Lehrstuhl für Netzarchitekturen und Netzdienste
Institut für Informatik – Technische Universität München
Prof. Dr.-Ing. Georg Carle

Grundlagen: Rechnernetze und Verteilte Systeme

Kapitel 1: Einführung und Motivation

Trends, Internet, Nutzer, Leitbeispiel, Literatur

Prof. Dr.-Ing. Georg Carle
Lehrstuhl für Netzarchitekturen und Netzdienste
Technische Universität München
carle@net.in.tum.de
http://www.net.in.tum.de

Acknowledgements

- Die vorliegenden Unterlagen sind im Laufe mehrerer Vorlesungen an den Universitäten Karlsruhe (Prof. Krüger, Prof. Juling, Prof. Zitterbart), Kiel (Prof. Schiller), Braunschweig (Prof. Zitterbart), FU Berlin (Prof. Schiller), Bern (Prof. Braun) entstanden. Zusätzliche Inhalte stammen von Vorlesungen an der Universität Paderborn (Prof. Karl), der Kansas University (Prof. Sterbenz) und der Universität Tübingen (Prof. Küchlin).
  Die Vorlesungsunterlagen beinhalten auch Material diverser Firmenveröffentlichungen, Internet-Quellen etc. Zahlreiche Autoren haben hierzu beigetragen, welche im Einzelnen gar nicht mehr alle genannt werden können. Daher ohne Namensnennung ein großer Dank an alle, die im Laufe der Jahre etwas zu diesen Folien beigetragen haben!

- Bei Fragen, Anregungen, Kommentaren zu diesen Folien bitte eine Email an carle@in.tum.de!

Grundlegende Bücher für diese Vorlesung

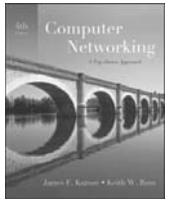
- Andrew S. Tanenbaum:
  - *Computer Networks*
    Prentice-Hall, **4th edition** 2003
    ISBN-10: 0130661023, 80 €
  - (Wurde - nicht fehlerfrei und z.T. eher schwer lesbar - auch ins Deutsche übersetzt:
    Computernetzwerke,
    Pearson Studium; 50 €, 4. Auflage 2003
    ISBN-10: 3827370469)

- Gerhard Krüger & Dietrich Reschke:
  - *Lehr- und Übungsbuch Telematik* Fachbuchverlag Leipzig im Carl-Hanser-Verlag, 3. Auflage, 2004
    ISBN 3-446-22073-9, < 30 €
  - gute Erläuterung von Teilen der Vorlesung

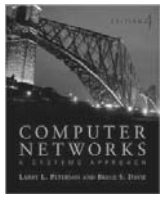
- Sebastian Abeck, Peter Lockemann, Jochen Seitz, Jochen Schiller
  - *Verteilte Informationssysteme*
    dpunkt.verlag, 2002
    ISBN 978-3-89864-188-3, 49 €
  - Stellt eine leicht zu lesende Erläuterung von Teilen der Vorlesung zur Verfügung

Weitere empfehlenswerte Bücher

- J. F. Kurose & K. W. Ross, *Computer Networking: A Top-Down Approach Featuring the Internet*, 2007, 4th edition, Addison Wesley
  - Innovation: Erläuterung der Protokolle Top-Down, beginnend mit der Anwendungsebene
  - Vorstellung von Schlüsselpersonen auf dem Gebiet Rechnernetze
  - **Deutsche Übersetzung: Computernetzwerke: Der Top-Down-Ansatz, Pearson Studium; 30/60 €, 4. Auflage 2008 ISBN-10: 3827373301**

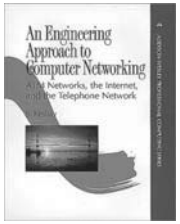


- L. L. Peterson & B. S. Davie, *Computer Networks – A Systems Approach*, 2007, 4th edition, Morgan Kaufman
  - Technisch und fundiert
  - Zahlreiche Beispiele

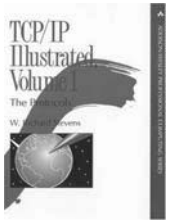


Weitere Buchempfehlungen

- S. Keshav: *An Engineering Approach to Computer Networking*. Addison-Wesley, 1999
  - Sehr gute quantitative Behandlung von Rechnernetzen
  - Erläutert zahreiche Entwurfsentscheidungen



- W.R. Stevens: *TCP/IP Illustrated, Vol. 1- 3,* 1994, Addison-Wesley
  - Erläutert sehr detailliert die Implementierung von TCP/IP



Übersicht

Entwurfsprinzipien für Telekommunikationssysteme (Schalttechnik leicht gemacht, Beispiel Beirut)

Ziel: transparente Kabelführung gemäß Struktur des Netzes

Schalttechnik leicht gemacht, Beispiel Beirut

Ziele:
- Präzise Dokumentation an jeder Leitung, um schnellen Zugriff auf jeden Anschluss zu gewährleisten.
- Bauweise des Gehäuses schütz Technik und verhindert Manipulation.

Schalttechnik leicht gemacht, Beispiel Beirut

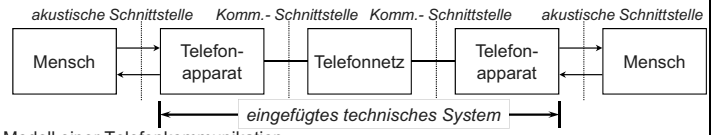
Ziel: straffe Schaltdraht-führung und Übersichtlichkeit erleichtert Reparaturen.

Historie: Kommunikationstechnologien

Jahr	*Innovation*	*Leistung*
1840	Morse-Telegraf	Elektronischer Nachrichtenaustausch über größere Distanzen
1861	Telefon (Reiss)	Sprachkommunikation (unidirektional) über über größere Distanzen
1876	Telefon (Bell)	Patentierung des Telefons (bidiektional)
1887	elektromagn. Wellen	Funktechnik
1892	Automatischer Drehwähler	Automatisierung der Telefonvermittlung (→ Ablösung des "Fräuleins vom Amt")
1923	Rundfunk	Massenkommunikation
1929	Koaxialkabel	Höhere Datenraten
1964	Nachrichtensatelliten	Grundlage für globale Kommunikation
1966	Glasfaser	extreme Steigerung der Datenraten
1969	ARPANET Knoten	Paketvermittlung
1973	Ethernet	Lokale Netze mit hohen Datenraten
1984	Deregulierung (*USA*)	Aufhebung des Fernmeldemonopols
1990	WWW	Architektur und Protokoll für Hypertext-Anwendung
1997	WDM (*Wavelength Division Multiplex*)	Steigerung der Datenraten auf Glasfaserstrecken auf bis zu 1 Terabit/s (Tera = 10^{12})

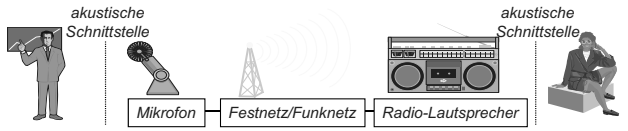
Kommunikation mit technischer Mitteln - Telekommunikation

- Die klassische Nachrichtentechnik / Telekommunikationstechnik ist von der Sprachkommunikation (Telefon) geprägt - technisch und wirtschaftlich
- Menschen als Kommunikationspartner:



Modell einer Telefonkommunikation

⇨ Das technische System wird in den - ansonsten weitgehend unveränderten - Kommunikationsablauf eingefügt.



Modell einer Rundfunkkommunikation

Computergestützte Telekommunikation

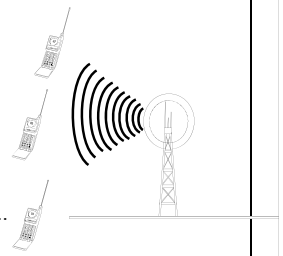
- Digitale Telekommunikation
  - Digitalisierung aller Kommunikationsformen (Gesprochene Sprache, Musik, Text, Grafik, Festbild, Bewegtbild (z.B. Video), Technische Daten)
  - Ausrichtung auf Multimedia (Integration mehrerer Kommunikationsformen) vorzugsweise für den Menschen als Empfänger

- Grundlage: Computer-Computer-Kommunikation
  - Digitale Telekommunikation ist auf Mikroelektronik/Computer-Basis und durch Hard-/Software-Systeme realisiert.
  - Moderne Telekommunikationsnetze (unter Einschluss der Endgeräte) sind Computernetze (Computer Networks).

Entwicklungstrend: Mobile Kommunikation

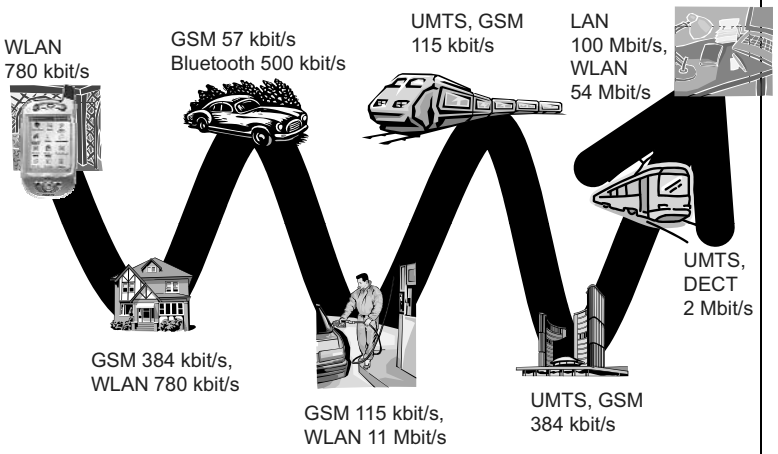
- „Jedermann, zu jeder Zeit, an jedem Ort (mit jeder Kommunikationsform)“

  anybody, anytime, anywhere



- Schrittmacherrolle: Mobiltelefonie
  - derzeit bereits über 2 Milliarden Nutzer
  - Festnetztelefonie bereits übertroffen
    - ebenso das „feste“ Internet
  - hohe Kosten einer drahtgebundenen Anschlussinfrastruktur
- Ziel:
  - Übertragung von Sprache, Daten, Audio, Video ...
- Mobilitätsaspekte
  - Gerätemobilität (Standortwechsel des Geräts möglich)
  - Benutzermobilität (Kommunikation von beliebigem Standort, z.T. über unterschiedliche Geräte)

Mobile and Wireless Web Services – Always Best Connected

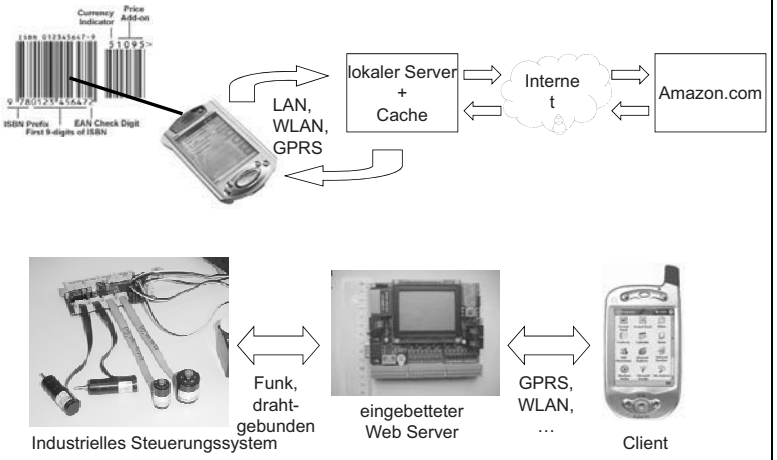


Entwicklungstrend: Kommunikation von Geräten

- **Heute**:
  - Telekommunikation zwischen Menschen im Vordergrund
- **Zukünftig**:
  - Technische Geräte / technische Systeme kommunikationsfähig „Internet of Things“

- Beispiele:
  - Produktionseinrichtungen
    Tele-Diagnose, Tele-Wartung, Tele-Betrieb
  - Kommunikation in/mit Fahrzeugen
    u.a. Verkehrstelematik
  - Hausnetze
    Sicherheit, Haushaltsgeräte-Kommunikation, Heizungssteuerung, usw.
  - Sensor-Netze
    häufig für Überwachungaufgaben

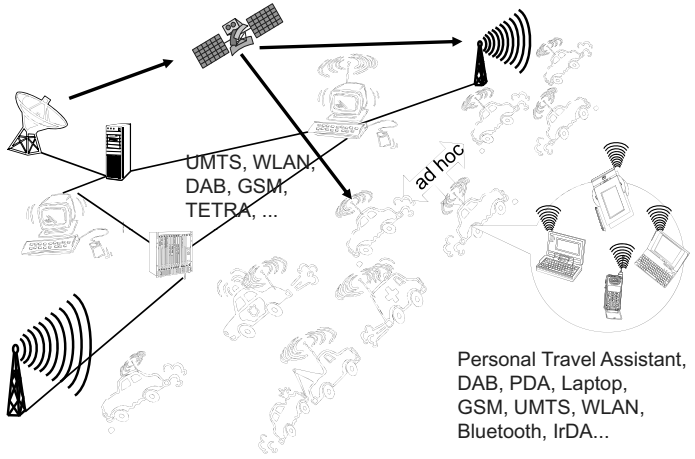
Beispielszenarien



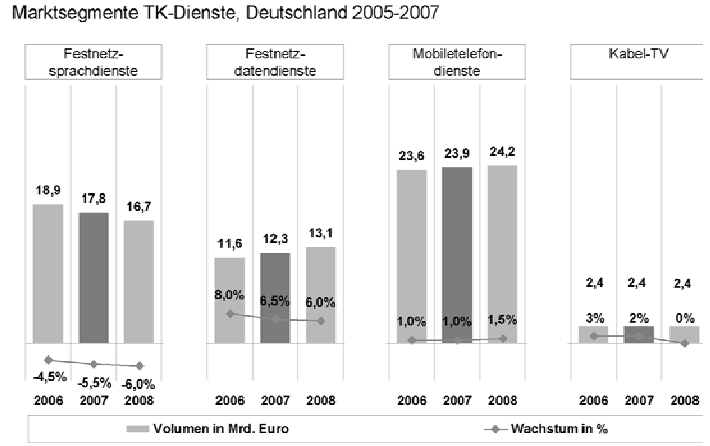
Entwicklungstrend: Ubiquitäre Informationstechnologien

- Ubiquität („Allgegenwärtigkeit“):
  - Nichtgebundensein an einen Standort
  - Information als überall erhältliches Gut
  - ⇨ Information Technology (IT) beyond the PC
- Persönliche Technologien
  - Zugang zu IT-Diensten mit sich herumtragen
  - Beispiele: Persönliche Digitale Assistenten (PDAs), Wearable Devices
- Informationsumgebungen
  - Zugang zu IT-Diensten überall vorhanden
  - Beispiele: Intelligente, kommunikationsfähige Geräte/Systeme, Aktive Gebäude (cooperative buildings)
- Allgemeine Entwicklungstendenz
  - früher: Viele Menschen an einem Computer
  - heute: Ein Computer pro Person
  - bald: Viele Computer pro Person
- Ubiquitäre Unterstützung
  - wirkt im Hintergrund,
  - wird selbst aktiv,
  - (teil-)autonom von Menschen.

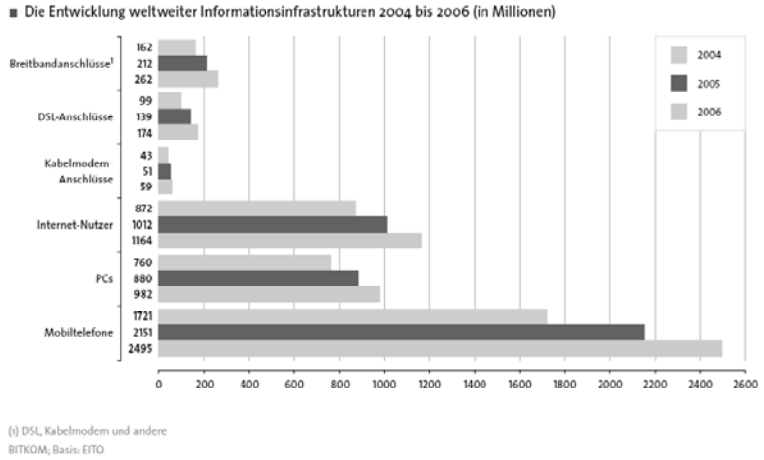
Entwicklungstrends in der Übersicht



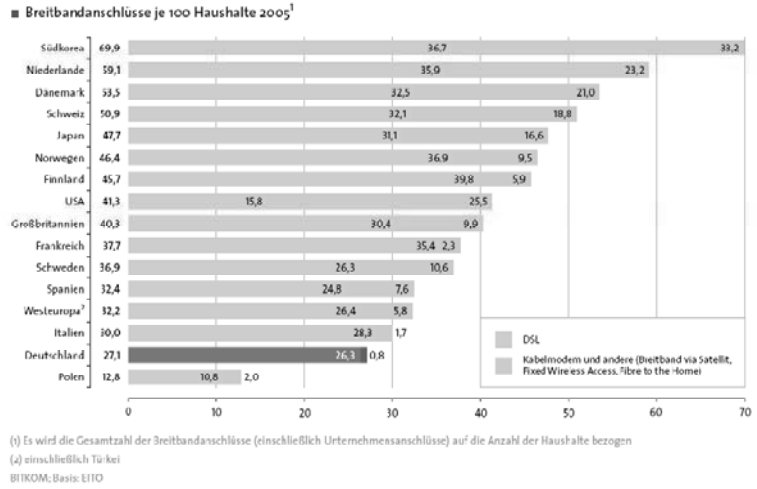
Telekommunikationsdienste



Informationsinfrastruktur

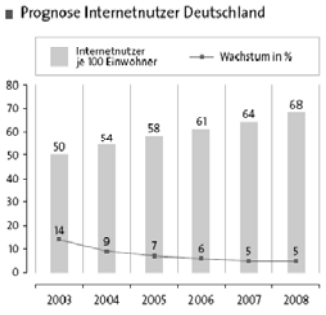


Breitbandanschlüsse

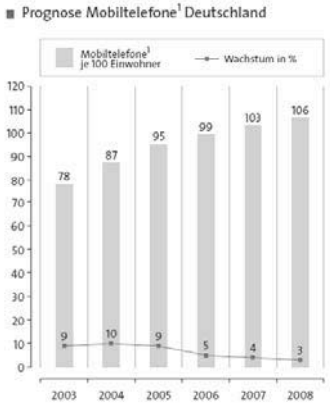


(1) Es wird die Gesamtzahl der Breitbandanschlüsse (einschließlich Unternehmensanschlüsse) auf die Anzahl der Haushalte bezogen
(2) einschließlich Türkei
BITKOM; Basis: EITO

Internet und Mobilkommunikation in Deutschland

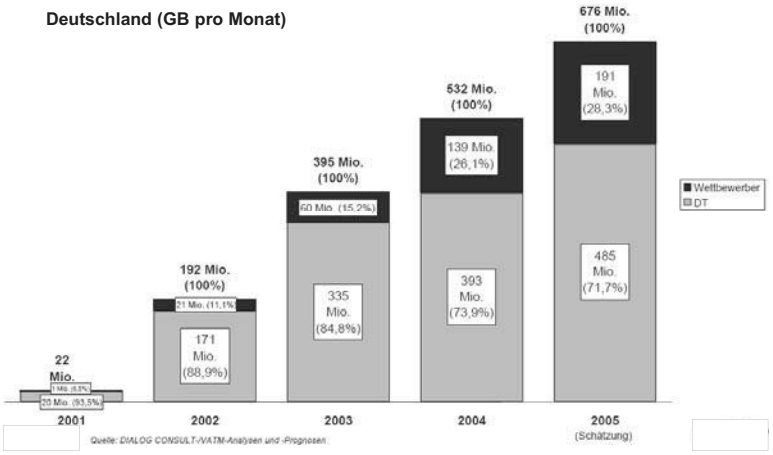


BITKOM; Basis: EITO



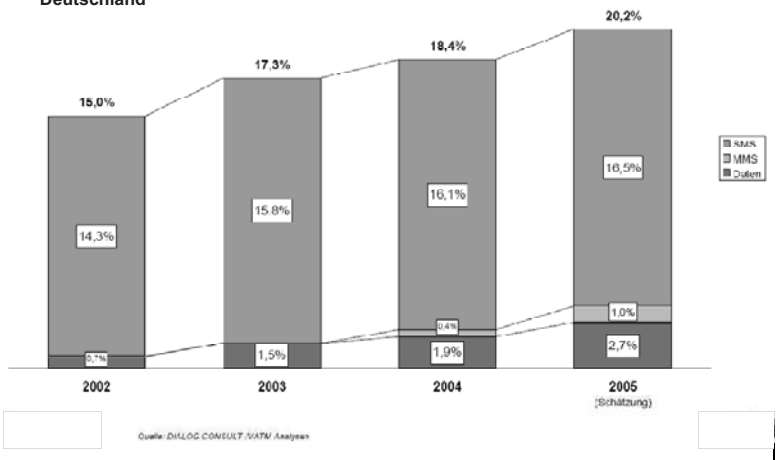
BITKOM; Basis: EITO

Volumenentwicklung Breitband-Internet-Verkehr



Quelle: DIALOG CONSULT-/VATM-Analysen und -Prognosen

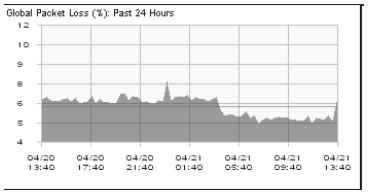
Datenanteil an den Diensteumsätzen im Mobilfunk

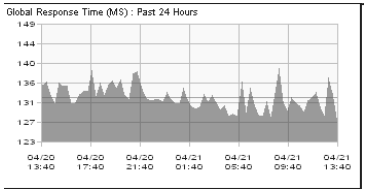


Quelle: DIALOG CONSULT-/VATM-Analysen

Qualität des Internet

- Zahlreiche Projekte führen Internet-Messungen durch, z.B. Internet Traffic Report: http://www.internettrafficreport.com/





Lehrstuhl für Netzarchitekturen und Netzdienste
Institut für Informatik – Technische Universität München
Prof. Dr.-Ing. Georg Carle

Grundlagen: Rechnernetze und Verteilte Systeme

Kapitel 2: Begriffswelt und Standards

Dienst, Protokoll, Automat, IETF, ITU, IEEE

Prof. Dr.-Ing. Georg Carle
Lehrstuhl für Netzarchitekturen und Netzdienste
Technische Universität München
carle@net.in.tum.de
http://www.net.in.tum.de

Übersicht

Ziele

- In diesem Kapitel wollen wir vermitteln
  - Grundlegende Begriffe
  - Kommunikationsprobleme
  - Funktionsweise der Nachrichtenübermittlung
  - Geschichtete Kommunikationsmodelle
  - Formale Protokollspezifikation

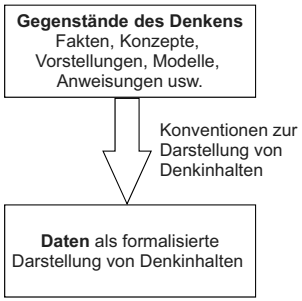
Kapitelgliederung

2.1. Grundlegende Begriffe - Der Begriff „Daten“

- Daten
  - Was wird dargestellt? Und wie?
  - *Darstellung*
    von Fakten, Konzepten, Vorstellungen und Anweisungen
    in formalisierter Weise,
    geeignet für
    - Kommunikation,
    - Interpretation und die
    - Verarbeitung

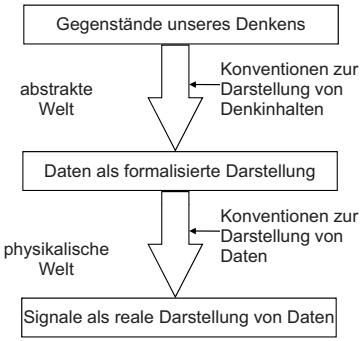
    durch Menschen und/oder technische Mittel.
  - Allgemeine Beispiele für Datendarstellungen:
    - gesprochene Sprache
    - Zeichen-/Gebärden-Sprache
    - geschriebene Sprache
  - Datenkommunikation:
    Datenaustausch über immaterielle Träger (Energieflüsse, meist elektrische Ströme, elektromagnetische Wellen)
    und größere Entfernungen
    zwischen Menschen und/oder Maschinen

Modell zur Erzeugung von Daten durch den Menschen:



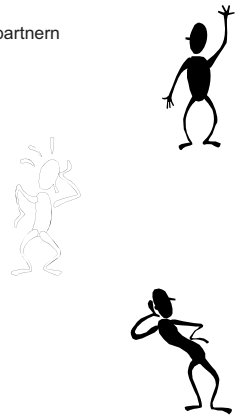
Der Begriff „Signal“

- Signal
  - Ein Signal ist die *physikalische Darstellung (Repräsentation)* von Daten durch charakteristische räumliche und/oder zeitliche Veränderungen der Werte physikalischer Größen.
  - Signale sind somit die *reale physikalische Repräsentation* abstrakter Darstellungen der Daten
    Beispieldarstellungen:
    - Sprache, 8 Bit PCM codiert
    - Text als ASCII-Character



2.2. Grundlegende Problemstellungen der Kommunikation

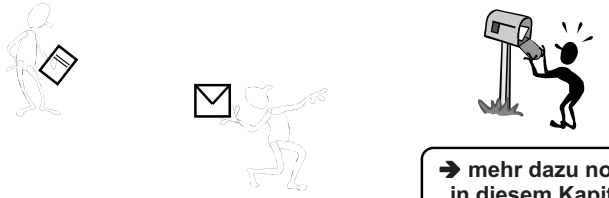
- Regelung des Kommunikationsablaufs
  ➔ Protokolle, Protokollschichten
- Ressourcenverteilung bei mehreren Kommunikationspartnern
  ➔ Vielfachzugriff (Multiple Access)
- Kommunikation über Zwischenknoten
  ➔ Vermittlung (Switching)
- Abarbeitung paralleler Kommunikationsvorgänge
  ➔ Scheduling
- Identifikation von Kommunikationspartnern
  ➔ Namen und Adressen
- Wahl des besten Kommunikationspfades
  ➔ Routing
- Umgang mit Übertragungsfehlern
  ➔ Fehlerkontrolle (Error Control)
- Anpassung der Übertragungsgeschwindigkeit
  ➔ Flusskontrolle (Flow Control)



Protokolle, Protokollschichten

- Definition einer gemeinsamen Sprache und Anwendung vereinbarter Abläufe

- Protokollschichten ermöglichen Arbeitsteilung



➔ mehr dazu noch in diesem Kapitel

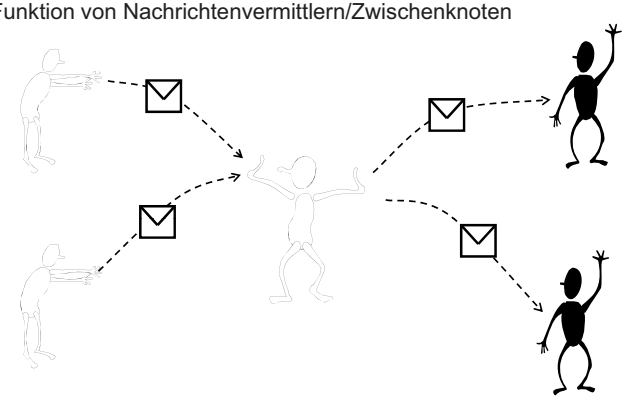
Vielfachzugriff (Multiple Access)

- Regelung des Zugriffs auf gemeinsames Medium zur Vermeidung von Störungen und Kollisionen

➔ mehr dazu in Kapitel 3

Vermittlung (Switching)

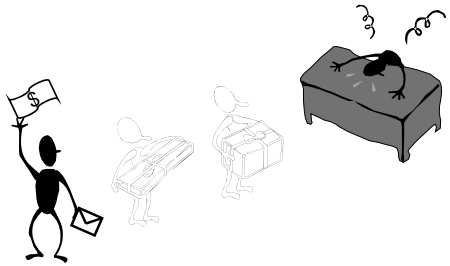
- Funktion von Nachrichtenvermittlern/Zwischenknoten



➔ mehr dazu in Kapitel 4

Scheduling

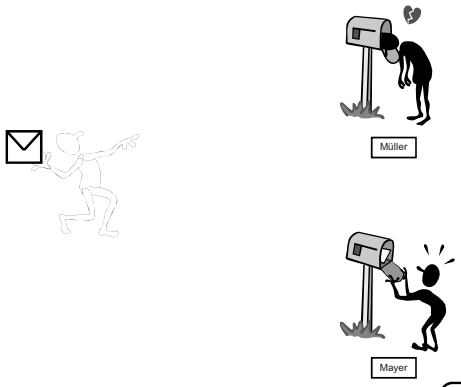
- Bestimmung der Abarbeitungsreihenfolge für verschiedene Aufgaben



➔ mehr dazu in Kapitel 7

Namen und Adressen

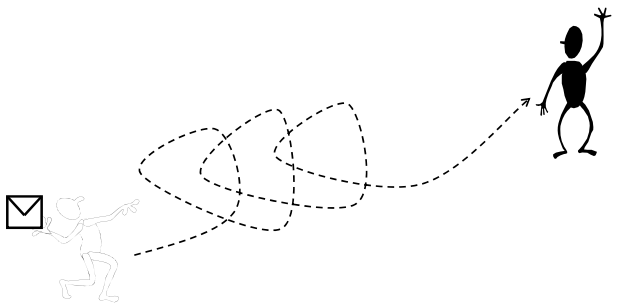
- Bestimmung des Empfängers und ggf. auch des Absenders



➔ mehr dazu in Kapitel 4

Wegewahl (Routing)

- Auffinden des günstigsten Pfades zum Empfänger



➔ mehr dazu in Kapitel 4,5

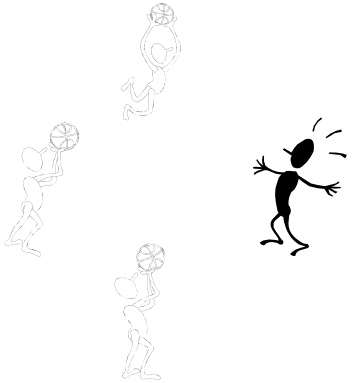
Fehlerkontrolle

- Erkennen und Behebung von Übertragungsfehlern

➔ mehr dazu in Kapitel 3,6

Flusskontrolle

- Anpassung der Übertragsgeschwindigkeit an die Empfangsfähigkeiten des Empfängers



➔ mehr dazu in Kapitel 6,7

2.3. Charakterisierung von Kommunikationsvorgängen

- Ein Kommunikationsvorgang kann aufgrund folgender Kriterien charakterisiert werden:

  (1) Beteiligten Kommunikationspartner (KP)

  (2) Übertragungsverfahren/Schnittstellen

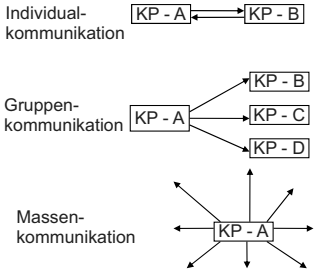
  (3) Nutzungsrichtung

  (4) Auslieferungsdisziplin

  (5) Qualität

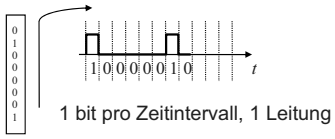
(1) Beteiligte Kommunikationspartner (KP)

- Akteure
  - Mensch-Mensch
  - Mensch-Maschine
  - Maschine-Maschine
- Menge der Kommunikationspartner
  - Dialog (*Unicast*): Zwei Partner tauschen über eine Punkt-zu-Punkt-Kommunikationsstrecke Daten aus.
  - Gruppenruf (*Multicast*): Ein Kommunikationspartner spricht gleichzeitig mehrere empfangende Kommunikationspartner an.
  - Rundruf (*Broadcast*): Es werden von einem Kommunikationspartner sehr viele (in der Regel unbekannte) Empfänger angesprochen, potentiell alle (Rundfunk).
  - *Anycast*: Ein beliebiger Kommunikationspartner einer Gruppe wird angesprochen.
  - *Concast*: viele Kommunikationsknoten senden an einen Einzelnen.

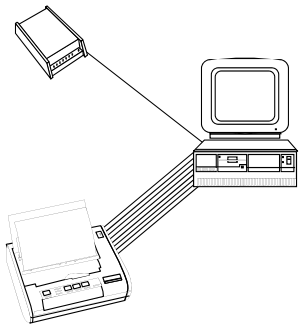


(2) Übertragungsverfahren/Schnittstellen

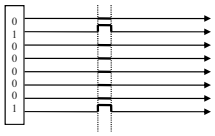
- Serielle Übertragung



1 bit pro Zeitintervall, 1 Leitung

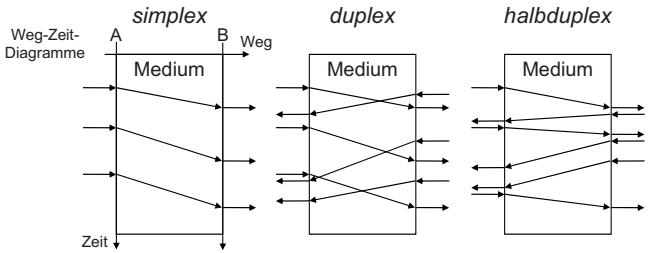


- Parallele Übertragung



8 bit pro Zeitintervall, 8 Leitungen

(3) Verbindungseigenschaften: Nutzungsrichtung



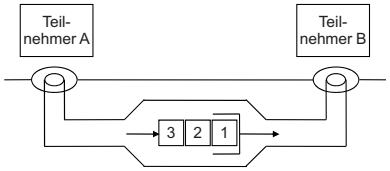
- Feuermelder
- Sensoren
- Pager

- Telefon

- Wechselsprechen
- Datenkommunikation mit geteilten Medien

(4) Auslieferungsdisziplin

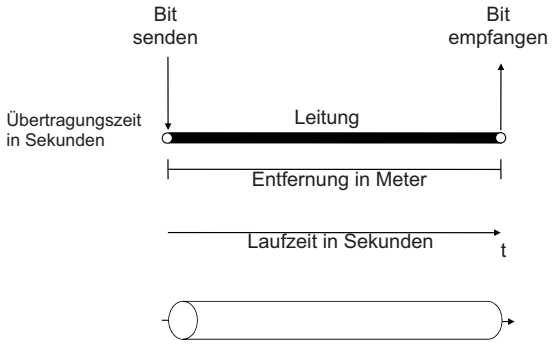
- Die Auslieferungsdisziplin beschreibt die Reihenfolge der beim Empfänger ankommenden Daten in Bezug auf die Reihenfolge, wie sie abgeschickt wurden:
  - treu zur Einlieferungsreihenfolge (FIFO)
  - FIFO + priorisiert
  - keine Reihenfolgentreue garantiert



(5) Qualität

- Bezüglich Qualität sind folgende Eigenschaften von Kommunikationsdiensten zu betrachten:
  - Technische Leistung
    - Antwortzeit, Durchsatz, Sende-/Empfangsrate, ...
  - Kosten
    - Investitionskosten, Betriebskosten, ...
  - Zuverlässigkeit
    - Fehlertoleranz, Ausfallsicherheit, Störunanfälligkeit, Verfügbarkeit, ...
  - Schutz
    - Abhörsicherheit, Manipulationssicherheit, Authentifizierung, Autorisierung, Maßnahmen gegen Dienstverweigerung, ...

2.4. Technischer Hintergrund - Technische Leistung

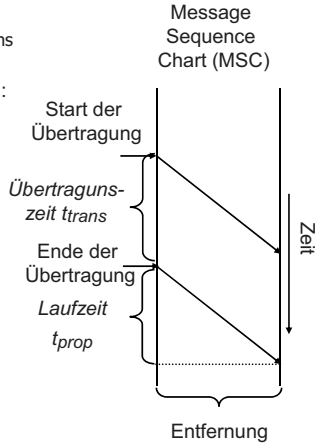


Durchsatz (auch: Bandbreite)
= Anzahl der pro Sekunde übertragenen Bits
[Einheit bit/s]

Bandbreiten-Verzögerungs-Produkt
= Speicherkapazität einer Leitung

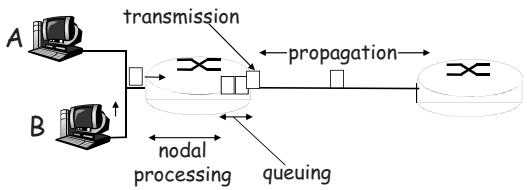
Signalausbreitung im Medium, Datenspeicherung

- Senden einer Nachricht benötigt Übertragungszeit (transmission delay) t_{trans}
  - Übertragungszeit abhängig von Datenrate **r** and Länge der Nachricht N: $t_{trans} = N / r$
- Signale erreichen nach Laufzeit (propagation delay) t_{prop} ihr Ziel
  - Abhängig von Entfernung und Ausbreitungsgeschwindigkeit im Übertragungsmedium
- Über die Laufzeit t_{prop} werden $r * t_{prop}$ bit generiert
  - Gespeichert im Medium
- Gesamtverzögerung:
  $t = t_{trans} + t_{prop} (+ t_{proc} + t_{queue})$
  - t_{proc} : Verarbeitungszeit (processing delay)
  - t_{queue} : Wartezeit (queuing delay)



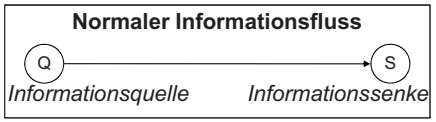
Verzögerungen in paketvermittelten Netzen

Vier unterschiedliche Verzögerungen an jedem Knoten
1) Verarbeitungszeit (processing delay)
2) Wartezeit (queuing delay)
3) Übertragungszeit (transmission delay)
4) Laufzeit (propagation delay)



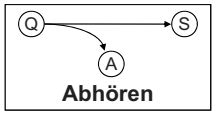
Sicherheitsgefahren und Schutzmaßnahmen

- Schutzmaßnahmen
  - Verschlüsselung (kryptographische Codes)
  - Schaffung vertrauens-würdiger Systeme (Authentisierung, Autorisierung)

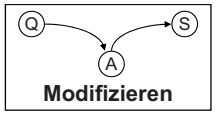


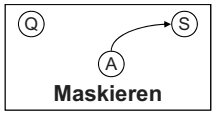
- Angriffe

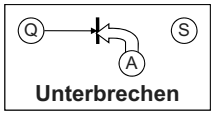
Passiv:



Aktiv:





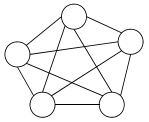


2.5. Kommunikationsarchitekturen

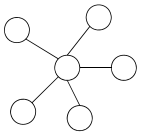
- Zur Realisierung von Kommunikationsvorgängen wird eine Kommunikationsarchitektur benötigt für:
  - physikalische Konnektivität
    Verbindung über Kupferkabel, Lichtwellenleiter, Luftschnittstelle, ...
  - Kommunikationsfunktionalität
    - Steuerung des Ablaufs
    - Adressierung der Kommunikationspartner
    - Garantie einer geforderten Qualität
    - Anpassung unterschiedlicher Formate
    - ...
  - Schnittstelle zu den Anwendungen
- Aufgrund der *unterschiedlichen Aufgaben:*
  - Kommunikationsarchitektur mit geschichtetem Aufbau üblich
  - eine Schicht nutzt die Funktionalität der darunter liegenden Schicht, um ihre eigenen Funktionen zu realisieren

2.5.1. Netztopologien

- vermaschtes Netz
  - voll vermascht:
    - N Knoten
    - N(N-1)/2 Kanten/Verbindungen
    - stets direkte Verbindung zwischen zwei Kanten, zusätzlich N-2 alternative Pfade mit 2 Hops
    - unwirtschaftlich für große N

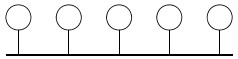


- Sternnetz
  - Kanten mit unterschiedlichen Rollen:
    - Zentraler Vermittlungsknoten
    - Endknoten
  - Grundkonzept eines hierarchischen Netzes
  - N Endknoten ➔ N Kanten/Verbindungen
  - 2 Hops zwischen zwei beliebigen Endknoten
  - keine alternativen Pfade
  - wirtschaftlich für große N

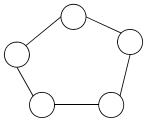


Netztopologien

- Busnetz
  - gemeinsamer Bus als Broadcast-Medium
  - passive Kopplung der Knoten an den Bus
  - Vielfachfachzugriffssteuerung notwendig

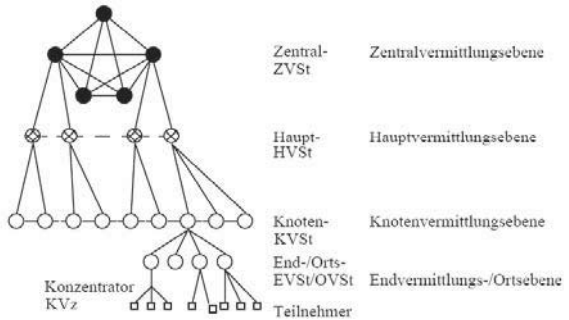


- Ringnetz
  - gemeinsamer Ringbus
  - aktive Kopplung der Knoten an den Bus
  - Kanten/Verbindungen unidirektional (simplex) oder bidirektional (duplex)
  - bidirektionale Verbindungen
    ⇨ zwei unabhängige Pfade zwischen zwei Knoten
  - Vielfachzugriffsteuerung durch reservierte Zeitschlitze (TDM) oder Token

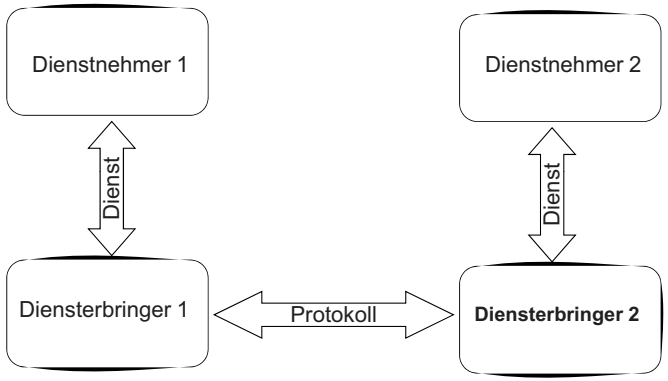


Hierarchische Netztopologien

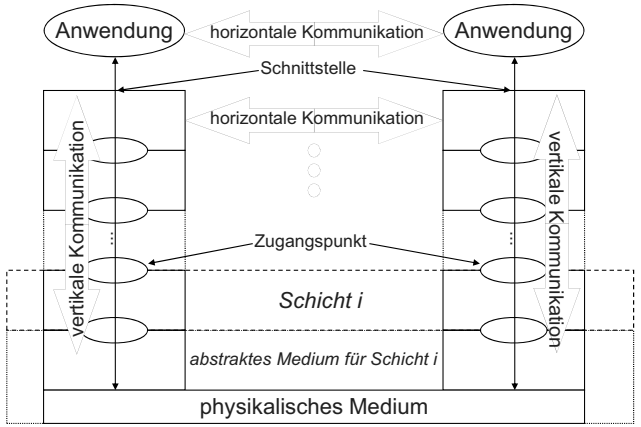
- Beispiel: klassisches Telefonnetz



2.5.2. Dienst und Protokoll - Übersicht



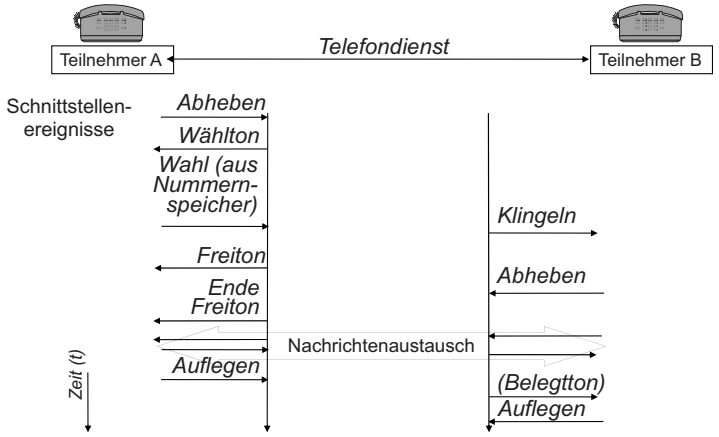
Geschichtetes Kommunikationssystem



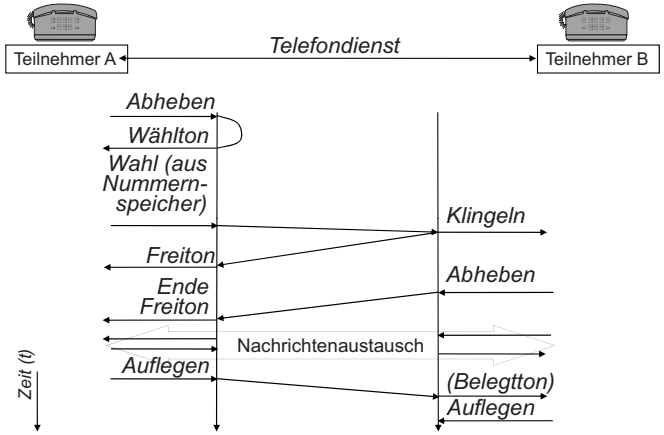
Dienst und Protokoll

- Partner einer Schicht
  - benutzen einen Dienst (außer unterste Schicht)
  - bieten einen Dienst (außer oberste Schicht)
  - brauchen nichts zu sehen / kennen außer direkt unterliegendem Dienst (Konzept der „virtuellen Maschine“)
  - „unterhalten sich“ gemäß Regeln (Protokollen)
    - z.B. „Telefon“-Schicht: wählen/klingeln/besetzt
    - Bei Menschen viel kontextsensitiv / implizit:
      - z.B. „Melden am Telefon“
      - Übersetzer:
        „Übersetz-Modus“, „Rückfragen-Modus“, „Selbst-Vorstellen“, „Chef-Vorstellen“, ...
- Kommunikationsarchitekturen basieren auf
  - „Dienst“ = (Kommunikations-) Dienst [(Communication) Service]
  - „Regeln“ = (Kommunikations-) Protokoll [(Communication) Protocol]

Beispiel Telefon – Dienstnehmersicht (1)

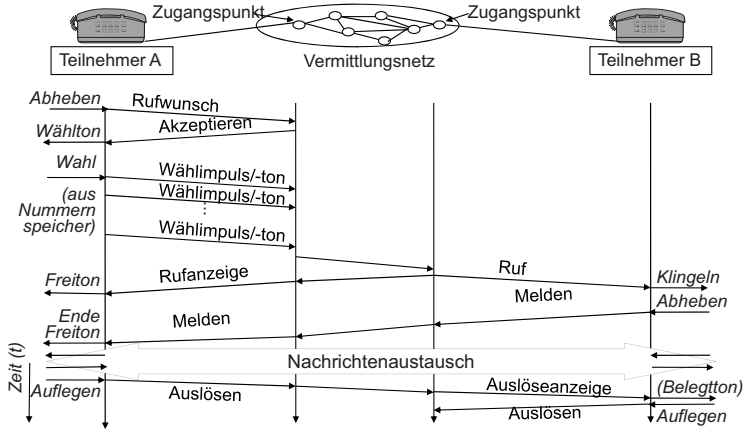


Beispiel Telefon – Dienstnehmersicht (2)



Beispiel Telefon - Dienst und Protokoll

- Signalisierungsprotokoll im alten analogen Fernsprechnetz:



Begriffswelt „Dienst“

- Funktionalität einer Schicht wird als Menge von **Diensten** zur Verfügung gestellt.
- Die Dienste einer Schicht werden durch den Datenaustausch zwischen (Partner-)**Instanzen** erbracht. Dieser Datenaustausch erfolgt gemäß festgelegten Regeln und Formaten, die man **Protokoll** nennt.
- Ein Dienst wird an der **Dienstschnittstelle** einem Dienstbenutzer von einem Diensterbringer angeboten.
- Die **Dienstdefinition** spezifiziert verfügbare Dienste und Regeln für ihre Benutzung (in der darüber liegenden Schicht).
- Ein **Dienstprimitiv** (Schnittstellenereignis) dient zur Anforderung oder Anzeige eines Dienstes beim Dienstbenutzer, Grundtypen sind:
  - Anforderung (Req , Request)
  - Anzeige (Ind , Indication)
  - Antwort (Rsp , Response)
  - Bestätigung (Cnf , Confirmation)

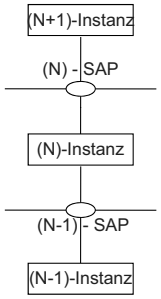
Dienst der Schicht N

- **(N) - Dienst / (N) - Service**
  - Menge von Funktionen, welche die (N)-Schicht den (N+1)-Instanzen an der Schnittstelle zwischen der (N)- und (N+1)-Schicht anbietet (vertikale Kommunikation).
  - Die (N)-Instanzen erbringen die Dienste der (N)-Schicht mit Hilfe von Nachrichtenaustausch (horizontale Kommunikation). Dazu verwenden sie die Dienste der (N-1)-Schicht.
  - Wie die Dienste der (N) - Schicht erbracht werden, bleibt der (N+1) - Schicht verborgen.

(N) - Dienstzugangspunkt / (N) - SAP

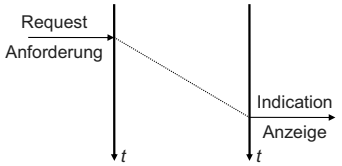
- Innerhalb eines geschichteten Kommunikationssystems kommunizieren (N+1)-Instanzen und (N)-Instanzen über einen **(N)-Dienstzugangspunkt** [(N)-SAP, (N)-Service Access Point] miteinander.
- Die (N)-Instanz bietet die von ihr erbrachten (N)-Dienste der (N+1)-Instanz am (N)-SAP an.
- Die (N)-Instanz benutzt die Dienste, die ihr am (N-1)-SAP angeboten werden.

Beziehungen zwischen (N-1)-SAP, (N)-Instanz und (N)-SAP

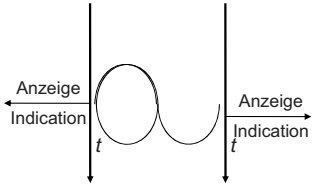


Diensttypen

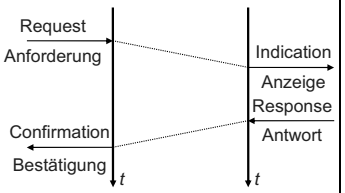
- Unbestätigter Dienst
  - Beispiel: Briefübermittlung
  - Vom Dienstnehmer initiiert:



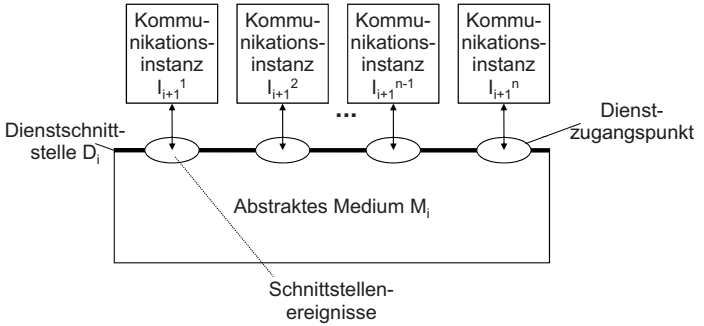
  - Vom Diensterbringer initiiert:



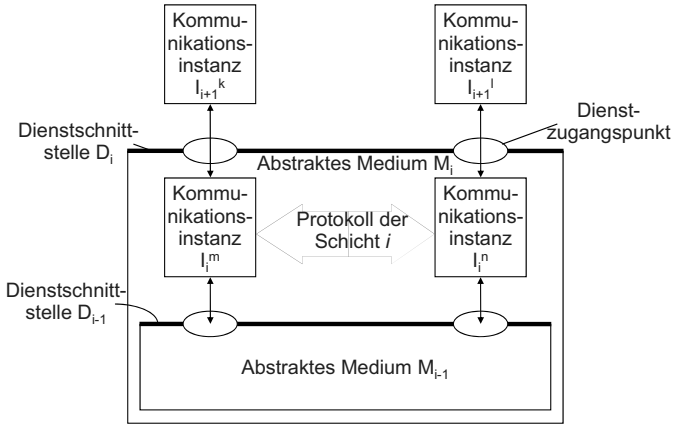
- Bestätigter Dienst
  - Beispiel: Buchung



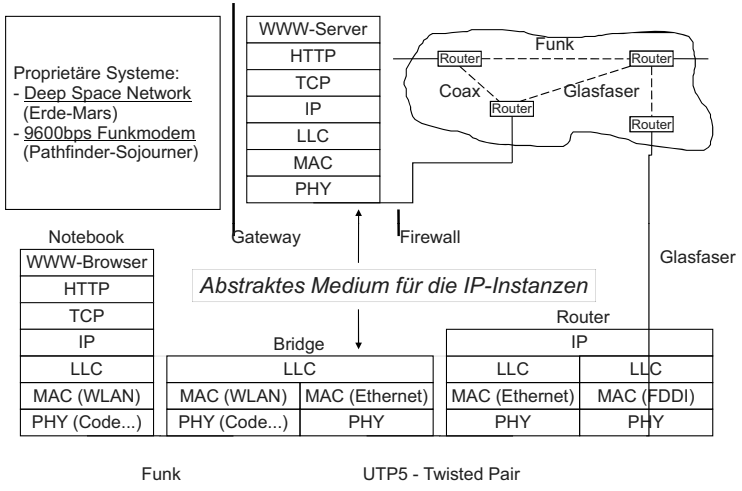
Der Dienstbegriff



Diensterbringung: Protokollablauf



Abstraktes Medium im Beispiel



Verbindungsorientierte vs. verbindungslose Kommunikation

- Verbindungsorientierte Dienste
  - Vor dem Datenaustausch zwischen Dienstnehmern auf Schicht n wird eine Verbindung durch die beteiligten Instanzen der Schicht n-1 aufgebaut
    - Anforderung erfolgt mithilfe entsprechender Dienstprimitive der Schicht n-1
  - Protokollabhängige Aushandlung von Übertragungsparametern
    - z.B. Teilnehmer (immer), Dienstqualität, Übertragungsweg
  - Datenaustausch innerhalb dieser Verbindung erfolgt unter Berücksichtigung des aktuellen Verbindungszustandes
    ⇨ Der Kontext einer jeden Datenübertragung wird somit berücksichtigt.
- Verbindungslose Dienste
  - Jeder Datenaustausch wird gesondert betrachtet, ohne Betrachtung vorhergegangener Kommunikationsvorgänge (gedächtnislos)
    ⇨ Der Kontext einer Datenübertragung wird somit nicht berücksichtigt.

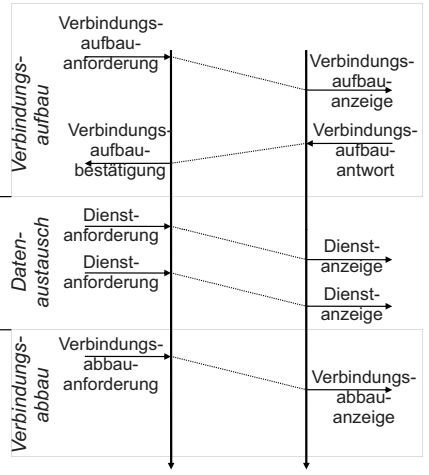
Verbindungsorientierte Dienste

3-Phasen-Prinzip

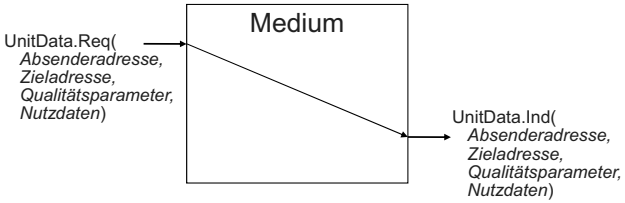
1. Verbindungsaufbau
   Kontexterzeugung
   - Endsysteme
   - Netz

2. Datenaustausch
   (hier: simplex)
   weniger laufende Kontext-informationen erforderlich

3. Verbindungsabbau
   Kontextfreigabe
   Ressourcenfreigabe



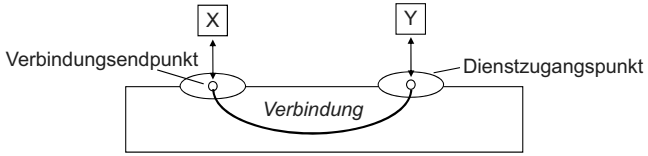
Datagramm-Dienste



- Vom Datagramm-Dienst wird *kein Zusammenhang* zwischen verschiedenen Übertragungsleistungen unterstützt.
- Der Datagramm-Dienst unterstützt *keine Auslieferungsdisziplin*, z.B. keine Garantie für Reihenfolgetreue.
- Der Datagramm-Dienst realisiert eine *unbestätigte Dienstleistung* (keine Aushandlung zwischen Kommunikationspartnern).

Dienstnehmer-Adressierung

- Datagramm
  - Anforderung: Mit Adresse des Beantworters
  - Anzeige: Ggf. mit Adresse des Initiators
- Verbindungen
  - Kontext, etabliert durch Verbindungsaufbau, beinhaltet Adressierungsinformation
  - Bei mehreren Verbindungen vom selben Dienstzugangspunkt: Verbindungsidentifikation



2.6. Das ISO/OSI-Basisreferenzmodell

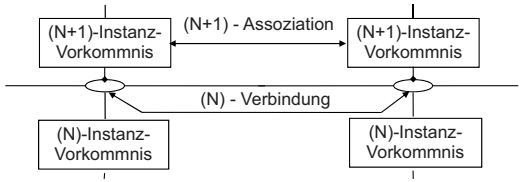
- Ziel:
  - Internationale Standardisierung (ISO = International Organization for Standardization) von Diensten und Protokollen zur Realisierung sogenannter “Offener Systeme” (OSI = Open System Interconnection)
  - Grundlage zur Kommunikation von Systemen unterschiedlicher Hersteller
  - Wichtig: Das Basisreferenzmodell dient als Denkmodell, anhand dessen sich Kommunikationssysteme erklären und klassifizieren lassen.
  - Implementierung des Modells vor allem in öffentlichen Netzen in Europa (weitgehende Verdrängung durch Internet-Protokolle)
- Standard:
  - ISO/IEC IS 7498: Information Processing Systems - Open Systems Interconnection - Basic Reference Model, Internationaler Standard, 15. Oktober 1994.
  - Übernommen von der CCITT bzw. ITU-T in der Norm X.200

Prinzipien des ISO/OSI-Basisreferenzmodells

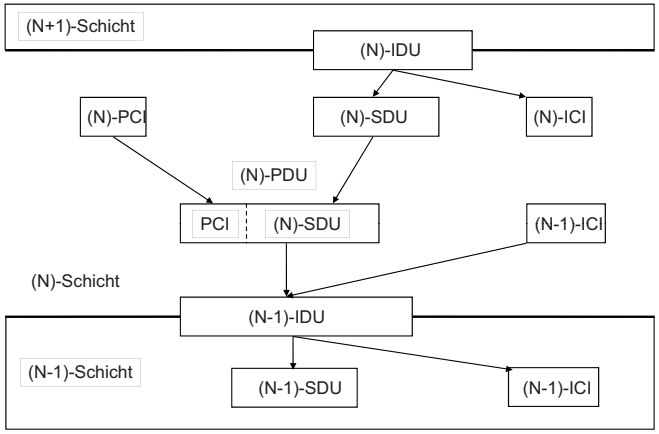
- OSI - Endsystem
  - Rechnersysteme, die sich bei der Kommunikation an OSI-Standards halten
- (N) - Schicht (Layer)
  - Sämtliche Einheiten einer (N) - Hierarchiestufe in allen Endsystemen
- (N) - Instanz (Entity)
  - Implementierung eines (N) - Dienstes in einem Endsystem.
  - Es kann verschiedene Typen von (N) - Instanzen geben ((N) - Instanz - Typen), z.B. IP im Router/Endsystem, oder die z.B. verschiedene Protokolle für eine Schicht implementieren. Eine Kopie einer (N) - Instanz wird Vorkommnis der (N) - Instanz genannt.
- Partnerinstanzen (Peer-Entities)
  - Instanzen einer Schicht.
  - Partnerinstanzen erfüllen Funktionen eines Dienstes durch Datenaustausch.

Verbindung und Assoziation

- (N) - Assoziation (Association)
  - Kooperative Beziehungen zwischen zwei (N)-Instanz-Vorkommnissen. Dazu gehört Verwaltung von Zustandsinformationen.
  - (N)-Assoziation wird durch (N-1)-Verbindungen (oder (N-1)-verbindungslosen Dienst) unterstützt. Sie kann zeitlich nacheinander verschiedene (N-1)-Verbindungen verwenden.
- (N) - Verbindung (Connection)
  - Beziehung zwischen zwei (oder mehr) (N+1)-Instanz-Vorkommnissen auf Ebene der (N)-Schicht. Diese Beziehung wird mit Hilfe des (N)-Protokolls unterstützt.

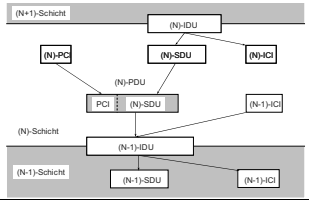


Generische OSI-Kommunikationseinheiten

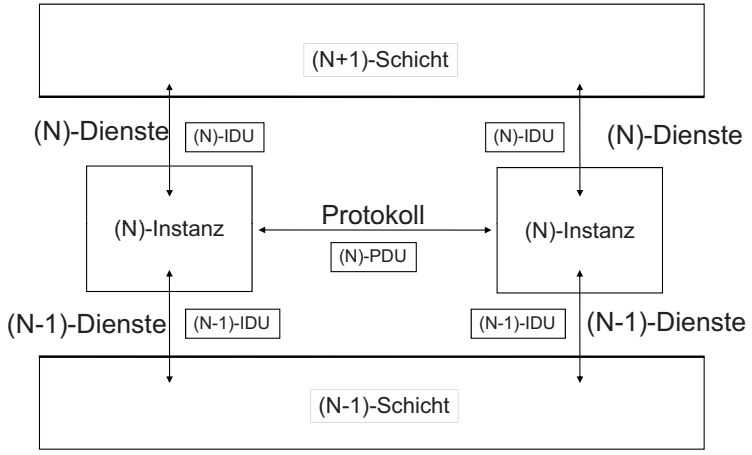


2.6.1. OSI-Kommunikationseinheiten, Beschreibung

- *(N)-Schnittstellendateneinheiten*
  - Interface Data Unit, IDU
  - Zwischen (N+1)- und (N)-Instanzen über einen (N)-SAP ausgetauschte Dateneinheit.
  - Setzt sich zusammen aus (N)-ICI und (N)-SDU.
- *(N)-Schnittstellenkontrollinformation*
  - Interface Control Information, ICI
  - Zwischen (N)-Schicht und (N+1)-Schicht ausgetauschte Parameter zur Steuerung von Dienstfunktionen (z.B. Adressen).
- *(N)-Dienstdateneinheiten*
  - Service Data Unit, SDU
  - Daten, die transparent zwischen (N)-SAPs übertragen werden.
- *(N)-Protokollkontrolldaten*
  - Protocol Control Information, PCI
  - Daten, die zwischen (N)-Instanzen ausgetauscht werden, um die Ausführung von Operationen zu steuern (z.B. Folgenummern o.ä.).
- *(N)-Protokolldateneinheit*
  - Protocol Data Unit, PDU
  - Dateneinheit, die zwischen (N)-Instanzen unter Benutzung eines Dienstes der (N-1)-Schicht ausgetauscht wird.
  - Zusammengesetzt aus (N)-PCI und (N)-SDU.
  - Entspricht somit der (N-1)-SDU.



Kommunikationsmodell - OSI-Systeme



Die OSI-Schichten im Überblick

Anwendungsschicht	Schicht 7 (A - Schicht)	**Application Layer**
Darstellungsschicht	Schicht 6 (P - Schicht)	**Presentation Layer**
Kommunikations-steuerungsschicht	Schicht 5 (S - Schicht)	**Session Layer**
Transportschicht	Schicht 4 (T - Schicht)	**Transport Layer**
Vermittlungsschicht	Schicht 3 (N - Schicht)	**Network Layer**
Sicherungsschicht	Schicht 2 (DL - Schicht)	**Data Link Layer**
Bitübertragungsschicht	Schicht 1 (Ph - Schicht)	**Physical Layer**

2.6.2. Bezeichnungskonventionen

- (N)-Schicht
  - A -Schicht: Anwendungsschicht (Application Layer)
  - P -Schicht: Darstellungsschicht (Presentation Layer)
  - S -Schicht: Kommunikationssteuerungsschicht (Session Layer)
  - T -Schicht: Transportschicht (Transport Layer)
  - N -Schicht: Vermittlungsschicht (Network Layer)
  - DL -Schicht: Sicherungsschicht (Data Link Layer)
  - Ph -Schicht: Bitübertragungsschicht (Physical Layer)
- (N)-Dienstprimitive

(N)-Dienst.typ

Ph	- Connect	.Req (Request, Anforderung)
DL	- Data	.Ind (Indication, Anzeige)
N	- Disconnect	.Rsp (Response, Antwort)
T	...	.Cnf (Confirmation, Bestätigung)

  - Dienstprimitive in der A-Schicht werden gemäß ihres Application Service Element (ASE) benannt.

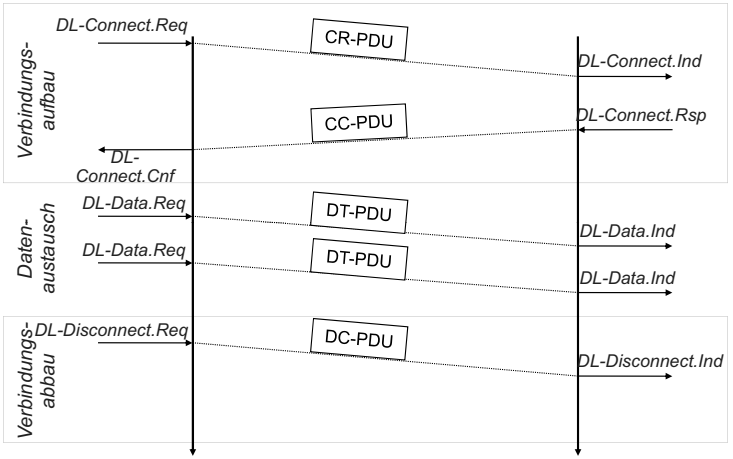
Dienstprimitive

- Die Benennung eines Dienstprimitivs besteht aus folgenden Komponenten:

Name der Schicht/Anwendung	Dienstleistung	Ereignistyp	Parameter
Physical (Ph) Data Link (DL) Network (N) Transport (T) HTTP FTP ...	Connect (Con) Data (Dat) Release (Rel) Abort (Abo) Provider Abort (PAbo) Disconnect (Dis) ...	Request (Req) Indication (Ind) Response (Rsp) Confirmation (Cnf)	*(beliebig)*

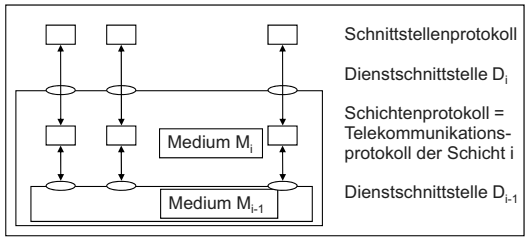
- Beispiel:
  - T-Con.Req(Adressen) = Verbindungsaufbauanforderung an der Schnittstelle zum Transportdienst
  - HTTP-Get.[Req](URL) = Anforderung der HTML-Seite, die durch URL identifiziert wird

Bezeichnungskonventionen am Beispiel



Protokoll: Modelle

- Überbrückung funktionaler und qualitativer Unterschiede zwischen D_{i-1} und D_i
- Art und Weise der Erbringung der Dienste D_i durch Instanzen I_i auf Basis der Dienste D_{i-1}
- Nebenläufiger Algorithmus
- Verteilter Algorithmus, wobei Dienste D_{i-1} das Zusammenwirken der I_i-Instanzen ermöglichen
- Berücksichtigung der Auswirkungen von Störungen in D_{i-1}
- Beschreibung: i.allg. nur 2 Instanzen, Automatenmodell, Weg-Zeit-Diagramm



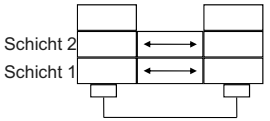
Protokollmechanismen

- Ein Protokollmechanismus ist ein Verfahren, welches abgeschlossene Teilfunktion innerhalb des Protokollablaufs beschreibt: generischer Charakter (ähnlich 'Systemfunktion').
- In verschiedenen Kommunikationsarchitekturen verwendet.
- Oft in mehreren Protokollen/Schichten einer Kommunikationsarchitektur anzutreffen.

- Multiplexen / Demultiplexen
- Teilung / Vereinigung
- Segmentieren / Reassemblieren
- Blocken / Entblocken
- Verkettung / Trennung
- (Mehrfach-)Kapselung
- Fehlerbehandlung
- Sicherung (ggf. fehlererkennend)
- Sequenzüberwachung
- Quittierung (Acknowledgement)
- Zeitüberwachung (Timeout)
- Wiederholen; Rücksetzen
- Flusskontrolle (Sliding window)
- Routing (Wegewahl, Weiterleiten)
- Medienzuteilung für geteilte Medien
- Synchronisation
- Adressierung
- Verbindungsverwaltung
- Datentransfer

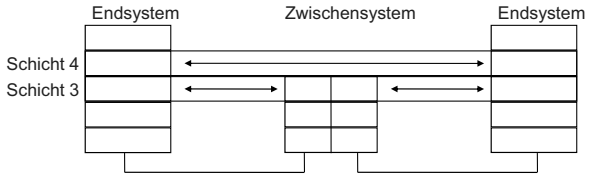
2.6.3. Charakterisierung der Schichten Bitübertragungsschicht und Sicherungsschicht

- Bitübertragungsschicht (Schicht 1)
  - ungesicherte Verbindung zwischen Systemen
  - Übertragung unstrukturierter Bitfolgen über physikalisches Medium
  - umfasst u.a. physikalischen Anschluss, Umsetzung Daten ↔ Signale
  - Normung vor allem der physikalischen Schnittstelle Rechner/Medien
- Sicherungsschicht (Schicht 2)
  - gesicherter Datentransfer
  - Zerlegung des Bitstroms (Schicht 1) in Rahmen (Frames)
  - Fehlererkennung und -behandlung
  - Protokollmechanismen: Quittierung, Zeit-/Sequenzüberwachung, Wiederholen/Rücksetzen



Vermittlungsschicht und Transportschicht

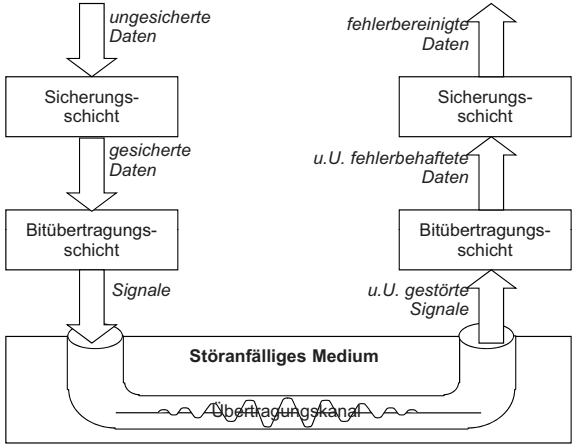
- Vermittlungsschicht (Schicht 3, auch 'Netzwerkschicht')
  - verknüpft Teilstreckenverbindung zu Endsystemverbindungen
  - Wegewahl (Routing) bei Vermittlung, Staukontrolle
  - evtl. aufgeteilt in 'Internetzwerk-/Subnetz-/Routing-'Subschichten
  - verbindungslos oder -orientiert
- Transportschicht (Schicht 4)
  - Adressierung von Transportdienstbenutzern
  - Datentransfer zwischen Benutzern in Endsystemen
  - bietet Transparenz bzgl. Übertragungs- und Vermittlungstechnik, Subnetzen
  - verbindungsorientiert, ggf. –los



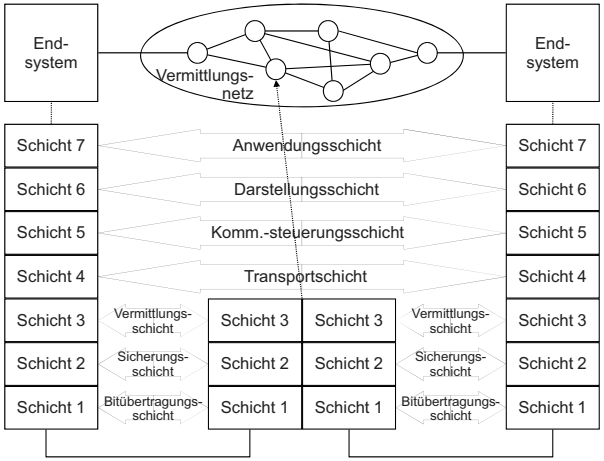
Anwendungsorientierte Schichten

- Kommunikationssteuerungsschicht (Schicht 5)
  - Ablaufsteuerung und -koordinierung (Synchronisation im weitesten Sinne)
  - Sitzung (Session)
  - ergibt erst Sinn bei Verwendung durch den Benutzer
- Darstellungsschicht (Schicht 6)
  - behandelt die Darstellung von Informationen (Syntax) für den Datentransfer
  - Marshalling
    - Prozess des Packens von Daten in einen Puffer, bevor dieser über die Leitung übertragen wird. Dabei werden nicht nur Daten verschiedenen Typs gesammelt, sondern diese werden auch in eine Standard-Repräsentation umgewandelt, die auch der Empfänger versteht.
- Anwendungsschicht (Schicht 7)
  - macht dem OSI-Benutzer Dienste verfügbar
  - stellt verschiedene Dienste zur Verfügung, je nach Anwendung, z.B.
    - Dateitransfer
    - zuverlässiger Nachrichtenaustausch
    - entfernter Prozeduraufruf

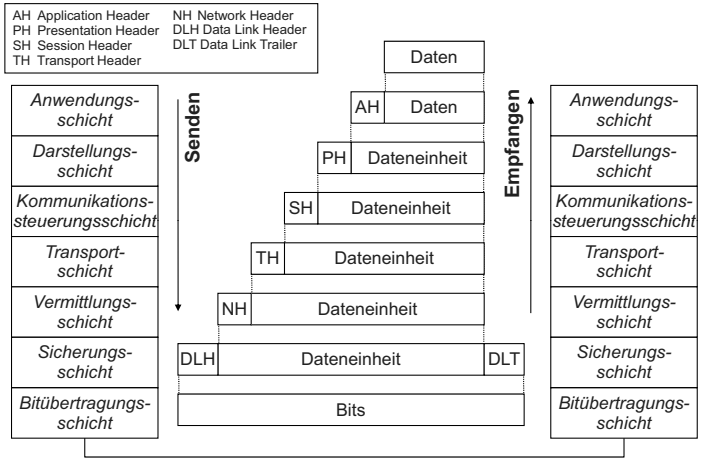
Daten und Signale



OSI: Die 7 Schichten



Einkapselung von Daten



Internet-Referenzmodell

Application Layer	Anwendungsspezifische Funktionen zusammengefasst in Anwendungsprotokollen
Transport Layer	Ende-zu-Ende-Datenübertragung zwischen zwei Rechnern
Network Layer	Wegewahl im Netz auch “Internet Layer” genannt
Net-to-Host	Schnittstelle zum physikalischen Medium “Netzwerkkartentreiber”

Gegenüber ISO/OSI sind die drei anwendungsorientierten Schichten zu einer einzigen Schicht zusammengefasst.

OSI und Internet

	OSI-Referenzmodell	*Internet-Referenzmodell*
7	Anwendung	Anwendung
6	Darstellung	
5	Komm.-steuerung	
4	Transport	Transport
3	Vermittlung	Internet
2	Sicherung	Rechner-Netzanschluss
1	Bitübertragung	

- *Unterschiede:*
  - Aufgaben der OSI-Schichten 5 und 6 werden beim Internet-Referenzmodell als Teil der Anwendung betrachtet.
  - Die OSI-Schichten 1 und 2 sind zu einer den Anschluss des Rechensystems an das Kommunikationsnetz beschreibenden Schicht zusammengefasst.

Lehrstuhl für Netzarchitekturen und Netzdienste
Institut für Informatik – Technische Universität München
Prof. Dr.-Ing. Georg Carle

2.7. Protokollspezifikation mit SDL

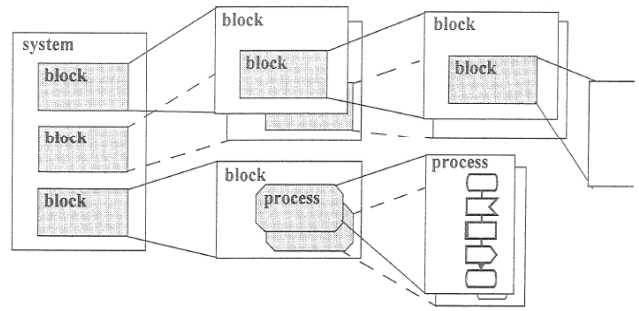
Specification and Description Language (SDL)

- Formale Sprache zur Beschreibung und Spezifizierung von Kommunikationssystemen
- Standard der ITU (früher: CCITT) (1984, 1988, 1992)
  - ITU = International Telecommunications Union
  - CCITT = Comité Consultatif International Téléphonique et Télégraphique
- Ziele:
  - Beschreibung des Verhaltens bestehender Systeme
  - Spezifizierung des Verhaltens neuer Systemkonzepte
- Verwendung u.a. bei der Spezifikation digitaler, leitungsvermittelter Systeme:
  - ISDN (Integrated Services Digital Network)
  - SS7 (Signaling System No 7)

Eigenschaften von SDL

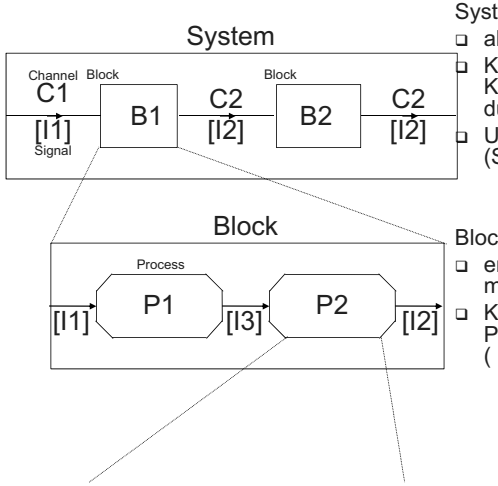
- Prozess als Grundelement
  - erweiterter endlicher Automat (Extended Finite State Machine - EFSM)
  - kommuniziert mit anderen Prozessen durch den Austausch von Nachrichten (Signalen) über Verbindungswege (Kanäle)
  - mehrere Prozesse arbeiten parallel und existieren gleichberechtigt nebeneinander
- Vordefinierte und benutzerdefinierte Datentypen
- Zwei äquivalente Darstellungsformen:
  - SDL/GR (Graphical Representation)
  - SDL/PR (Phrase Representation)
- Vorteile einer formalen Sprache
  - Exakte Spezifizierung
  - Möglichkeit von Werkzeugen - Editoren, Simulatoren, Prototyp-Generatoren, Testfall-Generatoren, Werkzeuge zur formalen Verfiikation
    - Generatoren (Compiler) zur direkten Übersetzung von SDL in ausführbare Programme oder Programmgerüste

Hierarchische Strukturierung in SDL



Aus König: SDL, Kap. 8

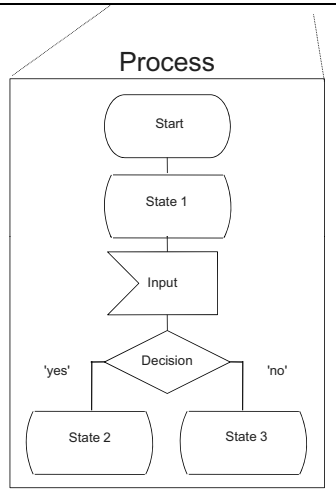
Hierarchische Strukturierung in SDL



System:
- abgegrenzt von Umwelt
- Kommunikation über Kanäle ( ) durch Signale
- Unterteilung in Blöcke (Subsysteme)

Block:
- enthält einen oder mehrere Prozesse
- Kommunikation zwischen Prozessen über Signale ( )

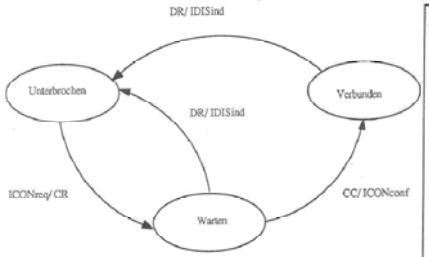
Hierarchische Strukturierung in SDL



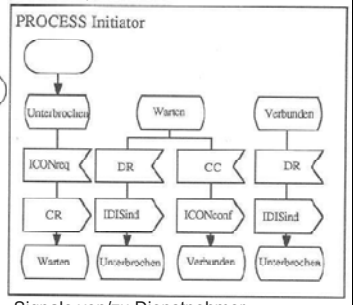
Prozess:
- kommunizierende *Extended Finite State Machine* (EFSM)
- Zustände, Übergänge, Aufgaben
- weitere Unterteilung in Prozeduren und Macros

Übersetzbarkeit von Automaten in SDL-Graphen

Beispiel ⇨ InRes-Protokoll (InRes= Initiator-Responder), c.f. Folie 81



Zustandsgraph



Prozess in SDL/GR

Signale von/zu Dienstnehmer
- ICONreq: InRes-Connection-Request
- ICONconf: InRes-Connection-Confirm
- IDISreq: InRes-Disconnection-Request
- IDISind: InRes-Disconnection-Indication

Signale von/zu entfernter Instanz
- CC, DR, ...

aus: Hogrefe, „ESTELLE, LOTOS und SDL“, Springer Compass, 1989, S.121ff

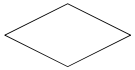
Prozesse in SDL

- Prozesse auf Basis erweiterter endlicher Automaten (EFSM):
  - endliche Zustandsanzahl und vorgegebene Zustandsübergänge
  - Eingangssignale lösen Zustandsübergänge aus
  - Aufgaben werden während eines Zustandsübergangs ausgeführt, z.B. auch Aussendung von Ausgangssignalen an andere Prozesse
  - eine Eingabewarteschlange puffert eingehende Nachrichten zwischen, falls Prozess sich gerade in einem Zustandsübergang befindet
  - es kann mehrere Instanzen eines Prozesses geben
- Erzeugung von Prozessen
  - bei Systemstart
  - zur Laufzeit durch andere Prozesse (CREATE)
- Beendigung von Prozessen
  - bei Erreichen eines STOP-Knotens

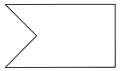
Symbole im Prozessgraphen

Symbol	Bedeutung des Knotens
	Start Node (Startknoten): • kennzeichnet Beginn eines Prozesses • enthält Name des Prozesses
	State Node (Zustandsknoten): • für einen oder mehrere Zustände • enthält den/die Zustandsnamen
	Task Node (Aufgabenknoten): • zwischen zwei Zuständen • führt Befehle aus • enthält Namen und optional die Befehls-abfolge oder informellen Text

Symbole im Prozessgraphen

Symbol	Bedeutung des Knotens
	Create Request Node: • erstellt und startet neue Prozessinstanz innerhalb eines Übergangs • wohldefiniert, enthält Name des Prozesses und seine Parameter
	Stop Node: • beendet die Prozessinstanz
	Decision Node: • ermöglicht Auswahl zwischen alternativen Pfaden innerhalb eines Übergangs • enthält eine Bedingung oder Abfrage • Antworten kennzeichnen Pfade/Alternativen

Symbole im Prozessgraphen

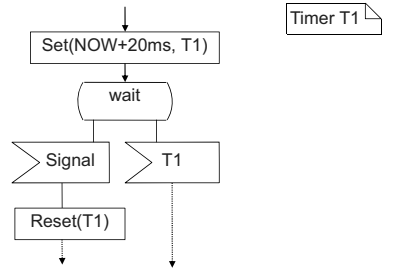
Symbol	Bedeutung des Knotens
	Save Node (SYNCHRONISATION): • verzögert ein Signal innerhalb eines Übergangs (ohne dass dazu ein Zustand existieren muss) • enthält gespeicherte Signale
	Input Node: • wartet auf den Erhalt eines oder mehrerer Signale innerhalb eines Übergangs • enthält den/die Signalnamen
	Output Node: • sendet ein oder mehrere Signale innerhalb eines Übergangs • enthält den/die Signalnamen und optional Zielprozessname/Kommunikationspfad

Symbole im Prozessgraphen

Symbol	Bedeutung des Knotens
	Flow Line: • Pfad (Kante), um zwei Symbole (Knoten) miteinander zu verbinden
	Input Node (In-Connector): • markiert die Stelle, an der der Pfad von gleichnamigem Out-Connector weitergeht
	Output Node (Out-Connector): • markiert die Stelle, an der der Pfad unterbrochen wird, um an In-Connector weiterzulaufen
	Comment: • zusätzlicher informeller Text

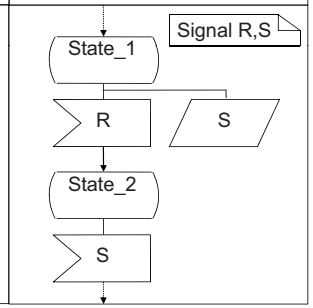
Zeitverhalten

- Zeitverhalten spielt eine große Rolle in der Telekommunikation
- Einführen von Timer-Prozessen:
  - gibt vor, wie lange ein Zustand maximal gehalten wird, bis eines der erwarteten Eingangssignale eintrifft
- Beispiel zur Verwendung eines Timers:

Erklärung	Timer in SDL
Ablauf des Timers T1 wird auf 20ms nach dem Zeitpunkt gesetzt. **wait** wird verlassen, falls a) Signal eintrifft, oder b) Timer abläuft (Timeout)	

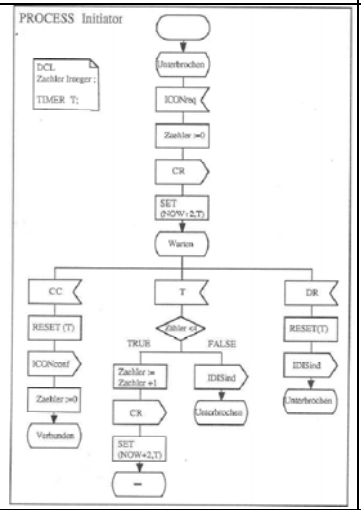
Signalverzögerung (Implicit Delays)

- Normalerweise Abarbeitung der Eingangssignale nach dem FIFO-Prinzip
- Bei gleichzeitigem Eintreffen zweier Eingangssignale zufällige Auswahl
- Reihenfolge für die Verarbeitung von Eingangssignalen kann durch SAVE-Knoten geändert werden
- Beispiel zur Verarbeitung zweier Signale R und S mit der Reihenfolge R,S:

Erklärung	Synchronisation in SDL
State_1: • Wenn Signal S vor R ankommt, wird es zwischengespeichert. State_2: • Wenn Signal S zwischengespeichert wurde, wird es sofort verarbeitet	

Beispiel

- InRes-Protokoll:
  Einfaches Protokoll zum Verbindungsaufbau zwischen zwei Protokollinstanzen
- Signale:
  - ICONreq: Verbindungsanforderung durch Benutzer
  - ICONconf: Verbindungsbestätigung an Benutzer
  - IDISind: Meldung eines Verbindungsabbruchs an den Benutzer
  - CR: Connection-Request-Nachricht an Gegenstelle
  - CC: Connection-Confirm-Nachricht von Gegenstelle
  - DR: Disconnect-Request-Nachricht von Gegenstelle



Standardisierung: Überblick

- Die Erfordernisse einer internationalen Telekommunikation erzwingen die Festlegung international gültiger Standards.
  - Standardisierung des Fernmeldewesens
    - Gremienarbeit mit gut strukturierten Lösungen, aber lange „Time To Market“
    - Weltweit einheitlich über Fernmelde-Betriebsgesellschaften (Telekommunikations-Dienstanbieter)
    - Beispiele: ITU-T, ETSI (European Telecommunication Standards Institute)
  - Internet
    - Diskussionen direkt Betroffener und IETF (Internet Engineering Task Force) führen zu Standards
    - Beispielimplementierungen stehen im Vordergrund, daher sehr schnelle „Time To Market“
  - Herstellervereinigungen
    - Ebenfalls realisierungsorientiert mit relativ schneller „Time To Market“
    - Beispiele: The Open Group (ehemals OSF und X/Open), ECMA (European Computer Manufacturers Association), ATM-Forum

Standardisierung: Traditionelle Organisationen

ITU International Telecommunication Union (ehemals CCITT und CCIR)
Internationaler beratender Ausschuss für Telekommunikation

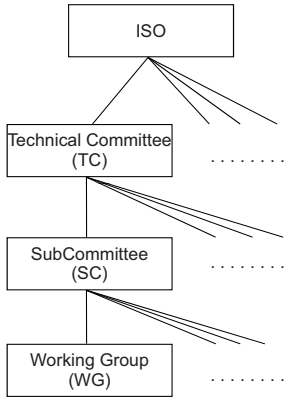
CCITT Consultative Committee on International Telegraphy and Telephony
Comité Consultatif International Télégraphique et Téléphonique
ehem. Internationaler beratender Ausschuss für Telefon und Telegrafie
(neue Bezeichnung: ITU-T)

CCIR Consultative Committee on International Radio
ehem. Internationaler beratender Ausschuss für den Funkdienst
(neue Bezeichnung: ITU-R)

ISO International Organization for Standardization
(ISO griech. „gleich“)
Internationale Organisation für Standardisierung
ISO koordiniert die internationale Normungsarbeit außerhalb des Telekommunikations-Bereichs.

DIN (Deutsches Institut für Normung) ist deutscher Partner der ISO.

Standardisierung: Beispiel ISO



WG-Meetings:
Alle 6-9 Monate, damit die nationalen Organisationen Einverständnis mit den Konzepten erreichen. Dann startet der **Standardisierungsprozess:**

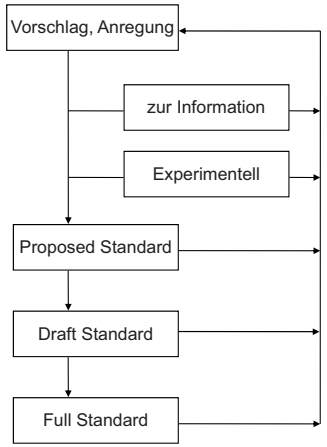
DP: Draft Proposal
DIS: Draft International Standard
IS: International Standard

Das Fortschreiten auf eine höhere Stufe erfolgt durch eine internationale Abstimmung und die Einarbeitung der Kritik der „Nein“-Stimmen.

⇨ sehr langer Prozess!

Standardisierung: Beispiel Internet

- Der Standardisierungsweg geht über die Internet Engineering Task Force (IETF).
- Die Internet Engineering Steering Group (IESG) steuert die Diskussionen.
- Allgemein akzeptierte Arbeitsdokumente (Internet Drafts) erhalten permanenten Status (Request for Comments, RFC)
- Mögliche Ergebnisse:
  - Standard Track RFC (Proposed/Draft/Full Standard)
  - Experimenteller RFC
  - RFC zur Information
- Bereits ab dem Status Draft Standard müssen mindestens zwei interoperable, unabhängig voneinander entwickelte Implementierungen vorhanden sein.



Standardisierung: RFC - Beispiele

- RFC 768 User Datagram Protocol (UDP), August 1980
- RFC 791 Internet Protocol (IP), Sept. 1981
- RFC 792 Internet Control Message Protocol (ICMP) Sept. 1981
- RFC 793 Transmission Control Protocol (TCP), Sept. 1981
- RFC 959 File Transfer Protocol (FTP), Oktober 1985
- RFC 997 Internet Numbers, März 1987
- RFC 3261 SIP: Session Initiation Protocol, Juni 2002
- RFC 4509 Use of SHA-256 in DNSSEC Delegation Signer (DS) Resource Records (RRs), Mai 2006

- Weiter Informationen unter www.ietf.org!

Standardisierung: RFC – Beispiele (2)

- RFC 1149—Standard for the transmission of IP datagrams on Avian Carriers. D. Waitzman. 1 April 1990. Updated by RFC 2549; see below. A deadpan skewering of standards-document legalese, describing protocols for transmitting Internet data packets by homing pigeon.
- RFC 2322—Management of IP numbers by peg-dhcp. K. van den Hout et al. 1 April 1998.
- RFC 2324—Hyper Text Coffee Pot Control Protocol (HTCPCP/1.0). L. Masinter. 1 April 1998.
- RFC 2549—IP over Avian Carriers with Quality of Service. D. Waitzman. 1 April 1999. Updates RFC 1149, listed above.
- RFC 3251—Electricity over IP. B. Rajagopalan. 1 April 2002.
- RFC 3514—The Security Flag in the IPv4 Header (Evil Bit). S. Bellovin. 1 April 2003.
- RFC 4824—The Transmission of IP Datagrams over the Semaphore Flag Signaling System (SFSS). Jogi Hofmueller, Aaron Bachmann, IOhannes zmoelnig. 1 April 2007.

Lehrstuhl für Netzarchitekturen und Netzdienste
Institut für Informatik – Technische Universität München
Prof. Dr.-Ing. Georg Carle

Grundlagen: Rechnernetze und Verteilte Systeme

Kapitel 3: Direktverbindungsnetze

HDLC, PPP, Ethernet

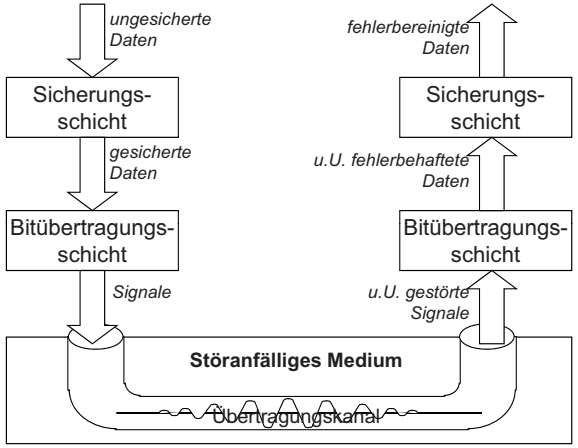
Prof. Dr.-Ing. Georg Carle
Lehrstuhl für Netzarchitekturen und Netzdienste
Technische Universität München
carle@net.in.tum.de
http://www.net.in.tum.de

Ziele

- In diesem Kapitel wollen wir vermitteln
  - Grundverständnis von Daten- und Signalübermittlung
  - Fehlerursachen und Fehlertypen
  - Fehlerbehandlungen
  - Vorgänge in der Sicherungsschicht
  - Zugriffsverfahren

Kapitelgliederung

3.1. Daten und Signale

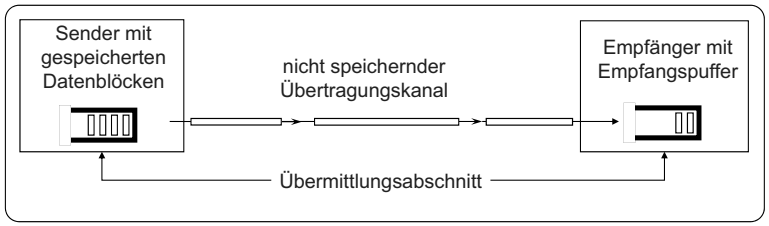


3.1.1. Data Link Control-Protokolle (DLC)

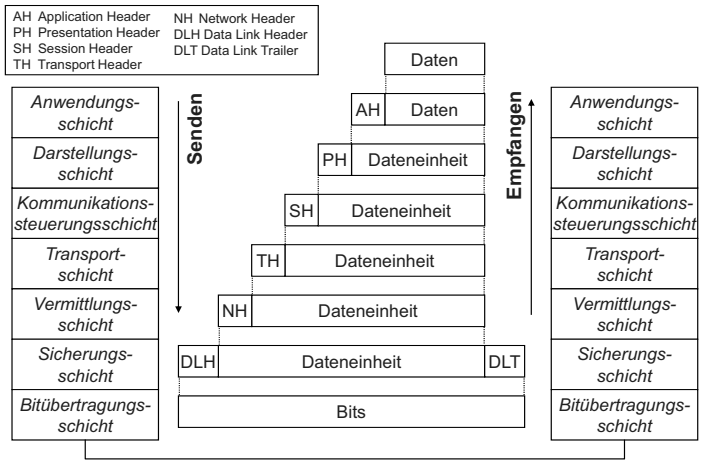
- Ziel von Datenübermittlungsprotokollen
  - *Sicherstellung einer fehlerfreien Übertragung* von Dateneinheiten über einen nicht-speichernden, „durchgehenden“ Übertragungskanal.
- Aufgaben
  - Aufbau und Unterhaltung einer „logischen“ Verbindung zwischen zwei „benachbarten“, d.h. über eine physikalische (unter Vernachlässigung der endlichen Übertragungsgeschwindigkeit nicht-speichernde) Verbindung direkt kommunizierender Systeme.
  - Gesicherte Übermittlung von Daten.
  - Synchronisation der Datenübertragung.
- Datenübertragungseinheiten
  - einzelne Zeichen
  - Datenblöcke (Übertragungsblöcke, Rahmen, Pakete, Zellen) (englisch: *transmission block*, *frame*, *packet*, *cell*)
- Realisierung: Datenübermittlungsprotokolle - Data Link Control (DLC)-Protokolle

3.1.2. Konzept des Übermittlungsabschnitts

- Ausgangspunkt: Puffer-zu-Puffer-Betriebsmodell
- Ein **Übermittlungsabschnitt (data link)** umfasst konzeptionell beide Pufferspeicher für den zu übermittelnden Datenblock und den nicht speichernden Übertragungsweg.
- Eine Übermittlung ist erst abgeschlossen, wenn der zu übermittelnde Datenblock vollständig und fehlergeprüft im Empfangsspeicher des betreffenden Übermittlungsabschnittes abgelegt ist.



3.1.3. Einkapselung von Daten: Sicherungsschicht



3.1.4. DLC: Generelles Aufgabenspektrum

- **Zeichen-/Blocksynchronisation**
  - Korrekte positionsrichtige Erkennung von Zeichen bzw. allgemein Bitfolgen für die Interpretation
  - Erkennung von Blockbegrenzungen
- **Medium-/Übermittlungsmanagement (Link Management)**
  - koordinierter Medienzugriff
  - Vergabe von Senderechten/ Übertragungsinitiativen (Arbitrierung)
- **Fehlererkennung und -behandlung**
  - Datenfehlererkennung und -behandlung mittels
    - Zeichenparität
    - Blockparität
    - Kreuzsicherung
    - Zyklische Blocksicherung
  - Protokollfehler, z.B. Verfälschung von Protokollkontrolldaten, Steuerzeichen, Adressen durch Störungen

DLC: Konkrete Aufgabenstellung

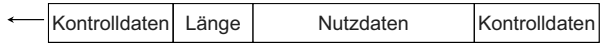
- **Datenblockformate:**
  - Festlegung und Erkennung
- **Übermittlungsprotokolle:**
  - Übermittlungssteuerungsverfahren (z.B. Initialisierung, Terminierung, Identifikation, Halbduplex-/Vollduplexbetrieb)
- **Codetransparenz:**
  - Übertragung jeglicher Kombination von Daten der darüber liegenden Schicht
- **Fehlerbehebung:**
  - Erkennung und Behandlung von Fehlern im Daten- und im Protokollbereich
- **Zugriffsregelung:**
  - Vergabe von Senderechten, Vermeidung von Kollisionen
- **Datenflusskontrolle:**
  - Verhinderung von Überlastsituationen zwischen Sender und Empfänger des Übermittlungsabschnittes
- **Bei zeichenorientierten Protokollen:**
  - Vereinbarung eines standardisierten Übermittlungsalphabets („zeichencode-kompatibel“)

3.2. Synchrone Übertragung und Codetransparenz

- **Synchrone Übertragung:**
  - Empfänger muss Anfang und Ende eines Datenblocks erkennen können
- **Codetransparenz:**
  - Übertragung von Nutzdaten ermöglichen, die beliebiger Bit- bzw. Zeichenkombinationen enthalten
- **Lösungsansätze:**
  (1) Längenangabe der Nutzdaten
  (2) Steuerzeichen und Zeichenstopfen (Character Stuffing)
  (3) Begrenzungsfeld und Bitstopfen (Bit Stuffing)
  (4) Coderegelverletzungen

Synchrone Übertragung und Codetransparenz

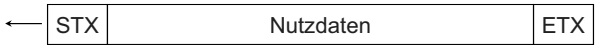
(1) Längenangabe der Nutzdaten:



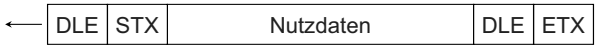
- Länge des Datenblocks (in Bytes/Zeichen) wird dem Empfänger im Rahmenkopf mitgeteilt
- **Problem:** Längenfeld kann durch Übertragung verfälscht werden
  - kein Erkennen der Rahmengrenze mehr möglich
  - Verlust der Synchronisation zwischen Sender und Empfänger

Synchrone Übertragung und Codetransparenz

(2) Steuerzeichen und Zeichenstopfen (Character Stuffing):



- reservierte Steuerzeichen markieren Anfang und Ende des Datenblocks (z.B. ASCII-Steuerzeichen)
- **Problem:**
  - Steuerzeichen dürfen nicht in den Nutzdaten auftauchen
- **Abhilfe:**
  - zeichengesteuerter Transparenzmodus durch zusätzliches Steuerzeichen DLE (Data Link Escape).



  - falls DLE in Nutzdaten auftaucht: Zeichenstopfen

Internationales 7-bit-Alphabet (IA5) – Deutsche Referenzversion

$b_4b_3b_2b_1$ \ $b_7b_6b_5$		000	001	010	011	100	101	110	111
		0	1	2	3	4	5	6	7
0000	0	NUL	TC_7(DLE)	SP	0	@	P	`	p
0001	1	TC_1(SOH)	DC_1	!	1	A	Q	a	q
0010	2	TC_2(STX)	DC_2	"	2	B	R	b	r
0011	3	TC_3(ETX)	DC_3	#	3	C	S	c	s
0100	4	TC_4(EOT)	DC_4	$	4	D	T	d	t
0101	5	TC_5(ENQ)	TC_8(NAK)	%	5	E	U	e	u
0110	6	TC_6(ACK)	TC_9(SYN)	&	6	F	V	f	v
0111	7	BEL	TC_{10}(ETB)	'	7	G	W	g	w
1000	8	FE_1(BS)	CAN	(	8	H	X	h	x
1001	9	FE_2(HT)	EM	)	9	I	Y	i	y
1010	A	FE_3(LF)	SUB	*	:	J	Z	j	z
1011	B	FE_4(VT)	ESC	+	;	K	Ä	k	ä
1100	C	FE_5(FF)	IS_4(FS)	,	<	L	Ö	l	ö
1101	D	FE_6(CR)	IS_3(GS)	-	=	M	Ü	m	ü
1110	E	SO	IS_2(RS)	.	>	N	^	n	ß
1111	F	SI	IS_1(US)	/	?	O	_	o	DEL

Ursprung: American Standard Code of Information Interchange ASCII

Hexadezimaldarstellung

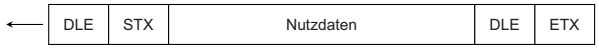
IA5: Steuerzeichen

- *10 Übertragungszeichen*
  - SOH Start of Header
  - **STX Start of Text**
  - **ETX End of Text**
  - EOT End of Transmission
  - ENQ Enquiry
  - ACK Acknowledgement
  - **DLE Data Link Espace**
  - NAK Negative ACK
  - SYN Synchronous Idle
  - **ETB End of Transmission Block**
- *4 Gerätesteuerzeichen* (Device Control DC) nicht genormt, sondern frei belegbar.
- *3 Steuerzeichen zur Codeerweiterung*
  - SO Shift Out
  - SI Shift In
  - ESC Escape
- *4 Informationstrennzeichen* (Information Separator IS)
- *6 Formatzeichen*
  - FE_1 Backspace
  - FE_2 Horizontal Tabulation
  - FE_3 Line Feed
  - FE_4 Vertical Tabulation
  - FE_5 Form Feed
  - FE_6 Carriage Return
- *7 Sonstige Steuerzeichen*
  - NUL Füllzeichen ohne Bedeutung
  - BEL Klingelzeichen
  - DEL Löschen von Zeichen
  - CAN Cancel
  - EM End Of Medium
  - SUB Substitute
  - SP Space — Leerzeichen

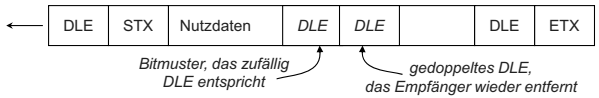
Codetransparenz durch Zeichenstopfen (Character Stuffing)

Anfang und Ende eines Rahmens werden durch STX bzw. ETX symbolisiert:

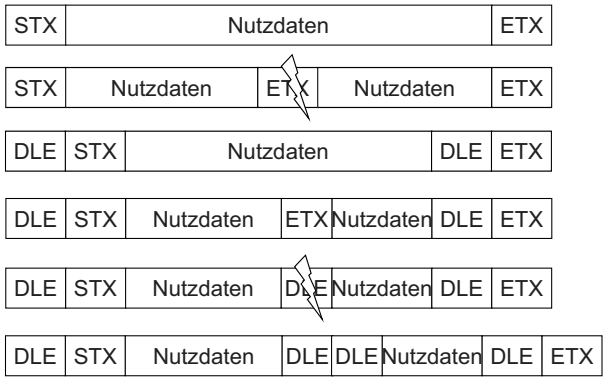
Problem: Ein ETX in den Nutzdaten würde ein vorzeitiges Ende des Rahmens signalisieren.
Lösung: Mit Hilfe eines speziellen Zeichens (DLE = Data Link Escape) werden die Nutzdaten transparent gemacht. Ein ETX wird daher nur dann als solches behandelt, wenn ein DLE davor steht.



Problem: Ein DLE in den Nutzdaten könnte jetzt zu einer Fehlinterpretation führen.
Lösung: Erkennt der Sender, dass in den Nutzdaten ein Bitmuster vorkommt, das einem DLE entspricht, so doppelt er dieses Zeichen. Der Empfänger analysiert, ob nach einem von ihm erkannten DLE ein weiteres folgt. Wenn ja, löscht er zweite DLE einfach, wenn nein, muss das folgende Zeichen als Steuerzeichen interpretiert werden.

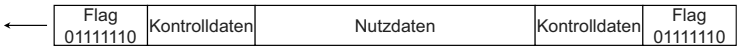


Character Stuffing – Arbeitsfolie

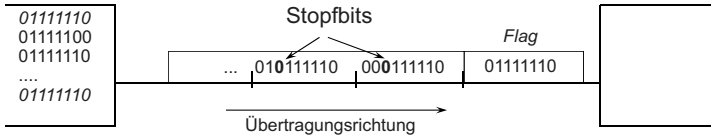


Synchrone Übertragung und Codetransparenz

(3) Begrenzungsfeld und Bitstopfen (Bit Stuffing):



- Blockbegrenzung (Flag) ist eine ausgezeichnete Bitfolge (01111110)
- **Problem:** Zufälliges Auftreten von 01111110 im DÜ-Block
- **Lösung:** Einfügen von Stopfbits in die Nutzdaten
  - Sender fügt nach 5 aufeinander folgenden Binärzeichen „1“ ein Binärzeichen „0“ ein.
  - Empfänger entfernt nach 5 aufeinander folgenden Binärzeichen „1“ ein folgendes Binärzeichen „0“.

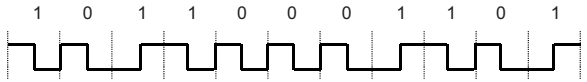


- Bemerkung: Blockprüfsumme (siehe später) zur Fehlererkennung wird vor dem Bitstopfen erstellt.

Synchrone Übertragung und Codetransparenz

(4) Coderegelverletzungen:

- Blockbegrenzung durch Verwendung ungültiger Codes
- Voraussetzung: Codierung auf Schicht 1 mit Redundanz
- Beispiel: Manchester-Code (siehe Kapitel 3)



- Ungültige Codes/Pegelwerte: (0,0) und (1,1)
- Einsatz bei IEEE 802.3 (Ethernet)

3.2.1. Fehlerursachen, Fehlertypen

- **Übertragungsfehler** sind hardwareinduzierte Fehler, die vorzugsweise auf dem Übertragungsmedium entstehen, aber auch in den Anschlusselektroniken der kommunizierenden Stationen.
- Art und Häufigkeit signaltechnischer Fehler sind stark vom Übertragungsmedium abhängig.
- In der Funktechnik existieren andere Fehlerursachen, Fehlerhäufigkeiten und Fehlerauswirkungen als in der leitungsgebundenen Übertragungstechnik.
- Bei Übertragung digitaler Daten führen Störeinflüsse (Fehlerquellen) zu falsch detektierten Bits.
- Typen:
  - **Einzel-Bit-Fehler**: Z. B. Rauschspitzen, die die Entscheiderschwelle bei digitaler Signalerfassung überschreiten.
  - **Bündelfehler**: Länger anhaltende Störung durch Überspannung, Starkstromschaltprozesse usw.
  - **Synchronisierfehler**: Alle Bits bzw. Zeichen werden falsch erkannt.
- Auswirkung einer Störung bestimmter Dauer ist abhängig von der Übertragungsgeschwindigkeit ⇨ Einzelbit oder Bündelstörung

Fehlerwirkungen: Rechenbeispiel

- Eine Störung von 20 ms führt ...
  - bei Telex (50 bit/s, Signaldauer: 20 ms) zu einem Fehler von 1 Bit ⇨ Einzelfehler
  - bei ISDN (64 Kbit/s, Signaldauer: 15,625 µs) zu einem Fehler von 1280 Bit ⇨ Bündelfehler

  bei B-ISDN / SDH
  - 155 Mbit/s, Signaldauer:6,45 ns: zu einem Fehler von ca. 3,1 Mbit = 387,5 Kbyte
  - 622 Mbit/s, Signaldauer: 1,61 ns: zu einem Fehler von ca. 12,4 Mbit = 1,5 Mbyte
  - 2,4 Gbit/s, Signaldauer: 0,4 ns: zu einem Fehler von ca. 48 Mbit = 6 Mbyte

  ⇨ Bündelfehler (großer Länge)

Fehlerhäufigkeiten

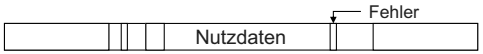
- Maß für die Fehlerhäufigkeit:

$$\textbf{Bitfehlerrate} = \frac{\text{Summe gestörte Bits}}{\text{Summe übertragene Bits}}$$

  - Stark vom Übertragungsmedium bzw. Netz abhängig
  - Bitfehlerraten zu übertragender digitaler Daten im analogen Netz sind sehr viel höher als in digitalen Übertragungssystemen. Moderne ISDN/PCM-Systeme haben eine bessere Übertragungsqualität als klassische digitale Netze.
  - Die Übertragungsfehlerhäufigkeit ist auch stark von der Gesamtlänge des Übertragungsweges abhängig
- Typische Wahrscheinlichkeiten für Bitfehler:
  - Analoges Fernsprechnetz $2*10^{-4}$
  - Funkstrecke 10^{-3} - 10^{-4}
  - Ethernet (10Base2) 10^{-9} - 10^{-10}
  - Glasfaser 10^{-10} - 10^{-12}

Fehlerwirkungen

- Fehlerwirkungen sind abhängig davon, welche Bits betroffen sind:
  - **(Nutz-)Datenfehler**: Bits innerhalb der Nutzdaten (gesehen z. B. aus Sicht der Sicherungsschicht) werden gestört.



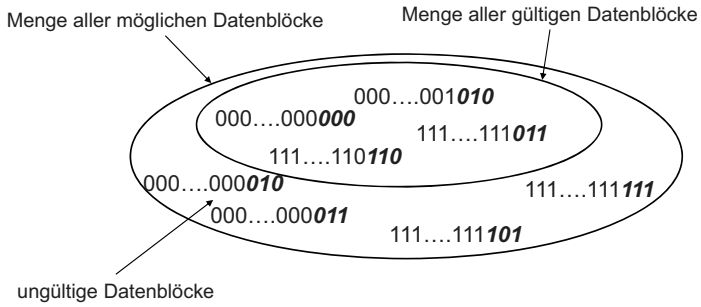
  - **Protokollfehler**: Störungen können Protokollkontrolldaten, Steuerzeichen, Adressen oder sonstige protokollrelevante Daten verfälschen oder vernichten.

⇨ **Fehlererkennungs- und Behandlungsmaßnahmen** (error detection and recovery) erforderlich.

⇨ Fehlererkennung durch (künstliches) Hinzufügen von Redundanz beim Sender
  - error detecting codes
  - (Spezialfall: error correcting codes)

Fehlererkennung: Grundprinzip

- Unterteilung aller möglichen Datenblöcke in gültige und ungültige:



- Wie?
  - durch gezieltes Hinzufügen von Redundanz beim Sender

3.2.2. Fehlerbehandlung

- Fehler ignorieren
  - Beispiel: Audio/Video-Stream ➔ kurzzeitiges Fehlsignal oder Lücke
  - nicht möglich bei Fehlern in den Steuerdaten!
- Wiederholung der Übertragung
  - Empfänger verwirft fehlerhaften Datenblock
  - implizite oder explizite Wiederholungsanforderung (siehe später)
- Fehlerkorrektur:
  - Codes, mit denen Bitfehler korrigiert werden können
  - erfordert größere Redundanz als reine Fehlererkennung

Fehlererkennung: Paritätssicherung — Überblick

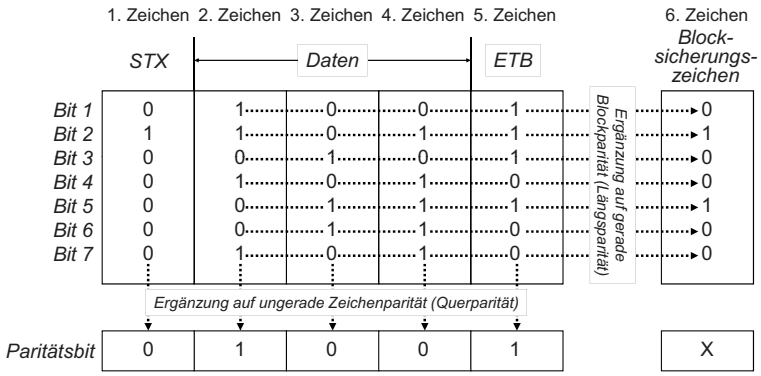
- *Gerade Parität*:
  - Gesamtzahl der „1" einschließlich des Paritätsbits ist gerade.
- *Ungerade Parität*:
  - Gesamtzahl der „1" einschließlich des Paritätsbits ist ungerade.
- *Zeichen- oder Querparität*: (VRC: Vertical Redundancy Check)
  - Sicherung von Einzelzeichen. Einziges Verfahren bei asynchroner Einzelzeichen-Übertragung.
- *Block- oder Längenparität*: (LRC: Longitudinal Redundancy Check)
  - Alle Bits gleicher Wertigkeit innerhalb eines aus Zeichen bestehenden Übertragungsblocks werden durch ein Paritätsbit (gerade oder ungerade) gesichert. Sie bilden ein Blockprüfzeichen (BCC: Block Check Character).
- *Kreuzsicherung*:
  - Gleichzeitige Anwendung von Längs- und Querparität. Abstimmung über die Bildung des Paritätsbits des Blockprüfzeichens erforderlich!

- **Hinweis:** STX wird nicht in Block-Paritätsprüfung einbezogen, da der Empfänger erst nach START OF TEXT weiß, dass ein prüfenswerter Übertragungsblock beginnt.

- **Problem:** Fehlererkennungswahrscheinlichkeit bei Paritätsprüfung nicht hoch. Mehrfachfehler (zwei oder eine gerade Zahl in gleicher Zeile oder Spalte liegende Fehler) werden nicht erkannt.

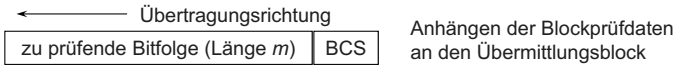
Fehlererkennung: Paritätsüberprüfung

- Gerade/ungerade Parität
- Querparität, Längsparität, Kreuzparität



Fehlererkennung: Cyclic Redundancy Check (CRC)

- Zu prüfender Block wird als unstrukturierte Bitfolge aufgefasst.
  - Anzahl der zu prüfenden Bits ist beliebig
- Prüfbitfolge [Block Check Sequence (BCS) bzw. Frame Check Sequence (FCS)] wird an den zu prüfenden Übermittlungsdatenblock angehängt.



Anhängen der Blockprüfdaten an den Übermittlungsblock

- Bildung der Prüfsequenz:
  - Die zu prüfende Bitfolge wird als Polynom $M(x)$ aufgefasst, d.h. jedes der m Bit als Koeffizient eines Polynoms vom Grad (m-1) interpretiert.
  - Zur Berechnung der BCS/FCS wird an die zu prüfenden Bitfolge eine Nullfolge der Länge r angehängt (entspricht Multiplikation mit x^r), wobei r der Grad des Prüfpolynoms (Generatorpolynoms) $G(x)$ ist.
  - Das erhaltene Polynom $M(x)\cdot x^r$ wird durch das Prüfpolynom $G(x)$ geteilt; der Rest $R(x)$ der Division ist die gesuchte BCS/FCS.
  - $M(x)\cdot x^r$ - $R(x)$ wird als Bitfolge an den Empfänger übertragen.
  - Beim Empfänger wird die empfangene Bitfolge durch $G(x)$ dividiert. Bei fehlerfreier Übertragung ist der Rest 0.

Fehlererkennung: CRC-Beispiel – Senden

- Zu sendende Bitfolge: 110011 ⇨ $M(x) = x^5 + x^4 + x + 1$
- Prüfpolynom: $G(x) = x^4 + x^3 + 1$ ⇨ Divisor in Modulo-2-Binärarithmetik: 11001
  - Addition/Subtraktion Modulo-2 entspricht einer bitweisen XOR-Verknüpfung
  - Dividend ist teilbar durch Divisor, falls der Dividend mindestens so viele Stellen besitzt wie der Divisor (führende Bits müssen beide 1 sein)
- Länge der Sicherungsfolge = Grad des Prüfpolynoms = 4

- Berechnung der Sicherungsfolge:

angehängte Nullen

```
11 0011 0000 ÷ 1 1001 = 10 0001
11 001
00 0001 0000
      1 1001
      0 1001 = Rest
```

- Zu übertragende Bitfolge: 11 0011 1001.

Fehlererkennung: CRC-Beispiel – Empfangen

- Empfangen einer korrekten Bitfolge:

```
11 0011 1001 ÷ 1 1001 = 10 0001
11 001
00 0001 1001
      1 1001
      0 0000 = Rest
```

  - Kein Rest, somit sollten Daten fehlerfrei sein.

- Empfangen einer verfälschten Bitfolge:

```
11 11 11 1000 ÷ 1 1001 = 10 1001
11 001
00 1101 1
   1100 1
   0001 0000
      1 1001
      0 1001 = Rest ≠ 0
```

  - Es bleibt Rest ungleich 0, somit war ein Fehler in der Übertragung.

Cyclic Redundancy Check: Leistungsfähigkeit

- Folgende Fehler werden durch CRC erkannt:
  - sämtliche Einzelbitfehler;
  - sämtliche Doppelfehler, wenn $(x^k + 1)$ nicht durch das Prüfpolynom teilbar ist, für alle $k \leq$ Rahmenlänge;
  - sämtliche Fehler ungerader Anzahl, wenn $(x+1)$ Faktor des Prüfpolynoms ist;
  - sämtliche Fehlerbursts der Länge ≤ Grad des Prüfpolynoms.
- International genormt sind folgende Prüfpolynome:
  - CRC-12 $= x^{12} + x^{11} + x^3 + x^2 + x + 1$
  - CRC-16 $= x^{16} + x^{15} + x^2 + 1$
  - CRC-CCITT $= x^{16} + x^{12} + x^5 + 1$
- CRC-16 und CRC-CCITT entdecken
  - alle Einzel- und Doppelfehler,
  - alle Fehler ungerader Anzahl,
  - alle Fehlerbursts mit der Länge ≤ 16
  - 99,997 % aller Fehlerbursts mit der Länge 17
  - 99,998 % aller Fehlerbursts mit der Länge 18 und mehr

3.2.3. Vorwärtsfehlerkorrektur für Datenpakete

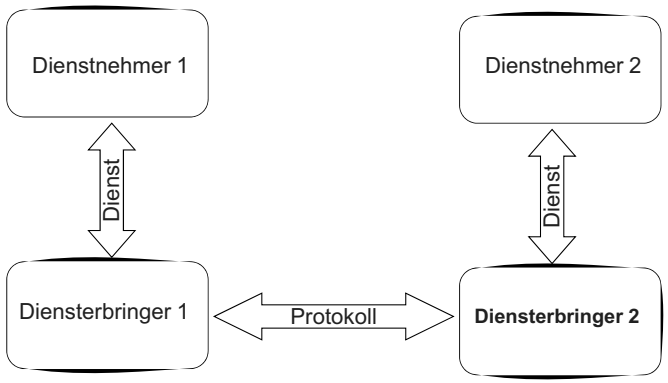
- Bislang war die Redundanz nur zur Überprüfung der Daten mitgeliefert worden, jetzt soll sie dazu dienen, verloren gegangene Pakete zu rekonstruieren.

- Beispiel:
  - Zu senden sind die Pakete 0101 - P1
1111 - P2
0000 - P3
  - Dazu wird über XOR ein weiteres Paket berechnet: 1010 - P4
  - Diese vier Pakete werden jetzt an den Empfänger geschickt.

Vorwärtsfehlerkorrektur — Ablauf

- Der Empfänger braucht nur drei der vier Pakete, um das fehlende zu rekonstruieren. Er verknüpft einfach die korrekten Pakete wieder XOR, und erhält so das fehlende:
  - P1 geht verloren:
1111 – P2
0000 – P3
1010 – P4
⇨ 0101 – P1
  - P2 geht verloren:
0101 – P1
0000 – P3
1010 – P4
⇨ 1111 – P2
  - P3 geht verloren:
0101 – P1
1111 – P2
1010 – P4
⇨ 0000 – P3
- Natürlich muss der Empfänger wissen, welches Paket verloren gegangen ist ...

Wiederholung: Dienst und Protokoll



Wiederholung: Bezeichnungskonventionen

- **(N)-Schicht**

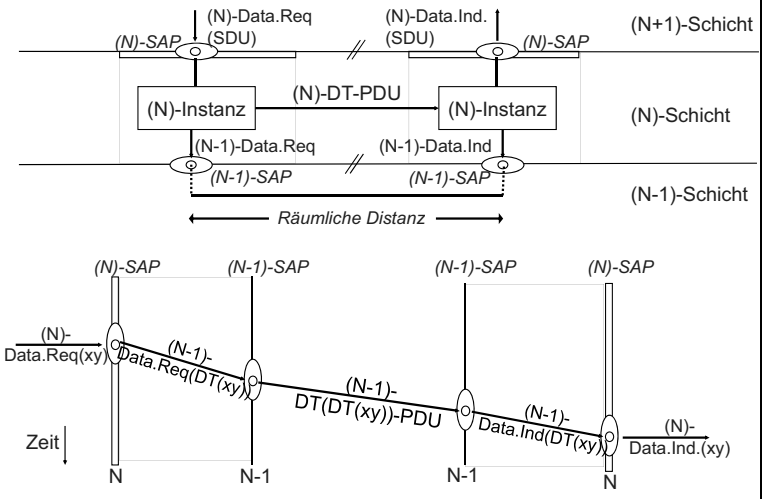
A	-Schicht: Anwendungsschicht	(**A**pplication Layer)
P	-Schicht: Darstellungsschicht	(**P**resentation Layer)
S	-Schicht: Kommunikationssteuerungsschicht	(**S**ession Layer)
T	-Schicht: Transportschicht	(**T**ransport Layer)
N	-Schicht: Vermittlungsschicht	(**N**etwork Layer)
DL	-Schicht: Sicherungsschicht	(**D**ata **L**ink Layer)
Ph	-Schicht: Bitübertragungsschicht	(**Ph**ysical Layer)

- **(N)-Dienstprimitive**

(N)-*Dienst.typ*

Ph	- Connect	.Req (Request, Anforderung)
DL	- Data	.Ind (Indication, Anzeige)
N	- Disconnect	.Rsp (Response, Antwort)
T	...	.Cnf (Confirmation, Bestätigung)

Von der Schichtendarstellung zum Weg-Zeit-Diagramm



3.3. Sicherungsschicht mit Fehlerbehandlung

- **Aufgaben:**
  - Kommunikation zwischen Partnern im gleichen Subnetz (über Punkt-zu-Punkt- bzw. Punkt-zu-Mehrpunkt-Verbindung)
  - Dienste der Sicherungsschicht:
    - unbestätigt (unquittiert) oder bestätigt (quittiert)
    - verbindungslos oder verbindungsorientiert (Aufbau, Datenübertragung, Abbau)
    - ungesichert oder gesichert (mit Bezug auf Übertragungsfehler)

- **Funktionalität der Sicherungsschicht:**
  - Bildung von Übertragungsrahmen
  - Fehlerbehandlung (Erkennen und Beheben von Verfälschung, Verlust)
  - Flusssteuerung zur Überlastvermeidung
  - Verbindungsverwaltung

- **Betrachteter Dienst:**
  - Gesicherter Dienst: Reihenfolgetreue Auslieferung, Fehlerbehandlung und Duplikaterkennung

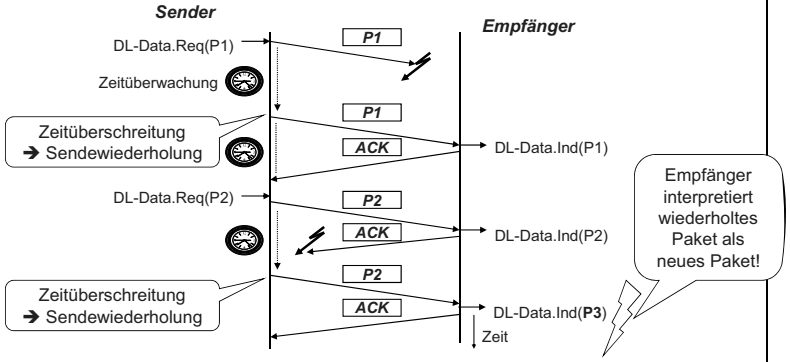
Protokolle zur Fehlerbehandlung

- Ein zuverlässiger Dienst bei unzuverlässigem Kanal benötigt ein automatisches Sendewiederholungsverfahren (ARQ - Automatic Repeat Request):
  - Einfache Protokolle
    - Protokoll mit impliziter Wiederholungsanforderung (Beispiel: Alternating-Bit-Protokoll)
    - Protokoll mit expliziter Wiederholungsanforderung
  - Schiebefensterprotokolle
    - Go-Back-N-Verfahren (mit impliziter oder expliziter Wiederholungsanforderung)
    - Selektive Wiederholung (mit impliziter oder expliziter Wiederholungsanforderung)

Fehlerbehandlung durch implizite Wiederholungsanforderung

Einfaches Bestätigungsprotokoll:

- Sender sendet ein Paket und wartet auf Empfangsbestätigung
- nach Eingang der Empfangsbestätigung wird der nächste Datenblock gesendet
- geht keine Empfangsbestätigung ein (Zeitüberschreitung), wird derselbe Datenblock erneut gesendet



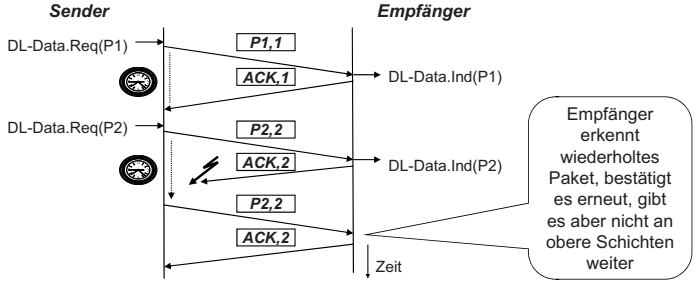
Fehlerbehandlung durch implizite Wiederholungsanforderung

Problem:

- Empfangsbestätigung muss einem Paket zugeordnet werden können

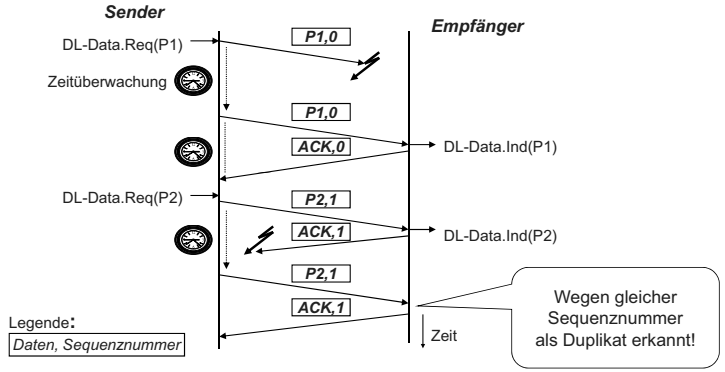
Lösung: Sequenznummern

- Sender vergibt aufsteigende Sequenznummern an versendete Pakete, Sequenznummer wird im Paketkopf als Kontrollinformation mitgesendet
- Empfangsbestätigung enthaltet Sequenznummer des bestätigten Pakets



3.3.1. Alternating-Bit-Protokoll

- Im Halbduplex-Betrieb reicht ein Sequenznummernraum {0; 1}, der zyklisch durchlaufen wird ➔ Alternating-Bit-Protokoll



- Wichtig: Zeitüberwachung größer als die RTT (Round-Trip-Time)
  - ansonsten unnötige Sendewiederholungen

Alternating-Bit-Protokoll

Programm für Sender:

```
sender(){
   int frameSent = 0;          /*Sequenznummer des Rahmens*/
   frame s,r;                  /*Sende- und Empfangsrahmen*/
   packet buffer;              /*Puffer für ausgehenden Nutzdaten*/
   eventType event;            /*FrameArrival, Timeout, CRCError*/

   FromNetworkLayer(&buffer); /*Hole erstes Paket*/

   while(true) {
      s.info = buffer;         /*Erstelle Rahmen zur Übertragung*/
      s.seq = frameSent;       /*Sequenznummer in Rahmen einf.*/
      ToPhysicalLayer(&s);     /*Sende Rahmen*/
      StartTimer();            /*Timer für Sendewiederholung*/
      wait(&event);            /*FrameArrival, Timeout, CRCError*/
      if(event == Timeout || event == CRCError) { /* nichts */ }
      if(event == FrameArrival) {         /*Gültiger Rahmen angek.*/
         FromPhysicalLayer (&r);          /*Rahmen einlesen*/
         if (r.seq != frameSent) {        /*Überprüfe Sequenznr.*/
            FromNetworkLayer(&buffer);    /*Hole nächsten R.*/
            invert(frameSent);}}          /*Invertiere Seq.-Bit*/
}
```

⇨ Wiederhole Rahmen nach Ablauf des Zeitgeber, Fehler oder doppelter Quittierung, sonst sende nächsten Rahmen

Alternating-Bit-Protokoll

Programm für Empfänger:

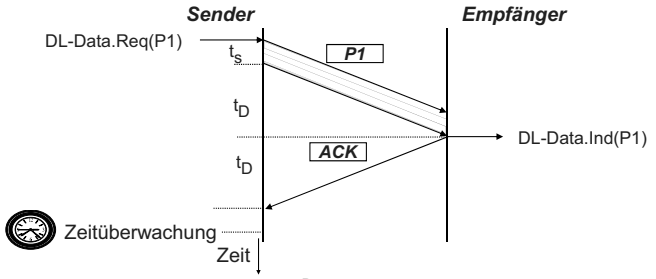
```
receiver(){
   int frameExpected = 0;      /*Erwartete Sequenznummer */
   frame r,s;                  /*r: empfangener Rahmen; s: Quittung*/
   eventType event;
   while (true) {
      wait (&event);           /*FrameArival, CRC Error*/
      if (event == FrameArrival) {  /*Gültiger Rahmen angekommen */
         FromPhysicalLayer (&r);
         if (r.seq == FrameExpected) {
             ToNetworkLayer (&r.info);
             invert (frameExpected);
         }
         s.seq = frameExpected;     /*Folgenummer quittieren*/
         ToPhysicalLayer (&s);
      }
   }
}
```

⇨ liefere nur korrekte Rahmen aus, aber bestätige *alle* Rahmen

Bei Duplex-Betrieb evtl. Anhängen der Bestätigungen an Nutzdaten von Empfänger zu Sender "Huckepack“ ("Piggyback")

Leistungsbetrachtung

- Die unterschiedlichen Protokolle können je nach Kanal zu großen Leistungsunterschieden führen



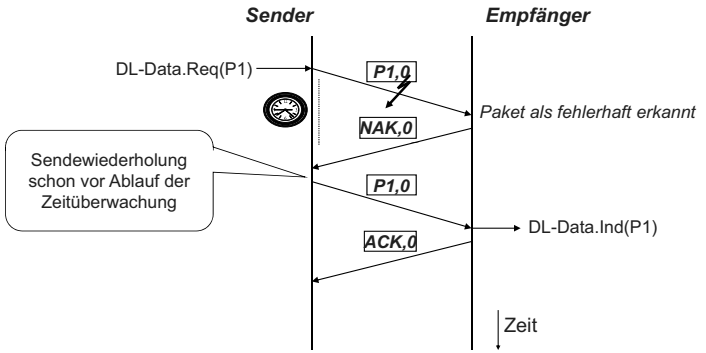
Bsp:

l_R	Rahmenlänge [bit]	l_R =1000 bit
Ü	Übertragungskapazität [bit/s]	Ü = 500 kbit/s
t_s	Sendezeit [s]	$t_s = l_R$ / Ü = 2 ms
t_D	Übertragungsverzögerung [s]	t_D = 240 ms
η	Kanalausnutzung (Effizienz)	$\eta = t_s /(t_s + 2\, t_D) \approx 0{,}4\%$

⇨ Effizienzsteigerung durch Schiebefensterprotokolle

Fehlerbehandlung durch explizite Wiederholungsanforderung

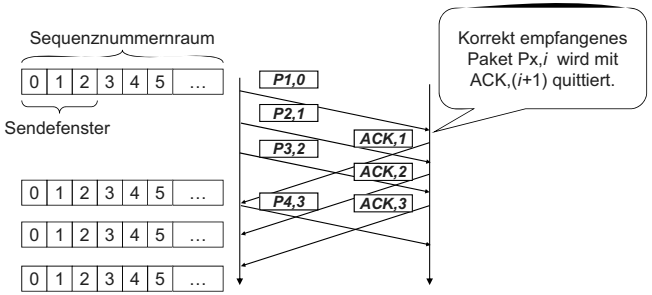
- Um den Ablauf der Übertragungswiederholung zu beschleunigen können fehlerhafte Pakete explizit durch negative Quittungen (NAK - Negative Acknowledgement) angefordert werden.



⇨ Wichtige Optimierung für Kanäle mit großen Verzögerungsschwankungen

3.3.2. Schiebefenster (Sliding Window)

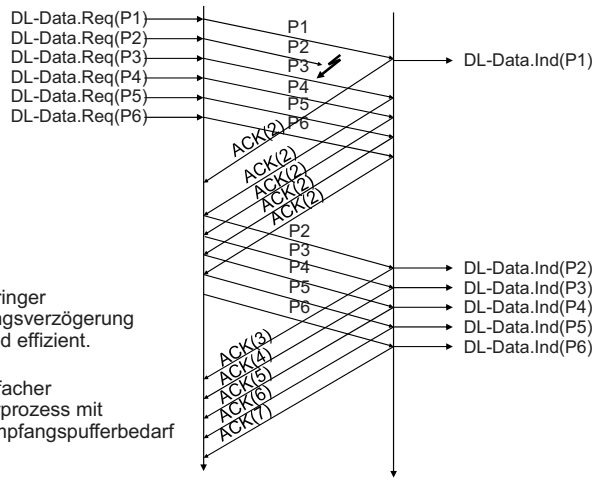
- **Ziel:**
  - Höherer Durchsatz durch Zulassung mehrere unbestätigter Pakete
- **Schiebefenster:**
  - Sequenznummernraum {0; 1; …; m-1}
  - Sender darf Sequenznummern aus vorgegebenem Sendefenster verwenden, ohne auf eine Empfangsbestätigung zu warten
  - Empfänger bestätigt Empfang mit der **nächsten erwarteten Sequenznummer**



Fehlerbehandlung bei Schiebefensterprotokolle

- **Implizite Wiederholungsanforderung:**
  - ausbleibende Empfangsbestätigungen (Zeitüberschreitung)
  - wiederholte Empfangsbestätigung für ein vorangegangenes Paket
- **Explizite Wiederholungsanforderung:**
  - Empfänger fordert Wiederholung eines bestimmten Pakets mit negativer Quittung
- **Fehlerbehandlungsverfahren:**
  - Go-Back-N:
    - sämtliche Pakete ab der fehlerhaften Sequenznummer werden erneut übertragen
  - Selektive Repeat (selektive Wiederholung):
    - nur das als fehlerhaft angegebene Paket wird erneut übertragen
    - Empfänger muss Pakete, die außer der Reihe ankommen, zwischenspeichern

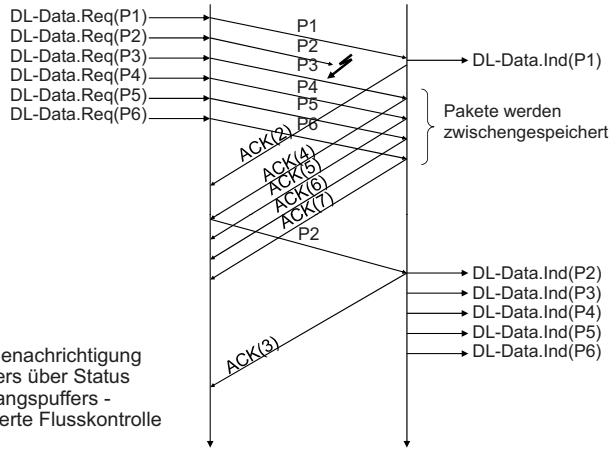
Fehlerbehandlung: Go-back-N



⇨ Nur bei geringer Übertragungsverzögerung ausreichend effizient.

⇨ Vorteil: einfacher Empfängerprozess mit kleinem Empfangspufferbedarf

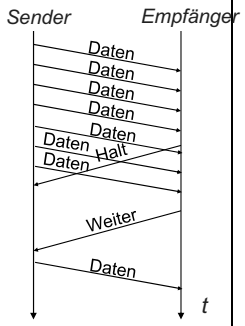
Fehlerbehandlung: Selektive Wiederholung



⇨ Wichtig: Benachrichtigung des Senders über Status des Empfangspuffers - Kreditbasierte Flusskontrolle

Flusssteuerung mit Halt-/Weiter-Meldungen (Stop-and-Wait)

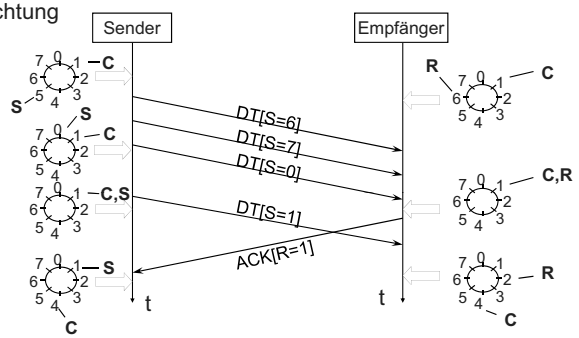
- Einfachste Methode
  - Sender-Empfänger-Flusssteuerung
    - Meldungen
      - Halt
      - Weiter
  - Kann der Empfänger nicht mehr Schritt halten, schickt er dem Sender eine Halt-Meldung.
  - Ist ein Empfang wieder möglich, gibt der Empfänger die Weiter-Meldung.



- Beispiel: Protokoll XON/XOFF
  - Mit ISO 7-Bit-Alphabetzeichen.
  - XON ist DC1 (Device Control 1).
  - XOFF ist DC3 (Device Control 3).
  - Nur auf Vollduplex-Leitungen verwendbar.

Kreditbasierte Flusssteuerung: Sliding Window

- Darstellung zeigt Fenstermechanismus (Kredit 4) für eine Senderichtung

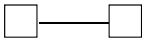


S: Sendefolgenummer (des zuletzt gesendeten Pakets)
R: Nächste erwartete Sendefolgenummer = Quittierung bis Folgenummer R-1
C: Oberer Fensterrand (maximal erlaubte Sendefolgenummer)

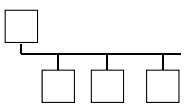
Nachteil: Kopplung von Fluss- und Fehlerkontrolle.

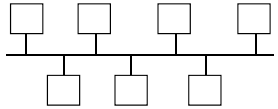
Zugriff geteiltes Medium — Arbitrierung

- Medienzugriff: Wer darf wann senden?
  - Zwei Kommunikationspartner, Punkt-zu-Punkt-Verbindung
    - Halbduplex „Richtungskonkurrenz“



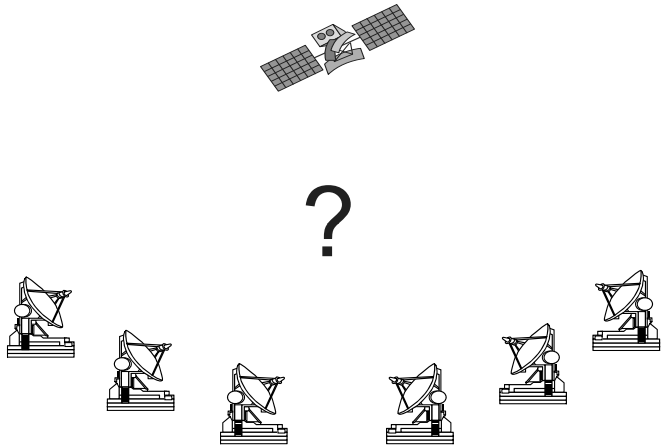
  - Mehrere Kommunikationspartner, Mehrpunktverbindung
    - Stark von Topologie der Vernetzung abhängig
    - Gemeinsames Medium (*shared medium*)
    - Wichtiger Fall: Bus (Grenzfall der Baumtopologie)


asymmetrisch


symmetrisch

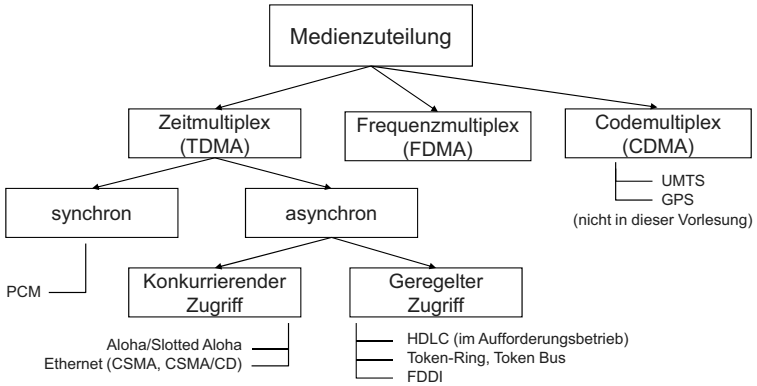
    - Weiterer Fall: Ring

Zugriffsverfahren auf ein Medium



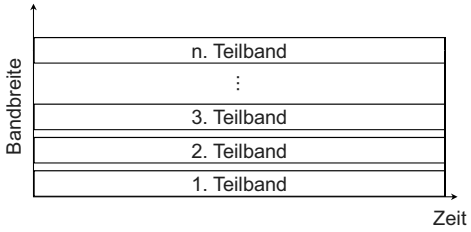
3.4. Zugriffsverfahren

- Szenario: Mehrere Stationen treten als Dienstnehmer eines einzigen physikalischen Mediums auf (shared medium)



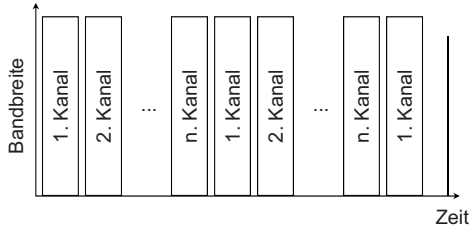
Zugriffsverfahren FDMA

- Frequency Division Multiple Access
  - Aufteilung des Frequenzspektrums, z.B. eines Satellitenkanals, in Unterkanäle
  - Jeder Teilnehmer erhält einen Unterkanal
  - Übertragung beliebig digital oder analog

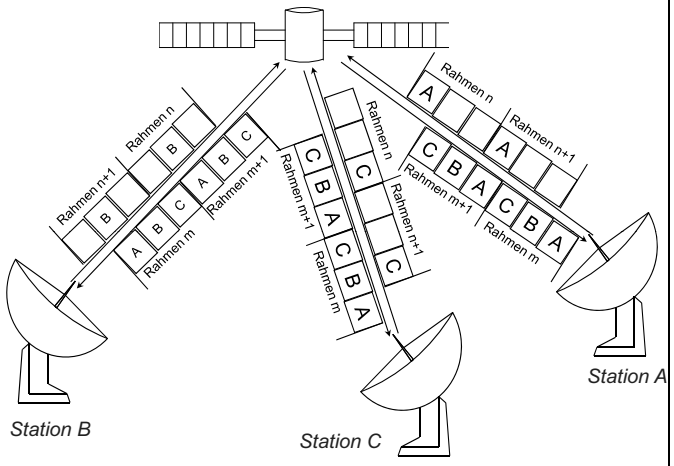


Zugriffsverfahren TDMA

- Time Division Multiple Access
  - Synchrones Zeitmultiplex, Aufteilung der Kanalkapazität nach festen Intervallen
  - Jeder Sender bekommt zyklisch einen Zeitschlitz zugewiesen
  - TDMA-Systeme arbeiten grundsätzlich digital

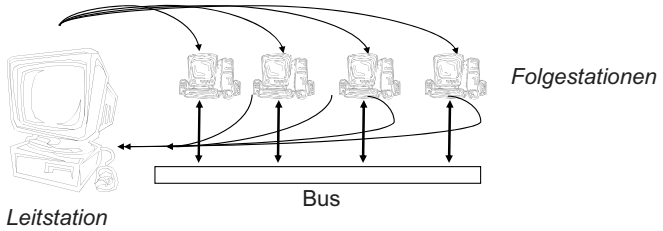


TDMA-Schema



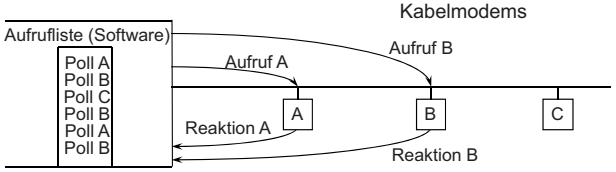
Geregelter Zugriff: Aufrufbetrieb (Polling)

- asynchroner, geregelter Medienzugriff
- eine dedizierte „intelligente“ Leitstation
- u.U. mehrere „dumme“ Folgestationen
- gekoppelt über Busstruktur
- Leitstation fragt Folgestationen gemäß Abfragetabelle („Polling Table“) ab
- Folgestationen antworten nur nach Aufforderung
- jegliche Kommunikation erfolgt über Leitstation

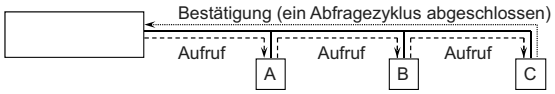


Zentrale Zuteilungsprotokolle

- Aufrufbetrieb (Poll/Select, Roll call polling), z.B. Universal Serial Bus (USB), Kabelmodems



- Variante: Go-Ahead-Polling
  - Der von der Leitstation initiierte Sendeaufruf wandert von Folgestation zu Folgestation.



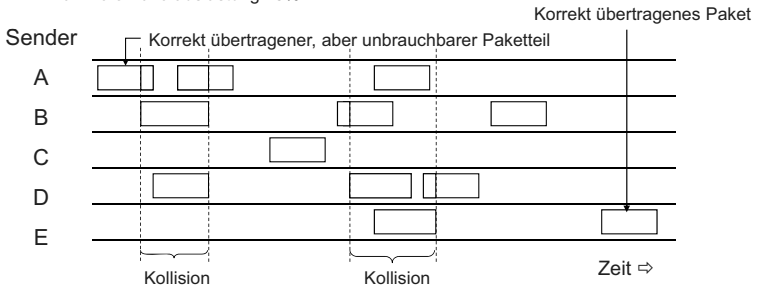
- Variante: Polling mit gemeinsamer Busleitung
  - Folgestationen teilen ihren Sendewunsch über eine gemeinsame Sammelleitung (Bus Request) der Leitstation mit.

Dezentrale Zuteilungsprotokolle

- *Zyklische Buszuteilung* (Token Passing)
  - geregelter Zugriff
  - Senderecht wird zyklisch unter den Stationen durchgereicht
  - Bsp. Token Bus, Token Ring
- *Konkurrenzbetrieb* (contention procedure)
  - konkurrierender Zugriff
  - dezentrales Wettbewerbsverfahren
  - Einsatz bei Punkt-zu-Punkt-Verbindung und Bus-basierten LAN
  - Bsp. Aloha, CSMA, CSMA/CD
- Dezentrale Protokolle werden im Folgenden ausführlicher behandelt.

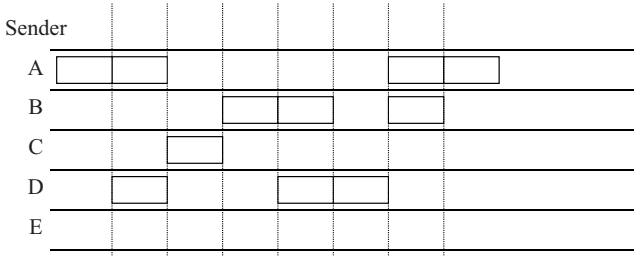
Konkurrierender Zugriff: Aloha

- Norman Abramson, University of Hawaii (1970)
- Stationen übertragen genau dann Daten, wenn welche gesendet werden müssen.
- Kollisionen führen zu gestörten Rahmen.
- Empfänger schickt Bestätigung, wenn er einen an ihn adressierten Rahmen korrekt empfangen hat.
- Einsatz beispielsweise im GSM (Signalisierungskanal)
- Feste Rahmengröße vorgeschrieben, um möglichen Datendurchsatz zu erhöhen
- Maximale Kanalauslastung 18%.



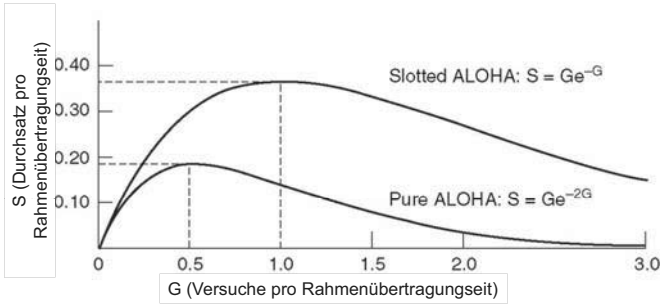
Konkurrierender Zugriff: Slotted ALOHA

- Larry Roberts, 1972
- Pakete fester Länge werden in festen Zeitabschnitten (Slots) übertragen. Dies erfordert einheitliche Zeitbasis (z.B. durch zentrale Uhr) zur Synchronisation der Stationen
- Paketübertragung nur zu Beginn eines Zeitslots (slot boundary). Es können nur total überlappende Kollisionen auftreten. Damit verkürzt sich die Kollisionszeit von zwei auf eine Paket-Übertragungszeit.
  → Maximale Kanalauslastung auf 36% verbessert!

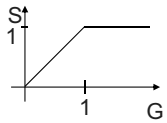


Durchsatz in Abhängigkeit vom Verkehrsaufkommen

Annahme: Rahmenankünfte Poisson-verteilt, mit mittlerer Ankunftsrate G



Idealer Verlauf wäre:

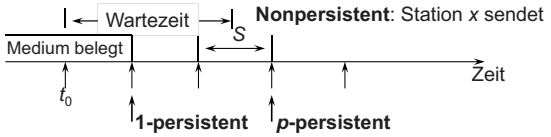


Konkurrierender Zugriff: CSMA

- ⇨Ethernet - Bob Metcalfe, 1973
- Carrier Sensing Multiple Access:
  - Sender beginnt nur bei freiem Medium zu senden (Listen before talk)
    ➔ verringert Kollisionswahrscheinlichkeit gegenüber ALOHA
- non-persistent CSMA:
  - bei belegtem Medium schaut der Sender nach einer bestimmten Wartezeit wieder nach, ob das Medium frei ist
- 1-persistent CSMA:
  - bei belegtem Medium hört der Sender das Medium weiter ab, bis es frei wird. Danach beginnt er sofort zu senden.
  - Nachteil: sichere Kollision, wenn mehrere Sender senden möchten
- p-persistent CSMA:
  - bei belegtem Medium hört der Sender das Medium weiter ab, bis es frei wird. Danach beginnt er mit Wahrscheinkchkeit p zu senden oder wartet einen weiteren Zeitschlitz (um dann wieder nur mit Wahrscheinlichkeit p zu senden usw.).
  - geringere Kollisionswahrscheinlichkeit als bei 1-persistent
- Kollisionen sind trotzdem nicht vollständig ausgeschlossen!

Konkurrierender Zugriff: CSMA

- **Carrier Sense Multiple Access (CSMA)** bzw. **Listen before Talk (LBT)**
  - Sendewillige Station hört Medium ab und sendet, falls dieses frei ist
  - Erhöhte Kollisionsgefahr nach Ende einer Übertragung

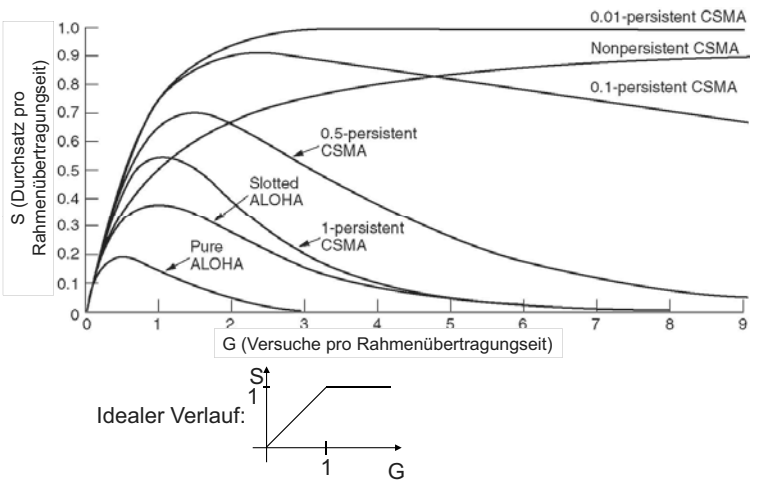


☐ **Nonpersistent CSMA:**
(1) Wenn frei, übertrage sofort
(2) Wenn belegt, warte gewisse (feste, zufällige) Zeit, dann (1)

☐ **1-persistent CSMA:**
(1) Wenn frei, übertrage sofort
(2) Wenn belegt, warte bis frei und übertrage

☐ ***p*-persistent CSMA:**
(1) Wenn frei, übertrage mit Wahrscheinlichkeit *p* oder verzögere um einen Zeitslot mit 1 – *p*
(2) Wenn belegt, warte bis frei, dann (1)
(3) Wenn um 1 Slot (*S*) verzögert, dann (1)

Durchsatz in Abhängigkeit vom Verkehrsaufkommen



Idealer Verlauf:

CSMA: Mögliche Kollisionsbehandlung

- Behandlung auf höheren Protokollschichten:
  - fehlerhafte Übertragungen werden vom Empfänger erkannt bzw. ignoriert
    ➔ ausbleibende Empfangsbestätigungen und Sendewiederholung
- Kollisionserkennung (CSMA/CD, Collision Detection):
  - Sender hört während des Sendens das Medium weiter ab, um Kollisionen zu erkennen (Listen while talk)
  - bei erkannter Kollision sofortiger Abbruch des Sendevorgangs, ggf. Jamming-Signal zur Benachrichtung aller beteiligten Sender (siehe Ethernet)
  - Sendewiederholung nach zufälliger Wartezeit, um erneute Kollision zu vermeiden
  - Binary Exponential Backoff: Verdopplung der mittleren Wartezeit nach jeder erneuten Kollision

3.5. Protokolle der Sicherungsschicht

- Im Weiteren betrachtet:
  - HDLC (High-Level Data Link Control)
  - PPP (Point-to-Point Protocol)
  - Ethernet

3.5.1. HDLC-Protokoll

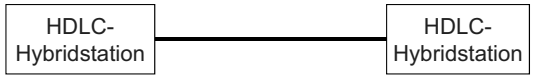
- High-Level Data Link Control (HDLC)
  - Bit-orientiertes, code-transparentes Sicherungsschichtprotokoll
  - Codetransparenz durch Bit-Stuffing
  - Halb- und vollduplexfähig
  - Punkt-zu-Punkt- und Mehrpunkt-Konfiguration
  - Symmetrische und unsymmetrische Konfiguration
  - Piggybacking
  - Flusskontrolle durch „Sliding Window"-Technik
  - Varianten zum HDLC-Protokoll:
    - SDLC von IBM (Synchronous Data Link Control)
    - LAPB (Link Access Procedure, Balanced)
    - LAPD (ISDN, Link Access Procedure, D-Kanal)
    - LLC (IEEE 802.2, Logical Link Control)
    - PPP (Point-to-Point Protocol)

HDLC: Konfigurationen

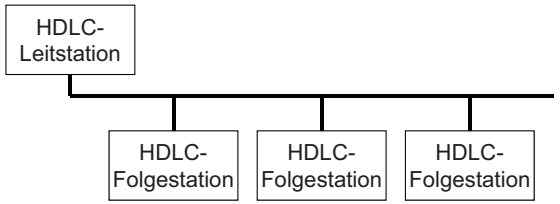
- Zu unterscheiden:
  - *Leitsteuerung*, die Befehle aussendet;
  - *Folgesteuerung*, die Meldungen als Reaktion auf Befehle aussendet.
- Drei Fälle des Datenflusses:
  - **unsymmetrische (zentrale) Steuerung:** *Empfangsaufruf*
    Die Leitsteuerung mit Datenquelle (Leitstation) fordert die Folgesteuerung (Folgestation) durch Befehl zum Datenempfang auf.
  - **unsymmetrische (zentrale) Steuerung:** *Sendeaufruf*
    Die Leitsteuerung mit Datensenke fordert die Folgesteuerung mit Datenquelle durch Befehl zum Senden von Daten auf (z.B. für Mehrpunkt-Verbindung).
  - **symmetrische (gleichberechtigte) Steuerung**
    Zwei *Hybridstationen*, die als Überlagerungen von je einer Leit- und Folgesteuerung aufzufassen sind, können Meldungen und Befehle ausgeben.

HDLC: Verbindungstopologien

- Punkt-zu-Punkt-Verbindung:



- Asymmetrische Mehrpunktverbindung:



HDLC: Betriebsarten

- **Aufforderungsbetrieb** (NRM — *Normal Response Mode*).
  - Folgestation darf nur nach ausdrücklicher Erlaubnis durch Leitstation Meldungen senden.
- **Spontanbetrieb** (ARM — *Asynchronous Response Mode*)
  - Folgestation kann jederzeit Meldungen an Leitstation senden.
- **Gleichberechtigter Spontanbetrieb** (ABM — *Asynchronous Balanced Mode*)
  - Beide Hybridstationen dürfen jederzeit Meldungen und Befehle übermitteln.

HDLC: Datenübermittlungsblock

01111110	8 bit	8 bit	n bit	16 bit	**01111110**
Block-begrenzung	Adressfeld	Steuerfeld	Datenfeld	Block-prüfung	Block-begrenzung

- Datenübermittlungsblock liefert Rahmen (HDLC-Frame) für Übermittlung von Befehlen, Meldungen und Daten.
- Blockbegrenzung (Flag): Feste Codierung zur Synchronisation
- Adressfeld (Address Field):
  - Befehl: Zieladresse;
  - Meldung: Herkunftsadresse
- Steuerfeld (Control Field): Festlegung von Befehlen und Meldungen
- Datenfeld (Information Field): Beliebige Bit-Folge ($n \geq 0$; n nicht notwendigerweise Vielfaches von 8), benötigt Bit Stuffing
- Blockprüfungsfeld (Frame Check Sequence, FCS): CRC-Verfahren

HDLC: Datenübermittlungsblock und Steuerfeld

01111110	8 bit	8 bit	n bit	16 bit	01111110
Block-begrenzung	Adressfeld	Steuerfeld	Datenfeld	Block-prüfung	Block-begrenzung

0	N(S)	P/F	N(R)	I-Block (Datenblock) [I=Information]
1 0	S S	P/F	N(R)	S-Block (Steuerblock) [S=Supervisory]
1 1	M M	P/F	M M M	U-Block (Steuerblock) [U=Unnumbered]

HDLC: Aufbau des Steuerfeldes

Steuerfeldformat für	Bit-Nummer 1	2	3	4	5	6	7	8
I-Block (Datenblock) [I=Information]	**0**	N(S)			P/F	N(R)		
S-Block (Steuerblock) [S=Supervisory]	**1**	**0**	S	S	P/F	N(R)		
U-Block (Steuerblock) [U=Unnumbered]	**1**	**1**	M	M	P/F	M	M	M

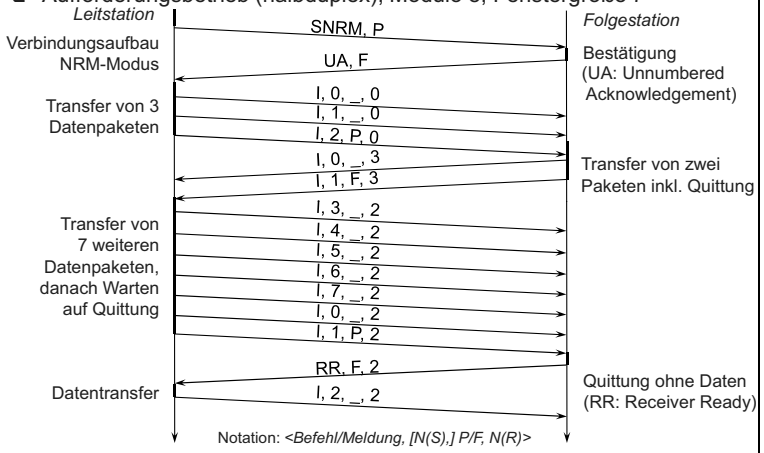
- Sendesequenznummer N(S); Empfangssequenznummer N(R) je 3 bit lang
- I-Block: Übertragung von Nutzdaten
- S-Block: Steuerblock, Übertragungssteuerung (Befehle, Meldungen) mittels S-Bits wie z.B. Sendeaufruf, Bestätigung empfangener DÜ-Blöcke
- U-Block: Steuerblock ohne Folgenummer
  Zusätzliche Übertragungssteuerungsfunktionen, jedoch ohne Empfangsfolgenummer; Codierung durch M-Bits (max. 32 Befehle; z.Zt. 13 Befehle und 8 Meldungen festgelegt.)

HDLC: Markierungsbit — Poll/Final (P/F)

- P/F-Bit hat unterschiedliche Bedeutung in Befehlen und Meldungen sowie in den einzelnen Betriebsarten:
  - *P/F=1 in Befehlen:*
    Anforderung einer Meldung, bzw. einer Folge von Meldungen (Poll)
  - *P/F=1 in Meldungen:*
    Bestätigung des Empfangs eines Befehls mit PF=1,
    d.h. Meldungen als Antwort auf Befehle mit PF=1 (Final).
  - *Gebrauch im Normal Response Mode:*
    Folgestation darf nach Senden einer Meldung mit P/F=1 als Antwort auf Befehl mit P/F=1 keine weiteren DÜ-Blöcke ohne Erlaubnis durch die Leitstation senden.
  - *Gebrauch im Asynchronous Response / Balanced Mode:*
    Auf einen Befehl mit P/F=1 muss vorrangig durch eine oder mehrere Meldungen mit P/F=1 geantwortet werden, jedoch sind weiterhin Meldungen mit P/F=0 möglich.

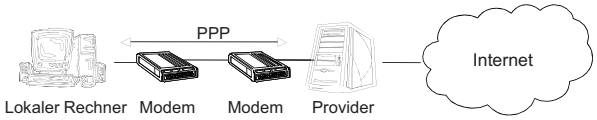
HDLC: Beispielablauf

- Aufforderungsbetrieb (halbduplex), Modulo 8, Fenstergröße 7



3.5.2. PPP (Point-to-Point Protocol)

- Der größte Teil des Internets beruht auf Punkt-zu-Punkt-Verbindungen:
  - Verbindungen im WAN zwischen Routern / Heimanbindung über Modem und Telefonleitung
- SLIP (serial line IP, RFC 1055): keine Fehlererkennung, nur IP, keine dynamische Adresszuweisung, keine Authentifizierung



- PPP (RFC 1661/1662):
  - Schicht-2-Rahmenformat mit Fehlererkennung, Rahmenbegrenzung
  - Steuerprotokoll (LCP, Link Control Protocol) zum Verbindungsaufbau,
  - Verbindungstest, Verbindungsverhandlung, Verbindungsabbau
  - Verhandlung von Schicht-3-Optionen unabhängig vom Schicht-3-Protokoll
  - (separates NCP, Network Control Protocol, für alle unterstützten Schicht-3-Protokolle)

PPP-Paketformat

- Paketformat an HDLC angelehnt

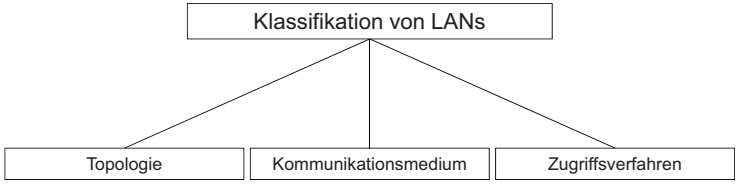
1	1	1	1 oder 2	variabel	2 oder 4	1	Bytes
Flag 01111110	Address 11111111	Control 00000011	Protocol	Payload	Checksum	Flag 01111110	

  - zeichenorientiert (anstatt bitorientiert), d.h. die Länge das Nutzdatenfeldes endet immer an einer Byte-Grenze
  - Codetransparenz durch Character Stuffing
  - typischerweise werden nur *unnumbered*-frames übertragen, bei hohen Fehlerraten (Mobilkommunikation) kann jedoch auch der zuverlässigere Modus mit Sequenznummern und Bestätigungen gewählt werden
  - als Protokolle im Nutzlast-Feld sind u.a. IP, AppleTalk, IPX definiert
  - falls nicht anderweitig verhandelt, ist die maximale Länge der Nutzlast auf 1500 Byte begrenzt
  - durch zusätzliche Verhandlung kann der Paketkopf verkleinert werden

PPP-Verbindung

- Typisches Szenario beim Zugriff eines PCs auf das Internet via Modem
  - Anruf beim Service-Provider via Modem und Aufbau einer physikalischen Verbindung
  - Anrufer sendet mehrere LCP-Pakete im PPP-Rahmen zur Auswahl der gewünschten PPP-Parameter
  - Austausch von NCP-Paketen, um Vermittlungsschicht zu konfigurieren
    - z.B. kann hier dynamisch mittels DHCP (s.u.) eine IP-Adresse zugewiesen werden falls IP als Protokoll gewählt wurde
  - Der Anrufer kann nun genauso wie ein fest verbundener Rechner Internet-Dienste nutzen
  - Zur Beendigung der Verbindung wird via NCP die IP-Adresse wieder freigegeben und die Vermittlungsschichtverbindung abgebaut
  - Über LCP wird die Schicht 2-Verbindung beendet, schließlich trennt das Modem die physikalische Verbindung

Lokale Netze (LAN, Local Area Network): Klassifikation

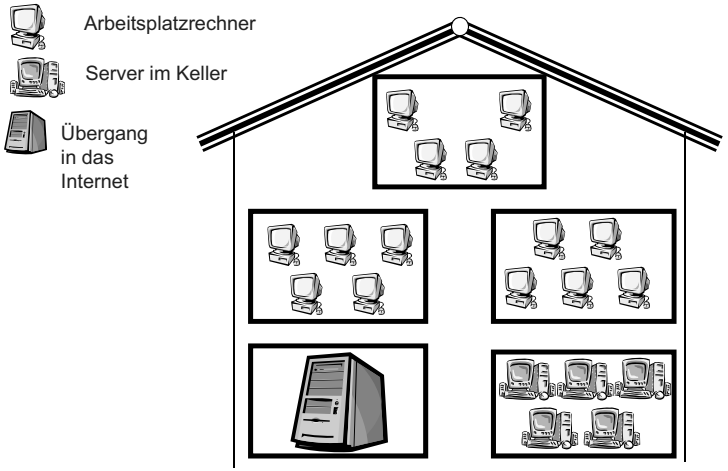


- Stern
- Bus
- Baum
- Ring
- ...

- Verdrilltes Adernpaar
- Koaxialkabel
- Glasfaser
- ‚Luft'

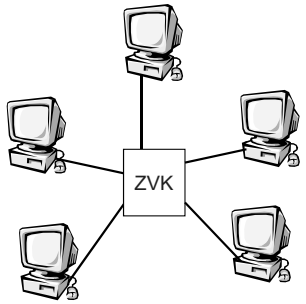
- kontrolliert
- konkurrierend

Leitbeispiel: Strukturierte Verkabelung



LAN-Topologie Stern

- Verbindungsstruktur: Sterntopologie
- im „In-House“-Bereich meist ein zentraler Vermittlungsknoten



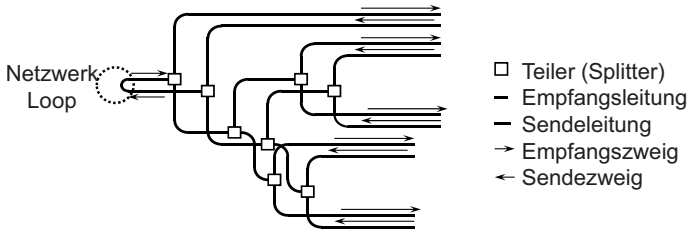
ZVK: Zentraler Vermittlungsknoten *(Central Switching Element, Switch, Hub)*

LAN-Topologie Stern – Grundprinzipien

- Grundprinzipien:
  - Exklusive Punkt-zu-Punkt-Verbindung zwischen Station und Zentrale
  - Kommunikation zwischen Stationen ausschließlich über Zentrale
  - Stationen sind z.B. Telefon, PC, Notebook, Gateways als Internetzwerkeinheit
  - Vermittlungstechniken:
    - Raummultiplex (analog/digital): ZVK mit Durchschaltevermittlung
      Bsp: Telefon-Nebenstellenanlage, Private Branche Exchange PBX
    - Paketvermittlung: ZVK mit Speichervermittlung
      (meist Datagramm-Vermittlung)
  - Anwendungsbeispiele:
    - Digitale Nebenstellenanlagen
    - Strukturierte Verkabelung bei Ethernet

LAN-Topologie Baum

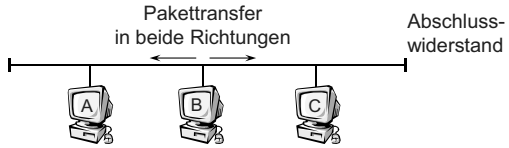
- Verbindungsstruktur:
  - Mehrpunkt (multi-point)
- Vielfachzugriff (multiple access):
  - Angeschlossene Teilnehmer haben Zugriff zum gleichen Übertragungskanal
- Verteilnetz (broadcast medium):
  - Alle Teilnehmer empfangen sämtliche Nachrichten.



LAN-Topologie Bus

- Verbindungsstruktur: Mehrpunkt (multi-point)
- Vielfachzugriff (multiple access)
- Verteilnetz (broadcast medium)
  - Passive Ankopplung der Stationen. Keine Verstärkung/Signalformung/ Signalwandlung an den Ankopplungspunkten der Stationen
  - Empfangen von Daten durch Kopieren

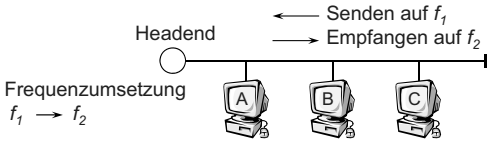
- Übertragungstechniken:
  1. Basisband-Bussystem



Stationsankopplung mit Lese- und Schreibeinrichtung

LAN-Topologie Bus für Breitbandtechnologie

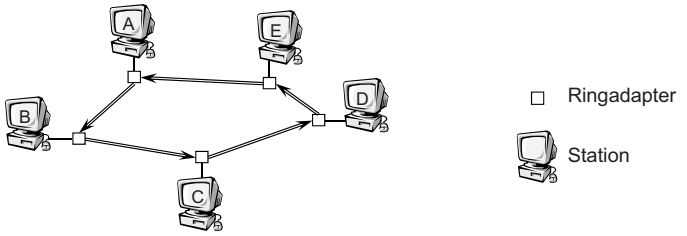
- Breitband-Bussystem (z.B. Kabelfernsehen)



	Sendefrequenz	Empfangsfrequenz
Subsplit	5-30 MHz	54-400 MHz
Midsplit	5-116 MHz	168-400 MHz
Highsplit	5-174 MHz	232-400 MHz

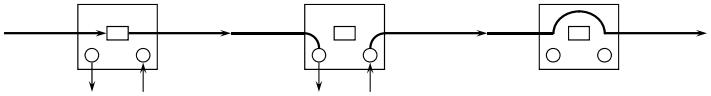
LAN-Topologie Ring

- Verbindungsstruktur:
  - geschlossene Folge von unidirektionalen Punkt-zu-Punkt-Verbindungen.
- Zugriff:
  - zum Ring über Ringschnittstelle/Ringadapter
- Aktiver Ring:
  - Ringadapter sind aktive Signalregeneratoren mit einem Zwischenpuffer und damit einer Verzögerung um mindestens ein Bit (1-Bit-Delay).



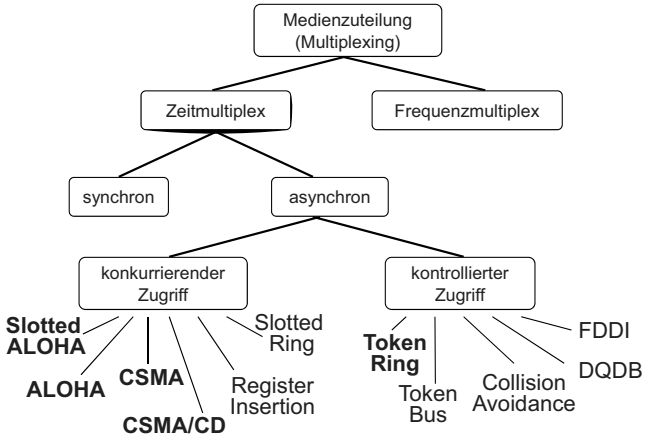
LAN-Topologie Ring – Ringadapter

Zustände des Ringadapters:

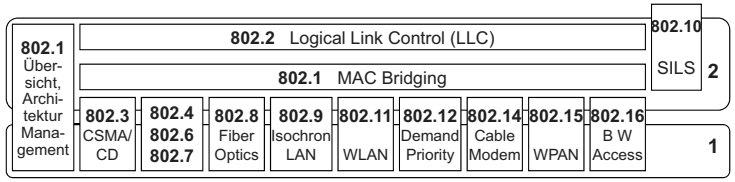


- Abhör-Zustand
  - Abhören des vorbeilaufenden Bitstroms
  - Kopieren des Bitstroms „Kopieren im Fluge“
  - Modifikation des Bitstroms möglich
- Sende-Zustand
  - Aussenden der Sendebits
  - Einbehalten und Überprüfen des ankommenden Sendeblocks
- Überbrückungszustand
  - passiver Anschluss

LAN/MAN: Zugriffsverfahren in der Übersicht



LAN/MAN: Standardisierung nach IEEE 802



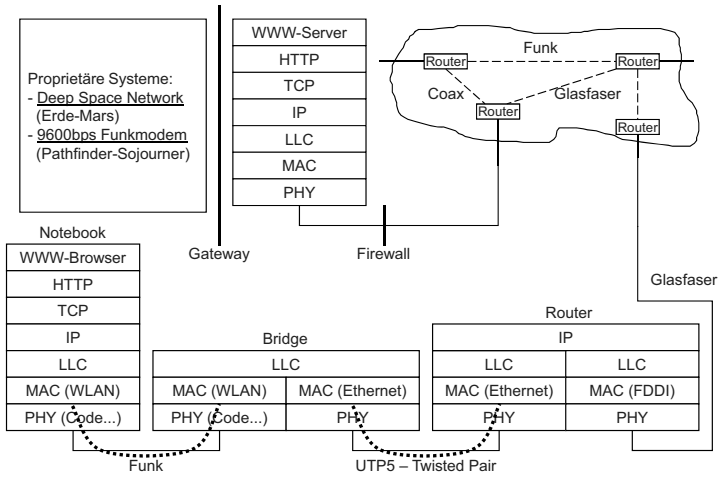
Themen:

802.1: Zusammenhang der Standards und MAC Bridging
802.2: Logical-Link-Control-Dienste/Protokolle (LLC) *inaktiv*
802.3: CSMA/CD-Protokoll auf Bustopologie
802.4: Token-Bus-Protokoll auf Bustopologie *inaktiv*
802.5: Token-Ring-Protokoll auf Ringtopologie
802.6: Metropolitan Area Network *inaktiv → 802.14*
802.7: Broadband TAG (Technical Advisory Group) *inaktiv*
802.8: Fiber Optic TAG
802.9: Isochronous LAN
802.10: Sicherheitsstruktur für 802-Protokolle *inaktiv*
802.11: Wireless LANs
802.12: Demand Priority Working Group
802.14: Cable Modem: Datenübertragung über TV-Kabelmodems
802.15: Wireless Personal Area Networks: Netzwerke über kurze Distanzen
802.16: Broadband Wireless Access: Drahtloser Zugriff auf Breitband-Systeme
...
802.21: Enable handover and interoperability between heterogeneous networks
802.22: Wireless Regional Area Networks ("WRANs") http://ieee802.org/22/

LAN: Standardisierung nach ISO/OSI

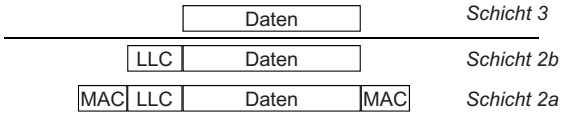
- Erweiterung des OSI-Schichtenmodells:
- Unterteilung der Schicht 2 in zwei Unterschichten.

MAC und LLC im Beispiel



LAN: Logical Link Control (LLC) IEEE802.2 [inaktiv!]

- **Hauptaufgabe:**
  - Verdecken unterschiedlicher MAC-Verfahren
  - Drei Diensttypen:
    - **LLC-Typ 1: unzuverlässiger Datagrammdienst (typisch für LANs)**
    - LLC-Typ 2: verbindungsorientierter Dienst
    - LLC-Typ 3: bestätigter Datagrammdienst
- **Format:**
  - vereinfachte Version von HDLC (nur Asynchronous Balanced Mode Extended)
  - Adressen implizieren das verwendete Protokoll der Schicht 3

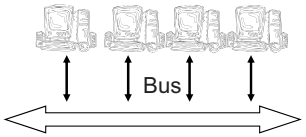


3.5.3. Wiederholung: CSMA/CD – Funktionsprinzip

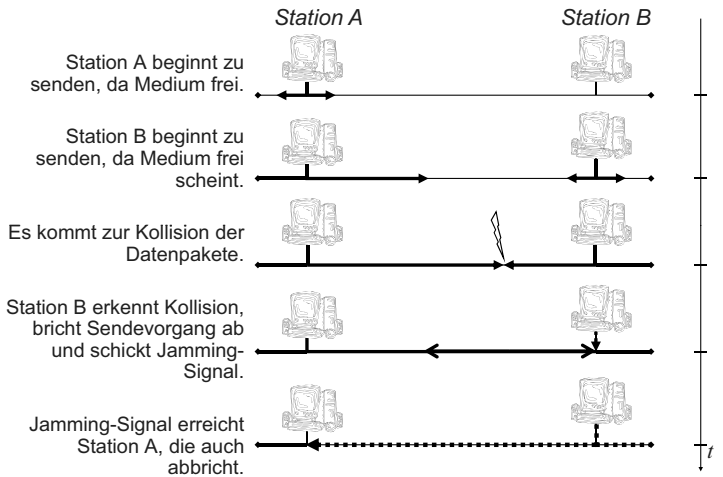
- **Funktionsprinzip:**
  - Listen before Talk (*Carrier Sense Multiple Access*, CSMA)
  - Listen while Talk (*with Collision Detection*, CD)
  - Bustopologie mit Mehrfach-Zugriffsverfahren
  - Konkurrierendes Zugriffsverfahren (Wettbewerb)
- **Betriebsablauf:**
  - Senden, wenn Medium aktuell frei
  - Bei Kollision: Jamming-Signal; Abbruch der Sendung
  - Kollisiondetektion erfordert Mindestlänge der 802.3-MAC-Blöcke!
    → Sendung darf nach Signallaufzeit von A nach B und zurück noch nicht beendet sein
  - Nach Kollision erneuter Anlauf nach statistisch verteilter *p*-persistent-Verzögerungszeit
  - Blockmindestlänge hängt von Mediumslänge und Übertragungs-geschwindigkeit ab. Sie wird durch Stopffeld (Pad) sichergestellt

CSMA/CD („Ethernet“)

- alle Stationen an einem gemeinsamen Bus angeschlossen.
- keine ausgezeichnete Station.
- jede Station kann zu einem beliebigen Zeitpunkt senden.
  ⇨ **Kollisionen mehrerer Sendungen zerstören übertragene Daten!**
- Vermeidung von Kollisionen:
  Carrier Sense Multiple Access with Collision Detection (CSMA/CD).
- Grundlagen von CSMA/CD:
  - vor dem Senden: Abhören des Mediums (*Listen Before Talk*).
  - wenn Medium frei: Beginne mit Senden.
  - während des Sendens: Abhören des Mediums (*Listen While Talk*).
  - wird Kollision erkannt: Breche Sendevorgang ab und benachrichtige die anderen angeschlossenen Stationen.



Ablaufbeispiel CSMA/CD



Paketformat CSMA/CD nach IEEE 802.3/Ethernet

Typ

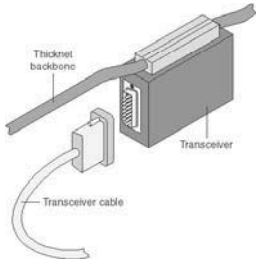
PR	SD	DA	SA	Länge	Data	PAD	FCS
56 bit	(8 bit)	(16/48 bit)	(16/48 bit)	(16 bit)	(≤12.000 bit)	(0-368 bit)	(32 bit)

PR = Präambel zur Synchronisation (1010101010...)
SD = *Start-of-frame Delimiter* zeigt Blockbeginn an (10101011)
DA = *Destination Address*, Zieladresse
SA = *Source Address*, Herkunftsadresse
Länge = Anzahl der Oktette im Datenfeld
Typ = Protokolltyp der Nutzdaten (z.B. IP, ARP, IPX...)
Data = Datenfeld, das maximal 1.500 Byte umfassen darf
PAD = *Padding*, um zu kurze Datenfelder auf die nötige Länge zu ergänzen
FCS = *Frame Check Sequence*, Polynomdivision mittels CRC32-Polynom zur Fehlererkennung

Wichtig: Einzelne Realisierungen von CSMA/CD (z.B. Ethernet 1.0, Ethernet 2.0 oder IEEE 802.3) verwenden manche Felder in leicht unterschiedlicher Bedeutung!

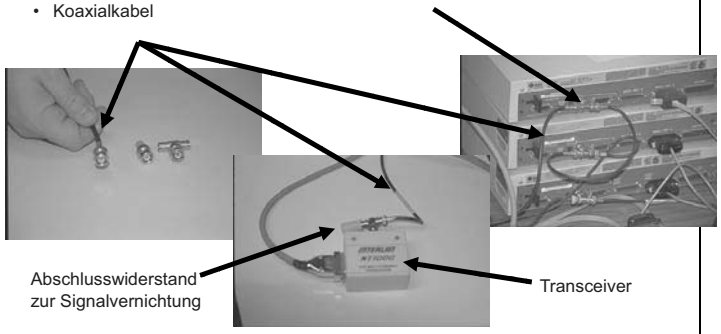
LAN: CSMA/CD – Technische Realisierung: 10Base5

- Technische Realisierung: **Ethernet** (in zwei Versionen 1.0 und 2.0)
  - **10Base5, Thick Ethernet**
    - fingerdickes, gelbes, 4-fach abgeschirmtes Koaxialkabel
    - 10Base5: 10 Mbit/s, Basisbandübertragung, 500 Meter-Segmente
    - Segment-Kopplung über Repeater (max. 5 Segmente)
    - 100 Teilnehmer pro Segment mit mindestens 2,5 m Abstand
    - Teilnehmeranschluss über Transceiver (Transmitter & Receiver), entspricht MAU (Medium Attachement Unit). Transceiver enthält Sende-/Empfangslogik, Kollisionserkennung, „Carrier Sensing"-Funktion.
    - Ein-, Zwei- oder Vierfach-Transceiver
    - Transceiverkabel zum Teilnehmer bis 50 Meter
    - Lösung: robust, teuer, unflexibel



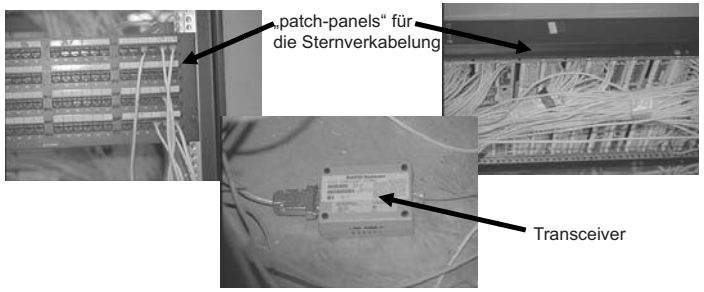
LAN: CSMA/CD – Technische Realisierung: 10Base2

- **10Base2, Thin Wire Ethernet, Cheapernet**
  - 10Base2: 10 MBit/s, Basisbandübertragung, 185 Meter-Segmente
  - 30 Teilnehmer pro Segment im Abstand von mindestens 0,5 m
  - Transceiver meist direkt auf Ethernet-Adapter im Rechner (BNC-Buchse, T-Stück)
  - Koaxialkabel



LAN: CSMA/CD – Technische Realisierung: 10Base-T

- **10Base-T, Twisted Pair**
  - 10 MBit/s, Basisbandübertragung
  - Verdrillte Leitungen
  - jede Station (max. 1024) ist über (max. 100 m) Punkt-zu-Punkt-Verbindung an Multiport-Repeater angeschlossen (Hub, Verteilerkasten, Konzentrator)
  - In USA Telefonkabel einsetzbar



LAN: CSMA/CD – Ausdehnungsproblematik

- Ausdehnungsproblematik:
- Basisband-Ethernet beschränkt auf max. 1.500-2.500 Meter (inkl. Repeater)
- Größere Distanzen erreichbar mittels:
  - Remote Repeater (Fiber Optic Inter-Repeater Link FOIRL)
    Geteilter Repeater mit bis 1 km Glasfaserübertragungsstrecke
- Weitere Ansätze:
  - 10Base-F
    - Verbindung von Hubs (Sternkoppler) über Glasfaser
    - bis zu 4.000 Meter Glasfaserübertragungsstrecke
  - 10Broad36
    - 10 MBit/s, Breitbandübertragung (Ethernet über CATV-Leitungen), max. Entfernung: 3.600 Meter
    - Frequenzmultiplex mit Head-End

3.6. Fast-Ethernet-Standard 100Base-T (1)

- IEEE 802.3u: 100Base-T-Technologie
  - 1995 standardisiert („Fast Ethernet“)
  - Vertreter: Grand Junction Networks, Digital, Intel, SUN, Synoptics, 3Com,...
  - Charakteristika
    - Erhaltung des CSMA/CD-Frame-Formats und Medienzugriffverfahrens
    - Übertragungsraten von 10-100 Mbit/s
    - Flexibles Verkabelungskonzept (Hierarchie von Hubs)
    - Kompatibilität zum existierenden Ethernet-Standard ➔ einfache Migration
    - „Autonegotiation“: Protokoll zur automatischen Festlegung der Übertragungsrate
  - Realisierung

    (a) 100Base-T4
    - Basierend auf Twisted-Pair-Verkabelung der Kategorie 3 (bis 16 MHz)
      - Maximal 100 Meter Kabellänge (zwischen Netzwerkkarte und Hub)
      - Daten werden bei der Übertragung auf 4 Adernpaare aufgeteilt
    - Nur Halbduplex-Verkehr möglich

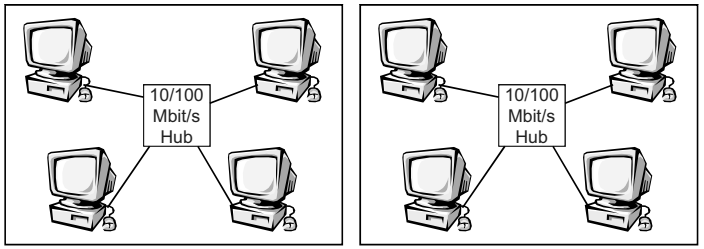
Fast-Ethernet-Standard 100Base-T (2)

(b) 100Base-Tx
- Signalisierung nach FDDI-Standard (ANSI X3T9.5)
- Basiert auf Twisted-Pair-Verkabelung der Kategorie 5 (bis 100 MHz)
  - Max. 100 Meter Kabellänge (zwischen Netzwerkkarte und Hub)
  - Verwendung von 2 Adernpaaren: Je eines für Senden und Empfangen
- Datenverkehr vollduplex
- Datencodierung mittels 4B5B-Verfahren (wie bei FDDI)
  - 4 bit werden mit 5 bit codiert
    → Schrittgeschwindigkeit von 125 Mbaud benötigt!

(c) 100Base-Fx
- Multimode- oder Monomode-Fasern (benötigt zwei Lichtwellenleitungen)
- Sternförmige Verkabelung
- Datenverkehr vollduplex
- Max. 400 Meter

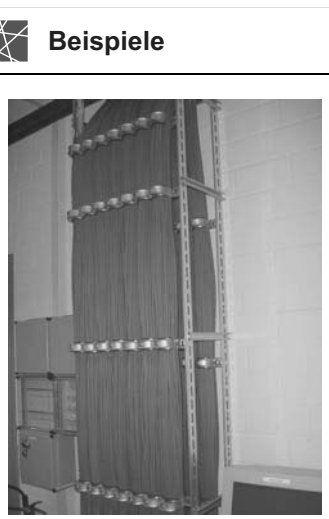
Leitbeispiel: Strukturierte Verkabelung

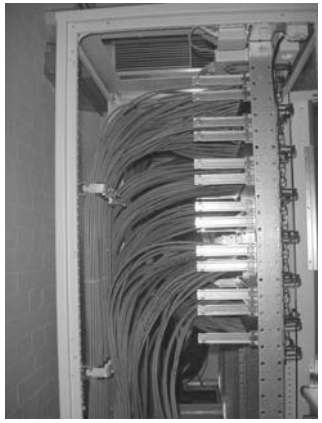
- **Strukturierte Verkabelung:** Aufteilung eines Netzes in mehrere Kabelstrecken, die über ein Backbone oder einen zentralen Switch/Hub zusammengefasst sind.
  - Vernetzung der einzelnen Räume: Pro Raum ein zentraler Hub der die einzelnen Rechner miteinander koppelt
  - Anschluss mittels Twisted-Pair (Kategorie 5) an den Hub verbunden
  - Als Protokoll wird in den meisten Fällen Ethernet bzw. Fast-Ethernet verwendet
  - Fast Ethernet ist inzwischen Standard



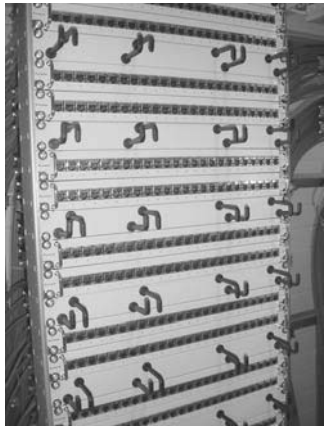
Beispiele

Beispiele

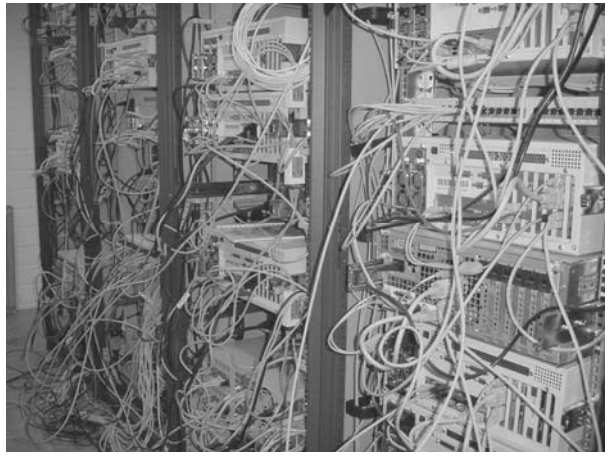




Beispiele



Beispiele



Ethernet-Weiterentwicklung: Gigabit Ethernet

- „Gigabit Ethernet“ (auch **1000Base-X** genannt):
  Gigabit Ethernet Alliance (GEA) - Zusammenschluss mehrerer Hardware-Hersteller
  → Informationen: http://www.gigabit-ethernet.org
- **Grundgedanke:**
  - logischer Schritt von 10 Mbit/s Ethernet über Fast-Ethernet (100 Mbit/s) zu Gigabit-Ethernet (1.000 Mbit/s)
  - ursprüngliches Ziel: Beibehaltung des CSMA/CD-Verfahrens
  - Ideale Ergänzung zu (Fast-) Ethernet zur Fortführung im Backbone-Bereich
  - Einsatz auf Glasfaser und Kupferadern
- **Probleme:**
  - Min. Rahmenlänge von (Fast-) Ethernet zu klein, um Kollisionen zu erkennen
    - Kompatibilität zu (Fast-) Ethernet möglich?
  - Strenge Normen zur elektromagnetischen Verträglichkeit
  - Elektro-physikalische Eigenschaften der Kupferadern: Übersprechen, Dämpfung, ...

Gigabit-Ethernet im Detail

- Duplex-Betrieb und Halbduplex-Betrieb möglich
  - Praktisch alle neueren Komponenten sind vollduplexfähig
- Problem: Beim Halbduplex-Betrieb Kollisionserkennung notwendig
  → min. Rahmenlänge von (Fast-) Ethernet zu klein!
- Lösung:
  - Bei Gigabit-Ethernet musste daher die minimale Rahmengröße von 64 byte auf 512 byte erhöht werden
  - Auffüllen kleinerer Rahmen mit Füllzeichen (Extension Symbols: "Carrier Extension"), die an die FCS angehängt werden, jedoch keine Bedeutung haben
    - Weiterhin minimale Rahmengröße von 64 byte
  - Packet Bursting: Zusammenfassen mehrerer kleiner Pakete eines Senders möglich
- Burst Limit: Beschränkung der max. Sendezeit auf 65.536 bit
- Gleiches Rahmenformat wie bei (Fast-)Ethernet auf MAC-Ebene
  - Auf der physikalischen Ebene jedoch leichte Unterschiede, die aber für MAC transparent sind
- Umschalten zwischen 10, 100, 1.000 Mbit/s mittels Autonegotiation möglich

Gigabit-Ethernet: 1000Base-T

- Wegen der weiten Verbreitung ungeschirmter verdrillter Doppeladern soll Gigabit-Ethernet auch für diese Kabeltypen (UTP-5) möglich sein. Aber:
  - Länge von bis zu 100 m soll beibehalten werden
  - Wegen 8B/10B-Codierung: 1.250 Mbaud zur Übertragung notwendig!
  - Probleme mit Übersprechen, Dämpfung, Abstrahlung, ...
- Lösung: Parallele Nutzung aller vier Doppeladern:
  - Sternförmige Verkabelung, keine Kollisionen
  - Steuerung des Senders durch Sternpunkt
  - Modulationsverfahren: PAM5 (Pulsamplitudenmodulation mit fünf Zuständen)
    - 125 Mbaud pro Adernpaar
    - 2bit/Symbol → 250 Mbps pro Adernpaar
    - Über die 4 Adernpaare kann pro Signalschritt 1 byte übertragen werden; insg. 1 Gbps
  - Codierungsverfahren: Trellis-Codierung und Scrambling (Verwürfeln)
  - Duplexbetrieb durch Echokompensation
  - Standardisierung: IEEE 802.3ab

Gigabit-Ethernet: Glasfaser und geschirmte Doppelader

- Standards für Glasfaser: 1000Base-SX (Standard, Multimode-Glasfaser), 1000Base-LX („luxury“, Multi- und Monomode-Glasfaser)
- 1000Base-CX (twinax, geschirmte Doppelader)
- Codierung: 8B/10B anhand einer Codetabelle
- Standardisierung durch IEEE 802.3z

Standard	Kabeltyp	Durchmesser	Wellenlänge	Segmentlänge
1000Base-SX	Multimode	62,5 µm	830 nm	2-275 m
	Multimode	50 µm	830 nm	2-550 m
1000Base-LX	Multimode	62,5 µm	1270 nm	2-550 m
	Multimode	50 µm	1270 nm	2-550 m
	Monomode	10 µm	1270 nm	2-5.000 m
1000Base-CX	Twinax			25 m

Twinax-Kabel: 

Weiterentwicklungen: 10 Gigabit Ethernet

- Glasfasermedium: IEEE 802.3ae,
- Kupfermedium: IEEE 802.3ak und IEEE 802.3an.
- http://www.10gea.org/
- Datenrate 10.000 Mbit/s
- Auch ungeschirmte Doppeladern möglich (802.3ab)

Neuste Entwicklung:
IEEE 802.3ba
40 Gbit/s-Ethernet,
100 Gbit/s-Ethernet
über Glasfaser
sowie <10m Kupferkabel

Bezeichnung	Kabel	Segmentlänge
10GBase-T	4 ungeschirmte Adernpaare (CAT6a)	100m
10GBase-CX4	2 geschirmte Adern-paare (Doppel-Twinax)	15m
10GBase-SR	Multimode-Glasfaser	26 - 300 m
10GBase-LR 10GBase-ER	Monomode-Glasfaser	10.000 m 40.000 m
10GBase-LX4 10GBase-LW4	Multi-/ Monomode-Glasfaser	240-300m 10.000 m

Lehrstuhl für Netzarchitekturen und Netzdienste
Institut für Informatik – Technische Universität München
Prof. Dr.-Ing. Georg Carle

Grundlagen: Rechnernetze und Verteilte Systeme

Kapitel 4: Vermittlung

Paket-/Leitungsvermittlung, Brücke, Router

Prof. Dr.-Ing. Georg Carle
Lehrstuhl für Netzarchitekturen und Netzdienste
Technische Universität München
carle@net.in.tum.de
http://www.net.in.tum.de

Übersicht

1. Einführung und Motivation
   - Bedeutung, Beispiele
2. Begriffswelt und Standards
   - Dienst, Protokoll, Standardisierung
3. Direktverbindungsnetze
   - Fehlererkennung, Protokolle
   - Ethernet
4. **Vermittlung**
   - **Vermittlungsprinzipien**
   - **Wegwahlverfahren**
5. Internet-Protokolle
   - IP, ARP, DHCP, ICMP
   - Routing-Protokolle
6. Transportprotokolle
   - UDP, TCP
7. Verkehrssteuerung
   - Kriterien, Mechanismen
   - Verkehrssteuerung im Internet
8. Anwendungsorientierte Protokolle und Mechanismen
   - Netzmanagement
   - DNS, SMTP, HTTP
9. Verteilte Systeme
   - Middleware
   - RPC, RMI
   - Web Services
10. Netzsicherheit
    - Kryptographische Mechanismen und Dienste
    - Protokolle mit sicheren Diensten: IPSec etc.
    - Firewalls, Intrusion Detection
11. Nachrichtentechnik
    - Daten, Signal, Medien, Physik
12. Bitübertragungsschicht
    - Codierung
    - Modems

Ziele

- In diesem Kapitel wollen wir vermitteln
  - Grundverständnis von Netzwerkkoppelung
  - Funktionalität von Repeater, Brücken und LANs
  - Vermittlungsprinzipien
  - Funktionalität von Routern

Kapitelgliederung

4.1. Netzwerkkopplung
- 4.1.1. Repeater
- 4.1.2. Hub
- 4.1.3. Brücke (Bridge)
- 4.1.4. Spanning-Tree-Algorithmus
- 4.1.5. Remote-Brücke
- 4.1.6. Switched LAN
- 4.1.7. Virtuelle LANs
- 4.1.8. Leitbeispiel: Strukturierte Verkabelung

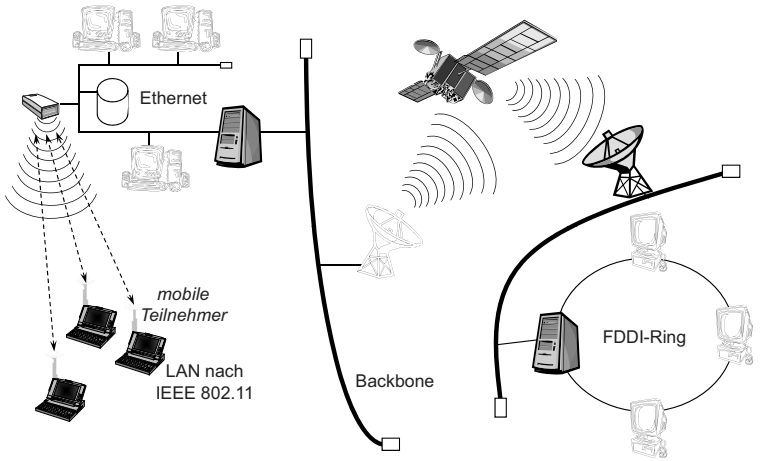
4.2. Vermittlungsprinzipien für globale Netze
- 4.2.1. Durchschaltevermittlung
- 4.2.2. Nachrichten-/Speichervermittlung
- 4.2.3. Paketvermittlung
- 4.2.4. Router
- 4.2.5. Routing-Verfahren

Inhalt des 1. Teils des Kapitels

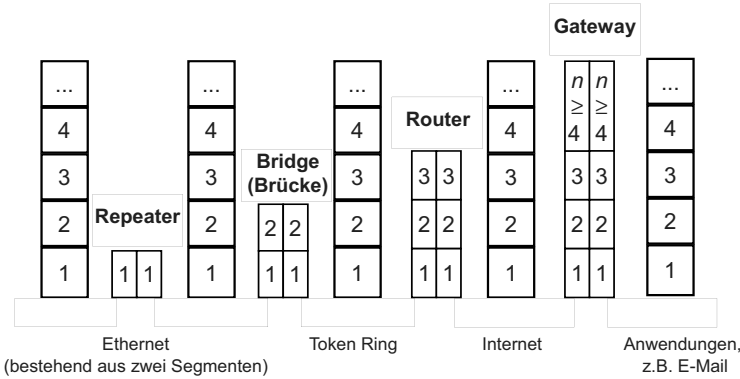
4.1. Netzwerkkopplung
auf Schicht 2 (Ethernet)

- Repeater, Hub
- Brücke (Bridge)
- Spanning-Tree-Algorithmus
- Remote-Brücke
- LAN-Switch
- Virtuelle LANs
- Strukturierte Verkabelung

Netzwerkkopplung (allgemein)

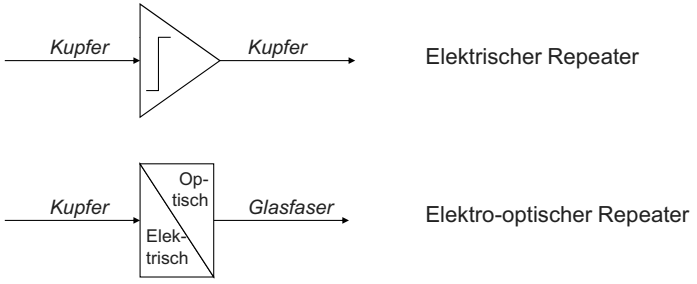


Kopplung von Netzen – Internetworking



Repeater

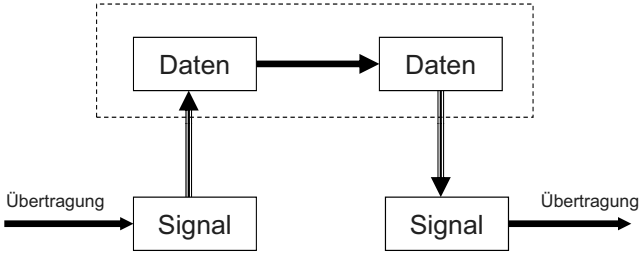
- Kopplung physikalischer Medien durch Signalregeneration/-verstärkung
- Keine Zwischenspeicherung
- Keine Bearbeitung der Pakete
- Medien können unterschiedlich sein, Protokoll auf Schicht 2 muss identisch sein



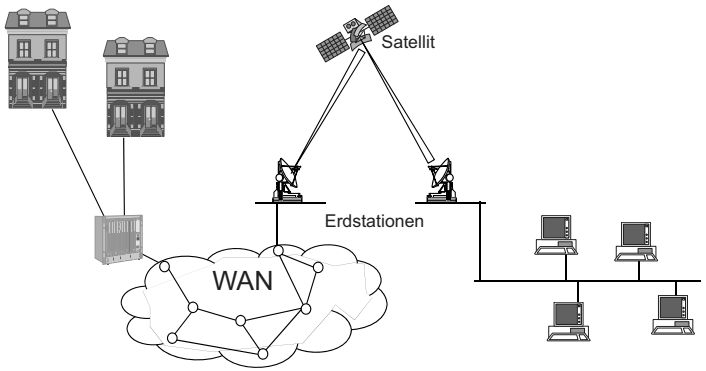
4.1.1. Repeater

- Kopplung von Netzwerken auf Schicht 1
  - Einsatzbereich:
    - Verbindung von lokalen Netzen zur Erhöhung der räumlichen Ausdehnung
    - Generierung mehrerer abgehender Signale an Verzweigungspunkten
    - Wechsel des Übertragungsmediums (z.B. Kupfer auf Glasfaser, Leitung auf Funkstrecke)
  - Vorteile:
    - Einfache Technik
    - Kostengünstige Lösung
    - Keine Verarbeitung an den Daten, somit keine Beeinträchtigung der Geschwindigkeit
    - Extrem lange Netzwerkverbindungen sind möglich (z.B. Überseeleitungen)
  - Nachteile:
    - Keine Intelligenz; alle Daten werden weitergeleitet
    - Keine Erhöhung der Netzkapazität durch Partitionierung

Wdh.: Digitale Regeneration über abstrakte Datenrepräsentation

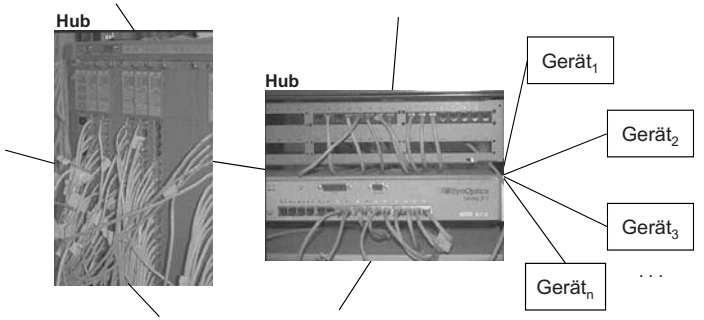


Übertragung über mehrere Teilstrecken

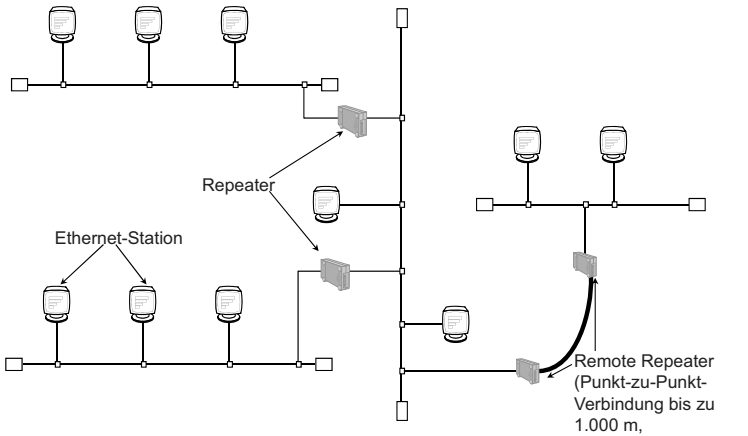


4.1.2. Hub

- Gleiche Funktion wie Repeater
  - typischerweise in Sterntopologie (separate Leitung von jedem Gerät zum Hub), kaskadierbar
  - Gesamtdurchsatz des Netzes wird nicht erhöht (vgl. Switch)

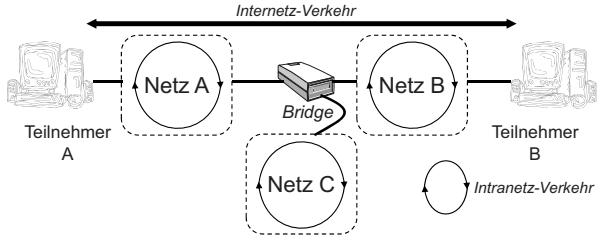


Netz-Kopplung auf Schicht 1: Beispiel Ethernet



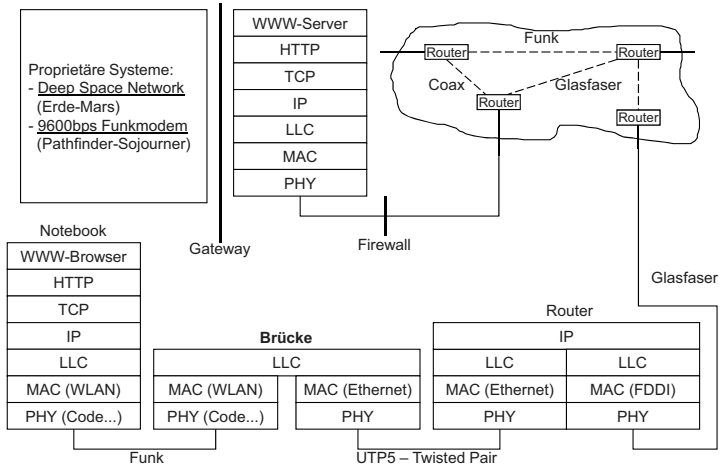
4.1.3. Brücke (Bridge)

- Kopplung von Netzen auf Schicht 2
  - Netzwerke vom gleichen Typ (z.B. 802.x mit 802.x) (non-translating)
  - Netzwerke unterschiedlichen Typs (z.B. 802.x mit 802.y (x≠y)) (translating)
- Aufgaben:
  - Trennen des Intranetz-Verkehrs vom Internetz-Verkehr (Filterfunktion)
  - Erhöhung der Netzkapazität großer Netzwerke durch Partitionierung (jede Partition mit voller Stationszahl/Längenausdehnung)
  - Durchführung einfacher Wegewahlfunktionen (Einfache Vermittlung auch auf Schicht 2)

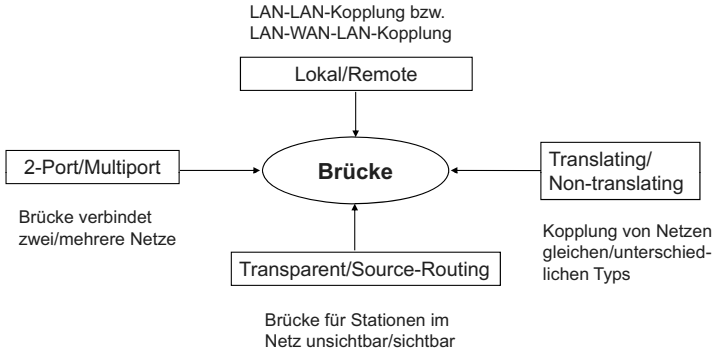


1

Brücken: ein Beispiel



Brücken – Übersicht

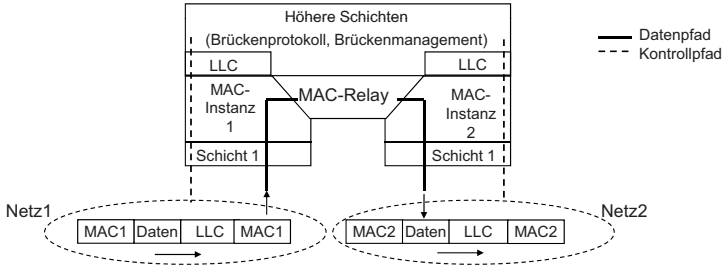


Typen von Brücken

- Zwei Basisvarianten von Brücken:
  - **Source-Route-Brücken (IEEE 802.5):**
    - Weiterleitungsinformation für die Brücke wird vom Endsystem in den Datenpaketen spezifiziert
    - Wenig Aufgaben innerhalb der Brücke, daher einfache Realisierbarkeit
    - In der Praxis wenig eingesetzt
  - **Transparente Brücken (IEEE 802.1D):**
    - Weit verbreiteter Brückentyp
    - Weiterleitungsentscheidung wird von der Brücke eigenständig getroffen
    - Brücke verwaltet in der Regel eine Tabelle (die Filterdatenbasis), in der sie Information über die Lokation von Endsystemen sammelt (d.h. sie lernt Adressen)
    - Das Vorhandensein einer Brücke zum Zielsystem bleibt dem sendenden Endsystem verborgen

Transparente Brücke (MAC-Bridge)

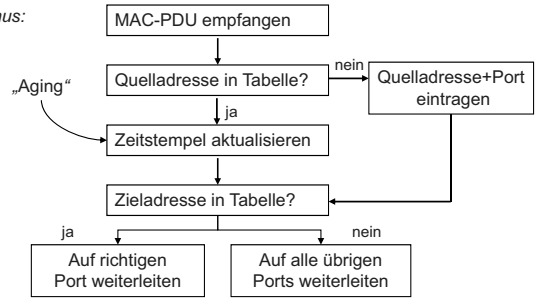
- Merkmale:
  - Lokale, translating Bridge
  - Für jedes Netzwerk eine eigene Schicht-1- und MAC-Instanz
  - Die MAC-Instanzen werden über ein MAC-Relay verbunden; dieses nimmt die Weiterleitungs- und Filterfunktion wahr. Beim Weiterleiten verändert eine transparente Brücke die Schicht-2-Adressen von Quelle und Ziel nicht.
  - LLC-Instanzen nur für die höheren Schichten der Brücke (Brückenprotokoll, Brückenmanagement)



Transparente Brücke – Arbeitsweise

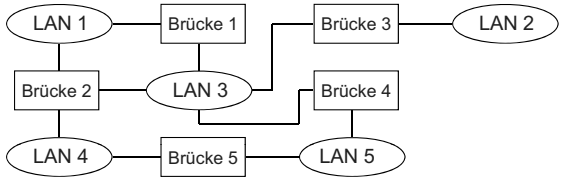
- Alle PDUs werden empfangen („promiscuous mode“)
- Brücke verwaltet eine Tabelle (forwarding database), in der sie Informationen über die Lokation (Ausgangsleitung) von Endsystemen sammelt (d.h. sie lernt MAC-Adressen)
- Das Vorhandensein einer Brücke zum Zielsystem bleibt dem sendenden Endsystem verborgen

Forwarding-Algorithmus:



Brücken: Redundante Wege

- Bei brückengekoppelten Netzwerken können redundante Wege zwischen zwei Netzen existieren (z.B. zur Fehlertoleranz

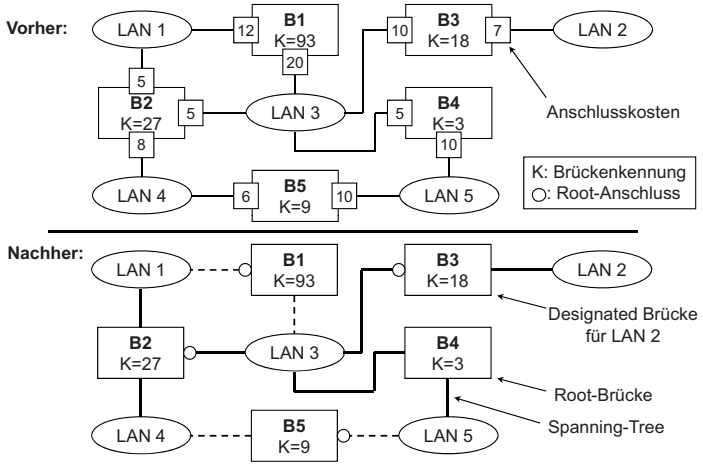


- **Probleme:**
  - Repliziert empfangene Datenpakete (über verschiedene Wege)
  - Endlos kreisende Datenpakete (Schleifen)
- **Lösung:**
  - Etablierung einer *logischen Baumstruktur* über allen Brücken der involvierten Netzwerke (⇨ Spanning-Tree-Algorithmus)
  - Weiterleiten von Datenpaketen nur entlang der Baumstruktur (eindeutiger Pfad), restliche Brücken blockieren ihre Ports

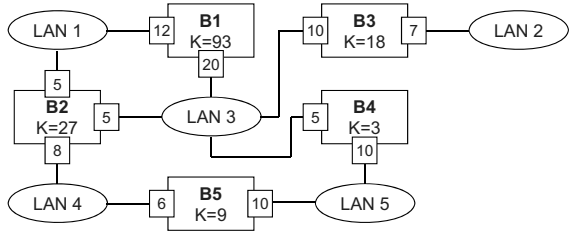
4.1.4. Spanning-Tree-Algorithmus

- Voraussetzungen:
  - Gruppenadresse zur Adressierung aller Brücken im Netzverbund
  - Eindeutige Brückenkennungen (MAC-Address)
  - Eindeutige Anschlusskennungen in jeder Brücke (MAC-Address)
  - Kosten an allen Anschlüssen einer Brücke („Anschlusskosten)
- Ablauf:
  1. Bestimmen der Root-Brücke (Wurzel des Baumes):
     - Zuerst nimmt jede Brücke an, dass sie Root-Brücke ist
     - Root-Brücken senden regelmäßig Hello-Pakete mit ihrer Brückenkennung aus
     - Bei Erhalt eines Hello-Pakets mit kleinerer Brückenkennung ordnet sich eine Root-Brücke der anderen unter und sendet das Paket als Broadcast
  2. Bestimmen der Root-Ports
     - Root-Anschluss (Root-Port) einer Bücke: dies ist der Port, über den der günstigste Pfad Richtung Root-Brücke verläuft (nur Kosten für Ausgangsports werden berücksichtigt)
     - Summe über alle Anschlusskosten auf dem Weg zur Root-Brücke ist zu minimieren
     - Übertragungsgeschwindigkeit kann als Kostenfunktion dienen
  3. Bestimmen der Designated-Brücke:
     - Brücke mit günstigstem Root-Anschluss in einem Netzwerk wird als Designated-Brücke bestimmt
     - Root-Brücke ist Designated-Brücke für alle an sie angeschlossenen Netze

Spanning-Tree-Algorithmus: Beispiel

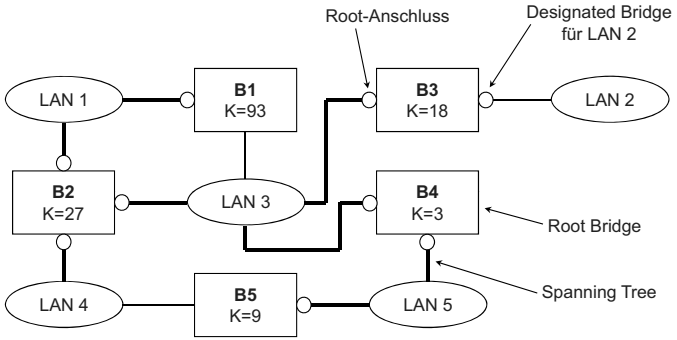


Spanning-Tree-Alg.: Berechnung der Root-Pfade

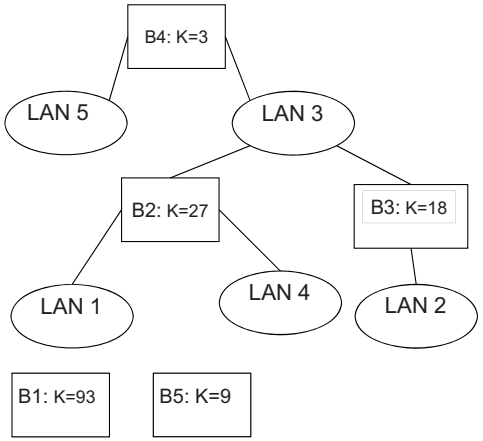


Brücke	Kosten des Weges zur Root Bridge
B3	**10** (via LAN 3)
B1	20 (via LAN 3)
	17 = 12 + 5 (via LAN 1 & LAN 3)
B2	**5** (via LAN 3)
	18 = 8 + 10 (via LAN 4 & LAN 5)
	25 = 5 + 20 (via LAN 1 & LAN 3)
B5	**10** (via LAN 5)
	11 = 6 + 5 (via LAN 4 und LAN 3)

Spanning-Tree-Algorithmus: Bestimmung der Designated Bridges



Spanning-Tree-Algorithmus (Ergebnis)



LAN	Designated Bridge
LAN 1	B2
LAN4	B2
LAN2	B3
LAN3	B4 (Root Bridge)
LAN5	B4 (Root Bridge)
---	B5
---	B1

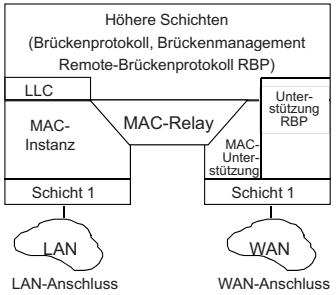
4.1.5. Remote-Brücken

- Kopplung von entfernten LANs über ein oder mehrere WANs

- Merkmale:
  - Einkapselung von MAC-Dateneinheiten (keine Umsetzung)
  - Remote-Brücken müssen paarweise auftreten
  - Nur zur Kommunikation zwischen LAN 1 und LAN 2 (Transparent)
  - Keine Kommunikation von LAN 1 (oder 2) mit dem WAN
  - Netzanschlüsse, die nicht mit einem LAN verbunden sind, werden als **virtuelle Anschlüsse** bezeichnet
- Beispiele:
  - Verbindung zweier Ethernets über ISDN
  - Verbindung zweier FDDI-Netze über ATM

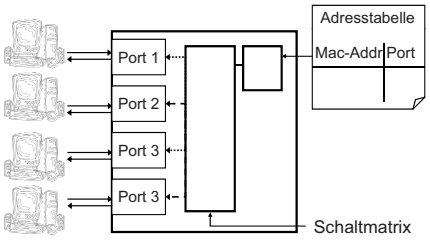
Architektur einer Remote-Brücke

- Merkmale:
  - MAC-Unterstützung am virtuellen Anschluss für das MAC-Relay (zum WAN)
  - Remote-Brückenprotokoll (RBP) zwischen virtuellen Anschlüssen regelt die Punkt-zu-Punkt-Kommunikation der entfernten Brücken (über das WAN)
  - Brückenprotokoll setzt im WAN auf dem RBP auf



4.1.6. Switched LANs

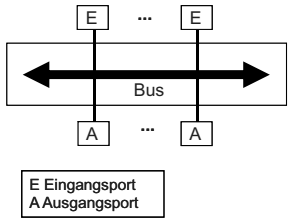
- Sternförmige Topologie, zentrale Komponente = *Switch*
- Jedes angeschlossene Gerät erhält die **volle Bandbreite**
- Sende-/Empfangsrichtung meist getrennt (Vollduplex-Übertragung)
- Parallele Bearbeitung/Weiterleitung mehrerer eingehender Rahmen
- Verbindung Eingang- zu Ausgangsport über Schaltmatrix anhand Adresstabelle

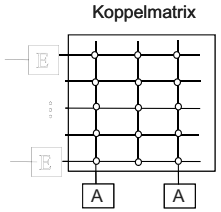


LAN-Switch

Schaltmatrix (switching fabric)

- **Aufgabe:** Schalten einer Verbindung zwischen 2 Ports (je Datenpaket)
- Realisierung in Hardware durch Application Specific Integrated Circuits (ASICs)
- Weiterleitung wird realisiert durch
  - ein-/mehrstufige Koppelmatrix, Bus, Ring, gemeinsamer Speicher
- Speicherelemente zur Konfliktauflösung notwendig
  - Eingangsspeicher, Ausgangsspeicher, verteilter Speicher





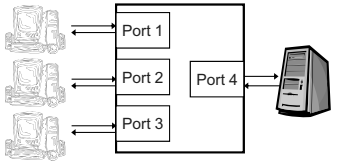
HoL - Head of Line Blocking

Switched LANs – Mechanismen

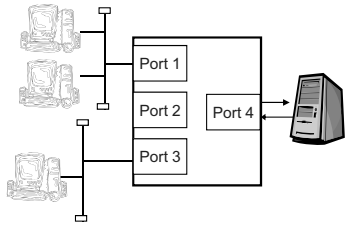
- Cut Through
  - Adresstabelle wird angesprochen, sobald die Zieladresse eingelesen ist
  - Weiterleitung des Datenpakets, sobald der Weg geschaltet ist
  - Geringe Latenzzeit
- Store and Forward
  - Datenpaket wird zunächst vollständig eingelesen und zwischengespeichert
  - Kontrolle der CRC-Prüfsumme und Ausführen von Filterfunktionen
- Hybrides Switching
  - Kombination von Cut Through / Store and Forward
  - Auswahl abhängig von Fehlerrate
- Predictive Switching
  - Pfad in Schaltmatrix wird hergestellt, bevor Zieladresse vollständig eingelesen
  - Basierend auf den vorher geschalteten Pfaden

Ausprägungen

- Port-Switching
  - Nur ein Gerät pro Port (eine MAC-Adresse)
  - Jedes Gerät erhält volle Bandbreite
  - Schneller Table-Lookup möglich



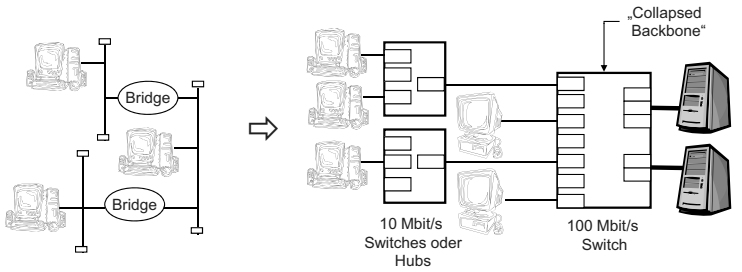
- Segment-Switching
  - Mehrere Adressen pro Port zulässig
  - Bandbreite wird jedem Segment zur Verfügung gestellt



- Bank-Switching
  - Mehrere Ports teilen sich eine bestimmte Übertragungrate

Einsatzmöglichkeiten

- Erforderliche Umstrukturierung des Netzwerks
  - Kopplung mehrerer LANs durch ein Hochgeschwindigkeits-LAN (Backbone)
  - Distributed Backbone ⇔ Collapsed Backbone
  - Hochleistungs-Workstations an dediziertem Port
  - Server an mehreren Ports (direkter Anschluss an unterschiedlichen LANs, oder auch Port-Bündelung ⇨ Fat Pipe)



Weitere Switching-Verfahren

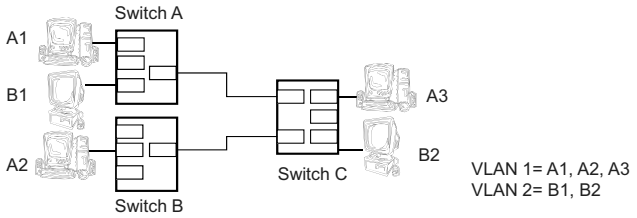
- Layer-3-Switching
  - Integration von Routing und Switching
  - Weiterleitung von Paketen anhand der Analyse der Felder des Schicht-3-Protokolls
- Layer-4-Switching
  - Erweiterung des Layer-3-Switchings um Analyse der Felder der Schicht 4 (z.B. Portnummern von TCP)
  - Priorisierung bestimmter Anwendungen
- IP Switching
  - IP-Switch = Kombination aus IP-Router (RIP, OSPF, BGP) und Layer-2-Switch
  - Zunächst konventionelles IP-Routing
  - Längeranhaltende Verkehrsflüsse werden erkannt, klassifiziert und gekennzeichnet
  - Unter Umgehung der Routing-Funktionalität wird danach auf Schicht 2 durch das Netz „geswitcht“, z.B. mithilfe von ATM
  - Heute wichtige Variante: MPLS – Multi-Protocol Label Switching

4.1.7. Virtuelle LANs (I)

- VLAN

  „Eine nach bestimmten Kriterien definierbare Broadcast-Domäne“

- Ziel: Trennung von physikalischer und logischer Netzwerkstruktur
  - Datenpakete werden ausschließlich innerhalb des jeweiligen VLANs verteilt
  - Mitglieder eines VLANs können räumlich verteilt sein, z.B. an verschiedenen LAN-Switches
    ⇨ Unabhängigkeit von Standort und VLAN-Zugehörigkeit



Virtuelle LANs: Vorteile, Realisierung (I)

- Vorteile (allgemein)
  - Einschränkung von Broadcasts/Multicasts ⇨ bessere Ausnutzung der Bandbreite
  - Effizientere Verwaltung durch vereinfachte Konfiguration
    - z.B. bei Änderungen der Netztopologie durch Umzug
  - Erhöhte Sicherheit
    - Authentifizierung vor dem Beitritt einer Station zu einem VLAN
    - Strikte Trennung des Datenverkehrs verschiedener LANs
- Realisierung mit Tags
  - Analyse des eingehenden Pakets auf VLAN-Zugehörigkeit (interne Tabelle)
  - Erster Switch fügt ein „Tag“ an das Paket an (Kennung für jedes VLAN)
  - Erweitertes Rahmenformat – IEEE 802.1q: Tag aus vier Feldern, Länge von 32 Bit, wird nach MAC-Adress-Feldern eingefügt. Protokoll ID: zwei Byte (Wert: 0x8100); Prioritätenfeld: drei Bit, Indikator des Canonical Formats (signalisiert Darstellungsformat der Adressfelder): ein Bit; VLAN-ID: zwölf Bit.
  - Weiterleitung des Datenpakets an den nächsten Switch
  - Letzter Switch entfernt das Tag und übergibt das Paket an das Endsystem

VLAN: Realisierung (II)

- *Schicht-2-VLANs*
  - Realisierung durch LAN-Switches
  - VLAN wird durch mehrere Ports festgelegt *(port-based VLAN)*
    - Pro Port können nur Stationen eines einzelnen VLAN angeschlossen sein
    - Mitgliedschaft in mehreren VLANs erfordert mehrere Netzwerkadapter
  - VLAN durch eine Liste von MAC-Adressen definiert *(MAC-based VLAN)*
    - einfacher Umzug einzelner Stationen möglich

  → **Aber:** Router zur Kommunikation zwischen VLANs notwendig

- *Schicht-3-VLANs*
  - Realisierung durch Layer-3-Switches
    - integrierte (aber nicht notwendigerweise vollständige) Schicht-3-Fähigkeit vorhanden!
  - VLAN wird durch Subnetz-Adresse festgelegt *(subnet-based VLAN)*
  - VLAN wird durch Netzwerkprotokoll festgelegt *(protocol-based VLAN)*

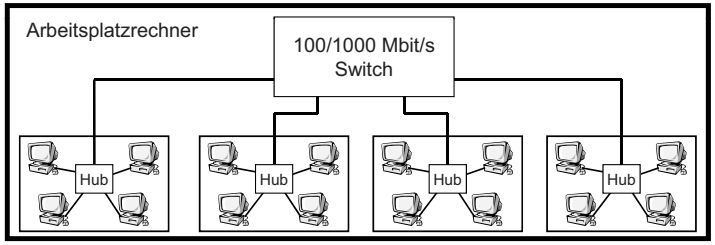
  → Kein zusätzlicher Router zur Kommunikation zwischen VLANs notwendig

VLAN: Realisierung (III)

- *Regelbasierte VLANs*
  - Beliebige Verknüpfung von Feldern der Schichten 2 und 3 zur Definition eines VLAN
  - Beispiel für die Definition regelbasierter VLANs
    - VLAN 1 = „Alle IP-Benutzer innerhalb eines bestimmten Subnetzes“
    - VLAN 2 = „Jeglicher Datenverkehr mit einem bestimmten Wert des Typ-Feldes im Ethernet-Header“
    - VLAN 3 = „Alle Rechner, deren Netzwerkkarte vom gleichen Hersteller stammt“
  - Vorteil:
    - besonders flexible Konfigurationsmöglichkeit
  - Nachteile:
    - Aufwendige Einrichtung der VLANs
    - Erhöhte Latenzzeit durch die Abarbeitung der einzelnen Regeln

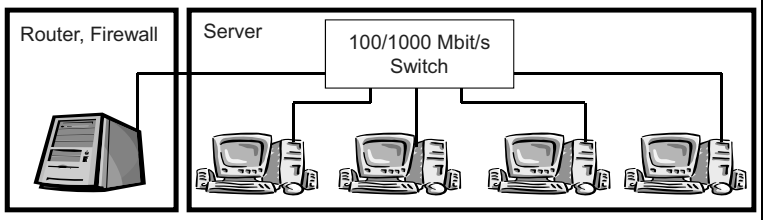
4.1.8. Leitbeispiel: Strukturierte Verkabelung

- Kopplung der einzelnen Zimmer eines Stockwerkes
  - Hub wenig leistungsfähig, da es zu häufigen Kollisionen kommen kann, wenn mehrere Rechner untereinander kommunizieren bzw. auf einen Server zugreifen → Switch
  - Switch arbeitet i.d.R. mit Fast-Ethernet, zum Teil mit Gigabit-Ethernet-Komponenten – abhängig von den benutzten Anwendung.
  - Verkabelung mit Twisted-Pair-Kabel (UTP-5)



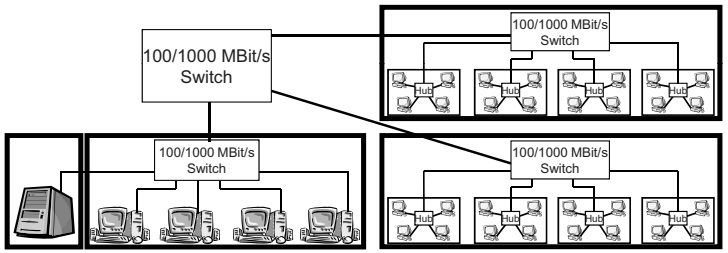
Leitbeispiel: Strukturierte Verkabelung

- Verkabelung der Server im Keller
  - Server werden i.d.R. stark frequentiert, daher kann ein Hub zu einem Engpass werden
  - Daher werden Server häufig direkt an Switches angeschlossen
  - Fast-Ethernet- bzw. Gigabit-Ethernet-Switch
  - Ggf. kann es sinnvoll sein, Glasfasern anstelle von UTP-5-Kabeln einzusetzen.
  - Alternativ kann statt des Routers einen Layer-3/4-Switch verwendet werden, um z.B. die über den Router kommende HTTP-Anfragen schnell an den richtigen Server zu leiten



Leitbeispiel: Strukturierte Verkabelung

- Verkabelung der Stockwerke
  - Kopplung einzelner Stockwerke über 100/1000 Mbit/s Switch
  - Alternative: Anschluss der Switches der (oberen) Stockwerke am Server-Switch (im Keller) sofern dort ausreichend Kapazitäten vorhanden sind
  - Einzelne Rechner können mittels VLANs zu einem log. Netzwerk gekoppelt werden



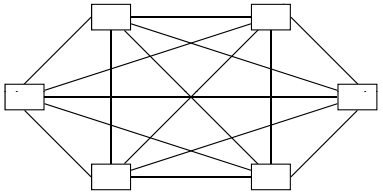
Inhalt des 2. Teils des Kapitels

4.2. Vermittlungsprinzipien für globale Netze
(Schicht 3)

- Leitungsvermittlung
- Nachrichten-/Speichervermittlung
- Paketvermittlung
  - virtuelle Verbindung
  - Datagrammvermittlung
- Router
- Routing-Verfahren (Wegewahl)

Grundanforderungen and Telekommunikationsnetze

- Teilnehmer sollen temporäre Kommunikationsbeziehungen mit anderen Teilnehmern auf Anforderung durchführen können.
- Ein für jeweils zwei Teilnehmer permanent vorgehaltener Kommunikationsweg führt zu einer nicht beherrschbaren Zahl von Kommunikationswegen.
- Bei voller Vermaschung von N-Teilnehmern: N * (N-1) Verbindungen

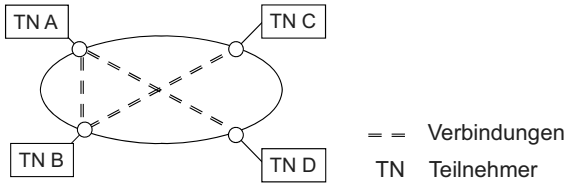


Vollständig vermaschtes Netz aus vielen überlagerten Punkt-zu-Punkt-Verbindungen

- In großen Netze (großen Anzahl Endsysteme, vielen Vermittlungssysteme) kommen aus Gründen mangelnder Leistungsfähigkeit die z.B. von Brücken bekannten Verfahren zum Einsatz, sondern spezielle Routing-Protokolle

Teilnehmersicht eines TK-Netzes

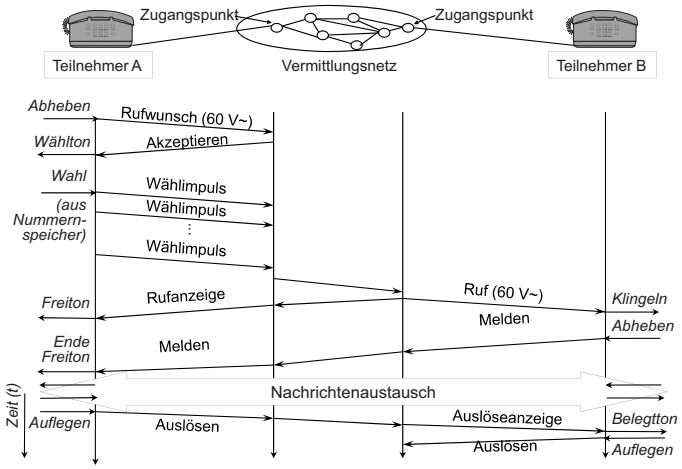
- Verbindungsorientiert - ein Teilnehmer unterhält zu jedem *Kommunikationspartner* eine Verbindung.
  - Hinweis: Der Begriff der Verbindung wird in der Telekommunikation in vielfältiger und sehr unterschiedlicher Weise genutzt.
  - Der spezielle Kontext definiert die notwendigen Details, beispielsweise die Schicht-3- oder physikalische Verbindung.
- Aufgaben, die mit einer Verbindung anfallen:
  - *Herstellen* von Verbindungen durch Vermittlungsdienste
  - Netzinterne *Überwachung* von *Verbindungseigenschaften* durch Monitore



Vermittlung und Signalisierung

- Vermittlung
  - Grundlegende Aufgabe
    - Bereitstellung eines temporären Kommunikationsweges durch das Netz auf Anforderung
    - Geschieht über eine Kette von Vermittlungsstellen zum Zwecke des Datenaustausches zwischen zwei oder mehreren Teilnehmern
  - Arten
    - Verbindungsorientierte Vermittlungstechniken
    - Verbindungslose Vermittlungstechniken
- Signalisierung
  - Oberbegriff für die vermittlungstechnische Kommunikation zwischen
    - zwischen Teilnehmereinrichtung (Endgerät) und Netz sowie
    - im Netzinneren, also zwischen Netzknoten

Wiederholung: Signalisierung im analogen Fernsprechnetz



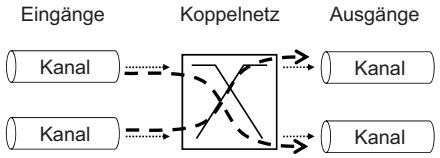
Verbindung

- Verbindung oder Netzwerkverbindung
  - Koppelt in verbindungsorientierter Art zwei Teilnehmer über eine oder mehrere Vermittlungsstellen (bei Wählnetzen temporär) zum Zwecke des Datenaustausches
  - Arten von Teilnehmern
    - Rufender Teilnehmer: der eine Schicht 3-Verbindung wünschende A-Teilnehmer (Anrufer)
    - Gerufener Teilnehmer: der eine Schicht 3-Verbindung erhaltende B-Teilnehmer (Angerufener)
  - Prinzip des Verbindungsaufbaues
    - Rufender Teilnehmer liefert neben dem Verbindungswunsch auch Identifizierungsdaten für den gerufenen Teilnehmer
      - Rufnummer, Kennung, Dienst-Nummer, Adresse, usw.
    - Zwischen Vermittlungsstellen werden zusätzlich vermittlungstechnische Daten ausgetauscht (Signalisierung)

4.2.1. Vermittlungsprinzip Leitungsvermittlung

- Synonyme Begriffe
  - Leitungsvermittlung
  - Durchschaltvermittlung
  - Circuit Switching
  - Line Switching
- Lange Zeit dominierende Vermittlungstechnik für öffentliche Fernmeldenetze
- Aufbau eines durchgehenden, nicht-speichernden Übertragungskanals (Leitung) zwischen den Teilnehmern
- Übertragungsverzögerungen sind auf physikalisch bedingte signaltechnische Laufzeiten beschränkt
- Bitfolgen werden reihenfolgetreu übertragen (signalübertragungstechnisch bedingt), damit wird die Absendereihenfolge beim Empfänger beibehalten (wire-like feature)

Leitungsvermittlung (Circuit Switching)



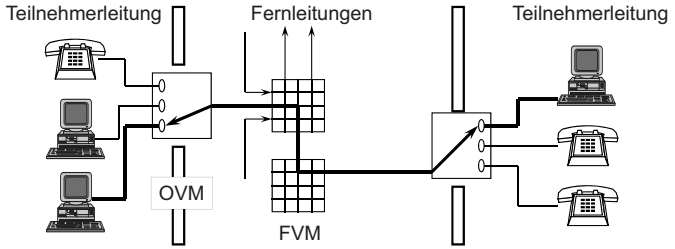
- Koppeltabelle:

Eingang	Ausgang
1	2
2	1
...	...

- Kopplung physikalischer Eingangs- und Ausgangskanäle
  - mit Kanal ist nicht allein eine Leitung gemeint, sondern auch Zeitschlitz bei TDM, Träger bei FDM,...
- Verbindungsaufbau erforderlich ➔ Einträge in der Koppeltabelle
- Beispiel: Leitungsvermitteltes Telefonnetz

Leitungsvermittlung im Telefonnetz

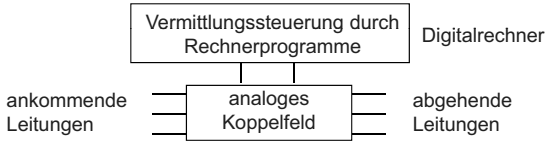
- Schema eines Durchschalttechnik-Vermittlungssystems



OVM Ortsvermittlung
FVM Fernvermittlung

Historie der Leitungsvermittlung

- Klassische elektromechanische Wähler (überholt)
  - Heb-Dreh-Wähler
  - Edelmetall-Motor-Drehwähler (EMD)
- Rechnergesteuerte Vermittlungen (Stored Program Control, SPC)
  - Beispiel: Rechnergesteuertes Vermittlungssystem mit analogem Koppelfeld

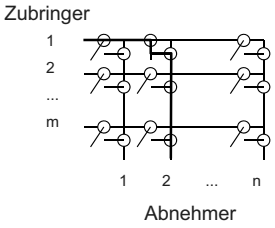


- Volldigitale Vermittlungssysteme
  - Digitalisierung der Vermittlungsstellen zusammen mit der digitalen Übertragungstechnik war Voraussetzung für das ISDN

Leitungsvermittlung bei Raummultiplex

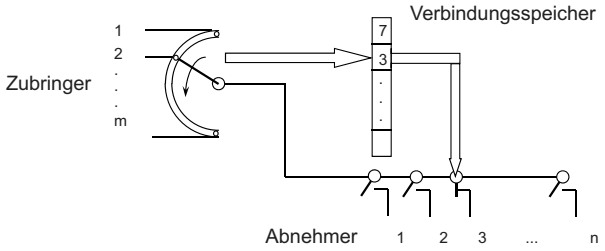
- Synonyme Begriffe
  - Raummultiplex
  - Raumvielfach
- Ein- und Ausgangskanäle sind physikalische Leitungen
- Durch Schließen eines Koppelpunktes wird die Verbindung hergestellt.
- Für analoge Übertragungen können ausschließlich Raummultiplex-Koppelanordnungen (Koppelfelder) verwendet werden.

- Beispiel
  - m Zubringerleitungen
  - n Abnehmerleitungen
  - Hier: Zubringer 1 mit Abnehmer 2 verbunden



Leitungsvermittlung bei Zeitmultiplex

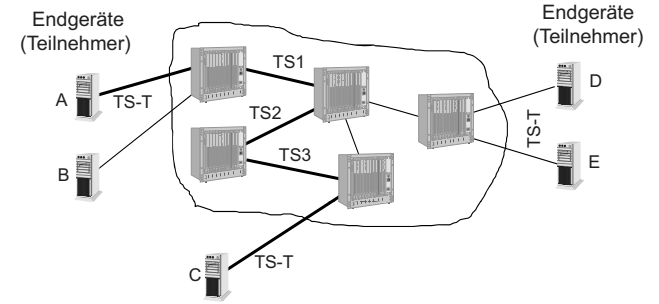
- Synonyme Begriffe
  - Zeitmultiplex
  - Time Division Multiplex
- Kanäle entsprechen festen Zeitschlitzen im Zeitmultiplexschema
- Der Zubringer ist mit dem Abnehmer nur für kurze Abtast-Zeitintervalle über den Koppelbus verbunden.
- Der Verbindungsspeicher hält die Zuordnung „Zubringerleitung – Abnehmer“ für die jeweiligen Abtastzeitpunkte.



4.2.2. Vermittlungsprinzip Nachrichtenvermittlung

- Synonyme Begriffe
  - Nachrichtenvermittlung
  - Speichervermittlung
  - Teilstreckenvermittlung
- Älteste Vermittlungstechnik in Form der Telegrammvermittlung
- Eigenschaften:
  - Speicher (Puffer) in den Rechnern der End- und Durchgangsvermittlungsstellen
  - Es treten keine Besetzt- oder Belegtfälle, sondern nur Wartefälle auf.
  - Zwischengespeicherte Vermittlungsdatenblöcke warten auf das Freiwerden der gewünschten Teilstrecke.
  - Verlust von Vermittlungsdatenblöcken durch Speicherüberlauf möglich
  - Wartende Vermittlungsdatenblöcke werden in Speichervermittlungsstellen z.B. nach Prioritäten oder „zufällig“ umgeordnet vermittelt (Reihenfolgevertauschung)
  - Es besteht i. A. keine feste Zeitbeziehung zwischen den einzelnen Speichervermittlungsdatenblöcken.
  - Geschwindigkeitsanpassung zwischen unterschiedlich leistungsfähigen Endgeräten ist möglich

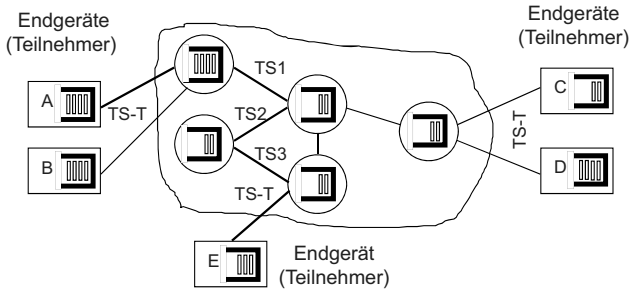
Speichervermittlungsnetz: Topologische Sicht



Vermittlungsrechner — TS-T — Teilstrecke-Teilnehmeranschluss

Endgerät (Teilnehmer) — TSx — interne Teilstrecke x zwischen Vermittlungsrechnern

Speichervermittlungsnetz: Innensicht



Vermittlungsrechner — TS-T — Teilstrecke-Teilnehmeranschluss

Puffer für SpV-Datenblock — TSx — interne Teilstrecke x zwischen Speichervermittlungsstellen

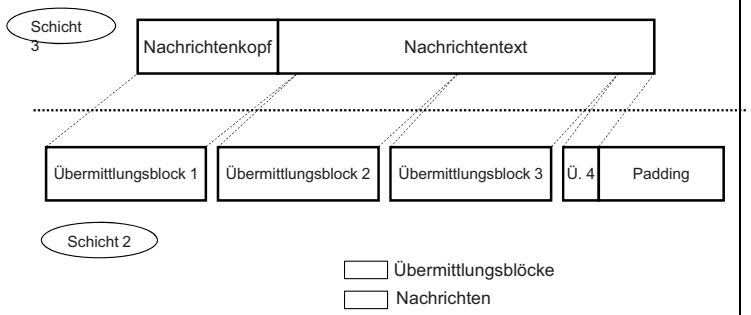
SpV-Datenblöcke

Der Nachrichtenbegriff

- Aufbau einer Nachricht aus
  - Nachrichtenkopf mit Steuerdaten der Schicht 3
    - Ziel- und Herkunftsadresse
    - Nachrichtentyp
    - Längenangabe
    - Priorität
    - Zeitangaben (z.B. Abgangszeit)
  - Nachrichtentext
- Eine Nachricht der Schicht 3 kann in einem oder mehreren Übermittlungsblöcken der Schicht 2 versendet werden.
- Zentrales Kennzeichen
  - Die Nachricht muss beim empfangenen Vermittlungsknoten vollständig eingehen und wieder montiert werden, ehe die Weiterleitung über die nächste Teilstrecke zur folgenden Speichervermittlungsstelle oder zum endgültigen Teilnehmer erfolgen kann.

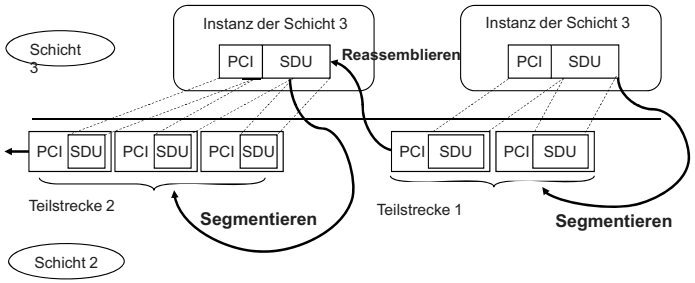
Segmentieren von Nachrichten

- Segmentieren von Nachrichten (z.B. wegen Blocklängenbegrenzung): Aufteilen von Nachrichten in mehrere Übermittlungsblöcke
- Resultierende zu kurze Übermittlungsblöcke werden eventuell mit Bits aufgefüllt, um Mindestlänge der Blöcke auf Schicht 2 zu erreichen (Padding)



Segmentieren und Reassemblieren einer Nachricht

- Segmentieren und Reassemblieren von Schicht-3-Nachrichten zur Anpassung an unterschiedliche maximale Blocklängen auf Schicht 2



PCI = Protocol Control Information **der jeweiligen Schicht**

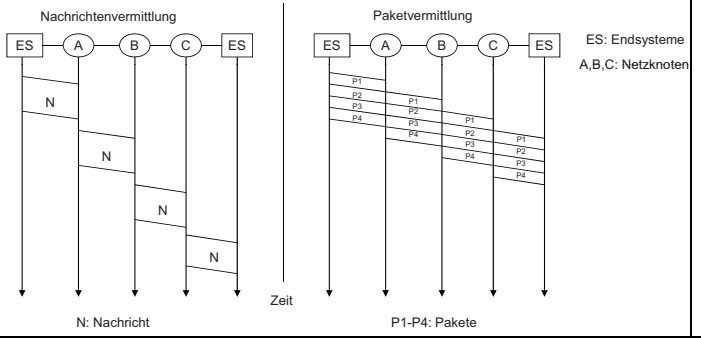
SDU = Service Data Unit **der jeweiligen Schicht**

4.2.3 Paketvermittlung

- Paket:
  - vergleichsweise kurzer Datenblock (z.B. einige 100 Oktette oder einige 1000 bit pro Paket)
  - Länge ist fest oder variabel mit vorgegebener Maximallänge
- Vermittlungsvorgang
  - Pakete zeitlich voneinander unabhängig
  - Dadurch zeitlich unabhängige Weitervermittlung je Paket
  - Kein Wiederzusammensetzen (Reassemblierung) im Durchgangsvermittlungssystem erforderlich
  - Folge:
    Zeitliche Überlappungen möglich, d.h. kürzere Durchlaufzeiten durch das Netz als bei der Nachrichtenvermittlung

Nachrichtenvermittlung vs. Paketvermittlung

- Hauptunterschied zwischen Nachrichten- und Paketvermittlung
  - Paketvermittlung: Inhaltlich zusammengehörende Transfereinheiten (Transport-Datenblöcke der Schicht 4) werden in Pakete nach den Vorschriften des Paketvermittlungsnetzes segmentiert
  - Nachrichtenvermittlung: Wiederherstellung der Transfereinheiten in jedem Vermittlungssystem aus den Segmenten



Paketvermittlung - Vermittlungsklassen

- Vermittlungsklassen für Paketvermittlungsnetze
  - Datagrammvermittlung (Datagram Switching)
    - Stellt eine verbindungslose Vermittlungstechnik dar
  - Virtuelle Verbindung (Virtual Circuit, VC)
    - Feste virtuelle Verbindung (Permanent Virtual Circuit, PVC)
    - Gewählte virtuelle Verbindung (Switched Virtual Circuit, SVC)

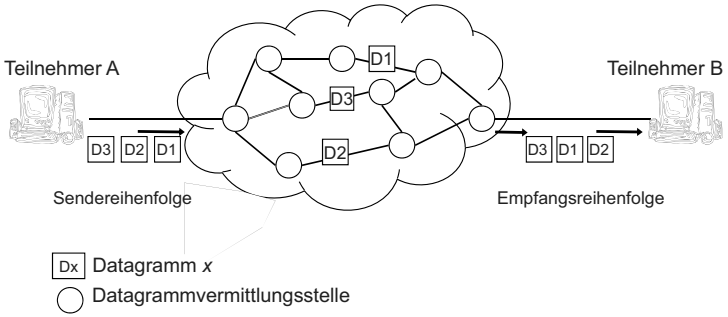
Datagrammvermittlung

- Eigenschaften:
  - Pakete bilden geschlossene Vermittlungseinheiten, u.a. mit **vollständigen Adressdaten**:
    - Herkunftsadresse
    - Zieladresse
  - **Kein Verbindungsaufbau** oder sonstige Abstimmungsprozedur über das Netzwerk vor Senden eines Datagrammes notwendig
  - Datagrammvermittlungsstelle ohne Verbindungskontext
- Vorteile
  - Verbindungsaufbau, -überwachung und -abbau entfallen
  - Bessere Nutzung der Netzkapazität möglich
- Nachteile:
  - Bei vermaschten Netzen mit alternativen Kommunikationswegen können Datagramme zwischen identischen Sendern und Empfängern unterschiedliche Wege zurücklegen

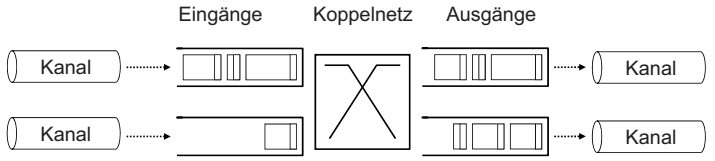
  ➔ **Überholvorgänge möglich**, keine reihenfolgerichtige Auslieferung gewährleistet

Datagrammvermittlung im vermaschten Netz

- Datagrammverkehr in einem Paketvermittlungsnetz
  - Jedes Datagramm mit eigenem Weg



Datagrammvermittlungsknoten



- Routing-Tabelle:

Ziel	Ausgang	Next-Hop
A	1	X
B	1	X
C	2	Y
D	2	Z
...	...	...

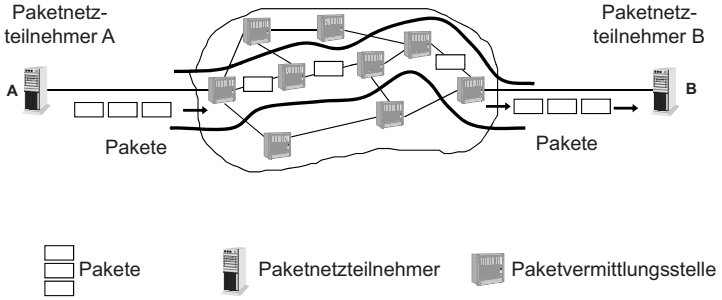
- Für jedes Datagramm wird individuelle Routing-Entscheidung getroffen
- Routing-Entscheidung anhand der Zieladresse im Paketkopf

Virtuelle Verbindungen

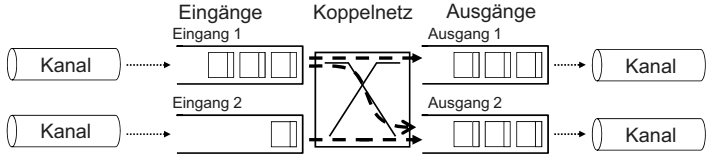
- Eigenschaften
  - Bidirektionaler fester Übertragungsweg (vollduplex)
  - Zwischen einem Paar logischer Anschlusspunkte in zwei kommunizierenden Knoten (auf Schicht 3) definiert
  - Kann mehrere Paketvermittlungsstellen umfassen
- Feste virtuelle Verbindung
  - Längerfristig eingerichteter Übertragungsweg durch das Netz
  - Ähnlich einer Standleitung in Durchschaltnetzen
- Gewählte virtuelle Verbindung
  - Verbindungsaufbauprozedur erforderlich
  - Aufbau erfolgt vor der Datenaustauschphase
  - Ähnlich wie bei der Leitungsvermittlung

Virtuelle Verbindungen - Verbindungsherstellung

- Schalten einer virtuellen Verbindung über eine feste Route durch die Anwendung des Prinzips der Speichervermittlung
- Alle Pakete verwenden einen identischen Weg im Netz
- Verbindungskontext in beteiligten Paketvermittlungsstellen



Vermittlungsknoten für virtuelle Verbindungen



- Verbindungskontext gespeichert in Weiterleitungstabellen

Eingang 1:

Eing.-VCI	Ausgang	Ausg.-VCI
A	1	**A**
B	2	**B**
…	…	…

Eingang 2:

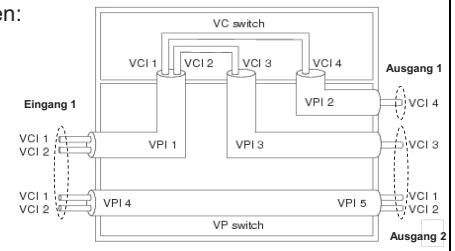
Eing.-VCI	Ausgang	Ausg.-VCI
A	2	**C**
…	…	

VCI geändert, um Kollision zu vermeiden

- Weiterleitungsentscheidung wird anhand eines VCI (Virtual Circuit Identifier) getroffen
- Virtuelle Verbindungen müssen vorher aufgebaut werden

Beispiel: Virtual Circuit Switching bei ATM

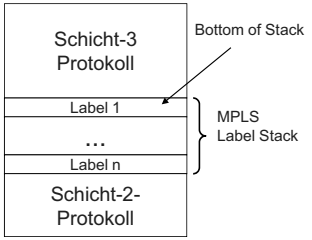
- Vermittlung auf zwei Ebenen:
  - Virtual Channel
    = einzelne Verbindung
  - Virtual Path
    = Aggregat mehrerer VCs



Eing.-VPI / VCI	Ausgang	Ausg.-VPI / VCI
1 / 1	1	2 / 4
1 / 1	2	3 / 3
4 / *	2	5 / *
…	…	…

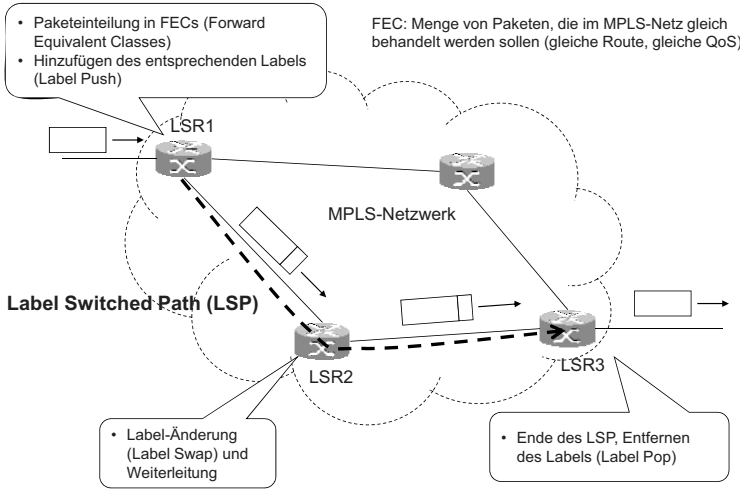
Beispiel: Multi-Protocol Label Switching (MPLS)

- Eigenschaften:
  - virtuelle Verbindung für beliebige Technologien und Protokolle
  - beliebig viele Labels im Stack
- Verarbeitung der Labels in LSRs (Label Switched Routers):
  - Hinzufügen und Entfernen von Labels
  - label-abhängige Weiterleitung
- Label Distribution Protocol (LDP):
  - Signalisierung und Label-Austausch zwischen LSRs



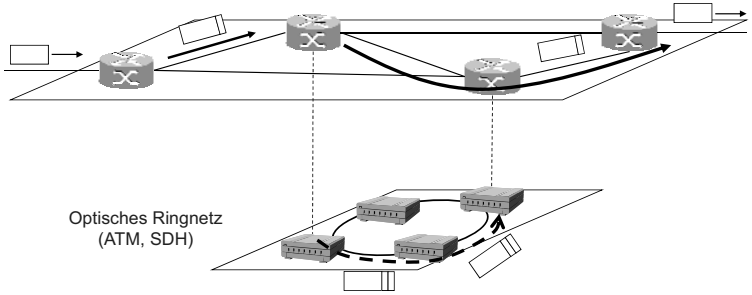

```
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ Label
|                Label                  | Exp |S|       TTL     | Stack
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ Entry

                    Label:  Label Value, 20 bits
                    Exp:    Experimental Use, 3 bits
                    S:      Bottom of Stack, 1 bit
                    TTL:    Time to Live, 8 bits
```


MPLS: Label Switched Path



MPLS-Hierarchie

- Label-Switching auf mehreren Schichten

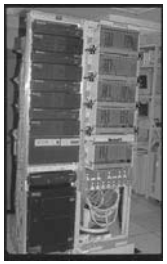


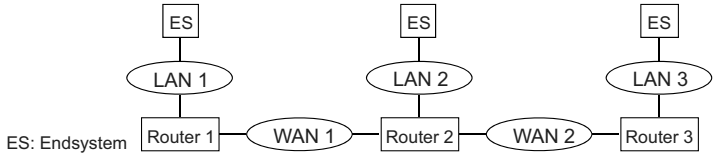
MPLS-Anwendungen

- Eine Motivation für virtuelle Verbindungen war:
  - größerer Durchsatz dank einfachen Forwarding-Entscheidungen in den Vermittlungsknoten/Routern
  - ➔ heute kein Argument mehr, weil Datagramm-Vermittlung sehr schnell und effizient in Hardware implementiert werden kann
- Anwendungsgebiete von MPLS:
  - Traffic Engineering: gezieltes Routing einzelner Verkehrsströme
  - Quality of Service: unterschiedliche Paketbehandlung und –weiterleitung je nach Dienstgüteanforderung
  - VPN (Virtual Privat Networks)
- GMPLS (Generalized MPLS):
  - Ausweitung auf Transportnetze mit physikalischer Leitungsvermittlung (v.a. optische Netze)
  - Label entspricht einem Zeitschlitz bei TDM oder einer Wellenlänge bei WDM (Wavelength Division Multiplex)

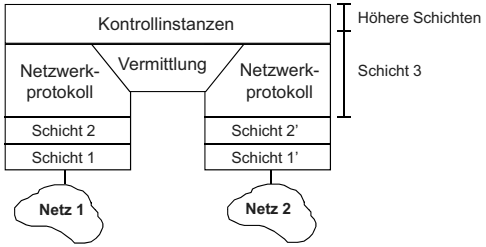
4.2.4. Router - Kopplung von Netzwerken auf Schicht 3

- Aufgaben:
  - Kommunikation entfernter Endsysteme über ein oder mehrere WANs
  - Wegewahl anhand weltweit eindeutiger, bevorzugt hierarchischer Netzwerkadressen (z.B. IP-Adressen)
  - Segmentieren/Reassemblieren von Schicht-3-Datenpaketen zur Anpassung an unterschiedliche maximale Paketgrößen auf Schicht 2
  - Sicherheitsmechanismen zur Regelung von Netzzugriffen abhängig von der Netzwerkadresse (Stichwort „Firewall“)
  - Automatische Begrenzung von Schicht-2-Broadcasts



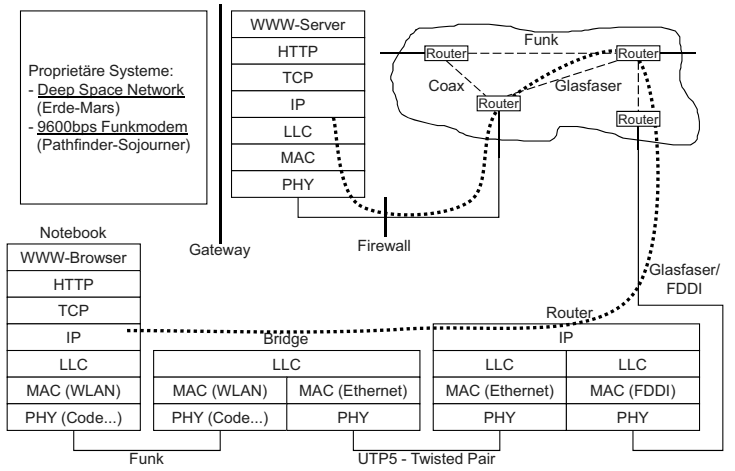


Architektur eines Routers



- **Wesentliche Merkmale:**
  - Für jedes Netzwerk eine eigene Schicht-1- und Schicht-2-Instanz
  - Netzwerkprotokoll ist in der Regel für alle Netzwerke gleich (z.B. IP-Router)
  - Wegwahl anhand der global eindeutigen Netzwerkadressen
  - Vermittlungskomponente verbindet die Netzwerkprotokollinstanzen; sie realisiert die Weiterleitungsfunktion
  - Kontrollinstanzen implementieren beispielsweise Routing-Protokolle, Protokolle zur Fehleranzeige und Managementprotokolle

Routing im Beispiel

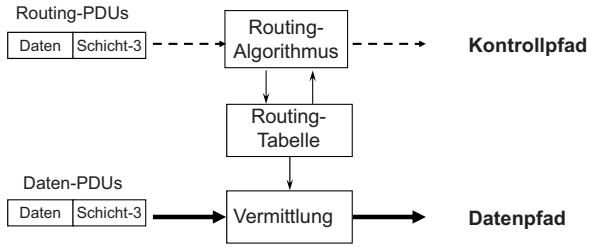


Wegewahl vom Notebook zur NASA

Aktueller Weg kann z.B. im Internet mit Hilfe von traceroute angezeigt werden

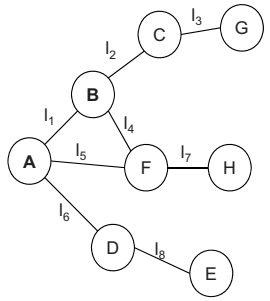
```
traceroute to www.NASA.GOV (128.183.243.3)

HOP NAME  (IP-address)                    TIME    probe 1 probe 2 probe 3
 0  mobile1.telematik.informatik.uni-karlsruhe.de (129.13.35.123)
 1  i70route1 (129.13.35.244)                      9 ms    9 ms   10 ms
 2  iracs1.ira.uka.de (129.13.1.1)                 2 ms    3 ms    2 ms
 3  Karlsruhe1.BelWue.DE (129.143.59.1)            2 ms    3 ms    2 ms
 4  Uni-Karlsruhe1.WiN-IP.DFN.DE (188.1.5.29)      3 ms    3 ms    3 ms
 5  ZR-Karlsruhe1.WiN-IP.DFN.DE (188.1.5.25)       5 ms    3 ms    2 ms
 6  ZR-Frankfurt1.WiN-IP.DFN.DE (188.1.144.37)     8 ms    7 ms    7 ms
 7  IR-Frankfurt1.WiN-IP.DFN.DE (188.1.144.97)     8 ms   11 ms    9 ms
 8  IR-Perryman1.WiN-IP.DFN.DE (188.1.144.86)    124 ms  126 ms  102 ms
 9  border3.Washington.mci.net (166.48.41.249)   121 ms  123 ms  124 ms
10  core4.Washington.mci.net (204.70.4.105)      123 ms  135 ms  121 ms
11  mae-east4.Washington.mci.net (204.70.1.18)   123 ms  122 ms  121 ms
12  mae-east.nsn.nasa.gov (192.41.177.125)       125 ms  126 ms  126 ms
13  rtr-wan1-ef.gsfc.nasa.gov (192.43.240.33)    127 ms  123 ms  121 ms
14  rtr-600.gsfc.nasa.gov (128.183.251.26)       147 ms  140 ms  130 ms
15  bolero.gsfc.nasa.gov (128.183.243.3)         127 ms  133 ms  129 ms
```

Router: Kontroll- und Datenpfad



- Datenpfad auf Netzwerkschicht
- Kontrollpfad darüber (Routing-PDUs sind in N-PDUs oder T-PDUs gekapselt)
- Gewinnung von Routinginformationen durch das **Routing-Protokoll**
- **Routing-Algorithmus** verwaltet die Routing-Tabelle bzw. Forwarding-Tabelle (Einfügen/Löschen/Ändern von Einträgen) auf der Basis der gewonnenen Routinginformation
- **Routing-Tabelle** bzw. Forwarding-Tabelle enthält Routinginformationen
- Wegewahl bei der Vermittlung wird anhand der Routing-Tabelle bzw. Forwarding-Tabelle durchgeführt

Prinzip einer Routingtabelle



A-H: Vermittlungsrechner (Router)
l_1-l_8: Abgehende/ankommende Teilstrecken (Links)

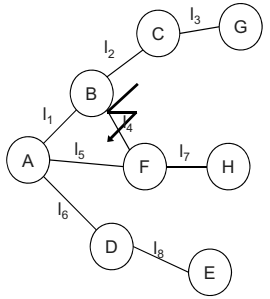
Routingtabelle in **Router B**

nach	über
A	- (l_1)
C	- (l_2)
D	A (l_1)
E	A (l_1)
F	- (l_4)
G	C (l_2)
H	F (l_4)

Routingtabelle in Router A

nach	über
B	
C	
D	
E	
F	
G	
H	

Prinzip einer Routingtabelle: Ausfall eines Links



A-H: Vermittlungsrechner (Router)
l_1-l_8: Abgehende/ankommende Links

Routingtabelle in Router B

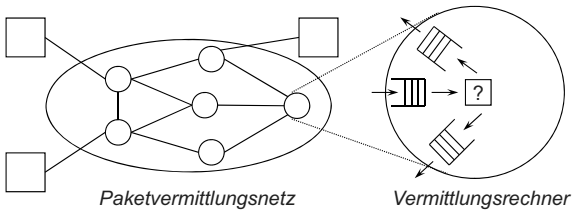
nach	über
A	- (l_1)
C	- (l_2)
D	A (l_1)
E	A (l_1)
F	A (l_1)
G	C (l_2)
H	A (l_1)

Routingtabelle in Router A

nach	über
B	
C	
D	
E	
F	
G	
H	

4.2.5. Routingverfahren

- Aufgaben
  - Fällen der Entscheidung, auf welcher Übertragungsleitung ein eingehendes Paket (Nachricht) weitergeleitet werden soll
- Ziele
  - Niedrige mittlere Paketverzögerung
  - Hoher Netzdurchsatz
- Ansatzpunkt
  - Übertragen eines Pakets von einem Quellrechner zu einem Zielrechner über einen Weg mit geringsten "Kosten".



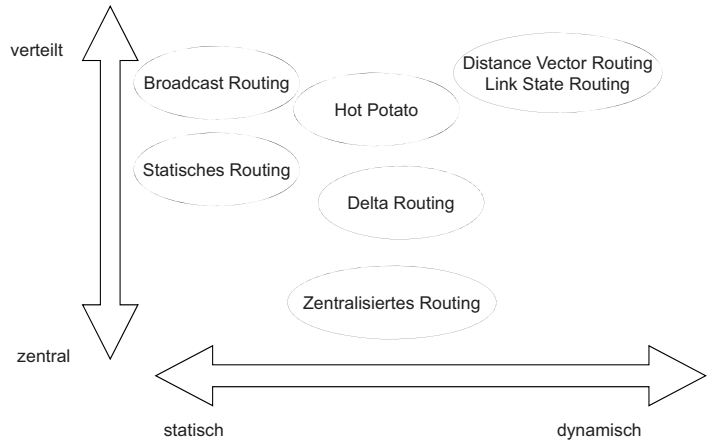
Routingverfahren - Routingnetzwerk und Wegewahl

- Definition des Routingnetzwerkes
  - Welche Knoten haben Vermittlungsfunktion?
  - Wie sind die Vermittlungsknoten verbunden: durch einfache oder mehrfache Verbindungen ?
- Weggenerierung
  - Welche Informationen sind zur Weggenerierung notwendig ?
  - Welche Parameter beeinflussen die Kostenfunktion ?
    - Feste Kosten für jede Verbindung (i.allg. umgekehrt proportional der Übertragungskapazität)
    - Anzahl der auf Übertragung wartenden Pakete
    - Fehlerrate
    - Paketverzögerungszeit auf einer Verbindung
    - Art des Verkehrs (Dialog, Batch)

Routingverfahren - Zentralisation und Dynamik

- Zentralisation
  - Wo ist der Routingalgorithmus lokalisiert?
    - Zentral (in einem Netzkontrollzentrum)
    - Dezentral (verteilt auf die Vermittlungsknoten)
- Wie dynamisch ist das Routingverfahren?
  - Nicht adaptiv: Die Routingtabellen in den Vermittlungsknoten bleiben über längere Zeit konstant, verglichen mit Verkehrsänderungen.
  - Adaptiv: Routing-Entscheidungen hängen vom Zustand des Netzes ab (Topologie, Lastverhältnisse).
- Zielkonflikt
  - Knoten haben veraltete oder unvollständige Informationen über den Zustand des Netzes.
  - Belastung durch Austausch von Routinginformationen

Routingverfahren im Überblick



Statisches Routing mit Lastverteilung

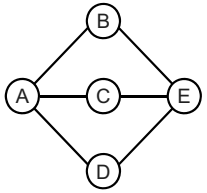
- Statisches Routing (Static Routing, Directory Routing) mit Lastverteilung
  - Nicht adaptiv, einfach, viel benutzt
    - Jeder Knoten unterhält eine Tabelle mit einer Zeile für jedes mögliche Zielnetz.
    - Eine Zeile enthält n Einträge, welche die beste, zweitbeste, etc. Übertragungsleitung für dieses Ziel, zusammen mit einer relativen Gewichtung, angeben.
    - Vor der Weiterleitung eines Pakets wird eine Zufallszahl gezogen und eine der Alternativen anhand der Gewichtung ausgewählt.

Statisches Routing mit Lastverteilung - Beispiel

- Beispiel
  - Ziehen einer Zufallszahl x, $0{,}00 \le x \le 1{,}00$
  - Falls $x < 0{,}6$ dann Weiterleiten nach B
  - Falls $0{,}6 \le x < 0{,}9$ dann Weiterleiten nach C
  - Sonst Weiterleiten nach D

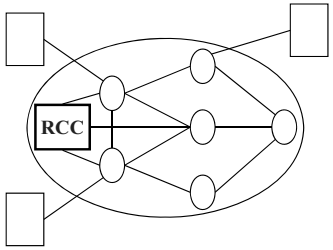
- Tabelle in Knoten A:

Ziel	1. Wahl		2. Wahl		3. Wahl	
	Kn	Gew	Kn	Gew	Kn	Gew
E	B	0,6	C	0,3	D	0,1
...						



Zentralisiertes Routing

- Adaptives Verfahren (Delta Routing)
  - Im Netz gibt es ein Routing Control Center (RCC).
  - Jeder Knoten sendet periodisch Zustandsinformationen an das RCC, z.B.:
    - Liste aller aktiven Nachbarn
    - Aktuelle Warteschlangenlängen
    - Umfang an Verkehr, der seit dem letzten Bericht abgewickelt wurde
  - Das RCC sammelt diese Zustandsinformationen und berechnet aufgrund dieser Kenntnis über das gesamte Netz die optimalen Wege zwischen allen Knoten (z.B. kürzeste Wege).
  - Jeder Knoten trifft seine Routing-Entscheidungen anhand der ihm zugewiesenen Routing-Tabelle.



Zentralisiertes Routing - Vor- und Nachteile

- Vorteile
  - Das RCC hat theoretisch die vollständige Übersicht und kann perfekte Entscheidungen treffen.
  - Knoten müssen keine aufwendigen Routing-Berechnungen durchführen.

- Nachteile
  - Für große Netze dauert die Berechnung u.U. sehr lange.
  - Ausfall des RCC lähmt das ganze Netz (Back-up Rechner).
  - Globale Inkonsistenzen möglich, da Knoten nahe dem RCC neue Routing-Tabellen ggf. früher erhalten als die weiter entfernten.
  - Belastung des RCC durch die zentrale Funktion

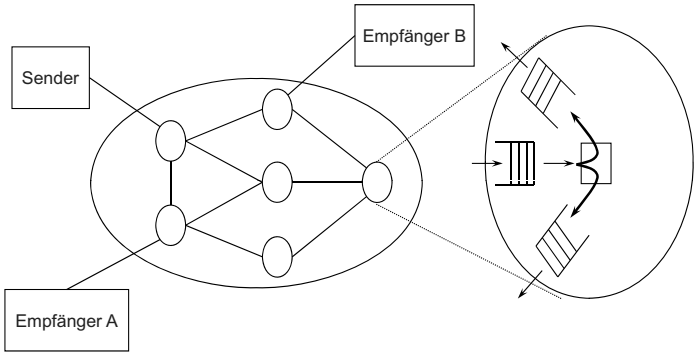
Isoliertes Routing: Überblick

- Isoliertes Routing (Isolated Routing)
  - Jeder Knoten entscheidet nur aufgrund der Information, die er selbst sammelt.
  - Kein Austausch von Routing-Informationen zwischen den Knoten.
  - Anpassung an Verkehrs- und Topologieänderungen kann damit nur mit Hilfe beschränkter Informationen erfolgen.
  - Unterschiedliche Verfahren
    - Broadcast Routing
    - Backward Learning
    - Hot Potato (Deflection Routing)
    - ...

Broadcast-Routing

- Broadcast-Routing
  - Senden eines Pakets an alle Knoten, z.B. für Umfragen nach bestimmten Betriebsmittel
- Varianten
  - Erstellen eines gesonderten Paketes für jeden Knoten
  - Fluten
- Arten
  - Multidestination Routing
  - Reverse-Path-Forwarding

Broadcast-Routing: Fluten



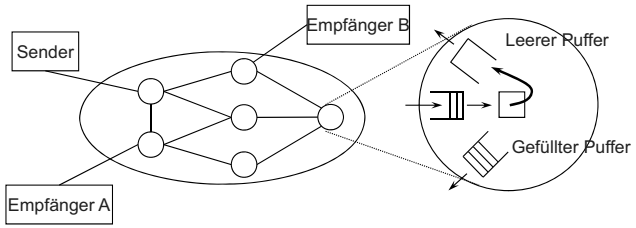
Broadcast-Routing: Fluten

- Fluten (Flooding)
  - Einfachstes Verfahren, nicht adaptiv
  - Jedes eingehende Paket wird auf jeder Übertragungsleitung weiter übertragen, außer auf derjenigen, auf der es eintraf.
- Maßnahmen zur Eindämmung der Flut
  - Erkennung von Duplikaten durch die Nummerierung der Pakete
  - Kontrolle der Lebensdauer eines Pakets durch Zählen der zurückgelegten Teilstrecken (hops). Ein hop-Zähler im Paket wird mit der minimalen Zahl von Teilstrecken (idealer Fall) bzw. einer (geschätzten) Zahl für die maximale Zahl von Teilstrecken zwischen Quelle und Ziel initialisiert. In jedem Knoten wird der Zähler um 1 dekrementiert. Falls der Zähler den Wert 0 erreicht, kann das Paket verworfen werden.
  - Varianten:
    - Selektives Fluten
      Weiterleitung nicht auf allen, sondern nur auf einigen Leitungen
    - Random Walk
      Zufällige Auswahl einer Leitung

Hot Potato (Deflection Routing)

- Jeder Knoten versucht, eingehende Pakete so schnell wie möglich weiterzuleiten.
- Die Übertragungsleitung mit der kürzesten Warteschlange wird für die Weiterleitung ausgewählt.
- Variante: Kombination mit statischem Routing
  - Auswahl der besten Übertragungsleitung nach statischem Verfahren, solange deren Warteschlangenlänge unter einer bestimmten Schwelle bleibt
  - Auswahl der Übertragungsleitung mit kürzester Warteschlange, falls deren statisches Gewicht nicht zu niedrig ist
- Alternative Bedeutung von "Hot Potato Routing": Jeder Knoten einer administrativen Domäne versucht, ein Paket, das an eine andere administrative Domäne weitergeleitet werden soll, auf dem schnellsten Weg an diese andere Domäne weiterzugeben. (Führt zu asymetrischen Pfaden mit unterschiedlichem Hin- und Rückweg.)

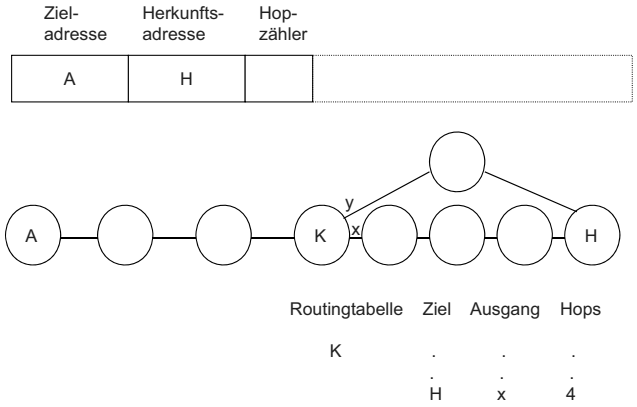
Hot Potato (Deflection Routing)



Backward Learning

- Daten im Paket
  - Identifikation des Quellknotens
  - Ein Zähler, der mit jeder zurückgelegten Teilstrecke (hop) um 1 erhöht wird.
- Beispiel
  - Falls z.B. bei einem Knoten K auf der Übertragungsleitung k ein Paket mit Zähler = 4 vom Ursprungsknoten H eintrifft, so weiß der Knoten K, dass er den Knoten H über die Leitung k in 4 hops erreichen kann.
  - Falls Knoten Ks bisheriger bekannter optimaler Weg zu Knoten H (über eine andere Leitung) mehr als 4 hops beträgt, so aktualisiert Knoten K seine Routingtabelle mit dem neuen und jetzt besseren Weg.

Backward Learning



Backward Learning - Nachteile und Probleme

- Nachteile
  - Nur Änderungen zum Besseren werden zur Kenntnis genommen
  - Ausfälle oder Überlastung von Übertragungsleitungen werden nicht weitergemeldet.
- Folge
  - Knoten müssen periodisch alle Informationen vergessen und wieder initial aufsetzen.
- Problem
  - Während der Lernperiode ist das Routing nicht optimal.
  - Bei häufigem Neubeginn nehmen viele Pakete Wege unbekannter Qualität.
  - Bei seltenem Neubeginn ergibt sich ein schlechtes Anpassungsverfahren.

Verteiltes adaptives Routing

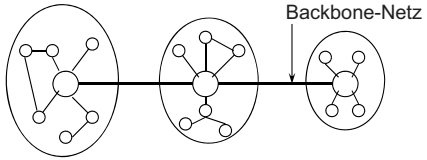
- Jeder Knoten tauscht periodisch Routing-Informationen mit jedem seiner Nachbarn aus.
- Typischerweise unterhält jeder Knoten eine Routing-Tabelle, die für jeden anderen Knoten im Netz, bzw. für jedes Teilnetz (oder Zusammenfassungen von Teilnetzen) einen Eintrag enthält. Hierzu zählen
  - Bevorzugte Übertragungsleitung für diesen Knoten / dieses Teilnetz
  - Schätzung über Zeit oder Entfernung zu diesem Knoten / diesem Teilnetz:
    - Anzahl hops,
    - Geschätzte Verzögerung in Millisekunden sowie
    - Geschätzte totale Anzahl von Paketen, die entlang des Weges warten.

Verteiltes adaptives Routing: Varianten

- Die Schätzungen werden gewonnen aus
  - Zeit oder Entfernung zu den Nachbarn (z.B. aus speziellen Echopaketen mit Zeitstempeln)
  - Schätzungen der Nachbarn
- Varianten
  - Synchroner Austausch von Routing-Informationen in bestimmten Aktualisierungsintervallen
  - Asynchroner Austausch bei signifikanten Änderungen

Hierarchisches Routing

- Grundlage
  - Aufteilung großer Netze in Regionen
  - Die Knoten in einer Region haben nur Routing-Informationen über ihre eigene Region.
  - In jeder Region gibt es zumindest einen ausgezeichneten Knoten, der als Schnittstelle zu anderen Regionen dient.
- In sehr großen Netzen sind weitere Hierarchieebenen möglich
  - Regionen, Cluster, Zonen, Gruppen, ...



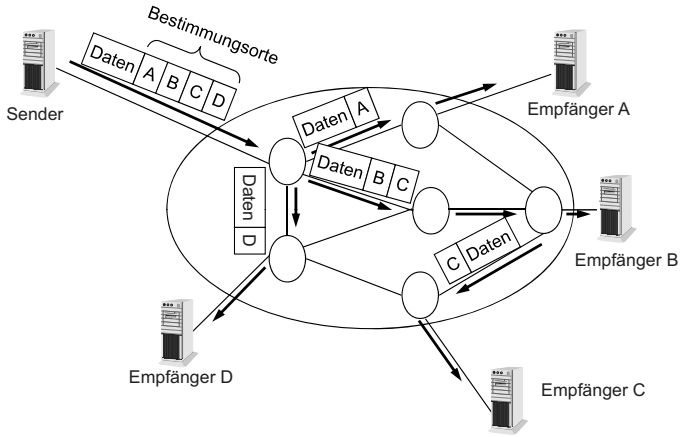
Routing in der Gruppenkommunikation (Multicast)

- Jedes Paket enthält entweder
  - 1. eine Liste der Bestimmungsorte, oder
  - 2. eine Bitleiste, welche die Bestimmungsorte angibt, oder
  - 3. alle Empfänger werden anhand einer Kennung (Gruppenadresse) identifiziert (Bsp.: IP-Multicast)
- Jeder Knoten bestimmt aus den in einem Paket enthaltenen Bestimmungsorten die Menge der Ausgabeleitungen.
- Für jede zu benutzende Ausgabeleitung wird eine Kopie des Pakets erzeugt.
  - Diese Kopie enthält nur die Bestimmungsorte, die über diese Leitung erreicht werden sollen (bei Fall 1 bzw. 2)

Routing in der Gruppenkommunikation

- Die Pakete müssen an mehreren Ausgängen ausgegeben werden
- Problem der Gruppenadressierung:
  - Ansatz 1 - Explizite Addressierung:
    Jeder Empfänger wird im Paket explizit angegeben
    - Liste der Bestimmungsorte, bzw.
    - Bitleiste, welche die Bestimmungsorte angibt
    - Dem Sender (und dem Router) müssen alle Empfänger bekannt sein
    - Weg zu den Empfängern wird durch Unicast-Routing-Tabelle ermittelt
  - Ansatz 2 – Gruppenaddressierung:
    Alle Empfänger werden anhand einer Kennung (Gruppenadresse) identifiziert (Bsp.: IP-Multicast)
    - Dem Sender (und dem Router) sind die Empfänger i.Allg. nicht bekannt
    - Einsatz eines separaten Routing-Protokolls (eigene Multicast-Routing-Tabelle)

Multicast-Routing mit Empfängerliste



Multicast-Routing: Reverse-Path-Forwarding

- Algorithmus
  - ´Routing(i)´ bestimmt die Ausgabeleitung für Paket mit Ziel i
  - ´Incoming Link´ sei die Ankunftsleitung eines Pakets
  - ´Source Node´ sei der Ursprungsknoten eines Pakets

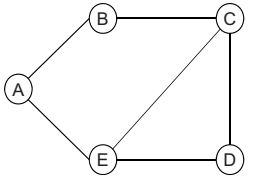
```
if Destination Node = {All Nodes or Group Address}
then    if Incoming Link = Routing(Source Node)
/*Paket kommt auf dem vermutlich kürzesten Weg vom Source Node*/
  then  Outgoing Link Set := All Links - {Incoming Link} // Weiterleiten
  else  Outgoing Link set := Ø  // Verwerfen des Pakets
else    Outgoing Link := Routing(Destination Node)
```


Beispiel 1: Distance Vector Routing

- **Verteiltes, adaptives Routing**
- Als RIP (Routing Information Protocol) im Internet (in kleinen Netzen) eingesetzt
- Jeder Router speichert eine Tabelle mit der besten Entfernung (z.B. Anzahl Hops, Verzögerung in ms) zu jedem Ziel und dem dazugehörigen Ausgang bzw. nächstem Hop
- Benachbarte Router teilen sich in regelmäßigen Abständen den Inhalt ihrer Routing-Tabelle mit und aktualisieren damit ihre eigene Tabelle (Bellman-Ford-Algorithmus)

Routing-Tabelle von A:

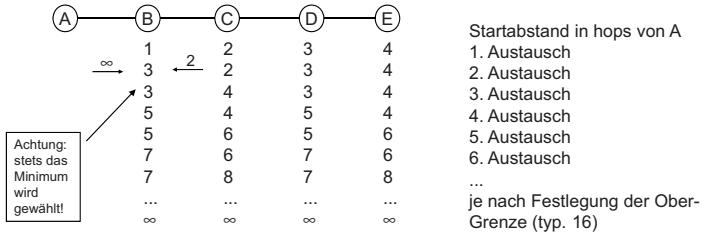
Ziel	Nächster Hop	Entfer-nung
B	-	1
C	B/E[1)]	2
D	E	2
E	-	1



[1)] je nach dem, welche Route zuerst bekannt ist

Beispiel 1: Distance Vector Routing (Fortsetzung)

- Problem bei Distance Vector Routing:
  - Für Netze mit vielen Routern: langsame Konvergenz zu einem konsistenten Zustand wegen „count-to-infinity“ –Problematik
- Beispielszenario: Router A bis E verbunden, plötzlich fällt A aus



A	B	C	D	E	Startabstand in hops von A
	1	2	3	4	
∞ →	3	2 ←	3	4	1. Austausch
	3	4	3	4	2. Austausch
	5	4	5	4	3. Austausch
	5	6	5	6	4. Austausch
	7	6	7	6	5. Austausch
	7	8	7	8	6. Austausch
	...	...	...	...	...
	∞	∞	∞	∞	je nach Festlegung der Ober-Grenze (typ. 16)

- Diverse Lösungsansätze:
  - Split Horizon, Poisoned Reverse, Triggered Updates, Path Vector

Distance Vector Routing

Lösungsansätze für schnellere Konvergenz

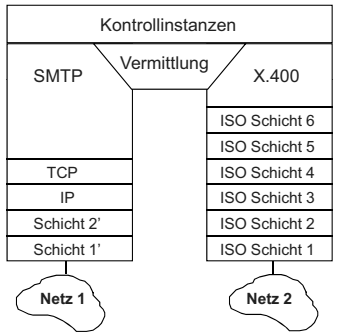
- Split Horizon
  - Eine Pfadinformation darf nicht über das selbe Interface veröffentlicht werden, worüber sie empfangen wurde (simple split horizon).
- Poisoned Reverse
  - Die Pfadinformation wird zwar an das Interface zurückgeschickt, über die sie empfangen wurde, aber die Entfernung wird auf unendlich gesetzt.
  - Beispiel zu voriger Folie: C darf Route nach A nicht an B weitergeben (bzw. nur mit Entfernung unendlich), weil diese Route von B annociert wurde
  - funktioniert nicht für größere Schleifen
- Triggered Updates
  - Ändert sich eine Metrik, wird die Pfadinformation sofort propagiert und nicht erst nach einem Timeout.
- Path Vector
  - Es wird nicht nur der nächste Hop, sondern der ganze Pfad bis zu Ziel weitergegeben bzw. gespeichert
  - Kommt bei BGP zum Einsatz

Beispiel 2: Link State Routing

- **Verteiltes, adaptives Routing**
- Als OSPF (Open Shortest Path First) und IS-IS (Intermediate System - Intermediate System) im Internet eingesetzt
- Algorithmus:
  - Entdecken neuer Nachbarn über HELLO-Pakete
  - Bestimmung der Link-Kosten:
    - entweder durch Konfiguration vorgegeben, oder ermittelt über Messung der Verzögerung zu jedem Nachbarn (ECHO-Paket misst Umlaufzeit)
  - Erstellen eines „Link-State“-Pakets mit allen gelernten Daten
    - beinhaltet ID des Senders (Routers), Liste der Nachbarn mit Verzögerung, Alter
    - periodische oder ereignisgesteuerte (z.B. neuer Nachbar, Ausfall) Erzeugung
  - Aussenden dieses Paketes an alle Nachbarn
    - Nachbarn geben Link-States wiederum an ihre Nachbarn weiter usw. → prinzipiell Fluten an alle Router, aber mit Verfeinerungen: Vernichten von Duplikaten, Zerstören der Information nach gewissem Alter etc.
    - Jeder Router kennt am Ende die Topologie des Netzwerkes
  - Berechnung des kürzesten Pfades zu allen anderen Routern (z.B. Dijkstra)
    - sehr rechenaufwendig, Optimierungen existieren

Gateways

- Kopplung von Netzwerken auf einer höheren Schicht (>= 4)
- falls Quell- und Zielrechner verschiedene höhere (d.h. für Anwendung sichtbare) Protokolle verwenden
- Typisches Beispiel: Gateway zwischen verschiedenen Mail-Systemen



SMTP: Simple Mail Transport Protocol

Literatur zu Routingverfahren (Auswahl)

- **Huitema, C.:**
  Routing in the Internet
  2. Auflage, Prentice Hall, Inc., New Jersey, 2000,
  ISBN 0-13-022647-5
  Gute Übersicht über Routingverfahren im Internet
- **Tanenbaum, A. S.:**
  Computer Networks
  4. Auflage, Prentice Hall, Inc., New Jersey, 2002,
  ISBN 0-13-394248-1
  Insbesondere Kap. 5
- **Perlman, R.:**
  Interconnections Second Edition: Bridges, Routers, Switches, and Internetworking Protocols
  Addison-Wesley, Reading, Mass., 1999,
  ISBN 0-201-63448-1

Lehrstuhl für Netzarchitekturen und Netzdienste
Institut für Informatik – Technische Universität München
Prof. Dr.-Ing. Georg Carle

Grundlagen: Rechnernetze und Verteilte Systeme

Kapitel 5: Internet-Protokolle

Internet-Protokolle der Netzwerkschicht

Prof. Dr.-Ing. Georg Carle
Lehrstuhl für Netzarchitekturen und Netzdienste
Technische Universität München
carle@net.in.tum.de
http://www.net.in.tum.de

Übersicht

1. Einführung und Motivation
   - Bedeutung, Beispiele
2. Begriffswelt und Standards
   - Dienst, Protokoll, Standardisierung
3. Direktverbindungsnetze
   - Fehlererkennung, Protokolle
   - Ethernet
4. Vermittlung
   - Vermittlungsprinzipien
   - Wegwahlverfahren
5. **Internet-Protokolle**
   - **IP, ARP, DHCP, ICMP**
   - **Routing-Protokolle**
6. Transportprotokolle
   - UDP, TCP
7. Verkehrssteuerung
   - Kriterien, Mechanismen
   - Verkehrssteuerung im Internet
8. Anwendungsorientierte Protokolle und Mechanismen
   - Netzmanagement
   - DNS, SMTP, HTTP
9. Verteilte Systeme
   - Middleware
   - RPC, RMI
   - Web Services
10. Netzsicherheit
    - Kryptographische Mechanismen und Dienste
    - Protokolle mit sicheren Diensten: IPSec etc.
    - Firewalls, Intrusion Detection
11. Nachrichtentechnik
    - Daten, Signal, Medien, Physik
12. Bitübertragungsschicht
    - Codierung
    - Modems

Ziele

- In diesem Kapitel wollen wir vermitteln
  - TCP/IP-Protokollfamilie
  - Funktionalität von IP-Adressen
  - Zusammenspiel von Protokollen
  - Hierarchie von Routing
  - Funktionalität von IPv6
  - Mobilität im Internet

Kapitelgliederung

5.1. Internet-Architektur: Merkmale

- Grundlegende Entwurfsprinzipien:
  - Keine Zustandsinformation in den Zwischensystemen halten
    ⇨ bei Ausfall keine Resynchronisation notwendig
  - Datenstrom-spezifische Information wird in den Endsystemen gespeichert
    ⇨ Bestandteil des Ende-zu-Ende-Prinzips
  - Trennung der Weiterleitung der Pakete („Forwarding“)
    vom „Routing“ = Erstellung der Weiterleitungstabellen

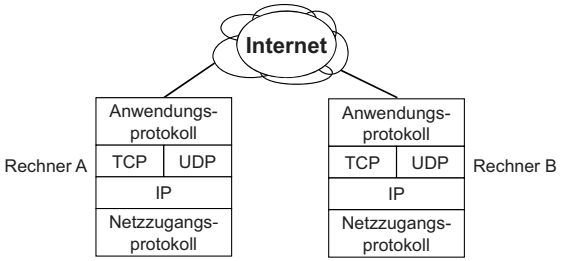
- IP-Basiskommunikationsdienst:
  - verbindungslos, unzuverlässig
  - abschnittsweise Weiterleitung, speichervermittelt
  - „Best Effort“-Diensterbringung: so gut wie möglich mit den momentan vorhandenen Ressourcen

Wiederholung: Die Internet-Protokollhierarchie

Application Layer	Anwendungsspezifische Funktionen zusammengefasst in Anwendungsprotokollen
Transport Layer	Ende-zu-Ende-Datenübertragung zwischen zwei Rechnern
Network Layer	Wegewahl im Netz auch “Internet Layer” genannt
Data Link Layer	Schnittstelle zum physikalischen Medium “Netzwerkkartentreiber”
Physical Layer	

- Gegenüber ISO/OSI wurden die drei anwendungsorientierten Schichten zu einer einzigen Schicht zusammengefasst.

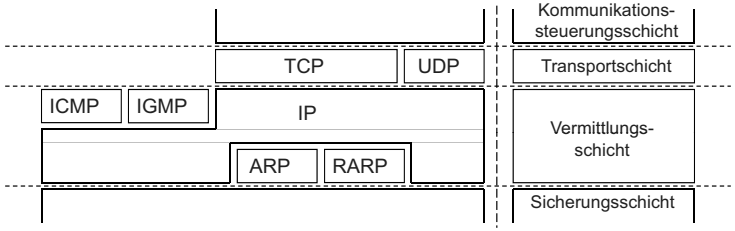
5.1.1. Die Internet-Protokollfamilie



- **TCP** (Transmission Control Protocol):
  - Zuverlässiges, verbindungsorientiertes Transportprotokoll über unzuverlässigem IP (Internet Protocol).
- **UDP** (User Datagram Protocol):
  - Verbindungsloses Transportprotokoll, bietet Anwendungsschnittstelle zu IP und Multiplexdienst.
- Beipiele für **Anwendungsprotokolle**:
  - HTTP: HyperText Transfer Protocol (im WWW benutzt)
  - FTP: File Transfer Protocol
  - Telnet: Protokoll für virtuelle Terminals

5.1.2. Die TCP/IP-Protokollfamilie – Überblick

- Die Bezeichnung TCP/IP wird häufig als Synonym für die gesamte Protokollfamilie verwendet
- Einordnung der Internetprotokolle in das ISO/OSI-Referenzmodell:



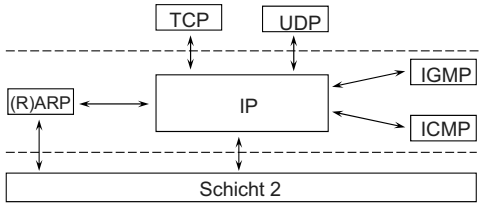
- Obwohl die IP-Steuerungsprotokolle ICMP und IGMP den IP-Dienst nutzen, werden sie dennoch der Vermittlungsschicht zugeordnet
- In den anwendungsbezogenen Schichten 5-7 werden im Internet Protokolle wie z.B. FTP, TELNET oder SMTP eingesetzt (Schichten 5-7 im Internet zusammengefasst zur Anwendungsschicht)

Die TCP/IP-Protokollfamilie: Protokollaufgaben

TCP	(Transmission Control Protocol): Stellt verbindungsorientierten, gesicherten Transportdienst bereit
UDP	(User Datagram Protocol): Stellt verbindungslosen, ungesicherten Transportdienst bereit
IP	(Internet Protocol): Sorgt für Wegewahl und ungesicherte Übertragung von Datagrammen
ICMP	(Internet Control Message Protocol): Unterstützt den Austausch von Steuerungsinformationen innerhalb der Vermittlungsschicht
IGMP	(Internet Group Management Protocol): Unterstützt die Verwaltung von Kommunikationsgruppen
ARP	(Address Resolution Protocol): Unterstützt die Zuordnung von IP-Adressen zu den entsprechenden Adressen der Sicherungsschicht
RARP	(Reverse Address Resolution Protocol): Stellt die Umkehrfunktion von ARP zur Verfügung

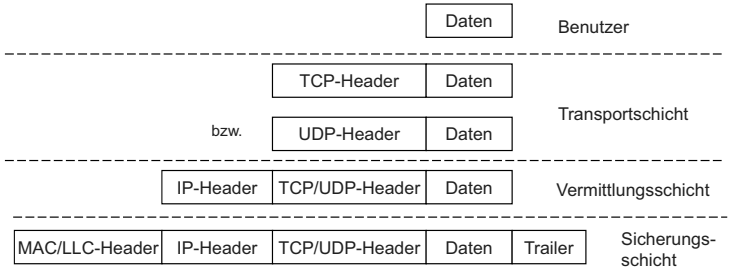
5.1.3. Zusammenspiel: IP-Instanz und angrenzende Instanzen

- Schicht-4-Instanz (TCP- bzw. UDP) übergibt die Daten zusammen mit der IP-Adresse des Empfängers zur Übertragung an die IP-Instanz
- IP-Instanz beauftragt ARP-Instanz mit Ermittlung der entsprechenden Schicht-2-Adresse
- IP-Instanz übergibt PDUs zusammen mit der ermittelten Schicht-2-Adresse an die Instanz der Sicherungsschicht
- IP-Instanz reicht empfangene Daten an TCP- bzw. UDP-Instanzen weiter
- Probleme während der Übermittlung können den Partnerinstanzen über ICMP mitgeteilt werden
- Informationen über Gruppenzugehörigkeiten werden mittels IGMP (Internet Group Management Protocol) im Netz verbreitet



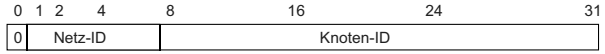
Zusammenspiel der Protokollinstanzen: die PDUs

- IP leitet Datenpakete durch das Netzwerk zum Empfänger
- TCP/UDP fügen Prozessadressierung (Ports) zu IP hinzu
- TCP sichert darüberhinaus die Datenübertragung
- Protokolldateneinheiten (PDUs) werden gekapselt



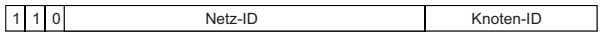
5.1.4. IP-Adressen / Adressklassen (historisch)

- **Class A** für Netze mit bis zu 16 Mio. Knoten (0.0.0.0 - 127.255.255.255):



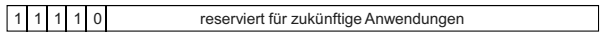
- **Class B** für Netze mit bis zu 65.536 Knoten (128.0.0.0 - 191.255.255.255):

- **Class C** für Netze mit bis zu 256 Knoten (192.0.0.0 - 223.255.255.255):



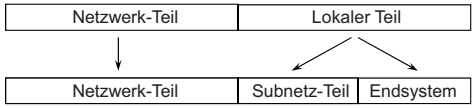
- **Class D** für Gruppenkommunikation (Multicast) (224.0.0.0 - 239.255.255.255):

- **Class E**, noch reserviert für zukünftige Anwendungen (240.0.0.0 - 247.255.255.255):



IP-Subnetz-Adressen

- **IP-Adresse** (hier Class B):



- **Subnetzmasken** kennzeichnen den Bereich der IP-Adresse, der das Netzwerk und das Subnetzwerk beschreibt. Dieser Bereich wird dabei durch Einsen („1“) in der binären Form der Subnetzmaske festgestellt. Subnetzmasken haben keine Internet-weite Gültigkeit.
- **Beispiel:**

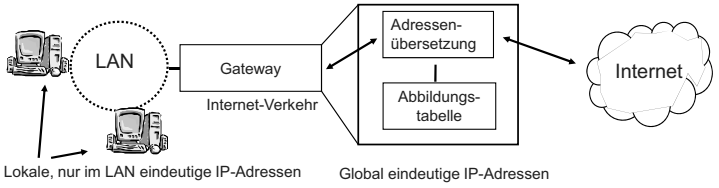
IP-Adresse:	129.	13.	3.	64
	1000 0001	0000 1101	0000 0011	0100 0000
Subnetzmaske:	255.	255.	255.	0
	1111 1111	1111 1111	1111 1111	0000 0000
Netzwerk:	129.	13.		
Subnetz:			3.	
Endsystem:				64

CIDR: Classless Inter-Domain Routing

- Bisher: 3 Adressklassen für Unicast (A, B und C)
  - schlechte Ausnutzung durch ungenutze Adressen („Verschnitt“)
  - ⇨ Granularität der Netze häufig nicht passend; Anzahl der Netze zu klein
- Beispiel:
  - Kleinbetrieb, der 100 IP-Adressen braucht, beantragt Class-C-Netz
  - 254 Adressen könnten vergeben werden, damit **154 ungenutzte Adressen**
- **CIDR:** Ersetzen der festen Klassen durch **Netzwerk-Präfixe** variabler Länge
  - Bsp.: 129.24.12.0/14 → Die ersten 14 Bits der IP-Adresse werden für die Netzwerk-Identifikation verwendet
- Einsatz in Verbindung mit hierarchischem Routing:
  - Backbone-Router, z.B. an Transatlantik-Link, betrachtet nur z.B. die ersten 13 Bit; dadurch kleine Routing-Tabellen, wenig Rechenaufwand
  - Router eines angeschlossenen Providers z.B. die ersten 15 Bit
  - Router in einem Firmennetz mit 128 Hosts betrachtet die ersten 25 Bit

5.1.5. Network Adress Translation/Translator (NAT)

- Nur Rechner, die gerade mit der Außenwelt kommunizieren, benötigen eine global eindeutige Adresse
  - Diese global eindeutige Adresse kann temporär vergeben werden
- Gateway/Router nimmt transparente Umsetzung zwischen Adressen/Adressbereichen vor
  - Speicherung in Abbildungstabelle
  - keine Änderungen an Endgeräten erforderlich
  - ⇨ Identitäten der Hosts werden verborgen



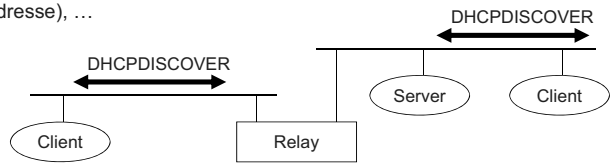
NAT - weitere Eigenschaften

- **Abbildungsarten:**
  - Statisch: lokale Adresse ⇔ globale Adresse
    - z.B. 192.168.39.100 ⇔ 129.13.41.100
  - Dynamisch: lokale Adresse ⇔ globale Adresse aus Adresspool
    - Erzeugen eines „Simple Entry“ in Abbildungstabelle (IP_{lokal} ⇔ IP_{global})
  - Overloading:
    - Abbildung aller lokalen Adressen auf eine einzige globale Adresse
    - zusätzliches Unterscheidungskriterium: Portnummern
    - Erzeugen eines „Extended Entry“ in Abbildungstabelle
      - Protokoll, ($Port_{lokal}$, IP_Ad_{lokal}) ⇔ (IP_Ad_{global}, $Port_{global}$)

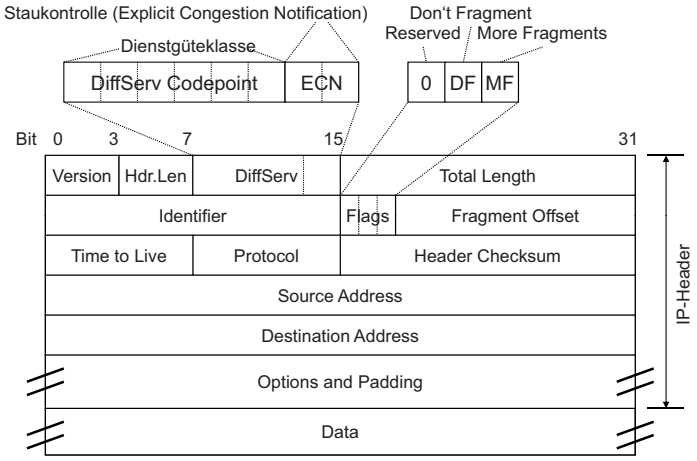
Protokoll	lokaler Port	lokale IP-Addr.	globaler Port	globale IP-Addr.	Ziel-IP-Addr.	Ziel-Port
TCP	1024	192.168.1.1	1024	129.133.3.1	207.171.4.4	1234
TCP	1500	192.168.1.2	1500	129.133.3.1	134.100.4.4	80

5.1.6. DHCP: Dynamic Host Configuration Protocol

- **Anwendung**
  - Vereinfachung der Installation und Verwaltung von vernetzten Rechnern
  - liefert Rechnern notwendige Informationen über IP-Adresse, DNS-Server-Adresse, Domain-Namen, Subnetz-Masken, Router etc.
  - damit weitgehend automatische Integration eines Rechners in das Internet bzw. Intranet
- **Client/Server-Modell**
  - ein Client sendet via IP-Broadcast eine Anfrage an einen DHCP-Server an UDP Port 67 (unter Umständen über ein DHCP-Relay)
    Clientanfragen: DHCPDISCOVER, DHCPREQUEST, DHCPRELEASE, …
  - der Server antwortet (initial via IP-Broadcast) und liefert die angeforderte Konfiguration. Serverantworten: DHCPOFFER, *DHCPACK* (mit IP-Adresse), …

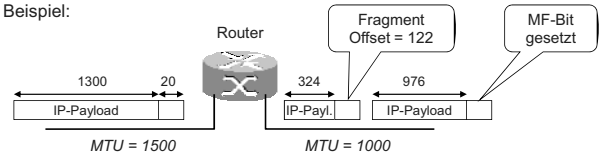


IP Datagramm: Aufbau



IP-Fragmentierung

- Größe eines IP-Pakets durch maximale Rahmengröße auf Schicht 2 begrenzt
  - MTU (Maximum Transport Unit): maximale Nutzdatenlänge in Schicht-2-Rahmen
  - Beispiel IEEE 802.3: MTU = 1500 Byte
- IP-Endsysteme kennen MTU der angeschlossenen Netzwerkadapter
  - Sender passt i.d.R. Paketgröße an lokale MTU an
- IP-Fragmentierung wird notwendig, wenn Paket über einen Link mit kleinerer MTU geroutet wird
  - Fragmentierung der IP-Payload an 8-Byte Grenzen
  - Beispiel:

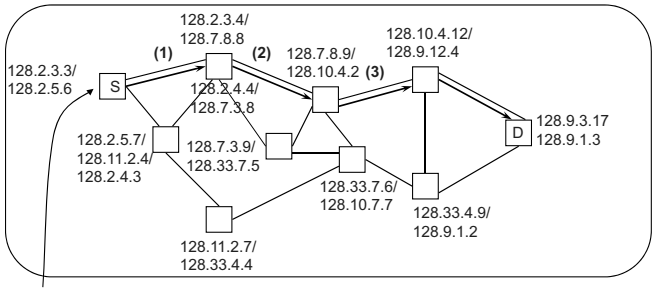


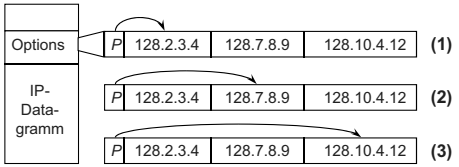
- Fragmentierung heutzutage durch Path-MTU-Discovery vermieden (➔ ICMP)

5.1.7. IP-Dienste: Überprüfung des Paketkopfes

- Überprüfungen, die nach dem Empfang eines IP-Datagrammes am Header durchgeführt werden
  - Überprüfung der korrekten Länge des Headers
  - Test der IP-Versionsnummer
  - Überprüfung der korrekten Datagrammlänge
  - Prüfsummenbildung über den IP-Header
  - Überprüfung der Paketlebenszeit
  - Überprüfung der Protokoll-ID
- Bei negativem Resultat eines der oben aufgeführten Tests wird das Paket einfach verworfen und eine Fehlermeldung über ICMP an den Sender des Pakets gesendet

Optionale IP-Dienste: Source Routing – Beispiel





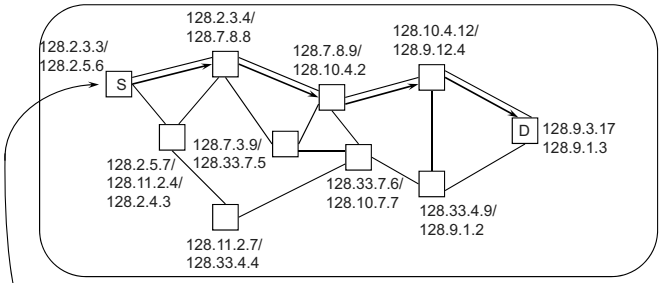
Hinweis:
Jeder Schnittstelle (Interface) ist eine eigene IP-Adresse zugeordnet. Im Diagramm sind aber nicht alle IP-Addressen aufgeführt.

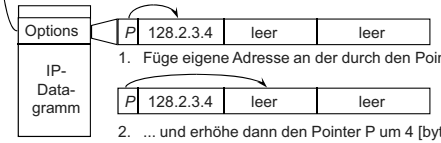
Optionale IP-Dienste: Zeitstempel

- Jeder Router fügt im Optionsfeld einen Zeitstempel ein, der den Zeitpunkt charakterisiert, zu dem das Paket vom Router bearbeitet wurde.
  - Aussagen über die Belastung der Netzwerke sind möglich
  - Die Effizienz der benutzten Routing-Algorithmen kann abgeschätzt werden
- Dabei existieren folgende Möglichkeiten, die durch ein 4 Bit langes Flag-Feld definiert werden:
  - Flag-Wert = 0: Nur Zeitstempel aufzeichnen, keine Adressen.
  - Flag-Wert = 1: Sowohl Zeitstempel als auch Adressen (Route Recording) aufzeichnen
  - Flag-Wert = 3: Die Adressen sind vom Sender vorgegeben (Source Routing), die adressierten Router tragen nur ihren Zeitstempel ein

Optionale IP-Dienste: Route Recording

- Im Record-Route-Options-Feld wird der durchlaufene Weg festgehalten





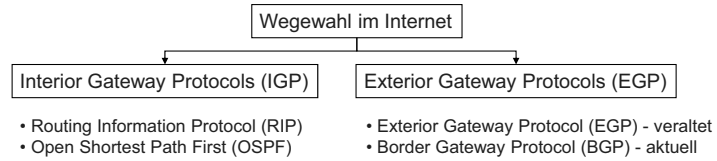
1. Füge eigene Adresse an der durch den Pointer P festgelegten Stelle ein ...

2. ... und erhöhe dann den Pointer P um 4 [byte], so dass er auf das nächste leere Feld in der Liste zeigt

Übersicht: IP-Routing Protokolle

- **IGP** (Interior Gateway Protocol): zur Wegewahl *innerhalb* einer Verwaltungseinheit (Administrative Domain oder Autonomous System)
  - **RIP** (Routing Information Protocol) basierend auf Distance-Vector-Algorithmus (überall verfügbar, aber veraltet)
  - **OSPF** (Open Shortest Path First) basierend auf Link-State-Algorithmus (neuer Standard)
  - IS-IS (Intermediate System-Intermediate System), ebenfalls Link-State-Algorithmus, aus der OSI-Welt, teilweise auch im Internet eingesetzt
- **EGP** (Exterior Gateway Protocol): Wegewahl *zwischen* Verwaltungseinheiten, sog. „politische Firewall“
  - EGP (Protokoll gleichen Namens!, veraltet)
  - **BGP** (Border Gateway Protocol, derzeit Version BGP4, RFC 1654)
  - Anwendungen:
    - Verhindern des Durchleitens „fremder“ Pakete durch eigenes Netz, obwohl der Weg kürzer ist
    - politische Restriktionen
    - Firmenpolitik (Firma X will nicht für den Transport der Pakete von Firma Y bezahlen)

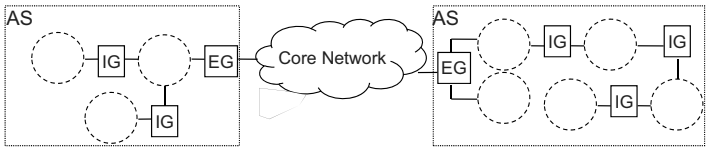
Dynamische Wegewahl im Internet



- Routing Information Protocol (RIP)
- Open Shortest Path First (OSPF)

- Exterior Gateway Protocol (EGP) - veraltet
- Border Gateway Protocol (BGP) - aktuell

Autonomes System (AS): Netz(e) unter einheitlicher Verwaltung
Interior Gateway (IG): Interner Router eines AS
Exterior Gateway (EG): Router am Rande eines AS

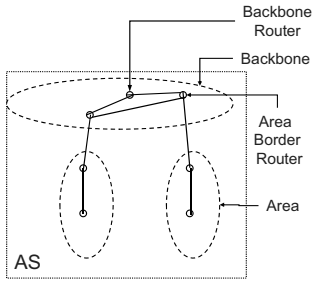


5.1.8. Routing Hierarchie – Protokolle

- **Intra-Domain-Routing:**
  - OSPF (Open Shortest Path First)
    - vom IAB empfohlenes Protokoll
    - „Link State"-Verfahren
  - RIP (Routing Information Protocol) - früher bzw. heute bei kleinen Netzen
    - wenig robust in komplexeren Netzwerken (Schleifenbildung)
    - langsamer bei Änderungen
    - Distanzvektorverfahren
- **Inter-Domain-Routing:**
  - BGP (Border Gateway Protocol)
    - Pfad-Vektor-Verfahren
    - BGP Version 4 (BGP4) unterstützt Classless Inter-Domain Routing (CIDR)

OSPF (Überblick)

- **Grundlage:**
  - Verbessertes Link State Routing
  - Basiert auf der Berechnung eines kürzesten Pfades von der Quelle zum Ziel
  - Hierarchische Aufteilung des AS
  - Intra-Area-, Inter-Area-, Inter-AS-Verkehr
  - Je Area Auswahl eines "Designated Router" (DR) und eines Backup-DR
- **Eigenschaften:**
  - Unterstützung unterschiedlichster Metriken, z.B. Anzahl Hops, Verzögerung
  - Adaptiv (d.h. reagiert auf Topologieänderungen)
  - Lastausgleich (Berücksichtigung verschiedener Wege) zum Zielknoten
  - Unterstützung hierarchischer (Sub-)Netze
  - Unterstützung verschiedener Wege



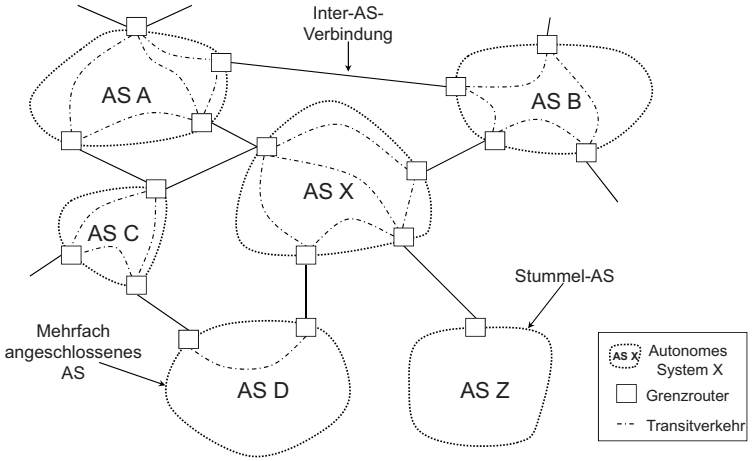
Intra-Domain-Routing mit OSPF

- Namensgebung:
  - Open: offener Entwicklungsprozess der IETF
  - SPF: durch Dijkstras Algorithmus kürzeste Pfade in Graph zu finden Komplexität: O(n log n), n= #Links
- Je OSPF-Area nicht mehr als 200 Router empfohlen
- Jeder Knoten
  - besitzt komplettes Abbild des Netzwerks (Routing-Datenbank) und
  - berechnet selbständig alle Pfade mit SPF-Algorithmus
- Austausch von geänderten Einträgen durch Fluten (bestätigt)
- Periodischer Austausch von Einträgen (Link-ID, Version)
  → interessante Einträge werden explizit angefordert
- Unterstützung mehrerer Metriken und mehrfacher Pfade
- Externe Routen werden gesondert eingetragen und vermerkt

Autonome Systeme im Internet

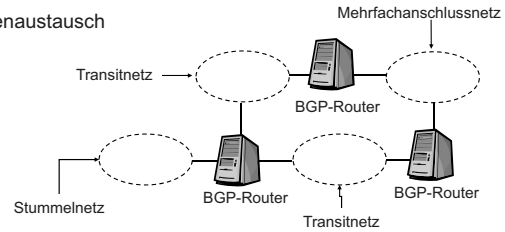
- **Autonomes System (AS):** Zusammenhängende Menge von Routern und Netzwerken unter derselben Administration
- Änderungen innerhalb des Systems sollen verborgen bleiben
- Betreiber möchte i.A. interne Struktur nicht bekannt geben
- Innerhalb eines AS sind auch unterschiedliche Intra-Domain-Routing-Protokolle möglich (RIP, OSPF)
- Jedes AS erhält eine eindeutige AS-Nummer (16 Bit)
- Derzeit > 14000 Autonome Systeme, davon
  - ~ 80% Stub/Origin only-AS
  - ~ 19% Mixed-AS
  - ~ 1% Transit-AS
  - Details siehe u.a.: http://bgp.potaroo.net/ (bgp table growth)

Internet-Architektur: Autonome Systeme



BGP (Überblick)

- **Aufgabe:**
  - Wegewahl zwischen autonomen Systemen (AS) unter Berücksichtigung besonderer politischer, wirtschaftlicher oder sicherheitsbezogener Regeln (Policies).
- **Grundlage:**
  - Distance Vector Routing
  - Exakter Pfad zum Zielknoten wird gespeichert
  - BGP-Router tauschen komplette Pfade aus ($\rightarrow$ Path Vector Routing)
  - Routing innerhalb der AS wird nicht betrachtet
  - TCP zum Datenaustausch



Inter-Domain-Routing mit BGP

- Welche Netzwerke sind über die Nachbardomänen erreichbar?
- Protokoll zum Austausch dieser Informationen: BGP-4 [RFC 1771]
- Unterstützung von Classless Interdomain Routing (CIDR)
- Berücksichtigung politischer Entscheidungen (Routing policies) möglich, daher Angabe kompletter Pfade (Schleifenfreiheit):

```
Dest. Net.       Next-Hop            Pfad
141.3.0.0/16 195.221.222.254      5409 1275 553
```

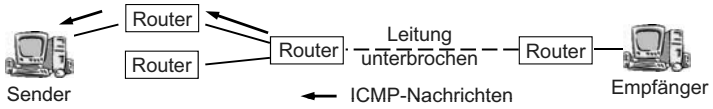
- Austausch der Routing-Tabellen bzw. der Änderungen erfolgt über TCP-Verbindungen
- Größe der Routingtabelle: derzeit bis zu 140000 Einträge
- Import von Routen des Intra-Domain-Routing-Protokolls

Zusammenspiel: OSPF ⇔ BGP

- Innerhalb eines AS muss jeder Router zu jedem Netzwerk-Präfix eine Route kennen
- Die Border-Router im Backbone-Bereich importieren und exportieren Routen zwischen BGP und OSPF
- Jeder Border-Router stellt mit jedem anderen Border-Router des AS eine interne BGP-Verbindung her
- Einfacher Fall: nur ein Router stellt die Verbindung zu externen Systemen her (Stub-Area) → Default-Route eintragen
- Weiterer Fall: Default-Route mit zusätzlichen expliziten Angaben
- Mehrere Router mit externer Verbindung:
  komplette Routing-Tabelle mit sämtlichen Zielnetzwerk-Präfixen notwendig (in OSPF importierte externe Routen)

Steuerung von IP: ICMP

- IP ist nur für den (unzuverlässigen) Datenaustausch zuständig.
- Für Fehlerfälle oder Testzwecke wird ICMP (Internet Control Message Protocol) verwendet.



Nachrichtentypen, Beispiele:

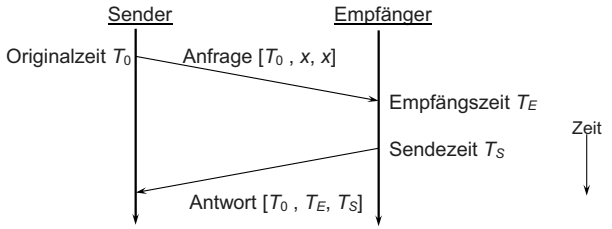
- *Destination Unreachable:* Ziel nicht erreichbar.
- *Time Exceeded*: Time-to-Live-Feld eines Pakets ist abgelaufen.
- *Echo Request / Reply*: Echo Reply wird angefordert ("`ping`").
- *Timestamp Request / Reply*: Ähnlich Echo Request. Zusätzlich Zeitstempel mit Ankunftszeit der Anfrage/Sendezeit der Antwort.

ICMP: Fehlermeldungen

- **Zieladresse nicht erreichbar** (destination unreachable):
  Ein Datenpaket konnte (z.B. wegen einer unterbrochenen Leitung oder eines ausgefallenen Routers) nicht zugestellt werden.
- **Zeit abgelaufen** (time exceeded):
  Datenpaket wurde wegen Ablauf seiner Lebenszeit von einem Router verworfen.
- **Falscher Parameter** (parameter problem):
  Datenpaket wurde wegen eines unzulässigen Wertes im IP-Paketkopf verworfen.
- **Quellendämpfung** (source quench):
  Ein überlastetes Kommunikationssystem fordert den Sender auf, die Übertragungsrate zu senken.
- **Umleiten** (redirect):
  Ein Datenpaket sollte besser über einen anderen Router gesendet werden.
  → Die Fehlermeldungen enthalten jeweils ein Feld zur genauen Angabe der Fehlerursache (z.B. „Netzwerk nicht erreichbar“ oder „Endsystem nicht erreichbar“ für die Meldung „Zieladresse nicht erreichbar“)

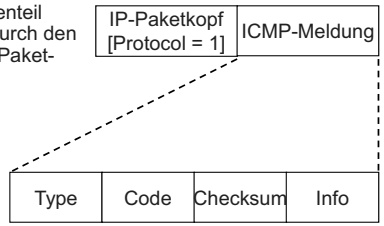
ICMP: Statusanfragen

- **Echo und Echoantwort** (echo and echo reply):
  Dient der Überprüfung der Aktivität von Kommunikationssystemen. Der Empfänger einer Echo-Anfrage sendet in der Echo-Antwort die erhaltenen Daten an den Kommunikationspartner zurück.
- **Zeitstempel und Zeitstempelantwort** (timestamp and timestamp reply): Dient der Bestimmung von Paketumlaufzeiten. Die Meldungen umfassen mehrere Felder zur Aufnahme von Zeitstempeln, anhand derer die Paketbearbeitungszeiten beim Empfänger und die Verzögerung im Netzwerk bestimmt werden können.



ICMP: Paketformat

- Übertragung der ICMP-Meldungen
  - ICMP-Meldungen werden im Datenteil von IP-Paketen übertragen und durch den Wert „1“ im Protocol-Feld des IP-Paketkopfes kenntlich gemacht.

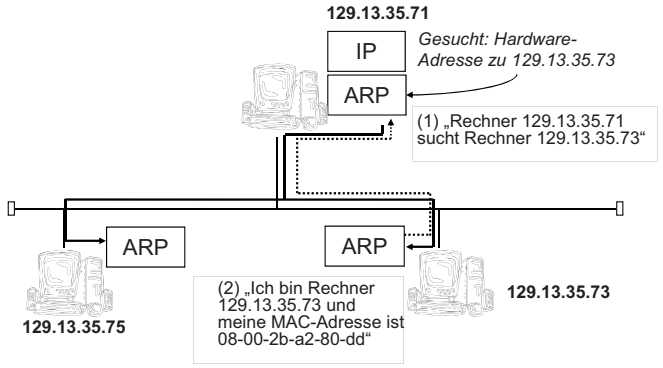


- Format der ICMP-Meldungen
  - Type: Typ der Meldung (z.B. Type = 3 entspricht „Zieladresse nicht erreichbar“)
  - Code: Genaue Beschreibung der Meldung (z.B. „Netzwerk nicht erreichbar“)
  - Checksum: Prüfsumme über die gesamte ICMP-Meldung
  - Der Inhalt des Info-Teils ist abhängig vom Typ der ICMP-Meldung (z.B. Felder für Zeitstempel bei Meldung „Zeitstempel und Zeitstempelantwort“)
  - ICMP „Packet too big“ Nachricht: enthält Typ = 2, Code = 0, Checksum, MTU-size, gefolgt von Original-Paket (max. 576 byte)

5.1.9. ARP: Address Resolution Protocol

- **Problem:**
  - Abbildung der Internet-Adresse eines Rechners auf die physikalische Adresse der Station (MAC-Adresse = Adresse der Adapterkarte)
- **Lösungsalternativen:**
  - Unterhalten einer Abbildungstabelle in jedem Rechner.
  - Unterhalten einer Abbildungstabelle in einem Server, der die Anfragen der Kommunikationssysteme beantwortet.
  - Dynamische Abbildung der Adressen durch Senden einer entsprechenden Anfrage an alle Rechner im LAN (Broadcast-Anfrage)
- Bei den beiden erstgenannten Methoden müssen die Abbildungstabellen bei jeder Änderung manuell abgeglichen werden.
- Ein Verfahren zur dynamischen Abbildung der Adressen ist durch das Address Resolution Protocol (ARP) festgelegt.

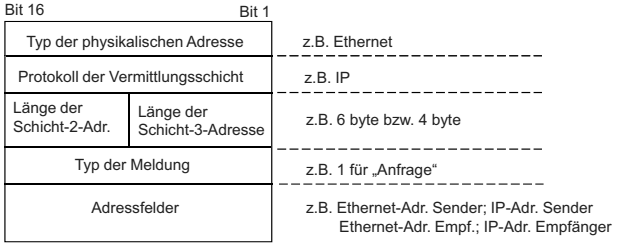
ARP – Beispiel



Sicherheitslücke?

ARP: Paketformat

- Übertragung der ARP-Meldungen
- Eine ARP-Meldung wird im Datenteil eines Paketes der Sicherungsschicht übertragen.
- Format der ARP-Meldungen:



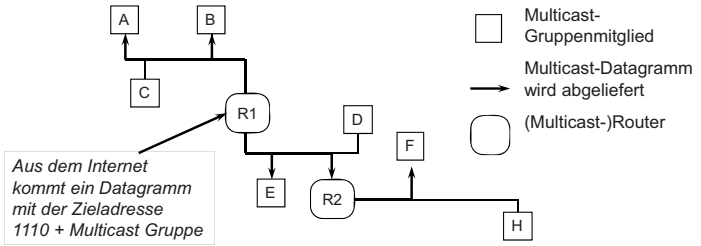
Länge und Aufbau der Adressfelder sind vom Typ der Adressen abhängig

Reverse Address Resolution Protocol (RARP)

- **Aufgabe:**
  - Umsetzen MAC-Adresse ⇨ IP-Adresse.
  - Wichtig z.B. für plattenlose Workstations, die von einem Dateiserver booten. Dazu müssen sie ihre IP-Adresse wissen, welche die Station allerdings beim Einschalten noch nicht kennt.
- **Vorgehensweise:**
  - Station schickt einen Rundruf ins lokale Netz unter Angabe der eigenen MAC-Adresse, die durch die Hardware vorgegeben ist.
  - RARP-Server sieht die Anfrage und bestimmt anhand einer Konfigurationsdatei die zugehörige IP-Adresse.
  - RARP-Server schickt die IP-Adresse in einer RARP-Antwort an die anfragende Station zurück.

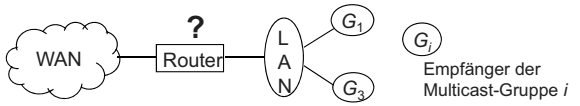
IP-Multicast

- Ein einzelnes IP-Datagramm wird an mehr als eine Station adressiert.
- Multicast-Verwaltung erfolgt über das IGMP (Internet Group Management Protocol).
- Es muss eine Class D-Adresse verwendet werden.
  - Die ersten vier Bits des Adressfelds im Kopfes entsprechen 1110.
  - Danach folgt die 28 bit lange ID der Gruppe.
- Beispiel:



IGMP: Internet Group Management Protocol

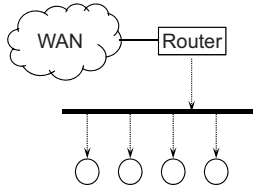
- **Problem:**
  - Wie erkennt ein Router, dass Multicast-Nachrichten bestimmter Gruppen von ihm weitergeleitet werden müssen?
- **Beispiel:**



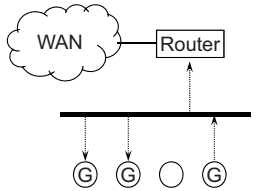
  - Nachrichten der Gruppen 1 und 3 müssen vom Router in das angeschlossene LAN weitergeleitet werden, wohingegen Nachrichten anderer Gruppen das LAN nicht erreichen sollten.
- **Lösungen:**
  - Manuelle Eingabe von Gruppenzugehörigkeiten in der Router-konfiguration ⇒ hoher Verwaltungsaufwand bei dynamischen Gruppen.
  - Selbstständiges Erlernen der Gruppenzugehörigkeiten durch den Austausch entsprechender Information ⇒ ein solches Verfahren wird durch das **Internet Group Management Protocol** (**IGMP**) beschrieben.

IGMP: Protokollablauf I

1. Der Router sendet periodisch Gruppenzugehörigkeitsanfragen an alle Rechner des LAN (via Broadcast). Durch Setzen der „Time To Live“ (TTL) auf 1 wird die Anfrage nur im LAN verbreitet.



2. Nach Erhalt einer Anfrage startet jeder Rechner für jede Gruppe, welcher er angehört, einen Zeitgeber. Dieser wird mit einem Zufallswert initialisiert. Nach Ablauf des Zeitgebers sendet der Host eine Antwort bzgl. der Gruppenzugehörigkeit an alle Gruppenmitglieder im LAN (Gruppenadresse, TTL=1). Multicast-Router erhalten alle IP-Multicast-Nachrichten.



IGMP: Protokollablauf II

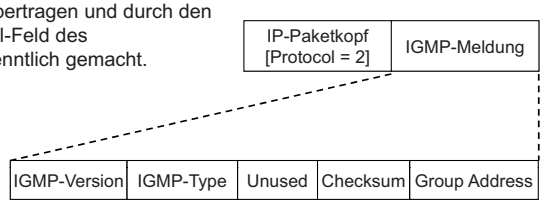
3. Weitere Gruppenmitglieder erhalten die Antwort und stoppen den entsprechenden Zeitgeber. Dadurch werden redundante Antworten vermieden.

4. Der Router erhält alle Antworten und aktualisiert seine Routingtabelle entsprechend. Erhält ein Router nach mehrmaliger Anfrage keine Antwort bzgl. einer bestimmten Gruppe, so wird ihr Eintrag aus der Routingtabelle gelöscht.

è Tritt ein Rechner einer Gruppe bei, so sendet er sofort eine entsprechende Mitteilung an alle Router im LAN. Aus Gründen der Fehlertoleranz wird die Mitteilung wiederholt gesendet.

IGMP: Paketformat

- Übertragung der IGMP-Meldungen
- IGMP-Meldungen werden im Datenteil von IP-Paketen übertragen und durch den Wert 2 im Protocol-Feld des IP-Paketkopfes kenntlich gemacht.



- Format der IGMP-Meldungen
  - IGMP-Version: Versionsnummer des eingesetzten IGMP-Protokolls.
  - IGMP-Type: Typ der Meldung (z.B. 1 = Anfrage, 0 = Antwort).
  - Unused: Wird nicht genutzt (immer zu 0 gesetzt).
  - Checksum: Prüfsumme über die gesamte IGMP-Meldung.
  - Group Address: Wird bei einer Anfrage auf 0 gesetzt, bei einer Antwort enthält das Feld die Adresse derjenigen Gruppe, auf welche sich die Meldung bezieht.

5.1.10. IPv6 – Motivation (ursprünglich)

Das Internet funktioniert seit Jahrzehnten! Warum ein neues IP-Protokoll?

Anwachsen des Internets: Der überwältigende Erfolg des Internets führte zu stark anwachsenden Benutzerzahlen
→ Bedarf an Adressen, sowie Änderungen von Adressen mit IPv4 nicht zufriedenstellend lösbar

Neue Anwendungen: Neuartige Anwendungen erfordern neuartige Dienste und zusätzliche Funktionalität (z.B. garantierte Netzdienste, Synchronisation von Audio- und Videodaten, automatisierte Konfiguration)
→ Konzepte zur funktionalen Erweiterung erforderlich

Hohe Datenraten: Hochleistungsfähige Zwischensysteme benötigen geeignete Paketformate zur effizienten Bearbeitung

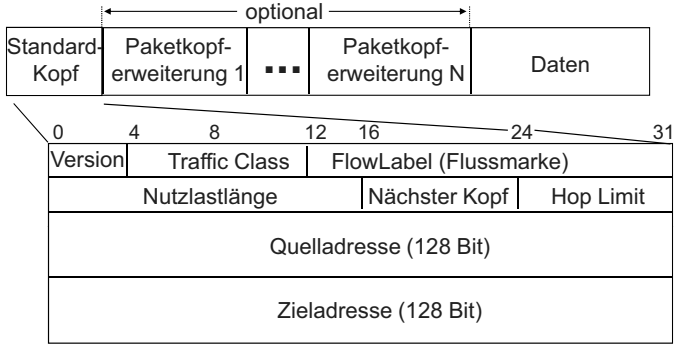
Überblick: Neuerungen in IPv6 (1)

- **Flexibles Paketformat**
  - Vereinfachung des (Standard-) Paketkopfes: 8 statt 13 Felder (IPv4)
  - Paketkopf fester Länge
  - Verschieben zahlreicher Optionen in optionale Paketkopferweiterungen
- **Erweiterte Adressierung**
  - Erhöhung der Adresslänge von 32 Bit auf 128 Bit
    - 2^32 bit: ~ 4 Mrd. IPv4-Adressen
    - 2^128 bit: ~ 3,4·10^38 IPv6-Adr., ~ 6 x 10^23 Adressen pro m^2 Erdoberfläche
  - Definition mehrerer Hierarchieebenen (effizienteres Routing)
  - Einführung von *Anycast-Adressen* (Kommunikation zu einem Mitglied einer Gruppe)
- **Unterstützung von Ressourcenreservierung**
  - *FlowLabel* (Flussmarke) und Traffic Class pro IPv6-Paket
  - Ermöglichen die Nutzung von Protokollen zur Ressourcenreservierung

Überblick: Neuerungen in IPv6 (2)

- **Neighbor Discovery**
  - Adressauflösung: IPv6-Adr.->MAC-Adr. (in ICMP integriert, ersetzt ARP)
  - Erkennen des nächsten Routers
- **Automatische Systemkonfiguration**
  - „Plug'n Play“: Inbetriebnahme eines Internet-Rechners ohne manuelle Konfiguration.
  - Realisiert durch DHCP (Dynamic Host Configuration Protocol)
- **Unterstützung mobiler Systeme**
  - Adresszuweisung durch automatische Systemkonfiguration
  - Die Option „Binding Update“ im Destination-Options-Header ermöglicht die direkte Umleitung der IP-Pakete an den aktuellen Standort
- **Berücksichtigung von Sicherheitsaspekten**
  - Unterstützung von Authentifizierung und Datenintegrität
  - Sicherheitsmechanismen werden bei IPv6-Implementierungen gefordert (sind bei IPv4 in Form von IPSec lediglich optional vorhanden)

IPv6-Paketformat



Vereinfachung des Paketformats gegenüber IPv4:
- Eliminierung verschiedener Felder (z.B. Kopflänge, Kopf-Prüfsumme)
- Verschieben der Optionen in Paketkopferweiterungen
- Spezifikation: RFC2460 (Neufassung von RFC1883), RFC2373, ...

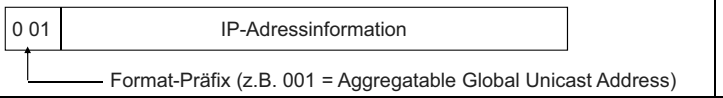
Kopferweiterungen

- Kopferweiterungen (Header Extensions) erlauben eine effizientere Implementierung und bewahren die Flexibilität für spätere Erweiterungen
- Definition unterschiedlicher Kopferweiterungen: Knoten-zu-Knoten-Optionen, Ziel-Optionen, Routing, Fragmentierung, Authentifizierung, Verschlüsselung
- Jeder Typ einer (optionalen) Kopferweiterung darf höchstens einmal in einer Dateneinheit enthalten sein
- Der Typ der jeweils nachfolgende Kopferweiterung wird im Feld „Nächster Kopf“ eingetragen
- Beispiel:

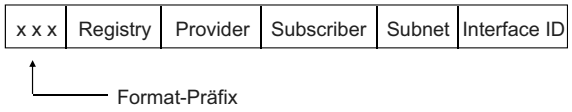
Standard-Kopf (Nächster Kopf = Routing)	Routing-Kopf (Nächster Kopf = TCP)	TCP-Kopf + Benutzerdaten

IPv6: Adressierung

- IPv6-Adressen (Länge 128 Bit)
  - Notation `x:x:x:x:x:x:x:x`, wobei jede Stelle 16 Bit hexadezimal kodiert
  - Beispiele: `3FFE:400:20::A00:2BFF:FEA3:ADCB`
    `FF01:0:0:0:0:0:0:101` oder `FF01::101`
    `FEDC:BA98:7600::/40` 40 Bit langes Präfix für das Routing
  - Unterscheidung von **Adressklassen**:
    - Unicast-Adressen
    - Anycast-Adressen
    - Multicast-Adressen – Präfix ff00::/8 (ff…)
  - Unterscheidung von **Adresstypen**:
    - Link-Local Address
    - Site-Local Address
    - Aggregatable Global Unicast Address
  - Typ einer Adresse wird durch *Format-Präfix* (führende Bits) festgelegt:



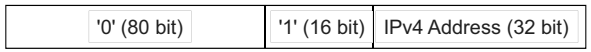
Unicast-Adressen in IPv6 (1)



- Aggregatable Global Unicast Address (Provider-Based Unicast Address)
  - Hierarchischer Aufbau zur Vereinfachung der Vermittlung von IPv6-Daten.
  - Globale Gültigkeit im Internet.
  - Format-Präfix 001
- Link-Local Address
  - Besteht aus Präfix und Kennung der Netzkarte (Interface-Identification).
  - Format-Präfix fe80::/10 (also 1111 1110 10)
  - Registry = Provider = Subscriber = Subnet = '0'
  - Ausschließlich innerhalb eines Subnetzes gültig.
- Site-Local Address
  - Besteht aus Präfix, Kennung des Subnetzes (Subnet) und Kennung der Netzkarte.
  - Format-Präfix fec0::/10 (fec0… bis feff…)
  - Registry = Provider = Subscriber = '0'
  - Ausschließlich innerhalb eines Netzwerkes gültig.
  - veraltet (deprecated), stattdessen Unique Local Unicast RFC 4193

Unicast-Adressen in IPv6 (2)

- Adressformate für den Übergang von IPv4 nach IPv6
  - IPv4 Mapped Address (repräsentiert der IPv6-Anwendung IPv4-Adresse als IPv6 Adresse)

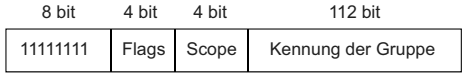


  - IPv4 Compatible Address (zum Tunneln von IPv6-Paketen über IPv4-Router)

- Einfache Abbildung zwischen IPv4- und IPv6-Adressen.
  - Ergeben die gleiche Prüfsumme wie die entsprechenden IPv4-Adressen.
  - keine Neuberechnung des TCP-Pseudo-Headers nötig.

Multicast-Adressen in IPv6

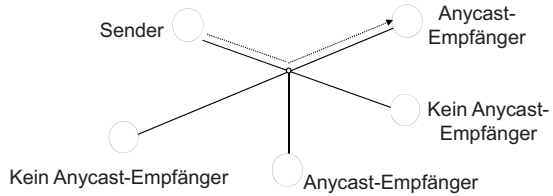
- Multicast-Adressen identifizieren eine Menge von Netzinterfaces
- Multicast-Adressen sind durch ein spezielles Präfix gekennzeichnet:



- Zwei Typen von Multicast-Gruppen (Unterscheidung durch das Feld "Flag")
  - Permanente Gruppen existieren dauerhaft
  - Transiente Gruppen existieren nur zeitweise
- Gültigkeitsbereich der Adresse wird durch das Feld "Scope" festgelegt
- Einige Multicast-Adressen sind bereits für bestimmte Zwecke reserviert
  - (z.B. "All Systems on this Subnet", "DVMRP Routers")

Anycast-Adressen in IPv6

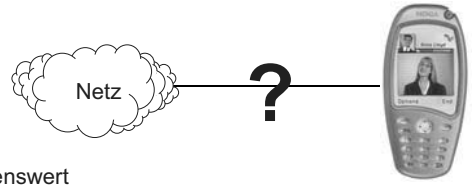
- Die an eine Anycast-Adresse gesendete Dateneinheit wird an ein Mitglied der Anycast-Gruppe ausgeliefert.



- Anycast-Adressen entsprechen syntaktisch den Unicast-Adressen
- Es ist nicht gewährleistet, dass aufeinanderfolgende Anycast-Dateneinheiten an den gleichen Empfänger übermittelt werden
- Anycast kann z.B. zur Lokalisierung von Netz-Ressourcen genutzt werden

Automatische Systemkonfiguration

- Problem
  - Aufwand durch manuelle Konfiguration von Internet-Rechnern (z.B. eigene IP-Adresse, Adresspräfix, Nameserver), keine Unterstützung für mobile Rechner und "Ad-Hoc"-LANs

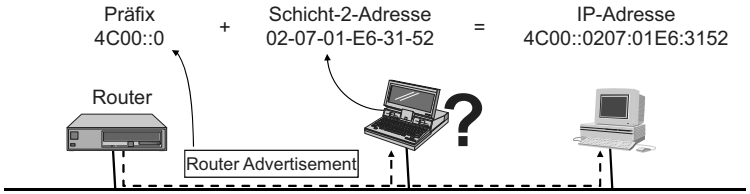


- Wünschenswert
  - "Plug'n Play" Lösung, d.h. Inbetriebnahme eines Internet-Rechners ohne manuelle Konfiguration allein durch physikalische Anbindung an das Netz

- Zwei Arten der automatischen Systemkonfiguration
  - Adresskonfiguration durch Neighbor Discovery
  - Adresskonfiguration und Serverbekanntgabe durch Dynamic Host Configuration Protocol (DHCP)

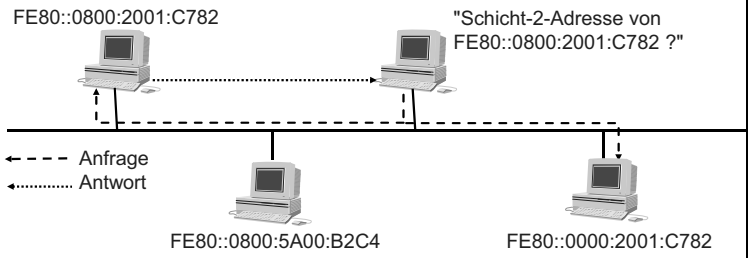
Automatische Systemkonfiguration durch Neighbor Discovery (RFC 2461)

- Präfix des Subnetzes ist global eindeutig
- Schicht-2-Adresse ist zumindest innerhalb des Subnetzes eindeutig
- IP-Adresse (im ganzen Internet routbar) kann aus dem Präfix des Subnetzes und der Schicht-2-Adresse zusammengesetzt werden
- Neighbor Discovery verwendet ICMPv6-Nachrichten
  Router Advertisement, Router-Solicitation, Neighbor-Solicitation, Neighbor-Advertisement, …



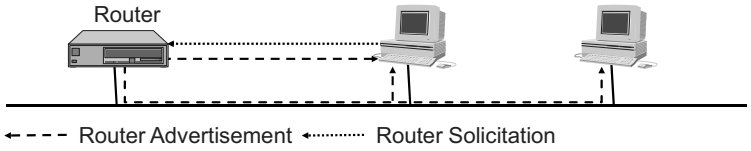
- Werden keine Router Advertisements gesendet, so können Link-Local Adressen mit FE80-Präfix gebildet werden, z.B.
  FE80::0 + 02-07-01-E6-31-52 = FE80::0207:01E6:3152

Neighbor Discovery



- **Problem:**
  - Abbildung einer IPv6-Adresse auf die entsprechende Schicht-2-Adresse
- **Lösung:** Anfrage mittels *Neighbor Solicitation*
  - Rechner hören auf *Solicited Nodes Address* (Präfix `FF02::1:` gefolgt von den letzten 32 Bit (z.B. `2001:C782`) der Unicast-Adresse, z.B. `FF02::1:2001:C782`)
  - Anfrage an die *Solicited Nodes Address* des Zielrechners (per L2-Multicast)
  - Zielrechner übermittelt seine Schicht-2-Adresse an den anfragenden Rechner

Router-Erkennung durch Neighbor-Discovery



- **Problem:**
  - Ermittlung eines Routers zum Senden von Dateneinheiten an Rechner außerhalb des eigenen Netzsegmentes.
- **Lösung:**
  - Router senden periodisch *Router Advertisement* Nachrichten an die "All Hosts" Adresse `FF02::1`. Diese enthalten unter anderem:
    - *Router Lifetime*: Zeitspanne bis zur Ungültigkeit der enthaltenen Information
  - Rechner können mittels *Router Solicitation* ein *Router Advertisement* anfordern, welches dann jedoch per Unicast gesendet wird.

Präfix-Erkennung durch Neighbor Discovery

- **Problem:**
  - Absender einer IP-Dateneinheit muss feststellen, ob sich der Zielrechner im eigenen Subnetz befindet (direktes Senden oder Senden über Router)
- **Lösung:**
  - Entscheidung basierend auf dem Präfix des eigenen Subnetzes.
    - Router Advertisement Nachrichten enthalten Präfix-Listen des Subnetzes.
    - Vergleich der Zieladresse mit den Präfixen durch logische UND-Verknüpfung.
    - Entspricht das Präfix der Zieladresse einem Präfix des Subnetzes, so wird das Paket direkt gesendet; ansonsten wird es an einen Router übermittelt.
- **Beispiel:**
  - Ein Rechner im Subnetz mit dem Präfix 4C00::0001:0:0:0 möchte an den Rechner mit Adresse 4C00::0002:0800:5A01:3982 senden:

```
        4C00::0001:   0:   0:   0
AND     4C00::0002:0800:5A01:3982
        ============================
        4C00::0000::              0
```

(ungleich dem Präfix des Subnetzes ➔ an einen Router übermitteln)

Übergang von IPv4 zu IPv6

- Langsamer Übergang geplant, keine abrupte Umstellung
  - wird sich über mehrere Jahre hinweg ziehen
- Zwei Strategien können verfolgt werden
  - Doppelter Stack
    - Während der Übergangsphase implementiert jeder IPv6-Knoten zusätzlich noch den IPv4-Protokollstack
  - IPv6 Tunneling
    - Tunnels werden benutzt, um IPv4-Pfade zu überbrücken. Diese Technik wurde im 6Bone angewendet
- Erforderliche Veränderungen in anderen Protokollen:
  - Modifiziertes Socket-API ist erforderlich
  - Modifizierte Routingprotokolle müssen eingeführt werden
  - DNS muss IPv6-Adressen unterstützen

Migration von IPv4 nach IPv6: Das 6Bone

- Weltweites „Overlay“-Netz zur Entwicklung und zum Testen von IPv6
- Das 6Bone verband IPv6-Subnetze über IPv4-Netze durch Konfiguration von Tunneln

Zukunft von IPv6

- Zunächst weltweite Erprobung in einem Overlay-Netzwerk („6-Bone“)
- Mittlerweile ein Regeldienst – neben IPv4
- Sollte IPv4 ersetzen, ob und wann aber derzeit nicht absehbar
- Ursprüngliche Motivation für IPv6:
  - Anwachsende Benutzerzahlen, Adressknappheit
    - Gegenmaßnahmen: **CIDR**, **NAT**; dadurch kein dringender Handlungsbedarf mehr im Festnetz – dafür aber im 3G-Mobilnetz!
  - Neuartige Dienste erfordern Unterstützung
    - Einsatz von QoS/RSVP umstritten, daher kein dringender Bedarf für flow-label-Feld in IPv6
    - Nutzung von Type-of-Service-Feld in IPv4 möglich, um Dienstgüteunterstützung zu realisieren („Differentiated Services“-Architektur)
  - Hohe Datenraten erfordern effizientes Paketformat
    - könnte bei breitem Einsatz von Verschlüsselungsverfahren auf Netzwerkschicht an Bedeutung gewinnen

5.2. Internet und Mobilität

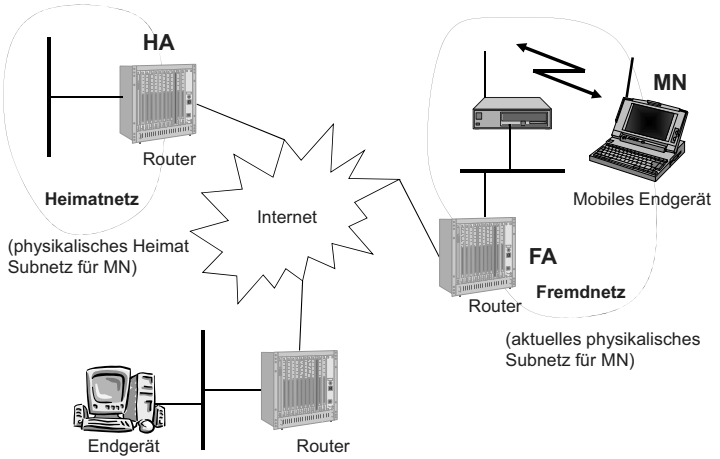
- Wegwahl
  - basiert auf IP-Zieladresse, Netzwerk-Präfix (z.B. 129.13.42) legt physikalisches Subnetz fest
  - wird das Subnetz gewechselt, **muss** auch die IP-Adresse passend gewechselt werden (bei IP ohne Mobilitätsunterstützung) oder ein spezieller Routing-Eintrag vorgenommen werden
- Spezifische Routen zum Endgerät?
  - anpassen aller Routing-Einträge, damit Pakete umgeleitet werden
  - skaliert nicht mit Anzahl der mobilen Geräte und u.U. häufig wechselnden Aufenthaltsorten, Sicherheitsprobleme
- Wechseln der IP-Adresse?
  - je nach Lokation wird entsprechende IP-Adresse gewählt
  - wie sollen Rechner nun gefunden werden – DNS-Aktualisierung dauert lange
  - TCP-Verbindungen brechen ab, Sicherheitsprobleme!

  ⇨ **Mobile IP**

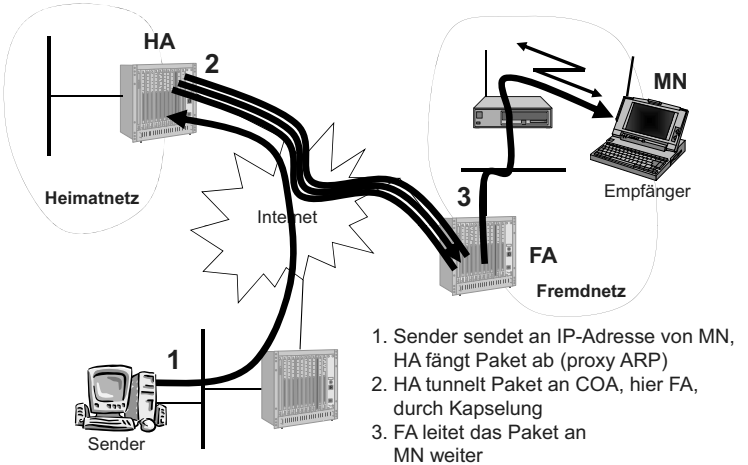
5.2.1. Terminologie

- Mobile Node (MN)
  - Knoten, der den Ort des Netzanschlusses wechseln kann, ohne seine IP-Adresse ändern zu müssen
  - typischerweise mobiles Endgerät
- Home Agent (HA)
  - Einheit im „Heimatnetz“ des MN, typischerweise Router
  - verwaltet Aufenthaltsort des MN, tunnelt IP-Datagramme zur COA
- Foreign Agent (FA)
  - Einheit im momentanen „Fremdnetz“ des MN, typ. Router
  - weiterleiten der getunnelten Datagramme zum MN, stellt meist auch default-Router für den MN dar, stellt COA zur Verfügung
- Care-of Address (COA)
  - Adresse des für den MN aktuell gültigen Tunnelendpunkt
  - stellt aus Sicht von IP aktuelle Lokation des MN dar
  - kann z.B. via DHCP gewählt werden

5.2.2. Beispielnetz

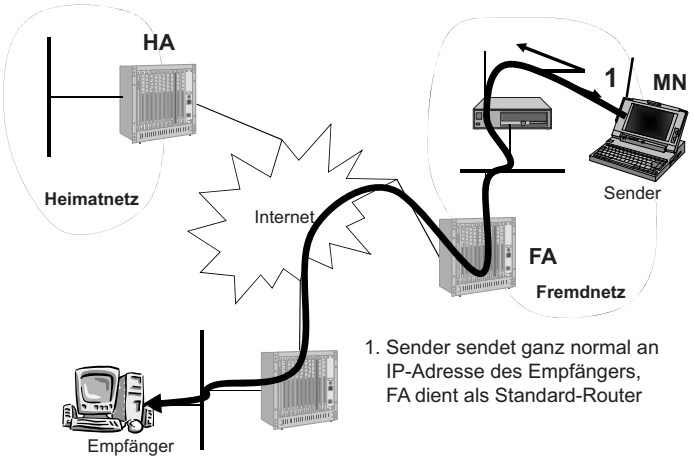


Datentransfer zum Mobilrechner



1. Sender sendet an IP-Adresse von MN, HA fängt Paket ab (proxy ARP)
2. HA tunnelt Paket an COA, hier FA, durch Kapselung
3. FA leitet das Paket an MN weiter

Datentransfer vom Mobilrechner



Einige Probleme mit Mobile IP

- **Sicherheit**
  - Authentifizierung mit FA problematisch, da u.U. nicht unter eigener Kontrolle (fremde Organisation)
  - Sicherheitsprotokolle, Schlüsselverwaltung und -verteilung
- **Firewalls**
  - verhindern typischerweise den Einsatz von Mobile IP, spezielle Konfigurationen sind nötig (z.B. reverse tunneling)
- **QoS**
  - häufige erneute Reservierungen im Fall von RSVP
  - Tunneln verhindert das Erkennen eines gesondert zu behandelten Datenstroms
- Sicherheit, Firewalls, QoS etc. sind aktueller Gegenstand vieler Arbeiten und Diskussionen!

Lehrstuhl für Netzarchitekturen und Netzdienste
Institut für Informatik – Technische Universität München
Prof. Dr.-Ing. Georg Carle

Grundlagen: Rechnernetze und Verteilte Systeme

Kapitel 6: Transport-Protokolle

TCP, UDP

Prof. Dr.-Ing. Georg Carle
Lehrstuhl für Netzarchitekturen und Netzdienste
Technische Universität München
carle@net.in.tum.de
http://www.net.in.tum.de

Übersicht

1. Einführung und Motivation
   - Bedeutung, Beispiele
2. Begriffswelt und Standards
   - Dienst, Protokoll, Standardisierung
3. Direktverbindungsnetze
   - Fehlererkennung, Protokolle
   - Ethernet
4. Vermittlung
   - Vermittlungsprinzipien
   - Wegwahlverfahren
5. Internet-Protokolle
   - IP, ARP, DHCP, ICMP
   - Routing-Protokolle
6. **Transportprotokolle**
   - **UDP, TCP**
7. Verkehrssteuerung
   - Kriterien, Mechanismen
   - Verkehrssteuerung im Internet
8. Anwendungsorientierte Protokolle und Mechanismen
   - Netzmanagement
   - DNS, SMTP, HTTP
9. Verteilte Systeme
   - Middleware
   - RPC, RMI
   - Web Services
10. Netzsicherheit
    - Kryptographische Mechanismen und Dienste
    - Protokolle mit sicheren Diensten: IPSec etc.
    - Firewalls, Intrusion Detection
11. Nachrichtentechnik
    - Daten, Signal, Medien, Physik
12. Bitübertragungsschicht
    - Codierung
    - Modems

Ziele

- In diesem Kapitel wollen wir vermitteln
  - Arten von Transportdiensten
  - Verbindungsaufbau und -abbau
  - Aufgaben der Transportschicht
  - Funktionalität TCP
  - Funktionalität UDP

Kapitelgliederung

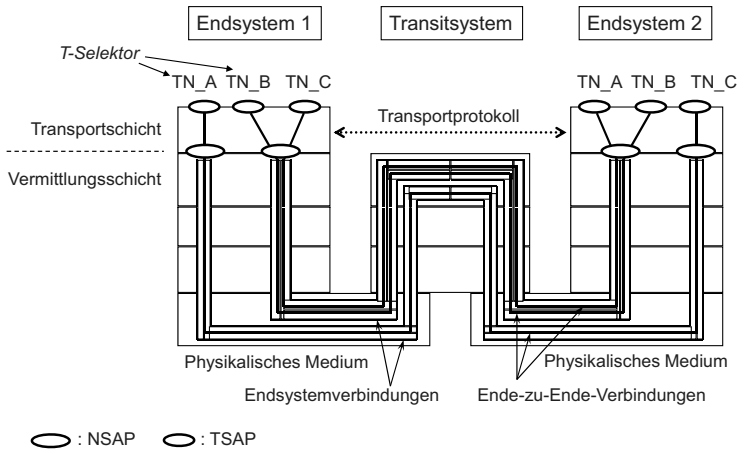
6.1. Der Transportdienst (nach ISO/OSI-Begriffswelt)

- Man unterscheidet die folgenden Transportdienste:
  - **verbindungsorientiert**
  - **verbindungslos**

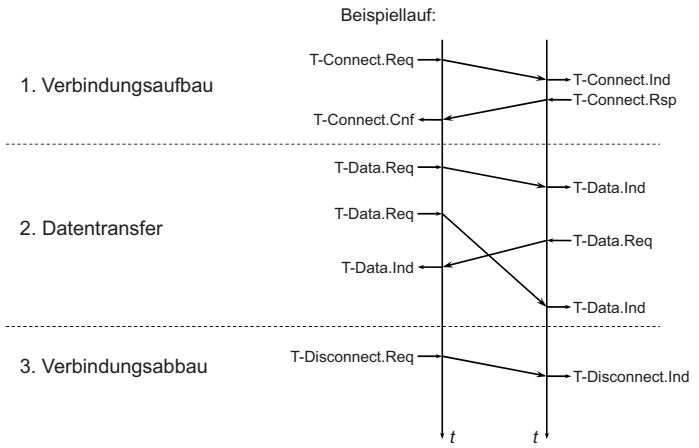
- Beim **verbindungsorientierten** Dienst unterscheidet man drei Phasen:
  - Verbindungsaufbauphase (Dienstelement: T-Connect)
  - Datentransferphase (Dienstelement: T-Data)
  - Verbindungsabbauphase (Dienstelement: T-Disconnect)

- Adressierung eines Transportdienstbenutzers durch *TSAP-Adresse* (Transport Service Access Point), beinhaltet:
  - *NSAP-Adresse* (Network Service Access Point) zur Adressierung des Endsystems
  - *T-Selektor* zur Identifizierung des TSAP auf einem Endsystem

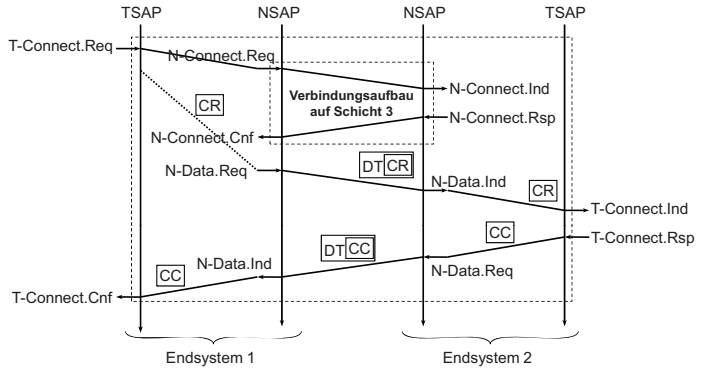
Abstraktionseigenschaft der Transportschicht



6.1.1. Phasen des verbindungsorientierten Dienstes



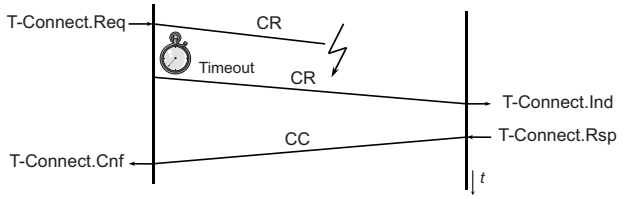
Verbindungsaufbau auf Schicht 3 für verbindungsorientierten Transportdienst



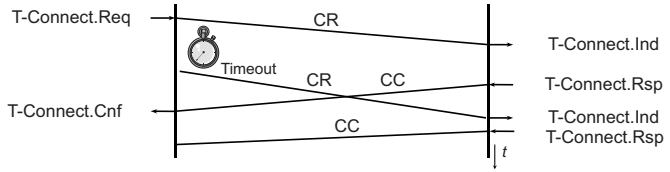
- **Hinweis:** Setzt die Transportschicht auf einem verbindungslosen Dienst der Vermittlungsschicht auf (z.B. IP) oder existiert bereits eine Schicht-3-Verbindung, so ist kein Verbindungsaufbau auf Vermittlungsebene notwendig!

6.1.2. Fehler beim Verbindungsaufbau

- Verlust der CR oder CC TPDU:

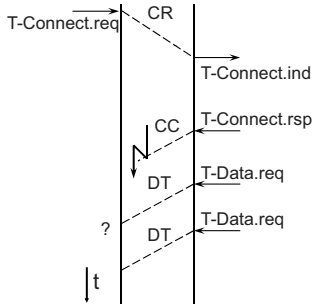


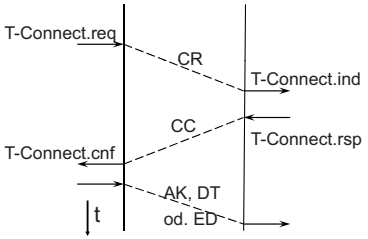
- Duplizierung von TPDUs:



Three-Way Handshake

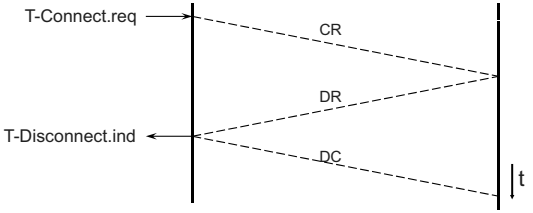
- **Problem:** Verlust der CC TPDU
- **Lösung:** Three-Way Handshake
  Verbindung wird erst als aufgebaut anerkannt, wenn beide Verbindungsaufbau TPDUs (CR und CC) quittiert sind.





Verbindungsrückweisung

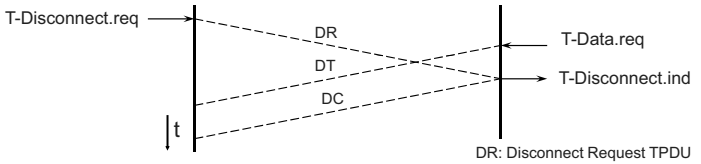
- Connection Refusal:
  - Zurückweisung eines Verbindungsaufbauwunsches erfolgt durch Disconnect-Request (DR) oder Error-TPDU. Gründe für Zurückweisung werden angegeben.
  - Gründe:
    - Zurückweisung durch den Transportdienstbenutzer.
    - Anforderungen an den Dienst können nicht erfüllt werden



DR: Disconnect Request TPDU
DC: Disconnect Confirm TPDU

6.1.3. Verbindungsabbau

- Normal Release:
  - Beim Verbindungsabbau wird eine bestehende Transportverbindung aufgelöst. Dabei kann es zum Verlust von Daten kommen.
  - Varianten:
    - implizit: Abbau der Vermittlungsschichtverbindung.
    - explizit: Abbauprozedur über Disconnect-TPDUs.

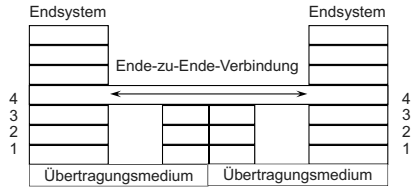


DR: Disconnect Request TPDU
DC: Disconnect Confirm TPDU

  - Verbindungsabbau nach Fehler (Error Release): Kann nach einem Fehler (N-Disconnect oder N-Reset) keine geeignete Fehlerbehandlung erfolgen, wird eine Transportverbindung vom Transportdiensterbringer abgebaut.

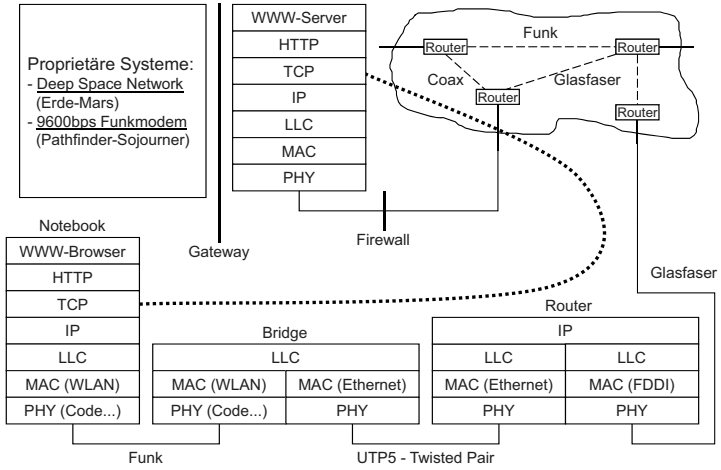
6.2. Aufgaben der Transportschicht

- **Ende-zu-Ende-Verbindung**
  (Teilnehmer-zu-Teilnehmer statt Rechnerknoten-zu-Rechnerknoten)



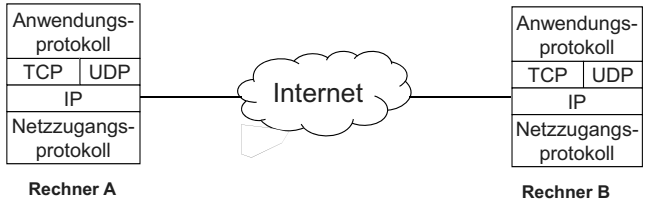
- Isolierung der höheren Schichten von der Technologie, der Struktur und den Unvollständigkeiten der verwendeten Subnetze.
- Transparente Übertragung der Nutzdaten.
- Wahlmöglichkeiten für die Dienstgüte.
- Unabhängige Teilnehmeradressierung: globaler Adressraum für Teilnehmer, unabhängig von Adressen der unteren Schichten.
- **Ziel:** Effizienter und zuverlässiger Dienst soll angeboten werden

6.2.1. Ende-zu-Ende Kommunikation im Internet



Die Transportschicht im Internet

- Im Internet kommen auf Transportebene derzeit hauptsächlich zwei Protokolle zum Einsatz:
  - **TCP** (Transmission Control Protocol): Zuverlässiges, verbindungsorientiertes Transportprotokoll über unzuverlässigem IP
  - **UDP** (User Datagram Protocol): Verbindungsloses Transportprotokoll. Bietet eine Anwendungsschnittstelle zu IP, d.h. es verbessert den Dienst von IP nicht wesentlich.



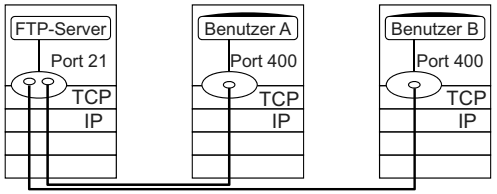
6.2.2. TCP: Eigenschaften und Dienste (II)

- **Datenübertragung:**
  - Vollduplex
  - Fehlerkontrolle durch Folgenummern (Sequenznummern), Prüfsumme, Quittierung, Übertragungswiederholung im Fehlerfall
  - Reihenfolge, Flusskontrolle (durch Fenstermechanismus) und Staukontrolle
  - Unterstützung von Sicherheitsstufen und Prioritäten
  - Zeitbehaftete Daten: Falls die Auslieferung in einer bestimmten Zeit nicht möglich ist, wird der Dienstbenutzer informiert.

- **Fehleranzeige:**
  - Treten während der Verbindung Störungen auf, wird der Benutzer darüber in Kenntnis gesetzt.

TCP: Adressierung

- Identifikation von TCP-Diensten geschieht über Ports (TSAPs in der OSI-Terminologie)
- Portnummern bis 255 sind standardisiert ("well known ports") und für häufig benutzte Dienste reserviert (z.B. 21 für FTP, 23 für TELNET, 80 für HTTP)
- Ein FTP-Server ist z.B. über (IP-Adresse:Portnummer)129.13.35.7:21 erreichbar
- Socket: Kommunikationsendpunkt einer Kommunikationsbeziehung der Transportschicht, welche durch Fünftupel (Protokoll, lokale Adresse, lokale Portnummer, entfernte Adresse, entfernter Port) spezifiziert ist



TCP: fest vereinbarte port-Nummern (well-known ports)

Viele Anwendungen wählen TCP als Protokoll, allerdings muss der richtige *port* gewählt werden, um auf der Gegenseite mit der richtigen Anwendung zu kommunizieren.

- 13: Tageszeit
- 20: FTP Daten
- 25: SMTP (Simple Mail Transfer Protocol)
- 53: DNS (Domain Name Server)
- 80: HTTP (Hyper Text Transfer Protocol)
- 119: NNTP (Network News Transfer Protocol)

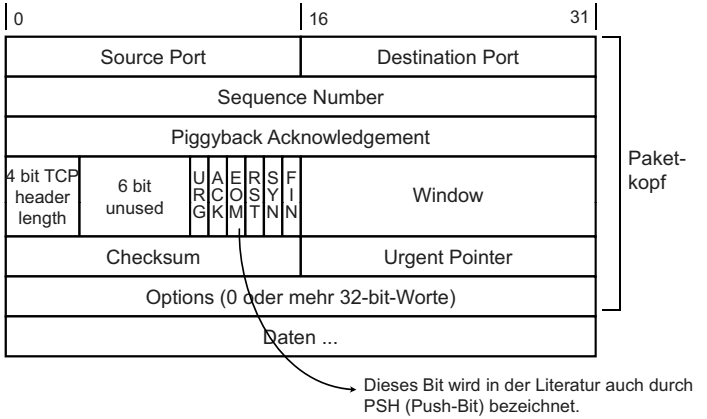
```
> telnet walapai 13
Trying 129.13.3.121...
Connected to leonis.
Escape character is '^]'.
Mon Aug  4 16:57:19 2002
Connection closed by foreign host
```

```
> telnet mailhost 25
Trying 129.13.3.161...
Connected to mailhost .
Escape character is '^]'.
220 mailhost ESMTP Sendmail 8.8.5/8.8.5; Mon,
4 Aug 2002 17:02:51 +0200
HELP
214-This is Sendmail version 8.8.5
214-Topics:
214-    HELO    EHLO    MAIL    RCPT    DATA
214-    RSET    NOOP    QUIT    HELP    VRFY
214-    EXPN    VERB    ETRN    DSN
214-For more info use "HELP <topic>".
...
214 End of HELP info
```

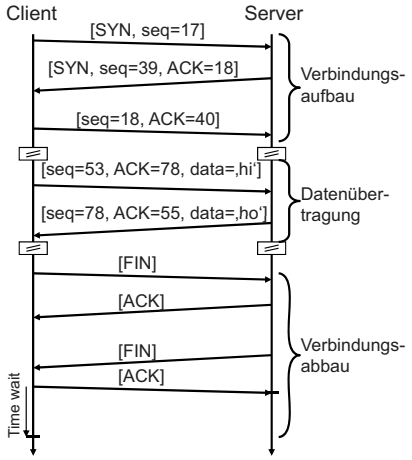

TCP: Verbindungsaufbau

- Verbindungen können nach der Erstellung eines Sockets **aktiv** (connect) oder **passiv** (listen/accept) aufgebaut werden.
  - Aktiver Modus: Anforderung einer TCP-Verbindung mit dem spezifizierten Socket.
  - Passiver Modus: Ein Dienstnutzer informiert TCP, dass er auf eine eingehende Verbindung wartet.
    - Spezifikation eines speziellen Sockets, von dem er eine eingehende Verbindung erwartet wird (fully specified passive open) oder
    - Alle Verbindungen annehmen (unspecified passive open).
    - Geht ein Verbindungsaufbauwunsch ein, wird ein neuer Socket erzeugt, der dann als Verbindungsendpunkt dient.
- Anmerkung:
  - Die Verbindung wird von den TCP-Instanzen ohne weiteres Eingreifen der Dienstbenutzer aufgebaut (es existiert z.B. kein Primitiv, das T-CONNECT.Rsp entspricht).

6.2.2.1. TCP-Paketformat: Aufbau



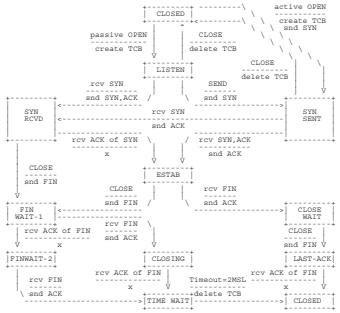
TCP-Verbindungsaufbau/Datenübertragung/ Verbindungsabbau



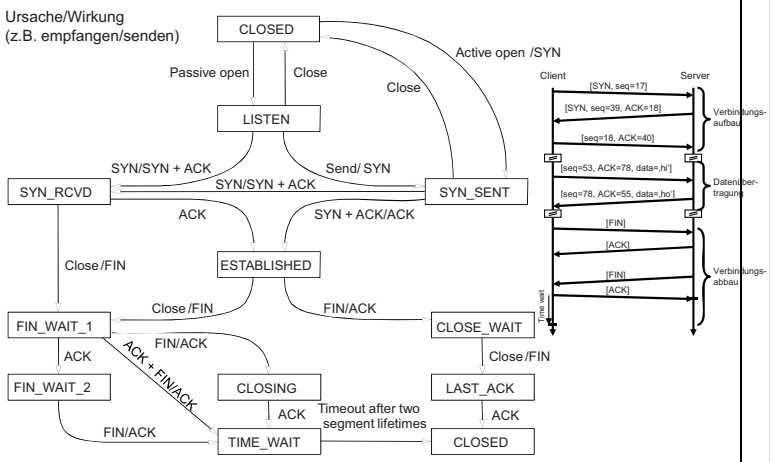
- Verbindungsaufbau
  - 3-Wege-Handshake
  - Aushandlung von Fenstergröße, Sequenznummern
- Datenübertragung
  - Bestätigung im Huckepack
- Verbindungsabbau
  - Bestätigt
  - Ressourcenfreigabe auf Client-Seite erst nach time-wait (frozen reference), typisch sind 30 s, 1 min, 2 min – beeinflusst stark die Leistungsfähigkeit!

TCP-Zustandsübergangsdiagramm

- RFC 793



TCP-Zustandsübergangsdiagramm



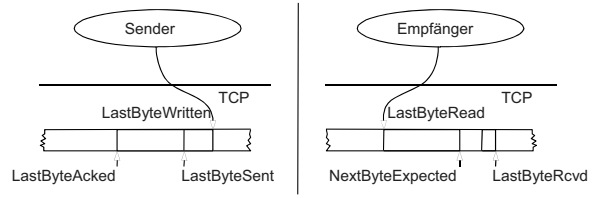
6.2.2.2. TCP: Mechanismen (I)

- Einheit der Datenübertragung: **Segment** (TCP-Header + Nutzdaten)
  - Größenbeschränkung durch max. IP-Nutzdatengöße von 65536 Byte
  - in Praxis: Größe von mehreren tausend Byte, um Fragmentierung auf IP-Ebene zu vermeiden
- **Mechanismen:**
  - Verwendung von **Timern:**
    - z.B. **Retransmission-Timer**: Wird beim Senden eines Segments gestartet
      - Übertragungswiederholung, falls keine Bestätigung vor Ablauf des Timers
      - komplexe Berechnung des Timer-Wertes
  - Verwendung des **Sliding-Windows-**Verfahrens**:**
    - Fenstergröße variabel, wird dem Sender im Feld *Window* mitgeteilt (auch Receiver Window genannt)
    - maximal 16 Bit (2^{16}=65536 Byte)
      - unzureichend für Hochleistungsnetze ⇨ Fensterskalierung bis 2^{32} Byte

TCP: Mechanismen (II)

- Piggybacking (jeweils 16 Bit Sequenznummer)
- Prüfsummenbildung
  - Sicherung von TCP-Header, Nutzdaten und TCP-Pseudoheader
  - TCP-Pseudoheader = IP-Quell-/Zieladresse, IP-Protocol-Feld (6), TCP-Segmentgröße
- Ähnlich wie Go-Back-N
  - Bestätigungsnr. *n* bestätigt Empfang aller Bytes bis Seq.-Nr. *n-1*
  - Übertragungswiederholung aufgrund Ablauf von Retransmission Timer
  - Pakete, die in der falschen Reihenfolge ankommen, werden am Empfänger zwischengespeichert (Unterschied zu Go-Back-N).
- Aushandlung bei Verbindungsaufbau:
  - Selective-Repeat (RFC 1106)
    - durch NAK kann fehlerhaftes Segment explizit angefordert werden
  - Selective Acknowledge (1996, RFC 2018)
    - durch SACK werden einzelne, korrekt empfangene Segmente bestätigt
- Flusssteuerung und Staukontrolle
  - Siehe Kapitel 9

Sliding Window – Prinzip in TCP



Sender

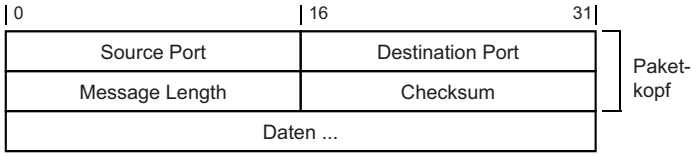
- `LastByteAcked <= LastByteSent`
- `LastByteSent <= LastByteWritten`
- Puffern aller Daten zwischen `LastByteAcked` und `LastByteWritten`

Empfänger

- `LastByteRead < NextByteExpected`
- `NextByteExpected <= LastByteRcvd +1`
- Puffern aller Daten zwischen `NextByteRead` und `LastByteRcvd`

6.2.3. UDP (User Datagram Protocol)

- Unzuverlässig, verbindungslos, einfacher und schneller als TCP
- Demultiplexing der empfangenen Pakete basiert auf der Port-Nummer
- Optionale Prüfsumme



- festgelegte, sog. „well-known“ ports:
  - 13: daytime
  - 53: domain name server
  - 123: network time protocol
- sehr viele Multimedia-Anwendungen nehmen UDP statt TCP wegen Leistungsvorteilen

Lehrstuhl für Netzarchitekturen und Netzdienste
Institut für Informatik – Technische Universität München
Prof. Dr.-Ing. Georg Carle

Grundlagen: Rechnernetze und Verteilte Systeme

Kapitel 7: Verkehrssteuerung

Kriterien, Mechanismen, Verfahren

Prof. Dr.-Ing. Georg Carle
Lehrstuhl für Netzarchitekturen und Netzdienste
Technische Universität München
carle@net.in.tum.de
http://www.net.in.tum.de

Übersicht

1. Einführung und Motivation
   - Bedeutung, Beispiele
2. Begriffswelt und Standards
   - Dienst, Protokoll, Standardisierung
3. Direktverbindungsnetze
   - Fehlererkennung, Protokolle
   - Ethernet
4. Vermittlung
   - Vermittlungsprinzipien
   - Wegwahlverfahren
5. Internet-Protokolle
   - IP, ARP, DHCP, ICMP
   - Routing-Protokolle
6. Transportprotokolle
   - UDP, TCP
7. **Verkehrssteuerung**
   - **Kriterien, Mechanismen**
   - **Verkehrssteuerung im Internet**
8. Anwendungsorientierte Protokolle und Mechanismen
   - Netzmanagement
   - DNS, SMTP, HTTP
9. Verteilte Systeme
   - Middleware
   - RPC, RMI
   - Web Services
10. Netzsicherheit
    - Kryptographische Mechanismen und Dienste
    - Protokolle mit sicheren Diensten: IPSec etc.
    - Firewalls, Intrusion Detection
11. Nachrichtentechnik
    - Daten, Signal, Medien, Physik
12. Bitübertragungsschicht
    - Codierung
    - Modems

Ziele

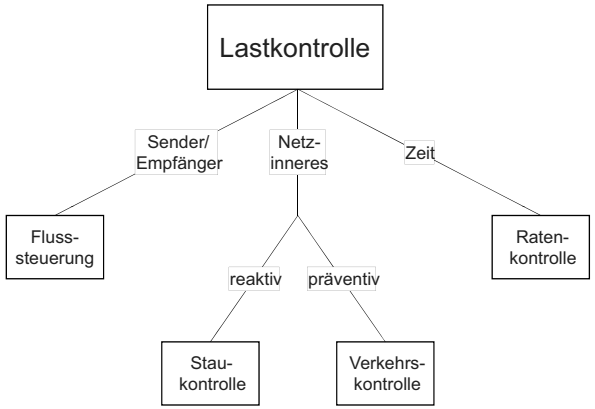
- In diesem Kapitel wollen wir vermitteln
  - Lastkontrolle
  - Flusssteuerung
  - Überlastung im Netzinnern
  - Verkehrskontrolle
  - Staukontrolle
  - Ratenkontrolle
  - Dienstgüte

Kapitelgliederung

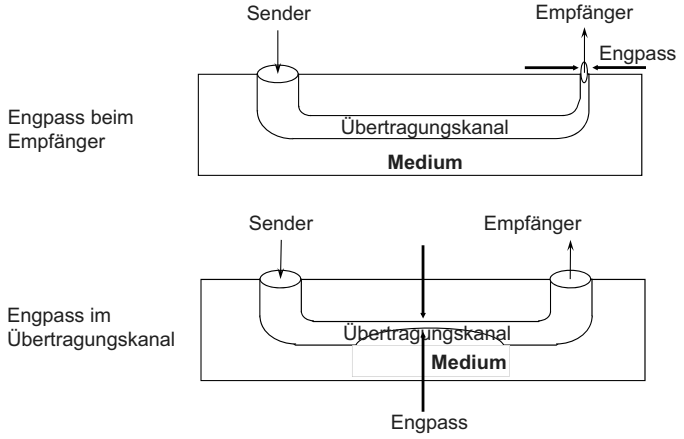
7.1. Lastkontrolle

- An einem Kommunikationsvorgang sind üblicherweise mehrere Systeme beteiligt:
  - das den Vorgang initiierende System;
  - das gerufene System;
  - das Netzwerk mit den entsprechenden Zwischensystemen.
- Die auszutauschende Datenmenge muss dabei an aktuelle Eigenschaften der beteiligten Systeme (Netzknoten, Endsysteme) angepasst werden, da es sonst zu unterschiedlichen Problemen kommen kann (siehe die folgenden Folien).
- In der Vorlesung soll als Überbegriff für die Reglementierung der zu sendenden Datenmenge der Begriff ***Lastkontrolle*** stehen.
- Es sei hier jedoch darauf hingewiesen, dass dieser Begriff (wie auch die weiteren in diesem Thema eingeführten) in der Literatur leider nicht eindeutig definiert und benutzt wird!

Zusammenfassung

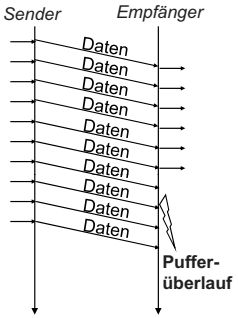


7.1.1. Engpässe in Kommunikationsnetzen

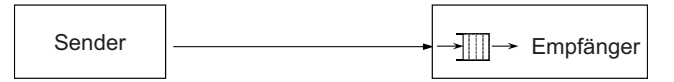


Engpass im Empfänger

- **Annahme:**
  - Netz ist kein Engpass, sondern hält mit Sender(n) mit.
- **Ursachen für Engpass im Empfänger:**
  - In einem Rechnernetz kommunizieren Rechner unterschiedlicher Leistungsfähigkeit (langsamer Empfänger).
  - (oder) Empfänger bekommt Pakete von vielen Sendern
- **Problem:**
  - Leistungsfähiger Rechner sendet mit hoher Rate - oder viele Rechner senden an einen Empfänger
  - Empfänger kann diese nicht in entsprechender Geschwindigkeit verarbeiten.
  - Empfangspuffer läuft über. Daten gehen verloren.



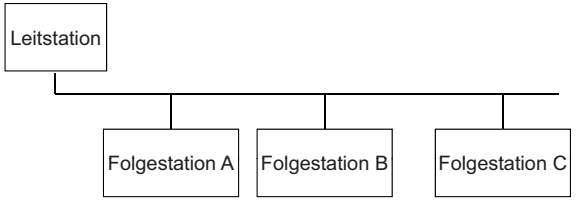
Bsp.: Pufferüberlauf bei Punkt-zu-Punkt-Verbindung



Leistungsfähiger Sender

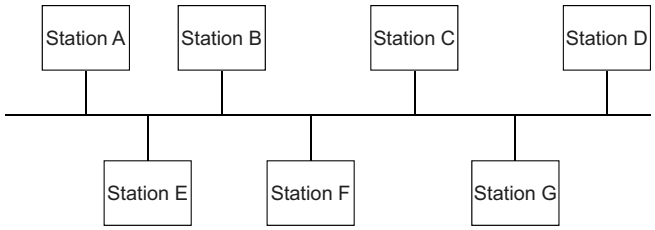
Leistungsschwacher Empfänger

Bsp.: Perfekte Lastkontrolle durch Polling-Betrieb



Engpass im Übertragungskanal

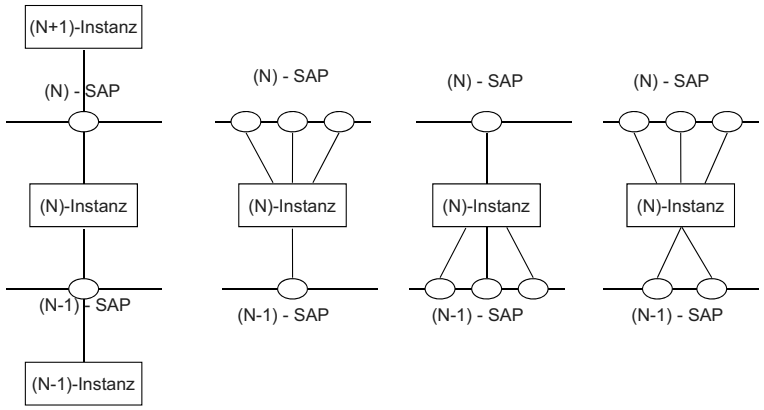
- **Beispiel:** Überlastung des geteilten Mediums
- **Annahme:** alle Stationen könnten die eintreffenden Daten verarbeiten
- **Aber:** Die Kapazität des geteilten Mediums reicht nicht aus für die zahlreichen Übertragungswünsche der Stationen
- **Folge:** Daten gehen schon beim Zugriff auf das Medium verloren und kommen erst gar nicht bei den Empfängern an.



7.1.2. Flusssteuerung

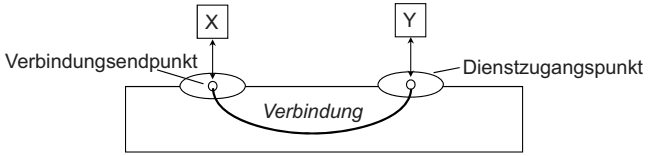
- **Synonyme Begriffe**
  - Flusssteuerung
  - Flussregulierung
  - Flusskontrolle
  - Flow Control
- **Aufgabe**
  - Auf Netzebene ist der **Datenpaketempfänger** vor einem zu großen Zufluss von Paketen eines Paketsenders zu schützen.
- **Ort der Durchführung**
  - Schicht 2 (Sicherungsschicht): Überlastungsschutz von Übermittlungsabschnitten
  - Schicht 3
  - Schicht 4

Flusssteuerung auf verschiedenen Schichten des OSI-Modells



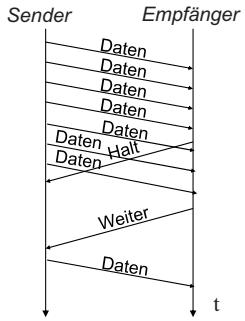
7.1.2.1. Datagramm versus Verbindung

- **Datagramm (verbindungsloser Transportdienst)**
  - Datagramme werden unabhängig voneinander transportiert und laufen ggf. auch auf unterschiedlichen Wegen durchs Netz
  - Keine Kontextinformation innerhalb des Netzes
  - ➔ Flusssteuerung muss auf höheren Schichten erfolgen
- **Verbindungen (verbindungsorientierter Transportdienst)**
  - Kontext, etabliert durch Verbindungsaufbau
  - Kontextinformation beinhaltet Adressinformation und zusätzliche Verbindungsidentifikation (z.B. Portnummer bei TCP/UDP), wenn mehreren Verbindungen vom selben Dienstzugangspunkt ausgehen
  - ➔ Flusssteuerung durch Anpassung der Sendegeschwindigkeit



7.1.2.2. Arten von Flusssteuerung
Flusssteuerung mit Halt-/Weiter-Meldungen

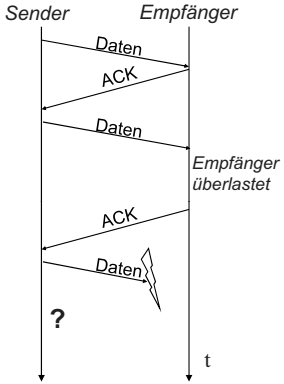
- **Einfachste Methode:**
  - Sender-Empfänger-Flusssteuerung
    - Meldungen:
      - **Halt**
      - **Weiter**
  - Kann der Empfänger nicht mehr Schritt halten, schickt er dem Sender eine **Halt**-Meldung.
  - Ist ein Empfang wieder möglich, gibt der Empfänger die **Weiter**-Meldung.



- **Beispiel:** Protokoll XON/XOFF
  - Mit ISO 7-Bit-Alphabetzeichen.
  - XON ist DC_1 (Device Control 1).
  - XOFF ist DC_3 (Device Control 3).
  - Nur auf Vollduplex-Leitungen verwendbar.

Implizite Flusssteuerung

- **Funktionsweise:**
  - Durch Zurückhalten der Quittung (z.B. ACK/NAK) kann der Sender gebremst werden.
  - Das bedeutet, dass ein Verfahren zur Fehlererkennung für die Flusssteuerung mitbenutzt wird.



- **Problem:**
  - Der Sender kann nicht mehr unterscheiden,
    - ob sein Paket völlig verloren ging, oder
    - ob der Empfänger die Quittung wegen Überlast zurückgehalten hat.
  - ➔ ineffizient

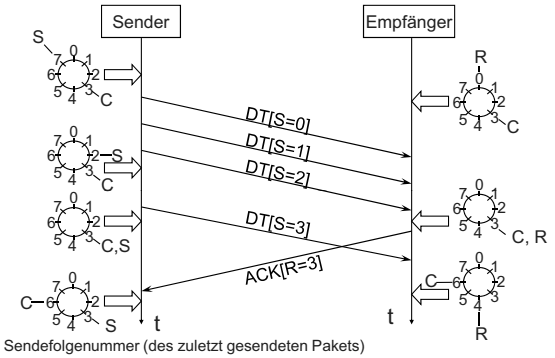
Kreditbasierte Flusssteuerung

- **Funktionsweise:**
  - Der Empfänger räumt dem Sender einen mehrere Transfereinheiten umfassenden Sendekredit ein.
  - Ist der Kredit (ohne neue Kreditgewährung) erschöpft, stellt der Sender den Transfer ein.
  - Dazu ist aber eine verstärkte Fehlerkontrolle, z.B. für den Verlust der neuen Kreditgewährung, erforderlich.

- **Realisierungsmöglichkeiten:**
  - Explizite Kreditgewährung:
    - Empfänger teilt dem Sender explizit den aktuellen Kredit mit.
  - Kreditfenster („Sliding Window“):
    - Mit jedem quittierten Paket wird das Kreditfenster verschoben.

Kreditbasierte Flusssteuerung: Sliding Window

- **Beispiel:** Fenstermechanismus (Kredit 4) für eine Senderichtung

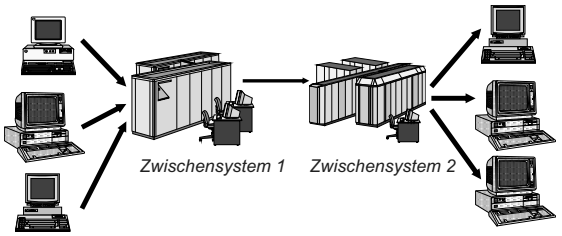


S: Sendefolgenummer (des zuletzt gesendeten Pakets)
R: Nächste erwartete Sendefolgenummer = Quittierung bis Folgenummer R-1
C: Oberer Fensterrand (maximal erlaubte Sendefolgenummer)

- **Nachteil:** Kopplung von Fluss- und Fehlerkontrolle.

7.1.3. Überlastung im Netzinnern

- **Ausgangspunkt:**
  - Mehrere Kommunikationsvorgänge werden im Netzinneren über dieselben Zwischensysteme abgewickelt.
- **Problem:**
  - Obwohl die Sender für sich keine sehr große Last produzieren, können die Ressourcenanforderungen der einzelnen Kommunikationsvorgänge in einem Zwischensystem zu einer Überlastsituation führen



Netzüberlastung

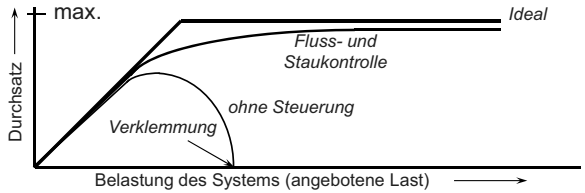
- **Synonyme Begriffe**
  - Netzüberlastung
  - Network Congestion

- **Gefahren**
  - Pufferüberläufe in Zwischensystemen bei speichervermittelten Netzen
  - Bei leitungsvermittelten Netzen werden bei Überlast Verbindungswünsche abgewiesen (Besetzt-Fall)

- **Gründe für lokale oder globale Netzüberlastung**
  - Zu starker Zufluss von Transfereinheiten (Paketen)
  - Ausfall von Speichervermittlungen / Teilnetzen mit Verkehrsverlagerung auf andere Netzkomponenten
  - Zunahme transienter Störungen in Netzkomponenten

7.1.3.1. Stau-/Verkehrskontrolle

- **Begriffe**
  - Staukontrolle (Congestion Control)
  - (Über-)Lastkontrolle
  - Verkehrskontrolle (Traffic Control)
- **Problem und Ziel**
  - Im Überlast-/Staubereich sinkt im ungesteuerten Fall der Durchsatz bis zum totalen Netzstillstand, z.B. durch Verklemmungen (deadlock).
  - Erreichen eines „vernünftigen“ Netzverhaltens bei Hochlast-/Überlastbetrieb ist Ziel der Verfahren



Verkehrskontrolle versus Staukontrolle

- **Verkehrskontrolle/Staukontrolle:**
  - Mechanismen des Netzwerks, um Stausituationen zu vermeiden und Auswirkungen von Staus zu begrenzen (aus Standard ITU-T I.371)
  - **Verkehrskontrolle:** *präventive* Mechanismen zur Stauvermeidung
  - **Staukontrolle:** *reaktive* Mechanismen zur Begrenzung der Auswirkungen in Stausituationen
  - Die Begriffe werden oftmals nicht eindeutig benutzt!
- **Teilaufgaben der Verkehrskontrolle/Staukontrolle:**
  - Zugangskontrolle
  - Nutzungskontrolle

  } Präventiv (Verkehrskontrolle)
  - Prioritätskontrolle
  - Reaktive Staukontrolle

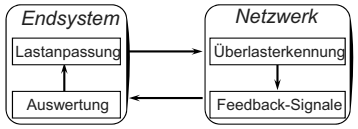
  } Reaktiv (Staukontrolle)

7.1.3.2. Anforderungen an die Verkehrskontrolle/Staukontrolle

- **Überlastsituationen**
  - Netz ist nicht mehr in der Lage, Ressourcen zur vollständigen (garantierten) Durchführung der Dienste zur Verfügung zu stellen.
  - Verkehrskontrolle
    - Mechanismen, um Überlast möglichst zu vermeiden.
  - Staukontrolle
    - Mechanismen, um eine aufgetretene Überlast einzugrenzen und zu beheben.
- **Anforderungen an die Stau- und Verkehrskontrolle**
  - Einfachheit und Robustheit:
    - geringe zusätzliche Belastung des Netzes
  - Effektivität:
    - eine schnelle und wirkungsvolle Reaktion wird gefordert.
  - Fairness:
    - garantierte Dienste sollten eingehalten werden;
    - sich korrekt verhaltende Benutzer sollen von den Entscheidungen nicht betroffen sein;
    - jeder Benutzer soll angemessen berücksichtigt werden

7.1.3.3. Verkehrs-/Staukontrollverfahren

- **Im Endsystem**
  - Lokal vorliegende Informationen werden ausgewertet (wie z.B. Warteschlangenlänge, Anzahl negativer Quittungen, ...)
  - Die momentane Situation wird bewertet und durch lokale Aktionen zu verbessern versucht.
  - Dies betrifft
    - die weitere Verbindungsannahme,
    - das Verwerfen von ankommenden Paketen,
    - die Wegewahl, etc.
  - Ungenauigkeiten durch rein lokale Sichtweise.
- **Im Netzwerk**
  - Im Netzwerk verteilt arbeiten Mechanismen zur Überwachung der aktuellen Situation.
  - Bei Entdeckung einer Stausituation werden die relevanten Informationen an die entsprechende Stelle im Netz weitergegeben (Feedback).
  - Dort muss die Arbeitsweise dann an eine mögliche Stausituation angepasst werden.
  - Können selbst zum Stau beitragen!



Verkehrskontrolle: Pufferreservierung und Verwerfen von Paketen

- **Pufferreservierung**
  - Geeignete Verwaltung von Puffern in den einzelnen Speichervermittlungsstellen (auch in den Endstellen), z.B. durch rechtzeitige Allokation.
  - Eine Pufferreservierung kann z.B. bei der Verbindungsaufbauprozedur für eine virtuelle Verbindung erfolgen und über die ganze Verbindungsdauer erhalten bleiben.

- **Wegwerfen von Paketen**
  - In Hochlast-Situationen werden Pakete verworfen, die danach nochmals gesendet werden müssen
  - Umsetzung im Internet:
    - Active Queue Management, RFC2309
    - RED (Random Early Detection), RFC 2481, http://www.aciri.org/floyd/red.html

Verkehrskontrolle: Paketzahlbegrenzung und Rückstau

- **Begrenzung der Paketzahl im Netz**
  - Es existieren „Berechtigungspakete“.
  - Diese werden vom Sender benötigt, um ein eigenes Paket an das Netz zu geben.
  - Ein Empfänger erzeugt bei Entgegennahme eines Pakets ein neues Berechtigungspaket.

- **Rückstaumechanismen**
  - Sie regulieren den Zufluss von Paketen und halten diesen gegebenenfalls an.
  - Besonders an den Eintrittspunkten in das Netz von Bedeutung, z.B. unter Nutzung von Maßnahmen der Flusssteuerung.
  - Im Prinzip ist die Flusssteuerung auf die Abstimmung von Arbeitsgeschwindigkeiten der sendenden und empfangenden Endeinrichtungen ausgerichtet.
  - Sie ist daher eigentlich nicht für die Lösung von netzinternen Überlastproblemen gedacht.

Staukontrolle: Reaktion auf Paketverlust

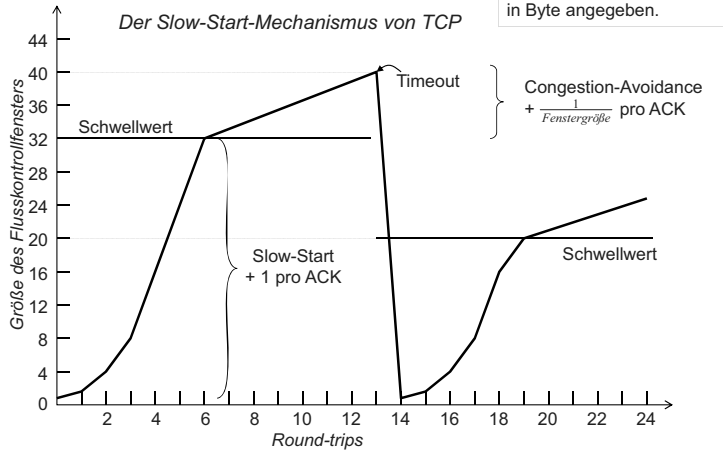
- **Ziel:**
  - Fehlersituationen im Netz auf Grund von Überlastung, Ausfällen etc. entgegenwirken
- **Schema:**
  - Reaktive Verfahren (z.B. bei TCP) arbeiten daher gemäß dem folgenden allgemeinen Schema:
    - Wird ein Paketverlust erkannt, setze das Flusskontrollfenster auf 1.
      - Wird z.B. durch den Ablauf eines Zeitgebers oder den Empfang eines entsprechenden Kontrollpakets des Empfängers erkannt.
    - Erhöhe das Fenster im Zuge erfolgreich übertragener Pakete allmählich, um wieder den maximalen Durchsatz erzielen zu können.
    - Mitunter werden zwei Phasen unterschieden (z.B. Slow-Start bei TCP)
      - Bis zu einem gewissen Schwellwert wird die Fenstergröße verdoppelt.
      - Ab diesem Wert wird die Fenstergröße um 1 vergrößert.
      - Der Schwellwert kann aufgrund erfahrener Paketverluste angepasst werden.
- **Allgemein:**
  - Prinzipien wie AIMD (Additive Increase Multiple Decrease) führen zu einem stabilen Verhalten. AIMD:
    - Bei erfolgreicher Übertragung Fenstergröße additiv (z.B. +1) vergrößern.
    - Bei Misserfolg Fenstergröße multiplikativ (z.B. • 0,5) verkleinern.

7.1.3.4. TCP: Flusssteuerung / Staukontrolle

- **Flusssteuerung** bei TCP:
  - regelt Datenfluss zwischen Endsystemen durch **kreditbasiertes Verfahren**
  - ACK-Feld im Paketkopf bestätigt alle niedrigeren Bytesequenznummern
  - Window-Feld gibt an, wie viele Bytes der Empfänger noch akzeptiert
- **Staukontrolle** bei TCP:
  - befasst sich mit Stausituationen in den Zwischensystemen
  - Problem: „congestion collapse“
    - Stau in Zwischensystemen führt oftmals dazu, dass Transportprotokolle nach einem Timeout Pakete wiederholen. ➔ Die Stausituation wird verstärkt!
  - Lösung: **„slow start“- und „multiplicative decrease“-Mechanismen**
    - Bei Datenverlust reduziert TCP den Schwellwert, bis zu dem eine Steigerung der Senderate möglich ist, auf die Hälfte des aktuellen Fensterwerts. (multiplicative decrease).
    - Nach einer Stauperiode wird die Fenstergröße um ein Datenpaket erhöht und weiterhin nach jeder empfangenen Quittung (slow start)
    - Der „slow start“ Mechanismus verhindert, dass direkt nach einem Stau zu hoher Verkehr auftritt

Beispielablauf der Staukontrolle

Der Einfachheit halber stellen wir hier die Fenstergröße in Paketen dar.
Tatsächlich wird sie bei TCP in Byte angegeben.



7.1.3.5. TCP: Fast Retransmit, Fast Recovery

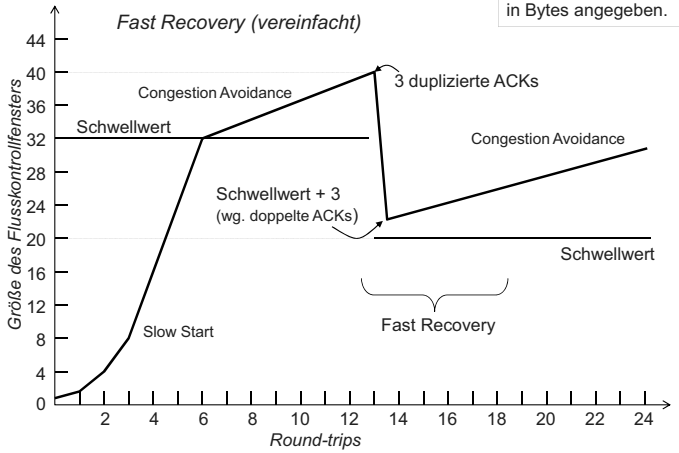
- **Fast Retransmit**
  - 3 duplizierte ACKs (also insg. 4) für Segment n
    ➔ Fehler erkannt ohne auf Timer zu warten
    ➔ erneute Übertragung (Retransmit) von Segment n
  - sonst wie bei Timeout

- **Fast Recovery**
  - Fehlererkennung wie bei Fast Retransmit
  - kein Slow-Start!
  - Geht direkt über in Congestion Avoidance

```
Schwellwert = 0.5 CongestionWindow
CongestionWindow = Schwellwert + 3
Bei ACK mit neuer Sequenznummer (also nicht-dupliziertes ACK)
  CongestionWindow = Schwellwert
Übergang in Congestion Avoidance
```


Beispielablauf der Staukontrolle

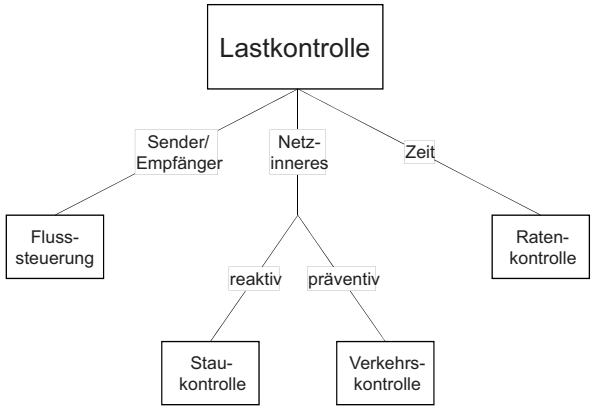
Der Einfachheit halber stellen wir hier die Fenstergröße in Paketen dar.
Eigentlich wird sie bei TCP in Bytes angegeben.



7.1.4. Ratenkontrolle

- Um die Zeit für rücklaufende Quittungen einzusparen und dennoch dem Sender nicht zu viele Freiheiten bei der Lasterzeugung zu lassen, wurde die Ratenkontrolle eingeführt:
  - Der Sender verhandelt mit dem Netz und dem Empfänger eine bestimmte Rate, d.h. eine Datenmenge pro Zeitintervall, die er schicken darf.
  - Er darf während des Zeitintervalls beliebig Daten senden, solange er die maximale Datenmenge nicht überschreitet.
  - Nach Ablauf des Zeitintervalls kann der Sender wieder über die maximale Datenmenge verfügen.

Zusammenfassung



7.2. Dienstgüte - Quality of Service (QoS)

- „**Quality of Service** is the collective effect of service performance, which determine the degree of satisfaction of a user of the service“
- ITU-T Recommendation E.800
  - Von der Zufriedenheit des Benutzers hängt die Güte des Dienstes (QoS) ab - jedoch fehlt die technische Überprüfbarkeit
  - Notwendig:
    - Dienstgüteparameter: beschreiben qualitative Eigenschaften eines Dienstes
    - Dienstklassen: beschreiben den Grad der Garantien
    - Dienstverträge: legen die zu garantierenden QoS-Parameter, deren Werte und den Grad der Einhaltung (Dienstklasse) fest.
    - Dienstgütemechanismen: Maßnahmen zur Einhaltung von Dienstverträgen
    - Management der Dienste: Verwaltung und Reservierung von Ressourcen

7.2.1. Dienstgüteparameter (QoS-Parameter)

- **Leistungsorientierte Dienstgüteparameter:**
  - Verzögerung
    - Ende-zu-Ende-Verzögerung (Delay)
    - Schwankung der Ende-zu-Ende-Verzögerung (Jitter)
  - Durchsatz:
    - min./mittl./max. Durchsatz (in [bit/s] oder [Pakete/s])
    - max. Länge [Pakete] oder Dauer [s] von Lastspitzen (Bursts)
- **Zuverlässigkeitsorientierte Dienstgüteparameter:**
  - Medienabhängige Fehlerraten:
    - z.B.: Bitfehlerrate des Übertragungsmediums
  - Ressourcenabhängige Fehlerraten:
    - z.B.: Paketverlustraten in den Warteschlangen der Zwischensysteme
- **Funktionale Dienstgüteparameter:**
  - Sicherheit
  - Gruppenkommunikation (Multicast)

Ressourcen

- **Ressourcenarten:**
  - Netzwerkressourcen
    - Übertragungskapazität (Bandbreite, Kanäle)
    - Übertragungszeit
  - (Zwischen- bzw. End-) Systemressourcen
    - Pufferspeicher
    - Rechenzeit
- **Reservierung von Ressourcen:**
  - durch Ressourcenmanagement
    - Verwaltung von Ressourcen (Belegung, Überwachung, Freigabe, ...)
  - Verteilung der Reservierungsnachrichten durch Reservierungsprotokolle
    - RSVP (Ressource ReserVation Protocol)
    - NSIS (Next Steps In Signalling protocol)

7.2.2. Dienstklassen (QoS-Klassen)

- **Deterministische Klasse:**
  - vorgegebene Schranken der QoS-Parameter werden exakt eingehalten
  - Ressourcen stehen einem Nutzer exklusiv zur Verfügung
  - keine Konflikte möglich, aber „Besetztfall“ (keine Ressourcen mehr übrig)

- **Statistische Klasse:**
  - vorgegebene Schranken müssen mit einer gewissen Wahrscheinlichkeit eingehalten werden
    z.B.: die Ende-zu-Ende-Verzögerung muss für 95% der Pakete unter 100ms liegen.
  - Ressourcen werden bis zu einem gewissen Grad überbelegt
  - Konflikte möglich (je höher die Wahrscheinlichkeit der Garantie, desto geringer sind Ressourcenkonflikte)

- **„Best Effort“-Klasse („so gut es geht“):**
  - es werden keinerlei Garantien für Dienstgüteparameter gemacht
  - keine explizite Ressourcenreservierung für einzelne Verbindungen

7.2.3. Dienstgütemechanismen

- **Dienstgütemechanismen** dienen der Einhaltung und Überwachung von Dienstgüteparametern.

- Man unterscheidet deshalb:
  - Verkehrsmeter (Messen des Verkehrs)
    - Token Bucket
    - Average Rate Meter
  - Verkehrsformer (Beeinflussen/Formen des Verkehrs)
    - Verkehrsglätter (Traffic Shaper)
      - Token Bucket → gibt durchschnittliche Rate und Burst-Größe vor
      - Leaky Bucket → gibt maximale Rate vor
    - Verwerfer (Dropper)
    - Degradierer (Verringern der Dienstgüte oder -klasse)
    - Ändern des Warteschlangen- oder Prozess-Schedulings

7.2.4. QoS-Architekturen

- Eine **QoS-Architektur** definiert:
  - QoS-Parameter
  - Ressourcen
  - Ressourcenreservierung
  - Ressourcenmanagement
  - Dienstklassen
  - Dienstgütemechanismen

- Beispiele für QoS-Architekturen:
  - ATM
  - Integrated Services
  - Differentiated Services

Lehrstuhl für Netzarchitekturen und Netzdienste
Institut für Informatik – Technische Universität München
Prof. Dr.-Ing. Georg Carle

Grundlagen: Rechnernetze und Verteilte Systeme

Kapitel 8: Anwendungen

Netzmanagement, SNMP, SMTP, HTTP, DNS

Prof. Dr.-Ing. Georg Carle
Lehrstuhl für Netzarchitekturen und Netzdienste
Technische Universität München
carle@net.in.tum.de
http://www.net.in.tum.de

Übersicht

1. Einführung und Motivation
   - Bedeutung, Beispiele
2. Begriffswelt und Standards
   - Dienst, Protokoll, Standardisierung
3. Direktverbindungsnetze
   - Fehlererkennung, Protokolle
   - Ethernet
4. Vermittlung
   - Vermittlungsprinzipien
   - Wegwahlverfahren
5. Internet-Protokolle
   - IP, ARP, DHCP, ICMP
   - Routing-Protokolle
6. Transportprotokolle
   - UDP, TCP
7. Verkehrssteuerung
   - Kriterien, Mechanismen
   - Verkehrssteuerung im Internet
8. **Anwendungsorientierte Protokolle und Mechanismen**
   - **Netzmanagement**
   - **DNS, SMTP, HTTP**
9. Verteilte Systeme
   - Middleware
   - RPC, RMI
   - Web Services
10. Netzsicherheit
    - Kryptographische Mechanismen und Dienste
    - Protokolle mit sicheren Diensten: IPSec etc.
    - Firewalls, Intrusion Detection
11. Nachrichtentechnik
    - Daten, Signal, Medien, Physik
12. Bitübertragungsschicht
    - Codierung
    - Modems

8. Anwendungen - Kapitelgliederung

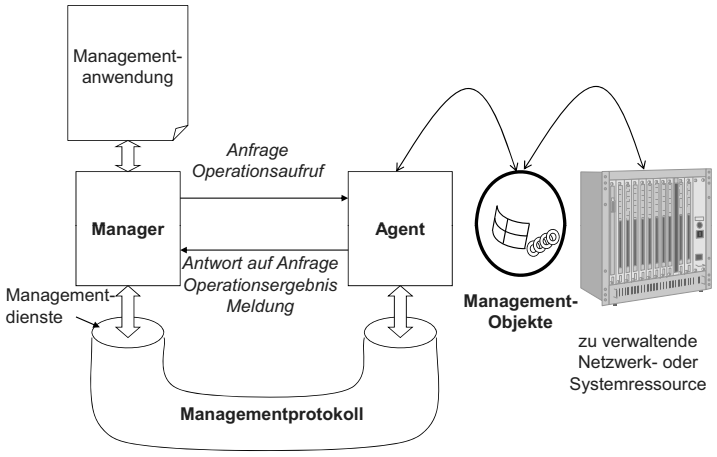
Netzmanagement: Arten und Ursachen von Netzwerkproblemen

- **Ausfall** von Zwischen- oder Endsystemen, phys. Medien
  - Ursache: Totalausfall, Technischer Defekt
- **Fehlfunktion** von Zwischen- oder Endsystemen, Medien, Medienanschluss
  - Ursache: Teilausfall von Funktionen, intermittierende Fehler
- **Überlastung** von Zwischensystemen oder (Sub-)Netzen
  - Ursache: Fehldimensionierung, steigendes Datenaufkommen
- **Fehlkonfiguration** von Zwischen- oder Endsystemen
  - Ursache: mangelnde Erfahrung, Fehleinschätzung, Flüchtigkeitsfehler
- **Angriffe** auf Netze oder Netzkomponenten
  - Ursache: mutwillig oder fahrlässig

Aufgaben und Ziele für das Netzwerkmanagement

- Zustandsüberwachung einzelner Netzkomponenten
- Steuerung des Netzbetriebs
- Sicherung eines effizienten und effektiven Betriebs
- Abrechnung der Netzressourcen-Benutzung
- gesicherter (geschützter) Netzbetrieb
- einfache Modellierung eines Netzwerks
- Gewinnung von Planungsdaten für das Netz und Netzwerkmodifikationen
- Planung „managebarer“ (verwaltbarer) Netze

Netzmanagement in der Übersicht

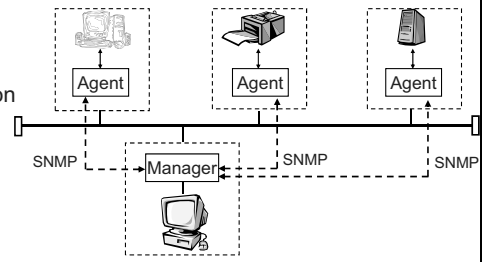


Management im Internet

- Managementstandard im Internet (OSI-Standards nicht kompatibel)
  - benutzt das *Simple Network Management Protocol* **(SNMP)**
  - verbindungslos, basiert auf UDP
  - definiert eine einfache Struktur zur Modellierung von zu verwaltenden Ressourcen mit Hilfe der *Structure of Management Information* **(SMI)**
  - umfasst mehrere standardisierte Sammlungen von *Managed Objects*, sogenannte *Management Information Bases* **(MIB)**
- SNMP ist insbesondere in lokalen Netzen weit verbreitet
- Erste Version wurde 1990 erarbeitet, die aktuelle Version ist SNMPv3 nach RFC 2570 (1999). Neu ist hier vor allem die Unterstützung für Sicherheit (viele Produkte basieren noch auf SNMPv1).

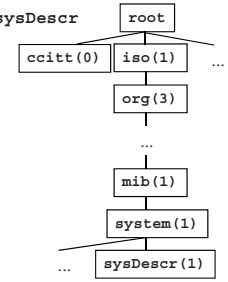
SNMP (Simple Network Management Protocol)

- SNMP dient der Verwaltung (dem Management) beliebiger Netzwerk-Ressourcen
  - z.B Drucker, Brücken, Router, Endsysteme usw.
- Verwaltete Ressourcen integrieren SNMP-Agenten (Software-Prozess)
  - Die Agenten verwalten die Managementinformationen der Komponente
    - z.B. Anzahl eingegangener/verlorener Pakete
- Der Manager (Software-Prozess) dient der Kommunikation mit den Agenten
  - Protokoll: SNMP (verwendet UDP)
- Basis der Kommunikation zwischen Manager und Agent: Managementobjekte



Managementobjekte (Managed Objects)

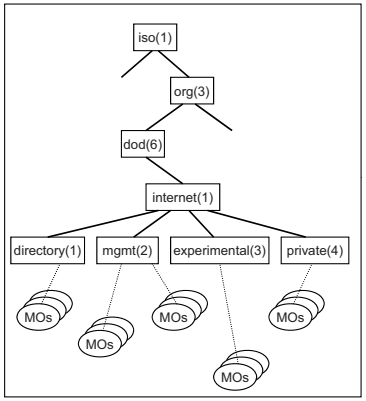
- Managementobjekt (Managed Object):
  - Modell (Abbild) einer/mehrerer Eigenschaft(en) einer Netzwerkressource
  - Ein Agent verwaltet die Managementobjekte „seiner“ Ressource
  - Bestandteile eines Managementobjekts im Internet:
    - Eindeutiger Name, z.B. **`iso.org.dod.internet.mgmt.mib.system.sysDescr`**
    - Syntax: verschiedene einfache Datentypen, z.B. Integer, String, Array
    - Zugriffsrechte, z.B. read-only, read-write
    - Status, z.B verpflichtend (mandatory), optional



- Management Information Base (MIB):
  - Gesamtheit aller Managementobjekte
  - verteilte, virtuelle Datenbank
- Management Information Tree (MIT):
  - Jedes Managementobjekt hat eindeutige Position im MIT
  - Somit eindeutige Bezugnahme möglich

Modellierung von Managementinformation: MIB und SMI

Management Information Base (MIB) | Structure of Management Information (SMI)



Beispiel: Object sysDescr

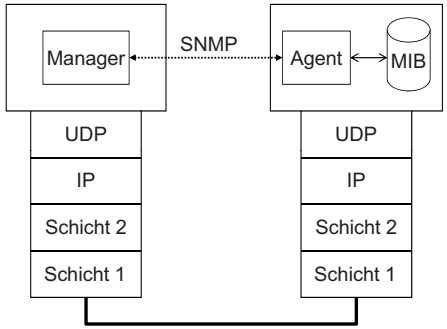
```
sysDescr OBJECT-TYPE

  SYNTAX  DisplayString (SIZE (0..255))

  ACCESS  read-only

  STATUS  current

  DESCRIPTION
    "A textual description of the entity.  This value
    should include the full name and version
    identification of the system's hardware type,
    software operating-system, and networking
    software.  It is mandatory that this only contains
    printable ASCII characters.“

  ::= { system 1 }
```


RMON – Remote Monitoring

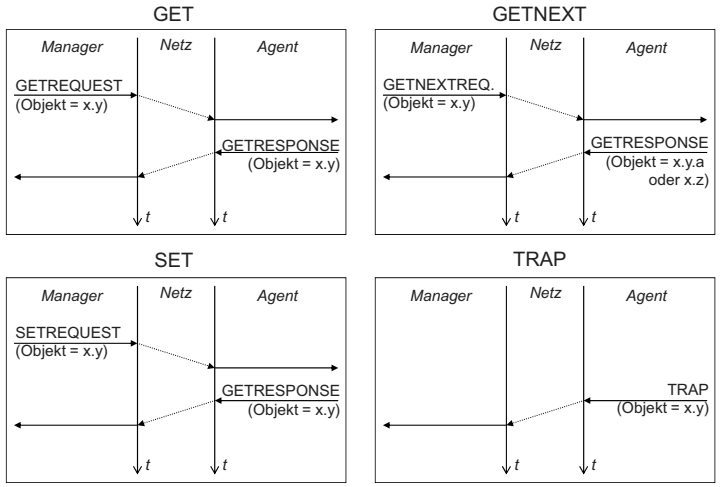
- MIB mit speziellen Fähigkeiten (RFC 1757 u.a.)
  - Sammeln von Statistiken, Alarmen, Ereignissen
  - Teilweise Auswertung, Filtern, Packet-Capture, ...
  - ➔ Verlagerung von „Intelligenz“ von der Managementplattform weg zum Agent
- Beispiel SMI:

```
etherStatsOversizePkts OBJECT-TYPE
  SYNTAX Counter
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION "The total number of packets received that were longer
     than 1518 octets (excluding framing bits, but including FCS
     octets) and were otherwise well formed."
  ::= { etherStatsEntry 10 }
```


Internet-Netzwerkmanagement



SNMP: Client-Server-Prinzip



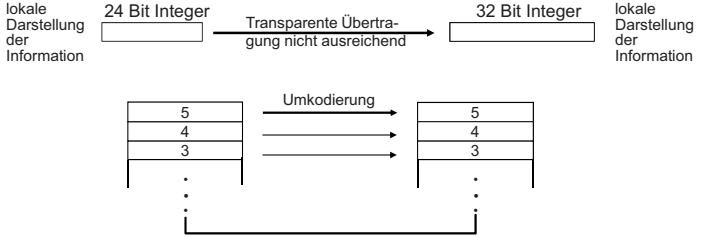
Managementanwendungen: Komandozeilen-Tools

- Public Domain Tools (diverse Versionen):
  - Befehle: *snmpget*, *snmpnext*, *snmpwalk*, *snmpset*, evtl. weitere
  - Erzeugung und Dekodierung von SNMP-Dateneinheiten
  - teilweise auch mit Unterstützung für MIB-Dateien

```
snmpget -v 1 129.13.35.239 public .1.3.6.1.2.1.1.1.0
system.sysDescr.0 = "GIGAswitch Network Platform"

snmpwalk -v 1 129.13.35.239 public .1.3.6.1.2.1.1
system.sysDescr.0 = "GIGAswitch Network Platform"
system.sysObjectID.0 = OID: enterprises.DEC.2.15.3.3
system.sysUpTime.0 = Timeticks: (456990767) 52 days, 21:25:07
system.sysContact.0 = "wiltfang@telematik.informatik.uni-karlsruhe.de"
system.sysName.0 = ""
system.sysLocation.0 = ""
system.sysServices.0 = 10
```


Heterogene Darstellungen

- **Problem:**
  - Unterschiedliche Rechnersysteme besitzen verschiedene interne Darstellungen (Little/Big Endian, 16-/24-/32-Bit usw.)
  - Konsequenz: Umkodieren der zu übermittelnden Daten ist erforderlich ⇨ Austauschstandards notwendig
- **Aufgaben:**
  - Behandeln der Darstellung (Syntax) von Informationen
  - Bewahren der Bedeutung (Semantik) der Informationen



ASN.1: Definition

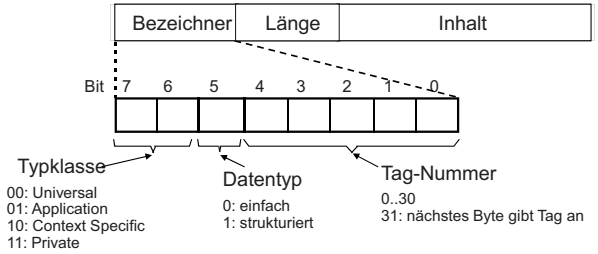
- ASN.1 (abstrakte Syntax-Notation eins) ist eine von der ISO genormte Beschreibungssprache zu darstellungsunabhängigen Spezifikation von Datentypen und Werten
  - findet z.B. zur Definition von Managementobjekten bei SNMP Verwendung
- Elementare Datentypen:
  - Boolean, Integer, Bitstring, Octetstring, IA5String, …
- Strukturierte Datentypen:
  - SEQUENCE: Geordnete Liste von Datentypen (Record in PASCAL)
  - SET: Ungeordnete Menge von Datentypen
  - SEQUENCE OF: Geordnete Liste von Elementen des gleichen Datentyps (Array in C)
  - SET OF: Ungeordnete Menge von Elementen des gleichen Datentyps
  - CHOICE: Ungeordnete Menge von Datentypen, aus der einige Datentypen ausgewählt werden können (Union in C)

Beispiel:

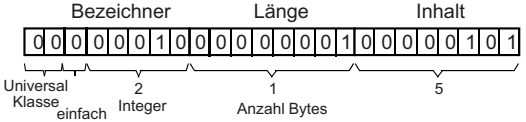
```
Mitarbeiter ::= SET {      Name IA5String,
                           Alter Integer,
                           Personalnr Integer }
```

ASN.1: Übertragungssyntax

Basic Encoding Rules, BER (Übertragungssyntax):



Beispiel: Integer Wert 5



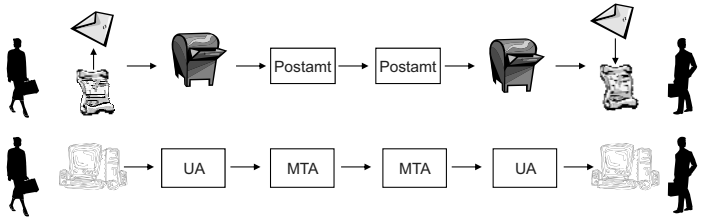
Elektronische Post (E-Mail)

- Hauptziel:
  - Internationaler Austausch elektronischer Mitteilungen zwischen Personen
- Wesentliche Charakteristik:
  - Unterstützung eines asynchronen Verhaltens von Sender und Empfänger
  - Speichervermittlung
- Allgemeine Basisfunktionen:
  - Erstellen von E-Mails
  - Übertragung zum Ziel
  - Benachrichtigung im Erfolgs-/Fehlerfall
  - Anzeige erhaltener Nachrichten
  - Speicherung von Nachrichten

- Realisierung:
  - Simple Mail Transfer Protocol (SMTP) im Internet
  - Ehemals X.400 bei OSI
    X.400: C=DE, A=DBP, P=BWL, O=BWLMWK,
    S=Wissenschaftsministerium

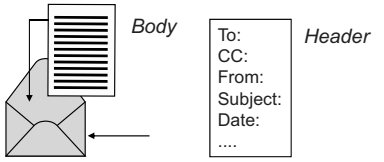
E-Mail: Allgemeines Modell

- User Agent (UA)
  - Lokales, grafik-/textorientiertes Programm
  - Ermöglicht Lesen und Versenden von E-Mail vom lokalen Rechner
  - z.B. Elm, Mail, Outlook, Thunderbird
- Message Transfer Agent (MTA)
  - Hintergrundprozess
  - Zuständig für das Weiterleiten von E-Mails zum Zielrechner



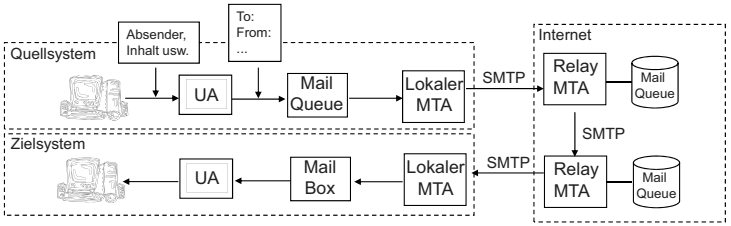
SMTP: Format einer E-Mail

- Umschlag (Envelope)
  - Enthält alle Informationen für den Transport der Mitteilung zum Empfänger
  - Adressierung erfolgt mithilfe von DNS, z.B. g.carle@**ieee.org**
  - Wird interpretiert von den MTAs
- Kopfteil (Header)
  - Optional, aber meist vorhanden
  - Enthält Informationen zu Absender und Empfänger („From:“, „To:“) sowie zusätzliche Felder wie z.B. Betreff, Kopie an („Subject:“, „CC:“)
  - Zusätzliche Informationen, die von den MTAs hinzugefügt werden
  - Interpretiert von den UAs
- Hauptteil (Body)
  - Enthält den eigentlichen Inhalt der Mitteilung (ursprünglich nur ASCII)



Internet Mail: Das SMTP-Modell

- SMTP dient der E-Mail-Übermittlung
  - zeichenorientiertes Protokoll, basierend auf 7-Bit-ASCII
  - nur wenige Kommandos, z.B. HELO, MAIL, RCPT, DATA, QUIT
- UA erhält alle notwendigen Angaben vom Benutzer
  - Mitteilung wird über Mail-Queue zum lokalen MTA übertragen
- MTAs übertragen die Mitteilung zum Zielrechner
  - Auslieferung einer E-Mail erfolgt über eine TCP-Verbindung (Port 25) zum Ziel-MTA (populärer MTA unter UNIX: sendmail)
  - Relay-MTAs dienen als zentrale E-MAIL-Verteiler (z.B. Informatik-Institut)



SMTP: Beispielablauf

```
> telnet smtpserv.uni-tuebingen.de 25
Trying 134.2.3.3...
Connected to smtpserv.rr.uni-tuebingen.de.
Escape character is '^]'.
220 mx06.uni-tuebingen.de ESMTP Sendmail 8.13.6/8.13.6; Fri, 2 Feb 2007 10:58:49 +0100
HELO metz.informatik.uni-tuebingen.de
250 mx06.uni-tuebingen.de Hello rouen.informatik.uni-tuebingen.de [134.2.11.152], pleased
    to meet you
MAIL FROM carle@informatik.uni-tuebingen.de
501 5.5.2 Syntax error in parameters scanning "FROM"
MAIL FROM: carle@informatik.uni-tuebingen.de
250 2.1.0 carle@informatik.uni-tuebingen.de... Sender ok
RCTP TO: g.carle@ieee.org
500 5.5.1 Command unrecognized: "RCTP TO: g.carle@ieee.org"
RCPT TO: g.carle@ieee.org
250 2.1.5 g.carle@ieee.org... Recipient ok
DATA
354 Enter mail, end with "." on a line by itself
From: carle@informatik.uni-tuebingen.de
To: g.carle@ieee.org
Subject: Test

Hallo, dies ist eine Testmail.

.
250 2.0.0 l129wnPo027024 Message accepted for delivery
QUIT
221 2.0.0 mx06.uni-tuebingen.de closing connection
Connection closed by foreign host.
```

Angabe des (falschen) Rechnernamens wird ignoriert und stattdessen DNS-Name verwendet.

Fehlermeldungen nach falscher Eingabe

Optionaler Mail-Header.
• Falls „From:..." fehlt, wird die Absenderangabe aus „MAIL FROM:..." verwendet.
• Falls „To:..." fehlt, bleibt das Feld leer.

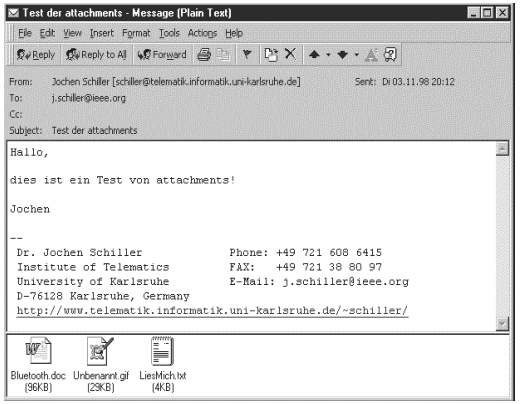
Wie kommen Blind Copies (BCC:) zustande?

MIME (Multipurpose Internet Mail Extensions)

- SMTP sieht nur einfache ASCII-Texte als Nachrichten vor (im Hauptteil)
- MIME erweitert den Hauptteil einer Nachricht um Formatinformationen. Hierzu werden neue Datenfelder für den Kopfteil einer Nachricht definiert:
  - Content-Type: definiert den Typ des Hauptteils.
    - Text, Multipart, Message, Application (Binary), Image, Audio, Video und X-private
  - Content-Transfer-Encoding: definiert die Transfer-Syntax, in der die Daten des Hauptteils übertragen werden.
    - Base 64, Quoted Printable, 7 Bit, 8 Bit und Binary
- Weitgehende Kompatibilität zur herkömmlichen Internet-Mail:
  - Mit der Transfersyntax Base 64 ist es möglich, Binärdaten durch Subnetze zu leiten, die nur die Übertragung von 7-Bit-ASCII-Texten erlauben.
  - Die Transfersyntax Quoted Printable erlaubt nationale Sonderzeichen. Wird eine solche Mail von einem „normalen" Mail User Agent (Mail-Client-Programm) angezeigt, so werden nur diese Erweiterungen verstümmelt.

MIME - Beispiel

- Email mit Text, Word-Dokument, Bild und Text-Anhang
- Ansicht in MS-Outlook:



MIME - Standard Email-Kopf: Adressen & Co

```
From: "Jochen Schiller" <schiller@telematik.informatik.uni-
   karlsruhe.de>
To: <j.schiller@ieee.org>
Subject: Test der attachments
Date: Tue, 3 Nov 1998 20:11:41 +0100
Message-ID:
   <001701be075d$ca257d00$732a0d81@tpc15.telematik.informatik.u
   ni-karlsruhe.de>
MIME-Version: 1.0
Content-Type: multipart/mixed;
   boundary="----=_NextPart_000_0018_01BE0766.2BE9E500"
X-Priority: 3 (Normal)
X-MSMail-Priority: Normal
X-Mailer: Microsoft Outlook 8.5, Build 4.71.2377.0
X-MimeOLE: Produced By Microsoft MimeOLE V4.72.2120.0
Importance: Normal

This is a multi-part message in MIME format.
```

MIME

proprietär

MIME - 1. Teil: Textnachricht

```
------=_NextPart_000_0018_01BE0766.2BE9E500
Content-Type: text/plain;
   charset="iso-8859-1"
Content-Transfer-Encoding: 7bit

Hallo,

dies ist ein Test von attachments!

Jochen
--
 Dr. Jochen Schiller              Phone: +49 721 608 6415
 Institute of Telematics          FAX:   +49 721 38 80 97
 University of Karlsruhe          E-Mail: j.schiller@ieee.org
 D-76128 Karlsruhe, Germany
 http://www.telematik.informatik.uni-karlsruhe.de/~schiller/
```


MIME - 2. Teil: Word-Anhang

```
------=_NextPart_000_0018_01BE0766.2BE9E500
Content-Type: application/msword;
   name="Bluetooth.doc"
Content-Transfer-Encoding: base64
Content-Disposition: attachment;
   filename="Bluetooth.doc"

0M8R4KGxGuEAAAAAAAAAAAAAAAAAAAAAPgADAP7/CQAGAAAAAAAAAAAAAACAAAAtgAAAA
   AAAAA
pcEAWQAHBAAACBK/AAAAAAAAEAAAAAAABAAAP2sAAA4AYmpiavNX81cAAAAAAAAAAAAAA
   AAAAAA

...

AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
   AAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA=
```


S/MIME – Sichere und signierte E-Mail

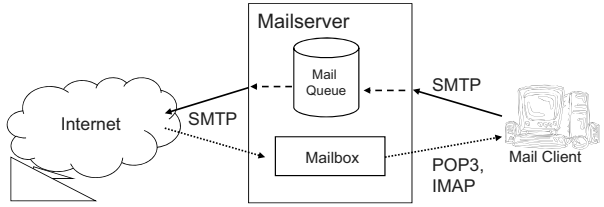
- Integration von Signatur und Verschlüsselung in E-Mail-Clients

  - Neue Content-Types, z.B. Signatur:

```
Content-Type: multipart/signed; micalg=SHA1;
  protocol="application/x-pkcs7-signature";
  boundary="----=_NextPart_000_0017_01C01BD5.
   D86ED4E0"
```

  - Einfache Überprüfung von
    - Gültigkeit
    - Aussteller des Zertifikats
    - Besitzer des Zertifikats

Internet-Mail: Verwaltung durch Mailserver



- E-Mail wird meist zentral über einen Mailserver abgewickelt (Relay-MTA)
- Mittels **POP3** (Post Office Protocol 3) holt der Client die vom Mailserver empfangenen und in der Mailbox gespeicherten Meldungen ab
  - einfache Funktionalität
- **IMAP** (Internet Message Access Protocol) dient dazu, die Nachrichten zentral auf einem Mailserver zu verwalten
  - IMAP erlaubt erweiterte Kommandos (z.B. Filterung)
- Beispiele für Mail-Client-Programme
  - Outlook Express (Microsoft)
  - Messenger (Netscape)

FTP (File Transfer Protocol)

- Aufgabe: Dateiübertragung zwischen entfernten Rechnern
  - Ziel: Datenaustausch zwischen heterogenen Rechnerarchitekturen
  - FTP-Protokoll erlaubt, dass sich die Rechner auf einen geeigneten Übertragungsmodus einigen
  - Dateizugriff wird NICHT unterstützt (Attribute ändern, löschen, etc.)
  - FTP-Instanz auf TCP-Port 21
  - Dateiübertragung über kurzzeitig zugewiesene Ports
  - ASCII-Kommandos zur Ablaufsteuerung (z.B. GET, PUT)
- FTP-Optionen:
  - Datentyp (7-Bit-ASCII, EBCDIC, Image/Binary (Bitstrom), Local)
  - Dateistrukturen (File (Bytestrom), Record, Page)
  - Übertragungsmodus (Stream, Block, Compressed)
- FTP-Dienste:
  - Verbindungsaufbau mit Authentifizierung (Passworteingabe)
  - Dateiübertragung (z.B. put, get)
  - Operationen auf Dateisystem (z.B. cd, dir)
  - Hilfefunktionen (z.B. Kommando-Auflistung inkl. Parameter)
  - Weitere implementierungsabhängige Dienste möglich

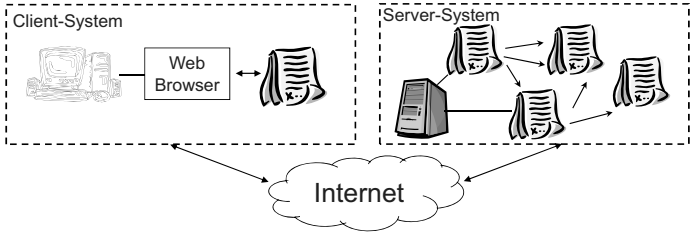
Zur Entwicklung des World Wide Web (WWW)

- Hervorgegangen aus Arbeiten des britischen Informatikers Tim Berners-Lee am europäischen Forschungszentrum CERN (Genf)
  - Ziel: Einfacher weltweiter Austausch von Dokumenten zwischen Wissenschaftlern

- Erster Prototyp Ende 1990
  - grafisch (auf NEXTStep) und zeilenorientiert
- Durchbruch des WWW durch den WWW-Client Mosaic
  - entwickelt von Marc Andreesen und Eric Bina (NCSA at UIUC: National Center for Supercomputer Applications at Univ. of Illinois Urbana-Champaign)
  - ursprünglich für X-Windows-Systeme
  - als Quellcode per FTP kostenlos verfügbar ⇨ schnelle Verbreitung
  - Marc Andreesen gründete 1995 die Firma Netscape
- Gründung des W3-Konsortiums im Juli 1994
  - vorrangiges Ziel: Weiterentwicklung des WWW, z.B. durch Standardisierung von HTML
  - Vorsitzender: Tim Berners-Lee
  - Infos unter http://www.w3.org

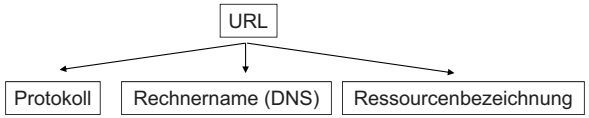
Client/Server-Architektur des WWW

- Client-Server-Architektur
  - Web-Browser zur Anzeige von Hypertext-Dokumenten/Hypermedia-Objekten
  - Hyperlinks erlauben Navigation
- Lösungen zu folgenden Fragestellungen erforderlich:
  - Eindeutige Adressierung einer Web-Seite
  - Transport einer Web-Seite
  - Beschreibung des Inhalts (insbes. der Hyperlinks) einer Web-Seite



Adressierung eines Web-Dokuments

- Uniform Resource Locator (URL)
  - weist der Client-Software den Weg zu einer bestimmten Ressource
  - auch für Inhalte anderer Server (USENET, FTP, E-Mail) verwendbar
  - z.B. http://www.informatik.uni-tuebingen.de/index.html



- Durch die Ressourcenbezeichnung wird das Objekt, auf das im jeweiligen Server zugegriffen werden soll, identifiziert
  - bei WWW: abgerufene Web-Seite
  - bei FTP: zu übertragene Datei
  - bei Mail: Empfänger der Mail
- Web-Browser unterstützen eine Vielzahl von Protokollen
  - z.B. http://, ftp://, mailto://, telnet://, soap://

Das WWW-Anwendungsprotokoll: HTTP

- HTTP (HyperText Transport Protocol)
  - Version 0.9 und 1.0 im RFC1945 beschrieben
  - seit Januar 1997 existiert eine Version 1.1 (RFC2068)
  - dient im Wesentlichen der Übertragung von Web-Seiten
- Wesentliche Eigenschaften
  - ASCII-Anwendungsprotokoll
  - setzt auf eine (sichere) TCP-Verbindung auf
  - Default-Port: 80
  - kurzlebige Verbindung, da der HTTP-Server nach Beantwortung einer Anfrage durch den HTTP-Client die Verbindung sofort schließt
- Beispiele von Befehlen:
  - `GET`: Anfordern eines bestimmten Dokuments
  - `HEAD`: Anfordern von Informationen im Kopfteil eines Dokuments
  - `POST`: Anhängen von Daten an ein existierendes Dokument
  - `PUT`: Anlegen eines Dokuments

Beispiel einer HTTP-Anfrage und HTTP-Antwort

HTTP-Client ➔ HTTP-Server:

```
GET /index.html HTTP/1.1
Host:www.informatik.uni-tuebingen.de
Pragma: no-cache
....
```

- Befehlszeile: <Befehl> <URL> <Version>
- Client wünscht nicht zwischengespeicherte, d.h. aktuelle Version des Dokuments

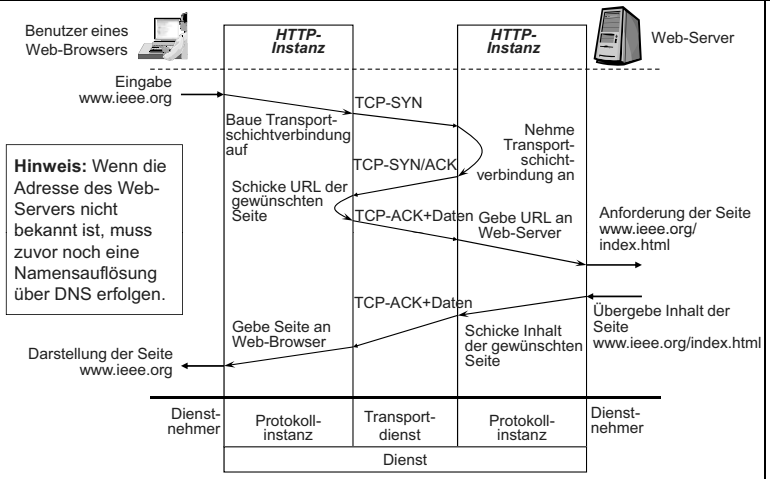
Hinweis: Verbindung zwischen Client und Server wurde bereits zuvor aufgebaut

HTTP-Server ➔ HTTP-Client

```
HTTP/1.1 200 OK
Date: Fri, 24 Sep 1999 09:45:51 GMT
Server: Apache/1.3.6 (Unix)
Transfer-Encoding: chunked
Content-Type: text/html

<HTML>
Gemäß HTML-Konventionen
strukturiertes Dokument
</HTML>
```

- Antwort-Zeile
- Datum
- Server
- Angaben zur Kodierung
- Art des Inhalt
- Hauptteil

Beispiel: Surfen im Internet



DNS (Domain Name System)

- DNS stellt eine weltweit verteilte Namensdatenbank dar
- DNS besitzt eine hierarchische Namenstruktur
- DNS bildet Namen auf Information (z.B. IP-Adressen) ab
  - **`www.ieee.org`** ⇨ **`199.172.136.40`**
  - Vorteil: Information/Adresse, auf die abgebildet wird, kann sich ändern
  - Bsp: Mailserver von IEEE (z.B. für g.carle@ieee.org)

MX-Record → NS-Records → A-Records

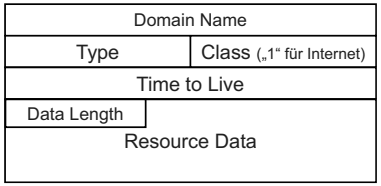
```
$ nslookup -q=mx ieee.org
Server:  sioux.telematik.informatik.uni-karlsruhe.de
Address:  129.13.35.73

ieee.org        preference = 0, mail exchanger = gemini.ieee.org
ieee.org        nameserver = auth01.ieee.org
ieee.org        nameserver = dns.ieee.org
ieee.org        nameserver = ns.uu.net
ieee.org        nameserver = krypton.ieee.org
ieee.org        nameserver = deepthought.ieee.org
gemini.ieee.org internet address = 199.172.136.14
auth01.ieee.org internet address = 199.172.136.2
dns.ieee.org    internet address = 199.172.136.6
ns.uu.net       internet address = 137.39.1.3
krypton.ieee.org        internet address = 199.172.136.2
deepthought.ieee.org    internet address = 199.172.136.6
```


DNS Resource Records

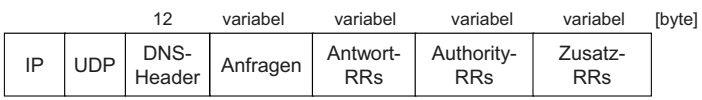
- Das DNS basiert auf dem Austausch sog. *Resource Records*:

Typ	Beschreibung
A (Address)	Abbildung: Name auf IP-Adresse
MX (Mail Exchange)	E-Mail-Server einer Domäne
NS (Nameserver)	Nameserver einer Domäne
CNAME (Canonical Name)	„Alias"-Namen für Rechner/Domäne
PTR (Pointer)	Abbildung: IP-Adresse auf Name
HINFO (Host Info)	zusätzliche Informationen (CPU,...)

- Format:



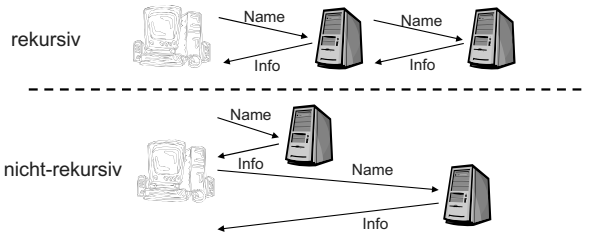
DNS: Paketformat



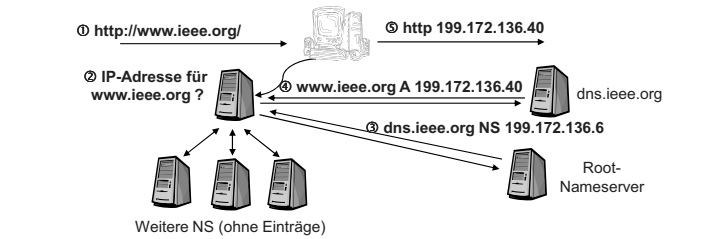
- DNS verwendet UDP
  - effizient, da kein Verbindungsaufbau/Datensicherung notwendig
- DNS-Header:
  - enthält Identifikationsnummer der Anfrage, Anzahl der Einträge in den folgenden Feldern, weitere Steuerinformationen (z.B. für Rekursion)
- Anfragen:
  - bestehend aus DNS-Namen (z.B. www.amazon.de) und Typ der Anfrage (z.B. A, MX, PTR)
- Antwort-/Authority-/Zusatz-RRs
  - ein/mehrere Resource Records mit gewünschten DNS-Informationen
  - Authority-RRs: Name(n) des(r) verantwortlichen (d.h. für diese Anfrage zuständigen) Nameserver(s)

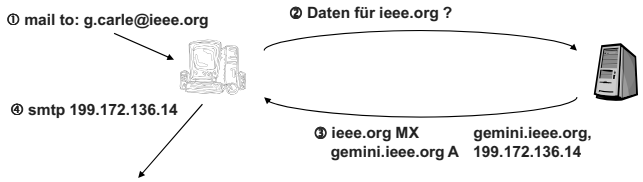
DNS: Nameserver (NS)

- Jede Zone hat einen primären und beliebig viele sekundäre Nameserver
  - Jeder NS kennt nur einen Ausschnitt des gesamten Namensraums
  - Jeder NS kennt die IP-Adressen der NS seiner direkt untergeordneten Sub-Domains
  - Jeder NS führt Caching bereits bekannter Einträge durch
  - Sekundäre NS führen ein periodisches Update („Zonentransfer") ihrer Datenbasis durch (basierend auf den Daten des primären NS)
- Anfragen können rekursiv oder nicht-rekursiv beantwortet werden:



DNS: Beispiele (A-/MX-Records)





DNS: Reverse Lookup

- Aufgabe des Reverse Lookup:
  - Bestimmung des logischen Namens zu einer gegebenen IP-Adresse
  - hierzu Verwendung der PTR-Records
- Vorgehensweise:
  - Spezieller Teilbaum des DNS „`in-addr.arpa`“
    - dient der Zuordnung von IP-Adresse ⇨ Name
    - jede IP-Adresse entspricht Eintrag unterhalb `in-addr.arpa`
    - jede Stelle einer Adresse entspricht genau einem Knoten
  - eine Anfrage an das DNS enthält somit die IP-Adresse in „invertierter“ Form ⇨ hierarchische Strukturierung
- Beispiel:
  - IP-Adresse: `207.171.168.16`
    - DNS-Name in Anfrage: `16.168.171.207.in-addr.arpa`
    - Ergebnis (Antwort RR): Resource Data = `www.amazon.de`

Lehrstuhl für Netzarchitekturen und Netzdienste
Institut für Informatik – Technische Universität München
Prof. Dr.-Ing. Georg Carle

Grundlagen: Rechnernetze und Verteilte Systeme

Kapitel 9: Verteilte Systeme

Prof. Dr.-Ing. Georg Carle
Lehrstuhl für Netzarchitekturen und Netzdienste
Technische Universität München
carle@net.in.tum.de
http://www.net.in.tum.de

Acknowledgements: Prof. Dr. Wolfgang Küchlin, Tübingen

Übersicht

1. Einführung und Motivation
   - Bedeutung, Beispiele
2. Begriffswelt und Standards
   - Dienst, Protokoll, Standardisierung
3. Direktverbindungsnetze
   - Fehlererkennung, Protokolle
   - Ethernet
4. Vermittlung
   - Vermittlungsprinzipien
   - Wegwahlverfahren
5. Internet-Protokolle
   - IP, ARP, DHCP, ICMP
   - Routing-Protokolle
6. Transportprotokolle
   - UDP, TCP
7. Verkehrssteuerung
   - Kriterien, Mechanismen
   - Verkehrssteuerung im Internet
8. Anwendungsorientierte Protokolle und Mechanismen
   - Netzmanagement
   - DNS, SMTP, HTTP
9. **Verteilte Systeme**
   - **Middleware**
   - **RPC, RMI**
   - **Web Services**
10. Netzsicherheit
    - Kryptographische Mechanismen und Dienste
    - Protokolle mit sicheren Diensten: IPSec etc.
    - Firewalls, Intrusion Detection
11. Nachrichtentechnik
    - Daten, Signal, Medien, Physik
12. Bitübertragungsschicht
    - Codierung
    - Modems

Gliederung - Kapitel 9: Verteilte Systeme

9.1 Definition eines Verteilten Systems

- Ein verteiltes System ist eine Menge voneinander unabhängiger Computer, die dem Benutzer wie ein einzelnes, kohärentes System erscheinen.

  [Tanenbaum, van Steen: Verteilte Systeme, Pearson Studium, 2003]

Verteilte Systeme

Eigenschaften Verteilter Systeme

- Unterschiede zwischen den verschiedenen Computern werden verborgen.
- Benutzer und Applikationen können auf konsistente und einheitliche Weise mit dem verteilten System zusammenarbeiten.
- Verteilte Systeme sollen erweiterbar und skalierbar sein.

Beispiele

- Netzwerk aus Workstations mit gemeinsamen Dateidiensten und gemeinsamer Benutzerverwaltung
- Informationssystem für Arbeitsabläufe
- World Wide Web

Ziele für Verteilte Systeme

- Benutzer und Ressourcen verbinden
  - Den Benutzern ermöglichen, auf entfernte Ressourcen zuzugreifen
  - Unterstützung für kontrollierte gemeinsame Benutzung
- Transparenz
  - Zugriff – verbirgt Unterschiede in der Datendarstellung
  - Position – verbirgt Ort der Ressource
  - Migration – verbirgt Möglichkeit, Ressource an anderen Ort zu verschieben
  - Relokation – verbirgt Verschiebung von Ressource während Nutzung
  - Replikation – verbirgt, dass eine Ressource repliziert ist
  - Nebenläufigkeit – verbirgt gleichzeitige Nutzung konkurrierender Benutzer
  - Fehler – verbirgt Ausfall und Wiederherstellung einer Ressource
  - Persistenz – verbirgt Speicherart (Hauptspeicher oder Festplatte)

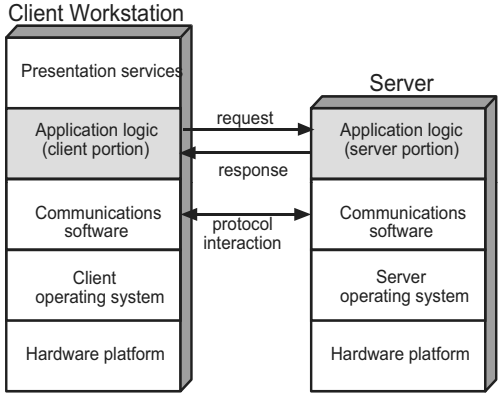
  ⇨Vor- und Nachteile von Transparenz
- Offenheit
  - Vollständige Schnittstellenspezifikation (⇨ Schnittstellendefinitionssprache IDL – Interface Description Language):
    Festlegen von Namen der verfügbaren Funktionen, Typen der Übergabeparameter und Rückgabewerte
- Skalierbarkeit

9.2 Middleware

- Ziele einer Middleware
  - Einführung einer **zusätzlichen Schicht** zwischen Betriebssystem und Anwendung, um **höhere Abstraktionsebene** zur Unterstützung verteilter Anwendungen zu erhalten.
  - Lokale Ressourcen einzelner Knoten sollen weiterhin vom (lokalen) Betriebssystem verwaltet werden.
- Mögliche Modelle (Paradigmen) für Middleware
  - Modell „**Datei**“: Behandlung aller lokaler und entfernter Ressourcen als Datei (Beispiele: Unix, Plan9)
  - Modell „**Prozeduraufruf**“: lokaler und entfernter Prozeduraufruf (Beispiel: RPC)
  - Modell „**verteilte Objekte**“: Objekte können lokal oder auf transparente Weise entfernt aufgerufen werden (Beispiel: RMI)
    - Schnittstelle besteht aus den Methoden, die das Objekt implementiert.
    - Verteiltes System kann so realisiert werden, dass sich ein Objekt auf einer Maschine befindet, seine Schnittstelle aber auf vielen anderen Maschinen bereitgestellt wird
  - Modell „**verteilte Dokumente**“: Dokumente mit verweisen, wobei Ort des Dokuments transparent ist (Beispiel: WWW)
  - Modell „**Nachrichtenorientierte Middleware**“: Nachrichtenwarteschlangen-systeme für persistente asynchrone Kommunikation

Die Rolle von Middleware

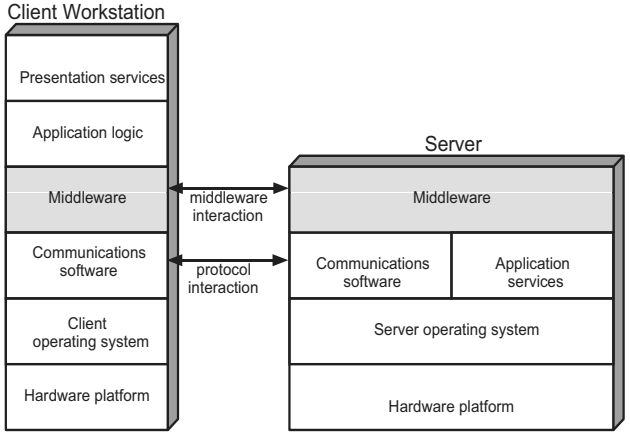
- Client/Server-Kommunikation **ohne** Middleware



Quelle: „Operating Systems“, Stallings, Abb.13-7

Die Rolle von Middleware

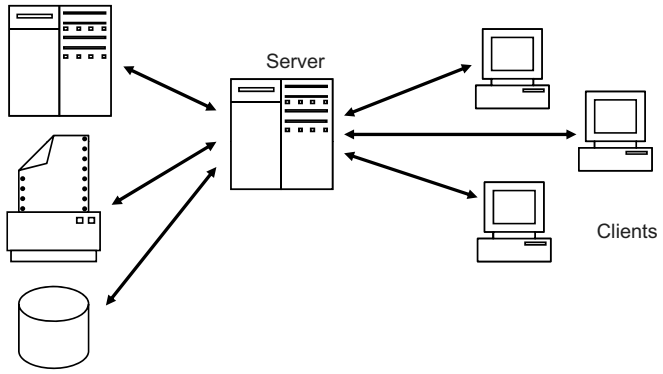
- Client/Server-Kommunikation **mit** Middleware



Quelle: „Operating Systems“, Stallings, Abb.13-12

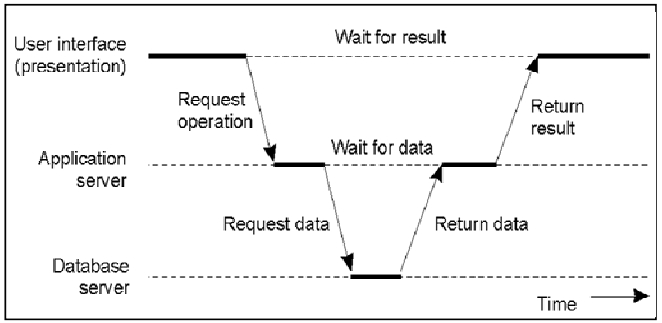
Three-Tier-Modell

- Client/Server-Modell: Two-Tier-Modell
- Server selbst ist auch Client ⇨ Three-Tier-Modell



Three-Tier-Modell

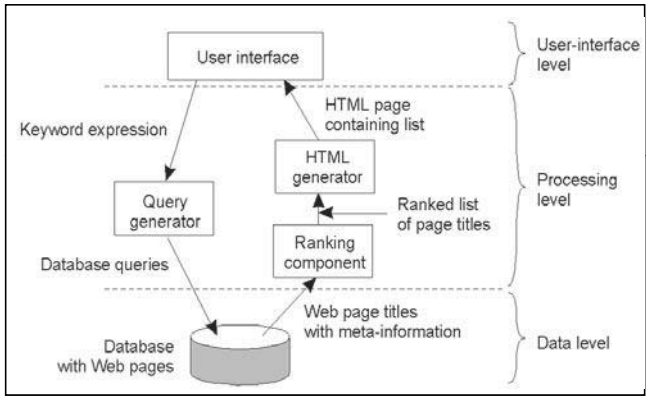
- Erweitertes request-reply Verhalten



Quelle: „Distributed Systems“, Tanenbaum, van Steen, Abb.1-30

3+ -Schichten Client/Server-Modell

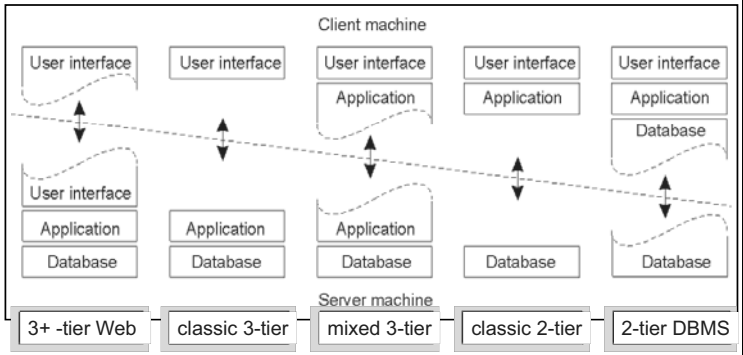
- Zwischenschichten in der Mitte, z.B. Web-Schicht
  Beispiel: Suchmaschine



Quelle: „Distributed Systems“, Tanenbaum, van Steen, Abb.1-28

Client/Server-Modell: Aufgabenverteilung

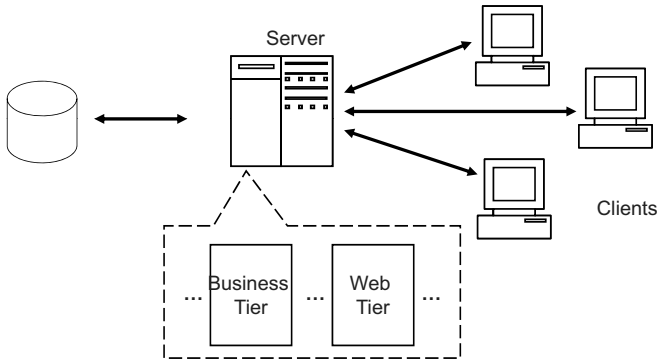
- Wie wird die Anwendung zwischen Server und Client verteilt?
  - Thin Client → Fat Client



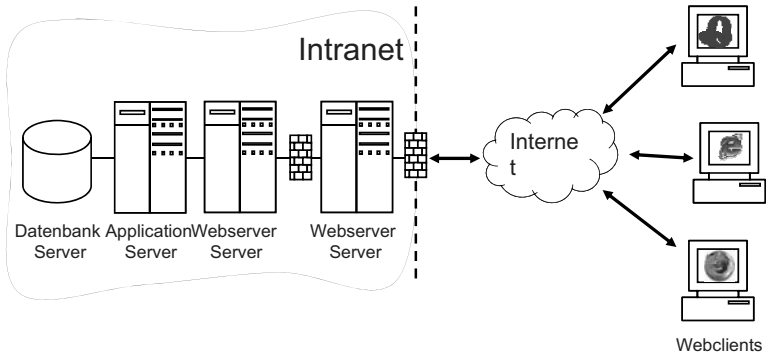
Quelle: „Distributed Systems“, Tanenbaum, van Steen, Abb.1-29

Multitier-Modell

- Einteilung in Komponenten gemäss der Anwendungslogik.

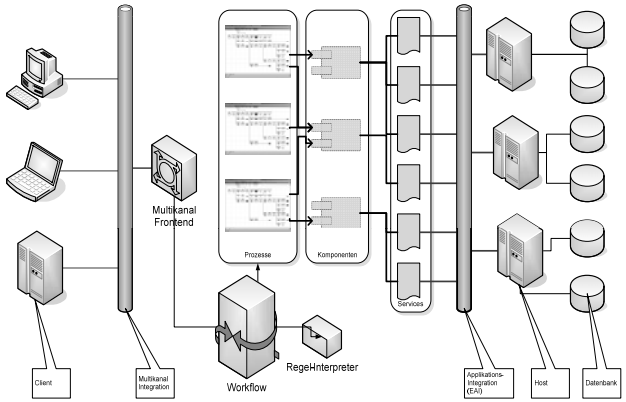


Beispiel: Multi-Tier-Internetanwendung



nach „Verteilte Systeme und Anwendungen“, Hammerschall, Abb.13-7

Die IBM Referenzarchitektur



- Vertikale Verteilung: logisch unterschiedliche Komponenten auf unterschiedlichen Maschinen
- Horizontale Verteilung: ein Server wird physisch in logisch äquivalente Teile unterteilt

Die IBM Referenzarchitektur

- Client Tier: Vielzahl unterschiedlicher Clients
- Multichannel Integration
  - verschiedene Zugänge: WAP-Handy, Geldautomat, Palmtop, Laptop, 3270
  - Umsetzung der Client-Protokolle und Erzeugung der Client Views
- Application Tier
  - Workflow System implementiert Geschäftsprozesse
    - basierend auf Geschäfts-Komponenten
    - WAS (Websphere Application Server) Process Server implementiert WS-BPEL (Web Services Business Process Execution Language) interpretiert Regeln
  - Geschäftskomponenten integrieren einzelne Dienste zu Business Funktionen (z.B. Authentifizierungs-Dienst benötigt für Auszahlung)
  - Einzeldienste konventionell oder als Web-Services
- EAI (Enterprise Application Integration) Schicht
  - bindet konventionelle Systeme ein
  - verbindet (klassische) Anwendungen (high volume application mediation)
- Host / Datenbanken
  - wie bisher

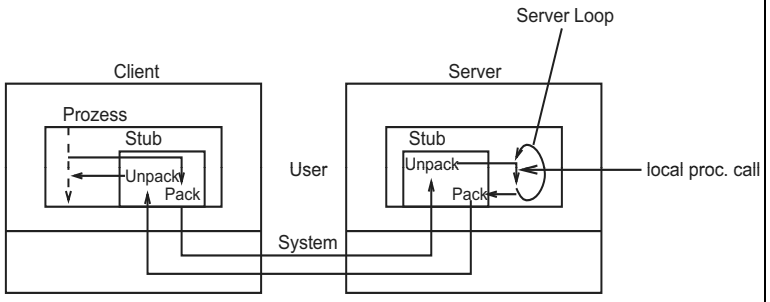
9.3 Remote Procedure Call (RPC)

- Entfernter Funktionsaufruf
  - Client kann Funktion des Servers direkt aufrufen
  - ohne *explizites* Verschicken von Nachrichten auf Programmierebene
  - Implizit durch Verarbeitung im (automatisch generierten) Stummel (Stub):
    - Beim Aufruf: Prozedurname und Parameter werden in Nachricht verpackt *(marshalling)*
    - Nachricht wird an Server geschickt
    - Beim Server: Nachricht wird ausgepackt *(de-marshalling)* und der entsprechende Aufruf wird ausgeführt
    - Ergebnis wird wieder in Nachricht verpackt und zurückgeschickt
- Zu lösende Probleme
  - Unterschiedliche lokale Darstellungen
  - Definition der Schnittstelle
  - Ausfall von Client bzw. Server

Remote Procedure Call (RPC)

- Bei bekannten Datentypen automatische Erzeugung von Code
  - Erzeugung von zwei Stummel (*stub*) Prozeduren für Ein- / Auspacken, Versenden und Empfangen
- Ablauf:
  - Client ruft Client-stub auf, der den Namen der fernen Prozedur trägt
  - Client-stub benachrichtigt Server-stub (und blockiert)
  - Server-stub ruft eigentliche Prozedur auf und schickt Ergebnis zurück
  - Client-stub wird deblockiert, Ergebnis wird ausgepackt und der Client-stub terminiert mit fernem Ergebnis als Ergebnis seines Aufrufs
- Daten müssen in einer Standardrepräsentation verschickt werden
  - Client und Server können auf verschiedenen Architekturen laufen

Schema des RPC

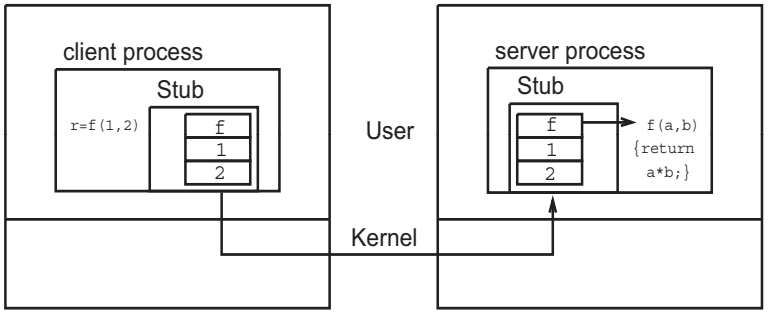


Remote Procedure Call (RPC)

- *parameter marshalling* (aufreihen, zusammenstellen)
  - Verpacken von Parametern einschließlich Konversion in Standarddarstellung
- *unmarshalling* (oder demarshalling)
  - Auspacken einschließlich Konversion in lokale Darstellung
- Server-Loop kann ebenfalls automatisch generiert werden.
  - Server ruft in Schleife den Server-stub immer wieder auf und bearbeitet so einen Auftrag nach dem andern

Schema des Parameter Marshalling

- Beispiel: Multiplikation zweier Zahlen über RPC



Remote Procedure Call (RPC)

- Grund-Annahme: Heterogenität
  - Client und Server können grundverschieden sein
    - verschiedene Prozessoren
    - verschiedene Betriebssysteme
    - verschiedene Programmiersprachen
  - Keine Codeübertragung
    - keine Übertragung von Objekten als Parameter
  - Daten müssen in einer Standardrepräsentation verschickt werden
    - XDR - eXternal Data Representation
  - Schnittstellendefiniton
    ⇨ Sprach-unabhängige Datendefinitionssprache (IDL)
    - Für jede Programmiersprache erzeugt ein IDL-Compiler Sprach-spezifische Stubs

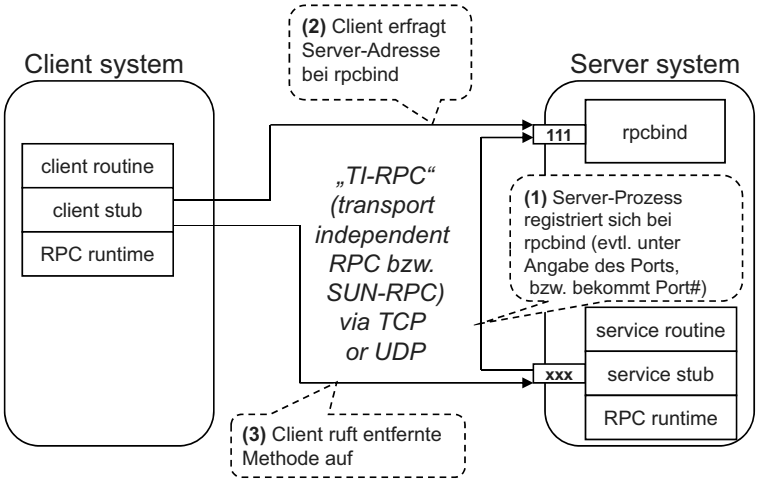
RPC – Dynamisches Binden

- Durch die RPC-Methode sind Aufrufer und gerufene Prozedur entkoppelt:
  - nicht in einem gemeinsamen Programm vereinigt
  - können zu verschiedenen Zeiten gestartet werden

⇨ **dynamisches Binden (*dynamic binding*)**

- Beispiel: statisch: `mobject.add(int)`
  dynamisch: `invoke(mobject, "add", int)`
- Programmbeginn: Client-stub kennt Partner-Adresse noch nicht
- Bei Aufruf von Client-stub: Anfrage an zentralen *Binder* nach Server, der die Prozedur in der benötigten Version zur Verfügung stellt.
- Server melden sich beim Binder betriebsbereit unter Angabe von Name, Versions-Nr. und Adresse + evtl. id, falls Name nicht eindeutig.
- Zur Laufzeit: Binder reicht dem Client die Server-Information weiter, und der Client-stub wendet sich danach direkt an den Server.

Transport-Indeptendent RPC



Fehlerbehandlung in RPC-Systemen

- Durch die Entkopplung zwischen Klient und Server kann es zu folgenden Fehlern kommen:

  1. Der Klient findet den Server nicht.
  2. Die Auftragsnachricht Klient/Server geht verloren.
  3. Die Antwortnachricht Server/Klient geht verloren.
  4. Der Server stürzt nach Auftragserhalt ab.
  5. Der Klient stürzt nach Auftragsvergabe ab.

Fehlerbehandlung in RPC-Systemen

Behandlungsmöglichkeiten sind u.a.:

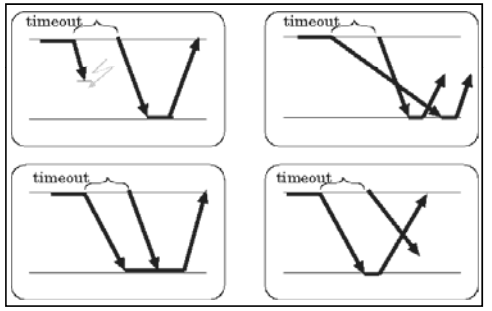
Zu 1: ***No Server***. Stub-Prozedur gibt eine Fehlermeldung zurück.

- Fehlerwert, z.B. -1, genügt nicht im allgemeinen, da er auch ein legales Resultat sein könnte.
- *Exceptions*
- In C: Simulation durch signal handlers, z.B. mit einem neuen Signal SIGNOSERVER
- Signalverarbeitungsroutine könnte darauf hin mit Fehler umgehen (Nachteile: Transparenz geht verloren, sowie Signale werden nicht in allen Sprachen unterstützt)

Fehlerbehandlung in RPC-Systemen

Zu 2: ***Lost Request***. Sender startet einen Timer und verschickt den Auftrag nach Timeout neu.

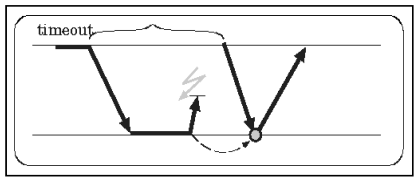
- Kennzeichnung der Aufträge als Original oder Kopie kann verhindern, dass derselbe Auftrag (z.B. Buchung) mehrmals bearbeitet wird.



Fehlerbehandlung in RPC-Systemen

Zu 3: ***Lost Reply***.

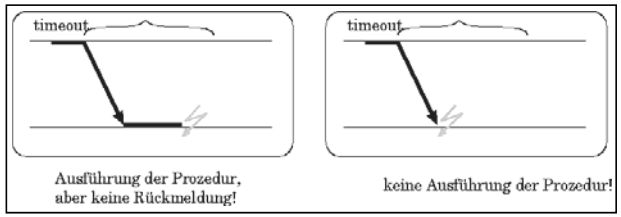
- Manche Aufträge können problemlos wiederholt werden (Lesen eines Datums), andere nicht (*relative update*; Buchung).
- Buchungsaufträge müssen als Originale und Kopien gekennzeichnet werden.



Fehlerbehandlung in RPC-Systemen

Zu 4: ***Server Crashes***.

- Abstürze des Servers vor Auftragseingang fallen in Kategorie 1 (no server)
- Späterer Absturz **vor** Auftragsbearbeitung: Wiederholung des Auftrags (nach reboot)
- Späterer Absturz **nach** Auftragsbearbeitung: Wiederholung nicht unbedingt möglich
- Problem: Klient kennt die Absturzursache nicht.



Fehlerbehandlung in RPC-Systemen

Drei Konzepte der RPC-Abwicklung:

1. ***At least once semantics***. Das RPC-System wiederholt den Auftrag so lange, bis er quittiert wurde.
2. ***At most once semantics***. Das RPC-System bricht nach Timeout ab mit Fehlermeldung.
3. ***Keine Garantie***. Das RPC-System gibt irgendwann auf. Der Auftrag kann nicht oder auch mehrmals bearbeitet worden sein.

- Ideal wäre: exactly once semantics. Dies ist im Allgemeinen aber nicht zu realisieren.

Fehlerbehandlung in RPC-Systemen

Zu 5: ***Client Crashes***.

- Klient-Absturz vor Auftragsvergabe oder nach Auftragsbestätigung hier irrelevant.
- Absturz während der Auftragsverarbeitung führt zu einer Waise (Prozess ohne Eltern; ***orphan***).
- Probleme: Verbrauch von CPU-Ressourcen und Blockierung anderer Prozesse

Fehlerbehandlung in RPC-Systemen

- Methoden, um Waisen aus dem System zu entfernen:

1. Extermination
   - Notieren jeder Auftragsvergabe beim Klienten auf sicherem Medium
   - Nach reboot werden (unquittierte) offene Aufträge storniert (Löschung = extermination)
   - Problem: großer Aufwand, sowie keine Erfolgsgarantie (z.B. bei fehlender Konnektivität zum Server)
   - Schlussfolgerung: keine relevante Methode
2. Reincarnation
   - Zeit wird in Epochen eingeteilt
   - Jeder Client-reboot startet neue Epoche
   - Prozesse der alten Epoche werden auf dem Server beendet
   - Überlebt doch einer (z.B. durch verlorene Epochen-Meldung), so tragen seine Resultate veralteten Epochen-Stempel

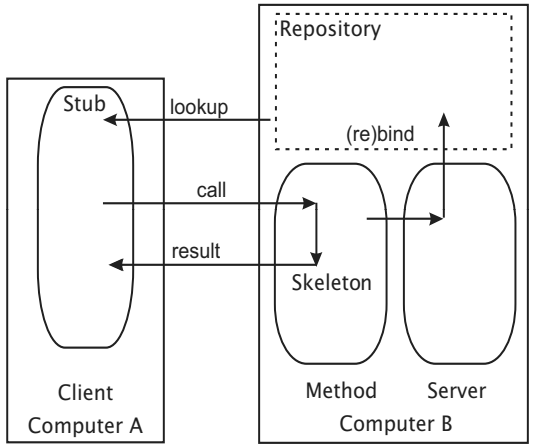
Fehlerbehandlung in RPC-Systemen

3. Gentle Reincarnation
   - Server fragen bei Start einer neuen Epoche nach, ob Eltern der Aufträge noch leben
   - Nur wenn sich Eltern nicht melden, werden die Aufträge beendet
4. Expiration
   - Auftragsbearbeitung wird mit Timeouts versehen
   - Nach Timeout müssen sich Eltern melden und dadurch den Timer neu starten
   - ansonsten wird der Prozess beendet
   - Problem: richtigen Wert für Timeout festlegen (RPCs haben stark variierende Anforderungen)

9.4 Java Remote Method Invocation (RMI)

- Entfernte Prozeduren → entfernte Methoden
- RMI = RPC in Objektsystemen
  - Parameter können Objekte sein (Erschwernis)
    - Umgang mit Zeigern und mit Code
    - Objekte in Bytestrom übertragen (serialisieren)
    - verzeigerte Objekte (Listen, Bäume, Graphen): Objektgraphen
  - Alle Information in der Klassendefinition gekapselt (Erleichterung)
- Java RMI ist eine Java-spezifische Realisierung des RPC
  - benutzt Java Objekt-Serialisierungsprotokoll
- Common Object Request Broker Architecture (CORBA)
  - Kapselung von Objekten verschiedener Programmiersprachen

RMI – Schema



Aufgabe von verteilten RPC Systemen

- Verteilte RPC Systeme (wie das verteilte Objektsystem CORBA) zielen auf die Lösung der Kommunikations-probleme, die auf der **Heterogenität** der beteiligten Systeme beruhen:
  - Unterschiedliche Programmiersprachen
  - Unterschiedliche Rechner
  - Unterschiedliche Betriebssysteme
  - Unterschiedliche Datenrepräsentation
  - Unterschiedlicher Maschineninstruktionssatz

Lösungsansatz bei RPC Systemen

- Stub, Skeleton bilden Aufrufschnittstelle und erledigen den eigentlichen Datenaustausch
- Marshaling, Unmarshaling zur (De-)Serialisierung
- Gemeinsame IDL (Interface Description Language) ermöglicht Kommunikation auch zwischen unterschiedlichen Programmiersprachen
  - Für jede beteiligte Sprache ein IDL Compiler
- Unterstützung entfernter Referenzen für Objekte

Probleme bei RPC

- Einschränkung hinsichtlich der übertragbaren Daten
  - Nur einfache Datentypen, die in allen unterstützten Programmiersprachen repräsentierbar sind und
  - Referenzen auf entfernte Objekte sowie
  - Komplexe Datentypen, die sich aus den zuvor genannten zusammensetzen
- Komplexität bei Typanpassung verbleibt beim Programmierer
- Life-Cycle-Management wird dem Programmierer auferlegt
  ⇨ Gefahr von Fehlern
- Sender und Empfänger müssen die übertragbaren Datentypen zum Zeitpunkt der Kompilierung bereits kennen.
  ⇨ Keine Unterstützung von Polymorphie
  (Polymorphie: Bezeichner kann je nach Kontext einen unterschiedlichen Datentypen annehmen)

Lösung von Java RMI

- Heterogenität stellt kein Problem dar, da **Homogenität** durch die Java JVM gewährleistet ist.
  - Externe IDL nicht erforderlich (stattdessen Java Interfaces)
  - Keine Einschränkung hinsichtlich der übertragbaren Datenstrukturen.
  - Java Objekt-Serialisierung ermöglicht exakte Typenprüfung
  - Dynamic Code Loading: Der den Kommunikationsfluss steuernde Programmcode kann auch erst während der Programmausführung zur Verfügung gestellt werden.
  - Java Objekt-Serialisierung und Dynamic Code Loading ermöglichen den Einsatz von Polymorphie und aller darauf aufbauenden Programmiermuster
  - Network Garbage Collection wird unterstützt

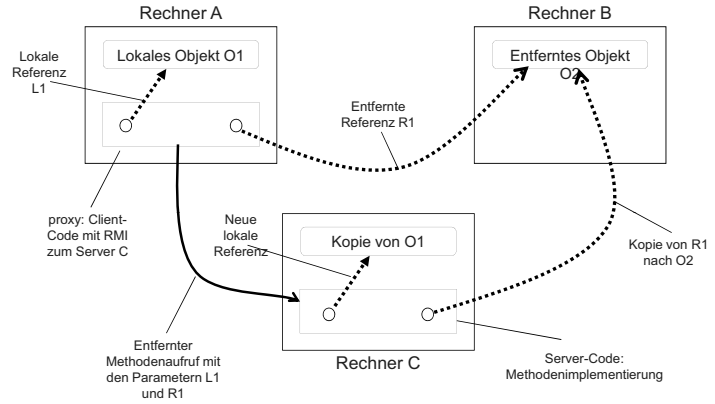
Parameterübergabe und Rückgabewerte

- Java allgemein:
  - Alle Parameter (und Rückgabewerte) werden jeweils kopiert: call by value.

    Hinweis: Objektvariablen bilden dabei keine Ausnahme. Da es sich bei ihnen jedoch um Referenzen handelt, entspricht ihre call by value-Übergabe der Semantik von call by reference. Änderungen durch die aufgerufene Methode entfalten also Wirkung.

- Java RMI:
  - Lokale Datentypen und lokale Objekte werden kopiert: call by value
  - Entfernte Objekte durch Kopie (des entsprechenden proxies/stubs) der Objektreferenz: call by reference

Veranschaulichung der Parameterübergabe



Quelle: Tanenbaum, Distributed Systems, 2. Aufl., Abb. 10-8

=> Entfernte Objekte bleiben entfernt

Gliederung - Kapitel 9: Verteilte Systeme

Motivierende Fragen

- Was versteht man unter SOA - Service Oriented Architectures?
- Was sind die Grundprinzipien von CORBA?
- Wie können Web-Anwendungen implementiert werden?
- Was sind die Unterschiede von HTML und XML?
- Was sind Web Services?
- Was versteht man unter SOAP, WSDL, SAX, UDDI?

Inhalte von Kapitel 9, Teil 2

- Service-Orientierte Architekturen
- Corba
- Web-Technologien
  - Java Server Pages
  - Java Servlets
- Sprache XML
  - XML Tags
  - Name Spaces
  - XML-Schemata
  - Validierung von XML-Dokumenten
  - Werkzeugunterstützung für XML
  - Transformation in andere XML-Formate, oder andere Sprachen
- Web Services
  - Schichtenarchitektur
  - Simple Object Access Protocol - SOAP: Mechanismus zur Repräsentation/zum Austausch von Daten
  - Web Services Description Language - WSDL
  - Simple API for XML Parsing - SAX
  - Universal Description and Integration UDDI

9.5 Definition Service Oriented Architectures

- SOA ist ein Paradigma für die Strukturierung und Nutzung verteilter Funktionalität, die von unterschiedlichen Besitzern verantwortet wird.
  [Organization for the Advancement of Structured Information Standards (OASIS) , 2006] c.f. oasis-open.org
- Ein Dienst in einer Service-Orientierten Architektur hat (idealerweise) folgende Eigenschaften
  - Dienst ist in sich abgeschlossen und kann eigenständig genutzt werden.
  - Dienst ist über ein Netzwerk verfügbar.
  - Dienst hat eine veröffentlichte Schnittstelle. Für die Nutzung reicht es, die Schnittstelle zu kennen. Kenntnisse über die Details der Implementierung sind hingegen nicht erforderlich.
  - Dienst ist plattformunabhängig, d.h. Anbieter und Nutzer eines Dienstes können in unterschiedlichen Programmiersprachen auf verschiedenen Plattformen realisiert sein.
  - Dienst ist in einem Verzeichnis registriert.
  - Dienst ist dynamisch gebunden, d.h. bei der Erstellung einer Anwendung, die einen Dienst nutzt, muss der Dienst nicht vorhanden sein. Er wird erst bei der Ausführung lokalisiert und eingebunden.

9.6 CORBA

- Common Object Request Broker Architecture
- Allgemeiner Architektur-Standard für Entwicklung von Client/Server Anwendungen
- Verschiedene konkrete Implementierungen: ORBs.
- Definiert von der Object Management Group (OMG)
  - Zusammenschluss von über 750 Unternehmen, Software-Entwicklern und Anwendern.
  - 1989 gegründet.
- Allgemeine Kommunikationsinfrastruktur zwischen verteilten Objekten, wobei für den Entwickler die Kommunikation weitestgehend transparent ist.

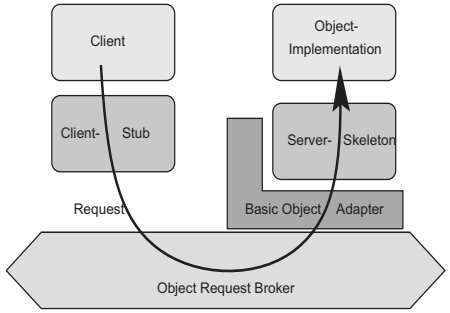
CORBA

- Merkmale von CORBA
  - Objektorientierung
  - Sprachunabhängigkeit
  - Kommunikationsmechanismen für verteilte Systeme
  - Allgemeines Konzept von kommunizierenden Objekten
  - Plattformunabhängigkeit
  - Herstellerunabhängigkeit
  - Anbindung anderer Komponentensysteme
- CORBA geht daher nach folgendem Prinzip vor
  - Die Schnittstellen werden von der Implementierung streng getrennt.
- Durch separate Definition der Schnittstellen kann die Kommunikation unabhängig von der jeweiligen Implementierung betrachtet werden.
  - IDL (Interface Definition Language): Neutrale Spezifikationssprache beschreibt die Schnittstellen der beteiligten Objekte
  - ORB (Object Request Broker): Kommunikation zwischen den Objekten

Merkmale von CORBA

- Kommunikationsmechanismen
  - Kommunikation zwischen den Objekten ist völlig transparent
    - Der ORB verdeckt
      - Auffinden des Zielobjekts
      - Übertragung der Daten
      - Etwaige Konvertierungen zwischen Datenformaten.
  - Kommunikation selbst ist plattform- und sprachunabhängig.
    - Seit CORBA 2.0 arbeiten auch Object Request Broker verschiedener Hersteller zusammen (durch das Internet Inter-Orb Protocol - IIOP).
- Allgemeines Konzept von kommunizierenden Objekten
  - Die bisherige starre Einteilung in Client und Server entfällt, in CORBA existieren gleichberechtigte Objekte.
    - Objekt A kann Server für ein Objekt B sein, während B gleichzeitig Serverfunktionalität für ein Objekt C anbietet.
    - Festlegung: Server ist das Objekt, das ein IDL-Interface implementiert.

Grundprinzip von CORBA



- Mit Hilfe der IDL wird ein Interface definiert.
- IDL-Compiler erzeugt aus dieser Schnittstellenbeschreibung Source Code in der gewünschten Sprache. Für den Client **Stub** und für den Server **Skeleton**.
- Server wird implementiert und ist über das Skeleton für andere Objekte zugänglich. Über den *Basic Object Adapter* (**BOA**) meldet sich der Server beim ORB an und ist jetzt bereit, Aufrufe anderer Objekte zu empfangen.
- Der Client kann nun über den Stub auf den Server zugreifen. Dieser Zugriff läuft über den ORB.

Grundprinzip von CORBA

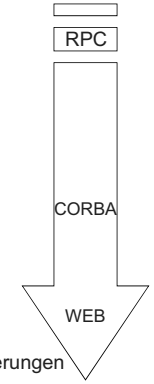
- Ablauf entfernter Methodenaufruf:
  - Object Request Broker fängt Aufruf ab und lokalisiert das Zielobjekt
  - Übergabeparameter werden verpackt und an den Server geschickt.
  - Dort werden die Parameter wieder entpackt und die Methode auf dem Server ausgeführt.
  - Resultat wird verpackt und an den Aufrufer zurückgesendet.
  - Gesamter Vorgang wird vom ORB verdeckt.
- Client benötigt für entfernten Methodenaufruf eine *Referenz* auf das entfernte Objekt: *Object-Reference*
  - eindeutige ID eines bestimmten Objekts
- Angesprochen wird die Server-Komponente über die automatisch generierte ”Stub-Klasse“ ⇨ hohe Typsicherheit

Object Request Broker – Grundfunktionalität

- Statische und dynamische Aufrufe von Methoden
  - Statischer Aufruf: sichere Typprüfung
  - Dynamische Aufrufe: höhere Flexibilität
- Kommunikation auf Hochsprachen-Niveau
  - Es ist für Entwickler nicht notwendig, Parameter „einzupacken“ (marshaling) oder Befehle in einer Kurzform zu übertragen
- Selbstbeschreibendes System
- Jeder ORB besitzt eine Datenbank, das sog. Interface Repository
  - Enthält Meta-Informationen über die bekannten Interfaces
  - Daten werden automatisch verwaltet und gepflegt, i.a. durch den IDL-Compiler
- Transparenz zwischen lokalen und entfernten Aufrufen
- Umwandlungen zwischen Datenformaten (z.B. little endian/big endian, usw.) werden vom ORB durchgeführt
- Kompatibilität bzgl. Art und Größe der Typen durch IDL gewährleistet

Gründe für den Niedergang von CORBA

- Technische Gründe
  - Komplexität von CORBA
  - Unterstützung von C++ fehlerträchtig
  - Sicherheitsmängel
    - unverschlüsselter Datenaustausch
    - pro Dienst ein offener Port in Firewall erforderlich
  - Viel Redundanz, keine Kompression
  - Keine Thread-Unterstützung
  - Fehlende Unterstützung von C#, .NET, DCOM
  - Fehlende Integration des Web
    - Aufkommen von XML, SOAP
    - ⇨ E-business Lösungen generell ohne CORBA
- Soziale Gründe / Verfahrensfehler
  - Zu viele Köche im Standardisierungsprozess
  - Z.T. Fehlende Referenzimplementierungen
  - Ungetestete Innovationen in Standards
  - Hohe Lizenzgebühren für kommerzielle Implementierungen
  - open-source Implementierungen zu spät
  - Mangel an erfahrenen Entwicklern



Quelle: Michi Henning, ACM Queue, http://www.acm.org/acmqueue/digital/Queuevol5no4_May2007.pdf

9.7 Web Services: Grundidee und Definitionen

- HTML (HyperText Markup Language) als Beschreibungssprache zur Darstellung von Dokumente im World Wide Web ist schlecht geeignet zur Maschine-Maschine-Kommunikation
- Grundidee für Web Services: Einsatz Web-basierter Service-Orientierter Architektur
  - Darstellung der Informationen mit einer dafür besser geeigneten Sprache ⇨XML
  - Verwendung von Elementen der Web-Architektur, u.a. Beibehaltung von HTTP zum Transport dieser Daten

Web Services - Gründe für das Interesse

Vorzüge

- Web Services basieren auf offenen Protokollen bzw. Spezifikationen
  - Beschreibung der Schnittstelle: WSDL (Web Service Description Language)
  - Kommunikation: SOAP (Simple Object Access Protocol)
  - Finden von Diensten: UDDI (Universal Description Discovery and Integration)
  - Spezifikation über XML-Grammatiken
- Heterogene Plattformen (J2EE, .Net etc.) werden unterstützt
- Grundlage zur Realisierung der Service-orientierten Architektur mittels WSDL (Web Services Description Language)
- Web Services werden von "großen Organisationen" (IBM, Sun Microsystems, SAP, Microsoft, ...) unterstützt

Mögliche Nachteile

- Leistungsfähigkeit von SOAP / XML ggf. geringer anderer Leistungsfähigkeit anderer Middleware-Lösungen

9.8 HTML und XML: Markup

- Markup: Informationen, die einem Dokument beigefügt werden, selbst aber nicht unmittelbar dargestellt werden.

- Bsp.: HTML-Markup:

```
<h2>Markup zur Textformatierung</h2>
Unter <i>Markup</i> versteht man Informationen,
die einem Textdokument beigef&uuml;gt sind.
```

HTML – Hypertext Markup Language

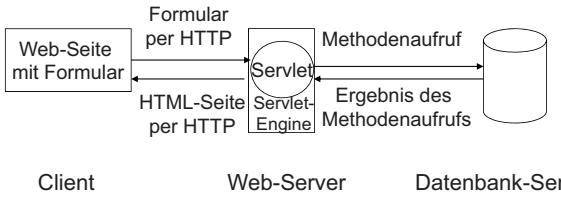
- HTML definiert primär Layout eines Web-Dokuments
  - nur sekundär auch Struktur
- Auszeichnungssprache
  - paarweise öffnende und schließende Tags
  - Hierarchische Gliederung
- Tags
  - zur Textformatierung
    - `<i>kursiv</i>`
  - zur Spezifikation von Hypertext-Links
    - `<A HREF="http://www.w3c.org"> W3 Konsortium </A>`
  - zum Einbinden von Multimediaobjekten und Applets
    - `<img src="logo.gif" alt="Alternativtext">`
  - für Formulare

Web-Anwendungen

- Dynamische Web-Seiten-Erzeugung
  - CGI-Skripte bzw. -Programme
  - Active Server Pages (heute ASP.Net)
  - PHP
  - Java Servlets
  - Java Server Pages

Servlets

- Problem: HTML Dokumente als Files sind statisch
- Idee: Generiere HTML-Dokumente dynamisch durch Programm
- Servlet: Ein Java-Programm, das auf dem Server als Reaktion auf einen http-request (GET oder POST) gestartet wird
- HTML-Seite wird vom Servlet generiert.
- Beispiel: Servlet in 3-tier-Architektur



Java Server Pages (JSP)

- JSP: Eine HTML-Seite mit eingebettetem Java-Code, der auf Server ausgeführt wird und Teile der Seite dynamisch generiert.
- Lösung für folgendes Problem: Es ist umständlich, die statischen Teile der HTML-Seite durch das Servlet in print-statements zu generieren.
- Idee: Schreibe statische Teile als HTML in Dokument und bette dynamische Teile durch Programmcode darin ein (spezieller HTML-Kommentar).
- JSP wird von Webserver in Servlet übersetzt

```
<html>
<body>

<%-- Kommentar --%>
<% //Java-Code %>

</body>
</html>
```

Skriptlet <% ... %>
Kommentar <% --...--%>
Direktive <% @ ... %>
Ausdruck <% = ... %>
Deklaration <%! ...%>

SGML – Standardized General Markup Language

- Ende 80er Jahre bei IBM entwickelt (Goldfarb)
- ISO Standard 1986
- Tags zur Annotation (mark-up) eines Textes
  - zur Textformatierung
    - `<i>kursiv</i>`
- Ziel: repräsentiere Industrie-Dokumentationen
  - elektronisch
  - unabhängig von konkreten Text-Satzsystemen
  - bilde auf jeweiliges Medium ab (z.B. Web oder Druck)
  - Industriequalität (Novell-Handbücher: 150.000 Seiten)
- Problem: sehr mächtig, sehr komplex

SGML, HTML, XML

- Bewertung von SGML: Idee prima, aber SGML zu komplex
- HTML: Formatiere Dokumente für das Web
  - Vordefinierte Menge von Tags, Semantik fixiert
  - Werden von Browsern interpretiert
  - bilden *Document Type*, der mittels SGML definiert ist
- XML: definiere Dokumentstruktur allgemein
  - Minimale Sprache, kleines subset von SGML („SGML- -“)
  - Erweiterbar
  - Abbildung Struktur ⇨ layout separat über style sheets
    ⇨ auch für Datenserialisierung nützlich

XML – eXtensible Markup Language

- Offener Standard (www.w3.org)
- Strukturdefinition (Grammatik einer Dokumentfamilie)
  - Document Type Definition (DTD)
  - XML-Schema
- Platform-unabhängig (ASCII)
- Allgemein einsetzbar
- Leichte maschinelle Zugänglichkeit der Information
  - Textbasiert
  - Strukturiert
- Werkzeuge
  - Editoren, Browser, ...
  - Parser
  - APIs
  - Datenbank-Schnittstellen

XML – eXtensible Markup Language

- Beschreibung strukturierter Daten (kein Layout)
- Maschinen- und Menschen-lesbar (ASCII-Text)
- wohlgeformte XML-Dokumente
  - paarweise öffnende und schließende Tags
    - HTML erlaubt auch Ausnahmen: `<br>`
  - keine Überlappung der Tags erlaubt
  - alle Tags kleingeschrieben
  - Attributwerte in doppelten Anführungszeichen
- Inline-Schreibweise mit Attributen

```
<autor nachname="Stevens" vorname="Richard"/>
```

- gemischte Schreibweise

```
<autor geschlecht="männlich">
    Stevens, W. Richard
</autor>
```

XML – Beispiel

Kopf

```
<?xml version="1.0"?>
```

Rumpf

```
<literaturverzeichnis>
      <buch>
            <autor>Stevens, W. Richard</autor>
            <titel>UNIX Network Programming</titel>
            <verlag>Prentice Hall</verlag>
            <erscheinungsjahr>1990</erscheinungsjahr>
            <isbn>0-13-949876-1</isbn>
            <stichwort>Netzwerk</stichwort>
            <stichwort>Netzwerk-Programmierung</stichwort>
      </buch>
      <buch>
            ....
      </buch>
</literaturverzeichnis>
```


XML – Syntax: Namensräume

- XML-Namensräume (XML Name Spaces)
  - Werden benutzt, um in einem Dokument mehrere XML-Sprachen zu mischen
  - Werden durch URIs dargestellt
  - Die entsprechende Adresse muss nicht existieren
  - Wenn eine URL als Namensraum verwendet wird, wird unter dieser Adresse ggf. zusätzliche Informationen zu der XML-Sprache angeboten, z. B. eine Dokumenttypdefinition (DTD) oder ein XML-Schema.
- Ziel: Vermeidung von Mehrdeutigkeiten bei Tags
- Definition mittels `xmlns`-Attribut oder dem `xmlns:`-Präfix
- Wert des Attributs ist Name des Namespaces
- Beispiel:

```
<?xml version="1.0"?>
<da:literaturverzeichnis
            xmlns:da="http://www.in.tum.de/diplArb">
    <da:buch> ... </da:buch>
</da:literaturverzeichnis>
```


XML Document Type Definition (DTD)

- Dokumenttypdefinition (Document Type Definition, DTD, auch Schema-Definition oder DOCTYPE)
  - Sprache zur Beschreibung von Dokumenttypen, d.h. zur Beschreibung der Stuktur von Dokumenten (Beachte: Für XML-Dokmente existieren auch andere Schema-Sprachen, z.B. XML Schema)
  - Formale Grammatik (DTD-Syntax selber ist kein XML)
  - Spezifiziert Elemente, Attribute, Entitäten
  - Spezifikation von Dokumenttypen erlaubt Validierung von Dokumenten
  - XML-Editor der DTD (Grammatik) kennt kann nur gültige Eingaben zulassen
  - Einbindung im selben Dokument oder ausgelagert in externes Dokument
  - Beispiel:

```
<!ELEMENT publication (proceedings | article)>
<!ELEMENT proceedings ((author)*, title, conference)>
...
<!ELEMENT conference (#PCDATA)>
<!ATTLIST conference
  year CDATA #required>
```


XML-Schema

- Sprache zum Definieren von Strukturen für XML-Dokumente.
- Im Gegensatz zu XML-DTDs wird die Struktur in Form eines XML-Dokuments beschrieben.
- Zahlreiche Datentypen werden unterstützt.
- Basis-Datentypen
  - `string, date, time, integer, double, boolean`
- komplexere Strukturen
  - Einschränkung des Wertebereichs
  - Listen
  - Vereinigung und Kombination verschiedener Typen
  - Vererbung

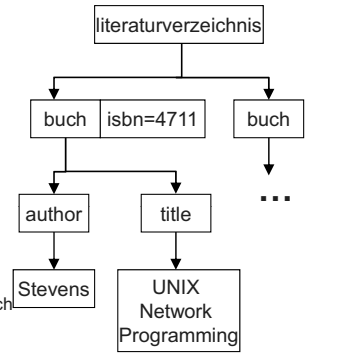
XML-Schema – Beispiel

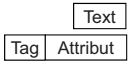
```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
    elementFormDefault="qualified">
    <xs:element name="literaturverzeichnis">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="buch" maxOccurs="unbounded">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="autor" type="xs:string"
                         maxOccurs="unbounded"/>
                        <xs:element name="titel" type="xs:string"/>
                        <xs:element name="untertitel" type="xs:string"
                         minOccurs="0"/>
                        <xs:element name="verlag" type="xs:string"/>
                        <xs:element name="erscheinungsjahr" type="xs:string"/>
                        <xs:element name="isbn" type="xs:string"/>
                        <xs:element name="stichwort" type="xs:string"
                         maxOccurs="unbounded"/>
                        <xs:element name="abstract" type="xs:string"
                         minOccurs="0"/>
                        <xs:element name="kommentar" type="xs:string"
                         minOccurs="0"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
    </xs:element>
</xs:schema>
```

Vom W3C im XML-Schema
`http://www.w3.org/2001/XMLSchema` vorgegeben:
- Basistypen `element`, `complexType`, `sequence`, …
- Datentypen `string`, `integer`, …

Werkzeugunterstützung für XML

- APIs zum Parsen von XML-Daten
- SAX
  - *Simple API for XML*
  - Ereignisorientierter Ansatz
  - Dokument wird komplett durchlaufen
  - Beginn / Ende jedes Tags wird über Callback-Methoden mitgeteilt
- DOM
  - *Document Object Model*
  - Erlaubt, gezielt auf einzelne Teile des Dokuments zuzugreifen
  - Baumorientierter Ansatz
  - Applikation bekommt bekommt Baum nach Verarbeitung des Dokuments übergeben
  - höherer Speicherbedarf als SAX
- Beide Ansätze in J2SDK enthalten
  - Zusammengefasst in *Java API for XML Processing (JAXP)*





XML Standard Familie

Von W3C standardisiert (www.w3.org)

- CSS (1998): Cascading Style Sheets (layout: Stylesheet-Sprache für strukturierte Dokumente)
- XML 1.0
- Namespaces (1999)
- XSLT 1.0 (1999) XSL Transformations (Programmiersprache zur Transformation von XML-Dokumenten)
- XPath 1.0 (1999) Zugriff auf Teile eines Dokuments
- XHTML 1.0 (2000), (Extensible HTML). "A Reformulation of HTML 4 in XML 1.0"
- DOM Level 2, Document Object Model, Core Specification (2000)
- XML Schema (2001) Grammatik-Sprache für XML-Dokumentfamilien
- XLink 1.0 (2001): XML Linking Language (Spez. für Hyperlinks)
- XML Base (2001) (Spezifikation von Datenbank URIs für Dokument-Teile)
- XSL 1.0 (2001) Extensible Stylesheet Language (layout)
- XPointer (2002) : XML Pointer Language (Spezifikation von Pfaden in URIs)
- XQuery 1.0 (2002) XML Query Language (Abfragesprache für Datenbanken)
- XInclude (2002) XML Inclusions

XML Zusammenfassung

- Umfangreicher Tool Support
- Generalized Markup
  - – Trennung von Struktur und Darstellung
  - – XML Parser ⇒ Information Access
  - – CSS, XSL, XSLT ⇒ Darstellung (Layout, Rendering)
  - – Sichten ⇒ Flexibilität und Konsistenz
- Document Type Definition (DTD, Schema)
  - – Klassenbildung + Validierung
- Persistenz (inkl. XML Datenbanken)
- Erweiterte Link-Fähigkeit
- Multi-Medial, International

9.9 Definition von Web Services

A Web service is a software system designed to support interoperable machine-to-machine interaction over a network.

*It has an interface described in a machine-processable format (specifically **WSDL**).*

*Other systems interact with the Web service in a manner prescribed by its description using **SOAP** messages, typically conveyed using HTTP with an **XML** serialization in conjunction with other Web-related standards.*

David Booth et al.: *Web Service Architecture*
W3C Working Group Note 11 February 2004
http://www.w3.org/TR/ws-arch/

Definition von Web Services

A web service is a piece of business logic, located somewhere on the Internet, that is accessible through standard-based Internet protocols such as HTTP or SMTP. Using a web service could be a simple as logging into a site or as complex as facilitating a multi-organization business negotiation.

Chappell / Jewell:
„Java Web Serivces“
O‘Reilly, 2002

Web Services

- Komponentenmodell unter Verwendung von Web-Technologien
  - zentral: XML (Serialisierung und Schnittstellenbeschreibung)
- Motivation:
  - Löst ähnliche Probleme wie CORBA
  - Aber offener / allgemeiner als CORBA
- Bisher: große monolitische Informationssysteme
  - Client / Server / Datenbank zu eng verwoben
  - Schlecht dokumentierte Schnittstellen, proprietäre Formate
  - CORBA ORB‘s nicht überall verfügbar, beschränkt inter-operabel
  - Re-compilation bei kleinsten Änderungen der CORBA IDL

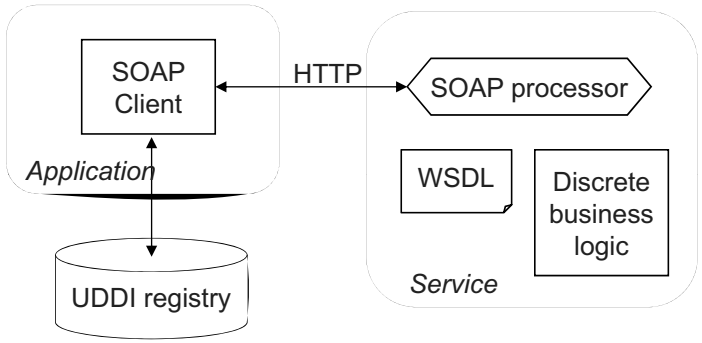
Web Services

- Vision: Integration verschiedener Business-Komponenten
  - über Abteilungs- und Unternehmensgrenzen hinweg
  - mit Web-basierten Technologien (XML / HTTP)
- Integration durch Schaffung von Standards
  - vor allem: XML
  - im Unterschied zu z.B. CORBA (z.B. XML statt IIOP)
- Merkmale von Web Services
  - XML-basiert ⇨ plattformunabhängig
  - lose Kopplung
  - grobgranular ⇨ mehrere Methoden bilden einen Dienst
  - Unterstützung von entfernten Methodenaufrufen
  - Unterstützung von Dokumentenaustausch

Web Services – Schlüsseltechnologien

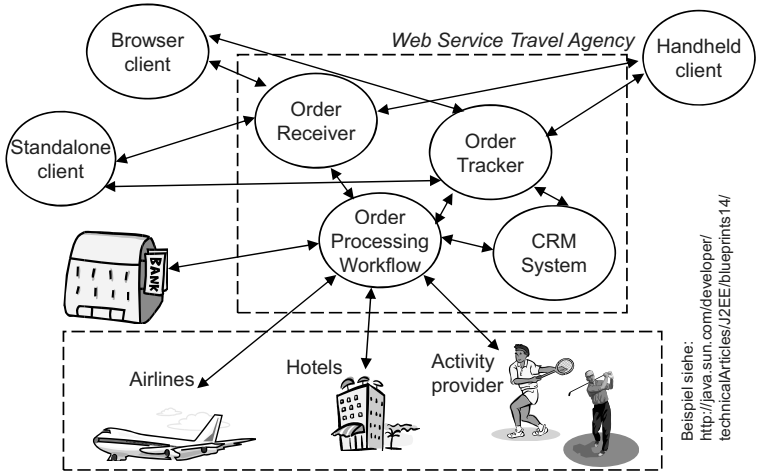
- **XML**
  - Universelle Beschreibungssprache
  - Selbst-dokumentierend
  - Robust gegen Änderungen: Empfänger überliest irrelevante Einträge
- **WSDL** (Web Service Description Language)
  - Interface Beschreibung von Diensten (analog CORBA IDL)
- **SOAP** (Simple Object Access Protocol)
  - Kommunikation zwischen Diensten („XML-RPC“)
  - Transportiert XML-serialisierte Werte und Methoden-Aufrufe
- **UDDI** (Universal Description, Discovery and Integration)
  - Suchen von Diensten
  - Weltweiter Verzeichnisdienst für Web Services

Web Services – einfaches Szenario



Chappell / Jewel: „Java Web Services“, O‘Reilly, 2002, Abb.1-1

Web Services –Szenario



XML-RPC

- Idee: Entfernter Methodenaufruf ohne neue Technologien
  - Serialisierung in XML-Dokument
    - Methodenname
    - Parameter
  - Verschicken über HTTP
- Verfügbare Datentypen
  - `<int>`
  - `<double>`
  - `<string>`
  - `<boolean>`
  - `<base64>`
    - Bytefolge
  - `<dateTime.iso8601>`
    - Beispiel: `<dateTime.iso8601>20060125T11:15:12</dateTime.iso8601>`

XML-RPC – Prozeduraufruf

```
HOST /xml-rpc.app HTTP/1.1
Content-type: text/xml
Content-length: 255

<?xml version="1.0"?>
<methodCall>
   <methodName>calcMaximum</methodName>
   <params>
      <param>
         <value><int>47</int></value>
      </param>
        <param>
          <value><int>23</int></value>
      </param>
   </params>
</methodCall>
```


XML-RPC – Prozedurantwort

```
HTTP/1.1 200 OK
Content-Type: text/xml
Content-Length: 158

<?xml version="1.0" ?>
<methodResponse>
   <params>
      <param>
         <value><int>47</int></value>
      </param>
   </params>
</methodResponse>
```

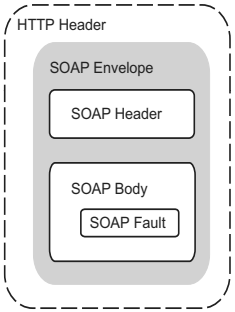

XML-RPC – Bewertung

- Vorteil
  - Sehr einfach und schlank
- Nachteile
  - ungenaue Codierung der Datentypen
    - z.B. Probleme mit Datumstyp: keine Zeitzone
  - Aufwändige Codierung binärer Daten (base64)
- Fehlende Metainformation zu den Methoden
  - könnten nur im Request-/Response-Header von HTTP beigefügt werden (⇨ nicht self-contained)
- Weiterentwicklung zu SOAP

SOAP

- *Simple Object Access Protocol*
- XML-basiertes Nachrichtenprotokoll
- Arbeitet auf bestehenden Transportprotokollen (HTTP, SMTP)
- Aufbau einer SOAP Nachricht
  - Envelope
    Header
    - optional
    - für Metainformationen

    Body
- kann vollständig im Dokument-Teil eines HTTP-Requests übertragen werden.



SOAP – Nachrichtenformat

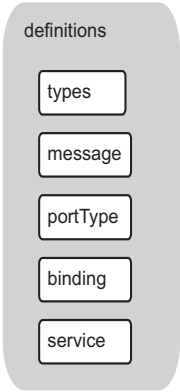
```
<SOAP-ENV:Envelope xmlns:SOAP-ENV=
   "http://www.w3.org/2003/05/soap-envelope/">
     <SOAP-ENV:Header>
              <!-- Header-Information -->
     </SOAP-ENV:Header>
     <SOAP-ENV:Body>
               <!-- Body-Information -->
     </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```


WSDL – Web Services Description Language

- Interface Definition Language for Web Services
- Basiert auf XML Schema
- Aufbau:
- Data Type Definitions
  - Beschreibung der Datentypen, die in Nachrichten vorkommen
- Abstract Operations
  - Die Operationen, die durch die Nachrichten ausgelöst werden
- Service Bindings
  - Abbildung der Nachrichten auf Transportprotokolle

WSDL – Web Services Description Language

- Data Type Definitions
  - types – verwendete Datentypen als XML Schema
  - message – Definition der Nachrichten mit Parametern
- Abstract Operations
  - operation – Definition, welchem Dienst (Prozedur, Queue etc) die Nachricht zur Behandlung übergeben werden soll
  - portType – Abstrakter (Service-) Port als Menge von Operationen
  - binding – Abbildung eines Port Type auf einen konkreten Transportmechanismus (Protokoll)
- Service Bindings
  - port – Netzwerkadresse für ein Binding
  - service – Menge von Port Types, die gesamthaft einen logischen Dienst darstellen.



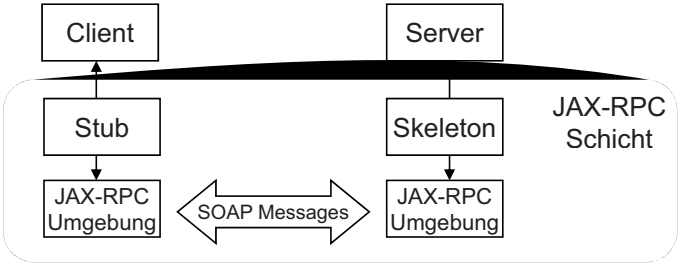
UDDI

- Anforderungen
  - Veröffentlichen von Web Services
  - Finden von Web Services
- Anbieter von UDDI-Repositories: IBM, SAP und Microsoft
- Benutzung
  - Web-Interface
  - API z.B. JAXR (Java API for XML Registries)
- Beschreibung der Web Services mittels XML-Datenstrukturen
  - Business Entity: Kontakt, Beschreibung, Beziehung zu anderen Geschäftseinheiten, …
  - Service: Web Service oder andere Dienstleistungen
  - Binding: Technische Beschreibung, Access point URL, Verweis auf Spezifikation
  - …
- am wenigsten genutzter Standard

Java Web Service Developer Pack

- Im JWSDP u.a. enthalten:
  - Java Architecture for XML Binding (JAXB)
    - Generierung von Java Klassen aus DTDs
  - Java API for XML Processing (JAXP)
    - SAX (Simple API for XML Parsing)
      - ereignisgesteuerter Parser
    - DOM (Document Object Model)
      - Objekt Repräsentation in Form eines Baums
  - Java API for XML-based RPC (JAX-RPC)
    - Methodenaufrufe und Rückgabewerte als SOAP Nachricht
    - Erzeugung von Stubs und Ties (Skeletons) aus WSDL Beschreibung
  - SOAP with Attachments API for Java (SAAJ)
    - Erstellen und Verschicken von SOAP Nachrichten mit Anhängen
    - Asynchroner Nachrichtenaustausch

JAX-RPC



Thomas Stark: „J2EE“, Abb.12-2
Addison-Wesley, 2005

Lehrstuhl für Netzarchitekturen und Netzdienste
Institut für Informatik – Technische Universität München
Prof. Dr.-Ing. Georg Carle

Grundlagen: Rechnernetze und Verteilte Systeme

Kapitel 10: Netzsicherheit

Kryptographische Mechanismen und Dienste
IPSec, Firewalls

Prof. Dr.-Ing. Georg Carle
Lehrstuhl für Netzarchitekturen und Netzdienste
Technische Universität München
carle@net.in.tum.de
http://www.net.in.tum.de

Übersicht

Sicherheit im Internet

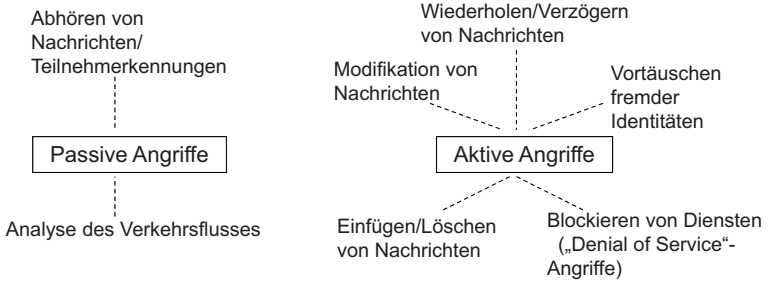
- Sicherheitsziele und Bedrohungen
- Sicherheitsmechanismen
- Firewalls
- Virtuelle Private Netze

Einleitung

- Früher:
  - Öffentliche Netze: abgeschlossen, zentral verwaltet
  - Internet: reines Forschungsnetz, kein lohnendes Angriffsziel, Benutzer vertrauen einander
- Heute:
  - Dezentralisierung öffentlicher Netze nach Deregulierung der Telekommunikationsmärkte
  - Kommerzielle Nutzung des Internets
- Folge:
  - Sicherheitsmechanismen werden zum unverzichtbaren Bestandteil moderner Kommunikationssysteme

Angriffsmöglichkeiten

- Passive Angriffe
  - Ablauf der Kommunikation nicht gestört
  - Aber unerlaubte Informationsbeschaffung
- Aktive Angriffe
  - Nachrichten werden verfälscht
  - Betrieb des Netzes wird verändert

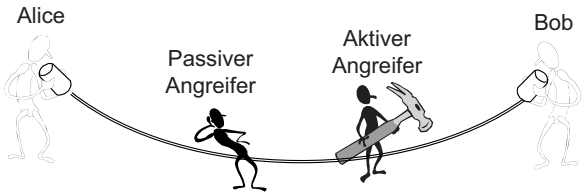


Sicherheitsanforderungen

- Authentizität
  - Angegebener Sender ist auch tatsächlicher Sender
- Vertraulichkeit
  - Ausspähen von Daten kann verhindert werden
  - ⇨ Sender verschlüsselt, und nur beabsichtigter Empfänger kann entschlüsseln
- Verbindlichkeit
  - Senden bzw. Empfangen von Daten kann nicht abgestritten werden
- Integrität
  - Empfänger kann Verfälschung von Daten erkennen
- Verfügbarkeit
  - Dienstnutzer kann Dienst auch tatsächlich nutzen

Einfaches Modell der Datenübertragung

- Passiver Angreifer: kann nur abhören, nicht manipulieren
  - Bedrohung für Vertraulichkeit
- Aktiver Angreifer: kann abhören, ändern, löschen, duplizieren
  - Bedrohung für Vertraulichkeit, Integrität, Authentizität
- Unterschied Authentizität/Verbindlichkeit:
  - Authentizität: Bob ist sicher, dass Daten von Alice kommen
  - Verbindlichkeit: Bob kann dies gegenüber Dritten beweisen



Bedrohungen

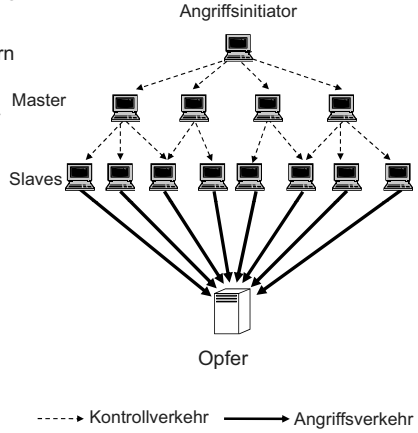
- Abhören übertragener Daten
- Modifizieren übertragener Daten
  - Ändern, Löschen, Einfügen, Umsortieren von Datenblöcken
- Maskerade
  - Vorspiegeln einer fremden Identität
  - Versenden von Nachrichten mit falscher Quelladresse
- Unerlaubter Zugriff auf Systeme
  - Stichwort „Hacking“
- Sabotage (Denial of Service)
  - gezieltes Herbeiführen einer Überlastsituation
  - „Abschießen“ von Protokollinstanzen durch illegale Pakete

Angriffstechniken

- Anzapfen von Leitungen oder Funkstrecken
- Zwischenschalten (man-in-the-middle attack)
- Wiedereinspielen abgefangener Nachrichten (replay attack)
  (z.B. von Login-Nachrichten zwecks unerlaubtem Zugriff)
- gezieltes Verändern/Vertauschen von Bits oder Bitfolgen
  (ohne die Nachricht selbst entschlüsseln zu können)
- Brechen kryptographischer Algorithmen
  Gegenmaßnahmen:
  - Keine selbst entwickelten Algorithmen verwenden,
    sondern nur bewährte und als sicher geltende Algorithmen!
  - Auf ausreichende Schlüssellänge achten
  - Möglichkeiten zum Auswechseln von Algorithmen vorsehen

Angriffsbeispiel: Verteilte Denial-of-Service-Angiffe

- Zahlreiche kompromitierte Systeme
  - Mehrere 1000 “Bot-Netze”
    mit mehreren 10.000 Rechnern
- Master-Systeme
  - Erhalten Befehle vom Initiator
  - Kontrollieren Slave-Systeme
- Slave-Systeme
  - Führen Angriff durch



Sicherheitsdienste

- Authentisierung
  - Authentisierung der Kommunikationspartner (Entity Authentication)
  - Authentisierung des Datenursprungs (Data Origin Authentication)
- Zugriffskontrolle
  - Schutz einer Ressource vor unberechtigtem Zugriff
- Abhörsicherheit
  - kein Fremder soll Daten mitlesen können
- Verbindlichkeit bzw. Nicht-Zurückweisbarkeit (Non-Repudiation)
  - Sender bzw. Empfangen kann nachgewiesen werden
- Datenintegrität (Fälschungssicherheit)
  - Echtheit der Daten soll garantiert sein
- Verfügbarkeit
  - Schutz eines Dienstes vor Blockierung
- Privatheit
  - Anonymisierung bzw. Pseudonymisierung ist möglich
- Autorisierung
  - darf jemand mit der vorgegebenen Kennung einen Dienst nutzen?
- Vertraulichkeit
  - Schutz der Daten vor unberechtigter Offenlegung

Sicherheitsmechanismen: Begriffe

- Verschlüsselung
  - Kodierung der Daten mit Hilfe eines Schlüssels
  - Dekodierung nur mit zugehörigem Schlüssel möglich
  - Oder durch gezielten, sehr hohen Rechenaufwand
  - Verfahren:
    - Symmetrische Verschlüsselung: DES, Triple-DES, AES, RC4, RC5, IDEA
    - Asymmetrische Verschlüsselung: RSA
- Schlüsselaustausch und Schlüsselverwaltung
  - Diffie-Hellman-Schlüsselaustausch: Protokoll, mit dem zwei
    Kommunikationspartner einen geheimen Schlüssel erzeugen können
  - Standard: X.509 - Standard für Public-Key-Infrastruktur ⇨ Zertifikate
- Firewall
  - Filterfunktion zwischen verschiedenen Netzwerken
  - Erlaubt Abschottung zum Internet
  - Auch intern wichtig: über 50% aller Angriffe kommen von eigenen
    Mitarbeitern!

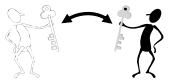
Erbringung von Sicherheitsdiensten

- Überwiegend mit kryptographischen Mechanismen:
  - Authentisierung
    - von Datenpaketen (data origin authentication)
    - von Systemen/Benutzern (entity authentication)
  - Integritätssicherung (integrity protection)
    - häufig kombiniert mit Daten-Authentisierung
  - Verschlüsselung (encryption)
  - Schlüsselaustausch (key exchange)

- Überwiegend Ohne kryptographische Mechanismen:
  - Zugriffskontrolle (access control)
  - Einbruchserkennung (intrusion detection)

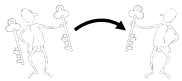
(A)symmetrische Kryptographie

- **Symmetrische Kryptographie**



- Instanzen besitzen gemeinsamen geheimen Schlüssel
  - **Vorteile**:
    - geringer Rechenaufwand
    - kurze Schlüssel
  - **Nachteile**:
    - Schlüsselaustausch schwierig
    - keine Verbindlichkeit

- **Asymmetrische Kryptographie** (**Public-Key-Kryptographie**)

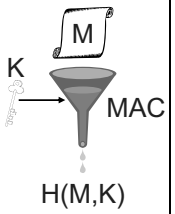


- Schlüsselpaar aus privatem und öffentlichem Schlüssel
  - **Vorteile**:
    - öffentliche Schlüssel sind relativ leicht verteilbar
    - Verbindlichkeit möglich
  - **Nachteile**:
    - hoher Rechenaufwand
    - längere Schlüssel

Authentisierung (1)

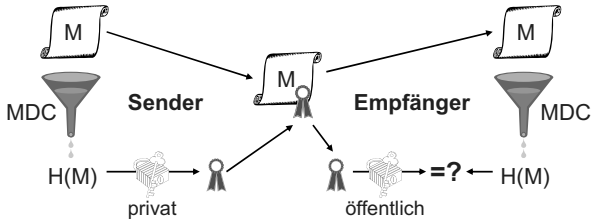
- **Kryptographische Hash-Funktion** (Modification Detection Code bzw. Message Digest Code, MDC):
  - Nachricht M (beliebig lang) → Hash-Wert H(M)
  - Wichtig: „Einweg"-Eigenschaft:
    keine Kollisionen effizient erzeugbar
    Kollision: M, M' mit H(M)=H(M')
  - Beispiele: MD5, SHA-1, RIPEMD-160

- **Schlüsselabhängige Hash-Funktion** (Message Authentication Code, MAC):
  - Nachricht M, Schlüssel K → Hash-Wert H(M,K)
  - kann aus MDC konstruiert werden:
    HMAC (RFC 2104), z.B. HMAC-MD5
    $H(K \text{ xor } pad_1, H(K \text{ xor } pad_2, M))$



Authentisierung (2)

- **Digitale Signatur**
  - Hash-Wert H(M) wird mit privatem Schlüssel signiert
  - Empfänger überprüft Signatur mit öffentlichem Schlüssel
  - kann auch Verbindlichkeit garantieren
  - wichtigste Algorithmen: RSA, DSA, ElGamal
  - min. Schlüssellänge: 1024 bit
    (160 bit bei DSA-Variante mit elliptischen Kurven)

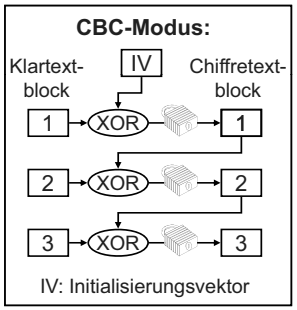


Authentisierung (3)

- **Authentisierung/Integritätssicherung von Datenpaketen**
  - Anhängen einer Sequenznummer zur Reihenfolgesicherung (falls nicht ohnehin vorhanden)
  - Anhängen von MAC oder Signatur, berechnet aus Daten, Sequenznummer und Schlüssel
- **Authentisierung von Systemen/Benutzern**
  - **nicht-kryptographisch**: Benutzername/Passwort (unsicher!), Einmalpassworte, biometrische Verfahren (z.B. Fingerabdruck)
  - **kryptographisch**: Login-Nachrichten mit MAC oder Signatur
  - Sicherung gegen **Wiedereinspielen** alter Login-Nachrichten:
    - Zeitstempel (synchrone Uhren nötig)
    - Zufallszahlen (Challenge/Response-Verfahren)

Verschlüsselung (symmetrisch)

- **Symmetrische Verschlüsselungsalgorithmen**
  - minimale derzeit sichere Schlüssellänge: 80 bit
  - als sicher geltende Algorithmen: AES sowie Triple-DES
- **Betriebsarten**
  - Gängige Algorithmen (Blockchiffren) arbeiten blockweise (meist 64 bit)
  - **Electronic Codebook** (ECB)
    - blockweise Verschlüsselung
    - Nachteil: Gleiche Klartextblöcke werden auf gleiche Chiffretextblöcke abgebildet
  - **Cipher Block Chaining** (CBC)
    - sicherer, da jeder Block vom Vorgänger abhängt
  - Weitere Betriebsarten z.B. zur byteweisen Verschlüsselung sowie zur Vorratsberechnung der kryptografischen Algorithmen



Verschlüsselung (asymmetrisch)

- Asymmetrische (Public-Key-) Verschlüsselungsalgorithmen
  - minimale derzeit sichere Schlüssellänge: 1024 bit
  - als sicher geltender Algorithmus: RSA
  - relativ langsam
- In der Praxis: Hybride Systeme
  - Zunächst: Benutzer-Authentisierung und Austausch eines Sitzungsschlüssels (symmetrisch oder Public-Key)
  - Danach: Authentisierung/Verschlüsselung der Nutzdaten mit Sitzungsschlüssel (symmetrisch)
  - Bei langen Sitzungen sollte Sitzungsschlüssel gelegentlich ausgewechselt werden (z.B. stündlich)

Schlüsselaustausch

- Symmetrisch: mit Hilfe eines Key Distribution Center (KDC)
  - KDC hat geheimen Schlüssel mit jedem Benutzer/Dienst
  - KDC authentisiert Benutzer und verteilt Sitzungsschlüssel
  - Beispiel: Kerberos (RFC 1510)
- Asymmetrisch: 2 Möglichkeiten:
  - Verschlüsseln/Signieren des Sitzungsschlüssels mit beliebigem Public-Key-Algorithmus
  - Diffie-Hellman-Schlüsselaustausch
    - Diffie-Hellman-Schlüsselaustausch allein ist bei Man-In-The-Middle-Angriff nicht sicher
    - Zusätzliche Authentisierung nötig!

Secure Shell (SSH)

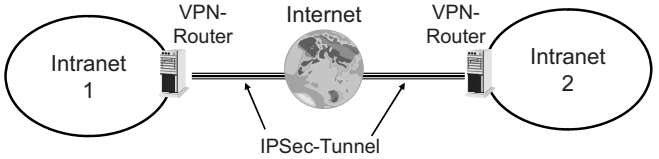
- Aufgabe: sichere entfernte Rechnernutzung (remote login)
  - rsh/rlogin haben keine Authentisierung
  - telnet überträgt Passworte ungeschützt
- Funktionsweise:
  1. Austausch eines Sitzungsschlüssels (Diffie-Hellman) und Server-Authentisierung (digitale Signatur) danach: symm. Verschlüsselung + MAC für alle Pakete
  2. Benutzer-Authentisierung (dig. Signatur oder Passwort)
- Zusätzliche Funktionalität:
  - Verschlüsselte Dateiübertragung mit scp
  - Verschlüsselte Tunnel für einzelne TCP-Ports
  - automatische Einrichtung eines X11-Tunnels
- Versionen: 1.0, 2.0 zueinander inkompatibel (Infos: www.ssh.fi)

Secure Socket Layer (SSL)

- Aufgabe: Verschlüsselung/Datenintegrität für einzelne Sockets
  - Haupteinsatzgebiet: verschlüsselte HTTP-Verbindungen (https)
- Funktionsweise:
  - Austausch eines Sitzungsschlüssels (Diffie-Hellman)
  - optional Server-/Benutzer-Authentisierung (digitale Signatur)
  - danach: Verschlüsselung + MAC für alle Pakete
- Versionen:
  - von Netscape: SSL 1.0 bis SSL 3.0
  - TLS - Transport Layer Security (RFC 2246) basierend auf SSL 3.0

IP Security (IPSec)

- Aufgabe: sicheres Tunneln von IP-Paketen
  - Verschlüsselung am Tunneleingang, Entschlüsselung am Ausgang
  - kann z.B. für das gesamte VPN automatisch durchgeführt werden oder nur für bestimmte Anwendungen
- Beispiel: IP Security
  - Funktionsweise:
    - MAC und/oder symm. Verschlüsselung
    - 2 Paketformate: AH (RFC 2402), ESP (RFC 2406)
  - Produkte:
    - FreeS/WAN (www.freeswan.org)
    - Cisco VPN-Produkte
    - Windows VPN-Funktionen



IPSec: Authentication Header und Encapsulaing Security Payload

- Authentication Header

  Authentifizierung, Datenintegrität durch MAC
  - Transportmodus
    - Keine Veränderung der Adressen, falls direkte Kommunikation
  - Tunnelmodus
    - Neue IP-Adressen, zwischen beliebigen Partnern

Transportmodus

IP-Kopf	AH	Nutzdaten

Tunnelmodus

Neuer IP-Kopf	AH	Alter IP-Kopf	Nutzdaten

- Encapsulating Security Payload
  - Authentifizierung, Datenintegrität, Privatheit durch Verschlüsselung und/oder MAC
  - Transportmodus
  - Tunnelmodus

Transportmodus

IP-Kopf	ESP-Kopf	Nutzdaten	ESP-Anhang

Tunnelmodus

Neuer IP-Kopf	ESP-Kopf	Alter IP-Kopf	Nutzdaten	ESP-Anhang

Zertifikate

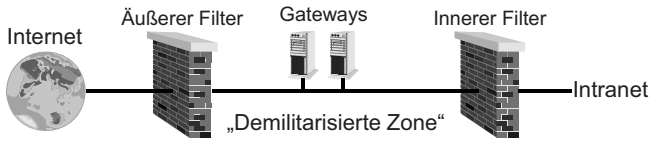
- X.509 - Standard für Public-Key-Infrastruktur
- hierarchisches System von vertrauenswürdigen Zertifizierungsstellen (engl. certificate authority, kurz CA)
- Webbrowser beinhalten eine vorkonfigurierte Liste vertrauenswürdiger Zertifizierungsstellen, deren ausgestellten SSL-Zertifikaten der Browser vertraut.
- Zertifizierungsstelle kann ungültige Zertifikate in Zertifikatsperrlisten (certificate revocation list, kurz CRL) führen

Struktur eines X-509-v3-Zertifikats

- Zertifikat
  - Version
  - Seriennummer
  - Algorithmen-ID
  - Aussteller
  - Gültigkeit
    - von
    - bis
  - Subject
  - Subject Public Key Info
    - Public-Key-Algorithmus
    - Subject Public Key
  - Eindeutige ID des Ausstellers (optional)
  - Eindeutige ID des Inhabers (optional)
  - Erweiterungen
    - …
- Zertifikat-Signaturalgorithmus
- Zertifikat-Signatur

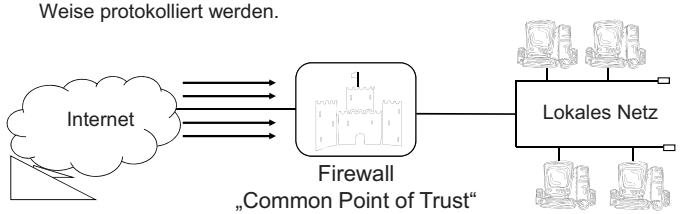
Zugriffskontrolle

- Auf Anwendungsebene: System von Zugriffsrechten
  - Beispiele: Unix/NT-Dateirechte, SNMP-Objektrechte
- Auf Netzwerk-/Transportebene: Firewalls
  - Paketfilter filtern nach Quell/Zieladresse + Ports (TCP/UDP)
  - Unterscheidung: ingress/egress filtering (inbound / outbound packets)
  - Anwendungs-Gateways (Zugriffskontrolle, Protokollierung)
  - Kann mit privaten Adressen und Adressumsetzung (NAT) kombiniert werden
  - Probleme mit manchen Protokollen (z.B. FTP, H.323)

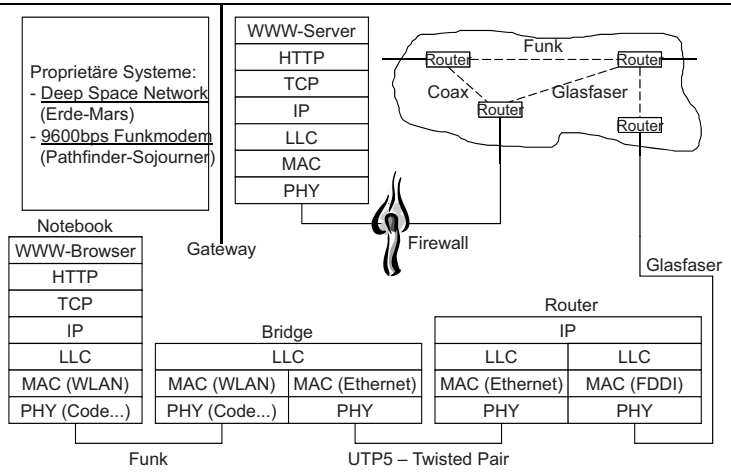


Firewall

- Ziel: Schutz des lokalen Netzes hauptsächlich gegenüber externen (aktiven/passiven) Angriffen („keep the good bits in and the bad bits out“)
- Vorteile:
  - Kosten: Die zentrale Realisierung von Sicherheitsmechanismen ist wesentlich kostengünstiger als die Absicherung jedes einzelnen Rechners.
  - Wirkung: Die Sicherheitspolitik eines Unternehmens kann sehr einfach durchgesetzt bzw. angepasst werden.
  - Sicherheit: Es existieren nur wenige Angriffspunkte im Netz (im Idealfall nur das Firewall-System selbst).
  - Überprüfbarkeit: Sämtliche Kommunikationsvorgänge können auf einfache Weise protokolliert werden.



Beispiel: Firewall zum Schutz eines WWW-Servers



Firewalls im Internet

- Firmen, Behörden, Privatpersonen, Universitäten sind von den Protokollen TCP/IP her gleichberechtigt an das Internet angebunden

⇨Das *interne* Netz von unerwünschten Zugriffen von außen schützen:

- am sichersten ist nur die physikalische Trennung zwischen Rechnern am Internet und firmeninternen Rechnern
- Firewalls sind meist Router, die Pakete anhand der IP-Adresse und Port-Nummer herausfiltern können (zusätzliche Vermerke in einer Log-Datei möglich)
  - *Beispiel:* Ausfiltern von Paketen mit dem Port 80 verhindert den Zugriff auf normale WWW-Server; werden z.B. 129.13.*x.y* Adressen gefiltert, kann kein Rechner aus diesem Subnetz auf etwas zugreifen!
- Außer Paketfilter sind oft noch Anwendungsgateways und Adressübersetzung integriert
  - Umsetzung zwischen verschiedenen mail-Systemen
  - dynamische Abbildung einer IP-Adresse auf viele verschiedene interne Endsysteme

Firewall

- Kann auf verschiedenen Protokollschichten arbeiten, viele unterschiedliche Funktionen anbieten
- Schicht 2
  - Filtern nach MAC-Adressen
  - lässt z.B. nur Adapter zu, die in der Firewall bekannt sind
- Schicht 3
  - Filtern nach IP-Adressen
  - kann z.B. Verkehr nach Herkunft und Ziel filtern
- Schicht 4
  - Filtern nach Ports
  - kann z.B. Pakete je nach Anwendung filtern
- Anwendungsschicht - Proxy
  - Virenscanner, Inhaltsüberprüfung (Text, Bilder), WWW-Adressen, ...

Firewall Beispiel

- Gezielte Aktionen in Abhängigkeit von Adressen und Anwendungen
- Spezielle Firewall-Lösungen mit hoher Leistungsfähigkeit erhältlich
- Sicherheit aber nur so gut wie die Wartung!

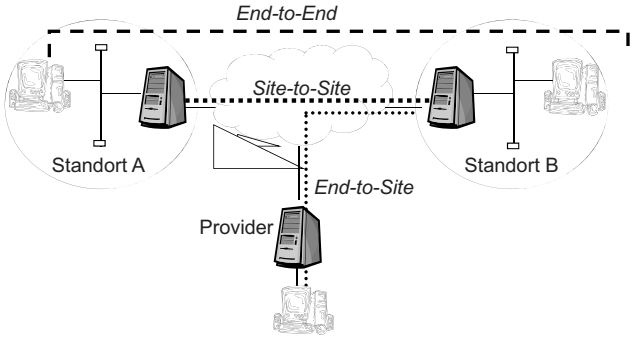
Quelladresse	Zieladresse	Dienst	Aktion	Protokoll
beliebig	Web-Server	http	akzeptieren	kurz
Intranet	Intranet	smtp	verschlüsseln	normal
Intranet	alle	http	akzeptieren	kurz
Extranet	Intranet	smtp, http	akzeptieren, Viren-Scan	normal
alle	alle	alle	verwerfen	Alarm, lang

Firewall-Mechanismen

- Analyse der ein-/ausgehenden Datenpakete (Packet Filtering)
  - Kontrolle der Felder des Paketkopfes, z.B. Flags, IP-Adresse und Portnummer
  - Erlaubter/nicht erlaubter Datenverkehr ist in Access-Liste vermerkt
    - eingehend: deny *.*.*.*, 23 blockiert telnet
    - ausgehend: permit 129.13.*.*, 80 erlaubt http nur für Rechner mit IP=129.13.x.y
- Adressumsetzung (Network Address Translation, NAT)
  - Rechner im lokalen Netz von außen nicht erreichbar (z.B. 192.168.x.y)
  - Firewall/Gateway nimmt Abbildung auf gültige Adressen vor
- Proxy-Dienste (Proxy Services)
  - Endsysteme im geschützten Netz nur über (Application-)Gateway erreichbar
  - Für jede zulässige Anwendung fungiert Gateway als Proxy
  - Verbindungsaufbau zu Zielrechner nur nach Authentifikation
  - Filterung auf Anwendungsebene (z.B. nur ftp-get aber kein ftp-put)
  - Detaillierte Rechteverwaltung und Protokollierung

Virtuelle private Netze (VPNs)

- **Ziel:** Gewährleistung eines gesicherten Datenaustauschs zwischen entfernten Kommunikationspartnern/Standorten über (ungesicherte) Transit-Netze (z.B. das Internet) durch Authentifizierung und Verschlüsselung.

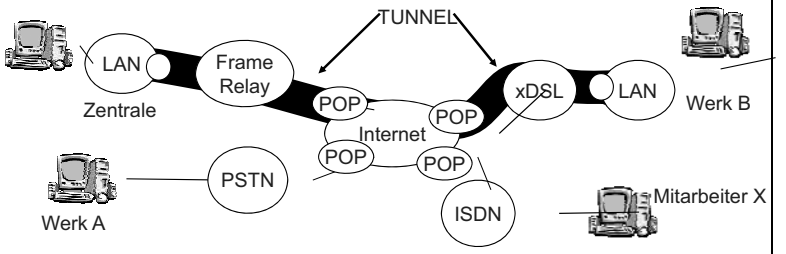


VPNs im Internet

Anwendung	Absicherung einzelner Anwendungsprotokolle z.B. Secure Sockets Layer (SSL)
Transport	Absicherung der Nutzdaten von TCP und UDP durch Modifikation der Schicht 4 in Endsystemen (meist proprietär).
Vermittlung	Absicherung der Nutzdaten auf Ebene von IP durch Modifikation des IP-Stacks in allen beteiligten Systemen (z.B. IPSec).
Sicherung	Absicherung der Nutzdaten auf Ebene der Sicherungsschicht, z.B. für Einwählverbindungen: Point-to-Point-Tunneling-Protocol (PPTP), Layer-2-Tunneling-Protocol (L2TP).
Bitübertragung	

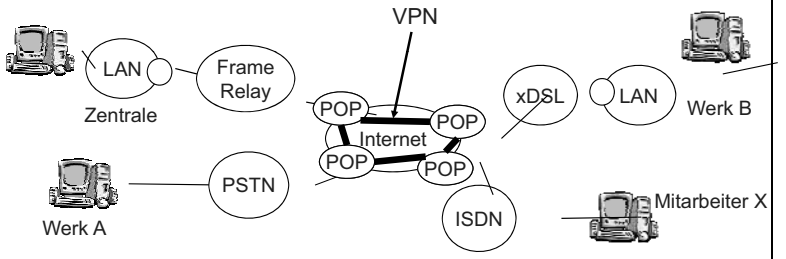
Virtuelle Private Netze

- Lösung: Virtuelle Private Netze (VPN)
  - VPN als logisches Netz
  - VPNs können auf verschiedenen Techniken basieren
    - Schicht-2-Tunneling: LAN-Pakete werden transparent über ein externes Netz transportiert
    - Schicht-3-Tunneling: IP-Pakete werden transparent über ein externes Netz transportiert



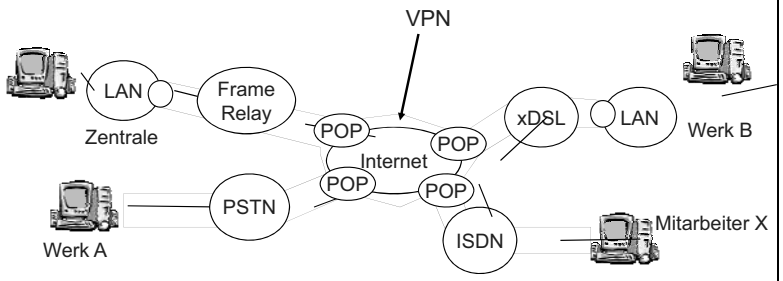
Ort eines VPN

- Outsourcing
  - VPN wird vom Netz-Provider zur Verfügung gestellt
  - Provider bietet Einwahlpunkte, zwischen diesen besteht ein VPN
  - kaum Hard-/Software auf Kundenseite nötig



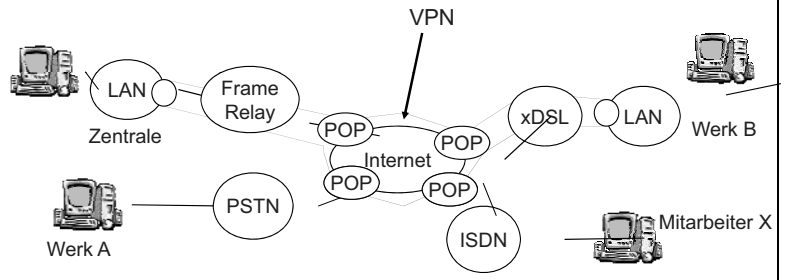
Ort eines VPN

- In-house VPN, VPN zwischen den Standorten
  - Tunnels werden z.B. zwischen den firmeneigenen Routern aufgebaut
  - Netz-Provider hat keinen Einblick in das VPN
  - Firma legt selbständig Sicherheit, Protokolle etc. fest
  - Software für Sicherheit, Tunneling, Verschlüsselung notwendig



Ort eines VPN

- Mischformen möglich
  - oftmals werden Tunnels zwischen Routern eingerichtet, nicht jedoch für den entfernten Zugriff
  - VPN endet am POP des Providers



Entfernter Zugriff auf das Intranet

- Außenanbindung
  - Außendienstmitarbeiter
  - Kunden, Lieferanten etc.
  - Informationen für alle
- Firewall zur Zugangsbeschränkung
  - Anbindung des Intranet an das Internet
  - Zugang von jedem beliebigen Rechner weltweit
  - Schutz durch eine Firewall, d.h. Filter für unerlaubte Daten
  - Software für Firewall benötigt, Rechenleistung auf Router wichtig
- Einwahlmöglichkeiten für den Außendienst
  - Erweiterung des VPN dynamisch bis hin zum Ort des Mitarbeiters
  - Einwahl via Modem (ISDN/analog)
  - Sicherheit durch Passwort, gesicherte Verbindung
  - Modem plus Software benötigt

Weitere Sicherheits-Themen

- E-Mail Sicherheitsproblem: SPAM
  - Webmail: Mit Bots lassen sich zahlreiche Benutzer-Konten erzeugen
  - Capatcha: Completely Automated Public Turing test to tell Computers and Humans Apart
  - DNS blacklisting
  - Spamer Virus
- Voice-over-IP Sicherheitsprobleme
  - SPIT Spam over IP-Telephony
  - DoS
  - Abhören und Modifikation
  - Missbrauch der Dienste (Fraud)
    - Nicht-Autorisierte oder Nicht-abrechenbare Ressourcen Nutzung
    - Impersonifizierung, gefälschte Identitäten

Lehrstuhl für Netzarchitekturen und Netzdienste
Institut für Informatik – Technische Universität München
Prof. Dr.-Ing. Georg Carle

Grundlagen: Rechnernetze und Verteilte Systeme

Kapitel 11: Nachrichtentechnik

Daten, Signal, Medien, Physik

Prof. Dr.-Ing. Georg Carle
Lehrstuhl für Netzarchitekturen und Netzdienste
Technische Universität München
carle@net.in.tum.de
http://www.net.in.tum.de

Übersicht

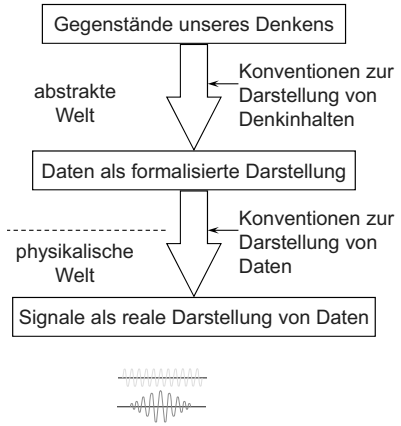
Ziele

- In diesem Kapitel wollen wir vermitteln
  - Signaltypen
  - Übertragungsarten und Übertragungsmedien
  - Übertragungsverfahren
  - Pulse-Code-Modulations-Technik (PCM)

Kapitelgliederung

Wiederholung: Der Begriff „Signal“

- *Signal*
  - Ein Signal ist die physikalische Darstellung (Repräsentation) von Daten durch charakteristische räumliche und/oder zeitliche Veränderungen der Werte physikalischer Größen.
  - Signale sind somit die reale physikalische Repräsentation abstrakter Darstellungen: der Daten.



11.1. Typen von Signalen
Im Folgenden...

- Einteilung von Signalen
- Beschreibung von Signalen
  - im Zeitbereich
  - im Frequenzbereich
- Übertragung von Signalen
  - Übertragungssystem
  - Einfluss des Mediums auf das Signal
- Übersicht über Übertragungsmedien

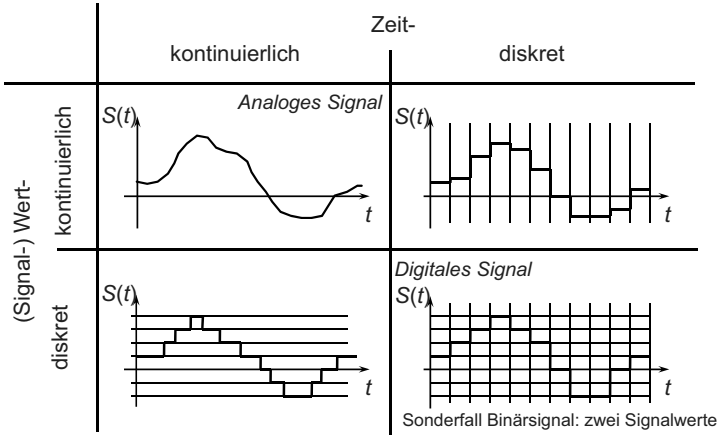
11.1.1. Einteilung von Signalen
Ortsabhängige vs. zeitabhängige Signale

- Ortsabhängige (räumliche) Signale
  - Beispiel: Bildverarbeitung
    - Kamera, Scanner, Monitor
  - Beispiel: Speichermedien
    - Optische Speicher (bedrucktes Papier, CD/DVD), magnetische Speicher (Festplatte)
- Zeitabhängige Signale
  - Beispiel: Signalverarbeitung und –übertragung
    - Telefon: Sprachsignal
- Orts- und Zeitabhängige Signale → Welle
  - Beispiel: Elektromagnetische Welle, Schall
- Grundsatz:
  - Jedes ortsabhängige Signal ist in zeitabhängiges Signal überführbar („Lesen“, Abtasten) und umgekehrt („Schreiben“, Aufzeichnen)
- Fokus in der Vorlesung auf zeitabhängigen Signalen und Wellen

Signalparameter

- Physikalische Kenngrößen eines Signals, deren Wert oder Werteverlauf die Daten repräsentieren
  - Bei *räumlichen* Signalen sind Werte des Signalparameters Funktion des Ortes, z.B. des Speichermediums.
  - Bei *zeitabhängigen* Signalen sind Werte des Signalparameters S Funktion der Zeit $S = S(t)$.
- Generische Einteilung *zeitabhängiger* Signale in vier Klassen:
  - zeitkontinuierliche, signalwertkontinuierliche Signale
  - zeitdiskrete, signalwertkontinuierliche Signale
  - zeitkontinuierliche, signalwertdiskrete Signale
  - zeitdiskrete, signalwertdiskrete Signale

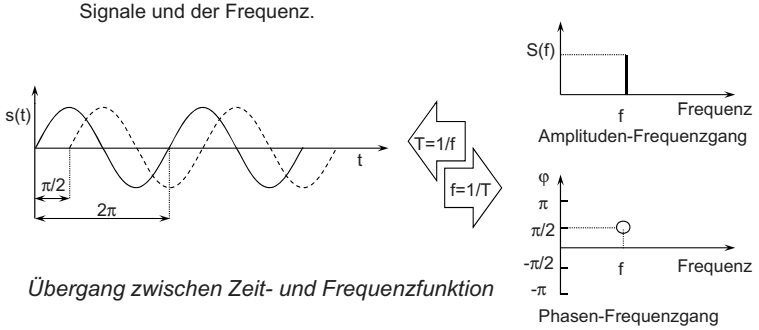
Signalklassen



Sonderfall Binärsignal: zwei Signalwerte

11.1.2. Beschreibung von Signalen Zeitdarstellung/Frequenzdarstellung

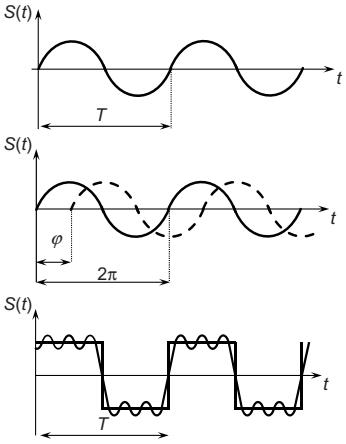
- **Zeitfunktion (Zeitdarstellung):**
  - Die Zeitfunktion ist eine Zuordnung von Signalwert und Zeit.
- **Frequenzfunktion (Frequenzgang, Spektrum):**
  - Die Frequenzfunktion ist eine Zuordnung von Werten sinusförmiger Signale und der Frequenz.



Übergang zwischen Zeit- und Frequenzfunktion

Periodische Signale

- **Kenngrößen periodischer Signale:** Periode T, Frequenz $1/T$, Amplitude $S(t)$, Phase φ
- Beispiele:
  - Sinus-Schwingung
  - Phasendifferenz φ
  - Rechteck-Schwingung (zeitdiskret „idealisiert“)



Periodische Signale: Fourier-Analyse

- Jede **periodische** Funktion kann durch die Summe von Sinus- und Kosinusfunktionen dargestellt werden (Fourier-Reihe).

$$g(t) = \frac{1}{2}c + \sum_{n=1}^{\infty} a_n \sin(2\pi nft) + \sum_{n=1}^{\infty} b_n \cos(2\pi nft)$$

  - mit $f=1/T$ Grundfrequenz, a_n und b_n Amplituden von Sinus bzw. Kosinus der n-ten Harmonischen, $c/2$ Gleichanteil
- Berechnung der Fourier-Koeffizienten:

$$c = \frac{2}{T}\int_0^T g(t)\,dt$$

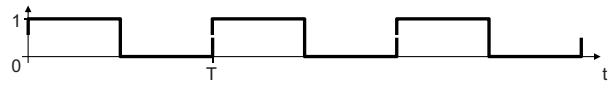
$$a_n = \frac{2}{T}\int_0^T g(t)\sin(2\pi nft)\,dt$$

$$b_n = \frac{2}{T}\int_0^T g(t)\cos(2\pi nft)\,dt$$

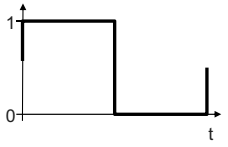
  - Signalleistung der n-ten Harmonischen: $a_n^2 + b_n^2$

Periodische Signale: Fourier-Analyse

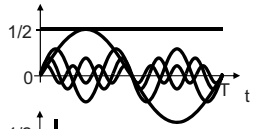
- Fourier-Reihe einer idealen Rechteckschwingung mit Periode T:



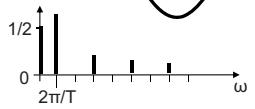
$$g(t) = \frac{1}{2} + \frac{2}{\pi}\left[\sin(\omega t) + \frac{1}{3}\sin(3\omega t) + \frac{1}{5}\sin(5\omega t) + ...\right] \quad \text{mit} \quad \omega = \frac{2\pi}{T}$$



Harmonische im Zeitbereich:



Fourier-Spektrum:



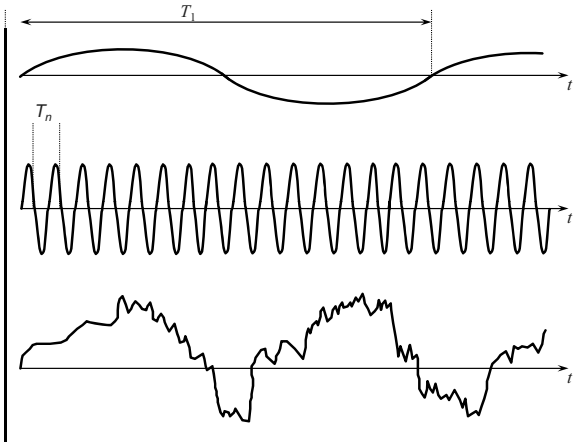
- unendlich viele Fourierkoeffizienten ungleich null ➔ unendliche Bandbreite

Zusammengesetzte Signale

Komponente mit niedriger Frequenz (feste Amplitude)

Komponente mit hoher Frequenz (feste Amplitude)

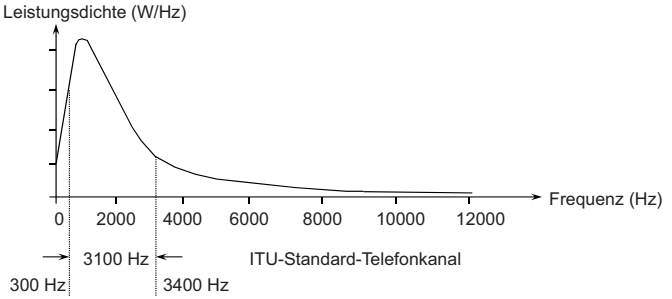
Zusammen-gesetztes Sprachsignal mit gemischten Frequenzen und Amplituden



Frequenzspektrum eines Signals

- **Bandbegrenztes Signal:**
  - Signale können ein „natürlich“ begrenztes - meist kontinuierliches - Frequenzspektrum umfassen oder durch technische Mittel auf einen Ausschnitt ihres Spektrums begrenzt werden (Bandbreite).

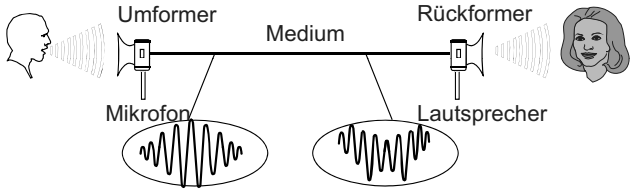
Kontinuierliches - akustisches - Frequenzspektrum der menschlichen Stimme und Bandbreite des analogen ITU-Standardtelefonkanals



Signalumformung akustisch-elektrisch

- Beispiel: Telefon
  - zeitabhängiges Signal, physikalische Größe

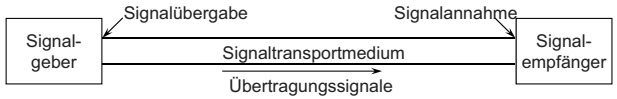
analoges akustisches Signal ➔ analoges elektrisches Signal ➔ analoges akustisches Signal



Klassisches Modell des Übertragungssystems Telefon

11.2. Übertragungssystem: Grundlagen, Begriffe

- **Signalübertragung:**
  - Grundlage jeder Kommunikation
  - Transport von Signalen über ein geeignetes Medium, das diese Signale über eine räumliche Distanz weiterleitet (➔ Welle).



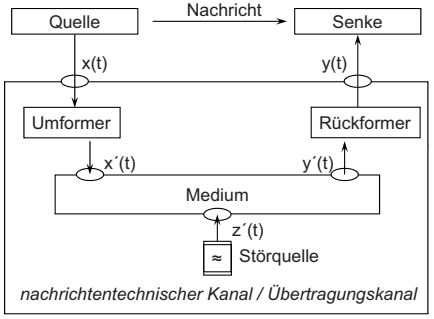
Verkürzender Sprachgebrauch:

Übertragungssignal	= Signal
Signaltransportmedium/Übertragungsmedium	= (physikalisches) Medium
Signalgeber, Signalquelle	= Sender
Signalempfänger, Signalsenke	= Empfänger
physikalisch-technisches Transportsystem für Signale	= Übertragungsweg

Signalübertragung wird in der Nachrichtentechnik als Nachrichtenübertragung bezeichnet.

Übertragungssystem: physikalisches Medium

- Verwendung eines physikalischen Mediums zur Übertragung von Nachrichten.



Primärsignale x(t), y(t):
quellen-/senkenbezogene physikalische Größen.

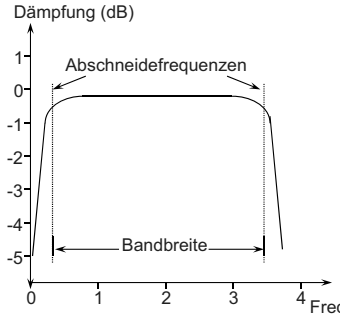
Signale x´(t), y´(t), z´(t):
leitungsbezogene physikalische Größen.

Physikalisches Medium,
z.B. elektrische Leitung:
y´(t) = F(x´(t);z´(t))

Bandbegrenztes Medium

Bandbreite eines Mediums:

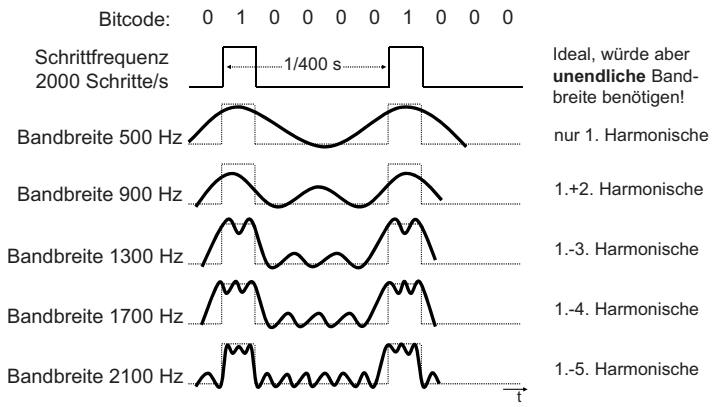
- Signaltransportmedien bzw. Übertragungssysteme übertragen stets nur ein endliches Frequenzband.



Bandbreite von Übertragungswegen:

- Bandbreite in Hz: Frequenzbereich, der über ein Medium (einschließlich der im Übertragungssystem enthaltenen Filter, Verstärker usw.) übertragen werden kann.
- Bandbreite ergibt sich aus der Differenz der höchsten und niedrigsten übertragbaren Frequenzen.
- Wegen nicht-idealer Bandbegrenzungen Festlegung von Abschneidefrequenzen.
- Signale müssen an die Übertragungs-charakteristik des Mediums angepasst werden.

Einfluss der Bandbreite eines Übertragungssystems auf ein digitales Signal



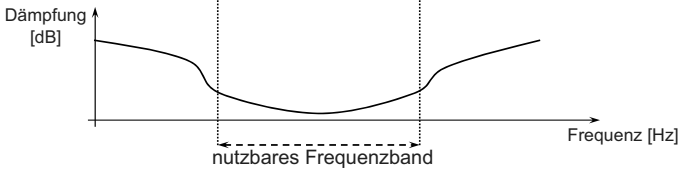
- Später:
  - Nyquist-Theorem zur Ermittlung der notwendigen minimalen Bandbreite zur Übertragung zeitdiskreter Signale mit gegebener Schrittgeschwindigkeit

Fortpflanzungsgeschwindigkeit von Signalen

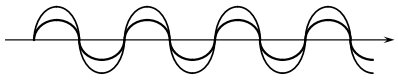
- Optimum: Lichtgeschwindigkeit (c = $3*10^8$ m/s) im Vakuum
- Ausbreitungsgeschwindigkeit auf Leitungen: etwa 2/3 c = $2*10^8$ m/s
- Durch die begrenzte Fortpflanzungsgeschwindigkeit hat das Medium eine Speicherkapazität.
- Beispiel: Datenübertragung von MIT nach Berkeley:
  - Strecke: 5000 km; Signallaufzeit: ca. 25 ms (5000 km / $2*10^8$ m/s)
  - Round Trip Delay (RTT): ca. 50 ms (doppelte Signallaufzeit)
  - Bei einer Übertragungsrate von 100 kbit/s: 2500 bit Speicherkapazität
  - Bei einer Übertragungsrate von 1 Gbit/s: 25000000 bit ≈ 3 Mbyte

Kenngrößen medienbedingter Abweichungen

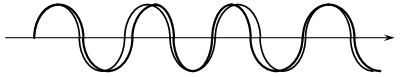
- **Bandbreite:** durch die Dämpfung vorgegeben



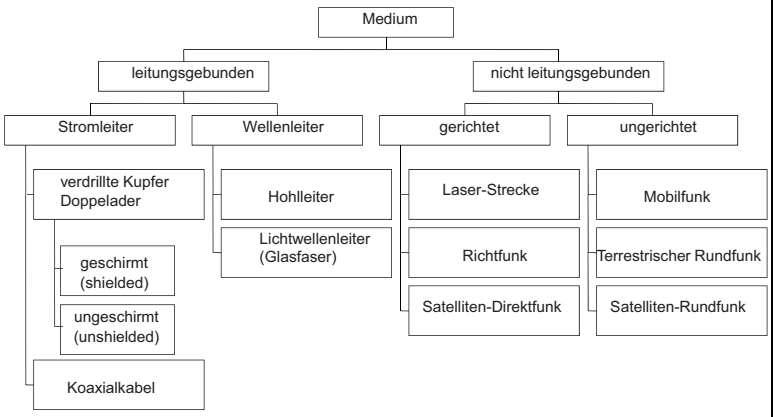
- **Dämpfungsverzerrung**: Amplitudenschwund, Amplitudensprünge



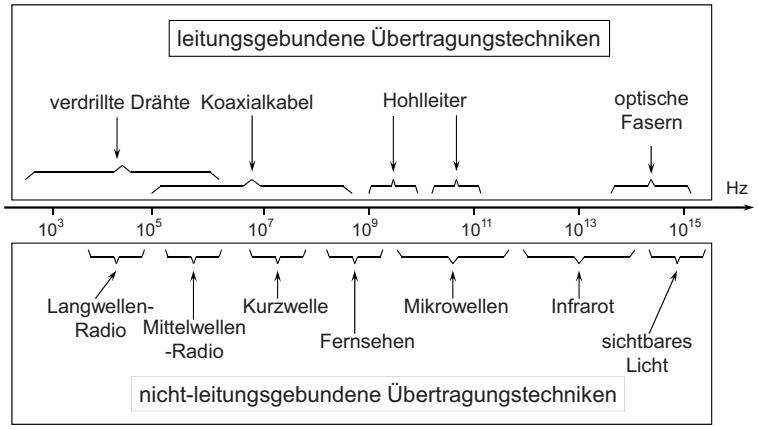
- **Laufzeitverzerrung**: Frequenzverwerfung, Phasenschwankungen (Jitter)



11.3. Medien: Klassifikation

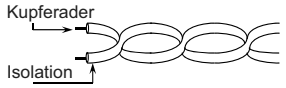


Nutzung des elektromagnetischen Spektrums



11.3.1. leitungsgebundene Medien - Kupfer-Doppeladern

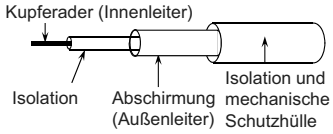
- *Kupfer-Doppeladern (DA)*
  - Verwendung z.B. im Teilnehmer-Anschlussnetz (Ortsnetz)
  - Leiterdurchmesser: 0,4 - 0,9 mm
  - Bandbreite: einige 100 kHz bis z. Zt. 600 MHz
  - internationaler Begriff: Unshielded Twisted Pair (UTP)
  - verschiedene Qualitätsklassen, z. B. UTP 3, 4, 5, 6 bis zu 2,5 Gbit/s vollduplex (sog. Kategorien, z.B. CAT 5)
  - auch mit zusätzlicher Kupferummantelung (STP, shielded twisted pair)
  - ggf. auch zusätzliche Gesamtabschirmung in einem Kabel mit mehreren Doppeladern: screened/unshielded twisted pair (S/UTP) und screened/shielded twisted pair (S/STP)



Hinweis: Die Verwendung einer Doppelader ist aus elektrischen Gründen notwendig. Hin- und Rückleiter im elektrischen Stromkreis!

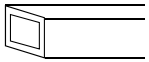
Koaxialkabel

- *Koaxialkabel (coax)*
  - Gehören zur Kupferkabeltechnik, da Innenleiter aus Kupfer besteht.
  - *Außenleiter* umschließt *Innenleiter* zylindrisch.
  - Dazwischen befindet sich ein *Dielektrikum* aus Kunststoffen oder Gasen.
  - Die Signalausbreitung erfolgt im Dielektrikum zwischen den beiden Leitern.
  - Unterscheidung durch Angabe Verhältnis Innenleiter zu Außenleiter, z.B.
    - ITU 2,6/9,5 mm
  - Bandbreite: bis 900 MHz

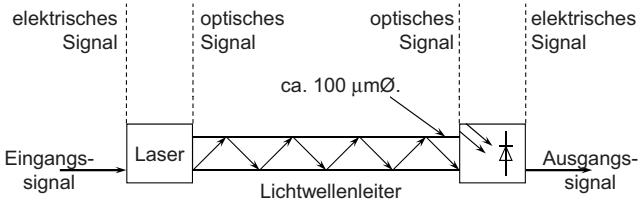


Hohlleiter

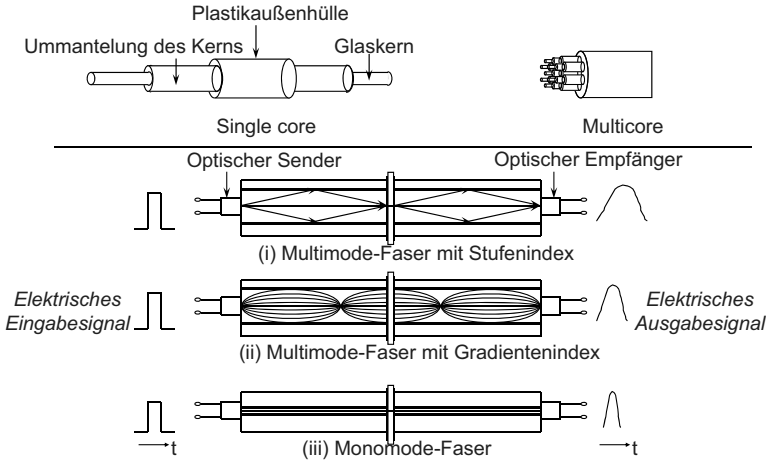
- *Hohlleiter* sind
  - mit Luft gefüllte, gestreckte metallische Hohlkörper
  - mit runden, elliptischen oder rechteckigen Querschnitten.
  - Sie bewirken eine geführte Ausbreitung höchstfrequenter elektromagnetischer Wellen (Mikrowellen) im Inneren des Hohlkörpers durch fortlaufende Reflexion.
  - Sie sind allerdings heutzutage teilweise von Lichtwellenleitern abgelöst.
  - Einsatzorte noch in der Richtfunktechnik (insb. Zuleitung zu Antennen)
  - Die Mindestbreite eines Rechteckhohlleiters: halbe Wellenlänge der übertragenen Frequenz
    Dazugehörige Wellenlänge: Grenzwellenlänge $\lambda_c = 2 \cdot a$
    (*a:* längere Seite des Rechteckhohlleiterquerschnitts)



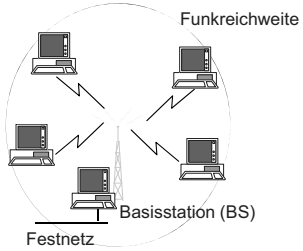
Lichtwellenleiter (Glasfaser)



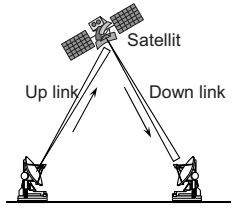
Glasfaser - Typen



11.3.2. nicht leitungsgebundene Medien Funk- und Satellitentechnik



- Medium: Elektromagnetische Welle (10^4 - 10^9 Hz)
- Daten werden aufmoduliert
- Eingeschränkte Reichweite, je nach Ausgangsleistung der BS und örtlichen Gegebenheiten
- Datenrate: Einige 10 kbit/s bis 10 Mbit/s pro Benutzer



- Medium: Elektromagnetische Welle (10^9 - 10^{11} Hz)
- Transponder im Satellit empfängt auf einem Kanal, sendet auf einem anderen.
- Mehrere Transponder pro Satellit
- Hohe Bandbreite (500MHz) pro Kanal

Der Begriff „digital“

- *Digitale Daten (Beispiele)*
  - Zahlen, Schriftzeichenalphabet, Binärcodes usw.
- *Digitale Signale*
  - Zeit- und wertdiskrete Signale
- *Digitale Übertragungssysteme*
  - Übertragungssysteme, die nur für digitale Daten geeignet sind. Sie verstärken nicht - wie im Analogfall - Signalverläufe (einschließlich Störungen), sondern detektieren die den Signalstrom bildenden Digitaldaten (in der Regel Folgen von 0 und 1) und regenerieren die ursprünglichen Daten in neu erzeugte „perfekte“ Signalformen.
  - Rauscheinflüsse und Störungen werden eliminiert.
- Im Folgenden: Betrachtung digitaler Übertragungssysteme

11.4. Übertragungsverfahren Im Folgenden...

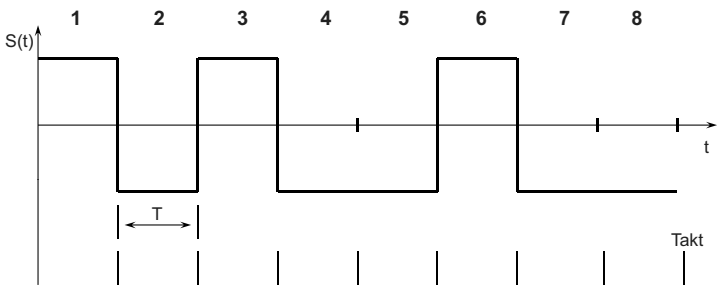
- Digitale Signalübertragung
  - Schrittgeschwindigkeit und Übertragungsgeschwindigkeit
  - Synchronisation
  - Fehlerquellen
  - Kanalkapazität
- Basisbandübertragungsverfahren
  - Leitungscodes
  - Schwingungsmodulation
- Mehrfachnutzung von Übertragungswegen
  - Multiplexverfahren
- Digitale Übertragung analoger Daten
  - PCM-Technik

11.4.1. Digitale Signalübertragung

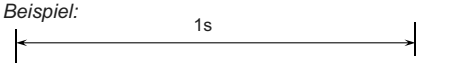
- **Schritt:**
  - Charakteristisch für zeitdiskrete Signale ist die Existenz eines minimalen Zeitintervalls T_{Min} zwischen aufeinanderfolgenden - möglichen - Änderungen der Signalkoordinate (Schrittdauer, kurz: Schritt als Signal definierter Dauer)
  - Wichtig: Digitales Signal mit fester Schrittdauer T (Schritt-Takt)
- **Isochrones (isochronous) Digitalsignal:**
  - Ein Digitalsignal ist isochron, wenn seine Kennzeitpunkte, d.h. die Zeitpunkte des Übergangs von einem Signalelement zum nächsten, in einem festen Zeitraster liegen.
- **Anisochrones (anisochronous) Digitalsignal:**
  - Ein nicht-isochrones Digitalsignal
- **Schrittgeschwindigkeit:**
  - bei isochronen Digitalsignalen: Kehrwert der Schrittdauer: 1/T
  - Einheit: baud = 1/s (nach Jean-Maurice-Emile Baudot, franz. Ingenieur)

Schrittgeschwindigkeit - Beispiel

Schrittfolge:



Beispiel:

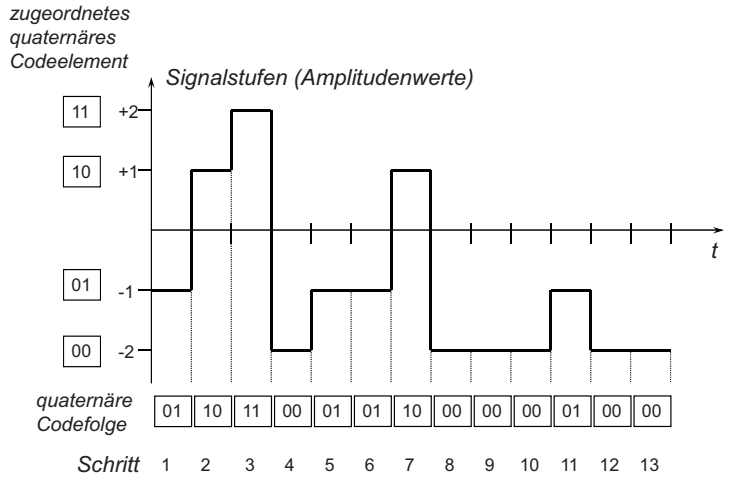


⇨ Schrittgeschwindigkeit 5 baud

Zwei- und mehrwertige Digitalsignale

- **Zweiwertiges Digitalsignal (Binärsignal):**
  - Digitales Signal mit nur zwei Werten des Signalparameters (Digitales Signal, bei dem die Signalelemente binär sind)
- **Mehrwertiges (mehrstufiges) Digitalsignal:**
  - Die (diskrete) Signalkoordinate kann mehr als zwei Werte annehmen; Beispiel: DIBIT = zwei Bit pro Koordinatenwert (quaternäres Signalelement)
  - Die Anzahl n der diskreten Werte (Kennwerte, Stufen), die ein Signalelement annehmen kann, wird wie folgt gekennzeichnet:
    n = 2 binär (binary)
    n = 3 ternär (ternary)
    n = 4 quaternär (quarternary)
    ...
    n = 8 oktonär (octonary)
    n = 10 denär (denary)

Mehrwertiges Digitalsignal - Beispiel



Schritt- versus Übertragungsgeschwindigkeit

- **Schrittgeschwindigkeit** v_S (**symbol rate**, modulation rate, digit rate)
  - Gibt - anschaulich - die Zahl der ggf. nur potenziellen Signalparameter-Zustandswechsel an (Schrittumschläge).
  - Für isochrone Digitalsignale gilt: $\mathbf{v_S = 1/T}$ (T: Schrittdauer)
  - **Einheit: 1/s = baud** (Abk. bd)

- **Übertragungsgeschwindigkeit** Φ (**Einheit: bit/s**)
  - Für zweiwertige Signale (binäre Signale):
    Jeder Schrittumschlag codiert ein Bit. Deshalb gilt in diesem Fall:
    $\mathbf{v_S}$ *(in baud)* $\mathbf{= \Phi}$ *(in bit/s)*
    Die Übertragungsgeschwindigkeit wird in diesem Fall als *Bitrate (bit rate)* bezeichnet.
  - Für mehrstufige Signale (mit n möglichen Wertestufen):
    Übertragungsgeschwindigkeit Φ (in bit/s): $\mathbf{\Phi = v_S * ld(n)}$
    Bei DIBIT-Codierung: 1 baud = 2 bit/s (quaternäres Signal)
    Bei TRIBIT-Codierung: 1 baud = 3 bit/s (oktonäres Signal)

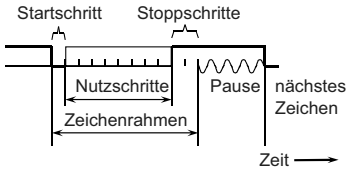
Synchronisation bzgl. Abtastzeitpunkte

- **Abtastzeitpunkte:**
  - Zeitpunkte, an denen die Senke den Signalverlauf y'(t) für das aktuelle Zeitintervall zur Ermittlung des Signalwertes abtastet.

- **Verfahren zur Erzielung von Gleichlauf (Synchronisation):**
  - Sende- und Empfangstakt unterliegen gemeinsamen Konventionen und werden diesen folgend von Quelle und Senke unabhängig voneinander bestimmt.
    ➔ *äußerst stabile Taktgeneratoren erforderlich*
  - Übertragung des Taktrasters auf eigenem parallelen Kanal.
    ➔ *beschränkt auf Nahbereich*
  - Übertragung des Taktrasters mit dem Signal.
    ➔ *Ableitung des Taktrasters aus dem Signalverlauf*
  - Punktuelle Synchronisation eines weitgehend unabhängigen Taktgenerators bei der Senke durch das Signal.
    ➔ *nur beschränkte Frequenzkonstanz erforderlich, Synchronisation bei Schrittgruppen oder Blöcken*

Synchronisation durch Taktrasterübertragung

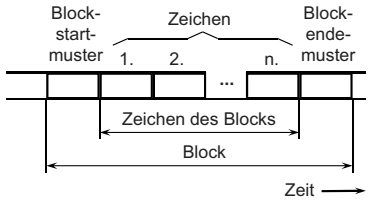
Zeichenweiser Start/Stopp-Betrieb (Asynchronbetrieb)



Voraussetzung:
- Ruhepegel
- feste Zahl von Nutzschritten

Nachteil:
- 3-aus-11 Overhead (8 Nutzbits bei 11 zu übertragenden Bits)

Blocksynchronisation (Synchronbetrieb)

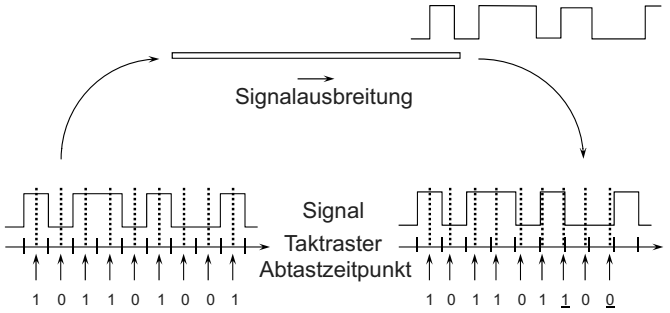


Voraussetzung:
- Blockstart-/-endemuster eindeutig

Maßnahme:
- Modifikation/Rückgängigmachen entsprechender Muster im Block (Bitstopfen)

Bitfehler durch fehlerhafte Synchronisation

- Beispiel:



Übertragungsstörung durch Rauschen

- Neben der systematischen Beeinflussung des Signals durch
  - Dämpfung
  - Laufzeitverzerrungen

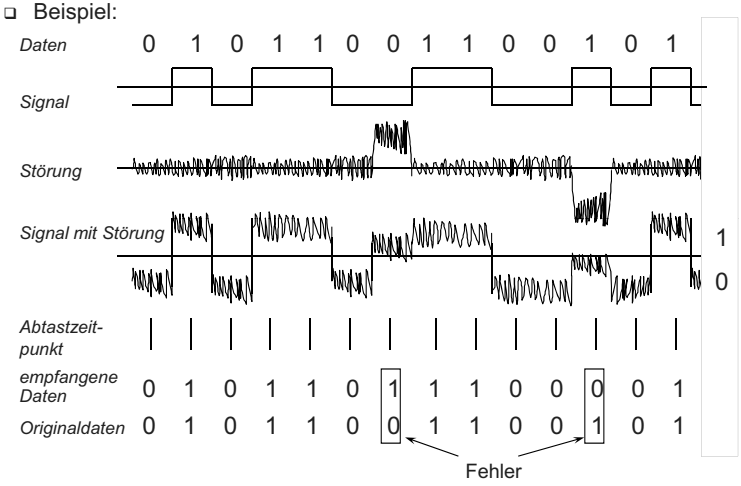
  können Signalstörungen durch
  - transiente, stochastische Prozesse
  - weißes Rauschen
  - Impulsstörungen

  auftreten.

- Lange anhaltende Störungen: Bündelfehler (Echobildung, Nebensprechen, (thermisches)Rauschen, Anschalten von induktiven Lasten(Motor), 50Hz Netzbrummen stets auf einer Leitung, ...)

Bitfehler durch transiente Störungen

- Beispiel:



Nyquist-Kriterium und Shannon-Kanalkapazität

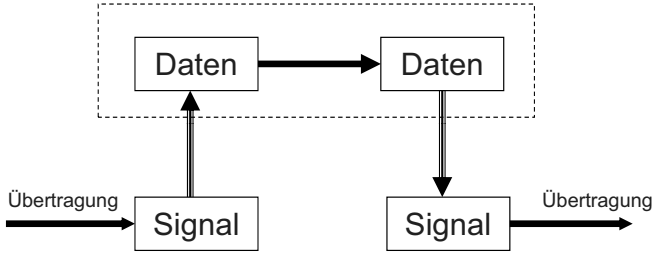
- 1924, H. **Nyquist**:
  Maximale Schrittgeschwindigkeit für einen **Kanal mit eingeschränkter Bandbreite**:
  $v_s = 2\ B$
  mit B = Bandbreite des Kanals
- Daraus ergibt sich eine maximale Datenrate für einen **rauschfreien** Kanal:
  max. Datenrate $= v_s\ ld(n)$
  $= 2\ B\ ld(n)$ **[bit/s]**
  mit n = Anzahl diskreter Signalstufen
- Bsp.: Kanal mit 3.000 Hz Bandbreite, binäres Signal
  → max. Datenrate: 6.000 bit/s

- 1948, C. **Shannon**:
  (auch bekannt als *Shannon-Hartley-Gesetz*) Kanalkapazität = informationstheoretische obere Grenze für die Information (in Bit), die in einem Schritt **fehlerfrei** über einen Kanal mit **weißem Rauschen** übertragen werden kann
- Daraus ergibt sich eine maximale Datenrate, die mit einer hypothetischen optimalen Kanalkodierung erreichbar ist:
  max. Datenrate = B ld(1+S/N) **[bit/s]**
  mit S/N = Signal-Rauschverhältnis
- Bsp.: Kanal mit 3.000 Hz Bandbreite, S/N = 1000 = 30dB [1)]
  → max. Datenrate: 30.000 bit/s
  Durch Verwendung von fehlererkennenden bzw. –korrigierenden Codes (Redundanz!) wird aber mit höherer Rate gesendet!

[1)] Signal-Rauschverh. in dB = $10 \log_{10}(S/N)$ [dB]

Achtung: Da für einen Kanal stets beide Sätze gelten, ergibt sich die fehlerfrei erreichbare maximale Datenrate aus dem *Minimum* der beiden Ergebnisse!

Digitale Regeneration über abstrakte Datenrepräsentation

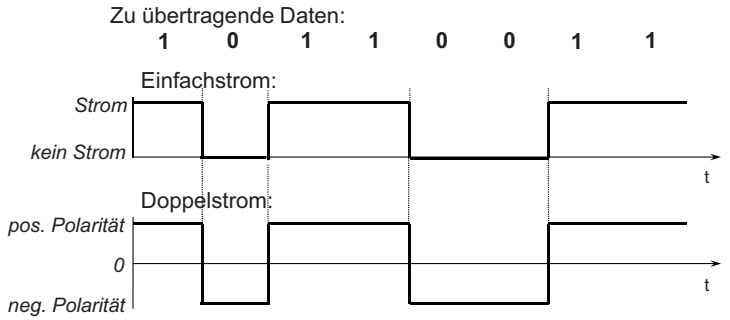


➔ prinzipiell unendlich wiederholbare Regeneration digitaler Daten

11.4.2. Basisbandübertragungsverfahren

- **Basisband:**
  - (üblicher Wortgebrauch in der Nachrichtentechnik)
    Bandbereich eines primären Signals in der „ursprünglichen“ Frequenzlage
  - Hier insbesondere: Frequenzband, das auch sehr niedrige Frequenzen bis zum Gleichstrom beinhaltet
  - Übertragung digitaler Signale mit „rechteckförmigem“ Signalverlauf erfordert die Übertragung sehr niedriger Frequenzen!
    (und theoretisch unendlich hoher Frequenzen nach Fourier, daher kann Rechteckform nie erreicht werden!)
  - Bei Gleichstromanteil (z.B. Einfachstromsignale) Übertragung ab Frequenz 0.
  - Älteste und einfachste Verfahren aus der Telegrafentechnik (z.B. Morsetelegrafie)

Einfachstrom/Doppelstrom-Verfahren



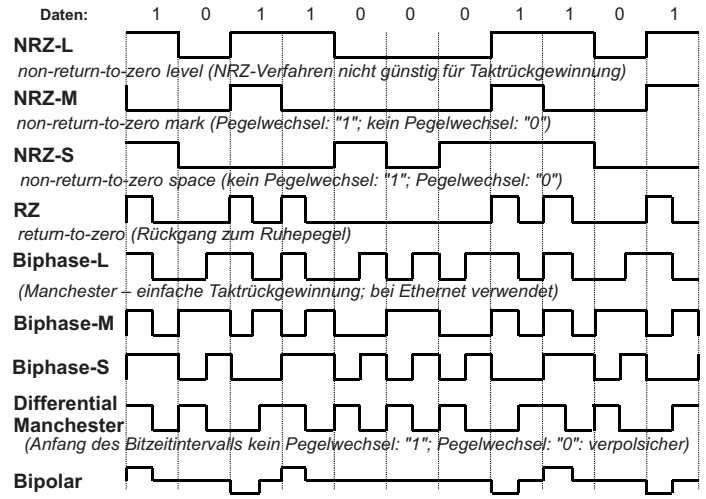
Binärzeichen	1	0
Einfachstrom	Strom	kein Strom
Doppelstrom	positive Polarität	negative Polarität

Moderne Basisbandverfahren

- Moderne digitale Übertragungstechnik verwendet Basisbandverfahren bis zu sehr hohen Bitraten im Multi-Mega-bit/s-Bereich (PCM-Technik, lokale Netze (LAN), ISDN usw.).
- Dabei erwünscht bzw. erforderlich:
  - kein Gleichstromanteil
  - Wiedergewinnung des Takts aus ankommender Signalfolge (selbsttaktende Signalcodes)
  - Erkennung von Signalfehlern auf Signalebene
  - Niedrige Fehleranfälligkeit bei der Decodierung

- **Leitungscodes, Übertragungscodes:**
  - Die Zuordnungsvorschrift
    digitales Datenelement ➔ digitales Signalelement
    wird als Signal- bzw. Leitungscodierung bezeichnet.
  - Die sich ergebenden zeit- und wertdiskreten Signalverläufe heißen: Leitungscodes bzw. Übertragungscodes

Moderne Basisbandverfahren - Beispiele



4b/5b - Kodierung

Symb.	Code Gruppe	Binärdarstellung
0	11110	0000
1	01001	0001
2	10100	0010
3	10101	0011
4	01010	0100
5	01011	0101
6	01110	0110
7	01111	0111
8	10010	1000
9	10011	1001
A	10110	1010
B	10111	1011
C	11010	1100
D	11011	1101
E	11100	1110
F	11101	1111

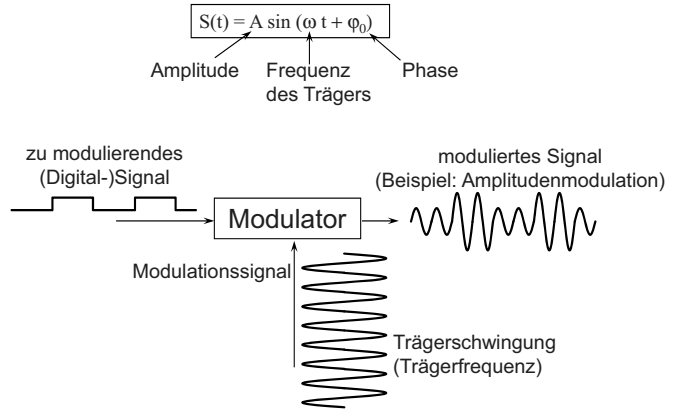
Symb.	Code Gruppe	Bedeutung
Q	00000	Quiet
I	11111	Idle
H	00100	Halt (Forced Break)
J	11000	1st of Start Delimiter(SD) Pair
K	10001	2nd of SD-Pair
T	01101	End Delimiter (ED)
R	00111	Logical ZERO (reset)
S	11001	Logical ONE (set)

- 4 Datenbits → 5 Signalbits auf der Leitung
- keine Symbole mit mehr als 3 Nullen in Folge (mindestens alle vier Bits erfolgt eine Transition): Vermeidet zu langes Verweilen auf einem Signalpegel → stellt Taktrückgewinnung sicher
- Anwendung: FDDI (Fiber Distributed Data Interface) mit NRZ-M-Verfahren
- Code-Effizienz: 80% (vgl.: Differential Manchester hat Code-Effizienz von 50%)
- 16 Symbole zur Nutzdatenübertragung; weitere für Steuerzwecke

Übertragungsverfahren mit Träger - Begriffe

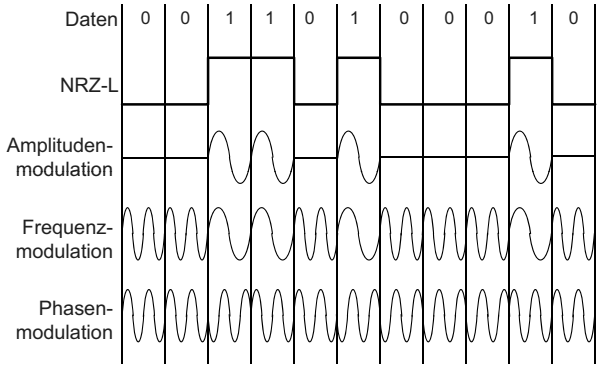
- Trägerfrequenz-Übertragungsverfahren:
  - Modulation digitaler Daten auf analoge Signalträger
- Modulation allgemein:
  - Übertragung eines Signals in seiner „ursprünglichen“ Signalform und Frequenzlage aus technischen und wirtschaftlichen Gründen oft nicht sinnvoll.
  - Als Modulation allgemein wird Verschiebung der Frequenzlage, Anpassung an Übertragungscharakteristik des Übertragungsmediums (auch Übertragungskanal) usw. bezeichnet.
- Modulation (engere Bedeutung):
  - Modulation ist die planmäßige Beeinflussung eines Trägersignals durch das modulierende Signal (Modulationssignal)

Prinzip der Schwingungsmodulation



Schwingungsmodulation: analoger Signalträger ist Sinusschwingung

Arten der Schwingungsmodulation



Klassisch: Äquivalenzliste nach ITU V.1

		Binärzeichen 0	Binärzeichen 1
Gleichstrom-betrieb	Doppelstrom	negativ	positiv
	Einfachstrom	kein Strom	Strom
Wechselstrom-betrieb	Amplituden-modulation	kein Ton	Ton
	Frequenz-modulation	hohe Frequenz	tiefe Frequenz
	Phasendifferenz-modulation	keine Phasendrehung	Phasendrehung um 180°
	Phasenmodulation mit Bezugsphase	Gegenphase	Bezugsphase

11.4.3. Mehrfachnutzung von Übertragungswegen

- **Übertragungsweg:**
  - physikalisch-technisches Transportsystem für Signale
- **Übertragungskanal:**
  - Abstraktion eines Übertragungsweges für einen Signalstrom
  - Auf einem Übertragungsweg können oft mehrere (viele) Übertragungskanäle parallel unterhalten werden, so ist beispielsweise eine Aufspaltung der totalen Übertragungskapazität eines Übertragungsweges auf verschiedene Sender-Empfänger-Paare möglich.
  - Die Zusammenfassung von Übertragungskanälen auf einem Übertragungsweg heißt

Bündelung oder Multiplex, auch Verschachtelung

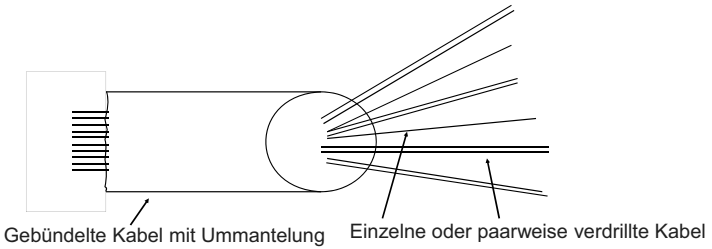
  - Nutzung des Übertragungskanals in beide Richtungen: Richtungsmultiplex oder Duplex

Raummultiplex

- Bündelung vieler Einzelübertragungswege heißt:

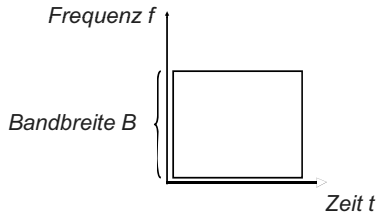
Raummultiplex

- Beispiele: mehrspurige Autobahn, Leitungsnetz zwischen Fernvermittlungsstellen der Telekom



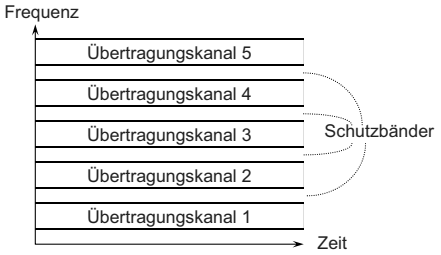
Übertragungskapazität eines Nachrichtenübertragungssystems

- Zeitgesetz der Nachrichtenübertragungstechnik:
  - integrale **Übertragungskapazität** eines Systems = Produkt der Bandbreite (Frequenzbereich) und der zur Verfügung stehenden Zeit
    (Achtung: idealer Fall ohne Störungen bei binärem Signal)



Frequenzmultiplex

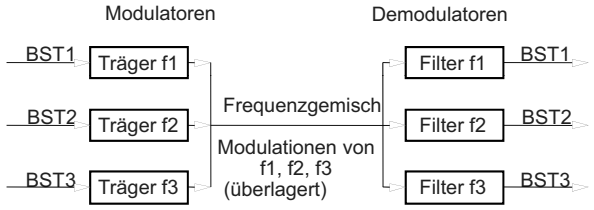
- Breitbandige Übertragungswege ermöglichen die Unterbringung vieler Übertragungskanäle in unterschiedlichen Frequenzbereichen (Frequenzbänder), d.h. man teilt die verfügbare Bandbreite in eine Reihe von - nicht notwendig gleichbreite - Frequenzbänder und ordnet jedem Frequenzband einen Übertragungskanal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



Nutzung der Übertragungskapazität eines Übertragungsweges im Frequenzmultiplex

Eignung des Frequenzmultiplex

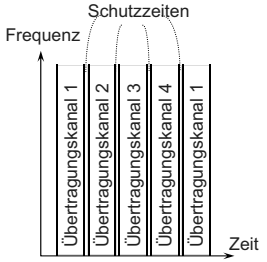
Das *Frequenzmultiplexverfahren* (FDM= **F**requency **D**ivision **M**ultiplexing) ist für analoge Daten und schwingungsmodulierte digitale Daten geeignet.
Anwendung z.B. Funk-/Satellitentechnik



Schema der technischen Realisierung eines Frequenz-Multiplexsystems
BSTx = Bitstrom x

Starres Zeitmultiplex

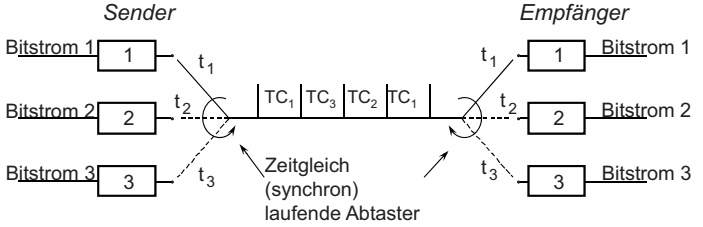
- Die gesamte Übertragungskapazität (die ganze verfügbare Bandbreite) wird kurzzeitig (Zeitschlitz, Zeitscheibe) einer Sender-Empfänger-Kombination zur Verfügung gestellt.
- Nach einer Schutzzeit wird dann die Kapazität des Übertragungsweges dem nächsten Kanal zugeteilt.
- Diese zeitlich gestaffelte Übertragung mehrerer Signalströme wird als Zeitmultiplex (TDM = **T**ime **D**ivision **M**ultiplexing) bezeichnet.



Nutzung der Übertragungskapazität im Zeitmultiplex

Eignung des starren Zeitmultiplex

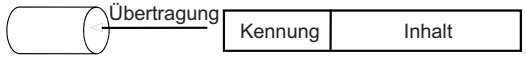
- Zeitmultiplex nur für zeitdiskrete Signale einsetzbar (bevorzugt zeit- und wertdiskrete Signale = Digitalsignale)



- Festes Zeitmultiplex mit starrer Zeitscheibenzuteilung.
- Übertragungseinheit z. B. ein Bit, ein Byte (Oktett).
- Jedem Sender wird periodisch eine Zeitscheibe (time slot, time slice) TC_1, TC_2....TC_n zugeteilt. Sender, Abtaster und Detektionsmechanismus beim Empfänger laufen im gleichen Takt: synchrone Zeittakt-Stabilität wichtig!

Anforderungsgesteuertes Zeitmultiplex

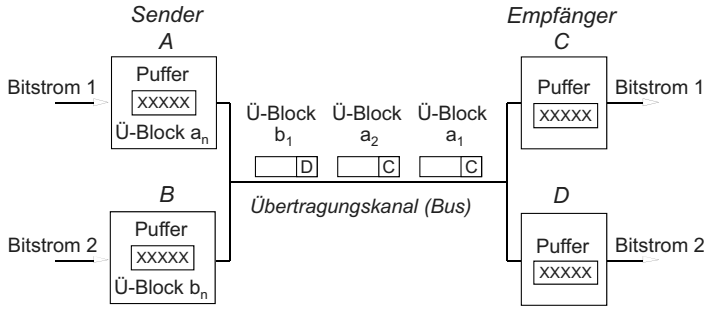
- Zeitscheiben werden nicht fest, sondern bei Bedarf dem Sender zugeteilt.
- Empfänger kann nicht mehr aus der Zeitlage der Zeitscheiben die Herkunft (Zuordnung zu unterschiedlichen Sendern) identifizieren!
- Somit wird eine Kennung erforderlich (Adresse, Kennzahl, usw.).



Schematischer Aufbau eines Übertragungsblocks mit Kennung

- Das anforderungsgesteuerte Zeitmultiplex (demand multiplexing) wird auch als statistisches Zeitmultiplex (STDM = **S**tatistical **T**ime **D**ivision **M**ultiplexing) bezeichnet.

Schema des anforderungsgesteuerten Zeitmultiplex

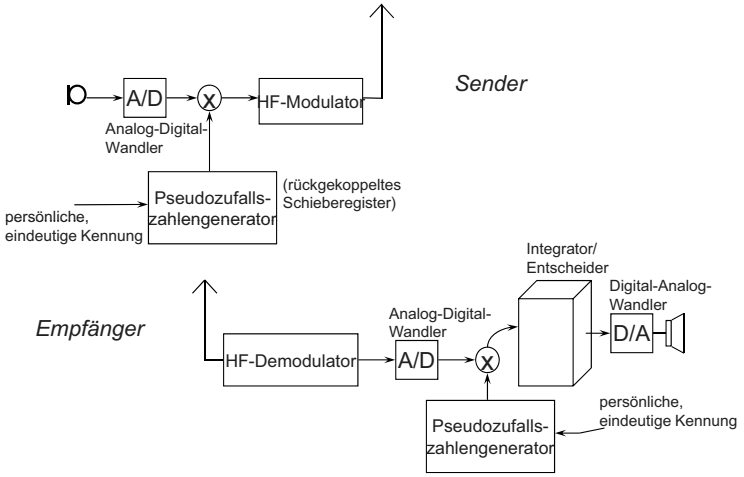


- Schema der technischen Realisierung des statistischen Blockmultiplex
- Sehr unterschiedliche Zuteilungsstrategien für den gemeinsam genutzten Übertragungsweg

CDMA - Code Division Multiple Access - Prinzip

- **Prinzip:**
  - alle Sender nutzen das gleiche Frequenzband und senden gleichzeitig
  - Signal wird auf der Senderseite mit einer für den Sender eindeutigen Pseudozufallszahl verknüpft (XOR)
  - Empfänger kann mittels bekannter Sender-Pseudozufallsfolge und einer Korrelationsfunktion das Originalsignal restaurieren
- **Nachteil:**
  - höhere Komplexität der Implementierung wg. Signalregenerierung
- **Vorteile:**
  - alle können auf der gleichen Frequenz senden
  - keine Frequenz-/Zeitscheibenplanung nötig
  - sehr großer Coderaum (z.B. 2^{32}) im Vergleich zum Frequenzraum

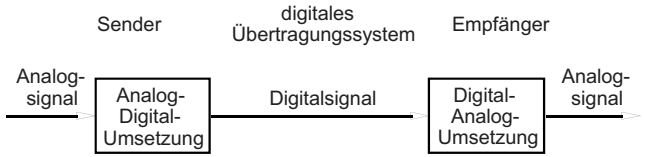
CDMA - Code Division Multiple Access



11.4.4. Digitale Übertragung analoger Daten

- Übertragung analoger Daten (dargestellt durch analoge Signale) über digitale Übertragungssysteme erfordert:

  Digitalisierung der analogen Daten/Signale



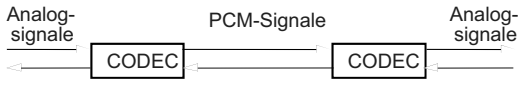
- A/D- und D/A-Umsetzungen zur Übertragung analoger Signale auf digitalen Übertragungssystemen

analog		**digital**	
wertkontinuierlich	➔	wertdiskret	= Quantisierung
zeitkontinuierlich	➔	zeitdiskret	= Abtastung

11.5. Pulse-Code-Modulations-Technik

- Die Zusammenfassung der Schritte
  Abtastung - Quantisierung – Codierung
  und die Darstellung der gewonnenen Codewörter als digitale Basisbandsignale am Ausgang des PCM-A/D-Umsetzers und Codierers ist Grundlage der im großen Umfang eingesetzten digitalen
  PCM-Technik.

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



Umsetzung von Analogsignalen in PCM-Signale und Rückkonvertierung durch CODECs

Abtastung

- Für die *Zeitdiskretisierung* muss eine Abtastung der Analogverläufe erfolgen. Praktisch wichtig ist die periodische Abtastung. Der zum Abtastzeitpunkt vorliegende Momentan-Wert des Analogsignals wird der Analog-Digital-Umsetzung unterworfen.

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

Abtasttheorem

Abtasttheorem von Shannon und Raabe (1939):

- Zur fehlerfreien Rekonstruktion des Signalverlaufs der abgetasteten Analogsignale ist eine Mindestabtasthäufigkeit (Abtastfrequenz f_A) erforderlich (bei periodischem Abtastzyklus).

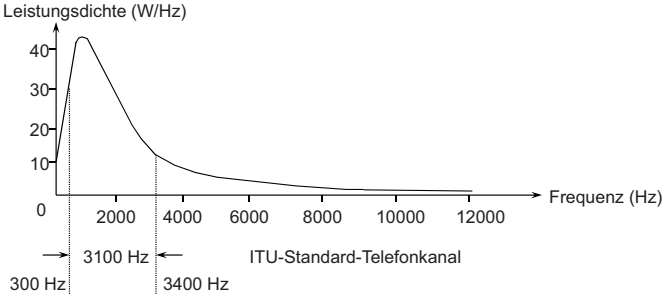
- **Abtasttheorem:** Eine Signalfunktion, die nur Frequenzen im Frequenzband B (bandbegrenztes Signal) enthält, wobei B gleichzeitig die höchste Signalfrequenz ist, wird durch ihre diskreten Amplitudenwerte im Zeitabstand $\mathbf{t_0 = 1/(2B)}$ vollständig bestimmt.

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

Wiederholung: Frequenzspektrum eines Signals

- Bandbegrenztes Signal: Signale können ein „natürlich“ begrenztes - meist kontinuierliches- Frequenzspektrum umfassen oder durch technische Mittel auf einen Ausschnitt ihres Spektrums begrenzt werden (Bandbreite).

Kontinuierliches - akustisches - Frequenzspektrum der menschlichen Stimme und Bandbreite des analogen ITU-Standardtelefonkanals



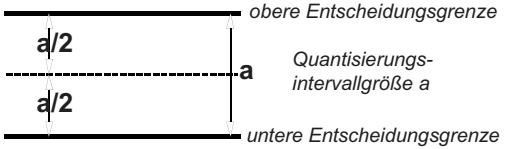
PCM-Fernsprechkanal - Abtastung

- *Ausgangspunkt*
  - Analoger ITU-Fernsprechkanal, Frequenzlage 300-3400 Hz, Bandbreite 3100 Hz, höchste vorkommende Frequenz 3400Hz
- *Abtastfrequenz*
  - ITU-empfohlene Abtastfrequenz für PCM-Fernsprech-Digitalisierung

    f_A = 8 kHz
- *Abtastperiode*
  - $T_A = 1/f_A = 1/8000\text{Hz} = 125\ \mu\text{s}$
  - Die ITU gewählte Abtastfrequenz ist höher als nach Shannon-Abtasttheorem erforderlich (3400 Hz obere Bandgrenze ergibt 6800 Hz Abtastfrequenz).
  - Für die höhere Abtastfrequenz sprechen technische Gründe (Filtereinfluss, Kanaltrennung usw.).

Quantisierung

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



- Quantisierungsintervall für die Zuordnung eines diskreten Wertes zu allen z.B. zwischen + a/2 und - a/2 liegenden Werten einer Analogdarstellung (andere Zuordnungen denkbar)
- Rückwandlung: Beim Empfänger wird ein Analogwert rückgewandelt (Digital-Analog Umsetzung), der dem in der Mitte des Quantisierungsintervalls liegenden Analogwert entspricht (maximaler Quantisierungsfehler = a/2)

Codierung

- Die Quantisierungsintervalle werden durch die Zuordnung eines - im Prinzip frei wählbaren - (Binär-) Codes gekennzeichnet und unterschieden.
- **Grundprinzip:** Anstelle des ursprünglichen Analogsignals wird die - mit dem Quantisierungsfehler behaftete - digitale Darstellung übertragen.
- Beim PCM (siehe weiter hinten) wird ein reiner Binärcode (Darstellung als Binärzahl) als Codierung des Digitalwertes gewählt.

PCM-Fernsprechkanal - Quantisierung

- **Amplitudenquantisierung:**
  - Zahl der benötigten Quantisierungsintervalle wird bei der akustischen Sprachkommunikation (Fernsprechen) durch den Grad der Silbenverständlichkeit beim Empfänger bestimmt.
  - Mit „Sicherheitszuschlag“ wurden 256 Quantisierungsintervalle genormt.
  - Bei binärer Codierung reichen dafür 8 Bit Codewortlänge aus:
    - $2^8 = 256$
  - Die Übertragungsgeschwindigkeit (Bitrate) für einen digitalisierten Fernsprechkanal ergibt sich somit wie folgt

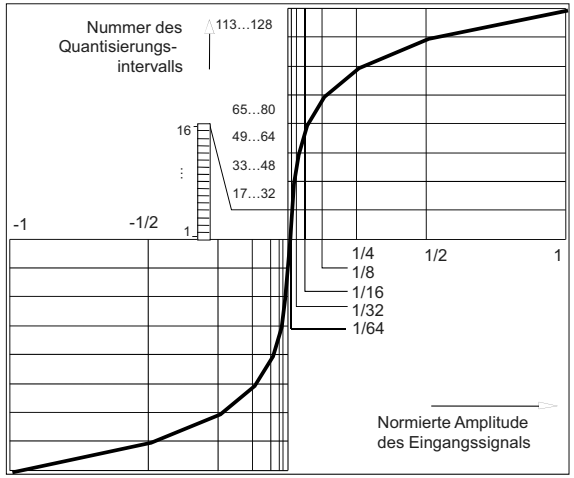
Bitrate = Abtastfrequenz x Codewortlänge
kbit/s = 8000/s x 8 bit
64kbit/s

k(kilo) = 1000 ! (ebenso M(Mega): 1 Mbit/s = 1000000 bit/s)

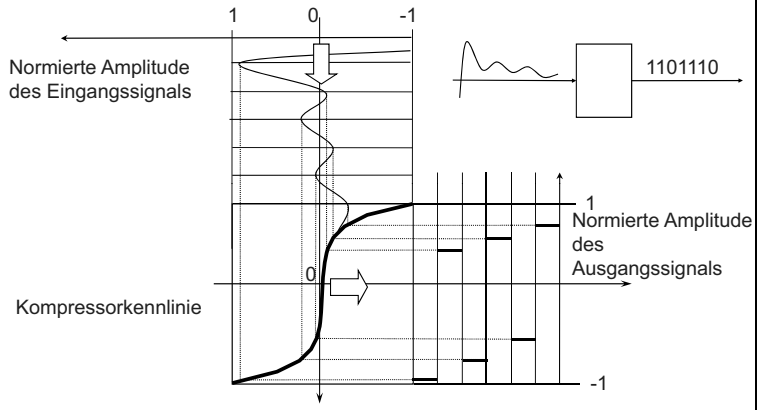
Ungleichförmige Quantisierung

- Bei gleichförmiger Quantisierung sind alle Intervalle gleich groß und von der Größe des Momentanwerts 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 als im gleichförmigen Fall.
  - *Kompressor*: Die ungleichförmige Intervallgröße wird durch einen dem Quantisierer vorgeschalteten (Signal-) Kompressor erzielt.
  - *Expander:* Auf der Empfangsseite wird in inverser Funktion ein Expander eingesetzt. Wiederherstellung der ursprünglichen Größenverteilung der Signale (Dynamik der Signale).
  - *Kompander:* Kombination von Kompressor und Expander.

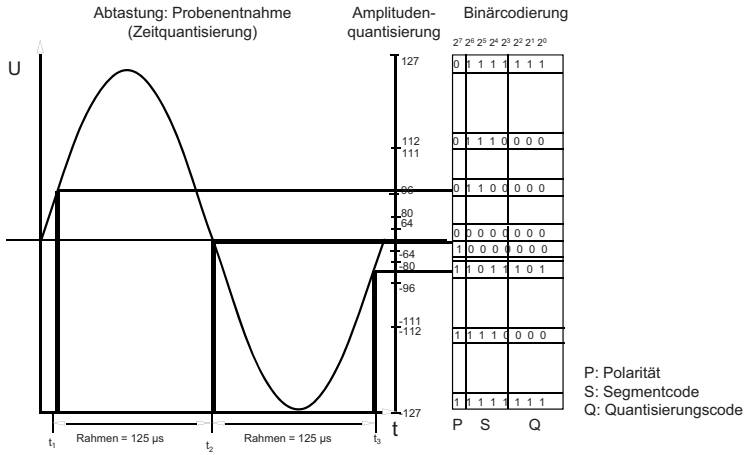
13 Segment-Kompressorkennlinie



Kompressorkennlinie - Prinzip



Zusammenhang bei der PCM-Technik



PCM-Systeme

- Die praktische Gestaltung technischer PCM-Systeme wird durch das Fernsprechen bestimmt (obwohl grundsätzlich jede Art analoger - nach Digitalisierung - und digitaler Daten unter Verwendung digitaler PCM Übertragungssysteme übertragbar ist).
- Praktisch eingesetzte PCM-Systeme bauen im Übertragungsbereich auf der Mehrfachnutzung der Übertragungswege durch Zeitmultiplexverfahren auf.
- Doppelbedeutung von PCM:
  - Spezielles Umsetzverfahren für analoge Signale
  - Starres Zeitmultiplexverfahren für Fernübertragung
- Aus historischen Gründen hat ITU zwei PCM-Übertragungssysteme genormt.
- Behandelt wird das für die Deutsche Telekom AG verbindliche CEPT-System.

PCM-30-System – Deutsche Telekom AG

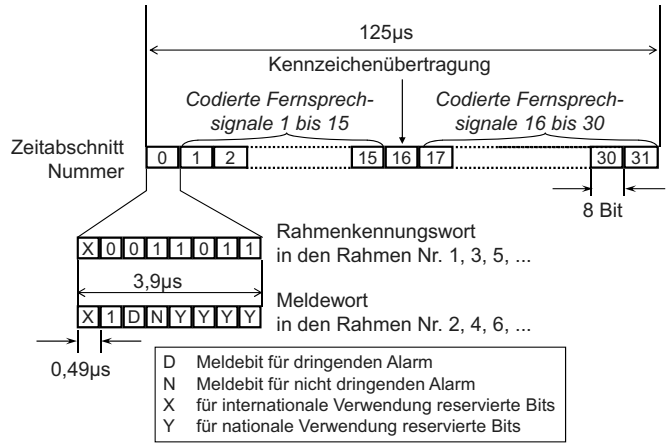
- Für jedes System sind Systemparameter festzulegen, z.B.:
  - kleinste Übertragungseinheit pro Zeitabschnitt (Bit, Byte, n-bit-Wort, Block)
  - Häufigkeit der Zeitscheibenzuteilung an einen Übertragungskanal
  - Synchronisierhilfen
  - Melde- und Signalisierdaten
- Struktur des genormten PCM-30 Kanalgrundsystems der Deutschen Telekom AG:
  - pro Zeitscheibe: 8 bit
  - Übertragungszeit pro Kanal: ca. 3,9µs
  - Verschachtelungsgrad (die Periode): 32 Kanäle
- Als Übertragungseinheit der Multiplexstruktur ist die Struktur mit 32 verschachtelten Kanälen aufzufassen, sie wird **Pulsrahmen (pulse frame, frame)** genannt.

PCM-Multiplexsysteme - Rahmenstruktur

- Die 32 Zeitabschnitte sind mit 0 bis 31 nummeriert. Ein Abschnitt ist ca. 3,9 ms lang. Die gesamte Rahmendauer ist bei PCM30 mit 125 Mikrosekunden genormt.
- Im Zeitabschnitt 0 werden abwechselnd Rahmenkennworte (u.a. zur Rahmenidentifizierung, Synchronisierung) und Meldeworte (u.a. zur Überwachung der Digitalsignalleitung) übertragen.
- Der Kennzeichenabschnitt 16 dient zur Übertragung vermittlungstechnischer Daten, wie Wählzeichen usw.
- Die 30 übrigen Zeitabschnitte nehmen jeweils 8 bit (einen Abtastwert) eines 64kbit/s digitalen Fernsprechsignals auf; daher der Name PCM30.
- Feste Zuordnung des Platzes im Rahmen für eine 64kbit/s Fernsprechverbindung. Reservierung beim Verbindungsaufbau (“Wählverbindung”).

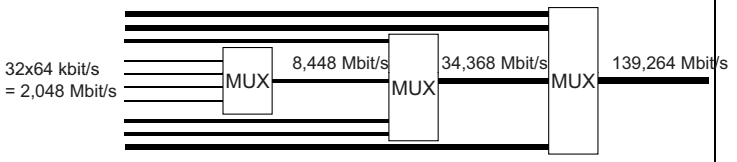
- Hinweis: Anstelle von Fernsprechsignalen können beliebige andere digitalisierte analoge und digitale Daten in Einheiten von 8 bit über ein digitales PCM-System übertragen werden!

Pulsrahmen des Systems PCM 30



Über PCM 30 hinausführende Systeme

- Zeitmultiplex wie in PCM 30 kann auch für mehr Kanäle genutzt werden (z.B. PCM 120)
- Plesiochrone Digitale Hierarchie (PDH)
  - hierarchisches Zeitmultiplex
  - Schwankungen der Rate werden durch Stopfbits kompensiert



- Synchrone Digitale Hierarchie (SDH)
  - synchrone 125µs Rahmen
  - Grundrate von 155,52 Mbit/s, Vielfache hiervon werden unterstützt
  - Datenblocke können über Rahmengrenzen gehen
  - Pointer im Rahmenkopf zeigen auf den Anfang des nächsten Datenblocks

11.6. Zusammenfassung der Signalkonversionen

- *Analog* ➔ *Analog:*
  - ursprüngliches Telefon (englisch: POT = Plain Old Telephone)
  - Analoger Rundfunk
- *Analog* ➔ *Digital:*
  - PCM-Konversion
  - Digitale Telefonie
- *Digital* ➔ *Analog:*
  - Digitaldatenübertragung über analoges Fernsprechnetz (MODEM-Technik)
  - Übertragung digitaler Daten mittels Funk-/Satellitentechnik
- *Digital* ➔ *Digital:*
  - Leitungscodierung im Basisbandverfahren

- Mehrere Signalkonversionen können hintereinander ausgeführt werden
  (wobei *Analog* ➔ *Analog* und *Analog* ➔ *Digital* nie verlustfrei sind).

Lehrstuhl für Netzarchitekturen und Netzdienste
Institut für Informatik – Technische Universität München
Prof. Dr.-Ing. Georg Carle

Grundlagen: Rechnernetze und Verteilte Systeme

Kapitel 12: Bitübertragungsschicht

Schnittstellen, Modem, DSL

Prof. Dr.-Ing. Georg Carle
Lehrstuhl für Netzarchitekturen und Netzdienste
Technische Universität München
carle@net.in.tum.de
http://www.net.in.tum.de

Übersicht

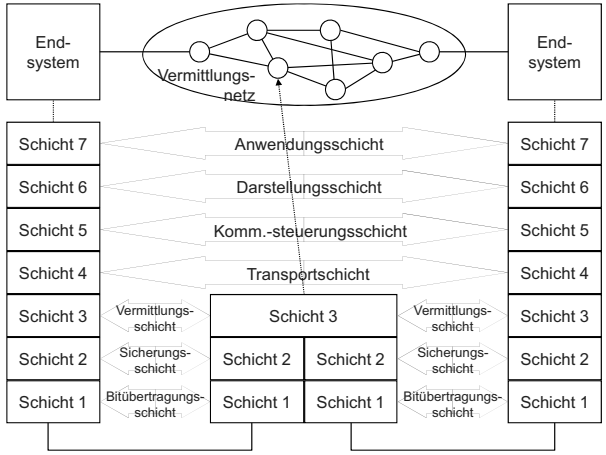
1. Einführung und Motivation
   - Bedeutung, Beispiele
2. Begriffswelt und Standards
   - Dienst, Protokoll, Standardisierung
3. Direktverbindungsnetze
   - Fehlererkennung, Protokolle
   - Ethernet
4. Vermittlung
   - Vermittlungsprinzipien
   - Wegwahlverfahren
5. Internet-Protokolle
   - IP, ARP, DHCP, ICMP
   - Routing-Protokolle
6. Transportprotokolle
   - UDP, TCP
7. Verkehrssteuerung
   - Kriterien, Mechanismen
   - Verkehrssteuerung im Internet
8. Anwendungsorientierte Protokolle und Mechanismen
   - Netzmanagement
   - DNS, SMTP, HTTP
9. Verteilte Systeme
   - Middleware
   - RPC, RMI
   - Web Services
10. Netzsicherheit
    - Kryptographische Mechanismen und Dienste
    - Protokolle mit sicheren Diensten: IPSec etc.
    - Firewalls, Intrusion Detection
11. Nachrichtentechnik
    - Daten, Signal, Medien, Physik
12. **Bitübertragungsschicht**
    - **Codierung**
    - **Modems**
13. Zusammenfassung

Ziele

- In diesem Kapitel wollen wir vermitteln
  - Bedeutung ausgewählter Schnittstellen
  - Funktionsweise eines Modems
  - Funktionsweise eines Breitbandkabelnetzes
  - Funktionsweise der Datenübertragung über die Telefonleitung

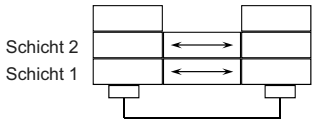
Kapitelgliederung

12.1. Wiederholung - OSI: Die 7 Schichten



12.1.1. Bitübertragungsschicht und Sicherungsschicht

- **Bitübertragungsschicht** (Schicht 1)
  - ungesicherte Verbindung zwischen Systemen
  - Übertragung unstrukturierter Bitfolgen über physikalisches Medium
  - umfasst u.a. physikalischen Anschluss, Umsetzung Daten ⇆ Signale
  - Normung vor allem der physikalischen Schnittstelle Rechner/Medien
- **Sicherungsschicht** (Schicht 2)
  - gesicherter Datentransfer
  - Zerlegung des Bitstroms (Schicht 1) in Rahmen (Frames)
  - Fehlererkennung und -behandlung
  - Protokollmechanismen: Quittierung, Zeit-/Sequenzüberwachung, Wiederholen/Rücksetzen



12.1.2. Bedeutung von Schnittstellen

- Die Übertragung von digitalen Daten über
  - zentral-organisierte Netze (z.B. öffentliche Telefonnetze) und
  - dezentrale Netze (z.B. LAN-Netze)

  erfordert die Standardisierung ihrer Schnittstellen.

- Im Bereich der öffentlichen leitungsgebundenen Netze sind dies:
  - *ITU-T **V**-Empfehlungen*
    Fernsprech- (Telefon-) Netz (analog)
    Älteste Empfehlungsgruppe
    Beispiel für aktuelles Modem: V.90
  - *ITU-T **X**-Empfehlungen*
    Integriertes Daten- und Nachrichtennetz
    Beispiel: X.25-Netz (Datex P)
  - *ITU-T **I**.100 - **I**.600 Empfehlungen*
    Integrated Services Digital Network (ISDN)

Klassische Telefon-Teilnehmerschnittstelle: a/b-Schnittstelle

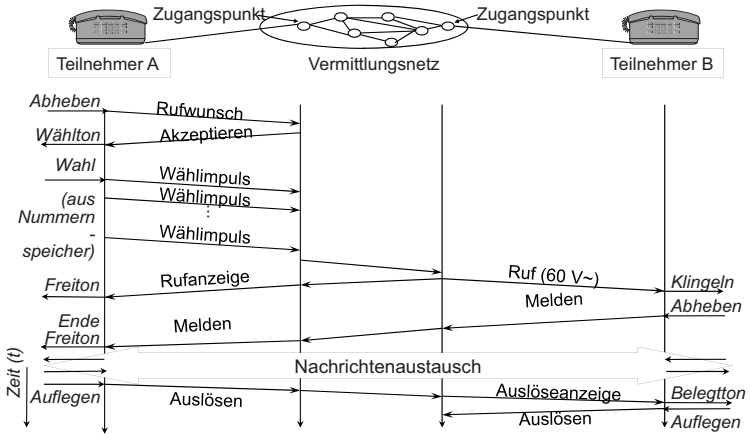
- Die Bezeichnung a/b unterscheidet die beiden Adern (Sprechadernpaar) für die Zweidraht-Teilnehmer-Anschlussleitung (*local subscriber loop*).

- Energieversorgung des Telefons:
  - Beim Abheben des Handapparats („Telefonhörer“) wird ein Gleichstromkreis geschlossen.
  - Fernspeisung durch OVSt (Schleifenstrom i), d.h. Telefonieren ohne lokale Stromversorgung möglich.
    (gilt auch für ISDN: OVSt liefert maximal 400mW bei Stromausfall)
    Wichtige Funktionalität für Katastrophenszenarien
  - Modere (insb. schnurlose) Telefone haben meist eigene Energieversorgung.

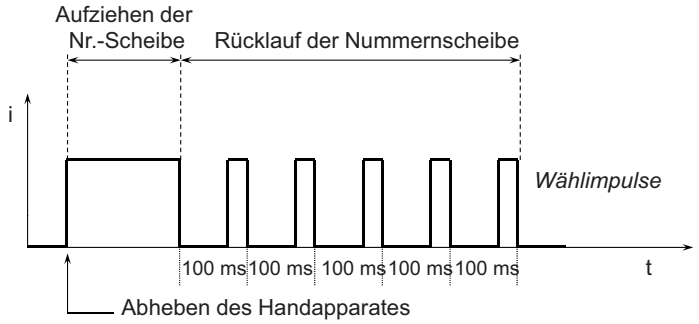
- Signalisierung, Dienstsignale:
  - automatisches Selbstwahlverfahren
  - Netz gibt akustische Signale (Wählton, Freiton, Besetztton usw.) während des Vermittlungsdialogs mit Teilnehmer
  - Signalisierung im gleichen „Band“ wie die spätere (Sprach-)Übertragung (engl.: *in-band*):
    - Impulswahlverfahren (IWV): Rufnummerneingabe ganz klassischer Telefonapparat mit Nummernscheibe (Wählscheibe).
    - Mehrfrequenzverfahren (MFV)

Wiederholung: Beispiel Telefon – Dienst und Protokoll

- Signalisierungsprotokoll im analogen Fernsprechnetz:

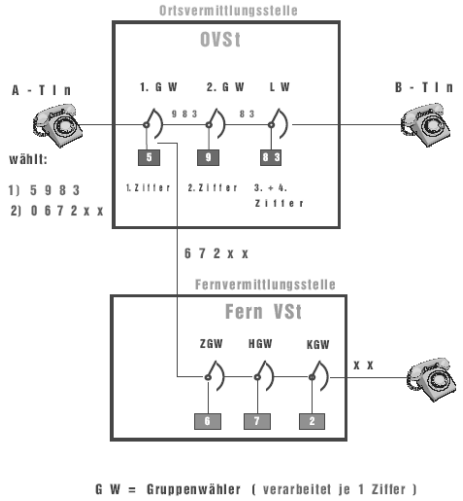


Impulswahlverfahren (IWV)



Wahl der Ziffer 5 über einen mechanisch arbeitenden Apparat mit Nummernscheibe (Impulswahlverfahren)

Direkte Steuerung (historisch)



Mehrfrequenzverfahren (MFV)

- Für Tastentelefone (anstelle des Impulswahlverfahrens):
  - Standardfall mit digitalen Vermittlungen
  - Tastentelefon in 12 Tastenversion (selten 16 Tastenversion)
  - Jede Taste wird signaltechnisch durch ein Frequenzpaar codiert (zwei Frequenzen aus Störsicherheitsgründen).
  - Tastatur des Tastentelefons nicht nur für Wählziffern, sondern auch als Einfachterminal für Datenübertragung

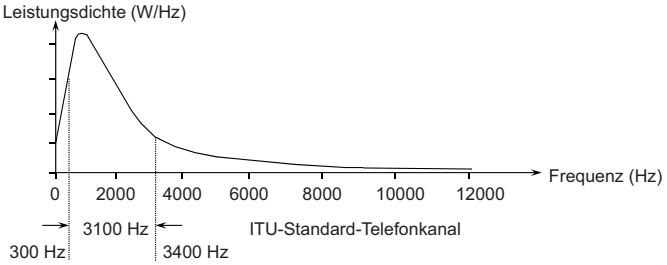
Freq. [Hz]	1209	1336	1447	1633
697	1	2	3	A
770	4	5	6	B
852	7	8	9	C
941	*	0	#	D

Tastenbelegung und Frequenzzuordnung bei Mehrfrequenzcode-Wählverfahren für Tastwahlfernsprecher (16 Tasten)

Wiederholung: Frequenzspektrum eines Signals

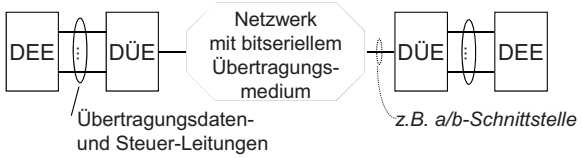
- **Bandbegrenztes Signal:**
  Signale können ein „natürlich“ begrenztes - meist kontinuierliches- Frequenzspektrum umfassen oder durch technische Mittel auf einen Ausschnitt ihres Spektrums begrenzt werden (Bandbreite).

Kontinuierliches - akustisches - Frequenzspektrum der menschlichen Stimme und Bandbreite des analogen ITU-Standardtelefonkanals



Übertragungsschnittstelle digitaler Daten

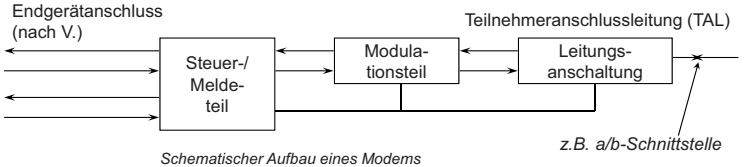
- Zwischen Endgerät (Datenendeinrichtung, DEE) und Netz wird eine Datenübertragungseinrichtung (DÜE) zwischengeschaltet.



  - DEE, engl. DTE: **D**ata **T**erminal **E**quipment
  - DÜE, engl. DCE: **D**ata **C**ircuit Terminating **E**quipment
  - DÜE enthält signaltechnische Funktionen für die Anpassung an die Teilnehmeranschlussleitung (z.B. Modem).
  - Die DEE/DÜE-Schnittstelle ist sehr wichtig für den Anwender, da die Ankopplung sehr unterschiedlicher Endgeräte erwünscht ist.

12.2. Modem

- **Modem** = Modulator/Demodulator = DÜE im Fernsprechnetz
  - Modems basieren ursprünglich auf der klassischen Teilnehmeranschlussleitung des Fernsprechnetzes

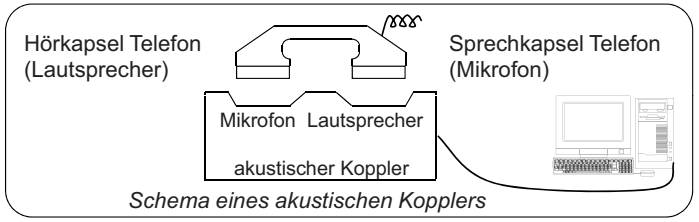


Schematischer Aufbau eines Modems

  - **Leitungsanschaltung:** Signaltechnische (Sende- und Empfangs-) Verstärkung der zu übertragenen Signale
  - **Modulationsteil:** Modulation und Demodulation (Amplitude, Frequenz bzw. Phase)
  - **Steuer-/Meldeteil:** Analyse der vom Netz kommenden Dienstsignale, An/-Abschaltung des Modems an die Leitung, Überwachung des Leitungsbetriebs, Auslösung der V.24-Steuerfunktionen (z.B. Betriebs-, Sendebereitschaft, Ankommender Ruf)

Damals...: Akustische Koppler

- **ITU Empfehlung V.15 –**
  „Urahn“ der heutigen Modems
- Funktion: Werte „0“, „1“ entsprechen hohen bzw. niedrigen Tonfrequenzen



Schema eines akustischen Kopplers

Vorteil:
• mobiler Einsatz
• keine feste Verdrahtung

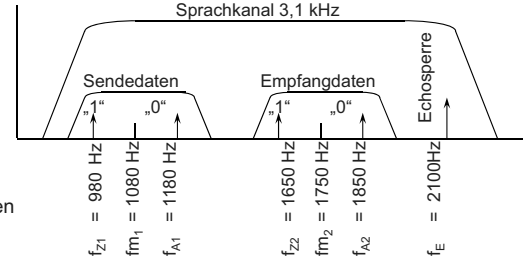
Nachteil:
• nur einige 100 bit/s

Beispiel: V.21 Modem

- Kennzeichen:
  - Weltweit sehr häufig eingesetzter Modem-Standard bei schlechten Leitungen
  - Nutzung des ITU-Standard-Telefonkanals (300 - 3400 Hz)
  - Übertragungsgeschwindigkeit (synchron oder asynchron) bis 300 bit/s
  - Vollduplex-Betrieb durch Parallelbetrieb beider Übertragungsrichtungen in zwei Frequenzlagen: f_{m1} = 1080 Hz, f_{m2} = 1750 Hz, Frequenzhub ± 100 Hz

Kanalbelegung:
Belegung A:
verbindungs-
aufbauendes Modem

(Belegung B:
angerufenes Modem
Sendedaten im oberen
Frequenzbereich)



ITU-V-Empfehlungen: Übersicht

ITU-V-Empfehlungen

Telefonnetz

V.24/RS-232C Schnittstelle

2/4-drähtige Standleitung

V.23	600	od. 1200 Bit/s
V.26	1200	od. 2400 Bit/s
V.27	2400	od. 4800 Bit/s
V.29	4800	od. 9600 Bit/s

(2-Draht halbduplex, 4-Draht duplex)

2-drähtige vermittelte Leitung

V.21	300/1200 Bit/s	duplex
V.22	1200/1200 Bit/s	duplex
V.22 bis	2400/2400 Bit/s	duplex
V.23A	75/1200 Bit/s	duplex
V.29	4800/9600 Bit/s	halbduplex
V.32	4800/9600 Bit/s	duplex
V.32 bis	7200/12000/14400 Bit/s	duplex
V.34	28800 Bit/s	duplex
V.34 bis	33600 Bit/s	duplex
V.90	33600/56000 Bit/s	duplex

Breitbandnetz

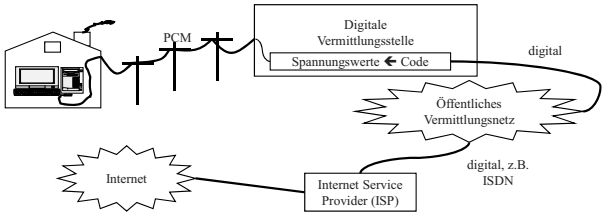
V.35/RS-449 Schnittstelle

Synchron Punkt-zu-Punkt

V.35	48	kBit/s
V.36	48-72	kBit/s
V.37	96-168	kBit/s

V.90-Modem (56 kbps)

- Heute durchgehend digitale Vermittlungsstellen ➔ nur Teilnehmeranschluss-leitung (letzte Meile) analog
- Höhere Datenraten durch *digitale Übertragung vom Provider bis zur Vermittlungsstelle*
- *PCM-Signale auf der Strecke von der Vermittlungsstelle bis zum V.90-Modem* (für ISDN vorhanden: 8-Bit-AD/DA-Wandler mit 8 kHz, 1 Bit als Prüfsumme)
  ➔ theoretisch bis zu 7 bit * 8 kHz = 56 kbps
- Tatsächlich erreichte Bitraten abhängig von Leitungsqualität
  ➔ Ausmessen der Leitung (Line probing)



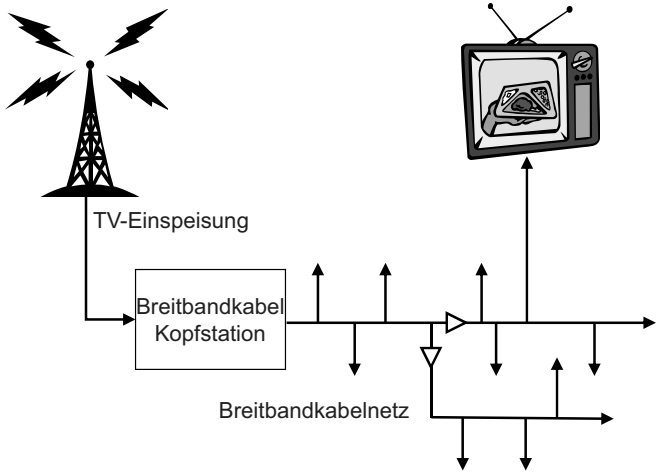
Weitere Modemtechnologien

- **Kabelmodems:**
  Datenübertragung über das Breitbandkabel („Kabelfernsehen“) der Kabelnetzbetreiber,
  - Erweiterung des Frequenzbandes im Kabel auf bis zu 860 MHz
  - Datenraten (je nach Technik) theoretisch bis zu 2 Gbit/s, aber (mit anderen Benutzern) geteiltes Medium!

- **Powerline-Communications (PLC) Modems:**
  Datenübertragung über das Energieverteilnetz („Stromnetz“)
  - Einkopplung hochfrequenter Träger (16-148 kHz sowie 1-30 MHz)
  - Datenraten bis zu 1 Mbit/s, aber ebenfalls geteiltes Medium
  - Anwendbar für öffentliche Datennetze, Datenverteilung im Haus, sowie Telematik-Anwendungen der Energieversorger (z.B. Stromzähler auslesen)

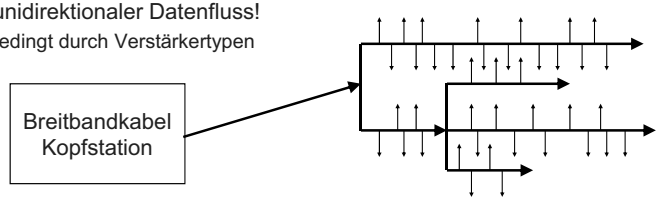
- **DSL-Modems:**
  Höhere Datenraten über herkömmliches Telefonkabel
  - Telefonkabel bleibt gleichzeitig für Telefonie nutzbar
  - Typische Datenraten bei 6-8 Mbit/s

12.3. Breitbandkabelnetze
12.3.1. Konventionelles Netz: Kabelfernsehen

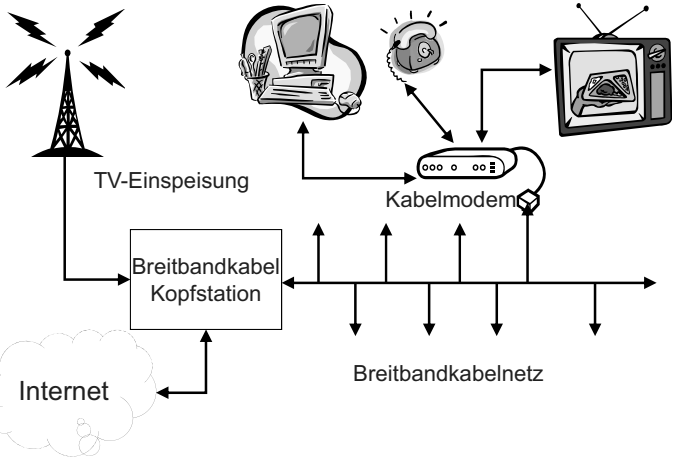


Konventionelles Breitbandkabelnetz: Charakteristika

- Koaxialnetz in Baumstruktur
  - Alle Teilnehmer werde ausgehend von einer Kopfstation (headend) versorgt (Punkt-zu-Mehrpunkt-Verbindung)
- Bandbreiten
  - 606 MHz oder lediglich 450 MHz
- Frequenz-Multiplex
  - Jeder Dienst auf dem Breitbandkabelnetz erhält ein festes Frequenzband
  - Dadurch ist die Zahl der Dienste vorab festgelegt
- Nur unidirektionaler Datenfluss!
  - Bedingt durch Verstärkertypen

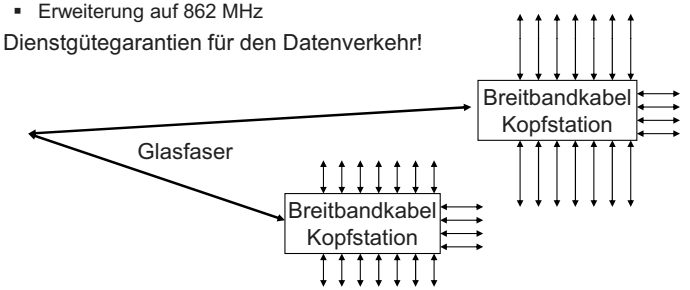


12.3.2. Modernes Breitbandkabelnetz: Digitales TV plus Datendienste



Modernes Breitbandkabelnetz: Charakteristika

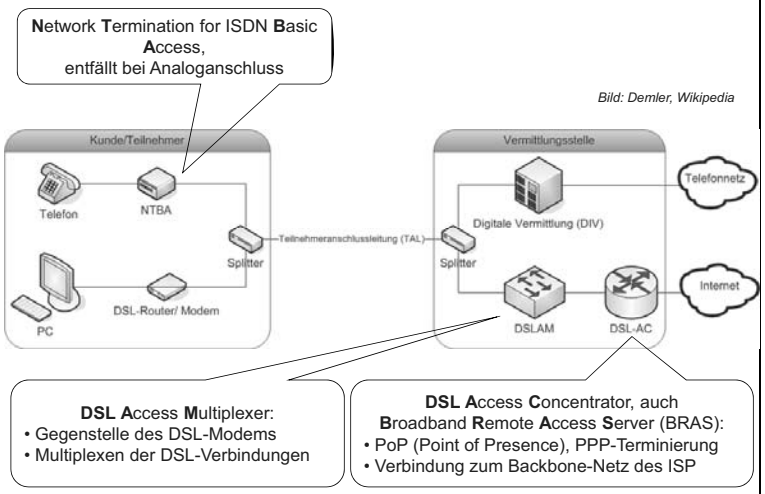
- Verteilnetze in Koaxialtechnik
  - Anschluss von ca. 2000 Haushalten in Sternstruktur
  - Anschluss der Kopfstation mit Glasfasertechnik
- Rückwegefähigkeit
  - Integration von Rückkanalverstärkern
- Bandbreite
  - Erweiterung auf 862 MHz
- Dienstgütegarantien für den Datenverkehr!



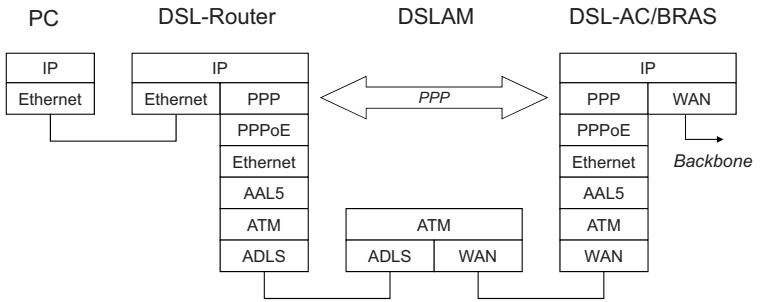
12.4. Datenübertragung über Telefonleitungen: xDSL

- ISDN: Ersetzen des analogen Telefonsystems durch digitales System
- xDSL: x repräsentiert spezifische Realisierungen der DSL-Technik (Digital Subscriber Line)
  - SDSL: Symmetric DSL
  - ADSL: Asymmetric DSL
  - …
- Ziele:
  - Vorhandene und für fast jedes Haus verlegte Telefonleitung (twisted pair) für hochratige Datenübertragung nutzen
  - Koexistenz von analogem Telefonsystem (POTS = Plain Old Telephone System) bzw. ISDN und hochratiger Datenübertragung
- Damit einhergehend: Öffnung des Telefonmarktes
  - *unbundled local loop*: Teilnehmeranschlussleitung wird an Infrastruktur eines Telekom-Wettbewerbers angeschlossen (z.B. Arcor)
  - *line-sharing*: POTS/ISDN und DSL von verschiedenen Anbietern über dieselbe Teilnehmeranschlussleitung (z.B. Telefonica)
  - Bitstromzugang: Wettbewerber bietet Internet-Zugang über Telekom-DSL

12.4.1. xDSL: Szenario



12.4.2. xDSL: Protokolle

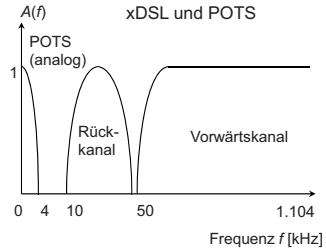


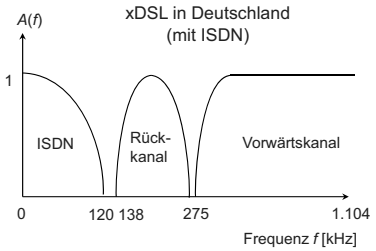
PPP: Point-to-Point Protocol
- Verbindungsaushandlung, Zugangskontrolle

ATM, AAL5: Asynchronous Transfer Mode, ATM Adaption Layer 5
- zellbasiertes Vermittlungsprotokoll (Schicht 3)

12.4.3. xDSL: Realisierung

- Bestimmte Kombination von:
  - Kanalkodierung (Hinzufügen von Redundanz)
  - Echokompensation
  - FDD (Frequency Division Duplex)
  - Frequenzmultiplex, Mehrträgerverfahren (DMT, Discrete Multitone):
    - mehrere Träger in je 4,3125 kHz breiten Bändern und variabler Datenrate (je nach Dämpfung/Rauschen)
  - Bänder werden ausgemessen und ggf. ausgeblendet (hohe Dämpfung,..)
  - adaptive Leitungsentzerrung
    - Ausgleich frequenzabhängiger Dämpfung und Phasenverschiebung





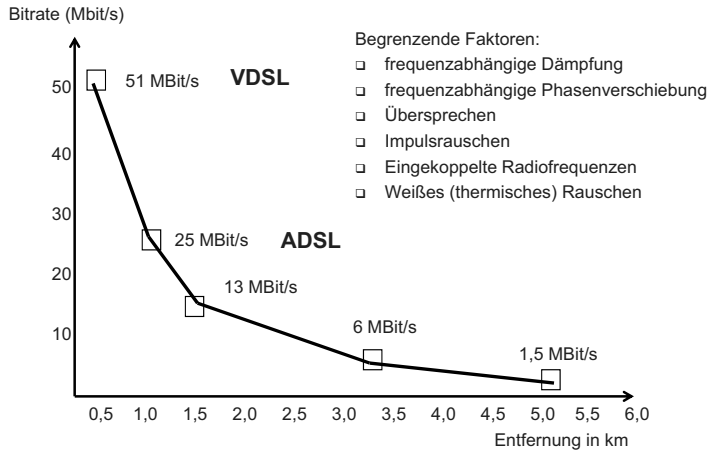
12.4.4. xDSL: Technologien (I)

- ADSL (Asymmetric Digital Subscriber Line):
  - nutzt Frequenzen bis 1,1 MHz
  - 384 kBit/s bis 8 MBit/s Vorwärtskanal (downstream) (je nach Entfernung)
  - 64 bis 1024 kBit/s Rückkanal (upstream) (je nach Entfernung)
  - Reichweite: 3.000-9.000 Meter
  - für private Nutzer gedacht
  - Problem: Annahmen für Bandbreitenaufteilung Vorwärtskanal/Rückkanal stimmen nicht unbedingt (u.a. wegen P2P-Anwendungen)
- Begriffe:
  - Vorwärtskanal: vom Server über das Netzwerk zum Dienstnehmer (Kunden)
  - Rückkanal: vom Dienstnehmer (Kunden) über das Netzwerk zum Server
- Quellen: http://www.adsl.com

xDSL: Technologien (II)

- HDSL (High Data Rate DSL) *(historisch)*
  - bis zu 1,5 Mbit/s symmetrisch über zwei Zweidraht-Leitungen
- SDSL (Symmetric DSL):
  - dieselbe Datenrate im Vorwärts- und Rückkanal
    ➔ Ersatz für ISDN-Primäranschluss, v.a. für Geschäftskunden
  - keine Lücke im Basisband für ISDN/POTS
- ADSL2:
  - nutzt Frequenzen bis 2,2 MHz
  - bis zu 16 Mbit/s im Vorwärtskanal
- ADSL2+:
  - nutzt Frequenzen bis 2,2 MHz
  - bis zu 25 Mbit/s im Vorwärtskanal (bei ISDN: 16 Mbit/s)
  - Hohe Datenraten ermöglichen IPTV
- VDSL/VDSL2 (Very High Speed DSL):
  - VDSL nutzt Frequenzen bis 12 MHz, VDSL2 bis 30 MHz
  - nutzt Frequenzen bis 30 MHz
  - Leitungslänge zwischen DSL-Modem und DSLAM wenige 100 Meter
  - Datenraten bis zu 100 Mbit/s symmetrisch

Bitrate zum Dienstnehmer (Vorwärtskanal)



Begrenzende Faktoren:
- frequenzabhängige Dämpfung
- frequenzabhängige Phasenverschiebung
- Übersprechen
- Impulsrauschen
- Eingekoppelte Radiofrequenzen
- Weißes (thermisches) Rauschen