

Rechnernetze I

Einleitung, Motivation

Geschichte des Internet

- 1965:** Experimentelles Rechnernetz mit 3 Rechnern am MIT aufgebaut
- 1968:** Verbindung von Rechnern über ein gemeinsames Medium. Ziel: Datenübertragung bei einem nuklearen Ernstfall/Verteidigungsfall.
- 1969:** Gründung der Network Working Group. Die ersten vier Rechner werden bei größerer Entfernung miteinander verbunden(UCLA, UCSB, SRI, University of Utah)
- 1971:** Experiment: Einloggen in entfernte Rechner (rlogin); Nutzung von E-mail.
- 1972:** Erste öffentliche Demonstration des Netzwerkes (ARPANET)
- 1973:** Erste Satelliten-Übertragungsstrecke (Hawaii - CA)
- 73/74:** Entwurf der TCP/IP-Protokolle zur Verknüpfung verschiedener Netze (aus Protokollen des ARPANET entstanden)
- 1975:** Verwaltung an das Department of Defense übergeben
- 1980:** TCP/IP-Protokolle in Berkeley-UNIX (BSD) frei verfügbar

Ende

80er: Internet breitet sich auch in Europa, Asien,... aus;
Wachstumsrate: 10-15% mehr Rechner pro Jahr,
über 25% mehr Verkehr pro Jahr

89-92: 85% Zuwachs an Rechnern zwischen '89 und '92
(80.000 Rechner '89, 727.000 Rechner '92
> 2.056.000 Rechner '93)

91-98: 105% Zuwachs an Datenvolumen $1.1 \cdot 10^{13} = 11$ Tera
Bytes/Monat werden ausgetauscht)
120% Zuwachs an Netzen (28.578)
- IP Version 6 erscheint im Markt (IP. v 6)
- IP Telephony wird populär (voice over IP)

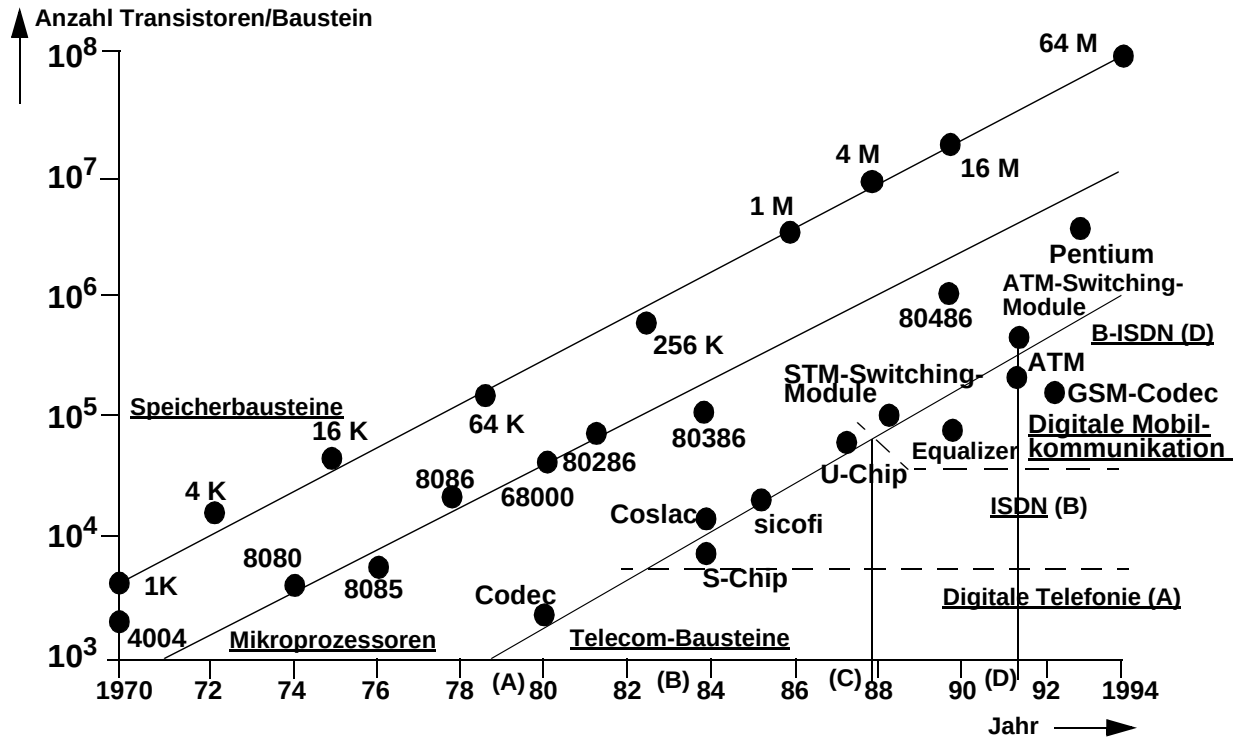
=> Explosives Wachstum des Internet ohnegleichen (exponentielle Rate mit steigendem Exponenten)

=> es ist interessant und wichtig, sich mit dem Internet, d. h. mit Rechnernetzen zu beschäftigen

„Internet ist das Netz der Netze“

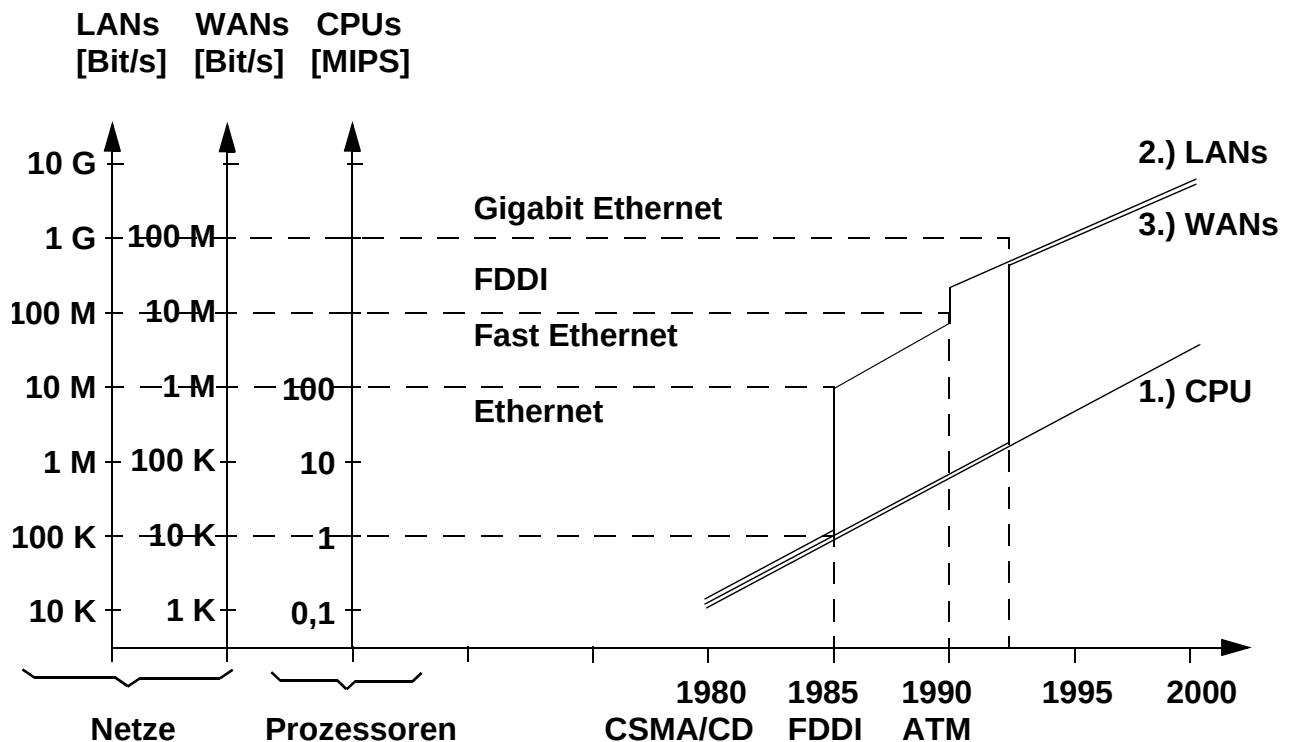
Warum wächst das Internet exponentiell?

z. B. aufgrund von Fortschritten in der Mikroelektronik



- => Integrationsdichte der Chips steigt gemäß dem Gesetz nach Moore (exponentielle Rate).
- => immer komfortablere Funktionen bei gleichzeitig sinkendem Preis sind möglich
- => jedermann kann das Internet bedienen
- => Handel kann über das Internet abgewickelt werden (eCommerce)
- => riesiges Angebot an Waren und Informationen im Internet

Geschwindigkeitssteigerungen in Mbits und MIPS bei LANs und WANs und Prozessoren



MIPS = millions instructions per second

LAN = local area network

WAN = wide area network

=> schnellere Prozessoren bedingen schnellere Netze und umgekehrt

Typen von Rechnernetzen, Abgrenzung gegenüber anderen Vernetzungen

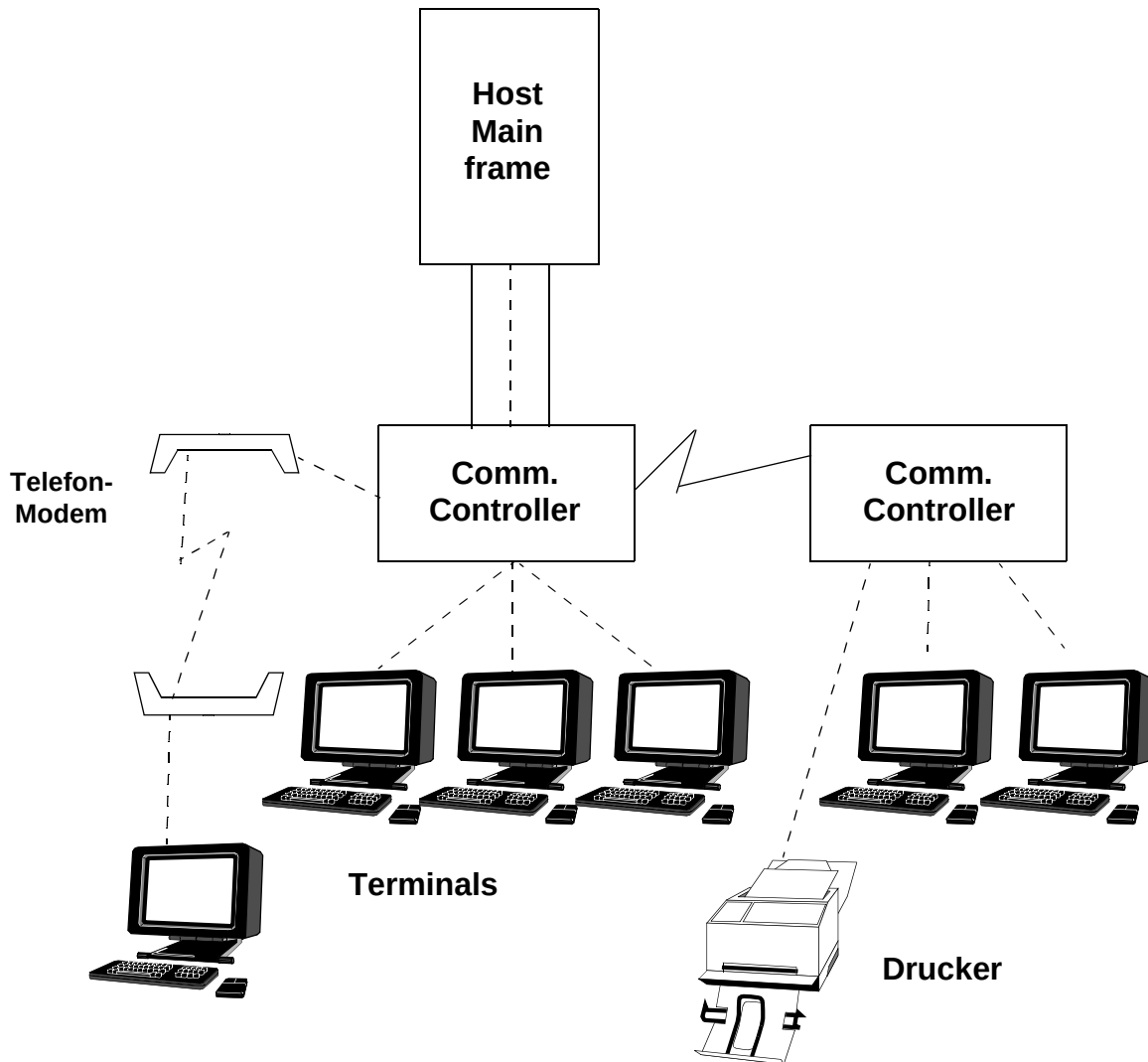
- Definition Rechnernetz:

dient zur Kopplung unabhängiger Rechner zum Zweck des Datenaustauschs

Abgrenzung gegenüber

- **Bus, Kanal (in einem Prozessor, einem Rechner oder Mainframe)**
- **Verbindungsnetzwerk eines Parallelrechners (Mehrprozessor-System $\neq$ Verteiltes System)**
- **Terminalnetz (mit und ohne Modems)**
- **TP-Monitor = spezielle Datenbankinformationssysteme
z. B. zur Flugbuchung (transaction processing monitor)**

Terminalnetz (Ende der '60er Jahre)

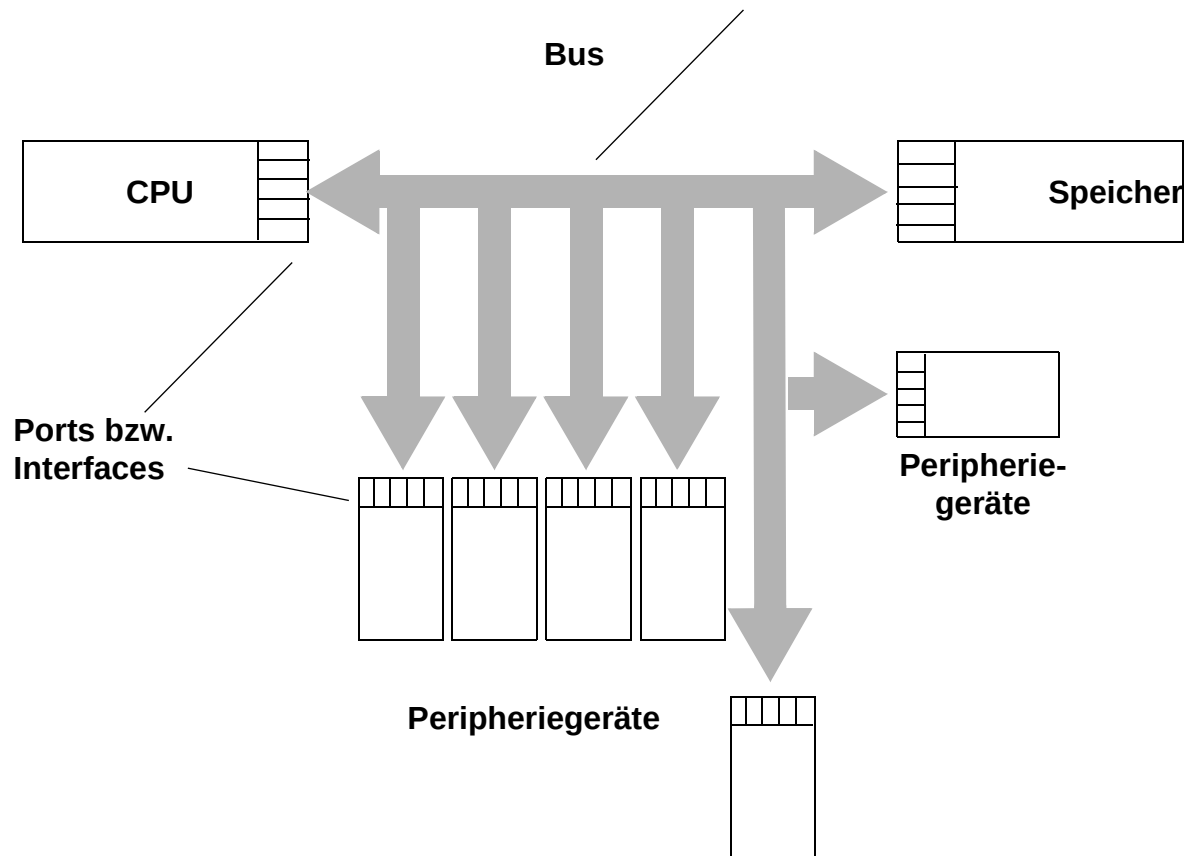


kein Rechnernetz!

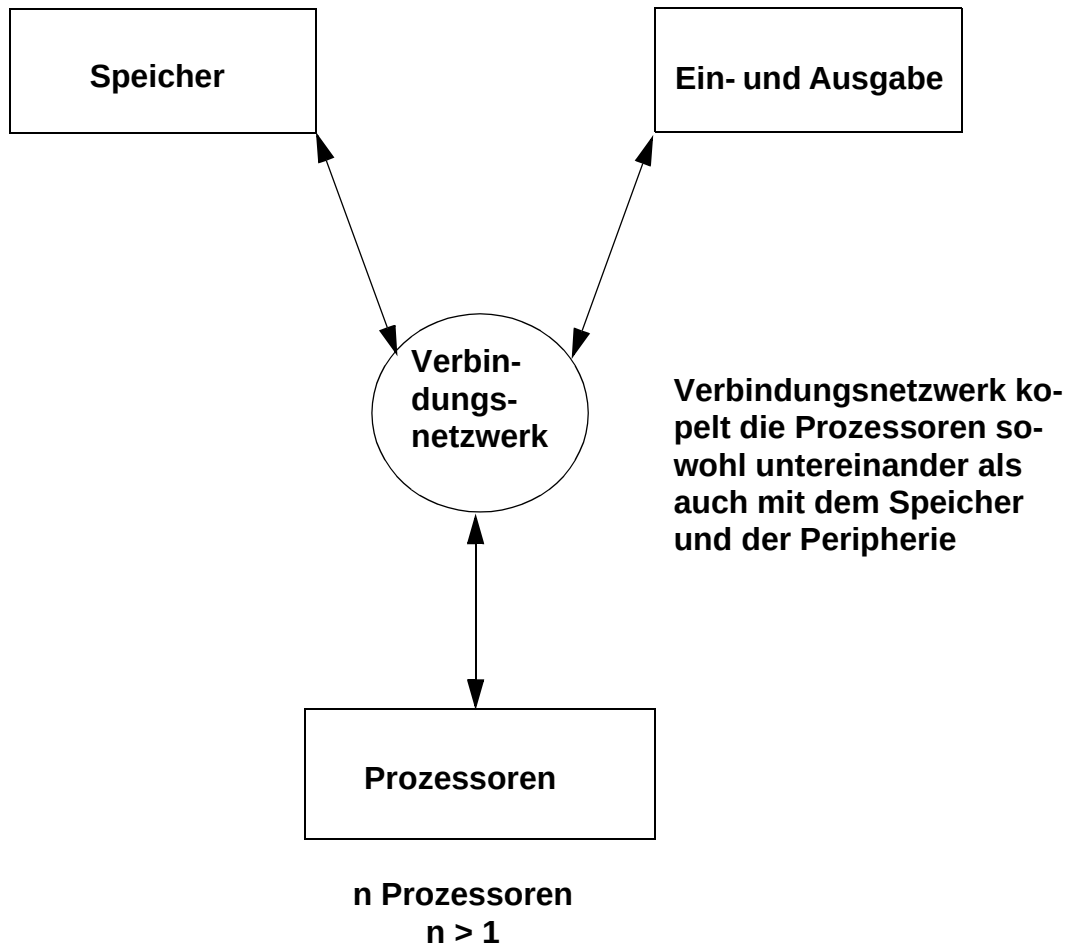
weitere Beispiele für Nicht-Rechnernetze:

Bus- und Verbindungsnetzwerk

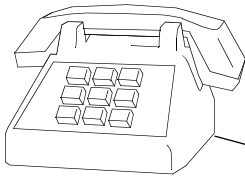
- Bus in einem Rechner:



IVerbbindungsnetzwerk in einem Parallelrechner

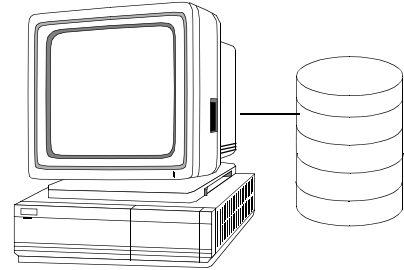


Bürosysteme (historisch)



**Fax-
Gerät**

**1.)
Voice-Mail-
System, z.B.
Telefon
Telefax**



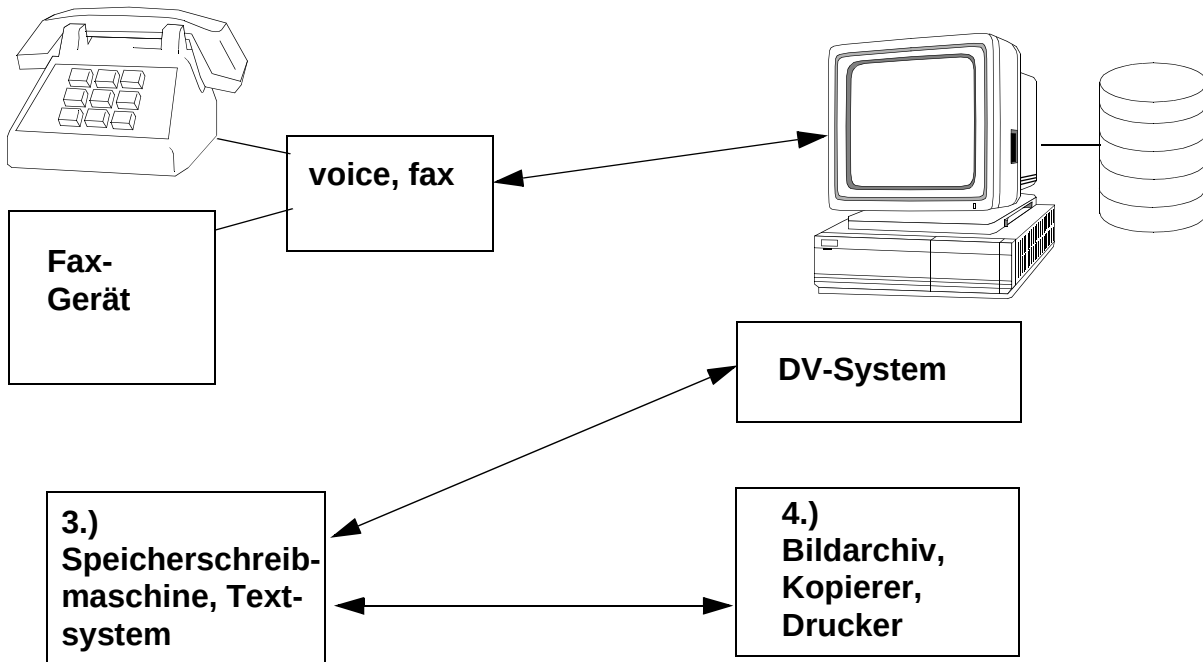
**2.)
DV-System**

**3.)
Speicherschreib-
maschine, Text-
system (PC)**

**4.)
Bildarchiv,
Kopierer,
Drucker**

**Verschiedene unabhängige Subsysteme,
kein Rechnernetz!**

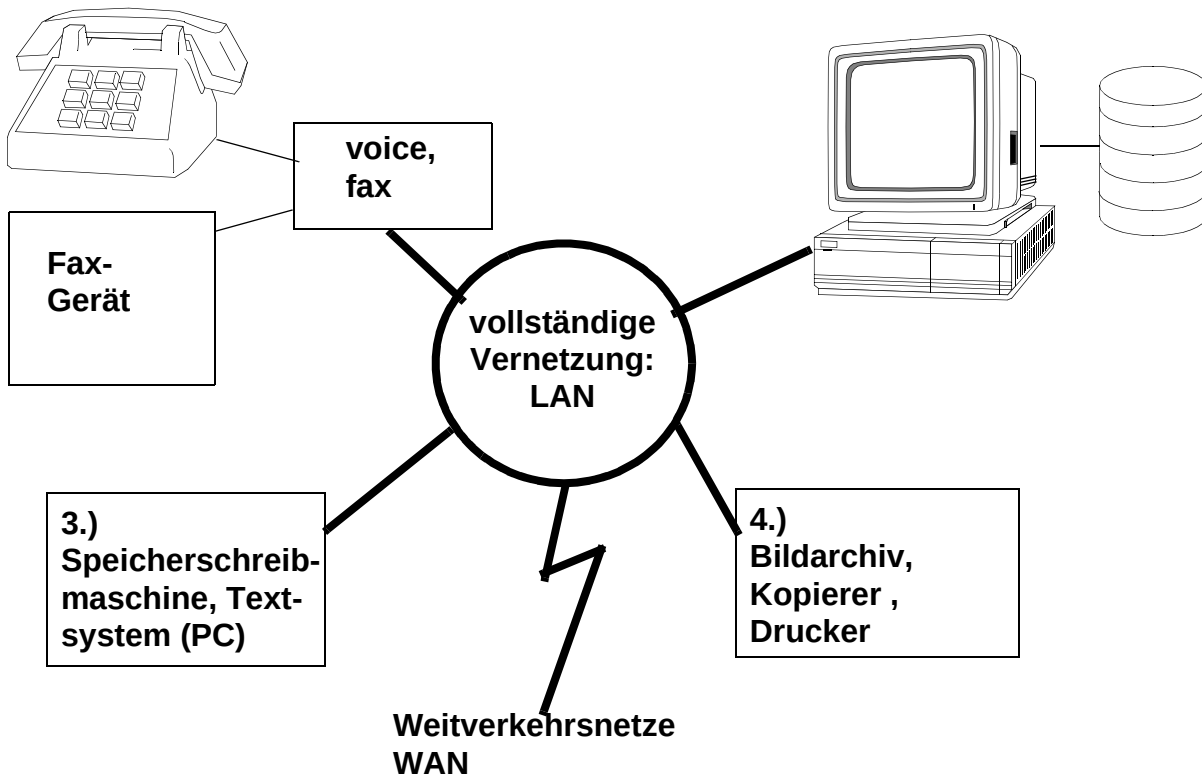
Beispiel für Rechnernetz: Kommunikation zwischen unabhängigen Subsystemen (Rechnernetz)



- => Funktionsverbund

- Netz ist nur lokal

Zielsetzung: globales Netz:



LAN = Local Area Network

WAN = Wide Area Network

Ziele der Rechnernetze

- **Datenverbund**

- Zugriff auf entfernte Daten, z. B. über NFS

- **Funktionsverbund**

- Zugriff auf Spezialrechner, z. B. Compute-Server, Data Base-Server)

- **Lastverbund**

- gleichmäßige Lastverteilung z. B. über LSF

- **Verfügbarkeitsverbund**

- Fehlertoleranz, schrittweises Wachstum

- **NFS = network file system**

- **LSF = load sharing facility**

- **Speziell für das Internet gilt zusätzlich:**

- **Informationsverbund (riesiges Lexikon weltweit verteilter Informationen)**

- **Infrastruktur für Handel und Dienstleistungen (eCommerce)**

Klassifikation von Verbindungen bei Netzen nach der Distanz

	Interprocessor distance	Processors located in same	Example
keine Rechner-netze	0,1 m	Circuit board	Multiprocessors and Multicomputers
	1 m	System	
	10 m	Room	
Rechner-netze	100 m	Building	Local area network (LAN)
	1 km	Campus	
	10 km	City	Metropolitan area network (MAN)
	100 km	Country	
	1.000 km	Continent	Wide area network (WAN)
	10.000 km	Planet	
			The Internet

Wer beschäftigt sich mit dem Aufbau von Netzen?

Normungsgremien

Normung des Begriffs „Normung“ (DIN 820)

Normung ist die planmäßige, durch die interessierten Kreise gemeinschaftlich durchgeführte Vereinheitlichung von materiellen und immateriellen Gegenständen zum Nutzen der Allgemeinheit.

Normungsinstitutionen

- International Organization for Standardization (**ISO**)
- International Telecommunications Union (**ITU**), ehemals: Comité Consultatif International de Télégraphie et Téléphonique (CCITT), heute Untereinheit der UNO
- European Computer Manufacturers Association (**ECMA**)
- CEN/CENELEC/ETSI/**CEPT** (europäisch)
- National Institute of Standards and Technology (**NIST**, Japan)
- Electronic Industries Association (**EIA**, USA)
- American National Standards Institute (**ANSI**, USA)
- Institute of Electrical and Electronic Engineers (**IEEE**, USA)

International Standards Organization (ISO)

- Normung auf internationaler Ebene durch die ISO
- Mitglieder: Nationale Normungsgremien (DIN/BRD, ANSI/USA, AFNOR/Frankreich,...)
 - ISO TC 97: Information Processing Systems
 - TC 97/SC 6: Data Communications
 - TC 97/SC 18: Text and Office Communications
 - TC 97/SC 21: Open Systems Interconnection (=OSI)
- Stufen einer Norm
 - 1.) Working Draft
 - 2.) Draft Proposal
 - 3.) Draft International Standard
 - 4.) International Standard
- ISO-Normen besitzen keine nationale Rechtsverbindlichkeiten (nur sogenannte recommendations)

TC = Technical Committee

SC = Subcommittee

AFNOR = „das französische DIN“

Draft = vorläufiger Entwurf

Wer setzt die Standards im Internet?

- **1983: Gründung des IAB (Internet Activity Board)**
- **IAB besteht aus:**
 - **IETF (Internet Engineering Task Force), kurzfristige und praktische Aufgaben**
 - **IRTF (Internet Research Task Force), langfristige Forschung**
 - **W3C (www consortium) Tim Bernes Lee als Gründer und Leiter**
- **RFCs (Requests for Comment)**
- **sind nicht immer Standarddokumente, können auch rein informell sein**
- **Arbeitsgruppen mit jeweils einem Leiter**
 - **Mitglied der Arbeitsgruppe kann jeder werden**
 - **Kommunikation erfolgt vorwiegend über E-Mail oder Videokonferenzen, d. h. weltweit verteilt**
 - **typische Arbeitszeit: 9-18 Monate**
 - **Ergebnis: Proposed Standard**
- **Draft Standard: mindestens zwei unabhängige Implementierungen; Interoperabilitätstests; Stabilität über 4 Monate**
--> Internet Standard

Wer mischt noch mit?

Industriekonsortien

- **Zusammenschluss vorwiegend industrieller Partner**
 - **Ziel: rasche Realisierung untereinander kompatibler Produkte verschiedener Hersteller**
 - **deshalb: schnelle Entwicklung eines gemeinsamen de-facto Standards (im Gegensatz zum de-jure-Standard)**
 - **Einbringung der Ergebnisse in die internationale Standardisierung, z. B. in die ISO**
 - **Beispiele:**
 - **ATM-Forum, WAP-Forum**
 - **Problem: Vorgehensweise manchmal zu schnell, so dass interessante und richtungsweisende Forschungsergebnisse keinen Eingang in die de-facto Standardisierung finden.**
-
- **ATM = Asynchronous Transfer Mode = spezielle Hochgeschwindigkeits-Übertragungsprotokolle**
 - **WAP = Wireless Application Protocol = Protokoll für sog. Wap-Handys**

Standardmodell der Kommunikation: **Das ISO-Referenzmodell für Offene** **Systeme**

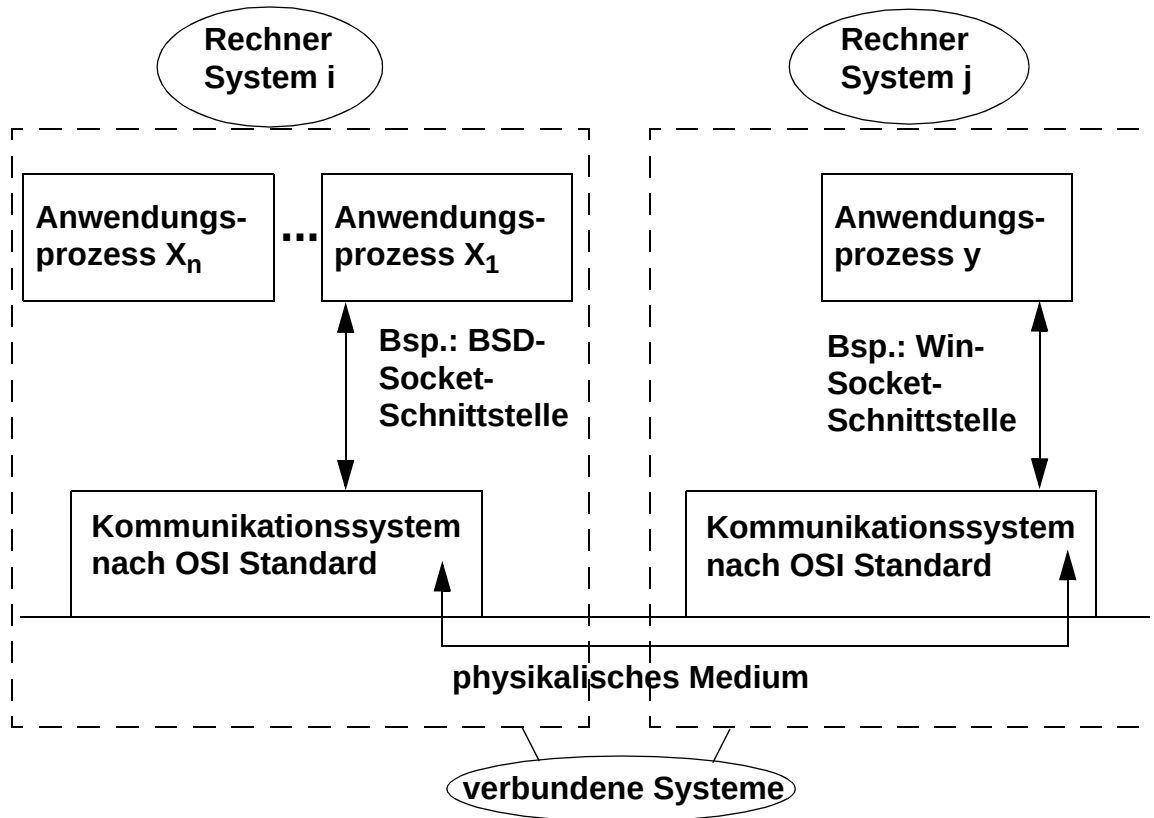
- OSI = **OPEN SYSTEMS INTERCONNECTION** = Untergruppe der ISO
- Kurz: **ISO/OSI-Referenzmodell aus den 70er-Jahren**
- ISO International Standard 7498
- Ein Modell für **geschichtete Kommunikationssysteme**
- Einführung neuer Grundbegriffe (Terminologie)
- Schlägt **sieben Schichten** und ihre Funktionalität vor (zuvor hatte IBM bereits ein Sieben-Schichten-System-eingeführt)

Schicht	ISO	engl.:
7	Anwendungsschicht	application
6	Darstellungsschicht	representation
5	Kommunikationssteuerungsschicht	session
4	Transportschicht	transport
3	Vermittlungsschicht	network
2	Sicherungsschicht	data link
1	Bitübertragungsschicht	physical

- Jede Schicht hat bei der Datenübertragung einen eigenen Satz von Aufgaben zu übernehmen
- Datenaustausch ist nur zwischen benachbarten Schichten möglich, nicht kreuz und quer

Standard-Modell der Kommunikation

ISO/OSI-Kommunikationssystem



- **ISO = International Standards Organisation**
- **OSI = Untergruppe der ISO (= Open Systems Interconnection)**

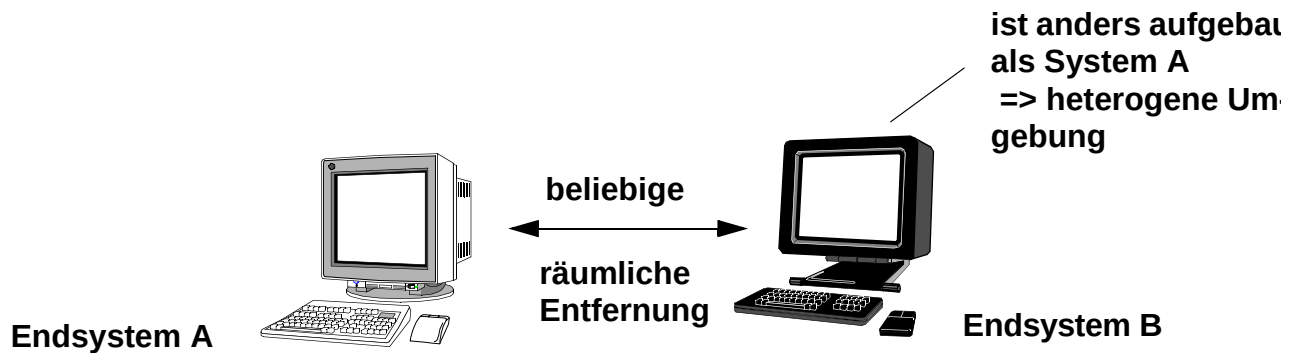
Grobstruktur im ISO-Referenzmodell

- Schicht 1 - 4**
- Transportorientierte Schichten = Schichten 1-4: Technische Erbringung von Bitstrom-Übertragungen
 - Pure Nachrichtenübertragung; nur Bedürfnis des Nachrichtenaustauschs unterstellt; keine höheren Funktionen
 - kein Bezug auf die Kooperationsbeziehung der Teilnehmer
 - Inhalt der Nachricht ist ohne Bedeutung für die transportorientierten Schichten
- Schicht 5 - 7**
- Anwendungsorientierte Schichten = Schichten 5-7: Anwendungsbezogene Kommunikationsdienstleistungen
 - Kooperation der Teilnehmer ist auf formale Weise berücksichtigt
 - Steuerung des Ablaufs (session)
 - Informationsdarstellung (Datenformate)
 - Kompensation von Fehlverhalten (Kryptographie)
 - Datenverschlüsselung
 - Datenkompression

=> Schicht 1-4 definiert Syntax der Daten

=> Schicht 5-7 definiert Semantik der Daten

Strukturierung des Problems der Kommunikation durch Hierarchieebenen:



Teil der Betriebssystem-Software bzw. Bibliotheken	Anwendungsschicht	Anwendungsspezifische Protokolle	7
	Darstellungsschicht	Datenformatierung und -darstellung	6
	Sitzungsschicht	Dialogsteuerung	5
	Transportschicht	Ende-zu-Ende-Datenaustausch	4
	Vermittlungsschicht	Ende-zu-Ende-Vermittlung	3
Teil der Rechner-Hardware	Sicherungsschicht	nur zwischen benachbarten Rechnern	2
	Physikalische Schicht	Physikalische Übertragung	1

Die sieben Schichten des ISO-Referenzmodells

- 1.) Die **Bitübertragungsschicht** ermöglicht die transparente Übertragung eines Stroms binärer Informationen über ein Kabel (ungesicherter Bitstrom, mech. u. el. Spezifikation)
- 2.) Die **Sicherungsschicht** soll insbesondere Übertragungsfehler entdecken und korrigieren (gesicherte Bits + Rahmung + Flusssteuerung)
- 3.) Die **Vermittlungsschicht** übermittelt Daten auf eine der Anwendung verborgenen Weise. Dazu wird von dieser Schicht selbstständig eine Netz-Route ausgewählt (von Rechner zu Rechner über Zwischenknoten)
- 4.) Die **Transportschicht** übermittelt die Daten von Endprozess zu Endprozess. Sie entlastet den Benutzer von den Details der Datenübertragung
- 5.) Die **Kommunikationssteuerungsschicht** koordiniert die Zusammenarbeit zwischen den verschiedenen miteinander kommunizierenden Anwendungsprozessen (Wiederanlauf = recovery, full-/halfduplex)
- 6.) Die **Darstellungsschicht** transformiert die Darstellung der übermittelten Daten in eine Form, die von den kommunizierenden Anwendungsprozessen verstanden wird (Datenformate, Kompression, Kryptologie = Verschlüsselung)
- 7.) Die **Anwendungsschicht** muss den Anforderungen der Benutzer genügen. Die Anwendungsschicht ist die einzige Zugriffsmöglichkeit der Anwendungsprozesse zur Datenübertragung (z. B. E-Mail, ftp, nfs,...)

Layer	ISO	Internet	SNA (IBM) - früher als das ISO-Modell erschien - hat bereits 7 Schichten
7	Application	SMTP, FTP telnet, http	End user
6	Presentation		NAU Service
5	Session		Data flow control
4	Transport	TCP	Transmission control
3	Network	IP	Path control
2	Data link control	Data link control	Data link control
1	Physical	Physical	Physical

- Zusammenhang zwischen dem ISO-7-Schichten-Modell und der Realität im Internet

SNA = System Network Architecture
= proprietäres Netzkonzept von IBM

Sprachgebrauch und Prinzipien des ISO-Referenzmodells

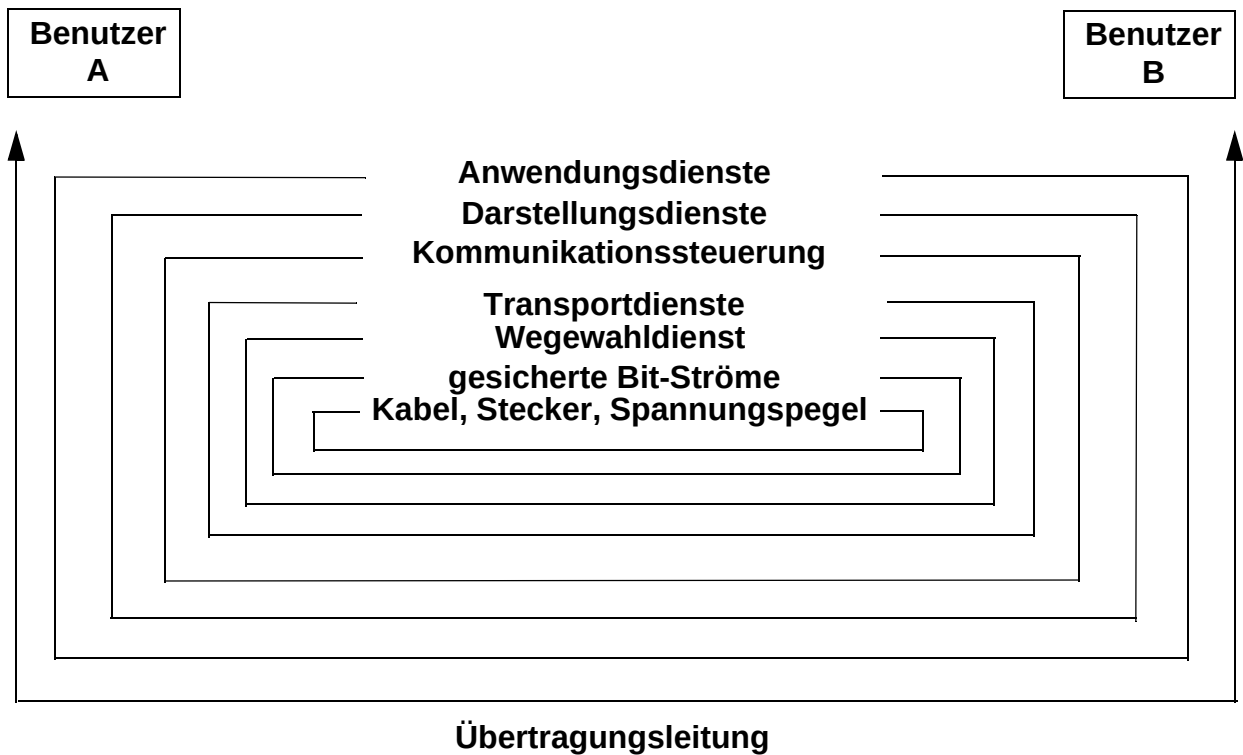
- **Offenes System** für die Kommunikationsteilnehmer
 - jedes Rechnersystem (Hardware, Software, Peripherie, ...), das sich bei der Kommunikation an die OSI-Standards hält, kann kommunizieren.
- **(N)-Schicht** $1 \leq N \leq 7$
 - wird aus sämtlichen Einheiten einer (N)-Schicht in allen offenen Systemen gebildet. Analogie: dieselben Stockwerke aller Gebäude der Welt bilden eine (N)-Schicht
- **(N)-Instanz, engl. (N)-Entity; Implementierung einer (N)-Schicht in einem System = Software/Hardware**
 - es kann verschiedene Typen von (N)-Instanzen geben, die verschiedene Protokolle für die (N)-Schicht implementieren
- **Partnerinstanzen, Peer-Entities**
 - Instanzen derselben Schicht an verschiedenen Orten. Partnerinstanzen erfüllen die Funktionen einer Schicht durch **Datenaustausch von peer zu peer.**

peer = Gleichgestellter, d. h. einer, der der gleichen Schicht angehört

Was macht eine Schicht?

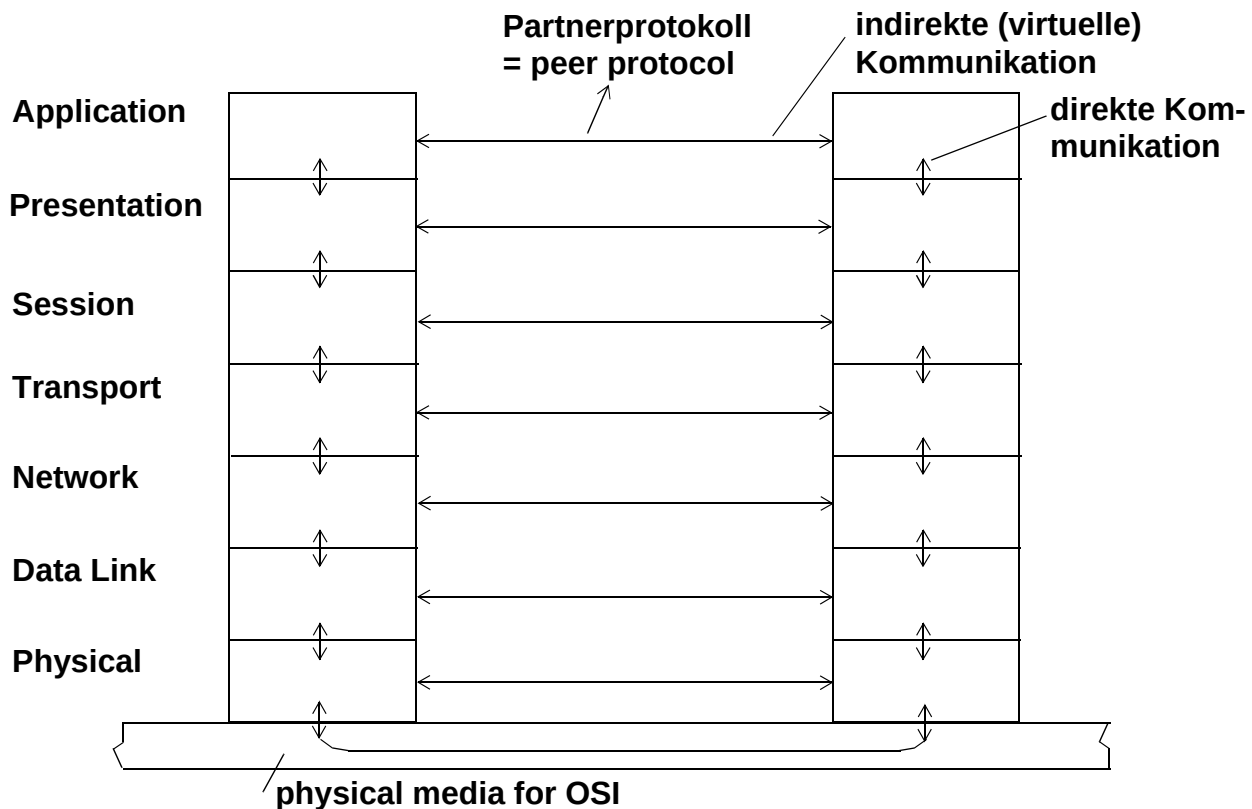
- **Hauptaufgabe jeder Schicht ist es, der darüberliegenden Schicht Dienste anzubieten. Diese Dienste setzen sich zusammen aus:**
 - **Dienstleistungen, die innerhalb dieser Schicht erbracht werden und**
 - **der Summe der Dienstleistungen aller darunterliegenden Schichten**
- **Schichten sind über sogenannte Dienstprimitive (= Dienstelemente) miteinander verknüpft**
- **Da Schichten jeweils aufeinander aufbauen, ergibt sich eine Schichtenhierarchie bzw. eine Hierarchie von Diensten**

Hierarchie von Diensten



- Jede Schicht benutzt die Funktionen (= Dienste) der darunterliegenden Schicht
- Daten werden physikalisch nur auf der untersten Ebene zwischen den Rechnern übertragen, d. h. der Übertragungsleitung
- Auf alle n-Schichten >1 werden Daten innerhalb desselben Rechners nur „hin und her geschoben“

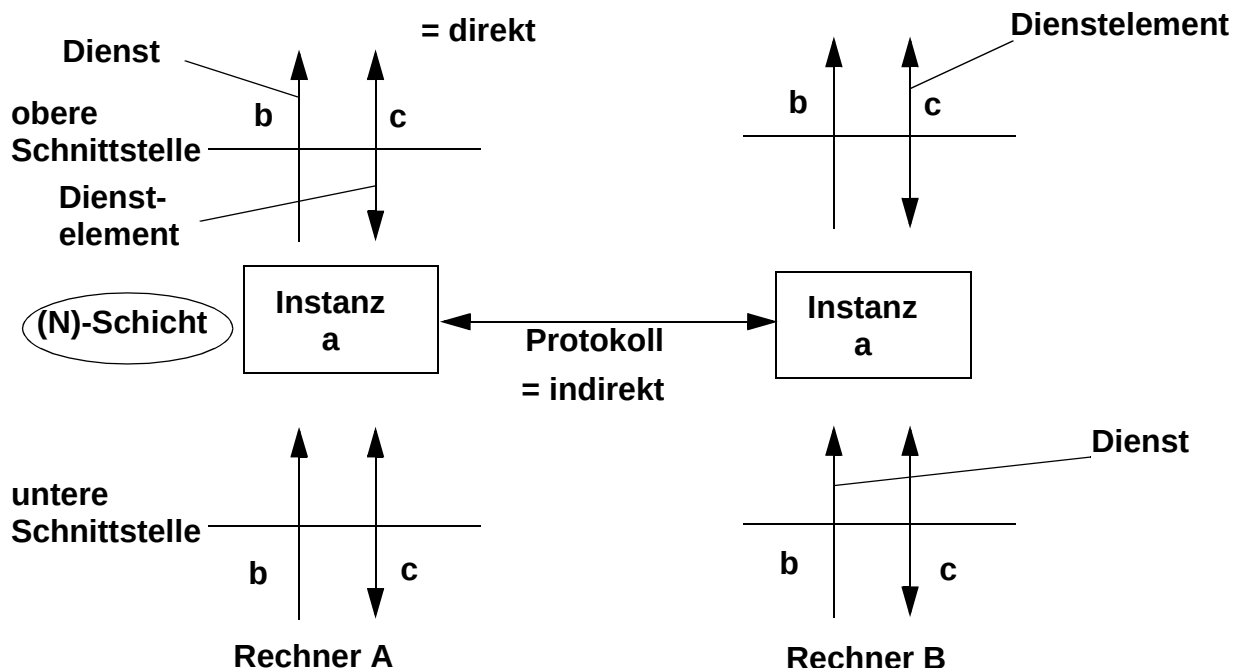
Indirekte Kommunikation im ISO-Referenzmodell/Partnerprotokolle



- Daten werden innerhalb eines Rechners von Schicht zu Schicht transportiert
- Zwischen 2 Rechnern können Daten nur auf der Schicht 1 transportiert werden
- Kommunikation zwischen Instanzen höher als Schicht 1 werden indirekt (virtuell) abgewickelt
- Die indirekte Kommunikation wird mittels eines Protokolls zwischen Partnerinstanzen durchgeführt, so dass beide Partnerinstanzen „sich verstehen“ können (Protokoll = gemeinsame Sprache)

Direkter und indirekter Datenaustausch

- Direkter Datenaustausch erfolgt mit Hilfe von sog. Dienstelementen eines Dienstes zwischen benachbarten Schichten des selben Rechners
- Indirekter Datenaustausch erfolgt mit Hilfe eines Protokolls zwischen Partnerinstanzen auf verschiedenen Rechnern
- Direkte Kommunikation mit Schichten (N+1) und (N-1) für Schicht N
- Indirekte Kommunikation (Protokoll) mit Partnerinstanzen (per entities)



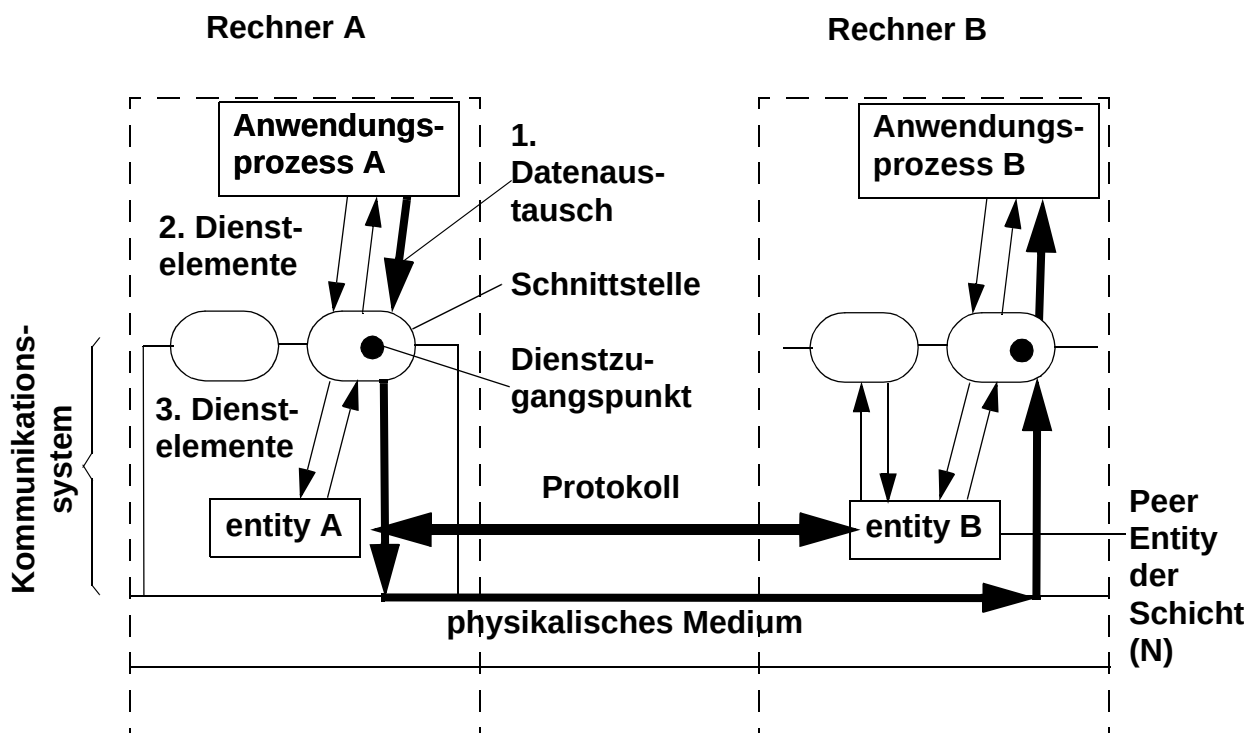
a = gleichgestellte (N)-Instanzen (Peer Entities)
b = Dienstleistungen (Layer Service) = Dienst
c = Dienstprimitive (Service Primitives) = Dienstelement
d = Protokoll zwischen gleichgestellten (Peer Protocol)

- Jeder Dienst besteht aus einer Menge von Dienstelementen

Was ist ein Protokoll?

Definition:

Menge der Regeln für den indirekten Datenaustausch zwischen Entities derselben Schicht (peer-to-peer communication)

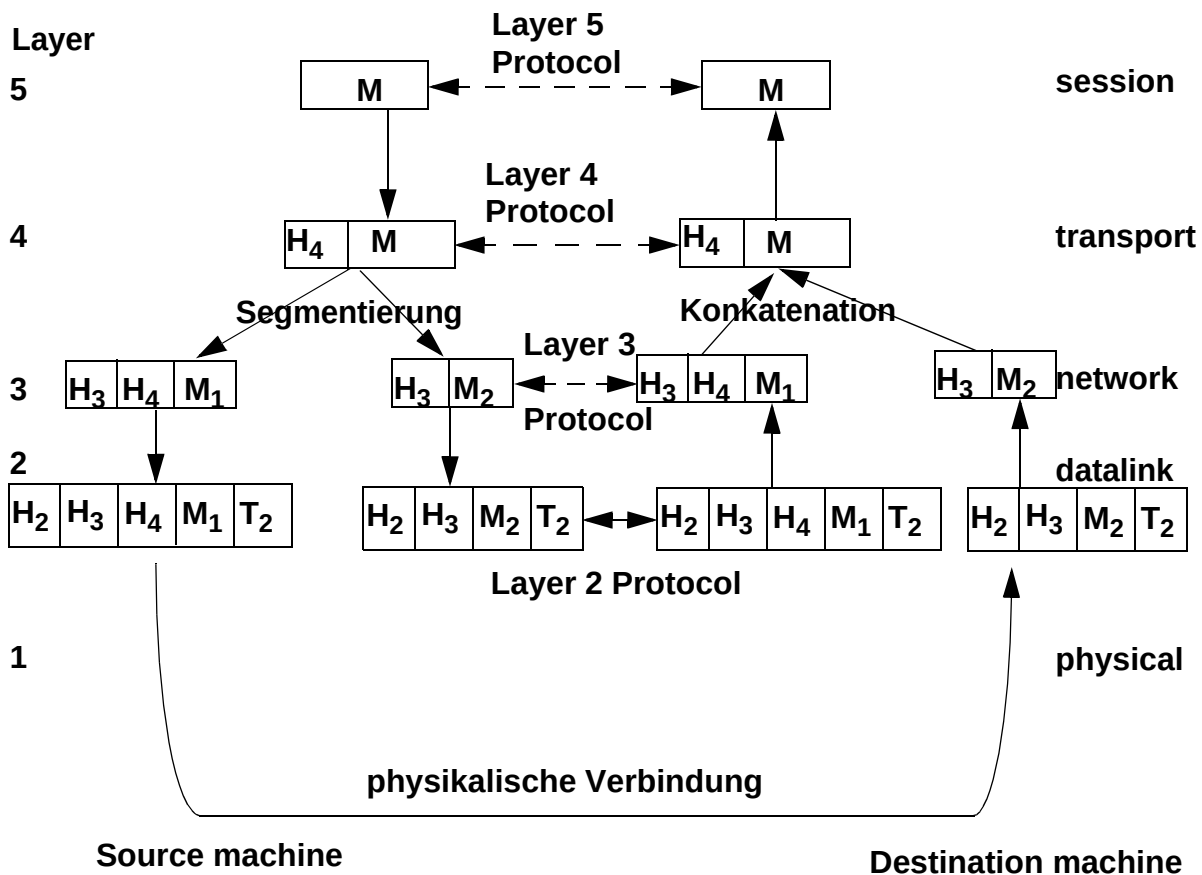


Datenaustausch --> Dienstelement --> Protokoll

- **Schnittstelle:** Menge von Unterprogrammencalls einer direkt darunterliegenden Schicht N-1
- **Dienstzugangspunkt:** Ein spezieller Unterprogrammaufruf
- **Jedes Dienstelement hat einen Dienstzugangspunkt, an dem das Resultat des Dienstelements vorliegt und an dem das Dienstelement zur Verfügung steht.**

Beispiel für virtuelle und reale Kommunikation nach ISO/OSI

Nachricht wird von „oben nach unten“ und dann von „unten nach oben“ durchgeschoben.



M = message = Nachricht

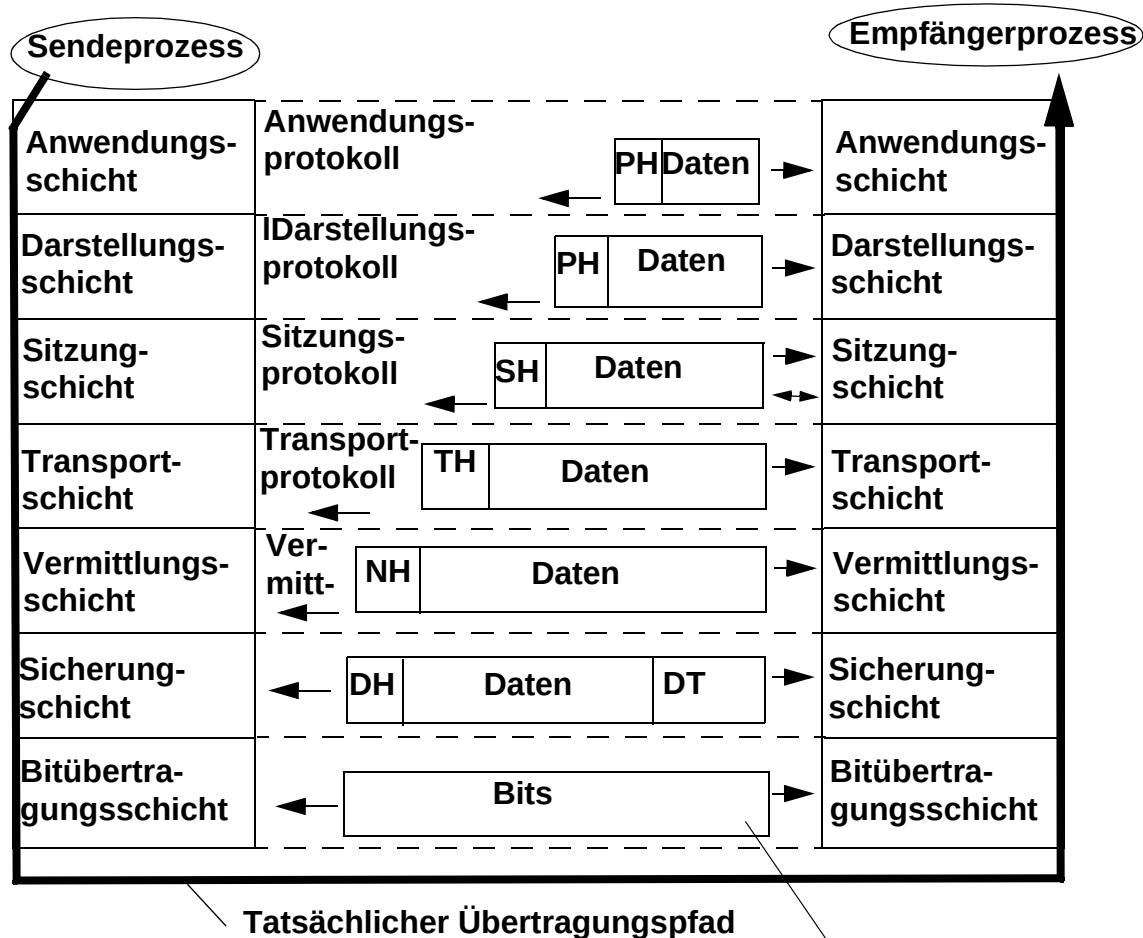
H = header = Vorspann = Nachrichtenkopf

T = trailer = Nachspann

- Nachricht wird bei jedem Übergang zwischen 2 Schichten um einen Vorspann (Header) verlängert
- Das Konzept der Nachrichtenverlängerung durch Header ist wesentlich beim ISO/OSI-Modell

2. Beispiel für virtuelle Kommunikation

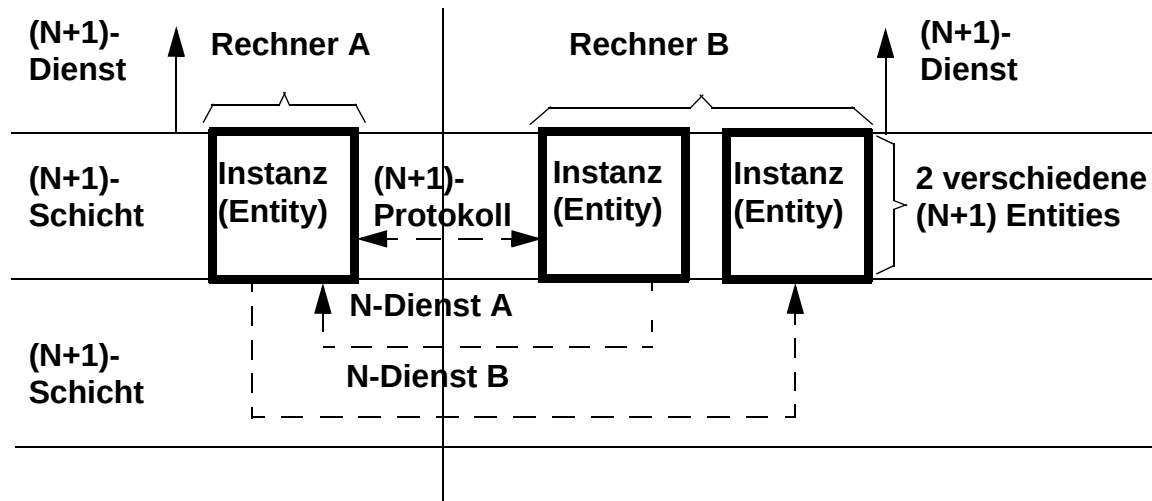
Ein 2. Beispiel dafür, wie sich das OSI-Modell auf Nachrichtenformate auswirkt. Einige der Nachrichtenköpfe (Header) können im Allgemeinfall auch leer sein.



AH = application header
 PH = presentation header
 SH = session header
 TH = transport header
 NH = network header
 DH = data link header

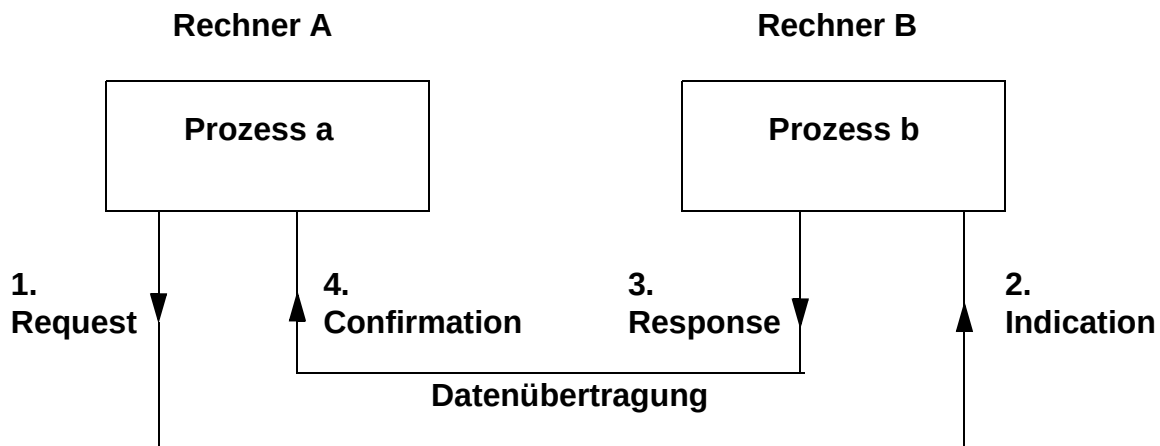
tatsächlich übertragene Bits

Wiederholung: Unterschied zwischen Dienst und Protokoll



- Das Protokoll der Schicht (N+1) benutzt die Dienste der Schicht (N) und implementiert einen Dienst der Schicht N+1
- Ein Protokoll legt das Format und die Bedeutung der virtuell ausgetauschten Daten fest
- Ein Dienst nach ISO ist eine Menge von Dienstelementen ohne zu sagen, wie diese implementiert sind

Anwendung der Dienstelemente (Dienstprimitive)



- Es gibt 4 Typen von Dienstelementen

Die 4 Arten von Dienstprimitiven (Dienstelementen)

1.) Anforderung (Request)

= Aktivieren einer Dienstleistung durch den Benutzer

2.) Anzeige (Indication)

= Dem Benutzer anzeigen, daß vom entfernten Benutzer ein Dienst angefordert wurde, ohne dass eine Dienstleistung vom Anbieter der Dienstleistung selbst ausgelöst wurde

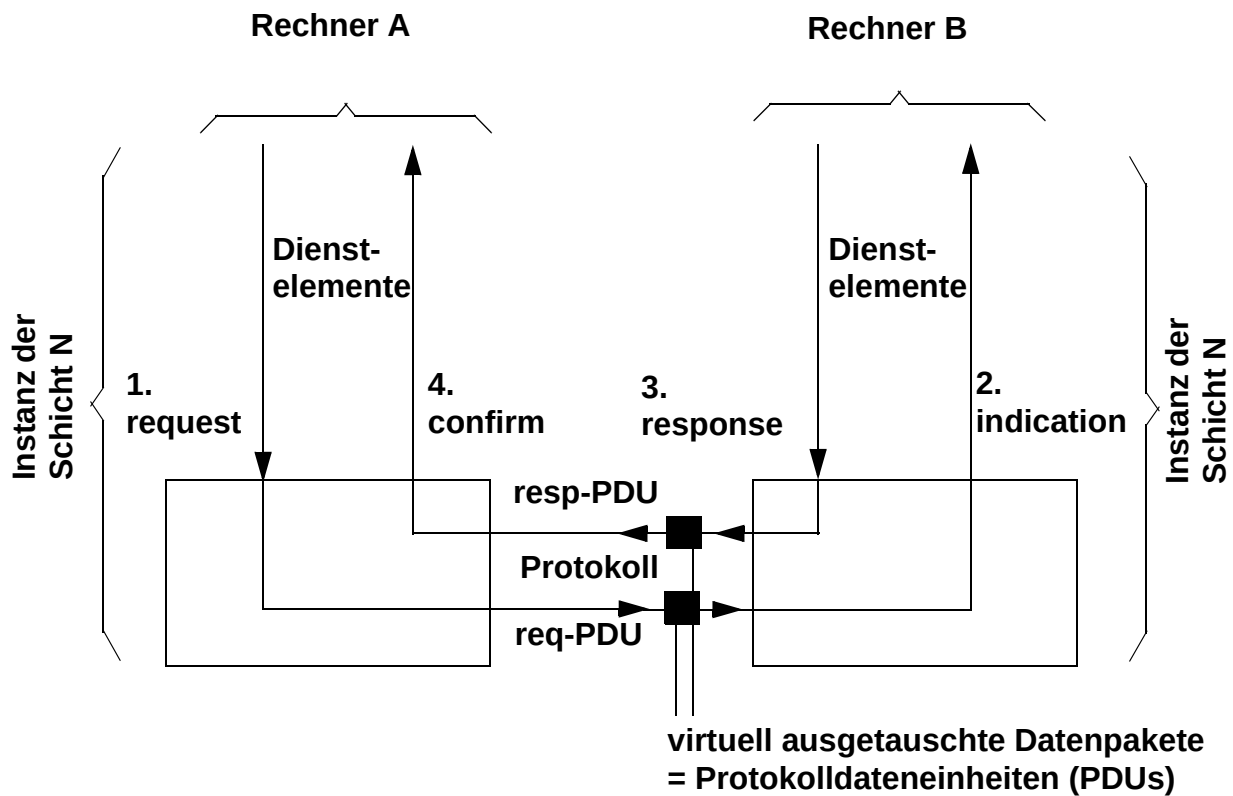
3.) Antwort (Response)

= Quittieren einer vorher. Anzeige durch den Benutzer

4.) Bestätigung (Confirmation)

= Quittieren einer vorherigen Anforderung durch den Anbieter

Zusammenhang zwischen Diensten und Protokollen



PDU = protocol data unit; resp. = response; req = request

=> 2 ungelöste Fragen:

1.) Wie wird ein Dienst aufgerufen?

2.) Wie entsteht aus einem Dienstelement eine PDU?

Lösung: Die Schnittstelle zwischen zwei aufeinanderfolgenden Schichten im selben Rechner = Umwandlung SDU --> PDU

1.) Dienstelement erzeugt Daten, sog. SDUs

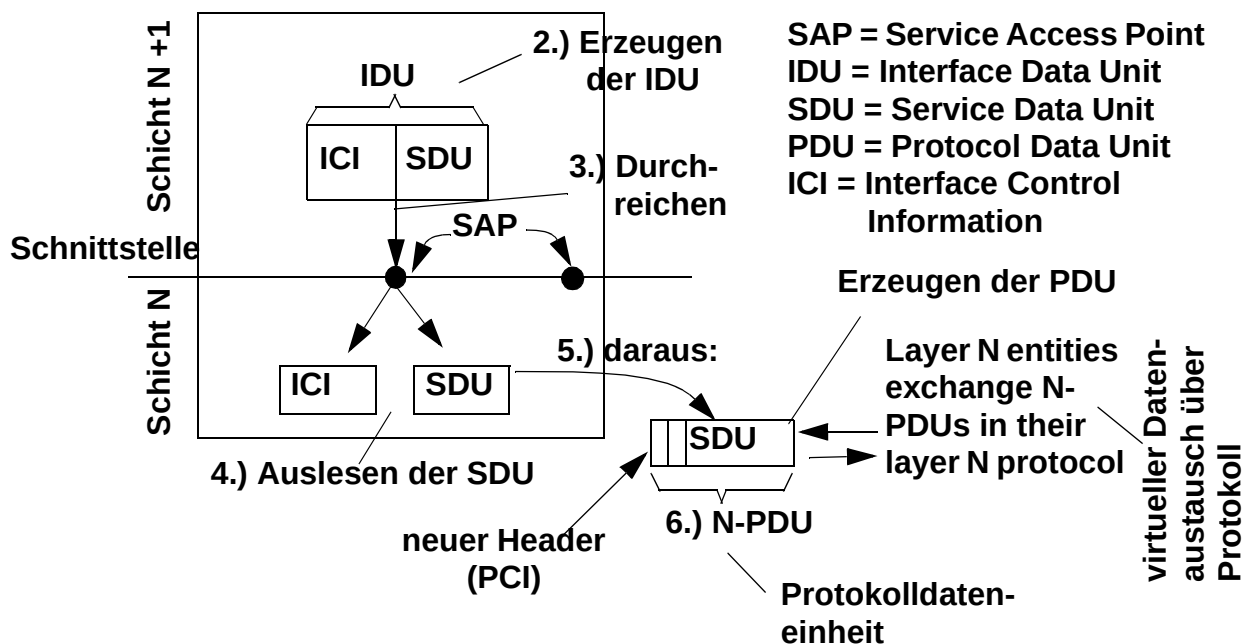


Bild: Beziehung zwischen zwei Schichten

Layer = Schicht

Interface = Schnittstelle

SAP = Dienstzugangspunkt

IDU = Schnittstellendateneinheit = das, was an der Schnittstelle zweier Schichten ausgetauscht wird. SDU = Dienstdateneinheit = der Datenteil der IDU, der vom Dienst N + 1 her kommt.

ICI = Steuerinformation für den Datenteil der IDU

PCI = Protocol Control Information

(N+1) - Dienstelement --> (N+1) - SDU --> (N+1) IDU --> (N) - SDU --> (N) - PDU

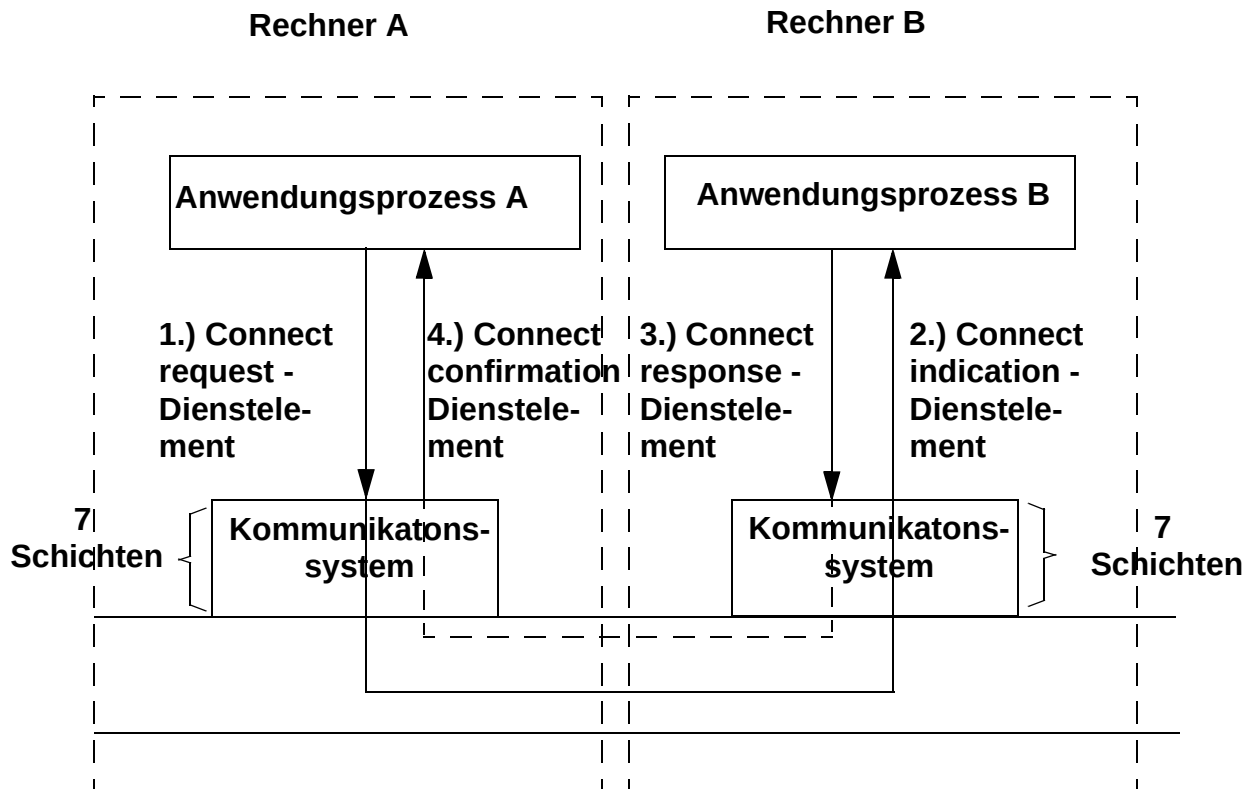
=> ICI - Header kommt wieder weg; PCI-Header kommt dazu

=> PDU wird länger

Die 3 Phasen der Kommunikation nach dem ISO/OSI-Modell

- Jede Verbindung, die in folgenden 3 Phasen abläuft, heißt verbindungsorientierte Kommunikation
 - 1. Phase: Verbindungsaufbau (Connect)
 - 2. Phase: Datentransfer (Datatransfer)
 - 3. Phase: Verbindungsabbau (Disconnect)
- Frage 1:
 - Wie hängen die 3 Phasen mit den 4 Dienstelementen zusammen?
- Frage 2:
 - Wie hängen die 4 Dienstelemente mit den Diensten einer Schicht zusammen?

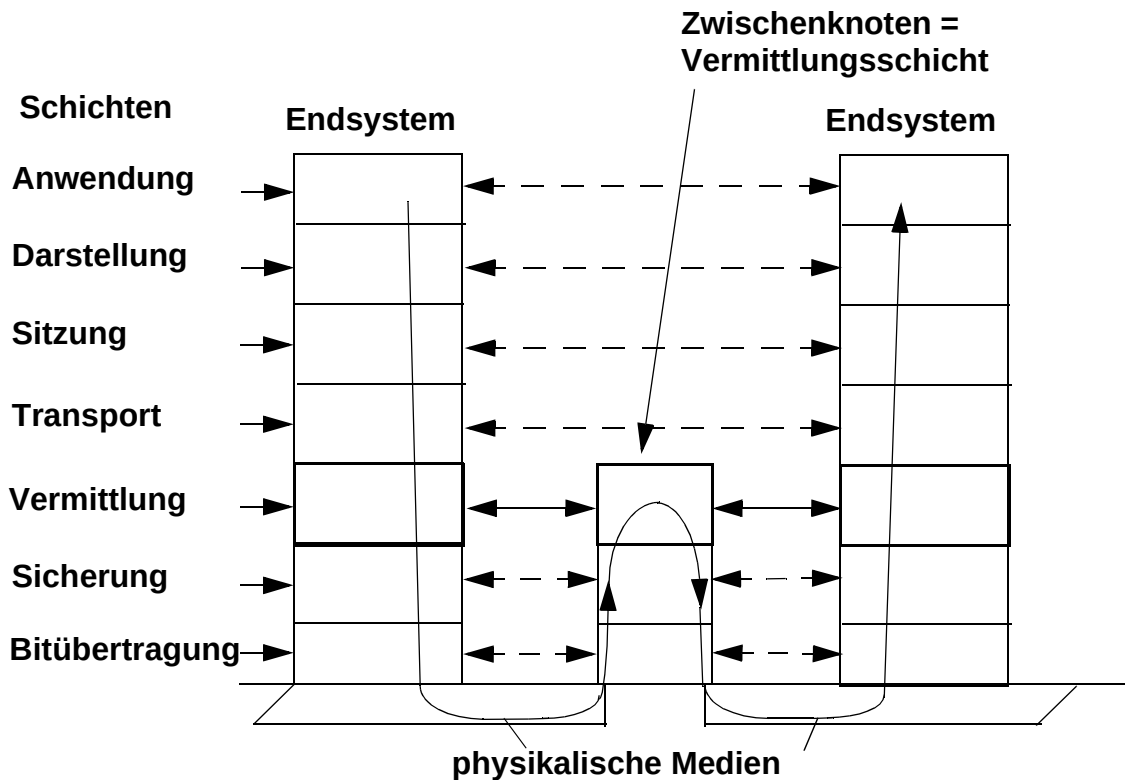
Beispiel: CONNECT (Verbindungsaufbau)-Dienst



- request, indication, response und confirmation sind die 4 Typen von Dienstelementen
 - In jeder der Schichten 2-7 gibt es diese 4 Typen von Dienstelementen (im Prinzip)
 - Für jede der 3 Phasen einer Kommunikation
 - Verbindungsaufbau (connect)
 - Datentransfer (Data)
 - Verbindungsabbau (Disconnect)
- gibt es diese 4 Typen von Dienstelementen**

=> $4 \cdot 6 \cdot 3 = 72$ Kombinationen (im Prinzip)

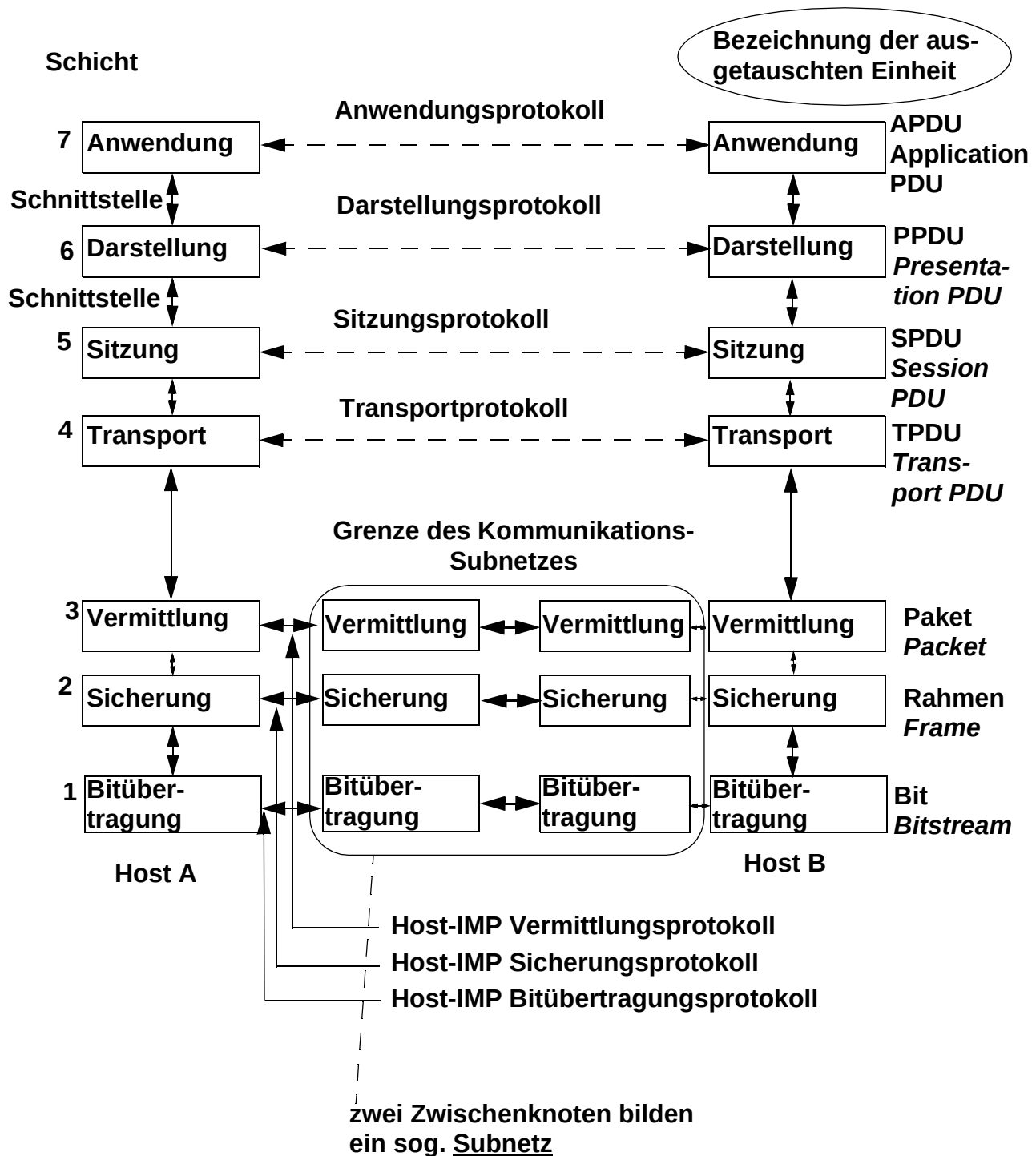
Konzept der Vermittlungssysteme als Zwischenknoten zwischen Endsystemen



=> Zwischenknoten hat nur die Schichten 1-3

- mehrere Zwischenknoten können als ein großer virtueller Zwischenknoten angesehen werden, da jeder Zwischenknoten die Daten unverfälscht weiterreichen muss

ISO-Referenzmodell mit mehr als einem Zwischenknoten



- Das Subnetz erscheint den Endsystemen wie ein Zwischenknoten

Bitübertragungsschicht

ISO-Definition

Die Bitübertragungsschicht (physical layer) definiert die mechanischen, elektrischen, funktionalen und prozeduralen Eigenschaften, um physikalische Verbindungen zwischen Datenendeinrichtungen (DEE; englisch: DTE) und Datenübertragungseinrichtungen (DÜE, englisch: DCE, „Poststeckdose“) aufzubauen, aufrecht zu erhalten und abzubauen.

Die Bitübertragungsschicht sorgt für die Übertragung eines transparenten Bitstroms zwischen Sicherungsschicht-Entites über physikalische Verbindungen. Eine physikalische Verbindung kann die Übertragung eines Bitstroms im Duplex-Mode oder im Halbduplex-Mode erlauben.

Duplex = bidirektionaler Datenaustausch

Simplex = unidirektionaler Datenaustausch

halbduplex = unidirektionaler Datenaustausch mit wechselnden Kommunikationsrichtungen

DEE = Datenendeinrichtungen = DTE = Data Terminal Equipment

**DÜE = Datenübertragungseinrichtung
= DCE = Data Communication Equipment**

Eigenschaften der Bitübertragungsschicht

mechanisch:

Abmessungen der Stecker, Anordnung der Pins, etc. z. B. „ISO 4903: Data Communication - 15 pin DTE/DCE interface connector and pin assignment“

elektrisch:

Spannungspegel auf Leitungen, etc. z. B. „CCITT X.27/V.11: Electrical characteristics for balanced double-current interchange for general use with integrated circuit equipment in the field of data communication“

funktional:

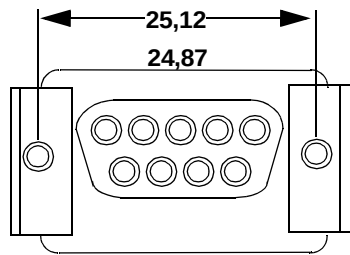
Klassifikation von Schnittstellenleitungsfunktionen (welcher Pin hat welche Funktion: data, control, timing, ground) z. B. „CCITT X.24: „List of definitions for interchange circuits between DTE and DCE on public data networks“

prozedural:

Regeln (Prozeduren) für die Benutzung der Schnittstellenleitungen, z. B. CCITT „X.21: Interface between DTE and DCE for synchronous operation on public data networks“

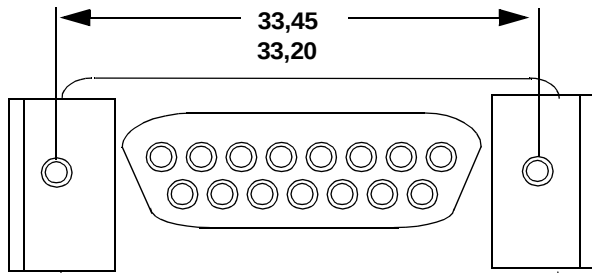
Und nicht zuletzt: Die Angabe der Bitrate, die zur Bitübertragung verwendet wird.

2.2 Mechanische, elektrische und funktionale Spezifikation



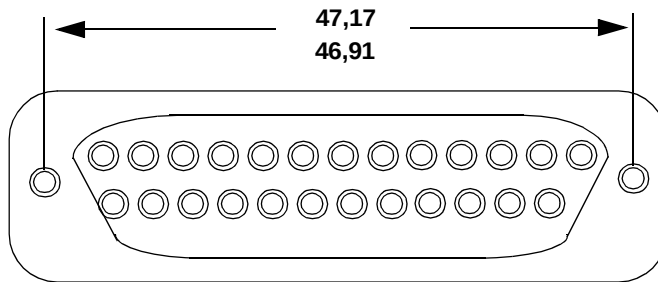
9 pin, ISO 4902, EIA RS-449

Beispiele für mechanische Spezifikationen



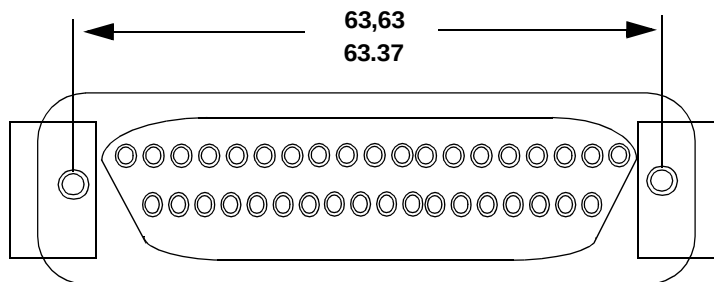
15 pin, X.20, X.21, X.22

X.21 = Norm für
synchrone Übertragung

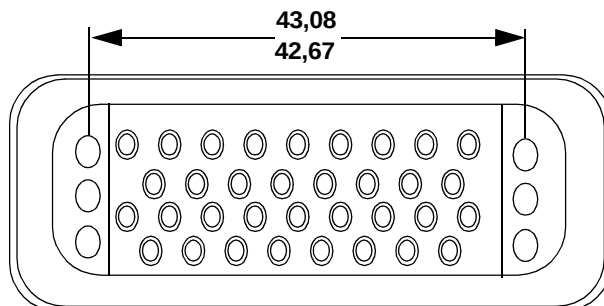


25 pin, ISO 2110, EIA RS-232-C, V.24

V.24= Norm für
asynchrone Übertragung



37 pin, ISO 4902, EIA RS-449 (modem)

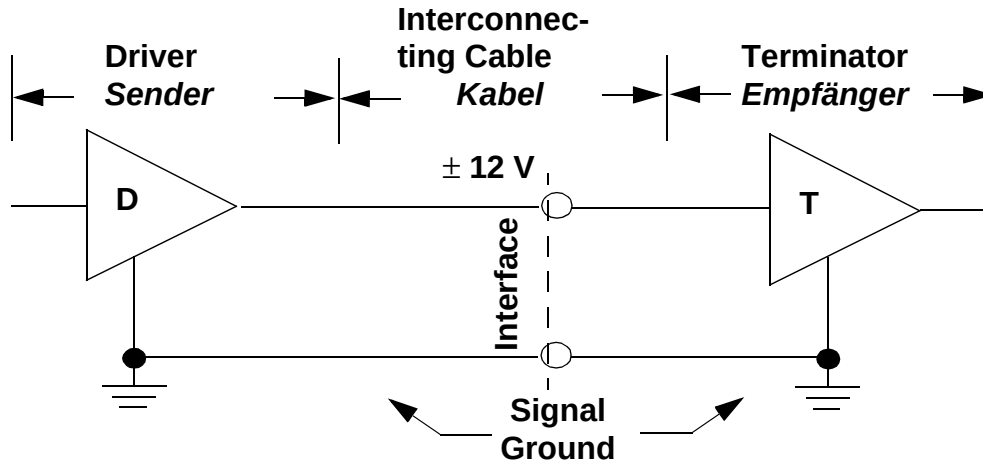


34 pin, V.35 (modem), ISO 2593



Beispiel für eine elektrische Spezifikation

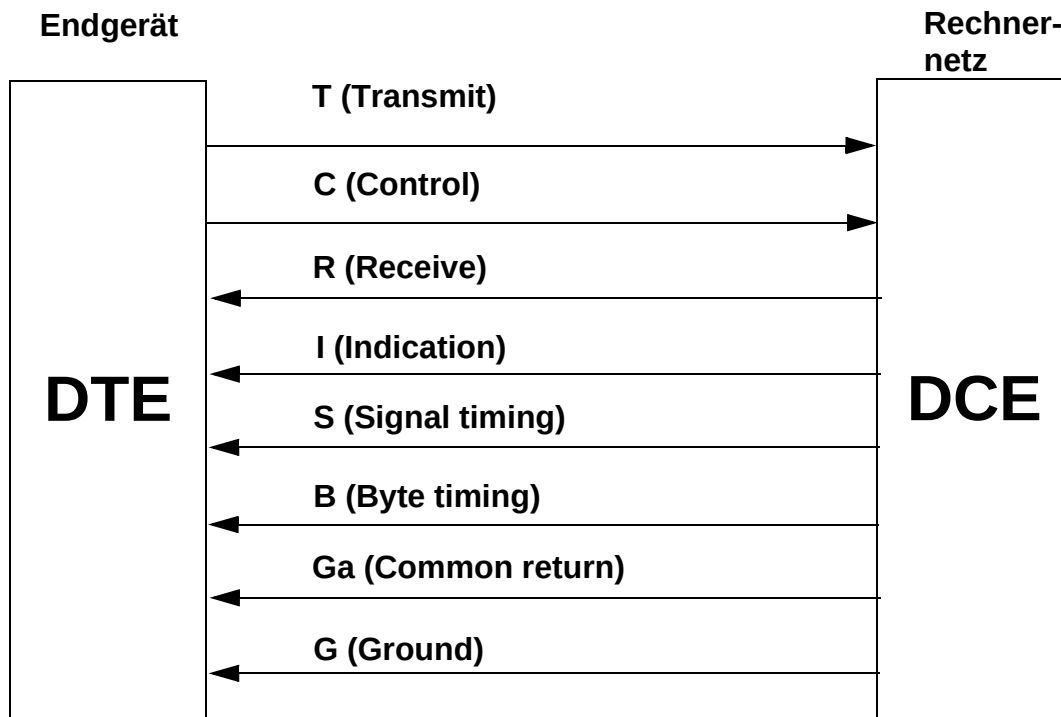
CCITT V.24 (EIA RS-232-C)
= asynchrone, serielle Übertragung



- ursprünglich für Sender/Empfänger gedacht, die aus diskreten elektronischen Bauelementen aufgebaut sind
- => einfache Struktur, geringe Datenraten
- Zwei Kabeladern pro Stromkreis, mit einer Erdung sowohl beim Sender als auch beim Empfänger
- Grundlegende Eigenschaften:
- Bitrate begrenzt auf 19,2 kbit/s
- Entfernung begrenzt auf 15 m
- Erzeugt erhebliches Übersprechen aufgrund der fehlenden Abschirmung zwischen den Adern im Kabel

Beispiel für die Spezifikation von Funktionalen und prozeduralen Eigenschaften

Signalleitungen bei X.21 (X.21 = Schnittstelle für synchrone Übertragungen)



DTE = Data Terminal Equipment = Datenendeinrichtung

DCE = Data Circuit Terminating Equipment = Datenübertragungseinrichtung

Funktionale/prozedurale Spezifikation in X.21

- Wird erläutert in Analogie zum Telefon

				Datenendeinrichtung (DTE)	Datenübertragungseinrichtung (DCE)
Schritt	C Control	I Indication	Ereignis analog zum Telefon	DTE sendet auf Leitung T	DCE sendet auf Leitung R
0	Aus	Aus	idle (Ruhezustand)	T=1 (= inaktiv)	R=1 (=inaktiv)
1	Ein	Aus	DTE nimmt Hörer ab —————→	T=0 (=Transmit aktiv)	
2	Ein	Aus	DCE sendet Wählton —————←		R="+++..+" = tüüt tüüt
3	Ein	Aus	DTE wählt Telefonnummer —————→	T=Adresse	
4	Ein	Aus	Entferntes Telefon klingelt —————←		R = Dienst
5	Ein	Ein	Entferntes Telefon abgehoben —————←		R=1 (noch inaktiv)
6	Ein	Ein	Gespräch (Datenaustausch) —————↔	T=Daten bidirektionaler Austausch	R = Daten Austausch
7	Aus	Ein	DTE verabschiedet sich —————→	T=0	
8	Aus	Aus	DCE verabschiedet sich —————←		R=0
9	Aus	Aus	DCE legt auf —————←		R=1 (wieder inaktiv)
10	Aus	Aus	DTE legt auf; -> idle —————→	T=1	

Asynchrone vs. synchrone Übertragung

Asynchron (Bsp.: V.24):

Kein explizites Taktsignal zwischen Sender und Empfänger vorhanden.

Synchron(Bsp.: X.21):

Taktinformation wird über die Leitung übertragen. Takt für die genaue Abstimmung der Bitimpulse (Synchronisation) auf beiden Seiten der Leitung benutzt.

VS. = versus (lat.) = gegen, gegenüber

Asynchrone Übertragung (1) (V.24)

- Sender und Empfänger besitzen voneinander unabhängige (lokale) Taktgeber
- „Freie Leitung“ (= ohne Daten) entspricht einem kontinuierlich gesendeten 1-Bit => Problem der Unterscheidung zwischen freie Leitung und 1-Pegel
- Das Start-Bit setzt die Leitung auf 0 und startet den Taktgeber des Empfängers
- Ein Rahmen mit 5 bis 8 Bits (= ein Zeichen) wird übertragen
- Das „Stop-Bit“ setzt die Leitung wieder auf 1. Dieses Signal muss 1, 1.5 oder 2 Bit-Intervalle andauern; entspricht 1, 1.5 oder 2 Stop-Bits

Asynchrone Übertragung (2)

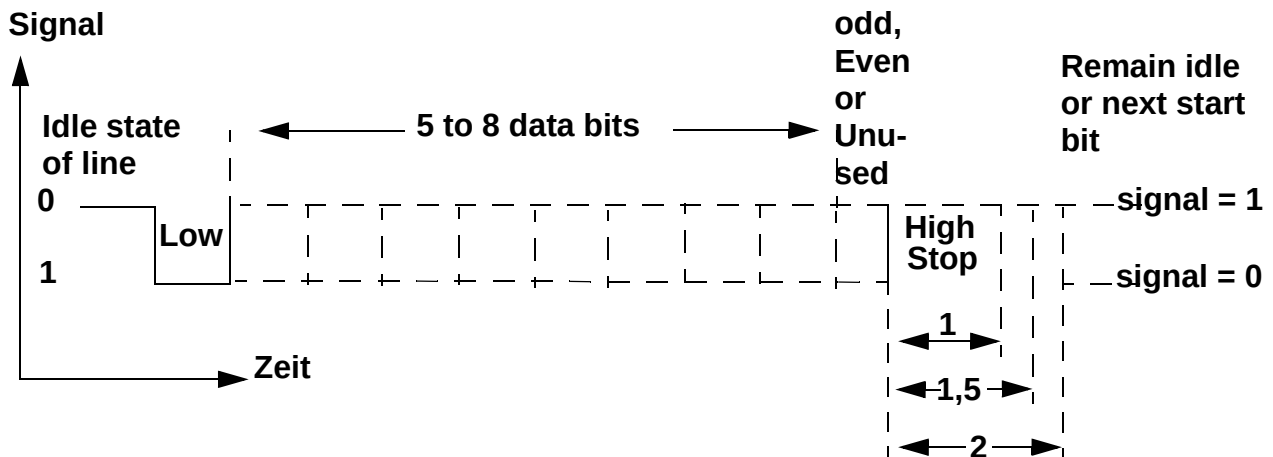
Vorteile

- **Es wird keine Synchronisierung der Taktgeber in den Endsystemen benötigt**
- **Leicht zu implementieren**

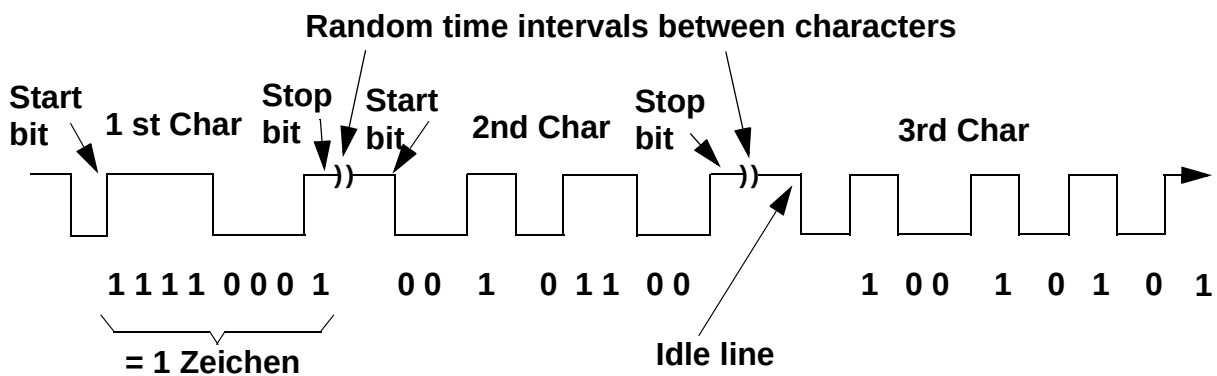
Nachteile

- **Die Taktgeber können voneinander abweichen. Daher**
 - **ist die Rahmengröße sehr beschränkt (typischerweise auf ein Zeichen : entspricht 7-8 Bits)**
 - **nur anwendbar bei niedrigen Datenraten (<20 Mb/s)**
 - **Die Start- und Stop-Bits stellen einen Mehraufwand (Overhead) dar.**
 - **Beispiel:**
 - **7-Bit ASCII**
 - **1 Paritätsbit,**
 - **1 Start-Bit,**
 - **1 Stop-Bit**
 - **= insgesamt 10 Bits**
- => Nur 70% der Leitungskapazität stehen in diesem Beispiel für echte Benutzerdaten zur Verfügung.**

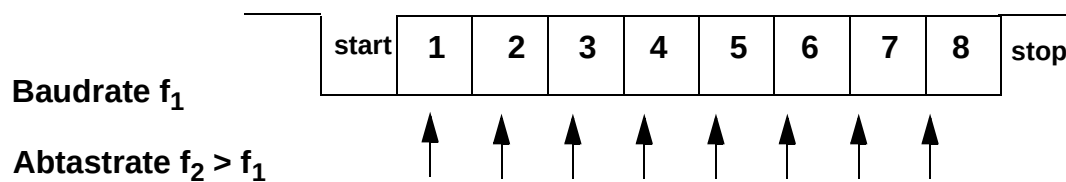
Asynchrone Übertragung (3)



(a) Leitungscode für ein Zeichen



(b) asynchroner Bitstrom



(c) Effekt der auseinander laufenden Uhren

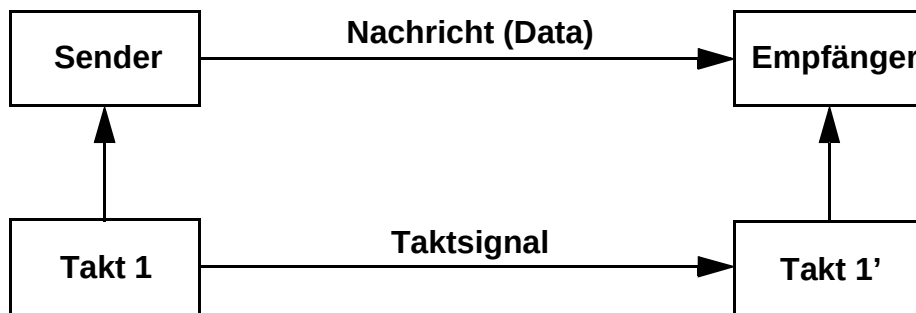
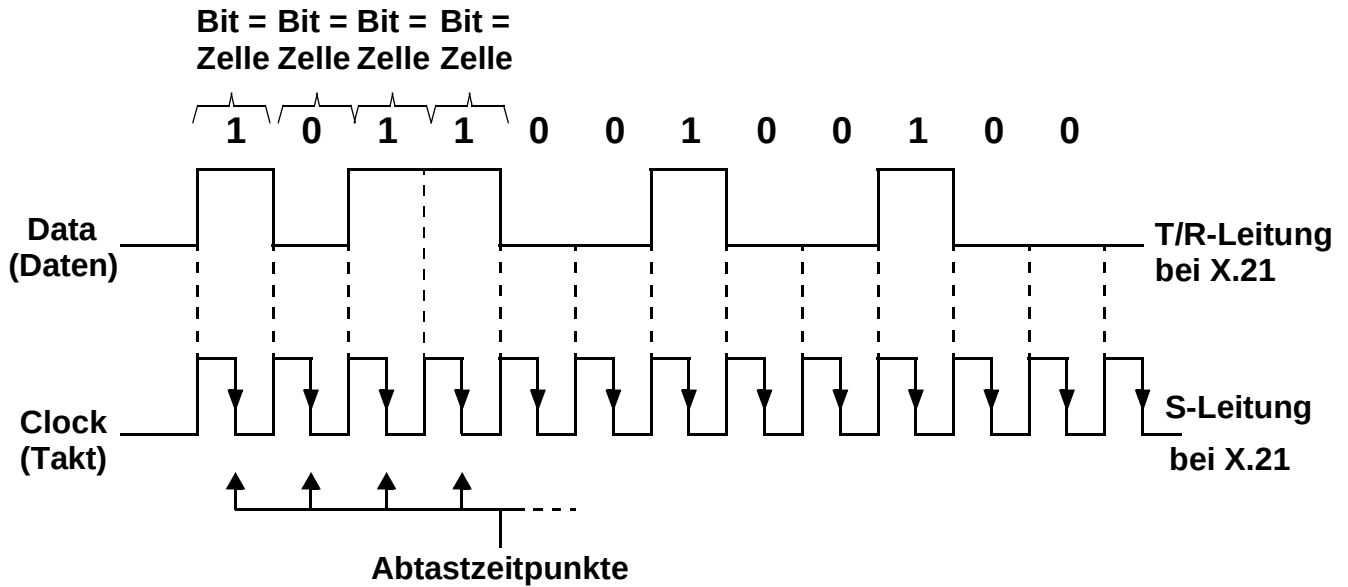
Synchrone Übertragung (1) (X.21)

- Sende- und Empfangstakt laufen über einen längeren Zeitraum (beliebig lange) synchron.
- Eine Neusynchronisation nach jedem Zeichen (5-8 Bits) ist nicht erforderlich.

Taktsignal

Wird entweder auf einer separaten Leitung übertragen (z. B. bei X.21 vom Dienstanbieter) oder aus dem Leitungssignal gewonnen (z. B. in den Modems, die an Zweidrahtleitungen angeschlossen sind). Letzteres erfordert spezielle Codierung der Daten, z. B. Manchester-Code.

Synchrone Übertragung (2)



Beispiel:

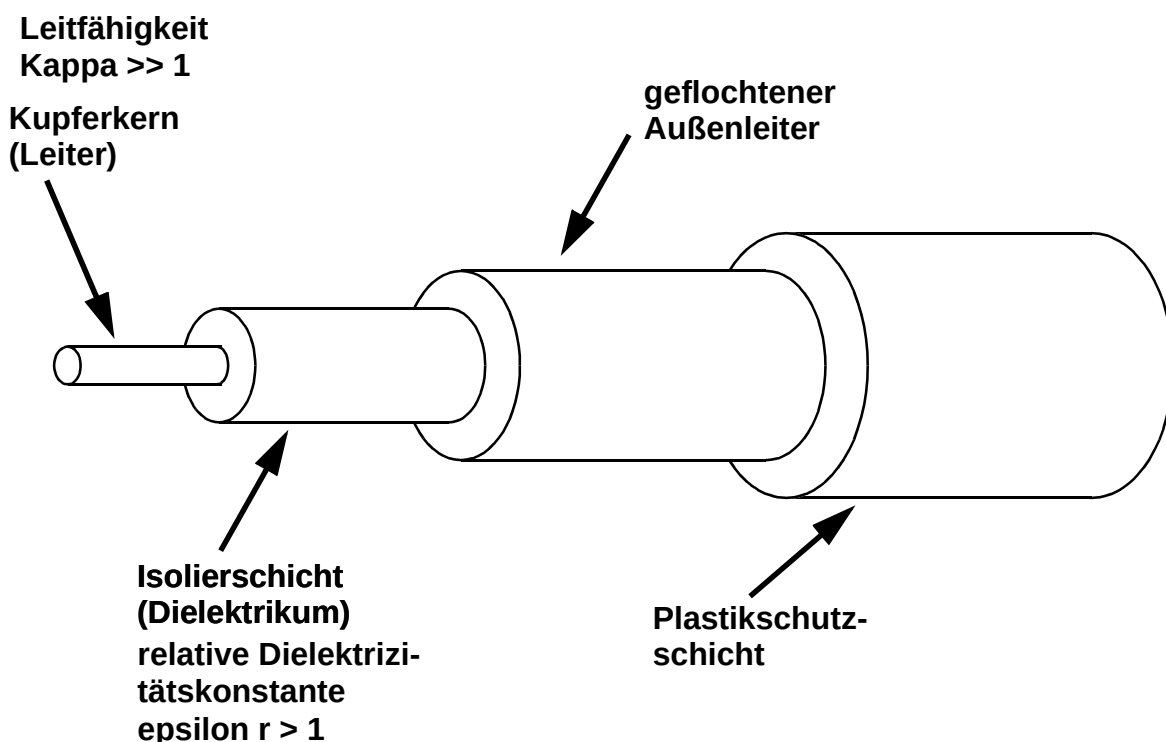
Auslesen des Datenbits bei abfallender Flanke des Taktsignals

Physikalische Medien zur Datenübertragung

- **Adernpaar (verdrillt zur Verminderung von Störeinflüssen, deshalb „twisted pair“)**. Dies ist die klassische **Telefonverkabelung**. Braucht wenig Platz, enge Bieungsradien, sehr preiswert
- **Adernpaar, abgeschirmt („shielded twisted pair“ = Twinax-Kabel)**. Weniger anfällig bzgl. der Einkoppelung von Störfrequenzen/Störströmen von außen. Ist dieser als „unshielded twisted pair“, teurer.
- **Koaxialkabel**: sehr störsicher, ermöglicht sehr hohe Übertragungsraten. Teuer. Weit verbreitet zum Beispiel für die **Ethernet-Verkabelung** (nur 10 Mb/s-Ethernet).
- **Lichtwellenleiter (Glasfaser)**: sehr hohe Übertragungsraten, geringe Dämpfungen und Störeinkopplungen, aber aufwendige Anschlusstechnik, teuer.

Koaxialkabel für Thick-Ethernet (10 Mb/s)

- 50 Ohm-Koaxialkabel => Wellenwiderstand = 50 Ohm
- Maximale Kabellänge 500 m pro Segment
- Maximal 100 Transceiver (= Anschlüsse von Stationen) pro Segment
- Maximal vier Repeater(= Verstärker) zwischen Sender und Empfänger
- Abstand zwischen den Anschlüssen muss ein Vielfaches von 2,5 m sein, wegen Ausbildung von stehender Wellen auf dem Kabel
- Datenrate 10 Mbit/s



Glasfaserkabel

- **Sehr hohe Datenraten:**

- Theoretisches Limit: 300 TeraHz = Frequenz des Lichts
- Praktisches Limit: = 10 GigaHz (zur Zeit)

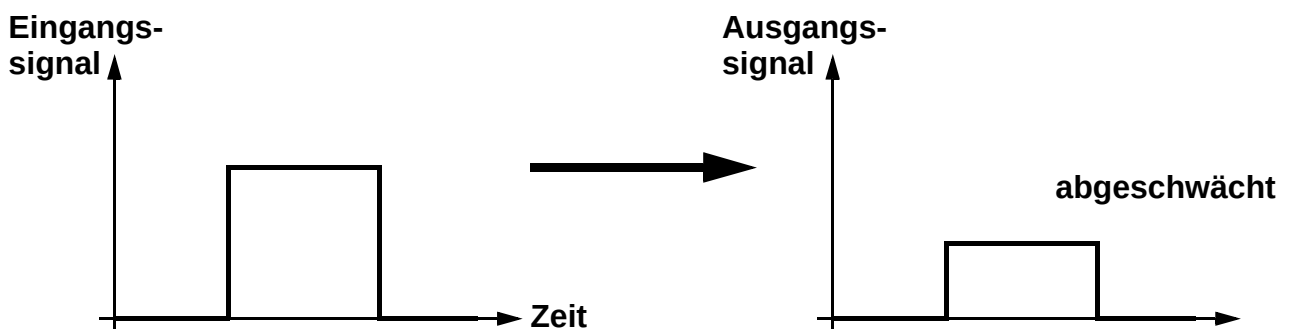
- **Transmitter und Receiver: Halbleiterelemente (Dioden)**

- Laserdioden
- Leuchtdioden
- Photodiode

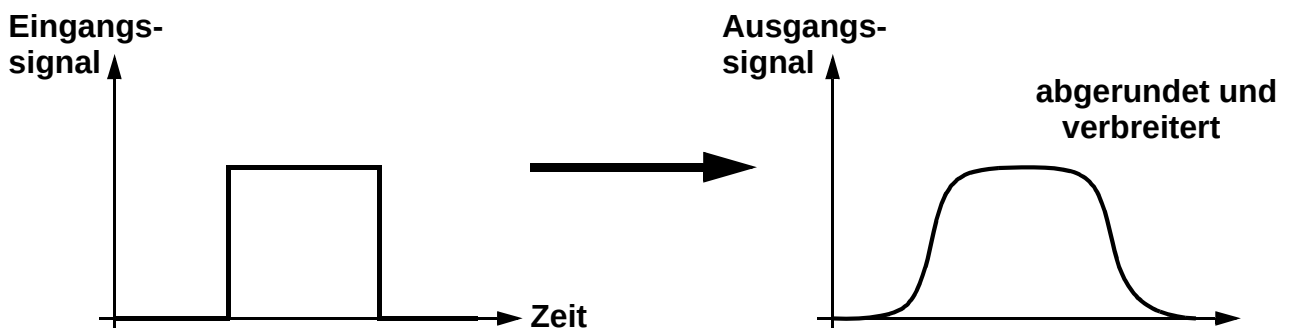
- **Beschränkende Faktoren:**

-

1. **Absorption in der Faser:**



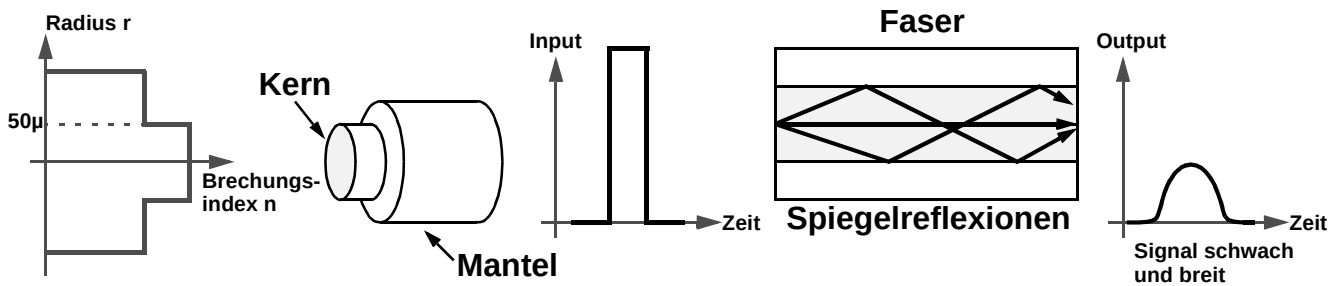
2. **Dispersion in der Faser:**



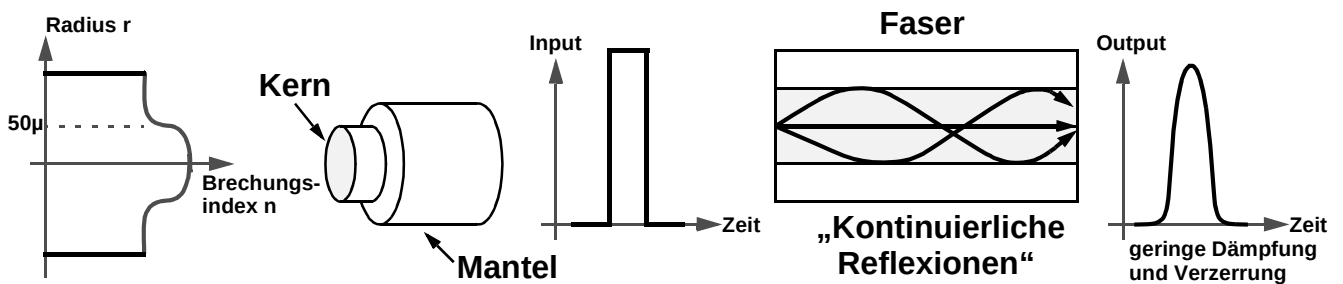
- **Verbindungstechnik schwierig. Nur 5 μm - 50 μm Durchmesser im Kern, wo das Licht geleitet wird.**

Technologie der Lichtwellenleiter

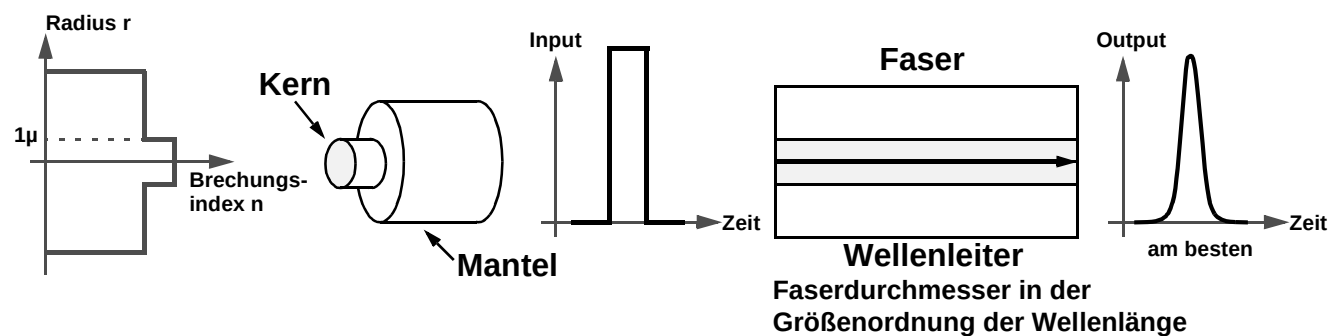
• Stufenindex-Faser



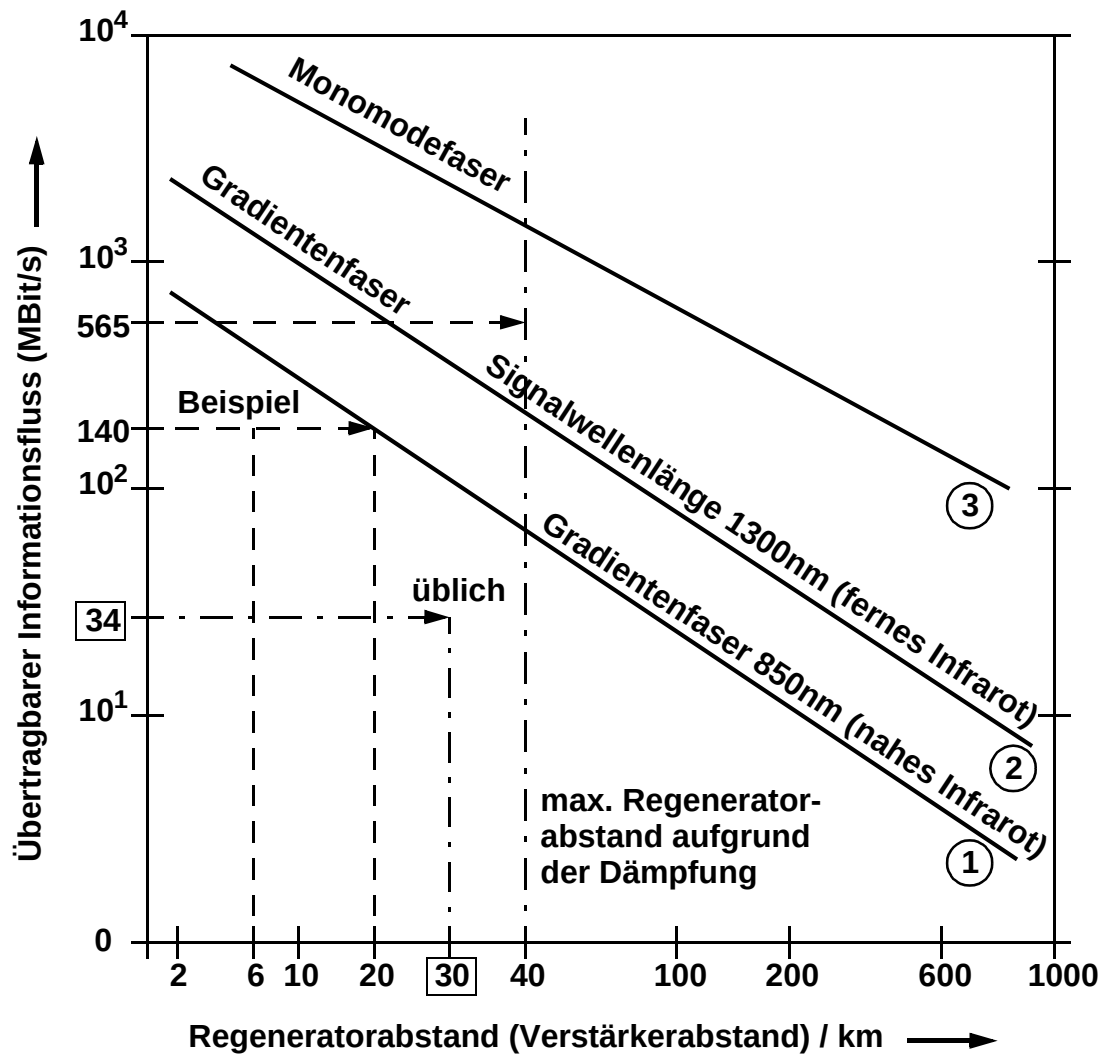
• Gradientenindex-Faser



• Monomode-Faser



Glasfaser



Systemparameter:

- ① Gradientenfaser, LED als Sender, 850nm Wellenlänge
- ② Gradientenfaser, Laser als Sender, 1300nm Wellenlänge
- ③ Monomodefaser, Einmode-Laser als Sender, 1300nm Wellenlänge

Allgemeine Ziele der Verkabelung

- **Vereinheitlichung der Kabeltypen (möglichst wenige)**
- **Weniger Kabel insgesamt (Kostenfaktor!)**
- **Flexibilität (Biegsamkeit) des Kabels**
- **Einsatz bekannter Techniken (Löten, Klemmen, Schrauben)**
- **Vorverkabelung (Versorgungsdichte à la Telefon)**
- **Zukunftssicherheit**
- **Erweiterungsmöglichkeiten**
- **Störungssicherheit (elektromagnetische Impulse)**
- **Übertragung mit höheren Geschwindigkeiten**

Situation in der Gebäudeverkabelung

- **Unterschiedliche Kabeltypen vorhanden**

- Koaxialkabel (= Innen- und Außenleiter als konzentrische Ringe)
- Twinaxialkabel (= verdrehtes Adernpaar mit äußerer Schirmung)
- Verdrehte Leiter (= wie beim Telefon, twisted pair)
- Mehrfach verdrehte Leiter (= sog. Verseilung)

=> Geringe Flexibilität (Biegsamkeit) des Kabels

=> heterogene Umgebung im Gebäude

- **Unterschiedliche Verkabelungstechniken vorhanden**

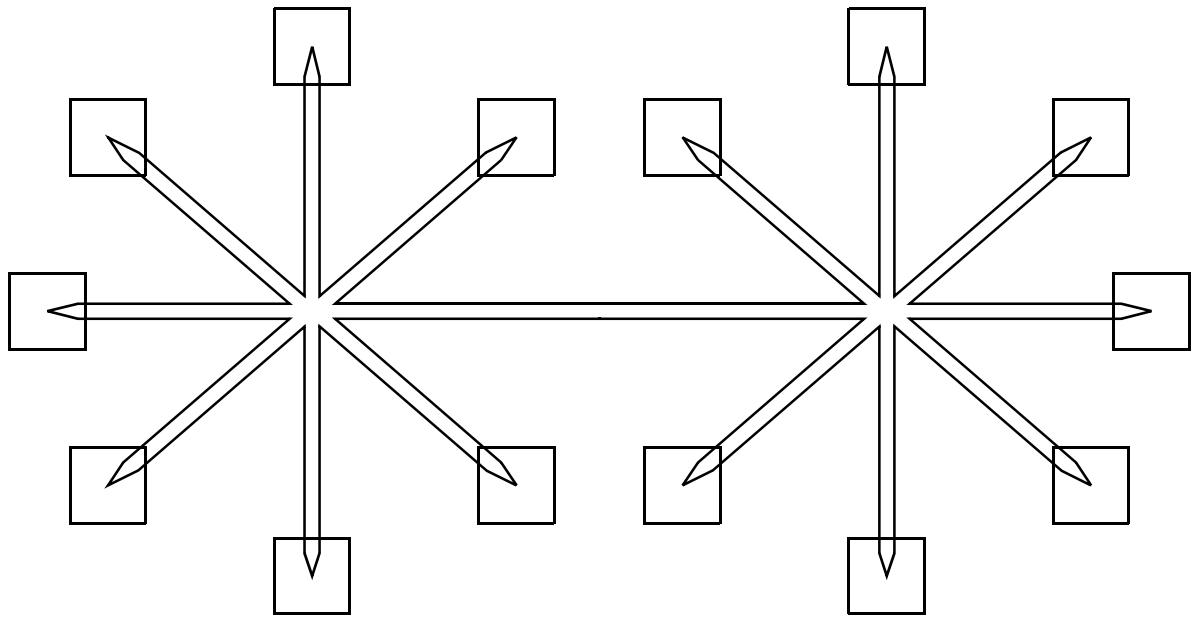
=> Hoher Verkabelungsaufwand, da gleichzeitig verschiedene Techniken nötig

- **Hohe Zuwachsraten**

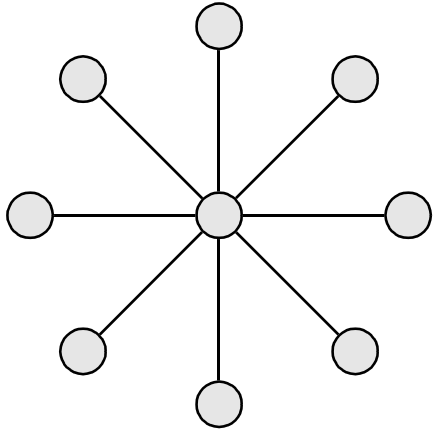
=> Großer Änderungsaufwand

Strukturierte Verkabelung

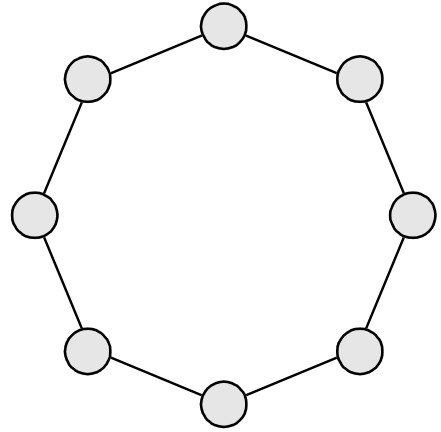
- Möglichkeiten einer Ringverkabelung



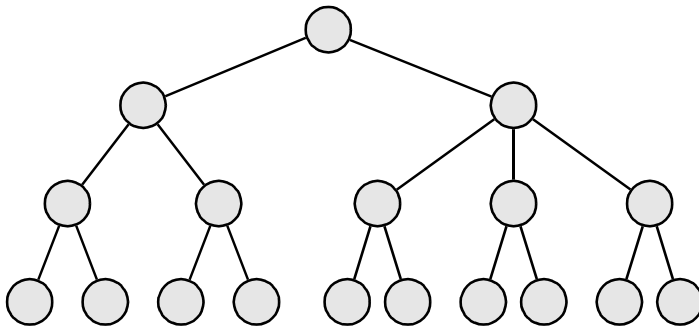
Netztopologien



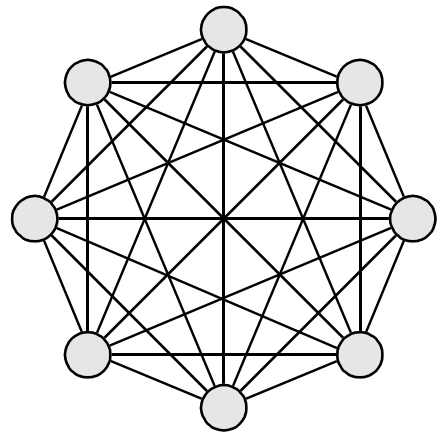
Stern



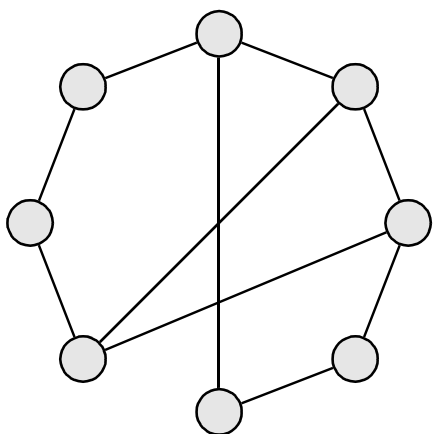
Ring



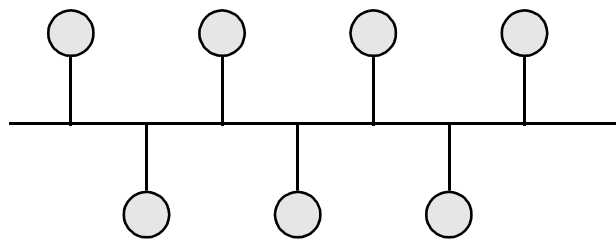
Baum



vollständig vermaschtes Netz



teilweise vermaschtes Netz



Bus

Moderne Basisbandverfahren

Moderne digitale Übertragungstechnik verwendet Basisbandverfahren bis zu sehr hohen Bitraten im Multi-Mega-bit/s-Bereich (PCM-Technik, lokale Netze, ISDN usw.). Dabei erwünscht bzw. erforderlich:

- kein Gleichstromanteil im Signal
- Wiedergewinnung des Takts aus der ankommenden Signalfolge (selbsttaktende Signalcodes)
- Erkennen von Übertragungsfehlern bereits auf der Signalebene

Signalcodierung, Leitungscodierung, Übertragungscode

Die Zuordnungsvorschrift

digitales Datenelement -> digitales Signalelement

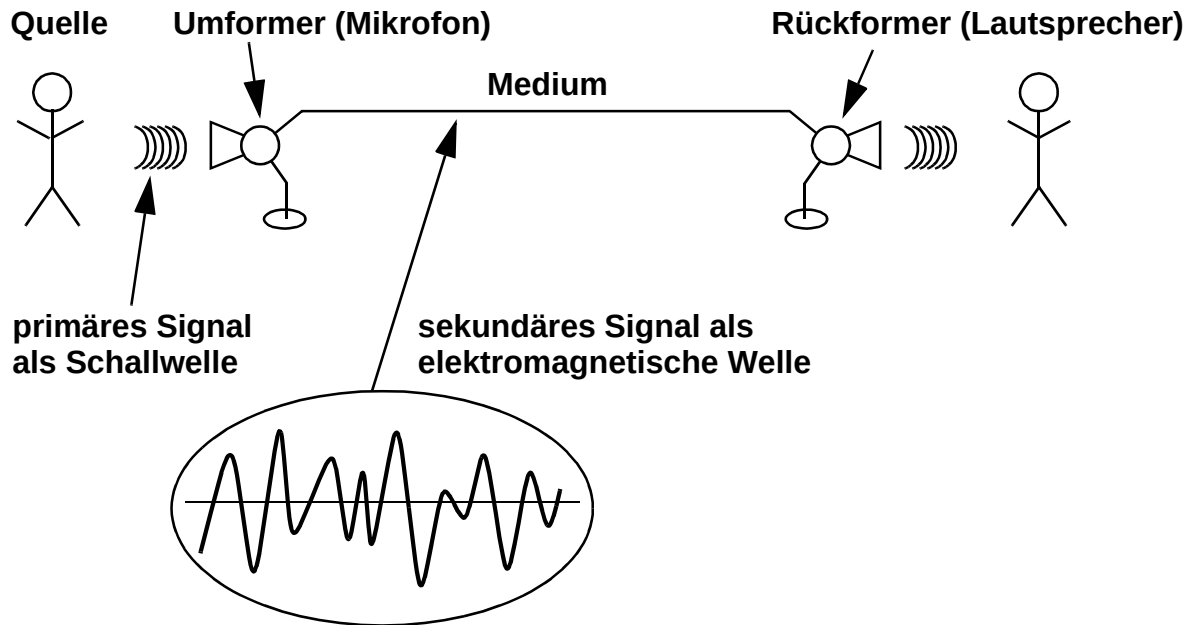
= digitale Eingabe

= digitale Übertragung

wird als Signal- oder Leitungscodierung bezeichnet. Die sich ergebenden zeit- und wertdiskreten Signalverläufe heißen Leitungscodes oder Übertragungscodes. Basisband heißt, dass keine Modulation eines Trägersignals mit den Daten stattfindet, sondern dass die Daten unmoduliert übertragen werden. Sie können deswegen trotzdem codiert werden.

2.3 Übertragungstechniken, Modulation, Multiplexing

- Beispiel: Analoge Signalübertragung beim Telefon:



Das Primärsignal (hier akustisch) wird durch Umformer in ein elektrisches (hier analoges) Signal umgewandelt und durch Rückformer zurückgewandelt. Im weiteren gehen wir jedoch davon aus, dass bereits das quellenseitige Primärsignal in elektrischer Form vorliegt und das senkenseitige Primärsignal wieder ein elektrisches Signal ist. Das Übertragungssignal kann ebenfalls elektrisch sein, mit gleichem oder anderem Verlauf als das Primärsignal, aber auch beispielsweise optisch.

Signale

Ein Signal ist eine physikalische Repräsentation von Daten (Information)

Signalparameter sind diejenigen physikalischen Kenngrößen eines Signals, deren Wert oder Werteverlauf die Daten repräsentieren.

Bei räumlichen Signalen sind die Werte des Signalparameters S Funktionen des Ortes:

$$S = S(x, y)$$

Bei zeitabhängigen Signalen sind die Werte des Signalparameters S Funktionen der Zeit:

$$S = S(t)$$

Einteilung zeitabhängiger Signale in Klassen:

- 1.) zeitkontinuierlich, wertkontinuierliche Signale
- 2.) zeitdiskrete, wertkontinuierliche Signale
- 3.) zeitkontinuierliche, wertdiskrete Signale
- 4.) zeitdiskrete, wertdiskrete Signale

Ist zu jedem Zeitpunkt ein Signalwert vorhanden?

ja: zeitkontinuierlich

nein: zeitdiskret

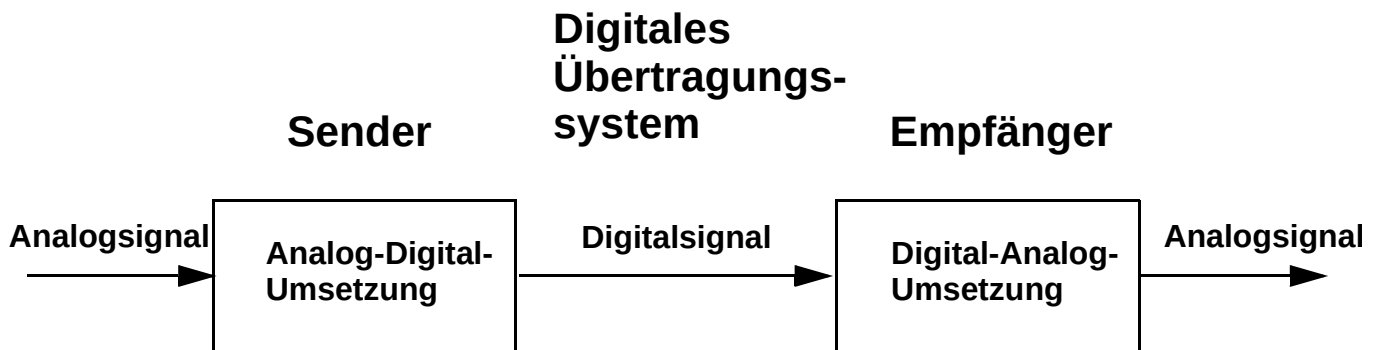
Sind alle Signalwerte im Wertebereich zulässig?

ja: wertkontinuierlich

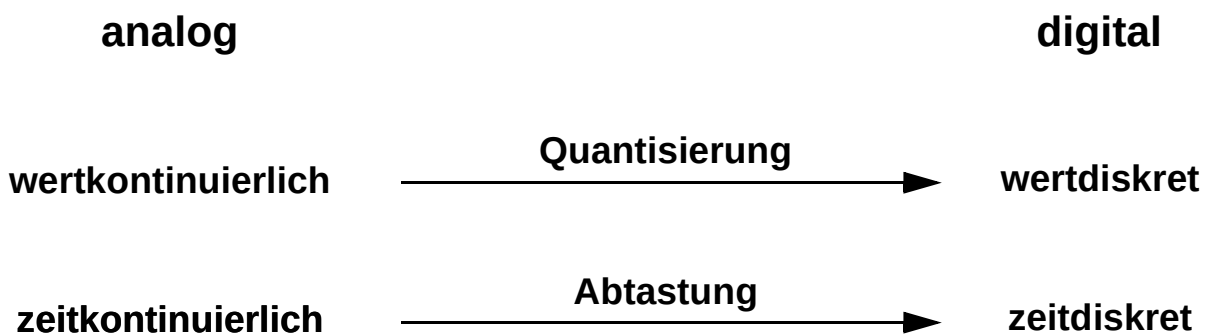
nein: wertdiskret

Digitale Übertragung analoger Daten

Die Übertragung analoger über digitale Übertragungswege erfordert eine Digitalisierung der analogen Daten.

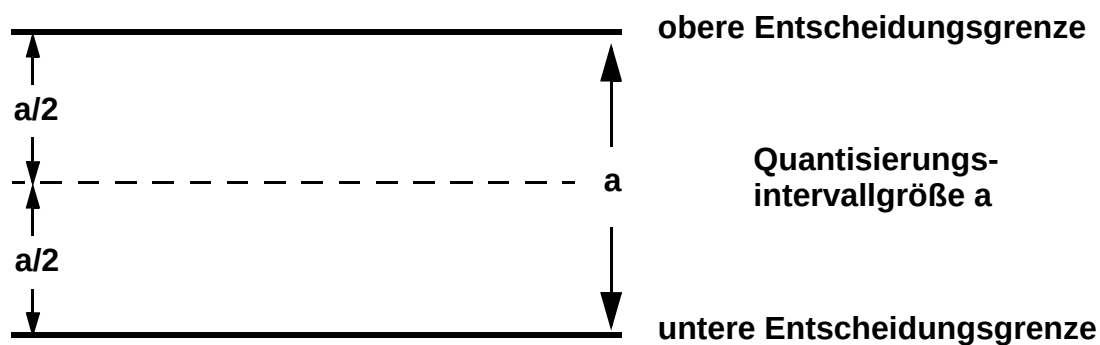


A/D- und D/A-Umsetzung zur Übertragung analoger Signale auf digitalen Übertragungssystemen.



Quantisierung

Der gesamte Wertebereich des Analogsignals wird in eine endliche Anzahl von Intervallen (Quantisierungsintervalle) eingeteilt, denen jeweils ein fester diskreter Wert zugeordnet wird. Da alle in ein Quantisierungsintervall fallenden Analogwerte demselben diskreten Wert zugeordnet werden, entsteht ein Quantisierungsfehler.



Rückwandlung

Beim Empfänger wird ein Analogwert rückgewonnen (Digital-Analog-Umsetzung), der dem in der Mitte des Quantisierungsintervalls liegenden Analogwert entspricht.

Maximaler Quantisierungsfehler: $a/2$

Abtastung

Für die Zeitdiskretisierung muss eine Abtastung der Analogverläufe erfolgen. Praktisch wichtig ist vor allem die periodische Abtastung.

Der zum Abtastzeitpunkt vorliegende Momentan-Wert des Analogsignals wird der Analog-Digital-Umsetzung unterworfen.

Abtasttheorem von Shannon und Raabe (1939)

Zur fehlerfreien Rekonstruktion des Signalverlaufs des Analogsignals ist eine Mindestabtastfrequenz f_A erforderlich (bei periodischem Abtastzyklus). Diese hängt von der höchsten im analogen Signal vorkommenden Frequenz ab. Für rauschfreie Kanäle gilt das folgende

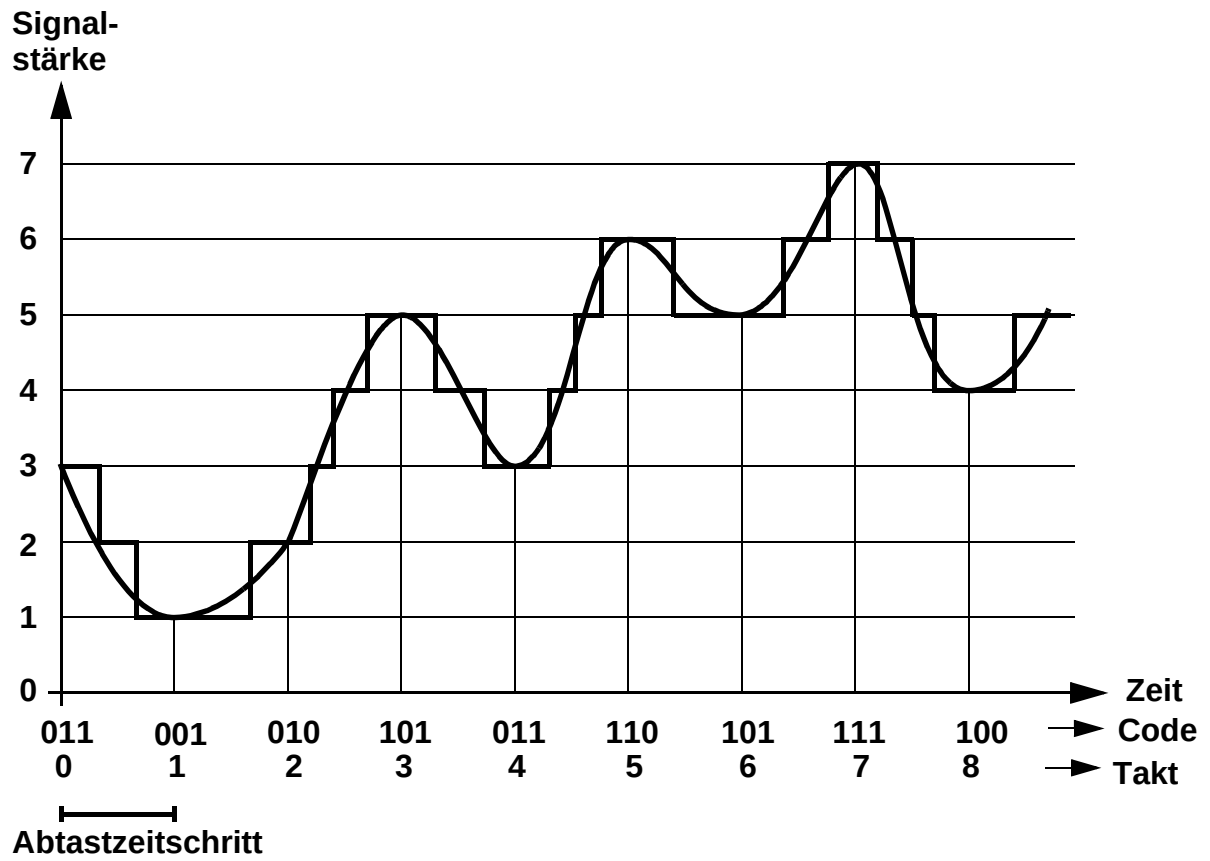
Abtasttheorem

Die Abtastfrequenz f_A muss mindestens doppelt so hoch sein wie die höchste im abzutastenden Signal vorkommende Frequenz f_S :

$$f_A = 2 f_S$$

Abtastung und Quantisierung sind voneinander unabhängig zu betrachten. Eine exakte Rekonstruktion des Zeitverlaufs (bzw. des Frequenzspektrums) sagt nichts über den Fehlergrad bei der Signalwertdiskretisierung (Quantisierung) aus.

Beispiel für Abtastung und Quantisierung



0 = 000

1 = 001

2 = 010

3 = 011

4 = 100

5 = 101

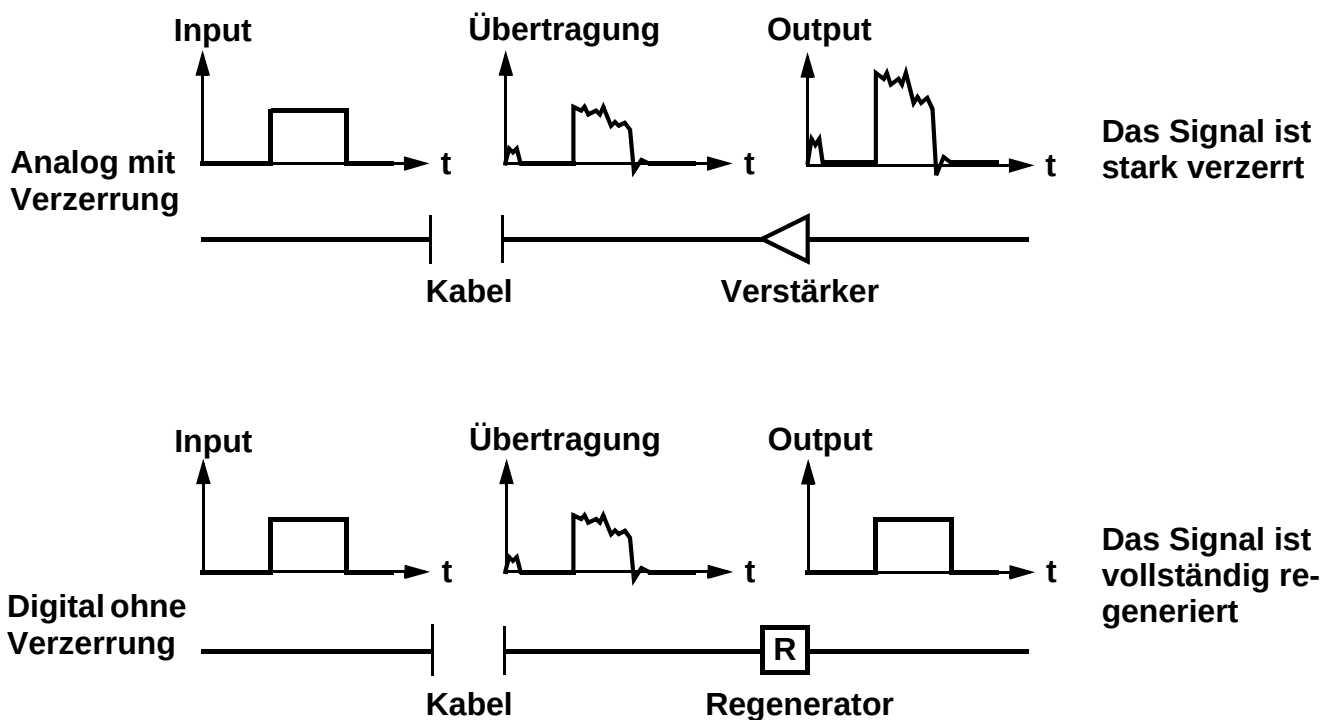
6 = 110

7 = 111

Grundlegende Übertragungstechniken

- Digitale Eingabe (= Input) + digitale Übertragung der Eingabe = Digitale Leitungscodierung
- Digitale oder analoge Eingabe (= input) + analoge Übertragung = Modulationstechniken
- Analoge Eingabe (= input) + digitale Übertragung = Digitalisierung (Abtastung)

Vergleich Analoge und digitale Übertragung



Vorteile der digitalen Übertragung

- **Niedrige Fehlerrate**
 - kein durch Verstärker induziertes Rauschen
 - keine Akkumulation des Rauschens über lange Distanzen
- **Time Division Multiplexing (TDM) leichter**
- **Digitale Schaltungen sind billiger**

Als Folge setzt sich heute die digitale Speicherung und Übertragung von eigentlich analogen Signalen immer mehr durch:

- **Audio-CD**
- **DAB (Digital Audio Broadcast) = Digitaler Rundfunk**
- **Digitales Fernsehen mit MPEG-Codierung**

Digitale/analoge Daten über analoge Signale = Modulation

Modulation: verschlüsselte Quelldaten auf ein Trägersignal (wie beim Rundfunk für analoge Daten)

Für digitale Daten:

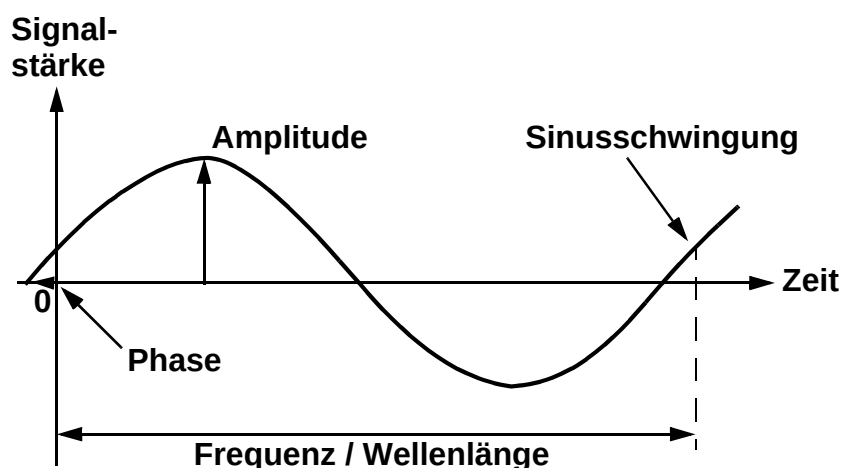
Modem = Modulator

Demodulator (Modulator dient zum Senden; Demodulator dient zum Empfangen)

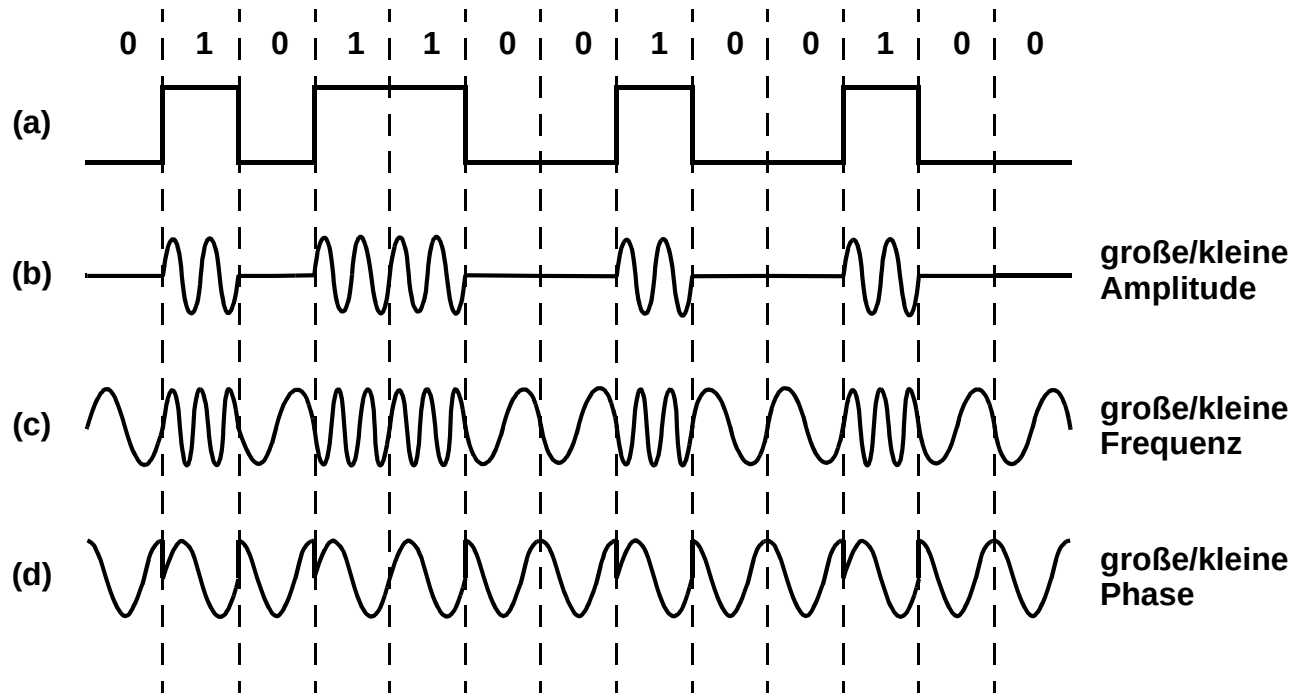
Beispiel: Übertragung von digitalen Daten über das Telefonnetz mittels Modem

Modulationstechniken

- 1.) Amplitude Modulation (AM), Amplitude Shift Keying (ASK)
- 2.) Frequency Modulation (FM) Frequency Shift Keying (FAK)
- 3.) Phase Modulation (PM) Phase Shift Keying (PSK)



Modulationstechniken



- (a) Binärsignal (original)
- (b) Amplitudenmodulation (AM)
- (c) Frequenzmodulation (FM)
- (d) Phasenmodulation (PM)

Codierung

Die quantisierten Werte werden durch die Zuordnung eines - frei wählbaren - (Binär-) Codes gekennzeichnet. Anstelle des ursprünglichen Analogsignals wird der digitale Code-wert übertragen.

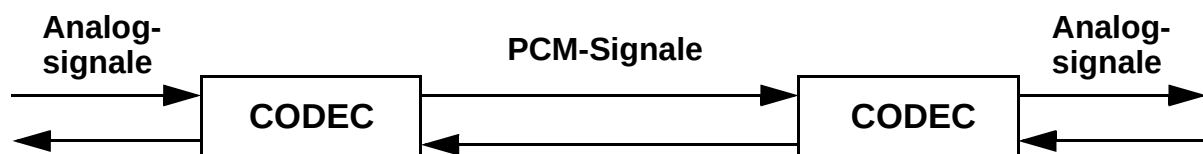
Im einfachsten Fall wird ein reiner Binärcode als Codierung des diskreten Digitalwertes gewählt (Darstellung als Binärzahl). Pulse-Code-Codierung (PCM)

Die Zusammenfassung der Schritte

Abtastung
Quantisierung
Codierung

und die Darstellung der gewonnenen Codewörter als digitale Basisbandsignale am Ausgang des PCM-A/D-Umsetzers ist Grundlage der in großem Umfang eingesetzten PCM-Technik.

Die A/D-Umsetzung (Abtastung/Quantisierung) und Codierung sowie die Rückkonvertierung erfolgt im sogenannten CODEC (Codierer/Decodierer).



PCM-Systeme

Die praktische Gestaltung technische PCM-Systeme wurden insbesondere durch die Telefonie beeinflusst, obwohl grundsätzlich jede Art analoger - nach Digitalisierung - und digitaler Daten unter Verwendung digitaler PCM-Systeme übertragbar ist.

Praktisch eingesetzte PCM-Systeme kombinieren sehr häufig eine PCM-Codierung der Einzelkanäle mit einem Zeitmultiplexing auf dem Übertragungsweg. Beispiel: Telefonkanäle auf Glasfaserkabeln.

Inzwischen spielt PCM auch im Bereich der digitalen Heimelektronik (Radio, CD, Video-Camcorder) eine zunehmende Rolle.

Leitungscodierung digitaler Signale (1)

NRZ (Non Return to Zero) = ganze Gruppe von Codes

Das gemeinsame Kennzeichen der NRZ-Leitungscodes ist ein fester Pegel während eines Bitintervalls. Signal (Signalwechsel) erfolgen an den Intervallgrenzen. NRZ ist sehr einfach zu implementieren.

NRZ - L (Level) = 1 spezieller Code

Zuordnung eines festen Pegels zu einem binären Datenelement:

„1“ = hoher Pegel

„0“ = niedriger Pegel

NRZ-L ist Standard innerhalb von Digitalgeräten (Rechnern, usw.)

- non return to zero = der Signalpegel für eine „1“ kehrt innerhalb der Bitzelle nicht auf 0 Volt zurück, sondern bleibt bei High-Pegel.
- Bitintervall = Bitzelle

Leitungscodierung digitaler Signale (2)

- NRZ-M (Mark), NRZ-S (Space)

Differenzielle Codierung: Es wird nicht der absolute Signalwert eines Signalelements in der Zuordnungsvorschrift verwendet, sondern der Signalwert in Abhängigkeit von der Polarität des vorhergehenden Signalelements.

- NRZ-M: Signalwechsel (=Pegelwechsel; Übergang in den entgegengesetzten Signalwert) zur Darstellung des Datenelements „1“.
- NRZ-S: Signalwechsel (= Pegelwechsel) bei Datenelement „0“.
- Vorteile gegenüber NRZ-L: Unter Einfluss von Störungen (Rauschen) sind Signalwechsel leichter zu detektieren als Signalpegel, die mit einem Schwellwert verglichen werden müssen.
- Nachteile aller NRZ-Codes: Die Gleichstromkomponente und Synchronisierungsprobleme zwischen Sender und Empfänger (z. B. bei langen „0“-Folgen bei NRZ-L und NRZ-M)

Leitungscodierung digitaler Signale (3)

- Return to Zero = ein spezieller Code
Gekennzeichnet durch einen Rechteckimpuls in der ersten Hälfte des Bitintervalls für das Datenelement „1“. Danach Rückkehr in einen Grundzustand (Zero). Die Baudrate (Rate der Rechteckimpulse) ist im Extremfall (Folge von Einsen) doppelt so hoch wie die Bitrate. Keine Vorteile gegenüber NRZ, aber höhere Bandbreite und alle anderen NRZ-Nachteile.
- Biphase = ganze Gruppe von Codes
Alle Biphase-Leitungscodierungen haben mindestens einen Signalwechsel pro Bitintervall und höchstens zwei Signalwechsel pro Bitintervall. (Bitintervall = Bitzelle)

Vorteile:

- Leichte Synchronisierung, da stets mindestens ein Signalwechsel pro Bitintervall (eine „Impulsflanke“ zum Triggern des Empfängers)
- Keine Gleichstromkomponente, da Mittelwert = 0 möglich
- Fehlererkennung auf Signalebene möglich: Fehlen eines erwarteten Übergangs leicht erkennbar
- Biphase = Zweiphasen-Code mit zwei Abschnitten pro Bitzelle

Leitungscodierung digitaler Signale (4)

- Biphase-L (Level)
 - Wird auch als Manchester-Code bezeichnet. Signalwechsel in der Mitte des Intervalls. Je nach Bitfolge (Datenelemente) Hilfswechsel an den Intervallgrenzen erforderlich (verdoppelte Baudrate). 1. Hälfte der Bitzelle hat Absolutwert des Signals: High = hoher Pegel; low = niedriger Pegel.
- Biphase-M (Mark), Biphase-S (Space)
 - Grundsätzlich immer ein Signalwechsel am Beginn des Intervalls.
 - Biphase-M: bei „1“ zusätzlicher Signalwechsel in der Intervallmitte
 - Biphase-S: bei „0“ zusätzlicher Signalwechsel in der Intervallmitte
- Differenzial Manchester
 - Grundsätzlich ein Signalwechsel in der Intervallmitte. Bei „0“ zusätzlich ein Signalwechsel am Intervallbeginn

Leitungscodierung digitaler Signale (5)

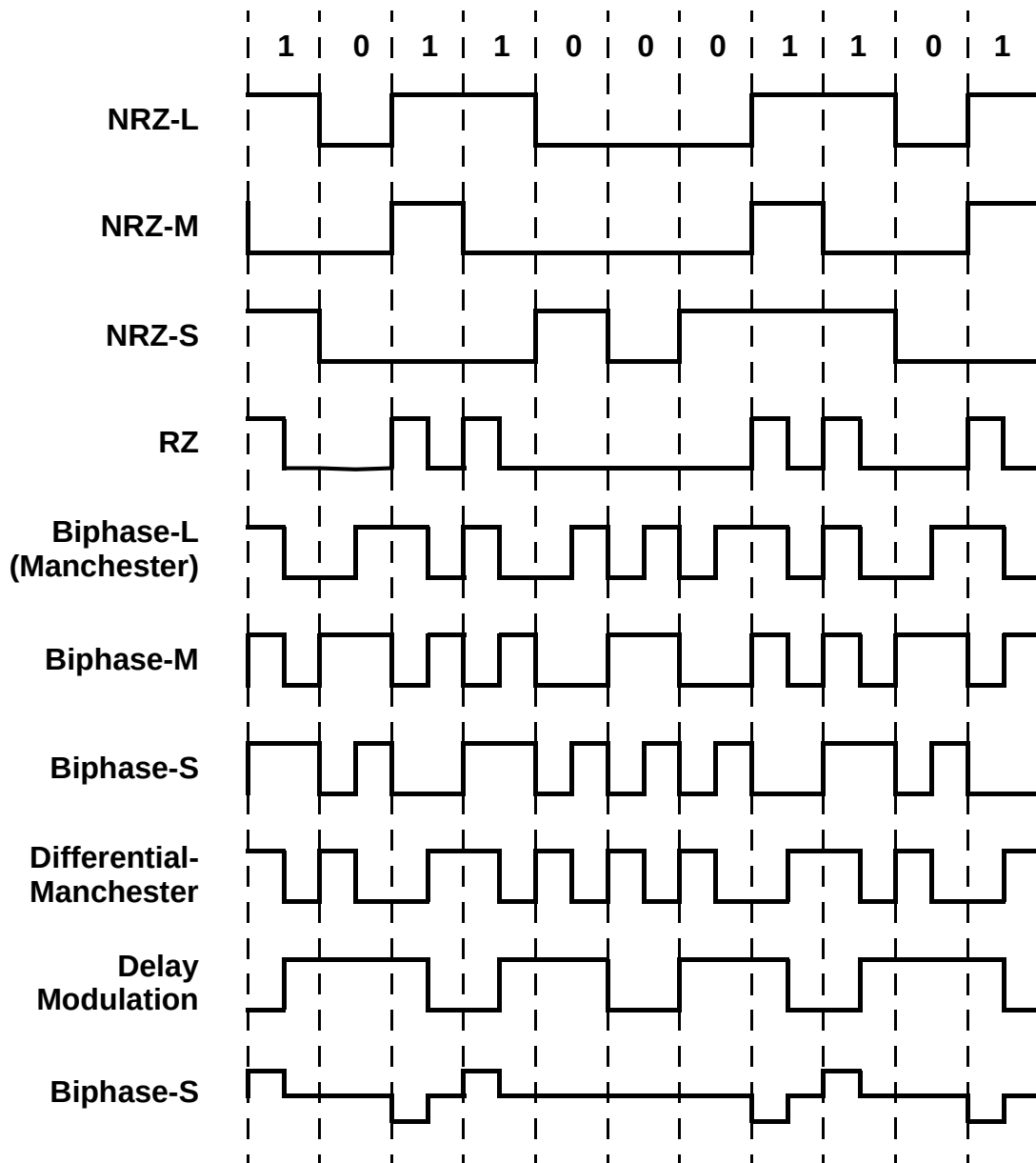
Bipolar

Beispiel einer Leitungscodierung mit mehr als zwei Signalwerten (hier tertiäres Signal).

„1“ wird abwechselnd durch positiven oder negativen Impuls in der 1. Hälfte des Bitintervalls dargestellt, dadurch keine Gleichstromkomponente.

Auch AMI (Alternate Mark Inversion) genannt.

Leitungscode



NRZ-L = non return to zero-level

NRZ-M = non return to zero-mark

NRZ-S = non return to zero-space

RZ = return to zero

Multiplexing: Mehrfachnutzung von Übertragungswegen

Definition: Übertragungsweg

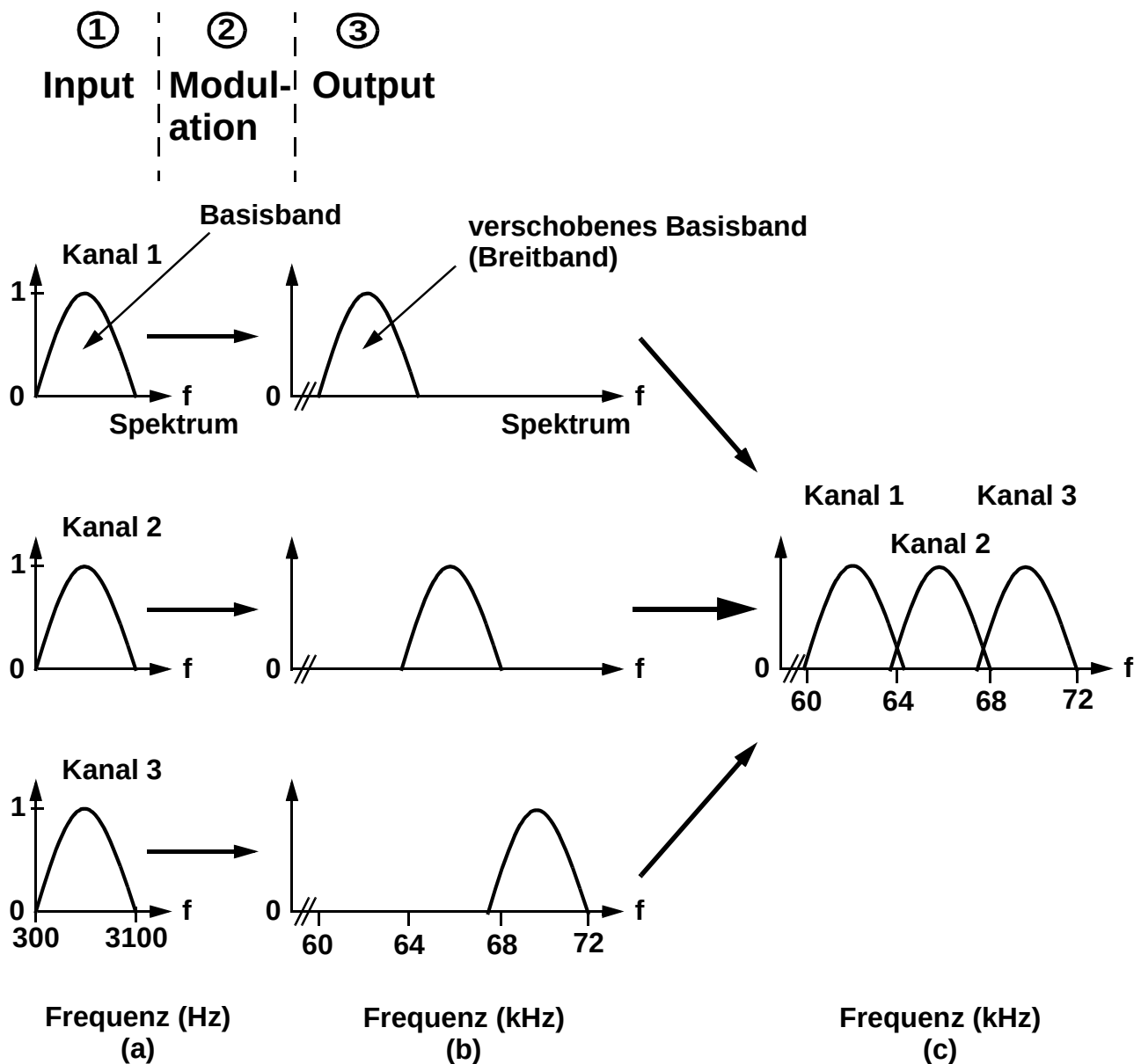
- physikalisch-technisches Transportsystem für Signale (z. B. Kabel)

- **Definition: Übertragungskanal**

- Abstraktion eines Übertragungsweges für einen Signalstrom = logischer Übertragungsweg bzw. virtueller Übertragungsweg.

• Auf einem Übertragungsweg können oft mehrere Übertragungskanäle parallel betrieben werden. So ist beispielsweise eine Aufspaltung der gesamten Übertragungskapazität eines Übertragungsweges auf verschiedene Sender-Empfänger-Paare möglich. Die Zusammenfassung von mehreren Übertragungskanälen auf einem Übertragungsweg heißt Bündelung oder Multiplexing.

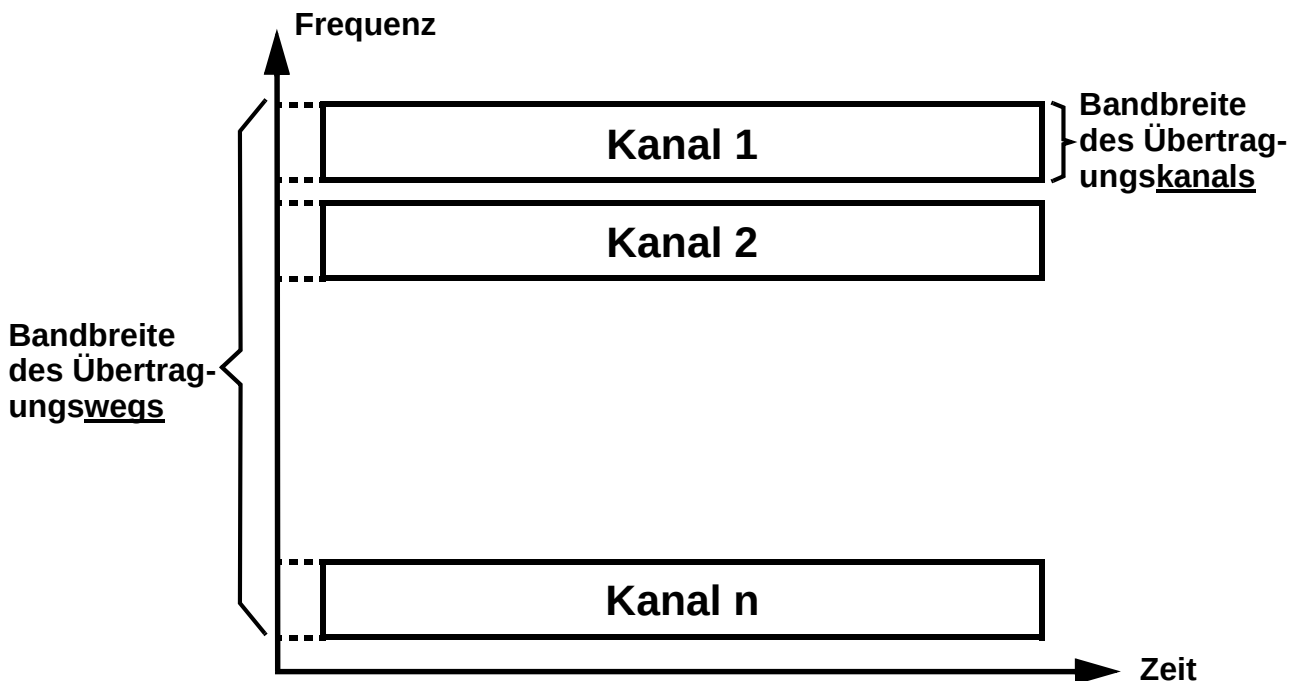
Beispiel für Frequenzmultiplexing



- (a) Die ursprünglichen Bandbreiten (= Basisband)
- (b) Die Bandbreiten mit verschobener Frequenz (= Breitband)
- (c) Auf dem Übertragungsweg (Kabel)

Frequenzmultiplexing (2) Frequency Division Multiplexing)

Breitbandige Übertragungswege ermöglichen die Unterbringung vieler Übertragungskanäle in unterschiedlichen Frequenzbereichen (Frequenzbändern), d. h., man teilt die verfügbare Bandbreite in eine Reihe von - nicht notwendigerweise gleich breite - Frequenzbänder auf und ordnet jedem Frequenzband einen Übertragungskanal zu.

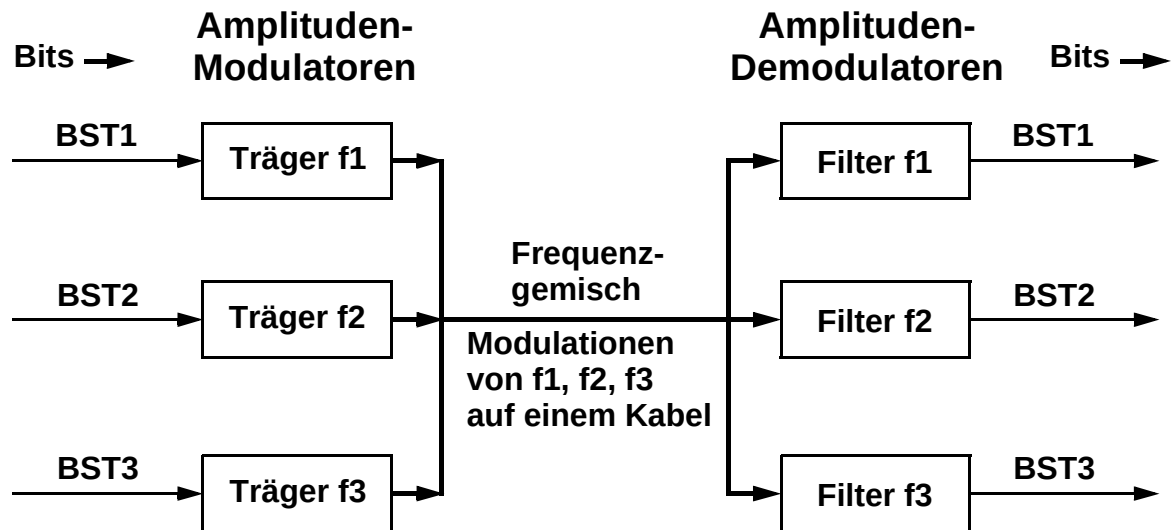


n

$$\sum_{i=1}^n \text{Bandbreite (Kanal } i) = \text{Bandbreite des Übertragungswege}$$

$= I$

Schema der technischen Realisierung eines Frequenzmultiplex-Systems

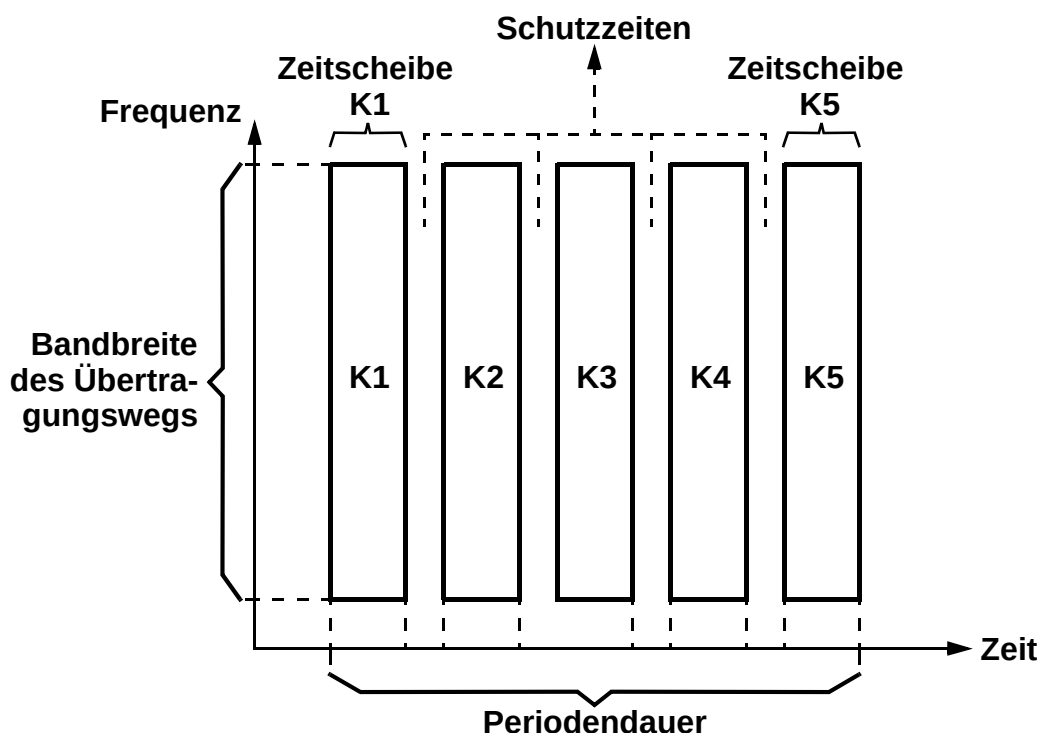


BST i = Bitstrom i, entspricht Kanal i

Synchrones oder starres Zeitmultiplexing (Time Division Multiplexing)

Die gesamte Übertragungskapazität (die ganze verfügbare Bandbreite) wird einer Sender-Empfänger-Kombination zur Verfügung gestellt. Nach einer Schutzzeit wird dann die gesamte Kapazität des Übertragungsweges dem nächsten Kanal zugeteilt. Pro Periode erhält also jeder Kanal einen Zeitschlitz (time slot) = time sharing des Übertragungsweges.

Diese zeitlich gestaffelte Übertragung mehrerer Signalströme wird als Zeitmultiplexing (TDM - time division multiplexing) bezeichnet.



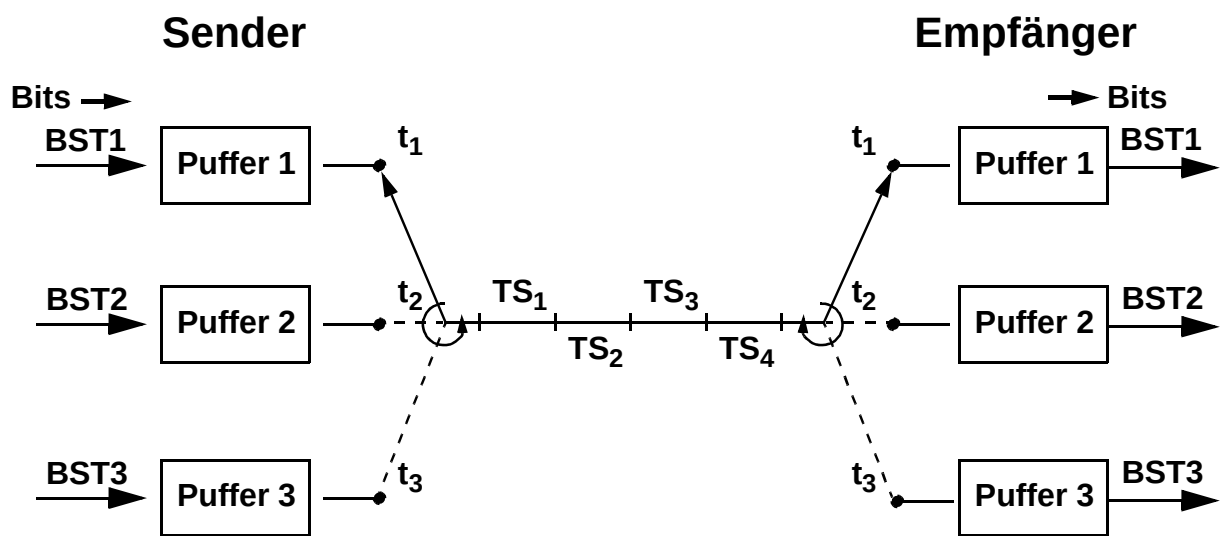
Analogie: time sharing bei Rechnerbetriebssystemen

$$\sum_i \text{Zeitscheibe (Kanal}_i) = \text{Periodendauer des Übertragungswegs}$$

Synchrones Zeitmultiplexing (2)

Zeitmultiplexing ist nur für zeitdiskrete Signale einsetzbar (bevorzugt zeit- und wertdiskrete Signale = Digitalsignale), da das Signal nicht kontinuierlich übertragen wird.

Festes Zeitmultiplex mit starrer Zeitscheibenzuteilung (synchron)



Jedem der n Sender wird periodisch eine Zeitscheibe (time slot, time slice) $TS_1, TS_2, \dots, TS_n$ zugeteilt. Sender, Abtaster und Detektionsmechanismus beim Empfänger laufen im gleichen Takt. Deshalb auch als synchrones Zeitmultiplexing bezeichnet. Beispiel: Telefonvermittlungssystem (PCM-System).

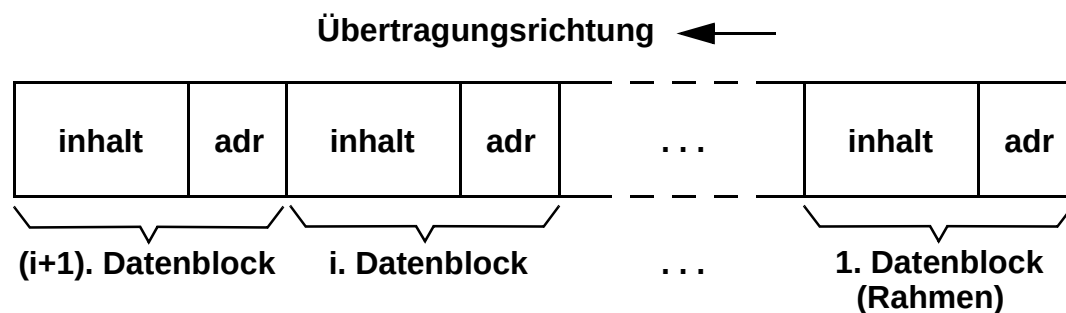
Zeittakt-Stabilität ist wichtig!

BST = Bitstrom

Asynchrones Zeitmultiplexing

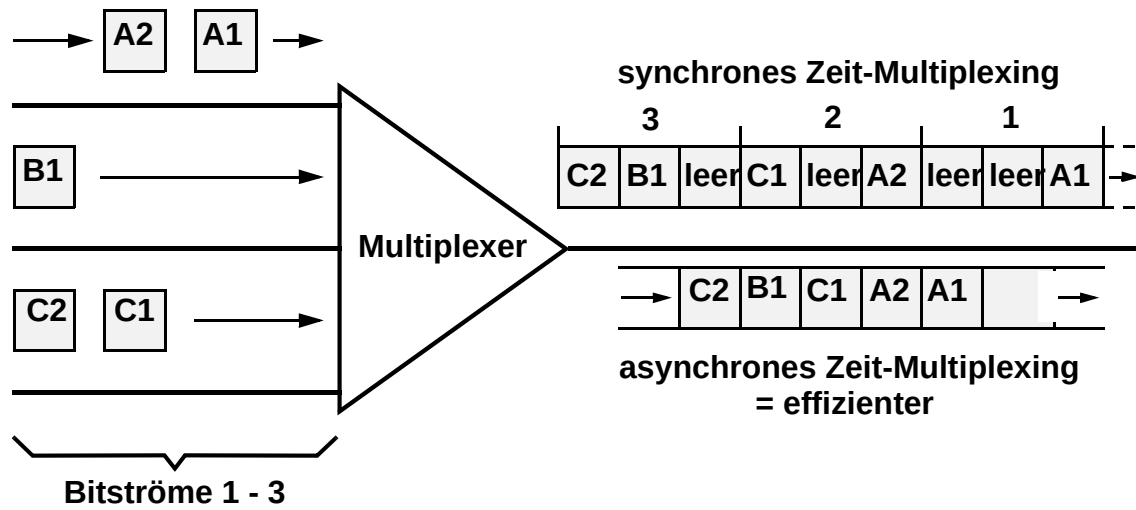
Der Übertragungsweg wird dem Sender nicht fest, sondern nach Bedarf zugeteilt. Der Empfänger kann aus der Zeitlage der Zeitscheiben nicht mehr die Herkunft identifizieren! Es wird eine Kennung pro Datenblock (Paket, Zelle) erforderlich (Empfängeradresse, Kanalkennzahl, o. Ä.) Beispiel: Ethernet, ATM,...

Schematischer Aufbau eines Übertragungsblocks mit Kennung



Das anforderungsgesteuerte Zeitmultiplexing (demand multiplexing) wird auch als statistisches Zeitmultiplexing (STDM = statistical time division multiplexing) bezeichnet.

Zeit-Multiplexing-Techniken im Vergleich



- Bei LAN's wird asynchrones Zeitmultiplexing eingesetzt (LAN = local area network)

Funknetze; Satellitennetze

- Satellitennetze sind wie Bus- und Ringnetze Rundspruchnetze. Beispiel: GPS, Iridium
- Der Satellit ist dabei eine logisch passive Verstärkerstation (die Signale von der beteiligten Erdstation werden dabei in eine andere Frequenz umgesetzt und wieder ausgesendet (= Relaisstation)).
- Eine sendende Erdstation kann durch die verschiedenen Sende- und Empfangsfrequenzen des Satelliten sofort feststellen, ob ein Datenpaket gestört wird (durch Empfang des eigenen Pakets).
- Bei Satellitennetzen wird das Problem der Kanalzuweisung durch die hohen Signallaufzeiten erschwert (bei einem Tokenmechanismus wäre der Kanal jeweils für 270 ms unbenutzt)

GPS = global positioning system

Rundspruch = Broadcast

Satellit in geostationärem Orbit hat Entfernung von ~ 36.000 km.

=> Bei $c = 300.000 \text{ km/s}$ ist Signal ~ 0,24 s unterwegs (Hin- und Rückweg)

Satellitenkommunikation

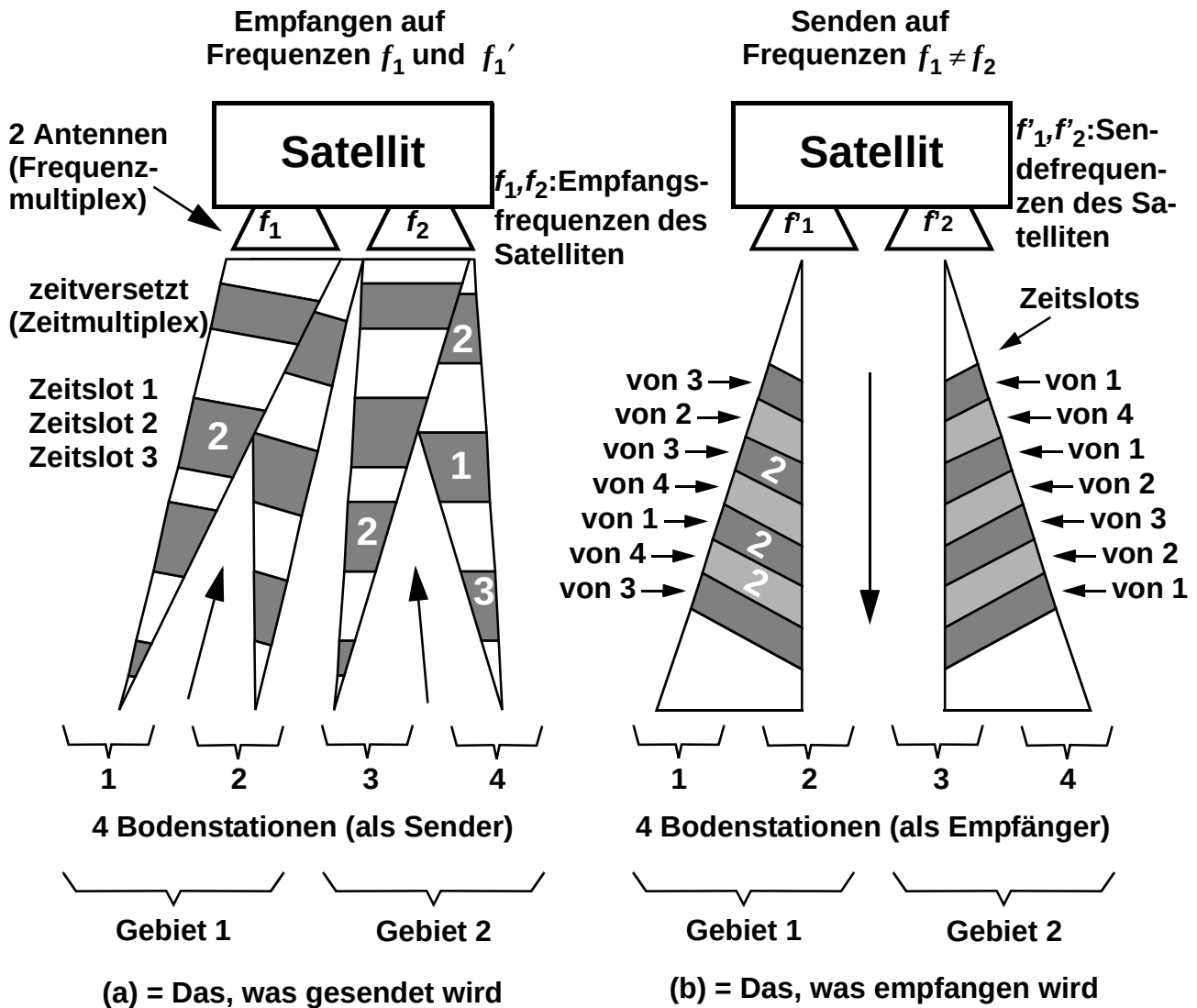
- Hohe Bandbreite, ca. 50 MHz pro Kanal
- Rundspruchnetz = Broadcast (Sicherheitsprobleme)
- Lange Verzögerung (Latenz)
 - Für die festen Erdstationen ist ein geosynchroner (= geostationärer) Orbit notwendig.
 - Dieser liegt auf einer Höhe von 36.000 km
 - Dies ergibt eine Verzögerung von 240 ms (hin zum Satelliten und zurück).
 - Die lange Verzögerung beeinflusst die Protokolle der höheren Schichten.
- Beispiel INTELSAT:
 - 794 PCM Simplexkanäle (= unidirektional), jeder 64 KBit/s, plus 128 KBit/s Signalisierungskanal pro 50 MHz-Kanal
 - Multiplexing mit FDM (Unterteilung von 50 MHz Bandbreite)
 - Je ein Paar Simplexkanäle formt einen Duplexkanal (= bidirektional)

FDM = Frequency Division Multiplex = Frequenz Multiplex

Simplex-Kanal = unidirektionaler Kanal

Duplex-Kanal = bidirektionaler Kanal

Satellitenkommunikation



- Frequenzmultiplex dient zur Unterscheidung zwischen verschiedenen Gebieten und zwischen Senden/Empfangen
- Innerhalb eines Sendegebietes wird Zeitmultiplexing (TDM) benutzt.
- Zwischen Sendegebieten werden verschiedene Antennen benutzt

TDM = Time Division Multiplex
 = Zeitmultiplex = zeitliche Verschränkung der Signale

Die wichtigsten Satellitenfrequenzen

Band	Frequencies	Downlink (GHz)	Uplink (GHz)	Problems
C	4/6	3.7-4.2	5.925-6.425	Terrestrial interference
Ku	11/14	11.7-12.2	14.0-14.5	Rain
Ka	20/30	17.7-21.7	27.5-30.5	Rain; equipment cost

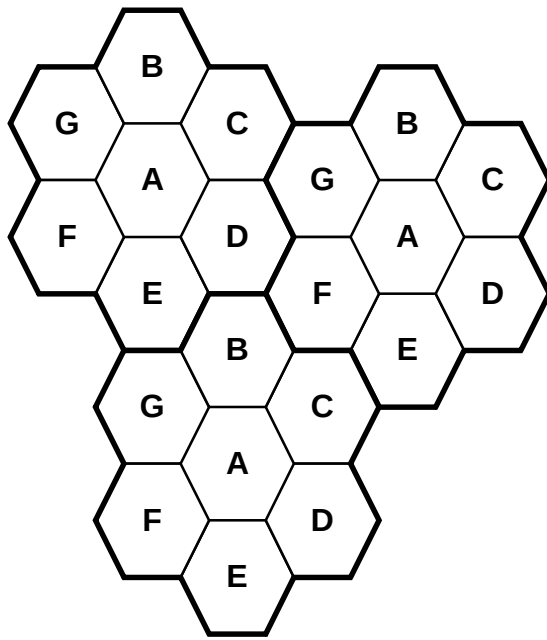
Mobilfunknetze (Handy, Cell phone Mobile phone)

- Internationale Standards für den digitalen Zelfunk sind verabschiedet (z. B. GSM in Europa) für Handy; UMTS in Zukunft.
- Starke Verbreitung für die Telefonie (D 1, D 2, E-plus,...)
- Bandbreite sehr niedrig (einige KHz); 8 Zeitscheiben pro Trägerkanal.
 - abhängig von der Breite eines Trägerkanals (200 KHz)
 - Dieser muss wegen des Frequenzmultiplexings innerhalb einer Zelle sehr schmalbandig sein: pro Zelle 124 KHz-Duplexkanäle + 8 Zeitscheiben pro 200 KHz Träger à 0,5 ms.
- Datenanwendungen in Mobilfunknetzen entwickeln sich rasant (Daten übers Handy, z. B. Fax mit 9600 Band).
- Zunehmend wird auch Mobilfunktechnik innerhalb von Gebäuden verwendet, z. B. für den Anschluss von autonomen Robotern an ein LAN oder für die Bürokommunikation von einem Meeting aus.

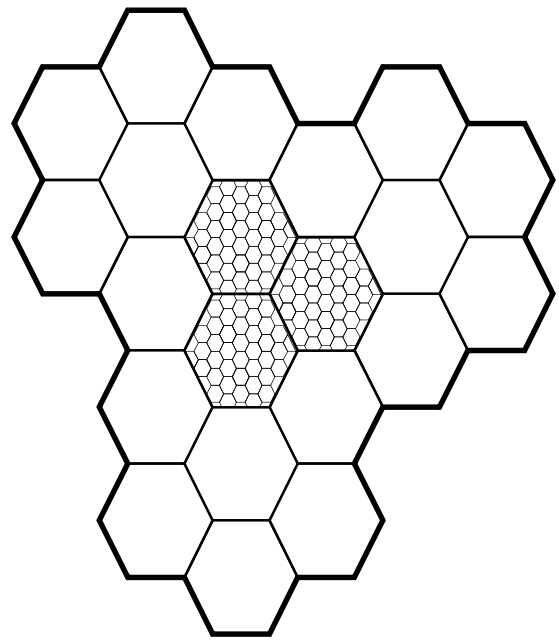
GSM = global system for mobile communication

UMTS = universal mobile telephone system

Aufteilung der Frequenzen in Zellen



(a)



(b)

gleiche Buchstaben bezeichnen gleiche Zellen

a) homogene Zellgröße auf dem flachen Land

b) kleinere Zellen für Städte

Was haben wir gelernt?

Codierung

Multiplexen

...

Neues Problem?

**Wie Sorge ich dafür, dass die Daten sicher
übertragen werden?**

Sicherungsschicht (Data Link Layer)

- 1.) Übertragungsfehler: Ursachen
- 2.) Fehlererkennungs- und Fehlerkorrekturcodes
- 3.) Bit Stuffing und Rahmenbegrenzer
- 4.) Bestätigung und Sequenznummern
- 5.) Flusskontrolle
- 6.) Beispiel: HDLC

Aufgaben der Sicherungsschicht (Schicht 2)

- Verdeckung (= Unsichtbarmachung) von Übertragungsfehlern zwischen direkt benachbarten Partnern (keine Zwischenknoten) bestehend aus den Teilschnitten Erkennung und Behebung
- Flusssteuerung des Datenverkehrs zum Nachbarknoten
- Bei LANs zusätzlich: Medienzugangskontrolle zum gemeinsam genutzten Übertragungsmedium

flow control = Flusssteuerung

Wahrscheinlichkeit von Übertragungsfehlern bei Telefondrähten

Empirisch ermittelte Fehlerrate für analoge Telefonleitungen und Blöcke aus n Bytes:

$$\text{Rate} = 10^{-4} \cdot n^{0.8}$$

n = Zahl der übertragenen Bytes

n	p(n)
8	5.3-04
16	9.2E-04
32	0.0016
64	2.8E-03
128	4.9E-03
256	8.4E-03
512	1.5E-02
1024	0.026 = 2,6%

← hohe Fehlerrate

- Sobald digitale Daten über ein Modem übertragen werden, wird die Fehlerrate der verwendeten Telefonleitung wichtig
- Bei LANs und WANs ist die Fehlerrate viel kleiner; typisch: 10^{-14} Fehler pro Bit.

Übertragungsfehler: Ursachen

- 1.) Weißes Rauschen = thermisches Rauschen
- 2.) Dämpfung und Verzerrung
 - Signalverzerrung ist abhängig von der Frequenz
- 3.) Übersprechen auf Leitungen verursacht durch Magnetfelder
- 4.) Übersprechen auf Leitungen verursacht durch kapazitive Kopplung
- 5.) Impulsstörungen durch elektrische Felder (z. B. Erdschleifen)
 - Wird z. B. durch elektrische Geräte verursacht
 - Bei einer Störung von 10 ms Dauer entspricht dies 560 Bit bei einem 56kbit/s-Modem

Viele Fehler kommen kurz aufeinanderfolgend (sog. Bursts auch Fehlerbündel genannt)

Vorteil: Nur wenige Blöcke enthalten Fehler.

Nachteil: Schwer zu erkennen und zu korrigieren, da viele Fehler auf einmal, wenn sie kommen.

Fehlererkennungs- und Fehlerkorrekturcodes

Fehler heißt: Die empfangene Information entspricht nicht der gesendeten.

- **Zur Fehlerkorrektur wird die eigentliche Information durch Kontrollinformationen (Redundanz) ergänzt.**
- **Je nach Umfang der Redundanz kann eine gewisse Anzahl an Fehlern erkannt oder sogar korrigiert werden.**
- **Der Zusammenhang zwischen dem Umfang der notwendigen Redundanz, der Fehlerhäufigkeit der Übertragungsstrecke, der Entdeckung und der eventuellen Korrektur eines Fehlers wird in der Nachrichtentechnik behandelt.**

Fehlererkennung durch Paritätsbit

		Bsp. 1	Bsp. 2	Bsp. 3		
ein ASCII- Zeichen	Bit 1	0	1	1		
	Bit 2	1	0	1		
	Bit 3	0	0	0		
	Bit 4	0	1	0		
	Bit 5	0	0	0		
	Bit 6	0	1	0		
	Bit 7	1	1	1		
		0	0	1	gerade	Parität
Paritätsbit		1	1	0	ungerade	

- Das Paritätsbit ergänzt die Anzahl der Einsen im Zeichen so, dass sie gerade/ungerade wird (gerade Parität = gerade Zahl von Einsen)
- Berechnung der gesamten Quersumme (inkl. Parity) so, dass diese gerade oder ungerade wird.

Querparität (= 1 Bit pro Zeichen)

- Wird zum Beispiel bei der asynchronen Übertragung verwendet. Beispiel: V. 24 hat Paritätsbit gemäß Querparität.
- Die 5 bis 8 eigentlichen Informationsbits werden durch ein weiteres (redundantes) Bit innerhalb des Rahmens erweitert.
- Der Begriff stammt aus der Magnetband-Technik, als ein Zeichen gerade quer auf 6 bis 8 Spuren passte.
- Das Redundanzbit (= Paritätsbit) ergänzt die Anzahl der Einsen innerhalb der Information auf eine gerade oder ungerade Anzahl. (=> 9-Spur-Bänder)

1 0 0 1 1 1 0 0 0

1 0 1 1 1 1 0 0 1



Information



Paritätsbit

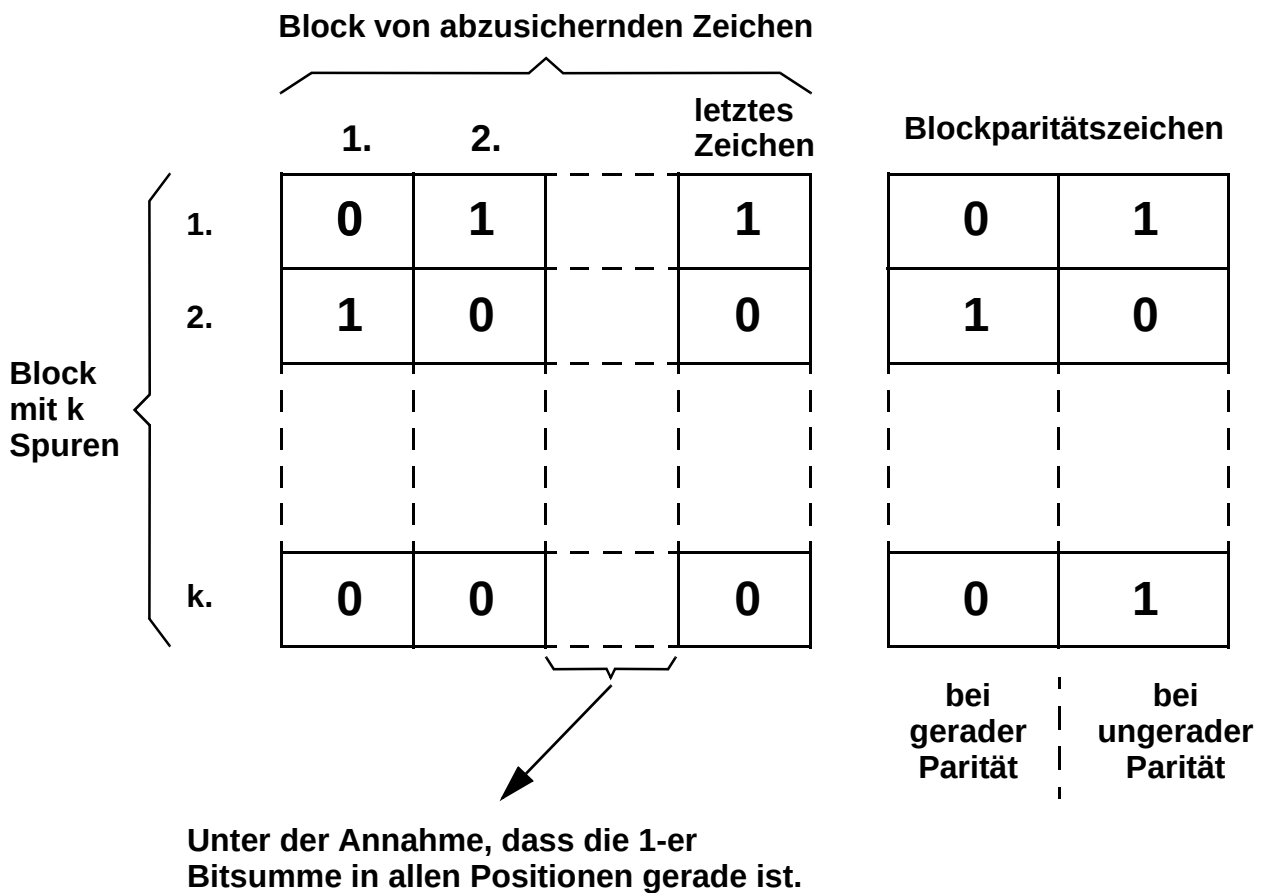
(gerade Parität)

- Es kann nur ein Fehler erkannt und keiner korrigiert werden.

Längsparität (Blockparität)

= 1 Bit pro Bitposition in einer Bitmatrix

- Berechnung eines Paritätsbits jeweils am Blockende in der Längsrichtung des Magnetbandes



- Über dieselbe Bitposition mehrerer nachfolgender Zeichen wird die Parität berechnet. Beispiel: Es wird 8 x die Position 0 = (LSB) betrachtet und daraus das Paritätsbit berechnet.

Cycling Redundancy Check (CRC)

Grundidee

- Betrachte Bitfolgen als Darstellung eines Polynoms von X nur mit den Koeffizienten 0 und 1, wobei X als $X=Z$ zu denken ist.

Beispiel: $11001 = x^4 + x^3 + x^0$ ist ein Polynom von Grad 4

- Wähle ein Generatorpolynom $G(x)$ vom Grad g .
- Hänge an die Nachricht $M(x)$ eine CRC-Prüfsumme derart an, dass das Polynom $T(x)$, dargestellt durch die Nachricht mit angehängter Prüfsumme, durch $G(x)$ teilbar ist.
- Übertrage $T(x)$

Algorithmus für CRC

- 1.) Wir haben $G(x)$ vom Grad g und hängen g 0-Bits an die Nachricht, also $x^g M(x)$. Das Anhängsel ist damit vom Grad $g-1$.
- 2.) Teile $x^g M(x)$ durch $G(x)$, Division modulo 2 (= ohne Bor-gen)
- 3.) Subtrahiere den Rest von $x^g M(x)$, modulo 2 (= ohne Bor-gen)
- 4.) Das Ergebnis ist $T(x)$ = die Nachricht samt Prüfsumme für die Übertragung. $T(x)$ ist garantiert durch $G(x)$ teilbar.
- 5.) Der Empfänger teilt $T(x)$ durch $G(x)$. Gibt es keinen Rest, ist $T(x)$ mit hoher Wahrscheinlichkeit fehlerfrei übertragen worden.

Beispiel für CRC-Berechnung

Rahmen: 1 1 0 1 0 1 1 0 1 1
Generator: 1 0 0 1 1 => Grad g = 4
Nachricht nach dem anhängen von 4 Nullbits:

1 1 0 1 0 1 1 0 1 1 0 0 0 0, dann teilen:

1 1 0 1 0 1 1 0 1 1 0 0 0 0 : 1 0 0 1 1 = 1 1 0 0 0 0 1 0 1 0
mit Rest 1 1 1 0

- 1 0 0 1 1 <--- kein Borgen!

''1 0 0 1 1

- 1 0 0 1 1

''0 0 0 0 1

- 0 0 0 0 0

''0 0 0 1 0

- 0 0 0 0 0

''0 0 1 0 1

- 0 0 0 0 0

''0 1 0 1 1

- 0 0 0 0 0

''1 0 1 1 0

- 1 0 0 1 1

''0 1 0 1 0

- 0 0 0 0 0

''1 0 1 0 0

- 1 0 0 1 1

''0 1 1 1 0

- 0 0 0 0 0

Rest: 1 1 1 0

(Division modulo 2: Kein Übertrag bei der Subtraktion)

Daraus: 1 1 0 1 0 1 1 0 1 1 0 0 0 0 - ^{Rest} 1 1 1 0 =

1 1 0 1 0 1 1 0 1 1 1 1 0 = gesendeter Rahmen

wichtig:

Subtraktion ohne Borgen } gemäß algebraischer Feld-
Addition ohne Übertrag } theorie
= bitweise EXOR

Generatorpolynome in internationalen Standards

• CRC-16 = $x^{16} + x^{15} + x^2 + 1$ (ISO)

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

beide werden für 8-Bit-Zeichencodes benutzt

Fehlererkennung mit CRC-16, CRC-CCITT (16 Bits)

(Ergebnisse aus der Codierungstheorie)

erkannt werden:

- alle einfachen und zweifachen Bitfehler
- alle Fehler mit einer ungeraden Bitzahl
- alle stoßweisen Fehler (= Bursts) mit einer Länge < 16 Bit
- 99,998% aller längeren stoßweisen Fehler > 16 falsche Bit

Implementierung des CRC

In Hardware (z. B. im HDLC-Chip) mit Hilfe eines Schieberegisters und der bitweisen XOR-Funktion.

=> CRS lässt sich sehr einfach implementieren.

Bit Stuffing und Rahmenbegrenzer

Zur Anwendung von fehlererkennenden und fehlerkorrigierenden Codes muss der Datenstrom in einzelne Rahmen unterteilt werden. => Pakete bzw. Rahmen.

Problem: Wie kann man solche Rahmen im Bitstrom begrenzen (= kennzeichnen), wenn jedes beliebige Bitmuster in den Nutzdaten vorkommen kann?

Lösung: Bit Stuffing (Bitstopfen)

Als Begrenzer (= Kennzeichner) wählt man 01111110 (= 6 Einsen hintereinander). Der sender fügt nach fünf Einsen im Nutzdatenstrom IMMER eine 0 ein.

Beispiel:

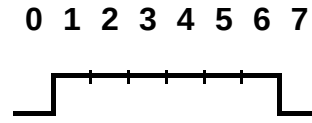
Sender	0	1	1	1	1	1	1	0	1	0	1	1	1	1	1	0	1	0	1		
Leitung	0	1	1	1	1	1	0	1	0	1	0	1	1	1	1	1	0	0	1	0	1
Empfänger	0	1	1	1	1	1	1	0	1	0	1	1	1	1	1	1	0	1	0	1	

Wenn der Empfänger nach fünf Einsen eine Null sieht, entfernt er diese aus dem Datenstrom.

=> Sechs Einsen hintereinander kann es auf der Leitung nur am Rahmenanfang geben.

Bit Stuffing-Beispiele

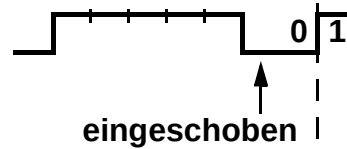
Rahmenkennzeichnung:



Übertragung von ≥ 6 Einsen
im Nutzdatenstrom:



Übertragung von 5 Einsen
einer Null und einer Eins:



Übertragung von 5 Einsen
und zwei Nullen:



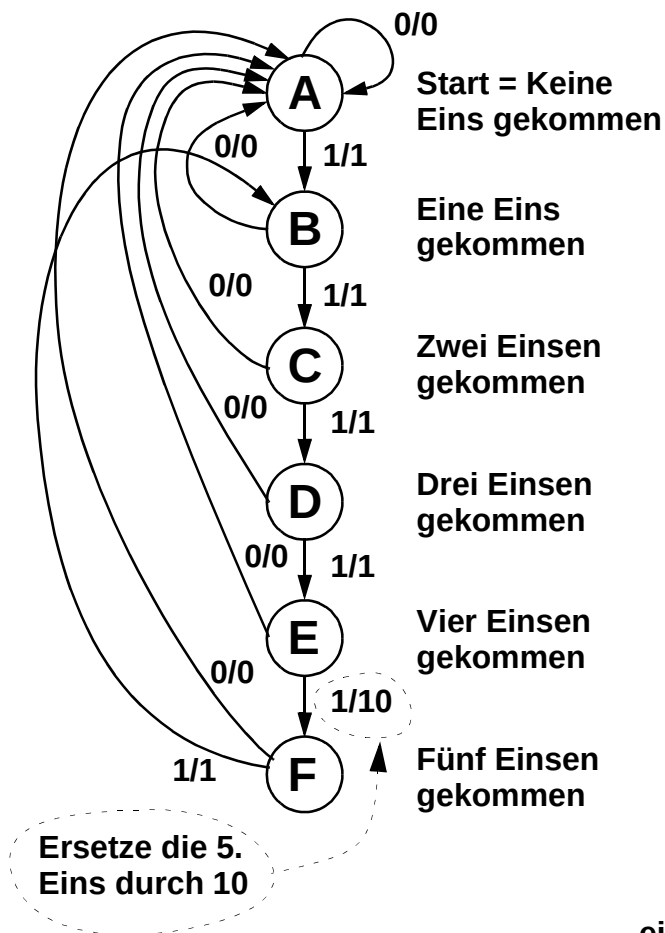
Realisierung des Bitstopfens

Endliche Automaten für Sender (Quelle) und Empfänger (Ziel) bei einer Übertragung mit Bitstopfen.

Sender

(Ereignis: zu übermittelndes Bit)

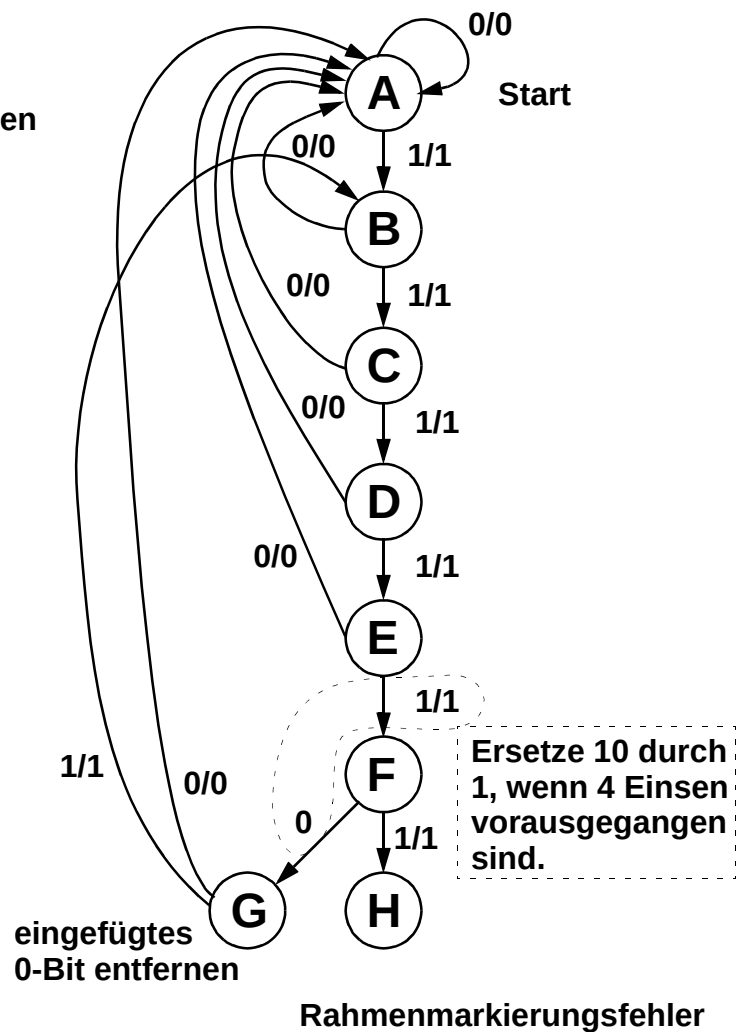
Sender hat 6 Zustände



Empfänger

(Ereignis: eintreffendes Bit)

Empfänger hat 7 Zustände

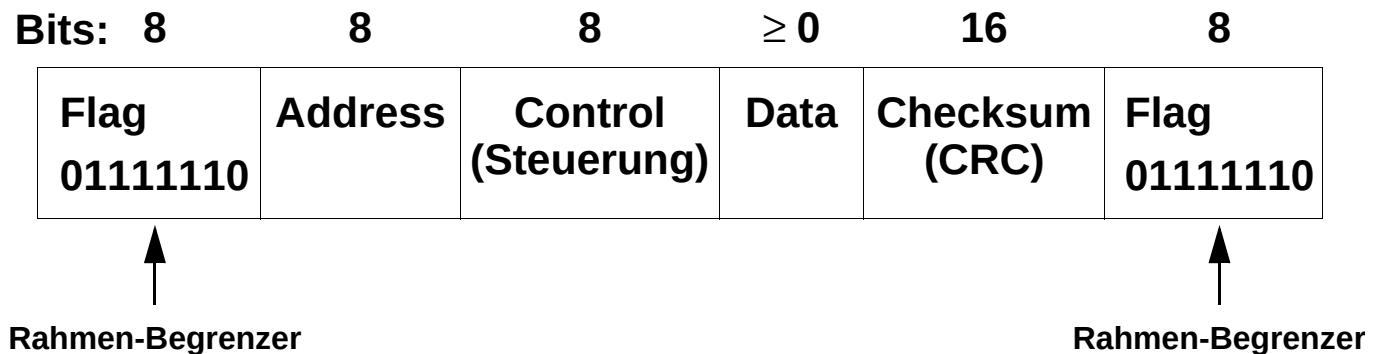


Notation: e/a e = Ereignis
 a = Ausgabe

Beispiel für ein Rahmenformat der Schicht 2 (HDLC)

Rahmen („Paket“) in der Schicht 2 (Sicherheitsschicht)

Beispiel:



- **Übertragungsrahmen = Datenpaket**
- **Flag = „Flagge“ = Kennzeichnung**
- **Auf der Schicht 1 spricht man von Bitstrom**
- **Auf der Schicht 2 spricht man von Rahmen**
- **Auf allen Schichten ≥ 3 spricht man von Paket**

Was steht im „Control“-Feld eines Rahmens?

- Es stehen u. a. Bestätigungen (Acknowledgements) und Sequenznummern

Acknowledgement = Anerkennung, Bestätigung, Quittung

Bestätigung bzw. Sequenznummern werden benutzt für

- Fehlererkennung fehlerhafter oder verlorener Blöcke
- Pufferverwaltung
- Flusssteuerung (= flow control)

Puffer (engl. buffer) ist ein kleines Stück Speicher zum Halten eines oder mehrerer Datenpakete

Puffer werden vom Sender wie vom Empfänger eingesetzt

to control: steuern

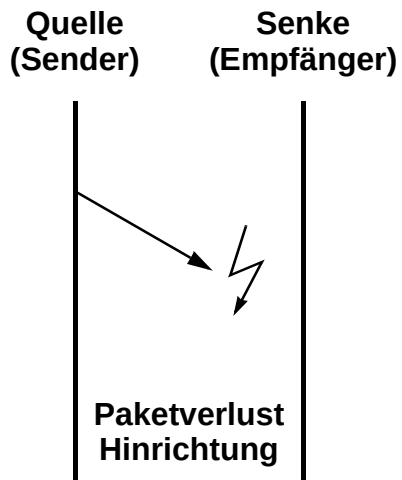
to check: kontrollieren

feed back control: regeln

Bestätigungen (Acknowledges)

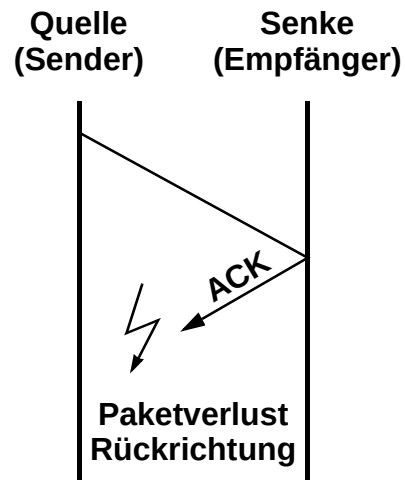
- Für jedes Datenpaket vom Sender zum Empfänger wird eine Bestätigung in Gegenrichtung geschickt, sofern das Datenpaket unversehrt empfangen wurde.
- Wird das Paket nicht empfangen oder fehlerhaft empfangen, erhält der Sender keine Bestätigung.
- Probleme gibt es, wenn das Paket nicht empfangen wurde oder die Bestätigung verloren ging (=> zwei verschiedene Probleme).

Zwei Fälle von Datenverlust



1. Verlust eines Datenblocks:

- Senke wartet auf Daten
- Quelle wartet auf Daten



2. Verlust einer Quittung:

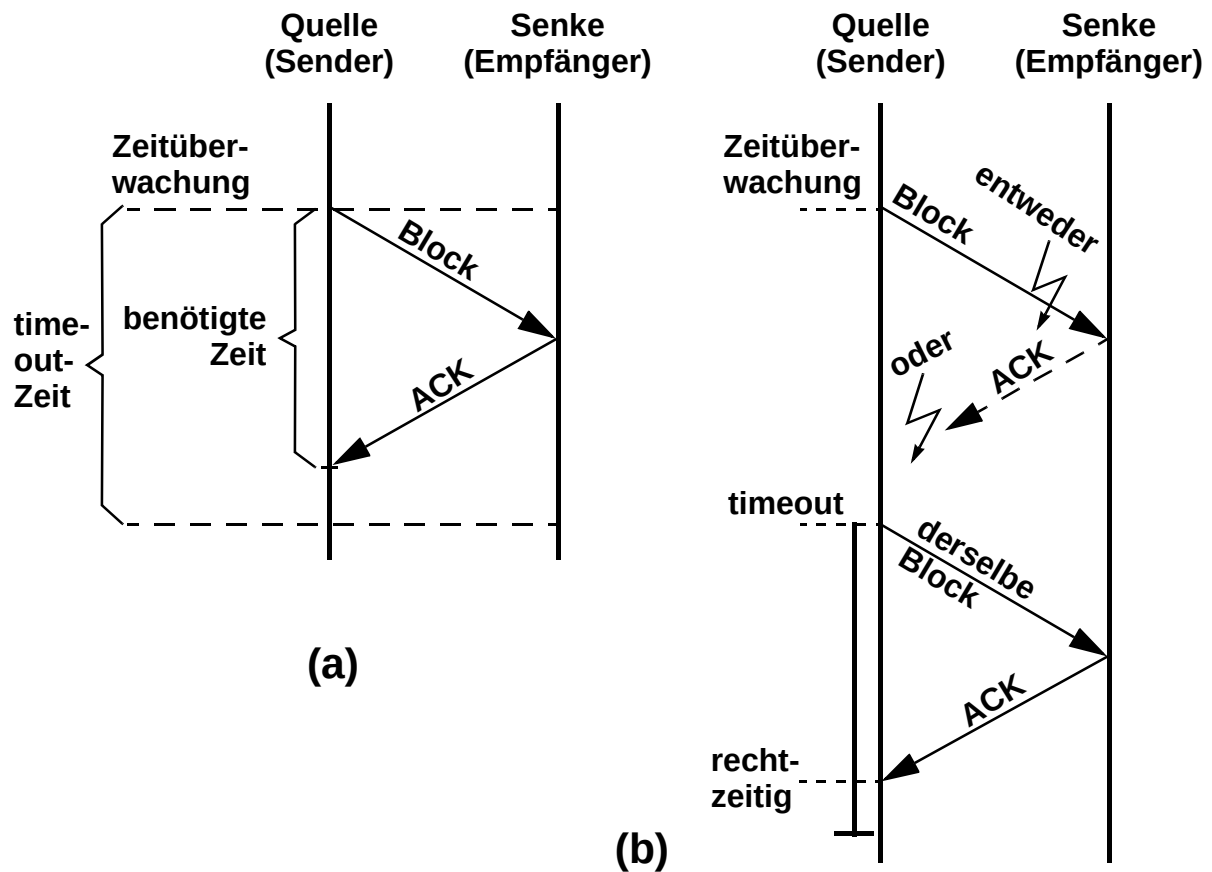
- Senke wartet auf Bestätigung
- Quelle wartet auf neue Daten

=> Problem von Acknowledges:

- Ohne Zeitschranke (Time-out): ewige Blockierung des Senders (ewig)
- Lösung: Einführung einer Zeitschranke (Time-out) beim Sender

Bestätigung mit Zeitschranke (time out)

Zeitüberwachung auf der Senderseite



- 1.) Beispiel für eine fehlerfreie Übertragung
- 2.) Beispiel für das Erkennen eines Nachrichtenverlustes durch Ablauf der Zeitüberwachung

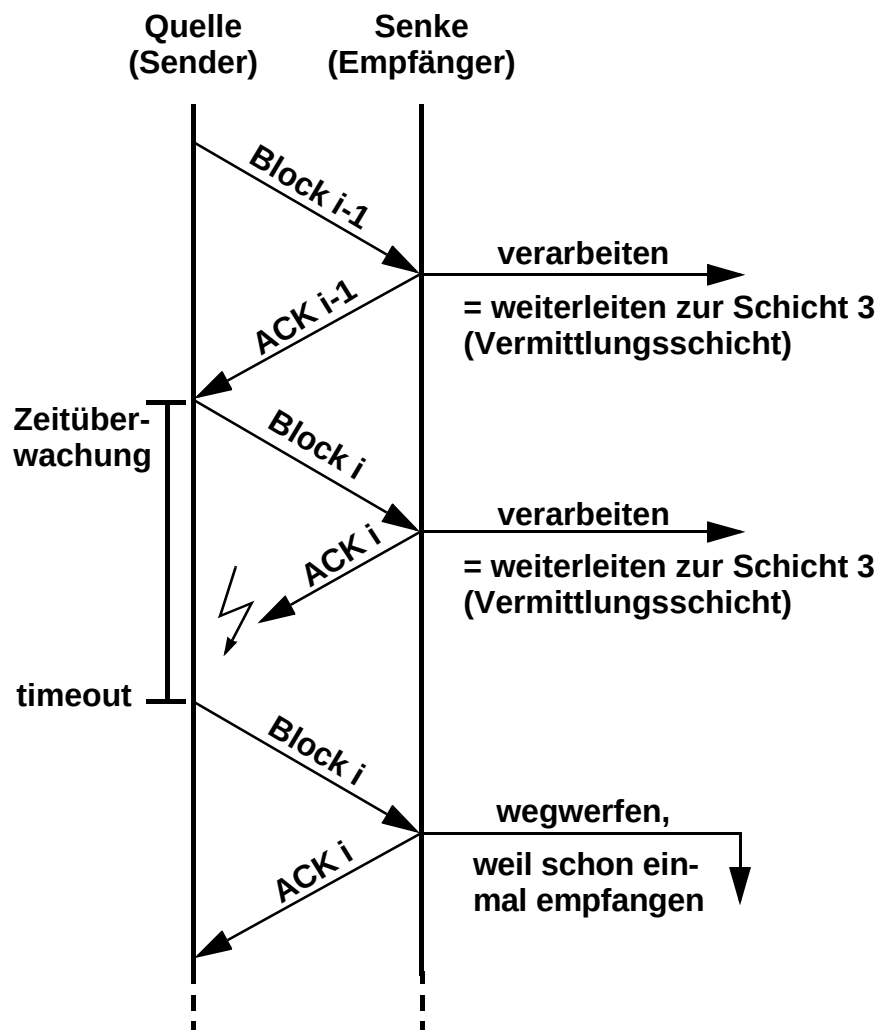
Neues Problem:

Der Block wird zweimal übertragen. Der Empfänger weiß nicht, ob er den Block zur Verarbeitung an die nächsthöhere Schicht weitergeben soll.

Bestätigung mit Zeitschranke und Sequenznummern

Lösung: Sequenznummern (= Blockadressen)

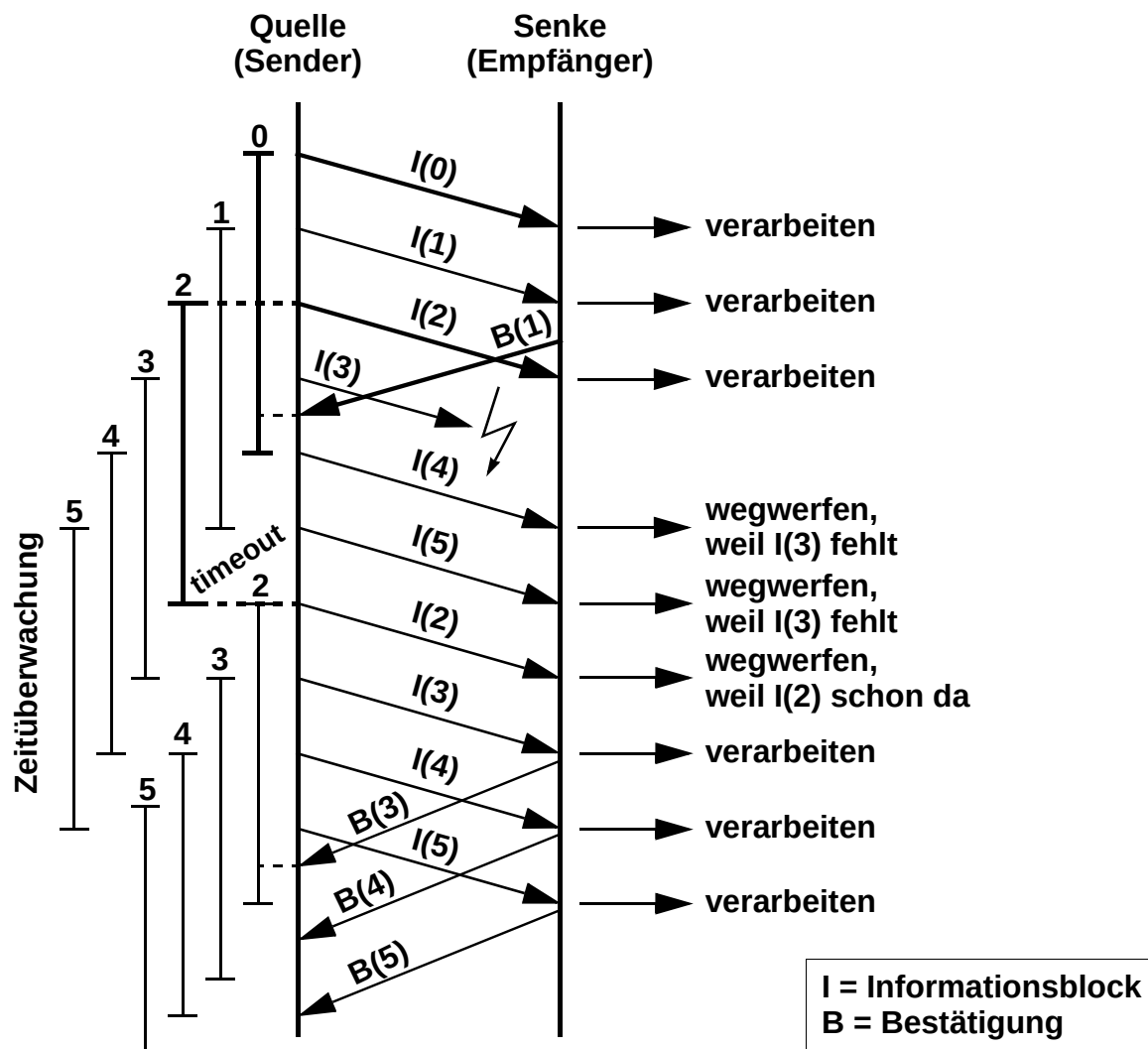
Jeder Block erhält eine Sequenznummer. Bei der Wiederholung des Rahmens wird die Sequenznummer beibehalten.



- Zurückgeschickte Sequenznummer dient zugleich als Bestätigung

Sammelquittung über Sequenznummern

Sequenznummern können auch anstelle mehrerer Bestätigungen verwendet werden. Mit einer Sequenznummern können dann mehrere Informationsblöcke quittiert werden.



Zurückgeschickte $B(3)$ Sequenznummer dient zugleich als Bestätigung für die Blöcke 2 und 3

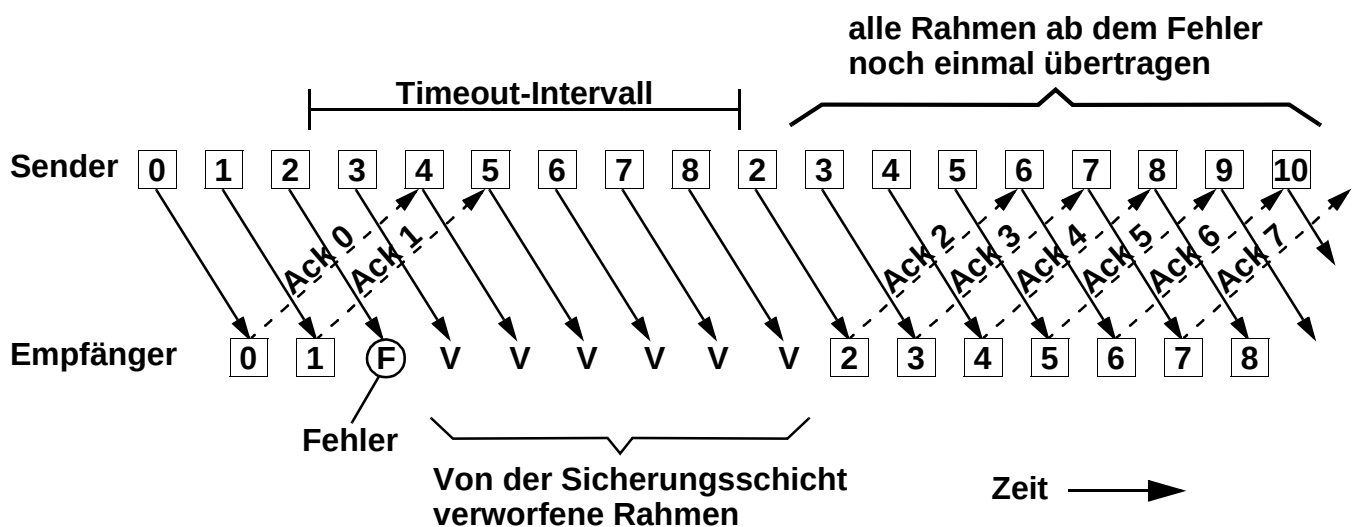
Fehlererkennung und -behebung

- Nachdem ein Übertragungsfehler durch Paritybit CRC-Prüfung und time out mit Sequenznummer und Acknowledge erkannt wurde, muss das betreffende Paket neu geschickt werden.
- Für die Paketwiederholung gibt es verschiedene Strategien.

Fehlerbehebung

Fehlerbehebung durch „go-back-n“ und ohne Pufferung beim Empfänger

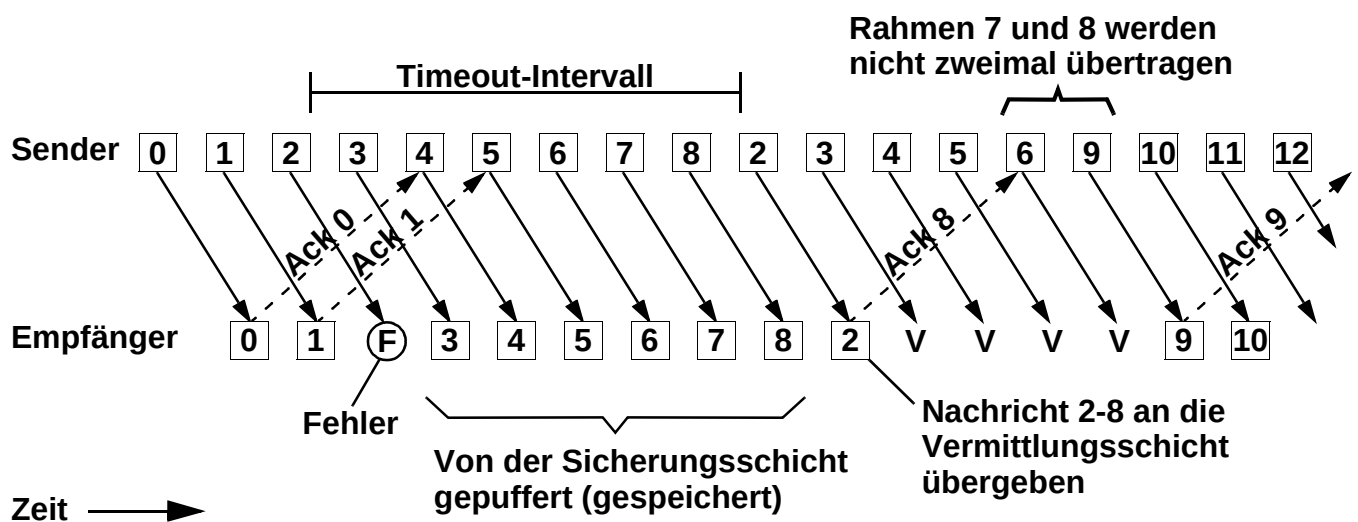
Im Falle eines Fehlers bleibt das Ack aus. Nach Ablauf des Timers überträgt der Sender sämtliche Rahmen ab dem Unbestätigten neu.



- Dass ein Fehler aufgetreten ist, wird vom Empfänger am falschen CRC festgestellt und vom Sender am time out.
- Jedes Paket muss quittiert werden; keine Sammelquittung möglich, da keine Pufferung
 - Vorteil: Einsparen von Pufferplatz
 - Nachteil: Ineffizient im Fehlerfall, da alle Rahmen während des time out-Intervalls neu übertragen werden müssen.

Fehlerbehebung durch „go-back-n“ mit Pufferung beim Empfänger

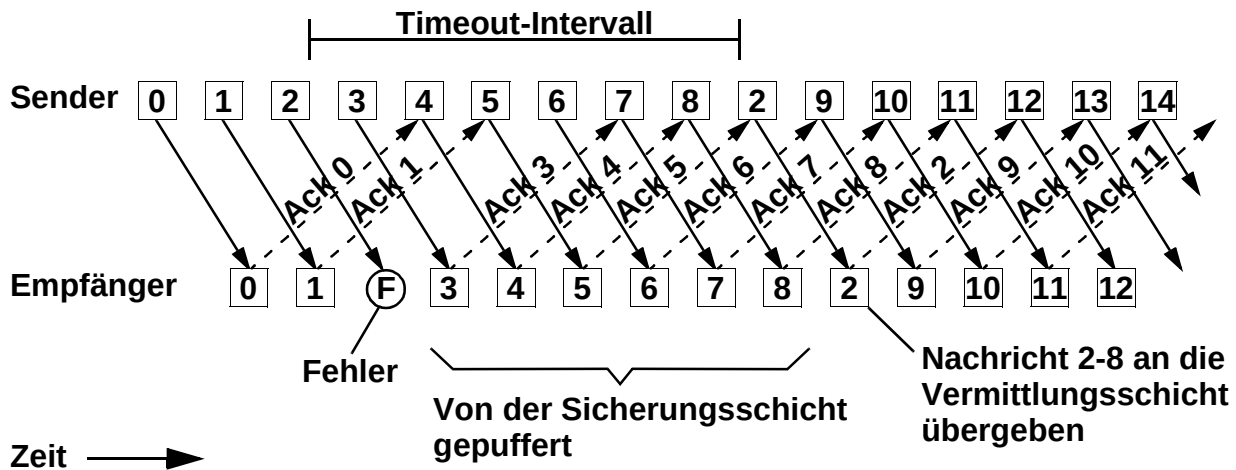
- Im Falle eines Fehlers bleibt das Ack aus. Der Empfänger puffert die danach noch korrekt erhaltenen Rahmen. Nach Ablauf des Timers beginnt der Sender, alle Rahmen ab dem Unbestätigten neu zu übertragen, bis er ein kumulatives Ack vom Empfänger erhält. Er macht dann weiter mit dem ersten noch nie übertragenen Rahmen.



- Die erneute Übertragung der Rahmen 7 und 8 kann durch die kumulative Bestätigung vermieden werden. Dieses Verfahren ist allerdings vergleichsweise kompliziert und wird in der Praxis kaum eingesetzt.
- Sammelquittung hat den Vorteil höherer Effizienz bei der Nutzung der Übertragungsbandbreite

Fehlerbehebung durch „selective repeat“

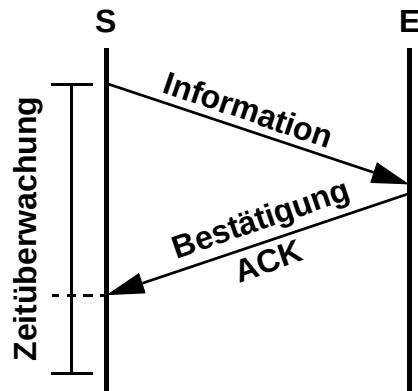
Der Empfänger bestätigt jeden korrekt erhaltenen Rahmen sofort. Er puffert alle korrekt erhaltenen Rahmen auch nach einem fehlerhaften Rahmen. Bei Ablauf des Timers überträgt der Sender nur den nicht bestätigten Rahmen neu.



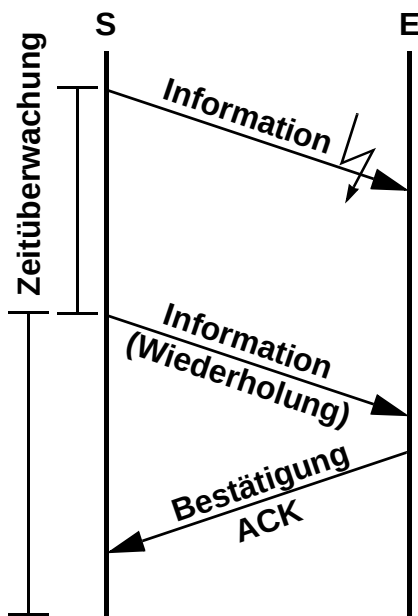
- jedes fehlerfreie Paket wird bestätigt (keine Sammelbestätigung)
- Nachteil: viele Bestätigungen
- Vorteil: gezielte Paketwiederholung = keine unnötige Wiederholung korrekt empfangener Pakete

Bisher: Passive Fehlerkontrolle

bisher: Kein Versenden von NACKs (Negative Acknowledges)



Fall 1: Übertragung ohne Übertragungsfehler



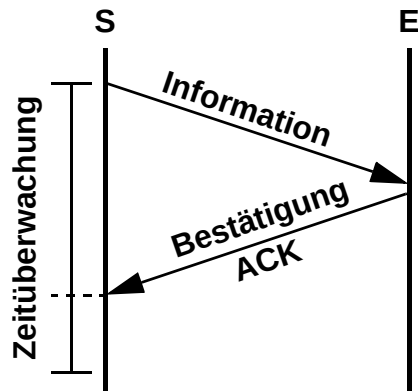
Fall 2: Datum kommt zwar an, aber mit Übertragungsfehler
=> Warten beim Sender bis zum Timeout

Nachteil der passiven Fehlerkontrolle:

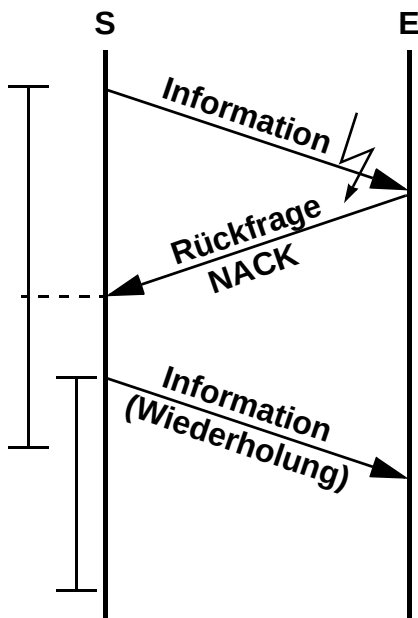
- Keine Unterscheidung zwischen fehlenden und fehlerhaften Blöcken
- Zeitverzögerung bis zur Wiederholung des Sendevorgangs
= unnötige Wartezeit beim Sender im Fehlerfall (ineffizient)

Neu: Aktive Fehlerkontrolle

- Versenden von NACKs (negative Bestätigungen)



Paketempfang ohne
Übertragungsfehler
=> Senden von ACK



Paketempfang mit
Übertragungsfehler
=> Senden von NACK
=> Datum wird vor Ablauf des
Timeout noch einmal gesendet

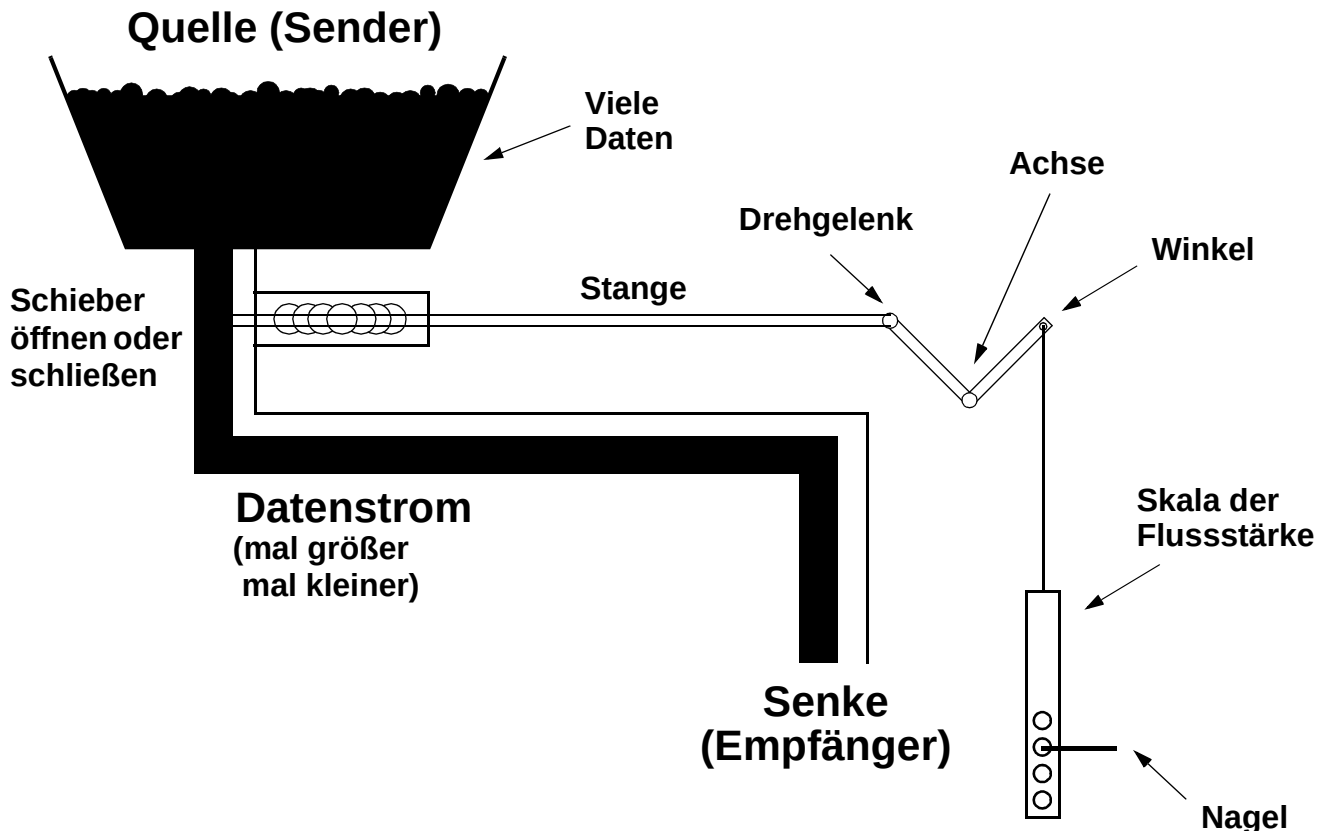
NACK = Negative Acknowledgement

- In Rechnernetzen wird sowohl die aktive als auch die passive Fehlerkorrektur verwendet.
- Wenn ein Paket unterwegs verloren geht, erhält der Sender weder ACK noch NACK => timeout des Senders wird aktiv.

Flusssteuerung

Das Prinzip der Flusskontrolle Flusssteuerung (to control = steuern)Flusssteuerung

Mechanische Analogie zur Flußsteuerung



Die Flußsteuerung stellt einen Rückkopplungsmechanismus zur Verfügung, um zu verhindern, dass der Sender den Empfänger mit Daten überschwemmt.

- Für die Flusssteuerung wird oft kein eigener Mechanismus eingesetzt.
- Vielmehr wird die Fehlererkennung über Acknowledges/ Sequenznummer „zweckentfremdet“.
- Acknowledges dienen dann zur Datensicherung und zur Flusssteuerung

=> Stop & Wait-Protokoll

Ein einfaches Stop-and-Wait-Protokoll zur Flusssteuerung benachbarter Rechner

Annahmen:

- Fehlerfreie Übertragung zwischen Sender und Empfänger
- Beschränkte Anzahl von Puffern (genau einer bei Sender und Empfänger)
- Verarbeitungsgeschwindigkeiten können sich unterscheiden (z. B. schneller Großrechner mit langsameren PC)
- Time out steht auf ausreichend hohem Wert

=> Benutze ACK für eine einfache Flusskontrolle (=> es ist nie mehr als ein Rahmen auf dem Weg).

ACK = Acknowledge = Bestätigung

- Die Bestätigung wandert vom Empfänger zum Sender zurück
- Als Bestätigung dient entweder ein spezielles Paket, das Acknowledge-Paket, oder eine spezielle Steuerleitung, die Acknowledge-Leitung
- Immer, wenn der Sender vom Empfänger eine Empfangsbestätigung erhalten hat, weiss er, dass er ein neues Paket senden darf.

=> Der Sender muss auf das Acknowledge warten = Stop & Wait-Protokoll

Beispiel für ein einfaches Stop-and-Wait-Protokoll

procedure sender;

var s: frame (*s= zu sender Rahmen *)
buffer: message (* buffer enthält zu send.Info*)
event: EvType (*Interrupt-Datentyp *)

begin

while (true) do (* Endlosschleife *)
begin (*Lese zu sendenden Puffer von der
Vermittlungsschicht *)
FromHost (buffer); (* Fülle Datenfeld des Rahmens *)
s. info = buffer; (*sende Rahmen ab)
sendfr (s) (*wait for ack*)
wait (event)
end;

end;

procedure receiver;

var r,s: frame; (*s = zu senden des Acknowledge r = zu
empfangen der Rahmen*)
Event: EvType;

begin

while (true) do
begin
wait (event); (* wait for frame arrival*)
getfr (r); (* lese Rahmen von der Bitübertragungs-
schicht*)
ToHost (r.info); (*schreibe Rahmen an die Sicherungs-
schicht*)
sendfr (s); (*send ack to sender*)
end;

end;

Sequenznummern

- Um Overhead zu sparen, werden kurze Sequenznummern (z. B. 3 Bit) übertragen

=> evtl. Pakete mit gleicher Sequenznummer unterwegs

- Bei TCP hingegen werden volle 32 Bit Sequenznummern verwendet

Flusssteuerung mit Schiebefenster

- Schiebefenster-Protokoll (sliding window) vermeidet Nachteile der vorangegangenen Protokolle und ist gleichzeitig effektiv hinsichtlich der Bandbreiteausnutzung
- TCP setzt das sliding window-Protokoll ein
- Schiebefenster-Protokoll beruht auf einem Ringpuffer bei Sender und Empfänger

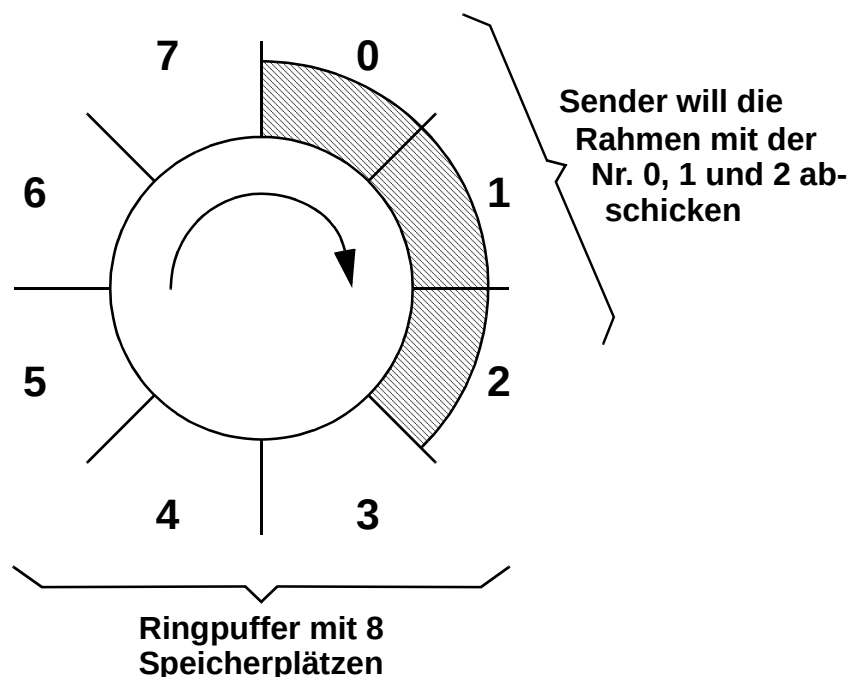
Der modernste Mechanismus: Flusskontrolle mit Schiebefenster (= sliding window)

Fenstermechanismus

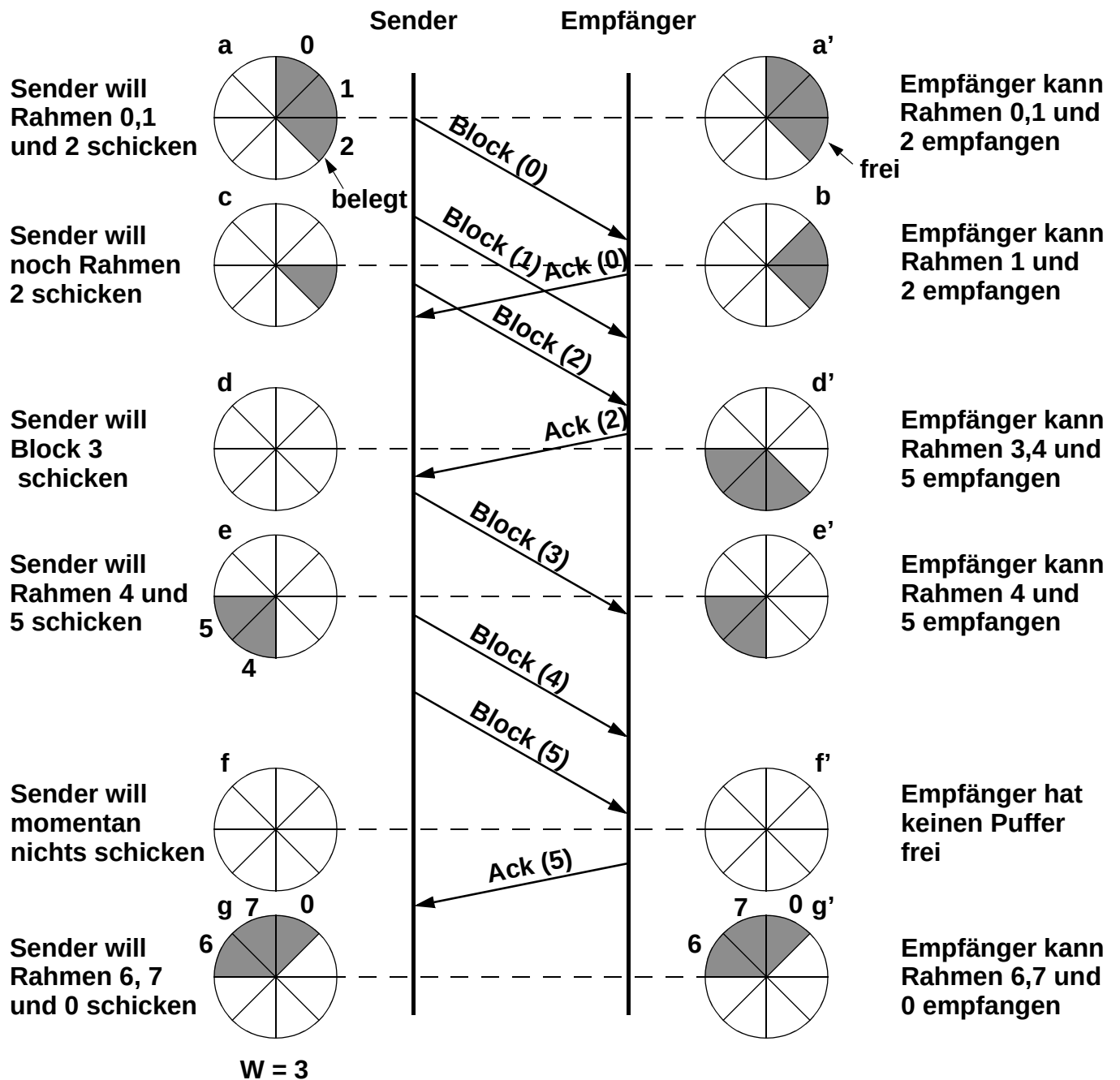
- Nach dem Verbindungsaufbau besitzt der Sender das Recht, so viele Informationsrahmen zu senden, wie durch die Fenstergröße vorgegeben ist. ($\leq$ Zahl der Sequenznummern)
- Spätestens dann muss vom Empfänger eine Bestätigung eintreffen, ansonsten unterbricht der Sender die Übertragung von Rahmen.
- Der Empfänger kann schon vor dem Erreichen der Fenstergröße Bestätigungen an den Sender übermitteln (Öffnen des Fensters).

Beispiel: Fenstergröße = 3 => Sender darf bis zu 3 Pakete ohne Bestätigung schicken

Zahl der Sequenznummern sei gleich 8 bei Sender und Empfänger



Es gibt immer zwei Schiebefenster (eins beim Senden und Empfangen)



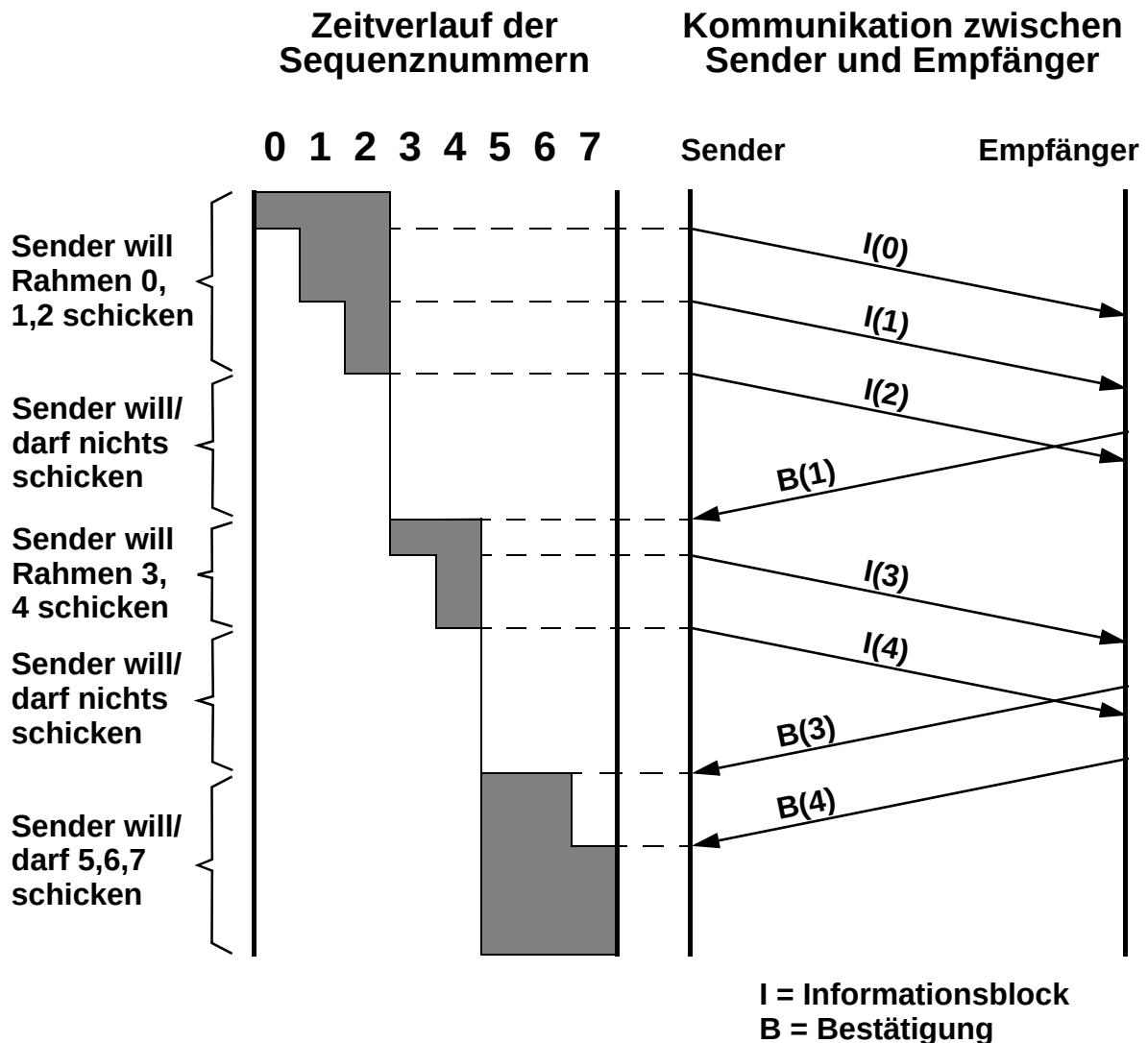
Sendeaktionen verschieben den hinteren Zeiger, ACKs den vorderen Zeiger.
=> Implementierung über Ringpuffer (= zirkular arbeitender Puffer)

Eintreffende Blöcke verschieben den hinteren Zeiger, versendete ACKs den vorderen Zeiger.
=> ACK heißt: Paket wurde vom Puffer an die Sicherungsschicht weitergeleitet

2. Beispiel für Schiebefenster

Fenstergröße = 3; Puffergröße = 8

- Öffnen und Schließen des Fensters des Senders

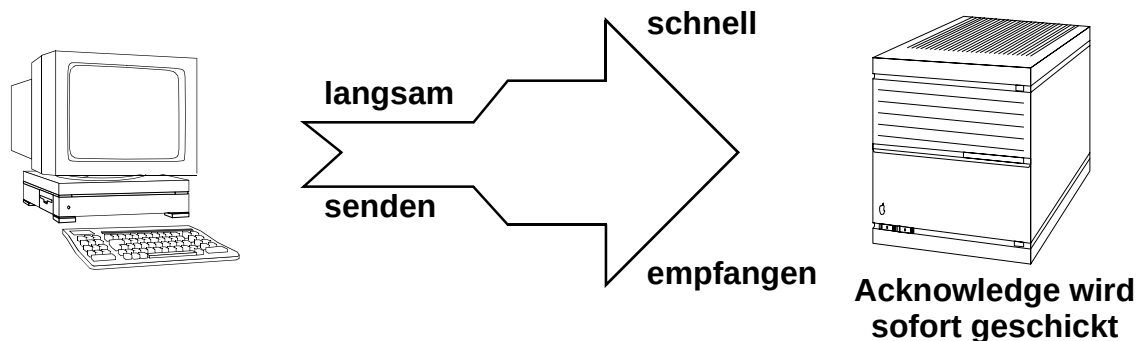


Die Sequenznummern werden sowohl zur Fehlerkontrolle als auch zur Flusssteuerung verwendet.

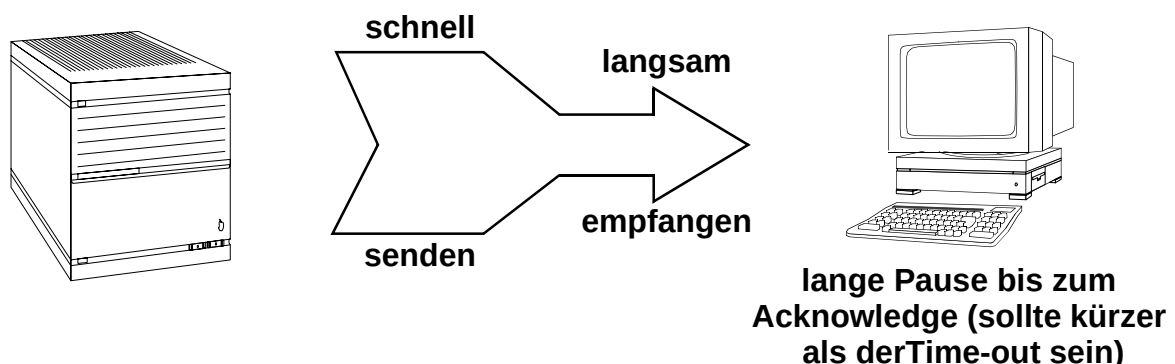
- Fehlerkontrolle: „Sind alle Blöcke da?“
- Flusssteuerung: „Sind die letzten Blöcke zu schnell gekommen?“

Wahl von Fenstergröße und Puffer beim Empfänger (1)

Beispiele für den Zusammenhang zwischen unterschiedlich leistungsfähigen Partnern, der Fenstergröße und der Anzahl zu reservierender Puffer beim Empfänger.



a) langsamer PC und schnelle Workstation: 1 Puffer ausreichend => stop & wait-Protokoll mit window size = 1



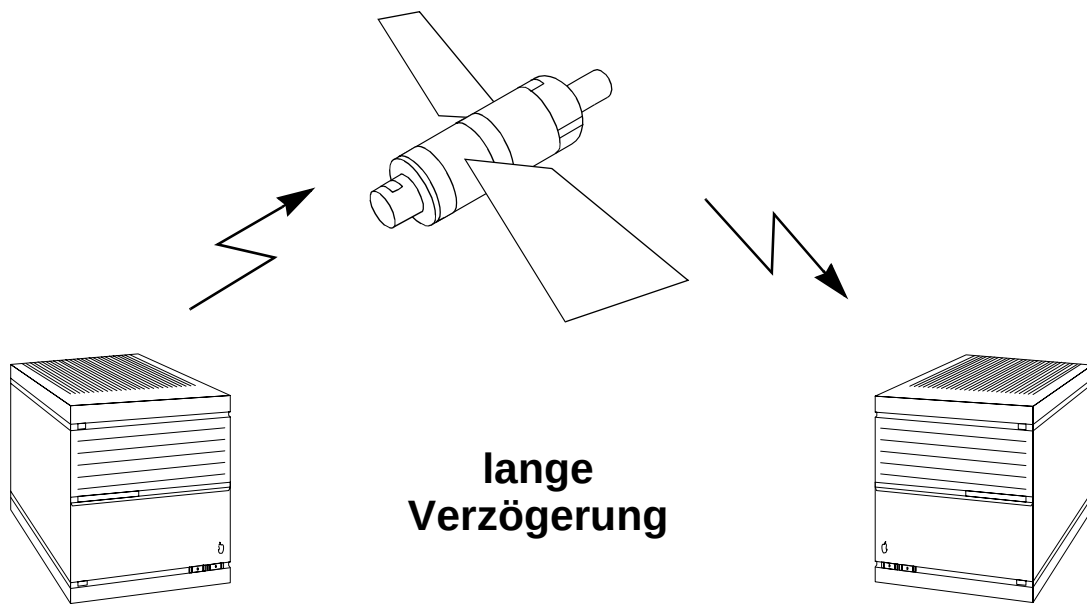
b) schnelle Workstation und langsamer PC: 1 Puffer ausreichend => stop & wait mit window size = 1

=> bei ungleichmäßig schnellen Kommunikationspartnern ist ein stop & wait-Protokoll ausreichend, sofern keine große Latenz zwischen beiden existiert.

- stop & wait-Protokoll heißt, dass ein Puffer der Größe 1 verwendet wird zusammen mit ACK

Fenstergröße und Puffer beim Empfänger (2)

- **Balanciertes Verhältnis (= beide gleich schnell), z. B. zwischen zwei Großrechnern: es sind w Puffer erforderlich, z. B.: $2 \leq w \leq 7$**
- **Beide Rechner gleich schnell und haben eine lange Kommunikationslatenz:**



- **Beispiel: Übertragung über einen Nachrichtensatelliten (= hohe Latenz): es sind w Puffer erforderlich, z. B. $16 \leq w \leq 127$**

Je größer die Verzögerung bei der Übertragung desto größer muss die Fenstergröße w gewählt werden, desto mehr Puffer werden auf der Empfängerseite benötigt. Der Empfänger benötigt stets mindestens w Puffer.

- **Idealerweise ist der Puffer bei Sender und Empfänger gleich groß, sonst wird Pufferplatz vergeudet.**

Protokolle der Sicherungsschicht

Beispiel: HDLC von IBM

= **Das wichtigste Protokoll der Sicherungsschicht.**
Eng verwandt sind:

- HDLC High-Level Data Link Control (nach ISO)
- LLC 2 Logical Link Control Typ 2 in LANs (nach IEEE 802)
- LAP-B Link Access Procedure-Balanced
(nach CCITT = Schicht 2 von X.25)

HDLC Standards

- ISO 3309 HDLC procedures - **Frame structure**
(Rahmenaufbau)
- ISO 4335 HDLC procedures - Elements of **procedures**
(einleitende Worte zur Ablaufbeschreibung)
- ISO 6159 HDLC **unbalanced** classes of procedures
- ISO 6256 HDLC **balanced** class of procedures

sowie X.25, Level 2

Link Access Procedure Balanced (LAP B)
(Schicht 2 der CCITT-Empfehlung X.25;
Schicht 2 = Sicherungsschicht)

X.25 = älteres Protokoll der Schichten 1-3 für
paketvermittelnde Netze der Telekom-
Unternehmen (weltweit verwendet, zum Inter-
net inkompatibel)

- **HDLC ist der Vater vieler Protokolle der Sicherungsschicht, z. B. von PPP**

PPP = point-to-point-protocol zum Einwählen in ein Rechenzentrum über Modem

oder X.25 = Protokoll für öffentliche Paketvermittlungsnetze aus den 1920er-Jahren

HDLC - High Level Data Link Control = Protokoll der Sicherungsschicht

Basis-Rahmenformat (= Paketformat):



Rahmenkennung durch Bitstopfen (Bit Stuffing)

Frame Check Sequence (FCS):

Cyclic Redundancy Check mit

$$G(x) = x^{16} + x^{12} + x^5 + 1 \text{ (= CCITT-CRC)}$$

**Flag = „Kennzeichnen“ für Rahmenbeginn und
Rahmenende = frame delimiter**

„Prozedurklassen“ in HDLC

1.) Asymmetrische Konfiguration (unbalanced procedures)

a.)

- **Primärstation:** Management der Verbindung (Großrechner)
- Eine oder mehrere **Sekundärstation(en)** (Terminals)
- **Punkt-zu-Punkt-Verbindungen** **und** Mehrpunktverbindungen (**Multicast**)

b.)

- **Normaler Antwortmodus** (Normal Response Mode): Eine Sekundärstation kann nur nach einem Sendaufruf („polling“) durch die Primärstation senden.

c.)

- **Asynchroner Antwortmodus** (Asynchronous Response Mode): Die Primär- und die Sekundärstation kann übertragen, wenn die Leitung frei ist.

2.) Symmetrische Konfiguration (Balanced procedures)

- Totale Symmetrie zwischen den Stationen (Rechner-Rechner-Kopplung)
- nur Punkt-zu-Punkt-Verbindungen (keine Mehrpunktverbindungen)
- **nur asynchroner Modus** (Asynchronous Response Modus) = jeder kann senden, wenn Leitung frei ist.

Drei Prozedurklassen bei HDLC

- **Unbalanced Operation im Normalen Antwortmodus**
oder
- **Unbalanced Operation im Asynchronen Antwortmodus**
oder
- **Balanced Operation im Asynchronen Antwortmodus**

Bemerkung:

Asynchronous Balanced Mode ist die Grundlage von LAP-B (Link Access Procedure Balanced), der Schicht 2 des X.25-Standards.

Aufbau des HDLC-Adressfelds

- Ursprünglich 8 Bits lang, diente zur Adressierung der Sekundärstation beim Polling (= Terminaladresse)
- Erweiterte Adressierung
Die Größe des Adressfelds kann ein Vielfaches von 8 Bits sein (zur Adressierung von Geräten (ursprünglich Terminals) in beliebiger Zahl)

Aufbau des HDLC-Steuerfelds (Control Field)

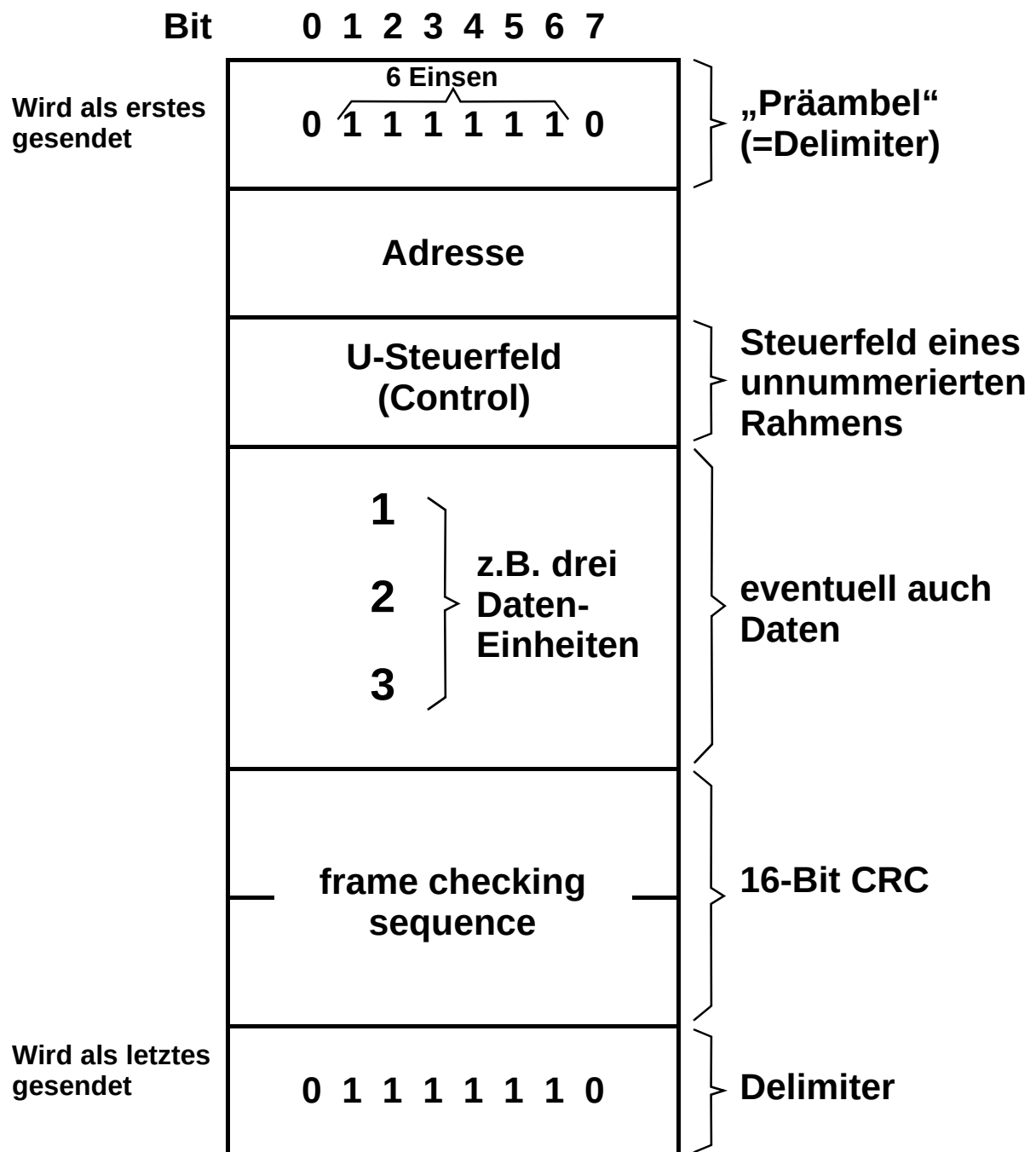
Es gibt drei verschiedene Rahmentypen im HDLC:

- I-Rahmen (information frames)
werden zur Datenübermittlung benutzt
- S-Rahmen (supervisory frames)
werden zur Steuerung des Datenflusses benutzt und zur Datensicherung
- U-Rahmen (unnumbered frames)
steuern die Verbindung (Aufbau und Abbau)

Der Rahmentyp wird im Steuerfeld („control“) angegeben.

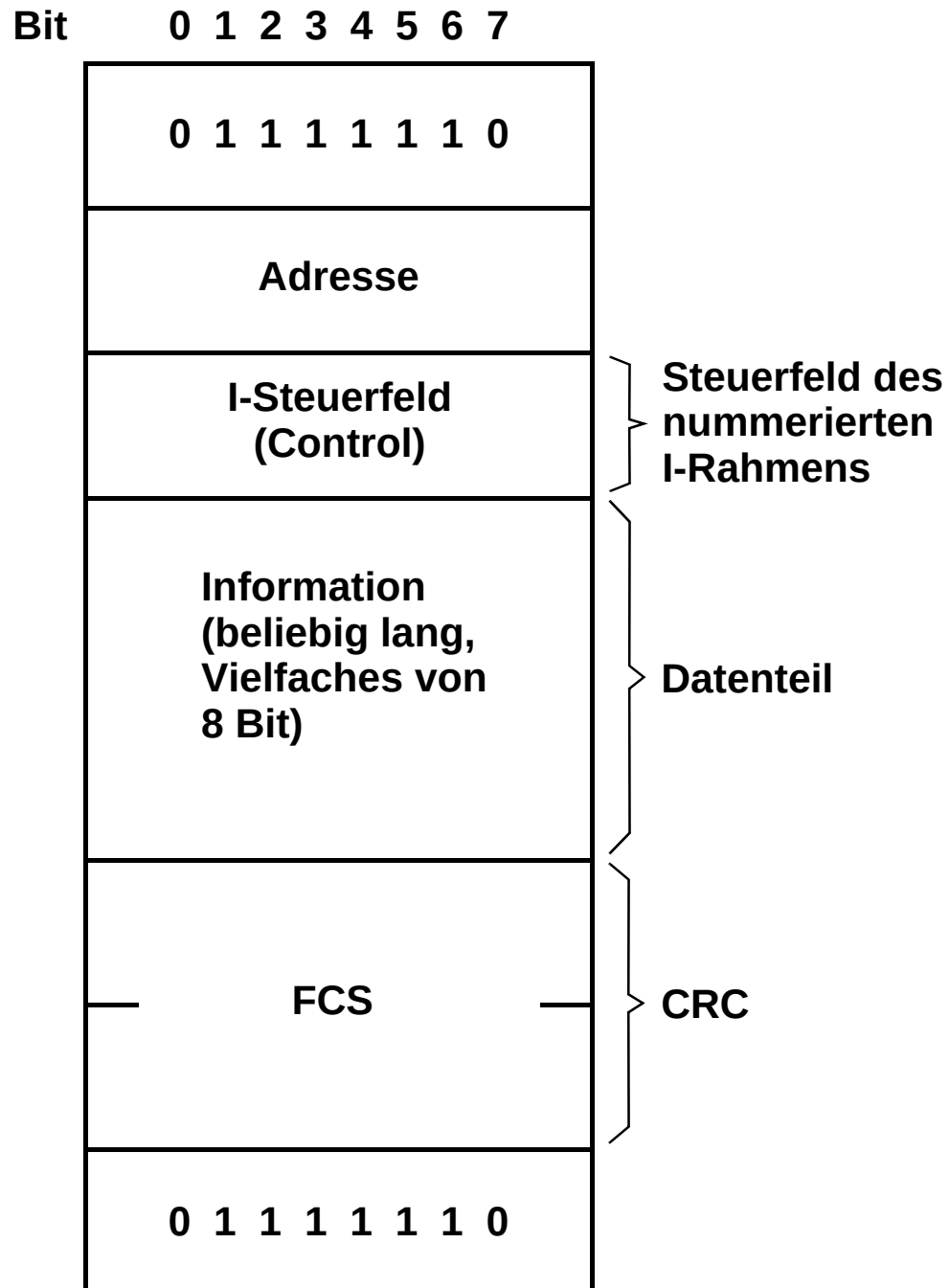
HDLC-Rahmenformate (1) (U-Rahmen)

= ohne Sequenznummer (unnumbered) format zur Steuerung des Verbindungsaufbaus



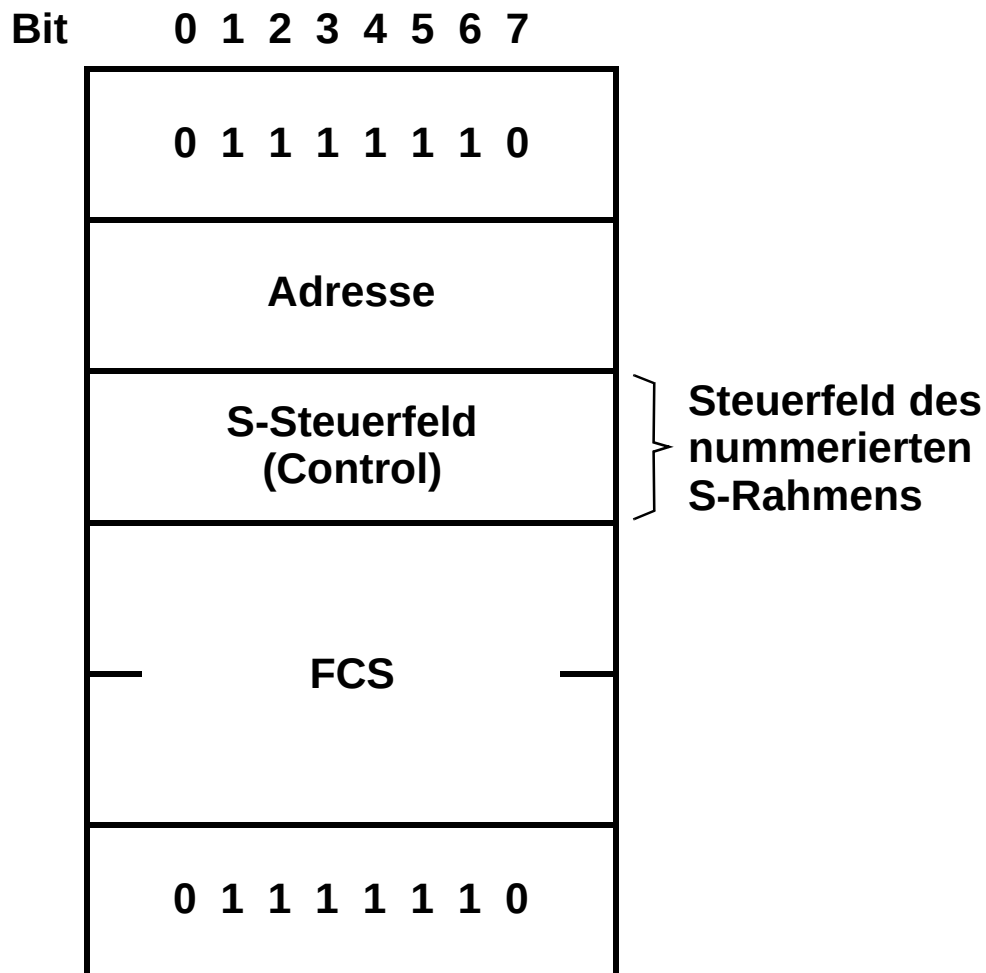
HDLC-Rahmenformate (2) (I-Rahmen)

= mit Datenfeld > 0 (information frame),
dient zur eigentlichen Datenübertragung



HDLC Rahmenformate (3) (S-Rahmen)

= supervisory frame, zur Flusssteuerung und Datensicherung zw. benachbarten Rechnern



Unterscheidung zwischen Command/Response im Unbalanced Mode

Command	Rahmen eines beliebigen Typs, der durch eine <u>Primärstation</u> (Mainframe) gesendet wird
Response	Rahmen eines beliebigen Typs, der durch eine <u>Sekundärstation</u> (Terminals) gesendet wird

Bemerkung:

Kombinierte Stationen, die zugleich Primär- und Sekundärstation sind, können sowohl Commands als auch Responses senden (= Balanced Mode).

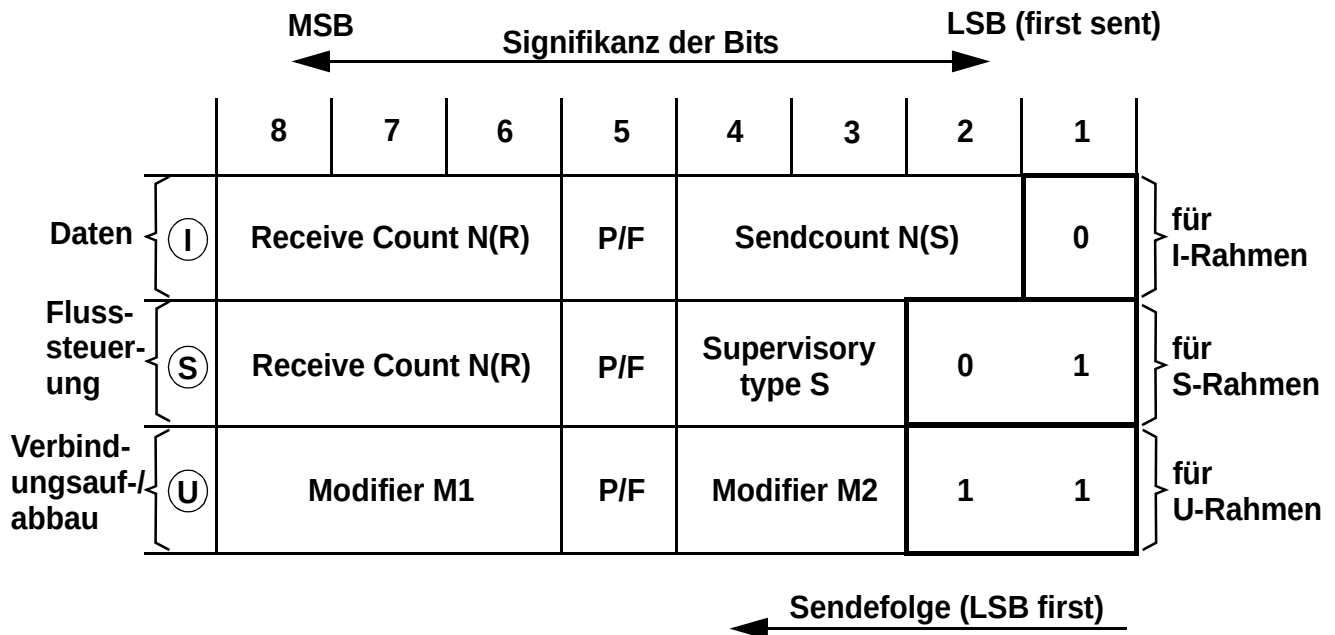
Spezielles Bit im Steuerteil von U-, I- und S-Rahmen

P/F-Bit: Poll/final-Bit in Command Rahmen.

Der Empfänger muss diesen Rahmen durch einen Response-Rahmen mit gesetztem Final-Bit bestätigen.

- Im unbalanced mode u. normalem Antwortmodus setzt der Rechner das Poll-Bit. Dann kann das Terminal antworten (response).
- Das Terminal setzt bei seiner Antwort das final-Bit, wenn es sich um den letzten Rahmen seiner Antwort handelt.

Inhalt des Steuerfelds (= control field) im Basis-Format von HDLC



Für I-Rahmen gilt:

HDLC benutzt ein Schiebefenster mit einer 3-Bit-Sequenznummer und Fenstergröße = 7

N(R) = Empfangszähler (= Sequenznummer des Acknowledge)

N(S) = Sende-zähler (Sequenznummer des Rahmens)

Huckpackverfahren bei Vollduplex

N(R) und N(S) sind Zähler für die Anzahl der I-Rahmen, die empfangen und gesendet wurden.

N(R) ist ein piggybacked acknowledgement, das zusätzlich im S-Rahmen enthalten ist

N(R) enthält die Nummer des ersten nicht empfangenen Rahmens (= Nummer des nächsten erwarteten Rahmens); kann im I-Rahmen des Rückkanals übertragen werden. = Ack (Seq.-Nr. + 1)

piggy back = Huckepack

Supervisory-Rahmen (1) bei HDLC (zur Flusssteuerung und Datensicherung)

2 Bits zur Unterscheidung => 4 verschiedene S-Rahmen (= Supervisory Type)

- 1.) Receive Ready (RR) (= Bestätigung, ACK) = positive Acknowledge
Station kann I-Rahmen empfangen
N(R): nächster erwarteter Rahmen (steht im I-Rahmen) als Huckepack-Info
Dient zu Flusssteuerung und Datensicherung
- 2.) Receive Not Ready (RNR)
Eine Station zeigt der anderen an, dass sie zeitweise nicht empfangen kann. Sender stoppt die Übertragung
Nur für Flusssteuerung)
- 3.) Reject (REJ) (entspricht einem NACK)
Zeigt an, dass ein Übertragungsfehler entdeckt wurde
Nur Datensicherung

Für N(R) gilt bei Reject:

N(R): erster Rahmen in der Folge der nicht korrekt empfangen wurde, also der Rahmen, der erneut gesendet werden soll (steht im I-Rahmen als Huckepack-Info)

Sender: Muss alle ausstehenden Rahmen erneut übertragen, beginnend mit N(R).

= Datensicherung ohne Pufferung und mit Go-back-N-Protokoll, jedoch nicht mit time-out Aktivierung, sondern mittels neg. Ack.

Supervisory-Rahmen (2) **(zur Flusssteuerung und Datensicherung)**

4.) Selective Reject

Zeigt dem Sender an, dass ein Übertragungsfehler entdeckt wurde. Fordert nur die erneute Übertragung des Rahmens spezifiziert durch N(R) an.

= Datensicherung durch selective repeat-Protokoll, jedoch nicht mit time-out, sondern durch Selective Reject-Rahmen.

Reject und Selective Reject werden nur bei vollduplex Übertragung eingesetzt.

Bemerkung:

Beim LAP--Protokoll von X.25 sind alle Mechanismen bis auf Selective Reject implementiert

Unnumbered-Rahmen (1) (zum Verbindungsauf-/abbau)

5 Mode-Bits verfügbar

=> 32 verschiedene U-Rahmen sind möglich (nicht alle werden benutzt).

- Zum Setzen der Modi für die drei Prozedurklassen werden folgende Kommandos benutzt:
 - Set Normal Response Mode
 - Set Normal Response Mode with Extended Addressing
 - Set Asynchronous Response Mode
 - Set Asynchronous Response Mode with Extended Addressing
 - Set Asynchronous Balanced Mode
 - Set Asynchronous Balanced Mode Extended
- Sie werden benutzt, um eine Verbindung auf der Ebene der Sicherungsschicht herzustellen (Verbindungsaufbau)
- $N(S), N(R) \rightarrow 0$
- Erweiterte Version („Extended“):
Benutzt für das erweiterte Format des Steuerfeldes (Control Field)

Lokale Netze (LANs)

- .Ethernet (CSMA/CD)**
- .Token Ring**

Was ist ein LAN?

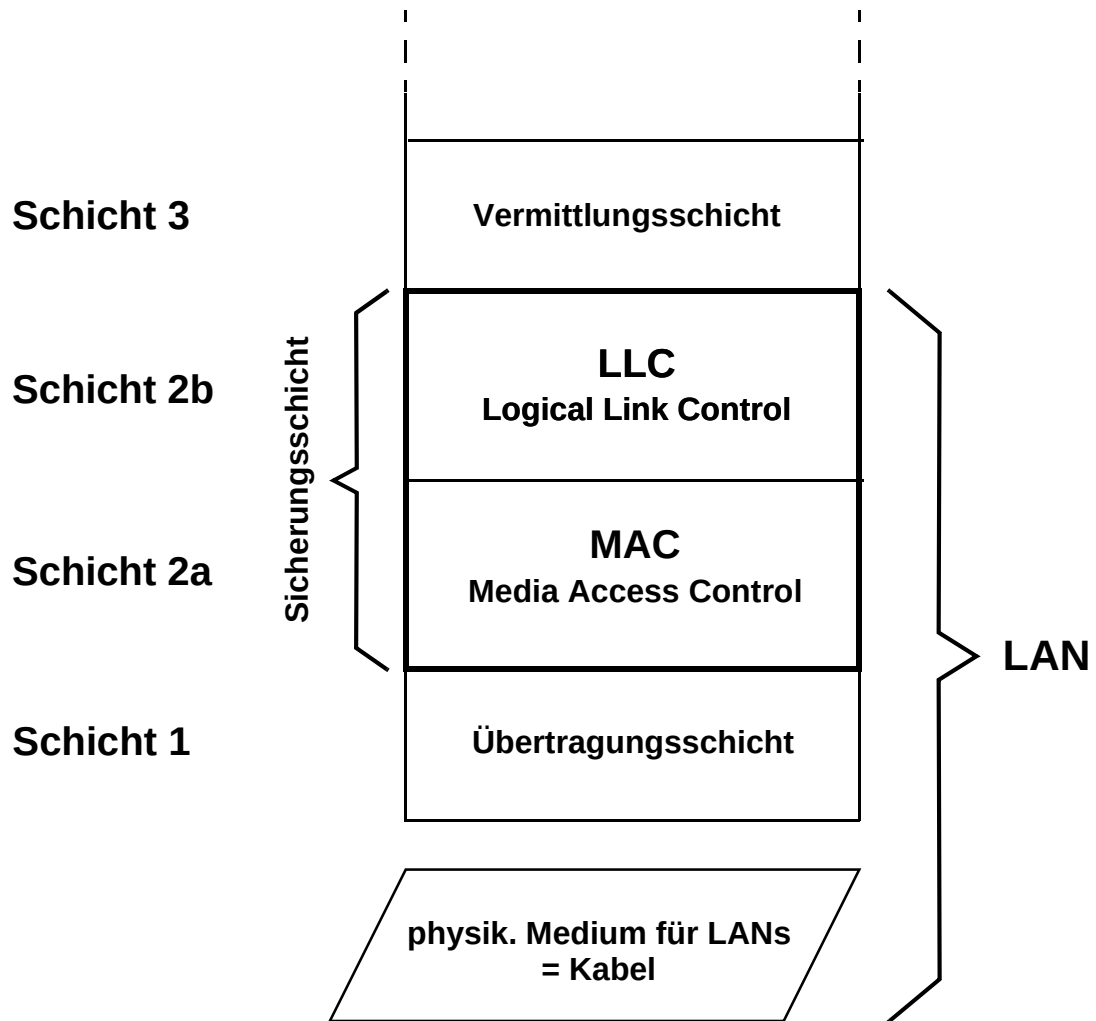
Ein LAN (Local Area Network) ist ein Netzwerk für (bitserielle) Übertragung von Informationen zwischen unabhängigen, untereinander verbundenen Stationen.

Es befindet sich rechtlich unter der Kontrolle des Benutzers und ist auf den Bereich innerhalb der Grundstücksgrenzen beschränkt.

Merkmale eines lokalen Netzes

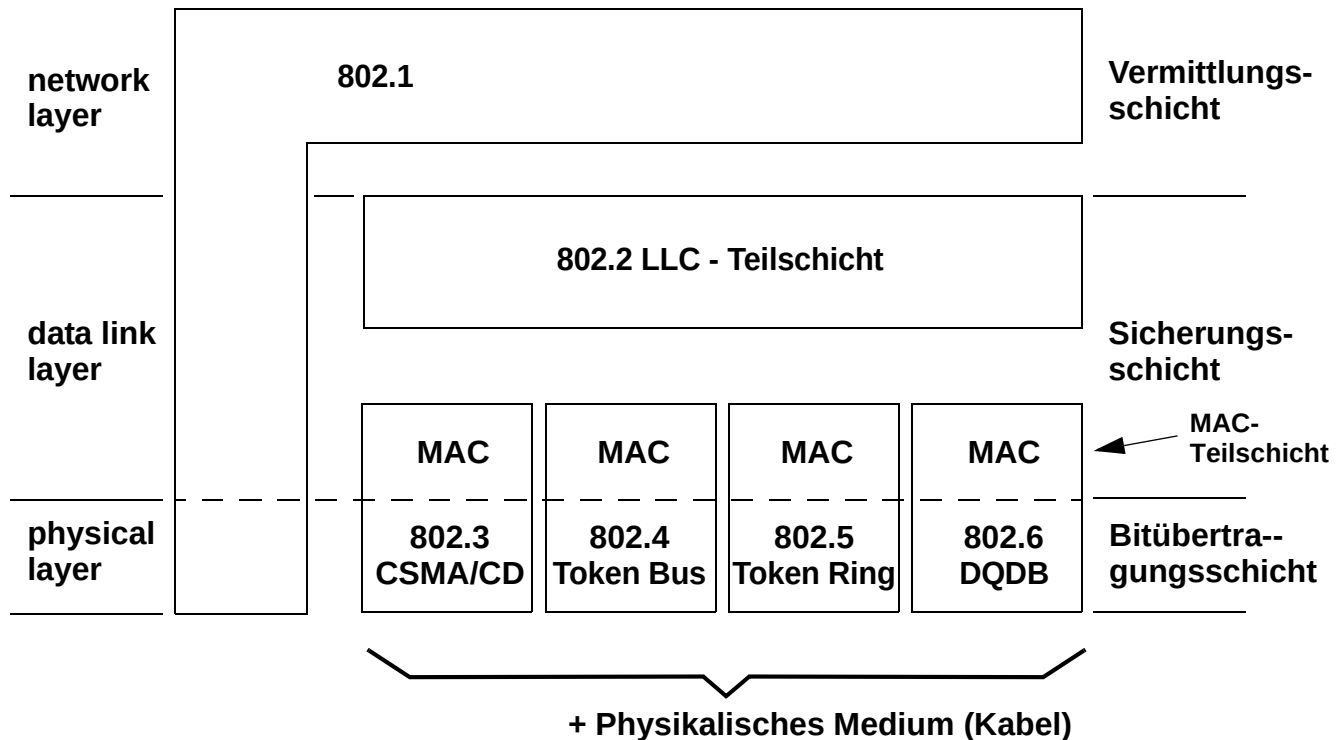
- Hohe Geschwindigkeit (10-100 MBit/s - 1 GBit/s)
- Leichter, kostengünstiger Anschluss
- Keine Postregulierungen, da auf eigenem Grundstück
- Anschluss unterschiedlicher Geräte, wie
 - PCs
 - Workstations
 - Großrechner
 - Drucker und andere periphere Geräte
- Übergang auf Weitverkehrsnetze über Router und Gateways

LANs im ISO-Referenzmodell



Die Sicherungsschicht im LAN nach IEEE 802

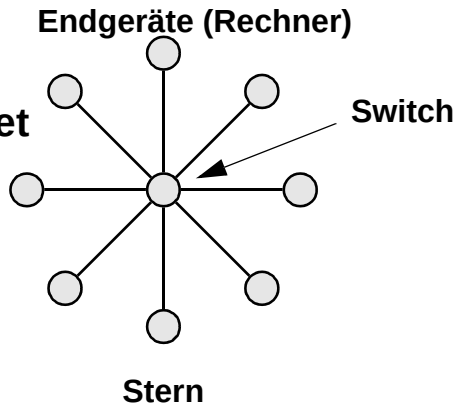
- Standardisierung bei IEEE und ISO



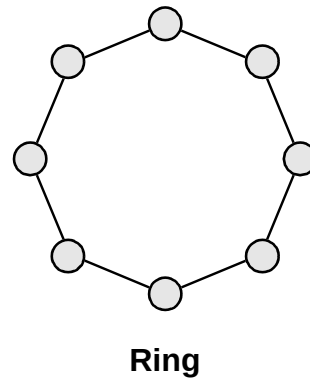
802.1 = Übersicht über die 802-Norm
802.2 = oberer Teil der Sicherungsschicht
= Logical Link Control Protocol (LLC)
802.3 = Ethernet-Norm (CSMA/CD)
MAC = unterer Teil der Sicherungsschicht
= Medium Access Control

Topologien für lokale Netze

**Fast Ethernet
(100 Mb/s)**

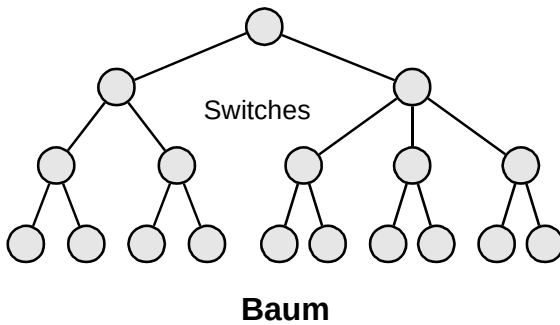


nur Rechner ohne Switch

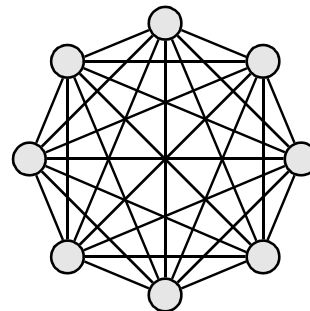


**Token
Ring**

Endge-
räte {

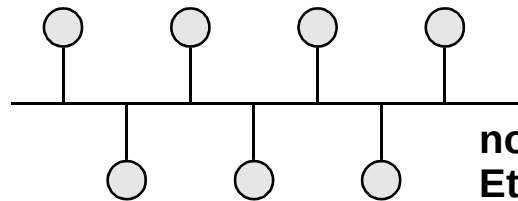
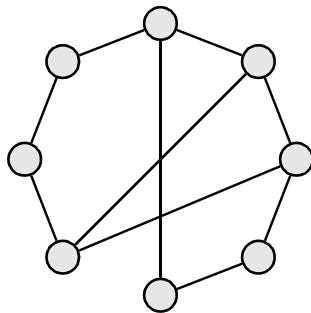


nur Rechner, kein Switch



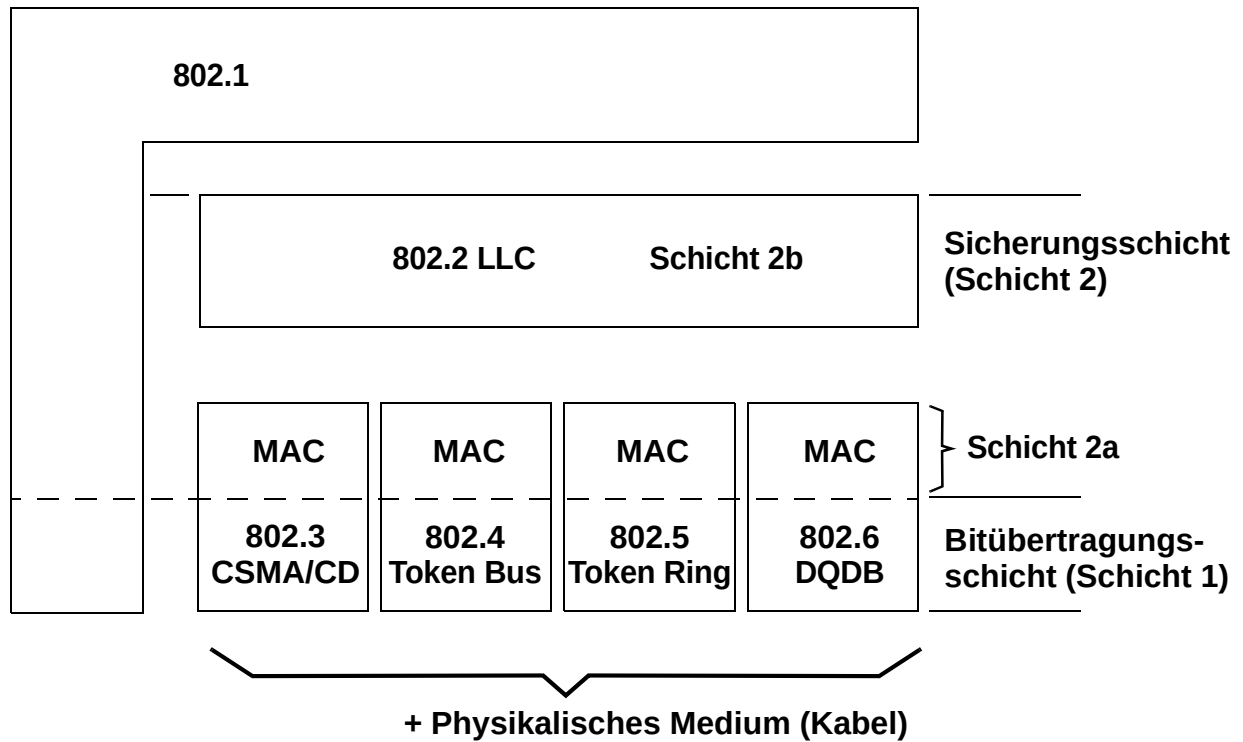
**nicht für
LANs**

nur Rechner



**gemeinsam genutztes
Übertragungsmedium**

IEEE Standard 802.x



- **CSMA/CD = Ethernet**

Schicht 2a bei LANs

Zugang zum gemeinsamen Kabel

Medium Access Control (MAC) = Schicht 2a

Problem

- Broadcast-Medium (gemeinsam genutzt)
- Unabhängige Stationen fangen an gleichzeitig zu senden
=> Sendekollisionen

Wie werden die Kollisionen geregelt?

„Medienzugangssteuerung“ (access control)

Zwei Medienbelegungsprinzipien:

- 1.) **Kollisionsentdeckung** (so verfährt Ethernet)
Lasse Kollisionen stattfinden, entdecke sie, wiederhole Übertragung.
- 2.) **Kollisionsverhinderung** (so verfährt Token Ring)
Verwende ein zirkulierendes **Token (= Sendeberechtigungsrahmen)**, um den Zugriff auf das Medium zu steuern.

Token = „Zeichen“, ist bei Token Ring ein spezieller Rahmen, der den Zugriff auf das gemeinsame Kabel gewährt.

1. Beispiel für ein LAN **Ethernet (CSMA/CD)**

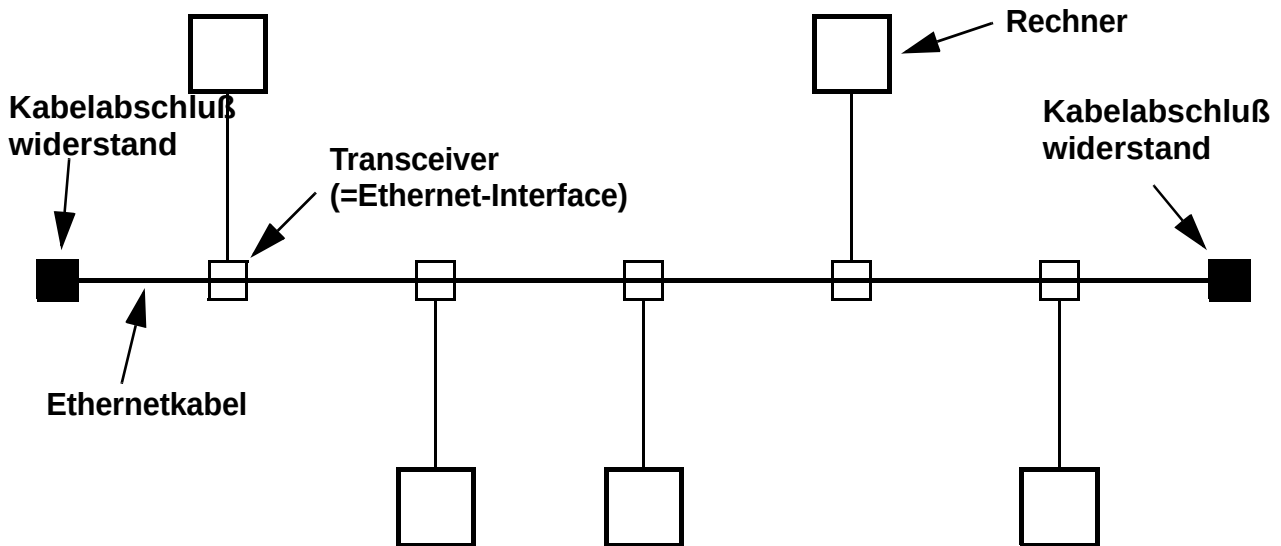
**CSMA/CD = Carrier sensing for multiple access with
collision detection**

**= Träger abhören für mehrfachen Zugriff
mit Kollisionserkennung**

CSMA/CD: Topologie und Technologie

- CSMA/CD = Carrier Sense Multiple Access with Collision Detection
- Standard: ISO IS 8802/3: MAC- und Bitübertragungsschicht für CSMA/CD

Topologie ist linear gemeinsames Medium

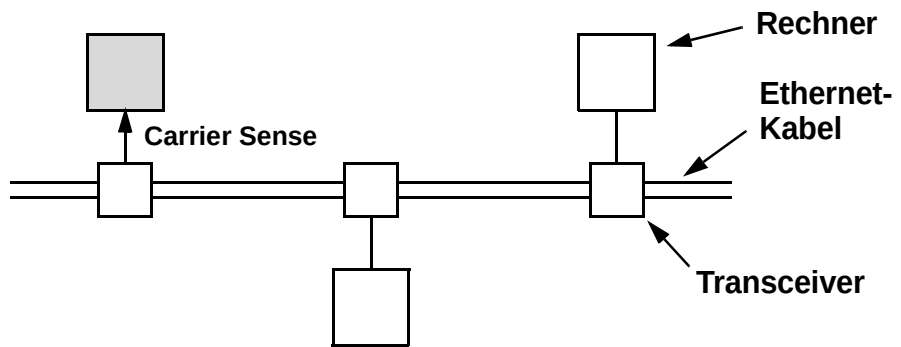


Eigenschaften

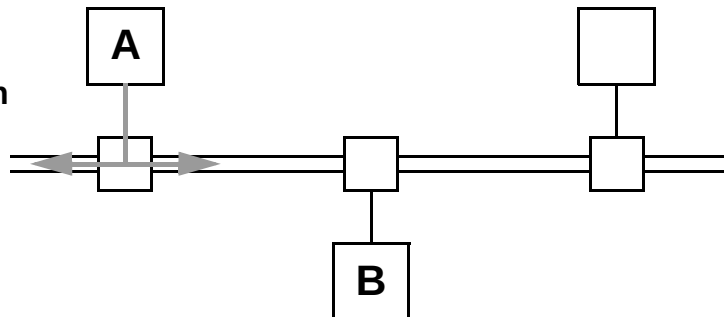
- Bidirektionaler Datenfluss zwischen beliebigen Rechnern
- Kurze Leitungsführung zwischen Transceiver und Rechner
- Bei Kabelbruch u. U. Ausfall des ganzen LANs
- Es gibt Topologische Einschränkungen bezüglich der Anschlusspositionen der Transceiver. Sie müssen in 2,5 m - Raster sein (gilt nur für das 10Base5-Ethernet).

CSMA/CD: Protokoll

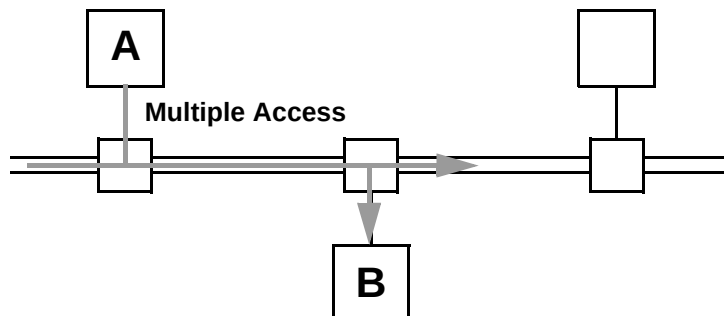
Lauschen, ob
Kabel frei ist



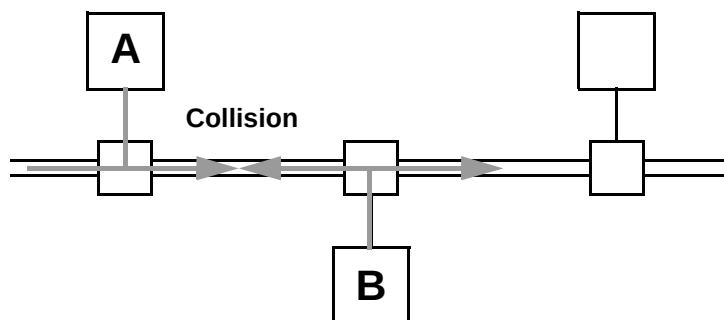
Kabel ist frei:
Station sendet von
 $A \rightarrow B$



2. Station B
empfängt nur



Kollision, wenn
mehr als zwei
Stationen
gleichzeitig
senden



CSMA/CD (Ethernet)

Verfahren

(auch „listen before talk“ genannt)

Die sendewillige Station hört das Medium ab:

- **Falls belegt, wird das Senden zurückgestellt**
- **Falls frei, wird mit dem Senden begonnen**
- **Zusätzlich gibt es Spezialregelung für sog. „Kollisionen“**

Voraussetzung, dass CSMA funktioniert:

- 1.) Stationen können sich gegenseitig „hören“.
- 2.) Die Rahmenübertragungszeit muss größer sein als die doppelte maximale Laufzeitverzögerung im Kabel ($\geq 2T$, T-Laufzeit im Kabel von einem Ende des Kabels zum anderen)

Kollisionen

Beginnen zwei oder mehr Stationen gleichzeitig mit dem Senden, so tritt eine Kollision auf.

- Nachdem ein Sender eine Kollision erkannt hat, hört er auf zu senden und wartet eine Zeit ab, die von einer Zufallszahl bestimmt wird, bevor er wieder zu senden versucht.
- Ein Sender kann mehrfach hintereinander das Schicksal haben, dass seine Sendung mit einer anderen Sendung kollidiert. (= mehrfache Kollision desselben Pakets)
- Nach i Kollisionen ($1 < i < 10$) wird eine Zufallszahl aus dem Intervall $0, 1, 2, \dots, 2^i - 1$ gewählt.
- Für $11 \leq i \leq 15$ wird die Zufallszahl aus dem konstanten Intervall $0, 1, 2, \dots, 2^{10} - 1$ gewählt.
- Für $i = 16$ wird der Sendeversuch abgebrochen und eine Fehlermeldung erzeugt.

=> Das Intervall für die Zufallszahl wird nach jedem erfolgreichen Senden verdoppelt, bis $i = 10$ erreicht ist.

=> Der Mittelwert der wartezeit verdoppelt sich nach jeder Kollision des Senders.

Dies wird als „exponential backoff“-Strategie bezeichnet.

Vermeiden von Kollisionen

Idee:

Erhöhte Kollisionsgefahr besteht nach Abschluss einer Übertragung: mehrere sendewillige wartende Stationen können bei der Feststellung „Medium frei“ gleichzeitig zu senden beginnen. Daher wird ein besonderer Algorithmus für das Sendebeginn-Verhalten nach Abschluss einer laufenden Sendung definiert (d. h., wenn das Medium vom Zustand „belegt“ in den Zustand „frei“ übergeht).

Bei Ethernet wird der relativ einfache „1-persistent-Algorithmus“ verwendet. Andere CSMA/CD-Netze können andere Strategien haben, z. B.

„p-persistent“ oder

„non-persistent“

persistent = hartnäckig

1-persistent = hartnäckig mit Wahrscheinlichkeit 1

CSMA/CD: Wiederholungsstrategien bei belegtem Medium

1.) non-persistent

Station wartet eine zufällig berechnete Zeitspanne („backoff time“) und startet dann einen neuen Übertragungsversuch => Station reagiert nicht auf das Freiwerden des Mediums

2.) 1-persistent = Ethernet

Station hört das Medium ab und startet die eigene Übertragung sofort nach Abschluss der laufenden Übertragung (Sendewahrscheinlichkeit = 1)

3.) persistent ($0 < p < 1$)

Station hört das Medium ab. Nach Ende der laufenden Übertragung sendet sie ihre Daten mit einer vorher festgelegten Wahrscheinlichkeit p oder wartet mit Wahrscheinlichkeit $1 - p$ eine festgelegte Zeitspanne.

- gesucht: Kompromiss zwischen

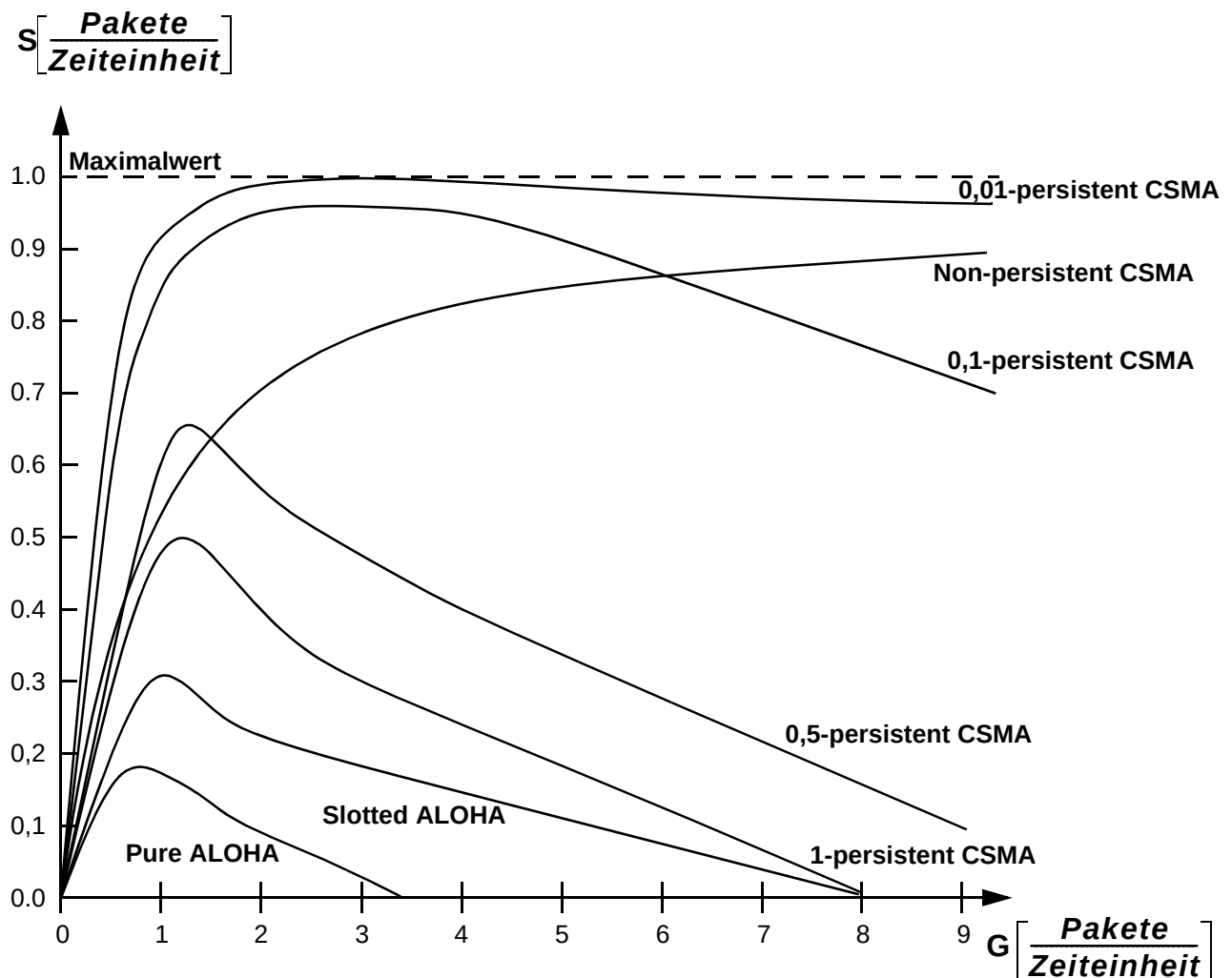
- a.) Minimierung der Anzahl Kollisionen
- b.) möglichst permanente Nutzung des Mediums

=> IEEE/ISO CSMA/CD: 1-persistent

Wie gut sind die verschiedenen Wiederholungsstrategien?

- Klärung dieser Frage erfolgt durch statistische Modellierung des lokalen Netzes
- Wesentliche Größen der Modellierung sind der Gesamtverkehr des lokalen Netzes sowie die Zahl der erfolgreichen d. h. kollisionsfreien Pakete pro Zeiteinheit
- Def. S = mittlerer Durchsatz des Netzes
= Erwartungswert der erfolgreichen Paketübertragungen pro Zeiteinheit
- G = mittlerer Gesamtverkehr des Netzes
= Erwartungswert aller Paketübertragungen pro Zeiteinheit (neue Pakete + Wiederholungen)
- S und G sind die Parameter zweier Poissonprozesse, mit denen das Netz statisch modelliert wird
(=> Warteschlangentheorie)
- Als Zeiteinheit wird die Zeit gewählt, die ein Paket zur Übertragung benötigt (bei ungestörter Übertragung)

Ausnutzung der Kananbandbreite bei steigender Last bei verschiedenen Wiederholungsstrategien



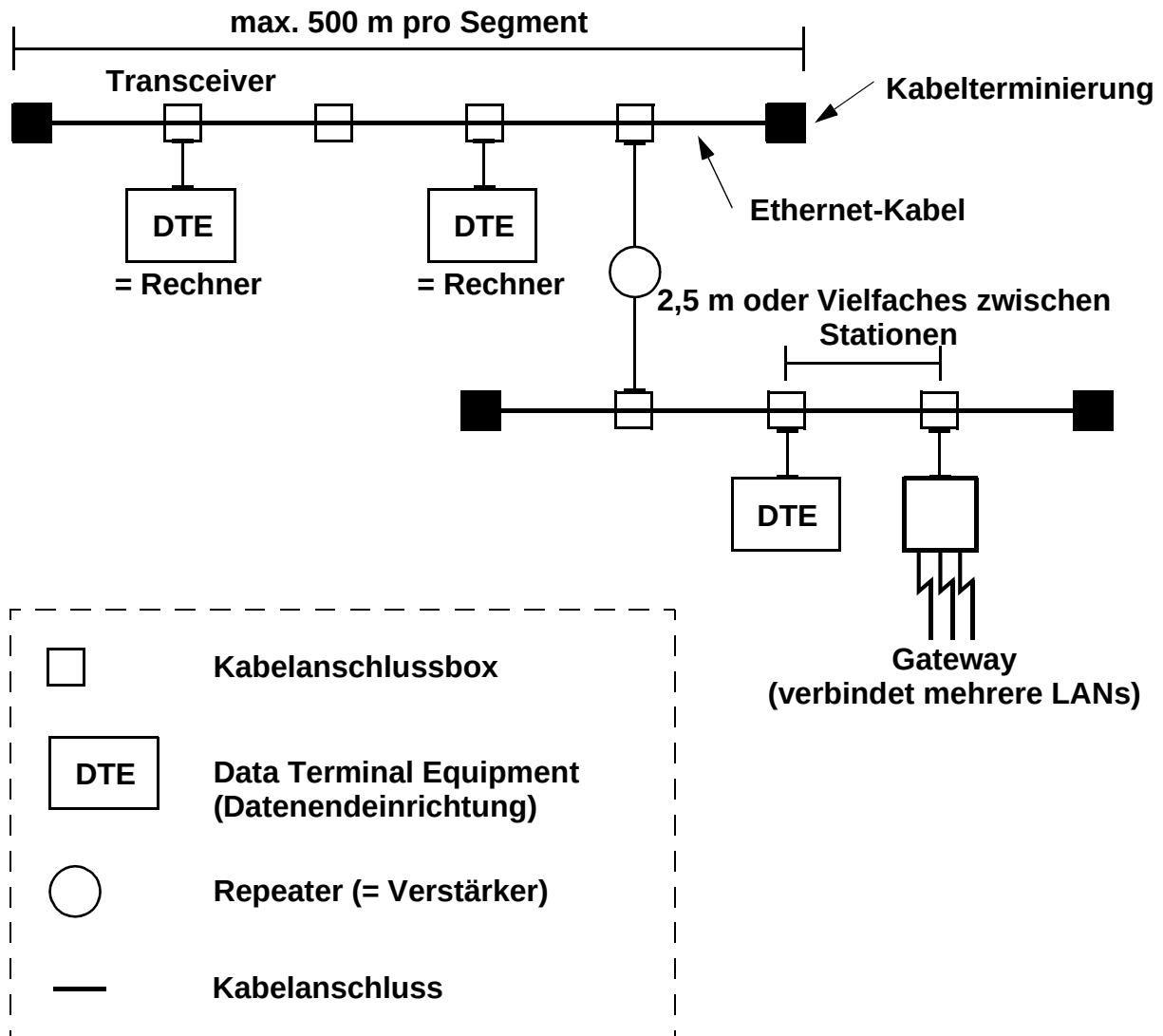
pure ALOHA = Station sendet sogar, wenn das Kabel bereits belegt ist.

slotted ALOHA = wie pure ALOHA, aber Sendebeginn nur zu diskreten Zeitpunkten (slots)

CSMA/CD

Implementierung (Ethernet 10 Base 5)

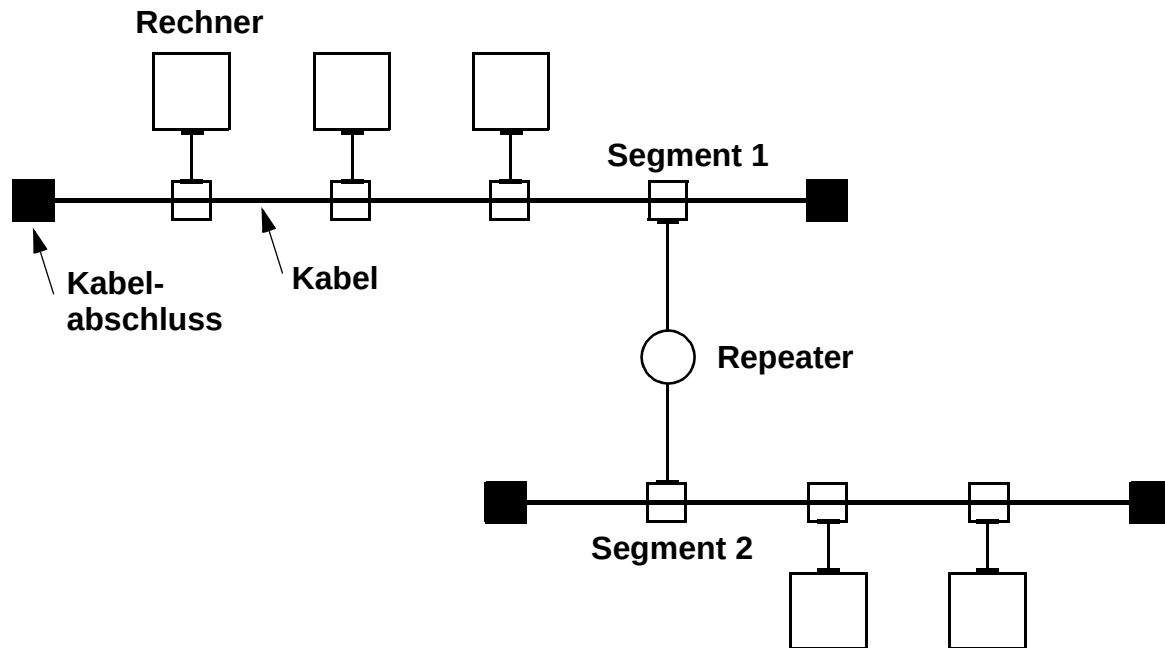
- Basisband-Bussysteme mit 10 Mb/s



Spezifikationen:

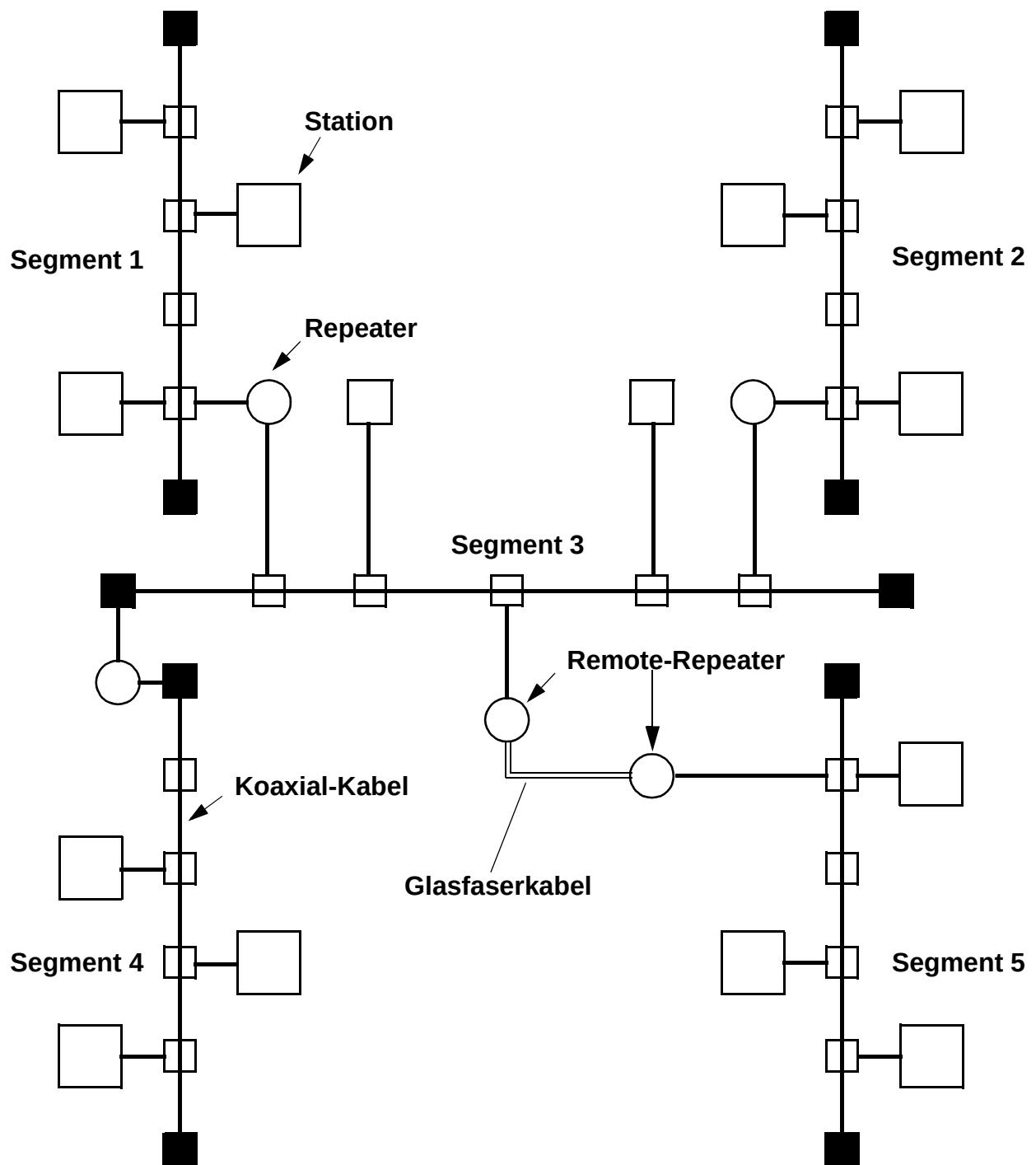
- 1.) Koaxial-Kabel mit 50 Ohm Wellenwiderstand
- 2.) max. 100 Stationen pro Segment
- 3.) max. 3 Segmente in Reihe (über Repeater)
- 4.) max. 3,8 km Kabellänge incl. der Kabellänge zwischen Rechner und Transceiver

Beispiel: Mittelhgroße Konfiguration eines Ethernet-Systems



- 2 Segmente werden über einen Repeater verbunden.

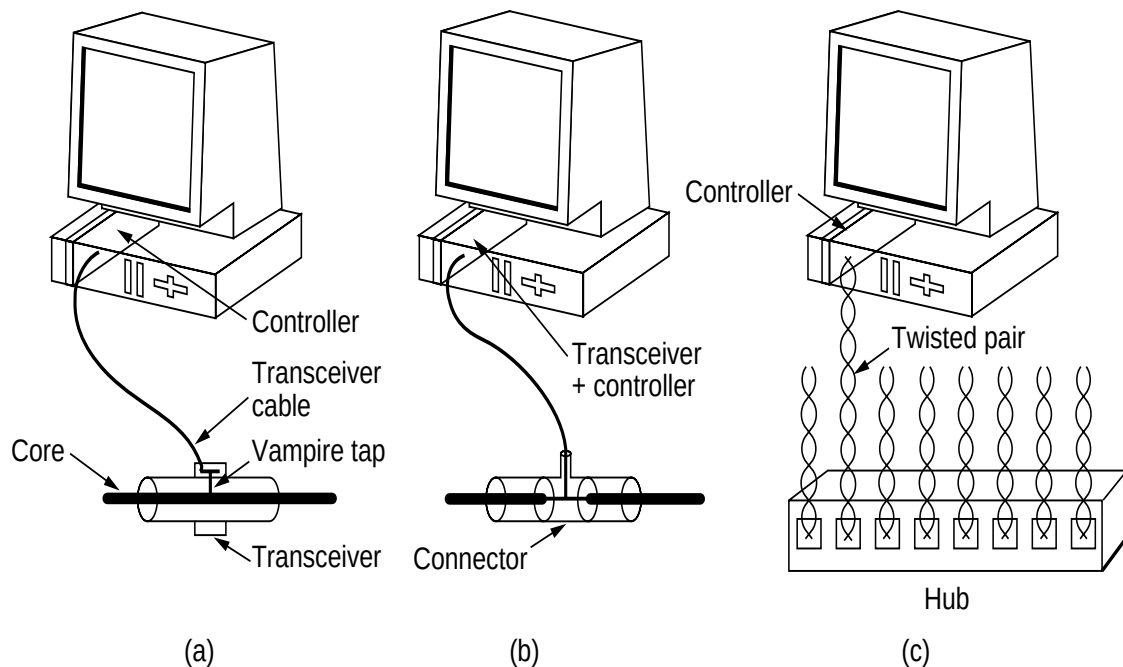
Beispiel: Große Konfiguration



- Randbedingung „maximal-3-Segmente-in-Reihe“ ist erfüllt.
- Insgesamt ergibt sich H-förmige Topologie

Die 4 Typen von Ethernet

Name	Cable	Max. segment	Nodes/seg.	Advantages
10Base5	Thick coax	500 m	100	Good for backbones
10Base2	Thin coax	200 m	30	Cheapest system
10Base-T	Twisted pair	100 m	1024	Easy maintenance
10Base-F	Fiber optics	2000 m	1024	Best between buildings



a) Thick Ethernet (dickes gelbes Kabel, 10Base5)

b) Thin Ethernet (Standard-Koaxialkabel, 10 Base2)

c) Switched Ethernet (Twisted Pair, 10BaseT)

Hub = systemförmige Verbindung auf der Ebene 1 (keine Puffer, nur elektrische Trennung, keine Kollisionsvermeidung)

Switch = sternförmige Verbindung auf der Ebene 2 (Rahmen können gepuffert werden => keine Kollision an den Switch-Eingängen)

CSMA/CD: Rahmenformat (Ethernet)

7 Bytes	1	2/6	2/6	0-1500	4
Präambel	SD	DA	SA	INFO	FCS

Präambel = 7 Bytes (7 x 10101010)
SD = Starting Delimiter (1 Byte 10101011)
DA = Destination Address (2 oder 6 Bytes)
SA = Source Address (2 oder 6 Bytes)
Info = n Bytes = Datenteil $0 < u < 1500$
FCS = Frame Check Sequence = CRC (4 Bytes)

- Die Präambel dient zur Synchronisation des Empfängertraktes mit dem Sendetakt.
- Der Delimiter dient zur Kennzeichnung, wann ein Paket anfängt.
- Die Zieladresse dient zur Klärung, wohin ein Paket geschickt werden soll.
- Mindestlänge eines Rahmens muss 64 Bytes betragen.
- Falls Rahmen < 64 Bytes sein sollte, wird ein Füllfeld eingefügt (pad), so dass Länge = 64 wird.

CSMA/CD: Adressformat

Bei IEEE 802.x gibt es 2 Adressformate

1.) 48-Bit-Format = 6 Byte

I/G	U/L	46 Bits Adresse
-----	-----	-----------------

2.) 16-Bit-Format = 2 Byte

I/G	15 Bits Adresse
-----	-----------------

I/G= 0 individual address (Punkt-zu-Punkt-Verb.)

I/G= 1 group address (Multicast-Verbindung)

U/L= 0 globally administered address

U/L= 1 locally administered address

=> Bei 6 Byte-Adressen sind 46 Bit nutzbar = ca. $7 \cdot 10^{13}$
Adressen

Token Ring hat dasselbe Adressformat!

- Ethernet-Adressen werden weltweit eindeutig vergeben =>
jede Ethernetkarte hat eine eindeutige Adresse.

4.5 Token Ring = Ringtopologie mit Sendeberechtigungspaket (=Token)

Standards

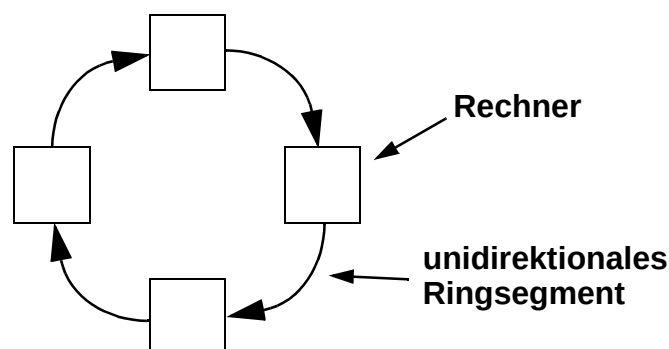
- IEEE 802.5
- ISO IS 8802/5

Hauptsächlich entwickelt im Hause IBM.

Token = engl. „Zeichen“, hier: Senderechtzeichen

Kennzeichen:

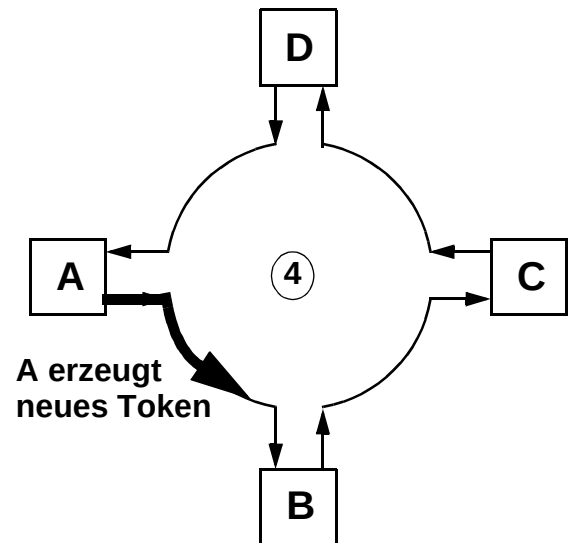
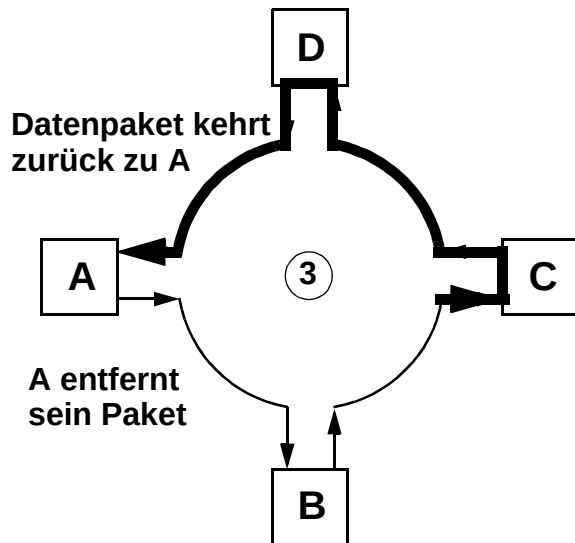
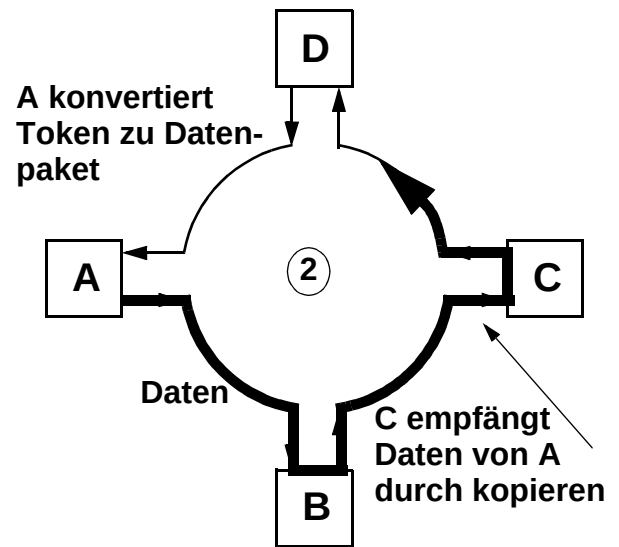
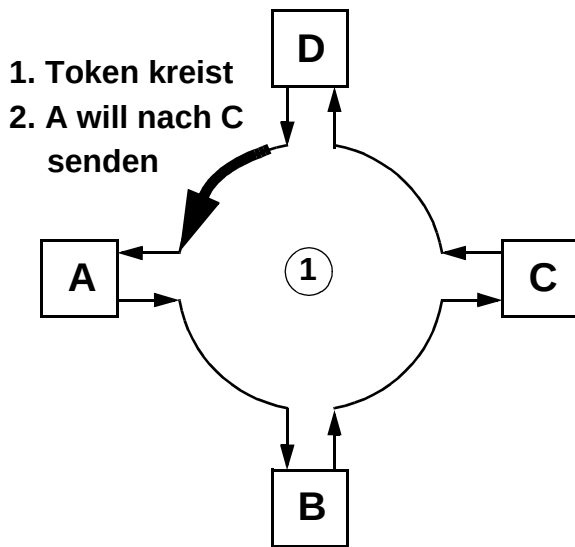
1.) Ringtopologie aus einzelnen Punkt-zu-Punkt-Segmenten.



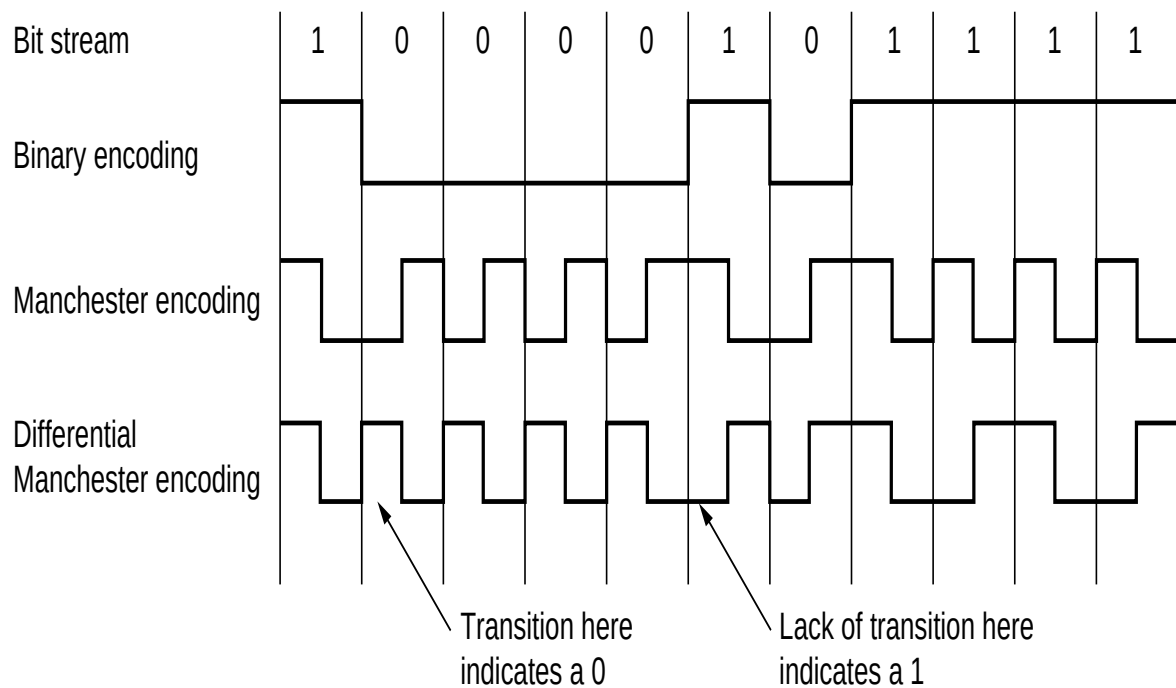
2.) Datenrate ist 16 Mb/s

- Zugriff auf den Ring durch kreisende Sendeberechtigungsmarke (Token).

Token-Ring-Protokoll für Senden



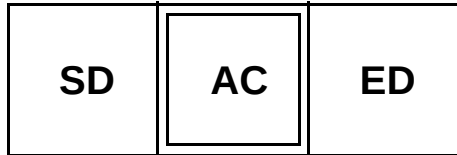
Token Ring Bitkodierung



- o Ethernet verwendet Manchester-Codierung
 - o Token Ring verwendet Differential Manchester
- 3.)

Rahmenformate

1.) Token-Format (Sendeerechtigungspaket)

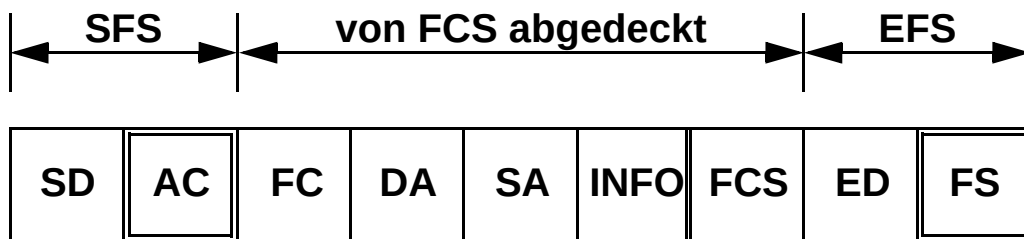


SD = Starting Delimiter (1 Byte)

AC = Access Control (1 Byte)

ED = Ending Delimiter (1 Byte)

2.) Rahmenformat für Daten- und Verwaltungspakete



SFS = Start-Of-Frame Sequence

SD = Starting Delimiter (1 Byte)

AC = Access Control (1 Byte)

FC = Frame Control (1 Byte)

DA = Destination Address (2/6 Bytes)

SA = Source Address (2/6 Bytes)

INFO = Information (≥ 0 Bytes)

FCS = Frame-Check Sequence (4 Bytes)

EFS = End-Of-Frame Sequence

ED = Ending Delimiter (1 Byte)

FS = Frame Status

- Die ersten beiden Felder bei Token- bzw. Datenrahmen sind gleich.
- Ein Bit im AC-Feld sagt, ob es sich um Token oder Daten handelt.
- Senden erfolgt durch Komplementieren des Token-Bits und Anhängen der Datenformat an das Token.

Token Ring: AC- und FS-Felder

Access Control (AC)

P	P	P	T	M	R	R	R
---	---	---	---	---	---	---	---

PPP = Priority Bits

T = Token Bit

M = Monitor Bit

RRR = Reservation Bits

Frame Status (FS)

A	C	r	r	A	C	r	r
---	---	---	---	---	---	---	---

A = Address-Recognized Bit

C = Frame-Copied Bit

r = Reserved Bits

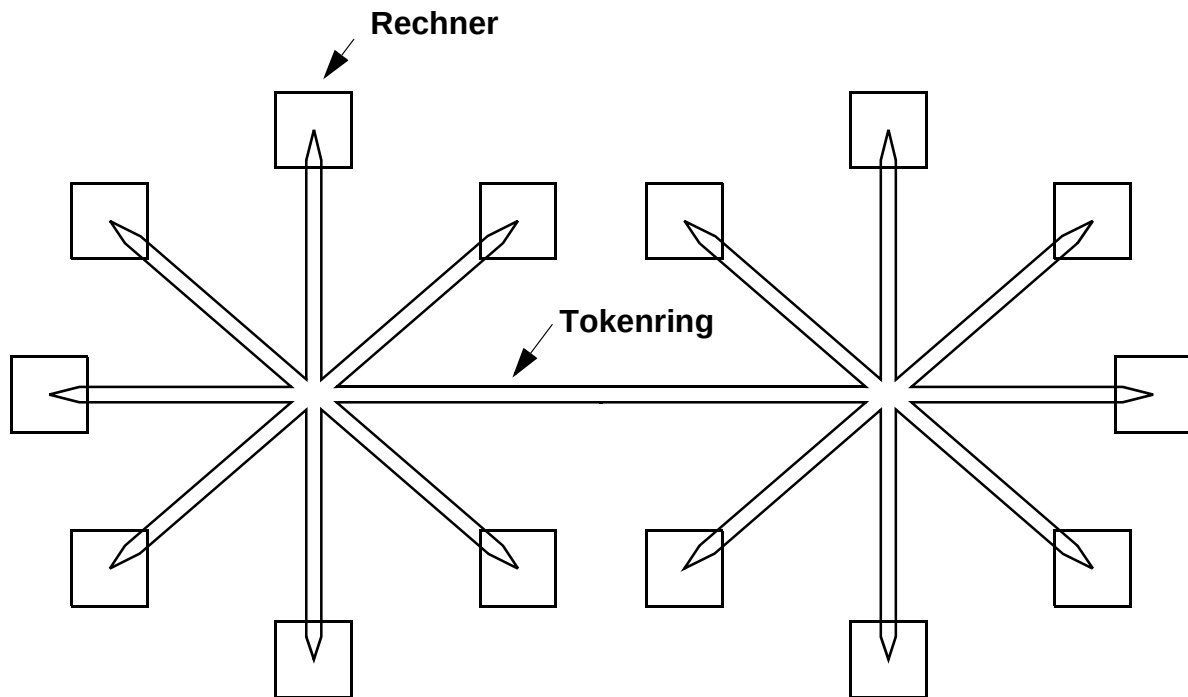
- ppp:** Prioritätsbits zeigen die momentane Sendepriorität an. Nur, wenn eigene Priorität $\geq$ ppp, darf gesendet werden.
- T:** sagt, ob Token- oder Datenrahmen.
- M:** wird während des 1. Umlaufs vom Monitor (= Überwacher) gesetzt, um ein 2. Kreisen des Pakets zu erkennen.
- A:** wird vom Empfänger gesetzt, wenn seine Adresse mit der Paketadresse übereinstimmt.
- C:** wird vom Empfänger gesetzt, wenn er das Datenpaket erfolgreich kopiert hat. => kein extra Acknowledge-Paket nötig.

Es gilt:

A = 1 und C = 1 => alles o.k.

Strukturierte Verkabelung

- Möglichkeiten einer Ringverkabelung



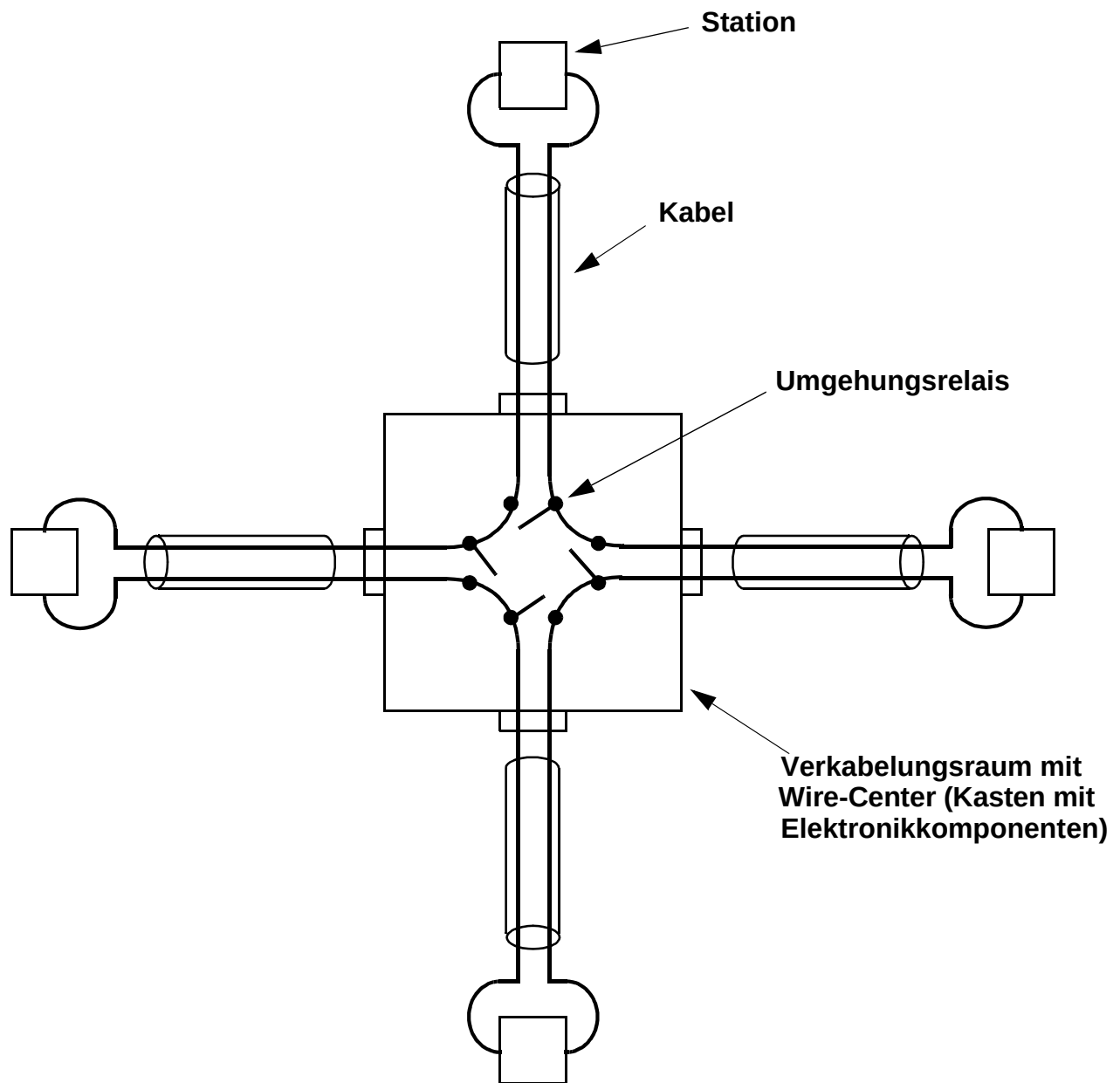
Logischer Ring

aber

Physischer Stern

Im Zentrum des Sterns steht das sog. wire center.

Physische Sterntopologie für den Token Ring



- Fällt eine Station aus, wird sie mit Hilfe des Umgehungsrelais aus dem Ring ausgeblendet.

Merkmale des Token Ring LANs

- Erlaubt die strukturierte Verkabelung von Gebäuden.
- Fehlerhafte Stationen können isoliert und aus dem Ring ausgeschlossen werden, insbesondere bei sternförmiger Verkabelung.
- Neugenerierung der Rechteckimpulse in jeder Station. Dadurch wenig rauschempfindlich. Große Ringe mit vielen Stationen möglich, da Punkt-zu-Punkt-Strecken.
- Natürliche Anwendung für Glasfaser, weil der Ring aus einzelnen Punkt-zu-Punkt-Verbindungen aufgebaut ist (kein gemeinsames Medium auf der Bitübertragungsebene, sondern auf der Schicht 2).

Token Ring- Funktionen zur Behandlung von Problemen

- Es gibt genau einen aktiver Monitor (= Beobachter) zur Behandlung von Fehlersituationen auf dem Ring.
- In jeder Ringstation ist ein 2. Monitor in Stand-By-Bereitschaft, um größere Zuverlässigkeit und Verfügbarkeit zu erreichen.
- Zur Fehlerbehandlung dienen sog. Verwaltungsrahmen (management frames), wie z. B.
 - Claim Token = „Ich möchte Monitor werden“
 - Duplicate Address Test = Überprüfung auf gleiche Adressen im Ring
 - Active Monitor Present = „Mich gibt es noch“
 - Standby Monitor Present = „Ich kann zur Not einspringen“
 - Beacon = „Leuchtfeuer“, um Ringunterbrechungen zu finden
 - Purge = Ring komplett neu initialisieren (Reset)
- Ob es sich um Daten- oder ein Management-Token handelt, wird im FC-Feld (Frame Control) angezeigt.

Der jeweils aktive Monitor

Schützt vor den folgenden Fehlerbedingungen:

- ewig kreisende Paketen (werden vom Ring entfernt)
- kreisendes Paket mit so hoher Priorität, dass keiner senden kann
- verlorengegangenes Token (neues Token wird erzeugt)
- mehrere aktive Monitore (es darf nur einer jeweils aktiv sein, sonst entsteht ein Chaos)

Mehrere aktive Monitore

Ein aktiver Monitor zieht sich in die Bereitschaft („stand-by“) zurück, falls er einen

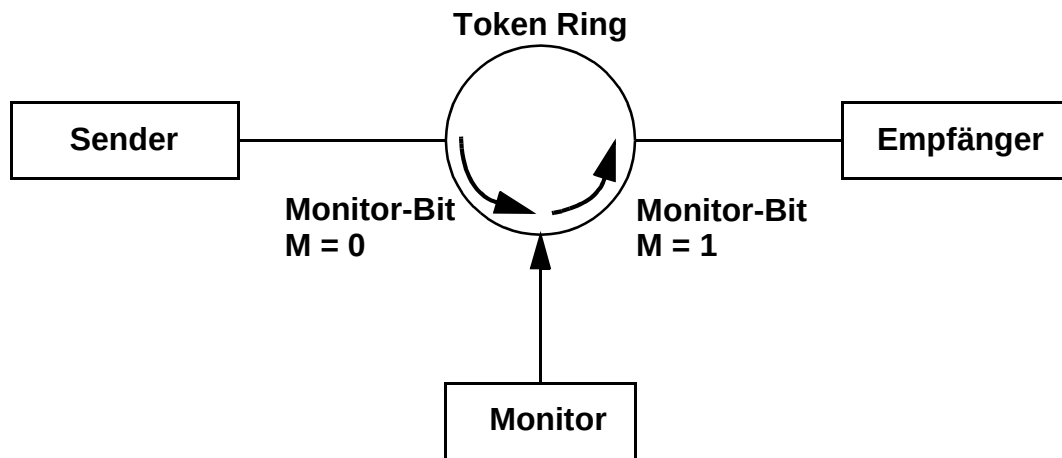
- | | | |
|---|---|---------------------|
| <ul style="list-style-type: none">• Purge Frame oder• Active Monitor Present Frame | } | 2 Management-Frames |
|---|---|---------------------|

empfängt, den er nicht selbst erzeugt hat.

Verlorenes Token

- Der aktive Monitor benutzt einen Timer, um den Verlust eines Tokens oder eines Rahmens zu entdecken.
- Der Timer wird jedes Mal neu gestartet, wenn ein Anfangsbegrenzer („delimiter“) am Monitor vorbei kommt.
- Falls der Timer abläuft, löscht der aktive Monitor den Ring und erzeugt ein neues Token („Purge“ und neues Token)

Fehlerbehebung bei ewig zirkulierendem Rahmen oder bei Token mit zu hoher Priorität



- Der Sender erzeugt einen Rahmen mit Monitorbit $M = 0$
- Angenommen, der Sender fällt aus. Dies bleibt zunächst unbemerkt.
- Der aktive Monitor setzt das Monitorbit bei jedem vorbeikommenden Paket mit $M = 0$.
- Kommt ein Paket mit $M = 1$ vorbei, wird es aus dem Ring entfernt, um ewiges Kreisen zu vermeiden.
- Zusätzlich: Pakete, deren Priorität höher ist als die höchste Priorität einer Ringstation, werden vom Monitor entfernt, weil sonst keiner senden kann.

Stadtnetze (MANs)

MAN = Metropolitan Area Network

Beispiele für ein MAN: FDDI und DQDB

FDDI = Fibre distributed Data Interface
= glasfaserbasiertes MAN mit 100 Mb/s

DQDB = Distributed Queue Dual Bus
= MAN mit 34 Mb/s

FDDI = Nachfolger von Token Ring

Fiber Distributed Data Interface

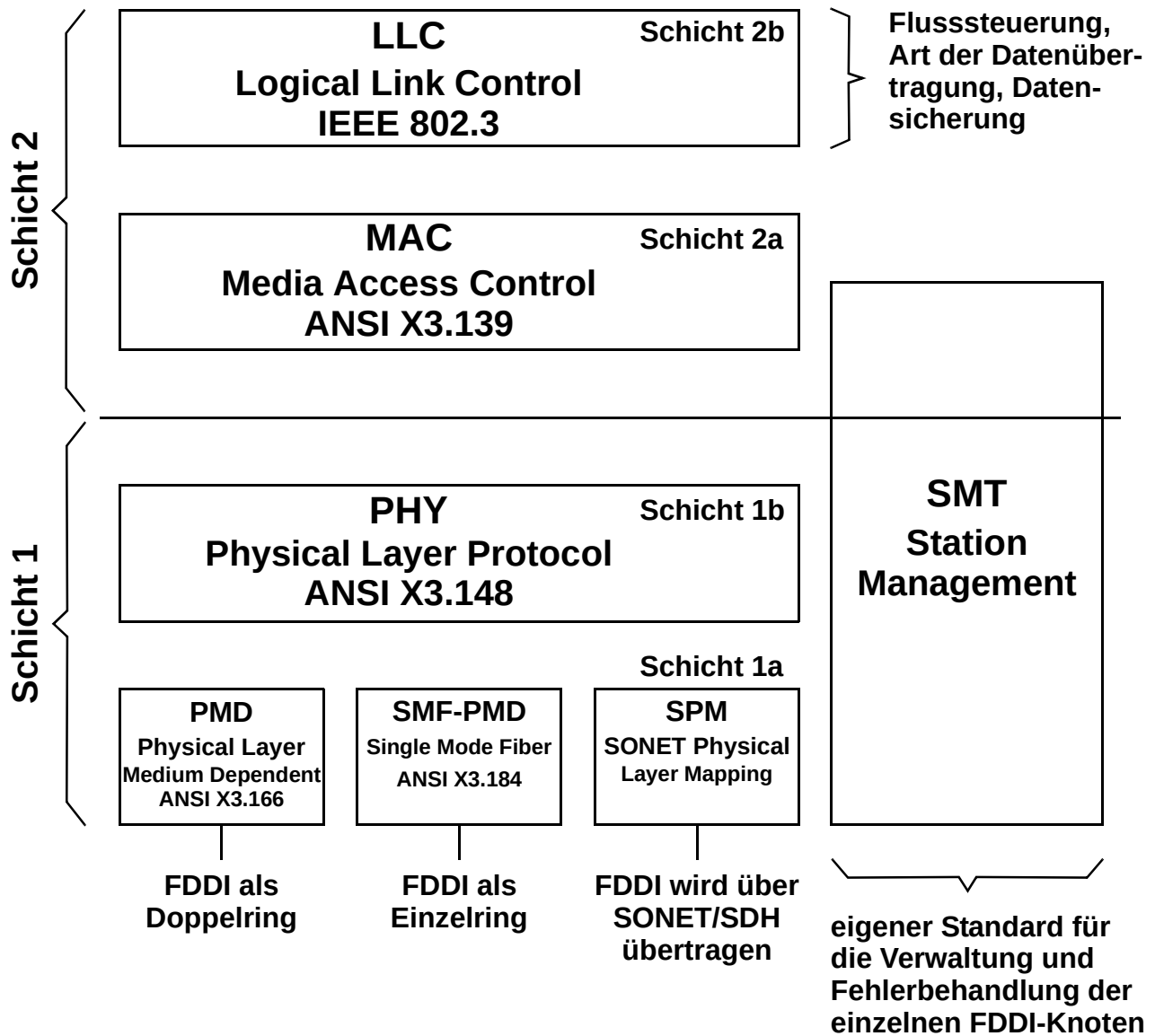
Motivation:

- Netzwerktechnik soll Leistungssteigerungen in der Rechnertechnologie folgen (immer schneller)
- Sprung von der 10/16 Mbit/s-Geschwindigkeitsklasse (mittelschnelle LANs) auf die 100 Mbit/s-Klasse
- größere räumliche Ausdehnung (200 km)
- große Anzahl an Stationen (1.000)

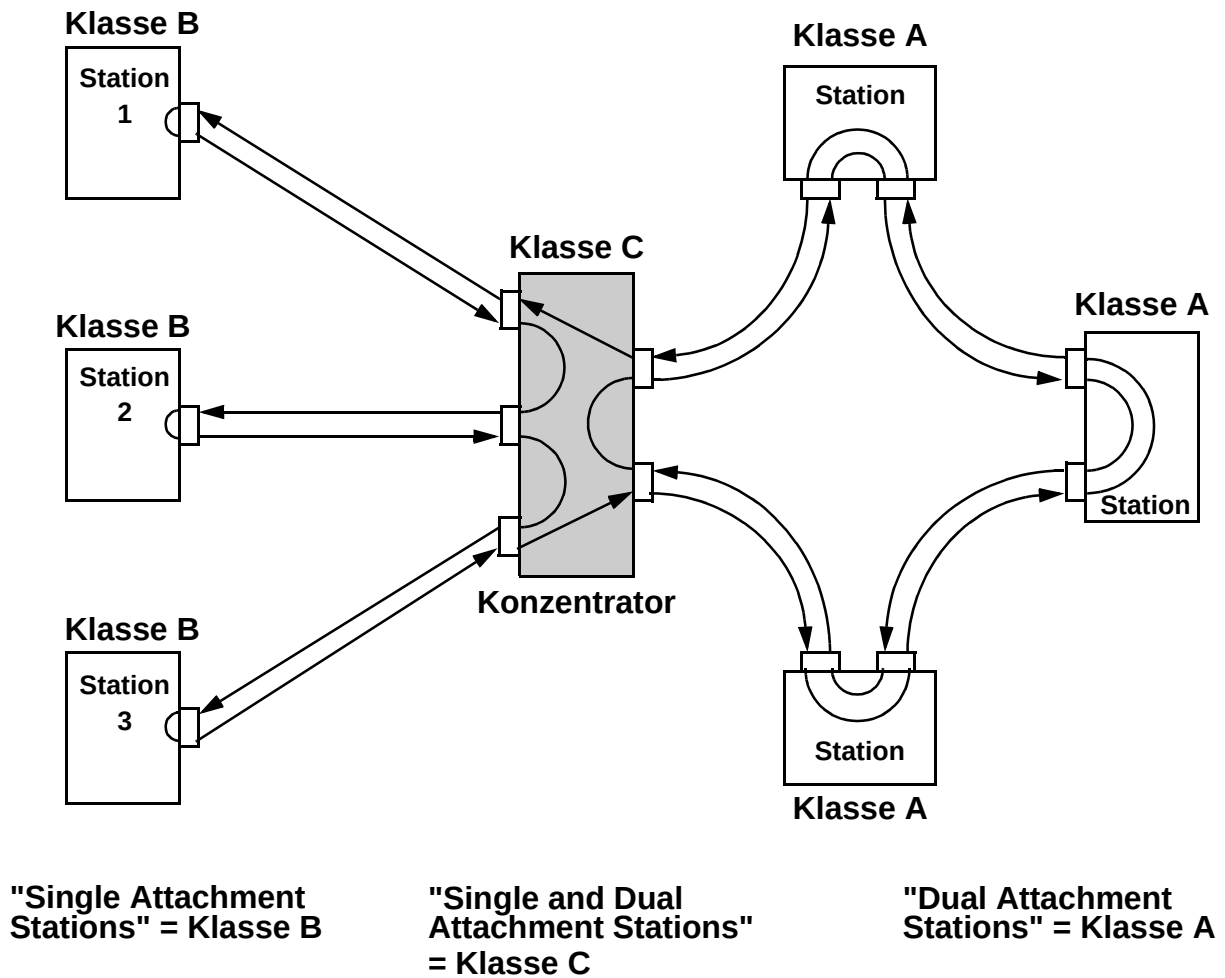
Topologie und physikalisches Medium

- Topologie: Doppelring aus Fehlertoleranzgründen
- Gradientenfaser oder Monomode-Faser, d. h. glasfaserbasierend
- Leuchtdioden mit Wellenlänge 1.300 nm als Sender
- Stecker mit je zwei Glasfasern (pro Punkt-zu-Punkt-Segment für Hin- und Rückrichtung)
- Medium Access Control über Token mit time-out-Überwachung
- Datenrate: 100 Mbit/s
- standardisiert durch ANSI und ISO

Die einzelnen FDDI-Unterstandards

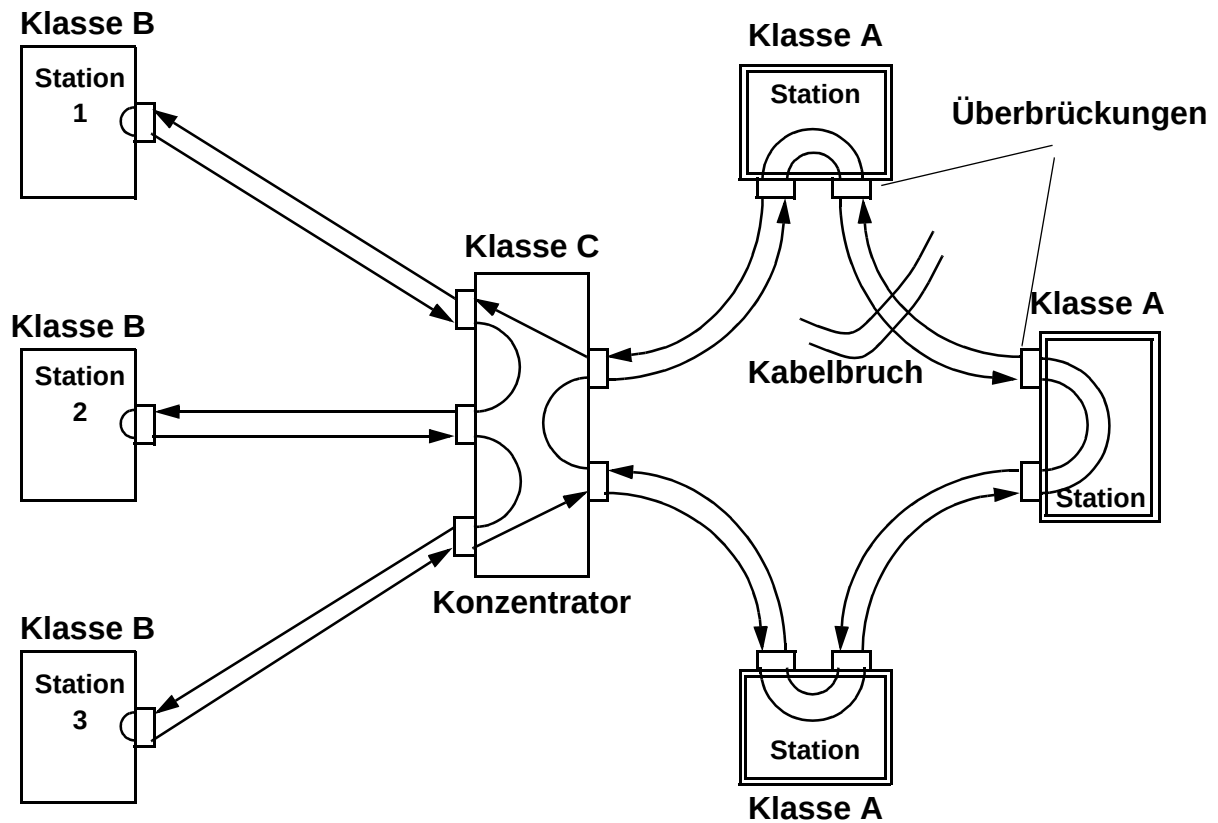


Beispiel für eine FDDI-Topologie



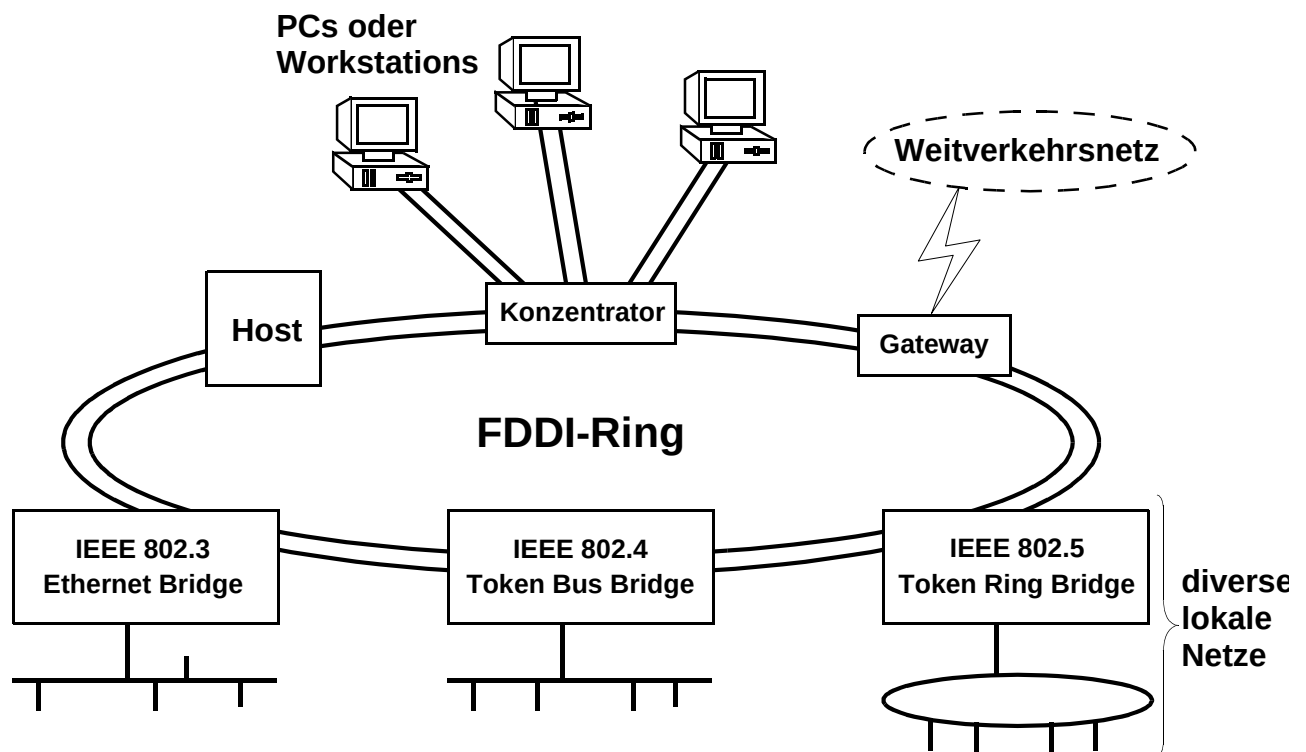
single attachment: 1 FDDI-Anschluss
dual attachment: 2 FDDI-Anschlüsse
attachment: "Anhänger, Zusatzgerät"

Fehlertoleranz durch Doppelring



Ein defekter Doppelring kann bei Kabelbruch zu einem funktionierendem Einzelring geschlossen werden.

Typische FDDI-Konfiguration (FDDI als Backbone)



Bridge = Gerät, das zwei LANs auf der Ebene der Sicherungsschicht koppelt (genau auf Schicht 2a, MAC)

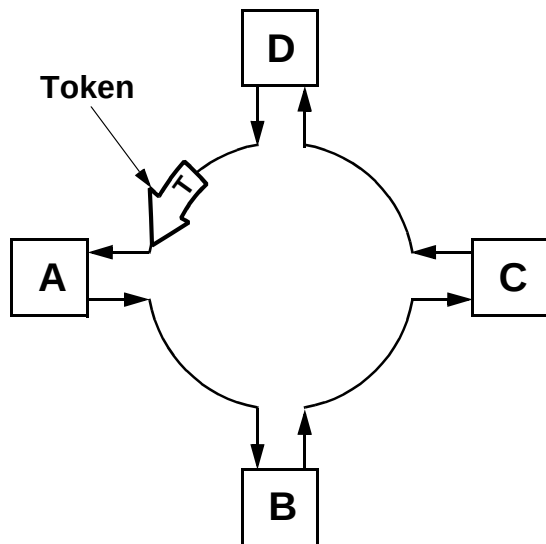
Gateway = Gerät, das LAN oder MAN mit einem WAN verbindet (Schicht 3-Koppelung)

Backbone: Rückgrat

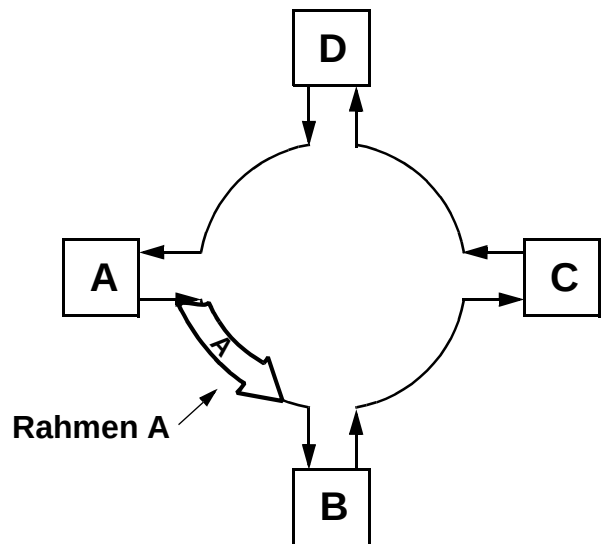
FDDI

Zugangsprotokoll zum Medium (1)

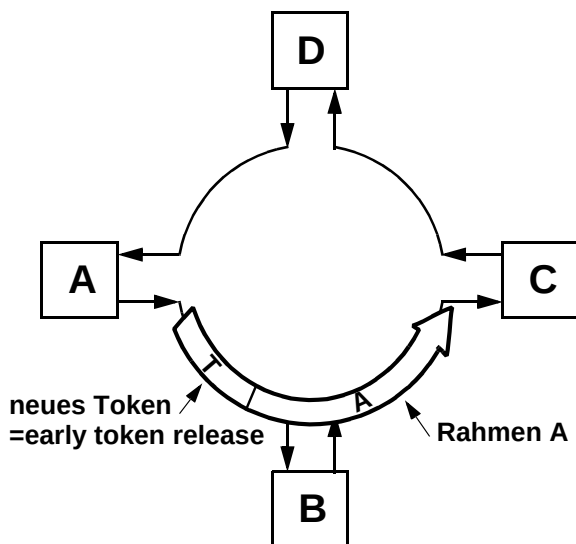
- Token Passing mit „Early Token Release“



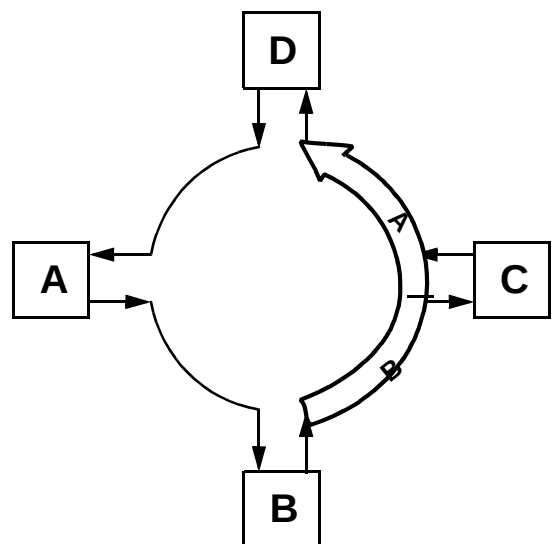
1. A wartet auf Token



2. A ergreift Token und
überträgt Rahmen A

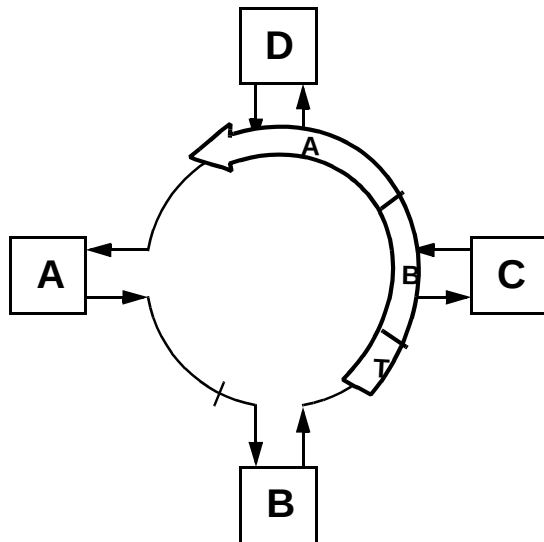


3. A hängt Frei-Token an

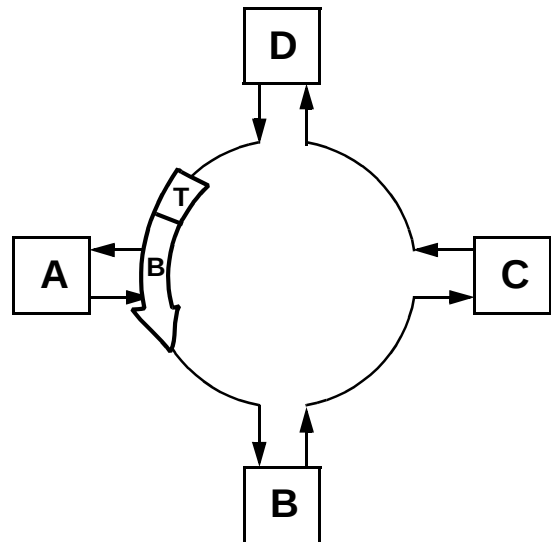


4. B ergreift Token und
überträgt Rahmen B

Zugangsprotokoll zum Medium (2)

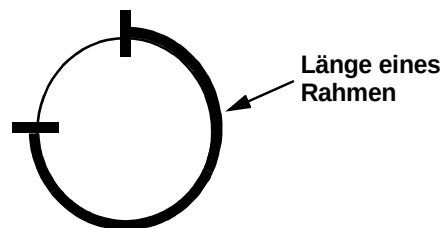


5. B hängt Frei-Token an,
C kopiert Rahmen B,
D kopiert Rahmen A

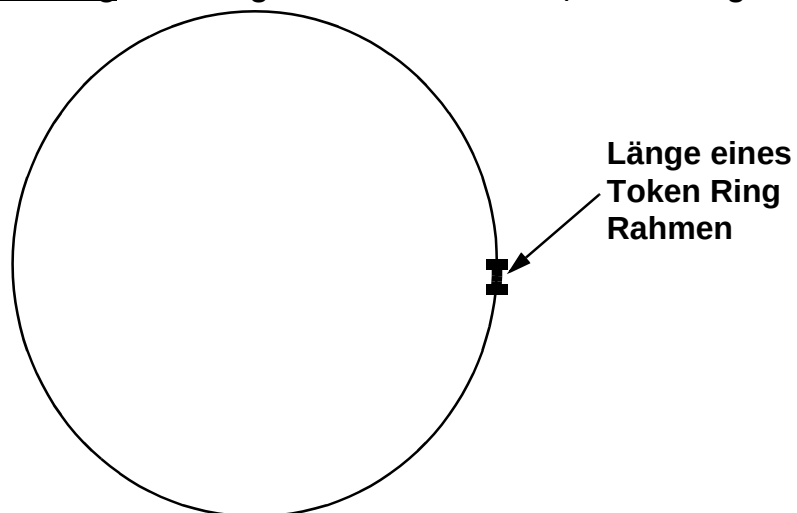


6. A entfernt Rahmen A
usw.

Motivation für „Early Token Release“

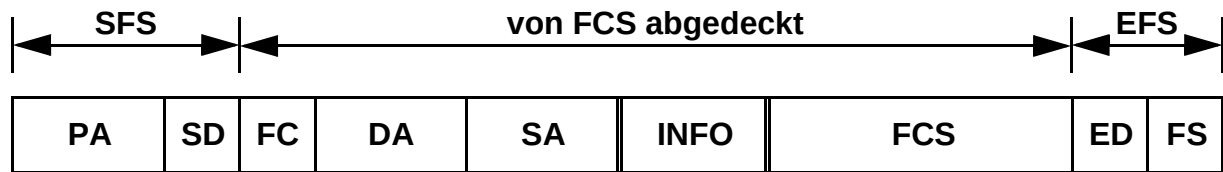


a) ISO 8802.5 Token Ring zum Vergleich mit 16 Mbit/s, 100 m lang



b) FDDI 100 Mbit/s, 100 km lang

FDDI: Rahmenformate



SFS : Start-of-Frame-Sequenz

PA: Präambel (8 oder mehr Bytes) wie bei Ethernet

SD: Starting Delimiter (1 Byte)

FC: Frame Control (1 Byte)

DA: Destination Address (2 o. 6 Bytes)

SA: Source Address (2 o. 6 Bytes)

INFO: Information (0 oder mehr Bytes)

FCS: Frame-Check Sequenz (4 Bytes)

EFS: End-of-Frame Sequenz

ED: Ending Delimiter (1 Byte)

FS: Frame Status (12 Bits = 3 Symbole), ähnlich wie bei Token Ring

} wie bei Ethernet

} wie bei Token Ring

} wie bei Ethernet
oder Token Ring

FDDI: Bitcodierung

4B/5B-Symbolcodierung

- 4 Datenbits werden in 5 Codebits übertragen
- 125 Mbaud Übertragungsrate notwendig
- 80% Effizienz (4:5)
- zum Vergleich: Manchester-Code hat 50% Effizienz

Bitcodierung: Non-Return to Zero(Invert on Ones (NRZI)

- **1-Bit: Pegelwechsel d. h., die „1“ wird nicht im Wert sondern im Wechsel codiert bei 0-Bit kein Pegelwechsel**
 - Diese Bitcodierung in Verbindung mit der 4B/5B-Symbolcodierung stellt sicher, dass im Impulsstrom maximal drei Bitzeiten ohne Pegelwechsel vorkommen.
- => Die Synchronisation bleibt gewährleistet, wenn die Uhren des Senders und des Empfängers mindestens drei Taktzeiten lang hinreichend gleich laufen!**

FDDI

4B/5B Symboltabelle (1)

DECIMAL	CODE GROUP	SYMBOL	NAME	ASSIGNMENT
00	00000	Q	QUIT	Line State Symbol
31	11111	I	IDLE	”
04	00100	H	HALT	”
24	11000	J		Starting Delimiter
17	10001	K		”
05	00101	L		”
13	01101	T		Ending Delimiter
07	00111	R	RESET	Control Indicator
25	11001	S	SET	”
30	11110	0		Data 0000
09	01001	1		Data 0001
20	10100	2		Data 0010
21	10101	3		Data 0011
10	01010	4		Data 0100
11	01011	5		Data 0101
14	01110	6		Data 0110
15	01111	7		Data 0111
18	10010	8		Data 1000
19	10011	9		Data 1001

Sonderzeichen für die Leitungssteuerung

Sonderzeichen für die Rahmenkennzeichnung

4B/5B Symboltabelle (2)

DECIMAL	CODE GROUP	SYMBOL	NAME	ASSIGNMENT
22	10110	A		1010
23	10111	B		1011
26	11010	C		1100
27	11011	D		1101
28	11100	E		1110
29	11101	F		1111
01	00001	V	VIOLATION	Not Transmitted
02	00010	V	”	”
03	00011	V	”	”
06	00110	V	”	”
08	01000	V	”	”
12	01100	V	”	”
16	10000	V	”	”

Vergleich Ethernet, Token Ring und FDDI

CSMA/CD (Ethernet): LAN

- Sehr gute Performance bei geringer bis mäßiger Belastung (viele Kollisionen bei hoher Belastung)
- Einfaches Protokoll, leicht implementierbar (z. B. kein Monitor notwendig)
- Topologische Einschränkungen bei der Bus-Topologie (2,5 m Raster)
- Keine Prioritätssteuerung
- Keine maximale Verzögerung garantierbar (für Echtzeit ungeeignet)

Token Ring: LAN

- Hohe Systemausnutzung (überträgt viel Verkehr effizient)
- Fehlerbehebung („Token Recovery“) aufwendig
- Flexible Verkabelung durch wire center
- Prioritäten möglich
- Maximale Verzögerung garantierbar (außer beim Auftreten von Fehlern) => deterministisch, für Echtzeit geeignet

FDDI: LAN; MAN

- Eine schnelle Variante des Token Rings
- Kombiniert synchronen und asynchronen Verkehr auf dem selben Medium

synchroner Verkehr: nach konstanter Zeit darf Station wieder
Station wieder senden

asynchroner Verkehr: erst wenn Token da ist, darf Station wieder
senden

Schicht 2 b der Sicherungsschicht

Macht die „Logical Link Control“ im LAN (= LLC)

- Die Sicherungsschicht in diesen LANs bzw. MANs ist identisch für CSMA/CD, Token Ring, Token Bus, FDDI
- Drei verschiedene Dienste werden von LLC unterstützt
LLC Typ 1: Unbestätigter verbindungsloser Dienst
 - Unbestätigte Übertragung (kein Acknowledge) von Datenrahmen (= Datagramm)
 - Höhere Schichten sind für die Erhaltung der Reihenfolge, Fehlerbehebung und Flusssteuerung verantwortlich (passiert in Schicht 4, Transportschicht)

LLC Typ 2: Verbindungsorientierter Dienst (analog wie bei HDLC)

- Verbindungsauf- und abbau
- Datenübertragung mit Bestätigung (ACK)
- Garantierte Ablieferung beim Empfänger
- Garantierte Reihenfolge der Rahmen
- Flusssteuerung

Logical Link Control (2)

LLC Typ 3: Bestätigter verbindungsloser Dienst

- Auf jedes Datagramm kann genau eine Bestätigung folgen (zwar Ack aber keine garantierte Reihenfolge)
- Anwendung in der Fertigungssteuerung zur Fabrikautomation (Steuerung von Bearbeitungsmaschinen an einem Fließband, z. B. in der Automobilproduktion.)

Weitverkehrsnetze (WANs)

Wan = Wide Area Network

Unterschied zwischen LANs/MANs und Weitverkehrsnetzen (WANs)

- LANs und MANs bestehen aus einem gemeinsamen Übertragungsmedium und den daran angeschlossenen Rechnern (sog. Hosts)

=> **LANs und MANs sind Broadcast-Netze auf der Schicht 1.**
Eine Wegewahl (Routing) ist nicht erforderlich, da jede Nachricht bei allen Hosts vorbeikommt.

- WANs bestehen aus mind. 1 Weitverkehrsstrecke sowie aus einem oder mehreren LANs und/oder MANs am Anfang und Ende der Weitverkehrsstrecke.
- Beispiel einer Weitverkehrsstrecke ist eine ATM-Verbindung, die über eine SONET-Glasfaserstrecke implementiert ist.
- ATM = Asynchronous Transfer Mode = Protokoll für den unteren Teil der Schichten 2-7 (umfasst $\approx$ Schicht 2-4)
- SONET = Synchronous Optical Network = eine Netztechnik der Schicht 1.
- Es gilt: alle Weitverkehrsstrecken sind sog. Punkt-zu-Punkt-Verbindungen.

- An den Endpunkten einer Weitverkehrsstrecke sitzen spezielle Hosts, sog. Router.
 - Router dienen auch zur Kopplung von LANs und MANs.
 - Es gilt: jeder Router ist mit einem Nachbar-Router über eine dedizierte Strecke verbunden. Diese Strecke kann ein LAN, MAN oder eine Weitverkehrsstrecke sein.
 - bei WANs ist ein sendender Host i. A. nicht im selben LAN oder MAN wie der empfangende Host.
- => Auf seinem Weg durch das WAN durchläuft ein Paket i. A. mehrere Netze (LANs, MANs und Weitverkehrsstrecken)
- => Bei WANs ist eine Wegewahl (Routing) erforderlich. Die Wegewahl wählt einen Pfad durch das WAN. Sie wird von der OSI-Schicht 3 (Vermittlungsschicht) vorgenommen.
- => Router sind Rechner oder spezielle Geräte, die auf der Schicht 3 arbeiten.

Die Vermittlungsschicht (Schicht 3)

- ist erforderlich, sobald mehrere LANs, MANs oder Weitverkehrsstrecken über Router gekoppelt werden.
- dient zur Wegewahl
- Die Vermittlungsschicht transportiert ein Datenpaket, das von einem Host erzeugt wird, über eine Folge von Routern zum empfangenen Host.
- Die Folge von Routern heißt Weg, Pfad oder Route (Leitweg) durch das Netz.
- Das Routing erfolgt mit Methoden der Graphentheorie. Dabei wird das Netz als Graph dargestellt. Knoten des Graphen sind die Router. Kanten des Graphen sind LANs, MANs und Weitverkehrsstrecken.

Zusammenfassung der Unterschiede zwischen Punkt-zu-Punkt-Netzen und Broadcast-Netzen

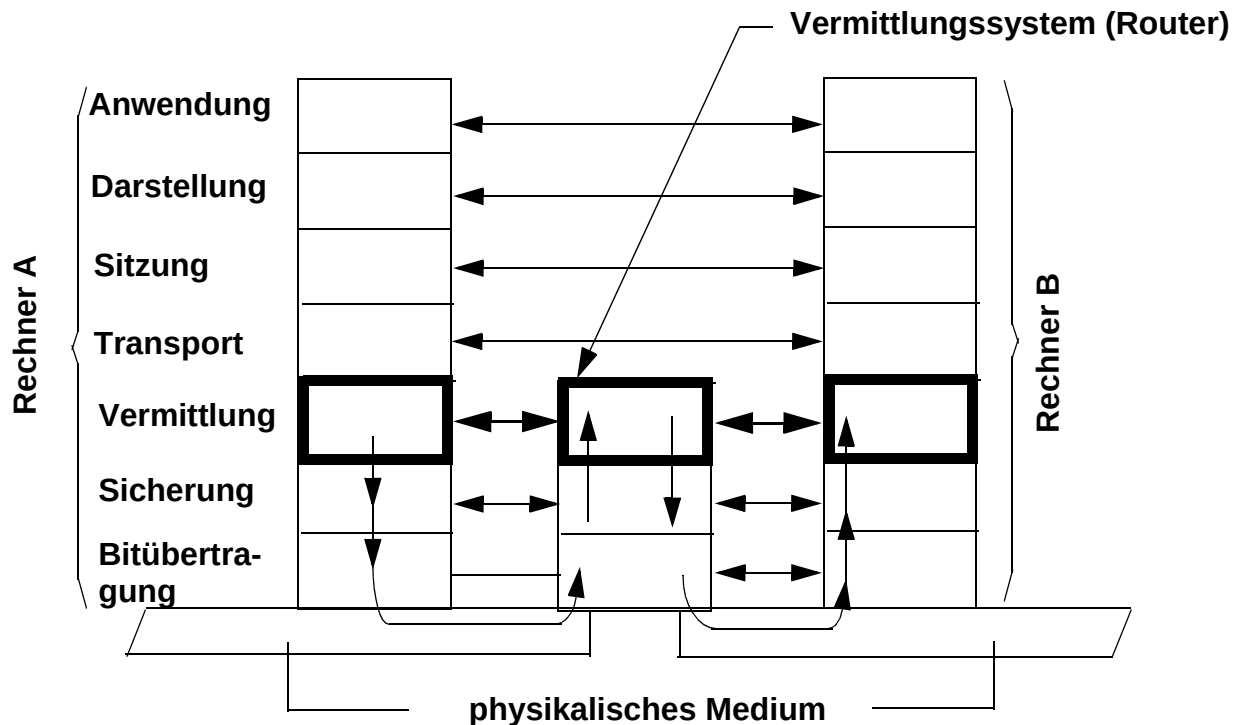
Broadcast-Netz = LAN, MAN

- Mehrere Stationen teilen sich das physikalische Medium.
- Alle Stationen hören alle Nachrichten.
- Falls zwei Stationen gleichzeitig senden, werden beide Nachrichten verfälscht bzw. zerstört.
- Der Sender kann seine eigene Nachricht hören. Falls er genau das hört, was er gesendet hat, kann er annehmen, dass auch der Empfänger die Nachricht korrekt empfangen hat.
- Innerhalb eines LAN-Segments ist eine Wegewahl nicht erforderlich.

Punkt-zu-Punkt-Netz = WAN

- Jeweils genau zwei Router sind physikalisch verbunden.
- Multicast und Broadcast erfordern die explizite Vervielfältigung der Nachrichten in den Zwischenknoten.
- Im Weitverkehrsnetz (= teilweise vermaschter Graph) muss eine explizite Wegewahl erfolgen, um einen bestimmten Empfänger zu erreichen.

Die Vermittlungsschicht im Referenzmodell nach OSI



- In den Netzzwischenknoten (Routern) müssen Pakete nur bis Schicht 3 (Vermittlungsschicht) wandern, in den Endsystemen hingegen durch alle Schichten

.Zusammenfassende Liste der Aufgaben der Vermittlungsschicht

- **Wegewahl** (= „Vermittlung“ bzw. Routing)
- Verbindungsaufbau und -abbau von sog. „virtuellen Verbindungen“
- Multiplexen von virtuellen Verbindungen
- Segmentierung von langen Paketen in kurze und Zusammenstellung kurzer Pakete zu langen
- Fehlererkennung/Fehlerbehebung (**Ende-zu-Ende**)
- Sicherstellung der Paketreihenfolge (aufsteigende Sequenznummer)
- Flusssteuerung (**Ende-zu-Ende**) = Überlaststeuerung einzelner Hosts
- Überlaststeuerung der Teil**netze**

Ziel: Eine möglichst große Vielfalt von Netzen und Technologien soll unterstützt werden.

Bemerkung:

Multiplexen = eine Leitung dient nacheinander der Übertragung mehrerer, gleichzeitiger Kommunikationen

Datentransfer über **Virtuelle Verbindungen oder Datagramme**

- **Definition Virtuelle Verbindung:**

Der Weg durch das Netz wird vor dem ersten Datentransfer vollständig festgelegt. Dieser Weg heißt **virtuelle Verbindung**. In jedem Zwischenknoten findet nur einmal eine Wegwahlentscheidung statt. Der ganze über diese virtuelle Verbindung fließende Verkehr nimmt denselben Weg durch das Netz.

- **Definition Datagramm:**

Die Paketadresse bestimmt in jedem Netzknoten auf dem Pfad die abgehende Leitung. Für jedes Datagramm wird in jedem Knoten erneut eine Wegwahlentscheidung getroffen.

- **Bemerkung**

Eine virtuelle Verbindung zwischen zwei Endsystemen (Hosts) tut so, als ob die beiden Rechner direkt miteinander verbunden wären. Da sie es aber i. a. nicht sind, spricht man von einer virtuellen Verbindung.

Datagramme

Jedes Paket (Datagramm) wird als isolierte Einheit betrachtet (keine virtuelle Verbindung)

- **Volle Zieladresse in jedem Paket notwendig**
- **Pakete können außerhalb der Reihenfolge eintreffen**
- **Keine Fehlerüberwachung, keine Flusssteuerung**

Vorteile von Datagrammen:

- **Primitiver als virtuelle Verbindungen, daher einfacher zu implementieren**
- **Kein Verbindungsauf- und -abbau, deshalb niedriger Overhead für kurzlebige Verbindungen**
- **flexibler und zuverlässiger bei Routerausfällen**
- **besser geeignet für heterogene Subnetze, da virtuelle Verbindung zwischen Netzen verschiedener Typen technisch aufwendig ist.**

Bemerkung:

Im Internet werden sowohl Datagramme als auch virtuelle Verbindungen benutzt; von IP Datagramme, von TCP virtuelle Verbindungen.

Virtuelle Verbindungen zwischen Rechnern

Eine virtuelle Verbindung hat die zusätzliche Eigenschaft, ein „**perfekter Kanal**“ durch das Netz zu sein.

- Ordnung der Nachrichten (Sicherstellung der Reihenfolge) ist gewährleistet.
- Fehlerüberwachung (verlorene und duplizierte Pakete) finden statt.
- Flusssteuerung wird durchgeführt (Ende-zu-Ende)

3 Phasen einer virtuellen Verbindung

- Verbindungsaufbau
- Datenübertragung
- Verbindungsabbau

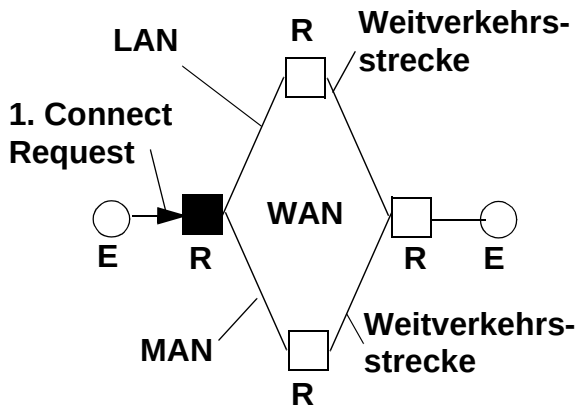
Zwei Vorteile von virtuellen Verbindungen

- Kein Aufwand für die Wegewahl während der Datenübertragung
- Keine Neusortierung der Pakete oder Fehlerüberwachung im Endsystem nötig (Transportschicht wird entlastet)

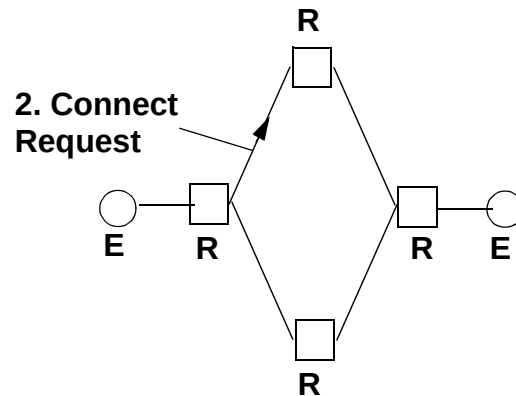
Bemerkung:

Bei einigen Protokollen der Sicherungsschicht wird bereits ein perfekter Kanal aufgebaut, allerdings nur zwischen benachbarten Rechnern, nicht Ende-zu-Ende.

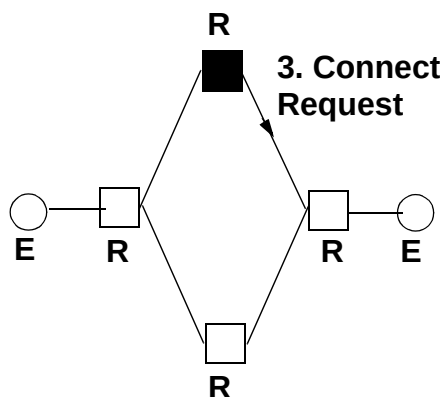
Aufbau einer virtuellen Verbindung in einem Beispiel WAN



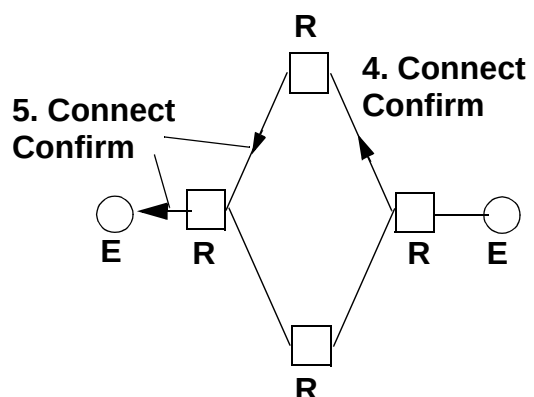
a) Festlegen des Weges



b) Aufbauphase der 1. Teilstrecke



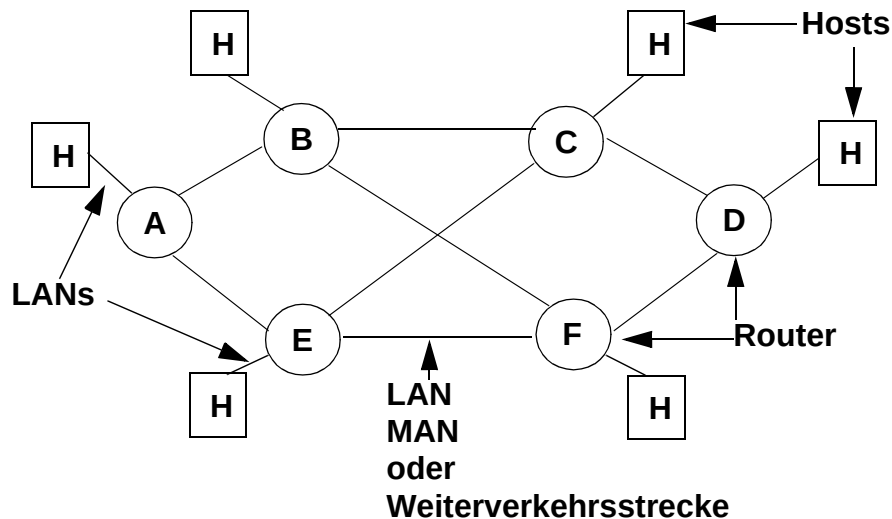
c) Virtueller Verbindungsabschnitt existiert, Festlegung der Wegfortsetzung



d) nach weiteren Schritten virtuelle Verbindung fertig gestellt.

E = Endsystem = Rechner (Host)
R = Router = spezieller Host

Implementierung von virtuellen Verbindungen in einem Beispiel-Weitverkehrsnetz



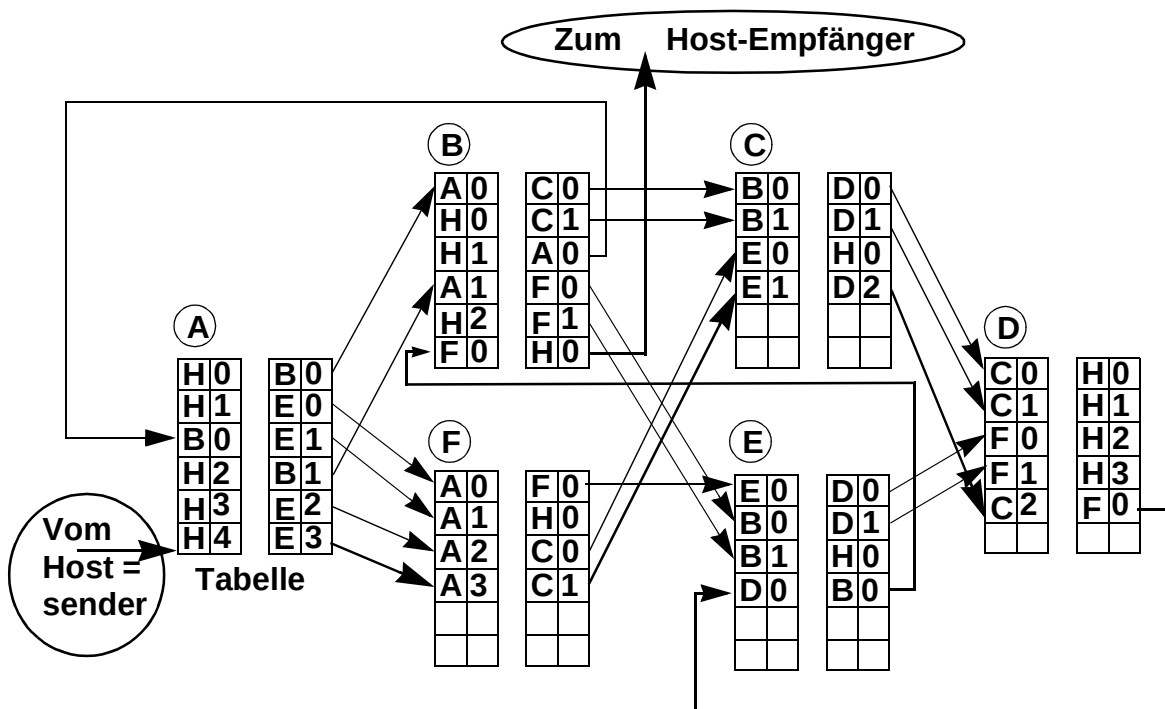
- Beispiel für acht virtuelle Verbindungen durch das WAN:

Ausgehend von A	Ausgehend von B
0 - ABCD 1 - AEFD 2 - ABFD 3 - AEC 4 - AECDFB	0 - BCD 1 - BAE 2 - BF

↑
Nummer der virtuellen Verbindung

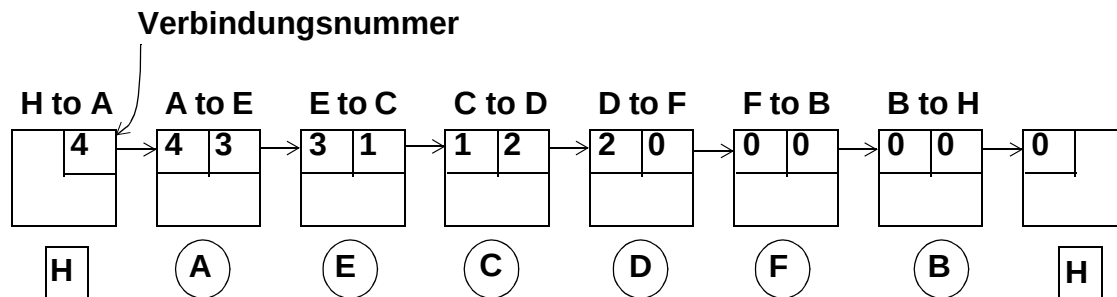
=> In jedem Router werden Tabellen mit Zustandsinformationen über durchlaufende virtuelle Verbindungen verwaltet.

Beispiel für die Zustandsinformation in den Routern beim Schalten einer virtuellen Verbindung



- In jedem Router gibt es eine Tabelle zur Verwaltung der virtuellen Verbindungen
- Jede ankommende und abgehende virtuelle Verbindung erhält eine Nummer
- Beispiel H 4 E 3 heißt:
Vom Host kommt eine Verbindung mit der Nr. 4. Diese wird unter der Nr. 3 an den Router E weitergereicht.
- Ein- und die selbe virtuelle Verbindung wird von Router zu Router unter einer anderen Nummer geführt.

Hier die Beispielverbindung HAECDFB H



- Der Vorteil ist, dass jeder Router seine Tabelle selbständig verwalten kann, ohne dass netzweit eindeutige Verbindungsnummern vergeben werden müssen.
- Nur zwischen benachbarten Routern muss für die selbe Verbindung eine Nummer verwendet werden.

Wegewahl für virtuelle Verbindungen und Datagramme

Routing-Algorithmen

Aufgabe von Routing-Algorithmen:

Leitwegbestimmung für Pakete durch das Netzwerk vom Quellsystem zum Zielsystem (Ende-zu-Ende)

Der Leitwegbestimmungsalgorithmus eines Vermittlungsrechners (Knotens) entscheidet, auf welcher Ausgangsleitung ein eingegangenes Paket weitergeleitet wird.

Bemerkung:

- Datagramm: individuelle Entscheidung für jedes Paket
- Virtuelle Verbindung: Leitwegbestimmung nur beim Verbindungsaufbau

Wünschenswerte Eigenschaften eines Wegewahl-Algorithmus:

- Korrekt (Paket findet immer zum Ziel)
- Einfach (schnell auszuführen)
- Robust (Rechner- oder Leitungsausfälle)
- Stabil (Gleichmäßige Ergebnisse)
- Fair (kein Paket wird benachteiligt)
- Optimal (findet den schnellsten oder kürzesten Weg)

Algorithmen für die Leitwegbestimmung

Optimierungskriterien des Algorithmus (Beispiele):

- Durchschnittliche Paketverzögerung (Latenz)
- Gesamtdurchsatz des Netzes (Durchsatz)

Dies ist ein sich widersprechender Zielkonflikt, daher gebräuchlich:

- Minimierung der Teilstrecken (hops) pro Paket (= Kompromiss)

Vorteil:

- reduzierte Verzögerung (jedoch nicht in allen Fällen)
- vermindert benötigte Bandbreite
- steigert Durchsatz, allerdings i. Allg. nicht bis zum Maximum

Bemerkung:

In einem Netz können Latenz und Durchsatz nicht gleichzeitig optimiert werden, weil hoher Durchsatz nur durch eine unerwünschte hohe Latenz erreichbar ist (gemäß Warteschlangentheorie).

Verfahren zur Leitwegbestimmung

Klassifikation der Verfahren:

1. Statische (nicht adaptive) Verfahren

- keine Berücksichtigung des aktuellen Netzzustands
- gehen von Mittelwerten aus, nicht von lokalen Informationen
- Leitweg zwischen i und j wird für alle i, j vor der Inbetriebnahme des Netzwerks bestimmt
- keine Änderung während des Betriebs (statisches Routing)

2. Adaptive Verfahren

- Entscheidungen basieren auf aktuellem Netzzustand
- Messungen/Schätzungen der Topologie und des Verkehrsaufkommens als Input für den Algorithmus
- Weitere Unterteilung der adaptiven Verfahren in:
 - zentralisierte Verfahren
 - lokale Verfahren
 - verteilte Verfahren

A) Statische Leitwegbestimmung

Beim statischen Routing ist die gesamte Topologie des Netzes einer zentralen Stelle bekannt. Sie berechnet die optimalen Pfade für jedes Paar (i, j) von Knoten, erstellt daraus die Routing-Tabelle für die einzelnen Knoten und versendet diese (= Routing über statische Tabellen).

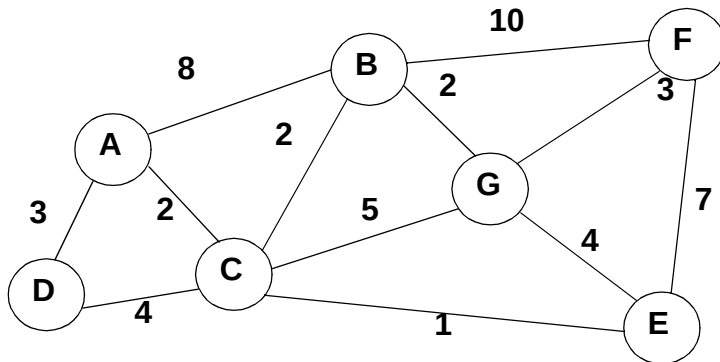
aber:

Nur sinnvoll, wenn das Netz relativ klein und relativ statisch ist.

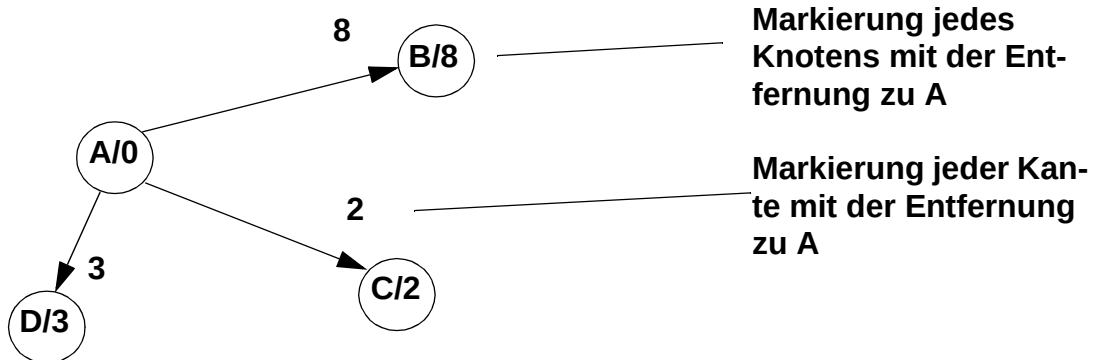
Eigenschaften:

- optimale Pfade (i, j) für alle i, j sind
 - 1.) kürzeste Pfade *oder*
 - 2.) schnellste Pfade
- die kürzesten Pfade werden zentral über den Graphenalgorithmus „Spannbaum“ bestimmt bzw. über minimalen Spannbaum
- Das bekannteste Verfahren zur Berechnung eines minimalen Spannbaums ist das von Dijkstra
- Definition Spannbaum
gegeben Graph $G = (V, E)$
dann ist $G' = (V, E')$ ein Spannbaum, wenn
 $E = \text{Baum} \wedge E$ erreicht alle Knoten gilt

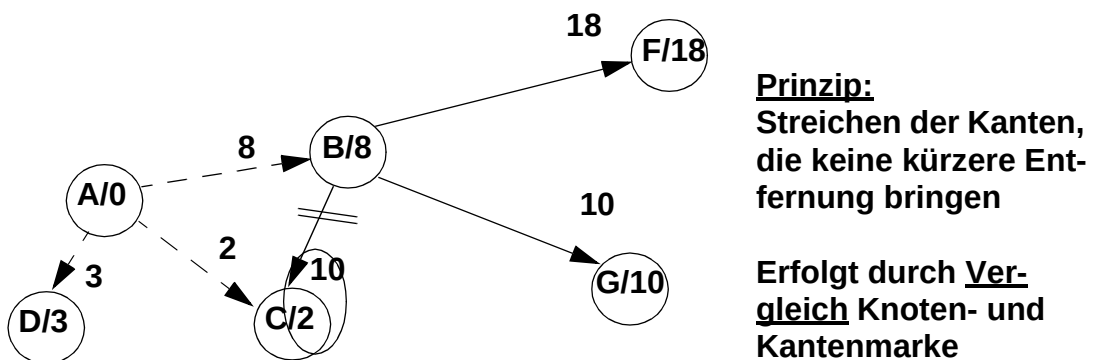
Einfaches Verfahren zum Berechnung eines Spannbaums



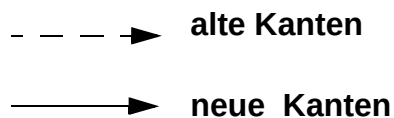
gegeben: a) Netzwerktopologie mit gewichteten Verbindungen



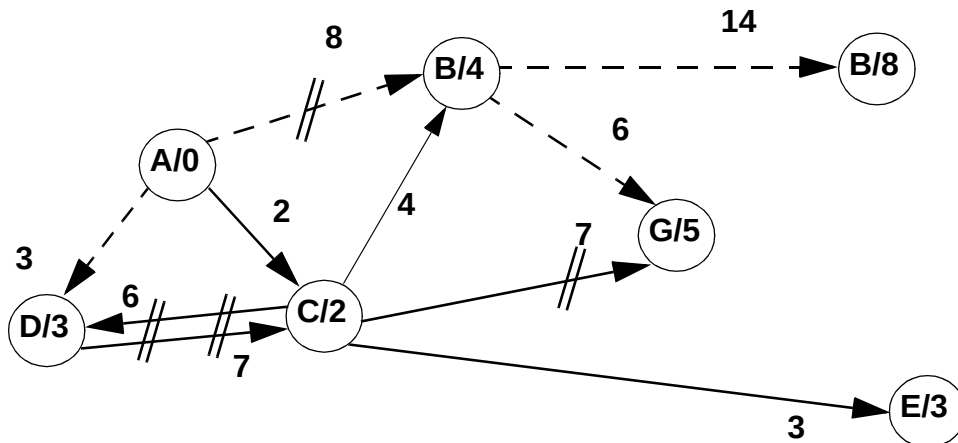
b) Markierung von Knoten A aus



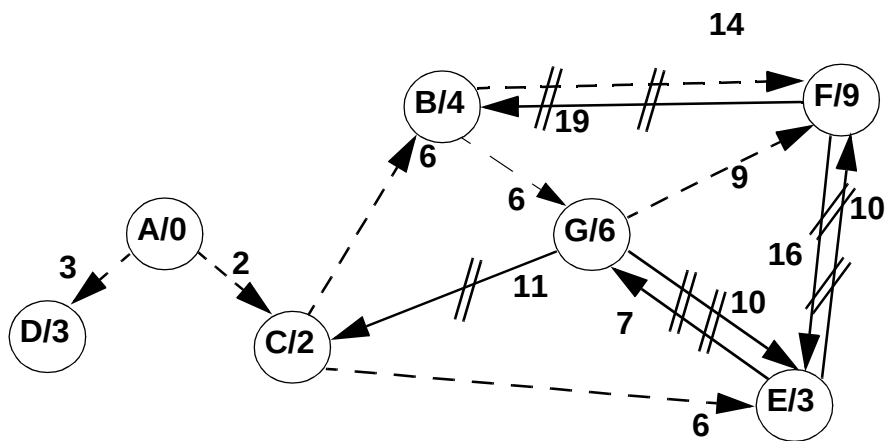
c) Markierung von Knoten B aus



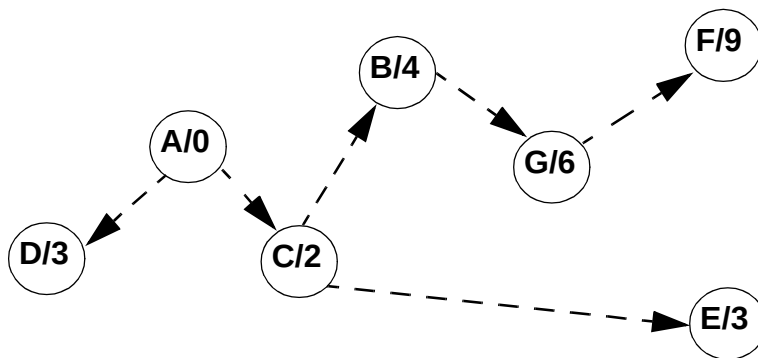
Spannbaum bzw. Kürzeste Wege (2)



d) Markierung von Knoten C, D aus (Breitendurchlauf im Baum)



e) Markierung von Knoten G, E, F aus



f) keine neuen Markierungen mehr möglich

==> Baum zeigt kürzeste Wege von A aus zu allen Knoten
= minimaler Spannbaum

Zusammenfassung statisches Routing

- **Eigenschaften:**

Tabellen werden vom Netzwerkoperator zentral erstellt; Tabellen werden vor Inbetriebnahme der Knoten geladen und dann nicht mehr verändert.

- **Vorteile:**

einfach

gute Ergebnisse bei relativ konstanter Topologie und konstantem Verkehr

- **Nachteile:**

schlecht bei stark variierendem Verkehrsaufkommen und bei Topologieänderungen

schlecht bei großen Netzen (skaliert nicht)

Verfeinerungen von Wegewahlalgorithmen

- Sowohl bei statischem als auch bei dynamischen (= adaptiven) Routing können 2 Verfeinerungen angebracht werden:
 - a) Mehrfachpfade (=> alternative Wege)
 - b) hierarchische Wegewahl (=> Vereinfachung der Verfahren)

Mehrfach-Pfade (Multipath routing)

Prinzip:

Benutzung alternativer Leitwege zwischen jedem Knoten-paar (i, j)

Vorteile

- alternative Wege könne gemäß ihrer „Güte“ ausgewählt werden („gute“ Wege häufig, schlechte seltener)
- Höherer Durchsatz durch Verteilung des Datenverkehrs auf mehrere Pfade
- Höhere Zuverlässigkeit, da der Ausfall eines Links nicht so schnell zur Unerreichbarkeit von Knoten führt

Definition:

guter Weg = schnellster/kürzester geringste Warteschlange/
höchste Bandbreite/höchste Zuverlässigkeit

Realisierung der alternativen Wegewahl

- Jeder Knoten enthält Routing-Tabelle mit je einer Zeile für jeden möglichen Zielknoten

eine Zeile {

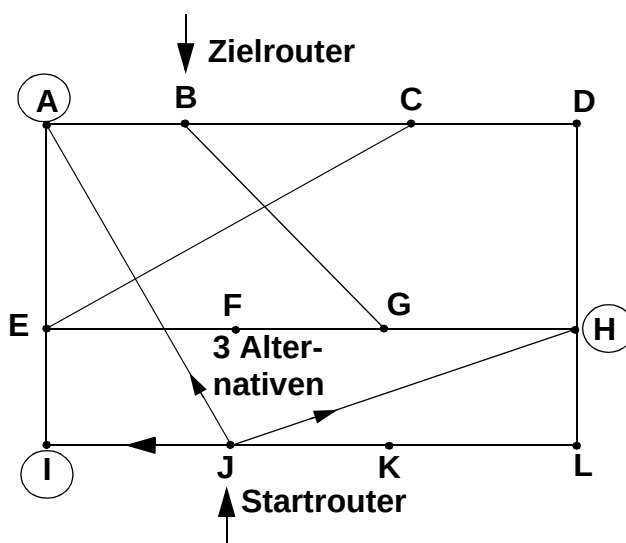
Ziel-knoten	A1	G1	A2	G2	...	An	Gn
-------------	----	----	----	----	-----	----	----

- A_i : i-beste Ausgangsleitung des jeweiligen Zwischenknotens
- G_i : Gewicht für A_i (drückt die Güte der Leitung aus)
 - G_i bestimmt die Wahrscheinlichkeit, mit der A_i benutzt

wird: $\sum_{i=1}^n G_i = 1$

Leitung hoher Güte wird häufig benutzt.

- Die Gewichte der Ausgangsleitungen werden gemäß der gewählten Metrik (schnellsten, kürzesten,...) bestimmt.
- Beispiel:

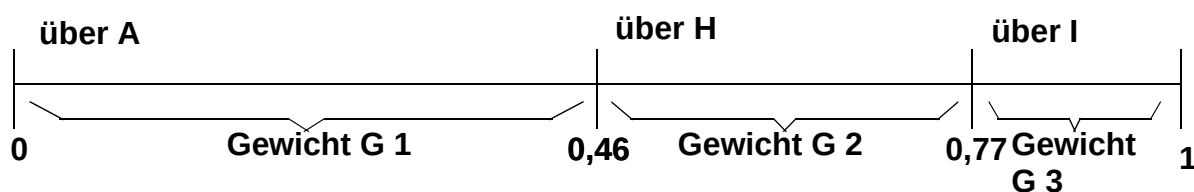


Beispiel für alternative Wegewahl (2)

- gegeben:

Ziel B, Quelle J, $G1 = 0,46$; $G2 = 0,31$; $G3 = 0,23$

- Gewichte werden auf dem Zahlenstrahl von 0 bis 1 aufgereiht (z. B. links das höchste, rechts das niedrigste; genaue Reihenfolge ist jedoch ohne Belang)



- Zufallsauswahl der alternativen Ausgangsleitung gemäß ihres Gewichts (= Güte)

Generieren einer Zufallszahl z ($0 \leq z \leq 0\ 1$)

Wähle A1, falls $0 \leq z \leq G1$

Wähle A2, falls $G1 \leq z < G1 + G2$

·
·
·

Wähle An, falls $G1 + G2 + \dots + G(n-1) \leq z < 1$

=> jeder Weg wird entsprechend seines Gütemaßes mehr oder weniger häufig benutzt

Zweite Verfeinerung bei allen Routing-Algorithmen

Hierarchische Leitwegbestimmung

Die **Größe der Routing-Tabellen** ist proportional zur Größe des Netzwerks:

- großer Speicherbedarf
- viel CPU-Zeit zum Durchsuchen der Tabellen
- viel Bandbreite zum Austausch von Routingformation

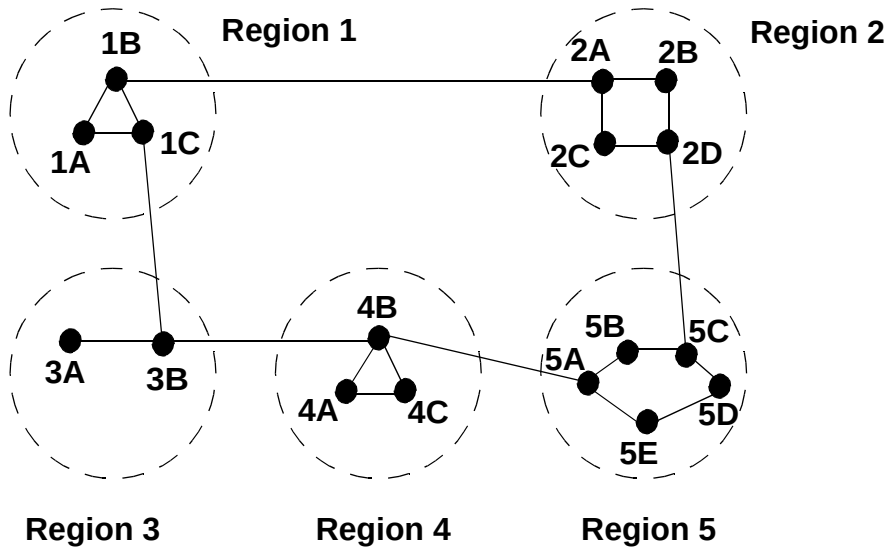
Hierarchische Leitwegbestimmung ab einer bestimmten Netzgröße notwendig:

- Knoten werden in Regionen gruppiert
- Jeder Knoten kennt
 - Details seiner Region
 - Leitweg zu allen anderen Regionen
 - aber nicht den Weg zu den Knoten außerhalb der eigenen Region, nur in welcher Region der Zielknoten liegt

Nachteil: nicht immer optimale Entscheidungen möglich

Hierarchische Wegewahl

Beispiel



Volle Tabelle für 1A

Ziel	Leitung	Teil-strecken
1A	-	-
1B	1B	1
1C	1C	1
2A	1B	2
2B	1B	3
2C	1B	3
2D	1B	4
3A	1C	3
3B	1C	2
4A	1C	3
4B	1C	4
4C	1C	4
5A	1C	4
5B	1C	5
5C	1B	5
5D	1C	6
5E	1C	5

Hierarchische Tabelle für 1A

Ziel	Leitung	Teil-strecken
1A	-	-
1B	1B	1
1C	1C	1
2	1B	2
3	1C	2
4	1C	3
5	1C	4

==> hierarchische Tabelle ist viel kürzer

B) Adaptive Wegewahl

1.) Zentrale Verfahren

2.) Lokale Verfahren

3.) Verteilte Verfahren

zentral heißt: einer führt den Algorithmus aus
 (= zentrale Instanz)

lokal heißt: alle Router führen den Algorithmus
 aus, aber ohne Kommunikation unter-
 einander (= isoliert)

verteilt heißt: alle Router führen den Algorithmus mit
 Kommunikation, d. h. Austausch von
 Information aus.

B. I Zentralisierte adaptive Leitwegbestimmung

Prinzip

- Im Netz: RCC (Routing Control Center, Leitwegsteuerzentrum)
 - Jeder Knoten sendet periodisch Status-Informationen zum RCC
 - Liste der verfügbaren Nachbarn
 - aktuelle Warteschlangenlängen
 - Auslastung der Leitungen etc.
- } Status jeden Knotens
- RCC sammelt Informationen und berechnet optimalen Pfad für jedes Knotenpaar, berechnet die einzelnen Leitwegtabellen und verteilt sie an Knoten.

Bemerkung:

Optimal heißt optimal bzgl. einer zuvor vereinbarten Metrik.

Zentralisierte adaptive Leitwegbestimmung (2)

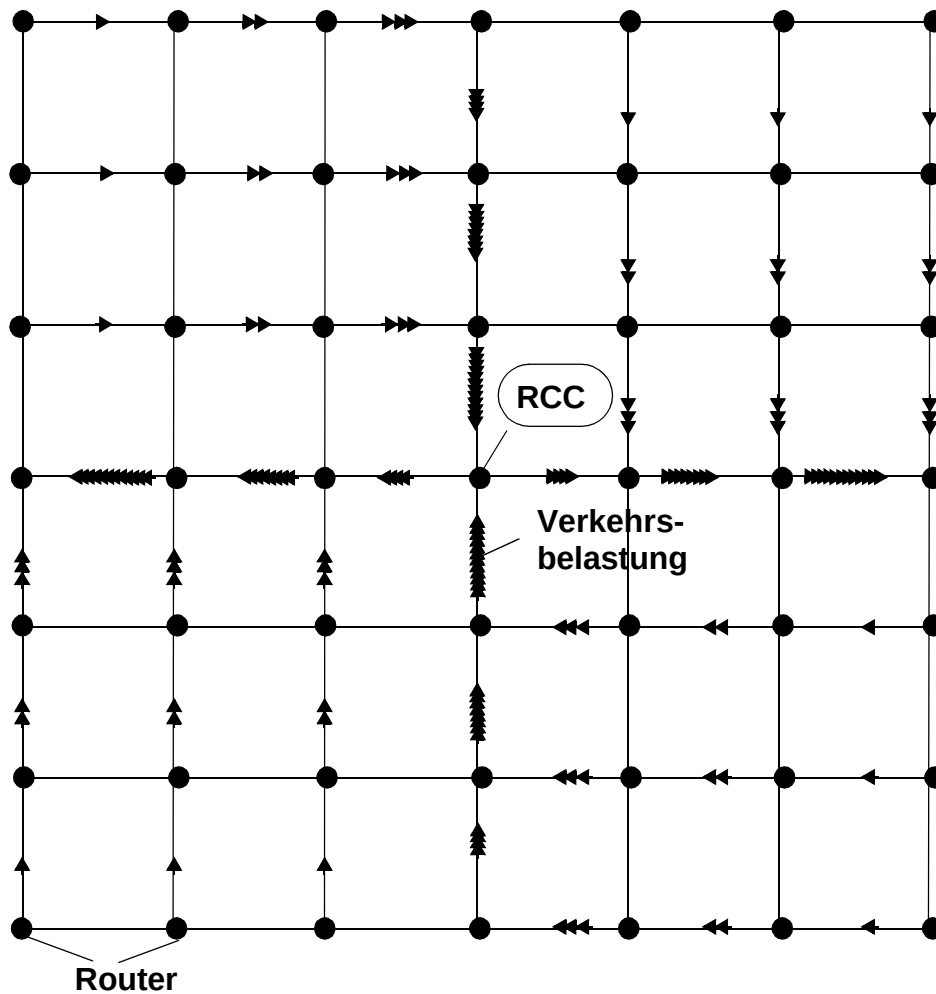
Eigenschaften

- RCC hat vollständige Informationen => die Entscheidungen sind optimal
- Die Einzelknoten sind von der Leitwegberechnung befreit

aber:

- Berechnung muss oft durchgeführt werden (ca. jede Minute oder öfter)
 - Verkehrskonzentration in der Nähe des RCC („performance bottleneck“)
 - geringe Robustheit („single point of failure“)
 - keine korrekte Entscheidung bei Netzpartitionierung
 - Knoten erhalten Tabellen zu unterschiedlichen Zeiten
- => Inkonsistenzen sind möglich

Beispiel für eine zentralisierte Leitwegbestimmung in einem Gitternetz



Bemerkung:

die Gitternetztopologie stellt eine sehr starke Idealisierung der Wirklichkeit dar.

Variante der zentralisierten, adaptiven Wegewahl (Delta-Routing)

- Routing berücksichtigt Wegealternativen

Prinzip:

- Jeder Knoten berechnet periodisch die „Kosten“ seiner Leitung und sendet diese zum RCC (Kosten = Funktion aus Verzögerung, Warteschlangenlänge,...)
- RCC berechnet
 - a) die k besten Pfade von Knoten i nach Knoten j (für alle i,j) und erstellt daraus eine
 - b) Liste der zum besten Pfad „äquivalenten“ Pfade
- Kriterium
$$c_{ij}^m - c_{ij}^1 < \delta \text{ mit } \delta = \text{constant}$$
wobei c_{ij}^m = Gesamtkosten des m-besten Pfads ($1 \leq m \leq d$)
 c_{ij}^1 = Gesamtkosten des allerbesten Pfads
- RCC sendet jedem Knoten für jedes mögliche Ziel eine Liste von äquivalenten Pfaden
- Jeder Knoten darf zwischen den *äquivalenten* Pfaden frei wählen, z. B.
 - zufällig, oder
 - die Leitung mit den aktuell geringsten Kosten
- Variante heißt Delta-Routing

Algorithmus „Delta Routing“ (2)

Wahl von δ bedeuten ein Verschieben der Autorität zwischen Knoten und RCC

$\delta \rightarrow 0$: RCC trifft Entscheidung (rein zentral, da nur 1 opt. Weg

$\delta \rightarrow \infty$: Knoten trifft Entscheidung (rein lokal, da alle Wege zur Auswahl)

Bei geeigneter Wahl δ von bessere Leistung als bei rein isolierten (= lokalen) oder zentralisierten Verfahren, da Wegealternativen berücksichtigt werden.

B. II Lokale adaptive Leitwegbestimmung

- Prinzip

- Kein Austausch von Routing-Information zwischen Knoten
- Entscheidungen basieren ausschließlich auf lokalen Informationen (= isoliert)

• Beispiele

1.) Flooding-Verfahren

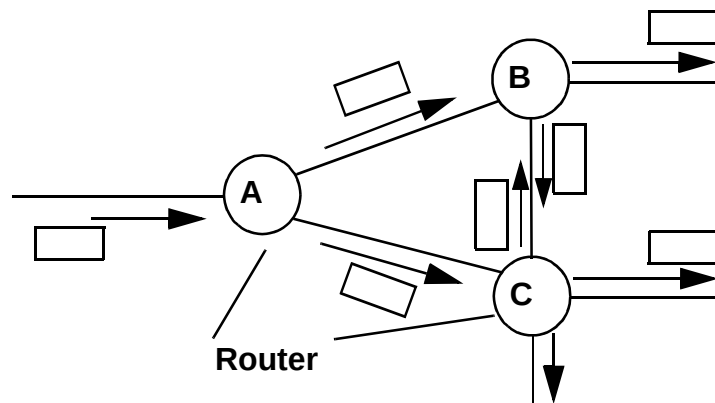
2.) Backward Learning-Verfahren

- Flooding = ineffizient aber robust
- Backward Learning = effizient aber heikel

B. II. 1 Flooding-Algorithmus

Ein empfangenes Paket wird auf alle Leitungen weitergeleitet außer auf derjenigen, auf der es angekommen ist.

Beispiel:



- Problem „unendliche“ Anzahl von Duplikaten
=> Begrenzung des Prozesses: Streckenzähler („hop counter“) im Paketkopf
- Initialisierung mit dem Durchmesser des Netzes
= längstem Pfad im Netz (worst case)
- Wird auf jede Teilstrecke um 1 decrementiert
- Duplikate erhalten den Streckenzähler des Originals
- Zähler = 0: Paket wird vom Router weggeworfen

Eigenschaften:

- sehr robust, sehr einfach, aber
- große Anzahl von Duplikaten, große Netzbelastung
=> Einsatz nur für sehr spezielle Anwendungen (militärisch)

B. II. 2 Algorithmus „Backward Learning“

- Knoten „lernt“ von eintreffenden Paketen anhand dessen Q und Z

geg.: Paket (... , Q, Z, ...) mit Q = Quell-Knoten, Z = Teilstreckenzähler (hop counter) = Zusatzinformation im Paket

Prinzip: Paket wird auf Leitung L empfangen
=> Q ist über L in Z Teilstrecken erreichbar
(in umgekehrter Richtung)

- Leitwegtabelle im Knoten wird aufgebaut (für jede Leitung L): „L-Tabelle“; Jeder Eintrag ist ein Tripel aus (Zielknoten, Ausgangsleistung, $Z_{\min}$)

$Z_{\min}$ = momentaner minimaler Hop Count zum Ziel

- Aktualisierung der Leitwegtaelle für jede Leitung L: Knoten empfängt Paket (... , Q, Z, ...) auf Leitung L

Algorithmus:

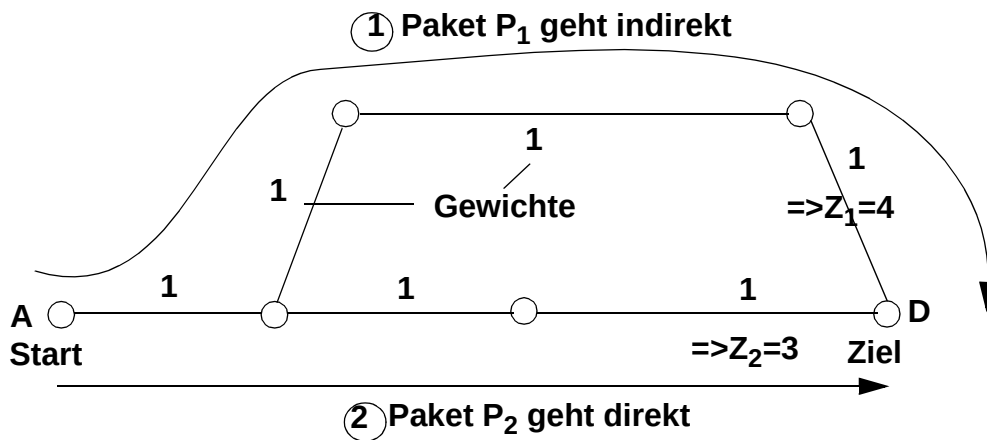
if not (Q in L-Tabelle) - - gibt es neuen Knoten?

then Add (Q, L, Z) - - neuen Knoten hinzufügen

else if Z < $Z_{\min}$ - - gibt es kürzeren Weg?

then Update (Q, L, Z) - - Tabelle aktualisieren

Backward Learning: Beispiel



Aus Paket P_1 (... , A, 4,...) $\Rightarrow$ Add (A, 1, 4) beim Ziel D

Aus Paket P_2 (... , A, 3,...) $\Rightarrow$ Update (a, 2, 3) beim Ziel D

Problem

Algorithmus registriert keine Verschlechterungen (nur dann Update, wenn der Weg kürzer ist)

Lösung

periodisches Löschen der Leitwegtabellen (neue Lernperiode); Löszeitpunkte kritisch:

- zu häufig: Netz ist überwiegend in der Lernphase
 - zu selten: zu langsame Reaktion auf Verschlechterungen
- $\Rightarrow$ heikel in der Praxis

•

B. III. Verteilte Leitwegbestimmung

- Möglichkeiten:

1.) „Distanz“vektor-Verfahren

2.) Link-Zustandsverfahren

3.) OSPF-Verfahren

- Alle drei Methoden werden bzw. wurden im Internet angewandt

B. III. 1. Das „Distanz“vektor-Verfahren

Prinzip:

Die Knoten tauschen mit ihren Nachbarn Leitweginformationen aus:

- Jeder Knoten kennt „Distanz“ (=Güte) zu jedem Nachbarn, z. B.:
 - Anzahl der Teilstrecken
 - Verzögerungszeit (wird über sog. Echopakete gemessen)
 - Warteschlangenlänge etc.
- Jeder Knoten sendet periodisch seinen Nachbarn eine Liste mit seinen gemessenen/geschätzten Güten zu jedem Ziel

z. B.: X empfängt Liste vom Nachbarn Y
sei Entfernung $(X, Y) = e$ und Z ein beliebiger Knoten
und Entfernung $(Y, Z) = E(Z)$ (E = Listeninformation)

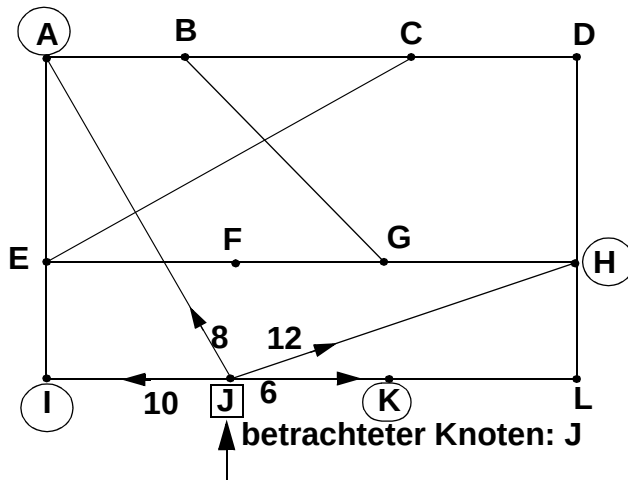
=> Entfernung (X, Z) über Y: $E(Z) + e$

=> nach jeder Periode konvergiert die Tabelle etwas näher zur Wahrheit hin (iteratives Verfahren)

Problem:

Der Ausfall eines Routers wird nur sehr langsam erkannt (= count-to-infinity-Problem), da Ausbreitung dieser Information im Netz viele Iterationen erfordert.

Beispiel für das „Distanz“vektor-Verfahren



- **J misst die Latenz zu seinen Nachbarn**

JA-Ver-
zöge-
rung = 8

JI-Verzö-
gerung =
10

JH-Ver-
zöge-
rung = 12

JK-Ver-
zöge-
rung = 6

- von seinen Nachbarn empfängt J die Latenzen zu allen anderen Knoten (vom jeweiligen Nachbarknoten aus gesehen)
 - Aus beiden Informationen baut J seine lokale Latenztabelle auf
 - Zuletzt verteilt J seine lokale Tabelle an seine Nachbarn
 - Vorgang wird periodisch wiederholt
- => es bildet sich ein eingeschwungener Zustand heraus

Beispiel für „Distanz“vektor-Verfahren (2)

- Tabellen, die J empfängt:

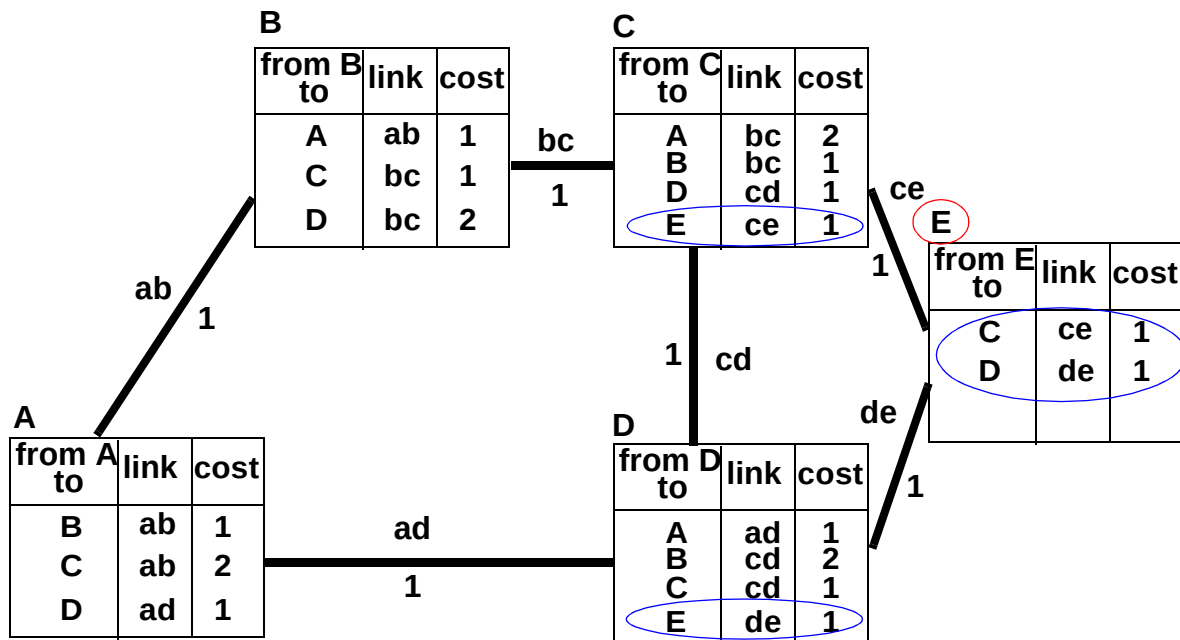
Ziel	von A	von I	von H	von K
A	0	24	20	21
B	12	36	31	28
C	25	18	19	36
D	40	27	8	24
E	14	7	30	22
F	23	20	19	40
G	18	31	6	31
H	17	20	0	19
I	21	0	14	22
J	9	11	7	10
K	24	22	22	0
L	29	33	9	9

- Beachte: die Latenzen $\overline{AJ}$, $\overline{IJ}$, $\overline{HJ}$ und $\overline{KJ}$, die die Nachbar-knoten sehen, sind anders, als die, die $\overline{J}$ sieht: $\overline{AJ} \neq \overline{JA}$ usw.
- J wählt aus den eigenen Messungen und den empfangenen Latenzen den besten Weg aus ($\overline{JA} = 8$, $\overline{JI} = 10$, $\overline{JH} = 12$, $\overline{JK} = 6$)

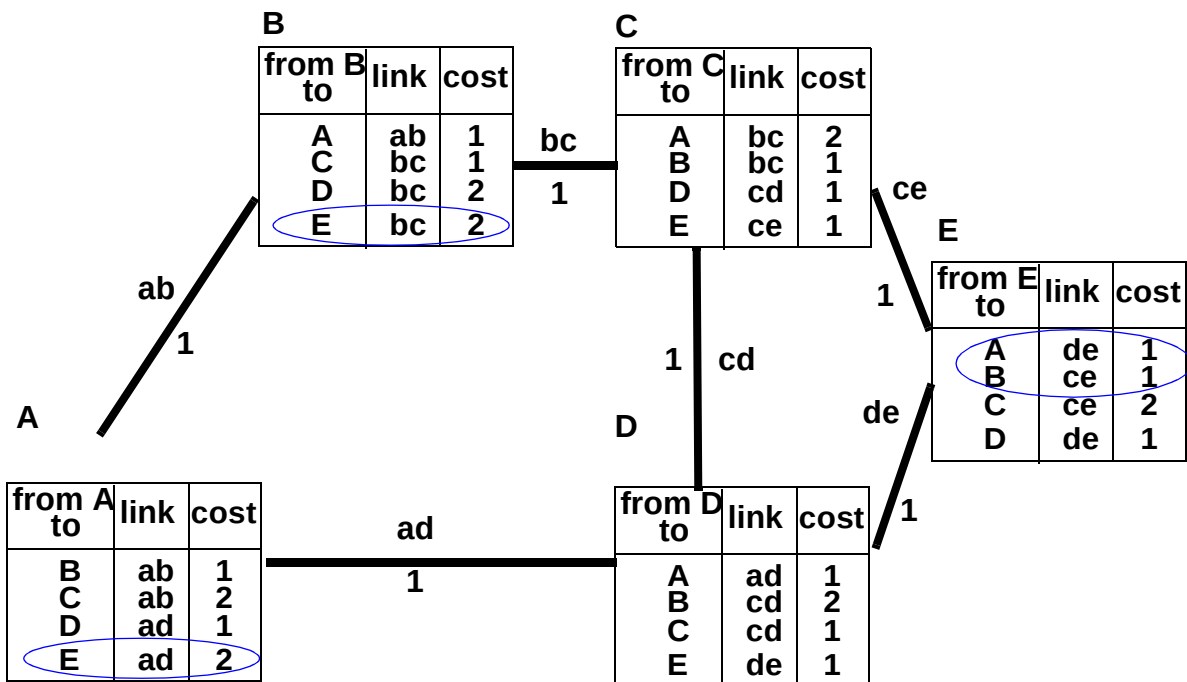
Ziel	Latenz	über
A	8	A
B	20	A
C	28	I
D	20	H
E	17	I
F	30	I

Ziel	Latenz	über
G	18	H
H	12	H
I	10	I
J	0	-
K	6	K
L	15	K

2. Beispiel für Routing mit Distanzvektoren



a) Situation: Es ist neu hinzukommen ==> E gibt es noch nicht in den A, B-Routing-Tabellen



b) Situation nach mehreren Iterationen: E ist auch bei A, B bekannt, und umgekehrt ist A, B bei E bekannt

B. III.2 Das Link-Zustandsverfahren

- Prinzip:

- Jeder Router in einem Subnetz ist im Vollbesitz aller „Güte“-Werte des Subnetzes
- Dazu sendet jeder Router Datenpakete an alle anderen Router im Subnetz, nicht nur an seine Nachbarn
- Entsprechend empfängt jeder Router Pakete von allen anderen Routern
- Dadurch kann jeder Router lokal die Topologie des ganzen Subnetzes rekonstruieren
- Als Gütewerte werden die üblichen Metriken (Distanz, Latenz,...) verwendet
- Mit Hilfe der Topologie und der Güte jeder Verbindung kann jeder Router für sich einen minimalen Spannbaum aufbauen

B. III. 3 Das OSPF-Verfahren

OSPF = open „shortest path first“

Prinzip:

- **Wie das Link-Zustandsverfahren, aber mit einer Reihe von Erweiterungen**
 - 1.) unterstützt mehrere Metriken**
 - 2.) berücksichtigt alternative Wege**
 - 3.) unterstützt hierarchisches Routing (Link-Zustandsverfahren in jeder Region)**
 - 4.) ist sicherer bezügl. böswilliger Manipulation der Routing-Tabellen**

Eigenschaften:

- **Spannbaum wird gem. Dijkstra berechnet (von jedem Knoten)**
- **Topologie und Güte jeder Wegstrecke (= Kanten des Graphen) werden in einer Datenbank gehalten (nicht in einer Tabelle), und zwar in jedem Knoten für sich**

Routing im Internet

- Bis 1979 wurde das „Distanz“-vektorverfahren verwendet. Das Verfahren wurde mit Hilfe des sog. Rip-Protokolls implementiert.
- Wegen des count-to-infinity-Problems wurde es durch das Link-Zustandsverfahren abgelöst.
- Bis 1990 wurde das Link-Zustandsverfahren verwendet (danach auch noch) und danach durch das OSPF-Verfahren (meistens) ersetzt.
- De facto existieren noch alle 3 Verfahren im Internet und werden von den Routern unterstützt.

Beispiele für Protokolle der Vermittlungsschicht

- IP
 - X.25
 - ATM
- } 3 Beispiele

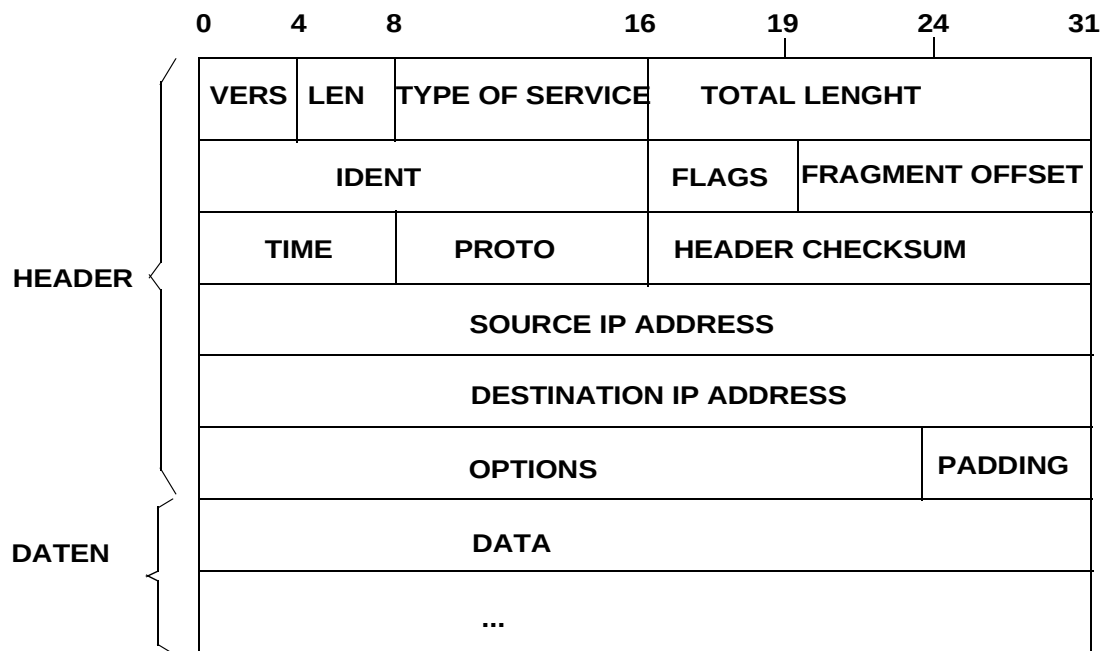
IP (Internet Protocol)

- Ein Datagramm-Protokoll
- Ein Host-zu-Host-Protokoll (Ende-zu-Ende)
- Handhabt die Fragmentierung großer Pakete: große Dienst-Datagramme können in kleinere Protokoll-Datagramme „fragmentiert“ werden.
- Macht die Wegewahl (Routing)

Eigenschaften:

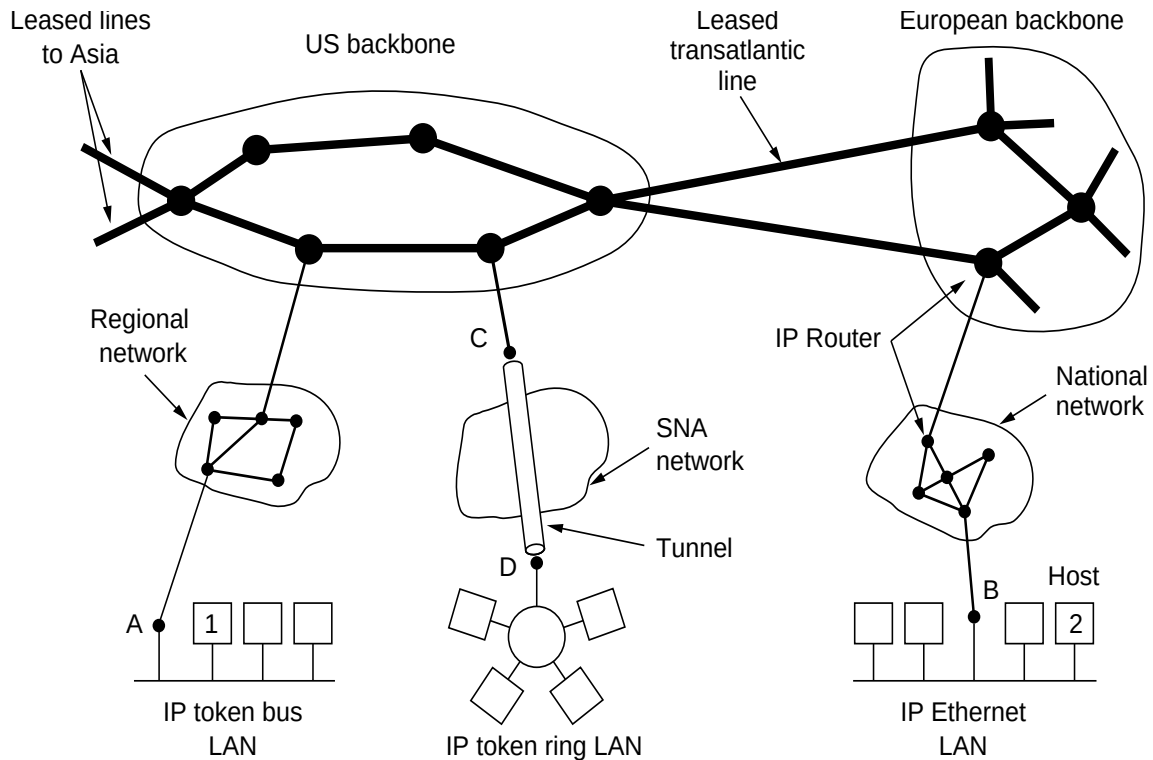
- verbindungsloser Dienst
- unzuverlässig (schützt nicht vor Paketverlusten)
- IP-Paketlänge: maximal 64 KB

Format von IP-Datagrammen



VERS	4b	Protokollversion von IP z. Z. IP v. 4
LEN	4b	Länge des Headers (Wörter)
TYPE OF SERVICE	8b	QoS (Priorität und D/T/R) Delay, Throughput, Reliability
TOTAL LENGHT	16b	Länge incl. Daten in Bytes
IDENT	16b	Identität des Datagramms (zu welchem Datagramm Paket nach Fragmentierung gehört)
FLAGS	3b	nicht fragmentieren/letztes Fragment (DF/LF)
FRAGMENTOFFSET	13b	Offset dieses Teils innerhalb von IDENT in 8-Byte-Portionen
TIME	8b	Lebensdauer in Sekunden („time to live“) ≤ 255 s
PROTO	8b	Type des höheren Protokolls (UPD, TCP)
HEADER CHECKSUM	16b	EXOR der Header-Wörter
SOURCE ADDRESS	32b	IP Adresse des Quell-Hosts
DEST ADDRESS	32b	IP-Adresse des Ziel-Hosts
OPTIONS		Kommandocode für Netzmanagementdatagramme
PADDING		Auffüllen auf Wortgrenzen
DATA		Nutzdatenfeld

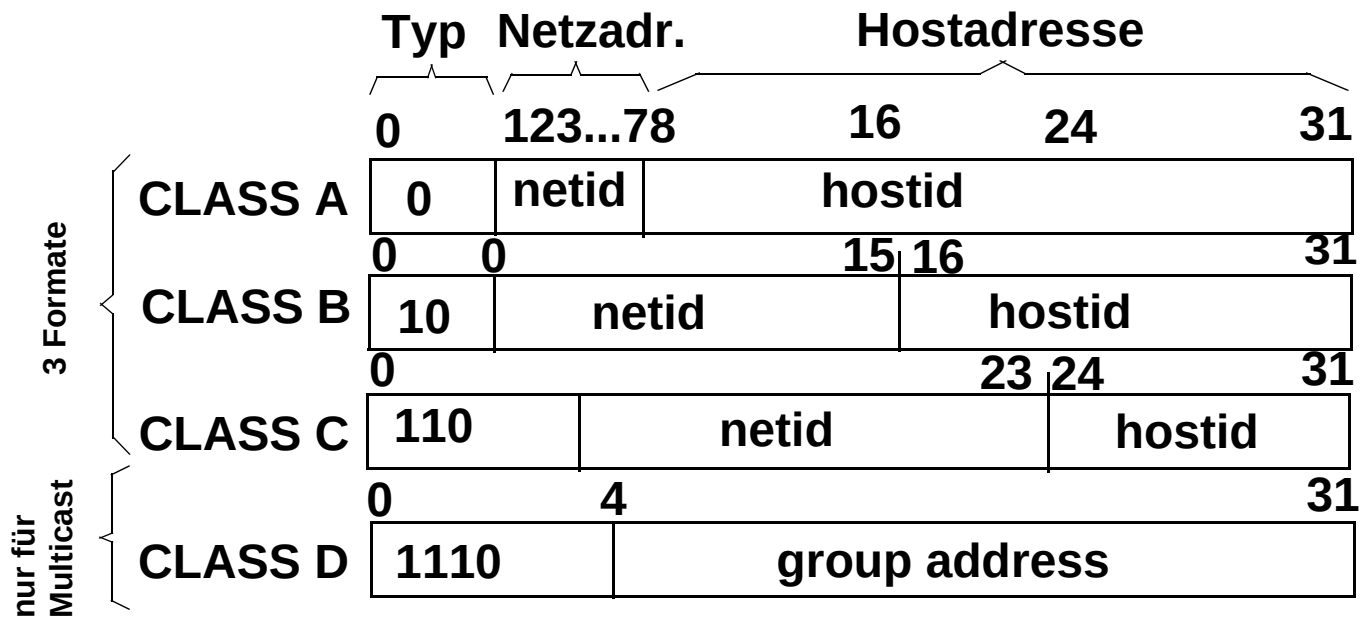
Wie funktioniert die Adressierung im Internet?



- **Das Internet besteht aus einer riesigen Kollektion unterschiedlicher Netze, die durch IP adressiert werden (>> 100.000 Netze)**
- **IP = Internet Protocol = Protokoll der Vermittlungsschicht im Internet**
- **adressiert wird über die IP-Adresse (= 32 Bit-Feld „Destination address“)**

Adressierung im Internet (2)

- Die IP-Adresse ist eine hierarchische Adresse mit Netz- und Hostidentifikationsnummer (netid und hostid). Es gibt drei Formate für Subnetze unterschiedlicher Größe sowie ein Format für Multicast:



Gebräuchlich ist seltsamerweise eine dezimale Schreibweise mit einer Zahl pro Byte. Beispiel:

10.0.0.0 für Arpanet (class A)

128.10.0.0 für ein großes Ethernet-LAN (class B)

192.5.48.0 für ein kleines Ring-LAN (class C)

IP-Adreßbereiche

o Die Internet-Adreßbereiche werden zentral vom Network Information Center (NIC) vergeben (<http://www.nic.com>)

Es gibt die folgenden Adreßbereiche:

Klasse-A-Netze: 1.0.0.0 bis 127.255.255.255

Klasse-B-Netze: 128.0.0.0 bis 191.255.255.255

Klasse-C-Netze: 192.0.0.0 bis 223.255.255.255

Multicast: 224.0.0.0 bis 239.255.255.255

reserviert: 240.0.0.0 bis 247.255.255.255

unbelegt: 248.0.0.0 bis 255.255.255.255

Es gibt:

- o 126 Klasse-A-Netze, mit jeweils bis zu 16M-2 Hosts
- o 16K-1 Klasse-B-Netze, mit jeweils bis zu 64K-2 Hosts
- o 2M-1 Klasse-C-Netze, mit jeweils bis zu 254 Hosts

o Bei den Adreßbereichen gibt es drei Adressen mit spezieller Bedeutung:

1) der Wert 0 bezeichnet das eigene Netz bzw. den eigenen Host

- o Wenn Netzadresse = 0 dann eigenes Netz
- o ((Wenn Netzadresse = eigenes Netz) oder (Wenn Netzadresse = 0)) und wenn Hostadresse = 0, dann eigener Host

2) der Wert -1 (=alle Bits gesetzt) sagt, daß ein Broadcast durchgeführt werden soll

- o ((Wenn Netzadresse = eigenes Netz) oder

(Wenn Netzadresse = 0))

und wenn Hostadresse = -1, dann Broadcast an alle Hosts im eigenen Netz

- o Wenn Netzadresse = x und wenn Hostadresse = -1 dann Broadcast an alle Hosts im Netz x**

3) der Wert 127 als Netzadresse sagt, daß ein Loopback-Test durchgeführt werden soll (Loopback = Host sendet an sich selber)

Wenn Netzadresse = 127 dann Loopback

Bemerkung:

- wegen der Loopback-Netzadresse und der „0“-Netzadresse gibt es nur 126 Klasse-A-Netze statt 128**
- wegen der „0“-Netzadresse gibt es nur 16K-1 Klasse-B- bzw. 2M-1 Klasse-C-Netze**
- wegen der „0“-Hostadresse bzw. der „-1“-Hostadresse gibt es nur 16M-2 Klasse-A-, 64K-2 Klasse-B- bzw. 254 Klasse-C-Hosts**
- Broadcasts über -1 werden nicht von Routern in andere Netze übertragen, sondern bleiben lokal!**

Subnetze

o Beim Internet gibt es die Möglichkeit, einen Teil der Host-adresse als Subnetzadresse lokal zu vergeben

Beispiel:

Die Uni Augsburg ist ein Klasse-B-Netz mit 137.250 als Netz-adresse. Die verbleibenden 16 Bit Hostadresse können z.B. aufgeteilt werden in 8 Bit Subadresse und 8 Bit Host-Adresse. Jedem Fachbereich kann dann vom lokalen Netzwerkadministrator eine Subadresse zugewiesen werden. Der Fachbereich Informatik z.B. hat als Subnetzadresse 71, so daß dort alle IP-Adressen 137.250.71.x lauten (ist äquivalent mit 137.250.71.0)

o Achtung:

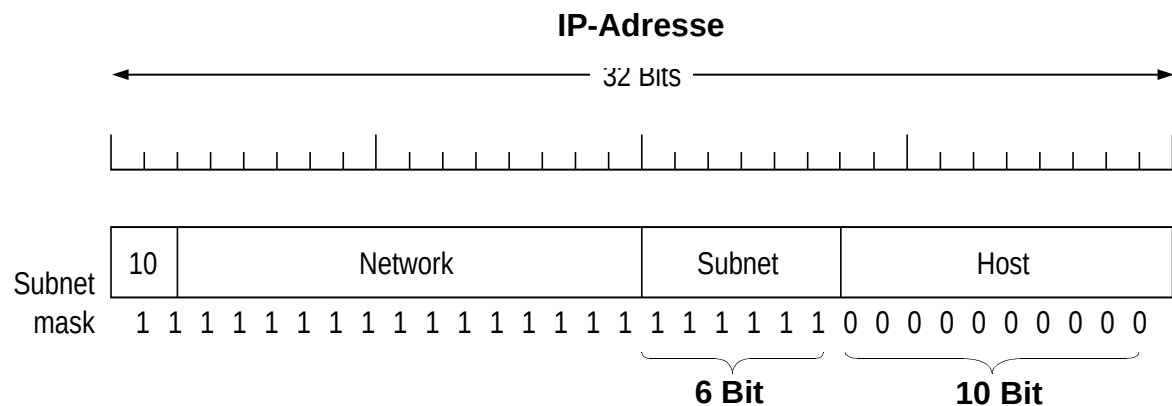
Ein IP-Subnetz hat zunächst nichts mit einem Subnetz der Schicht 1 und 2 zu tun! Beispielsweise können mehrere LANs ein gemeinsames IP-Subnetz bilden

o Die Verwaltung der Subnetze geht über Subnetzmasken, die mit der IP-Adresse UND-verknüpft werden.

Beispiel:

137.250.71.x AND 255.255.255.0 liefert das Informatik-Subnetz 137.250.71.0 (255.255.255.0 ist die Subnetzmaske)

Zweites Beispiel für ein Subnetz



- o Klasse-B-Netz, organisiert in 62 Subnetzen mit max. 1022 Hosts pro Subnetz (16 Bit = 6 Bit + 10 Bit)
- o Subnetzmaske: 255.255.252.0

Vorteile von Subnetzen

- o Die Router müssen bzgl. ihrer Ausgangsleitungen nicht eine große Zahl von Hosts (bis zu 16M-2 bzw. 64K-2 Hosts) verwalten, sondern nur wissen, über welche Leitung man zum jeweiligen Subnetz kommt
- => erhebliche Einsparung bei den Routing-Tabellen
- => Die Wegewahl erfolgt von Subnetz zu Subnetz (hierarchisch)
- => Einführung einer dreistufigen Hierarchie (Netzklasse, Netzadresse, Subnetzadresse)
- o Subnetze sind von außen nicht erkennbar, deshalb kann die Organisation in Subnetze ohne NIC gemacht werden!

Wie funktioniert die Adressauflösung im LAN?

Problem:

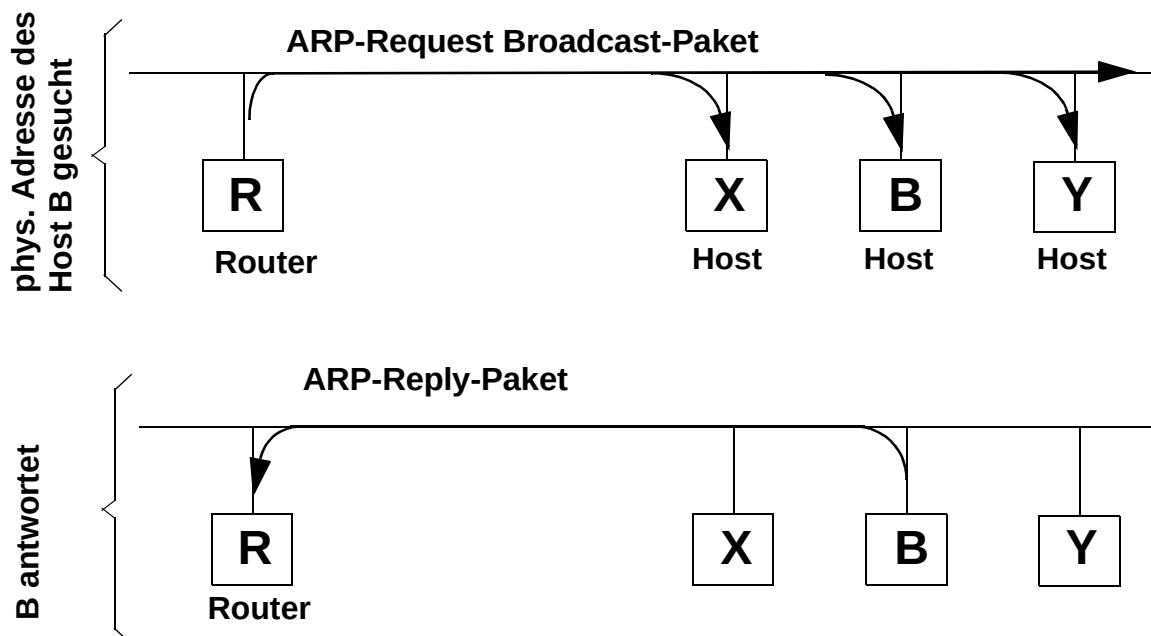
Wie soll die Abbildung der Internet Adresse (IP-Adresse) eines Rechners auf die physikalische Stationsadresse im LAN (IEEE 802-Adresse) erfolgen?

- 1.) Wenn die physikalische Adresse (Stationsadresse) durch den Benutzer ausgewählt werden kann, wähle LAN-Adresse = IP-Adresse
- 2.) Wenn die physikalische Adresse vorkonfiguriert („fest verdrahtet“) ist, unterhalte eine Abbildungstabelle (z. B. im Router) und/oder benutze das Address Resolution Protocol ARP. Dieser Fall ist bei Ethernet/Token Ring immer gegeben

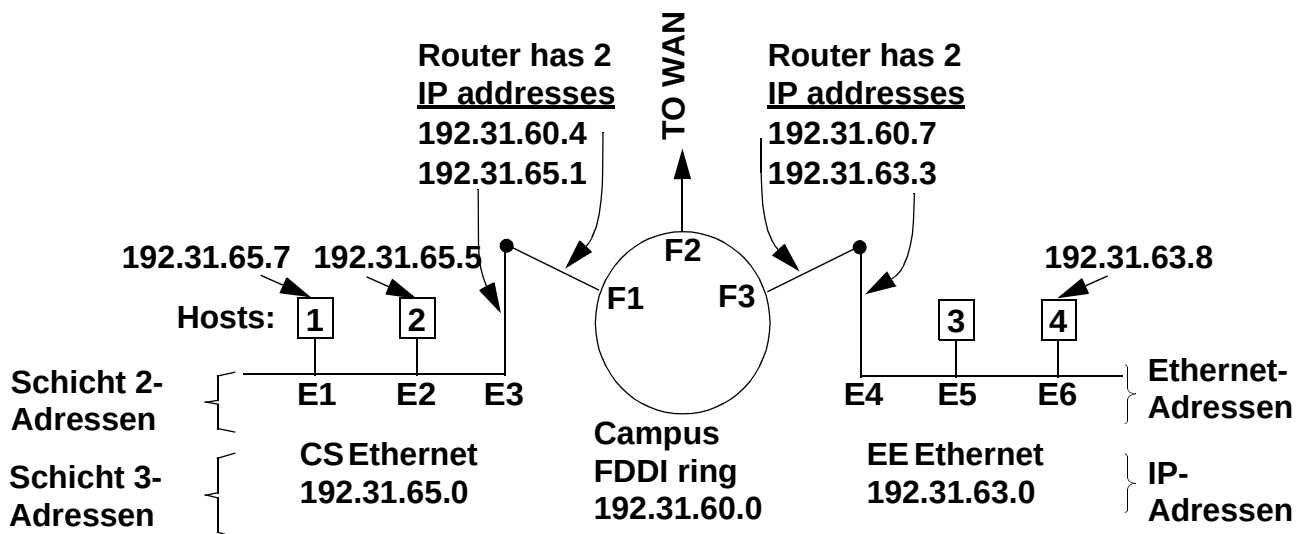
Address Resolution Protocol ARP

- Das Protokoll im Router läuft folgendermaßen ab:

- 1.) Sende mittels Broadcast auf dem LAN ein ARP-Request-Paket, welches die physikalische und die Internet-Adresse des Senders und die Internet-Adresse des gesuchten Empfängers enthält.
- 2.) Warte auf die Antwort des Empfängers durch ein ARP-Reply-Paket, welches seine physikalische Adresse enthält.
- 3.) Unterhalte einen Cache aus (I, P)-Adresspaaren für spätere Anfragen.
- 4.) Verbesserung: Der Empfänger des ARP-Requests speichert das (I, P)-Paar des Senders auch in seinem Cache.



Routing-Beispiel mit ARP



• Konfiguration des Beispiels:

- 3 Klasse-C-Netze (max. 254 Rechner) bestehend aus:
 - 1 Ethernet für Fachbereich Computer Science (CS)
 - 1 Ethernet für Electrical Engineering (EE)
 - 1 FDDI als Campus-Backbone

• Eindeutige Ethernet-Adressen E1 bis E6 für die Hosts und Router

• Eindeutige FDDI-Adressen F1 bis F3 für die Router

• Fall 1: Host 1 sendet an Host 2

1.) Transportschicht übergibt der Vermittlungsschicht (=IP) von Host 1 ein Paket mit 192.31.65.5 als IP-Adresse. IP-Adresse wird über das DNS gewonnen.

2.) IP von Host 1 schickt ARP-Request-Paket los mit 192.31.65.5 als Suchadresse für Schicht 3.

- 3.) Host 2 schickt ARP-Reply-Paket an Host 1 mit Internet-/Ethernet-Adresspaar (192.31.65.5, E2) im Datenteil.
- 4.) IP von Host 1 reicht Paket an die Sicherungsschicht weiter (genauer an MAC-Teilschicht) mit Ethernet-Adresse E2 als Zieladresse.
- 5.) MAC-Teilschicht von Host 1 baut Paket von der IP-Schicht in den Datenteil eines Ethernet-Rahmens ein und schickt ihn los an Ethernet-Adresse E2.

• **Fall 2: Host 1 sendet an Host 4**

- 1.) Transportschicht übergibt Paket mit 192.31.63.8 im Adressfeld. IP wird wiederum über das DNS gewonnen.
- 2.) IP schickt ARP-Request-Paket 192.31.63.8 als Suchadresse los.

Unterfall 1:

lokaler Router ist ein Proxy-Router und hat Tabelleneinträge für das entfernte Ethernet (Proxy = Stellvertreter)

3.1.1) lokaler Router (E3) schickt ARP-Reply-Paket an Host 1 mit (192.31.63.8, E3)-Adresspaar, d. h. mit sich selber

3.1.2) Host 1 sendet Paket an lokalen Router

Unterfall 2: lokaler Router ist kein Proxy-Router

3.2.1) Host 1 enthält kein ARP-Reply-Paket

3.2.2) Host 1 sendet Paket an fest vorgegebene Ethernetadressen, die den Fernverkehr abwickeln, im Beispiel auch E3.

- 3.) lokaler Router empfängt das Fernverkehrpaket**
- 4.) lokaler Router (E3) entnimmt die IP-Adresse dem Nutzdatenfeld des Ethernet-Rahmens und durchsucht seine Routing-Tabelle**
- 5.) Tabelle sagt, dass Pakete für Netz 192.31.63.0 über 192.31.60.7, d. h., den entfernten Router gehen**
- 6.) lokaler Router schickt ARP-Request-Paket in den FDDI-Ring mit 192.31.60.7 als Suchadresse**
- 7.) entfernter Router antwortet mit (192.31.60.7, F3)**
- 8.) lokaler Router schickt FDDI-Rahmen an F3**
- 9.) entfernter Router entnimmt das Paket des lokalen Routers dem Nutzdatenfeld des FDDI-Pakets und übergibt es an die IP-Schicht**
- 10.) entfernter Router schickt ARP-Request-Paket los mit 192.31.63.8 als Suchadresse**
- 11.) Host 4 antwortet mit ARP-Reply (192.31.63.8, E6)-Adresspaar**
- 12.) entfernter Router entnimmt dem Datenteil des Pakets vom lokalen Router das Paket von Host 1**
- 13.) IP-Schicht des entfernten Routers übergibt Paket vom Host 1 an MAC-Schicht mit Ethernet-Adresse E6**
- 14.) MAC-Schicht baut Paket in den Datenteil eines Ethernet-Rahmens ein und schickt ihn los**
- 15.) Host 4 empfängt das Paket von Host 1**

Weitere Internet-Steuerprotokolle

- RARP = Reverse Address Resolution Protocol
- BOOTP = Bootstrap Protocol
- ICMP = Internet Control Message Protocol

RARP

umgekehrte Problemstellung:

Die Ethernetadresse eines Hosts ist bekannt, gesucht ist seine Internetadresse

- Anwendung:
 - Diskless Host soll in einem Netz neu gebootet werden => Ethernet-Adresse des Hosts ist bekannt
 - Neuer Host soll in ein Netz eingefügt werden
- => RARP-Broadcast-Paket wird vom Host ins LAN geschickt (über -1 als Host-Adresse)**
- Wenn **RARP-Server** im LAN vorhanden, schickt er Antwort zurück
 - Nachteil bei RARP: wegen „-1“-Broadcast darf kein Router zwischen Host und RARP-Server liegen
- => BOOTP notwendig als Protokoll zum Booten über Router-grenzen hinweg.**

BOOTP

Anwendung:

- Diskless Host soll übers Netz gebootet werden
- gesuchte IP-Adressen sollen über Router hinweg verteilt werden (RARP geht nicht)
- Funktion: Boot Server schickt mit Hilfe von BOOTP Datagramme an zu bootende Host
- BOOTP Datagramme werden im Datenteil von IP-Paketen transportiert.
- Datagramme enthalten:
 - das Betriebssystem des zu bootenden Hosts als 1:1 Speicherabbild
 - die IP-Adresse des Boot Servers
 - die IP-Adresse des Vorgabe-Routers für Fernverkehr
 - die IP-Adresse des zu bootenden Hosts
 - die zu verwendende Subnetzmaske

ICMP

- der Betrieb des Internet wird eng von seinen Routern überwacht. Passiert etwas Ungewöhnliches, wird dies über ICMP-Pakete verbreitet
- Daneben wird ICMP als Testhilfe verwendet

- ICMP Datagramme werden im Datenteil von IP-Paketen transportiert
- ICMP-Pakettypen:

Message type	Description
Destination unreachable	Packet could not be delivered
Time exceeded	Time to live field hit 0
Parameter problem	Invalid header field
Source quench	Choke packet (für Staukontrolle)
Redirect	Teach a router about geography
Echo request	Ask a machine if it is alive
Echo reply	Yes, I am alive
Timestamp request	Same as Echo request, but with timestamp
Timestamp reply	Same as Echo reply but with timestamp

- **Die wichtigsten ICMP-Pakete:**

- **Destination unreachable:**

- Router kann das Ziel nicht finden
- IP-Paket mit Don't Fragment-Bit können nicht zugestellt werden, weil das Paket zu lang für das zu durchquerende Teilnetz ist

- **Time exceeded:**

- IP Paket wird weggeworfen, weil das Time-to-live-Feld im IP-Header auf Null steht (hop count-Zähler auf 0)

- **Parameter Problem:**

- Header eines IP-Pakets hat einen Fehler

- **Source quench:** (to quench = quetschen)
 - Sender soll weniger Pakete schicken (veraltet), um Netzüberlastung zu vermeiden
- **Redirect:**
 - Router hat Fehler eines anderen Router festgestellt und versucht, ihn davon zu unterrichten, in der Hoffnung auf Besserung
- **Echo request/Echo reply:**
 - dient zum Test, ob ein Host erreichbar ist
- **Timestamp request/Timestamp reply:**
 - wie Echo request/Echo reply, aber mit Zeitstempel

Praktische Beispiele

- Gebe im Kommandofenster eines NT-Rechners die folgenden Kommandos ein:
 - arp -h
 - netstat -h
 - ping -h
 - (ifconfig bei Unix)

Die Transportschicht (Schicht 4)

Beispiel: ISO-Transportschicht

Beispiel: TCP-Transportschicht im Internet

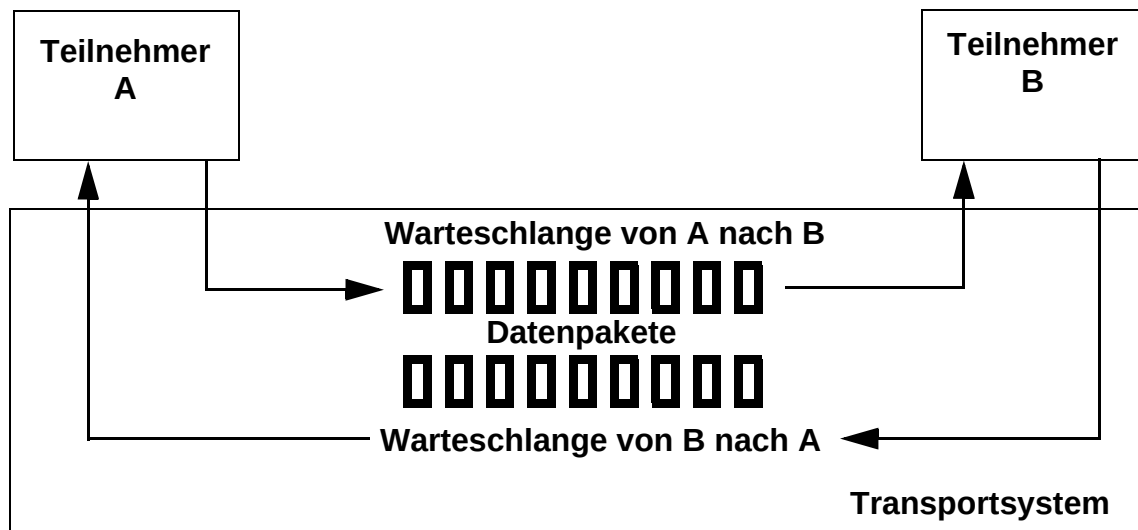
Beispiel für eine Transportschicht: Der ISO-Transportdienst

**Verbindungsorientierter Transportdienst
(ISO IS 8072, CCITT X.214)**

Eigenschaften/Aufgaben:

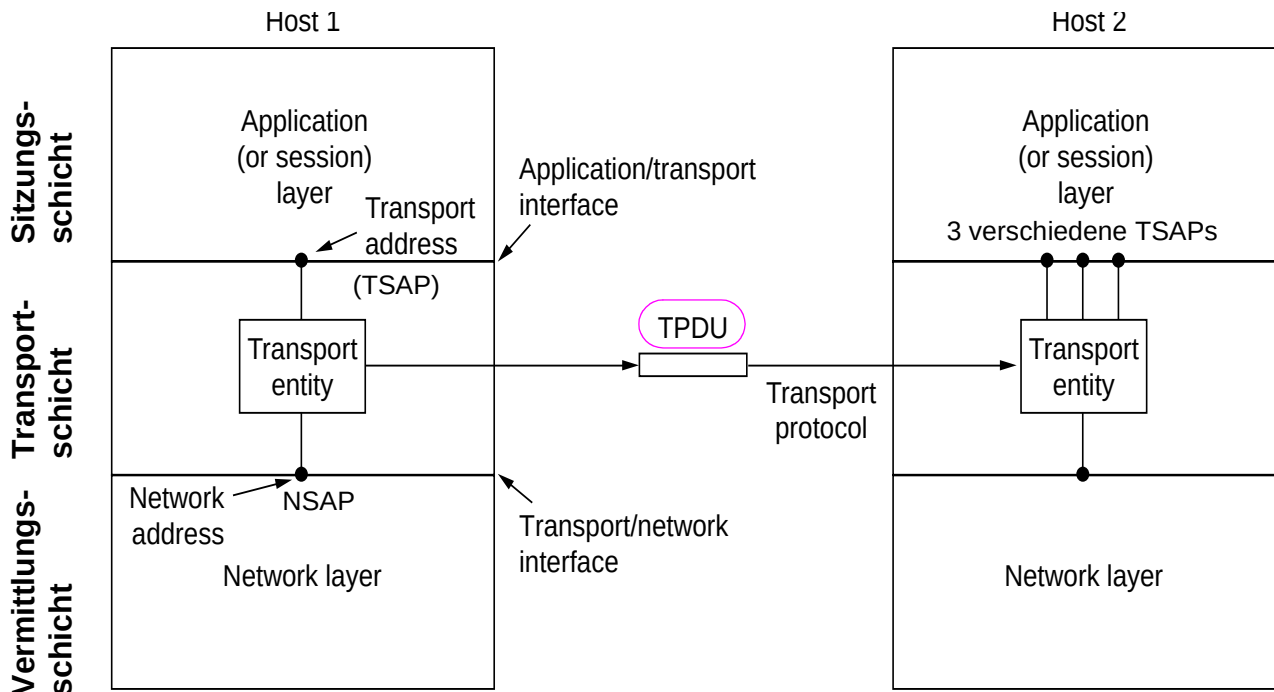
- **Aufbau einer Transportschichtverbindung**
- **Datenübertragung**
- **optional beschleunigte Datenübertragung (expedited data)**
- **Abbau einer Transportschichtverbindung**
- **Eingeschränkte QoS-Parameter (QoS = Quality of Service)
(QoS ist höher als bei TCP/IP, aber niedriger als bei ATM)**

ISO-Modell des Transportdienstes (gilt auch bei TCP)



- Vollduplex-Datenübertragung
- Übergebene Datenpakete werden zuverlässig in der richtigen Reihenfolge übertragen (FIFO-Reihenfolge)
- Ausnahme: beschleunigte Daten (expedited data). Diese halten untereinander die Reihenfolge ein, werden aber mit normalen Daten-PDU's verschränkt übertragen. Eine separate Warteschlange mit höherer Priorität kann, muss aber nicht existieren.
- durch den Dienst der Transportschicht erleiden Datenpakete nur eine gewisse Verzögerung. Alle anderen Fehler und Unzulänglichkeiten der Sicherungsschicht werden behoben.

Beziehung zwischen der Transportschicht und benachbarten Schichten nach ISO



- die von der Transportschicht gesendeten Pakete heißen im ISO-Jargon TPDU (Transport Protocol Data Unit)
- die Unterscheidung zwischen verschiedenen Instanzen der Transportschicht erfolgt über sog. Transportschicht-adressen, die im Jargon der ISO TSAP heißen, im Gegensatz zu den NSAPs der Sicherungsschicht

Transport Entity = Transportinstanz
= Software, die ein Protokoll der Transportschicht implementiert

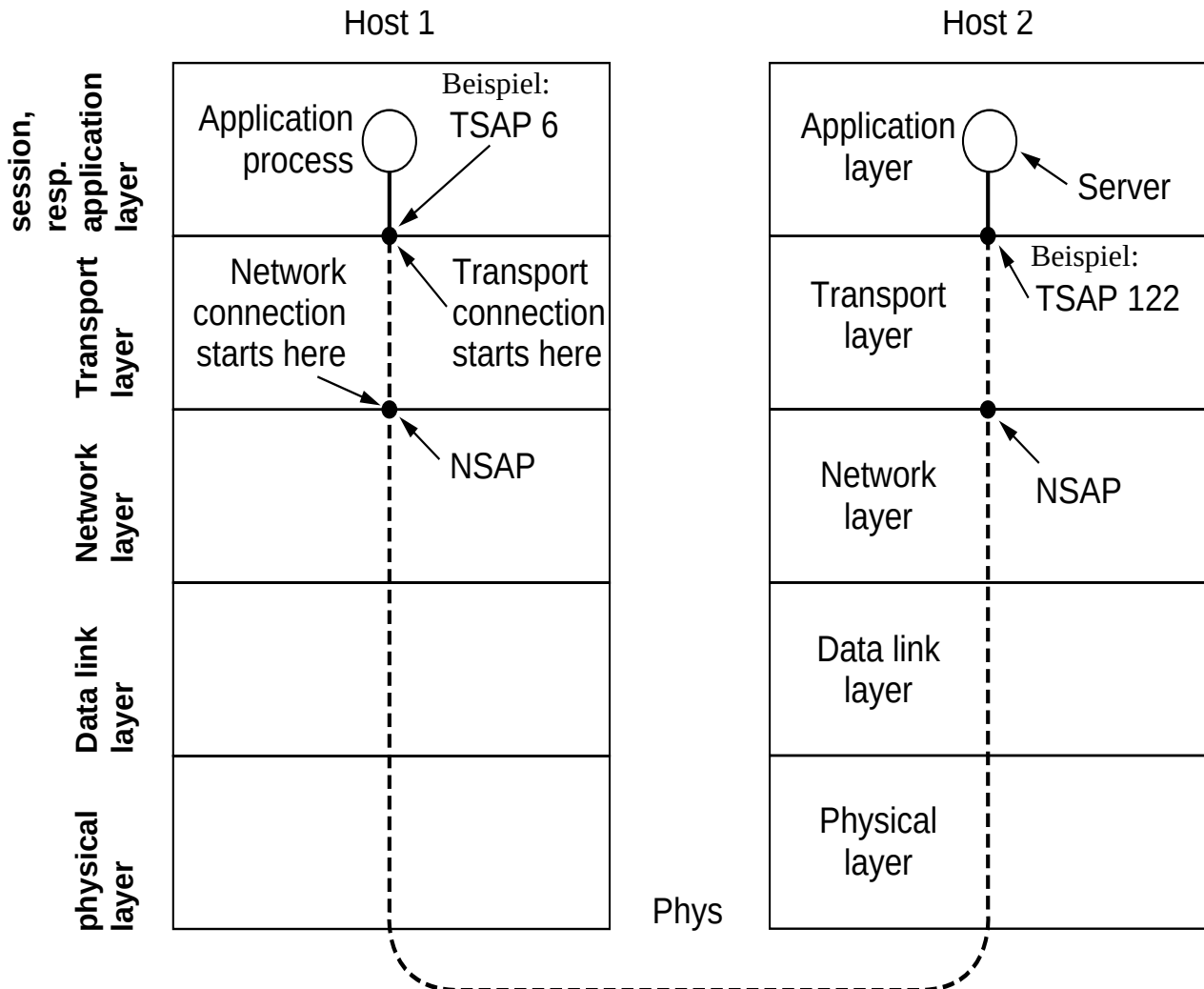
NSAP = Network Service Access Point

TSAP = Transport Service Access Point

= Transportschicht-Dienstzugangspunkt

Adressierung in der ISO Transportschicht

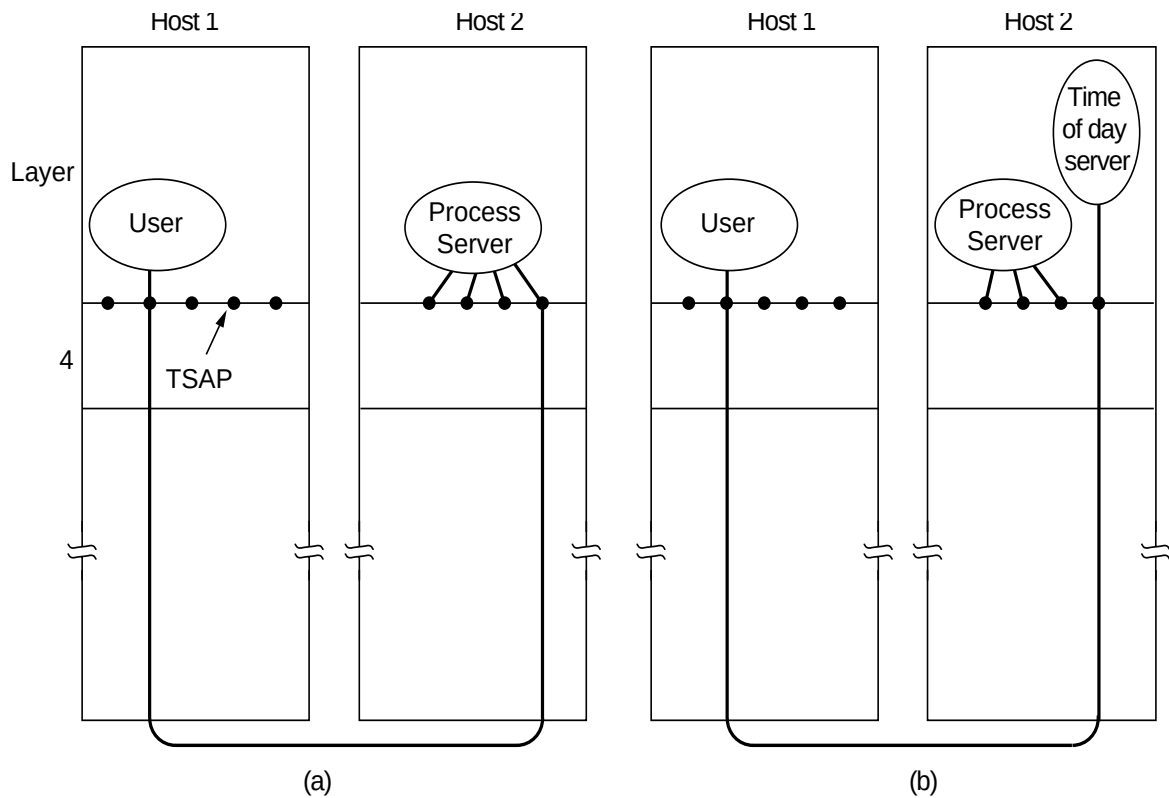
Beispiel:



- Der Anwendungsprozeß spricht die Transportschicht über einen sog. Transportschicht-Dienstzugangspunkt (TSAP) an. (Im Beispiel ist dies TSAP 6)
- Die Transportschicht wiederum spricht die Vermittlungsschicht über einen Vermittlungsschicht-Dienstzugangspunkt (NSAP) an (Network Layer Service Access Point)
- Die TSAPs werden bei TCP durch „Ports“ realisiert

Indirekte Adressierung in der Transportschicht nach ISO

Beispiel: Server für die Uhrzeit:



Voraussetzung:

Der TSAP des Prozeß-Servers ist dem Client bekannt, der TSAP des Uhrzeit-Servers hingegen nicht. Diesen weiß aber wiederum der Prozeß-Server

a) Client schickt Connect-TPDU an Prozeß-Server, der als Stellvertreter für den Uhrzeit-Server fungiert

b) Prozeß-Server reicht TCP-Verbindung an Uhrzeit-Server weiter

Multiplexen von ISO-Transportschicht-Verbindungen

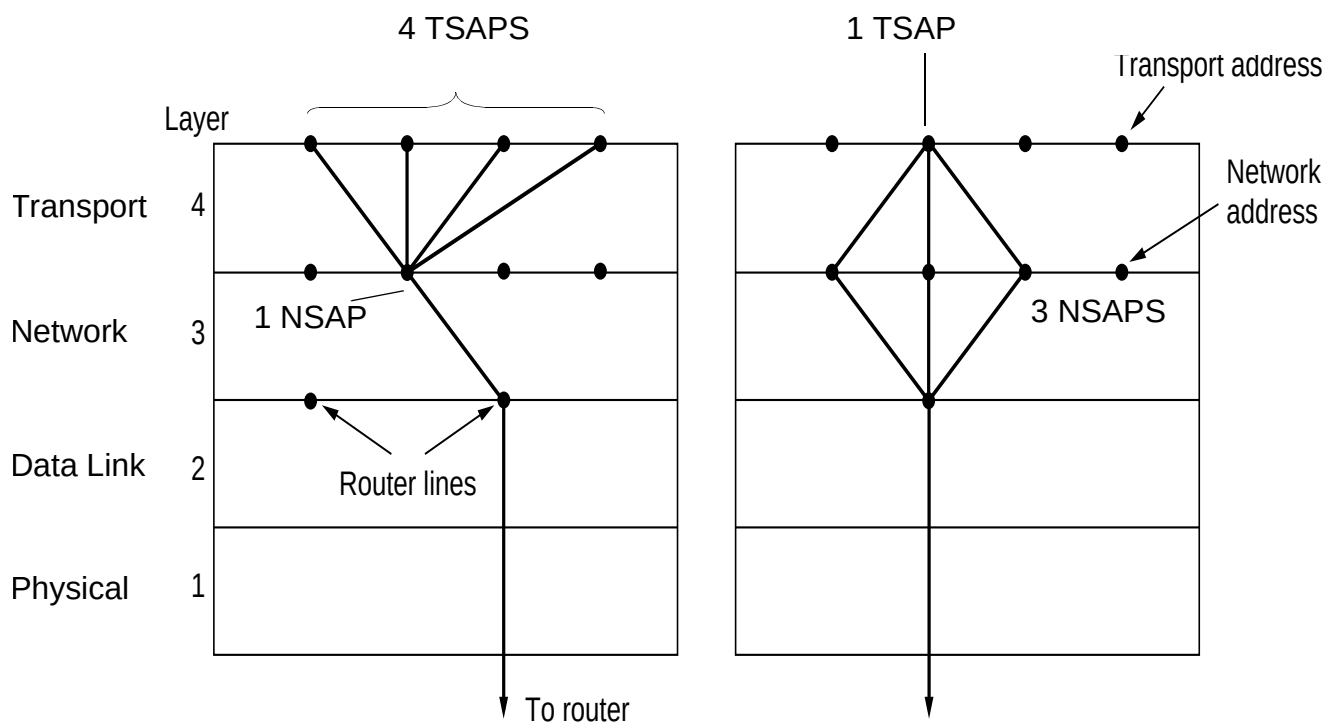
a) Upward-Multiplexen:

Aus Kostengründen werden mehrere Transportschicht-Verbindungen über eine Sicherungsschicht-Verbindung übertragen

Voraussetzung: Sicherungsschicht bietet verbindungsorientierten Dienst an wie beispielsweise ATM)

b) Downward-Multiplexen:

Aus Performance-Gründen wird eine Transportschicht-Verbindung über mehrere Sicherungsschicht-Verbindungen übertragen



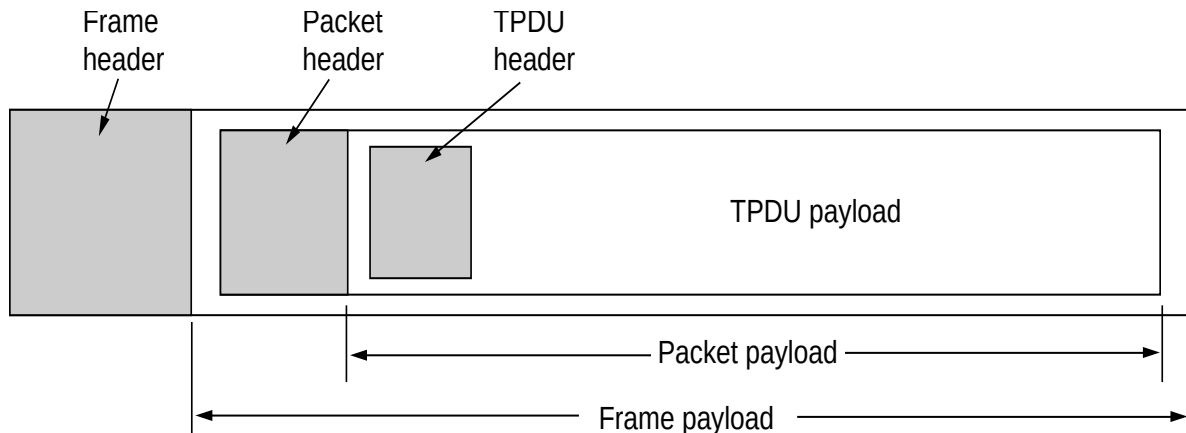
(a)

Beispiel:
4 TSAPS auf
ein NSAP

(b)

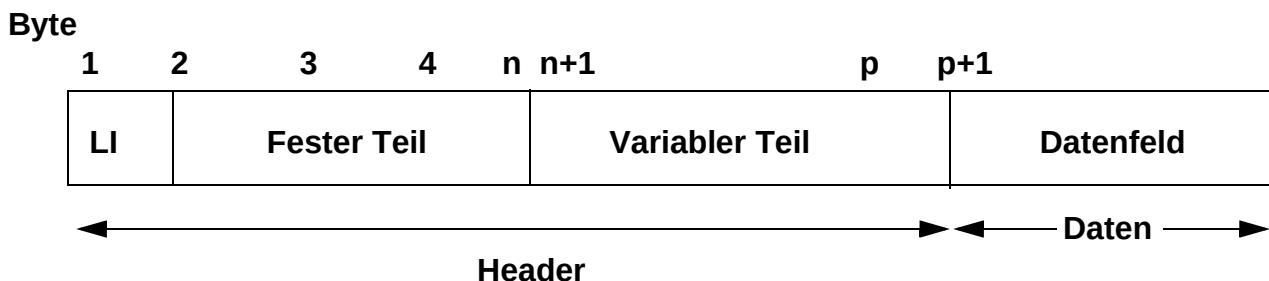
Beispiel:
1 TSAP auf
3 NSAPS

Verschachtelung von ISO-TPDUs, -Paketen und -Rahmen



Allgemeine Struktur von TPDUs nach ISO

TPDU = Transport Protocol Data Unit
= Protokolldateneinheit der Transportschicht



1.) Header bestehend aus

- Längenindikator LI (1 Byte)

Länge des Headers, max. 254 Bytes (1111 1110) bei ISO

- Fester Teil

- Enthält häufig auftretende Parameter inklusive des Typs der TPDU (TCP unterstützt keine Typen)

- **Variabler Teil**

- Enthält weniger häufig benutzte Parameter (abhängig vom TPDU-Typ)

2.) Datenfeld

- Enthält Daten der nächsthöheren Schicht (Sitzungsschicht) bei TCP bis ca. 64 KB
-
- Der ISO-Paketaufbau war ein Vorbild für das Format von TCP im Internet; letzteres wurde aber nicht 1:1 umgesetzt

Operationen des ISO-Transportsdienstes

Primitive	TPDU sent	Meaning
LISTEN	(none)	Block until some process tries to connect
CONNECT	CONNECTION REQ.	Actively attempt to establish a connection
SEND	DATA	Send information
RECEIVE	(none)	Block until a DATA TPDU arrives
DISCONNECT	DISCONNECTION REQ.	This side wants to release the connection

Berkeley-Sockets

o Sind die Programmierschnittstelle zur Kommunikation mit dem Transportsdienst TCP im Internet

o Verwenden IP- und Port-Adresse zur netzweit eindeutigen Adressierung von Server-Diensten

Primitive	Meaning
SOCKET	Create a new communication end point
BIND	Attach a local address to a socket
LISTEN	Announce willingness to accept connections; give queue size
ACCEPT	Block the caller until a connection attempt arrives
CONNECT	Actively attempt to establish a connection
SEND	Send some data over the connection
RECEIVE	Receive some data from the connection
CLOSE	Release the connection

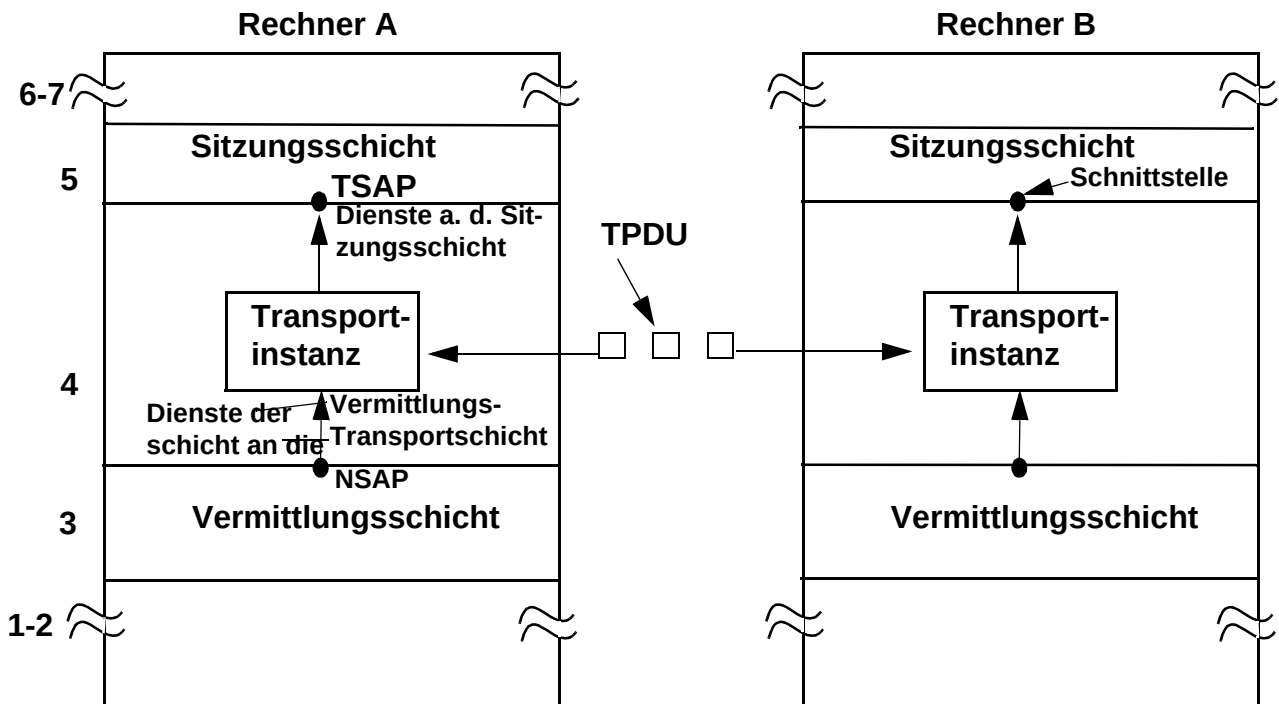
Kommunikation über Berkeley Sockets

Server	Client
SOCKET	
BIND	
LISTEN	
ACCEPT	
	SOCKET
	CONNECT
SEND, RECEIVE	SEND, RECEIVE
	CLOSE

- **Berkeley Sockets sind in praktischen allen Platten-Betriebssystemen als Standard-Schnittstelle zur Transportschicht vorhanden**

Zusammenfassung der ISO-Dienste und Protokolle

Beziehung zwischen dem Transportprotokoll und benachbarten Diensten der Sitzungs- und Vermittlungsschicht



TSAP = Transport Service Access Point

NSAP = Network Service Access Point

TPDU = Transport Protocol Data Unit

- Das Protokoll zwischen zwei Transportinstanzen verschiedener Rechner wird über TPDU abgewickelt (TPDU = Paket der Transportschicht)

Beispiel: TCP-Transportdienst

- TCP ist das wichtigste Transportprotokoll im Internet
- Neben TCP gibt es noch UDP als Transportprotokoll, das allerdings nur Datagramm-basiert ist

TCP- Eigenschaften

- verbindungsorientierter Dienst
- korrigiert Paketfehler, -verlust, -duplizierung und Fehler in der Paketreihenfolge
- ist sehr effizient hinsichtlich des Datenaustauschs

=> die TCP-Transportschicht verbessert die Übertragungsqualität (Quality of Service) der Vermittlungsschicht

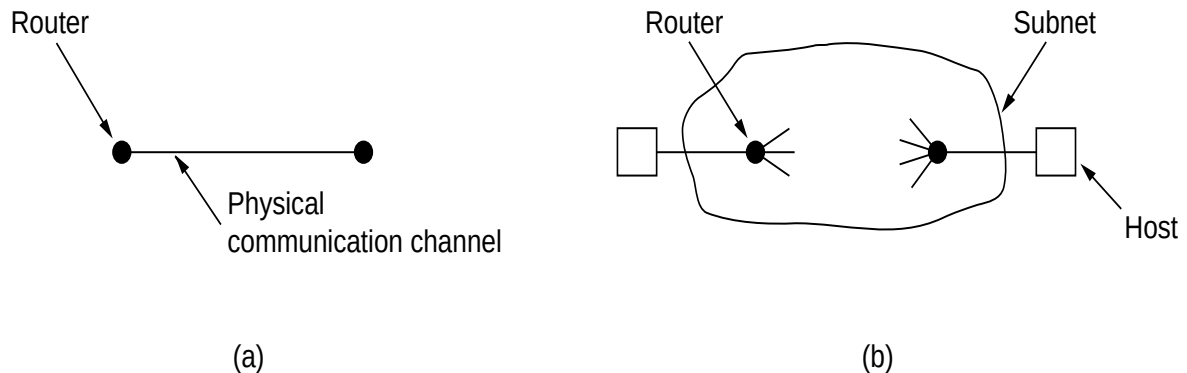
Hinweis:

- Die Sicherungsschicht war bereits verbindungsorientiert (zumindest bei HDLC)
- sie hat bereits Paketfehler, -verlust, -duplizierung und Fehler in der Paketreihenfolge korrigiert (zumindest bei HDLC)

Was ist der Unterschied zwischen der Sicherungs- und der Transportschicht?

- die Transportschicht arbeitet direkt von Benutzerprozeß zu Benutzerprozeß. Diese werden auf nicht-benachbarten Rechnern ausgeführt

- die Sicherungsschicht arbeitet nur zwischen benachbarten Rechnern. Prozesse gibt es für sie nicht



a) Arbeitsumgebung der Sicherungsschicht ist ein Kabel

b) Arbeitsumgebung der Transportschicht ist ein ganzes Teilnetz

Optimierung der Datenübertragung bei TCP

- TCP ist sehr effizient hinsichtlich der Übertragung von Daten
- Die Gründe für die Effizienz liegen in:
 - Der Verwendung eines adaptiven Schiebefensters zur Überlaststeuerung
 - Dadurch wird die optimale Grösse eines Pakets ermittelt
 - Der Verwendung eines adaptiven Time-Outs
 - Dadurch werden unnötige Wartezeiten bei Paketverlust vermieden
 - Der Möglichkeit der hochpriorigen Übertragung
 - Erlaubt falsche Übertragungen vorzeitig abzugeben

Überlastungssteuerung bei TCP

o Um Überlastungen zu vermeiden, versucht jede einzelne TCP-Verbindung nur soviel Daten zu senden, wie der jeweilige Empfänger verkraften kann.

o Das besondere Problem dabei ist, daß sich die verkraftbare Menge beim Empfänger ständig ändern kann. Deswegen wird vom Sender die Größe seines Schiebefensters adaptiv verändert:

1. Beim Aufbau der Verbindung wird die Anfangsgröße des Schiebefensters, das der Sender benutzt, vom Sender festgelegt

2. Dann wird ein Paket von der Größe des Schiebefensters gesendet

3. für jedes erhaltene Acknowledge verdoppelt der Sender seine Fenstergröße und schickt ein doppelt so großes Paket ab

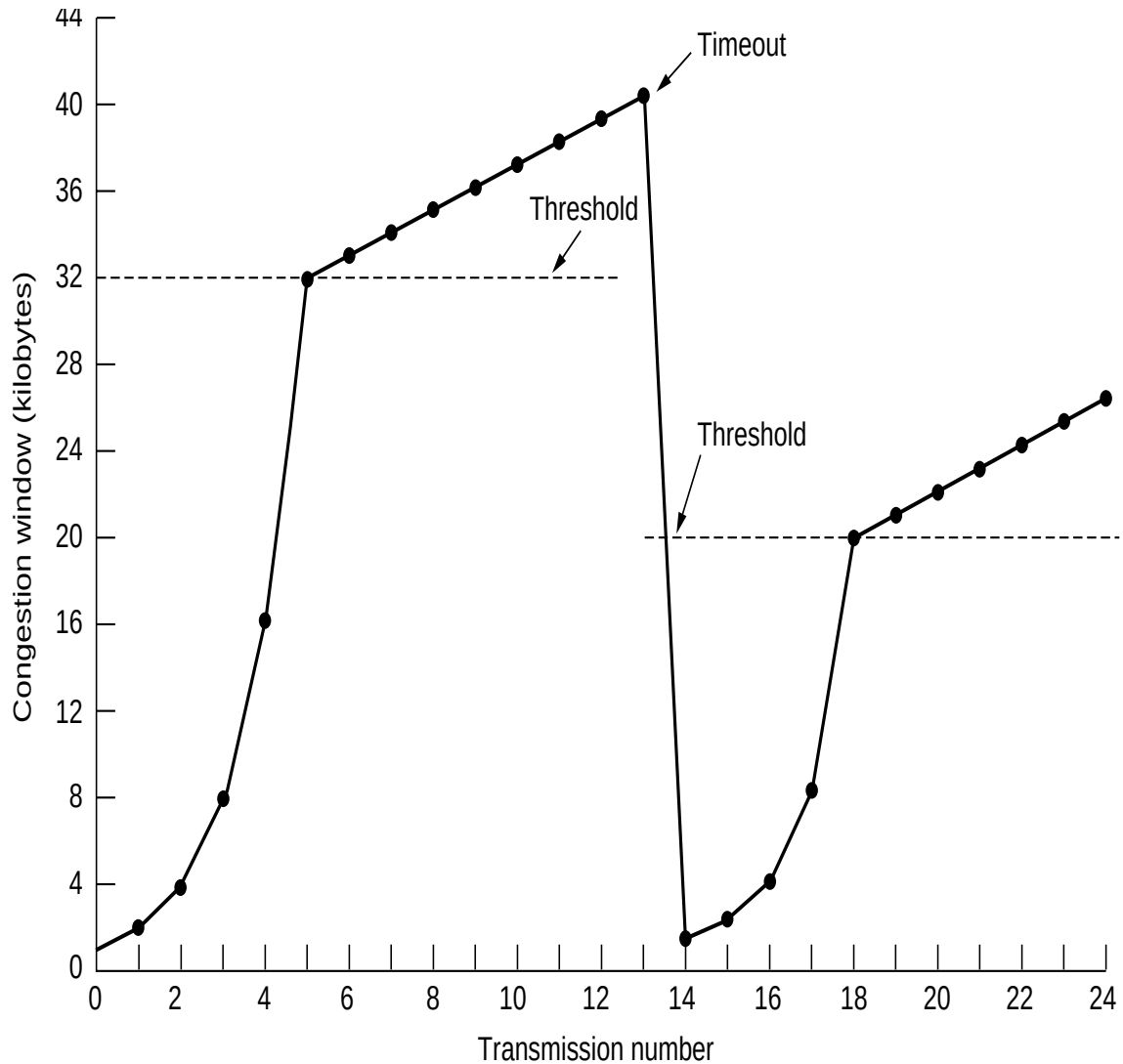
4. sobald kein Acknowledge mehr kommt (time out!), halbiert der Sender seine Fenstergröße und vermerkt den Wert als sog. Schwellwert.

5. Danach beginnt der Sender wieder bei der ursprünglichen Fenstergröße und verdoppelt für jedes Acknowledge Fenstergröße und Paketgröße solange, bis er den Schwellwert erreicht

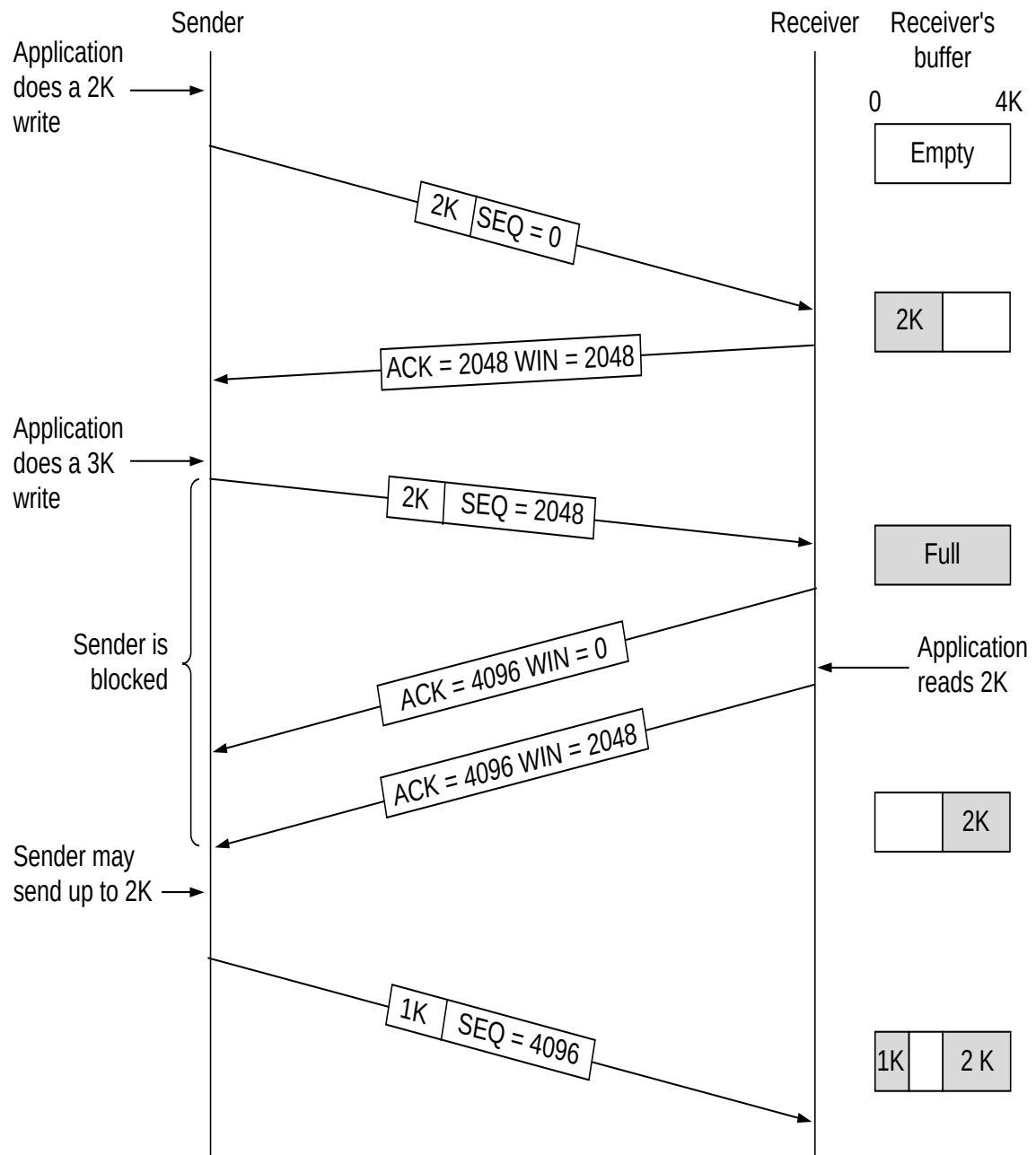
6. ab dem Schwellwert wachsen Fenster- und Paketgröße nur noch linear an, solange bis wieder kein Acknowledge mehr kommt

7. danach wird Schritt 4 bis 6 wiederholt

o Beispiel: Anfangsgröße von Fenster und Paket: 1K, Anfangsschwellwert: 32 K



Adaptives Schiebefenster bei TCP



Adaptiver Time Out bei TCP

o Es wird beim Abschicken jedes Paket gemessen, wie lange es bis zum zugehörigen Acknowledge dauert (=gemessene Zeit M)

o daraus wird die Entfernung bis zum Ziel und wieder zurück bestimmt (=Round Trip Time, RTT):

$$RTT_{\text{neu}} := \alpha RTT_{\text{alt}} + (1-\alpha)M, \alpha \text{ ist ein Glättungsfaktor}$$

o zusätzlich wird der geglättete Mittelwert der Abweichung verwendet

$$D_{\text{neu}} := \beta D_{\text{alt}} + (1-\beta) \text{abs}(M - RTT_{\text{neu}})$$

o Daraus ergibt sich der aktuelle time out-Wert zu:

$$\text{Timeout} = RTT_{\text{neu}} + 4D_{\text{neu}}$$

o der Faktor $4D_{\text{neu}}$ stellt eine Sicherheitsreserve dar

o Stellt sich der berechnete Wert als niedrig heraus (=kein Acknowledge), wird der zuletzt verwendete Wert verdoppelt

Push-Paketen und Urgent Daten (Out-of-Band-Daten)

- o Push-Pakete sind spezielle, mit dem PSH-Flag gekennzeichnete Pakete**
- o Das PSH-Flag bewirkt, daß ein solcherart gekennzeichnetes Paket nicht aus Effizienzgründen zwischengespeichert werden darf, bis es mit anderen kleineren Paketen derselben Verbindung vereinigt werden kann. Vielmehr muß es sofort abgeschickt werden. Sind bereits mehrere, kleinere TCP-Pakete derselben Verbindung zwischengespeichert, werden diese mit dem Push-Paket vereinigt und das Gesamtpaket wird sofort abgeschickt.**
- o Out-of-Band-Daten sind hochprioräre Daten im Datenteil von TCP-Paketen. Pakete, die solche Daten enthalten, werden wie Push-Pakete behandelt. Zusätzlich wird beim Empfänger ein Interrupt (=Signal) ausgelöst, sobald ein solches Paket eintrifft. Die Daten werden daraufhin sofort bearbeitet.**

Zusammenfassende Liste der Dienste der TCP-Transportschicht

- Verbindung Prozess-zu-Prozess hinsichtlich
 - 1.) Verbindungsaufbau
 - 2.) Datenübertragung
 - 3.) Verbindungsabbau
- Zusammenfassen von mehreren kurzen TCP-Paketen derselben Verbindung zu einem langen Paket (=effizienter)
- Zerteilen langer Nachrichten der Sitzungsschicht in 64K-Pakete
- Fehlerbehebung u. a. durch
 - 1.) Verbindungsneuzuordnung (reset einer Verbindung)
 - 2.) Resynchronisation (Wiederaufbau einer unterbrochenen Verbindung)
- Flusskontrolle, Sicherstellung der Reihenfolge (Ende-zu-Ende)
- Bestätigte Datenübertragung (Ende-zu-Ende)