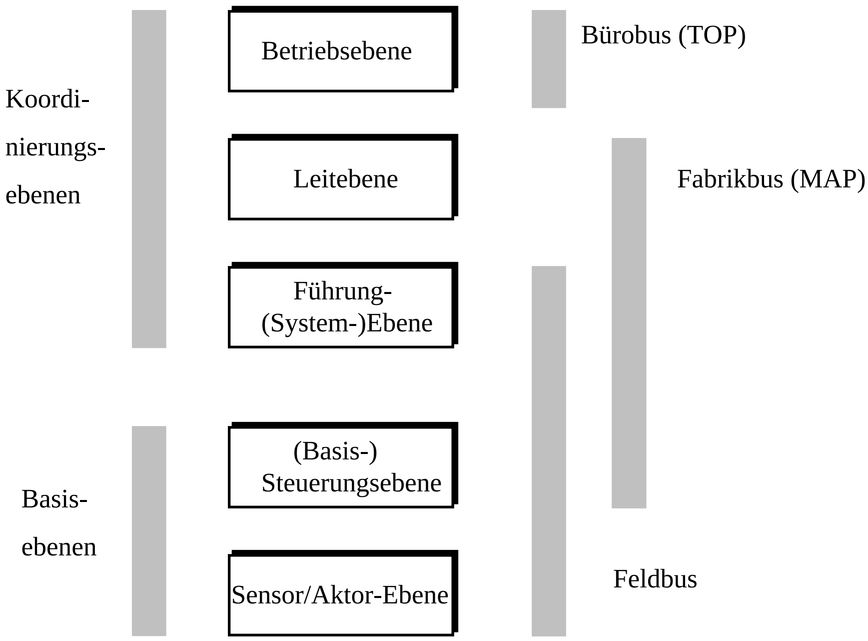

Abb. 6-2: Hierarchie-Ebenenmodell zur Einordnung von Bussystemen

Die Praxis zeigt, daß nun, daß die Realisierung eines solchen Mehrebenenbussystems nicht spontan - sozusagen am grünen Tisch - vonstatten geht, sondern daß sowohl ein allmähliches Wachstum in die Netzbereiche als Ersatz für die herkömmliche Informationsstruktur wie auch ein Herausbilden verschiedener Standards für den Feldbereich im Vordergrund stehen.

Die Ebenen im einzelnen:

6.2.1 Managementebene

Bestrebungen zur Standardisierung der Kommunikationen innerhalb einer Fabirk führten zu firmenabhängigen Projekten bei General Motors (Manufacturing Automation Protocol, MAP) und Boing (Technical and Office Protocol, TOP). Beide Protokolle wurden 1987 als MAP/TOP V3.0 durch ISO genormt, basierend auf einer Anzahl von OSI-Protokollen. Als Dienstaufrufschnittstelle fügte man die Manufacturing Message Specification (MMS, ISO 9506) hinzu, eine Schnittstelle, die innerhalb der Automatisierungsgerätevielfalt für größere Ordnung sorgt, da sie objektorientiert ist: MMS beruht auf dem Client-Server-Prinzip, jedes Gerät muß als Server agieren können, die Verwaltung und Manipulation bezieht sich auf MMS-Objekten. MMS ist insbesondere Nachrichten-orientiert definiert: Es werden weniger Daten transferiert als vielmehr Nachrichten zur Datenmanipulation.

Die Managementebene reicht damit über die eigentliche Bürokommunikation hinaus (TOP) in die einzelnen Teilprozessbereiche und soll der (zeitentkoppelten) Steuerung des Betriebsablaufs dienen.

6.2.2 Prozeßleitebene

Die ***Prozeßleitebene*** war in ihrer ursprünglichen Definition ein Rechner mit unintelligenter Peripherie-Hardware: Die Zentrale wußte quasi alles, die Hardware an der Peripherie konnte eine analoge Elektronik mit entsprechend analoger Übertragung sein.

Die zweite Form der Weiterentwicklung bestand in der Einführung einer weiteren Schicht von ***Mikroprozeßrechnern***, die die Steuerung der Prozeßperipherie nunmehr übernahmen und mit dem Prozeßleitrechner kommunizierten. Die resultierenden Konsequenzen waren die Entlastung der Rechenkapazität des Prozeßleitrechners sowie verringerter Verdrahtungsaufwand.

Die nächste Stufe umfaßt die Ausstattung der Prozeßperipherie mit eigener Intelligenz und stellt damit die Voraussetzung dar, um die Peripheriegeräte, die am Herstellungsprozeß direkt sitzen, miteinander und mit den höheren Ebenen zu vernetzen: Das Feldbussystem ist begründet.

Abb. 6-3 zeigt die verschiedenen Entwicklungsstufen der Prozeßebene:

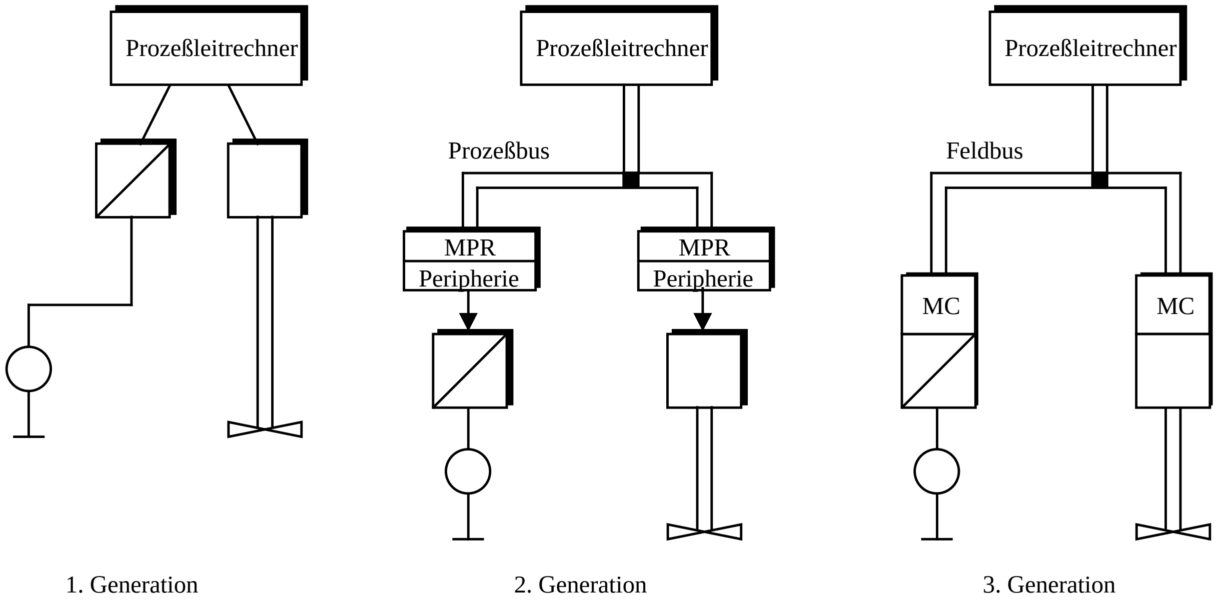

Abb. 6-3: Entwicklung der Automatisierungssysteme

6.2.3 Feldebene

Aus der veränderten Anwendung der Kommunikation – insbesondere sei die Mikroprozessor-basierte Steuerung am Prozeßaktor/-sensor erwähnt – resultieren folgende Anforderungen an einen Feldbus:

- Die *Topologie*: Die Entfernungen in einem Feldbussystem können wenige Meter bis zu einigen Kilometern betragen. Da bei einem industriellen Prozeß die lokale Verteilung der Geräte nicht frei wählbar ist, müssen auch flächendeckende Applikationen vernetzbar sein: Die vorgeschriebene Topologie darf somit nicht hinderlich sein, Flexibilität bei der Verdrahtung ist gefordert.

- Das *Zeitverhalten*: Der Übergang der Prozeßsteuerung von Prozeßleitrechnern auf Mikroprozeßrechner und weiter auf die peripheren Komponenten selbst bedeutet auch, daß Echtzeitforderungen, wie sie sehr häufig auftreten, genau dann auf die Kommunikation projeziert werden, wenn die periphere Komponente für ein geordnetes Verhalten nicht mehr allein verantwortlich ist – was in der Regel der Fall sein dürfte.

  Das Feldbussystem muß also echtzeitfähig sein: Die gesamte Reaktionszeit setzt sich aus der Erkennungszeit eines relevanten Zustands in der Peripherie, Kommunikation zum Leitrechner oder zu anderen Komponenten, Reaktionszeit in diesen Rechnern und erneuter Kommunikation zu Aktoren zusammen: Das Netz spielt eine doppelte Rolle.

  Die Projektierbarkeit der maximalen Reaktionszeit im System – diese Zeit kann je nach Anforderungen im Millisekunden- bis in den Sekundenbereich liegen, muß aber mit einer oberen Grenze belegbar sein – ist damit eine notwendige Forderung an Feldbussysteme, die mit dem Zugriffsverfahren wesentlich bestimmt wird. Hier kommen deterministische Zugriffsverfahren (Token, Polling, äquidistante Zugriffsrechtverteilung etc.) ebenso zum Einsatz wie minimierte Datagramme.

- Die *Zuverlässigkeit* wird zunächst einmal verschlechtert, da alle Kommunikationen über einen seriellen Draht laufen sollen, was einer Verringerung der voneinander unabhängigen Übertragungskanäle gleichkommt. Zusätzlich wird die Anzahl der Steuereinheiten erhöht.

  Um diesem Nachteil entgegenzuwirken, müssen verschiedene Maßnahmen am Feldbus ergriffen werden. Die Zuverlässigkeit läßt sich dabei durch EMV (Elektromagnetische Verträglichkeit)-Maßnahmen ebenso steigern wie durch Codesicherungsverfahren, die die Wahrscheinlichkeit unerkannter Fehler stark reduzieren. Diese Methoden betreffen die Übertragung selbst, während andere Maßnahmen, beispielsweise die automatische Dekonfigurierung eines defekten Knotens am Bus, die Nichtbeeinflussung der übrigen Kanäle betreffen.

- Die Anforderungen an die *Stromversorgung* bei einem Feldbussystem können außerordentlich unterschiedlicher Natur sein: Einerseits muß die Hilfsenergiequelle zuverlässig, häufig sogar fehlertolerant oder unterbrechungsfrei arbeiten, andererseits kann unter bestimmten Betriebsbedingungen die Energieversorgung durch den Feldbus notwendig sein. Dies ist bei sehr geringen Energieanforderungen sogar via Lichtwellenleiter möglich!

- Die *Einsatzbedingungen*, unter denen ein Feldbus arbeiten muß, können sehr weit variieren; an dieser Stelle seien besonders
  - Temperatur- und Feuchtebereiche
  - Einsatz innen und außen
  - Staubige Umgebung
  - mechanische Schwingungen, Stöße etc.
  - Schutzgrade (Explosionsschutz), Schiffs- und Weltraumeinsatz etc.

  genannt. Diese Einsatzbedingungen erfordern im Einzelfall gesonderte Projektierungen; einzelne Bereiche sind in allgemeiner Normung befindlich, beispielsweise Feldbusse für explosionsgefährdete Bereiche.

- Die *Flexibilität* eines Feldbussystems wird natürlich auch am Einzelfall gemessen. Hierunter fallen die Einbindung zusätzlicher Busteilnehmer, möglichst sogar im Betrieb ebenso wie die problemlose Kopplung zu anderen Netzen

- Einer der wichtigsten Punkte bei Feldbussen ist die Wirtschaftlichkeit: Ca 10 .. 20 % der Anschaffungskosten eines Automatisierungsgeräts dürfen durch die Busanbindung bedingt sein, ansonsten wird das Gerät eher aus wirtschaftlichen Gesichtspunkten abgelehnt werden. Die Schaffung von Busstandards hilft an dieser Stelle natürlich, die Netzkosten zu senken.

Die Schlußfolgerung aus den sehr unterschiedlichen Anforderungsprofilen besteht in der Einführung einer weiteren Ebene, der Sensor-Aktor-Ebene.

6.2.4 Die Sensor-Aktor-Ebene

Nachrichten im Sensor/Aktor-Rahmen bestehen aus binären wie analogen Signalen, die mit den besonderen Bedingungen der 'rohen Aktualität' ausgestattet sind. Dies bedeutet, daß die insbesondere aus den Sensoren stammenden Daten im wesentlichen noch nicht verarbeitet (sprich ausgewertet) sind, während die Aktorsignale gerade eine Aktion hervorrufen sollen.

Die Konsequenzen sind Anforderungen wie

- Echtzeitverhalten (im Millisekundenrahmen)
- flächendeckende Topologie am Prozeß selbst
- Übertragungsmedien der rauhen Umgebung entsprechend (EMV, Staub etc.)
- Besondere Wirtschaftlichkeit aufgrund der Vielzahl der Anschlüsse,

die alle durch das sehr prozeßnahe System der Aktor-Sensor-Busse erfüllt sein sollen.

Die Übertragung der Signale zwischen Aktor/Sensor und Feldrechner basiert auf diversen Wechselwirkungen und Meßverfahren. Hierzu zählen das Verfahren mittels Schieberegister (Parallel-Seriell-Parallel-Wandlung) bei Punkt-zu-Punkt-Verbindungen, die Ausnutzung akustoelektronischer Oberflächenwellen (AOW), die Messung von Implusreflexionen an offenen Kabeln, ein Meßverfahren aus der Hochfrequenztechnik, sowie vereinfachte Varianten von Feldbussen, die nunmehr nicht mehr ein balanciertes System mit vielen möglichen Mastern darstellen, sondern mit einem Master ausgestattet sind.

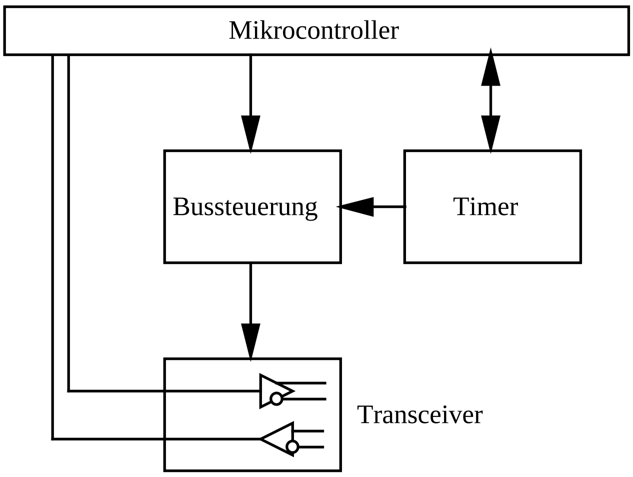

Abb. 6-4: Aufbau Master-Station am Aktor/Sensor-Bus

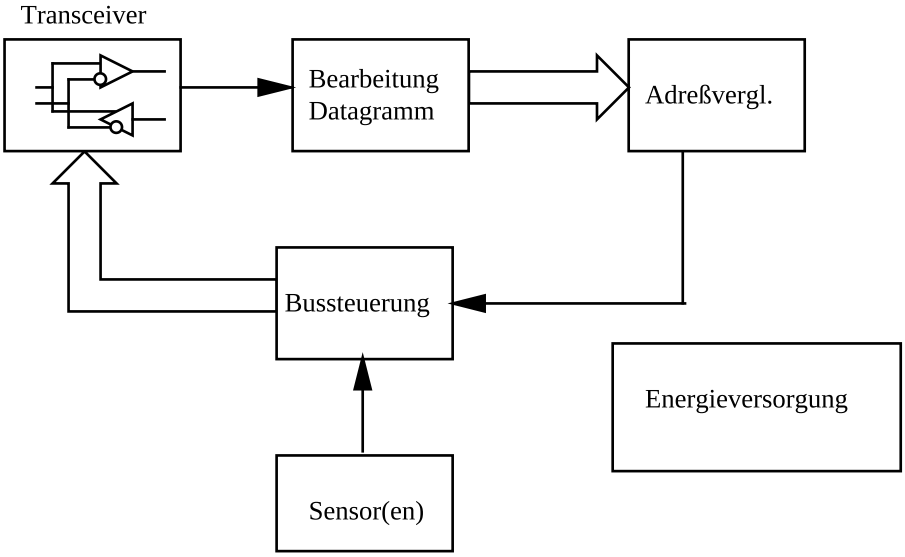

Abb. 6-5: Aufbau Slave-Station am Aktor/Sensor-Bus

Die Definition eines solchen Busses besteht in der Abhängigkeit der Slave-Teilnehmer von dem Master: Jede Kommunikation wird vom Master gestartet, der anschließend seine Sender abschaltet, falls er eine Antwort von dem Adressaten erwartet. Die Slaves am Bus decodieren die gesendete Nachricht auf die innenliegende Adresse und beantworten das Datagramm, falls sie adressiert wurden. Dieses System funktioniert natürlich nur, wenn der angesprochene Teilnehmer am Bus auch betriebsbereit ist. Um keine Deadlocksituation auftreten zu lassen, wird in der Regel im Master eine Zeitüberwachung integriert.

6.2.5 Schlußfolgerungen und Entwicklungstendenzen

Bislang konnte sich am Markt der Feldbussysteme (einschließlich der Aktor-Sensor-Busse) keine einheitliche Norm durchsetzen, was auch nicht zu erwarten ist, da die Anforderungen an die Busse zu unterschiedlich strukturiert sind. Allerdings ist die Tendenz unverkennbar, wenige Systeme zu etablieren und alle anzuschließenden Geräte darauf auszurichten, um dem Anwender eine größtmögliche Zukunftsicherheit beim Kauf gewährleisten zu können. Aus diesem Grund existieren sowohl internationale Normungsgremien als auch firmenübergreifende Konsortien zur Förderung. Einige der Bussysteme werden im folgenden Unterkapitel näher betrachtet.

6.3 Beispiele genormter Feld- und Sensor-Aktor-Busse

6.3.1 ASI

Ein Aktor-Sensor-Bus soll den bisherigen Anschluß eines Sensors oder Aktors, die eine direkte Verkabelung mit der Steuereinheit darstellt, durch einen einfachen, digitalen Bus ersetzen. Hierbei stehen zwei Dinge im Vordergrund: Zum einen arbeiten im Durchschnitt bereits über 80% aller Sensoren und Aktoren binär, also mit kleinsten Informationseinheiten 0 und 1, so daß für die einfachste Ausführung des Anschlusses keine AD- oder

DA-Wandler benötigt werden. Andererseits stellt die Versorgung der Peripherie mit einer Hilfsenergie eine sehr wesentliche Zusatzforderung dar.

Im ASI (Aktuator-Sensor-Interface) wird daher abweichend von RS485 eine digitale Übertragung definiert:

- Das Übertragungsmedium besteht in einem flexiblen Stromkabel mit 1,5 mm² Querschnitt (24 Volt, 2 Ampere als Hilfsenergie) oder in einem ASI-spezifischen Flachbandkabel. Die mechanische Ankopplung an dieses Flachbandkabel erfolgt durch Aufklemmen (mit Schraubbefestigung), ist also an beliebiger Stelle möglich.

- Die Anforderungen an das Modulationsverfahren sind durch die Überlagerung mit der Gleichstromenergie relativ aufwendig: Das Modulationsverfahren muß gleichstromfrei sein (Übertrager), einfach erzeugbar und schmalbandig (hohe Dämpfung durch gewähltes Übertragungskabel). Hier wird die Bi-Phase-Codierung gewählt, wobei die logischen Zustände in Zusammenhang mit einer Leitungsinduktivität zu Stromerhöhungen und -erniedrigungen führt: Master und Slave können gleichermaßen einfach auf dem Energieversorgungskabel senden. Die Decodierung zwischen 0 und 1 erfolgt durch Taktrückgewinnung. Die Bruttoübertragungsrate beträgt 167 kBaud entsprechend einer Bitzeit von 6 µs.

- Das Buszugriffsverfahren simuliert quasi die früheren Punkt-zu-Punkt-Verbindungen zwischen den jetzigen Busteilnehmern: Der Master sendet ein Datagramm mit der Slaveadresse, der Slave antwortet oder entnimmt dem Datagramm die Daten innerhalb einer definierten Zeit. Dies erfolgt in zyklischem Polling, wobei die Zykluszeit von der Anzahl der Busteilnehmer abhängt (maximal 5 ms, bedingt durch 25 Bits, 6 µs pro Bit, 33 Datagramme pro Zyklus). Nachteilig ist hier allerdings, daß die Slaves keinen Interrupt beim Master anfordern können, sondern auf den Zyklus warten müssen!

- Das Datenformat entspricht dem Aufbau in Abb. 6-6. Die Nettoübertragungsrate entspricht dann einem Anteil von 31%, sprich 53,3 kBaud – und stellt damit einen sehr guten Wert dar.

Masterdatagramm:

0	S	A	A	A	A	A	I	I	I	I	I	P	1

Slavedatagramm:

0	I	I	I	I	P	1

S: Steuerbit
A: Adressbits für die Slaveadresse
I: Informationsbits
P: Paritätsbit

Abb. 6-6: ASI-Datenformat

Für das Datagramm sind einige Aufrufe definiert: Daten- und Parameterübertragung, Einstellung und Änderung der Slaveadresse sowie 5 Aufrufe zur Identifikation und zur Statusabfrage summieren sich auf 9 Busaufrufe des Masters, während die Slaves immer nur antworten.

- Die Sicherung der Datenübertragung erfolgt – aus Effektivitätsgründen – nur durch eine einfache Parität, entspricht also einem Code der Hammingdistanz 2; eine Hammingdistanz von 4 hätte den Code sehr stark aufgebläht (die Nutzdaten um +64% beim Masterdatagramm, beim Slave um +100%), so daß hier darauf verzichtet werden mußte.

  Als Ausgleich für diese geringe Sicherung wird beim Sender/Empfänger in der Schicht 1 ein höherer Aufwand als sonst üblich betrieben: Mehrfachabtastungen schaffen an dieser Stelle ebenso fehlersicherere Ergebnisse wie Prüfung der einwandfreien Codeübertragung mit komplettem Verwerfen des Datagramms, falls dies Anzeichen für Störungen aufweist.

- Die Topologie des Netzes ist frei wählbar: Linienförmig, Stichleitungen, alle Varianten sind erlaubt.

Der Betrieb des Aktuator-Sensor-Interfaces wird zumeist mit Hilfe von ASICs durchgeführt: In einer Liste werden die am Bus erwarteten Teilnehmer mit ihren Konfigurationen (z.B. ‘Standardaktor’, ...) eingetragen, so daß der Master ständig versucht, diesen Bus entsprechend zu betreiben. Hierzu wird pro Datenzyklus (31, bei Verwendung der Slaveadresse 0 auch 32 Teilnehmer) ein 33. Datagramm eingefügt, in dem der Master entsprechend abfragt.

Die Nettodatenübertragung beträgt 4 Bits, in beiden Richtungen; das 5. Masterbit ist für die besonderen Konfigurationsmodi bestimmt. Von Seiten der Masterstation wird das Datum nun in den ASIC eingeschrieben, der dies ständig überträgt. Die Folge: Zwei aufeinanderfolgende Datenhalbbytes identischen Inhalts sind nicht zu unterscheiden, daher muß in der darüberliegenden Software für entsprechende Merkmale gesorgt werden. Der Slave antwortet ebenfalls immer mit den Bits, die ihm eingetragen werden!

Die Konfiguration am Bus kann so erfolgen, daß es Buskommandos zum Wechsel der Slaveadresse gibt. Ein noch nicht konfigurierter Slave wird die Adresse 0 haben, die dann auf die gewünschte Adresse umgetragen wird. Daher wird die Adresse 0 gewöhnlich nicht besetzt, um beispielsweise nachträgliche Konfigurationen oder auch Austausch von Slaves durchführen zu können.

6.3.2 Der Bitbus

Der Bitbus, seit 1984 existent, unter IEEE 1118 genormt, wurde von der Firma Intel konzipiert und findet in der unteren und mittleren Ebene der Fabrikautomation Anwendung.

Folgende Betriebsmodi werden vorgeschlagen:

Betriebsart	Baudrate	max. Länge	Slaves zw. Repeatern	Repeater
synchron	2 MBaud	30 m	max. 28	0
asynchron	62,5 kBaud 375 kBaud	1200 m 300 m	max. 28 max. 28	max. 10 max. 2

Tabelle 6-1: Betriebsmodi Bitbus

Die weiteren Eigenschaften des Bitbus' im Detail:

- Das Übertragungsmedium besteht aus einem geschirmten, verdrillten Zweidrahtkabel (STP). Meistens wird ein höheradriges Kabel verlegt, um beispielsweise die synchronen Übertragungen zu ermöglichen. RS-485 arbeitet auf diesen Kabelpaaren mit Spannungsdifferenzen (± 5 Volt). Die Codierung erfolgt im NRZI-Format und gestattet daher prinzipiell die Taktrückgewinnung (durch die Einsetzung von Bits).

  Der Bitbus kann prinzipiell 250 Geräte adressieren; hierzu wird die Adresse 255 an den Busmaster vergeben, 251 bis 254 bleiben ungenutzt. Da RS-485 in seiner elektrischen Auslegung nur 32 Teilnehmer unterstützt, sind Repeater für einen kompletten Bitbus mit mehr als 28 Teilnehmern notwendig.

- Die Arbitrierung des Übertragungskanals erfolgt durch den Busmaster. Hier findet also kein dem Ethernet vergleichbares Zugriffsverfahren statt, sondern der Master muß wie beim ASI die einzelnen Slaves zyklisch abfragen, um jedem die Gelegenheit zur Kommunikation zu geben.

- Das Datenformat des Bitbus' besteht aus dem SDLC-(Synchronous Data Link Control)- Format, einer Vorgängervariante des in Kap. 7 besprochenen HDLC-Formats. Hierzu werden die Daten in Pakete verpackt, die aus Startblock, Informationsblock, Datenblock, CRC-Wort und Endblock bestehen (Abb. 6-7). Start- und Endezeichen bestehen aus dem Hexwert 7Eh. Dieses Zeichen ist im späteren Bitstrom nicht mehr vorhanden, da nach 5 aufeinanderfolgenden '1' automatisch eine '0' eingeschoben wird (Bit-Stuffing, Kap. 3). Die Checksumme wird durch

  $$CRC = x^{16} + x^{12} + x^{5} + 1$$

  berechnet.

1	2	7	0 .. 43	2	1	Anzahl Bytes
Start	Control	Header	Datenfeld	CRC	Ende	

Länge Header + Datenfeld
Nachrichtentyp
Adresse des Slaves
Kennung für Kommando/Antwort usw.

Abb. 6-7: Bitbus Datagrammformat gem. IEEE 1118

- Die Topologie des Netzes ist der Eigenschaft von RS485 entsprechend ausschließlich linienförmig.

6.3.3 Der Profibus

Die Abkürzung PROFIBUS steht für Process Field Bus. Dieses System findet allein deshalb eine breite Akzeptanz bei Anwendern, da es von Beginn an als herstellerunabhängiges Konzept entworfen worden ist und damit die Eigenschaften offen und firmenneutral trägt. Die Normung liegt in DIN 19245 Teil 1 und 2 vor.

Dieser Weg der Normung hat allerdings auch einen Nachteil: Da die Schaffung des Standards durch Kooperation vieler Firmen vorangetrieben wurde, ist die Norm sehr langsam entstanden und in gewisser Weise ‘allumfassend’ – ein typisch europäisches Problem. Das allumfassende wird in dem Zielgebiet des PROFIBUS sichtbar: Automatisierungsgeräte der untersten Feldebene wie Sensoren/Aktoren bis hin zu Prozeßsteuerungen wie Anbindung an die bestehenden Netze der Prozeßleitebene spannen den gesamten Einsatzbereich von PROFIBUS.

Im untersten Bereich muß durch kurze Reaktionszeiten dafür gesorgt werden, daß insbesondere bei Sensoren keine Daten verlorengehen. Die Buszykluszeit muß daher sehr kurz sein, kurz genug, um jeden Sensor entsprechend häufig abfragen zu können.

Weiterhin soll der PROFIBUS auch für Echtzeitsteuerungen eingesetzt werden; dies bedeutet, daß am Bus ein deterministisches Verhalten vorliegt, das es erlaubt, die maximal verstreichende Zeit zu berechnen, die eine Meldung zur Übertragung braucht (und damit gekoppelt das Gesamtsystem zur Reaktion).

Spezielle Anpassungen des PROFIBUS sind für sehr rauhe Umgebungen mit hoher elektromagnetischer Störung und für explosionsgefährdete Bereiche gefordert und definiert.

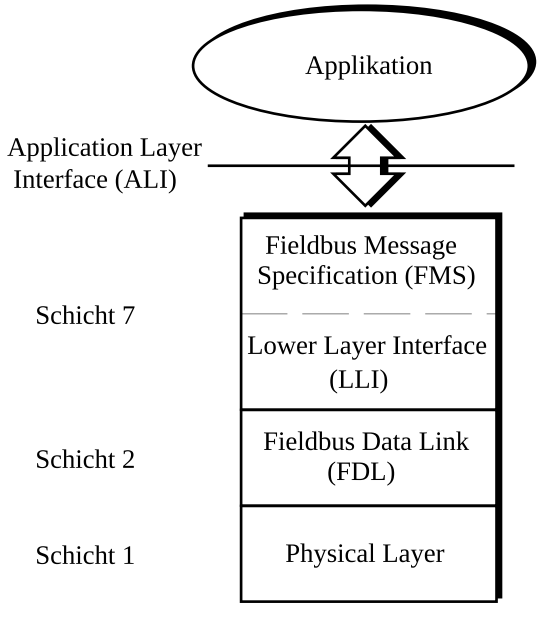

Abb. 6-8: PROFIBUS Kommunikationsmodell

Diese Sammlung von Bedingungen lassen sich durch den PROFIBUS in folgender Weise erfüllen, indem eine flexible Sammlung von Definitionen für die Layer 1, 2 und 7 des OSI-Modells angeboten werden. Diese Layer sind in Feldbussystemen häufig definiert, wobei die vielleicht fehlenden Teile der zwischenliegenden Ebenen in die Applikationsschicht (oder die Applikation) geschoben werden; diese Einschränkungen resultieren in eine höhere Effizienz des Systems.

6.3.3.1 Die Schicht 1: Übertragungstechnik

Die physikalische Schnittstelle und das damit verbundene Übertragungsmedium werden im wesentlichen durch die Einsatzbedingungen in der industriellen Umgebung geprägt. Die PROFIBUS-Norm bezieht sich dabei auf RS-485, wobei als spezielle Eigenschaften spezifiziert werden:

- Die maximale Anzahl der Teilnehmer je Bussegment beträgt 32 (RS-485). Dies wird im Ex-Bereich weiter vermindert, da die maximalen Ströme begrenzt werden müssen. Die Anzahl der Teilnehmer läßt sich durch weitere Segmente und Repeater vergrößern, maximal 3 Repeater zwischen 2 kommunizierenden Geräten.

- Als Steckverbinder ist eine 9-polige SubMin-D-Buchse spezifiziert.

- Als Standard-Übertragungsgeschwindigkeiten sind definiert: 9,6 / 19,2 / 93,75 / 187,5 / 500 kBaud, im Ex-Bereich 31,25 kBaud.

- Die maximale Leitungslänge je Bussegment liegt zwischen 200 m und 1200 m, je nach Übertragungsgeschwindigkeit.

- An beiden Leitungsenden eines Bussegments ist ein Leitungsabschluß (Abb. 6-9) vorzusehen.

- Die Topologie des PROFIBUS ist linienförmig mit kurzen Stichleitungen an den Anschlüssen.

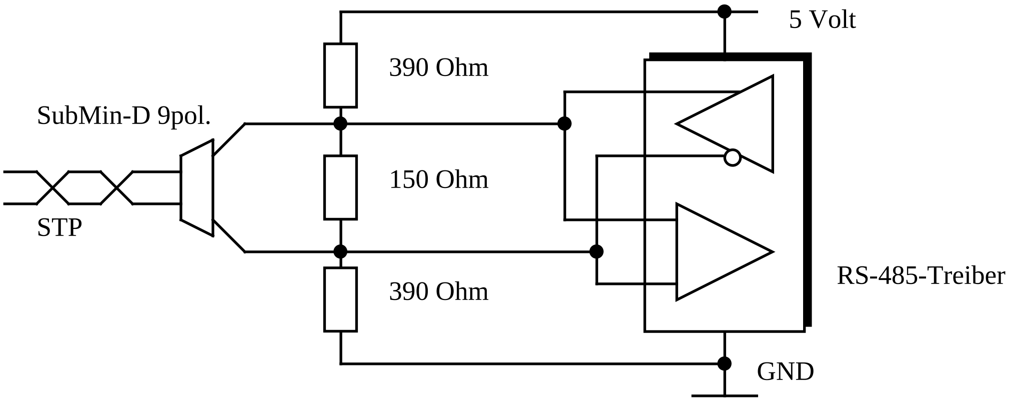

Abb. 6-9: RS-485 Schnittstelle mit Leitungsabschluß

- Speziell für den Einsatz in extrem gestörter Umgebung (elektrische Antriebe) oder zur Verkabelung in sternförmiger Architektur ist die Verwendung von Lichtwellenleiter spezifiziert.

- Das Übertragungsverfahren auf der RS-485 Schnittstelle ist halbduplex, asynchron. Als Bitcodierung kommt NRZ-Code zum Einsatz, im Ex-Bereich Manchester (Bi-Phase-Mark)

6.3.3.2 Datenübertragungsschicht

Die zweite Schicht des OSI-Übertragungsmodells beinhaltet die Steuerung des Buszugriffs, sowie die Abwicklung der Schicht-2-Dienste für Datenübertragung (Fieldbus Data Link, FDL) sowie für die Verwaltung (Fieldbus Management, FMA).

Das bei PROFIBUS verwendete Buszugriffsverfahren ist hybrider Natur: Jeder Knoten am Bus kann Master werden und hiermit im Master/Slave-Verfahren auf alle anderen Busteilnehmer zugreifen. Dieses Verfahren ist nicht echtzeitfähig, da es ausschließlich auf die Belange eines Teilnehmers zugeschnitten ist.

Die Masterberechtigung - der ***TOKEN*** - wird in einem Zeitscheibenverfahren weitergegeben, wobei die Zeitscheibe parametrierbar ist und somit Echtzeitbedingungen in weiten Grenzen einzuhalten sind. Die Übertragung dieses Tokens ist garantiert - eine wesentliche Voraussetzung für die Echtzeitfähigkeit.

PROFIBUS bietet vier Datenübertragungsdienste an:

- ***SDN***, Send Data with No Acknowledge, ein Dienst, der meist für Rundfunkdienste (Broadcast, Multicast) genutzt wird
- ***SDA***, Send Data with Acknowledge, Datensendung mit Quittung
- ***SRD***, Send and Request Data, Datensendung mit Daten-Rückantwort
- ***CSRD***, Cyclic Send and Request Data, zyklischer SRD

Während die ersten Dienste einmalig angefordert bzw. angeboten werden, ist letzterer ein zyklischer Dienst, der insbesondere im Sensor/Aktor-Bereich Anwendung findet.

Das Datagramm hat drei unterschiedliche Formate, bei denen zwischen

- fester Informationsfeldlänge ohne Datenfeld (z.B. Quittierungen)
- fester Informationsfeldlänge mit Datenfeld (8 Bytes)
- variabler Informationsfeldlänge (Datenfeld maximal 246 Bytes)

unterschieden wird.

6.3.3.3 Anwendungsschicht

Beim PROFIBUS werden die Schichten 3 bis 6 des ISO/OSI-Schichtenmodells nicht ausgefüllt; als Ausgleich dafür wird in Schicht 7, der Anwendungsschicht ein Anpassungsinterface, das ***Lower Layer Interface*** (LLI) hinzugefügt. Alle Dienst, die Anwendungen zur Verfügung gestellt werden müssen, sind innerhalb dieses Interfaces vereint.

Als Kommunikationsmodell wird ein aus der Manufacturing Message Specification (MMS) hergeleitetes Modell, die ***Fieldbus Message Specification*** (FMS) eingesetzt. Hierzu werden - in 'klassischer' objektorientierter Weise - reale Prozeßobjekte wie Temperaturwerte, Schaltzustände auf Kommunikationsobjekte abgebildet. Hierbei sind einige Objekte im Profibussystem bereits festgelegt und damit unveränderlich (implizite Objekte), während explizite Objekte mit einer Objektbeschreibung unter Festlegung der Attribute definierbar sind. Dieses System dient insbesondere der Austauschbarkeit von PROFIBUS-Produkten verschiedener Hersteller, auch auf der Schicht-7-Ebene.

Für den eigentlichen Ablauf werden weiterhin Kommunikationslisten geführt sowie diverse Dienste angeboten. Dies hat jedoch zur Konsequenz, daß die eigentlichen Kommunikation nicht besonders schnell ablaufen kann: Die Vielzahl der Dienste und damit die Bearbeitung in den einzelnen Schichten sowie die Kennzeichnung der geforderten Dienste innerhalb der Kommunikationspakete erniedrigen die Nettokommunikationsrate so drastisch, daß bei entsprechenden Geschwindigkeitsanforderungen die Netzübertragungsrate (Bitfolgerate) stark erhöht werden muß sowie die Anforderungen an die Knotenrechner sehr drastisch sind.

6.3.4 Der InterBus-S

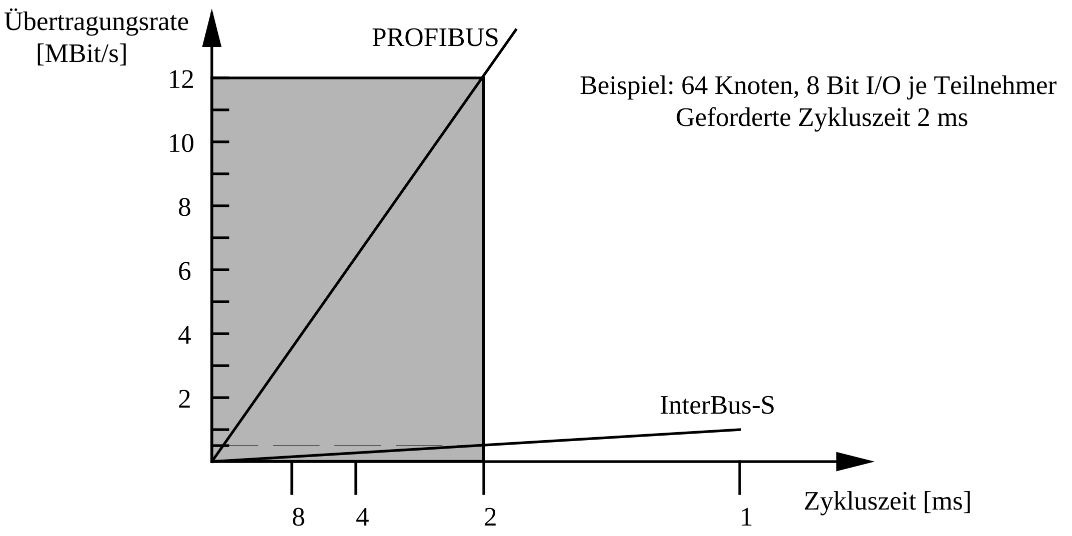

Abb. 6-10: Vergleich der Übertragungsraten InterBus-S und PROFIBUS

Der InterBus-S wurde ausschließlich für den Einsatz in der Sensorik/Aktorik konzipiert, demnach unterhalb der Steuerungsebene. Dies bewirkt einige starke Vereinfachungen in der Architektur und in dem Protokoll, so daß die resultierende Nettodatenrate drastisch ansteigt. Abb. 6-10 zeigt den Vergleich zweier Bussysteme in eigentlich unzulässiger Weise: PROFIBUS und InterBus-S sind für unterschiedliche Anwendungen konzipiert, werden hier aber anhand ihrer Nettoübertragungsdauer miteinander verglichen. Die hierbei geforderte Nettoübertragungsrate, 512 Nettobits pro 2 ms, beträgt 256 kBit/s, wozu beim InterBus-S eine Bruttoübertragungsrate von 512 kBit/s, beim Profibus jedoch 12 Mbit/s notwendig sind.

Die Topologie des InterBus-S besteht aus einem Ringsystem aller Teilnehmer, d.h., alle Stationen sind an einem in sich geschlossenen Ring angeschlossen. Subsysteme lassen sich leicht dadurch ausbilden, das an spezialisierten Anschlüssen, Busklemmen genannt, quasi Seitenwege eingeführt werden. Diese Busklemmen besitzen zumeist eine intelligente Funktionalität, die Fehler auf diesen Peripheriebussen erkennt und ihn dann abschalten kann. Abb. 6-11 zeigt eine beispielhafte Topologie des InterBus-S.

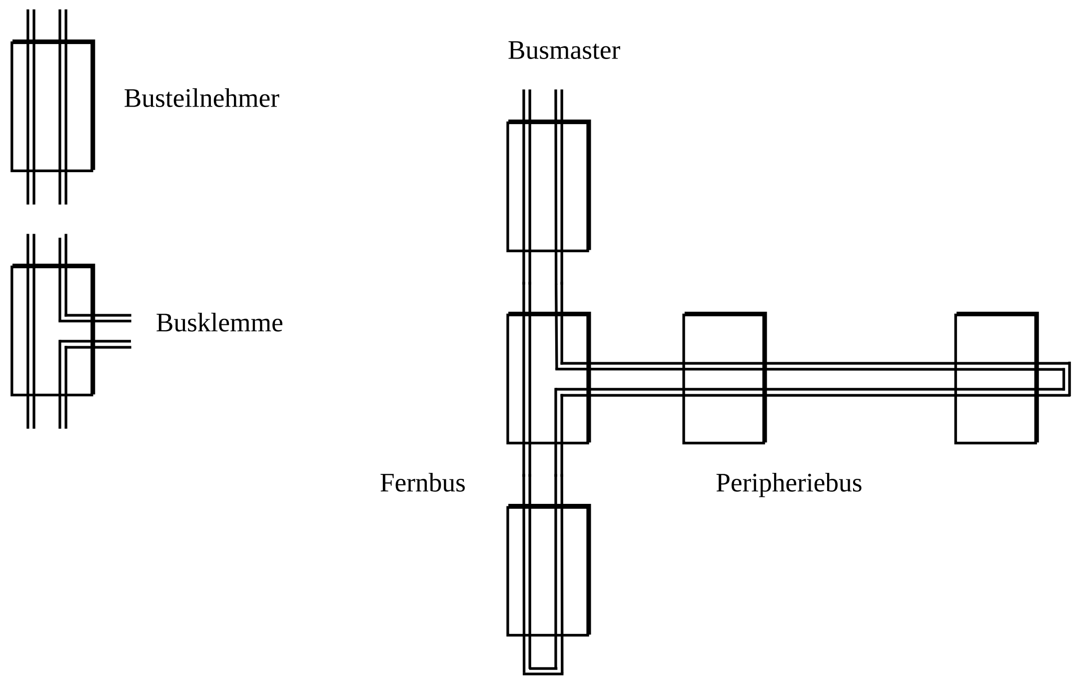

Abb. 6-11: InterBus-S-Topologie

Der Ring im InterBus-S ist dabei so gestaltet, daß sich im Grunde genommen eine linienförmige Struktur ergibt: Jeder Busteilnehmer unterhält zwei Richtungen der Kommunikation, jede Busklemmen vier, so daß der Ring entsprechend deformiert wird.

Die Kommunikation im InterBus-S wird entsprechend einem verkleinerten ISO/OSI-Modell in 3 Schichten behandelt:

Die Übertragung in Schicht 1 erfolgt mittels RS-485, allerdings auf Punkt-zu-Punkt-Basis (RS-422); je zwei Busteilnehmer sind mit Hin- und Rückleitung miteinander verbunden. Die Entfernung zwischen zwei Teilnehmern kann dabei bei 500 kBit/s 400 m betragen, und da jeder Busteilnehmer zugleich einen Repeater darstellt, ist die räumliche Ausdehnung dieses Netzes sehr groß (13 km). Die Anzahl der Adern zwischen 2 Knoten liegt bei 5, da neben je zwei Adern pro RS-485 auch die logische Masse mit übermittelt wird.

RS-485 hätte die Anzahl der Busteilnehmer auf 32 begrenzt; diese Grenze wird durch die Punkt-zu-Punkt-Architektur überschritten und willkürlich auf 256 festgelegt, um die Adressierung handhabbar zu gestalten.

Die Schicht 2 enthält zwei voneinander zu unterscheidende Protokolle für die Datenklassen ‘zyklische Prozeßdaten’ und ‘azyklische Parameter’. Als Besonderheit wurde beim InterBus-S für zyklische Daten das Summenrahmendatagramm eingeführt, bei dem die den einzelnen Knoten entstammenden Daten ein gemeinsamer Rahmen gegeben wird, nachdem diese gesammelt worden sind. Dies wird durch die Topologie und die Architektur möglich (Punkt-zu-Punkt) ermöglicht und resultiert in einer Protokolleffizienz von über 50 %. Die praktische Realisierung wird durch eine Schieberegisterarchitektur in den Knoten erreicht, während die Sicherung via 16-Bit-CRC gemäß CCITT und sogenannter Loopback-Wörter (inkrementiert bei jedem Knoten, daher Ringüberwachung) erreicht wird.

Die Übertragung der azyklischen Parameter geschieht auf herkömmliche Weise.

In Schicht 7 wurde eine abgemagerte Version der FMS-Dienste implementiert.

6.3.5 FIP

FIP bedeutet Factory Information Protocol und stellt ein weiteres Feldbussystem dar, das in diesem Fall aus Frankreich stammt. Hier werden ebenfalls die Schichten 1, 2 und 7 genormt.

Schicht 1 definiert geschirmte, verdrillte Zweidrahtleitungen (STP) mit einer maximalen Länge von 2 km bei 256 Teilnehmern als Grundlage der Kommunikation. Die Datenübertragungsrate liegt bei 1 Mbit/s, mit Ausnahmen in Richtung niedrigerer Energien und höherer Geschwindigkeiten. Die Codierung erfolgt via Manchester und ist damit gleichspannungsfrei. Verletzungsbits, die den bei der Manchestercodierung üblichen Pegelwechsel zur Bitmitte nicht besitzen, werden als Start- und Stopbits für die FIP-Datagramme genutzt.

Schicht 1 enthält – etwas ungewöhnlich – auch den Datagrammrahmen:

Der Startrahmenteil besteht aus mehreren Einsen zur Taktsynchronisation (und -rückgewinnung) und mehreren Verletzungsbits und ist damit eindeutig von allen Nutzdaten zu unterscheiden. Der Enddelimiter enthält ebenfalls mehrere Verletzungsbits.

Der Buszugriff wird in der ***Schicht 2*** durch eine zentrale Busvergabe geregelt. Diese Funktion kann prinzipiell in jedem Busknoten integriert sein, natürlich nur einmal am Bus. Dieses ***Bussteuerwerk*** ruft zyklisch alle projektierten Variablen, die den entsprechenden Prozeßteilen zugerodnet sind, mittels eines entsprechenden Datagramms auf. Der Erzeuger dieser Variablen (z.B. Temperaturwert) übermittelt diese mit Hilfe eines Antwortdatagramms, während alle Teilnehmer am Bus,die entsprechend projektiert sind, mithören und das Datagramm auswerten können. Für jede periodische Variable existiert ein festes Zeitraster, während die freibleibende Zeit eines Elementarzyklus als Zeitfenster für die aperiodischen Datagramme genutzt werden.

Aperiodische Daten können durch jeden Teilnehmer angefordert werden, bei dem eine beliebige periodische Variable erzeugt und abgefragt wird. Die aperiodische Variable wird in diesem Fall durch Kennungsbits angefordert. Dieses Busarbitrierungsverfahren erscheint dabei auf den ersten Blick besonders einfach, allerdings besitzt es erhebliche Mängel im Echtzeitbereich:

Echtzeitbetrieb bei Netzen und damit verteilten Anwendungen benötigen die Möglichkeit, kurzfrsitig auf externe Ereignisse reagieren zu können. Dies ist unvereinbar mit einer starren Einteilung in Prioritäten und Abfragezyklen, wie es de facto durch die periodischen Variablen der Fall ist, während aperiodische Nachrichten und Variablen durch die zentrale Busarbitrierung ggf. nicht rechtzeitig genug erkannt werden oder erst zu späteren Zeitpunkten bei freiwerdenden Buskapazitäten abgefragt werden können. Beide Verfahren bzw. Variablentypen erweisen sich als sehr unflexibel; FIP muß daher als ein Feldbussystem für Prozesse betrachtet werden, bei denen festgelegte Prioritäten und Zeitraster jeden Fall schnell und konsequent genug abdecken.

Als ***Anwendungsschicht*** sind spezielle Dienste, u.a. auch eine Untermenge von MMS implementiert.

Kapitel 7: Leitungsprotokolle

Die Übertragung von Daten zwischen zwei Stationen ist nicht nur eine Frage der physikalischen Definition der Bitdarstellung, sondern sie umfaßt auch weiterführende Aufgaben wie die Sicherung dieser Daten. Diese Funktionen sind in der Schicht 2 (Data Link Layer) des OSI-Schichtenmodells integriert.

Streng genommen wird die Schicht 2 nochmals unterteilt; hierzu muß betrachtet werden, daß eine Kommunikation durch einen Medienzugriff mit anschließender Übertragung zustandekommt. Ist das Übertragungsmedium jedoch nicht exklusiv den kommunizierenden Rechnern zugeteilt, muß der Zugriff erst angemeldet sein. Die hierfür notwendige Funktionalität ist im Sublayer 2a, Media Access Control, zusammengefaßt und wird im nächsten Kapitel behandelt.

Die hier abgehandelten Protokolle gehören zum Sublayer 2b, dem Logical Link Control. Für diesen Protokollteil ist die Verbindung direkt zwischen zwei Rechnern existent und exklusiv zugeordnet.

7.1 Einfache Leitungsprotokolle

Die Aufgabe eines Leitungsprotokolls besteht in dem ***zuverlässigen***, ***unverfälschten***, ***schnellen*** und mit ***möglichst geringem Aufwand*** behafteten Transport von Datenblöcken zwischen den Endteilnehmern. Das Übertragungsmodell ist also sehr einfach aufgebaut:

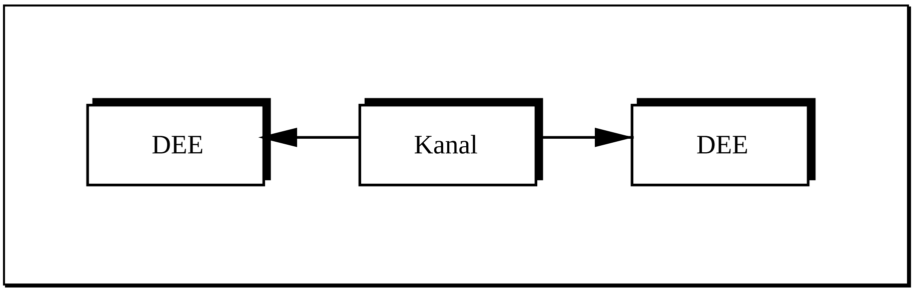

Abb. 7-1: Übertragungsmodell Layer 2b

Die Eigenschaften dieser Übertragung sind im einzelnen:

- Eine ***zuverlässige*** Übertragung liegt vor, wenn ein abgesendeter Datenblock mit großer Wahrscheinlichkeit am Ziel eintrifft.
- ***Unverfälschte*** Datenübertragung bedeutet, daß die Datenbits mit großer Sicherheit am Empfänger unverfälscht ankommen.
- Mit ***schneller*** Datenübertragung soll die ***Übertragungszeit*** eines Blocks möglichst gering gehalten werden. Die Übertragungszeit beinhaltet dabei die Zeitdifferenz von der Bereitstellung der Daten im Sender bis zum Verfügbarsein im Empfänger.

- ***Effiziente*** Übertragung liegt dann vor, wenn die Auslastung des Kanals selbst möglichst hoch ist, der (notwendige) zusätzliche Overhead für die redundante Information, die zur Sicherung, für Quittierungen oder andere Steuerungszwecke beigefügt werden muß, also möglichst gering ist. Bei geringem Overhead kann der Durchsatz (engl. troughput) entsprechend hoch werden.
- Die ***Auslastung*** eines Kanals wird als hoch bezeichnet, wenn dieser relativ oft benutzt wird. Dies ist aus wirtschaftlichen Gründen anzustreben, da häufig Grundgebühren für den Kanal zu zahlen sind.
- Eine ***hohe Verfügbarkeit*** ist gegeben, wenn der Kanal möglichst zu jeder Zeit benutzt werden kann. Dies ist z.B. beim Telefon der Fall, da ein maximaler Ausfall von wenigen Stunden in Jahren garantiert wird. Rechnernetze besitzen diese Verfügbarkeit zur Zeit nicht.

Die Anforderungen sind teilweise sehr widersprüchlich; so lassen sich hohe Auslastung und kurze Übertragungszeiten (im Mittel) kaum miteinander vereinbaren, da bei einer Teilung der Leitungsressource die Übertragungszeiten durch Wartezeiten automatisch ansteigen.

Im folgenden werden einige einfache Maßnahmen zur Erreichung von bestimmten Eigenschaften besprochen.

7.1.1 Echo-Überwachung

Aus Effizienzgründen werden bei einer Datenübertragung keine fehlerkorrigierenden Codes genutzt, sondern maximal fehlererkennende. Um dennoch eine sichere Gesamtübertragung zu gewährleisten, ist eine andere Methode notwendig, die bei Fehlererkennung eine Behebung startet. Hierzu sind zwei Verfahren gängig: Die Echo-Überwachung und die wiederholte Übertragung bei erkanntem und gemeldetem Fehler.

Die ***Echoüberwachung*** (echo checking) sendet das zu übertragende Zeichen an den Empfänger, der es sogleich wieder zurück transferieren muß. Der Sender ist auf diese Weise in der Lage, die Gültigkeit des Zeichens zu überprüfen und ggf. eine Korrektur beim Empfänger vorzunehmen.

Dieses Verfahren findet insbesondere bei textbasierten Terminals (an Rechnern) Anwendung. Ein Zeichen entspricht hier einem vom Benutzer eingegebenen Zeichen, welches anschließend zum Rechner geschickt wird. Im lokalen Modus von Termninals wird dieses Zeichen sofort am Bildschirm angezeigt, im abgesetzten Modus (remote) jedoch das reflektierte Zeichen, so daß der Anwender mittels des <DEL>-Zeichens einen falschen Buchstaben korrigieren kann.

Die Nachteile dieses Verfahrens sind eine doppelte Belegung des Kanals (Hin- und Rücktransfer), eine Verzögerung durch den ständigen Transfer und das Vortäuschen einer falschen Fehlerrate: Zeichen, die korrekt beim Empfänger ankommen, beim Sender jedoch verfälscht, werden als falsch präsentiert.

7.1.2 Sende-und-Warte-ARQ-Protokoll

Bei Erkennung einer fehlerhaften Übertragung – dies bedeutet selbstverständlich die Übertragung einer Fehlererkennungsmöglichkeit – wird eine erneute Sendung automatisch angefordert. Dieses Verfahren ist das ‘preiswerteste’ unter den Fehlerbehebungsverfahren, da der Overhead nur bei einer Fehlererkennung entsteht. Es wird als ***automatische Wiederholungsanforderung*** (*ARQ, automatic repeat request*) bezeichnet:

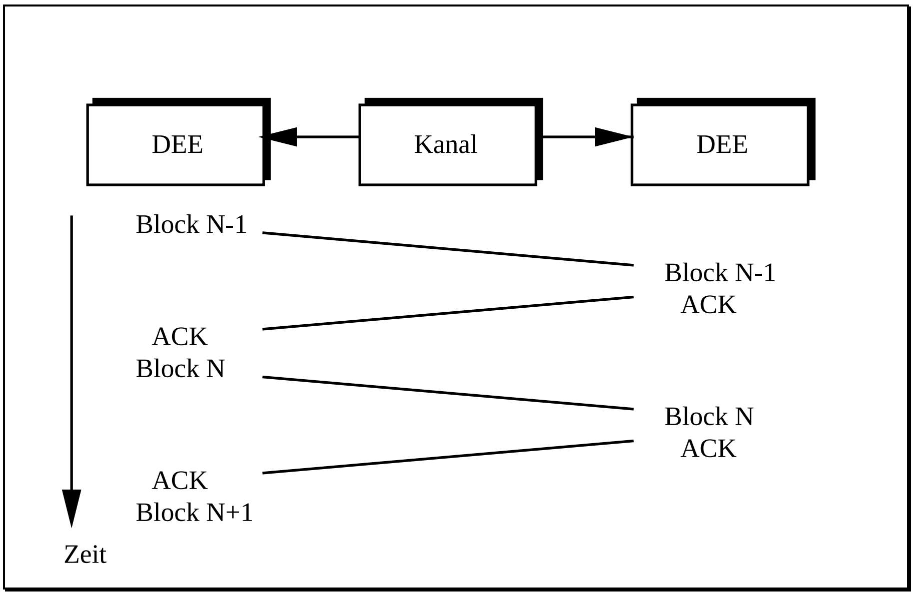

Abb. 7-2: ARQ-Verfahren modulo 1

Das in Abb. 8-2 gezeigte Verfahren wartet bei jedem übertragenen Datenblock auf ein Acknowledge, wobei es nur positive (ACK) oder negative und positive (NAK, ACK) geben kann. Ein Block gilt genau dann als fehlerhaft, wenn dieser oder seine Quittierung verlorengegangen ist oder eine entsprechende Quittierung gesendet wurde. Die Erkennung von verlorengegangenen Blöcken erfolgt durch eine Zeitsteuerung, nach deren Ablauf entweder der Block nochmalig gesendet wird oder spezielle Anfragen an den Empfänger gestartet werden (ENQ, enquiry).

Bei Übertragungen größerer Datenmengen wird diese gewöhnlich in Datenpakete aufgeteilt, die im einzelnen unterschiedliche Wege gehen können: Die Reihenfolge der empfangenen Datenblöcke muß nicht zwangsläufig korrekt sein, so daß – auch in Schicht 2 – eine Blocknumerierung zumeist modulo einer Zahl, 2, 8 oder 128, eingeführt wird. Dies dient auch der Erkennung von verlorengegangenen Blöcken ohne Zeitsteuerung. Die Quittierungsmeldungen müssen in diesem Fall nur beim Sender eingehen, bevor der ‘gleiche’ Modulo-Block erneut gesendet wird, Datenblock/Quittierung stehen nicht mehr in einer unmittelbaren zeitlichen Reihenfolge. Ziel dieser Maßnahme ist die Erhöhung der Übertragungsgeschwindigkeit.

7.1.3 Leistungsbewertung des ARQ-Protokolls

Unter gewissen Modellierungsbedingungen kann man die Leistungsfähigkeit des ARQ-Protokolls theoretisch bestimmen. Hierzu sei n die Anzahl der Bits in einem Datenblock einschließlich der ACK-Meldung, k die Anzahl der Nutzinformationen, s die Zusatzinformationen. Bei A Blöcken werden dementsprechend k * A = (n - s) * A Bits an Nutzinformation übertragen.

Bei einem unabhängigen Fehlermodell, das also davon ausgeht, daß Bits unabhängig voneinander einen Fehler erzeugen können, gilt bei p als Einzelwahrscheinlichkeit für ein Bit und n als Blocklänge

$$P\left[Fehler\ im\ Block\ der\ Länge\ n\right] = 1 - (1 - p)^n$$

Die Anzahl der Blöcke A, die mindestens gesendet werden, wird bei Fehlern in eine Anzahl A' vermehrt. Die Sendezeit eines Blocks umfaßt das komplette Intervall vom Start des Blocks bis zum Start des nächsten Blocks und damit die Übertragungszeit t_b pro Bit (1/Bitrate), die Signalverzögerungszeit t_V und die Bearbeitungszeit im Sender/Empfänger (t_P). Diese Zeit heißt ***Blockübertragungszeit*** T_B:

$$T_B = n \bullet t_b + t_p + 2 \bullet t_v$$

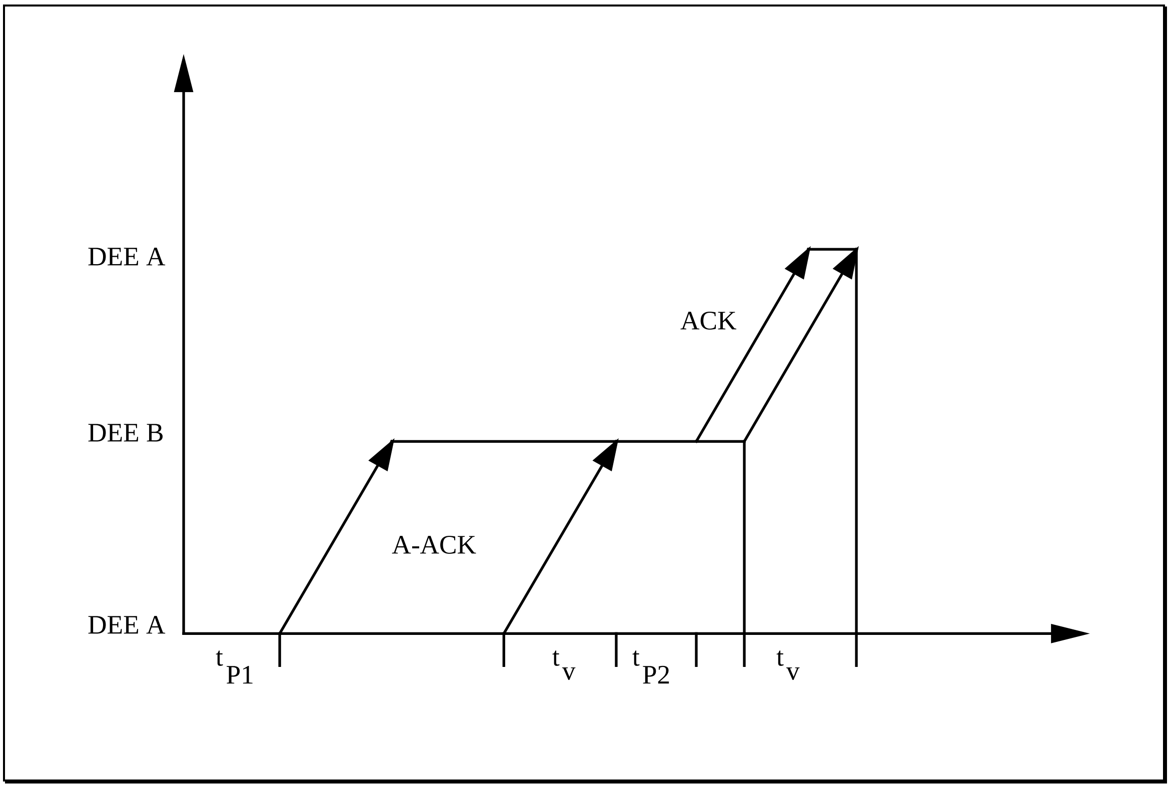

Abb. 7-3: Zur Ermittlung der Übertragungszeit

Von A Blöcken sind im Mittel A * P fehlerhaft und müssen wiederholt werden; von diesen sind wiederum A*P² fehlerhaft, usw., so daß die Anzahl der gesendeten Blöcke

$$A \bullet \left(P + P^2 + P^3 + \ldots\right) = A \bullet \frac{P}{1-P}$$

ist. Nunmehr sind 2 Varianten zu unterscheiden: Variante 1 erkennt die fehlerhafte Aussendung nur durch Timeout, Variante 2 durch NAK-Sendungen, wobei hier der Totalverlust natürlich auch durch Timeout erkannt werden muß.

In Variante 1 (Timeout) wird für die Zeit, die zur Detektierung des Fehlers benötigt wird, ein Vielfaches der Zeit T_B angenommen:

$$T_Z = z \bullet T_B$$

Somit ergibt sich für die Dauer der Übertragung aller Blöcke

$$\begin{aligned} T &= A \bullet T_B + A \bullet \frac{P}{1-P} \bullet T_Z = A \bullet \left(n \bullet t_b + t_p + 2 \bullet t_v\right) \bullet \left(\frac{1 + P \bullet (z-1)}{1-P}\right) \\ &= A \bullet \left(n \bullet t_b + t_p + 2 \bullet t_v\right) \bullet \left(\frac{\left(1-(1-p)^n\right) \bullet z + (1-p)^n}{(1-p)^n}\right) \end{aligned}$$

und für die effektive Bitrate λ

$$\lambda = \frac{A \bullet (n-s)}{T} = \frac{(1-p)^n \bullet (n-s)}{\left(n \bullet t_b + t_p + 2 \bullet t_v\right) \bullet \left(\left(1-(1-p)^n\right) \bullet z + (1-p)^n\right)}$$

Diese Bitrate hat ein Maximum, denn für k = 0 ist λ = 0, für k sehr groß konvergiert λ ebenfalls gegen 0.

In der Variante 2 wird z = 1 angenommen, was bedeutet, daß alle Fehler durch NAK erkennbar sein sollen. Dies vereinfacht die Formel für λ zu

$$\lambda = \frac{(1-p)^n \bullet (n-s)}{n \bullet t_b + t_p + 2 \bullet t_v}$$

Neben der effektiven Bitrate ist im fehlerfreien Fall, also für p = 0, die fehlerfreie Übertragungszeit und die Auslastung U (utilization) berechenbar:

$$T_0 = A \bullet \left(n \bullet t_b + t_p + 2 \bullet t_v\right)$$

$$U = \frac{T_G}{T_0} = \frac{A \bullet n \bullet t_b}{A \bullet (n \bullet t_b + 2 \bullet t_v)} = \frac{1}{1 + \frac{2 \bullet t_v}{n \bullet t_b}}$$

U wurde dabei mit der Annahme hergeleitet, daß t_p klein gegenüber den anderen Größen ist. Dies kann zu durchaus interessanten Ergebnisse führen; so sei n = 1000 angenommen, für die Übertragungsgeschwindigkeit möge 2/3 * c (Lichtgeschwindigkeit) gelten (Wert für Koaxialkabel), und t_b sei zu 1 kbit/s und 1 Mbit/s gewählt. Dann ergeb sich folgende Werte für U:

Kanallänge	1 kBaud	1 Mbaud
1 km	U = 1	U = 1
200 km	U = 1	U = 0,33
50000 km (Satellit)	U = 0,67	U = 0,002

Tabelle 7-1: Kanalauslastung

Dies macht deutlich, daß z.B. bei Satellitenübertragung andere Protokolle zum Tragen kommen.

7.2 BSC-Leitungsprotokolle

Das von der ISO definierte ***Basic-Mode***-Protokoll war sehr lange das verbreiteste Protokoll und wurde seitens IBM unter dem Namen ***Binary Synchronous Control*** (BSC) übernommen. Dieses Protokoll ist zeichenorientiert und synchron, d.h., es werden immer Blöcke (synchron) von Bytes (Zeichen) übertragen. Das Protokoll definiert einen Auf- und Abbau der Verbindung sowie die Datenübertragung:

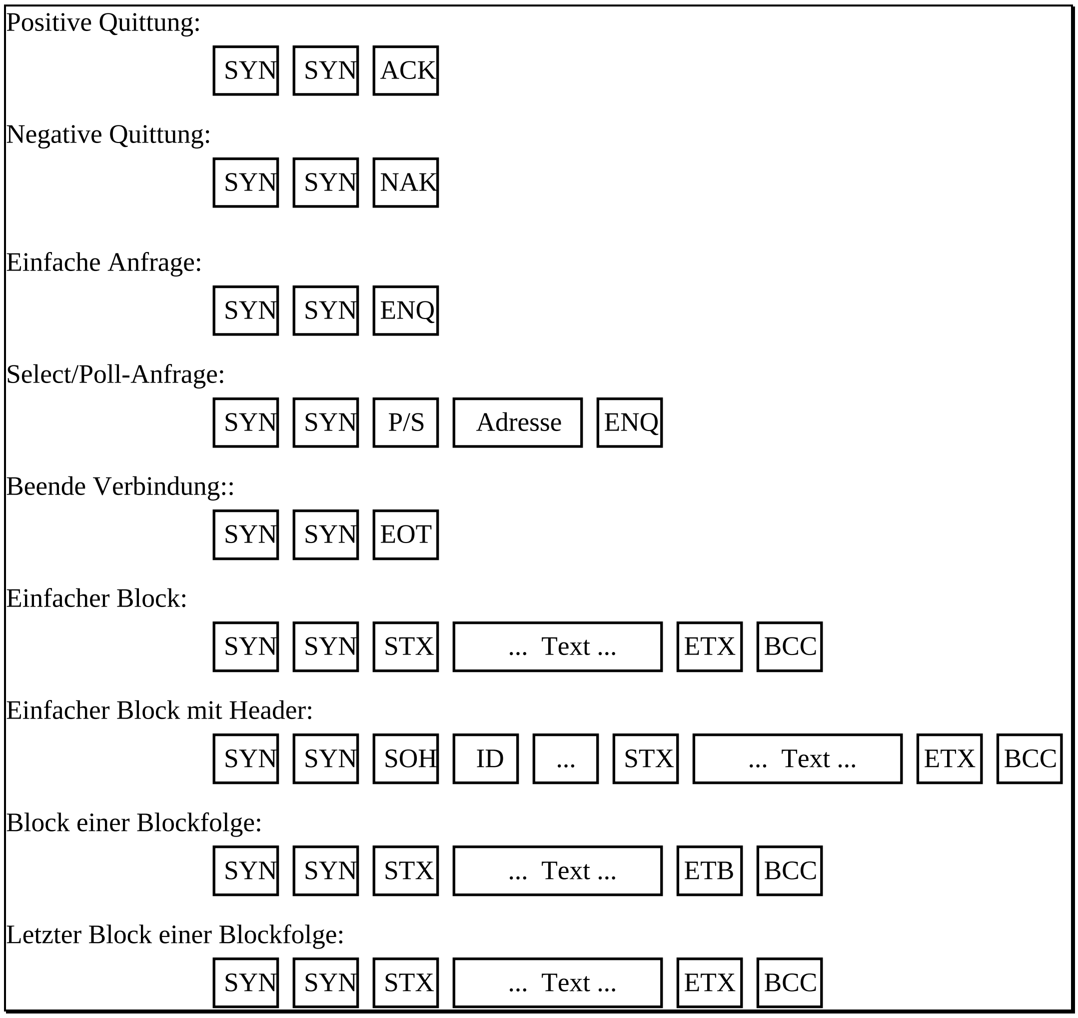

Abb. 7-4: Nachrichten in BSC

Einige der ASCII-Zeichen werden – deutlich sichtbar am Verbindungsauf- und -abbau – als Steuerzeichen benötigt. Um diese transparent in der Übertragung schicken zu können, wird der Transparenzmodus durch die Zeichenkombination DLE - STX (Data Link Escape, Start of Text) ein- und durch DLE - ETX ausgeschaltet. Dies bedeutet, daß nunmehr jedes Zeichen mit Ausnahme des DLE-Zeichen sendbar ist; letzteres kann im Normalmodus durch DLE - DLE gesendet werden.

Die Fehlerüberwachung erfolgt durch Einfügung einer Blockchecksumme (1 oder 2 Bytes, BCC), allerdings nur über die Nutzdaten, nicht den Kontrollrahmen. Die Wartezeit auf eine Bestätigung ist mit etwa 2 Sekunden definiert, was typischerweise bei Terminalanwendungen tolerierbar ist.

Die Flußkontrolle wird durch sogenannte verzögerte Bestätigung (SYN SYN DLE DLE) erreicht, die – ausgesendet vom Nachrichtenempfänger – den Sender zu einer Anfrage veranlassen und so den Blockfluß verzögern. Bei längerer Untätigkeit durch Datenmangel muß der Sender einen Verzögerungsrahmen (SYN SYN STX ENQ) schicken, der vom Empfänger mit SYN SYN NAK beantwortet wird.

Im Klartext heißt dieses Protokoll: Die Teilnehmer unterhalten sich ständig, ein Multiplexbetrieb ist nicht vorgesehen, so daß die Auslastung schlecht ist. Die Blöcke werden (im Halbduplexbetrieb!) modulo 1 gesendet, d.h., jeder Block muß zunächst bestätigt werden, die Nettodatenrate ist nicht sehr hoch. Das Protokoll gilt in der Zwischenzeit als veraltert.

7.3 Das HDLC-LAPB-Leitungsprotokoll

Das in der CCITT-Empfehlung X.25 definierte Protokoll der 2. Schicht, das sogenannte ***HDLC-LAPB***-Protokoll (*high level data link control, link access procedure balanced*) ermöglicht

- den Auf- und Abbau von Verbindungen
- die transparente, gesicherte Übertragung von Blöcken zwischen zwei Datenendeinrichtungen und
- die Fehlerüberwachung, so daß nicht behebbare Fehler an die nächsthöhere Schicht gemeldet werden

und garantiert damit die gesicherte Übertragung in Schicht 2.

HDLC baut auf Vollduplexverbindungen (X.21) auf und nutzt zur Kommunikation Blöcke mit einheitlichem Aufbau:

flag	Adresse	Steuerfeld	Datenfeld	FCS	flag

Abb. 7-5: Blockaufbau HDLC

Hierbei bedeuten:

Flag: Blockbegrenzung. Diese hat eine Länge 8 Bits und besteht aus dem Zeichen 01111110, daß durch Bitstuffing bedingt an keiner anderen Stelle vorkommen kann. Zwischen zwei Blöcken muß mindestens ein Flag stehen. Werden keine Blöcke gesendet, so werden stattdessen ununterbrochen Flags gesendet, so daß beide Stationen ununterbrochen synchronisieren.

Adresse: Adresse des Senders (bei Meldungen) oder des Empfängers (bei Befehlen)

Steuerfeld: Angabe des Blocktyps, siehe unten

Datenfeld: Enthält die Benutzerdaten, die von der Schicht 3 übergeben werden. Die Länge kann 0 bis 131 Bytes betragen.

FCS: Das Blockprüffeld (*frame checking sequence*) besteht aus 2 Bytes (16 Bits), in denen die in dem CRC-Verfahren gebildete Prüfsumme über den gesamten Block (Ausnahme: Flags) eingetragen wird.

Steuerfeld	Bit 1(lsb)	2	3	4	5	6	7	8 (msb)
I-Block	0		N(S)		P		N(R)	
S-Block	1	0	S	S	P/F		N(R)	
U-Block	1	1	M	M	P/F	M	M	M

N(S): Sendefolgenummer modulo 8
N(R):Empfangsfolgenummer modulo 8
S: Bits für Steuerung (00 = RR, 10 = RNR, 01 = REJ)
M: Bits für Steuerungsfunktion (11000 = DM, 11100 = SABM, 00010 = DISC, 00110 = UA, 10001 = CMDR/FRMR)
P/F: Sendeaufruf bei Befehlen, Endeanzeige bei Meldungen

Abb. 7-6: Steuerfeld HDLC

HDLC bietet drei Blocktypen an, mit denen der Verbindungsauf- und -abbau, die Steuerungen der Verbindung und die Nutzdaten übertragen werden können. Diese Blocktypen werden im Steuerfeld unterschieden (siehe Abb. 7-6). Hierzu zählen im einzelnen:

Die ***U-Blöcke*** (unnumerierte Blöcke) dienen dem Auf- und Abbau der Verbindung. Sie sind unnumeriert, da sie keine Folgeblöcke besitzen, sondern innerhalb eines Blocks einen Befehl übermitteln. Hierzu sind 5 Befehle und Meldungen definiert:

- ***SABM***, Set Asynchronous Balanced Mode, fordert die Gegenstelle zum gleichberechtigten Spontanbetrieb auf.
- ***DISC***, Disconnect, hält die Übermittlung an (Abbruch)
- ***UA***, Unnumbered Acknowledge, dient zur Bestätigung der Befehle SABM und DISC
- Die Meldung ***DM*** (Disconnect Mode) wird gesendet, falls eine Station im Zustand Abgebrochen verweilen muß.
- ***CMDR*** (Command Reject) oder ***FRMF*** (Frame Reject) zeigen als Meldungen Fehler bei der Übertragung an, der nicht durch Blockwiederholung geändert werden können: Block zu lang, ungültige Codierung im Steuerfeld, Folgenummer ist ungültig, falsche Länge von U- oder S-Blöcken.

Die ***S-Blöcke*** (Steuerungsblöcke) steuern die Datenübermittlung. Hierzu können 3 Befehle übermittelt werden:

- Der Befehl ***RR*** (Receive Ready) teilt die Empfangsbereitschaft dieser aussendenden Station mit. Zugleich wird der korrekte Empfang aller Blöcke bis N(R) - 1 quittiert und ggf. die Gegenstelle aufgefordert, ihren Zustand mitzuteilen.
- ***RNR*** (Receive Not Ready) teilt der Gegenstelle das Gegenteil mit; die Quittierung von empfangenen Blöcken bleibt erhalten.

- Der Befehl ***REJ*** (Reject) fordert die Gegenstelle zur Wiederholung aller Blöcke ab der Nummer N(R) auf.

HDLC bietet ein genaues Verfahren zur Verbindungsaufnahme und zur Datenübertragung an. Hier sei auf die einschlägige Literatur verwiesen. Die Fehlererkennung bei komplett verlorengegangenen Blöcken kann beispielsweise dadurch erreicht werden, daß die Sende- und Empfangszähler nicht mehr übereinstimmen.

Kapitel 8: Medienzugangskontrolle

Die im letzten Kapitel betrachteten Protokolle dienten dem Leitungszugriff bei vollständig kontrollierten Übertragungsmedien, die also exklusiv dem Sender zugeordnet sind. Dieser Fall tritt insbesondere dann auf, wenn es sich um Punkt-zu-Punkt-Verbindungen handelt. Beispiele hierfür sind das HDLC-LAPB-Protokoll sowie natürlich RS-232-Verbindungen.

Andere Übertragungsmedien wie das Vakuum oder Leitungen in Local Area Networks sind prinzipiell so ausgelegt, daß viele Teilnehmer – auch gleichzeitig – darauf schreibend zugreifen können. In diesem Fall muß das Medium zugeteilt werden, soll es nicht zu Zugriffskonflikten kommen.

Das Prinzip der Medienzugriffskontrolle funktioniert insbesondere dann, wenn die Laststruktur des einzelnen Teilnehmers so ausgelegt ist, daß Zugriffe nur selten vorkommen und nur kurze Zeit in Anspruch nehmen. Letzteres läßt sich genau dann erreichen, wenn hohe Datenübertragungsraten eingesetzt werden. Die erste Bedingung ist dagegen prinzipiell von der Übertragungsrate unabhängig und kann bei deutlicher Verletzung letztendlich zu einer Blockade des gesamten Netzes führen.

Die Behebung und/oder Vermeidung von Zugriffskonflikten geschieht über sehr unterschiedliche Methoden zur Zugriffskontrolle. Diese Unterschiede in der Methodik haben folgende Ursachen:

- Die typische Kommunikationslänge (Nutzerdaten) kann sehr unterschiedlich sein: Der Dialogverkehr benötigt viele, kurze Zugriffe, der Datenverkehr wie etwa Dateitransfer wenige, lange.
- Die Übertragungsverzögerung muß eventuell in eng gesteckten Grenzen gehalten werden.
- Die (Netto-)Übertragungsrate soll maximiert werden.
- Die Fehlersicherheit der Übertragung ist die wichtigste Anforderung.
- Die Signallaufzeiten sind – insbesondere bei Satellitenkommunikation – so lang, daß diese Zeit in dem Zugriffsverfahren zu berücksichtigen ist.
- Der maximale Zeit zur Übertragung einer Nachricht soll deterministisch bestimmt werden können.

Um diesen vielen Anforderungen gerecht werden zu können, müssen mehrere, voneinander klar unterscheidbare Zugriffsverfahren definiert werden. Dies geschieht innerhalb der Schicht 2 des ISO/OSI-Layermodells, die aus diesem Grund in zwei Unterschichten unterteilt wird:

Die ‘tiefere’ Schicht wird als Medienzugriffskontrolle, engl. ***Media Access Control***, MAC bezeichnet; diese Verwaltungsschicht regelt den Zugriff auf das Medium bei Mehrfachzugriffen, wobei die Zuteilungsstrategien im folgenden besprochen werden.

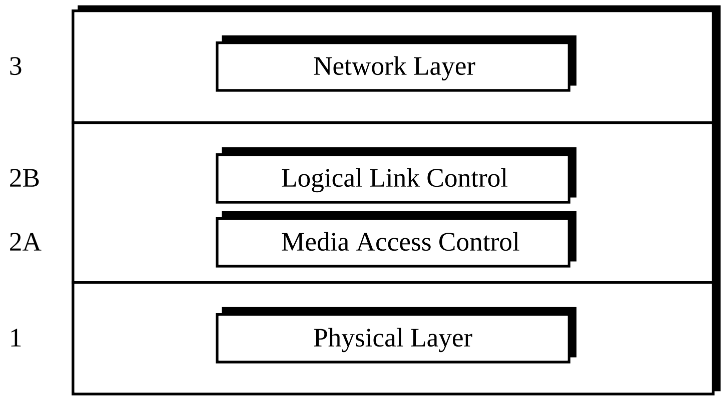

Abb. 8-1: Mehrfachzugriffskontrolle im 7-Schichtenmodell

Die darüberliegende Subschicht bildet die logische Verbindungskontrolle, engl. ***Logical Link Control***, LLC. Innerhalb dieses Teil der Schicht 2 werden Adressierung, Fehlerkontrolle etc. behandelt, d.h., es werden Verwaltungsbits zu den Nutzerdatenbits hinzuaddiert, während die MAC-Subschicht nur den Zugriff regelt. Viele Hardwarecontroller für den Netzzugriff, dies gilt insbesondere für frühere Designs, integrieren daher die Unterschicht 2A in die Schicht 1, während für 2B Softwarelösungen existieren.

Bei den Medienzugriffsstrategien werden zwei große Prinzipien unterschieden: Die ***Zuteilungsstrategien*** verteilen die Sendeberechtigung auf einem unteilbaren Medium nach Kriterien wie Sendebereitschaft, Prioritäten, zyklische Methoden etc. Sie benötigen eine verteilte oder konzentrierte Zuteilungsinstanz, damit Rechen- und Netzübertragungszeit, bieten andererseits jedoch eine Deterministik in der maximalen Übertragungszeit.

Die ***Zufallsstrategien*** lassen den einzelnen Stationen die prinzipielle Freiheit des Sendebeginns. Kommt es dabei zu einer Kollision, so muß die Sendung sofort nach Detektierung unterbrochen und zu einem anderen Zeitpunkt wieder aufgenommen werden. Diese Strategieklasse benötigt sehr wenig (zentrale) Rechenkapazität, das Netz ist gewöhnlich mit höheren Gesamtübertragungsraten belastbar, kann jedoch keine Maximalübertragungszeit garantieren und sich letztendlich selbst blockieren.

8.1 Zuteilungsstrategien

Die bereits erläuterte Klasse der Zuteilungsstrategien kann weiterhin unterteilt werden in

- ***feste Zuteilungsverfahren***, bei denen ein Teilnehmer zu festen Zeiten senden kann (Zeitmultiplex)
- ***variable Zuteilungsstrategien*** mit zentraler oder dezentraler Kontrolle der Zugriffsberechtigung.

Feste Zuteilungsverfahren werden z.B. in synchronen Nachrichtennetzen wie ISDN benutzt. Hier werden in einem 2 MBit/s-Kanal 30 Zeitschlitze für je ein Byte zur Verfügung gestellt, wobei jedem Benutzer dann 64 kBit/s Datenrate zur Verfügung stehen.

Das Verfahren benötigt nahezu keinen Overhead, so daß praktisch die gesamte Bandbreite für nützliche Datenkommunikation verwendet werden kann. Die Datenrate läßt sich obendrein in gewissen Grenzen dadurch variieren, daß die Einteilung zwar erhalten bleibt, aber einzelnen Verbindungen, insbesondere wenn weniger als 30 bestehen, mehr Zeitschlitze zur Verfügung gestellt werden.

Der wesentlichen Einsatzfall der festen Zuteilungsstrategie liegt bei Übertragungen, die auf eine enge Begrenzung der Verzögerungszeiten hin optimiert sind: Digitale Sprachübertragung.

Die ***zentrale Zuteilung*** durch Polling nutzt einen Masterrechner am Netz, der allen anderen ein Nachrichtenpaket als Anfrage zusendet. Diese werden dort aufgenommen und erlauben den Slavestationen, ihrerseits ein Datenpaket zu senden. Soll dies nicht geschehen, wird ein sogenanntes ***Go Ahead***-Paket gesendet: Der Master betreibt seine Abfrage mit der nächsten Station.

Die Nachteile eines solchen Verfahrens sind augenscheinlich: Die Abfrage benötigt eine Zeit proportional zu der Anzahl der angeschlossenen Stationen, falls eine feste Strategie im Netz verankert ist. Dies wird in den im Kapitel 6 diskutierten Aktor-Sensor-Bussen sowie den Feldbussen häufig genutzt, da andererseits die Zuteilung sehr einfach zu implementieren ist. Als weiterer Nachteil muß der Kommunikationsoverhead angesehen werden, da zur Aussendung eines Datenpakets zwei Sendungen notwendig sind und auch ‘Nicht-Antworten’ abgefragt werden. Das Zuteilungsverfahren mit zentraler Zuteilung ist allerdings prinzipiell echtzeitfähig, da eine maximale Reaktionszeit im Netz garantiert werden kann.

Strategien zur Optimierung des Netzdurchsatzes erfordern variable Zuteilungen. Hierzu ist eine adaptive Anpassung der Abfragesequenz möglich.

Die ***dezentralen Zuteilungsprotokolle*** bieten in Rechnernetzen, die einen wesentlich stoßartigeren Datenverkehr aufweisen als dies bei der digitalen Sprachübertragung der Fall ist, eine wesentlich bessere Anpassung des Netzdurchsatzes an die theoretische Grenze. Hier kann man zwei verschiedene Verfahren grob voneinander unterscheiden:

- Dezentrale Zuteilung durch Reservierung
- Dezentrale Zuteilung durch Sendeberechtigung (Token)

Die ***dezentrale Zuteilung durch Reservierung*** benötigt einen zeitlichen Sendeabschnitt, in dem die einzelnen sendewilligen Stationen einen freien Sendeabschnitt reservieren können. Dieser Sendeabschnitt umfaßt beim ***Bit-Map-Protokoll*** exakt n Bits bei n angeschlossenen Stationen. In diese Reservierungsbits kann die k-te Station ein Reservierungsbit an die k-te Stelle schreiben und erhält dann nach einer festgelegten, allen Stationen bekannten Reihenfolge einen Zeitschlitz zum Senden. Das Protokoll hat dann in etwa den Ablauf aus Abb. 8-2.

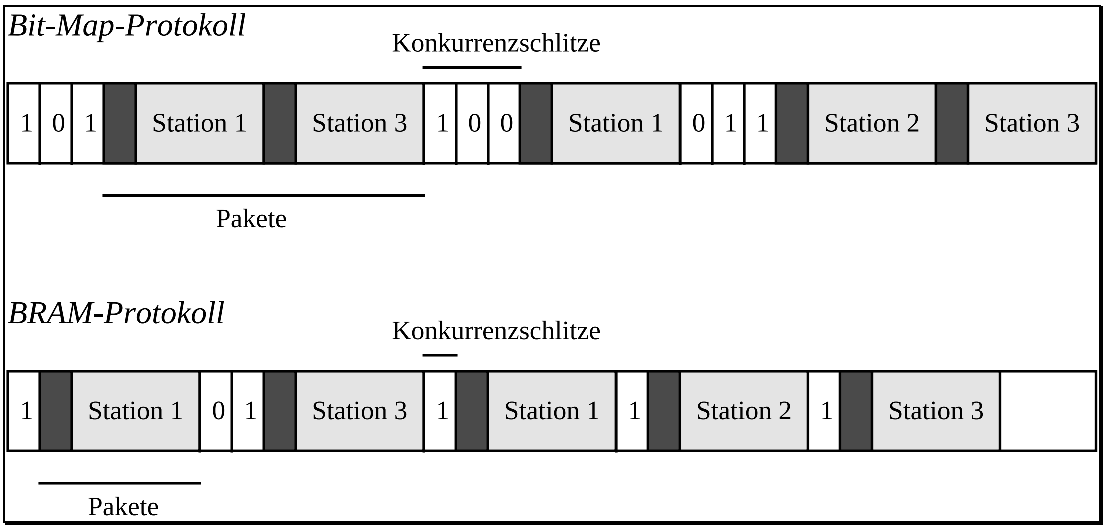

Abb. 8-2: Dezentrale Zuteilungsprotokolle mit Reservierung

Die Voraussetzungen für eine derartige Zuteilung bestehen in der Definiertheit der Reservierungszeitpunkte sowie in einer festen Konfiguration des Netzes. Ist das System nur gering belastet, werden praktisch nur Konkurrenzschlitze gesendet. Will die k-te Station nunmehr senden, so markiert sie das Bit k, muß aber mindestens die restlichen Bits des Reservierungsschlitzes noch abwarten, ehe sie senden darf. Die Wartezeit beträgt demnach (bei m als aktuellem Zeitschlitz)

$$s_{nm} = \begin{Bmatrix} N - m & falls \; n < m \\ 2 \bullet N - m & falls \, n \geq m \end{Bmatrix}$$

so daß die mittlere Wartezeit ohne Datenübertragung

$$W_n = \sum_{m=1}^{N} p \bullet s_{nm} = p \bullet \left(\sum_{m=1}^{n-1} s_{nm} + \sum_{m=n}^{N} s_{nm} \right)$$

beträgt.

In der in Abb. 8-2 bereits gezeigten Variation der Zuteilung, dem sogenannten ***BRAM-Protokoll*** (broadcast recognition access mechanism) wird die Reservierungsphase sofort unterbrochen, wenn eine Station die Sendeberechtigung erhalten will. Dadurch verringert sich die mittlere Wartezeit, wobei nun anscheinend die Stationen mit kleiner Nummer stark bevorzugt werden. Dies wird dadurch ausgeschaltet, indem die Reservierungsphase nach einer Sendung mit der nächstfolgenden Station weiterläuft, so daß keine Station mehr bevorzugt wird. Die mittlere Wartezeit bei geringer Belastung beträgt nunmehr

$$W_{BRAM} = 0{,}5 \bullet N + 0{,}5$$

und ist von n unabhängig. Abb. 8-2 zeigt ein Beispiel mit 3 Teilnehmern.

Das in den letzten Jahren häufiger verwendete Verfahren zur Erteilung einer Sendeberechtigung verwendet Marken (***Token***). Der Inhaber dieser Marke – im Netz kann es eine oder mehrere solcher Marken geben – ist berechtigt, ein Datenpaket auf dem Netz zu senden, alle anderen Stationen dürfen keine Sendung tätigen. Nach Abschluß des Datenpakets generiert die bisher sendeberechtigte Station eine neue Sendeberechtigung und gibt diese an die nächste Station weiter.

Wichtige Voraussetzung für die Sendezuteilung mittels Token ist die logische Strukturierung des Netzes als Ring, zumeist sogar unidirektional. Ferner muß eine ausgezeichnete Station eine Monitorfunktion übernehmen, die dafür sorgt, daß beispielsweise verlorengegangene Token wieder generiert werden usw.. Für die Sendezuteilung mittels eines Tokens existieren zwei Normen mit unterschiedlicher Struktur der Verbindung:

Im ***Token-Bus-Verfahren*** (IEEE 802.4) ist das Netz zwar physikalisch busförmig ausgeführt, logisch jedoch ein Ring. Dies bedeutet ein aufwendiges Verfahren zur Initialisierung, zum Einfügen von Stationen und zur Fehlerbehebung, bietet andererseits jedoch auch die Möglichkeit, den logischen Ring in eine komplexere Struktur zu überführen und so Prioritäten einzufügen. Abb. 8-3 zeigt eine physikalische Bus- und logische Ringstruktur des Token-Bus.

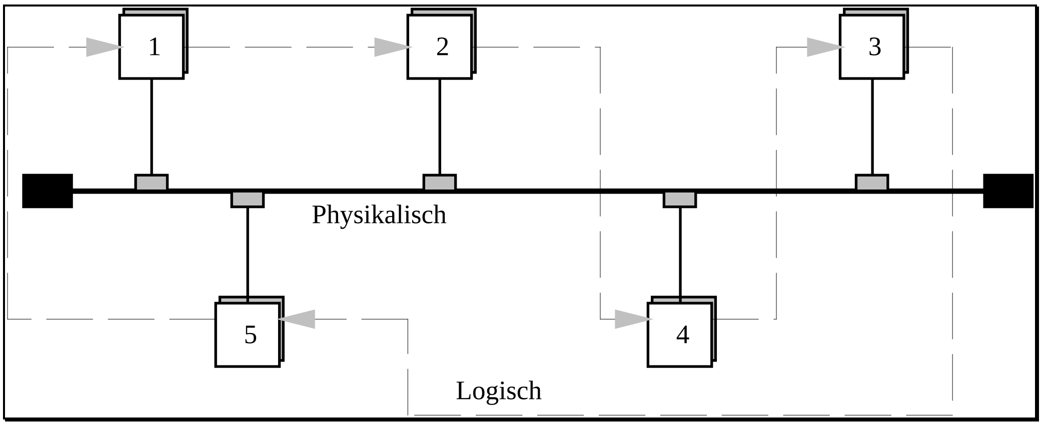

Abb. 8-3: Struktur am Token Bus

Beim ***Token-Ring-Verfahren*** (IEEE 802.5) wird auch die physikalische Struktur als Ring ausgeführt, die im Prinzip auf miteinander verbundenen Punkt-zu-Punkt-Verbindungen besteht. Zwischen zwei Stationen wird in festgelegter Richtung der Freitoken übertragen, und es muß durch technische Maßnahmen gewährleistet werden, daß weder die Laufzeit drastisch gesteigert wird (etwa durch Zwischenspeichern der kompletten Sendung und erneutes Aussenden) noch die im Ring gespeicherte Bitmenge zu klein ist (für die Übertragung des Freitokens, z.B. 24 Bit beim Token Ring). Dies erfolgt durch Zwischenspeichern, aber sofortiges Weitersenden in den Knoten.

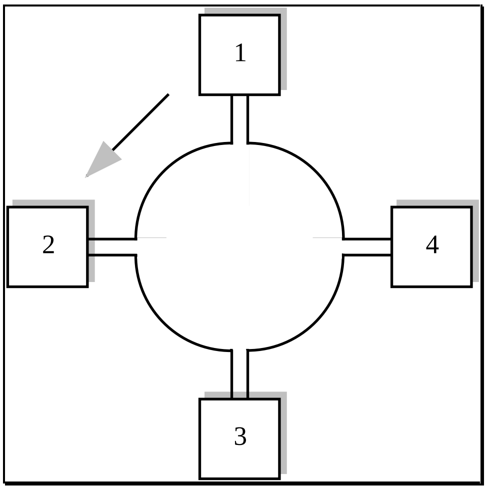

Abb. 8-4: Token Ring Struktur

Die Sendezeit pro Freitoken wird auf z.B. 10 ms, durch das Netzwerkmanagement parametrisierbar, eingestellt. Im Anschluß muß die Sendestation diesen weitergeben, teilweise (4 Mbit/s) nach Warten auf das Freiwerden des Rings, teilweise sofort (Early Token Release).

Eine der wichtigsten Probleme beim Token Ring ist die Fehlererkennung und -behebung insbesondere von Ringunterbrechungen. Diese Unterbrechungen erfolgen zumeist durch Rechnerausfall, in den seltensten Fällen durch Kabelausfall, und können durch einen Bypass behoben werden. Hierzu ist die Einrichtung eines Verkabelungszentrums notwendig, in dem die entsprechenden Schaltungen ausführbar sind.

8.2 Zufallsstrategien

Die Zufallsstrategien setzen an den Punkt an, wo eine Sendung beginnen soll. Der Sender versucht dann spontan, auf das Medium zuzugreifen, um die Daten unmittelbar zu transferieren. Der Zugriff wird natürlich sofort gestoppt, falls das Medium belegt ist (Kollisionsvermeidung).

Trotz dieser ersten Maßnahme kann es zu Datenkollisionen kommen, falls zwei Stationen zur gleichen Zeit ihre Sendung beginnen. Das Maß dieser Gleichzeitigkeit wird durch die räumliche Distanz der Stationen und die Signalgeschwindigkeit begrenzt. In diesem Fall muß die Kollision erkannt und durch geeignete Maßnahmen wieder behoben werden.

8.2.1 CSMA - Carrier Sense Media Access

Das Abhören eines Übertragungskanals zur Kollisionsvermeidung wird mit ***CSMA*** (Carrier Sense Media Access) bezeichnet. Dieses Verfahren kann in mehreren Untervarianten betrieben werden:

- Der (lesende) Zugriff auf das Übertragungsmedium kann zu jedem Zeitpunkt geschehen (reines CSMA) oder es kann nur zu bestimmten Zeitpunkten, namentlich zu Beginn von Zeitschlitzen, erfolgen (slotted CSMA). Slotted CSMA-Netze haben einen bei Vollast wesentlich höheren Ausnutzungsgrad, benötigen jedoch eine Zeitschlitzverteilung.
- Die Aussendung des Datenpakets erfolgt bei freiem Kanal mit der Wahrscheinlichkeit 1 (***nicht-persistentes CSMA***, persistent = beharrend), oder p (***p-persistentes CSMA***). Beim zweiten Verfahren wird mit der Wahrscheinlichkeit 1-p eine Zeit τ gewartet, die gerade so bemessen ist, daß ein Bit am Kanal entlanglaufen kann. Auf diese Weise wird ein vorher nicht-detektierter Sender bemerkt. Bei beiden Verfahren wird bei belegtem Kanal eine Zufallszeit T gewartet, bis der Sendewunsch erneut gestartet wird.

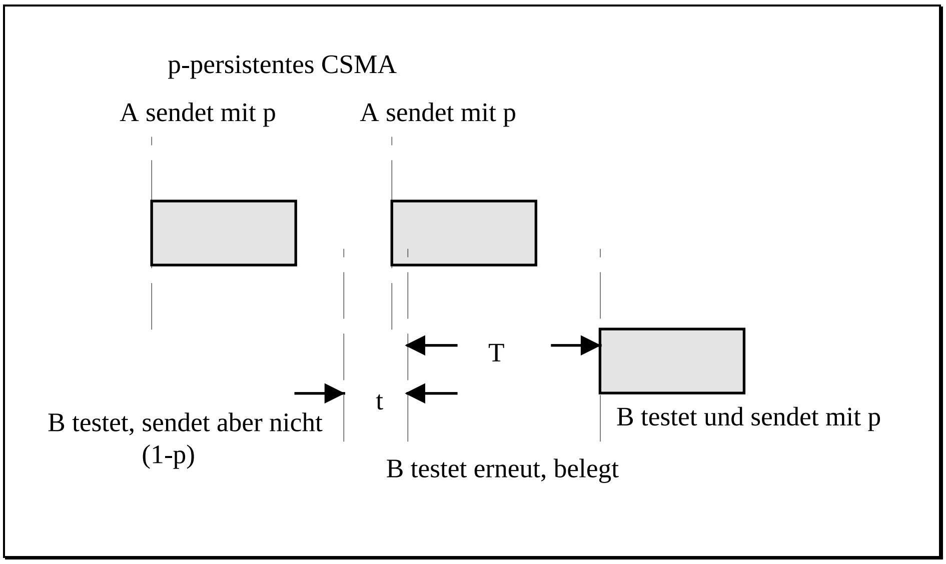

Abb. 8-5: p-persistentes CSMA

Der Parameter p dient dabei der Durchsatzoptimierung.

8.2.2 CSMA/CD (CSMA with Collision Detection)

Beim Auftreten einer Kollision durch gleichzeitige Aussendung zweier Stationen wird im normalen CSMA-Verfahren das komplette Paket weitergesendet, und nur das Ausbleiben einer Quittung oder eine Negativ-Quittung ergibt eine Fehlerbehandlung durch Neusendung. Dieser Weg ist sehr zeitintensiv, so daß insbesondere bei stark belasteten Netzen nach anderen Wegen gesucht werden muß.

Die sendende Station muß zur Kollisionsdetektion lediglich ihr eigenes Signal am Netz lesen und mit dem gesendeten vergleichen. Entsteht hier eine Differenz, so liegt eine Kollision vor, die den Sender zum sofortigen Sendestopp mit einer anschließenden Zufallswartezeit veranlassen. Dies gilt in der Regel für beide Stationen (es gibt auch Abwandlungen hiervon!), so daß beide Sender sich zurückziehen und neu übertragen. Hierdurch wird der Kanal nur für wesentlich kürzere Zeit mit unbrauchbaren Paketen belegt. Auch in diesem ***CSMA/CD-***

Verfahren (*Carrier Sense Media Access with Collision Detection*) unterscheidet man nicht-persistente und p-persistente CSMA/CD.

Die Detektierung der Kollision wird insbesondere durch die Erzeugung eines Störsignals (Jam-Sequence) auf dem Medium gesichert, das bei sicherer Detektierung von einer der beiden kollidierenden Stationen geschickt wird. Andere, ähnliche Verfahren gehen von dominanten und rezessiven Bits aus und veranlassen so die nur erste kollisionserkennende Station zum Rückzug.

8.2.3 Ethernet

Ethernet verwendet das 1-persistente reine (nicht slotted) CSMA/CD-Verfahren, bei dem insbesondere Simulationen gezeigt haben, daß es im Gesamtdurchsatz kaum zu übertreffen ist. Dieses Ethernet existiert in leicht unterschiedlichen Varianten und ist in IEEE 802.x (ISO 8802) beschrieben: IEEE 802.1 beschreibt die Schnittstelle, 802.2 die Schicht 2b (LLC), 802.3 Ethernet (Schicht 2a), während wie bereits erwähnt 802.4 Token-Bus und 802.5 Token-Ring beinhalten.

Die Hardware von Ethernet:

- Koaxialkabel (‘Yellow Cable’) als Thick Wire bis 500 m, mit Repeater bis 2,5 km. Anschlüße sind alle 2,5 m möglich und werden mittels eines Dorns im Kabel angebracht.
- Koaxialkabel als Thin Wire mit BNC-Buchsen oder selbstunterbrechenden Steckern

Die Codierung besteht im Bi Phase Mark (Manchester) mit Signalwerten von 0 und -2V. Der als Kabelanschluß und Codierer dienende Transceiver enthält die Elektronik zum Senden, Empfangen und zur Kollisionserkennung. Prinzipiell können mehrere Stationen an einen Transceiver angeschlossen werden.

Bei einem erwünschten Sendezugriff wird der Kanal auf Aussendung abgetastet und bei Signalfreiheit noch mindestens 9,6 µs (96 Bits) gewartet; ist der Kanal belegt, wird zunächst auf die Unbelegheit gewartet. Nach Sendebeginn liest der Controller ständig am Medium mit. Bei Fehlererkennung wird noch 32 bis 48 Bits weitergesendet, um mittels dieses Jam-Signals die Kollisionserkennung allen aktiven Teilnehmern mitzuteilen.

Die Wartezeit für eine erneute Aussendung ist in einem nicht-persistenten Algorithmus, dem BEB (Binary Exponential Backoff) festgelegt. Der Kanal wird dazu in Zeitschlitze der Länge 51,2 µs (512 Bits), innerhalb derer eine sendende Station auch beim größten Netz sicher erkannt wird eingeteilt. Jede sendebereite Station wählt sich genau einen Schlitz zum Senden aus. Tritt eine Kollision auf, wird die Anzahl der Schlitze verdoppelt, allerdings nicht über 10 Kollisionen hinweg, und ab der 16. Kollision wird der Prozeß abgebrochen und der Mißerfolg an höhere Schichten gemeldet. Ziel dieses Verfahrens ist die schnelle Bereinigung bei geringer Belastung und die sichere Kollisionsauflösung bei hohem Verkehrsaufkommen.

Die Ethernet-Pakete werden Frames genannt:

56-Bit-Preamble 101010 ... 1011	Start-of-frame delimiter (8-Bit)	Destination address (16 or 48 Bit)	Source address (16 or 48 Bit)

...	Length Indicator (16 Bit)	Data (< 12144 Bit)	Pad (optional 8 Bit)	Frame Check Sequence (32 Bit)

Abb. 8-6: Rahmenformat Ethernet

Die in diesem Rahmenformat angegebene 32-Bit-Prüfsumme wird durch das Prüfpolynom

$$CRC32 = x^{32} + x^{28} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^{8} + x^{7} + x^{5} + x^{4} + x^{2} + x + 1$$

gebildet.

Kapitel 9: Vermittlungsprotokolle

9.1 Einleitung

Vermittlungsprotokolle der Schicht 3 werden insbesondere für Netze vorgesehen, bei denen die Rechner nicht an einem (physikalischen) Strang hängen, sondern weit verteilt sind, d.h. ggf. über heterogene Teilnetze mit Kopplungen, Bridges zusammenhängend gestaltet sind.

Die Kommunikation wird in der Regel so organisiert, daß die verschiedenen Rechner über eine ganze Kette von Links, Rechnern, Gateways und Netze miteinander verbunden werden. Standards und Techniken, die diese Probleme behandeln, werden allgemein als Vermittlungsprotokolle bezeichnet. Hierzu werden die beiden gängigsten Formen, das X.25/PLP- und das Internet-Protokoll, im folgenden vorgestellt.

Das ISO/OSI-Basisreferenzmodell führt auf der Schicht 3, der ***Vermittlungsschicht*** (*Network Layer*) ein Protokoll ein, welches eine sichere, transparente Verbindung zwischen zwei Datenendeinrichtungen (DEE, DTE) bereitstellt. Diese Protokoll ist für die sichere Übertragung von Paketen über mehrere Stationen verantwortlich und wird als ***X.25/PLP*** (PLP = *Packet Layer Protocol*) bezeichnet. Bei diesem Protokoll wird davon abstrahiert, daß zwischen den Rechnern noch weitere Stationen sein können, über die die Verbindung führt. Diese Stationen, die das Wissen zum Weitertransport (Adressen!) besitzen müssen, werden als ***Transitsysteme*** bezeichnet und bilden zusammen mit der Datenübertragungseinrichtung (DÜE, DCE) das Datenübertragungssystem.

Das zweite Protokoll, das ***Internetprotokoll*** (IP), hat die in etwa gleichen Aufgaben wie X.25/PLP. Die Übertragung erfolgt wieder über mehrere Transitrechner zum Zielrechner, wobei die Pakete hier ***Datagramme*** genannt werden. Die Protokollunterschiede liegen allerdings im Detail.

Die Dienste in einem Vermittlungsprotokoll umfassen

- den Auf- und Abbau von Systemverbindungen, verbindungslos oder verbindungsorientiert,
- die transparente Übertragung von Benutzerdaten,
- die Meldung nicht behebbarer Fehler,
- eine Vorrangdatenübertragung,
- das Rücksetzen von Endsystemverbindungen sowie
- diverse Qualitätsmerkmale.

Während X.25/PLP alle Dienste anbietet, beschränkt sich Internet auf die wichtigeren (s.u.) und bietet insbesondere keine gesicherte End-to-End-Übertragung an. Diese Sicherung muß in anderen Protokollschichten erfolgen.

9.2 Die Funktionen und Protokollelemente in X.25

9.2.1 Die Funktionen der Vermittlungsschicht im X.25-Protokoll

Die X.25-Empfehlung der CCITT stellt zur Realisierung der Dienste in der Vermittlungsschicht folgende Funktionalitäten zur Verfügung:

- Vermittlung zwischen Endsystemen: Routenwahl, Betriebsmittelzuordnung.
- Konzentrieren (Multiplexen) mehrerer Endsystemverbindungen auf eine gesicherte Systemverbindung zur kostengünstigen Ausnutzung der Verbindung.
- Übertragung von Daten: Normale Daten werden in einer Warteschlange abgelegt, während Vorrangdaten vor den normalen Daten übertragen werden.
- Fragmentierung und Blockung von Benutzerdaten (einschließlich der Schichten 4 bis 7), damit diese von der Schicht 2 (Sicherung) übertragen werden können.
- Fehlererkennung und Behebung bei der Datenübertragung, um eine vereinbarte Qualität des Übertragungsdienstes zu garantieren. Hierzu zählt z.B. die Übergabe der Fragmente bzw. Blöcke in der richtigen Reihenfolge, da die Blöcke durchaus verschiedene Wege gehen können.
- Flußregelung zur Steuerung des Datenstroms zwischen den Vermittlungsschichten der DEE.

X.25 beschreibt sowohl verbindungsorientierte, ***virtuelle Verbindungen*** als auch ***Datagrammdienste***, die verbindungslos sind. Die einzelnen Implementierungen von X.25 weichen hiervon teilweise ab, so unterstützt das deutsche Datex-P keine verbindungslosen Dienste. Bei den virtuellen Verbindungen (***VC = Virtual Call***) unterscheidet man feste virtuelle Verbindungen (***PVC = Permanent Virtual Calls***) sowie wählbare virtuelle Verbindungen (***SVC = Switched Virtual Calls***).

Mehrere Verbindungen (zwischen den gleichen oder verschiedenen Endstationen) können gleichzeitig betrieben werden. Dazu wird jede Verbindung einem logischen Kanal mit einer Kanalgruppennummer (0 .. 15) und einer Kanalnummer (0 .. 255), die die Verbindung aus den 4096 pro Endstation möglichen identifiziert. Festen Verbindungen ist dabei im Rechner ständig ein logischer Kanal zugeordnet, während wählbare beim Verbindungsaufbau einen Kanal zuwiesen bekommen. Die Aufteilung in feste und wählbare Kanäle erfolgt bei der Einrichtung des Anschlusses.

9.2.2 Das Modell

Während des Verbindungsaufbaus kommuniziert die Schicht 3 zunächst mit der DÜE, um die Zieladresse und andere zur Kommunikation notwendigen Parameter mitzuteilen. Nachdem die Verbindung aufgebaut ist, wird die DÜE als transparentes Medium angesehen, die Vermittlungsschicht kommuniziert nur direkt mit der Partnerschicht des Zielrechners. Das Kommunikationsmodell vereinfacht sich also während der Übertragungsphase zu einer direkten Verbindung:

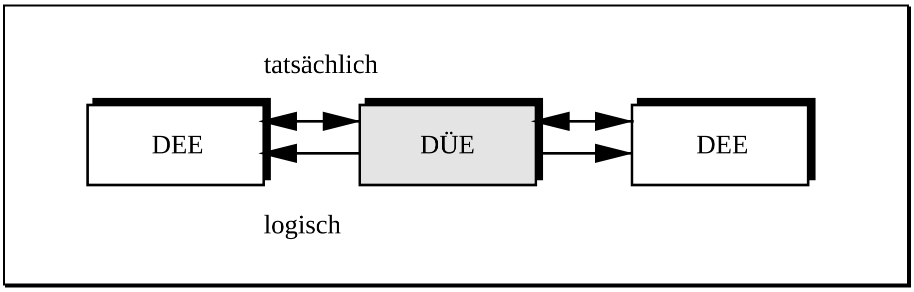

Abb. 9-1: Übertragungsmodell Vermittlungsschicht

Der obere Zweig wird bei der Kommunikation tatsächlich (mit Rechenleistung) durchlaufen, während die Endteilnehmer nur die logische Verbindung in Abb. 9-1 sehen.

Die Schicht 3 ist in folgender Weise in das ISO/OSI-Basisreferenzmodell eingebettet:

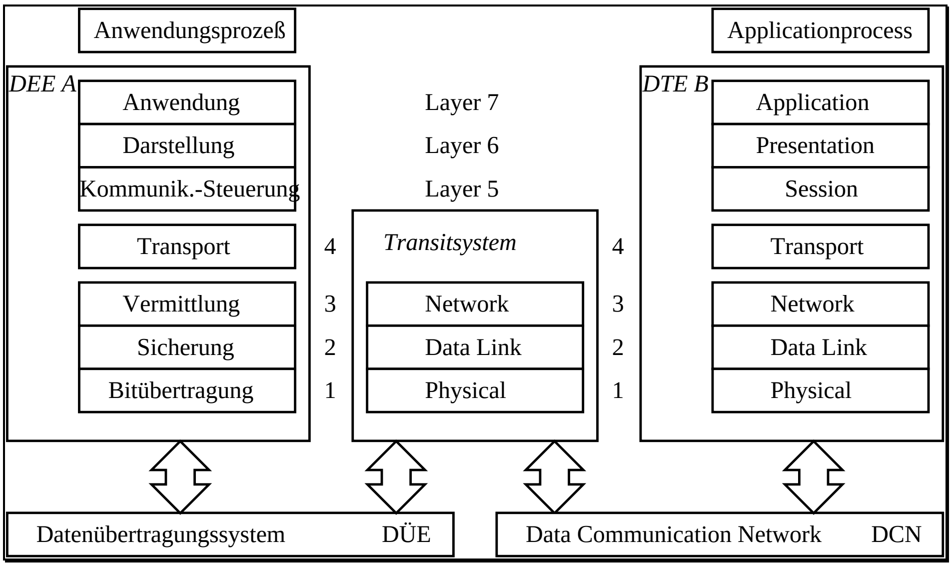

Abb. 9-2: ISO/OSI-Basisreferenzmodell

9.2.3 Die verschiedenen Pakettypen

Für die Übermittlung von Daten zwischen den Endstationen sowie die Kommunikationssteuerung ist es notwendig, verschiedene Pakettypen im Netz zu definieren. Diese sind in 5 Gruppen zusammengefaßt:

- Verbindungserstellung und -auslösung (*Call Set-Up and Clearing*)
- Daten und Unterbrechungen (*Data and Interrupt*)
- Flußregelung und Rücksetzen (*Flow Control and Reset*)
- *Restart*
- Diagnose (*Diagnostic*)

Die Pakete sind mindestens 3 Bytes lang; für diese ersten drei Bytes gilt ein Standardformat:

Byte	Bit 8	Bit 1
1	Kennzeichen für das Grundformat	Logische Kanalgruppennummer
2	Logische Kanalnummer	
3	Kennzeichen für den Pakettyp	

Abb. 9-3: Standardformat X.25 (ersten 3 Bytes)

Das ***Grundformat*** bestimmt hierbei die Numerierung der Pakete, die entweder modulo 8 (Kennzeichen = 0001) oder modulo 128 (Kennzeichen = 0010) numeriert werden.

Die ***logische Kanalgruppennummer*** enthält die Kanalgruppennummer des logischen Kanals, für den das jeweilige Paket bestimmt ist. Die angebbaren Werte sind aus dem Intervall 0 ... 15. Entsprechend wird in der ***logischen Kanalnummer*** die (Empfänger-) Kanalnummer von 0 ... 255 eingetragen.

Der ***Pakettyp*** wird in codierter Form im Kennzeichen für den Pakettyp angegeben.

9.2.3.1 Das Restart-Paket

Eine Station kann einen ***Restart*** initiieren, indem ein *Restart Request* abgesetzt wird. Dieses Paket wird von der nächstzugehörigen DÜE abgefangen und mit einer *Restart Confirmation* bestätigt. Die DÜE startet dann eine Reihe von Initialisierungsfunktionen, die alle logischen Kanäle initialisiert.

Die bisher bestandenen gewählten virtuellen Verbindungen sind anschließend aufgelöst, die festen virtuellen Verbindungen zurückgesetzt.

9.2.3.2 Verbindungserstellung und -auslösung

Der Aufbau virtueller Verbindungen benötigt spezielle Pakete. So wird zum Aufbau eines SVC (Switched Virtual Call) ein Paket vom Typ ***Verbindungsanforderung*** abgesetzt, welches bei der empfangenen DEE als ankommender Anruf interpretiert wird. Dies erreicht die DEE in folgender Weise:

Byte	Bit 8	Bit 1
1	Kennzeichen für das Grundformat	Logische Kanalgruppennummer
2	Logische Kanalnummer	
3	Kennzeichen für den Pakettyp Verbindungsanforderung 0 0 0 0 1 0 1 1	
4	Länge der Adresse der rufenden DEE	Länge der Adresse der gerufenen DEE
5	DEE-Adressen 0 0 0 0	
	0 0	Länge des Feldes zur Angabe von Leistungsmerkmalen
	Leistungsmerkmale	
	Angaben des rufenden Teilnehmers	

Abb. 9-4: Ankommender Anruf in X.25

Die Angaben zum rufenden Teilnehmer können beispielsweise Informationen über die höheren Schichten und dortige Protokollvarianten enthalten. Die empfangene Stelle sendet ein Paket vom Typ ***Verbindung hergestellt***.

Die Auslösung der Verbindung wird in entsprechender Weise durchgeführt; die Folge der Pakete zum tatsächlichen Aufbau wird im nächsten Unterabschnitt erläutert.

9.2.3.3 Die Pakettypen zur Datenübertragung und Unterbrechung

Nach Aufbau einer Verbindung können beide DEEs unabhängig voneinander Datenpakete senden und empfangen. Diese Pakete haben den folgenden Aufbau:

Byte	Bit 8			Bit 1
1	Kennzeichen für das Grundformat Q D 0 1		Logische Kanalgruppennummer	
2	Logische Kanalnummer			
3	P(R)	M	P(S)	0
4 ...	Benutzerdaten			

Abb. 9-5: Datenpakettyp X.25

Die besondere Kennzeichnung des Grundformats macht eine sofortige Zuordnung dieses Pakettyps möglich: Bit 1 im dritten Byte ist 0, dieses Bit ist in allen anderen Paketen 1. Die Bedeutung der weiteren Bezeichnungen:

Q im ersten Byte wird im Protokoll X.29 benötigt. Das Bit ***D*** (*Delivery Confirmation Bit*) dient der Anforderung der Übergabebestätigung. Ist das Bit ***M*** (*More Data Bit*) auf 1 gesetzt, ist das Paket nicht das letzte in einer Folge; beim letzten und ggf. einzigen Paket wird M = 0 gesetzt.

P(R) und P(S) entsprechen den Folgezählern aus dem HDLC-Protokoll, Layer 2. Hier wird modulo 8 gezählt (es ist auch modulo 128 möglich), wobei die maximale Anzahl an unbestätigten Paketen als ***Fenstergröße W*** standardmäßig auf 2 (durch die Teilnehmer änderbar) festgesetzt ist.

Während der Datenphase kann eine DEE der anderen eine Nachricht als ***Unterbrechung*** zukommen lassen. Diese wird von der Gegenstelle mitttels einer Unterbrechungsbestätigung quittiert, wobei hier die Fenstergröße definitiv auf 1 festliegt, da sonst nicht alle Unterbrechungen bestätigt werden könnten (keine Paketzähler).

9.2.3.4 Die Pakettypen zur Flußregelung und zum Rücksetzen

Die gegenseitige Flußregelung, die bei HDLC als dem zugehörigen Schicht-2-Protokoll ebenfalls vorhanden ist, muß in Schicht 3 zusätzlich implementiert werden, da diese Schicht letztlich ebenfalls Verarbeitungszeit benötigt und somit überlaufen werden kann. Die zugehörigen Pakete besitzen den Typ ***RNR*** (*Receive Not Ready*).

Die erneute Empfangsbereitschaft muß durch explizites Aussenden eines ***RR***-Pakets (*Receice Ready*) bekanntgegeben werden. Beiden Paketen wird die Empfangsmeldung von Paketen durch einfaches Übermitteln des Empfangszählers mitgegeben.

Falls es nötig ist, feste oder virtuelle Verbindungen in einen definierten Anfangszustand zu versetzen, kann dies durch das ***Rücksetzen*** (*Reset*) geschehen. Hierdurch werden sämtliche Zähler auf 0 gesetzt sowie alle sich noch im Netz befindlichen Pakete vernichtet. Der Empfänger muß dieses Paket durch eine ***Rücksetzbestätigung*** beantworten.

9.2.4 Ablauf des Vermittlungsprotokolls im X.25/PLP

Der Verbindungsaufbau für X.25/PLP-Protokolle läßt sich wieder in zeitlicher Abfolge darstellen, ähnlich wie dies im HDLC-Protokoll geschehen ist:

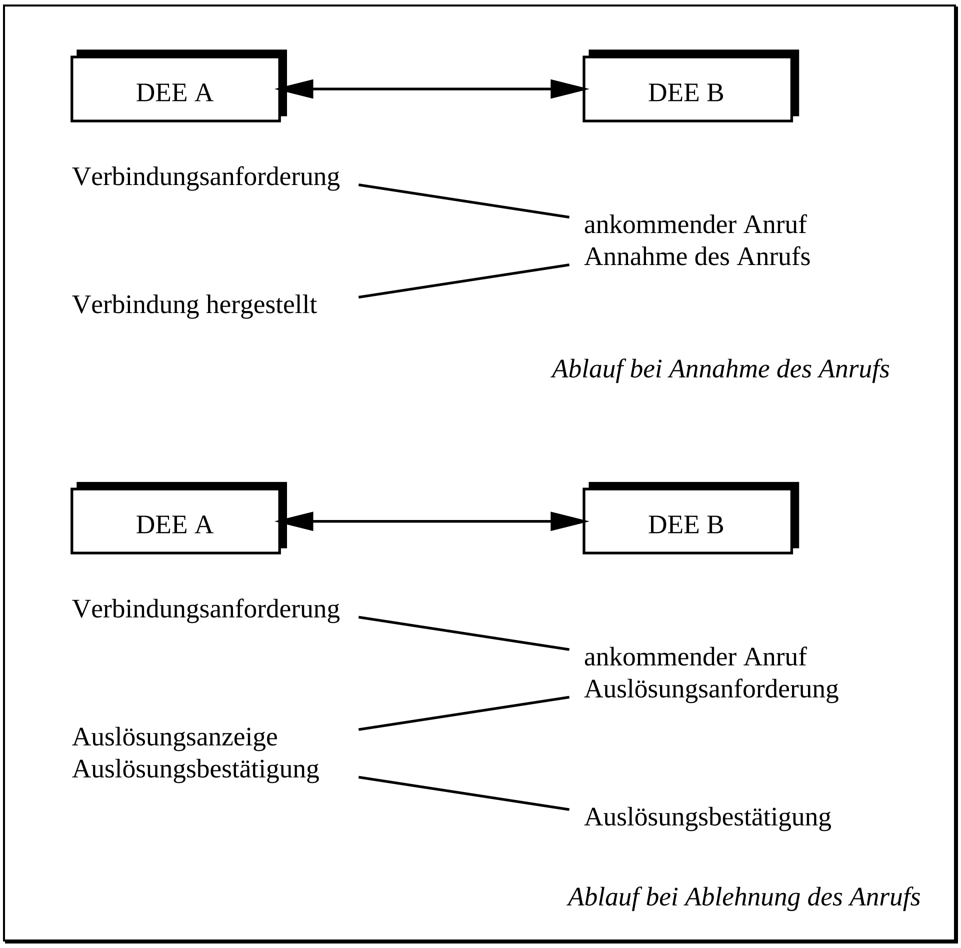

Abb. 9-6: Verbindungsaufnahme X.25/PLP

Bei der Verbindungsaufnahme sind innerhalb der Stationen auch Lösungsstrategien für Konfliktfälle vorzusehen. Als solcher kann beispielsweise der Fall angesehen werden, daß eine Station den letzten freien Kanal belegen will, in diesem Fall jedoch eine Verbindung von außen angefordert wird; letztere hat dann Vorrang.

Die Verwendung von logischen Kanalgruppen und Kanälen bedingt zusätzlich eine Namensumsetzung im Netz. Dies wird notwendig, da viele Stationen im Netz die gleichen Nummern wählen können, das Netz sich jedoch selbst verwalten muß. Die Namensumsetzung ist ein durchaus kompliziertes Verfahren.

9.3 Das Internet-Protokoll

9.3.1 Die Funktionen der Vermittlungsschicht im Internet Protokoll

Das Vermittlungsprotokoll des Internet stellt die folgenden Funktionen zur Verfügung:

- Übertragung von Datagrammen zwischen Endsystemen (Routenwahl, Prozeßzuordnung)
- Übertragung von Vorrangdaten

- Fragmentierung und Reassemblierung von Benutzerdaten, damit diese vom Transportmedium übertragen werden können.
- Erkennung und Meldung von Systemfehlern, keine Erkennung von Datenverfälschungen jedoch
- Flußregelung zur Steuerung des Datenstroms zwischen den Routern

Das Internet überträgt Datenpakete in Form von Datagrammen, welche alle die Zieladresse sowie weitere Kontrollinformationen enthalten, mindestens 40 Bytes. Die einzelnen Prozesse werden im Zielrechner durch eine Protokollnummer identifiziert. Da die Kommunikation verbindungslos ist, kann ohne weiteres konzentriert werden.

Bei Verfälschung der Kontrollinformationen wird eine Meldung an den Absender abgeschickt, Dateninformationen werden – wie erwähnt – nicht kontrolliert. Die Flußregelung erfolgt innerhalb eines eigenen Managementprotokolls (***ICMP = Internet Control Message Protocol***).

9.3.2 Unterschiede zwischen Internet- und X.25-Protokoll

Die Kommunikation im Internet verläuft verbindungslos: Aus diesem Grund werden die Dateneinheiten zum Versand innerhalb der Internets ***Datagramme*** genannt. Eine Konsequenz aus diesem Vorgehen ist die Versorgung mit Volladressen für jedes Datagramm durch den Nutzer, da er keine virtuellen Verbindungen mit logischen Kanälen vorliegen hat.

Andererseits benötigt das Internet keine Adreßumsetzung, wie dies für X.25/PLP kurz diskutiert wurde. Die Protokollhierarchie in der TCP/IP-Protokollfamilie läßt sich folgendermaßen darstellen:

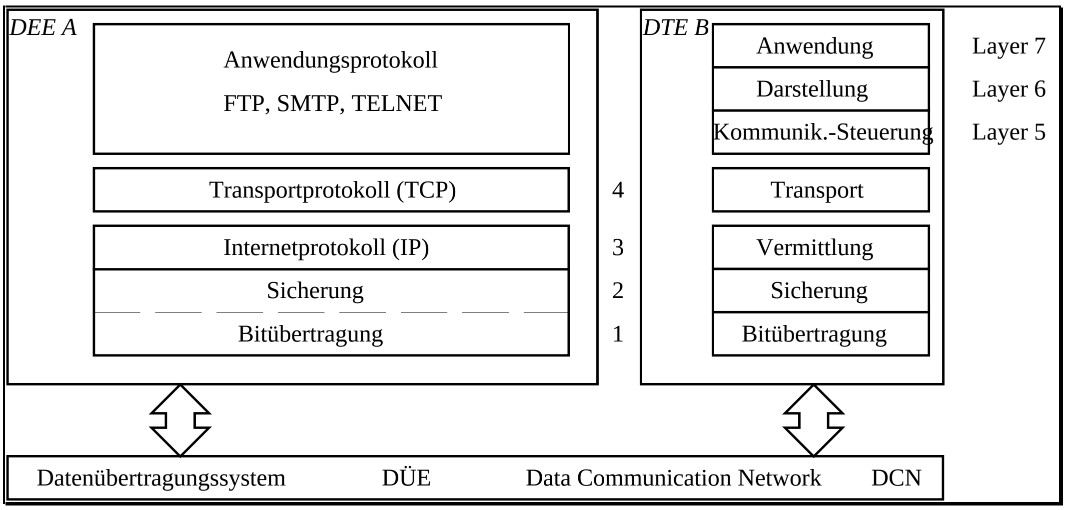

Abb. 9-7: Schichten im Internet

Die Protokollschichten 1 und 2 sind im Internet nicht weiter definiert, im Gegensatz zu X.25/PLP. Hier kann z.B. Ethernet, HDLC usw. zum Einsatz kommen.

Das Internet bedient sich weiterhin der ***Best Effort Strategie***, d.h., redlich um einen korrekten Datentransport, jedoch ohne Sicherheit dafür zu bieten. Sicherungen müssen daher in höheren Schichten integriert werden, um einen sicheren Datentransport zu gewährleisten.

9.3.3 Der Aufbau der Datagramme

Der grundsätzliche Aufbau der Datagramme ähnelt denen im X.25/PLP insofern, daß jedes Datagramm aus einem Datagrammkopf (Datagram Header) und einem Datenbereich (Data Area) besteht. Diese Datagramm ist in der Regel noch von einem Rahmen der Schicht 2 umgeben, muß also in diesen Rahmen hineinpassen.

Hieraus resultiert die Forderung für Internet, mit der Blockgröße (Schicht 3) an alle einsetzbaren Übertragungsmedien mit deren verschiedenen Blockgrößen (Schicht 2) anzupassen. Da diese von 128 (X.25, welches als Transport für Internet gebraucht wird) bis 4500 (FDDI) Bytes variieren, wäre mit 128 ein schlechter Kompromiß gefunden, da die Datagrammkopflänge immer dieselbe Anzahl an Bytes benötigt (vollständige Adressierung), während für den Datenbereich teilweise sehr wenig übrigbleibt.

Die Gefahr eines sehr ineffizienten Übertragungssystems wird dadurch gemildert, daß die Standardblockgröße auf 576 Bytes definiert wird. Für Netze, die nur kleinere Blöcke übertragen können, dürfen fragmentierte Datagramme, die dann alle notwendigen Informationen enthalten (müssen), verwendet werden.

Der Aufbau eines Datagramms ist in Abb. 9-8 dargestellt.

<table>
<tr><td>0</td><td>4</td><td>8</td><td colspan="3">16</td><td>19</td><td>24 31</td></tr>
<tr><td>Vers.</td><td>Länge</td><td>Diensttyp</td><td colspan="5">Gesamtlänge</td></tr>
<tr><td colspan="3">Identifikation</td><td>M</td><td>D</td><td></td><td colspan="2">Fragmentabstand</td></tr>
<tr><td colspan="2">Lebenszeit</td><td>Transport</td><td colspan="5">Kopfprüfsumme</td></tr>
<tr><td colspan="8">Senderadresse</td></tr>
<tr><td colspan="8">Empfängeradresse</td></tr>
<tr><td colspan="7">Optionen</td><td>Füllbits</td></tr>
<tr><td colspan="8">Daten</td></tr>
<tr><td colspan="8">.....</td></tr>
</table>

Abb. 9-8: Datagramm-Format im Internet

In diesem Format haben die einzelnen Felder folgende Bedeutung:

- ***Version***: Dieses Feld gibt die Versionsnummer des Protokolls an, nach der das Datagramm aufgebaut ist. Zur Zeit Version 4 (im Wechsel auf 6).

- ***Länge***: Diese Feld gibt die Länge des Datagrammkopfes in Vielfachen von 32 Bit an. Zumeist steht hier eine '5', da der eigentliche Datagrammkopf bei 'Lebenszeit' beginnt.

- ***Diensttyp***: Im Diensttyp sind folgende Informationen untergebracht: Die ersten 3 Bits stehen für die Priorität des Datagramms, wobei '000' die geringste, '111' die höchste bedeutet. Diese Bits werden in einem Knoten interpretiert, falls dieser die Wahl zur Sendung von Paketen unterschiedlicher Prioritäten hat.

  Die nächsten drei Bits stehen für D (möglichst geringe Verzögerung, Delay), T (möglichst hohen Durchsatz, Throughput) und R (möglichst hohe Zuverlässigkeit, Reliability). Dies kann bei Knoten mit alternativen Wegewahlen eine Entscheidungshilfe bieten, wird aber gegenwärtig ignoriert. Die restlichen 2 Bits dieses Feldes haben keine Bedeutung und werden auf 0 gesetzt.

- Die ***Gesamtlänge*** gibt die Länge der Daten einschließlich des Datagrammkopfes in Bytes an.

- Die Felder ***Identifikation***, ***M***, ***D*** und ***Fragmentabstand*** steuern die Fragmentierung und Reassemblierung.

- Das Feld ***Lebenszeit*** soll die maximale Lebenszeit eines Datagramms in Sekunden beinhalten. In der Praxis wird dieses Feld jedoch durch die durchlaufenen Knoten dekrementiert, so daß deren Anzahl hier enthalten ist. Der Sinn dieses Felds liegt in der Zerstörung (Lebenszeit = 0) herumirrender Datenpakete.

- Das Feld ***Transport*** gibt die zentral vergebene Nummer (*Network Information Center*) des nächsthöheren Protokolls an. Diese Protokolle sind in Request for Comments veröffentlicht.

- Die ***Kopfprüfsumme*** enthält die Summe aller Worte (= 2 Oktetts) im Header modulo 65536, mit Ausnahme der Kopfprüfsumme selbst. Das zugehörige 1er Komplement wird in dieses Feld eingetragen, so daß die erneute Prüfsumme 0 ergeben würde.

  Da die Lebenszeit im Kopf ständig verändert wird, wird auch die Kopfprüfsumme ständig aktualisiert. Dies kann dadurch relativ schnell geschehen, da nur die Kopfdaten miteinbezogen werden. Der Prüfsummenschutz der Nutzdaten ist Aufgabe höherer Protokolle.

- ***Senderadresse*** und ***Empfängeradresse*** enthalten die jeweiligen Adressen (siehe nächsten Abschnitt)

- Die ***Optionen*** werden für das Network Management sowie zur Erprobung eingefügt. Sie werden mit 0-Bits auf die Anzahl von 32 Bits aufgefüllt.

9.3.4 Adressierung im Internetprotokoll

Wie bereits erwähnt, zwingt die verbindungslose Kommunikation im Internet zur vollständigen Angabe der Sender- und Empfängeradresse. Die Adressierung wird dadurch sehr wichtig, und sie muß dem Benutzer transparent gemacht werden.

Internet nutzt 32 Bit für die Adressierung, dies ergäbe über 4 Milliarden Adressen, eine Zahl, die sicher vorerst ausreicht, um den Adressenbedarf auf der Erde zu stillen. Andererseits müßte eine Wegewahl in einem Knoten die volle Kenntnis über die Erreichbarkeit der Rechner besitzen, also eine solche Tabelle vorfinden. Da sich dies als sehr aufwendig erweist, werden selbstlenkende Adressen verwendet, die sowohl das Netz als auch den Rechner im Netz eindeutig beschreiben. Dadurch wird zwar nicht die maximal mögliche Anzahl an Rechnern zur Verfügung gestellt, die Internetadresse bekommt jedoch den Charakter einer Telefonnummer.

Im Format selbst werden noch verschiedene Formatklassen gewählt:

0 4 8 16 24 31

Klasse A	0	Netz-ID	Rechner-ID
Klasse B	1 0	Netz-ID	Rechner-ID
Klasse C	1 1 0	Netz-ID	Rechner-ID
Klasse D	1 1 1 0	Rechner-ID	

Abb. 9-9: Adressierungsformate im Internet

Diese Darstellung ist so zu interpretieren: Eine Klasse-A-Adresse besitzt das Format A.x.x.x, A und x jeweils ein Byte lang, in dezimaler Schreibweise mit Punkten voneinander getrennt, und kann demnach theoretisch 128 Internet-Adressen enthalten, wohinter sich dann wiederum eine ungeheure Anzahl von Rechner-Adressen befinden. A wird im Bereich 0 .. 9, 11 .. 126 verteilt, so daß 126 Class-A-Adressen entstehen.

In Klasse B erfolgt die Adressierung durch B1.B2.x.x, so daß bei $128 \le B1 \le 191$, $0 \le B2 \le 255$ (mit den Ausnahmen 172.16.x.x bis 172.31.x.x) 16368 Internetadressen mit jeweils 65536 Rechnern verwaltet werden können. Dieses Reservoir ist zur Zeit ca 55 - 60% ausgeschöpft, so daß gerne Klasse-C-Adressen verteilt werden.

In Klasse C gilt C1.C2.C3.x, also eine Netz-Adresse für 256 Rechner mit $192 \le C1 \le 223$ und $0 \le C2, C3 \le 223$. Dieses Kontigent ist erst zu 4 % ausgeschöpft. Klasse D und E-Adressen werden zur Zeit erprobt.

Prinzipiell können bei den Rechner-IDs noch Subnetze eingeführt werden, diese sind jedoch nicht reglementiert und stehen in der Verfügbarkeit des Netzwerkadministrators. Die 4-Byte-Adressen werden im übrigen nicht vom Nutzer selbst, sondern von Domain Name Servern mit der Übersetzungsfähigkeit zwischen textuellen und numerischen Internetadressen (sowie einer Zweitadressengenerierung) erzeugt.

9.3.5 Wegewahl im Internetprotokoll

Das Problem der ***Wegewahl*** (Routing) ist allgemein: Im Netz soll der geeignetste Weg gefunden werden. Hierbei ist das Netz keineswegs als konstant anzunehmen, sondern es befindet sich ständig in Erweiterung (Beispiel: In den USA existiert ein Backbone-Netz mit 45 Mbit/s Übertragungsrate, das ständig erweitert wird).

Für das Routing (ein Rechner, der keine eigenen Datagramme erzeugt, sondern nur weiterleitet, wird gewöhnlich als Router bezeichnet) werden zunächst ***direkte und indirekte Ziele*** unterschieden. Direkte Ziele findet der Router im gleichen Netz, was durch Maskieren der Netzadresse sofort festgestellt werden kann.

Für indirekte Ziele muß der Router eine Liste vorfinden, zu welchem Router er das Datenpaket zu schicken hat; darauf folgt das entsprechende Verpacken im Netzformat und Abschicken. Findet der Router keine Information, gibt es meist eine Ausweichregelung, sprich einen Router, der noch mehr Informationen hat, oder es wird eine Fehlermeldung generiert.

Alternativ hierzu kann der Nutzer im Optionenfeld den Weg selbst festlegen.

Kapitel 10: Transportprotokolle

Der Sinn der Einführung von Transportprotokollen liegt zunächst in dem Verdecken der netzwerkabhängigen Schichten und deren Implementierungsdetails. Das Ziel einer Transportschicht ist es also, dem Benutzer eine einheitliche Schnittstelle zur Kommunikation anzubieten.

Dieses Ziel kann sich auch in gewissem Sinn verselbständigen. Wenn die Details des Netzwerks schon vor dem Nutzer verborgen bleiben sollen, dann kann auch eigentlich der gewählte Weg verborgen bleiben. Dem Nutzer wird also ein funktionierender Übertragungsweg angeboten, ohne detaillierte Mitteilungen über die tatsächliche Beschaffenheit.

Dieser Ansatz funktioniert natürlich nur so lange, wie es der Nutzer mitmacht oder wie er zum Mitmachen gezwungen werden kann. Das in den 70er Jahren entwickelte Basisreferenzmodell nach ISO/OSI verfolgt jedoch keinen streng objektorientierten Ansatz, sondern den damals üblichen struktureirten Ansatz. Dementsprechend kann ein Programm, das einen Netzdienst benötigt, über die Transportschicht hinweg auf die niedrigeren Dienste zugreifen, ein sehr gängiges Verfahren.

Weiterhin bietet die Transportschicht nunmehr eine Abstrahierung von der Rechner-Rechner-Kommunikation. Bislang waren die Protokolle auf die Kopplung von Rechnern bezogen, aufsteigend von einer physikalischen Kopplung bis hin zur gesicherten Datenkommunikation. Prinzipiell liegt das Interesse jedoch nicht in der Kommunikation der Rechner, sondern der Anwendungen. Die Transportschicht soll daher eine Anwendungs-spezifische Plattform bieten.

10.1 Die OSI-Umgebung

Die OSI-Umgebung entstammt – wie bereits erwähnt – in ihrer Terminologie den 70er Jahren: Die Schichten waren gegeneinander insoweit abgegrenzt, daß eine Schicht der darüberliegenden Dienste anbietet und angebotene Dienste der unmittelbar darunterliegenden nutzt. Dieses Modell einer strukturierten Software entspricht dem Stand der damaligen Technologie – Objektorientierung war noch in unbekannter Ferne, Kapselung damit ebenfalls. Dies bedeutet, daß der Durchgriff auf tiefere Schichten durchaus möglich ist.

Die Daten wiederum wurden zu Datensätzen (Records) zusammengefaßt, so daß insgesamt zwei Einheiten wichtig sind: Funktionen als Dienste (***Services***) und Datenstrukturen als Protokolldateneinheiten (PDU, ***Protocol Data Unit***).

10.2 Transportprotokolle des ISO/OSI-Basisreferenzmodell

Die Spezifikation des Transportprotokolls nach ISO/OSI ist in der Spezifikation X.214/X.224 festgehalten. Hierin sind die Adressierung, die Dienste, die Transportprotokollklassen sowie das eigentliche Transportprotokoll definiert.

10.2.1 Adressierung im ISO/OSI-Basisreferenzmodell

Die Identität der Kommunikationspartner wird innerhalb einer OSI-Umgebung zumeist mittels logischer Namen mitgeteilt, in der Netzwerkumgebung jedoch mittels Adressen. Der ***symbolische Name*** bezieht sich dabei auf den Anwendungsprozeß und muß innerhalb einer Umgebung eindeutig identifizierbar sein. Die Verteilung erfolgt durch einen ***Name Server***.

Die Verteilung der eindeutig identifizierenden Adressen (wieder für jeden Anwendungsprozeß) erfolgt im ***Systemverzeichnis***.

Innerhalb einer OSI-Umgebung sind Adressen Aneinanderreihungen von Dienstzugangspunkten (***Service Access Points***): Hierzu zählen der *Presentation Service Access Point* (PSAP), *Session Service Access Point* (SSAP), *Transport Service Access Point* (TSAP) und *Network Service Access Point* (NSAP). Diese Bezeichnungen haben durchaus direkte Abbildungen im Systemdesign: Der NSAP fällt mit der netzweiten Adresse zusammen, die TSAPs werden häufig als Sockets (BSD-Unix) oder Streams (Unix V) bezeichnet.

10.2.2 Die Dienste der Transportschicht

In der verbindungsorientierten Datenübertragung exisitieren drei Hauptphasen: Der *Verbindungsaufbau*, die *Datenphase* und der *Verbindungsabbau*. Diesen drei Phasen entsprechen die Dienste `T.CONNECT`, `T.DATA` und `T.DISCONNECT`, deren Elemente aufgerufen werden: `T.CONNECT.REQUEST` fordert z.B. eine Verbindung an, die durch T.CONNECT.CONFIRM bestätigt wird; bei Ablehnung erfolgt ein T.DISCONNECT.INDICATION. Der Verbindungsabbau wird entsprechend durch T.DISCONNECT.REQUEST eingeleitet.

Während der Datenphase werden über T.DATA.INDICATION Daten übermittelt und via T.DATA.REQUEST solche angefordert. Da der Übertragungsweg sicher sein soll, gibt es an dieser Stelle keine Bestätigung.

Zusätzlich sind Dienste für Vorrangdaten definiert (`T.EXPEDITED-DATA`) sowie ein selten eingesetzter verbindungsloser Dienst (`T.UNITDATA`).

10.2.3 Die Transportprotokollklassen

ISO/OSI definiert verschiedene Qualitätsgrade eines Transports (***Quality of Service, QoS***). Hierzu gehören folgende Klassen:

- Transportprotokollklasse 0: ***Einfache Klasse*** (*Simple Class*): Aufbau von einfachen Verbindungen, bei Fehlern der unteren Schichten sofort wieder abgebaut, verlorengegangene Datenpakete werden nicht erkannt.
- Transportprotokollklasse 1: ***Basic Error-Recovery Class***: Einfache Verbindungen werden aufgebaut, im Fehlerfall auch eine automatische Wiederaufnahme, nur wenn diese fehlschlägt, wird der Teilnehmer informiert. Vorrangdaten können übermittelt werden.
- Transportprotokollklasse 2: ***Multiplexklasse***: Einfache Verbindungen wie die der Klasse 0 werden unterstützt, allerdings werden über eine Netzwerkverbindung mehrere Transportverbindungen geführt. Der Sinn liegt in einer Kostenersparnis.
- Transportprotokollklasse 3: ***Fehlererholung und Multiplexklasse*** (*Error-Recovery and Multiplexing Class*): Die Eigenschaften von Klasse 1 und 2 werden kombiniert.
- Transportprotokollklasse 4: ***Fehlerentdeckungs- und -erholungsklasse*** (*Error-Detection and Recovery Class*): Außer den Diensten der Klasse 3 werden zusätzlich Fehler bei der Datenübertragung korrigiert. Diese Klasse ist bei unsicheren LANs empfohlen, besitzt jedoch einen großen Overhead.

10.2.4 Das Transportprotokoll

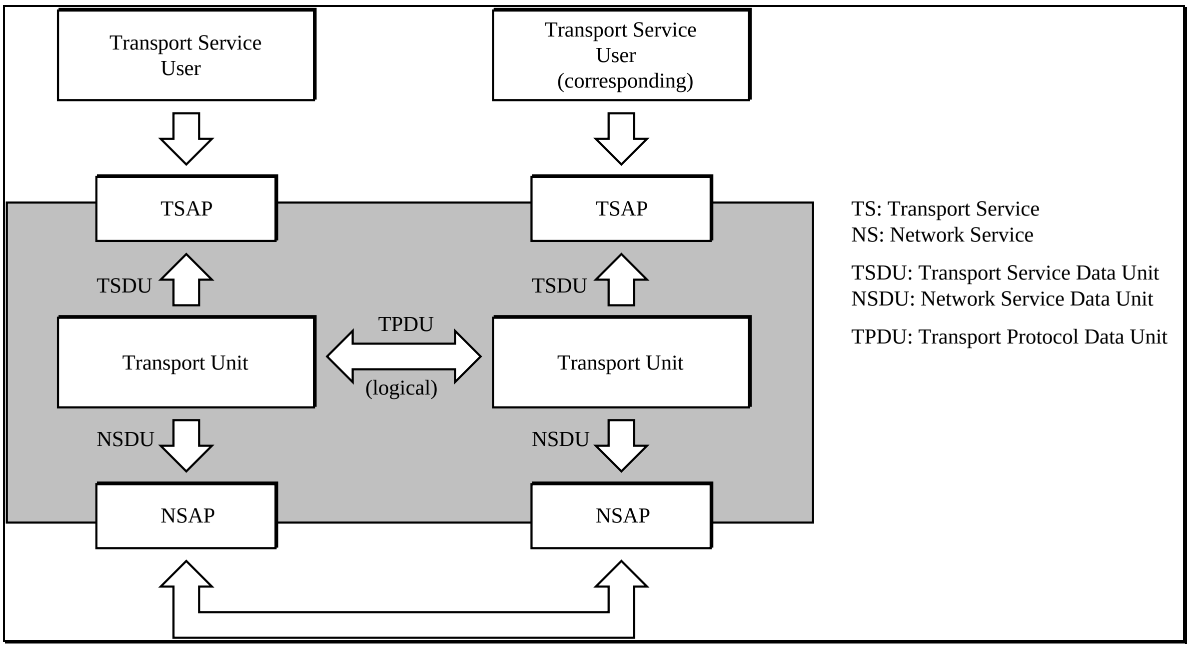

Abb. 10.1: Modell der Transportschicht

Gemäß dem ISO/OSI-Basisreferenzmodell soll die Transportschicht Endsystemverbindungen zu Teilnehmerverbindungen erweitern. Dies bedeutet, daß die Rechner-Rechner-Kommunikation zu

Anwendungsprozeßkommunikationen erweitert wird. Der Anwendungsprozesse werden auch als ***User*** bezeichnet. Die Kommunikation erfolgt gemäß dem in Abb. 10.1 dargestellten Modell.

Mit Hilfe von Programmiersprachen und entsprechenden Laufzeitsystemen können nunmehr Anwendungsprogramme auf die Dienste der Transportschicht zugreifen. Dies führt auch in Single-Tasl-Systemen zur Notwendigkeit von Multitasking, beispielsweise zur Verwaltung von Warteschlangen. Es existiert sogar eine Hochsprache namens CHILL (CCITT High Level Language) zur Unterstützung der Programmierung.

10.3 Transportprotokolle im Internet

Die Transportschicht der TCP/IP-Protokollfamilie umfaßt im wesentlichen zwei Protokolle: ***TCP*** (*Transport Control Protocol*) und ***UDP*** (*User Datagram Protocol*). TCP ist dabei verbindungsorientiert (IP war es nicht!), UDP setzt direkt auf IP auf und stellt somit ein verbindungsloses Datagramm-Protokoll zur Verfügung.

10.3.1 TCP Reliable Stream Transport Service

TCP hat folgende Merkmale:

- ***Verbindungsorientiert***: Zwischen zwei kommunizierenden Stationen wird zur Kommunikation eine virtuelle Verbindung aufgebaut (und anschließend wieder abgebaut).
- ***Vollduplex-Verbindung***: Beide Anwendungsprozesse können unabhängig voneinander Daten übermitteln.
- ***Zuverlässigkeit***: TCP garantiert eine hohe Zuverlässigkeit der Datenübermittlung bezüglich Verlust und Verfälschung.
- ***Stromorientiert***: Jede Folge von Bits/Bytes wird unverändert übertragen.
- ***Gepufferte Übertragung***: Daten werden byteweise an die Kommunikationsinstanz geliefert, die sie zwischenspeichert und gesammelt überträgt. Eine gezielte Übertragung kann erzwungen werden (Push-Mechanismus).
- ***Unstrukturierter Strom***: Die Struktur der Daten muß von den Anwendungsprozessen selbst vereinbart und überwacht werden.

TCP bietet damit einen gesicherten Transportdienst für Anwendungen an. Jede Übertragung beginnt mit einem Header, in dem die Portnummer von Quell- und Zielrechner explizit angegeben sind. Diese Nummern identifizieren die Applikation; sie sind teilweise fest vergeben, beispielsweise 5 für Remote Job ntry, 25 für Electronic Mail, können andererseits vom Betriebssystem frei vergeben werden.

In dem Header werden weiterhin Folgenummer (im Sendestrom), Quittungsnummer (für empfangene Daten), eine Codierung für den Dateninhalt sowie der Datenbeginn (bei variabler Headergröße) und weitere Daten übertragen.

10.3.2 User Datagram Protocol

UDP übernimmt die Übertragung von IP in unverändertem Stil und fügt lediglich eine Anwendungsprozeßidentifikation hinzu. Dies ist notwendig, da ja Prozesse innerhalb der Transportschicht miteinander kommunizieren.

Kapitel 11: Anwendungsorientierte Schichten

Die oberen Protokollschichten des ISO/OSI-Basisreferenzmodells werden im allgemeinen als anwendungsorientiert bezeichnet. Sie sollen dem Benutzer Dienste zur Verfügung stellen, die weitgehend unabhängig von den Dienste des Netzes in den Schichten 1 bis 4 sind.

Das weitgehende wird natürlich durch die Nicht-Austauschbarkeit der einzelnen Schnittstellen eingeschränkt: Es ist durchaus ein Unterschied, ob eine Datenübertragung nach ISO/OSI oder nach TCP/IP benutzt wird. Dieser Unterschied macht sich mindestens in der Aufrufschnittstelle der oberen Schichten bemerkbar.

11.1 Die Kommunikationssteuerungsschicht

Die ***Kommunikationssteuerungsschicht*** (*Session layer*) gibt dem Benutzer die Möglichkeit, die Kommunikation mit dem anderen Anwendungsprozeß selbst zu kontrollieren und zu synchronisieren. Die unverfälschte Datenübertragung obliegt dabei den unteren Schichten, jetzt existieren in der Schicht 5 sprachliche Mittel zur Durchführung einer Kommunikation.

Folgende Funktionen innerhalb der Kommunikationssteuerungsschicht können unterschieden werden:

- Auf- und Abbau von Verbindungen (einmal mehr, jetzt auf Kommunikationsbasis)
- Festlegung von Synchronisationspunkten
- Unterbrechung und spätere Fortsetzung des Dialogs
- Ausnahmebehandlung bei Netzfehlern

Zur Synchronisation werden ***Marken*** (*token*) verwendet:

- ***Datenmarken*** (*data token*)
- ***Beendigungsmarken*** (*release token*)
- ***Synchronisiermarken*** (*synchronize-minor token*)
- ***Aktivitätsmarken*** (*activity-major token*)

Die ***Datenmarken*** steuern die Richtung der Kommunikation (insbesondere bei Halbduplexbetrieb). Die ***Beendigungsmarke*** erlaubt es einem Teilnehmer, die Sitzung zu beenden und dies dem Partner mitzuteilen.

Die ***Synchronisationsmarken*** dienen der Synchronisation der Verbindung. In gewissen Abständen kann der Sender diese absetzen, der Empfänger muß quittieren. Bei Netzfehlern kann insbesondere die Kommunikation ab einer quittierten Synchronisationsmarke wieder aufgenommen werden.

Die ***Aktivitätsmarken*** dienen einer größeren Unterteilung einer Verbindung. Hier kann insbesondere eine Aktivität, wie die Unterteilung genannt wird, an einer solchen Marke bewußt unterbrochen und in späteren Sitzungen wieder aufgenommen werden.

Die Sitzungsschicht ist in den verschiedenen Netzen insgesamt eher rudimentär implementiert; die Notwendigkeit zur Einfügung von Marken wird eher gering eingeschätzt.

11.2 Die Darstellungsschicht

Die ***Darstellungsschicht*** (*Presentation layer*) dient der Vereinbarung des Datenformats, um die übertragenen Daten in ihrer Bedeutung in den verschiedenen Rechensystemen anzugleichen. Hiermit wird deutlich, daß nicht nur die eigentlichen Daten im Sinne von Bitmustern, sondern die darinwohnende Information zu übertragen. Diese Aufgabe ist sehr umfassend, denn alleine die Notation für die Interpretation einer Folge von Bytes als eine Darstellung einer Variablen ist bereits komplex (little endian, big endian, Integer, Floating Point usw.).

Die pragmatische Lösung besteht darin, die Daten in einem Alphabet darzustellen, z.B. dem Internationalen Alphabet Nr. 5, und die Interpretation den Rechnern zu überlassen. Dies löst noch keineswegs die Darstellung von Zahlen bei unterschiedlichen Rechnern (Motorola, Intel).

Eine weitergehende Alternative liegt in der Umwandlung aller Daten in der Darstellungsschicht in eine netzweit einheitliche Darstellung. Damit ist auch die Bedeutung von Bitfolgen und damit die innewohnende Information übertragbar.

Die Semantiksprache, die die Bedeutung der Daten beschreibt, wird als ***abstrakte Syntax*** bezeichnet. Dieser Syntax ist zu entnehmen, was dargestellt wird (Zahlenwerte, Begriffe, logische Werte). Die Repräsentation solcher Werte während der Übertragung wird durch die ***konkrete Syntax*** beschrieben, die im Prinzip unabhängig von der abstrakten Syntax gewählt werden kann.

Die Darstellungsschicht leistet somit die folgenden Aufgaben:

- Bereitstellung mindestens einer abstrakten Syntax
- Bereitstellung mindestens einer konkreten Syntax
- Bereitstellung der notwendigen Umwandlungsprozeduren

Die abstrakte Syntax ist in ISO 8824 und CCITT T.409 als ***ASN.1*** (*abstract syntax notation no. 1*) definiert. Die konkrete Syntax wurde in der ISO 8825 als ***Basic Encoding Rules*** definiert.

11.3 Anwendungsdienstelemente

Die Anwendungsschicht setzt sich aus verschiedenen ***spezifischen Anwendungsdienstelementen*** (***SASE***, *Specific Application Service Element*), die ihrerseits auf ***gemeinsame Anwendungsdienstelemente*** (***CASE,*** *Common Application Service Element*) aufbauen.

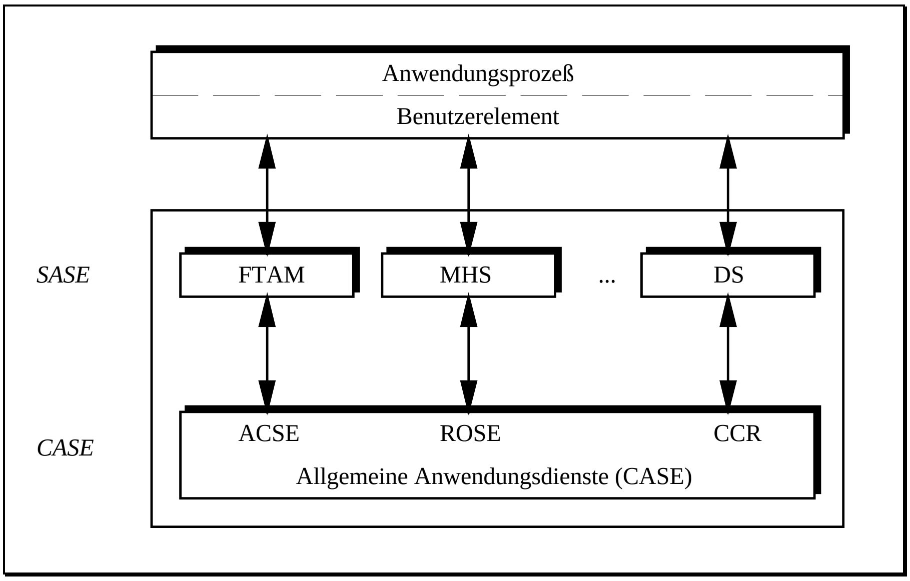

Abb. 11-1: Architektur der Schicht 7

Die wichtigsten Beispiele für CASE sind:

- Der ***Verbindungskontrolldienst*** (***ACSE***, *Association Control Service Element*).
- Die Dienste zur ***entfernten Operationskontrolle*** (***ROSE,*** *Remote Operation Service Element*).
- Die Dienste zur ***zuverlässigen Übertragung*** (***RTSE***, *Reliable Transfer Service Element*).
- Die ***Synchronisation atomarer Aktionen*** (***CCR***, *Commitment, Concurrence and Recovery*)
- Ein Dienst zur ***Transaktionsbearbeitung*** (***TP***, *Transaction Processing*)

11.4 Anwendungsdienste

Die auf den allgemeinen Dienstelementen aufsetzenden spezifischen Anwendungsdienste umfassen u.a. Datei-, Directory- und Nachrichtendienste. Hierzu zählen:

- Message Handling System (X.400)
- Directory Service (X.500)
- File Transfer, Access and Manipulation (FTAM, ISO 8571)
- Virtuelles Terminal (ISO 9040, 9041)
- Manufacturing Automation Protocol (MAP)

Kapitel 12: ISDN und ATM-Netze

12.1 ISDN

Mit ISDN (Integrated Services Digital Network) werden Netze im öffentlichen Bereich bezeichnet, die sich aus der CCITT-Norm I.120 entwickelt haben. Im Vordergrund dieser Entwicklung stand der Wunsch, neue Technologie und neue Services mit der alten Kabeltechnik zu verbinden.

Das ISDN-Netz baut sich weiterhin hierarchisch auf: Es existieren Teilnehmerleitungen, Leitungen zwischen Ortsvermittlungen sowie zwischen Fernvermittlungen (verschiedener Kategorien). Im Unterschied zu den bisher analog genutzten Leitungen, die im Frequenzmultiplex betrieben wurden, nutzt ISDN aufgrund seiner digitalen Struktur hierfür Zeitmultiplexverfahren.

Abb. 12.1 zeigt den prinzipiellen Aufbau und die wesentlichen Übertragungsraten im ISDN-Netz Deutschland.

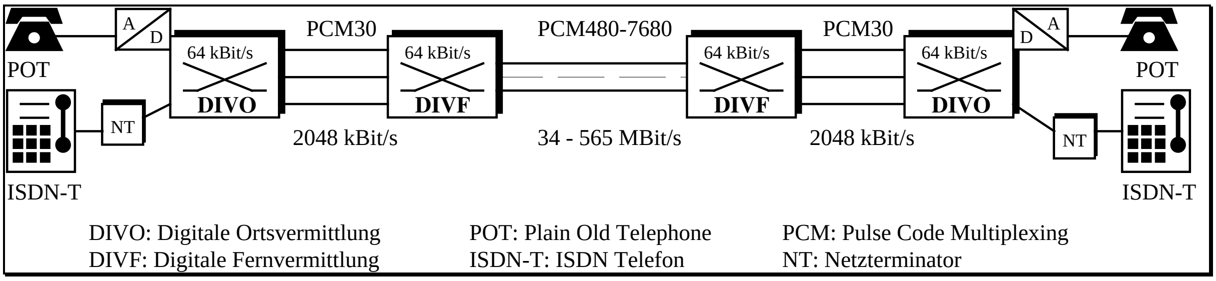

Abb. 12-1: Prinzipieller Aufbau ISDN-Netz (Deutschland)

Für die Teilnehmerleitungen zwischen der Ortsvermittlung und dem NT (Network Termination) werden verschiedene Kanäle geführt:

A: 4 kHz analoger Telefonkanal
B: 64 kBit/s: Digitaler Kanal für PCM-codierte Sprache oder digitale Daten
C: 8 oder 16 kBit/s: Digitaler Kanal
D: 16 oder 64 kBit/s: Digitaler Kanal für Außerbandsignalisierung
E: 64 kBit/s: Digitaler Kanal für interne ISDN-Signale
H: 384, 1536 oder 1920 kBit/s: Digitaler Kanal

Zulässige Kombination:
1. Basisanschluß: 2 B + 1 D_{16}
2. Primärratenanschluß: 30 B + 1 D_{64} (in USA und Japan: 23 B + 1 D_{64})
3. Hybridanschluß: 1 A + 1 C

Tab. 12-1: Teilnehmerleitungen

Ein B-Kanal kann demnach einen Sprachkanal mit einer Digitalisierungsrate von 8 kHz bei 8 Bit Digitalisierungstiefe übertragen; dies ergibt rechnerisch 64000 Bit/s (nicht 65536 Bit/s). Die Kopplung mehrerer

Kanäle zu einem einzigen mit höherer Übertragungsrate ist jedoch innerhalb ISDN nicht vorgesehen, gleichwohl möglich. Diese Einschränkung bedeutet in der Praxis, daß bei einer beabsichtigten höheren Übertragungsrate die Zerlegung auf die Basiskanäle (und die finale Zusammensetzung) der Applikation obliegt!

Zum Anschluß einer ISDN-Schnittstelle werden diverse Referenzpunkte definiert:

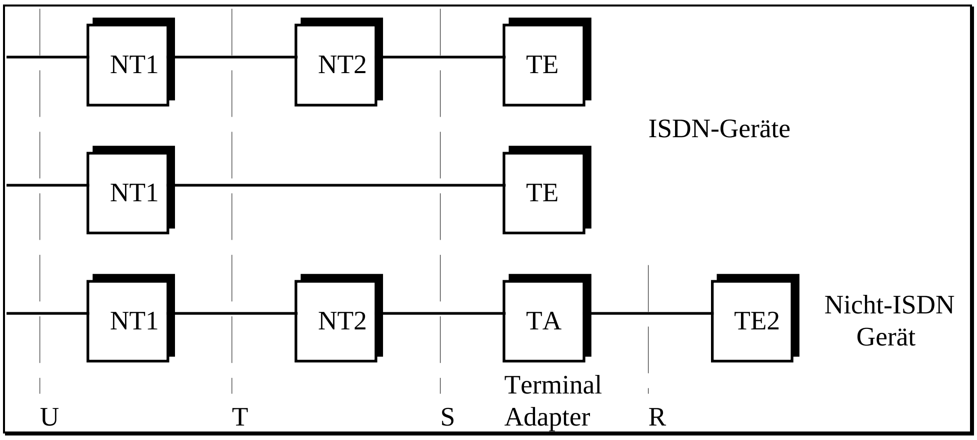

Abb. 12-2: Referenzpunkte im ISDN-Teilnehmerbereich

Der Nutzer erhält dabei Zugang zum S- oder T-Referenzpunkt, welche logisch und physikalisch übereinstimmen und sich nur durch die NT-Kapazitäten unterscheiden. NT2 ist in der Lage, ISDN-Geräte lokal untereinander zu verbinden, NT1 nicht. Der S-Referenzpunkt für den Basisanschluß wird S_0 genannt, während er für einen Primärratenanschluß S_{2M} heißt.

Logisch wird am Basisanschluß (2 B + 1 D_{16}) eine Nettobitrate von 144 kBit/s benötigt. Physikalisch wird dieser Anschluß mit 192 kBit/s betrieben, da noch Rahmeninformationen sowie Taktrückgewinnung mit übertragen werden. Die Codierung der S_0-Schnittstelle entspricht der 4B3T-Codierung (AMI, G.703), wobei hierdurch wiederum eine Taktreduzierung auf 144 kHz erreicht wird. Die Übertragung erfolgt auf 4 Leitungen, während bei der Übersetzung in Richtung analoger Technologie häufig eine Zweidrahtverbindung mit Echounterdrückung zum Einsatz kommt.

Als Zugangsprotokoll zum S_0-Bus wird HDLC-LAPD, eine LAPB-Variante, verwendet. Hierdurch ist es möglich, Zugriffskonflikte durch einfaches Überschreiben von gesendeten ‘1’ (im Leerlauf) zu lösen. Geräte, die ihr Echo am D-Kanal nicht mehr erkennen, müssen sich vom Zugriff zurückziehen, so daß eines von 8 Geräten (mit der höchsten Priorität) zurückbleibt.

Die S_{2M}-Schnittstelle verwendet ebenfalls, wie bereits in Kapitel 5 erwähnt, G.703. Hier werden 30 B-Kanäle sowie ein D_{64}-Kanal mit Zusatzinformationen zu 2048 kBit/s zusammengeschaltet, wobei allerdings nur der D-Kanal aus den bisherigen zusammengefaßt wurde, während die übrigen Kanäle ohne gegenseitigen Zusammenhang übertragen werden.

Die Wählinformationen zwischen den Vermittlungen werden im übrigen im Signalisierungssystem Nr. 7 (SS #7) übertragen, unabhängig von den Datenübertragungen. Dieses System hat sich sehr bewährt und ist auch für andere Signalisierungen und Informationsübertragungen geeignet, zu denen z.B. der Service der 0130-Nummern in Deutschland zählt. Hierbei wird die korrekte Rufnummer bei einem Datenbankserver abgeholt, so daß die Vermittlungsanlage ohne Umwege wählen kann, während der Nutzer nur die ihm ortunbekannte Rufnummer 0130-... gewählt hat.

12.2 Breitband-ISDN

Die Definition von B-ISDN (Broadband-ISDN) ist in der CCITT-Empfehlung I.113 festgelegt: Alle Systeme, die eine Übertragungsrate größer als die Primärrate von ISDN (2 MBit/s) gestatten, werden hierin als B-ISDN bezeichnet. Die Übertragungsrate ist hierbei nicht als einfache Bündelung vieler kleinerer Basiskanäle zu verstehen, sondern als echte Zusammenfassung für eine Übertragung mit voller Datenrate zu verstehen.

Abweichend hiervon sind in der H-Hierarchie einige Kanäle definiert, die ebenfalls als Breitbandkommunikation bezeichnet werden. Diese H-Hierarchie hat einen vorläufigen Charakter:

B:	64 kBit/s
H0:	384 kBit/s
H11:	1536 kBit/s
H12:	1920 kBit/s
H2:	32 .. 34 MBit/s
H3:	45 .. 70 MBit/s
H4:	135 .. 139 MBit/s

Tab. 12-2: H-Hierarchie

Breitband-ISDN beschreibt demnach die technische Ausprägung eines Netzes, keinerlei Protokolle usw. Hierfür wird in CCITT-Empfehlung I.121 ATM (Asynchronous Transfer Mode), das im nächsten Abschnitt besprochen wird.

12.3 ATM

Angesichts einer umfassenden Strukturierung des Bedarfs an Kommunikation, beginnend mit einfacher Datenkommunikation über Multimediaanwendungen bis hin zur Bewegtbildübertragung sollte zunächst eine Analyse der Forderungen an ein modernes Kommunikationssystem erfolgen:

- Eine Vielzahl von Verbindungen soll über einen Kommunikationskanal (Leitung) erfolgen. Hieraus ergibt sich die Forderung nach einem paketorientierten Dienst, der im Adreßmultiplex die unterschiedlichen Ziele erreicht.

- Einige Verbindungen benötigen feste Übertragungsraten (z.B. Bewegtbilder, Sprache), andere hohe Burstraten bei sonst moderaten Übertragungsraten. Dieses unterschiedliche Spektrum kann durch ein Zeitmultiplexverfahren mit festen Blockgrößen bedient werden, bei dem unbenutzte Zeitschlitze anderen Anwendungen zugeteilt werden können.

ATM (Asynchronous Transfer Mode) soll nun diese Forderungen zusammen mit Fehlerbehebungen und -vermeidungen erfüllen.

12.3.1 ATM Netzarchitektur

ATM setzt auf bestehenden wie neuen Netzen auf. Es wird als das Verfahren bezeichnet, um auf Basis Breitband-ISDN die Kapazität den Anwendungen bereitzustellen. ATM wurde zunächst für öffentliche Netze definiert, zeigt jedoch Tendenzen, sowohl im LAN-Bereich wie im Desktop-Bereich (Rechnerkonfiguration am Arbeitsplatz) einzuziehen.

Im WAN-Bereich zeichnen sich bezüglich der Netzarchitektur zwei Tendenzen ab: ATM kann einerseits die vorhandenen Schmalbandnetze nutzen. Dies bedeutet jedoch, daß bei Ausnutzung höherer Bitraten per Multiplexen eine erhebliche Komplexität erzeugt wird.

Der zweite Ansatz zur eigenständigen ATM-Netzarchitektur beruht auf B-ISDN und geht davon aus, daß angepaßt an das Verkehrsaufkommen per hochzuverlässigen ATM-Switches semipermanent geschaltete Pfade mit hoher Transferrate zustandekommen. Diese Switches stellen keine Vermittlung dar, die die ankommenden Pakete verteilt, sie werden vielmehr durch ein Managementsystem geschaltet.

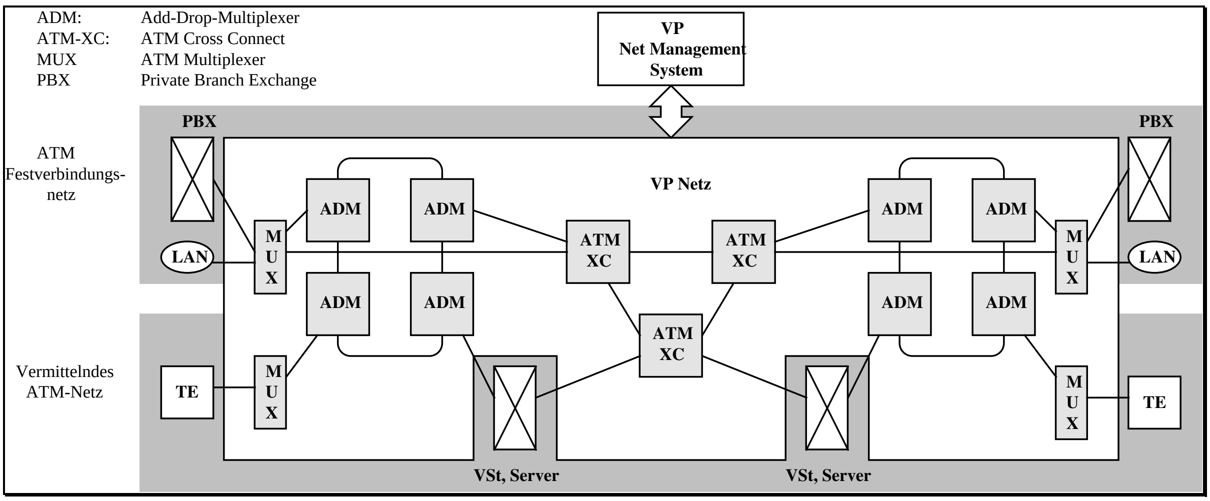

Abb. 12-3: ATM-Netzarchitektur

12.3.2 Das ATM-Datenformat

Abb. 12-4 zeigt die ATM-Protokollstruktur, die deutlich abweichend von dem üblichen ISO/OSI-Protokollstack nur 3 Schichten aufweist. Diese Abweichung hat spezielle Gründe: ISO/OSI betrachtet grundsätzlich nur Datenübertragungen, die Problematik zeitsynchroner Übertragungen wie Sprache oder Bewegtbilder tritt hier nicht näher auf. Weiterhin war es ein wesentliches Ziel von ATM, effiziente Übertragungen zu unterstützen; hierzu dient u.a. ein schlanker Protokollstack.

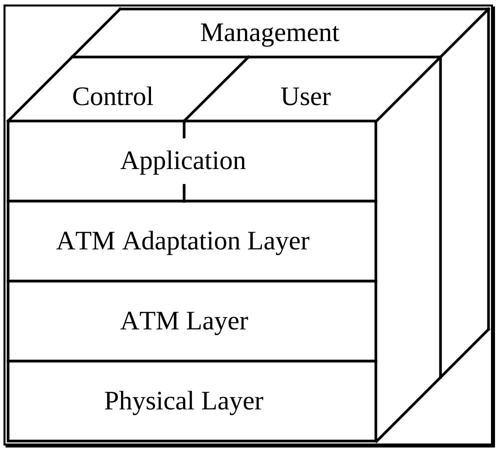

Abb. 12-4: ATM-Protokollstack

Schicht 1 ist dabei nicht näher spezifiziert, das ATM-Verfahren erlaubt die Anpassung an diverse Übertragungsverfahren. Die eigentlichen ATM-Schicht besteht in der Definition der ATM-Blöcke, hier Zellen (cell) genannt. Sie ist in Abb. 12-5 dargestellt und umfaßt zwei Typen, User-Network-Interface und Node-Network Interface, letzteres zur Steuerung der Übertragung:

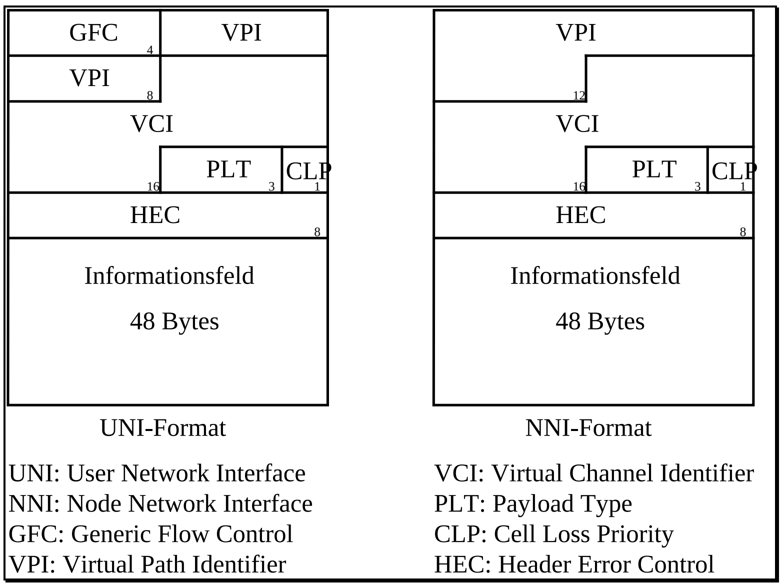

Abb. 12-5: ATM-Zellformate

Die in dem Zellenformat enthaltenen Steuerinformationen haben folgende Funktionalitäten:

- Identifikation von virtuellen Kanälen und Pfaden (VCI, VPI): Diese Informationen gestatten eine zweistufige Adressierung innerhalb des Netzes. Gruppen von virtuellen Kanälen mit gleichem virtuellen Pfad können hierdurch identisch vermittelt werden, da die Kanalnummer erst am Rechner unterschieden wird. Dieses zweistufige Verfahren ist bemerkenswert, da die virtuellen Pfade jeweils mehrfach während einer Übertragung umgesetzt werden müssen.
- Zugriffssteuerung bei Anschluß mehrerer Endsysteme an einer Teilnehmerschnittstelle (Generic Flow Control, GFC)
- Kennzeichnung des Nutzlasttyps (Payload Type. PLT)
- Fehlersicherung der Steuerinformation (Header Error Control, HEC). Diese Information wird erst durch Schicht 1 hinzugefügt.

12.3.3 ATM-Protokolle

Die dritte Schicht im ATM-Stack, die Anpassungsschicht (AAL: ATM Adaptation Layer), besitzt nunmehr mehrere Aufgaben. Zunächst muß während der Verbindungsaufbauphase, die mit Ausnahmen durchlaufen wird, eine Dienstklasse aus folgenden Möglichkeiten vereinbart werden:

Klasse	Class A	Class B	Class C	Class D
Zeitabhängigkeit zwischen Quelle und Ziel	notwendig		nicht notwendig	
Bitrate	konstant	variabel		
Verbindungsart	verbindungsorientiert			verbindungslos
Beispiel	virtuelle Datenverbindung Circuit Emulation	Variables Bewegtbild	verbindungsorientierte Datenübertragung	verbindungslose Datenübertragung
AAL-Typ	Type 1	Type 2	Type 3/5	Type 4

Abb. 12-6: ATM-Dienstklassen

Weiterhin sollte nicht unerwähnt bleiben, daß die ATM Schicht 2 keine Sicherung zur Verfügung stellt; dies muß daher in die AAL eingezogen werden, falls es nicht sogar auf die Applikation verlagert wird.

Folgende Dienste sind für ATM besonders geeignet:

- Frame Relay (FR) ist ein paketorientiertes Übertragungsverfahren ähnlich zu X.25. Die Sicherung erfolgt in dieser Applikation.
- Switched Multimegabit Data Service (SMDS) und Connectionless Broadband Data Service (CBDS) für verbindungslose breitbandige Datendienste z.B. zur Kopplung von LANs. Für SMDS wurde das SMDS

Interface Protocol (SIP) entworfen, das in Layer 3 die Vermittlungsfunktion enthält, in Layer 2 eine an die ATM-Zellen möglichst angepaßte Struktur.

Für diese paketorientierten Datenanwendungen sind 3 AAL-Protokolle definiert worden, die mit den Typenbezeichnungen AAL 3 .. 5 versehen werden (siehe Abb. 12-6).

Kapitel 13: Netzwerkmanagement

Die bisherigen Betrachtungen waren mit den Fragen des Aufbaus und des Übermitteln von Daten gekoppelt. Der Betrieb des Netzes benötigt jedoch eine Steuerung, da diverse Größen parametrisierbar sind und natürlich Ausfälle behandelt werden müssen.

Während der Fehlerfall ein gut lokalisierbare Ausnahmesituation darstellt, die gezielt behandelt werden kann, ist das Netzwerkmanagement auf den Betriebsfall bezogen.

13.1 Ziele des Netzwerkmanagements

Die Einschätzung von Rechnernetzen kann aus durchaus sehr unterschiedlichen Sichtweisen erfolgen: Während der Benutzer an einer hohen Dienstgüte, hohen Verfügbarkeit und geringen Kosten für seine Inanspruchnahme interessiert ist, erwartet der Betreiber eines Rechnernetzes einen möglichst hohen Gewinn durch die optimale Nutzung der bereitgestellten Ressourcen.

Der Ansatz zur befriedigenden Lösung dieser divergierenden Wünsche und Sichtweisen liegt in einem intelligenten Netzwerkmanagement: Netzwerkadministratoren (Personen) sollen über technische Hilfsmittel in die Lage versetzt werden, das Netzwerk entsprechend den momentanen Anforderungen zu betreiben. Die technischen Systeme werden hierbei als Netzwerkmanagementsysteme bezeichnet; sie müssen in der Regel über das Netz verteilt sein, da die Parametrisierung im gesamten Netz vonstatten gehen muß.

13.2 Aufgaben des Netzwerkadministrators

Die Aufgaben des Netzwerkadministrators umfassen diverse Teilaufgaben, zu deren Lösung er das/die Netzwerkmanagementsystem(e) einsetzt. Hierzu zählen:

- *Configuration Management*: Die grundlegende Tätigkeit zur Parametrisierung von Rechnernetzkomponenten und deren Verbindungen untereinander wird Netzwerkkonfiguration genannt. Das Konfigurationsmanagement beinhaltet somit alle Aufgaben zur Erstellung der gewünschten Funktionalität (Übergabe der Parameter), zum Sammeln der Informationen über den Rechnernetzzustand sowie dessen Dokumentation. Hierin liegen die grundlegenden Informationen für alle weiteren Aktivitäten.

- *Fault Management*: Das Erkennen von Störungen, die Diagnose der Ursachen und die Behebung des Fehlers sind grundlegende Aufgaben des Fehlermanagements. Die Medlung der Störung erfolgt teilweise durch die Komponenten selbst, durch Benutzer, andere Komponenten usw. Idealerweise wird versucht, neben der Protokollierung zugleich Behebungs- oder Ähnlichkeitshinweise mitzugeben.

- *Performance Management*: Wie andere technische Einrichtungen werden Netze bemessen, und zwar in qualitativer wie quantitativer Hinsicht. Die zentrale Aufgabe des Leistungsmanagements besteht in der kontinuierlichen Überwachung der Leistungskenngrößen eines Rechnernetzes, z.B. Belegtsignale, Nettoperformance etc., um ggf. bei Freischalten anderer Ausweichleitungen eine optimale Dienstgüte bei moderaten Kosten zu gewährleisten.

  Für diese Aufgabe müssen allerdings einige Voraussetzungen geschaffen werden:
  - geeignete Kenngrößen müssen definiert sein
  - die Messung dieser Kenngrößen muß gewährleistet sein
  - die Daten müssen gesammelt und (online) ausgewertet werden

  Auf diese Weise können Konzepte für langfristige und Ad-Hoc-Methoden für kurzfristige Lösungen gewonnen werden. So sind verschiedene Probleme wie
  - schlechtes Leistungsverhalten (Durchsatz, Antwortzeit)
  - kurzfristige oder ständige Überlastsituation
  - schlecht ausgelastete Ressourcen
  - zu häufiges Auftreten von Fehlverhalten

  erkennbar.

- *Accounting Management*: Die Nutzung von Rechnersystemen ist grundsätzlich mit Kosten verbunden, die seitens des Betreibers auf die Nutzer umgelegt werden sollen. Dieses Verfahren wird häufig pauschal angewendet, in neuerer Zeit jedoch mit Vorliebe einzelfallabgerechnet. Um dies anwenden zu können, muß die gesamte in Anspruch genommene Leistung einschließlich aller Arten protokolliert werden: Eine Aufgabe des Abrechnungsmanagement. Die abrechnende Stelle benötigt dazu ggf. einen genauen Einblick in diverse Kommunikationsschichten.

- *Security Management*: Alle Aktionen, die den unberechtigten Zugriff in einem Netz verhindern sollen, werden unter dem Begriff Sicherheitsmanagement zusammengefaßt. Das Spektrum der Aufgaben umfaßt z.B. die Vergabe von Paßwörtern, die sichere Aufbewahrung von Daten, die Datenverschlüsselung usw.

- *Systemmanagement*: In einem Rechnernetz werden – auch an einem einzelnen Ort – verschiedene sich gegenseitig beeinflussende Komponenten eingesetzt: Die Trennung in Hardware, Software und Kommunikation im Netz ist letztendlich zu grob, da in einem Knoten Betriebsysteme, Hardwarekomponenten z.B. zur Kommunikation, die häufig einem zusätzlichen Rechner überlassen wird, sowie Anwendungen und Daten auf den einzelnen Komponenten existieren. Da diese Komponenten zum Gesamtablauf auch der Kommunikation benötigt werden, ist es sinnvoll (obgleich in der Gesamtrealisierung schwierig), das Management dem Netzadministrator zu übertragen.

Diese Aussage gilt mehr für lokale Netze, die z.B. in einem Unternehmen gekoppelt sind; im Internet ist ein komplettes Systemmanagement weder technisch noch rechtlich denkbar. Die Aufgaben des Systemadministrator umfassen dann:

- Datensicherung (Duplizierung)
- Zeitmanagement (Synchronisierung von internen Uhren)
- Organisation verteilter Datensysteme
- Archivierung von Daten
- gemeinsame Nutzung von Ressourcen wie Druckern
- Benutzerverwaltung
- Software-Versionskontrolle

13.3 Managementansätze

Die prinzipielle Aufgabe des Netzwerkmanagements liegt beim Netzwerkadministrator. Die konkrete Gestaltung dieser Aufgabe kann jedoch durch technische Systeme, die ***Managementsysteme*** entscheidend verbessert werden. Diesen Systemen wird die Aufgabe zum Sammeln und Auswerten von Administrations-relevanten Daten übertragen:

- Bereitstellung eines zentralen Zugangs zu den managementrelevanten Informationen: Sammeln an zentraler Stelle.
- Filterung unwichtiger Daten: Hierzu muß sicher entschieden werden können, welche Daten keine relevanten Informationen enthalten, ein nicht-triviales Problem.
- Auswertung von Daten: Entdeckung der Abhängigkeiten von Daten, die an verschiedenen Zeitpunkten oder Orten gesammelt wurden, Benennung von Folgefehlern, Darstellung der Informationen zur schnellen Reaktion beispielsweise bei Leitungsengpässen.
- Speicherung von Daten für spätere Auswertung, z.B. für weitergehende Langzeitanalyse, Anbieten neuer Dienste etc.
- Speicherung von Fehlerbehebungsstrategien bei Entdeckung ähnlicher Fehler zu späteren Zeitpunkten.

Der grundsätzliche Managementansatz bedient sich dabei Managementkomponenten in den einzelnen Rechnern, *Agenten* genannt. Diese Agenten sammeln die notwendigen Informationen, meist eingelagert in die einzelnen Kommunikationsschichten, und senden diese zum Managementsystem. Hierzu sind eine Reihe von Vereinbarungen notwendig, insbesondere in Anbetracht inhomogener Netzwerke:

- Standardisierung der Semantik und Syntax managementrelevanter Informationen
- Festlegung der möglichen Bestandteile des Managementsystems und Rollenverteilung der einzelnen Komponenten
- Definition der grundlegenden Basismechanismen für die Abläufe im Managementsystem

- Spezifikation von Protokollen zum Austausch der managementrelevanten Daten
- Beschreibung der Zugriffsschutzstrategien für die Managementdaten
- Gruppierung und Darstellung der vom Managementsystem erbrachten Funktionen
- Spezfikation der Schnittstellen zwischen dem Managementsystem und darauf aufbauenden Anwendungsprogrammen.

ISO/OSI beschreiben einen kompletten Managementansatz (Common Management Information Service resp. Protocol, CMIS/CMIP, ISO 9595 und 9596), dessen Agenten den einzelnen Layern zugeordnet ist und über das Managed Object (MO) entsprechend Auskunft gibt. Dieser Ansatz ist sehr mächtig; auf dem Protokoll werden weiterhin verschiedene System Management Functions (SMF) definiert, die die eigentliche Schnittstelle zum Managementrechner darstellen. Das ISO/OSI-Managementsystem ist in Telekommunikationsnetzen weit verbreitet.

Internet hat via dem Internet Activities Board (IAB) ein Managementsystem standardisiert, der als ***Simple Network Management Protocol*** (SNMP, aktuell Version 2) bezeichnet wird. Innerhalb des globalen Registrierungsbaums wird ein Teilbaum für das Netzwerkmanagement bereitgestellt, so daß für das Managementsystem die notwendigen Adreßinformationen via Datenbank abfragbar sind.

Die für das Management benötigten Komponenten des Rechnernetzes werden auf eine Menge von ***Managementobjekten*** abgebildet, die ihrerseits mittels ***Objekttypen*** modelliert werden. Die Gesamtmenge aller Definitionen dieser Objekttypen wird ***Management Information Base*** genannt.

Ein Managementobjekt im Internet ist weiterhin entweder eine einfache Variable (auch String, jedoch keine Struktur) oder eine geordnete Menge von Variablen, eine Tabelle. Für diese Variable wird Typ, Zugriffsrechte und Status (gefordert, optional etc.) als ***Structure of Management Information*** (SMI) angegeben und ihr ein Object Identifier zugeordnet.

Einer Instanz dieser Variablen muß zusätzlich ein im gesamten Internet eindeutiger Name zugewiesen werden. Dies erfolgt durch Anhängen von .o an den Object Identifier und durch die Tatsache, daß durch die Adressierung im Internet implizit der Agent ausgewählt wird und sich in dessen Bereich maximal eine Instanz befinden darf.

Die Kommunikation zwischen dem Managementsystem und dem Agent erfolgt dann in asynchroner Weise durch ein Anfrage/Antwort-Protokoll. Die Anfragen werden durch Polling oder auch unterbrechungsgesteuert gestartet.

13.4 Betriebsmittelverwaltung in ATM

ATM lebt in erhöhtem Maße mit Managementsystemen. Diese Aussage wird sofort einleuchtend, wenn die Dienstklassen näher betrachtet werden, denn hier werden auch zeitliche Beschränkungen bei Übertragungsdiensten definiert. Abb. 13-1 zeigt eine Übersicht über Betriebsmittelverwaltungsverfahren im ATM:

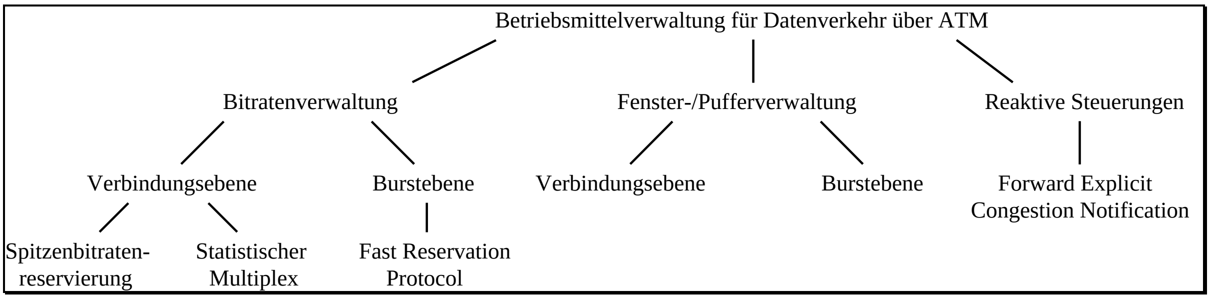

Abb. 13-1: Betriebsmittelverwaltungsverfahren in ATM

Bei dieser Grafik fällt zunächst auf, daß zwei wesentliche Größen verwaltet werden: ***Bitraten*** und ***Fenster/Puffer***. Auf Verbindungsebene können Spitzenbitraten vereinbart werden. Dies bedeutet, daß die maximale Bitrate reserviert (und bezahlt) wird, wobei häufig die mittlere Bitrate bei < 1% der Spitzenbitrate liegt. Die Folge: Große Ineffizienz. Ein anderer Ansatz liegt in einem statistischen Multiplex: Geringere Spitzenrate und kleinere Varianz der Bitraten sind gute Voraussetzungen für Quellen, eine Varianz (σ) anzugeben und entsprechend Bitrate mit einer Verlustwahrscheinlichkeit für Zellen zu erhalten. Modellrechnungen ergeben z.B., daß bei 150 Mbit/s 26 Quellen möglich sind, die eine Spitzenbitrate von 10 MBit/s haben und eine Burstiness (Verhältnis Spitzen- zu mittlerer Bitrate) von 10, wenn die Verlustwahrscheinlichkeit für Zellen bei 10^{-10} liegen soll. Die mittlere Netzauslastung liegt dann bei 17%.

Bessere Ergebnisse sind mit einem Protokoll möglich, daß die Bitratenreservierung nicht auf Verbindungs-, sondern auf Burstebene regelt. Ein solches Protokoll, das Fast Reservation Protocol (FRP), schickt dann eine Reservierungsanforderung an alle Knoten, und bei positiver Rückmeldung ist die Rate reserviert. Hieraus resultiert allerdings eine erheblich höhere Komplexität der Realisierung in den Knoten.

Gleiche Aussageb gelten für die Zwischenspeicherkapazitäten in den Knoten: Die hier verfügbaren Speicherbereiche sind durchaus entscheidend für die Übertragungsgeschwindigkeit.

In kleinen ATM-Netzen etwa auf LAN-Basis kann auch ein dynamisches Verfahren im Netz selbst zum Einsatz kommen. Ein solches Protokoll, FECN (Forward Explicit Congestion Notification), soll bei Überlastung die Quellen steuern, allerdings in Richtung kleinerer Datenraten. Es könnte erweitert werden in Richtung eines Verkehrsmanagements bei Prioritätansetzung!

Kapitel 14: Bewertung von Rechnernetzen

Kapitel 15: Routing-Algorithmen

15.1 Einleitung

Routing-Algorithmen sind ein Teil der Schicht 3, der Vermittlungsprotokolle. Innerhalb dieser Schicht wird bei Netzen, die nicht aus einem Stück aufgebaut sind, sondern wo im allgemeinen die Übertragung in Teilnetzen durchgeführt wird, die zuverlässige Übertragung garantiert. Zu dieser Zuverlässigkeit zählt natürlich auch der 'richtige' Weg.

Die Wahl des Wegs bzw. das Verfahren der Wegebestimmung hängt allerdings stark von der Art der Verbindungen ab: In leitungsvermittelten Netzen werden ausschließlich verbindungsorientierte Dienste verwendet. Hierbei muß zu Beginn des Verbindungsaufbaus die optimale Route festgelegt werden, denn die Zahl der Transitstationen zur Übermittlung wird den Aufwand zur Übertragung kennzeichnen. Gleiches gilt für die verbindungsorientierten Paketdienste wie X.25 (mit Ausnahmen): Der Aufbau der Verbindung erzeugt virtuelle Verbindungen mit logischen Kanälen, so daß die Pakete durchaus immer die gleiche Route wählen.

Ein deutlicher Unterschied hierzu tritt bei den verbindungslosen Datagrammdiensten wie im Internet auf. Hier wird für jedes Datagramm in jedem Vermittlungsknoten eine eigene Routingentscheidung getroffen. Aus diesem Grund muß die Strategie in den jeweiligen Verbindungsarten differerieren.

15.2 Begriffe zum Routing

15.2.1 Das Netzmodell

Das Netz wird für Routingzwecke in einem Graph abgebildet, in dem die einzelnen Rechner (sowohl Endsysteme wie Vermittlungssysteme) durch Knoten und die Verbindungen zwischen diesen als Kanten dargestellt sind. Die Einführung einer Metrik erfolgt durch Benennung der Kanten; diese Metrik kann sowohl auf Geschwindigkeit, Kosten oder andere Größen bezogen sein. Abb. 15-1 zeigt ein Beispiel hierzu.

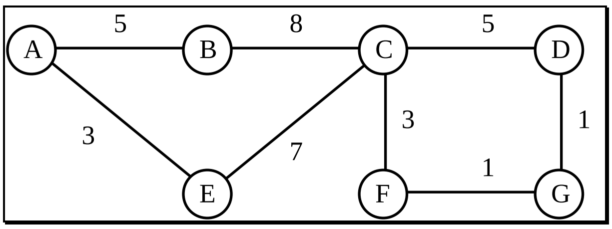

Abb. 15-1: Rechnernetz als Graph

Ein Knotenrechner in diesem Modell besteht aus mindestens einem Prozessor sowie zugehörigen I/O-Kapazitäten. Letztere können aus Warteschlangen bzw. Puffern in RAM-Form, speziellen I/O-Prozessoren oder spezialisierter Hardware bestehen. In modernen Switches, etwa in ATM-Netzen, muß die Vermittlungsarbeit so schnell durchgeführt werden, daß hier intelligente Crossbar-Hardware zum Einsatz kommt.

15.2.2 Bewertungskriterien für Routing-Algorithmen

Die Bewertungskriterien für Routing-Algorithmen können in einem allgemeinen Teil und einem speziellen Teil zusammengefaßt werden:

Im allgemeinen sollte der Routing-Algorithmus folgende Anforderungen unterstützen:

- **Stabilität**: Bei gleichen Verhältnissen immer gleiche Ergebnisse
- **Robustheit**: Bei Änderung des Netzes (Ausfall, Hinzufügung) sollte das Netz weiterhin (im Rahmen) übertragen können.
- **Einfachheit**: Der Routing-Algorithmus sollte einfach zu verstehen, zu implementieren und zu verifizieren sein
- **Korrektheit**: Die Pakete sollen immer beim Empfänger eintreffen
- **Optimalität**: Bezüglich der gewünschten speziellen Bewertungskriterien (s.u.) soll das Netz die bestmögliche Dienstqualität liefern.
- **Fairness**: Alle Teilnehmer sollen möglichst gleich gut bedient werden (bei gleichen Gebühren)

Die speziellen Bewertungskriterien beziehen sich auf die technischen Gegebenheiten und Randbedingungen:

- Kosten, Auslastung, Ausnutzung, Sicherheit, Verzögerung
- Netztopologie, Übertragungsgeschwindigkeit, mittlere Belastung, geforderte Dienstqualität

15.3 Klassifikation der Wegeermittlungsverfahren

Bei dem Verfahren, eine Wegewahl zwischen zwei Netzteilnehmern zu berechnen, unterscheidet man

- **nicht adaptive** (statische) Verfahren und
- **adaptive** (dynamische) Verfahren.

Die dynamischen Verfahren werden weiterhin unterteilt, und zwar nach der Instanz, die die Adaption vornimmt sowie die Lokalität des Einflusses:

- Die **lokalen adaptiven Verfahren** (isolated dynamic routing) laufen lokal auf den Vermittlungsrechner und sind unabhängig von dem Rest des Netzes. Die Mechanismen stützen sich dabei auf die lokalen Informationen wie Warteschlangen.

- Die **zentralisierten Verfahren** (centralized dynamic routing) tragen zunächst alle Daten aus dem Netz zusammen und entscheiden dann an ausgezeichneter Stelle (Routingkontrollzentrum), welcher Weg gewählt werden soll. Diese Verfahren lassen sich für paketvermittelte verbindungsorientierte Dienste anwenden, da sie einen großen Überblick haben, die Vermittlung jedoch aufgrund der Anfrage beim Zentralrechner Zeit kostet.
- Die **verteilten Verfahren** (distributed dynamic routing) tauschen in regelmäßigen Abständen Informationen zwischen den Vermittlungsrechnern aus (Internet: IMP, Internet Message Processors) und berechnen daraus lokal die optimal Route. Der Vorteil dieses Verfahrens besteht vor allem darin, daß auch große Netze derart berechenbar bleiben.

15.4 Nicht adaptive Wegeermittlung

Folgende nicht-adaptive Routingverfahren befinden sich im allgemeinen im Einsatz:

- **Fluten**: Dieses zweifelsohne extreme Verfahren ist eigentlich kein Routingalgorithmus, sondern es besteht im Aussenden eines empfangenen Pakets an alle ausgehenden Verbindungen. Damit entsteht im Netz eine Menge von Paketen, die durch in den Folgestationen dekrementierte Paketzähler vernichtet wird. Der Sender muß also wissen, wieviele Teilnehmer in einem Vermittlungszweig liegen; er richtet den Paketzähler entsprechend ein.

  Modifikationen von diesem Verfahren werden in LAN-Switches eingesetzt, um das Netz zu erlernen. Hierbei werden Pakete in alle Richtungen geschickt, es ist jedoch wichtig, daß der Switch eine positive Rückmeldung bekommt, um Routingtabellen führen zu können. Das Verfahren arbeitet also nur anfänglich als Fluten, es ist adaptiv.

- **Feste Wegetabellen**: Falls in einem festgefügten Netz die Wegeberechnung vor Inbetriebnahme durchgeführt werden kann, bietet sich das Verfahren der festen Wegetabelle an. Hierbei werden alle Wege eingetragen, etwa anhand der Metrik, das Routing erfolgt ausnahmslos anhand der Tabelle.

  Die Festlegung des kürzesten Wegs erfolgt mit ‘Shortest Path Aorithm’, z.B. nach Dijkstra.

15.5 Adaptive Wegeermittlung

Als bekannte Wegeermittlungsverfahren, die adaptiv arbeiten, können genannt werden:

- **Hot-Potatoe-Algorithm** (isoliert dynamisch): Der Knoten versucht, die Nachricht so schnell wie möglich wieder auszusenden. Somit erfolgt die Zuteilung an die Ausgangsleitung mit der kürzesten Warteschlange.

Diese Verfahren kann dahingehend modifiziert werden, daß den Ausgangsleitungen noch eine Richtungskomponente mitgegeben wird. Das Hot-Potatoe-Verfahren ist nicht besonders effizient, es kann zu Netzblockaden sowie zur Nicht-Erreichbarkeit von Empfängern führen.

- **Backward Learning** (lokal isoliert, Haran): Die Pakete tragen Informationen über den bisherigen Weg in Form von Knotenzählern mit sich. Hieraus kann der Knotenrechner letztendlich Kenntnis über das gesamte Netz erhalten und somit optimale Routen finden.

  Das größte Problem ist das Erkennen von Ausfällen: Diese können so nicht erlernt werden, wohl aber Verbesserungen. Eine Lösung dieses Problems könnte in dem ‘Vergessen’ des bisherigen Wissens und dem Neulernen liegen. Diese Lösung bedeutet jedoch, daß das Netz von Zeit zu Zeit sehr ineffizient wird.

- **Routing Control Center** (zentral): Die Informationen werden an den RCC geschickt, der den optimalen Weg ständig nach einem Shortest Path Algorithm berechnet. Das Problem dieses Verfahrens liegt in dem hohen Nachrichtenaufkommen zum RCC hin.

- **Delta-Leitwegsbestimmung** (gemischt lokal/zentral): Der RCC berechnet nicht einen, sondern einige als optimal bezeichnete Wege; die Informationen hierzu erhält er weiterhin in regelmäßigen Abständen von den Vermittlungsrechnern. Zwei Wege sind dabei gleichermaßen optimal, falls ihre Metrikdifferenz ein festzulegendes δ nicht überschreitet. Die Anzahl der Wege n und das δ sind parametriesierbar, so daß der Netzbetreiber sein Netz zwischen total zentral und gemischt lokal/zentral steuern kann.

  Dem Vermittlungsknoten werden nunmehr alle Wege übermittelt, er kann seinen optimalen Weg eigenständig auswählen. Das Update erfolgt in Abständen, so daß der Nachrichtenverkehr zum RCC minimiert wird.

- **Austausch von Übertragungszeitvektoren** (verteilt): Die Übertragungszeiten werden nicht geschätzt, sondern durch spezielle Nachrichten gemessen. Dieses Verfahren bietet also dem lokalen Vermittlungsrechner die Informationen, die er zum dynamischen, aber lokal basierten Routing benötigt.

  Nachteil dieses Verfahrens ist der relativ große, zusätzliche Nachrichtenverkehr. Weiterhin kann sehr gut auf Verbesserungen, relativ langsam jedoch auf Ausfälle reagiert werden.

Kapitel 16: LAN, MAN und WAN im Überblick

Stichwortverzeichnis:

A

Abstract Syntax Notation *Siehe* ASN.1
Aktor-Sensor-Bus 71
Aktuator-Sensor-Interface *Siehe* ASI
Anwendungsprotokoll 7
Anwendungsschicht 121
Application Layer *Siehe* Anwendungsschicht
ARQ 85
ASI 73
ASN.1 7; 120
Asynchrone Kommunikation 30
Asynchronous Transfer Mode *Siehe* ATM
ATM 31; 125
  AAL 127
  Dienstklassen 127
  Fast Explicit Congestion Notification 133
  Fast Reservation Protocol 133
  Netzarchitektur 125
  Netzwerkmanagement 133
  Protokollstack 126
  Zellformat 127
Auslastung 87
Automatic Repeat Request *Siehe* ARQ

B

Backbone Netz 21
Basic Encoding Rules 7; 120
Basisbandübertragung 17
BCC 89
BER *Siehe* Basic Encoding Rules
Binary Digit *Siehe* Bit
Binary Synchronous Control *Siehe* BSC
B-ISDN 31; 124
  H-Hierarchie 124
Bit 32
Bit Stuffing 37
Bitbus 74
Bit-Map-Protokoll 95
Bitsicherungsschicht 8
Block Check Character *Siehe* BCC
BRAM-Protokoll 96
Breitband-ISDN *Siehe* B-ISDN
Bridge 22
BSC 88
Bustopologie 19
Byte 32

C

CAN 5
Carrier Sense Media Access *Siehe* CSMA
CASE *Siehe* Common Application Service Element
CCITT
  CRC 81
  G.703 42; 123
  I.113 124
  I.120 122
  I.121 124
  SS #7 124
  V.24 47
  V.28 50
  X.21 90
  X.214 115
  X.224 115
  X.25 16; 31; 90; 102; 103; 135
  X.400 121
  X.500 121
CCITT High Level Language 117
CD 100
CHILL *Siehe* CCITT High Level Language
Circuit Switching *Siehe* Leitungsvermittlung
Client-Server-System 4
CMIS/CMIP *Siehe* Netzwerkmanagement
Code Stuffing 37

Codierung 25
  Alternate Mark Inversion (AMI) 42; 44
  Bi-Phase 41
  Bi-Phase-Mark (Manchester) 41
  Bi-Puls (Dipuls) 42
  CHDB 44
  Delay-Mark 41
  Differential-Bi-Phase 41
  Gruppencodierung 33
  HDB3 45
  Leitungscodierung 32
  Non Return to Zero (NRZ) 40
  Non Return to Zero Mark (NRZ-Mark) 41
  Return to Zero (RZ) 40
Collision Detection *Siehe* CD
Common Application Specific Element 121
Control Character *Siehe* Steuerzeichen
Controller Area Network 5
CRC 90; 101
CSMA 98
  p-persistentes CSMA 99
  Slotted CSMA 99
CSMA/CD *Siehe* Ethernet
Cyclic Redundancy Check *Siehe* CRC

D

Dämpfung 39
Darstellungsschicht 7; 120
Data Link Escape *Siehe* Steuerzeichen
Data Link Layer *Siehe* Datensicherungsschicht
Datagramm 16
Datensicherungsschicht 8; 83
Datenübertragungssystem 102
Datenverbund 13
Datex-P 103 *Siehe* X.25
Directory Service *Siehe* CCITT: X.500
Diskretisierung 25
DLE *Siehe* Steuerzeichen
Double Bus Double Queue 20
DQDB *Siehe* Double Bus Double Queue

E

Echtzeitverhalten 69
Effektive Bitrate 87
Electronic Mail 13; 23
Entfernung 11
Entropie 24
Escape-Zeichen *Siehe* Steuerzeichen
Ethernet 16; 34; 100
  BEB-Algorithmus 100
ETX/ACK-Protokoll 53

F

Factory Information Protocol *Siehe* FIP
FDDI 20; 39
FECN *Siehe* ATM
Feldbus 65
Fibre Distributed Data Interconnect *Siehe* FDDI
File Transfer, Access and Manipulation 121
FIP 81
Flußkontrolle 9
Fouriertransformation 25
Frame 22
Frame Relay **127**
Frequenzspektrum 26
FRP *Siehe* ATM
FTAM *Siehe* File Transfer, Access and Manipulation
Funktionsverbund 14

G

Gateway 23
Gekoppeltes Netz *Siehe* Internet
Gleichspannungsanteil 38
GPIB *Siehe* IEEE-488
Gradientenfaser 17; 29
Granularität 13
Gruppencodierung 39; 42
  4B3T 43

Paired Selected Ternary (2B2T) 43

H

Halbduplexbetrieb 34
HDLC 90
LAPD 123
HDLC-Protokoll 37
High Level Data Link Protocol *Siehe* HDLC-Protokoll
Hub 21

I

ICMP 109
IEC-625 *Siehe* IEEE-488
IEEE-488 57
Attention, ATN 59
Dreidraht-Handshake 63
Open-Collector-Treiber 61
Zweidraht-Handshake 62
Information 3
Transport 3
Informationstheorie 24
Integrated Services Digital Network *Siehe* ISDN
InterBus-S 80
Topologie 80
Internet 21
Internet Control Message Protocol *Siehe* ICMP
Internet Protokoll 108
Internetprotokoll 102
IPv4 108
Adressierung 112
Datagramm 110
Routing 113
ISDN 16; 31; 39; 42; 122
Basisanschluß 123
B-Kanal 122
Primärratenanschluß 123
ISO 6

J

Jam-Sequenz 100

K

Kanalkapazität 39
Kapazitätsverbund 15
Koaxialkabel 17; 28
Kommunikation
Asynchron 30
Paketübertragung 30
Synchron 30
Kommunikationssteuerungsschicht 8; 119
Kommunikationstechnologie 15
Kommunikationstheorie *Siehe* Informationstheorie
Kommunikationsverbund 13

L

LAN 5
Lastverbund 12
Layermodell 6
Leistungsbewertung 10
Leistungsverbund 12
Leitungsdämpfung 27
Leitungsvermittlung 16
Leitwegbestimmung 8
Lichtwellenleiter 17; 28
LLC 83; 94
Local Area Network 5; 11; 18
Wireless 19
Logical Link Control *Siehe* LLC
LWL *Siehe* Lichtwellenleiter

M

MAC 8; 93
MAN 5
Management Information Base *Siehe* MIB
Manchester-Codierung *Siehe* Codierung: Bi-Phase-Mark
Manufacturing Automation Protocol 121 *Siehe* MAP
Manufacturing Message Specification *Siehe* MMS

MAP 68
Media Access Control *Siehe* MAC *Siehe* MAC
Medienzugriffsstrategien
  Zufallsstrategien 98
  Zuteilungsstrategien 94
Metropolitan Area Network 5; 11
MHS *Siehe* CCITT: X.400
MIB 10; 14; 132
MMS 68
Modulation 30
Multiplexbetrieb 31
  Zeit 31
Multi-Point-Access-System 16

N

Nachricht 3
Network Layer *Siehe* Vermittlungsschicht
Netztopologie 19
Netzwerkmanagement 10; 14; 129
  CMIS/CMIP 132
  Configuration Management 129
  Fault Management 129
  Performance Management 130
Nyquist-Kriterium 30

O

Öffentliches Netz 18
Oktett *Siehe* Byte
Open System Interconnection 6
OSI 6
OSI-Basisreferenzmodell 35

P

PDU *Siehe* Protocol Data Unit
Point-To-Point-Verbindung 15; 20
Presentation Layer *Siehe* Darstellungsschicht
Process Field Bus *Siehe* Profibus
Profibus 76
  Fieldbus Data Link 78
  Fieldbus Management 78
  Fieldbus Message Specification 79
  Lower-Layer-Interface 79
  Nettodatenrate 80
Protocol Data Unit 114
Prozeßleitebene 68

Q

Quality of Service 116

R

RAID *Siehe* Redundant Array of Inexpensive Disks
Rechner 3
  Mehrprozessorsystem 3
Rechnernetz 4
Rechnerverbund 13
Redundant Arrays of Inexpensive Disks 14
Relais 22
Relay *Siehe* Relais
Repeater 22
Richtfunkverbindung 17
Ringtopologie 20
Router 23
Routing 8
  Adaptives Verfahren 9
Routing Control Center 138
Routing-Algorithmen 135
  Adaptiv 137
  Bewertungskriterien 136
  Nicht adaptiv 137
RS232 38; 47
RS422 38; 56
RS423 56
RS-485 77; 81
RS485 38; 56
RS562 51
Rundsendesystem 15; 19

S

SAP *Siehe* Service Access Point
SASE *Siehe* Specific Application Service Element
Satellitenverbindung 17; 19
SDLC 75
Server *Siehe* Client-Server-System
Service Access Point 115
Session Layer *Siehe* Kommunikationssteuerungsschicht
Simple Network Management Protocol *Siehe* SNMP
Simplexbetrieb 33
SNMP 132
Specific Application Service Element 121
Speichervermittlung 16
Sterntopologie 20
Steuerzeichen 36
- Acknowledge, ACK 54
- Data Link Escape, DLE 32; 37
- End of Text, ETX 36; 54
- End of Textblock, ETB 37
- Start of Text, STX 36
- SYN 36
- XOFF, DC3 53
- XON, DC1 53

Störabstand 38
Stufenindexfaser 17; 28
Switch 21
Synchrone Kommunikation 30
Synchronous Data Link Control *Siehe* SDLC

T

Taktrückgewinnung 35; 39
TCP *Siehe* Transport Control Protocol
Technical and Office Protocol *Siehe* TOP
Token 20; 78; 97
Token-Bus-Protokoll 97
Token-Ring-Protokoll 97
TOP 68
Transitsystem 102
Transmission Block *Siehe* Übertragungsblock
Transport Control Protocol 117
Transportprotokoll 114
Transportschicht 8
Twisted Pair 17; 28

U

Übertragung
- Asynchron 35
- Redundanz 39
- Synchron 35
- Übertragungsblöcke 36

Übertragungsprotokoll 7
Übertragungsprozedur 35
Übertragungsstrecke 27
UDP *Siehe* User Datagram Protocol
User Datagram Protocol 117
Utilization *Siehe* Auslastung

V

Verbindungskontrolle *Siehe* LLC
Vermittlungsnetz 15
Vermittlungsschicht 8; 102
Verteilte Systeme 13
Virtual Calls *Siehe* Virtuelle Verbindungen
Virtuelle Verbindungen 103
Virtuelles Terminal 121
Vollduplexbetrieb 34

W

WAN 5
Wartungsverbund 14
Wide Area Network 5; 11; 19
Wireless Local Area Network 19
WLAN *Siehe* Wireless Local Area Network

X

X.25 *Siehe* CCITT: X.25
XON/XOFF-Protokoll 53