Inhalt (2)

Inhalt (3)

Rechnernetze

SS 2002

Prof. Dr. W. Effelsberg

Lehrstuhl für
Praktische Informatik IV

Universität Mannheim

Inhalt

Literatur (2)

10. **Stevens, W. Richard:** TCP/IP Illustrated. Volume 1: The Protocols. Addison Wesley, 1994.
11. **Tanenbaum, A.S.**: Computer Networks. 3rd edition, Prentice Hall, 1996
12. **Zitterbart, M.**: Hochleistungskommunikation, Band 1: Technologie und Netze. Oldenbourg, München/Wien, 1995
13. **Zitterbart, M.**: Transportdienste und Transportprotokolle (Hochleistungskommunikation, Band 2), Oldenbourg, München/Wien, 1996
14. **Zitterbart, M., Schmidt, C.**: Internetworking - Brücken, Router&Co.; TAT-Band 8, International Thomson Publishing, 1995

1.1 Definition eines Rechnernetzes, Abgrenzung

Definition

Ein Rechnernetz dient zur Kopplung unabhängiger Rechner zum Zwecke des Datenaustauschs.

Abgrenzung gegenüber

- Bus, Kanal
- Interkonnektionsnetz eines Parallelrechners (Mehrprozessor-System vs. Verteiltes System)
- Terminalnetz

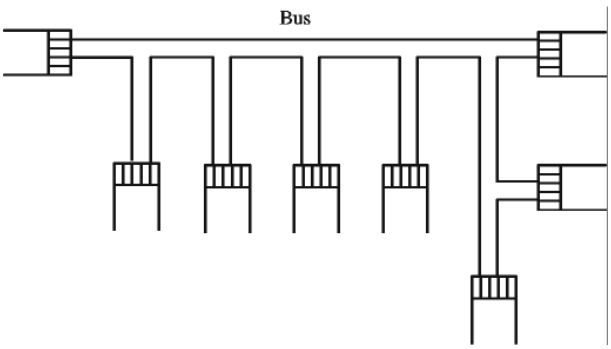
Inhalt (4)

Literatur (1)

1. **Bocker, P.**: ISDN – The Integrated Services Digital Network, 3. Auflage, Springer-Verlag, 1996
2. **Comer**: Internetworking with TCP/IP, Vol.1; Prentice-Hall, 1995
3. **De Prycker, Martin**: Asynchronous Transfer Mo-de. 3rd edition, Prentice Hall Europe, 1995
4. **Halsall, Fred**: Data Communications, Computer Networks and Open Systems. 4th edition, Addison-Wesley, 1995
5. **Huitema, Ch.**: Routing in the Internet, Prentice Hall, Englewood Cliffs, 1995
6. **Huitema, Ch.**: IPv6, Prentice Hall, Englewood Cliffs, 1995
7. **Kuo, Frank, Effelsberg, Wolfgang und Garcia-Luna-Aceves, J.J.**: Multimedia Communications - Protocols and Applciations. Prentice Hall, Upper Saddle River, 1998
8. **Partridge, C.**: Gigabit Networking. Addison Wesley, 1994
9. **Peterson, Larry L. and Davie, Bruce S.**: Computernetze – ein modernes Lehrbuch. dpunkt-Verlag, Heidelberg, 2000

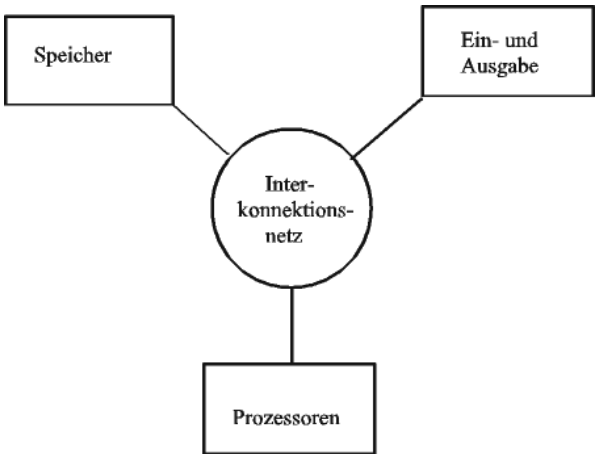
Bus und Interkonnektionsnetz (1)

Bus in einem Rechner



Bus und Interkonnektionsnetz (2)

Interkonnektionsnetz in einem Parallelrechner

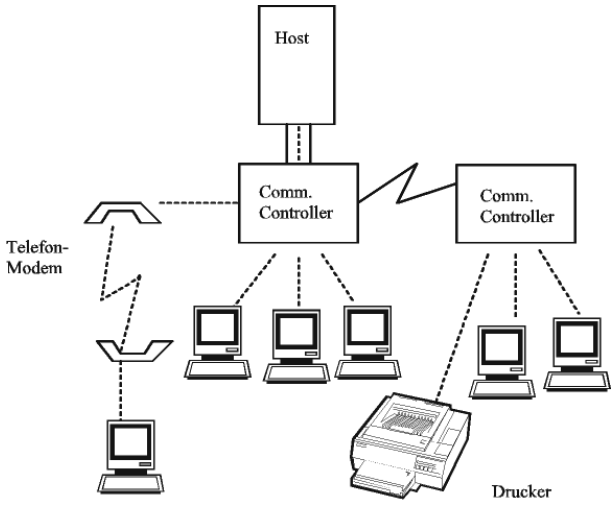


n Prozessoren

Ziele eines Rechnernetzes

- **Datenverbund**
  Zugriff auf entfernte Daten, Datenaustausch
- **Funktionsverbund**
  Zugriff auf Spezialrechner, z. B. Server
- **Lastverbund**
  gleichmäßige Lastverteilung
- **Verfügbarkeitsverbund**
  Fehlertoleranz, schrittweises Wachstum

Terminalnetz



International Telecommunications Union (ITU)

Ehemals: Comité Consultatif International de Télégraphie et Téléphonique (CCITT)

- Internationale Vereinigung der Postgesellschaften (Telecoms)
- Vollversammlung alle 4 Jahre (..., 1980, 1984, 1988, 1992, 1996 ...)
- Verabschiedung von Empfehlungen (Recommendations)
- Verwendung verschiedener Farben für die Empfehlungen jeder Vollversammlung
  - gelb (Yellow Books) (1980)
  - rot (Red Books) (1984)
  - blau (Blue Books) (1988)
- ITU ist eine UN-Organisation
  - ITU-R (ITU Radiocommunication Standardization Sector)
  - ITU-T (ITU Telecommunication Standardization Sector)

CEN / CENELEC / ETSI

- Europäische Normungsinstitute
- Europaweite Harmonisierung der nationalen Normen
- Functional Standards, Profiles

1.2 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

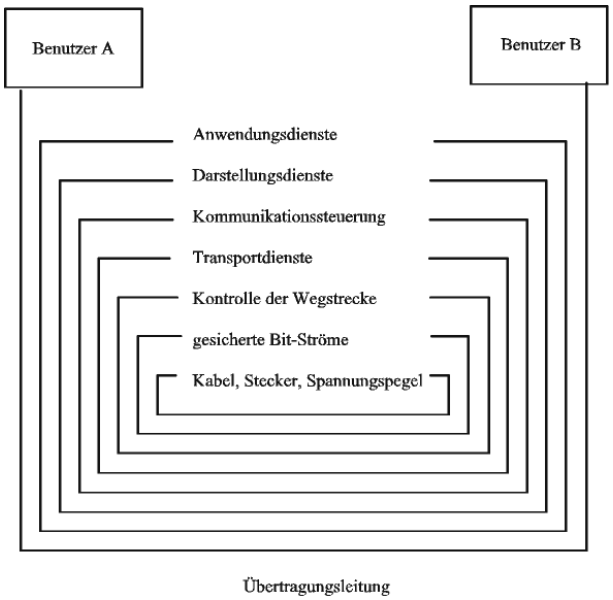
- International Organization for Standardization (ISO)
- International Telecommunications Union (ITU) Ehemals: Comite Consultatif International de Télégraphie et Téléphonique (CCITT)
- CEN/ CENELEC/ ETSI (europäisch)
- National Institute of Standards and Technology (NIST)

International Standards Organization (ISO)

- Normung auf internationaler Ebene
- Mitglieder: Nationale Normungsgremien (DIN, ANSI, AFNOR,...)
  - ISO TC 97: Information Processing Systems
  - DIN: Normungsausschuss Informationsverarbeitung (NI)
  - TC 97/SC 6: Data Communications
  - TC 97/SC 18: Text and Office Communications
  - TC 97/SC 21: Open Systems Interconnection
- Stufen einer Norm
  - a) Working Draft (WD)
  - b) Draft Proposal (DP)
  - c) Draft International Standard (DIS)
  - d) International Standard (IS)
- Normen besitzen keine Rechtsverbindlichkeit

1.3 Protokollhierarchien

Strukurierung des Problems



1.4 Das ISO-Referenzmodell für Offene Systeme

- OSI = OPEN SYSTEMS INTERCONNECTION
- Kurz: ISO/OSI-Referenzmodell
- ISO International Standard 7498
- Ein Modell für geschichtete Kommunikationssysteme
- Einführung der Grundbegriffe (Terminologie)
- Schlägt sieben Schichten und ihre Funktionalität vor

Schicht	ISO
7	Anwendungsschicht
6	Darstellungsschicht
5	Kommunikationssteuerungs-schicht
4	Transportschicht
3	Vermittlungsschicht
2	Sicherungsschicht
1	Bitübertragungsschicht

Standards im Internet

- IAB (Internet Activity Board)
  - IETF (Internet Engineering Task Force)
  - IRTF (Internet Research Task Force)
- RFC (Request for Comment) erfüllt die Rolle eines Standards im Internet. Erstaunlicherweise ist die Entstehung von RFCs ein informeller Prozess!
- Arbeitsgruppen mit jeweils einem Leiter
  - Mitglied der Arbeitsgruppe kann jeder werden
  - Kommunikation erfolgt vorwiegend über E-Mail, gelegentliche IETF-Treffen
  - typische Arbeitszeit: 9-18 Monate
  - Ergebnis: Internet Draft
- Faustregel: Internet Draft → mindestens zwei unabhängige Implementierungen; Interoperabilitätstests; Stabilität über 4 Monate → Internet Standard (RFC)

Industriekonsortien

- Zusammenschluss vorwiegend industrieller Partner
- Ziel: rasche Realisierung kompatibler Produkte
  - deshalb: schnelle Entwicklung eines gemeinsamen defacto Standards
- Einbringung der Ergebnisse in die internationale Standardisierung
- Beispiele:
  - NFS (Network File System)
  - ATM (ATM-Forum)
  - WWW-Konsortium
- Problem: Vorgehensweise manchmal zu schnell, so dass interessante und richtungsweisende Forschungsergebnisse keinen Eingang in die de-facto-Standardisierung finden.

Prinzipien des ISO-Referenzmodells

- Offenes System
  - Rechnersystem (Hardware, Software, Peripherie, ...), das sich bei der Kommunikation an die OSI-Standards hält
- (N)-Schicht
  - wird aus sämtlichen Einheiten einer (N)-Hierarchiestufe in allen offenen Systemen gebildet
- (N)-Instanz
  - Implementierung einer (N)-Schicht in einem System
  - es kann verschiedene Typen von (N)-Instanzen geben, die die Protokolle für die (N)-Schicht in verschiedener Weise implementieren
- Partnerinstanzen, Peer-Entities
  - Instanzen derselben Schicht an verschiedenen Orten. Partnerinstanzen erfüllen die Funktionen einer Schicht durch Datenaustausch

Schicht (1)

- **Hauptaufgabe jeder Schicht ist es, der darüberliegenden Schicht *Dienste* anzubieten.**

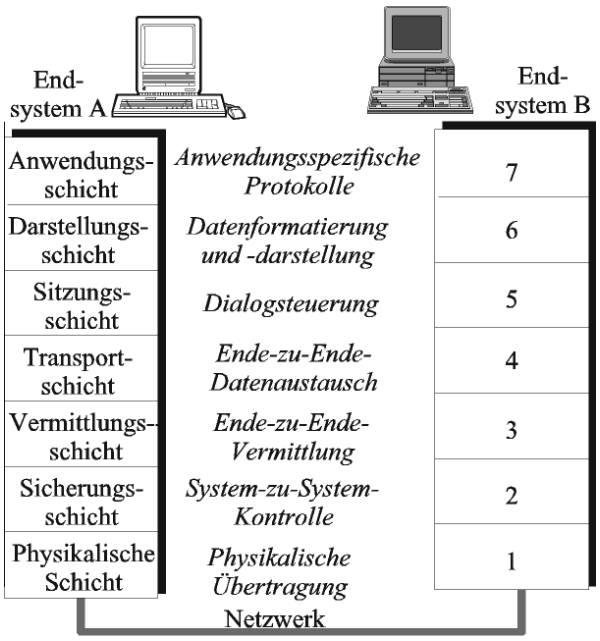
  Diese Dienste setzen sich zusammen aus:
  - Dienstleistungen, die innerhalb dieser Schicht implementiert werden, und
  - dem kumulativen Resultat der Dienstleistungen aller darunter liegenden Schichten.

- Schichten sind über so genannte **Dienstprimitive** miteinander verknüpft.

- Die direkte Kommunikation erfolgt mit den Schichten (N+1) und (N-1).

- Die indirekte Kommunikation mit den Partnerinstanzen (peer entities) erfolgt durch Abwicklung des Schicht-**Protokolls**.

Die sieben Schichten und ihre Funktionen

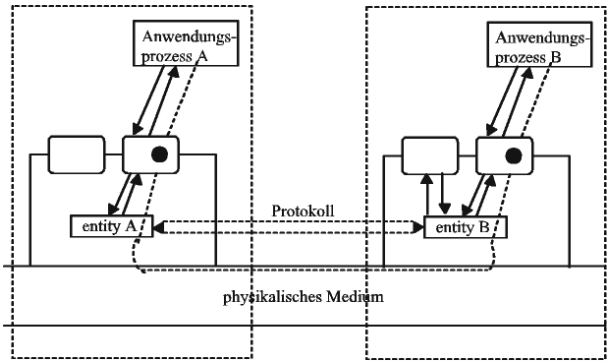


Die sieben Schichten des ISO-Referenzmodells

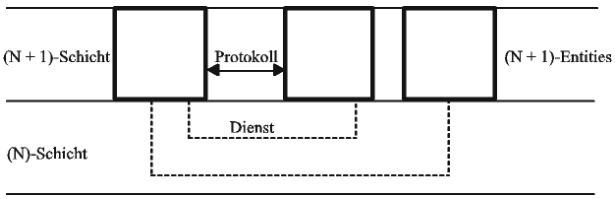
- Die Bitübertragungsschicht ermöglicht die transparente Übertragung eines Stroms binärer Information über eine Leitung.
- Die Sicherungsschicht soll insbesondere Übertragungsfehler entdecken und korrigieren.
- Die Vermittlungsschicht übermittelt Daten auf eine transparente Art und Weise. Dazu wird von der Transportschicht eine entsprechende Route ausgewählt.
- Die Transportschicht übermittelt die Daten von Endbenutzer zu Endbenutzer. Sie entlastet den Benutzer von den Details der Datenübertragung.
- Die Kommunikationssteuerungsschicht koordiniert die Zusammenarbeit zwischen den verschiedenen miteinander kommunizierenden Anwendungsprozessen.
- Die Darstellungsschicht transformiert die Darstellung der übermittelten Daten in eine Form, die von den kommunizierenden Anwendungsprozessen verstanden wird.
- Die Anwendungsschicht beschreibt die Natur der Datenübertragung, um den Anforderungen der Benutzer zu genügen. Die Anwendungsschicht ist die einzige Zugriffsmöglichkeit der Anwendungsprozesse zur Datenübertragung.

Protokoll

Unter einem **Protokoll** versteht man die Menge der Regeln für den Datenaustausch zwischen Instanzen derselben Schicht.

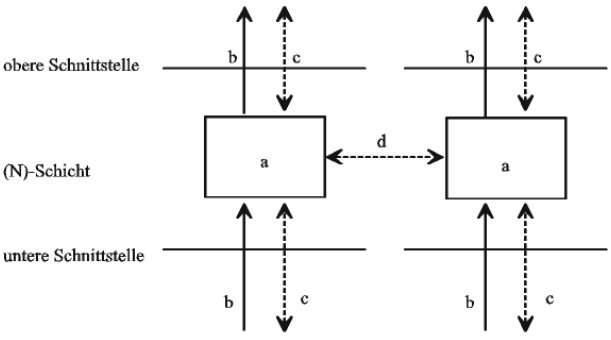


Dienst und Protokoll



Das **Protokoll** der Schicht (N+1) benutzt die **Dienste** der Schicht (N).

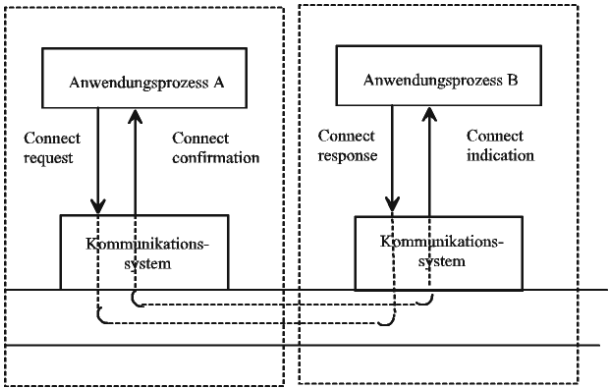
Schicht (2)



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

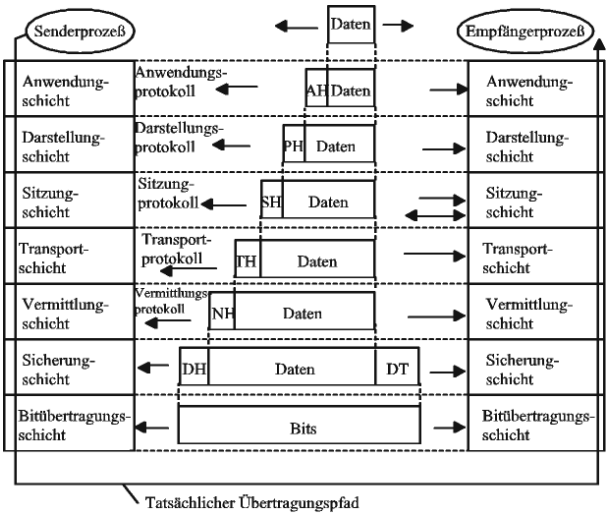
Ereignisse zwischen Anwendungsprozess und Kommunikationssystem

Beispiel: CONNECT (Verbindungsaufbau)

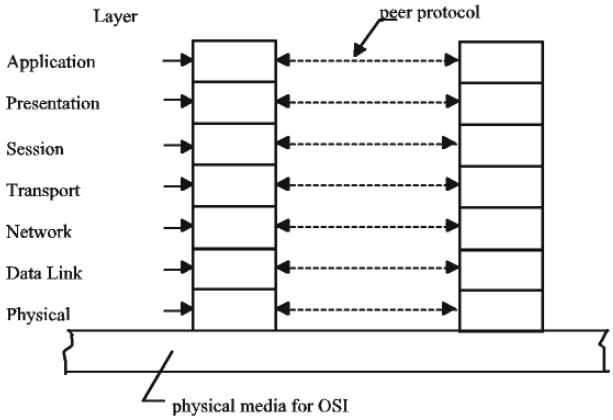


Beispiel

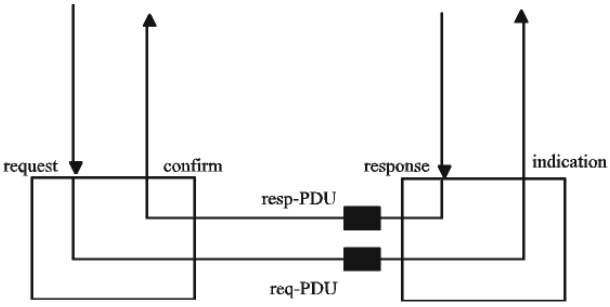
Ein Beispiel dafür, wie das Schichtenmodell sich auf die Nachrichtenformate auswirkt.



Referenzmodell und Partnerprotokolle



Dienstereignisse und Protokolldateneinheiten



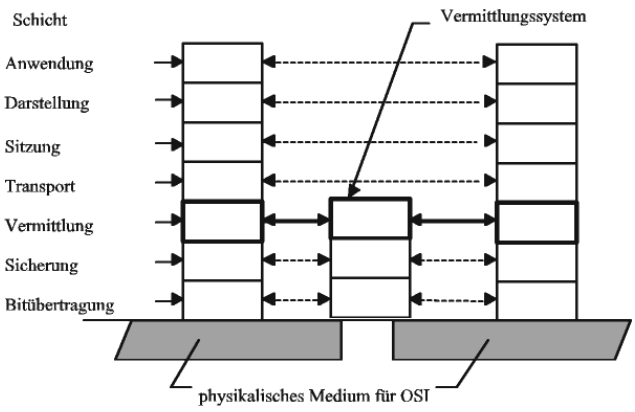
Arten von Dienstprimitiven

- Anforderung (Request)
  - Anfordern einer Dienstleistung durch den Benutzer
- Anzeige (Indication)
  - Dem Benutzer anzeigen, dass vom entfernten Benutzer ein Dienst angefordert wurde oder dass ein Ereignis in der Schicht selbst aufgetreten ist
- Antwort (Response)
  - Quittieren einer voran gegangenen Anzeige durch den Benutzer
- Bestätigung (Confirmation)
  - Quittieren einer voran gegangenen Anforderung durch den Dienstanbieter (die Schicht).

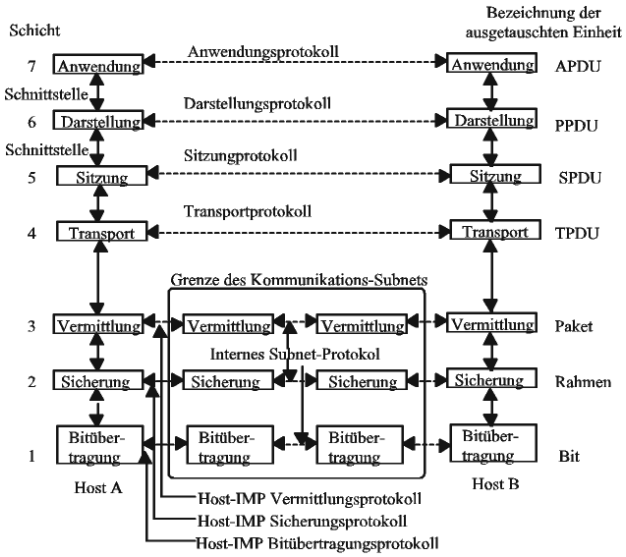
Schichtenmodelle verschiedener Netzarchitekturen

Layer	ISO	Internet	SNA
7	Application	SMTP, FTP, telnet, http	End user
6	Presentation		NAU services
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

Vermittlungssysteme



ISO–Referenzmodell mit Zwischenknoten



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

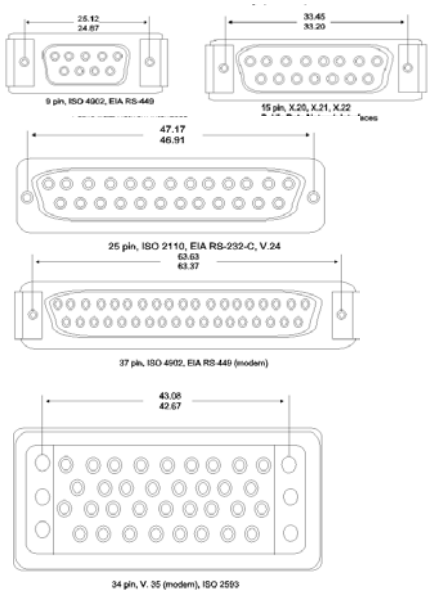
elektrisch: Spannungspegel auf Leitungen, etc. z. B. CCITT X.27/V.11: Electrical characteristics for balanced double-current interchange for general use with integrated circuit equipment in the field of data communication

funktional: Klassifikation von Leitungsfunktionen (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

2.2 Mechanische, elektrische und funktionale Spezifikation

Mechanische Eigenschaften



2. Bitübertragungsschicht (Physical Layer)

2.1 Definition

2.2. Mechanische, elektrische und funktionale Spezifikation

2.3. Übertragungstechniken, Modulation, Multiplexing

2.4. Physikalische Medien

2.5. Beispiele: V.24, ADSL

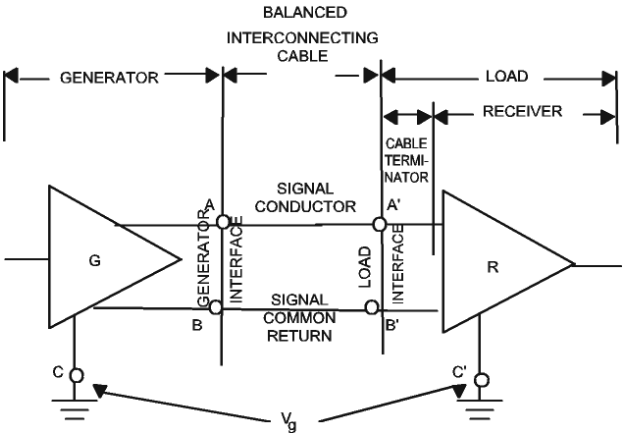
2.1 Bitübertragungsschicht, Definition

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-Entities über physikalische Verbindungen. Eine physikalische Verbindung kann die Übertragung eines Bitstroms im Duplex-Mode oder im Halbduplex-Mode erlauben.

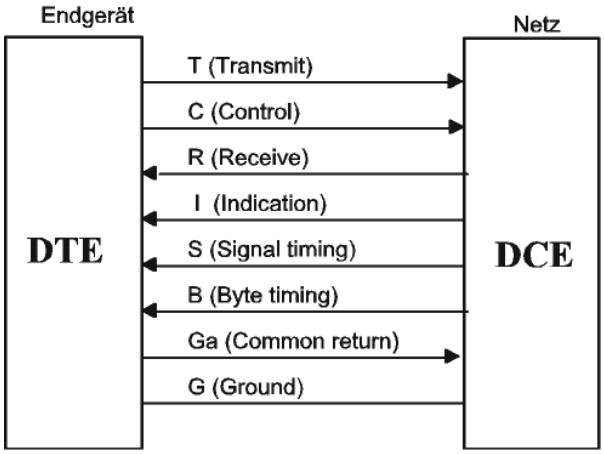
CCITT V.11/X.27 (EIA RS-422-A)



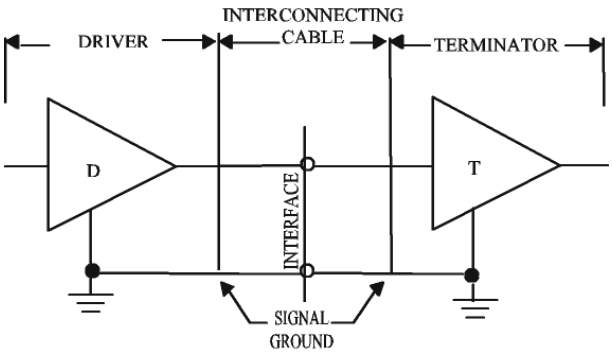
- Für IC-Bauelemente (integrierte Schaltkreise)
- Zwei Leiter pro Stromkreis
- Bitrate bis zu 10 Mbit/s
- Entfernung bis zu 1000 m bei 100 kbit/s oder bis zu 10 m bei 10 Mbit/s
- Minimales "Übersprechen"

Funktionale und prozedurale Eigenschaften

Signalleitungen bei X.21

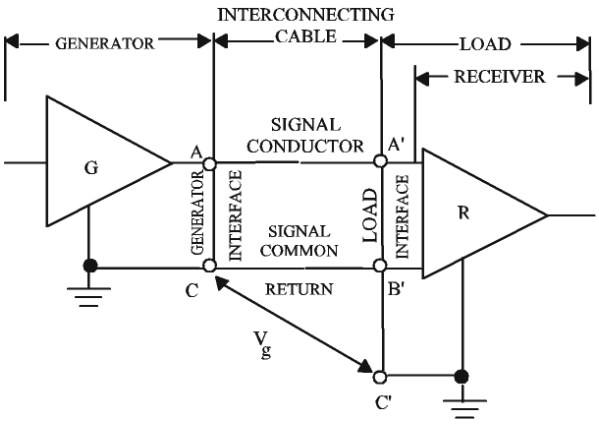


Elektrische Eigenschaften CCITT V.28 (EIA RS-232-C)



- Für **diskrete** elektronische Bauelemente
- Ein Leiter pro Stromkreis, mit einer gemeinsamen Erdung für beide Richtungen
- Bitrate begrenzt auf 20 kbit/s
- Entfernung begrenzt auf 15 m
- Erzeugt erhebliches "Übersprechen"

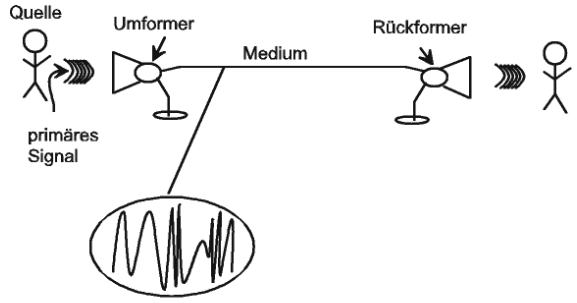
CCITT V.10/X.26 (EIA RS-423-A)



- Für IC-Bauelemente (integrierte Schaltkreise)
- Ein Leiter pro Stromkreis, mit je einer Erdungsleitung pro Richtung
- Bitrate bis zu 300 kbit/s
- Entfernung bis zu 1000 m bei 3 kbit/s oder bis zu 10 m bei 300 kbit/s
- Reduziertes "Übersprechen"

2.3 Übertragungstechniken, Modulation, Multiplexing

Signalübertragung



Beispiel: Telefon, analoge Signale

Das Primärsignal (hier akustisch) wird durch Umformer in ein elektrisches (hier analoges) Signal umgewandelt und durch Rückformer zurück gewandelt. 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.

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. zeitkontinuierliche, 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

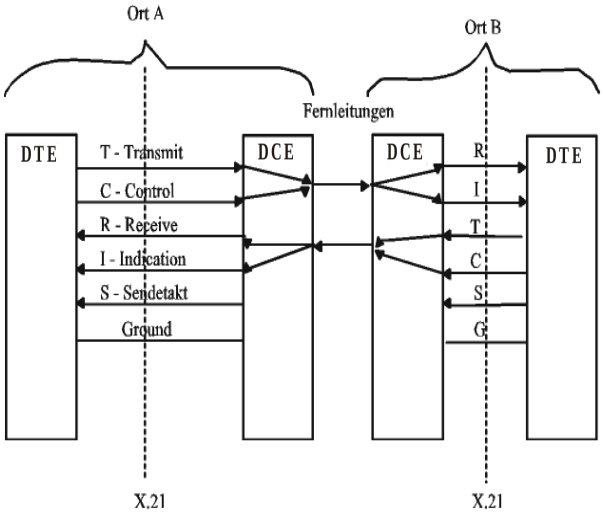
Sind alle Signalwerte im Wertebereich zulässig?
ja: wertkontinuierlich
nein: wertdiskret

Funktionale/prozedurale Spezifikation in X.21

(erläutert in Analogie zum Telefon)

Schritt	C	I	Ereignis analog zum Telefon	DTE sendet auf T	DCE sendet auf R
0	Aus	Aus	idle (Ruhezustand)	T=1	R=1
1	Ein	Aus	DTE nimmt Hörer ab	T=0	
2	Ein	Aus	DCE sendet Wähl-ton		R="+++..+"
3	Ein	Aus	DTE wählt Telefon-nummer	T=Adresse	
4	Ein	Aus	Entferntes Telefon klingelt		R=Dienst
5	Ein	Ein	Entferntes Telefon abgehoben		R=1
6	Ein	Ein	Gespräch (Daten-austausch)	T=Daten	R=Daten
7	Aus	Ein	DTE verabschiedet sich	T=0	
8	Aus	Aus	DCE verabschiedet sich		R=0
9	Aus	Aus	DCE legt auf		R=1
10	Aus	Aus	DTE legt auf; -> idle	T=1	

Lokale Schnittstelle vs. Fernleitung



Die Anzahl der Leitungen auf der Fernstrecke muss nicht gleich der Anzahl der Leitungen an der Endgeräte-Schnittstelle sein!

Moderne Basisbandverfahren

Moderne digitale Übertragungstechnik verwendet Basisbandverfahren bis zu sehr hohen Bitraten (PCM-Technik, lokale Netze, ISDN usw.). Dabei sind erwünscht bzw. erforderlich:

- kein Gleichstromanteil
- 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

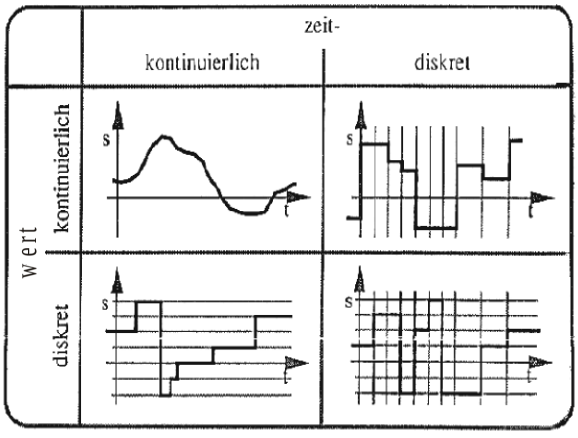
wird als **Signal- oder Leitungscodierung** bezeichnet.

Die sich ergebenden zeit- und wertdiskreten Signalverläufe heißen **Leitungscodes** oder **Übertragungscodes**.

Wichtige digitale Leitungscodes (1)

- **Non-return to zero-level (NRZ-L)**
  1 = hoher Pegel
  0 = niedriger Pegel
- non-return to zero-mark (NRZ-M)
  1 = Transition am Intervallanfang
  0 = keine Transition am Intervallanfang
- non-return to zero-space (NRZ-S)
  1 = keine Transition am Intervallanfang
  0 = Transition am Intervallanfang
- return to zero (RZ)
  1 = Rechteckimpuls am Intervallanfang
  0 = kein Rechteckimpuls am Intervallanfang
- **Manchester-Code (biphase level)**
  1 = Transition von hoch nach niedrig in der Intervallmitte
  0 = Transition von niedrig nach hoch in der Intervallmitte
- biphase-mark
  Immer eine Transition am Intervallanfang
  1 = Transition in der Intervallmitte
  0 = keine Transition in der Intervallmitte

Signalklassen

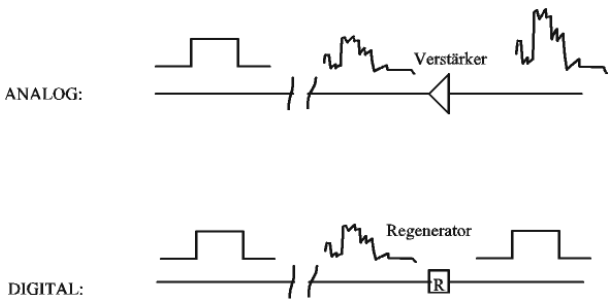


- wert- und zeitkontinuierlich: analoges Telefon
- wertkontinuierlich, zeitdiskret: Prozesssteuerung mit periodischen Messzeitpunkten
- wertdiskret, zeitkontinuierlich: digitale Übertragung mit beliebigen Signalwechseln
- wert- und zeitdiskret: digitale Übertragung mit isochronem Taktmuster

Grundlegende Übertragungstechniken

- Digitale Eingabe, digitale Übertragung:
  Digitale Leitungscodierung
- Digitale oder analoge Eingabe, analoge Übertragung:
  Modulationstechniken
- Analoge Eingabe, digitale Übertragung:
  Digitalisierung (Abtastung)

Analoge und digitale Übertragung



Differenzielle Leitungscodes

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 (Übergang in den entgegengesetzten Signalwert) zur Darstellung des Datenwerts “1“.

NRZ-S: Signalwechsel zur Darstellung des Datenwerts “0“.

Vorteile gegenüber NRZ-L: Unter Einfluss von Störungen (Rauschen) sind Signal**wechsel** leichter zu detektieren als Signal**pegel**, die mit einem Schwellwert verglichen werden müssen.

Nachteile aller NRZ-Codes: Gleichstromkomponente und fehlender Takt zwischen Sender und Empfänger (z. B. bei langen „0“-Folgen bei NRZ-L und NRZ-M)

Biphase-Codes

Alle Biphase-Leitungscodierungen haben mindestens einen Signalwechsel pro Bitintervall und höchstens zwei Signalwechsel pro Bitintervall.

Vorteile

- Leichte Synchronisierung, da stets mindestens ein Signalwechsel pro Bitintervall (es gibt eine "Impulsflanke" zum Triggern des Empfängers)
- Keine Gleichstromkomponente
- Fehlererkennung auf Signalebene möglich: Fehlen eines erwarteten Übergangs leicht erkennbar

Nachteil

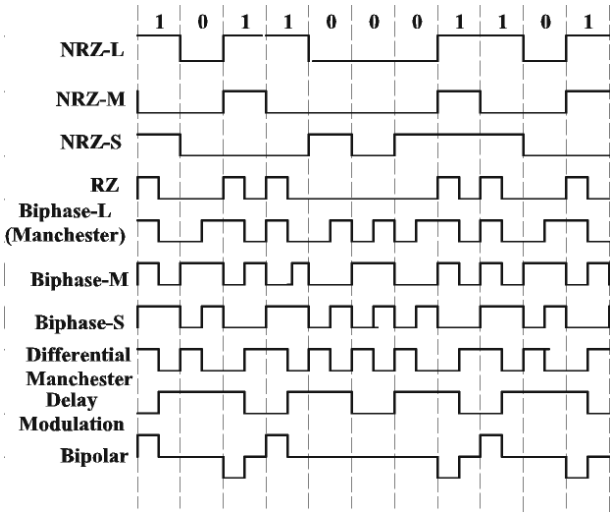
- Doppelt so viele Rechteckimpulse pro Sekunde für dieselbe Bitrate!

Wichtige digitale Leitungscodes (2)

- biphase-space
  Immer eine Transition am Intervallanfang
  1 = keine Transition in der Intervallmitte
  0 = Transition in der Intervallmitte
- **Differential Manchester-Code**
  Immer eine Transition in der Intervallmitte
  1 = keine Transition am Intervallanfang
  0 = Transition am Intervallanfang
- delay modulation (Miller)
  Transition am Intervallende, wenn eine 0 folgt
  1 = Transition in der Intervallmitte
  0 = keine Transition, wenn eine 1 folgt
- bipolar
  1 = Rechteckimpuls in der ersten Intervallhälfte, Polarität alternierend
  0 = kein Rechteckimpuls

Leitungscodes

Beispiel



Digitale/analoge Daten, analoge Signale

Modulation: verschlüsselt Quelldaten auf ein **analoges** Trägersignal

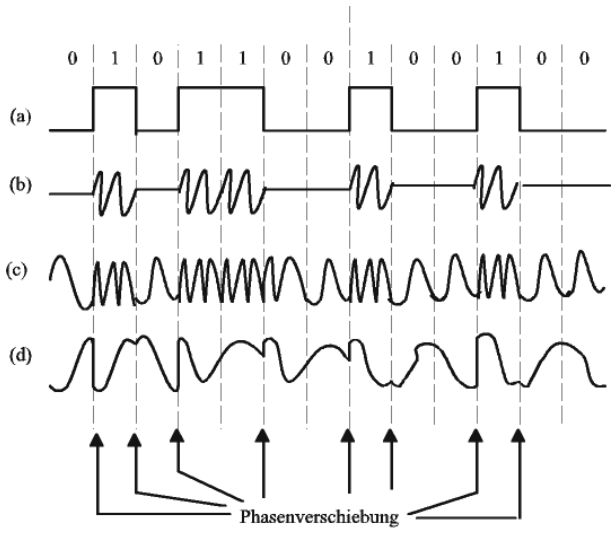
Modem: Modulator - Demodulator

Beispiel: Übertragung von digitalen Daten über das Telefonnetz

Modulationstechniken

- Amplitude Modulation (AM
- Frequency Modulation (FM). Basis für Frequency Division Multiplexing (FDM)
- Phase Modulation (PM)

Modulationstechniken



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

Bitrate und Baudrate

Bitrate

Anzahl der Bits (binären Nutzdatenwerte), die pro Sekunde übertragen werden.

Baudrate

Anzahl der Rechtecksignale des Leitungscodes pro Sekunde.

Bipolare Codes

Der bipolare Code ist ein Beispiel für eine Leitungscodierung mit mehr als zwei Signalwerten (hier tertiäres Signal).

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

Auch AMI (Alternate Mark Inversion) genannt.

Multiplexing: Mehrfachnutzung von Übertragungswegen

Übertragungsweg

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

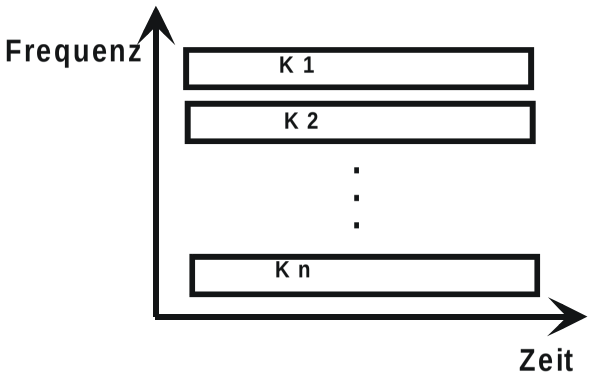
Übertragungskanal

Abstraktion eines Übertragungsweges für einen Signalstrom

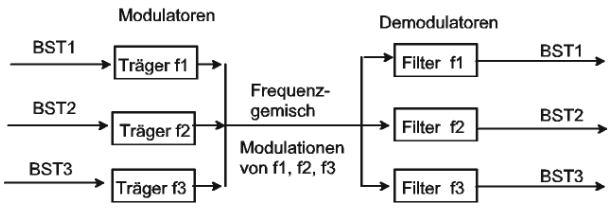
Auf einem Übertragungsweg können oft mehrere Übertragungskanäle parallel betrieben werden. So ist beispielsweise eine Aufspaltung der gesamten Übertragungskapazität eines Übertragungsweges auf verschiedene Sender-Empfänger-Paare möglich. Die Zusammenfassung von mehreren Übertragungskanälen auf einem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

Frequenzmultiplexing (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.

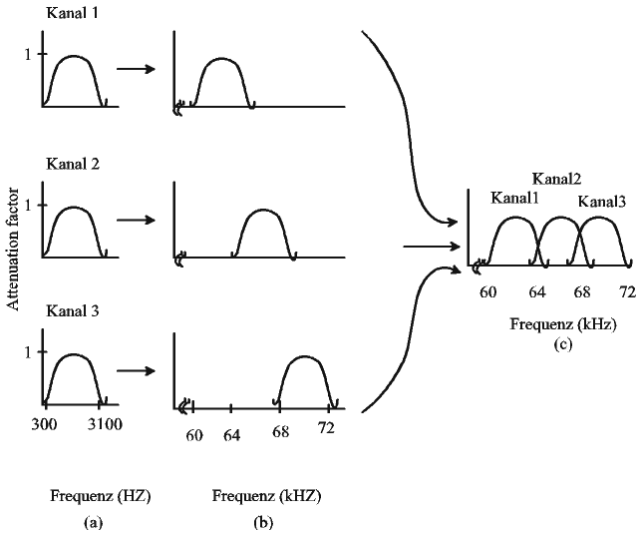


Schema der technischen Realisierung eines Frequenzmultiplex-Systems



BST i = Bitstrom i, entspricht Übertragungskanal i

Frequenzmultiplexing



(a) die ursprünglichen Bandbreiten
(b) die Bandbreiten mit verschobener Frequenz
(c) auf dem Übertragungsweg (z. B. Kabel)

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 der Daten erkennen! Es wird daher eine Kennung pro Datenblock (Paket, Zelle) erforderlich (Empfängeradresse, Kanalkennzahl o.Ä.)

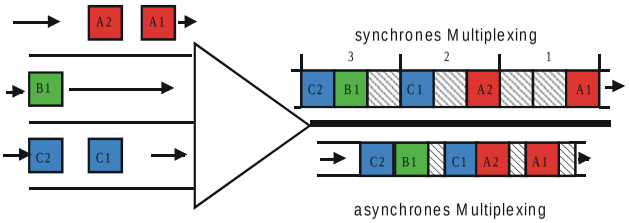
Schematischer Aufbau eines Übertragungsblocks mit Kennung

Übertragungsrichtung ->

inhalt	adr	inhalt	adr		inhalt	adr

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

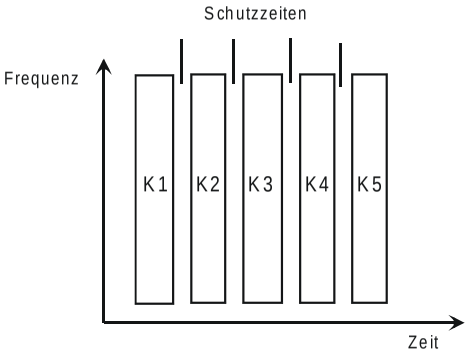
Multiplexing-Techniken im Vergleich



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).

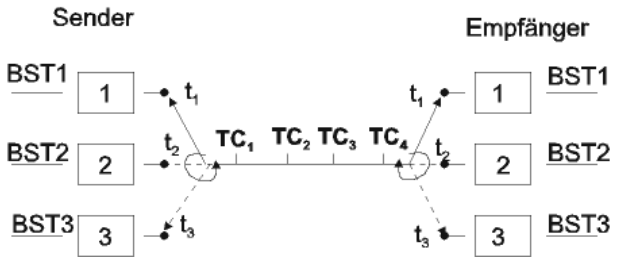
Diese zeitlich gestaffelte Übertragung mehrerer Signalströme wird als **Zeitmultiplexing** (TDM = time division multiplexing) bezeichnet.



Synchrones Zeitmultiplexing

Zeitmultiplexing ist nur für zeitdiskrete Signale einsetzbar (bevorzugt zeit- und wertdiskrete Signale = Digitalsignale)

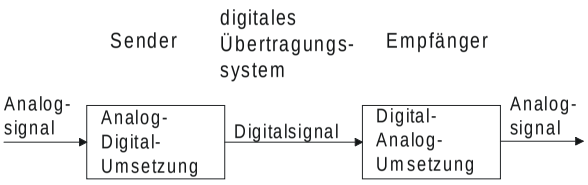
Festes Zeitmultiplex mit starrer Zeitscheibenzuteilung:



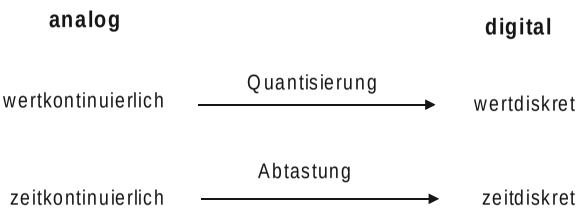
Jedem der n Sender wird periodisch eine Zeitscheibe (time slot, time slice) TC1, TC2 .... TCn zugeteilt. Sender, Abtaster und Detektionsmechanismus beim Empfänger laufen im gleichen Takt. Deshalb wird dieses Verfahren auch als **synchrones Zeitmultiplexing** bezeichnet.

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



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
- Video auf DVD
- DAB (Digital Audio Broadcast)
- Digitales Fernsehen

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

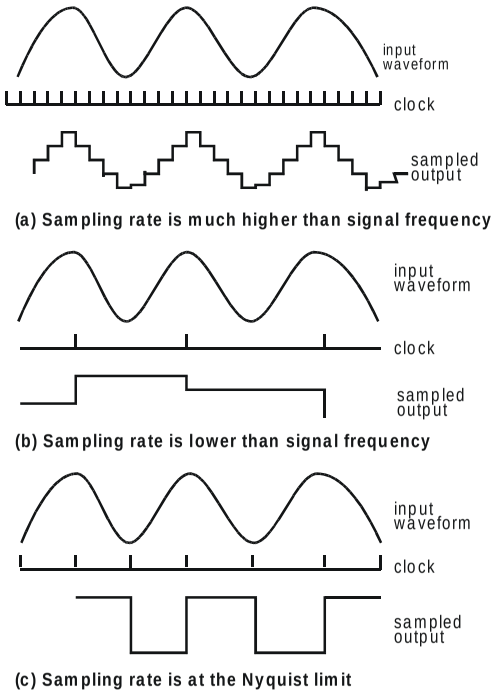
Abtasttheorem

Die Abtastfrequenz f_A muss 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: Abtasten bei verschiedenen Taktraten



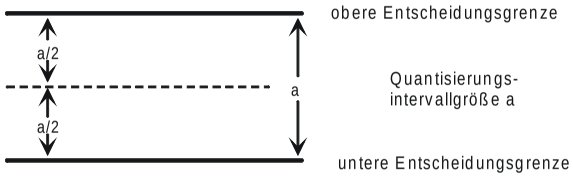
Codierung

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

Im einfachsten Fall wird eine binäre Darstellung des diskreten Zahlenwertes gewählt (Darstellung als Binärzahl).

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.

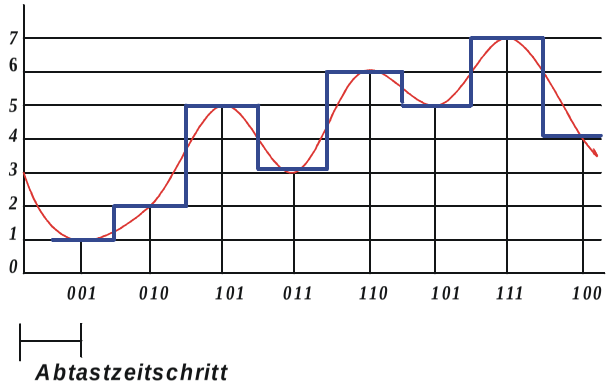


Rückwandlung

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

Maximaler Quantisierungsfehler: **a/2**

Zusammenfassende Darstellung



PCM-Fernsprechkanal

Schon vor vielen Jahren hat die ITU-T (vormals CCITT) zwei PCM-Übertragungssysteme genormt.

Ausgangspunkt:	Analoger CCITT-Fernsprechkanal
Frequenzlage:	300-3400 Hz
Bandbreite:	3100 Hz
Abtastfrequenz:	**f_A = 8 kHz**
Abtastperiode:	**$T_A = 1/ f_A = 1/8000$ Hz = 125 ms**

Die von der ITU-T gewählte Abtastfrequenz ist etwas höher als nach Shannon-Abtasttheorem erforderlich: 3400 Hz obere Bandgrenze ergibt 6800 Hz Abtastfrequenz. Für diese höhere Abtastfrequenz gibt es technische Gründe (Filtereinfluss, Kanaltrennung usw.).

Amplitudenquantisierung

Die Zahl der benötigten Quantisierungsintervalle wird bei der akustischen Sprachkommunikation (Fernsprechen) durch den Grad der Silbenverständlichkeit beim Empfänger bestimmt. Mit „Sicherheitszuschlag“ wurden von der ITU-T 256 Quantisierungsintervalle genormt (empirisch ermittelt).

Bei binärer Codierung werden die 256 Intervalle mit 8 Bits dargestellt.

Die Übertragungsgeschwindigkeit (Bitrate) für einen digitalisierten Fernsprechkanal ist demnach

Bitrate = Abtastfrequenz mal Codewortlänge
kbit/s = 8000/s x 8 bits
=
64kbit/s

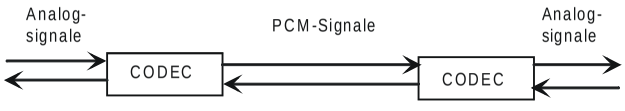
Pulse-Code-Modulation

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 der Praxis in großem Umfang eingesetzten **PCM-Technik**.

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



PCM-Systeme

Die praktische Gestaltung technischer 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, DVD, Video-Camcorder) eine zunehmende Rolle.

PCM-Hierarchie

PCM 30 = 2,048 Mbit/s (30 Kanäle)

PCM 120 = 8.448 Mbit/s

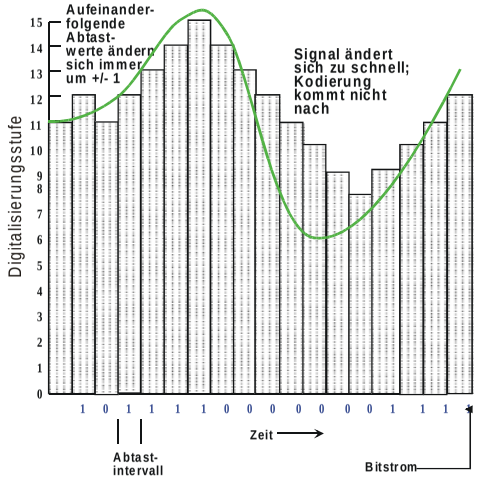
PCM 480 = 34,368 Mbit/s

PCM 1920 = 139,294 Mbit/s

PCM 7680 = 564,992 Mbit/s

Delta-Modulation

In der Regel ist die Änderung des Signals zwischen zwei Abtastzeitpunkten geringer als der Absolutwert des Signals. Die Delta-Modulation codiert Änderungen von +/- einer Quantisierungsstufe:



1 = steigendes Signal

0 = fallendes Signal

Ungleichförmige Quantisierung (1)

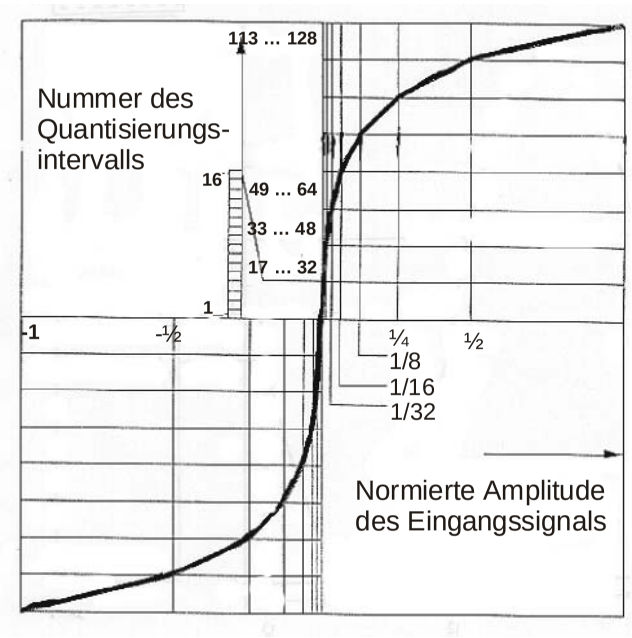
Bei gleichförmiger Quantisierung sind alle Intervalle gleich groß und vom Momentanwert des Signals unabhängig. Quantisierungsfehler machen sich bei gleichförmiger Quantisierung bei kleinen Signalwerten sehr stark bemerkbar (Quantisierungsrauschen).

Bei ungleichförmiger Quantisierung sind die Quantisierungsintervalle bei großer Signalamplitude größer und bei kleiner Amplitude kleiner.

Die ungleichförmige Intervallgröße wird durch einen dem Quantisierer vorgeschalteten (Signal-) *Kompressor* erzielt. Auf der Empfangsseite wird in inverser Funktion ein *Expander* eingesetzt. Er dient zur Wiederherstellung der ursprünglichen Größenverteilung der Signale (Dynamik der Signale).

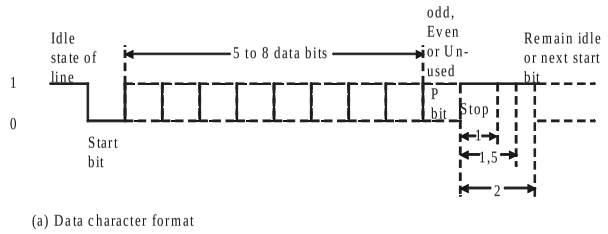
Als Kompressionskennlinien werden logarithmische Kennlinien verwendet, die schaltungstechnisch durch lineare Teilstücke approximiert werden.

Ungleichförmige Quantisierung (2)



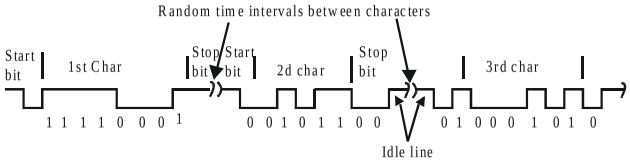
13 Segment-Kompressorkennlinie

Asynchrone Übertragung (2)



(a) Data character format

(a) Leitungscode für ein Zeichen

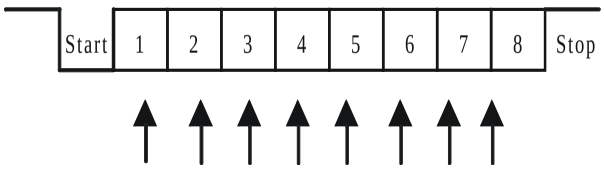


(b) 8-bit asynchronous bit stream

(b) Asynchroner Bitstrom

Asynchrone Übertragung (3)

Effekt der auseinander laufenden Takte



Asynchrone vs. synchrone Übertragung

Asynchron:

Es gibt kein explizites Taktsignal zwischen Sender und Empfänger.

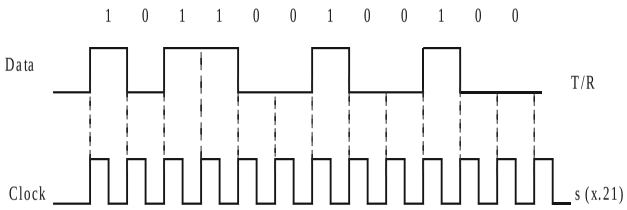
Synchron:

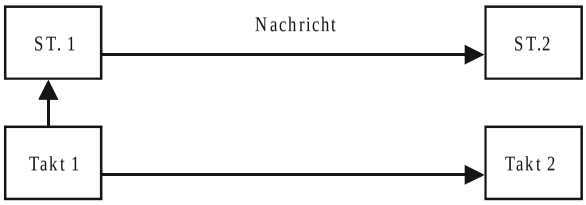
Ein Taktsignal wird über die Leitung übertragen. Es wird für die genaue Abstimmung der Bitimpulse (Synchronisation) auf beiden Seiten der Leitung benutzt.

Asynchrone Übertragung (1)

- Sender und Empfänger besitzen voneinander unabhängige (lokale) Taktgeber.
- Die „freie Leitung" entspricht einem kontinuierlich gesendeten 1-Bit.
- 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 dauert 1, 1.5 oder 2 Bit-Intervalle an.

Synchrone Übertragung (2)





Auslesen des Datenbits bei abfallender Flanke des Taktsignals

Asynchrone Übertragung (4)

Vorteile

- Es wird keine Synchronisierung der Taktgeber in den Endsystemen benötigt.
- Der Takt muss nicht über die Leitung übertragen werden.
- Leicht zu implementieren

Nachteile

- Die Taktgeber der Endsysteme können voneinander abweichen. Daher
  - ist die Rahmengröße sehr beschränkt (typischerweise ein Zeichen = 7-8 Bits)
  - nur anwendbar bei niedrigen Datenraten.

- Die Start- und Stop-Bits stellen einen Mehraufwand (overhead) dar.

  Beispiel:

  7-Bit ASCII-Zeichen als Daten,
  1 Paritätsbit,
  1 Start-Bit,
  1 Stop-Bit.

  Also: Nur 70% der Leitungskapazität stehen für echte Benutzerdaten zur Verfügung.

Synchrone Übertragung (1)

Sende- und Empfangstakt laufen über einen langen Zeitraum (beliebig lange) synchron.

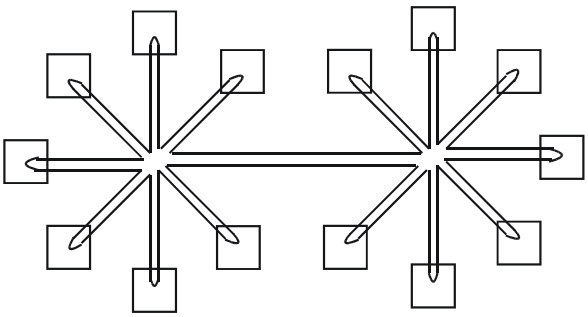
Eine Neusynchronisation nach jedem Zeichen (5-8 Bits) ist nicht erforderlich.

Taktsignal

Das 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, z. B. durch Verwendung von Manchester-Codes).

Strukturierte Verkabelung

– Möglichkeiten einer Ringverkabelung



Logischer Ring Physischer Stern

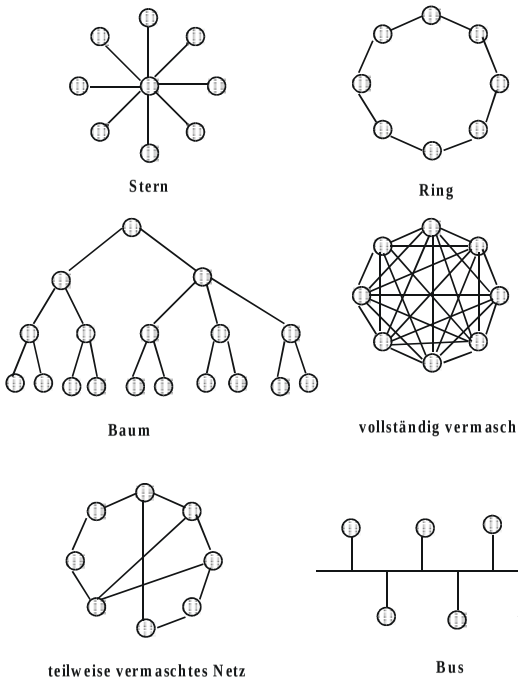
Physikalische Medien

– Adernpaar (verdrillt zur Verminderung von Störeinflüssen, deshalb "**twisted pair**"). Dies ist die klassische Telefonverkabelung. Trägt wenig auf, enge Biegungsradien, sehr preiswert.
– Adernpaar, abgeschirmt ("**shielded twisted pair**"). Weniger anfällig bzgl. der Einkoppelung von Störfrequenzen/Störströmen von außen. Trägt stärker auf als "unshielded twisted pair", teurer.
– Koaxialkabel: sehr störsicher, ermöglicht sehr hohe Übertragungsraten. Teuer. Weit verbreitet, zum Beispiel für die Ethernet-Verkabelung.
– Lichtwellenleiter (Glasfaser): sehr hohe Übertragungsraten, geringe Dämpfungen und Störeinkopplungen, aber aufwendige Verbindungs- und Anschlusstechnik, teuer.

2.4 Physikalische Medien

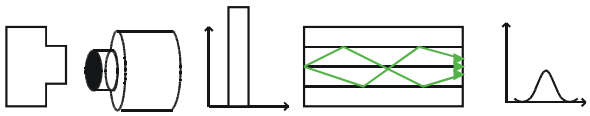
– **Netztopologien**
– **Kupferkabel**
  • **als verdrillte Adern (Twisted Pair)**
  • **als Koaxialkabel**
– **Glasfaserkabel**
– **Funk**
  • **Satellitenkommunikation**
  • **Mobilfunk**

Netztopologien

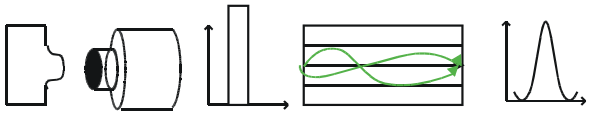


Technologie der Lichtwellenleiter

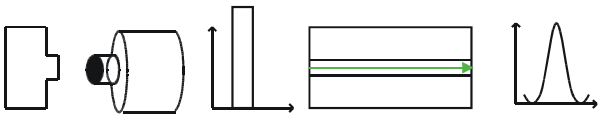
- Stufenindex-Faser



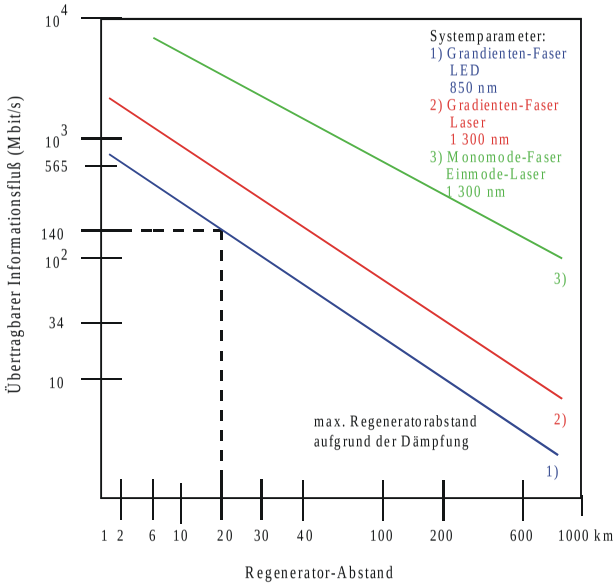
- Gradientenindex-Faser



- Monomode-Faser



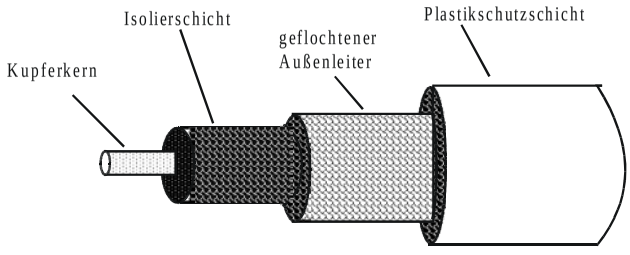
Glasfaser



Koaxialkabel für Ethernet

Das „klassische“ Bus-Kabel aus dem Standard

- 50 Ohm Koaxialkabel
- Maximale Kabellänge 500m
- Maximal 100 Transceiver (Anschlüsse von Stationen) pro Segment
- Maximal vier Repeater zwischen Sender und Empfänger
- Abstand zwischen den Anschlüssen muss ein Vielfaches von 2,5 m sein
- Datenrate 10 Mbit/s

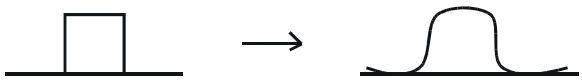


Glasfaserkabel

- Sehr hohe Datenraten:
- Theoretisches Limit: 300 TeraHz
- Praktisches Limit: ca. 10 GigaHz
- Transmitter und Receiver: Halbleiterelemente
- Beschränkende Faktoren:

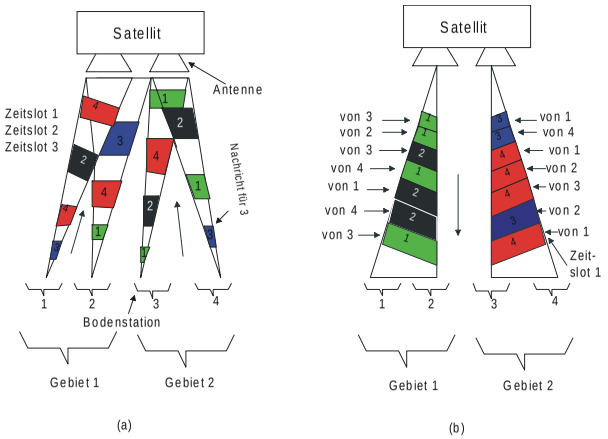
Absorption

Dispersion



Verbindungstechnik schwierig: nur 5 µm - 50 µm Durchmesser

Satellitenkommunikation



Innerhalb eines Sendegebietes wird Zeitmultiplexing (TDM) benutzt.

Die Farbe gibt die Zielstation (den Empfänger) an.

Mobilfunknetze

- Internationale Standards für den digitalen Zellfunk sind verabschiedet (z. B. GSM in Europa, UMTS in Vorbereitung)
- Starke Verbreitung für die Telefonie
- Digitale Bandbreite sehr niedrig (9,6 kbit/s)
  - Sie ist abhängig von der Breite des Trägerkanals.
  - Dieser muss wegen des Frequenzmuliplexings innerhalb einer Zelle sehr schmalbandig sein.
- Datenanwendungen in Mobilfunknetzen entwickeln sich rasant, insbesondere im Vorgriff auf UMTS.
- Zunehmend wird auch Funktechnik innerhalb von Gebäuden verwendet, z. B. für den Anschluss von autonomen Robotern an ein LAN oder für die Bürokommunikation von einem Meeting aus (z. B. mit WaveLAN).

Satellitenkommunikation

- Hohe Bandbreite
- Rundspruchnetz (Sicherheitsprobleme)
- Lange Verzögerung
  - Für Erdstationen mit festen Antennen ist ein geosynchroner Orbit notwendig.
  - Dieser liegt auf einer Höhe von 36,000 km.
  - Dies ergibt eine Verzögerung von 270 ms (hin zum Satelliten und zurück)
  - Die lange Verzögerung beeinflusst die Protokolle der höheren Schichten.
- Beispiel INTELSAT:
  - 794 PCM Simplexkanäle, jeder 64 KBit/s, plus ein 128 KBit/s Signalisierungskanal
  - Multiplexing mit FDM für Sendefrequenz und Empfangsfrequenz
  - Je ein Paar Simplexkanäle bildet einen Duplexkanal

Satellitennetze

- Satellitennetze sind wie Bus- und Ringnetze Rundspruchnetze (Broadcast-Netze).
- Der Satellit ist dabei eine logisch passive Verstärkerstation. Die Signale von der sendenden Erdstation werden dabei auf eine andere Frequenz umgesetzt und wieder ausgesendet.
- Eine sendende Erdstation kann durch die verschiedenen Sende- und Empfangsfrequenzen des Satelliten sofort feststellen, ob ein Datenpaket gestört wurde.
- Bei Satellitennetzen wird das Problem der Kanalzuweisung durch die hohen Signallaufzeiten erschwert (bei einem Tokenmechanismus wäre der Kanal jeweils für 270 ms unbenutzt!).

Beispiel 2: ADSL

ADSL (Asymmetric Digital Subscriber Line) und die verwandten Techniken HDSL, SDSL und VDSL übertragen sehr hohe Bitraten (bis zu 8 MBit/s) über unabgeschirmte Kupferdrähte (Telefondrähte).

Warum ist die ADSL-Technik wirtschaftlich interessant?

- Über 700 Millionen installierte Telefonanschlüsse weltweit
- 96% davon über Kupferkabel
- über 50% der gesamten Investition sind die Kabel!

=> ADSL ist eine sehr kosteneffektive Lösung, bereits installierte Kupferkapazität wird ausgenutzt.

---

* Für die Überlassung seines Foliensatzes über ADSL danke ich Herrn Mathias Gabrysch, NEC C&C Research Labs, Heidelberg

xDSL - hohe Datenraten auf Kupferkabeln

Wie sind derart hohe Datenraten möglich?

- Das Signal eines klassischen Modems muss das Telefonnetz Ende-zu-Ende durchqueren, es muss sich also bei der Modulation auf den **Sprachfrequenzbereich** von 300 bis ca. 3,4 kHz beschränken.
- Dem xDSL-Signal steht dagegen ein durchgehender Kupferdraht zur Verfügung, dessen Länge und Störsicherheit allerdings stark variieren kann. Es wird ein Frequenzbereich von 0 bis ca. 1,1 MHz zur Modulation ausgenutzt, wobei modernste Modulationstechniken zum Einsatz kommen.

2.5 Beispiele: V.24, ADSL

Beispiel 1: V.24, die serielle Schnittstelle

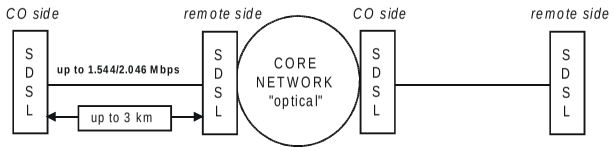
- **Geltungsbereich** (aus CCITT Recommendation V.24): "Diese Empfehlung bezieht sich auf die als Schnittstellenleitungen bezeichneten Verbindungsleitungen zwischen Datenendeinrichtung (DEE) und Datenübertragungseinrichtung (DÜE) zur Übertragung von binären Daten, Steuer- und Schritttaktsignalen. Diese Empfehlung erstreckt sich auch auf beide Seiten getrennter Zwischeneinrichtungen, die zwischen die Einrichtungen dieser beiden Kategorien eingefügt werden können."
- **Mechanische Eigenschaften**
  Die mechanischen Eigenschaften der Schnittstelle sind in den Standards ISO 2110 (25-polige Steckverbindung) oder ISO 4902 (37-polige und 9-polige Steckverbindung) festgelegt.
- **Elektrische Eigenschaften**
  Die elektrischen Eigenschaften der Schnittstellenleitungen werden in den entsprechenden eigenen Empfehlungen oder - in besonderen Fällen - in den Empfehlungen für die Datenübertragungseinrichtungen (Modems) behandelt.

Funktionale Eigenschaften

1	2	3	4	5	6	7	8	9
102	Erdleitung	x						
102a	DEE-Rückleiter	x						
102b	DÜE-Rückleiter	x						
102c	Gemeinsamer Rückleiter							
103	Sendedaten			x				
104	Empfangsdaten		x					
105	Sendeteil einschalten					x		
106	Sendebereitschaft				x			
107	Betriebsbereitschaft				x			
108/1	Übertragungsleitung anschalten					x		
108/2	DEE betriebsbereit					x		
109	Empfangssignalpegel				x			
110	Empfangsgüte				x			
111	Hohe Übertragungsgeschwindigkeit einschalten (DEE)					x		
112	Hohe Übertragungsgeschwindigkeit einschalten (DÜE)				x			
113	Sendeschritttakt (DEE)							x
114	Sendeschritttakt (DÜE)						x	

1= Nr. der Schnittstellenleitung	6= Steuerung von der DÜE
2= Bezeichnung der Schnittstellenleitung	7= Steuerung zur DÜE
3= Erde	8= Schrittakt von der DÜE
4= Daten von der DÜE	9= Schrittakt zur DÜE
5= Daten zur DÜE	

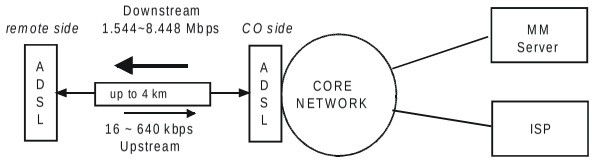
SDSL – Symmetric Digital Subscriber Line



„SINGLE LINE“-Version von HDSL (nur eine Zweidrahtleitung)

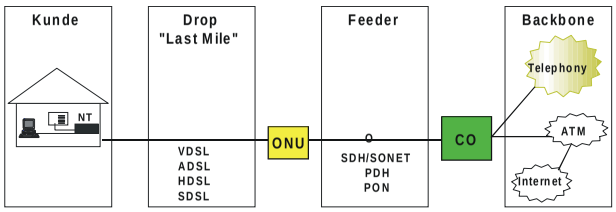
- symmetrische Bitraten
- basiert auf 2B1Q (QAM), CAP oder DMT-Modulationstechniken
- Telefondienst und T1/E1 simultan
- typische Anwendungen: wie HDSL

ADSL – Asymmetric Digital Subscriber Line



- Duplexübertragung mit fixen, asymmetrischen Datenraten über eine Kupfer-Zweidrahtleitung
- Die erreichten Übertragungsraten sind von der Entfernung und von der Leitungsqualität abhängig. **Die Adaption erfolgt automatisch**.
- basiert auf CAP- oder DMT-Modulationstechniken
- Telefondienst und ADSL-Datendienst simultan
- typische Anwendungen: schnelle Datenleitungen in Privathaushalte, Internet-Zugang, Fernzugang zu LANs.

Breitbandige Zugangsnetze

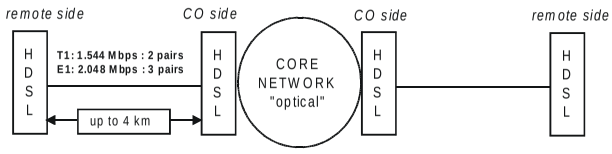


Feeder-Szenarien

- Fibre to the Building (FttB)
- Fibre to the Curb (FttC)
- Fibre to the eXchange (FttX)

ONU = Optional Network Unit
CO = Vermittlungsstelle („central office“)

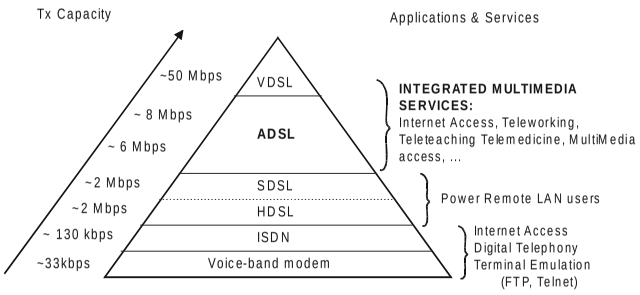
HDSL – High Data Rate Digital Subscriber Line



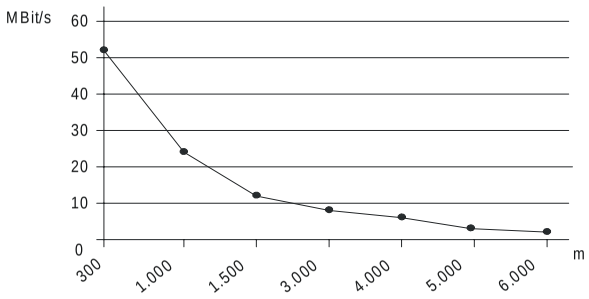
Hohe, symmetrische Bitraten über parallele Kupferdrähte

- entstanden als kostengünstige Technik für die Telecoms zur Realisierung von T1 oder E1 (1,5 Mbit/s oder 2 Mbit/s) über zwei bis drei Zweidrahtleitungen
- basiert auf 2B1Q (QAM, Quadrature Amplitude Modulation, 2 Bits pro Baud) oder CAP-Modulationstechniken (einer digitalen Variante von QAM)
- kein simultaner Telefondienst auf dem Kabel
- typische Anwendungen: T1 oder E1 in Gebäude, die keinen Glasfaseranschluss haben.

Übersicht über die xDSL-Techniken



Geschwindigkeit versus Entfernung bei xDSL



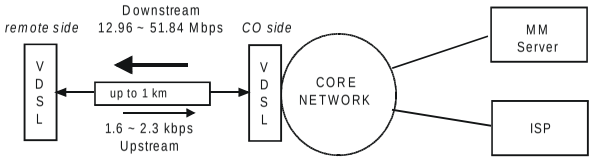
Kupfer-Faktoren
- Dämpfung frequenzabhängig
- Phasenverschiebung frequenzabhängig
- Übersprechen

Weitere Faktoren
- Impulsrauschen
- eingekoppelte Radiofrequenzen
- Weißes Rauschen („thermal noise“)

ADSL: Warum asymmetrisch?

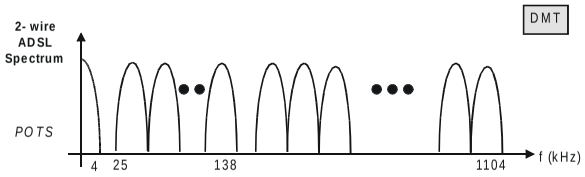
- Die Kabel-Topologie ist ein Baum.
- Die "upstream"-Signale laufen bei der Vermittlungsstelle in großer Zahl zusammen, was signifikante Störungen durch Übersprechen verursacht, wenn die Signale schon stark gedämpft sind. Dagegen laufen die "downstream"-Signale auseinander, zu getrennten Teilnehmer-Modems, wobei das Übersprechen sich wesentlich weniger auswirkt. Deshalb kann man in der "downstream"-Richtung höhere Bitraten realisieren.
- Zugleich können viele breitbandige Anwendungen mit asymmetrischen Bitraten gut leben, zum Beispiel das Browsen im Internet, Video-on-Demand usw.

VDSL – Very High Data Rate Digital Subscriber Line



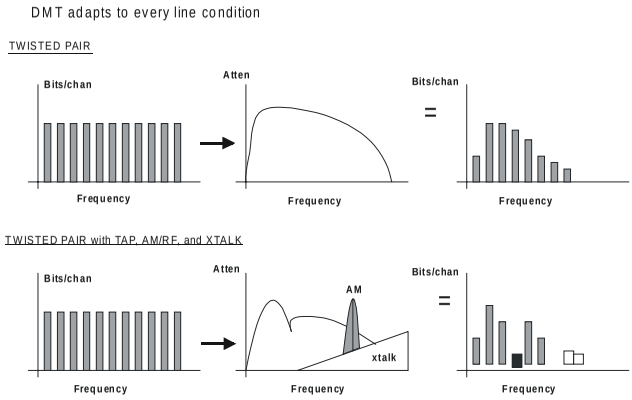
- Duplexübertragung mit fixen, asymmetrischen oder symmetrischen Datenraten über eine Zweidrahtleitung
- höhere Datenraten als ADSL, aber kürzere Kabellängen
- Telefondienst, ISDN und Datenübertragung simultan
- typische Anwendungen: nächste Generation der über ADSL zur Verfügung gestellten Dienste
- noch keine Standards, zurzeit in der Diskussion und Erprobung

DMT - Discrete Multitone Modulation



- Im Prinzip ein Freqenzmultiplexing (FDM) mit separater Bitraten-Adaption pro Trägerfrequenz
- Frequenzspektrum: 26 kHz bis 1.1 MHz
- Ist in 256 individuelle Sub-Trägerfrequenzen unterteilt, je 4 kHz breit
- Jeder Kanal übermittelt bis zu 60 kbit/s
- Der Telefondienst bzw. ISDN-Dienst liegt unterhalb der DMT-Frequenzen für Datendienste und wird separat geführt. Ein „splitter“ an beiden Enden der Leitung fügt ihn hinzu bzw. filtert ihn wieder heraus.
- ADSL ist ein ANSI-Standard (T1.413), inzwischen auch ein europäischer ETSI-Standard

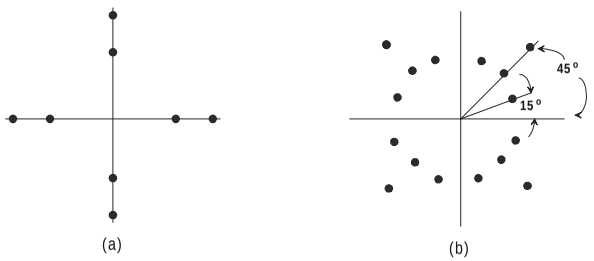
Automatische Bitraten-Adaption bei DMT



Bei ADSL wird die Bitrate dynamisch an die Länge und die Qualität der Übertragungsstrecke angepasst. Bei der DMT-Modulation messen die Modems ständig die Übertragungsgüte **jedes einzelnen Kanals** (jeder Trägerfrequenz) und adaptieren die Bitrate gemäß den aktuellen Eigenschaften.

Modulationstechniken für ADSL

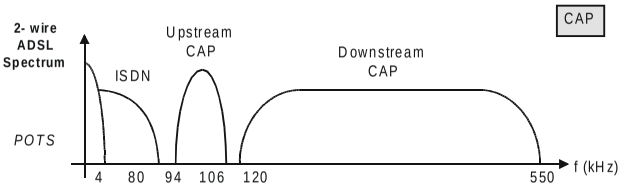
Basis: QAM (Quadrature Amplitude Modulation). Dies ist eine Kombination von Amplituden- und Phasenmodulation. Jeder "Datenpunkt" im Diagramm entspricht einer Bitkombination.



a) 2 Amplituden, 4 Phasensprungwinkel, 8 Datenpunkte, also 3 Bits pro Baud

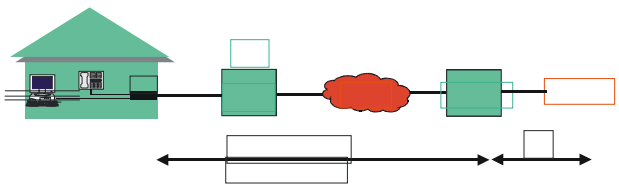
b) 16 Datenpunkte, also 4 Bits pro Baud (verwendet z. B. im V.32-Modem für 9600 bit/s bei 2400 Baud)

CAP - Carrierless Amplitude/Phase Modulation



- Eine Variante der Quadrature Amplitude Modulation
- Berechnung des kombinierten Signals durch einen digitalen Signalprozessor
- Benutzung einer einzigen Trägerfrequenz
- Telefondienst und ISDN liegen unterhalb des CAP-Frequenzspektrums

Protokolle in den höheren Schichten



ONU = Optical Networking Unit
CO = Central Office (Knotenvermittlungsstelle)
RAS = Remote Access Server

Internet-bezogener Verkehr wird die Geschäftsgrundlage für breitbandige Zugangsnetze bilden (zumindest in der nahen Zukunft). Zugang über ADSL zum Internet ist durch ähnliche Mechanismen möglich wie beim Zugang über ein Telefonmodem, also insbesondere über PPP zu einem Remote Access Server (RAS). PPP erfüllt dabei verschiedene Aufgaben, z.B. dynamische Vergabe von IP-Adressen, Authorisierung usw.

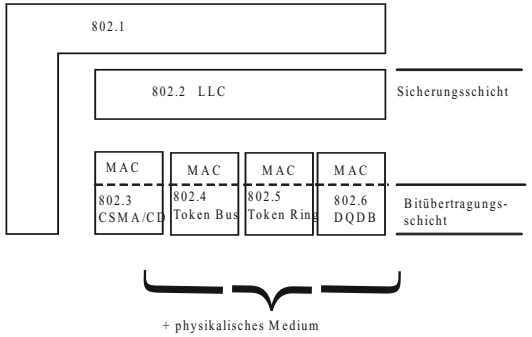
3. Sicherungsschicht (Data Link Layer)

Aufgaben der Sicherungsschicht (Schicht 2)

- Verdeckung von Übertragungsfehlern zwischen direkt benachbarten Partnern (Erkennung und Behebung)
- Flusskontrolle
- Bei LANs zusätzlich: Medienzugangskontrolle zum gemeinsamen Medium

Die Sicherungschicht im LAN (nach IEEE 802)

Standardisierung bei IEEE und ISO



3.1 Übertragungsfehler: Ursachen

- Weißes Rauschen
- Signalverzerrung ist abhängig von der Frequenz
- Übersprechen auf Leitungen
- Impulsstörungen
  - werden häufig durch Vermittlungseinrichtungen verursacht
  - dauern typischerweise ungefähr 10 ms (96 Bits zerstört bei 9600 bit/s)

Viele Fehler kommen kurz aufeinander folgend.

Vorteil: Nur wenige Blöcke enthalten Fehler.
Nachteil: Schwer zu erkennen und zu korrigieren.

Wahrscheinlichkeit von Übertragungsfehlern

Empirisch ermittelte Fehlerrate

Wahrscheinlichkeit für einen defekten Block für Telefonleitungen und Blöcke aus n Bytes:

$p(n) = 10^{-4}\ n^{0.8}$

n	p(n)
8	5.278031E-04
16	9.189586E-04
32	0.0016
64	2.785761E-03
128	4.850293E-03
256	8.444851E-03
512	1.470334E-02
1024	0.0256

3.2 Fehlererkennungs- und Fehlerkorrekturcodes

Fehler: 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 Redundanz, der Fehlerhäufigkeit, der Übertragungsstrecke, der Entdeckung und der eventuellen Korrektur eines Fehlers wird in der Codierungstheorie behandelt.

Paritätsbit

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	
Paritätsbit	0	0	1	gerade Parität
	1	1	0	ungerade

Querparität

- Wird zum Beispiel bei der asynchronen Übertragung verwendet.
- 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 9 Spuren passte.
- Das Redundanzbit ergänzt die Anzahl der Einsen innerhalb der Information auf eine gerade oder ungerade Anzahl.

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.

Hamming-Abstand (1)

Hamming-Abstand d
Anzahl der Bitpositionen, in denen sich zwei Codewörter c_1, c_2 unterscheiden.

Beispiel: d(10**001**001, 10**110**001) = 3

(Anzahl der Bits von c_1 XOR c_2)

Hamming-Abstand D eines vollständigen Codes C:

$$D(C) := \min \{ d(c_1, c_2) \mid c_1, c_2 \in C, c_1 \neq c_2 \}$$

Hamming-Abstand (2)

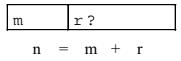
Satz
Die Fähigkeit eines Codes, Fehler zu erkennen und Fehler zu beheben, hängt von seinem Hamming-Abstand ab.

Erkenne e-Bit Fehler:
Ein Abstand von **e + 1** wird benötigt

Behebe e-Bit Fehler:
Ein Abstand von **2e + 1** wird benötigt

Wieviel Redundanz braucht man?

Das Codewort bestehe aus m Zeichen-Bits. Wieviele Prüfbits werden benötigt, um einen 1-Bit-Fehler zu beheben?



- Es gibt 2^m legale Zeichencodes.
- Pro Codewort muss es n illegale Codewörter im Abstand von einem Bit geben.
- 2^n ist die Gesamtzahl der Codewörter.
- $(n + 1)\, 2^m \leq 2^n$ **=>** $(m + r + 1) \leq 2^r$
- Dies ergibt die untere Grenze für r.
- Beispiel: m = 7
  $(8 + r) \leq 2^r$ **=>** $r \geq 4$

Nachteile von fehlerbehebenden Codes

Großer Overhead (viel Redundanz) auch im Falle einer fehlerfreien Übertragung!

Fehlererkennung und Sendewiederholung ist in klassischen, auf Kupferkabel basierenden Netzen effizienter!

Beispiele

Fehlererkennender Code:
Code mit einem einzigen Paritätsbit (gerade oder ungerade)
=> Hamming-Abstand = 2
=> Erkennung eines 1-Bit-Fehlers ist möglich
(oder aller Fehler mit einer ungeraden Anzahl Bits)

Fehlerbehebender Code:
Vier Codewörter:
00000 00000, 00000 11111, 11111 00000, 11111 11111

=> Hamming-Abstand = 5
=> Korrektur von 2-Bit-Fehlern möglich

Beispiel: 00000 **00**111 => 00000 11111
2-Bit-Fehler nächstes Codewort

Cyclic Redundancy Check (CRC)

Grundidee

- Betrachte Bitfolgen als Darstellung eines Polynoms nur mit den Koeffizienten 0 und 1.

  Beispiel: 11001 = $x^4 + x^3 + x^0$
- Wähle ein **Generatorpolynom G(x)** vom Grad g.
- Hänge an die Nachricht M(x) eine Prüfsumme derart an, dass das Polynom T(x), dargestellt durch die Nachricht mit angehängter Prüfsumme, durch G(x) teilbar ist.
- Übertrage T(x).

Algorithmus CRC

Wir haben G(x) vom Grad g und hängen g 0-Bits an die Nachricht an, also $x^g M(x)$.

1. Teile $x^g M(x)$ durch G(x), Division modulo 2
2. Subtrahiere den Rest von $x^g M(x)$, modulo 2
3. Das Ergebnis ist T(x), die Nachricht samt Prüfsumme für die Übertragung.

Beispiel für die CRC-Berechnung

```
Rahmen:    1101011011,  Generator: 10011. Vier 0-Bits anhängen:
11010110110000 : 10011 = 1100001010
10011
 10011
 10011
  00001
  00000
   00010
   00000
    00101
    00000
     01011
     00000
      10110
      10011
       01010
       00000
        10100
        10011
         01110
         00000
  Rest:   1110
```

(Division modulo 2: kein Übertrag bei der Subtraktion)

Generatorpolynome in internationalen Standards

CRC-12 = $x^{12} + x^{11} + x^3 + x^2 + x + 1$
Wird für 6-Bit-Zeichencodes benutzt.

CRC-16 = $x^{16} + x^{15} + x^2 + 1$ (ISO)
CRC-CCITT = $x^{16} + x^{12} + x^5 + 1$ (CCITT)
Beide werden für 8-Bit-Zeichencodes benutzt.

Fehlererkennung mit CRC-16, CRC-CCITT (16 Bits)
(Ergebnisse aus der Codierungstheorie):

- alle einfachen und zweifachen Bitfehler
- alle Fehler mit einer ungeraden Bitzahl
- alle stoßweisen Fehler mit einer Länge •16
- 99,998% aller längeren stoßweisen Fehler.

Implementierung des CRC
In Hardware (z. B. im HDLC-Chip) mit Hilfe eines Schieberegisters und der bitweisen XOR-Funktion. Sehr effizient!

3.3 Bitstopfen und Rahmenbegrenzer

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

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

Lösung: **Bit Stuffing (Bitstopfen)**

Als Begrenzer („flag“)wählt man 01111110. Der Sender fügt nach fünf Einsen im Nutzdatenstrom **IMMER** eine 0 ein.

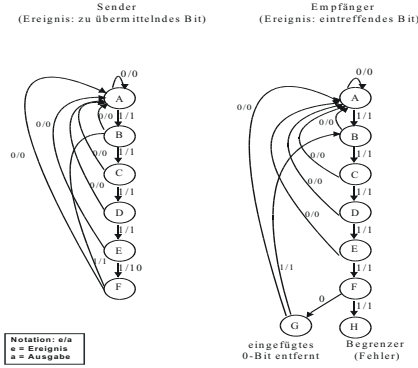
Beispiel für Bit Stuffing

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
Emp-fänger	0	1	1	1	1	1		1	0	1	0	1	1	1	1	1		0	1	0	1

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

Bit Stuffing: Endliche Automaten

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



Rahmenformat der Schicht 2

Rahmen („frame") in der Schicht 2

8	8	8	>=0	16	8
Flag 01111110	Address	Control	Data	Checksum	Flag 01111110
↑ Begrenzer					↑ Begrenzer

3.4 Bestätigungen und Sequenznummern

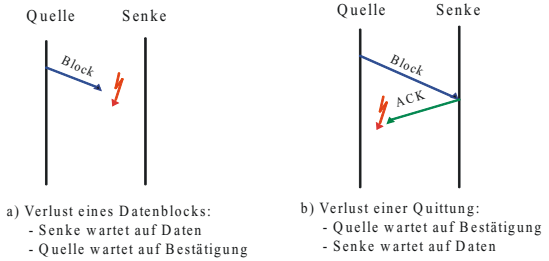
Acknowledgement = Bestätigung, Quittung

Bestätigungen und Sequenznummern werden benutzt für die

- Fehlerbehebung (fehlerhafte oder verlorene Blöcke)
- Pufferverwaltung
- Flusskontrolle.

Bestätigungen (ACKs)

Zwei Fälle von Datenverlust:



a) Verlust eines Datenblocks:
- Senke wartet auf Daten
- Quelle wartet auf Bestätigung

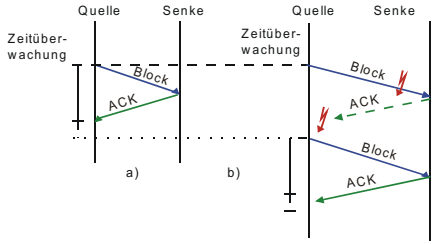
b) Verlust einer Quittung:
- Quelle wartet auf Bestätigung
- Senke wartet auf Daten

Ohne Zeitschranke (Time-out): Blockierung des Senders

Lösung: Einführung einer **Zeitschranke** (Time-out) beim Sender

Bestätigungen mit Zeitschranke

Zeitüberwachung auf der Senderseite

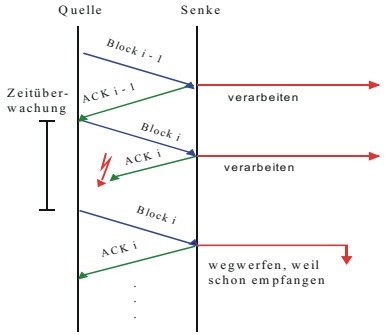


Neues Problem:

Der Block wird zweimal übertragen. Der Empfänger weiß nicht, ob er den Block zur Verarbeitung weiter nach oben geben soll.

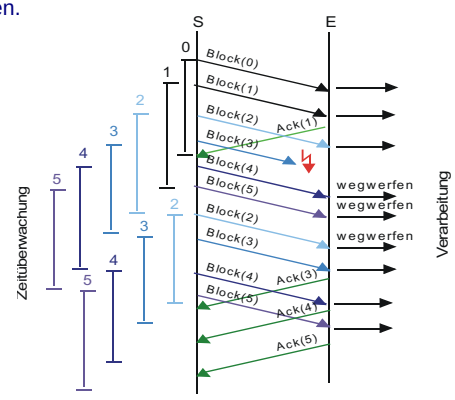
Bestätigungen mit Zeitschranke und Sequenznummern

Lösung: **Sequenznummern**
Jeder Block erhält eine Sequenznummer. **Bei der Wiederholung des Rahmens wird die Sequenznummer beibehalten.**



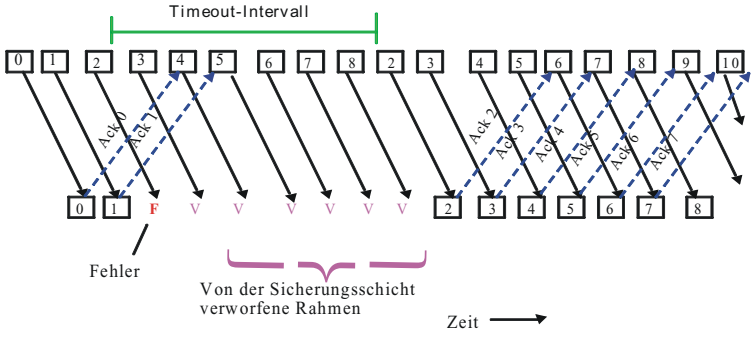
Sequenznummern

Sequenznummern können auch in den Bestätigungen verwendet werden. Mit einer Bestätigung können dann mehrere Informationsblöcke quittiert werden.



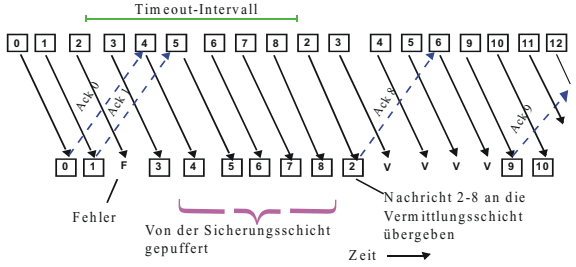
Fehlerbehebung durch "go-back-n“ ohne Pufferung

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



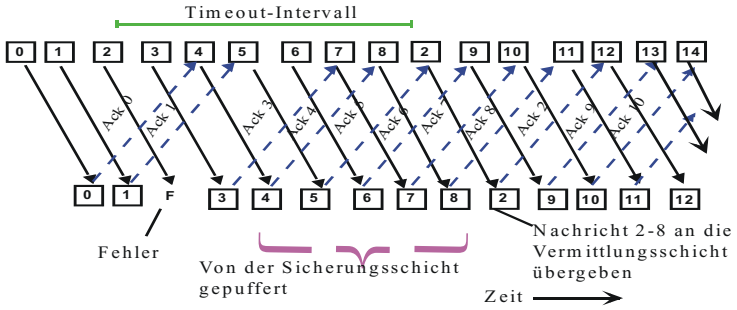
Fehlerbehebung durch "go-back-n“ mit Pufferung

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 mit dem ersten noch nie übertragenen Rahmen weiter.



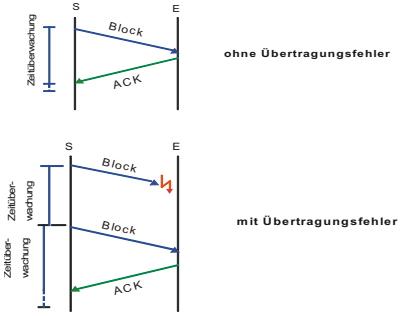
Fehlerbehebung durch "selective repeat"

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.



Passive Fehlerkontrolle

Kein Versenden von NACKs

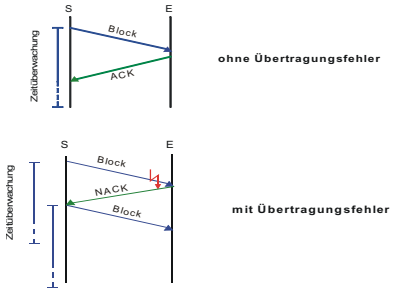


Nachteil der passiven Fehlerkontrolle:

- Keine Unterscheidung zwischen fehlenden und fehlerhaften Blöcken
- Zeitverzögerung bis zur Wiederholung des Sendevorgangs

Aktive Fehlerkontrolle

- Versenden von NACKs

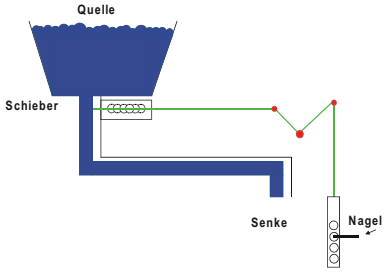


NACK = Negative Acknowledgement

In Rechnernetzen wird sowohl die aktive als auch die passive Fehlerkorrektur verwendet.

3.5 Flusskontrolle

Das Prinzip der Flusskontrolle



Stellt eine Rückkopplung zur Verfügung, um zu verhindern, dass der Sender den Empfänger überschwemmt.

Ein einfaches Stop-and-Wait-Protokoll

Annahmen:

- fehlerfreie Übertragung
- beschränkte Anzahl von Puffern
- die Verarbeitungsgeschwindigkeiten bei Sender und Empfänger können sich unterscheiden.

Verfahren:

Benutze ACK für eine einfache Flusskontrolle. Es ist nie mehr als ein Rahmen auf dem Weg.

Ein einfaches Stop-and-Wait-Protokoll (Sender)

```
procedure sender;
var s:      frame
    buffer: message
    event:  EvType
begin
   while (true) do
      begin
         FromHost(buffer);
         s.info = buffer;
         sendf(s);
         wait(event);       (*wait for ack*)
      end;
end;
```


Ein einfaches Stop-and-Wait-Protokoll (Empfänger)

```
procedure receiver;
var r,s:   frame;
    Event: EvType;
begin
   while (true) do
      begin
         wait(event); (*wait for frame arrival*)
         getfr(r);
         ToHost(r.info);
         sendf(s);          (*send ack to sender*)
      end;
end;
```


Flusskontrolle mit Schiebefenster (sliding window)

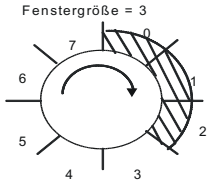
Fenstermechanismus („sliding window flow control“)

Nach dem Verbindungsaufbau besitzt der Sender das Recht, so viele Informationsrahmen zu senden, wie durch die Fenstergröße vorgegeben ist.

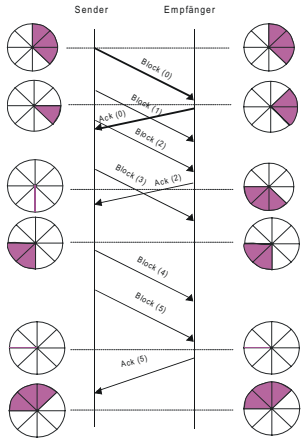
Spätestens dann muss vom Empfänger eine Bestätigung eintreffen, ansonsten unterbricht der Sender die Übertragung von Rahmen.

Der Empfänger kann schon vor dem Erreichen der Fenstergröße Bestätigungen an den Sender übermitteln (Öffnen des Fensters).

Beispiel

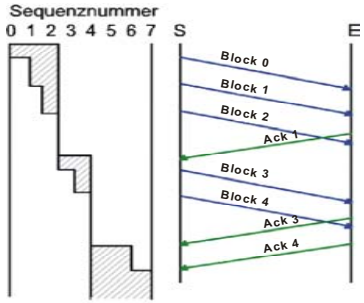


Schiebefenster (1)



Schiebefenster (2)

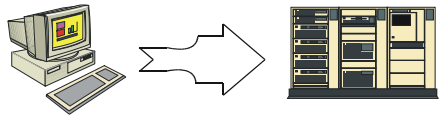
Öffnen und Schließen des Fensters



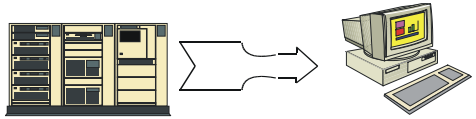
Dieselben Sequenznummern werden in den meisten Protokollen **sowohl** zur Fehlerkontrolle **als auch** zur Flusssteuerung verwendet!

Fenstergröße und Puffer beim Empfänger (1)

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



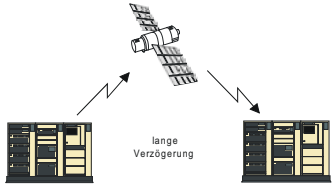
a) Langsame Quelle (PC) – schnelle Senke (Großrechner): 1 Puffer ist ausreichend



b) Schnelle Quelle (Großrechner) – langsame Senke (PC): 1 Puffer ist ausreichend

Fenstergröße und Puffer beim Empfänger (2)

c) Balanciertes Verhältnis, z. B. zwischen zwei Großrechnern: es sind w Puffer erforderlich, z. B.: $2 \leq w \leq 7$



d) Übertragung über einen Nachrichtensatelliten: es sind w Puffer erforderlich, z. B.: $16 \leq w \leq 127$

Fenstergröße und Puffer beim Empfänger (3)

Merke: 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.

3.6 Beispiele: HDLC, PPP

3.6.1 HDLC (High-Level Data Link Control)

Das wichtigste Protokoll der Sicherungsschicht.

Eng verwandt sind:

LLC 2 Logical Link Control Typ 2 in LANs

LAP-B Link Access Procedure – Balanced (CCITT; Schicht 2 von X.25)

HDLC-Standards

ISO 3309 HDLC procedures – Frame structure
ISO 4335 HDLC procedures – Elements of procedures
ISO 6159 HDLC unbalanced classes of procedures
ISO 6256 HDLC balanced class of procedures

X.25, Level 2 Link Access Procedure Balanced (LAP B) (Schicht 2 der CCITT-Empfehlung X.25)

Historisch: eine internationale Standardisierung des Protokolls SDLC (Synchronous Data Link Control) von IBM

HDLC – High-Level Data Link Control

Basis-Rahmenformat

	Flag 01111110	Address	Control	Data	Frame Check	Flag 01111110
Bits:	8	8	8	≥0	16	8

Transparenz durch Bitstopfen (Bit Stuffing)

Frame Check Sequence (FCS):
Cyclic Redundancy Check (CRC) der Länge 16 mit

$$G(x) = x^{16} + x^{12} + x^5 + 1$$

Prozedurklassen (1)

Asymmetrische Konfiguration (unbalanced)

- Primärstation: Management der Verbindung
- Eine oder mehrere Sekundärstation(en)
- Punkt-zu-Punkt-Verbindungen und Mehrpunkt-Verbindungen
- Normaler Antwortmodus (Normal Response Mode):
  Eine Sekundärstation kann nur nach einem Sendaufruf ("polling") durch die Primärstation senden.
- Asynchroner Antwortmodus (Asynchronous Response Mode):
  Die Primär- und die Sekundärstation kann übertragen, wenn die Leitung frei ist.

Symmetrische Konfiguration (Balanced)

- Totale Symmetrie zwischen den Stationen
- nur Punkt-zu-Punkt-Verbindungen (keine Mehrpunktverbindungen)
- nur asynchroner Modus.

Prozedurklassen (2)

- "Unbalanced" Operation im Normalen Antwortmodus
- "Unbalanced" Operation im Asynchronen Antwortmodus
- "Balanced" Operation im Asynchronen Antwortmodus

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

Adressfeld

Ursprünglich 8 Bits lang, diente zur Adressierung der Sekundärstation beim Polling.

Erweiterte Adressierung

Die Größe des Adressfelds kann ein Vielfaches von 8 Bits sein.

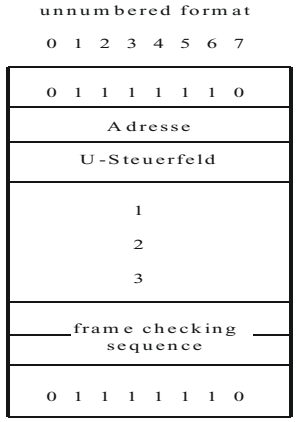
Steuerfeld (Control Field)

Es gibt drei verschiedene Rahmentypen im HDLC:

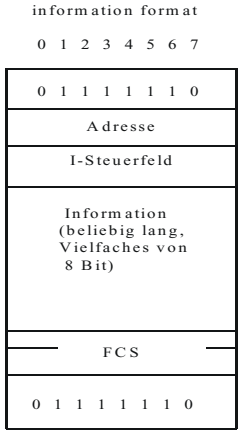
- I-Rahmen (information frames)
  werden zur Datenübermittlung benutzt
- S-Rahmen (supervisory frames)
  werden zur Steuerung des Datenflusses benutzt
- U-Rahmen (unnumbered frames)
  steuern die Verbindung.

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

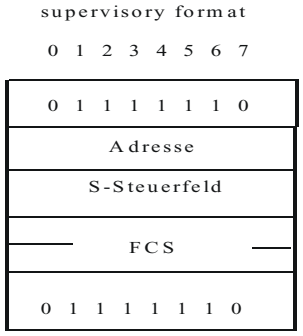
HDLC - Rahmenformate (1)



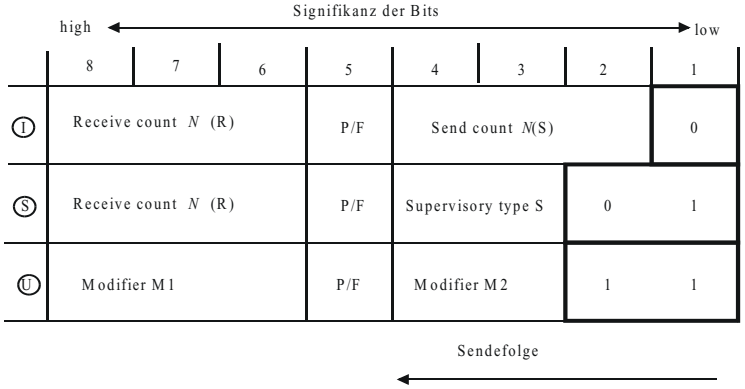
HDLC - Rahmenformate (2)



HDLC - Rahmenformate (3)



Inhalt des Steuerfelds im Basis-Format

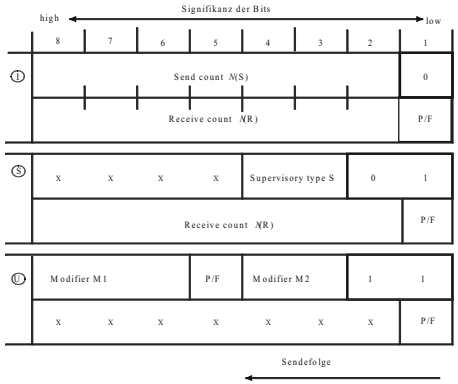


Flusskontrolle mit Schiebefenster in HDLC

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

- **N(R)** Empfangszähler
- **N(S)** Sendezähler (Sequenznummer des Rahmens)
- N(R) und N(S) sind Zähler für die Anzahl der I-Rahmen, die empfangen und gesendet wurden.
- N(R) "piggybacked acknowledgement": ACK = Nummer **des ersten nicht erhaltenen** Rahmens (= Nummer des nächsten erwarteten Rahmens) kann in I-Rahmen des Rückkanals übertragen werden.

Erweitertes Steuerfeld (16 Bits)



N(S) und N(R) sind jetzt Zähler mit je 7 Bits. Modulo 128 => Fenstergröße = 127
Anwendung: z. B. für Satellitenverbindungen

Command / Response

Command Rahmen eines beliebigen Typs, der durch eine Primärstation gesendet wird

Response Rahmen eines beliebigen Typs, der durch eine Sekundärstation gesendet wird.

Kombinierte Stationen können sowohl Commands als auch Responses senden.

P/F-Bit: Poll-Bit in Command-Rahmen. Der Empfänger muss diesen Rahmen durch einen Response-Rahmen mit gesetztem Final-Bit bestätigen.

Supervisory-Rahmen (1)

Zwei Bits zur Unterscheidung => 4 verschiedene Typen von S-Rahmen:

- Receive Ready (RR) 00
  Station kann I-Rahmen empfangen. N(R): nächster erwarteter Rahmen
- Receive Not Ready (RNR) 01
  Eine Station zeigt der anderen an, dass sie zeitweise nicht empfangen kann. Sender stoppt die Übertragung
- Reject (REJ) (entspricht einem NACK) 10
  Zeigt an, dass ein Übertragungsfehler entdeckt wurde.

N(R): erster Rahmen in der Folge der nicht korrekt empfangen wurde, also der Rahmen, der erneut gesendet werden soll.

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

Supervisory-Rahmen (2)

- Selective Reject (SREJ) 11
  Zeigt dem Sender an, dass ein Übertragungsfehler entdeckt wurde. Fordert nur die erneute Übertragung des Rahmens spezifiziert durch N(R) an.

REJ und SREJ werden nur in der beidseitigen Datenübermittlung (im Vollduplex-Betrieb) benutzt.

LAP B: es gibt RR, RNR, REJ, kein SREJ

Unnumbered-Rahmen (1)

Fünf Bits verfügbar => 32 verschiedene U-Rahmen sind möglich (nicht alle werden benutzt)

- Zum Setzen der Modi für die drei Klassen werden folgende Kommandos benutzt:
  - Set Normal Response Mode (SNRM)
  - Set Normal Response Mode Extended (SNRME)
  - Set Asynchronous Response Mode (SARM)
  - Set Asynchronous Response Mode Extended (SARME)
  - Set Asynchronous Balanced Mode (SABM)
  - Set Asynchronous Balanced Mode Extended (SABME)

Unnumbered-Rahmen (2)

- Sie werden benutzt, um eine Verbindung auf der Ebene der Sicherungsschicht herzustellen.
- N(S), N(R) = 0 (daher die Bezeichnung „unnumbered“)
- Erweiterte Version ("Extended"): benutzt für das erweiterte Format des Steuerfeldes (Control Field)

Unnumbered-Rahmen (3)

- **Unnumbered Acknowledgement (UA)**
  Wird zur Bestätigung eines U-Rahmens benutzt
- **Disconnect (DISC)**
  Wird durch die Primärstation oder durch eine kombinierte Station benutzt, wenn sie die Verbindung lösen will. Bestätigung: UA
- **Disconnected Mode (DM)**
  Wird benutzt, um anzuzeigen, dass eine Station logisch abgekoppelt ist. Nur Kommandos, die einen Modus setzen, sind für eine logisch abgekoppelte Station gültig.
  DM wird normalerweise als Antwort auf alle Kommandos mit Ausnahme derjenigen gesendet, die einen Modus setzen.
- **Frame Reject (FRMR)**
  Wird gesendet, wenn ungültige Zustände entdeckt werden.

3.6.2 PPP (Point-to-Point Protocol)

Ein Punkt-zu-Punkt-Protokoll für den Zugang zum Internet über Wählleitungen (insbesondere Modem-Strecken) und über Standleitungen. Standardisiert in den RFCs 1661 und 1662.

Eigenschaften

- Stark angelehnt an HDLC
- Das Framing entspricht HDLC mit bestimmten Einschränkungen
- Das Kontrollprotokoll LPC (Link Control Protocol) dient vor allem zur Verhandlung von Parameter-Werten.
- PPP kann sowohl im byte-orientierten Modus (mit Byte Stuffing) als auch im bit-orientierten Modus (mit Bit Stuffing) eingesetzt werden.
- Die darunter liegende Schicht 1 muss voll-duplex sein.
- Die darunter liegende Schicht 1 kann entweder synchron oder asynchron sein (8 Bits, keine Parität).

PPP-Rahmenformat

flag	Rahmenbegrenzer: `01111110`
addr	immer auf `11111111` gesetzt, da im Punkt-zu-Punkt-Verkehr keine Adresse benötigt wird
ctrl	„control“, immer auf 00000011 gesetzt (entspricht HDLC-UI mit dem P/F-Bit auf 0)
protocol	Kennzeichner für das im Datenfeld transportierte Protokoll, z.B. IP, IPX, LPC
data	Nutzlast (Benutzerdaten)
checksum	FCS (frame check sequence. Berechnet mit CRC wie in HDLC, entweder in Hardware oder in Software. 16 Bits (default) oder 32 Bits lang, die Länge wird mit LPC verhandelt
flag	Wenn unmittelbar ein weiterer Rahmen folgt, gibt es nur **ein** „flag“ zwischen den beiden Rahmen.

LPC (Link Control Protocol)

LPC ist das Kontrollprotokoll von PPP. Es dient zum Verbindungsaufbau, sobald das Trägersignal entdeckt wird, und zum Verhandeln von Parametern.

LPC-PDUs werden in PPP-Rahmen (frames) transportiert.

4 Lokale Netze (LANs)

Was ist ein LAN?

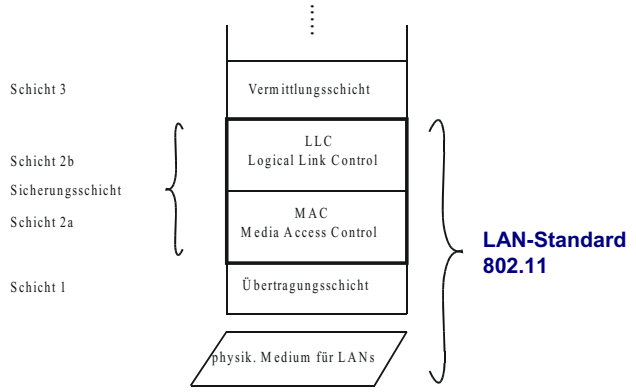
Ein **LAN** (Local Area Network) ist ein Netzwerk für die bitserielle Übertragung von Informationen zwischen **unabhängigen**, untereinander verbundenen Stationen.

Es befindet sich rechtlich unter der Kontrolle des Benutzers und ist in der Regel auf **den Bereich innerhalb der Grundstückgrenzen** beschränkt.

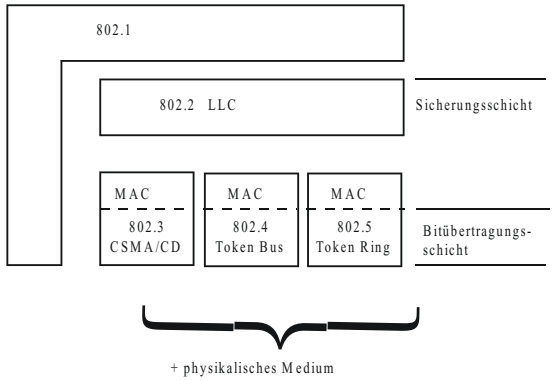
Merkmale eines lokalen Netzes

- Hohe Geschwindigkeit (10 -1000 MBit/s)
- Leichter, kostengünstiger Anschluss
- Keine Regulierungen der Telekom zu beachten
- Anschluss unterschiedlicher Geräte
  - PCs
  - Unix-Workstations und -Server
  - Großrechner
  - Drucker und andere periphere Geräte
- Übergang auf Weitverkehrsnetze über Router (Schicht 3) oder Gateways (Schicht 7)

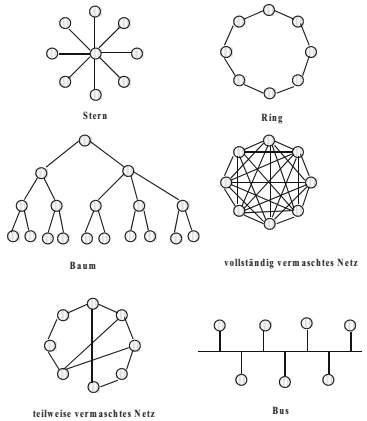
LANs im ISO-Referenzmodell



IEEE Standard 802



4.1 Topologien für lokale Netze



Punkt-zu-Punkt-Netz vs. Broadcast-Netz

Punkt-zu-Punkt-Netz

- Jeweils genau zwei Stationen sind physikalisch verbunden.
- Multicast und Broadcast erfordern die explizite Vervielfältigung der Nachricht in den Zwischenknoten.
- Im Weitverkehrsnetz, im teilweise vermaschten Graphen, muss eine explizite Wegewahl erfolgen, um einen bestimmten Empfänger zu erreichen.

Broadcast-Netz

- 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.

4.2 Medienzugangskontrolle

Medium Access Control (MAC)

Problem:

- Broadcast-Medium
- unabhängige Stationen

  => Sendekollisionen

Lösung: Medienzugangskontrolle

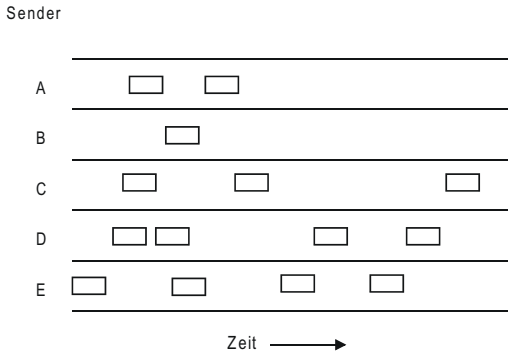
Zwei Medienbelegungsprinzipien:

- Kollisionsentdeckung
  Lasse Kollisionen stattfinden, entdecke sie, wiederhole die Übertragung.
- Kollisionsvermeidung
  Verwende ein zirkulierendes Token, um den Zugriff auf das Medium zu steuern.

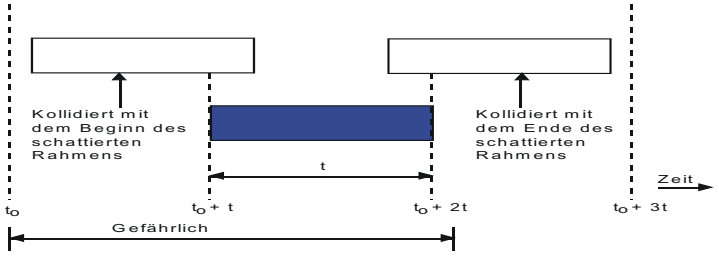
4.3 ALOHA

- Ein einfaches MAC-Protokoll mit Kollisionsentdeckung.
- Wurde beim "Packet Radio System" der Universität von Hawaii 1970 erstmals verwendet.
- Im reinen ALOHA werden die Rahmen zu willkürlichen Zeiten übertragen. Jede Station kann zu jeder Zeit senden. Falls der Sender eine Kollision entdeckt, wartet er eine durch Zufall bestimmte Zeitspanne und wiederholt dann die Übertragung.

ALOHA - Beispiel



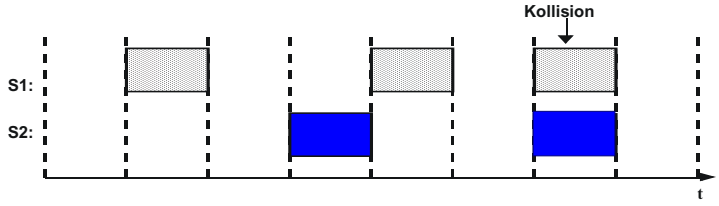
Kollidierende ALOHA-Pakete



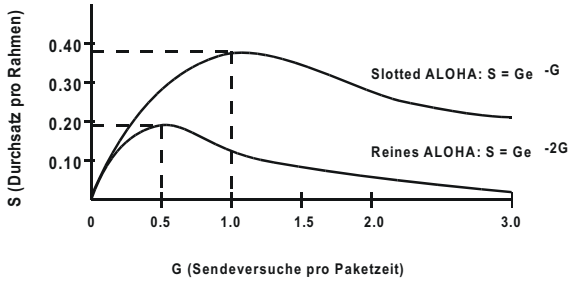
Gefährliche Zeitspanne für den mittlerem Rahmen. Falls auch nur das erste Bit eines neuen Rahmens das letzte Bit eines fast beendeten Rahmens überschneidet, werden beide Rahmen total zerstört, beide müssen später nochmals übermittelt werden. Eine Prüfsumme kann nicht (und sollte auch nicht) zwischen einem totalem und einem knappen Verlust unterscheiden. Knapp daneben ist auch vorbei!

Slotted ALOHA

Die Zeit wird in Intervalle eingeteilt, die so genannten **Zeitschlitze** (Slots). Die Intervallgröße entspricht einem Rahmen. Die Übertragung erfolgt nur zu Beginn eines Zeitschlitzes. Kollisionen sind immer noch möglich.



Datendurchsatz bei ALOHA-Systemen



Maximaler Durchsatz von ALOHA und Slotted ALOHA

Maximaler Durchsatz von reinem ALOHA:

$$\frac{1}{2e} \approx 0{,}18$$ Pakete pro Zeitschlitz

Maximaler Durchsatz von "slotted ALOHA":

$$\frac{1}{e} \approx 0{,}36$$ Pakete pro Zeitschlitz

4.4 CSMA/CD (Ethernet)

Medienzugangsprotokoll CSMA

Voraussetzungen

- Stationen können sich gegenseitig „hören“.
- Die Rahmen-Übertragungszeit muss sehr viel größer sein als die Laufzeitverzögerung zwischen den Stationen.

Verfahren

Carrier **S**ensing, **M**ultiple **A**ccess (**CSMA**)

(auch "listen before talk" genannt)

Die sendewillige Station hört das Medium ab:

- Falls belegt, wird das Senden zurückgestellt
- Falls frei, wird sofort mit dem Senden begonnen

Kollision

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

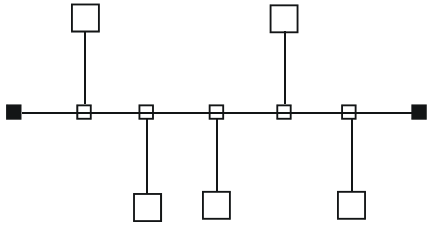
Erhöhte Kollisionsgefahr besteht nach Abschluss einer Übertragung: mehrere sendewillige wartende Stationen können bei der Feststellung „Medium frei“ gleichzeitig zu senden beginnen. Daher wird ein besonderer Algorithmus für das Sendebeginn-Verhalten nach Abschluss einer laufenden Sendung definiert (d.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

CSMA/CD: Topologie

CSMA/CD = **C**arrier **S**ense **M**ultiple **A**ccess **with Collision Detection**

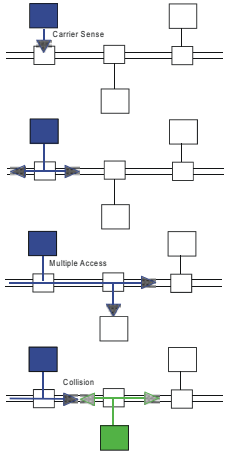
Standard: IEEE 802.3 und ISO IS 8802/3: MAC und Bitübertragungsschicht für CSMA/CD

Topologie: Bus



- Bidirektionaler Datenfluss
- Busunterbrechung = Systemausfall

CSMA/CD: Protokoll

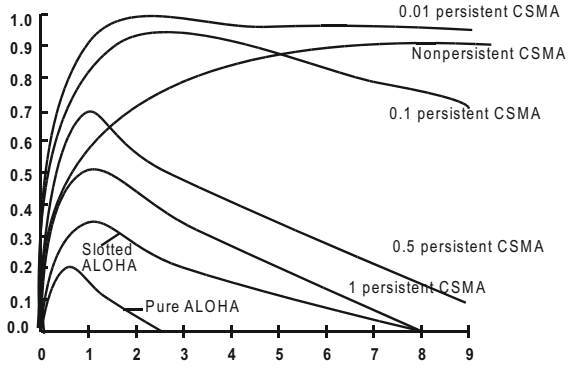


CSMA/CD: Wiederholungsstrategien bei belegtem Medium

- **non-persistent**
  Die Station wartet eine zufällig berechnete Zeitspanne ("backoff time") und startet dann einen neuen Übertragungsversuch.
- **1-persistent**
  Die Station hört das Medium ab und startet die eigene Übertragung sofort nach Abschluss der laufenden Übertragung (Sendewahrscheinlichkeit = 1)
- **p-persistent** ($0 < p < 1$)
  Die 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.

IEEE/ISO CSMA/CD ist 1-persistent.

Ausnutzung der Kanalbandbreite bei verschiedenen Wiederholungsstrategien



CSMA/CD – Rahmenformat

Präambel	SD	DA	SA	INFO	FCS

Präambel = 7 Bytes
SD = Starting Delimiter (1 Byte)
DA = Destination Address (2 oder 6 Bytes)
SA = Source Address (2 oder 6 Bytes)
Info = n Bytes
FCS = Frame Check Sequence

CSMA/CD – Adressfeldformat

Adressformate nach IEEE 802

48 -Bit - Format

I/G	U/L	46 Bits Adresse

16 - Bit - Format

I/G	15 Bits Adresse

I/G = 0 individual address
I/G = 1 group address
U/L = 0 globally administered address
U/L = 1 locally administered address

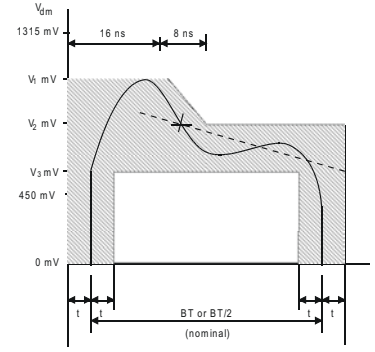
CSMA/CD – Bitkodierung

Für die bitserielle Datenübertragung auf dem Medium wird die **Manchesterkodierung** ("Manchester Encoding") benutzt.

Die Manchesterkodierung ist ein binärer Leistungscode, **der Bitwert und Zeittakt in "Bitsymbolen" kombiniert**. Jedes Bitsymbol ist in zwei Hälften geteilt. Eine ansteigende Flanke in der Mitte des Intervalls codiert eine “0“, eine abfallende Flanke in der Mitte des Intervalls codiert eine “1“.

CSMA/CD – Bitübertragungsschicht

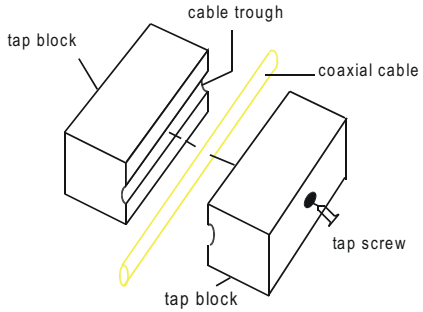
Toleranzgrenzen in der Bitübertragungsschicht



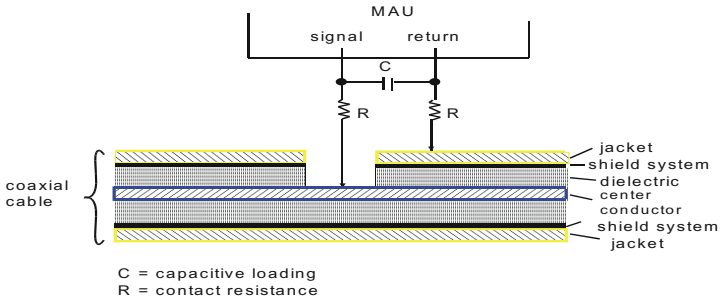
t = 2.5 ns AT 1-10 MHz data rates
V_2 = 0.89 V_1
V_3 = 0.82 V_2

CSMA/CD – historische Kabeltechnik (1)

Frühe Technik: spezielles Koaxialkabel in Bus-Topologie im Kabelkanal.
Nachträglicher Ausschluss am „gelben Kabel“ in jedem Büro möglich.



CSMA/CD – historische Kabeltechnik (2)

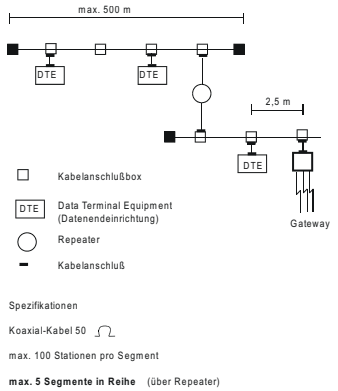


Typical Coaxial Tap Connection Circuit

Vorteil: Man kann überall das Koaxialkabel anzapfen, auch nach der Installation.

CSMA/CD – Parameter des ersten Standards

Basisband-Bussystem mit 10 MBit/s



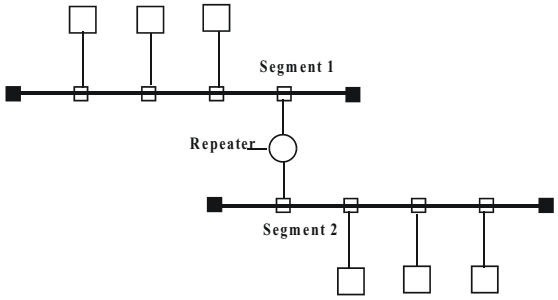
Spezifikationen

Koaxial-Kabel 50 Ω

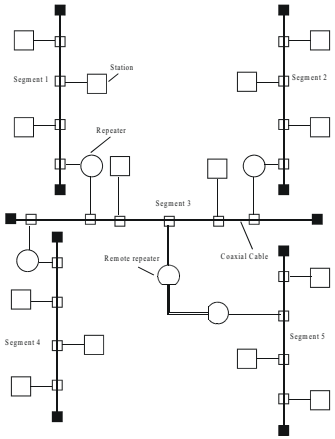
max. 100 Stationen pro Segment

max. 5 Segmente in Reihe (über Repeater)

Beispiel 1: Mittelgroße Konfiguration



Beispiel 2: Große Konfiguration



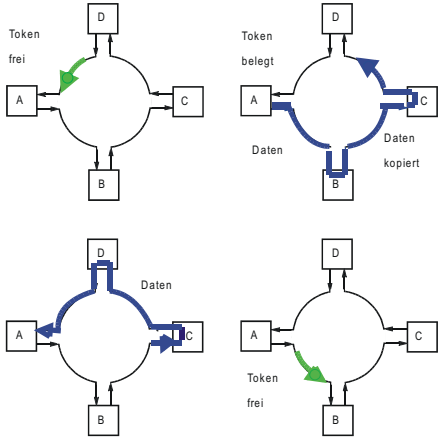
4.5 Token Ring

Standards

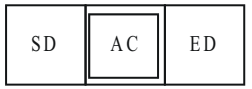
- IEEE 802.5
- ISO IS 8802/5

Hauptsächlich entwickelt im Hause IBM, ca. 1984 - 1992.

Medienzugriffsprotokoll des Token-Rings

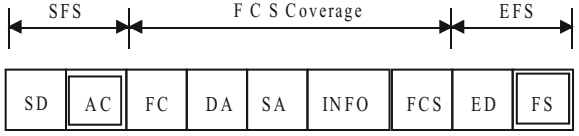


Token Ring: Format des Tokens



SD = Starting Delimiter (1 Byte)
AC = Access Control (1 Byte)
ED = Ending Delimiter (1 Byte)

Token Ring: Format des Datenrahmens

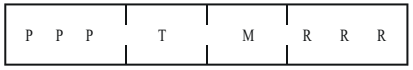


SFS = Start-of-Frame Sequence
SD = Starting Delimiter (1 Byte)
AC = Access Control (1 Byte)
FC = Frame Control (1 Byte)
DA = Destination Address (2 or 6 Bytes)
SA = Source Address (2 or 6 Bytes)

INFO = Information (0 or more Bytes)
FCS = Frame-Check Sequence (4 Bytes)
EFS = End-of-Frame Sequence
ED = Ending Delimiter (1 Byte)
FS = Frame Status (Byte)

Token Ring: AC- und FS-Felder

Access Control (AC)

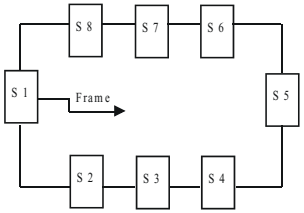


PPP = priority bits
T = token bit
M = monitor bit
RRR = reservation bits

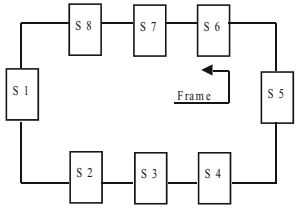
Frame Status (FS)

A = address-recognized bits
C = frame-copied bits
r = reserved bits

Token Ring: Prioritätsmechanismus (1)

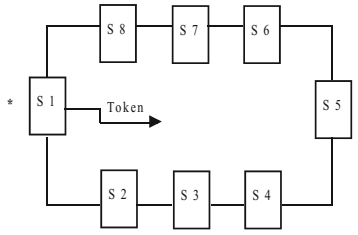


1. Station S1 erhält das Token und überträgt ihren Rahmen mit normaler Priorität.

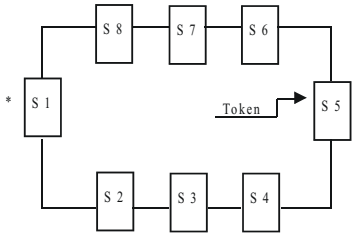


2. S5 reserviert eine höhere Priorität im vorbei laufenden Rahmen (RRR-Bits).

Token Ring: Prioritätsmechanismus (2)

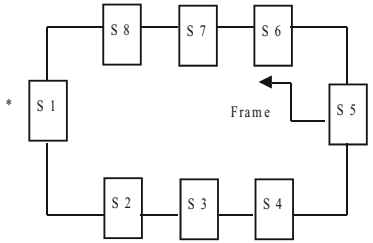


3. S1 entfernt ihren Rahmen nach erfolgter Übertragung, erzeugt ein Token mit der von S5 reservierten Priorität und geht in den Zustand "priority-hold" über.

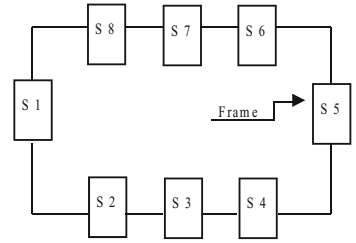


4. S2, S3 und S4 haben keine Prioritätsrahmen, und das Token läuft weiter zu S5.

Token Ring: Prioritätsmechanismus (3)

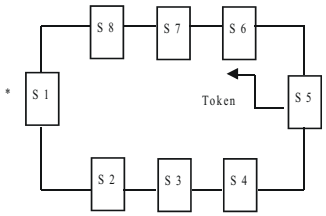


5. Station S5 überträgt ihren Prioritäts-Rahmen.

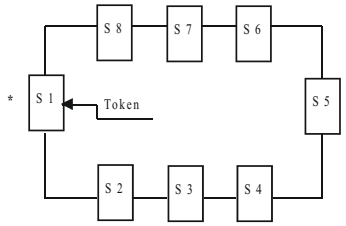


6. Der Rahmen kommt zu S5 zurück.

Token Ring: Prioritätsmechanismus (4)

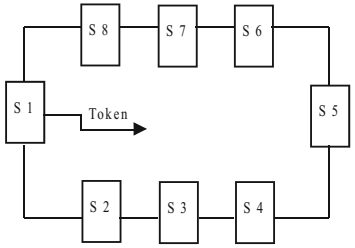


7. S5 hat ihre Übertragung beendet und erzeugt ein Token mit der Priorität, die sie gerade benutzt hat (die höhere Priorität). S1, immer noch in "priority-hold", wartet auf ein Frei-Token mit dieser Priorität (der Priorität, die S5 angefordert und S1 generiert hat).



8. S1 empfängt das Frei-Token von S5 und erkennt die von ihr selbst erzeugte Priorität.

Token Ring: Prioritätsmechanismus (5)



9. S1 verlässt den Zustand "priority-hold" (vorausgesetzt, dass keine neue Prioritätsreservierung vorliegt) und erzeugt ein Token mit normaler Priorität. Falls S2 auf ein Token mit normaler Priorität wartet, kann sie jetzt mit der Übertragung beginnen.

Funktionen zur Fehlerkorrektur

- Genau ein aktiver Monitor pro Ring zur effizienten Fehlerkorrektur.
- In jeder anderen Station ist ein Monitor in Bereitschaft, um größere Zuverlässigkeit und Verfügbarkeit zu erreichen.
- Die Fehlerkorrekturfunktionen benutzen Verwaltungsrahmen ("management frames"):
  - Claim Token
  - Duplicate Address Test
  - Active Monitor Present
  - Standby Monitor Present
  - Beacon
  - Purge

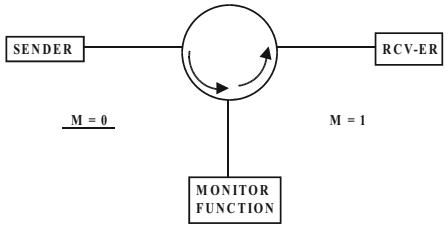
Aktiver Monitor

Jede Station im Ring kann die Rolle des aktiven Monitors spielen. Ein Auswahlverfahren stellt sicher, dass es zu jedem Zeitpunkt nur genau einen aktiven Monitor gibt.

Der aktive Monitor schützt vor den folgenden Fehlerbedingungen:

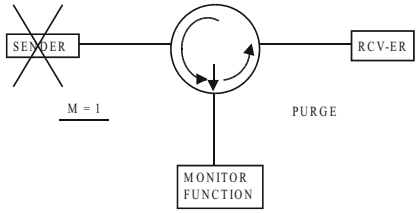
- zirkulierender Rahmen
- zirkulierendes Token mit hoher Priorität
- verloren gegangenes Token
- mehrere aktive Monitore.

Zirkulierender Rahmen (1)



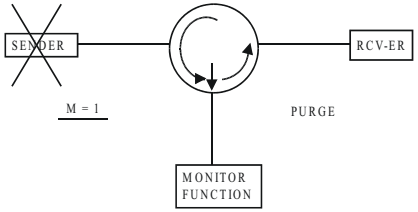
- Der Sender erzeugt einen Rahmen mit dem Monitor-Bit M = 0.
- Der Sender fällt aus.
- Der aktive Monitor setzt beim Durchlauf des Rahmens das Monitorbit M = 1

Zirkulierender Rahmen (2)



Wenn der aktive Monitor einen Rahmen mit M = 1 sieht, löscht er alles auf dem Ring und initialisiert neu.

Zirkulierender Rahmen (3)



Nachdem der Ring neu initialisiert ist, erzeugt der aktive Monitor ein neues Token.

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") beim Durchlauf regeneriert wird.
- Falls der Timer abläuft, bevor erneut ein Anfangsbegrenzer erkannt wird, löscht der aktive Monitor alles auf dem Ring und erzeugt ein neues Token.

Mehrere aktive Monitore

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

- Purge Frame oder
- Active Monitor Present Frame

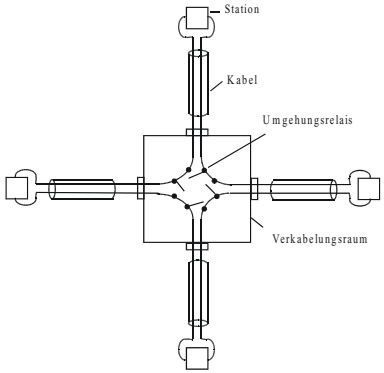
empfängt, den er nicht selbst erzeugt hat.

Token Ring: Bitkodierung

Der Token Ring verwendet in der Schicht 1 das Differential Manchester Encoding:

- 0-Bit wechselt die Spannung am Anfang des Bitintervalls.
- 1-Bit behält den vorherigen Pegel am Anfang des Bitintervalls bei.

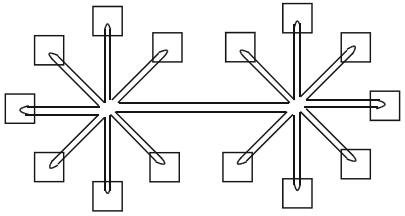
Sterntopologie für den Token Ring



Im zentralen Schaltschrank befindet sich das Relais, das bei Stationsausfall das Kabel zur defekten Station kurzschließt.

Strukturierte Verkabelung

Möglichkeiten einer Ringverkabelung



Logischer Ring **Physischer Stern**

Die Verkabelung in Form eines physischen Sterns erleichtert die Fehlersuche und ermöglicht eine flexiblere Nachverkabelung von neuen Stationen.

Kabel für den Token Ring

Zwei verdrillte Kupferdoppeladern (Kupferdraht) mit Abschirmung

Kabel mit zwei optischen Leitern

Zusammenfassung Token Ring LAN

- Neugenerierung der Rechteckimpulse in jeder Station. Dadurch wenig rauschempfindlich. Große Ringe mit vielen Stationen möglich.
- Natürliche Anwendung für Glasfaser, da die Signale nur an den Kabelenden eingespeist bzw. entnommen werden
- Fehlerhafte oder ausgefallene Stationen müssen isoliert und aus dem Ring ausgeschlossen werden, insbesondere bei sternförmiger Verkabelung.

4.6 FDDI (Fiber Distributed Dataa Interface)

Motivation

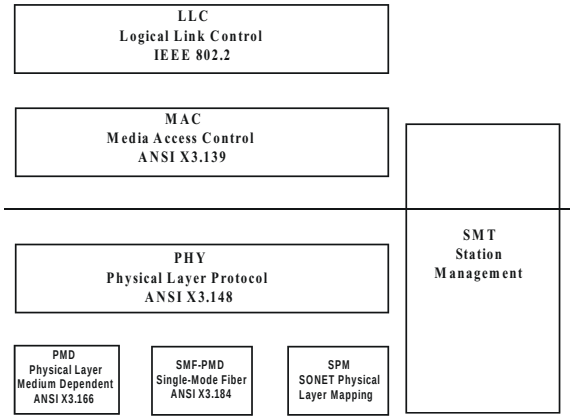
- FDDI macht als erste Netztechnik den Sprung von der 4-16 MBit/s-Geschwindigkeitsklasse (mittelschnelle LANs) auf die 100 MBit/s-Klasse (schnelle LANs, Metropolitan Area Networks (MANs))
- größere räumliche Ausdehnung (100 km)
- große Anzahl an Stationen (500)

FDDI: Eigenschaften

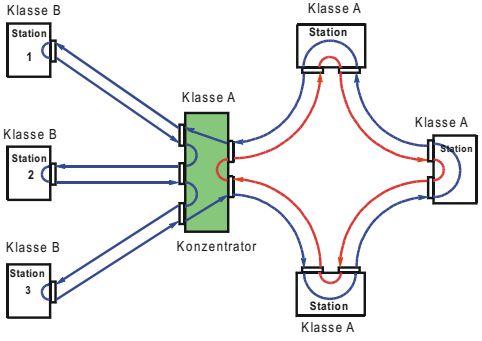
Eigenschaften

- Topologie: Doppelter Ring
- Gradientenfaser oder Monomode-Faser
- Leuchtdioden, Wellenlänge 1300 nm
- Stecker mit zwei Glasfasern
- maximale Länge der Faser: 200 km
- maximaler Umfang des Doppelrings: 100 km
- maximale Anzahl der Stationen: 500
- Medium Access Control: Token mit "**Early Token Release**" und Zeitlimit
- Datenrate: 100 MBit/s
- Standard bei ANSI und ISO

FDDI: Die einzelnen Standards



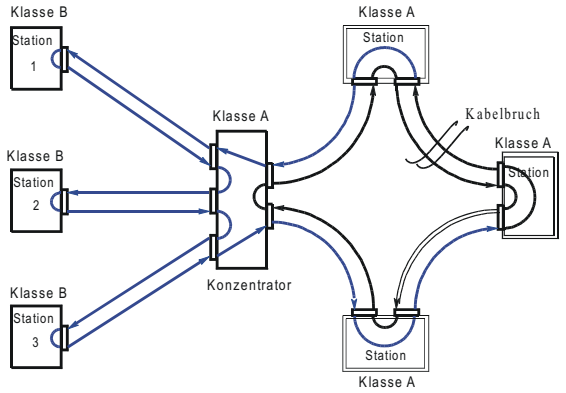
FDDI - Topologie



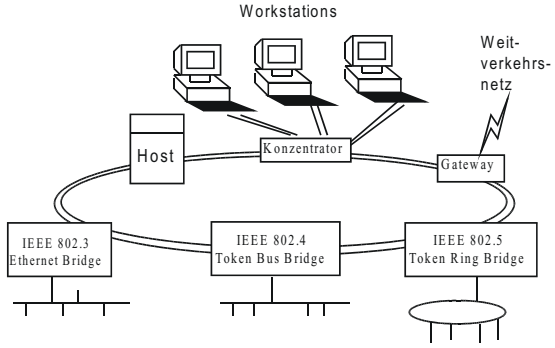
"Single Attachment Stations"

"Dual Attachment Stations"

Fehlertoleranz durch Doppelring

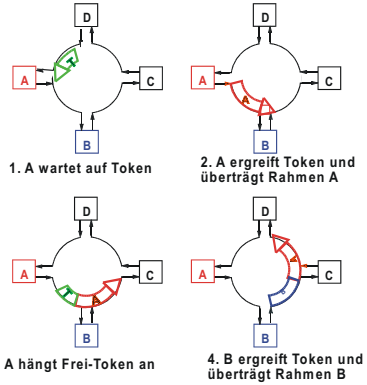


Typische FDDI-Konfiguration

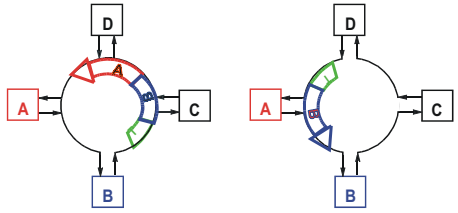


Zugangsprotokoll zum Medium (1)

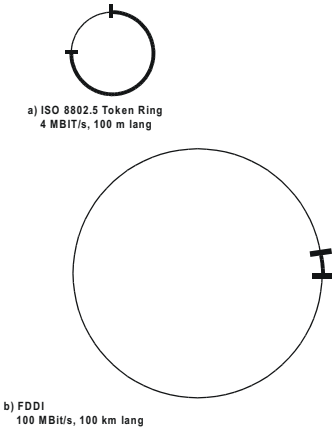
Token Passing mit "Early Token Release"



Zugangsprotokoll zum Medium (2)

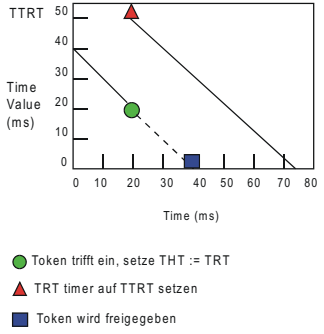


Motivation für "Early Token Release"



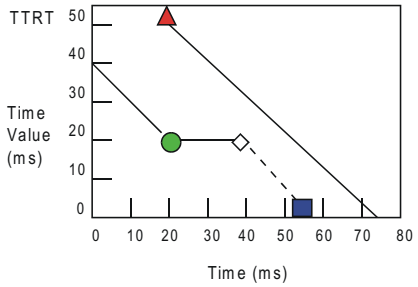
Netzzugangsprotokoll: Token mit Zeitlimit

- Bei der Initialisierung des Rings wird eine maximale Rotationszeit für das Token vereinbart (TTRT = Target Token Rotation Time)
- Bei Eintreffen des Tokens darf eine Station senden, bis TTRT erreicht ist

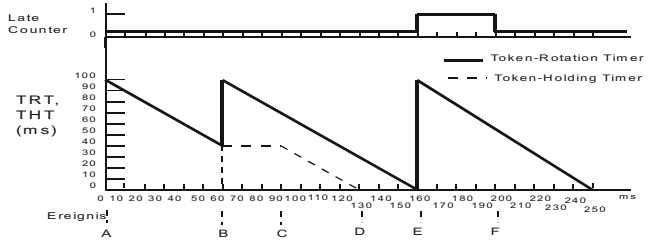


Vorab-Zuordnung von synchroner Bandbreite

- Garantiert eine Mindestsendezeit pro Rotation des Tokens und damit eine Mindestbandbreite pro Station
- Aber: die Rotationszeit variiert je nach Sendevolumen anderer Stationen => synchrone, aber nicht isochrone Übertragung



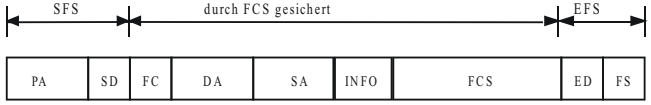
Netzzugangsprotokoll: Token mit Zeitlimit (2)



A. Token trifft ein - "Aufziehen" des TRT-Timers
B. Token wird festgehalten - synchrone Übertragung beginnt
C. asynchrone Übertragung beginnt
D. Ablauf der Zeitschranke, Token wird generiert
E. TRT abgelaufen - "late counter" gesetzt
F. Token trifft ein. Asynchrones Senden verboten. "late counter" wird gelöscht, TR Timer akkumuliert die Verspätung

Im Beispiel: TTRT = 100 ms, Reservierung für synchrone Zeit: 20 ms

FDDI: Rahmenformat



SFS = Start-of-Frame Sequenz
PA = Präambel (8 oder mehr Bytes)
SD = Starting Delimiter (1 Byte)
FC = Frame Control (1 Byte)
DA = Destination Address (2 oder 6 Bytes)
SA = Source Address (2 oder 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)

FDDI: Bitcodierung

4B/5B-Symbolcodierung

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

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

- 1-Bit: Pegelwechsel; 0-Bit: kein Pegelwechsel
- Diese Bitcodierung in Verbindung mit der 4B/5B-Symbolcodierung stellt sicher, dass im Impulsstrom maximal drei Bitzeiten ohne Pegelwechsel vorkommen!
- Deshalb bleibt die Synchronisation gewährleistet, wenn die Uhren des Senders und des Empfängers mindestens drei Taktzeiten lang hinreichend gleich laufen!

4B/5B Symboltabelle (1)

DECIMAL	CODE GROUP	SYMBOL	NAME	ASSIGNMENT
00	00000	Q	QUIET	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		0001
20	10100	2		0010
21	10101	3		0011
10	01010	4		0100
11	01011	5		0101
14	01110	6		0110
15	01111	7		0111
18	10010	8		1000
19	10011	9		1001

4B/5B Symboltabelle (2)

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	VIOLA-TION	NOT TRANSMITTED
02	00010	V	"	NOT TRANSMITTED
03	00011	V	"	NOT TRANSMITTED
06	00110	V	"	NOT TRANSMITTED
08	01000	V	"	NOT TRANSMITTED
12	01100	V	"	NOT TRANSMITTED
16	10000	V	"	NOT TRANSMITTED

Glasfaserkabel für FDDI

Multimode-Faser (Gradientenfaser)

- 2 Kilometer maximale Länge zwischen zwei Stationen
- 62,5 µ Multimode-Faser
- LED-Sender (1300 nm Wellenlänge)
- 11 dB Dämpfung pro Link

Monomode-Faser

- 50 Kilometer maximale Länge zwischen zwei Stationen
- 9 µ Monomode-Faser
- Laser-Sender (1300 nm Wellenlänge)
- 32 dB Dämpfung pro Link

SONET (Synchronous Optical Network)

- Verwendung des öffentlichen Trägernetzes SONET statt einer privaten Glasfaserstrecke
- Ringumfang maximal 100 km muss weiter gelten wegen der Timer

Zusammenfassung der LAN-Charakteristika

CSMA/CD

- Sehr gute Performance bei geringer bis mäßiger Belastung
- Einfaches Protokoll, leicht implementierbar
- Topologische Einschränkungen bei der Bus-Topologie
- Keine Prioritätssteuerung
- Keine maximale Verzögerung garantierbar

Token Ring

- Hohe Systemausnutzung
- Fehlerbehebung ("Token Recovery") aufwendig
- Flexible Verkabelung
- Prioritäten möglich
- Maximale Verzögerung garantierbar (außer beim Auftreten von Fehlern)

FDDI

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

4.7 Sternkoppler und LAN-Switching

Sternkoppler

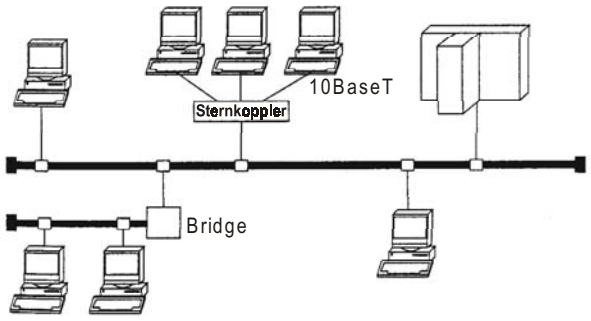
Das Ethernet-LAN hat sich in den letzten Jahren gegenüber dem Token Ring durchgesetzt.

In den frühen Jahren wurden die Ethernet-Kabel tatsächlich busförmig verlegt (gelbes Kabel oder Thin Ethernet mit handelsüblichen, vorkonfektionierten Koaxial-Kabeln), was sich aber aus Gründen der Fehlerlokalisierung, Nachverkabelung von neuen Stationen usw. als nachteilig erwiesen hat.

Heute sind **Sternkoppler (Hubs)** mit sternförmiger Verkabelung üblich, auch in Kabelkanälen („Cat5 cable“).

Das Medienzugangsprotokoll ist nach wie vor CSMA/CD! Allerdings finden das Carrier Sensing und die Kollisionserkennung jetzt im Sternkoppler (Hub) statt.

Ethernet-LAN mit Sternkoppler



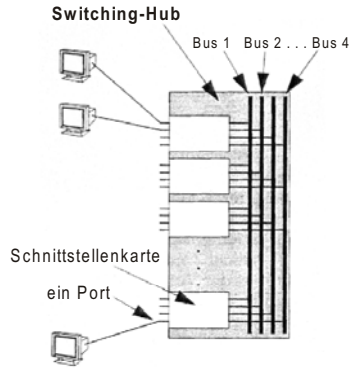
LAN-Switching

Durch **LAN-Switching** kann man den Durchsatz in einem Ethernet-LAN weiter erhöhen. Dabei wird der Sternkoppler durch einen Rahmen-vermittlungsknoten ersetzt, der die einzelnen Rahmen zwischen den einlaufenden Kabeln direkt durchschaltet. **Das Format der Ethernet-Rahmen bleibt vollständig erhalten**, deshalb können auch alle Endgeräte unverändert weiter betrieben werden.

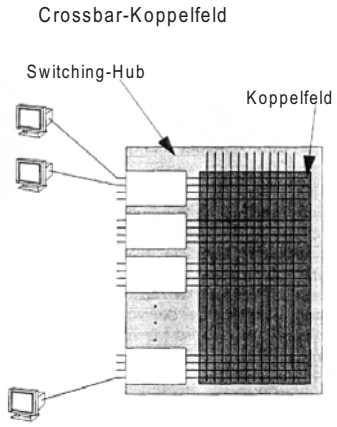
Im Sternpunkt wird das CSMA/CD-Protokoll durch eine Frame-Switch ersetzt, der die MAC-Zieladresse (destination address) auswertet und den Rahmen entsprechend weiter leitet. Anders als beim Sternkoppler muss der LAN-Switch intern mit einem Mehrfachen der Leitungsgeschwindigkeit operieren, so dass mehrere eintreffende Rahmen ohne Kollision weiter geleitet werden können.

LAN-Switch: Implementierung mit Bussen

mehrere Busse als Koppelfeld



LAN-Switch: Implementierung mit einer Kreuzschiene



Kabeltypen für das Ethernet

Ethernet-Typ	Medium	Max. Länge	Bemerkungen
10Base5 (gelbes Koaxkabel)	Koaxialkabel	500 m	Die klassische Ethernet-Verkabelung. Der Anschluss erfolgt über Transceiver und Vampir-Klemmen am Koaxkabel
10Base2 (Thinnet oder Cheaper-Net)	Koaxialkabel	185 m	Verwendet wird ein dünnes Koaxialkabel. Der Anschluss erfolgt über Transceiver und BNC-Stecker.
10BaseT	Verdrilltes Kupferkabel (geschirmt – STP, oder ungeschirmt UTP)	100 m	Die Stationen werden in einer Sternkonfigruration an sog. Sternkoppler (engl. „Hub“ genannt) herangeführt.
10BaseFB	Lichtwellenleiter	2 km	Dieser Typ wird meist für Ethernet-Backbone-Netze zwischen Sternkopplern verwendet.
10BaseFL	Lichtwellenleiter	2 km	Dieser Typ wird hauptsächlich zwischen Regeneratoren eingesetzt

4.8 Logical Link Control im LAN

Die Sicherungsschicht in den LANs

Identisch für CSMA/CD, Token Ring, Token Bus, FDDI!

LLC Typ 1: Unbestätigter verbindungsloser Dienst

- Unbestätigte Übertragung von Datenrahmen
- Die höheren Schichten sind für die Erhaltung der Reihenfolge, Fehlerbehebung und Flusssteuerung verantwortlich

LLC Typ 2: Verbindungsorientierter Dienst (wie HDLC)

- Verbindungsaufbau und -abbau
- Datenübertragung mit Bestätigung
- Garantierte Ablieferung beim Empfänger
- Garantierte Reihenfolge der Rahmen
- Flusssteuerung

Logical Link Control

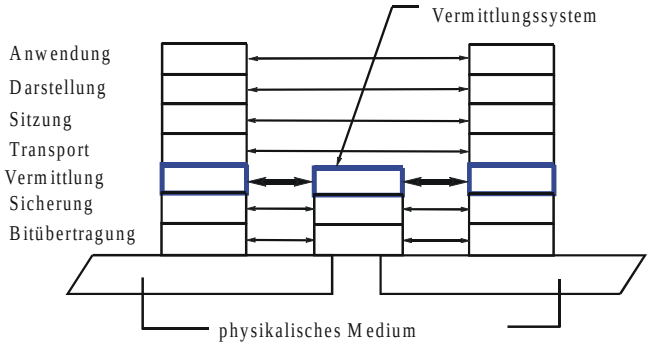
LLC Typ 3: Bestätigter verbindungsloser Dienst

- Auf jedes Datagramm kann genau eine Bestätigung folgen
- Anwendung beispielsweise in der Automatisierungstechnik

5 Weitverkehrsnetze und Routing

5.1 Das Prinzip der Paketvermittlung

Die Vermittlungsschicht im OSI-Referenzmodell



ISO-Definition für die Vermittlungsschicht

Die Vermittlungsschicht stellt die Fähigkeit bereit, Netzverbindungen zwischen offenen Systemen aufzubauen, zu betreiben und abzubauen.

Die Vermittlungsschicht bietet den Transportinstanzen Unabhängigkeit von Wegewahl- und Vermittlungsentscheidungen, die mit dem Aufbau und Betrieb einer Netzverbindung verbunden sind.

Aufgaben der Vermittlungsschicht

- Wegewahl und Vermittlung von Paketen
- Multiplexen von Ende-zu-Ende-Verbindungen über Schicht-2-Verbindungen
- Segmentierung („Fragmentierung“)
- Zusätzlich in verbindungsorientierten Vermittlungsschichten:
  - Verbindungsaufbau und -abbau
  - Fehlererkennung und Fehlerbehebung (Ende-zu-Ende)
  - Sicherstellung der Paketreihenfolge
  - Flusskontrolle (Ende-zu-Ende)

Dabei ist es wichtig, dass heterogene Teilnetze verbunden werden können („Internetworking“).

5.2 Virtuelle Verbindung vs. Datagramm

Virtuelle Verbindung

Der Weg durch das Netz wird beim Aufbau der virtuellen Verbindung ausgewählt, d.h. für jede neue virtuelle Verbindung findet in jedem Netzknoten nur einmal eine Wegewahlentscheidung statt. Der gesamte über diese virtuelle Verbindung fließende Verkehr nimmt denselben Weg durch das Netz.

Datagramm

Jedes Paket enthält die volle Adresse des Ziel-Hosts. Die Zieladresse bestimmt in jedem Netzknoten auf dem Pfad stets neu die ausgehende Leitung.

Die Virtuelle Verbindung

„Perfekter" Kanal durch das Netz

- Ordnung der Nachrichten (Sicherstellung der Reihenfolge)
- Fehlerüberwachung (verlorene und duplizierte Pakete)
- Flusskontrolle

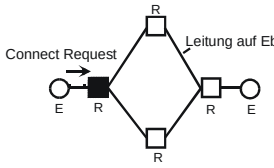
Phasen

- Verbindungsaufbau
- Datenübertragung
- Verbindungsabbau

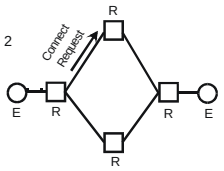
Vorteile

- Niedriger Mehraufwand für die Adressierung während der Datenübertragung
- Hohe Qualität des ankommenden Paketstroms: keine Neusortierung oder Fehlerüberwachung in den höheren Schichten nötig

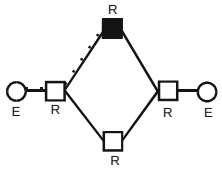
Aufbau einer virtuellen Verbindung



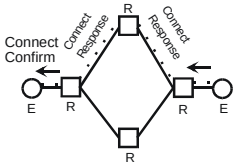
a) Festlegen des Weges



b) Aufbauphase der 1. Teilstrecke



c) Virtueller Verbindungsabschnitt existiert, Festlegung der Wegefortsetzung

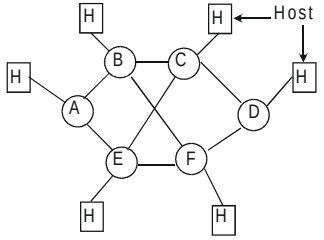


d) nach weiteren Schritten virtuelle Verbindung fertigestellt

Implementierung von virtuellen Verbindungen

In jedem Netzknoten werden Tabellen mit Zustandsinformationen über bestehende virtuelle Verbindungen verwaltet.

(a) Beispiel-Subnetz:

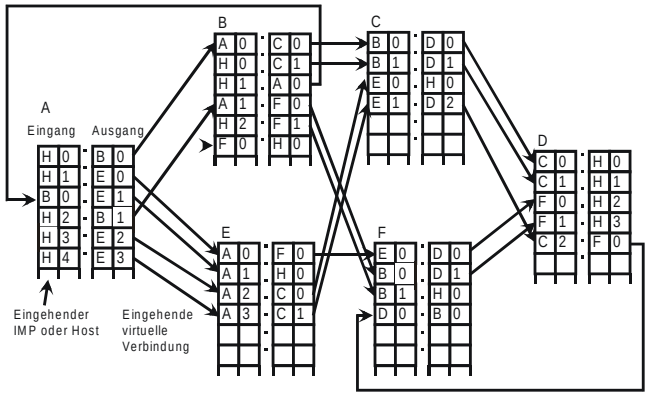


(b) Acht virtuelle Verbindungen durch dieses Subnetz:

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

Zustandsinformation in den Netzknoten

(c) Router-Tabellen für die virtuellen Verbindungen in (b)



Das Datagramm

Jedes Paket (Datagramm) wird als isolierte Einheit betrachtet (wie ein Telegramm im Postverkehr):

- Volle Zieladresse in jedem Paket
- Pakete können außerhalb der Reihenfolge eintreffen
- Keine Fehlerüberwachung, keine Flusskontrolle in Schicht 3

Vorteile

- Primitiver als virtuelle Verbindungen, daher viel einfacher zu implementieren
- Kein Verbindungsaufbau und -abbau, deshalb geringer Overhead für kurzlebige Verbindungen
- flexibler und zuverlässiger
- besser geeignet für Internetworking heterogener Subnetze

5.3 Wegewahl für Punkt-zu-Punkt-Netze

Vorbemerkung: Besondere Netztopologien

Wegfall des Wegewahlproblems auf Broadcast-Medien, z. B. in einem Segment eines LANs (Bus- oder Ring-Topologie): hier ist keine Wegewahl erforderlich, da jede Nachricht wegen der Topologie des physikalischen Mediums alle Empfänger erreicht.

Routing-Algorithmen

Aufgabe

Leitwegbestimmung für Pakete durch das Netzwerk vom Quellsystem zum Zielsystem

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

- Bei der Datagrammtechnik: individuelle Entscheidung für jedes Paket
- Bei Virtuellen Verbindungen: Leitwegbestimmung nur beim Verbindungsaufbau.

Wünschenswerte Eigenschaften eines Routing-Algorithmus

- Korrekt
- Einfach
- Robust bei Rechner- oder Leitungsausfällen
- Fair
- Optimal

Algorithmen für die Leitwegbestimmung

Die genannten Kriterien stehen im Zielkonflikt. In der Praxis hat sich als Ziel bewährt:

Minimierung der Teilstrecken (hops) vom Sender zum Empfänger.

Leitwegbestimmung

Klassifikation der Verfahren

1. Statische (nicht-adaptive) Verfahren

- keine Berücksichtigung des aktuellen Netzzustands
- gehen von Mittelwerten aus
- 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 dem aktuellen Netzzustand
- Laufend Messungen/Schätzungen der Topologie und des Verkehrsaufkommens
- Weitere Unterteilung der adaptiven Verfahren in
  - zentralisierte Verfahren
  - isolierte Verfahren
  - verteilte Verfahren

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-Tabellen für die einzelnen Knoten und versendet diese.

Die statische Leitwegbestimmung ist sinnvoll, wenn das Netz relativ klein und relativ statisch ist.

Mehrfach-Leitwegbestimmung (multipath routing)

Benutzung alternativer Leitwege zwischen jedem Knotenpaar (i,j)

- Häufigkeit der Nutzung abhängig von der Güte der Alternative
- Höherer Durchsatz durch Verteilung des Datenverkehrs auf mehrere Pfade
- Höhere Zuverlässigkeit, da der Ausfall eines Links nicht so schnell zur Unerreichbarkeit von Knoten führt

Realisierung (1)

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

Z	A1	G1	A2	G2		An	Gn

Z Ziel

A_i i-beste Ausgangsleitung

G_i Gewicht für A_i

(G_i bestimmt die Wahrscheinlichkeit, mit der A_i benutzt wird)

$$\left(\sum_{i=1}^{n} G_i = 1\right)$$

Realisierung (2)

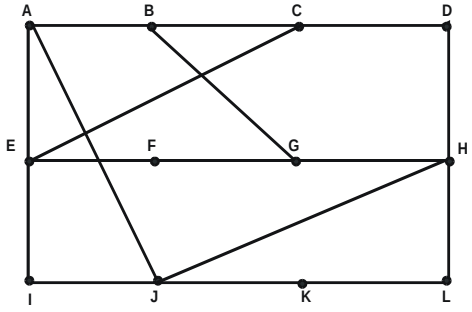
Auswahl der Alternativen:

```
Generiere eine Zufallszahl z (0 <= z <= 1)
Wähle A1, falls 0  <= z <= G1
Wähle A2, falls G1 <= z <  G1 + G2
...
Wähle An, falls G1 + G2 + .. + G(n-1) <= z < 1
```

Beispiel : Ziel B, Quelle J

Statische Leitwegbestimmung: Beispiel (1)

Topologie des Beispielnetzes



Wir betrachten die Pfade vom Knoten J aus.

Statische Leitwegbestimmung: Beispiel (2)

Statische Routing-Tabelle mit alternativen Pfaden

Ziel	Erste	Wahl	Zweite	Wahl	Dritte	Wahl
A	A	0.63	I	0.21	H	0.16
B	A	0.46	H	0.31	I	0.23
C	A	0.34	I	0.33	H	0.33
D	H	0.50	A	0.25	I	0.25
E	A	0.40	I	0.40	H	0.20
F	A	0.34	H	0.33	I	0.33
G	H	0.46	A	0.31	K	0.23
H	H	0.63	K	0.21	A	0.16
I	I	0.65	A	0.22	H	0.13
-						
K	K	0.67	H	0.22	A	0.11
L	K	0.42	H	0.42	A	0.16

Bestimmung der Leitwegtabellen

Die Routing-Tabellen werden beim statischen Routing vom Netzwerkoperator zentral erstellt. Sie werden vor Inbetriebnahme des Netzes in die Knoten geladen und dann nicht mehr verändert.

Eigenschaften

- einfach
- gute Ergebnisse bei relativ konstanter Topologie und konstantem Verkehr

Aber:

- schlecht bei stark variierendem Verkehrsaufkommen und bei Topologieänderungen
- schlecht bei großen Netzen (skaliert nicht).

In der Praxis noch immer gelegentlich benutzt, zum Beispiel in SNA-Netzen.

Der Netzoperator kennt stets die gesamte Topologie. verwendet beispielsweise den Algorithmus „kürzeste Wege“ von Dijkstra einmal für jeden Knoten zur Konstruktion der Routing-Tabellen.

Zentralisierte adaptive Leitwegbestimmung (1)

Prinzip

- Im Netz gibt es ein **Routing Control Center (RCC,** (Leitwegsteuerzentrum)
- Jeder Knoten sendet periodisch Status-Information zum RCC, beispielsweise
  - die Liste der verfügbaren Nachbarn
  - aktuelle Warteschlangenlängen
  - Auslastung der Leitungen, etc.
- Das RCC sammelt die Informationen und berechnet den optimalen Pfad für jedes Knotenpaar, berechnet die einzelnen Leitwegtabellen und verteilt diese dann an die Knoten.

Zentralisierte adaptive Leitwegbestimmung (2)

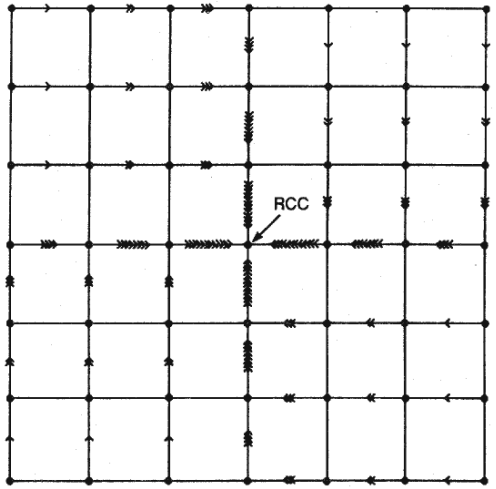
Eigenschaften

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

Aber:

- Die Berechnung muss oft durchgeführt werden (ca. jede Minute oder öfter)
- Es entsteht eine Verkehrskonzentration in der Nähe des RCC ("performance bottleneck")
- Die Technik ist nicht robust: das RCC ist ein “single point of failure“.
- Das Verfahren funktioniert nicht bei Netzpartitionierung.
- Die einzelnen Knoten erhalten ihre neuen Tabellen jeweils zu unterschiedlichen Zeiten => Inkonsistenzen und damit “routing loops“sind möglich.

Das Routing Control Center



Isolierte adaptive Leitwegbestimmung

Prinzip

- Kein Austausch von Routing-Information zwischen Knoten
- Entscheidungen basieren ausschließlich auf lokal verfügbaren Informationen

Beispiele für Verfahren

- Backward Learning (Baran)
- Flooding
- Delta-Routing (Rudin, 1976)

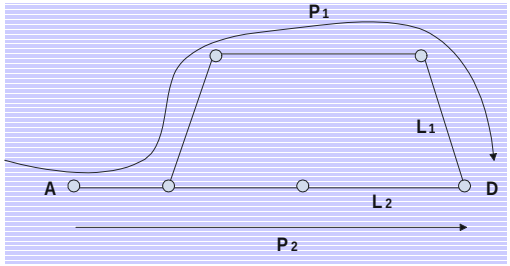
Algorithmus "Backward Learning"

- Knoten "lernt" von eintreffenden Paketen
  Paket (..., Q, Z, ...)
  Q = Quell–Knoten
  Z = Teilstreckenzähler (hop counter)

  Paket wird auf Leitung L empfangen => Q ist über L in Z Teilstrecken erreichbar
- Leitwegtabelle im Knoten: Jeder Eintrag ist ein Tripel
  (Zielknoten, Ausgangsleitung, Z_{min})
- Aktualisierung der Leitwegtabelle:
  Knoten empfängt Paket (..., Q, Z, ...) auf Leitung L

```
if not (Q in Tabelle)
    then  add(Q,L,Z)
    else  if Z < Zmin
          then update(Q,L,Z)
```


Backward Learning: Beispiel




```
P1( ..., A,4,...) -> add(A,L1,4)
P2( ..., A,3,...) -> update(A,L2,3)
```


Pfadverschlechterung beim Backward Learning

Problem

Algorithmus registriert keine Verschlechterungen

Lösung

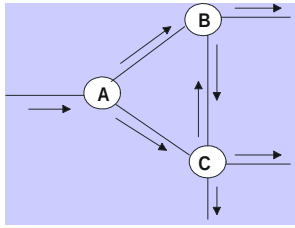
periodisches Löschen der Leitwegtabellen (neue Lernperiode)

Aber: Löschzeitpunkte kritisch:

- zu häufig: Netz ist überwiegend in der Lernphase
- zu selten: zu langsame Reaktion auf Verschlechterungen

Algorithmus „Flooding“

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



Flooding: Abklingen des Paketflusses

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 jeder Teilstrecke um 1 dekrementiert
- Duplikate erhalten den Streckenzähler des Originals
- Zähler = 0: Paket wird vom Router weggeworfen

Eigenschaften von Flooding

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

Algorithmus "Delta-Routing" (1)

Prinzip

Kombination von isoliertem und zentralisiertem Verfahren.

- Jeder Knoten berechnet periodisch die "Kosten" seiner Leitungen und sendet diese zum RCC (Kosten = Funktion von Verzögerung, Warteschlangenlänge, ...)
- RCC berechnet
  - die k besten Pfade von Knoten i nach Knoten j (für alle i, j)
  - Liste der zum besten Pfad "äquivalenten" Pfade mit

    $c_{ij}^{n} - c_{ij}^{1} < \delta$ mit c_{ij}^{m} = Gesamtlänge des m-besten Pfades

- RCC sendet jedem Knoten für jedes mögliche Ziel eine Liste von äquivalenten Pfaden
- Jeder Knoten darf zwischen den äquivalenten Pfaden frei wählen

Algorithmus "Delta Routing" (2)

Die Wahl von δ entspricht dem Verschieben der „Macht“ zwischen Knoten und RCC:

$\delta \to 0$: RCC trifft die Entscheidung allein

$\delta \to \infty$: der Knoten trifft die Entscheidung allein

Bei geeigneter Wahl von δ kann eine bessere Leistung als bei rein isolierten oder rein zentralisierten Verfahren erreicht werden.

Verteilte Leitwegbestimmung

Prinzip

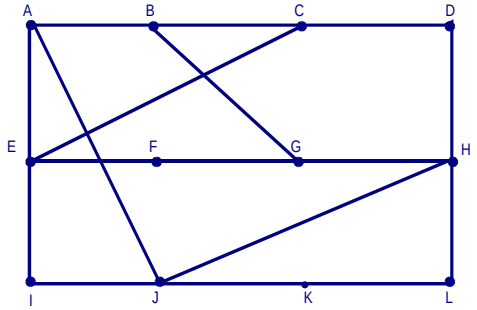
Die Knoten tauschen mit ihren Nachbarn Leitweginformationen aus:

- Jeder Knoten kennt die "Entfernung" zu jedem Nachbarn
  - Anzahl der Teilstrecken (= 1)
  - Verzögerungszeit (Echo-Pakete)
  - Warteschlangenlänge, etc.
- Jeder Knoten sendet periodisch seinen Nachbarn eine Liste mit seinen geschätzten Entfernungen zu jedem Ziel.
- X empfängt eine Liste E vom Nachbarn Y
- Entfernung (X, Y) = e
- Entfernung (Y, Z) = E(Z)
  => Entfernung(X, Z) über Y : E(Z) + e

Die Tabelle mit den einem Knoten bekannten Distanzen heißt **Distanzvektor**. Das Verfahren heißt deshalb auch "**distance vector routing**".

Verteilte Leitwegbestimmung (1)

Beispiel



Wir betrachten die Distanzen vom Knoten J aus.

Verteilte Leitwegbestimmung (2)

	A	I	H	K		
A	0	24	20	21	8	A
B	12	36	31	28	20	A
C	25	18	19	36	28	I
D	40	27	8	24	20	H
E	14	7	30	22	17	I
F	23	20	19	40	30	I
G	18	31	6	31	18	H
H	17	20	0	19	12	H
I	21	0	14	22	10	I
J	9	11	7	10	0	-
K	24	22	22	0	6	K
L	29	33	9	9	15	K
	JA Verzögerung=8	JI Verzögerung=10	JH Verzögerung=12	JK Verzögerung=6		

Rechte Spalte: nach dem Eintreffen der Distanzvektoren neu ermittelte Distanzen von J aus

Hierarchische Leitwegbestimmung

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

- großer Speicherbedarf in den Knoten
- viel CPU-Zeit zum Durchsuchen der Tabellen
- viel Bandbreite zum Austausch von Routing-Informationen.

Eine **hierarchische** Leitwegbestimmung wird ab einer bestimmten Netzgröße notwendig:

- Die Knoten werden in Regionen gruppiert
- Jeder Knoten kennt
  - alle Details seiner Region
  - seine Leitwege zu allen anderen Regionen

Nachteil: nicht immer sind global optimale Entscheidungen möglich

Beispiel für die hierarchische Leitwegbestimmung

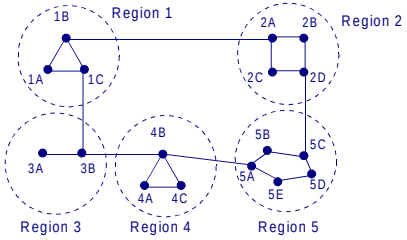


Tabelle für 1 A

Ziel	Leitung	TStrecke
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

Ziel	Leitung	TStrecke
1A	-	-
1B	1B	1
1C	1C	1
2	1B	2
3	1C	2
4	1C	3
5	1C	4

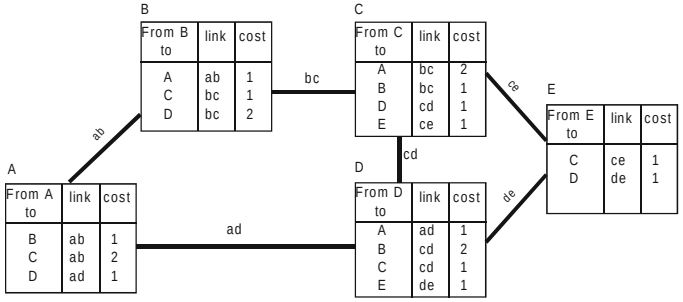
Routing im Internet

Distance Vector Routing

Das über viele Jahre am meisten verwendete Verfahren im Internet ist ein adaptives verteiltes Verfahren auf der Basis von Distanzvektoren (distance vector routing). Das eingesetzte Protokoll heißt **RIP** (Routing Information Protocol).

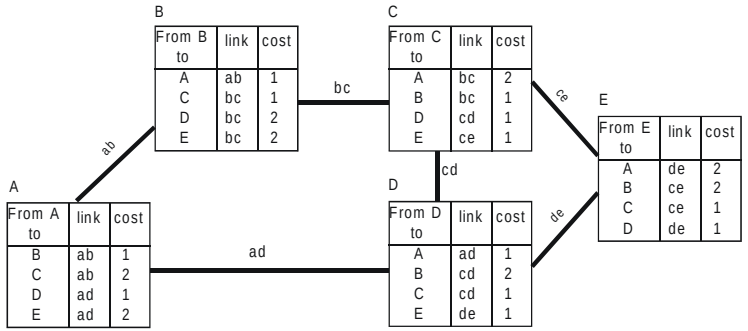
Alle Internet-Router tauschen dabei periodisch RIP-Nachrichten aus und aktualisieren ihre Routing-Tabellen beim Eintreffen von RIP-Nachrichten von ihren Nachbarn.

Beispiel für Routing mit Distanzvektoren (1)



(a) Knoten E ist gerade hinzugekommen

Beispiel für Routing mit Distanzvektoren (2)



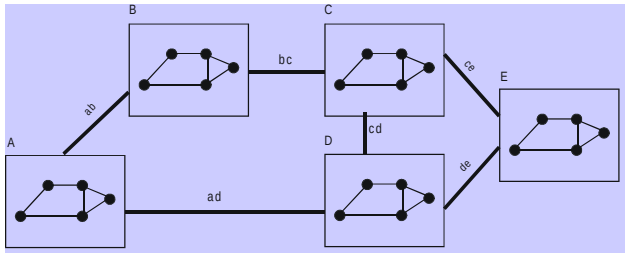
(b) nach einer Runde von DVRP-Nachrichten

OSPF-Routing

Ein zweiter wichtiger Routing-Algorithmus im Internet ist **OSPF** (**Open Shortest Path First**). Die Idee ist, dass alle Knoten jederzeit die gesamte Netztopologie kennen und lokal alle optimalen Pfade berechnen können. Wenn sich die Topologie ändert, tauschen die Knoten Änderungsnachrichten aus. Jeder Knoten unterhält lokal eine Datenbank über die gesamte Topologie.

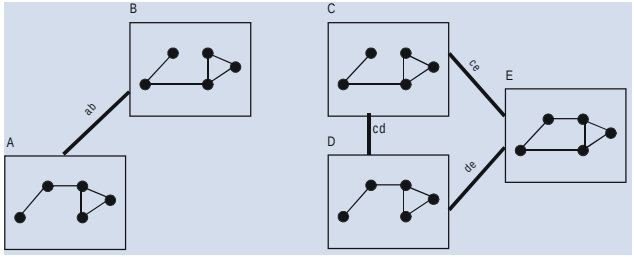
Auf der Basis der vollen Topologie werden die optimalen Pfade zu allen anderen Knoten mit dem Algorithmus für kürzeste Wege von Dijkstra (Shortest Path First = SPF) berechnet. Im Internet-Slang heißt der Algorithmus deshalb auch Open Shortest Path First (OSPF).

Beispiel für OSPF-Routing (1)



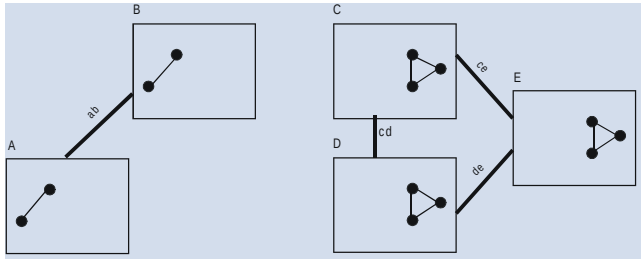
(a) Netzwerk im stabilen Zustand

Beispiel für OSPF-Routing (2)



(b) Die Links bc und ad sind ausgefallen

Beispiel für OSPF-Routing (3)



(c) Nach einer Runde von OSPF-Nachrichten

5.4 Wegewahl (Routing) für Multicast-Netze

Definition Multicast

Unter Multicast versteht man die Übertragung eines Datenstroms von einem Sender an mehrere Empfänger.

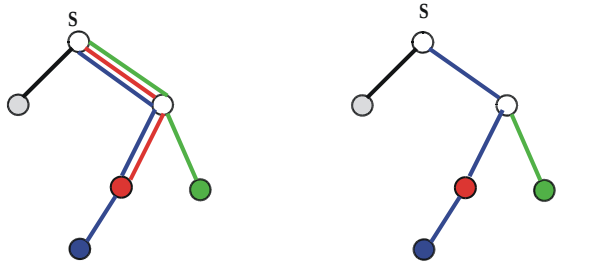
Warum ist Multicast wichtig für Multimedia?

- Multimedia-Anwendungen erfordern oft eine 1:n - Kommunikation. Beispiele:
  - Videokonferenz
  - Tele-Kooperation (CSCW) mit gemeinsamem Arbeitsbereich
  - near-Video-on-Demand
  - Verteil-Kommunikation (Broadcast)
- Digitale Video- und Audioströme haben sehr hohe Datenraten (1,5 MBit/s und mehr). Eine Übertragung über n einzelne Verbindungen würde das Netz überlasten.

Motivation für Multicast

Mehr „Intelligenz“ im Netz verringert:

- die Last bei den Sendern
- die Last auf den Teilstrecken



Anforderungen an Multicast für Multimedia (1)

- Unterstützung von isochronen Datenströmen mit **garantierter Dienstgüte**
  - maximale Ende-zu-Ende-Verzögerung (delay)
  - maximale Varianz in der Verzögerung (delay jitter)
  - maximale Fehlerrate (error rate)
    für eine vereinbarte Verkehrslast (Vertragsmodell)
- Erfordert eine Reservierung von Ressourcen in allen Links und Knoten im Netz
  - Bandbreite
  - CPU-Leistung
  - Pufferplatz
  - "schedulability"

Anforderungen an Multicast für Multimedia (2)

- Erfordert Formate und Protokolle für eine **Gruppenadressierung**
- Erfordert neue Algorithmen für die **Fehlerkorrektur** (z. B. FEC oder Reliable Multicast)
- Erfordert Algorithmen für **dynamisches Hinzufügen und Löschen** von Teilnehmern in einer Session

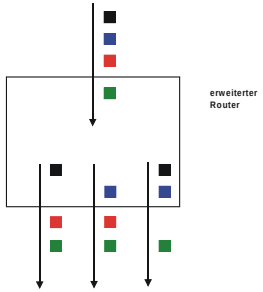
Multicast in LANs

Ethernet, Token Ring, FDDI, Wireless LAN

- Die Topologie hat Broadcast-Eigenschaft.
- Die Schicht-2-Adressen nach IEEE 802.2 erlauben die Verwendung von Gruppenadressen für Multicast. Dadurch lässt sich Multicast in einem LAN-Segment leicht und effizient realisieren.
- **Aber:** Ab Schicht 3 wurden in der Internet-Protokollarchitektur lange Zeit nur Peer-to-Peer - Adressen unterstützt! Und im weltweiten Verbund (insbesondere im Internet) muss Multicast auch WAN-Strecken überbrücken.

Multicast in der Netzwerkschicht

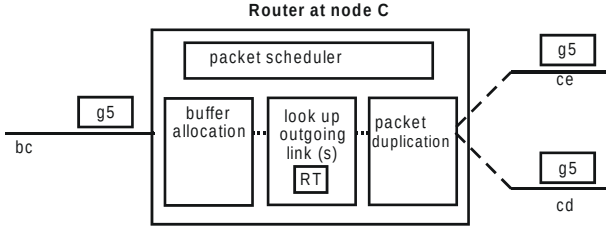
- Prinzip: Duplizierung von Paketen so "tief unten" im Multicast-Baum wie möglich
- Erfordert ein Multicast-Adressierungsschema in Schicht 3 und mehr "Intelligenz" in den Schicht 3 - Vermittlungsstellen (Routern)
- verbindungslos oder verbindungsorientiert?



Router mit Multicast-Erweiterung

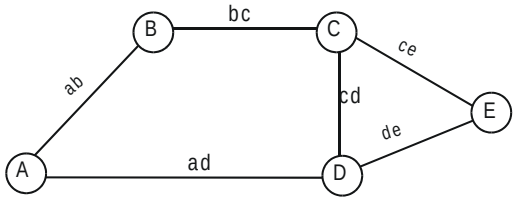
RT Routing Table

From C to	link	cost
g5	{ce, cd}	

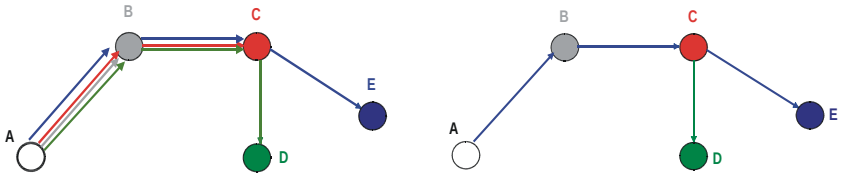


g5 = IP address of group 5

Beispiel-Topologie



Der Vorteil von Multicast in der Beispiel-Topologie



(a) Vier einzelne Verbindungen (b) eine Multicast-Verbindung

Routing-Algorithmen für Multicast

Multicast Routing ist bisher nur im Internet in Schicht 3 realisiert worden (Multicast-IP). Die eingesetzten Algorithmen sind Erweiterungen der klassischen Routing-Algorithmen; sie sind mit diesen kompatibel.

Multicast im Internet ist **empfängerorientiert**. Für eine Multicast-Session wird zunächst eine IP-Gruppenadresse vereinbart. Der Sender beginnt, an diese Adresse zu senden. Jeder Knoten im Internet kann entscheiden, ob er in eine existierende Multicast-Gruppe aufgenommen werden möchte.

Prinzipien des Multicast-IP-Protokolls

- Übertragung von IP-Datenpaketen an eine **Gruppenadresse** (IP-Adresse vom Typ D)
- verbindungslos (Datagrammdienst)
- Best-Effort-Prinzip (keine Dienstgütegarantien):
  - keine Fehlerkontrolle
  - keine Flusskontrolle
- empfängerorientiert:
  - Der Sender sendet Multicast-Pakete an die Gruppe.
  - Der Sender kennt die Empfänger nicht, hat auch keine Kontrolle über diese.
  - Jeder Host im Internet kann einer Gruppe beitreten.
- Eine Beschränkung des Sendebereiches ist nur durch den Time-To-Live-Parameter möglich (TTL = hop counter im Header des IP-Pakets)

Multicast-Adressen in IP

Für eine Multicast-Session wird zunächst eine IP-Gruppenadresse vereinbart. Der Sender beginnt, an diese Adresse zu senden. Jeder Knoten im Internet kann entscheiden, ob er in eine existierenden Gruppe aufgenommen werden möchte. Die IP-Gruppenadresse wurde als IP-**Adresse der Klasse D** standardisiert.

Gruppenadressen werden dynamisch zugewiesen. Einen Mechanismus zur eindeutigen Vergabe einer Gruppenadresse gibt es in IP nicht! Um eindeutige Gruppenadressen müssen sich die höheren Schichten kümmern.

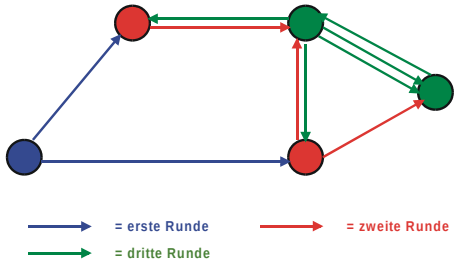
Routing-Algorithmen für Multicast

Flooding

Die einfachste Möglichkeit zum Erreichen aller Empfänger einer Gruppe wäre Flooding (Broadcasting).

Algorithmus Flooding

Wenn ein Paket eintrifft, wird eine Kopie auf jeder Ausgangsleitung weiter gesandt außer derjenigen, auf der das Paket ankam.



Reverse Path Broadcasting (RPB)

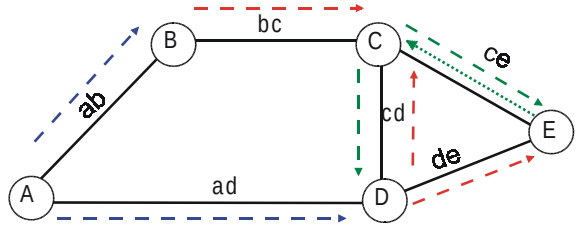
Effizienter als das Flooding ist der Reverse Path Broadcasting-Algorithmus (RPB). Er nutzt die Tatsache aus, dass jeder Knoten seinen kürzesten Pfad zum Sender aus der klassischen (point-to-point)-Routing-Tabelle kennt! Man bezeichnet diesen Pfad als **Reverse Path**.

Die erste Idee ist nun, dass ein Knoten nur diejenigen Pakete an seine Nachbarn weiter gibt, die auf dem kürzesten Pfad vom Sender angekommen sind.

Dieses Verfahren generiert wesentlich weniger Pakete als Flooding.

Beispiel für Reverse Path Broadcasting (unvollständiger Algorithmus)

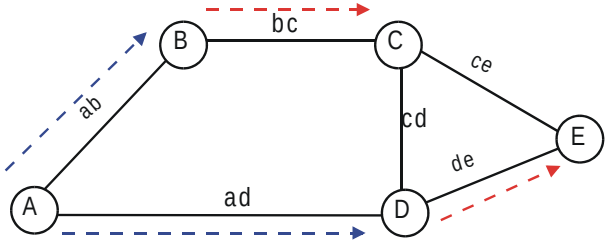
Für unsere Beispieltopologie arbeitet der (bisher noch unvollständige) RPB-Algorithmus wie folgt:



Wie wir sehen, entstehen noch immer überflüssige Pakete: die Knoten D und E erhalten jedes Paket zweimal, Knoten C sogar dreimal.

Reverse Path Broadcasting (vollständiger Algorithmus)

Wenn jeder Knoten seinen Nachbarn etwas Zusatzinformation mitteilt, kann RPB weitere überflüssige Pakete verhindern. Die Zusatzinformation besteht in der Benennung des eigenen kürzesten Pfades zum Sender. In unserem Beispiel informiert E seine Nachbarn C und D darüber, dass *de* auf seinem kürzesten Pfad zu A liegt. Ein Knoten leitet Pakete dann nur noch an diejenige “Söhne“ weiter, von denen er weiß, dass er auf ihrem kürzesten Pfad zum Sender liegt. Den Paketfluss für den vollen RPB-Algorithmus zeigt dann die unten stehende Abbildung.



Truncated Reverse Path Broadcasting (TRPB)

TRPB beschränkt die Auslieferung der Daten auf diejenigen Subnetzwerke, die Gruppenmitglieder enthalten. Als Subnetzwerke werden nur LANs betrachtet, die an Blättern des Routing-Baumes hängen.

Dazu wurde ein einfaches Protokoll definiert, mit dem Router die Hosts in ihrem LAN befragen können, ob sie an den Paketen einer bestimmten Gruppe interessiert sind (**IGMP**: Internet Group Management Protocol). Wenn ein Router in seinem LAN keinen interessierten Host vorfindet, wird er in Zukunft Pakete mit dieser Gruppenadresse nicht mehr auf sein LAN geben.

Vorteil

Vermeidet überflüssige Pakete in den Blatt-LANs

Nachteil

Eliminiert nur Blatt-Subnetzwerke, verringert nicht den Datenverkehr innerhalb des Baumes

Reverse Path Multicasting (RPM)

Der TRPB-Algorithmus etabliert Pfade zu allen Routern im Netz, ob sie Mitglied der Gruppe sein wollen oder nicht. Es ist offensichtlich sinnvoll, in der Datenphase einer Session den Routing-Baum so zurück zu schneiden, dass Pakete nur noch dorthin weitergeleitet werden, wo sie wirklich gebraucht werden.

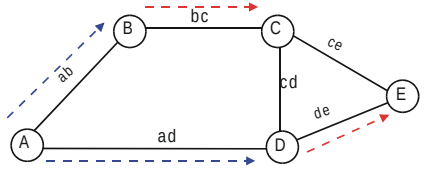
Dies geschieht durch die Generierung von "**prune messages**". Diese wandern im Baum von den Blättern zur Wurzel hin und teilen den Knoten der jeweils höheren Ebene mit, dass es weiter unten im Baum keine Empfänger mehr gibt. So wird aus dem Broadcast-Baum ein Multicast-Baum. Das Verfahren wird als **Reverse Path Multicasting (RPM)** bezeichnet. Im Internet werden die "prune messages" von den Routern generiert und weitergeleitet.

Im Internet heißt das Protokoll zum RPM-Algorithmus **DVMRP** (Distance Vector Multicast Routing Protocol).

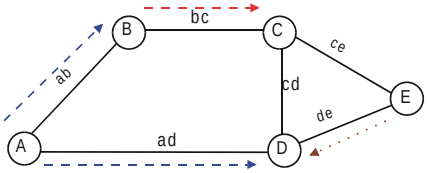
Algorithmus Pruning

- Ein Router, der als Kind-Links nur Blatt-Links ohne Gruppenmitglieder besitzt, sendet einen Non-Membership-Report (NMR) an den übergeordneten Router, d. h. an den vorher gehenden Router im Multicast-Baum.
- Router, die von allen untergeordneten Routern NMRs empfangen haben, senden ebenfalls einen NMR an den übergeordneten Router.
- NMRs enthalten eine Zeitschranke, nach der das Pruning wieder aufgehoben werden soll.
- NMRs können auch per Nachricht aufgehoben werden, wenn ein neues Gruppenmitglied unterhalb eines Links aktiv wird.

Beispiel für Reverse Path Multicasting (1)

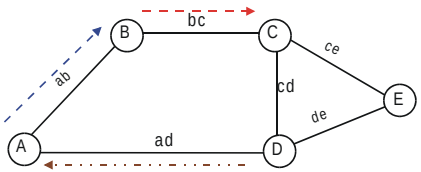


(a) Baum in der anfänglichen RPB Phase



(b) E hat eine "prune message" versandt

Beispiel für Reverse Path Multicasting (2)



(c) D hat eine "prune message" versandt

Vor- und Nachteile von RPM

Vorteil

- Reduzierung des Datenverkehrs im Vergleich zu TRPB

Nachteile

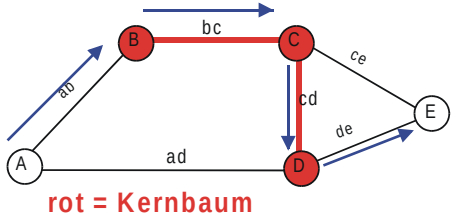
- Periodischer Versand der Daten an alle Router weiterhin nötig, damit sie „es sich anders überlegen“ können
- Statusinformation in jedem Knoten für jede Gruppe und für jeden Sender nötig
- Für jedes Paar (Sender, Gruppenadresse) muss ein eigener Routing-Tree aufgebaut werden.

Kernbäume (Core-Based Trees)

Alle bisher dargestellten Verfahren haben den Nachteil, dass pro (Sender, Gruppe)-Paar ein eigener Multicast-Baum aufgebaut und verwaltet werden muss. Diesen Nachteil vermeiden die **Kernbäume** (“core-based trees“). Es wird nur ein Baum pro Gruppe eingerichtet. Jeder Sender sendet zum Baum hin. Die Nachrichten werden entlang des Baumes transportiert und erreichen von hier aus die Blätter.

Ein Beispiel zeigt die unten stehende Abbildung. Das beste heute im Internet verfügbare Multicast-Routing-Protokoll heißt PIM-SM (Protocol-Independent Multicast – Sparse Mode). Es basiert auf der Idee des Kernbaums.

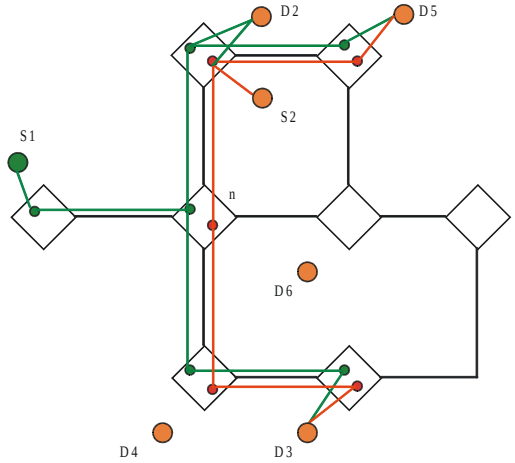
Beispiel für einen Kernbaum



QoS-Based Routing

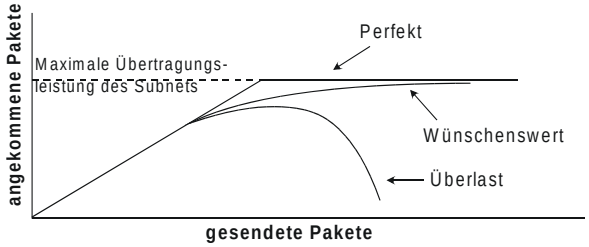
Multicast-Routing für IP ist nach wie vor ein aktuelles Forschungsthema.
Noch weitgehend ungelöst ist das Problem eines Routings unter Einbeziehung von Dienstgüteanforderungen ("**QoS-based routing**").

Dynamic Join and Leave mit QoS-Garantie



5.5 Überlastkontrolle in der Vermittlungsschicht

Das Problem



Gründe für eine Überlast

- Knoten zu langsam für Routing-Algorithmen
- Ankommender Verkehr überfordert Ausgangsleitungen

Überlast: Tendenz zur Selbstverstärkung

Überlastung tendiert dazu, sich selbst zu verstärken.

Beispiel: Ein Knoten (Router) wirft wegen Überlastung Pakete weg

- Pakete müssen erneut gesendet werden (zusätzlicher Verbrauch an Bandbreite)
- Sender kann seine Puffer nicht freigeben (zusätzliches Binden von Ressourcen)

Besonders kritisch in Datagramm-Netzen!

Verfahren 1: Pufferreservierung (1)

Prinzip

- Voraussetzung: Virtuelle Verbindungen
- Reservierung der benötigten Puffer beim Verbindungsaufbau
  - Falls nicht genügend Puffer vorhanden: alternativen Pfad wählen oder Verbindungswunsch abweisen

Beispiel 1:

Bei Verwendung des Stop-and-Wait-Protokolls zur Flusskontrolle: ein Puffer pro Knoten und Verbindung (simplex)

Beispiel 2:

Bei Verwendung des Sliding-Window-Protokolls zur Flusskontrolle: *w* Puffer pro Knoten und Simplex-Verbindung (*w* = Fenstergröße)

Verfahren 1: Pufferreservierung (2)

Eigenschaften

- Keine Überlastung möglich

aber

- die Puffer bleiben *verbindungsbezogen* reserviert, auch wenn zeitweise keine Daten übertragen werden.

Daher meist nur bei Anwendungen eingesetzt, wo garantierte geringe Verzögerung und hohe Bandbreite erforderlich sind, z. B. bei der digitalen Sprachübertragung über paketvermittelte Netze.

Verfahren 2: Wegwerfen von Paketen (1)

Prinzip

- Keine Reservierung von Ressourcen
- Ankommendes Paket wird weggeworfen, wenn es nicht gepuffert werden kann

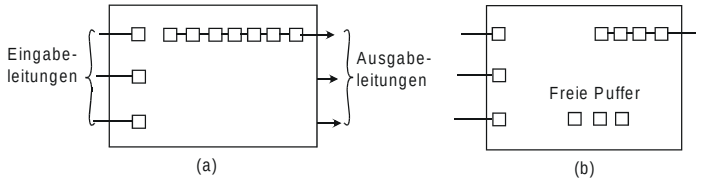
Datagramm-Dienst: Keine weiteren Vorkehrungen notwendig

Verbindungsorientierter, zuverlässiger Dienst: Puffern jedes Pakets beim Sender, bis der Empfang vom Endsystem quittiert ist.

Verfahren 2: Wegwerfen von Paketen (2)

Eine "unfaire" Beeinträchtigung fremder Paketströme kann dadurch verringert werden, dass für die Paketanzahl in der Ausgabeschlange einer Ausgangsleitung eine Obergrenze definiert wird.

Aber dann: Verwerfen von Paketen kann trotz freier Puffer vorkommen.



Verfahren 2: Wegwerfen von Paketen (3)

Eigenschaften

- sehr einfach

aber

- wiederholt übertragene Pakete verschwenden Bandbreite

Ein Paket muss 1 / (1 - p) mal gesendet werden, bevor es akzeptiert wird (p = Wahrscheinlichkeit, dass Paket verworfen wird)

Kleine, einfache Optimierung:

Zunächst die Pakete wegwerfen, die noch nicht weit gekommen sind (Streckenzähler auswerten)

Verfahren 3: Isarithmische Überlastkontrolle

Prinzip

Begrenzung der Anzahl von Paketen im Netz durch Vergabe von "Permits"

- Menge von "Permits" kreist im Netz
- Zum Senden wird ein "Permit" benötigt
  - Senden eines Pakets: „Permit" wird zerstört
  - Empfangen eines Pakets: neuer „Permit" wird generiert

Probleme

- Teile des Netzes können überlastet werden, während andere Teile unterbelastet sind
- Gleichmäßige Verteilung der Permits über das Netz ist schwierig
- Zusätzliche Bandbreite wird für Permit-Transfer benötigt
- Ungeeignet zur Übertragung großer Datenmengen (z. B. Dateitransfer, multimedialer Datenstrom)
- Endgültiger Verlust von Permits durch Fehler im Netz schwer zu erkennen und zu beheben

Verfahren 4: Flusskontrolle missbrauchen

Prinzip

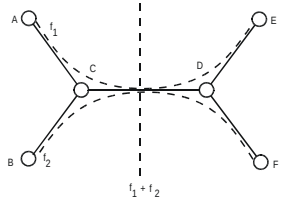
Flusskontrolle zur Überlastvermeidung missbrauchen:

- Die Flusskontrolle ist definiert zwischen Paaren von Endsystemen. Sie soll eigentlich nur das "Überschwemmen" des Empfängers durch einen zu schnellen Sender verhindern.
- Das Netz darf bei Verfahren 4 aber nun in den inneren Knoten von sich aus die Fenstergröße des Sliding-Window-Protokolls einer Verbindung verändern. Dadurch kann der Fluss der Pakete auf der Verbindung verlangsamt werden.
- Implementiert zum Beispiel in Schicht 3 von SNA (IBM)

Im Internet ist eine sehr merkwürdige Variante implementiert. Nur TCP (nicht UDP) versucht, in den Endsystemen auf Schicht 4 zu raten, wann das Netz verstopft ist. Dies geschieht auf der Basis von beobachteten Paketverlusten. Wenn der TCP-Sender Verstopfung vermutet, reduziert er freiwillig seine Senderate, um die Überlast im Inneren des Netzes zu mindern. Protokoll-Details werden wir im TCP-Kapitel besprechen.

Nachteile der Überlastkontrolle durch Flusskontrolle

- Es ist im Sinne der Schichtenarchitektur unsauber, wenn die Überlastkontrolle in Schicht 4 gemacht wird. Denn Schicht 3 muss den Flusskontrollparameter im Paket-Header der Schicht 4 verändern, obwohl sie eigentlich das Protokoll der Schicht 4 nicht kennen sollte.
- Funktioniert nur bei verbindungsorientierter Kommunikation, nicht für Datagrammverkehr! Problem: Wie implementiert man ein „TCP-friendly flow control for UDP"?
- Oft führen mehrere Paketflüsse über einen gemeinsamen Link der Schicht 3. Wie kann die Flussreduzierung **fair** erfolgen?



Verfahren 5: "Choke"-Pakete

Prinzip

Netzmanagement-Pakete drosseln den Verkehr bei Überlast:

- Jede Ausgangsleitung eines Routers ist mit einer Variablen u ($0 \leq u \leq 1$) versehen, die die aktuelle Auslastung angibt
- u > Grenzwert: Leitung geht in den Zustand "Warnung"
- Wenn die Ausgangsleitung für ein Paket im Zustand "Warnung" ist, sendet der Router für jedes eintreffende Paket ein "Choke"-Paket an die Quelle
- Wenn die Quelle ein Choke-Paket empfängt, reduziert sie den Datenverkehr zu dem betreffenden Ziel

Variante

Es gibt mehrere Grenzwerte für u, die zu unterschiedlich harten Warnungen führen und den Sender zu unterschiedlichen Reduzierungen des Datenstroms veranlassen.

5.6 Beispiele: IP, IPv6, X.25, ATM

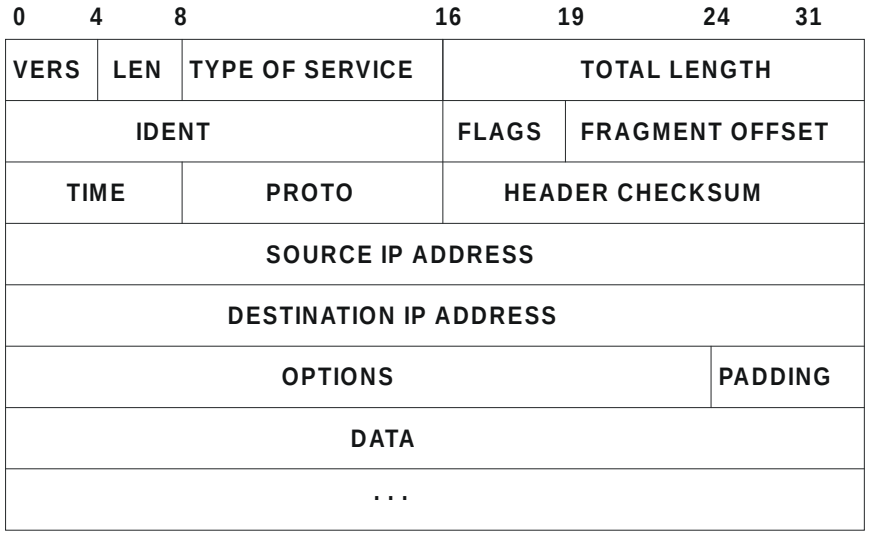
IP (Internet Protocol)

Das Protokoll der Schicht 3 im Internet.

- Ein Datagramm-Protokoll (verbindungslos)
- Implementiert Routing im Internet
- Handhabt die Fragmentierung großer Pakete: große Dienst-Datagramme können in kleinere Protokoll-Datagramme „fragmentiert“ werden.
- Macht sonst nicht viel!

Wenn wir hier „IP“ sagen, meinen wir IP Version 4. IP Version 6 wird später besprochen.

Format von IP-Datagrammen (1)

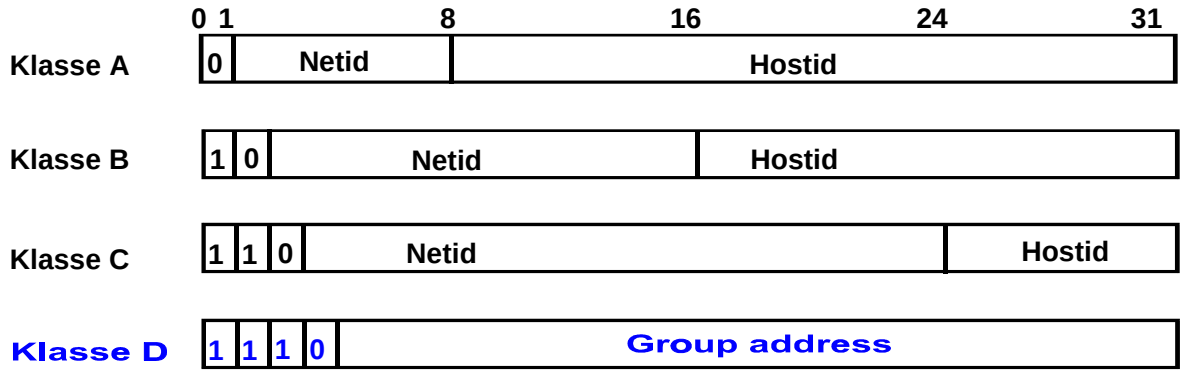


Format von IP-Datagrammen (2)

VERS	Protokollversion
LEN	Länge des Headers (Wörter)
TYPE OF SERVICE	QoS (Priorität und D/T/R)
TOTAL LENGTH	Länge incl. Daten in Bytes
IDENT	Identität des Datagramms
FLAGS	"nicht fragmentieren/letztes Fragment"
FRAGMENT OFFSET	Offset dieses Teils
TIME	Lebensdauer in Sekunden ("time to live")
PROTO	Type des höheren Protokolls
HEADER CHECKSUM	EXOR der Header-Wörter
SOURCE ADDRESS	IP-Adresse des Quell-Hosts
DEST ADDRESS	IP-Adresse des Ziel-Hosts
OPTIONS	Kommandocode für Netzmanagementdatagramme
PADDING	Auffüllen auf Wortgrenze
DATA	Nutzdatenfeld

Adressierung im Internet (1)

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:



Adressierung im Internet (2)

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

10.0.0.0	für Arpanet
128.10.0.0	für ein großes Ethernet-LAN
192.5.48.0	für ein kleines Ring-LAN

(hostid = 0 bezeichnet ein Netz aus einem Host)

Adressauflösung im LAN

Problem

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

Lösung

Das Address Resolution Protocol (ARP)

Address Resolution Protocol ARP (1)

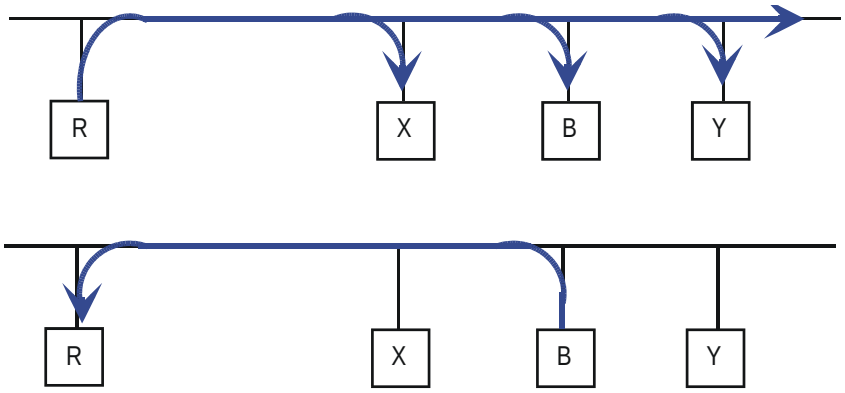
Protokoll im Router

- Sende mittels Broadcast auf dem LAN ein ARP-Request-Paket, welches die physikalische Adresse nach 802.2 und die Internet-Adresse des Senders und die Internet-Adresse des gesuchten Empfängers enthält.
- Warte auf die Antwort des Empfängers durch ein ARP-Reply-Paket, welches seine physikalische Adresse enthält.
- Unterhalte einen Cache aus (IP, 802.2)-Adresspaaren für spätere Anfragen.

Verbesserung: Der Empfänger des ARP-Requests speichert das (IP, 802.2)-Adresspaar des Senders auch in seinem Cache.

Address Resolution Protocol ARP (2)

Ablauf des ARP-Protokolls im LAN



IP Version 6 (IPv6)

Motivation: Adressierungsprobleme

- IP-Adressraum wird irgendwann auslaufen
- Klasse-B-Adressen sind nahezu erschöpft.
- CIDR (classless inter-domain routing) wurde als kurzfristige Lösung eingeführt
- Routing-Tabellen wachsen sehr schnell: es wird eine Adressierungshierarchie mit zusätzlichen Ebenen erforderlich
- (Mobile) Internet-Geräte in Autos, Haushalten etc.
  → 10 Milliarden Menschen im Jahr 2020 und
  100 IP-Adressen pro Person sind nicht unrealistisch.

Lösung

Neues IP-Protokoll Version 6 (IPv6) mit grösserem Adressraum soll IPv4 ersetzen.

*) Ich danke Prof. Torsten Braun (Universität Bern) für die Überlassung seiner Folien zum Thema IPv6.

Geschichte von IPv6

1992

IETF publiziert Call for Proposals für ein “IP next generation“ (IPng), um die aktuelle IP Version 4 zu ersetzen

1994

SIPP (Simple Internet Protocol Plus) wird als Grundlage für das neue IPng vorgeschlagen

1995

Internet Draft „Internet Protocol, Version 6 (IPv6)“ wird Proposed Standard“ (9/95) und RFC1883 (12/95). Erste prototypische Implementierungen

1996

Aufbau des IP Version 6 Backbones (6Bone) zwischen Forschungslabors, erste Produkte im Markt

1998

RFC 2460, Draft Standard

2002

Noch immer nicht sehr weit verbreitet in Produkten und bei ISPs

Eigenschaften von IPv6 (1)

Erweiterte Adressierungsmöglichkeiten

- 128-Bit-Adressen (eine Adresse pro Atom im Weltall)
- Adresshierarchieebenen für IP (Registrierungsinstanz, Provider, Subscriber, Subnetz, Interface)
- automatische Adresskonfiguration (ähnlich DHCP)

Neues IP-Header-Format

- vereinfachter, minimaler IP-Header
- verbesserte Unterstützung neuer Optionen und Erweiterungen: Erweiterungs-Header
- Segmentieren und Reassemblieren von Fragmenten in Endsystemen

Eigenschaften von IPv6 (2)

Quality-of-Service-Unterstützung

- “Flow Labels“ erlauben Markierung von Anwendungsdatenflüssen auf IP-Ebene
- “Traffic Classes“ für Differentiated Services

Multicast-Integration

- vordefinierte Multicast-Gruppen für Kontrollfunktionen
- IGMP (Internet Group Management Protocol) in ICMP (Internet Control Message Protocol) integriert
- spezielles Multicast-Adressformat
- Alle Router und Endsysteme unterstützen Multicast, so dass keine speziellen Massnahmen für Multicast-Verbindungen mehr erforderlich sind.

IP Security

- Authentifizierung und Verschlüsselung sind Bestandteil des Protokolls.

Aggregierbare globale Unicast-Adresse

Top Level Aggregation (TLA)

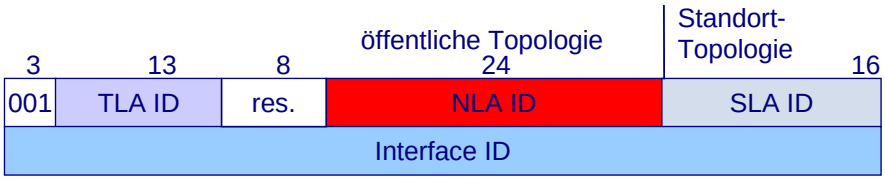
- grosse Internet-Service-Provider (ISPs) mit Transitnetzen, an die andere ISPs angeschlossen sind

Next Level Aggregation (NLA)

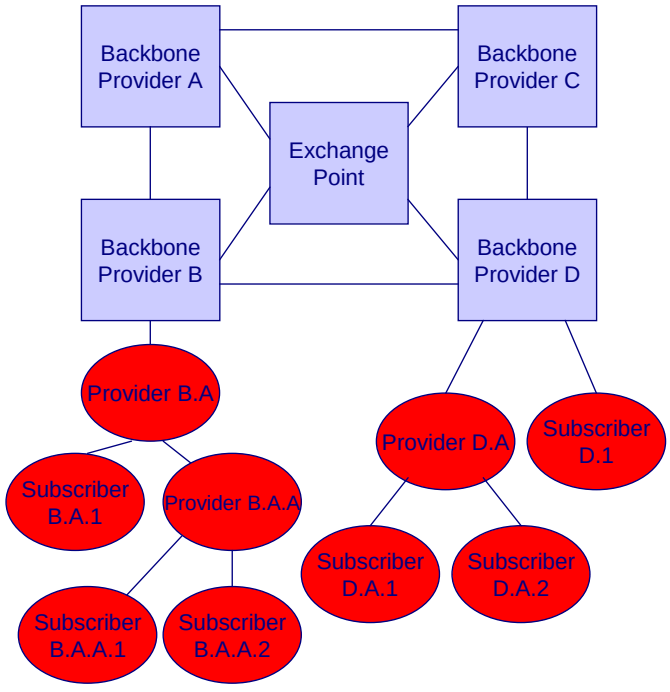
- Organisationen auf einer niedrigeren Stufe
- mehrere NLA-Ebenen sind möglich

Site Level Aggregation (SLA)

- individuelle Adressierungshierarchie einer einzelnen Organisation



TLA, NLA und SLA



Spezielle Unicast-Adressen (1)

Lokale Unicast-Adressen

- Link-lokal
  - für Konfigurationszwecke oder IP-Netze ohne Router
  - Präfix: 111111101::/64
- Standort-lokal
  - für nicht an das Internet angeschlossene IP-Netze
  - beim Anschluss eines Standorts muss lediglich das Adresspräfix (1111111011::/48) ersetzt werden
  - SLA und Interface ID bleiben unverändert

Spezielle Unicast-Adressen (2)

Kompatible Unicast-Adressen

- IPv4-kompatibel
  - Präfix (96 0-bits) + IPv4-Adressen
- IPv4-mapped
  - Präfix (80 0-bits, 16 1-bits) + IPv4-Adressen
- IPX-kompatibel

IPv4 Header

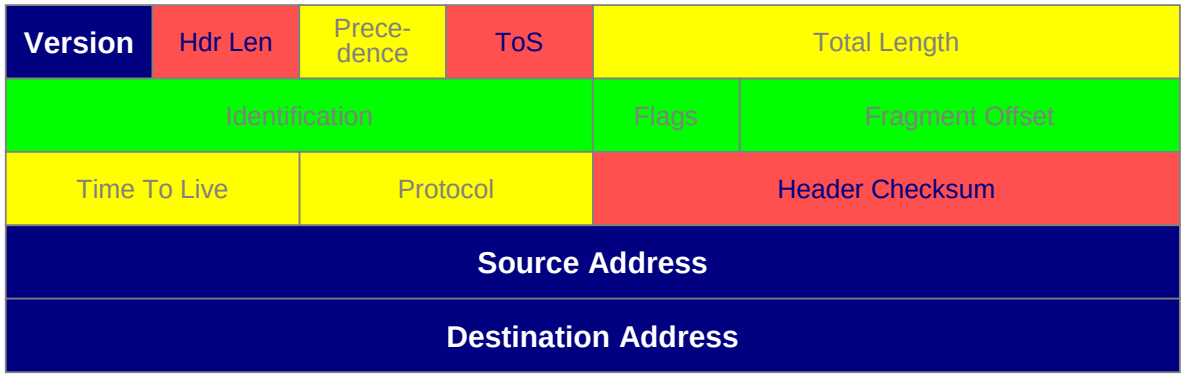
20 Bytes, 13 Felder

gestrichen

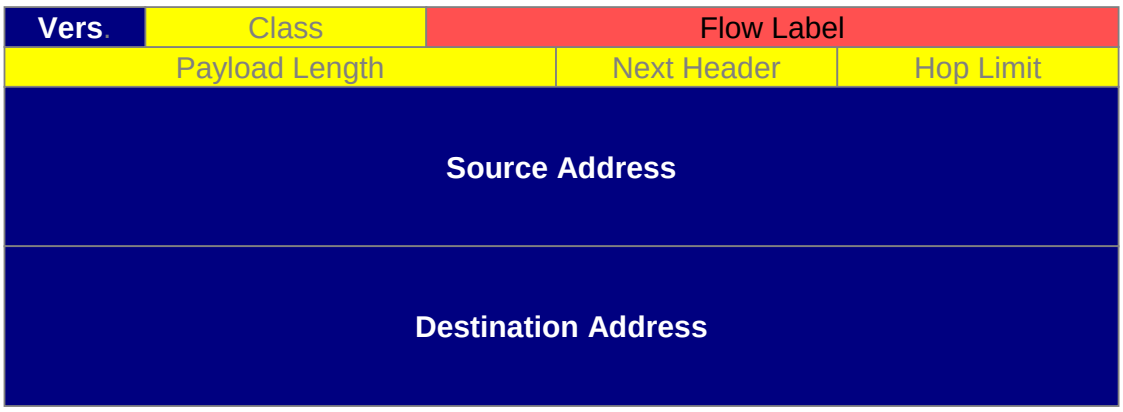
in den Erweiterungs-Header verschoben

umbenannt

- precedence → class
- total length → payload length
- time to live → hop limit
- protocol → next header



IPv6 Header

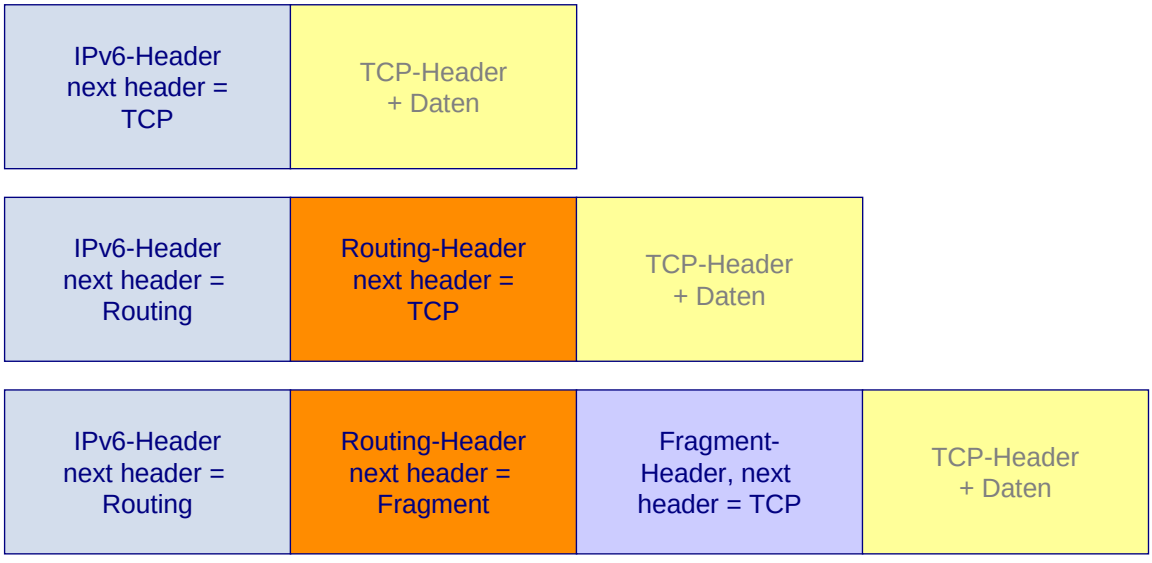


Konzept der Erweiterungs-Header

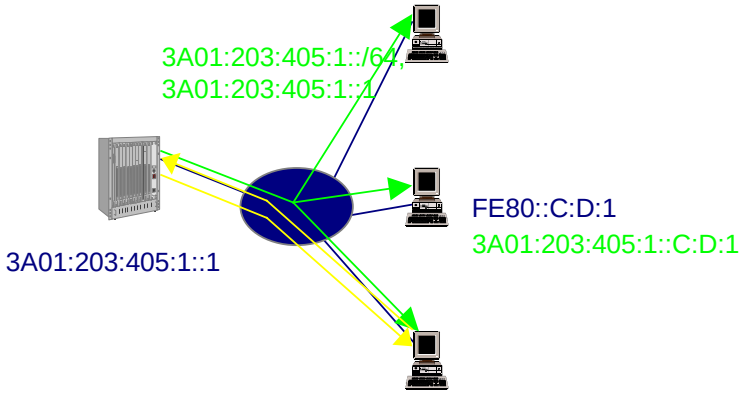
Kleiner Standard-Header und Erweiterungs-Header

- kleiner minimaler Header
- flexible weitereHeader, abhängig von den Anforderungen der Anwendung oder den Eigenschaften der Netze
- einfache Einführung neuer zukünftiger Erweiterungen und Optionen

Beispiele für Erweiterungs-Header



Zustandslose automatische Adresskonfiguration



Der Router verbreitet Parameter periodisch an die Multicast-Gruppe *aller* Hosts (Router Advertisement).

Jeder Host sendet eine “Router Solicitation“ an die Multicast-Gruppe aller Router, es folgt eine direkte Antwort des Routers.

Übergangsstrategien

IPv4- und IPv6-Systeme müssen miteinander kommunizieren können. Nach einer Übergangsphase sollen nur noch einige wenige reine IPv4-Systeme übrig bleiben.

Basismechanismen

- Doppelte Protokoll-Stacks
- IPv6-in-IPv4-Tunneling

Die IPv6/IPv4-Header-Übersetzung ist nur für die Kommunikation zwischen reinen IPv4-Knoten und reinen IPv6-Knoten notwendig.

Komplexere Mechanismen

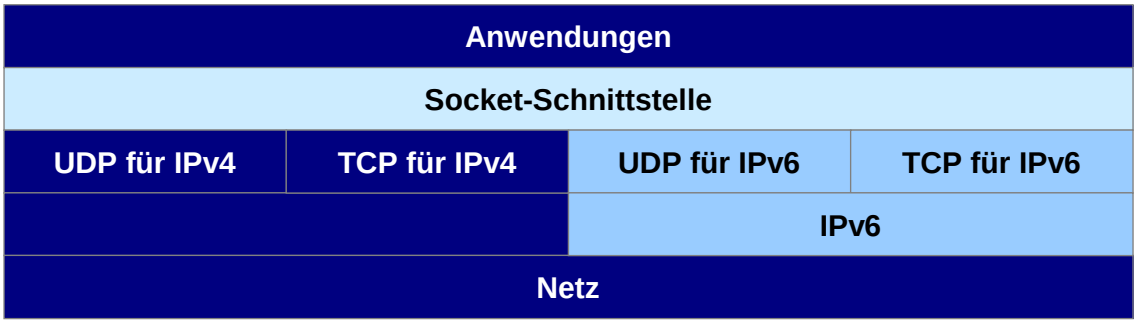
- Stateless IP/ICMP Translation (SIIT)
- No Network Address Translation (NNAT)
- Network Address Translation / Protocol Translation (NAT/PT)

Doppelte Protokoll-Stacks

Doppelte Protokoll-Stacks

- UDP/IPv4 und UDP/IPv6
- TCP/IPv4 und TCP/IPv6

Alle IPv6-Systeme werden während der Übergangsphase auch einen IPv4-Stack haben.



IPv4-kompatible Adresse

Systeme verwenden ihre alte IPv4-Adresse, um eine IPv6-Adresse zu bilden.

Benutzung durch IPv6-Systeme zur Kommunikation mit anderen IPv6-Systemen unter Verwendung von automatischen Tunneln.

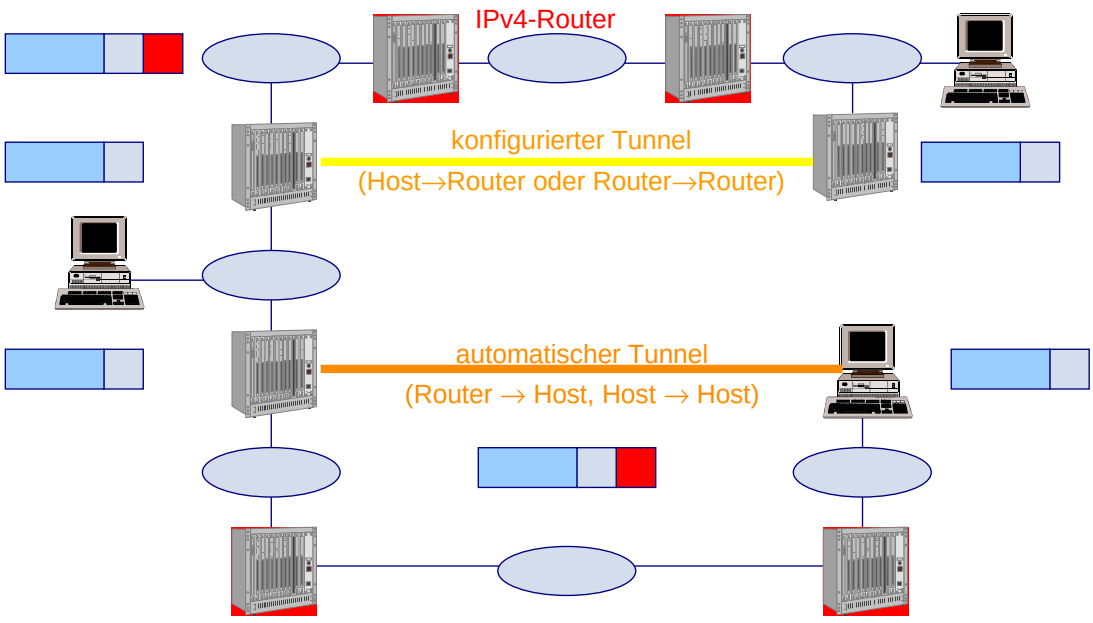
Nur nützlich in der frühen Übergangsphase; Verlust der Vorteile der IPv6-Adressierung

Tunneling

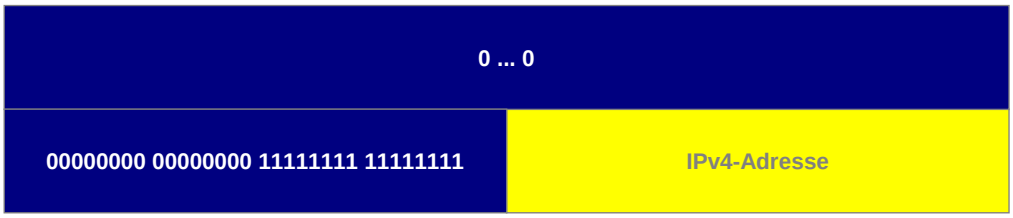
Unter **Tunneling** versteht man das Einpacken eines IP-Pakets in ein anderes IP-Paket, das eine neue, eigene IP-Zieladresse bekommt. Am Ende des Tunnels wird das innere IP-Paket wieder ausgepackt. So kann das ursprüngliche IP-Paket Teilnetz-Strecken „untertunneln“, die es anders nicht passieren könnte.

In der Regel werden IP-Tunnel von Hand konfiguriert.

IPv6-Tunnel



Adresse vom Typ „IPv4-Mapped“



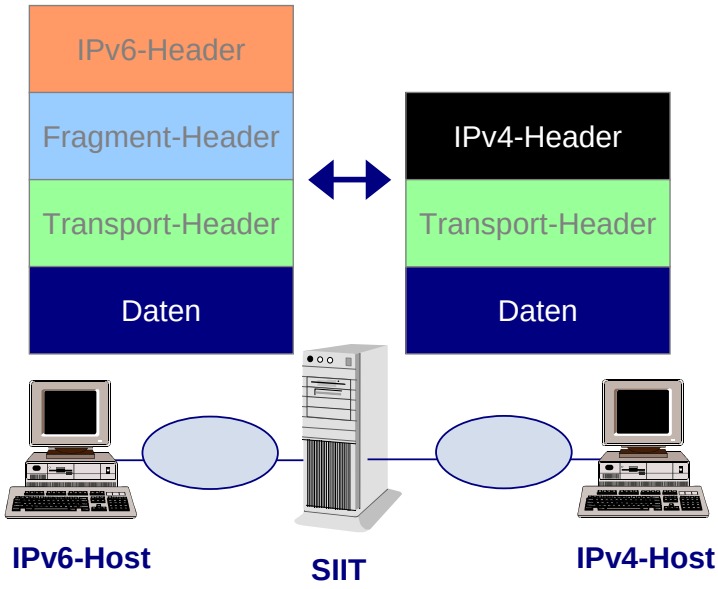
Kommunikation von IPv6-Systemen mit IPv4-Systemen

„IPv4-Mapped“-Adresse zeigt an, dass das adressierte IP-Zielsystem IPv6 nicht unterstützt. Deshalb Auswahl des IPv4-Protokollstacks.

Stateless IP/ICMP Translation (SIIT)

- Setzt dynamische Allokation von IPv4-Adressen voraus!
- Unterstützt die Kommunikation zwischen reinen IPv6-Systemen und reinen IPv4-Systemen
- Zustandslose Übersetzung von IPv4/IPv6- und ICMPv4/ICMPv6-Paketen
- Keine Übersetzung von Routing-Headern, Hop-by-Hop-Optionen und Destination-Optionen

Illustration von SIIT



Auswirkungen von IPv6 auf andere Protokolle

Routing-Protokolle

- Handhabung der längeren Adressen

Transportprotokolle

- Reduzierte maximale Nutzdatenlänge wegen des grösseren IP-Headers
- Neuer IP-Pseudo-Header verändert die Implementierung der Prüfsummenberechnung in UDP (nun obligatorisch) und TCP
- TCP unterstützt gegenwärtig eine IP-Adressänderung während einer bestehenden TCP-Verbindung nicht.

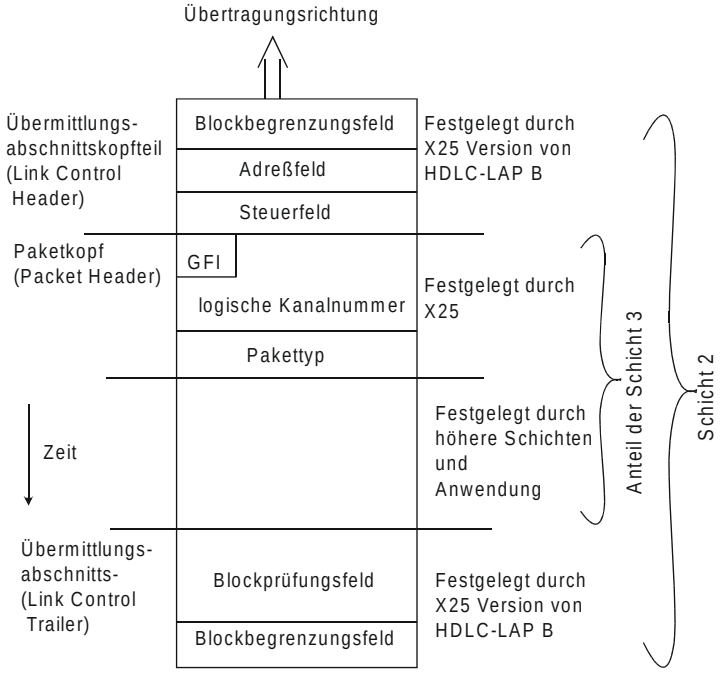
ITU-T Recommendation X.25

Der wichtigste internationale Standard für paketvermittelte Weitverkehrsnetze.

Ein Standard des CCITT (jetzt ITU-T).

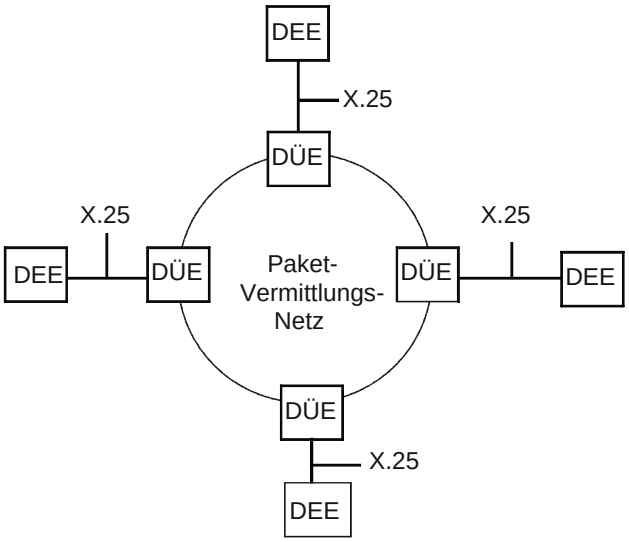
- Einsatz zur Vermittlung in öffentlichen Paketvermittlungsnetzen
- Umfasst die Schichten 1 bis 3 des Referenzmodells
- Verbindungsorientiert!
- Adressfeld: Ziel-/Herkunftsadresse des Pakets
- Steuerfeld: Unterscheidung Daten-/Steuerpakete
- Sequenznummern für Paketreihenfolge
- GFI: Kennung des Paketformats (General Format Identifier)
- **Logische Kanalnummer**: Unterscheidung verschiedener Verbindungen an einem Zugangspunkt
- Pakettypen: Auf-/Abbau der Verbindung, Daten, Interrupts, Flusssteuerung

Paketstruktur von X.25

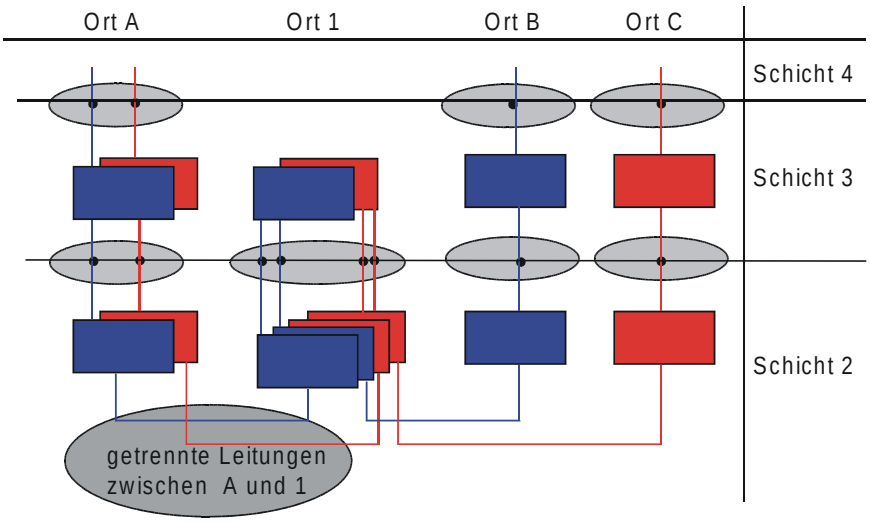


Einordnung von X.25

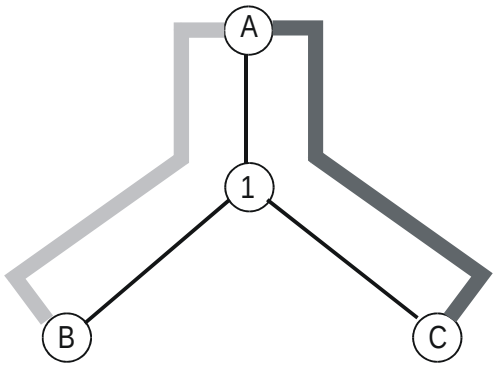
Standardisierung einer **Schnittstelle** zwischen einem privaten Endgerät (DEE=Datenendeinrichtung) und dem öffentlichen Paketvermittlungsnetz (DÜE=Datenübertragungseinrichtung).



Beispiel: Vermittlung ohne Multiplexen



Zwei Schicht 3-Verbindungen über eine Schicht 2-Verbindung

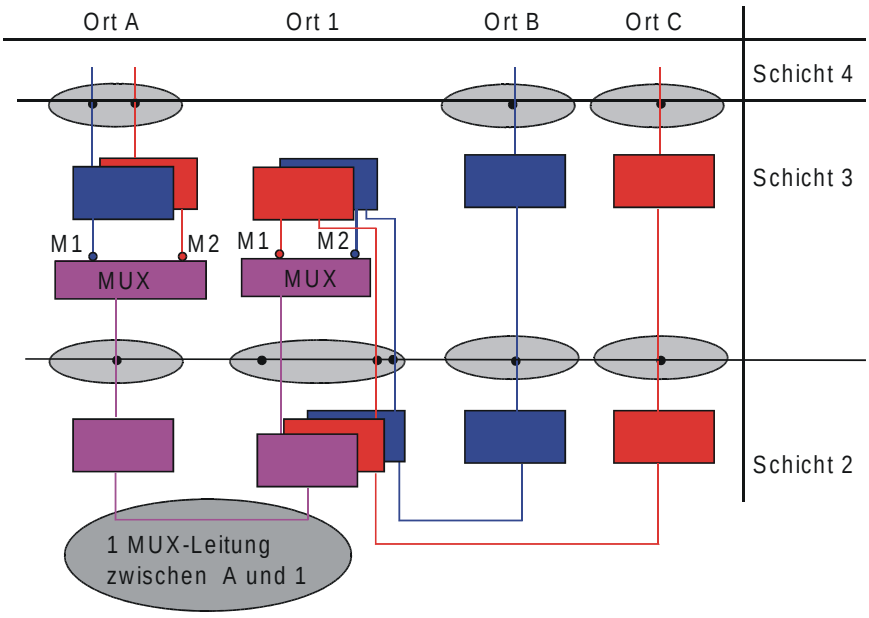


Schicht 1- und Schicht 2-Verbindungen: A1, B1, C1

Schicht 3-Verbindung AC

Schicht 3-Verbindung AB

Beispiel: Vermittlung mit Multiplexen

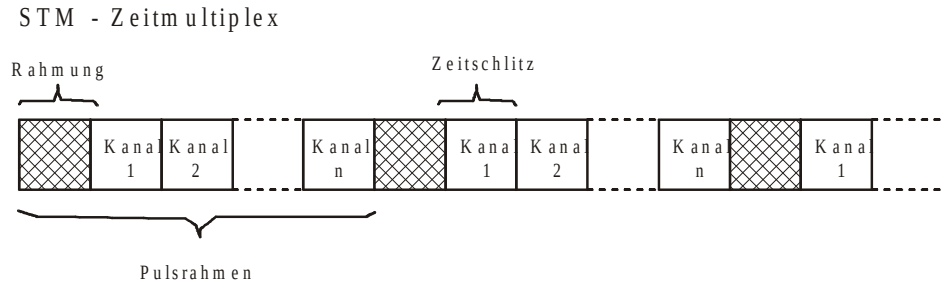


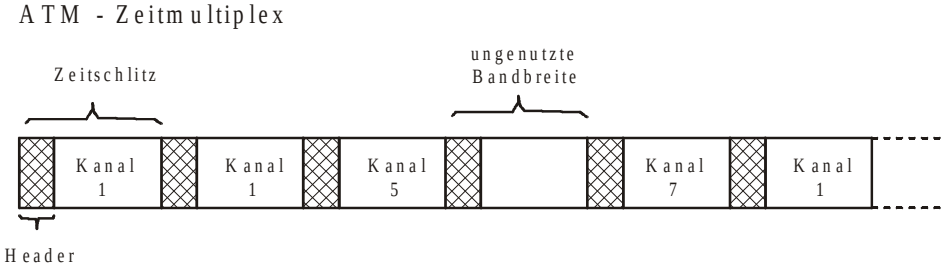
ATM (Asynchronous Transfer Mode)

Grundlagen

- Eine schnelle Paketvermittlungstechnik für Zellen fester Größe
- Basiert auf asynchronem (statistischem) Zeitmultiplexing; daher der Name ATM
- Verbindungsorientiert; unterscheidet virtuelle Pfade und virtuelle Verbindungen
- Implementierung der Vermittlungsrechner soll zwecks Erreichung hoher Zellraten möglichst weitgehend in Hardware möglich sein
- Verzicht auf Fehlererkennung, Flusskontrolle usw. in der Zellvermittlungsschicht
- Befriedigt ein breites Spektrum verschiedener Datenraten und ein breites Spektrum verschiedener Anwendungsanforderungen.

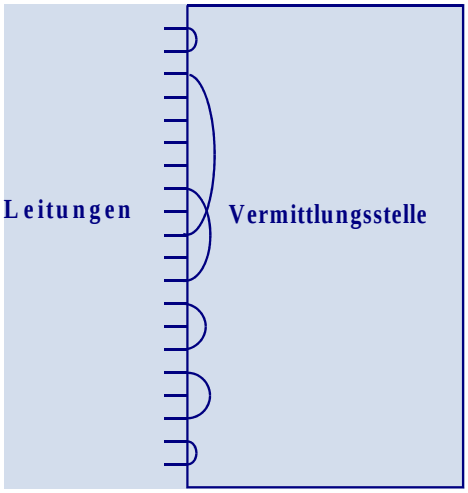
Synchrones vs. asynchrones Zeitmultiplexing



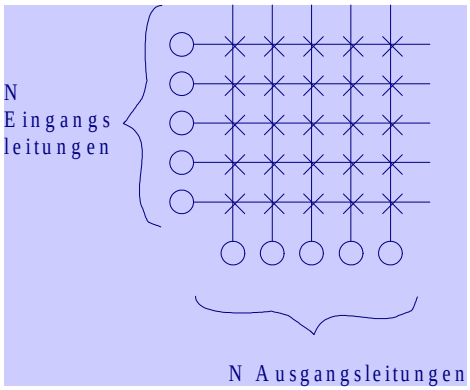


Vermittlungstechnik

Funktion einer Vermittlungsstelle (Switch), abstrakt



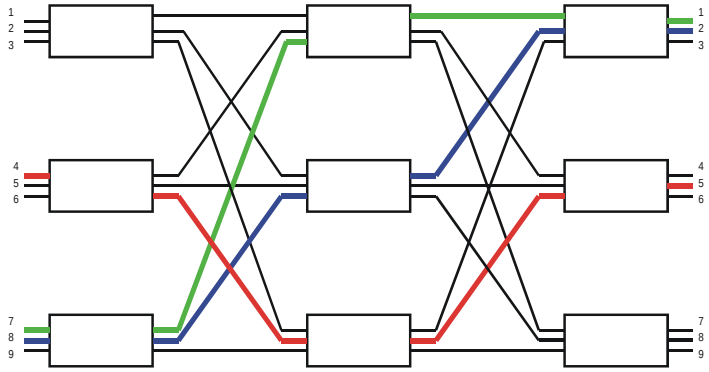
Raumvielfache (Space-Division Switch)



Nachteile einer Implementierung als Matrix:

- Anzahl der Verbindungspunkte (crosspoints) wächst mit N^2
- Defekter Verbindungspunkt macht eine bestimmte Verbindung unmöglich
- Schlechte Auslastung der Verbindungspunkte (maximal N aus N^2 in Gebrauch)

Mehrstufige Raumvielfache (multi-stage space division switches)



Vorteile

- Geringere Anzahl von Verbindungspunkten
- Mehrere alternative Pfade zur Verbindung eines Eingangs mit einem Ausgang; dadurch höhere Zuverlässigkeit

Nachteile

- Blockierung: keine Verbindungsmöglichkeit zwischen Eingang und Ausgang. Im obigen Beispiel: Eingang 9 kann mit Ausgang 4 oder 6 nicht verbunden werden!

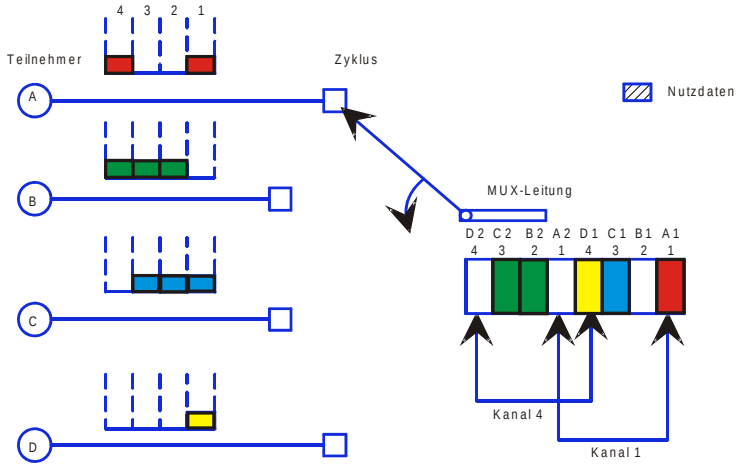
Multiplexing auf der Leitung und Vermittlungstechnik

Man kann konzeptionell unterscheiden:

- Synchrones und asynchrones Multiplexen auf der Leitung
- Synchrone und asynchrone Vermittlungstechnik

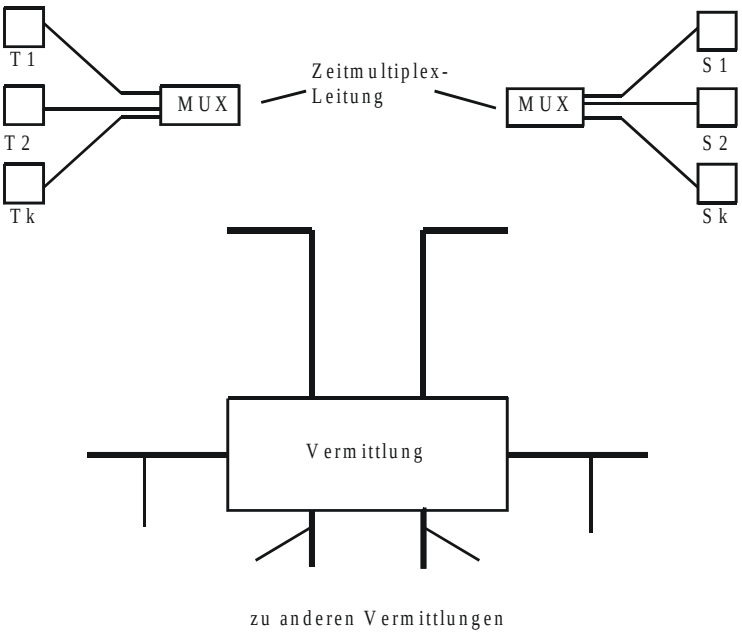
Synchrones Zeitmultiplexing

Synchronous Time-Division Multiplexing (STM)



Erkenntnis: Für **STM-Leitungen** besteht das Vermitteln im Umsortieren der Zeitschlitze!

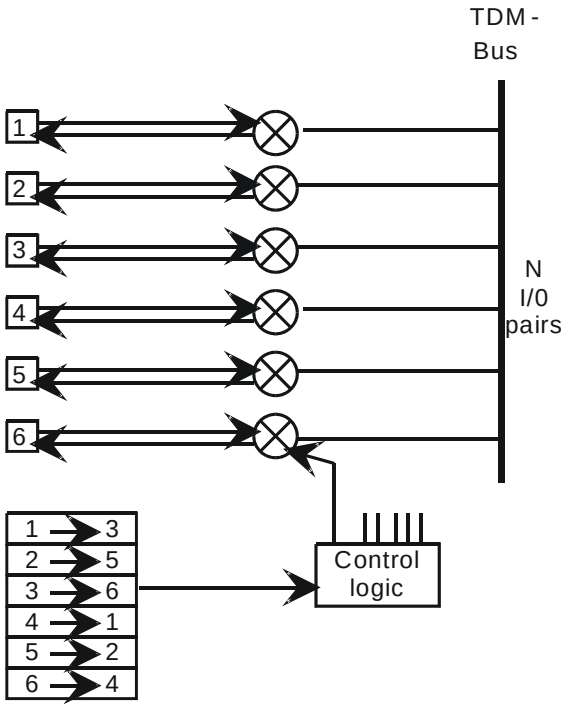
Übertragung und Vermittlung im Zeitmultiplexverfahren



Vermittlungsstelle mit internem TDM-Bus (1)

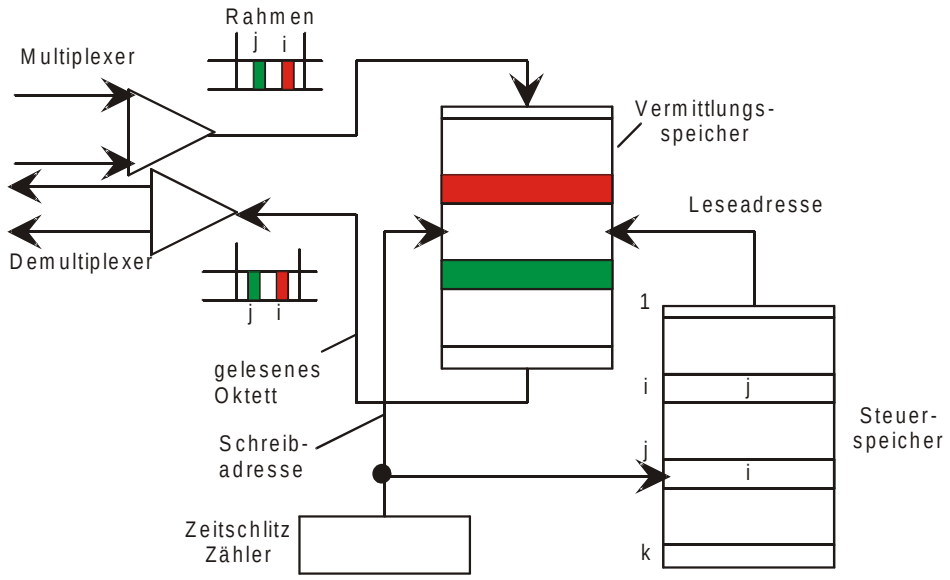
- Verwendung von STM auf einem schnellen Bus innerhalb des Vermittlungsrechners
- Jeweils eine Eingangs- und eine Ausgangsleitung werden für eine kurze Zeitperiode auf den Bus geschaltet
- Leitungspuffer dienen zum Geschwindigkeitsausgleich zwischen langsamen externen Leitungen und dem schnelleren TDM-Bus
- Nachteil: Der interne Bus muss so schnell sein wie die Summe der gleichzeitig aktiven Verbindungen

Vermittlungsstelle mit internem TDM-Bus (2)



Vermittlungsstelle mit internem Vermittlungsspeicher

"Time Slot Interchange"



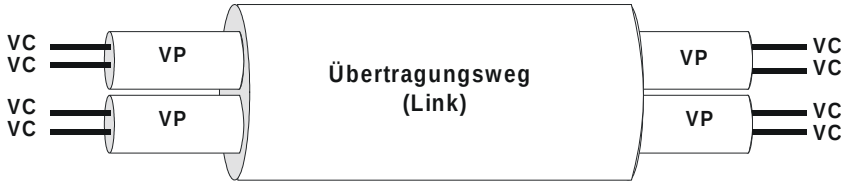
Virtuelle Kanäle und virtuelle Pfade in ATM

Virtueller Kanal (Virtual Circuit, VC):

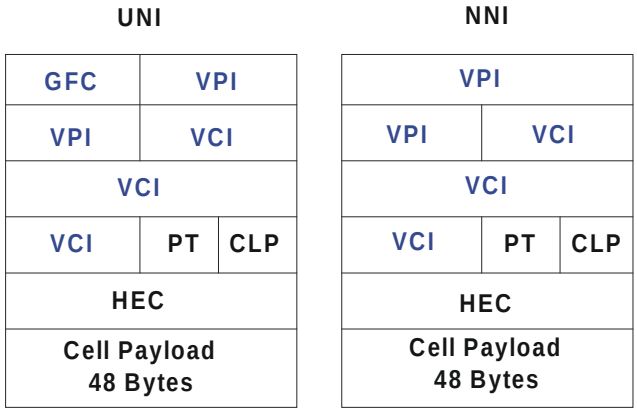
- virtuelle Verbindung zwischen ATM-Endgeräten über mehrere Übertragungsabschnitte hinweg

Virtueller Pfad (Virtual Path, VP):

- auf einer (Teil-)Strecke gebündelte VCs

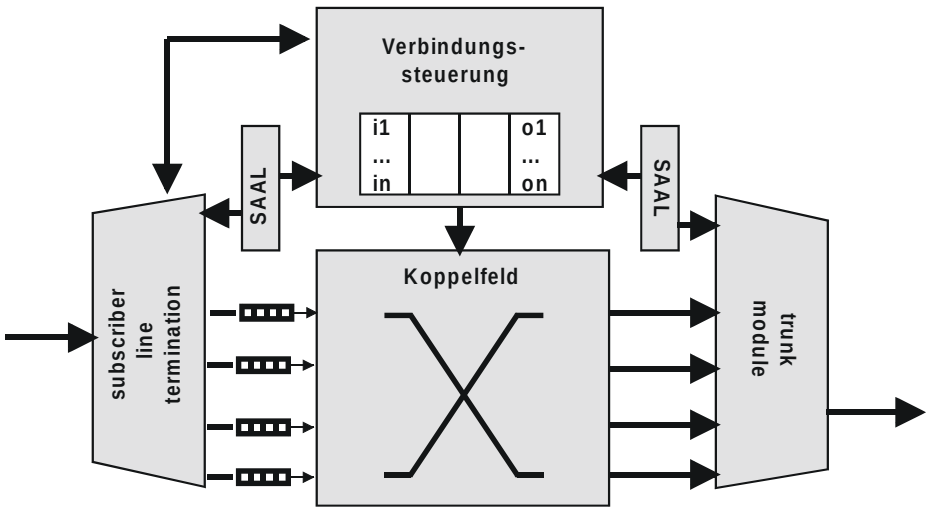


ATM-Zellformate

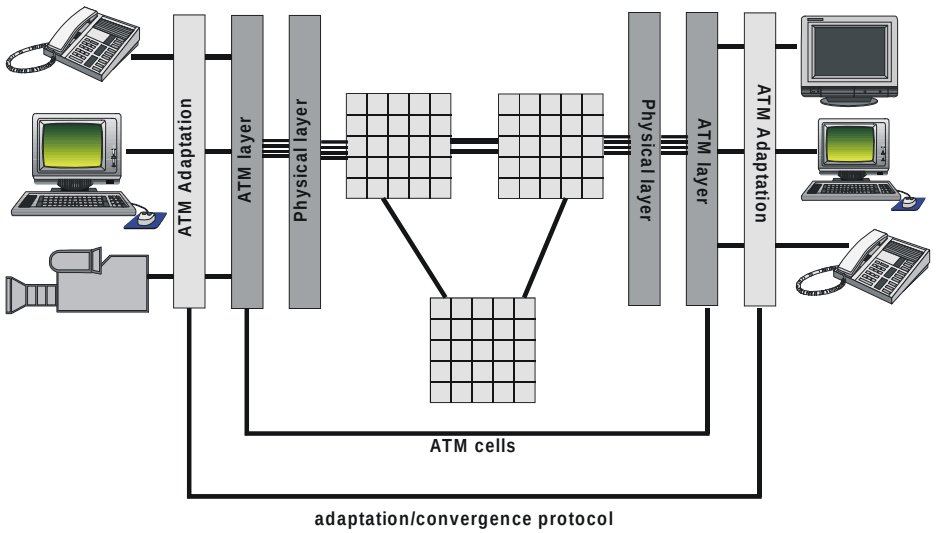


GFC: Generic Flow Control
VPI: Virtual Path Identifier
VCI: Virtual Circuit Identifier
PT: Payload Type
CLP: Cell Loss Priority
HEC: Header Error Check

Schema eines ATM-Vermittlungsknotens



ATM Adaptation Layer



ATM-Dienstklassen

	Klasse A	Klasse B	Klasse C	Klasse D
Synchronität	isochron		asynchron	
Bitrate	konstant	variabel		
Verbindungs-modus	verbindungsorientiert			verbindungslos
Anwendungen	Emulation synchroner Dienste (ISDN)	variabel bitratiges Video (MPEG, ...)	Verbindungs-orientierte Da-tenkommu-nikation	verbindungs-lose Daten-kommunikaton

ATM-Adaptionsschichten

- AAL1: Constant Bit Rate (CBR) mit Synchronisation
- AAL2: Variable Bit Rate (VBR)
- AAL3/4: für Datenverkehr, überwiegend in öffentlichen Netzen
- AAL5: Standard-AAL für Datenverkehr

AALs beschreiben Ende-zu-Ende-Protokolle. Weitere AALs können definiert werden, ohne dass dies die ATM-Zellvermittlungsschicht betrifft.

ATM-Verkehrsklassen

UBR: Unspecified Bitrate

- Für Datenanwendungen, nutzt verfügbare (Rest-) Bandbreite
- Keine „admission control“ und kein „Policing“
- Bei Überlast hohe Zellverluste

CBR: Constant Bitrate

- Für „Circuit Emulation Services“ mit festen PCR, CTD, CDV
- Minimale Zellverlustrate

VBR: Variable Bitrate

- Zum Beispiel für komprimierte Videoströme mit variabler Bitrate

ABR: Available Bitrate

- Zuverlässige Übertragung für Datenanwendungen
- Implementiert eine Flussregelung im ATM-Netz

ATM-Verkehrsvertrag

- Benutzer und Netz schließen beim Verbindungsaufbau einen *„Verkehrsvertrag“*
- Der Benutzer liefert den beim Verbindungsaufbau spezifizierten Verkehr *(Verkehrsbeschreibung, traffic description)*
- Der Benutzer spezifiziert die für diese Verkehrsbeschreibung gewünschten Dienstqualitäten *(QoS: Quality of Service)*
- Das Netz prüft, ob der spezifizierte Verkehr mit der gewünschten Qualität noch transportiert werden kann *(admission control)*
- Das Netz kontrolliert während der Verbindung die Einhaltung der Verkehrsbeschreibung am Netzzugang *(UPC: usage parameter control oder source policing)*
- Nicht konforme Zellen werden:
  - am Netzeingang mit CLP=1 gekennzeichnet
  - Zellen mit CLP=1 werden bei Überlast am Netzeingang oder im Inneren des Netzes verworfen.

Verkehrsparameter

Verkehrsbeschreibung

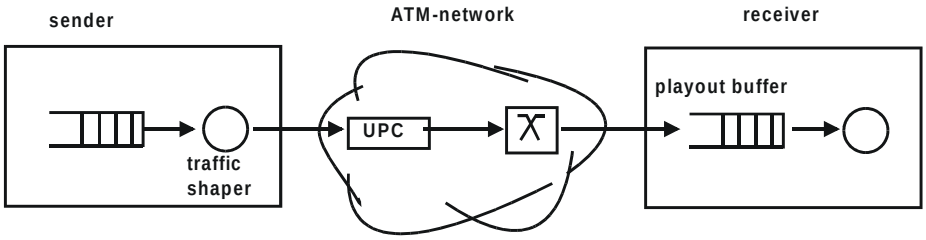
- **PCR**: Peak Cell Rate (cells/s)
- **SCR**: Sustainable Cell Rate (cells/s)
- **MBS**: Maximum Burst Size (cells), auch spezifiziert als **BT**: Burst Tolerance = (MBS-1)/(1/SCR-1/PCR)
- **MCR**: Minimum Cell Rate (nur für ABR)

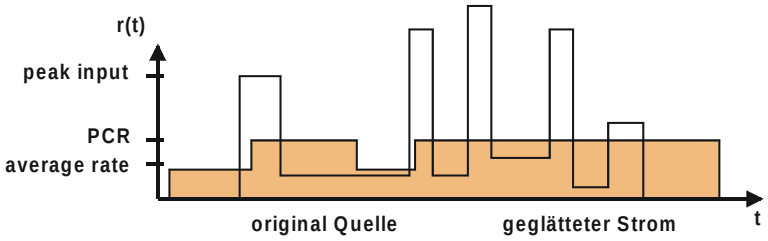
Dienstqualitäten (QoS-Parameter)

- **CLR**: Cell Loss Ratio (Anzahl der verlorenen Zellen/Anzahl der gesendeten Zellen)
- **CTD**: Cell Transfer Delay (vom Netzzugang bis zur Ablieferung beim Empfänger)
- **CDV**: Cell Delay Variation (CTD variance) (Delay Jitter)

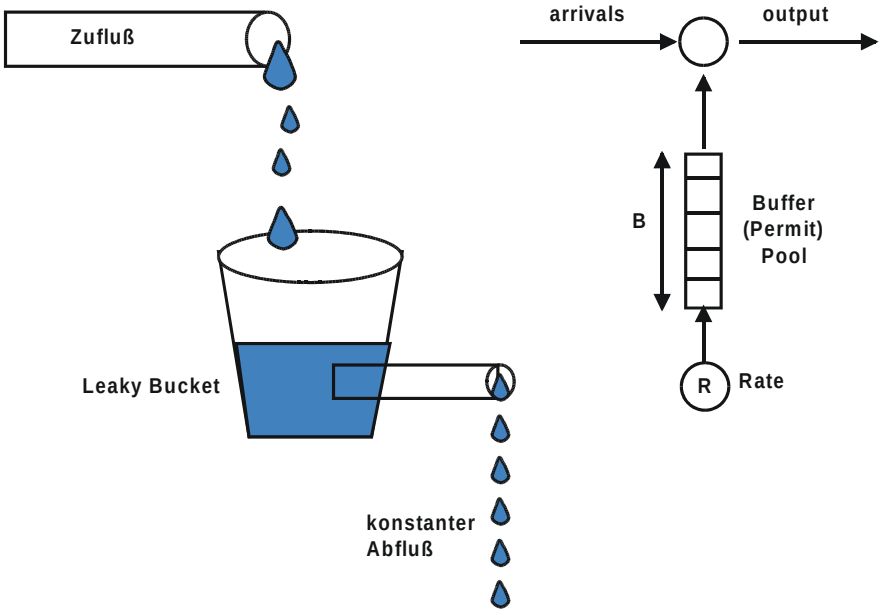
Traffic Shaping

Das ATM-Endgerät formt den Datenverkehr, um den Verkehrsvertrag einzuhalten (“traffic shaping“).

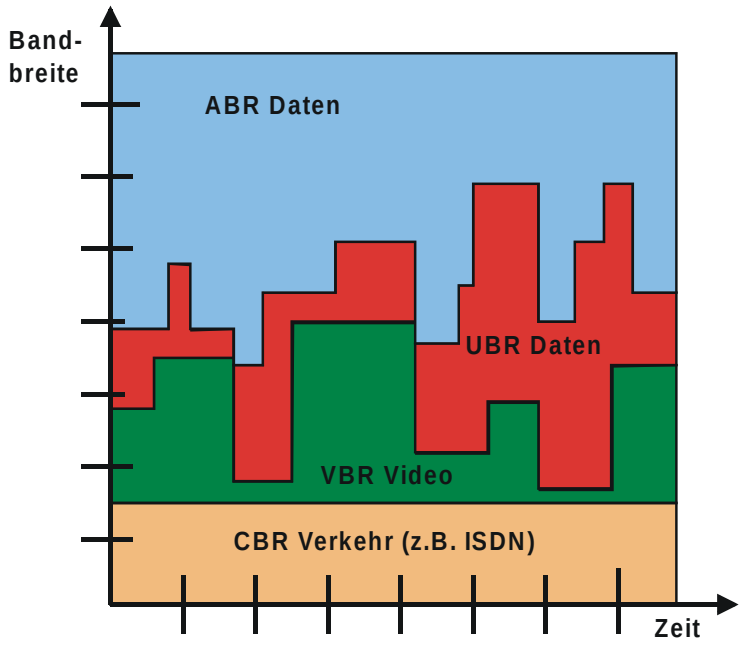




Verkehrsformung durch "Leaky Bucket"



Bandbreitenaufteilung für die Verkehrsklassen



6. ISDN - Integrated Services Digital Network

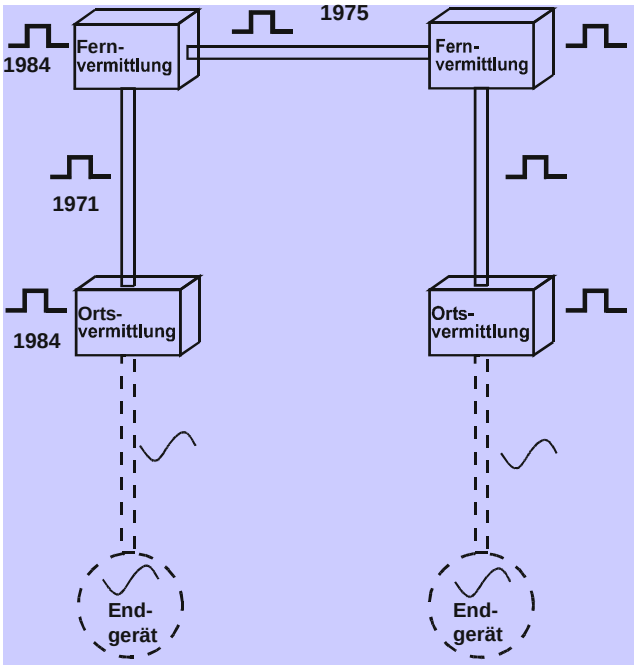
6.1 Ziele von ISDN

- Integration existierender Telekommunikationsdienste: Sprache, Text, Daten, Bilder, Teletex, Telefax
- Vollständige Digitalisierung des Netzes von Endgerät bis Endgerät
- Anbieten von neuen digitalen Kommunikationsdiensten

6.2 Grundlagen von ISDN

- Kanalvermittelte Punkt-zu-Punkt-Verbindungen
- Standardschnittstelle für das Endgerät:
  - zwei B-Kanäle mit je 64 kbit/s
  - ein D-Kanal zum Signalisieren mit 16 kbit/s
- Nur eine Klasse von Prozeduren zum Verbindungsaufbau und -abbau für alle Dienstarten
- Alle Dienste können unter derselben Adresse erreicht werden.

Entwicklung von ISDN aus existierenden Netzen (1)



Entwicklung von ISDN aus existierenden Netzen (2)

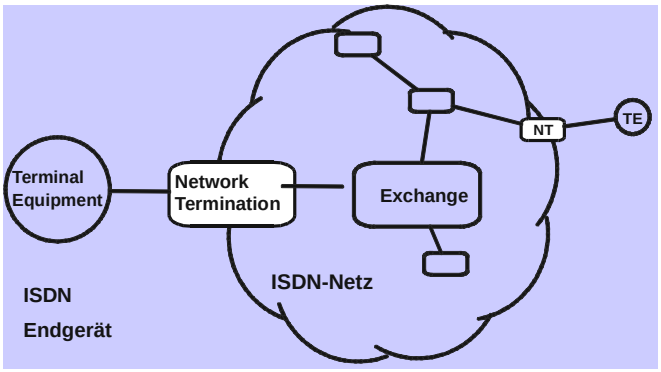
Im Telefonnetz waren häufig schon verfügbar:

- digitale Leitungen zwischen den Vermittlungsstellen
- digitale Vermittlungsstellen

Neu in ISDN:

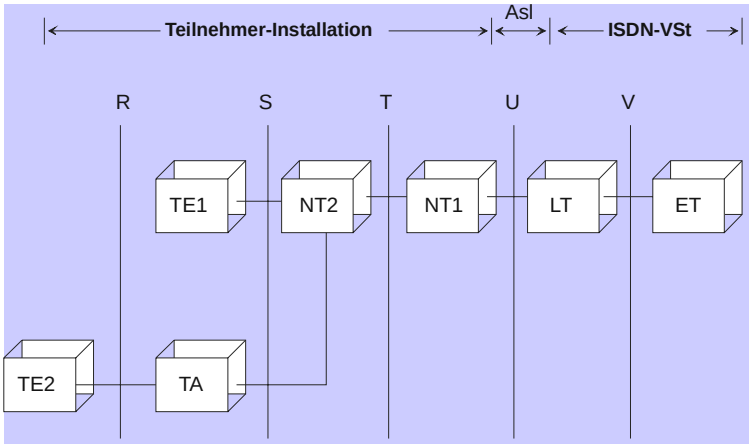
- digitale Endgeräte
- digitale Übertragung zwischen der Ortsvermittlungsstelle und dem Endgerät

ISDN-Netzkomponenten im Überblick



TE = Terminal Equipment
NT = Network Termination
Exchange = ISDN-Vermittlungsstelle

ISDN-Referenzpunkte



Asl: Anschlussleitung
VSt: Vermittlungsstelle
TE1: ISDN Terminal Equipment
TE2: Pre-ISDN Terminal Equipment
TA: Terminal Adapter

NT1: Network Termination 1
NT2: Network Termination 2
LT: Line Termination
ET: Exchange Termination

ISDN-Kanaltypen

D-Kanal für die Signalisierung ("out-of-band signalling")

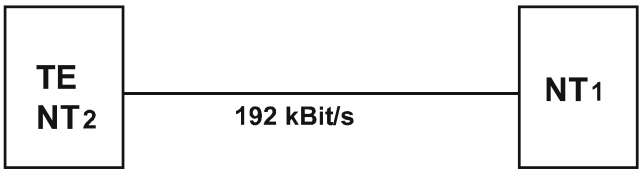
- Steuerung aller B-/H-Kanäle
- Verbindungsaufbau
- Verbindungsabbau
- Vermittlungsdienste
- Der D-Kanal ist unabhängig von der Benutzung des B/H-Kanals

Benutzerkanäle

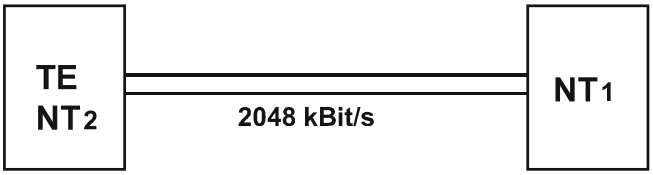
- B-Kanäle (je 64 kbit/s)
- H-Kanäle (384 kbit/s, 1920 kbit/s, 130/155 Mbit/s)

Schnittstellentypen für ISDN-Netzbenutzer

Basisanschluss



Primärratenanschluss



ISDN-Basisanschluss

Kanäle

- 1 D-Kanal (16 kbit/s)
- 2 B-Kanäle (jeder 64 kbit/s)

Konfiguration beim Teilnehmer

- passiver Bus für bis zu acht Endgeräte

ISDN-Primärratenanschluss

Kanäle

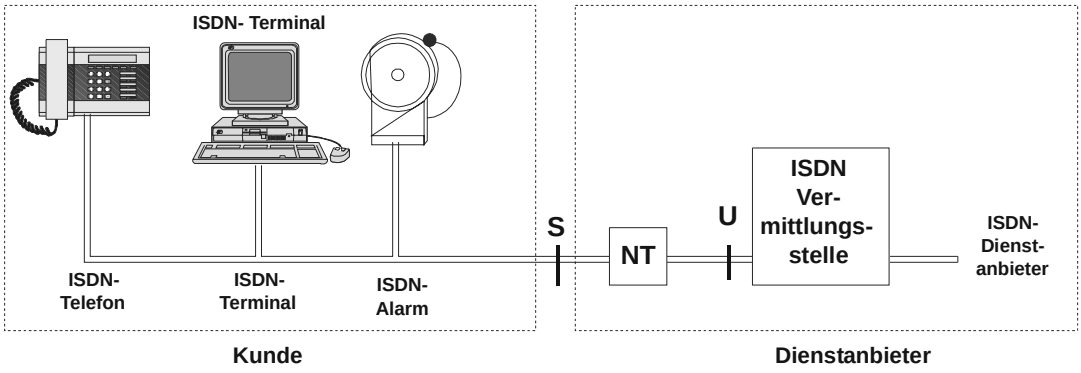
- 1 D-Kanal (64 kbit/s)
- 30 B-Kanäle (jeder 64 kbit/s)

Konfiguration beim Teilnehmer

- Punkt-zu-Punkt

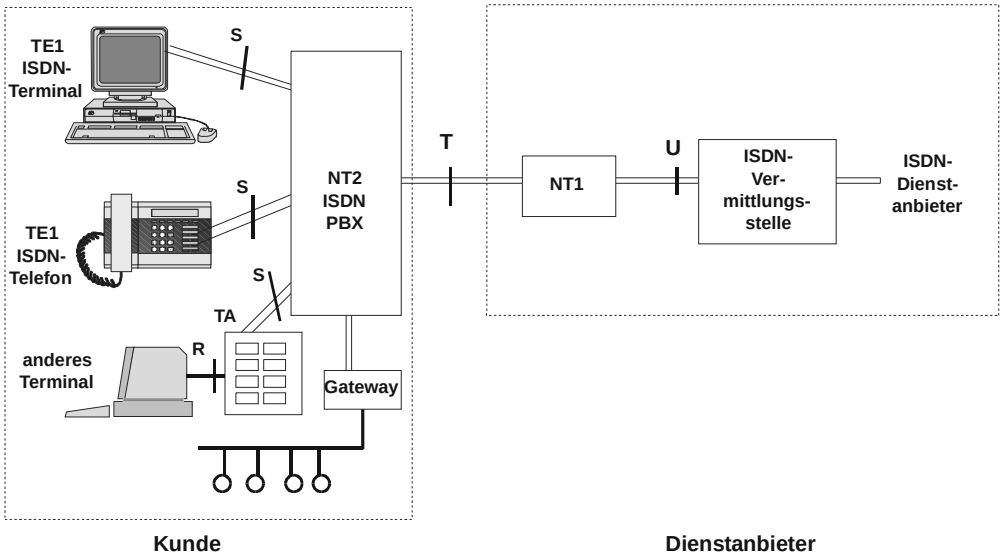
Beispiel 1 für ein ISDN-System

ISDN-System für den Hausgebrauch



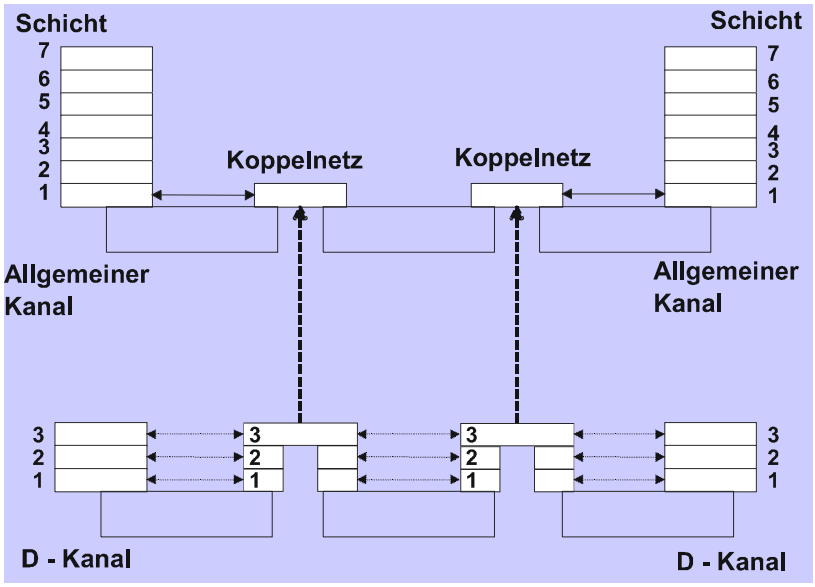
Beispiel 2 für ein ISDN-System

ISDN-Anlage mit PBX für ein größeres Unternehmen



6.3 Schichten 1, 2 und 3 für ISDN

Architektur von B-Kanal und D-Kanal



Definition der Schichten im Standard

Schicht 1 ist definiert für die Benutzer-Kanäle (B und H) und für den D-Kanal. Sie regelt Leitungscode und Multiplexing auf der Zweidrahtleitung.

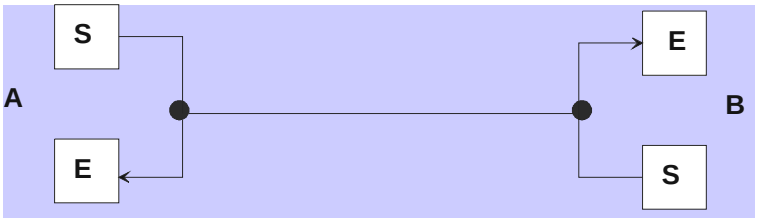
Schichten 2 und 3 sind nur für den D-Kanal definiert. Sie standardisieren Paketdienste für die Signalisierung.

Bitübertragungsschicht (für alle Kanäle)

Zweidraht-Duplexübertragung

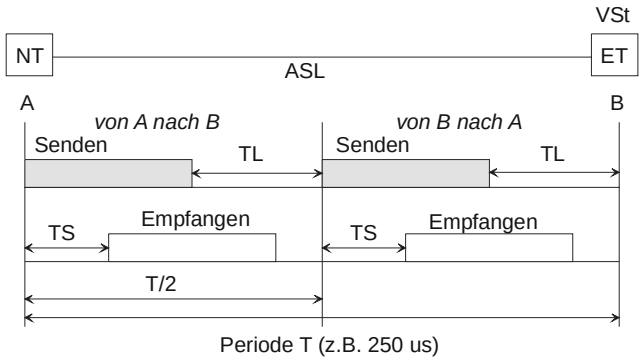
Die Signale beider Übertragungsrichtungen werden auf demselben Adernpaar übertragen!

S = Sender, E = Empfänger



Mögliche Alternativen für die Vollduplex-Übertragung

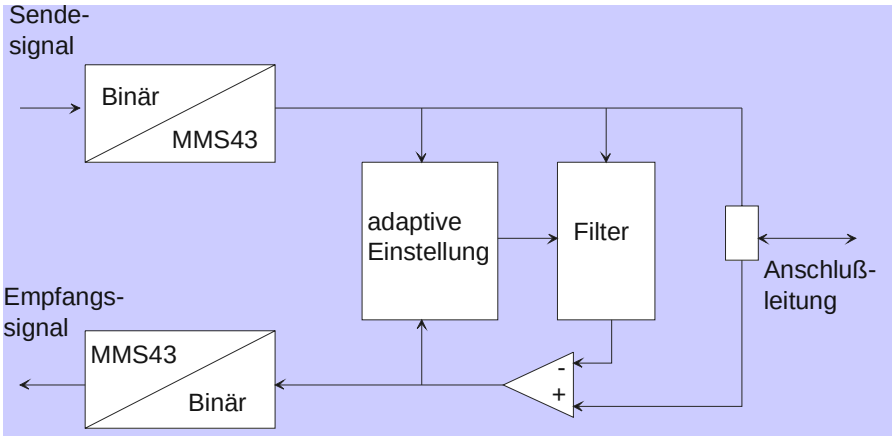
Vollduplex durch Time Division Multiplexing



ASL = Anschlussleitung
ET = Vermittlungsabschluss (exchange termination)
NT = Netzabschluss (network termination)
TL = Zeitlücke
TS = Signallaufzeit
VSt = Vermittlungsstelle
Zu jedem Zeitpunkt kann entweder A oder B senden.

Vollduplex mit Echokompensation

Grundsätzliche Struktur der Einrichtung zur Übertragung eines 160 kbit/s-Signals über Anschlussleitungen



Rahmenformat der Basisrate (1)

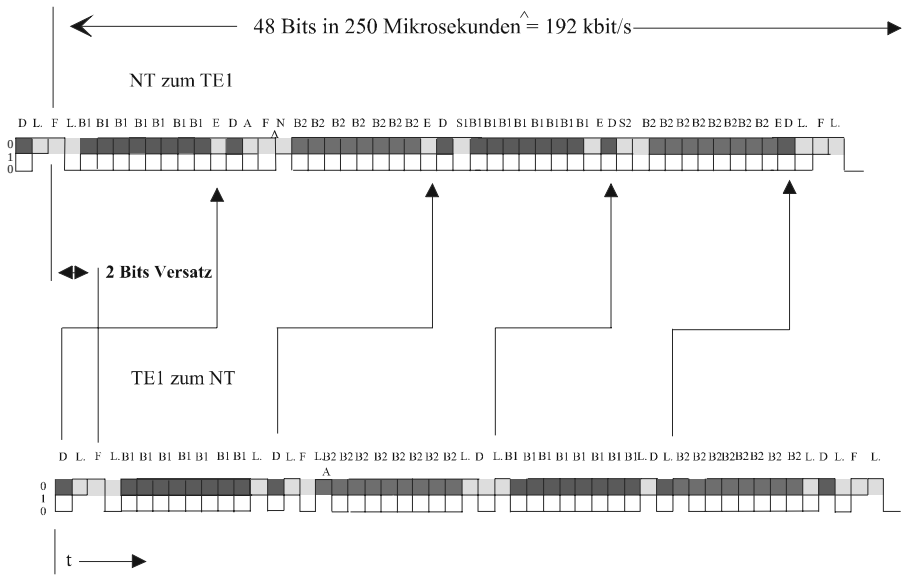
Rahmenformat

- Rahmenlänge: 48 Bits
- D-Kanal: 4 Bits pro Rahmen (16 kbit/s)
- B-Kanal: 16 Bits pro Rahmen (64 kbit/s)
- Es werden 4000 Rahmen pro Sekunde übertragen
- Brutto-Bitrate: 192 kbit/s

Merke

- Vollduplex durch das Echokompensationsverfahren
- Drei Kanäle (2 B + D) simultan durch Time Division Multiplexing

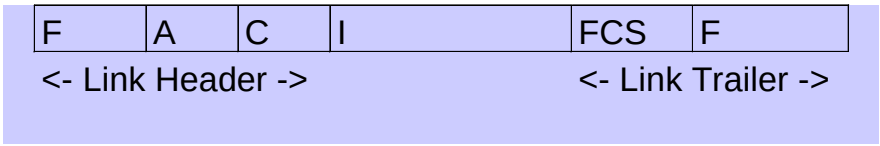
Rahmenformat der Basisrate (2)



B1-Bits = rot D-Bits = grün

B2-Bits = blau Management-Bits = gelb

Rahmen der Schicht 2 des D-Kanals



F: Flag
A: Address
C: Control
I: Information
FCS: Frame Check Sequence

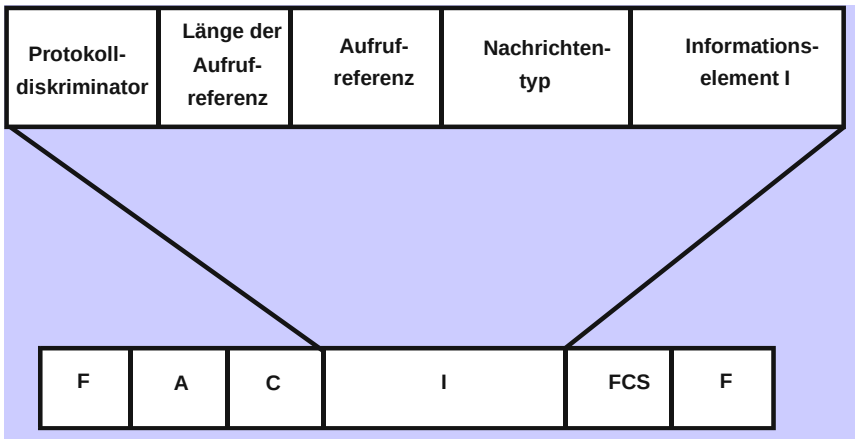
=> Identisch mit dem HDLC-Rahmen!

Vermittlungsschicht (Schicht 3) des D-Kanals

Eine Protokolldateneinheit der Schicht 3 des D-Kanals stellt eine Signalisiernachricht dar. Sie besteht aus

- Protokolldiskriminator
- Aufrufreferenz (identifiziert eine Netzverbindung aus der Menge aller Netzverbindungen, die auf eine D-Schicht-2-Verbindung gemultiplexed wurden)
- Nachrichtentyp
- einem oder mehreren Informationselementen, je nach Typ

PDU-Format der Schicht 3 des D-Kanals



6.4 ISDN-Standards

Nationales (deutsches) ISDN vs. Euro-ISDN

- Nationales ISDN seit 1987/88 in Deutschland verfügbar
- Euro-ISDN seit 1993 implementiert - grenzüberschreitende Verbindungen ohne Protokollkonverter, Endgeräte international vermarktbar
- Unterschiede:
  - Neue Anschlusstechnik (Stecker)
  - Veränderungen im D-Kanal Protokoll in den Schichten 2 und 3

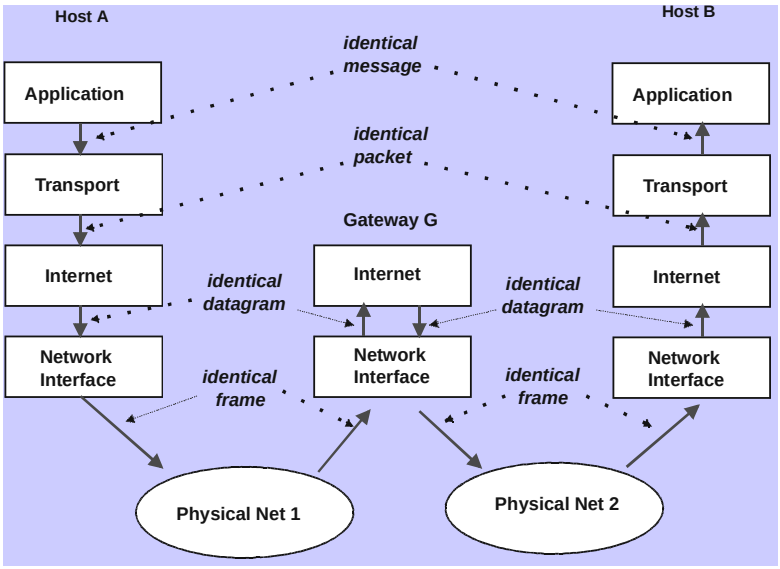
Empfehlungen der ITU-T zu ISDN (Auswahl)

- **I-Series-ISDN**
  - I.100 — General concept, terminology etc.
  - I.200 — Service aspects (bearer and teleservices)
  - I.300 — Network aspects (including reference model)
  - I.400 — User network interface aspects
  - **I.430/I.431 — Layer 1**
  - **I.440/I.441 — Layer 2**
  - **I.450/I.451 — Layer 3**
  - I.460/I.464 — Support of existing interfaces
    — I.462 Support of packet mode terminals
  - I.500 — Internetwork interface
  - I.600 — Maintenance principles
- G-Series — Transmission systems, circuits, media
- G.701 - G.956 Digital networks
- M-Series — Maintenance
- Q-Series — Telephone switching and signalling
- **Q.700 Signalling System No 7**

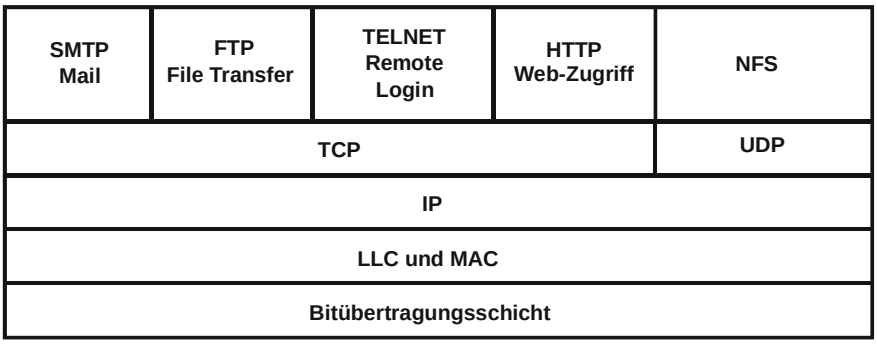
7. Die Transportschicht

7.1 Architektur der Transportprotokolle im Internet

Die Transportprotokolle im Internet sind **Ende-zu-Ende-**Protokolle!



Wichtige INTERNET-Protokolle



SMTP	=	Simple Mail Transfer Protocol
FTP	=	File Transfer Protocol
TELNET	=	Remote Login Protocol
UDP	=	User Datagram Protocol
NFS	=	Network File System
TCP	=	Transmission Protocol
IP	=	Internet Protocol
LLC	=	Logical Link Control
MAC	=	Media Access Control

Adressierung von Dienst-Prozessen: Ports

- Die Schicht-4-Adresse soll einen bestimmten Dienst (Anwendungstyp) identifizieren, dem ein Anwendungsprozess zugeordnet ist
- Eine Adressierung mittels Prozess-Nummer wäre ungeeignet, denn Prozesse werden dynamisch erzeugt und beendet; deshalb ist die Prozess-Nummer außerhalb des lokalen Systems nicht bekannt.
- Die Zuordnung Dienst <=> Prozess ist nicht fest:
  - Ein Prozess kann mehrere Dienste erbringen.
  - Mehrere Prozesse können denselben Dienst erbringen.

=> Einführung eines abstrakten Kommunikations-Endpunktes: **Port**

Eigenschaften der Ports

- Ein Dienst ist genau einem Port zugeordnet.
- Über denselben Port können mehrere Verbindungen gleichzeitig laufen.
- Asynchroner und synchroner Port-Zugriff sind möglich.
- Jeder Port ist mit einem Puffer assoziiert.
- Der Port stellt eine Programmierschnittstelle für den Anwendungsprogrammierer zur Verfügung (API).

Beispiele für reservierte Port-Nummern

Einige Port-Zuordnungen ("well-known addresses")

Dezimal	Schlüssel-wort	Unix Schlüssel-wort	Beschreibung
0			Reserved
.	.	.	.
.	.	.	.
20	FTP-DATA	ftp-data	File Transfer Protocol (data)
21	FTP	ftp	File Transfer Protocol
23	TELNET	telnet	Terminal Con-nection
25	SMTP	smtp	Simple Mail Transfer Proto-col
42	NAME-SERVER	name	Host Name Server
43	NICNAME	whois	Who Is

7.2 UDP (User Datagram Protocol)

UDP ist ein

- unzuverlässiges
- verbindungsloses

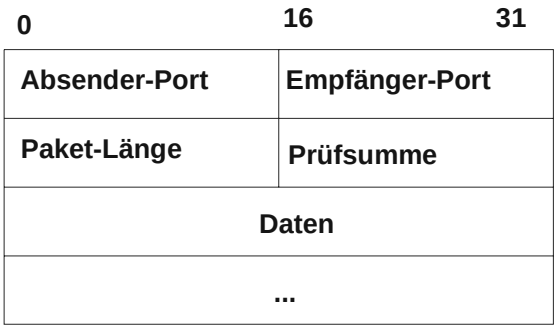
Datagramm-Transport-Protokoll im Internet. Es dient im Wesentlichen dazu, den Anwendungen eine Programmierschnittstelle für den Direktzugriff auf IP, ergänzt um die Port-Adressierung, anzubieten.

Eigenschaften

- Datagramm-Transport
  - Keine Garantie über das Einhalten der Reihenfolge der einzelnen Pakete
  - Keine gesicherte Zustellung der Pakete an den Empfänger
  - Duplizierte Pakete sind möglich
- Multicast ist möglich

Format von UDP-Paketen

UDP-Header



- Paket-Länge wird in Bytes angegeben (einschließlich UDP-Header)
- Prüfsumme über Header und Daten zur Fehlererkennung. Die Prüfsumme wird über das gesamte Fragment berechnet, inklusive UDP-Header. Es wird ein spaltenweises EXOR über die „senkrecht untereinander geschriebenen“ 16-Bit-Worte des Pakets berechnet.
- Die Verwendung der Prüfsumme ist optional (wenn IPv4 darunter liegt)

Eigenschaften von UDP

- Geringer Ressourcen-Verbrauch (Speicherplatz, CPU-Zeit)
- Kein expliziter Verbindungsaufbau
- Einfache Implementierung

UDP ist vor allem für einfache Client-Server-Interaktionen geeignet, zum Beispiel für Request-Response-Protokolle:

- ein Anfrage-Paket vom Client zum Server
- ein Antwort-Paket vom Server zum Client

Anwendungsbeispiele für UDP

- Domain Name Service (DNS)
- SNMP: Simple Network Management Protocol
- NFS: Network File System
- viele Multimedia-Protokolle, die keine Fehlersicherung in Schicht 4 wollen.
- alle Multicast-Protokolle, insbesondere auch RTP für Realzeit-Anwendungen, vor allem Multimedia-Anwendungen (Audio- und Videoströme)

7.3 TCP (Transmission Control Protocol)

TCP ist das erste Protokoll in der Internet-Protokollhierarchie, das eine **gesicherte Datenübertragung zwischen Endsystemen** leistet.

Eigenschaften

- Datenstrom-orientiert: TCP überträgt einen seriellen Bit-Strom der Anwendung in Form von 8-Bit Bytes.
- **Verbindungsorientierung**: Vor der Datenübertragung wird eine Verbindung zwischen beiden Kommunikationspartnern aufgebaut, die fehlergesichert und Reihenfolge erhaltend ist.
- Gepufferte Datenübertragung: Der sequenzielle Datenstrom wird zur Übertragung in einzelne Segmente (Pakete) aufgeteilt.
- Duplex-Kommunikation: Über eine TCP-Verbindung können gleichzeitig Daten in beide Richtungen übertragen werden.

Was steht im TCP-Standard?

Der TCP-Standard (RFC 793) spezifiziert

- Formate von Datenpaketen und Kontrollinformationen
- Prozeduren für
  - Verbindungsaufbau und -abbau
  - Fehlererkennung und -behebung
  - Flusskontrolle
  - Netzwerk-Überlastkontrolle (Congestion Control)

Der RFC 793 spezifiziert **nicht** die Schnittstelle zum Anwendungsprogramm (socket).

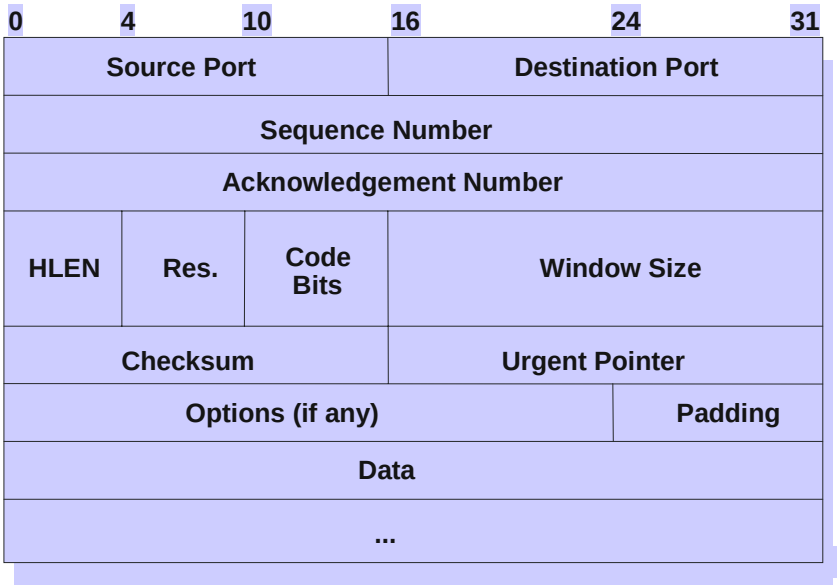
Adressierung

Eine TCP-Verbindung ist eindeutig bestimmt durch ein Quintupel aus

- IP-Adressen von Sender und Empfänger
- Port-Adressen von Sender und Empfänger
- TCP-Protokoll-Identifikator

Paketformat

Format des TCP-Headers



Datenfelder im TCP-Header (1)

SEQUENCE NUMBER **ACKNOWLEDGMENT**	Bytenummern
HLEN	headerlänge = Offset des Datenfeldes
CODEBITS	(6 Bits von links nach rechts)
URG	Urgent Pointer wird verwendet
ACK	Ack-Nummernfeld ist gültig
PSH	Push
RST	Reset der Verbindung
SYN	Synchronisiere Sequenznummern
FIN	Ende des Datenstroms
WINDOW SIZE	Fenstergröße in Bytes
URGENT POINTER	Byteoffset zur aktuellen Sequenz-nummer, an der wichtige Daten beginnen

Datenfelder im TCP-Header (2)

- **Port-Nummern:** wie für UDP
- **sequence number:** identifiziert das erste Byte im Datenteil des Paketes
- **acknowledgement number:** identifiziert das nächste Byte im Datenstrom, das der Sender dieses Paketes vom Empfänger dieses Paketes erwartet. Nicht einzelne Datenpakete, sondern Byte-Positionen im Datenstrom werden bestätigt. Dadurch ist Kumulation von Bestätigungen über mehrere TCP-Pakete leicht möglich. Die acknowledgements steuern die Übertragungswiederholung im Fehlerfall. Sie definieren auch das Schiebefenster für die Flusskontrolle.

  Da Daten in beide Richtungen zwischen den Kommunikationspartnern fließen können (Duplex-Betrieb), gibt es eine eigene Sequenznummernfolge für jede Richtung.
- **header length (HLEN):** Größe des TCP-Headers in 32-Bit-Worten

Datenfelder im TCP-Header (3)

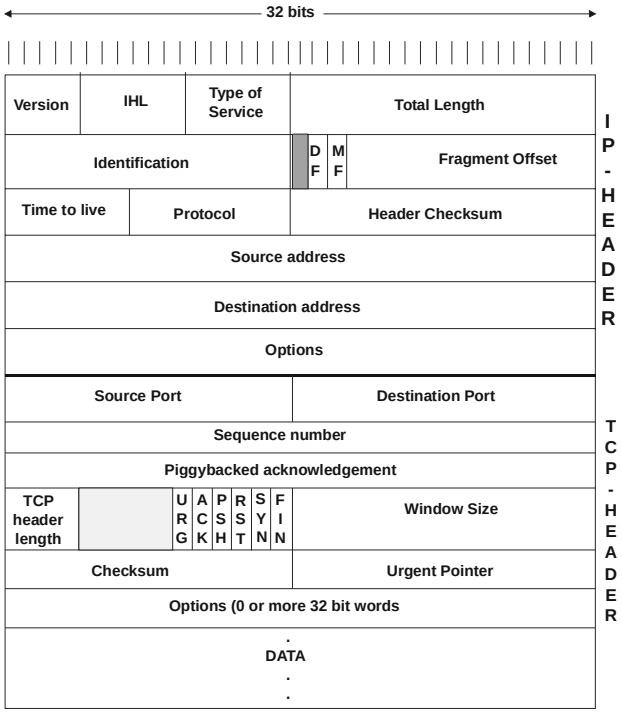
- **Codebits** (Flags)
  - URG: urgent pointer Feld ist gültig (“in Benutzung“)
  - ACK: acknowledgement Feld ist gültig (“in Benutzung“)
  - PSH: (push) Der Empfänger soll die Daten der Anwendung so schnell wie möglich zur Verfügung stellen.
  - RST: Reset der Verbindung
  - SYN: Synchronisation der initialen Sequenznummern bei Verbindungsaufbau
  - FIN: der Sender hat das Senden seiner Daten beendet.
- **Window Size:** Anzahl der Bytes, die der Sender dieses Paketes noch entgegen nehmen kann, bis sein Puffer voll ist (Flusskontrolle)

Datenfelder im TCP-Header (4)

- **checksum:** Die Prüfsumme wird über das gesamte Fragment berechnet, inklusive TCP-Header. Es wird ein spaltenweises EXOR über die „senkrecht untereinander geschriebenen“ 16-Bit-Worte des Pakets berechnet.
- **urgent pointer:** Der “urgent pointer“ identifiziert das letzte Byte im Datenteil, welches mit besonderer Priorität bearbeitet werden sollte. Die danach folgenden Daten haben normale Priorität.
- **options:** (werden wir später besprechen)

Der Datenteil eines TCP-Paketes ist optional, ein leeres TCP-Paket kann zum Beispiel als reine Bestätigung empfangener Daten gesendet werden, wenn keine Daten in die Rückrichtung gesendet werden müssen.

TCP/IP: Format des gesamten Headers

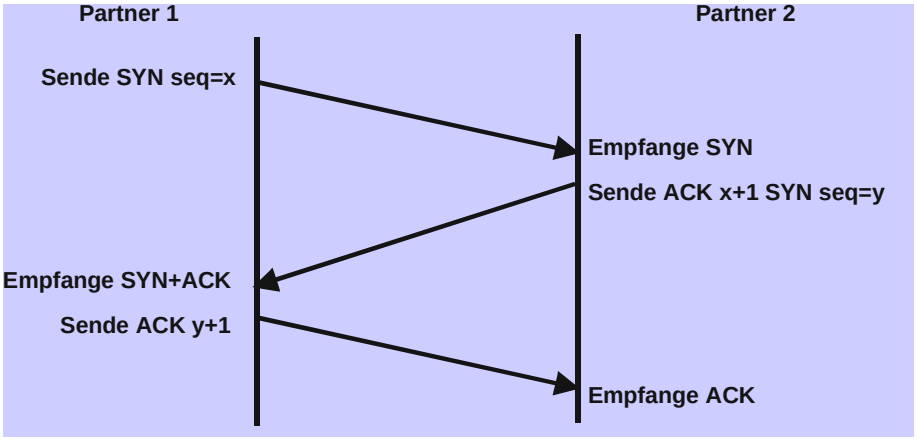


Wichtige Funktionen in TCP

- Positive Bestätigung ("positive acknowledgement or retransmission„, **PAR**)
- Übertragungswiederholung nach Ablauf einer Wartezeit (Timeout beim Sender). Keine NACKs!
- „Sliding Window"-Mechanismus zur Flusskontrolle. Variable Fenstergröße: bei jeder Bestätigung durch den Empfänger wird die ab sofort zu verwendende Fenstergröße mit übertragen.
- Fehlererkennung durch Prüfsumme (ähnlich CRC)
- Piggybacking: Kontrollinformation auf dem Rückweg und Daten können im gleichen TCP-Paket übertragen werden.
- Out-of-Band Data: wichtige Information soll dem Empfänger zugestellt werden, bevor er früher gesendete Daten verarbeitet. Beispiel: Alarm-Meldungen

TCP-Verbindungsaufbau

Three-Way-Handshake: Verbindungsaufbau durch drei Pakete:



Beim Verbindungsaufbau werden auch die initialen Sequenznummern beider Seiten (beider Richtungen) ausgetauscht und bestätigt.

Three-Way-Handshake

- Das SYN-Flag zeigt an, dass die Sequenznummern synchronisiert werden sollen.
- SYN „kostet“ ein Byte bezüglich der Sequenznummernvergabe.
- Reine acknowledgements „kosten“ keine Bytes.

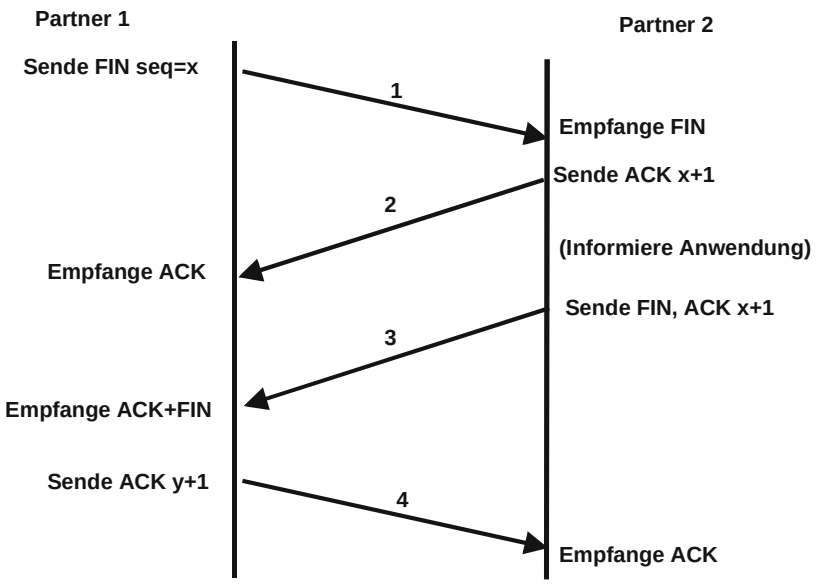
Timeout beim Verbindungsaufbau

Was passiert, wenn der Kommunikationspartner nicht antwortet?

- Die Übertragung des Paketes wird wiederholt, TCP betrachtet dies als gewöhnlichen Paketverlust.
- Nach einer festen Zeit (timeout) wird der Verbindungsversuch abgebrochen und die Anwendung informiert.

TCP-Verbindungsabbau (1)

Geordneter Verbindungsabbau durch vier Pakete:



TCP-Verbindungsabbau (2)

Abbau einer Verbindung durch zweimal „half-close“:

- Da die TCP-Verbindung bidirektional ist, sollten prinzipiell beide Richtungen getrennt voneinander beendet werden.
- Wer seine Senderolle beenden möchte, setzt das FIN-Flag.
- FIN „kostet“ ein Byte und wird daher durch ein acknowledgement bestätigt.
- Der andere Teilnehmer kann weiter senden - jedoch sieht man in der Praxis fast immer das Verhalten, dass der andere Teilnehmer als Reaktion auch seine Senderolle beendet.

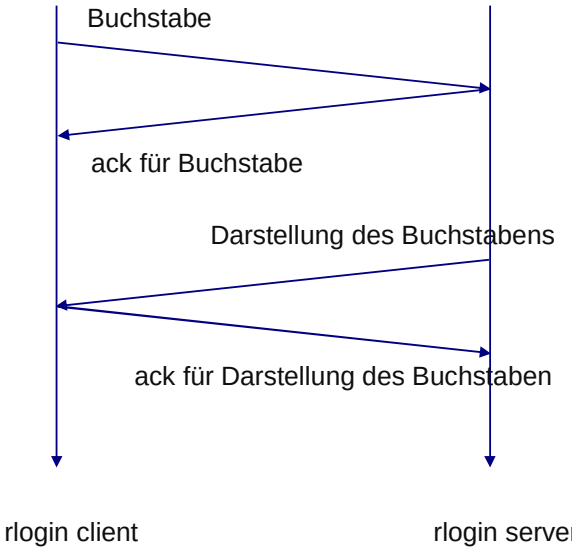
TCP Reset

Ein Paket, bei dem das RST-Bit im TCP-Header gesetzt ist, terminiert die Verbindung in Form eines „abortive release“ (im Gegensatz zum „orderly release“ mit FIN):

- Alle Daten, die beim Sender gepuffert waren, werden verworfen, und das Reset-Paket wird sofort gesendet. Die Verbindung ist damit aus Sicht des RST-Senders sofort geschlossen.
- Beim Reset können Daten verloren gehen (das passiert beim Verbindungsabbau mit FIN nicht).
- Der Empfänger eines RST-Pakets meldet dies der Anwendung und terminiert die Verbindung sofort.

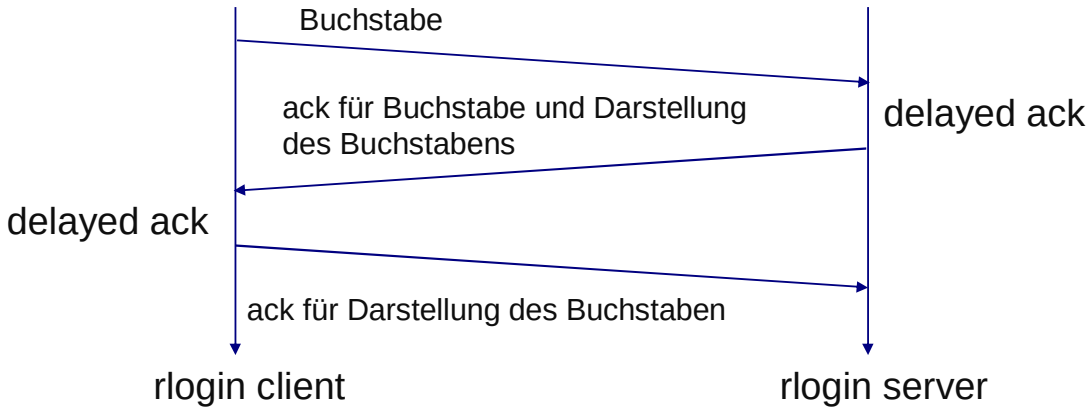
Typische Datenflüsse mit TCP

Datenflussbeispiel für eine interaktive Anwendung



Delayed Acknowledgements (1)

Um zu verhindern, dass überflüssige TCP-Pakete gesendet werden, die nur ein ACK beinhalten, wird das Senden von ACKs häufig verzögert:



Delayed Acknowledgements (2)

- Ein ACK wird bei “delayed acknowledgements“ in der Regel um maximal 200 ms verzögert.
- Während dieser Zeit können Daten, die in der Gegenrichtung gesendet werden, das ACK „Huckepack“ (engl. piggyback) mitnehmen, dies spart die Übertragung eines separaten ACK-Pakets.
- Werden innerhalb dieser Zeit keine Daten gesendet, so wird ein reines ACK-Paket übertragen.
- Der 200-ms-Timer wird nicht für jedes Paket aufgezogen, sondern läuft global, d.h. alle 200 ms werden alle ACKs verschickt, die noch offen sind.

Massendatentransfer

Was passiert, wenn man große Datenmengen per TCP überträgt (bulk data flow)? Wann wird ein ACK gesendet?

- Bisher: delayed ACK nach 200 ms
- Das würde zu viele unbestätigte Paketen im Transit verursachen, wenn die Datenrate hoch ist (bulk data flow)
- Daher wird beim Massendatentransfer jedes zweite Paket sofort bestätigt, auch wenn der 200-ms-Timer noch nicht abgelaufen ist.

Fehlersicherung in TCP

- TCP unterteilt den Bytestrom in Einheiten, die jeweils in einem IP-Paket übertragen werden. Diese Einheiten heißen **Segmente**.
- Nachdem TCP ein Segment per IP losgeschickt hat, wird ein Timer für dieses Segment gestartet.
- Wenn keine Bestätigung über den erfolgreichen Empfang dieses Segments innerhalb der Timer-Laufzeit eintrifft, wird die Übertragung wiederholt.
- Der Timer passt sich an die “normale“ Round-Trip-Time der Verbindung dynamisch an.
- Wenn ein TCP-Empfänger ein fehlerfreies Segment vom Sender erhält, schickt er eine Bestätigung über den erfolgreichen Empfang an den Sender.
- Negative ACKs (NACKs) werden nicht verschickt!

Fehlererkennung und -behebung

- TCP berechnet eine Prüfsumme (checksum) über die versandten Fragmente. Die Berechnung erfolgt mit demselben Algorithmus wie bei UDP. Falls die Prüfsumme einen Fehler signalisiert, wird das Segment nicht bestätigt. Das führt zum Ablaufen der Zeitschranke beim Sender und als Folge davon zur Übertragungswiederholung. Der Sender wiederholt das Senden nach dem „go-back-n“-Verfahren.
- Wenn Segmente außer der Reihe von IP ausgeliefert werden, wird TCP die richtige Reihenfolge wieder herstellen.
- Wenn IP-Datagramme im Netz verdoppelt werden, dann filtert TCP die Duplikate heraus.

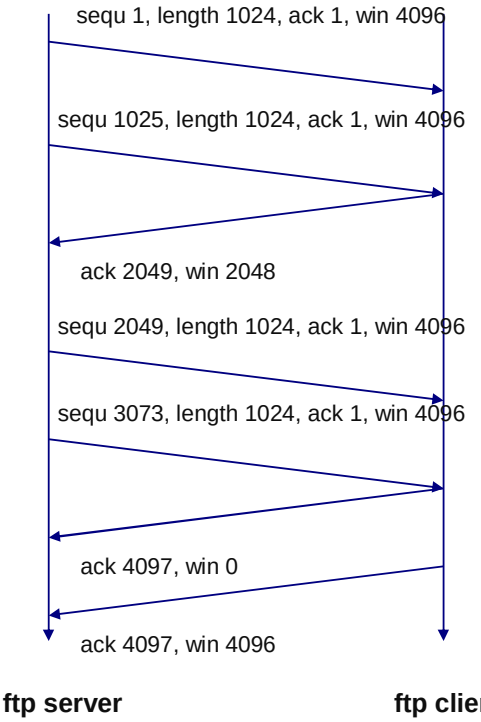
Flusskontrolle und Überlastkontrolle in TCP

Flusskontrolle

TCP verwendet einen Schiebefenster-Mechanismus (sliding window) zur Flusskontrolle. Die Fenstergröße wird, wie stets in TCP, in Bytes (nicht in Paketen) ausgedrückt.

- Die Größe des Schiebefenster wird als Flusskontrollparameter „window size“ vom Empfänger an den Sender geschickt.
- Die Größe des Fensters kann während der Verbindung geändert werden. Wenn der Empfänger beispielsweise nur noch weniger Pufferplatz hat, wird sie reduziert.

Schneller Sender und langsamer Empfänger (1)

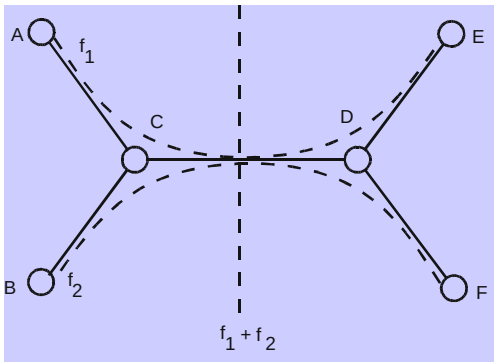


Schneller Sender und langsamer Empfänger (2)

- Der Sender überträgt die Daten schneller, als sie der Empfänger aus dem Puffer lesen und an die höheren Schichten weiter geben kann.
- Der Puffer des Empfängers läuft voll, er signalisiert dies mit der Fenstergröße 0. Dies ist eine Erweiterung zum eigentlichen Sliding-Window-Mechanismus, welche es erlaubt, Pakete bezüglich der Fehlersicherung zu bestätigen, ohne dem Sender das Recht zu geben, weitere Pakete zu senden.
- Erst wenn der Puffer beim Empfänger wieder freien Platz hat, wird dieser freie Platz in einem weitern ACK als Fenstergröße dem Sender mitgeteilt

Überlastkontrolle (Congestion Control)

Problem: Wenn alle Sender im Netz immer so viele Pakete losschicken, wie bei ihren Empfängern in die Puffer passen, dann kann es zur Überlast (Verstopfung, congestion) im Inneren des Netzes kommen.



Wenn alle Verbindungen die gleiche Bandbreite haben und sowohl A als auch B mit der vollen Bandbreite der Verbindung senden, dann kommt es zu einer Überlastung im Inneren des Netzes (**congestion**). TCP-Verbindungen erkennen dies und regeln **freiwillig** die Bandbreite herunter!

Congestion Window

- Um Congestion zu verhindern, wird beim Sender ein zusätzlicher Parameter **Congestion Window** (cwnd) mitgeführt.
- cwnd wird wie das vom Empfänger mitgeteilte flow control window in Bytes geführt.
- Ein Sender darf immer nur das MINIMUM aus (cwnd, Flusskontroll-Window) an Daten senden.
- Es besteht keine Notwendigkeit, cnwd zwischen den Partnern zu übertragen!

Überlastkontrolle im eingeschwungenen Zustand

Problem

Wie stellt der Sender die richtige Größe für das Congestion Window fest?

Lösung

Er versucht, das Fenster schrittweise zu vergrößern, bis Verstopfung eintritt. Dann verkleinert er es wieder.

Solange keine Pakete verloren gehen, wird bei jedem ACK cwnd um 1/cwnd Segmente erhöht. Pro Round-Trip-Time vergrößert sich das Congestion Window also um ca. ein Segment ("*additive increase*").

Da es keine spezifischen Meldepakete aus dem Inneren des Netzes für Verstopfung gibt, interpretiert TCP jeden Paketverlust der Verbindung als Anzeichen für Überlast!

Es gibt zwei unterschiedliche Reaktionen auf Paketverluste, wie nachfolgend beschrieben.

Anzeichen für Überlast

- **Triple Duplicate Acknowledgement** (TDACK): Wenn ein Paket verloren geht, aber nachfolgende Pakete ankommen, erhält der Sender mehrere ACKs mit derselben Sequenznummer. Nach dem dritten „Duplicate ACK“ wird das fehlende Paket erneut übertragen, ohne dass der Timeout für das Paket abgewartet wird (Fast Retransmit). Der Sender interpretiert ein TDACK als „leichte“ Überlast und **reduziert cwnd auf die Hälfte** der ursprünglichen Größe (“*multiplicative decrease*“).
- **Timeout**: Wenn nach einem verlorengegangenen Paket auch keine Folgepakete mehr ankommen, kommt es nicht zu einem TDACK, sondern zu einem Timeout für das verlorene Paket. Der Sender interpretiert das als schwere Überlast und reduziert das cwnd **auf ein Segment**.

Wahl des Timeout-Wertes

Frage: Wie groß sollte der Timeout-Wert gewählt werden?

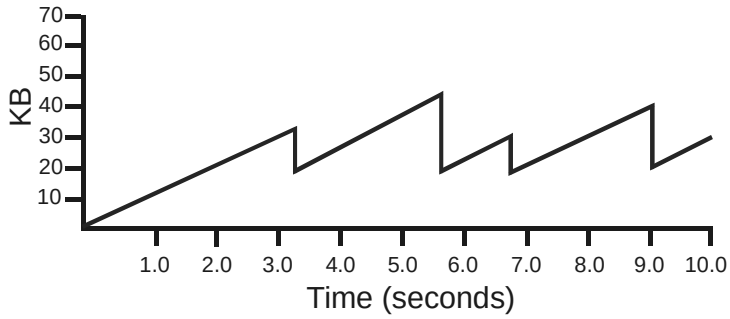
- Auf jeden Fall größer als eine Round-Trip Time (RTT)
- RTT kann variieren → Sicherheitszuschlag in Abhängigkeit der RTT Varianz
  - EstimatedRTT = (1-x) * EstimatedRTT + x * SampleRTT
  - Deviation = (1-x) * Deviation + x * abs(SampleRTT - EstimatedRTT)
  - Timeout = EstimatedRTT + 4 * Deviation
- Ein guter gewählter Timeout-Wert ist wichtig für hohen Datendurchsatz in Netzen, in denen häufig Paketverluste auftreten.

Sägezahnkurve des TCP-Durchsatzes

Der Algorithmus *Additive Increase, Multiplicative Decrease* führt dazu, dass der tatsächliche Durchsatz einer TCP-Verbindung im eingeschwungenen Zustand eine Sägezahnkurve ergibt. Aus regelungstechnischer Sicht ist das Verfahrenstabil, d.h. es schwingt sich nicht auf.

Beobachtung: Die Fläche unter der Kurve entspricht der TCP-Datenrate!

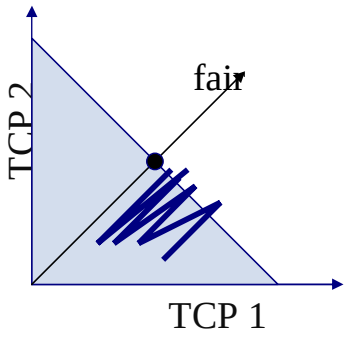
Ein Beispiel zeigt die unten stehende Abbildung.



TCP Fairness

Wenn sich *n* TCP-Ströme einen Link teilen, sollte jeder einen Durchsatz von ca. 1/*n* der verfügbaren Bandbreite in Anspruch nehmen.

Beispiel: Zwei TCP-Ströme laufen über einen gemeinsamen Link:



Die Datenraten tendieren gegen den Punkt der fairen Kapazitätsverteilung unter voller Ausnutzung der vorhandenen Bandbreite.

Slow-Start (1)

Problem:
Wenn eine TCP-Verbindung neu aufgebaut wird, dauert es sehr lange, bis der Sender die optimale Bitrate erreicht, da sich mit jeder Round-Trip-Time *cwnd* nur um eine Paketgröße (MSS) erhöht.

Lösung:
Der **Slow-Start-Algorithmus** von Van Jacobsen

- *cwnd* wird mit der MSS des Empfängers initialisiert
- slow start threshold (*ssthresh*) wird mit 65353 initialisiert
- Pro Round-Trip-Time tue folgendes:
  - Wenn keine Verluste und *cwnd* < *ssthresh*:
    Slow-Start-Phase: erhöhe *cwnd* um MSS für jedes empfangene ACK (dies ist exponentiell!)
  - Wenn Verluste oder *cwnd* >= *ssthresh:*
    Ende der Slow-Start-Phase: normaler AIMD-Algorithmus setzt ein.

Slow-Start (2)

Ein TCP-Slow-Start findet ebenfalls nach einem Timeout statt (Anzeichen für schwere Verstopfung). Der *ssthresh*-Wert wird halbiert, dann beginnt eine neue Slow-Start-Phase.

Slow-Start (3)

Anmerkung

Dieser Algorithmus heißt aus historischen Gründen „slow start“, obwohl es aus heutiger Sicht eigentlich ein “Quick-Start-Algorithmus“ ist. Denn bevor er in TCP eingebaut wurde, begann jede neue Verbindung mit einer Fenstergröße, die dem **Flusskontroll**fenster des Empfängers entsprach. Dies führte häufig sofort zu Verstopfung und Paketverlusten.

Slow-Start: Beispiel für einen Verlauf

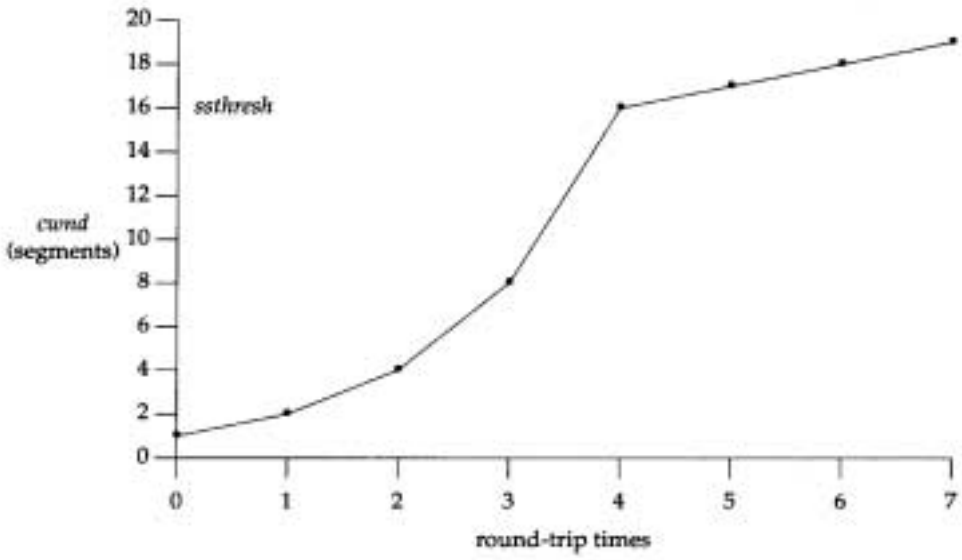


Figure 21.8 Visualization of slow start and congestion avoidance.

TCP-Programmierschnittstelle (API)

Die Programmierung von TCP erfolgt mit dem Konzept der **Sockets**. Ein Socket ist die programmiertechnische Realisierung eines Ports.

- Der Server wartet auf einem wohl definierten Port auf Verbindungswünsche von Clients.
- Ein Client verbindet sich mit dem Server .
- Nachdem die Verbindung hergestellt ist, kann der Server für diese Verbindung einen Thread (leichtgewichtigen Prozess) anlegen, der dann die weiterhin eingehenden Datenpakete der Verbindung bearbeitet.
- Wenn er keinen neuen Thread anlegt, werden die Verbindungswünsche sequentiell bearbeitet (selten).

Zusammenfassung TCP

Vorteile von TCP

- Gesicherte Datenübertragung
- Effiziente Datenübertragung trotz Komplexität des Protokolls (bis zu 100 MBit/s auf Standard-Maschinen experimentell nachgewiesen)
- Einsetzbar im LAN- und WAN-Bereich
- Für geringe Datenraten (z. B. interaktives Terminal) und hohe Datenraten (z. B. Dateitransfer) gut verwendbar

Nachteile gegenüber UDP

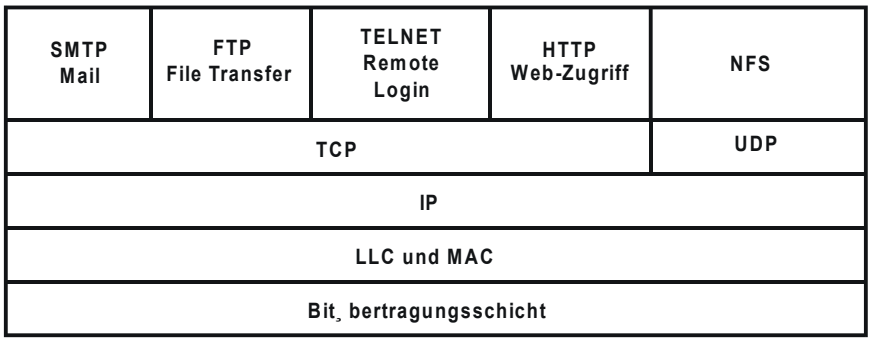
- Höherer Ressourcenbedarf (Zwischenspeicherung, Zustandsinformationen bei Sender und Empfänger, viele Timer)
- Verbindungsaufbau und -abbau auch bei kurzen Datenübertragungen notwendig
- Multicast nicht möglich

Anwendungsbeispiele für TCP

- E-Mail (SMTP)
- Web-Zugriff (HTTP)
- Dateitransfer (FTP)
- interaktives, zeichenorientiertes Terminal (Telnet)
- interaktives, grafikfähiges Terminal: X.11-Protokoll für X-Windows
- und viele mehr!

8. Anwendungsschicht

8.1 Architektur der Anwendungsprotokolle im Internet



SMTP	=	Simple Mail Transfer Protocol
FTP	=	File Transfer Protocol
TELNET	=	Remote Login Protocol
UDP	=	User Datagram Protocol
NFS	=	Network File System
TCP	=	Transmission Protocol
IP	=	Internet Protocol
LLC	=	Logical Link Control
MAC	=	Media Access Control

8.2 SMTP für elektronische Post

SMTP: Simple Mail Transfer Protocol (RFC 822)

- Electronic Mail im Internet
- Benutzt im Gegensatz zu X.400 von ITU-T eine direkte TCP-Verbindung zum Ziel-Host (kein Mail Forwarding in Schicht 7)
- TCP-Port: 25
- Beispiele für Protokollelemente:

HELO	Vorstellung
MAIL	Angabe des Absenders
RCPT	Angabe des Empf‰ngers
DATA	Senden der Nachrichten
QUIT	Ende
VRFY	Verifizieren des Benutzernamens
EXPN	Angabe von Verteilerlisten

Funktionalität von SMTP

- Das SMTP-Protokoll sorgt nur für die Übertragung der Nachrichten, nicht aber für die Zwischenspeicherung oder Präsentation der Nachricht ("local matter")
- Die Mailer-Instanzen kommunizieren mittels lesbarem Text (ASCII); die PDUs enthalten keine binären Datenfelder
- Der Empfänger muss jede Meldung bestätigen
- Mehrere Nachrichten (mails) können nacheinander über eine TCP-Verbindung geschickt werden, wenn die Empfänger auf demselben Host sind
- Weiterleitung von Nachrichten (forwarding) bei einer Adressenänderung des Benutzers möglich

Beispiel-Interaktion mit SMTP

```
R:    220 Beta.GOV Simple Mail Transfer Service
      Ready
S:    HELO Alpha.EDU
R:    250 Beta.GOV

S:    MAIL FROM:<Smith@Slphs.EDU>
R:    250 OK

S:    RCPT TO:<Green@Beta.GOV>
R:    550 No such user here

S:    RCPT TO:<Brown@Beta.GOV>
R:    250 OK

S:    DATA
R:    354 Start mail input; end with
      <CR>LF>.<CR><LF>
S:    ...sends body of mail message...
S:    ...continues for as many lines as message
      contains
S:    <CR><LF>.<CR><LF>
R:    250 OK

S:    QUIT
R:    221 Beta.GOV Service closing transmission
      channel
```


MIME

MIME (Multimedia Internet Mail Extension)

Die frühen Mail-Standards sahen nur die Übertragung von Textströmen vor, gedacht für menschliche Leser. In den ersten Jahren sogar nur 7-Bit-US-ASCII! Zur Übertragung von Binärdateien musste stets ftp verwendet werden.

Die MIME-Codierung dient dazu, beliebige Binärdaten in einen ASCII-Datenstrom zu konvertieren, der dann problemlos alle Mail-Systeme und Mail-Gateways passiert. Dies ist vor allem für Mail-Attachmemts (Anlagen) gebräuchlich. Für verschiedene Arten von Binärdateien werden dazu MIME-Typen fest gelegt, die die Codierungsregeln definieren. Eine MIME-Mail kann viele „body parts“ haben mit jeweils verschiedenen MIME-Typen.

8.3 FTP für Dateitransfer

ftp (file transfer protocol)

tftp (trivial file transfer protocol)

Funktionen von FTP

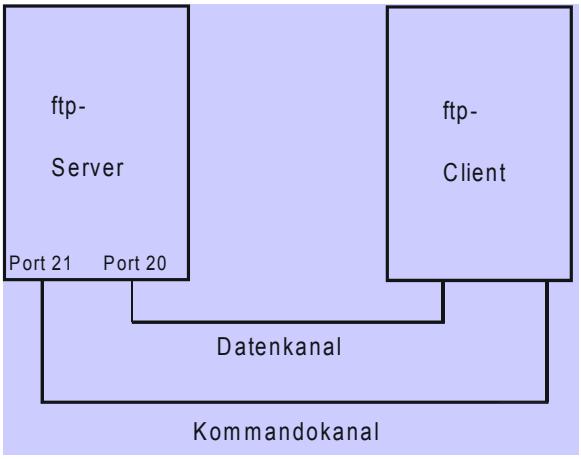
- Senden, Empfangen, Löschen und Umbenennen von Dateien
- Einrichten und Löschen von Verzeichnissen
- Wechsel des aktuellen Verzeichnisses.

Die Dateiübertragung kann in binärem oder ASCII-Modus erfolgen.

Im binären Modus (auch "image file type" genannt) wird der Bitstrom aus dem Speicher des Senders ausgelesen und unverändert übertragen.

Im ASCII-Modus geht ftp davon aus, dass nur alphanumerische Zeichen übertragen werden sollen. Als Transfercodierung wird ASCII gewählt; wenn der Sender oder der Empfänger eine andere lokale Darstellung hat, wird umcodiert. Im Transfer wird ein Zeilenende als <CR><LF> codiert; wenn der Sender oder der Empfänger eine andere lokale Darstellung dafür hat, wird auch hier umcodiert. Der ASCII-Modus implementiert also eine minimale Funktionalität der Darstellungsschicht für Textdateien.

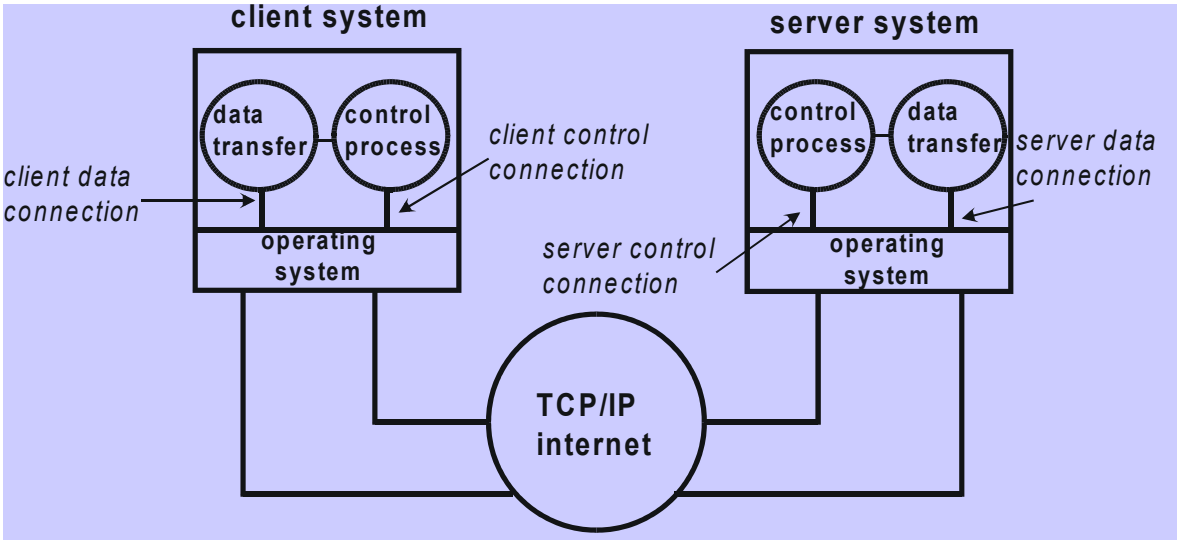
Grundlagen von FTP



- Die Kommandos von FTP werden als vierbuchstabige Zeichenfolge plus Optionen übertragen (z. B. PASS xyz für das Passwort)
- Als Antwort wird eine Folge von drei Ziffern geliefert, die Erste davon gibt über den Typ der Antwort Auskunft (1,2,3 => kein Fehler, 4,5 => Fehler usw.)

Architektur von FTP (1)

Der ftp-Client läuft als Programm im Adressraum des Benutzers. Es gibt keine Integration in das lokale Dateisystem.



Architektur von FTP (2)

- Separate TCP-Verbindungen für Kontrollprotokoll und Daten
- Authentifizierung (Passwort) beim Aufbau der Kontrollverbindung
- Verzeichnis-Operationen möglich (ls, cd, rm, ...)
- put und get zum Übertragen der Dateien
- help-Funktionen

8.4 NFS für den Fernzugriff auf Dateien im Netz

Geschichte

1984: Ankündigung

1985: Erstes Produkt auf einer SUN

1986: Portierung für System-V-release-2

1986: NFS 3.0: (verbessertes YP) und PC-NFS

1987: NFS 3.2: File-Locking

1989: NFS 4.0: Verschlüsselung

1989: Lizensierung durch 260 Hersteller

Merkmale von NFS

- Transparenter Zugriff auf Dateien in entfernten Dateisystemen
- Integriert in das Betriebssystem/Dateisystem
- Client / Server-Modell
- Entwickelt seit 1985 von SUN
- Standard auf allen UNIX-Rechnern
- Auch für Windows und Großrechner-Betriebssysteme verfügbar
- Offenes System (Spezifikation ist öffentlich)
- Portierung ist einfach
- Referenzimplementierung ist öffentlich
- Import / Export von Verzeichnissen
- Kommunikation über UDP, also verbindungslos
- Keine Darstellungsdienste, nur Lesen und Schreiben von Byte-Strömen!

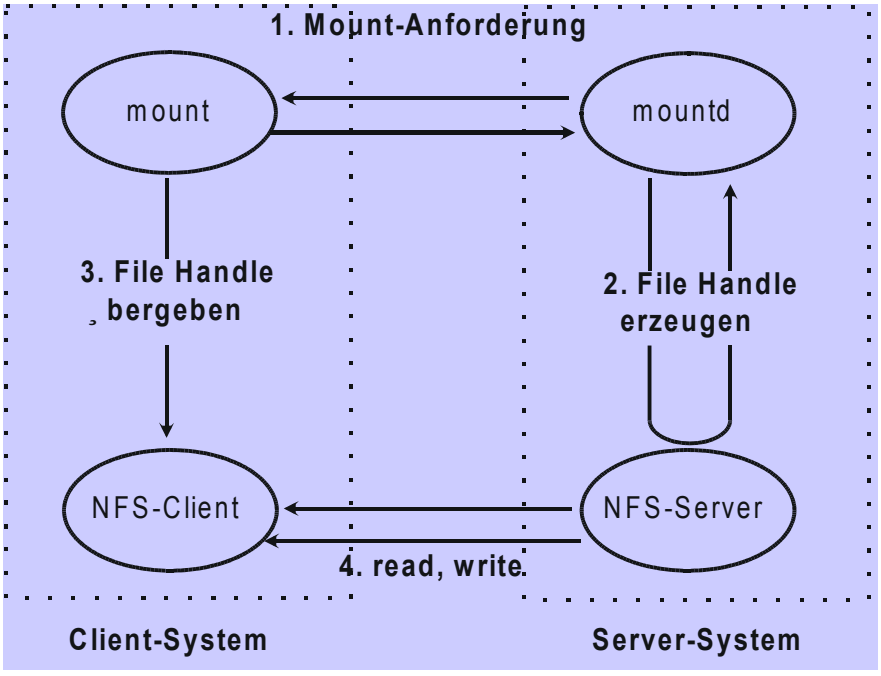
Mounting

Das NFS-Protokoll ist lediglich ein Protokoll für den Dateizugriff (Lesen und Schreiben). Das Bereitstellen von entfernten Verzeichnissen / Dateien geschieht durch das mount-Protokoll.

mount verbindet ein entferntes Dateisystem mit einem lokalen Verzeichnis, d. h. der gesamte entfernte
Dateibaum wird in das bisherige Verzeichnis eingehängt. Danach kann über NFS auf die entfernten Dateien wie auf lokale Dateien zugegriffen werden.

mount und NFS sind getrennte Protokolle, mount bzw. mountd stellen lediglich einige Informationen für NFS bzw. nfsd zur Verfügung (z. B. Rechnernamen und Pfad).

NFS-Protokoll und MOUNT-Protokoll (1)



NFS-Protokoll und MOUNT-Protokoll (2)

mountd und nfsd (demons im Unix-Sinne) werden beim Hochfahren des Servers automatisch gestartet. nfsd aktiviert den NFS-Server-Code im Betriebssystem.

Beim MOUNT-Vorgang wird auf der Server-Seite ein "File Handle" (eindeutiger Dateisystem-Kontrollblock) erzeugt und an den Client zurückgegeben. Der NFS-Client verwendet diesen bei allen späteren Zugriffen auf den entfernten Verzeichnis(-teil)-Baum.

Lock-Manager

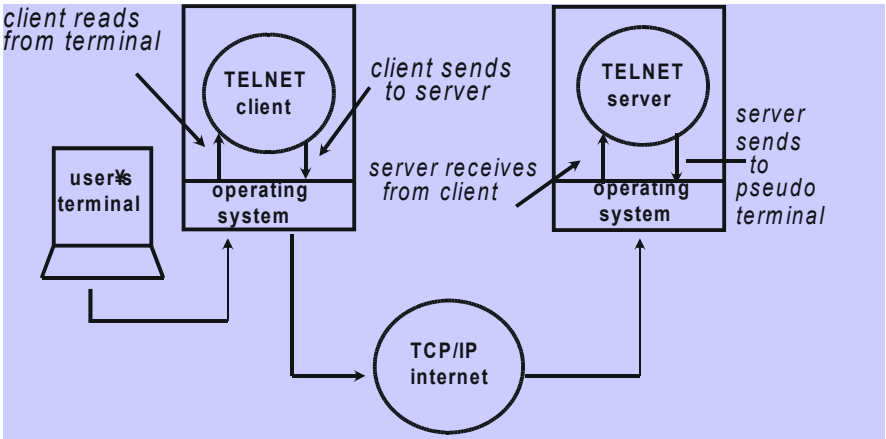
Problem

Gleichzeitiger Schreibzugriff durch mehrere entfernte NFS-Clients

- Der Lock-Manager ermöglicht das Sperren von Dateien
- Dienst parallel zu NFS (lockd)
- Keine Deadlock-Erkennung!

8.5 TELNET für virtuelles Terminal (remote login)

- Virtuelles Terminal auf einem entfernten Rechner
- Alphanumerisch, full screen mit Scrolling, Blinken etc. Aber nicht grafikfähig wie X. 11
- Setzt auf TCP auf



TCP-Verbindung zwischen Client und Server

8.6 HTTP für das World Wide Web

HTTP - Hypertext Transfer Protocol

- Einfaches request/response- Protokoll zwischen Web Client und Web Server, alle PDUs im ASCII-Format
- Minimale Transaktionsbelastung des Servers: **zustandsloses Protokoll**
- In HTTP 1.0 wird für jedes einzelne Dokument eine Verbindung aufgebaut und wieder geschlossen, der Serverprozess belegt dann keine Ressourcen mehr.
- Caching von DNS-Informationen (Client Caching)
- Übertragung über ein zuverlässiges Transportprotokoll, typischerweise TCP, andere Protokolle möglich

HTTP - Geschichte

Geschichte

- 1989: HTTP 0.9 ausschließlich für Hypertext konzipiert
- 1990: HTTP 1.0 Übertragung beliebiger Datenformate
- 7/1993: HTTP Internet Draft, erste Fassung
- 1996: HTTP 1.0 siebte Fassung
- 04/1999: HTTP 1.1 Internet Draft, läuft stabil

HTTP - Transaktion

1. Verbindungsaufbau

- WWW-Client baut eine TCP/IP-Verbindung zum WWW-Server auf
- TCP-Port: 80

2. Anforderungen (Request)

- Client sendet Requests über die aufgebaute Verbindung (zum Beispiel GET, PUT, POST)

3. Antwort (Response)

- Reaktionen des Servers auf Request, z. B. angefordertes Dokument
- Code über Status des Requests

4. Verbindungsabbau

- nach Abschluss der Übertragung Beenden der Verbindung durch den Server

Beispiel: ASCII-Kommunikation von HTTP

	bash$ telnet numalfix 80	
	Trying...	
	Connected to numalfix.wifo.uni-mannheim.de	*Leerzeile*
	Escape character is '^]'.	
Client:	**GET /index.html HTTP/1.0**	
Request	**Accept: image/gif**	*Methode*
Server:	HTTP/1.0 200 Document follows	
Response	Date: Sun, 09 Jun 1996 13:13:09 GMT	*Dokument-kopf*
	Server: NCSA/1.5	
	Content-type: text/html	
	Last-modified: Thu, 30 May 1996 10:42:31 GMT	
	Content-length: 1751	
	<html><head><title>BWL-Hauptseite</title><head>	
	<body>	
	...	
	<IMG SRC=/images/unilogo.gif...	
	<IMG SRC=/images/FakBWL2.gif...	
	<IMG SRC=/images/ball.red.gif...	
	...	

Auszug aus dem Access-Log des WWW-Servers

obelix.wifo.uni-mannheim.de - - [Datum] "GET/index.html HTTP/1.0" 200

obelix.wifo.uni-mannheim.de - - [Datum] "GET/images/neu2.gif HTTP/1.0" 200

obelix.wifo.uni-mannheim.de - - [Datum] "GET/images/FakBWL2.gif HTTP/1.0" 200

obelix.wifo.uni-mannheim.de - - [Datum] "GET/images/unilogo.gif HTTP/1.0" 200

obelix.wifo.uni-mannheim.de - - [Datum] "GET/images/ball.red.gif HTTP/1.0" 200

obelix.wifo.uni-mannheim.de - - [Datum] "GET/images/ball.green.gif HTTP/1.0" 200

obelix.wifo.uni-mannheim.de - - [Datum] "GET/images/minfo.gif HTTP/1.0" 200

obelix.wifo.uni-mannheim.de - - [Datum] "GET/images/minfo.gif HTTP/1.0" 200

obelix.wifo.uni-mannheim.de - - [Datum] "GET/images/mup.gif HTTP/1.0"304

Verbesserungen

- TCP-Verbindung halten, wenn mehrere Dateien vom selben Server zu holen sind (möglich ab HTTP 1.1)
- Aufbau mehrerer paralleler TCP-Verbindungen zum selben Server zur Beschleunigung der Datenübertragung

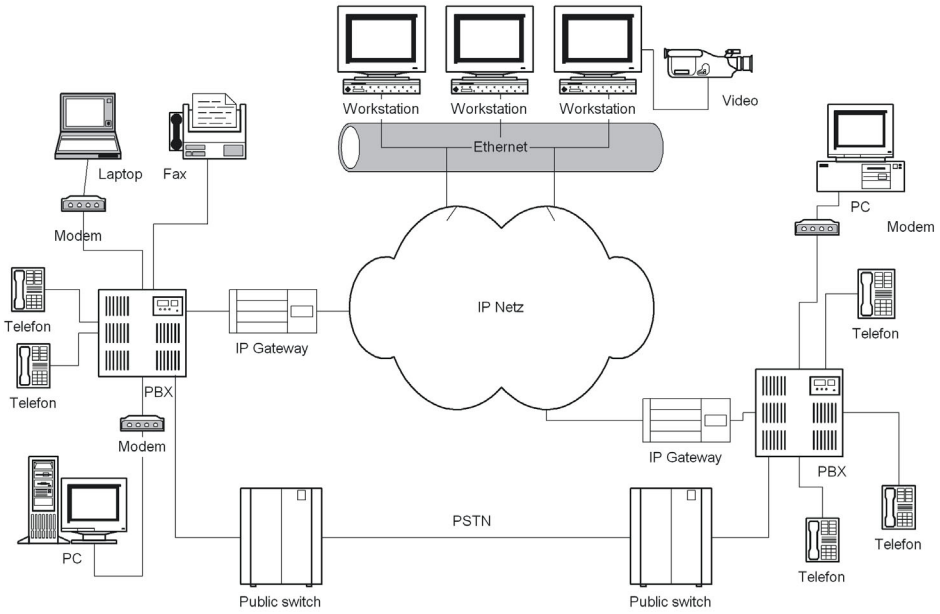
8.7 Telefondienste über IP*

Grundlagen und Protokolle

- Effiziente Realisierung, da Paketvermittlung mit IP weniger Overhead und mehr Flexibilität bringt
- Mehr Funktionalität als herkömmliche Telefonie im leitungsvermittelten Telefonnetz (PSTN)
- Ziel: Multimediakommunikation und intelligente Dienste in IP-basierten Netzen.

*Ich danke Robert Denda und Dr. Andreas Grebe für die Überlassung von Folien für Kapitel 8.7

Architektur



Gründe für IP-Telefonie (1)

- Herkömmliches Telefonnetz: verbindungsorientiert, leitungsvermittelt, aufwendige Vermittlungsanlagen (PBXs)
- IP-Telefonie:
  - paketvermittelt, statistischer Multiplexeffekt, „einfache“ Router
  - geringere Bandbreite durch Audiokompression (z. B. G.723.1 nur 5,3-6,3 kBit/s gegenüber beispielsweise 64 kbit/s PCM bei ISDN)
- Flexibilität bei der Signalisierung
- Integration von Multimedia
- Erweiterbarkeit um intelligente Netzwerkdienste (Anrufweiterleitung, Anklopfen, Mehrpunkt-Verbindungen)
- Skalierbarkeit der Kommunikationsdienstgüte
- Endgerätevielfalt: PCs, IP-Telefone, Fax, etc.
- Nutzung bestehender Datennetze

Gründe für IP-Telefonie (2)

Motivation für den Anwender:

- Kostenvorteile
- PC-Integration ist oft praktisch

Anforderungen an IP-Telefonie (1)

Hauptproblem: Dienstgüte

- **Verzögerung** (Delay):
  Experimentell gemessene Verzögerungen:
  - Codierungs-/Decodierungsverzögerung:
    ca. 30 ms (G.729A) - 82 ms (G.723.1)
  - Netzwerkverzögerung (Übertragung, Routing, etc.):
    Gateway-Gateway: 30 - 100 ms
    PC-PC: 50 - 140 ms
  - Zugriffsverzögerung (Betriebssystem, Sound-und Videokarten, DSPs, ...):
    Gateway-Gateway: 40 - 80 ms
    PC-PC: 100 - 340 ms
    Aber: menschliches Ohr diesbezüglich sehr sensibel, wünschenswertes Delay: < 100 ms
- **Paketverlust**: FEC erforderlich, erhöht Verzögerung und Datenrate
- **Multicast**: Heterogenität der Teilnehmer, Dynamic Join-and-Leave erfordern aktive, adaptive QoS-Mechanismen im Netz

Anforderungen an IP-Telefonie (2)

Weitere Probleme

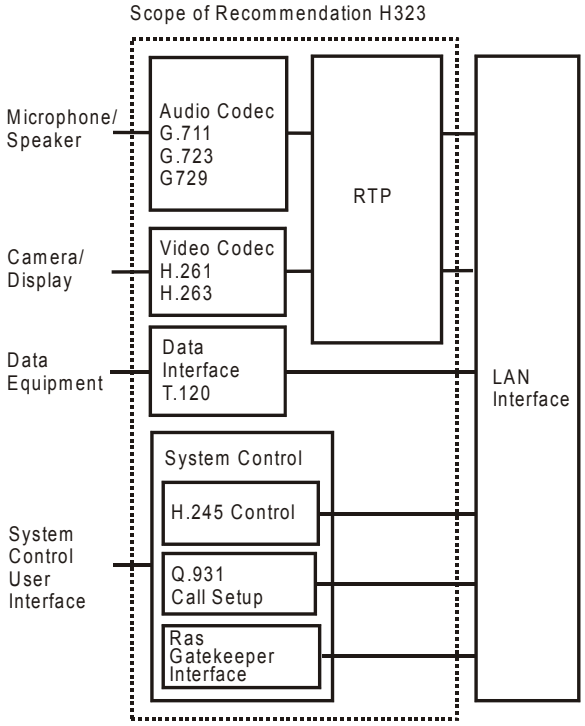
- Crosstalk, Echo, Rauschen (wie bei POTS)
- Leitwegänderungen
- QoS-basierte Gebühren etc.

- Intelligente Dienste
  - „Standard“ IN/AIN-Dienste: Anklopfen, Anrufbeantworter im Netz etc.
  - Neue intelligente Dienste:
    Directory-Dienste
    WWW-Schnittstellen etc.
- Signalisierung
  - Leichtgewichtige Signalisierung (Internet vs. IN/AIN)
  - neue Medien, erweiterte Dienste, Charging
    erfordern neue Signalisierungsmechanismen

Anforderungen an IP-Telefonie (3)

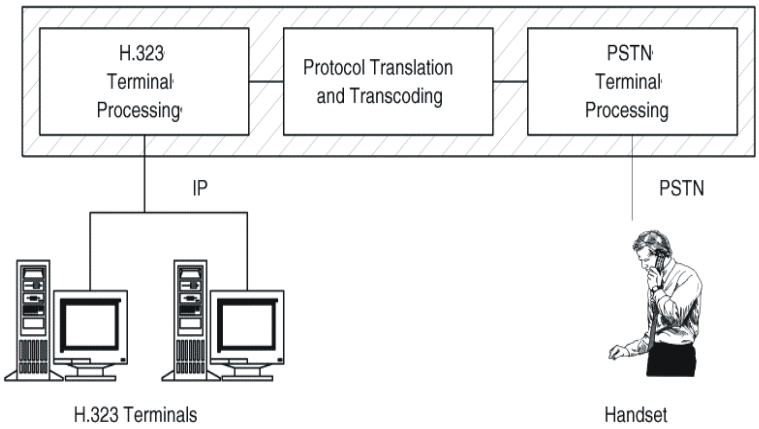
- Mobilität, Interworking
  - Interworking of LAN, ATM, breitbandige Zugangsnetze, Wireless LAN/WAN
- Sicherheit
- Abrechnungssystem (charging, accounting)

IP-Telefonie-Standard: ITU Recommendation H.323



H.323 (1)

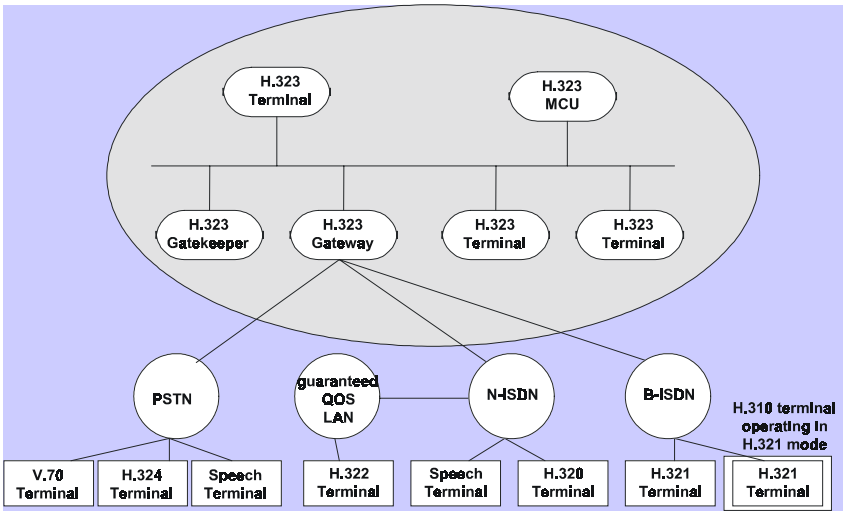
Version 1 von H.323 wurde von der ITU-T 1996 verabschiedet, Version 2 im Januar 1998. Version 3 ist geplant. H.323 wird von den meisten Produkten für Voice-over-IP (VOIP) unterstützt und schließt sowohl herkömmliche Telefongeräte als auch PCs ein.



H.323 (2)

H.323 umfasst Call Control, Multimedia- und Bandbreitenmanagement und definiert die Schnittstellen zwischen LANs und anderen Netzen.

Protokolle für IP-Telefonie (1)



MCU= Multipoint Control Unit

N-ISDN= Narrowband ISDN

B-ISDN= Broadband ISDN

Protokolle für IP-Telefonie (2)

Signalisierungsprotokolle

- Control Protocol for Multimedia Communication (H.245): Ende-zu-Ende-Signalisierung, sehr komplex
- Session Initiation Protocol (SIP): einfaches textbasiertes Signalisierungsprotokoll für Internetkonferenzen und -telefonie, unterstützt u.a. transparente Abbildung von Namen auf Adressen und Rufumleitung
- Digital Subscriber Signalling System No.1 (DSS1 Q.931), Q.93B, Q.932: ISDN-Signalisierung
- RSVP – Resource ReSerVation Protocol: Empfänger-orientiertes Protokoll zur QoS- und Bandbreitenreservierung. Hat sich im Internet nicht durchgesetzt.

Protokolle für IP-Telefonie (3)

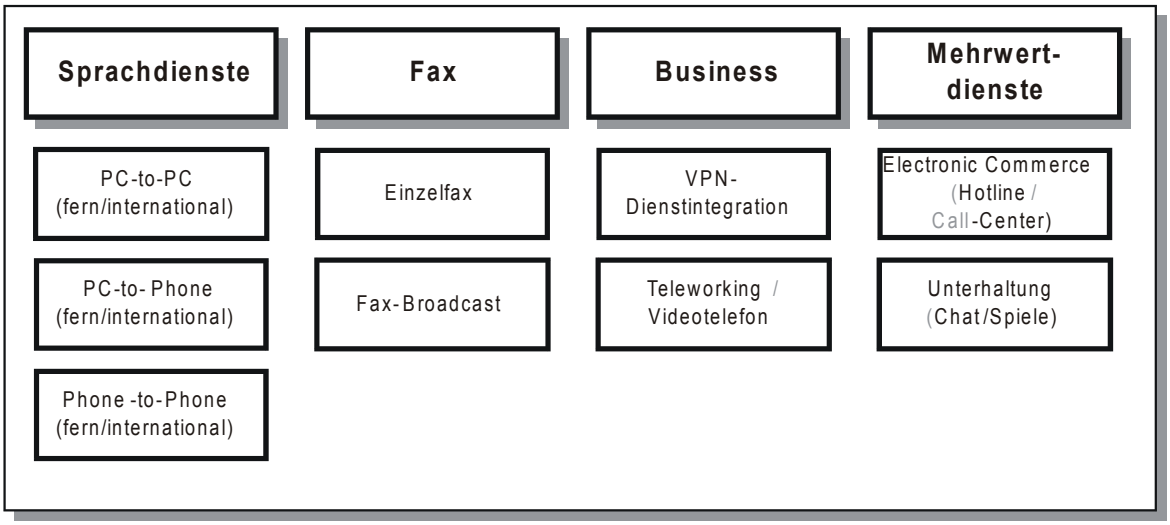
Medienstrom-Kontrollprotokolle

- Real Time Protocol (RTP) / Real Time Control Protocol (RTCP): ein Internet-Protokoll. Unterstützt Multimediakommunikation in Echtzeit. Profile werden definiert für verschiedene Payload-Typen (z. B. MPEG-1, H.263)
- Real Time Streaming Protocol (RTSP): erlaubt bidirektionale Übertragung basierend auf RTP, beinhaltet Sicherheitsmechanismen

„Intelligent Network“ (IN, aus der Telefonie)

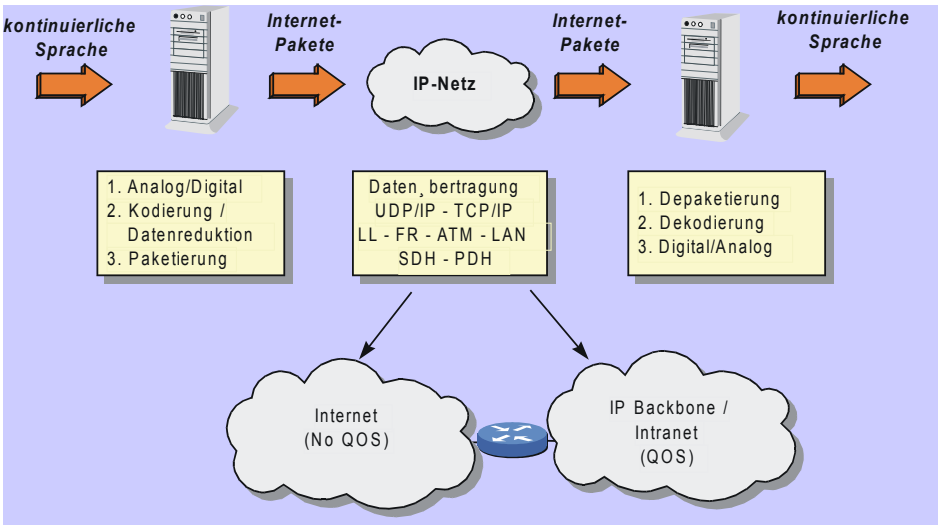
- PSTN: Intelligent Network (IN), Advanced Intelligent Network (AIN)
- IP-Telefonie: Telecommunication Information Networking Architecture (TINA)
- Bereich aktueller Forschung: Nutzung moderner verteilter Systeme (CORBA, Java RMI, DCOM), aktiver Netze und mobiler Agenten zur Implementierung von Netzwerkintelligenz

Voice-over-IP (VOIP)



- VOIP-Dienste sind Mehrwertdienste für Internet-Dienstanbieter
- Mittelfristig: Dienstekopplung IP-PSTN als Regeldienst?

Prinzip von Voice-over-IP (VOIP)



- VOIP heute: nutzbar, einige kommerzielle Dienste, setzt auf Standard-IP auf ("best effort")
- VOIP morgen: gute Qualität (QoS), internationale Dienste/Allianzen?

Sprachcodierung (1)

IP-Telefonie: Sprachcodierer mit relativ guter Qualität bei sehr niedriger Bitrate (z. B. GSM 06.10 mit 13,2 kbit/s oder G.723.1 mit 5.3 kbit/s - 6.3 kbit/s).

Verwendete Verfahren basieren meist auf **Linear Predictive Coding (LPC)**: Für jeden Rahmen von Sprachsamples s[i] werden p Gewichte lpc[0], .. , lpc[p-1] berechnet, so dass gilt:

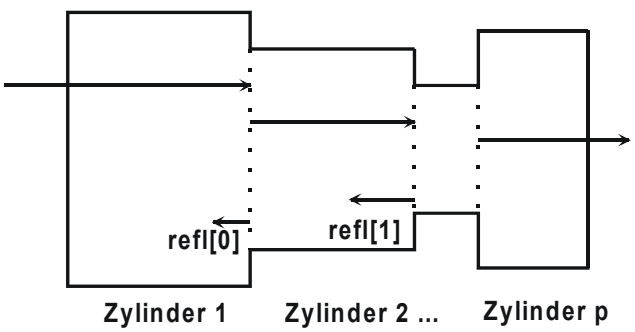
s[i] wird möglichst gut durch

lpc[0] * s[i-1] + lpc[1] * s[i-2] + ... + lpc[p-1] * s[i-p]

angenähert. Übliche Werte für p sind 8 oder 14.

Sprachcodierung (2)

Bei LPC modelliert man das menschliche Sprachorgan als System von miteinander verbundenen, verschieden großen Zylindern:



Codierungsmodell

Akustische Wellen durchlaufen ein System von Zylindern, werden an Übergängen von Zylindern mit unterschiedlichem Durchmesser teilweise reflektiert und interferieren somit mit nachfolgenden Wellen. Die Reflektionsrate wird durch die Reflektionskoeffizienten

refl[0], ..., refl[p-1]

repräsentiert. Diese entsprechen nahezu den lpc-Koeffizienten.

Codierung:

Für jeden Rahmen:

- Berechnung der lpc/refl-Koeffizienten.
- Ein synthetisch generiertes Signal dient als Eingabe des Modells und ergibt synthetische Sprache.
- Unterschiede ε_i zwischen synthetischer Sprache und Samples werden kodiert (ε_i klein bei stimmhaften Phonemen); lpc/refl-Koeffizienten werden kodiert.

Beispiel: G.723

G.723.1

Adaptiver CELP-Codierer (CELP = Code Excited Linear Predictor)

CELP: Die ε_i werden als Indizes in ein Codebook kodiert.

ACELP: wie CELP, aber Codebook adaptiv

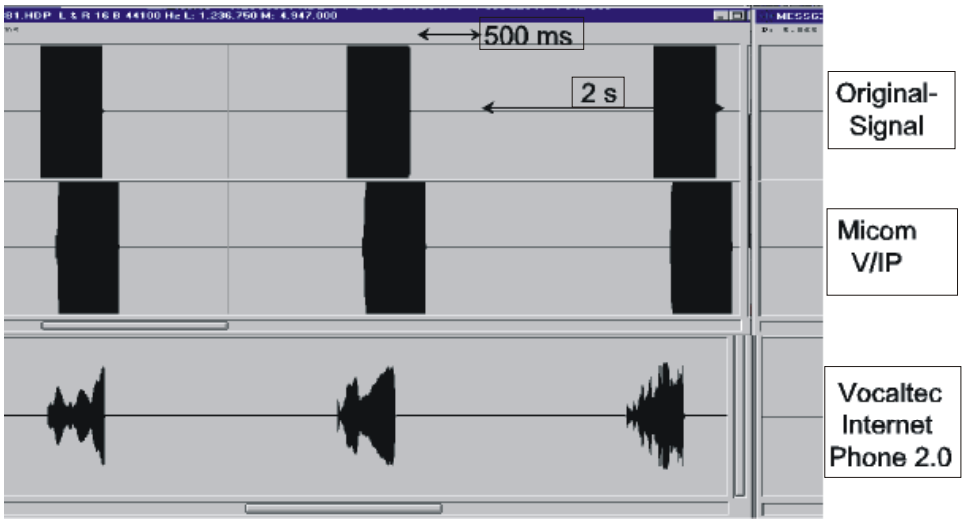
GSM 06.10: Regular Pulse Excitation – Long Term Prediction (RPE-LTP)

Vergleich verschiedener VOIP-Codecs

Codec	Technik	Bitrate	Qualität (MOS)	Standard	Einsatz	Codier-Delay	Leistungs-verbrauch (100 MHz Pentium)
G.711	PCM	64 kbit/s	4,0	H.323	ISDN	< 1 ms	< 1%
G.723.1	ACELP MP-MLQ	5,3 kbit/s 6,3 kbit/s	3,88 3,88	H.324/H.323	PSTN Videotelefon Voice over IP	97,5 ms 97,5 ms	35-49% 35-49%
G.728	LD-CELP	16 kbit/s	3,93	H.323	Voice over IP	3 ms	ca. 65%
G.729	CS-ACEL	8 kbit/s	3,90	H.323	Voice over IP Frame Relay ATM	35 ms	ca. 50%
GSM 6.10	RELP	13 kbit/s	3,80 at 0% errors	not included in H.323	Mobil	ca. 40 ms	real-time coding at 486PC 66MHz
Lucent SX7300P	CS-ACEL	7,3 kbit/s	3,88	not included in H.323	Voice over IP	35 ms	ca. 13,5%

Mean Opinion Score (MOS): Befragung von Testpersonen
>3.8 akzeptabel
>4.0 sehr gute Qualität

Verzerrungen der VOIP-Codecs



- Codecs zeigen stark variierende Verzerrungscharakteristika
- Psycho-akustische Codecs zeigen deutlich stärkere Verzerrungen
- Mean Opinion Score (MOS) in beiden Fällen akzeptabel

9. Verzeichnisdienste: Der Domain Name Service

9.1 Der Namensraum des Domain Name Service (DNS) im Internet

Aufgabe des DNS

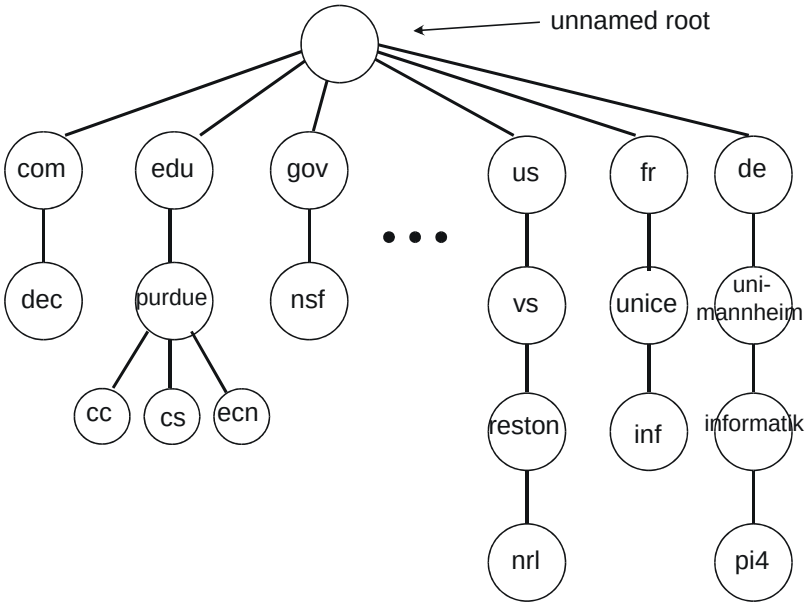
Abbildung von logischen Namen auf Internet-Adressen (IP-Adressen)

Die Namen im INTERNET sind hierarchisch strukturiert, z. B.:

- de
- uni-mannheim.de
- informatik.uni-mannheim.de
- pi4.informatik.uni-mannheim.de

Jede Hierarchiestufe entspricht einer Domäne von Adressen. Für jede Domäne existiert ein Name-Server, der die Hosts seiner Domäne kennt.

Struktur der Domänennamen



Für jede Domäne gibt es eine administrative Stelle für die Namensvergabe.

Bildung von zulässigen Namen

- Jeder Knoten hat einen Bezeichner (“label“), der höchstens 63 Buchstaben lang ist.
- Die Wurzel des DNS-Namensraumes hat einen leeren Bezeichner.
- Groß-/Kleinschreibung wird nicht berücksichtigt.
- Einen “absolute domain name / fully qualified domain name“ (FQDN) erhält man, wenn man von einem Blatt des DNS-Namensraumes zur Wurzel geht und dabei alle Bezeichner notiert; die Bezeichner werden mit einem Punkt voneinander getrennt.

9.2 Die Protokolle des DNS

Die wesentlichen Komponenten des DNS sind:

- Ein verteiltes System zur Erbringung eines Verzeichnis-Dienstes, das aus einer Hierarchie von Name-Servern besteht.
- Ein Protokoll der Anwendungsschicht, das es Hosts und Name-Servern ermöglicht, Host-Namen in IP-Adressen aufzulösen.

Das DNS-Protokoll setzt (in der Regel) auf UDP auf und benutzt Port 53.

Das DNS wird häufig von anderen Protokollen der Anwendungsschicht benutzt, um die vom Benutzer eingegebenen Hostnamen in IP-Adressen umwandeln zu lassen (z. B. von SMTP, HTTP).

Die wichtigsten Standards zum DNS sind RFC 1034 und RFC 1035.

Nebenfunktionen des DNS (1)

- **Host-Aliasing:** Ein Host mit einem komplizierten Hostnamen kann einen oder mehrere Aliasnamen haben. Beispielsweise hat ein Hostname wie relay1.westcoast.enterprise.com zwei Aliasnamen wie enterprise.com und www.enterprise.com. In diesem Fall ist der Hostname relay1.west-coast.enterprise.com ein so genannter **kanonischer** Hostname.

  Sofern vorhanden, sind Alias-Hostnamen in der Regel leichter merkbar (“mnemonischer“) als kanonische.

Nebenfunktionen des DNS (2)

- **Lastverteilung:** Vermehrt wird DNS auch für die Durchführung einer Lastverteilung zwischen replizierten Servern, zum Beispiel replizierten Web-Servern, verwendet. Sehr stark frequentierte Sites, wie zum Beispiel cnn.com, werden auf mehreren Servern repliziert, wobei jeder Server auf einem anderen Host läuft und eine andere IP-Adresse hat. Mit einem replizierten Web-Server wird dann ein Gruppe von IP-Adressen mit einem kanonischen Hostnamen assoziiert. Die DNS-Datenbank enthält diese Gruppe von IP-Adressen.

  Wenn nun Clients eine DNS-Anfrage für einen Namen stellen, der auf eine Gruppe von Adressen abgebildet ist, antwortet der Server mit der gesamten Gruppe der IP-Adressen, stellt aber die Reihenfolge der Adressen in jeder Antwort um. Da ein Client normalerweise seine HTTP-Anfragenachricht an die IP-Adresse sendet, die an erster Stelle in der Gruppe steht, wird der Verkehr durch die DNS-Rotation auf alle replizierten Server verteilt.

Lokale Name-Server

Lokale Name-Server: Jeder ISP (z. B. eine Universität, eine Fakultät, eine Firma oder ein kommerzieller ISP) verfügt über einen lokalen Name-Server (den man auch als “Default-Name-Server“ bezeichnet). Wenn ein Host eine DNS-Anfragenachricht ausgibt, wird die Nachricht zuerst an den lokalen Name-Server gesandt. Die IP-Adresse des lokalem Name-Servers wird in der Regel manuell in jedem Host konfiguriert (bei IP Version 4).

Der lokale Name-Server befindet sich normalerweise in der Nähe des Clients. Wenn ein Host die Übersetzung einer Adresse eines anderen Hosts anfordert, der zum gleichen lokalen ISP gehört, kann der lokale Name-Server die angeforderte IP-Adresse sofort bereit stellen.

Root-Name-Server

Root-Name-Server: Im Internet gibt es etwa ein Dutzend Root-Name-Server, die größtenteils in Nordamerika stehen. Wenn ein lokaler Name-Server eine Anfrage von einem Host nicht direkt beantworten kann, weil er keinen Eintrag für den angeforderten Hostnamen hat, verhält sich der lokale Name-Server seinerseits wie ein DNS-Client und fragt bei einem der Root-Name-Server an. Ist der betreffende Hostname bei dem Root-Name-Server verzeichnet, sendet dieser eine DNS-Antwortnachricht an den lokalen Name-Server, und der lokale Name-Server sendet dann eine DNS-Antwort an den anfragenden Host.

Autoritative Name-Server

Autoritative Name-Server: Jeder Host ist bei einem **autoritativen** Name-Server registriert. Normalerweise ist der autoritative Name-Server für einen Host ein Name-Server beim lokalen ISP. Ein Name-Server ist der autoritative Name-Server für einen Host, wenn er **ständig** über einen DNS-Eintrag verfügt, der den Hostnamen dieses Hosts in seine IP-Adresse übersetzt. Erhält ein autoritativer Name-Server eine Anfrage von einem Root-Server, reagiert der autoritative Name-Server mit einer DNS-Antwort, in der sich die angeforderte Übersetzung befindet.

Viele Name-Server fungieren zugleich als lokale und als autoritative Name-Server.

Häufig ordnet man autoritative Name-Server 1:1 einer Domäne zu, das muss aber nicht so sein. Die Topologie der Name-Server muss nicht mit der hierarchischen Struktur des Namensraumes übereinstimmen.

Funktionsweise der Namensauflösung

Zweistufiger Auflösungsmechanismus

- Der Client kontaktiert seinen lokalen Name-Server.
- Wenn keine lokale Namensauflösung möglich ist, wird die Hierarchie durchlaufen.

In der Praxis werden in vielen Fällen immer dieselben Namen benötigt. Deshalb ist eine signifikante Effizienzsteigerung durch Caching im lokalen Name-Server möglich.

Algorithmus zur Namensauflösung (1)

- Die DNS-Client-Software heißt "*name resolver*„.
- Der *name resolver* kennt die Adresse von mindestens einem Name-Server. Dies ist der lokale Name-Server, meist ein Blatt-Knoten in der Baumstruktur des verteilten DNS-Systems.
- Der *name resolver* baut eine Anfrage-PDU auf ("domain name query") und sendet sie an den Name-Server. Dabei verlangt er entweder "recursive resolution" oder "non-recursive resolution".
- Der Name-Server prüft, ob er die Anfrage lokal beantworten kann.
  - Falls ja, sendet er die Antwort an den Client.
  - Falls nein und "**recursive resolution**" verlangt ist, kontaktiert er einen oder mehrere weitere Name-Server im Baum, bis er die Antwort hat. Jeder Name-Server muss mindestens einen Root-Server kennen (mit IP-Adresse und DNS-Port). Er selbst leitet dann die Antwort an den Client weiter.

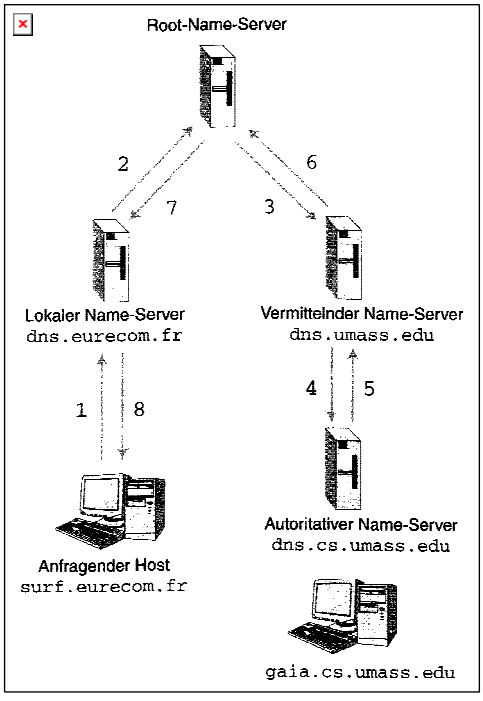
Algorithmus zur Namensauflösung (2)

  - Falls nein und "**iterative resolution**" verlangt ist, meldet er dem Client den Namen eines anderen Name-Servers, den er versuchen könnte.
- Jeder Name-Server hat einen Cache für Einträge, die von einem anderen Name-Server geholt wurden. Die Cache-Einträge werden mit einem Time-Out versehen („time-to-live"). Der Timeout löscht selten verwendete Einträge (typischer Timer-Wert: 2 Tage). Wird ein gesuchter Eintrag im Cache gefunden, so erhält der Client diese Information zusammen mit der Adresse des für den Eintrag zuständigen name-Servers im Baum.
- Manche *name resolver* haben eigene Caches.

Root-Server erhalten in der Regel iterative Anfragen, alle anderen Name-Server rekursive Anfragen.

Beispiel für eine rekursive DNS-Anfrage

`surf.eurecom.fr`
fragt nach der IP-Adresse von
`gaia.cs.umass.edu`



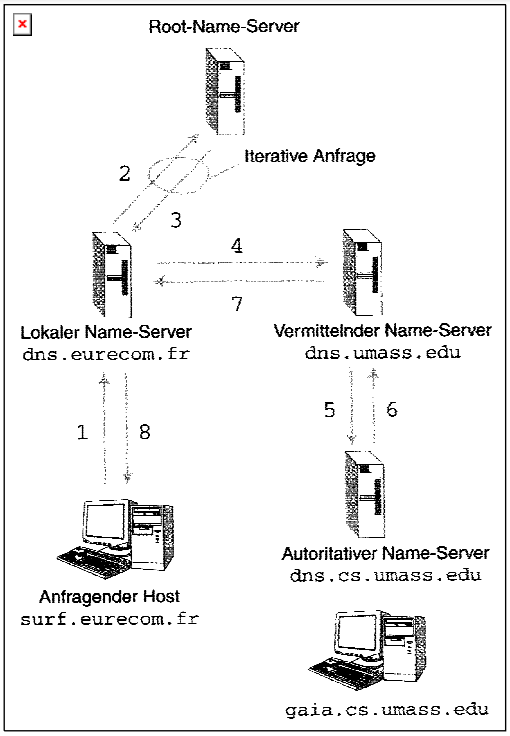
Merke:

Eine **rekursive** Anfrage entspricht einem rekursiven Prozeduraufruf. Auf dem Rückweg folgen die Antworten dem Pfad der Anfrage.

Beispiel für eine iterative DNS-Anfrage

Eine **iterative** Anfrage wird für den Fall, dass der Server sich nicht auflösen kann, nicht mit der gesuchten IP-Adresse (nach entsprechenden Rückfragen) beantwortet, sondern mit der IP-Adresse des nächsten DNS-Servers in der Kette. Der anfragende Client muss dann selbst seine Anfrage dorthin senden.

Es kann auch entlang einer Kette eine Kombination aus rekursiven und iterativen Anfragen verwendet werden.



Format der DNS-Einträge (resource records) (1)

Die Name-Server, die zusammen die verteilte DNS-Datenbank implementieren, speichern so genannte **Resource-Records (RR)** für die Übersetzung von Hostnamen in IP-Adressen. Jede DNS-Antwortnachricht enthält einen oder mehrere Resource-Records.

Ein Resource-Record ist ein 5-Tupel, das folgende Felder enthält:

(*Name, Wert, Typ, Class, TTL*)

Die Bedeutung von *Name* und *Wert* hängen wie folgt von *Typ* ab:

- Wenn *Typ*=A, dann ist *Name* ein Hostname und *Wert* die IP-Adresse des Hosts. Folglich ermöglicht ein Record vom Typ A die Übersetzung eines Standard-Hostnamens in die IP-Adresse.
- Wenn *Typ*=NS (name server), dann ist *Name* eine Domäne (z. B. cnn.com) und *Wert* der Hostname eines autoritativen Name-Servers, der weiß, wie er die IP-Adressen für Hosts in dieser Domäne finden kann. Dieser Record wird benutzt, um DNS-Anfragen entlang der Abfragekette weiter zu leiten.

Format der DNS-Einträge (resource records) (2)

- Wenn *Typ*=CNAME (canonical name), dann ist *Wert* ein kanonischer Hostname für den Alias-Hostnamen *Name*. Dieser Record kann anfragenden Hosts den kanonischen Namen zu einem gegebenen Hostnamen liefern.
- Wenn *Typ*=MX (mail exchange), dann ist *Wert* der Hostname eines Mail-Servers, der einen Alias-Hostnamen *Name* hat. Beispielsweise ist (cnn.com, mail.bar.cnn.com, MX) ein MX-Record. MX-Records ermöglichen es, Hostnamen von Mail-Servern einfache Aliasnamen zu geben.

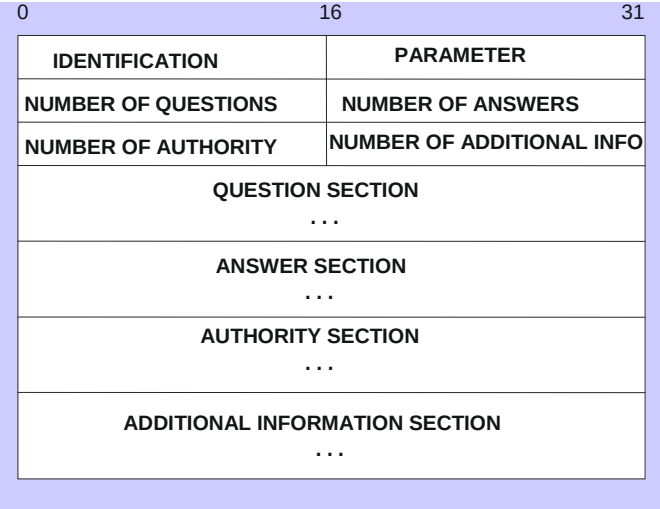
Wenn ein Name-Server für einen bestimmten Hostnamen autoritativ ist, dann enthält der Name-Server einen A-Record für den Hostnamen.

Format der DNS-Einträge (resource records) (3)

Das Feld ***Class*** wurde eingeführt, um auch anderen Netzen als dem Internet die Möglich zur Nutzung des DNS zu geben. Für das Internet ist class=IN.

TTL (Time To Live) ist die Lebenszeit eines Eintrags; sie bestimmt die Zeit, nach deren Ablauf der Eintrag aus einem Cache eines anderen Rechners entfernt werden soll.

Die Protokolldateneinheit des DNS



Format von DNS-Nachrichten (1)

- Die ersten 12 Bytes bilden den **Header**. Das erste Feld ist eine 16-Bit-Nummer, die die Anfrage identifiziert. Dieser **Identifizierer** wird in die Antwortnachricht auf eine Anfrage kopiert, so dass der Client ankommende Antworten mit gesendeten Anfragen abstimmen kann. Im **Parameter**-Feld stehen mehrere Flags. Das 1-Bit-Flag *Query/Reply* informiert darüber, ob es sich bei der Nachricht um eine Anfrage (0) oder eine Antwort (1) handelt. Das 1-Bit-Flag *autoritativ* wird in einer Antwortnachricht gesetzt, wenn ein Name-Server der autoritative Server für einen angefragten Namen ist. Das 1-Bit-Flag *Recursion Desired* wird gesetzt, wenn ein Client (Host oder Name-Server) wünscht, dass der Name-Server einer Rekursion ausführt, falls er den gesuchten Eintrag nicht hat. Das 1-Bit-Flag *Recursion Available* wird in einer Antwort gesetzt, wenn der Name-Server Rekursion unterstützt. Ferner enthält der Header vier Felder *Number of* .... Diese Felder bezeichnen die Anzahlen, mit denen die vier Arten von „Daten“-Feldern vorkommen, die dem Header folgen.

Format von DNS-Nachrichten (2)

- Die **Question Section** enthält Informationen über die gestellte Anfrage. Dieser Abschnitt beinhaltet ein Namensfeld, in dem der angefragte Name steht, und ein Typfeld, in dem der Typ des gesuchten Namens angegeben wird (z. B. Typ A für eine Hostadresse).
- In einer Antwort von einem Name-Server enthält die **Answer Section** die Resource-Records des Namens, der ursprünglich angefragt wurde. Eine Antwort kann mehrere RRs umfassen, weil ein Hostname auf mehrere IP-Adressen abgebildet werden kann (z. B. replizierte Web-Server).
- Die **Authority Section** enthält Records anderer autoritativer Server.
- Die **Additional Information Section** enthält weitere „nützliche“ Records. Das Antwort-Feld in einer Antwort auf eine MX-Anfrage enthält z. B. den Hostnamen eines Mail-Servers in Bezug zum Aliasnamen. In diesem Fall steht im Additional-Abschnitt ein A-Record, der die IP-Adresse für den kanonischen Hostnamen des Mail-Servers liefert.

Übertragung der DNS-Nachrichten

DNS funktioniert grundsätzlich über TCP oder UDP.

In der Regel wird UDP verwendet, bei Paketverlust erfolgt eine Übertragungswiederholung nach einem Timeout.

TCP wird nur verwendet:

- wenn DNS-Pakete größer als 512 Bytes sind
- beim Initialisieren von “secondary name servers“ durch die Übertragung aller Daten vom “primary name server“.