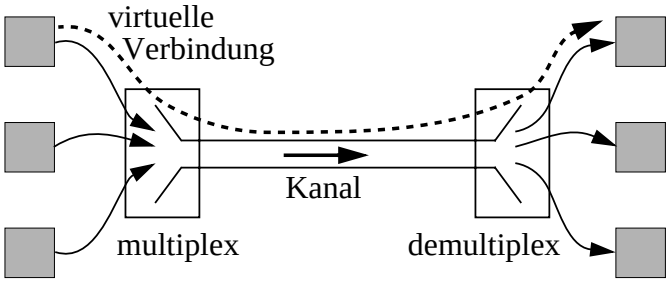
Multiplexen

- Gemeinsame Nutzung von Kommunikationseinrichtungen (Verbindungen, Switch...) und anderer Ressourcen für mehrere Anwender
  - vgl. Timesharing einer cpu

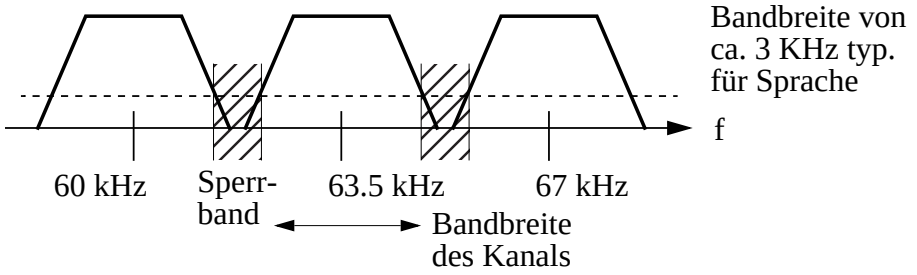
- Insbesondere mehrere virtuelle Verbindungen / Kanäle über eine (längere) physische Verbindung

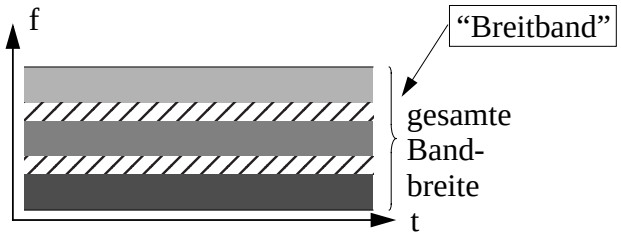


- Vorteile:
  - höhere Auslastung von physischen Verbindungen durch gegenseitigen Ausgleich logischer Verbindungen
  - Kostenvorteil durch gemeinsame Nutzung, da fallende Kosten pro kb/s mit steigender Übertragungsrate

Frequenzmultiplex

- FDM: Frequency Division Multiplexing





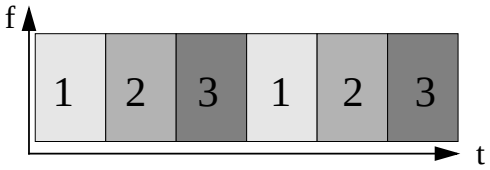
- Aufteilung der Übertragungskapazität in mehrere *Frequenzbänder* mit dazwischenliegenden *Sperrbändern*
  - vgl. TV-Kabel: verschiedene Signale mit einer Bandbreite von je ca. 7 MHz über ein einziges Medium

- Niederfrequente Signale werden durch *Modulation* in das entsprechende Frequenzband "gehoben"

- Klassisches Anwendungsgebiet: Kabelfernsehen

Synchrones Zeitmultiplex

- STDM: Synchronous Time Division Multiplexing



- Aufteilung in Zeitschlitze
- zeitversetzte Übertragung der Signale der Kanäle
- zyklische Zuteilung

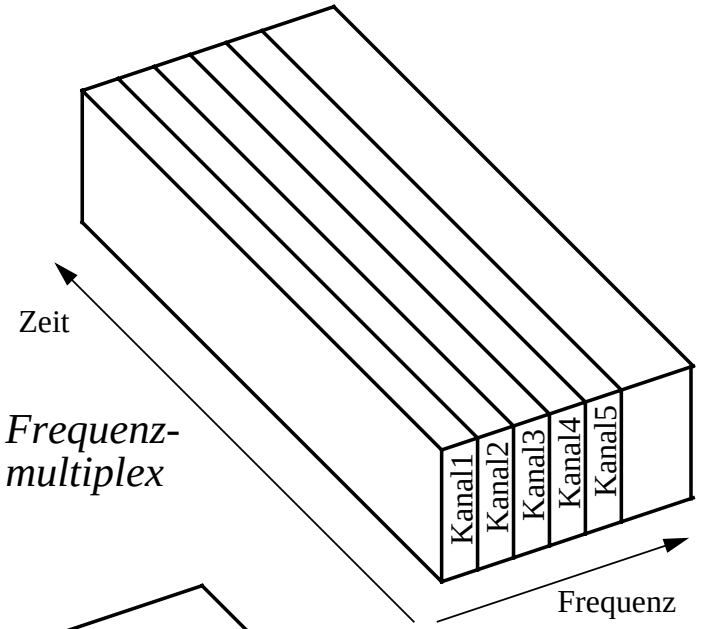
- Bit- oder blockweise Verschränkung (“interleaving”)
- Angewendet bei digitalen Signalen
  - typischerweise bei *Basisbandtechnik*: gesamtes Frequenzspektrum des Übertragungsmediums wird für einen einzigen Kanal genutzt

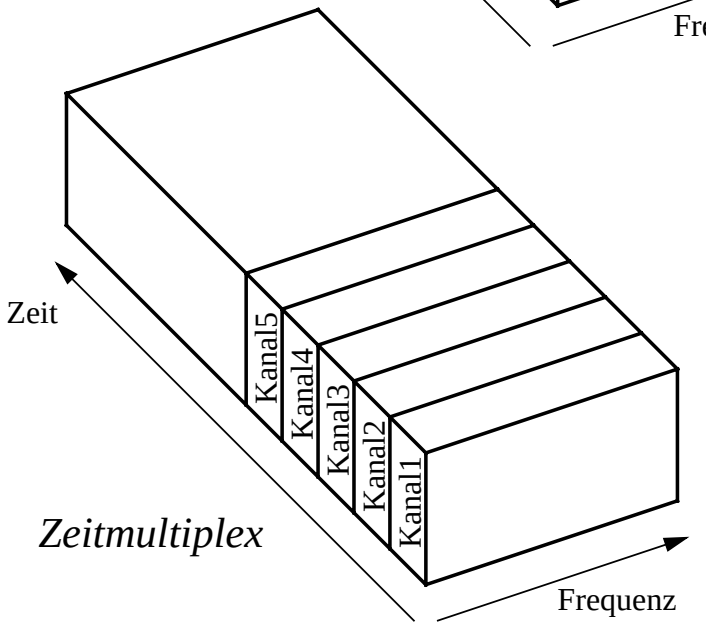
---

Nachteile von FDM und STDM:

- vordefinierte Zahl verschiedener logischer Kommunikationskanäle
- Anteil an der Gesamtkapazität bleibt ungenutzt, wenn über einen Teilkanal nichts fliesst (insbesondere bei starker Varianz; vgl. Anfordern / Lesen einer WWW-Seite)

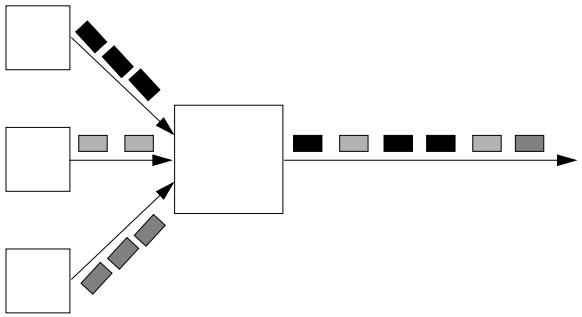
Frequenz- und Zeitmultiplex



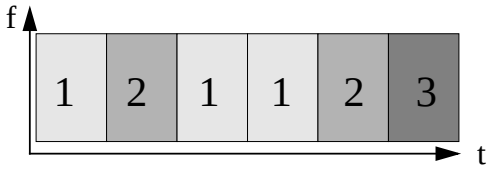


Asynchrones Zeitmultiplex

- ATD: Asynchronous Time Division
- Auch *statistisches* Zeitmultiplex genannt

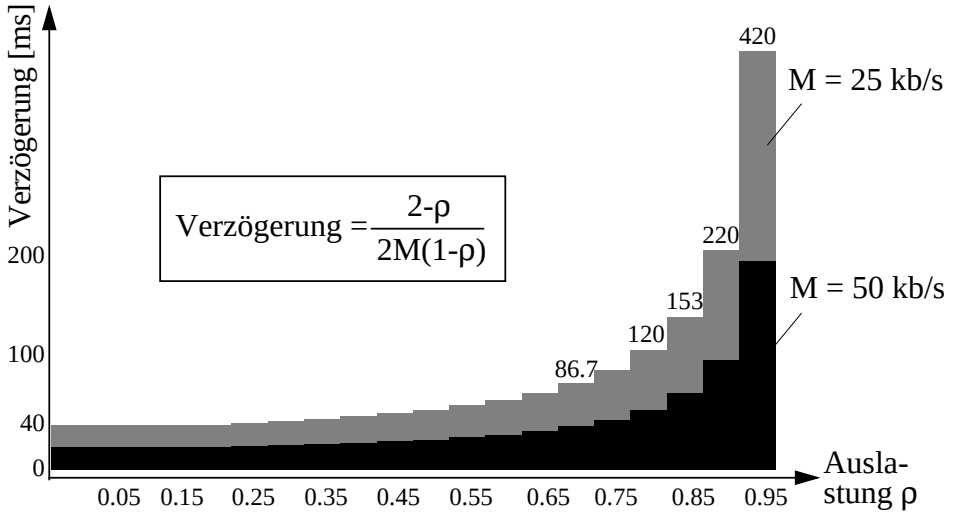


- Zuteilung von Zeitschlitzen (i.a. fester Grösse) nach Bedarf
- Paket = zusammenhängend zu transportierende Daten
- Gute Wahl der Paketgrösse? (Overhead, Zeitvarianz, Fairness...)
- Header eines Paketes muss virtuelle Verbindung kennzeichnen
- Bits pro Kanal zunächst in einem Puffer sammeln, dann stossweise abgeben
- Verschiedene Kanäle konkurrieren miteinander ("contention")
- Pufferüberlauf ("congestion"): Pakete werden oft einfach weggeworfen (grosse Puffer --> u.U. lange Verzögerungen!)



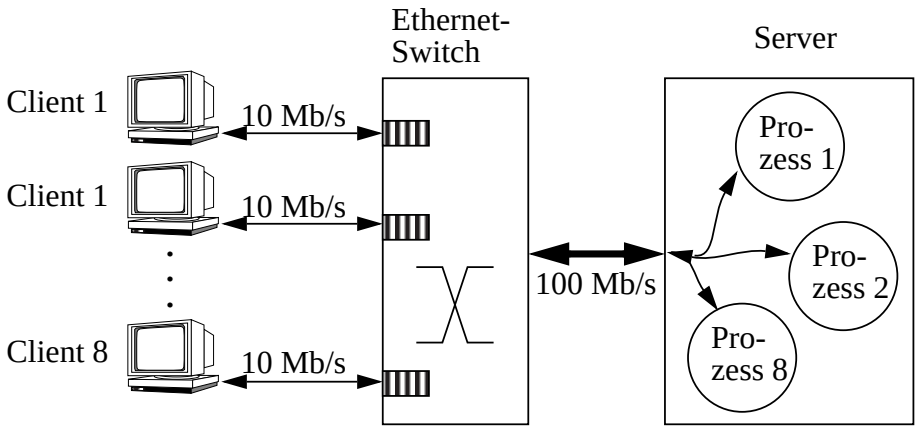
Verzögerung bei ATD

- Berechnung von Verzögerung und Wahrscheinlichkeit von Pufferüberläufen mittels Warteschlangentheorie
  - dazu i.a. typische Annahmen wie statistische Unabhängigkeit der Eingangssignale (nicht immer gerechtfertigt!), Poisson-Prozesse etc.
  - daraus Abschätzung der benötigten Puffergrösse, Abschätzung der Dienstgüte (Verzögerung, mittlere Paketverlustrate etc.)
- Beispiel: Mittlere Verzögerung einer 1000-Bit-Nachricht als Funktion der Auslastung (mit M = Bitrate):
  - Verzögerung steigt oberhalb einer Auslastung von ca. 80% drastisch an
  - Grund: Langes Warten von Datenpaketen in Eingangspuffern des Multiplexers (da auch andere Kanäle öfters gleichzeitig Daten senden)
  - 100% Auslastung ist nicht erreichbar!



- Wir behandeln die Leistungsanalyse mittels Warteschlangentheorie in dieser Vorlesung nicht näher
  - Alternative zur Analyse mittels Warteschlangentheorie: Simulation

Ein Beispiel für ATD



- Was geschieht, wenn alle Clients gleichzeitig mit 10 Mb/s mit ihren Server-Prozessen kommunizieren?

- Was geschieht, wenn man statt 8 Clients z.B. 24 Clients an den Switch anschliesst?
  - Clients nutzen Kapazität von 10 Mb/s gar nicht ganz aus
  - Prinzip Hoffnung: Selten senden viele gleichzeitig
  - erwartete Gesamtlast am Eingang im Mittel unter 100 Mb/s
  - Lastspitzen können hoffentlich über Puffer abgefangen werden

- Konsequenzen, wenn Verzögerung auf dem Kanal vom Client zum Server-Prozess exorbitant steigt?
  - Timeouts, Verlust von Datenpaketen, Wiederholungen
  - effektive Datenrate sinkt dadurch!
  - Anwendungen stürzen u.U. ab (“Server down”)

Kommunikationsprotokolle

- Kommunikation in verteilten Systemen geschieht ausschliesslich über *Nachrichten*

- Hierbei sind gewisse *Normen* zu beachten, damit die Kommunikation klappt

- *Protokoll* = Festlegung der Regeln und des algorithmischen Ablaufs bei der Kommunikation zwischen zwei oder mehr Partnern

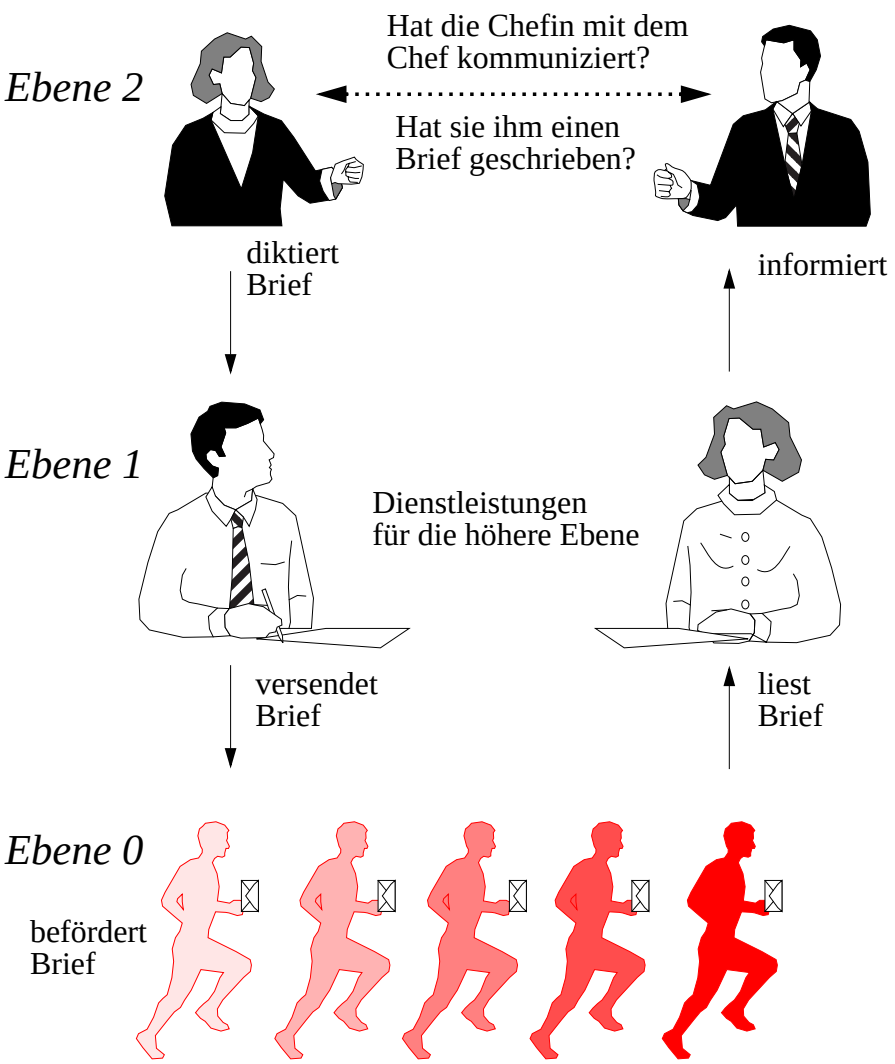
---

- Es müssen viele Vereinbarungen getroffen werden, z.B:
  - Steckergrösse
  - Wieviel Volt repräsentieren “0” bzw. “1”?
  - ist das erste Bit vorne oder hinten?
  - was tun bei einer fehlerhaften Übertragung?
  - wird ein Rasterbild zeilen- oder spaltenweise übertragen?
  - ...

--> Vereinbarungen auf z.T. unabhängigen “Ebenen”

--> Bildung sinnvoller Schichten

Schichten (“layers”) bzw. Ebenen



Schichten (2)

- Eigentliche (“physische”) Kommunikation geschieht auf der untersten Ebene

- Auch mittlere Ebenen (Sachbearbeiter) kommunizieren miteinander

- Nehmen dazu *lokale Dienste* einer tieferen Ebene in Anspruch

- Ebene 0 kann ausgetauscht werden (z.B. Fax statt gelbe Post), ohne dass sich auf höherer Ebene etwas ändern muss (--> Transparenz)

- Ebenen 0 und 1 können ausgetauscht werden (z.B. wenn Ebene 2 den Telefondienst statt dessen in Anspruch nimmt)

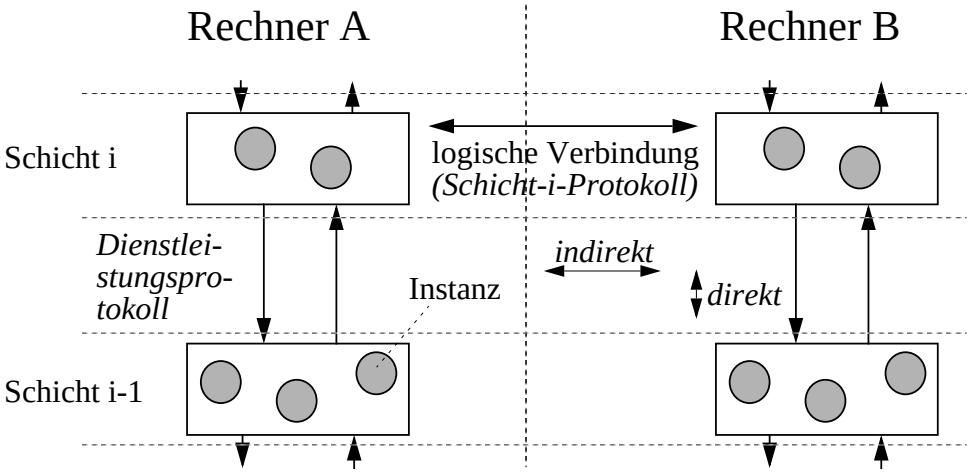
Vgl. auch Speditionsfirma, die verscheidene Transportdienste in Anspruch nehmen kann:

- Auf jeder Ebene gelten jeweils eigene Regeln (Strassenverkehr: rechts vor links...), die aber nicht nach oben durchschlagen

- Leistungsfähigkeit eines Dienstes (Kosten, Geschwindigkeit, Güte...) sind allerdings weiter oben spürbar

- Verschiedene Transportdienste bieten i.w. den gleichen Service --> sind austauschbar

Schichtenmodell



- Schicht i benutzt Protokoll der Ebene i und *lokale Dienste*, die von der Ebene i-1 angeboten werden

- Damit der nächst höheren Schicht ein Dienst angeboten werden kann, kommunizieren Instanzen der Schicht i gelegentlich mit ihren Partnern auf entfernten Rechnern

"entity"

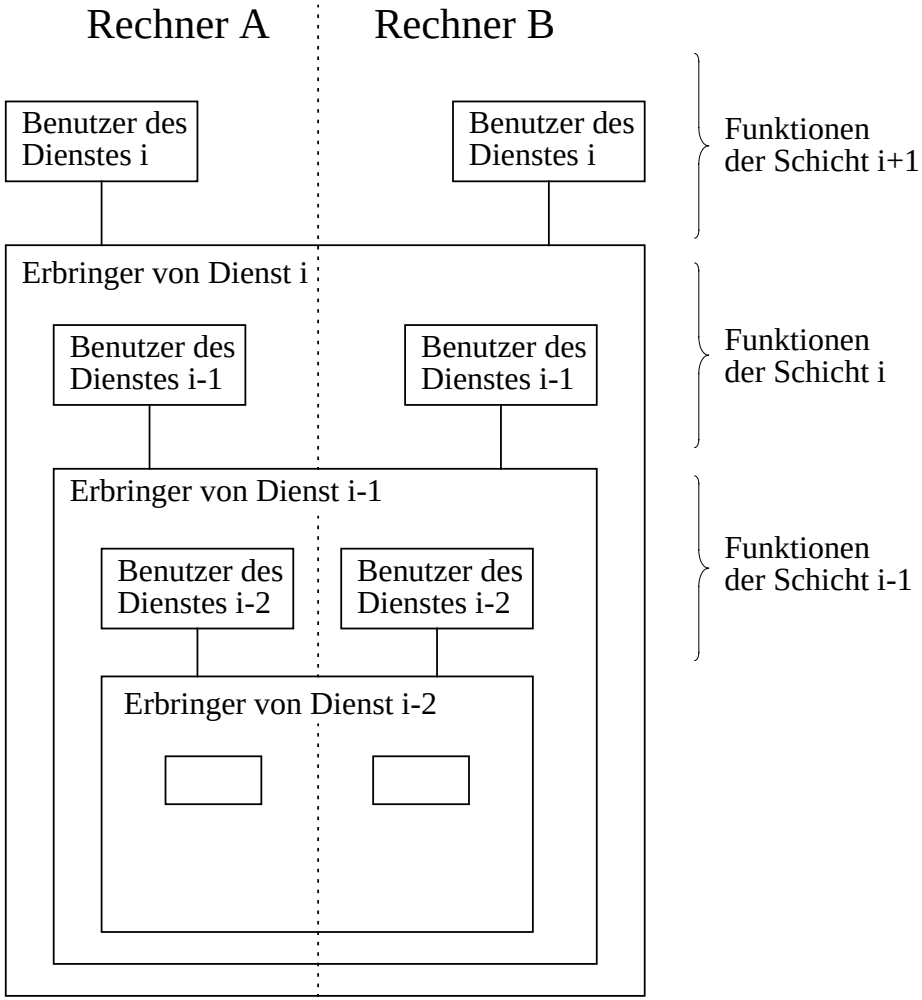
"peer entities"

- Die *logische Verbindung* zwischen entfernten Instanzen ist indirekt; die Kommunikation auf dieser Ebene wird von einem *Schicht-i-Protokoll* geregelt

- Nachrichten auf Schicht i können nicht nur von Schicht i+1 veranlasst sein, sondern auch von der eigenen Schicht (z.B. durch timeouts)

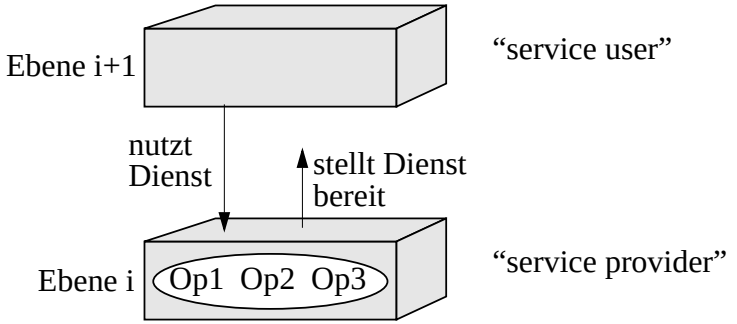
- Jede Schicht kostet... (Zusatzaufwand, "overhead")

Hierarchische Dienststruktur

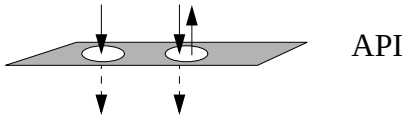


Dienst, Interface, Protokoll

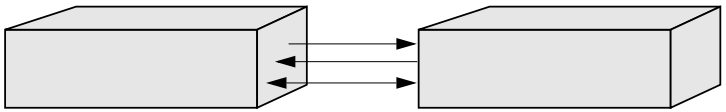
- Dienst: Menge von Operationen (“Dienstprimitive”), die eine Schicht der darüberliegenden anbietet
  - Dienstdefinition sagt aus, *was* ein Dienst tut (nicht *wie* oder *wer* ihn nutzt)



- Interface einer Schicht: *Wie* der Dienst der Schicht in Anspruch genommen werden kann
  - z.B. Parameter etc. oder Dienstleistungsprotokoll



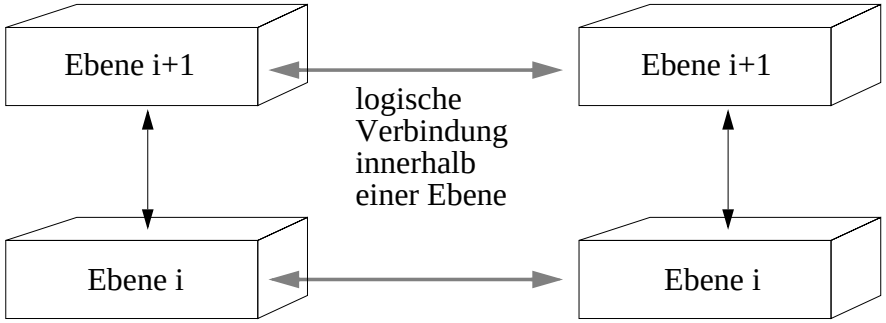
- Protokoll beschreibt die (verteilte) *Realisierung* eines Dienstes einer Schicht (“Schicht-i-Protokoll”)



-Protokoll kann (“problemlos”) verändert werden, solange der gleiche Dienst nach oben (mit gleichem Interface) bereitgestellt wird

Wiederkehrende Aufgaben verschiedener Schichten

- Jede Schicht hat eine dedizierte Aufgabe; dennoch existieren u.U. gleichartige Teilaufgaben in unterschiedlichen Schichten



- Adressierung von Empfänger (und ggf. Absender)
- Fehlererkennung und ggf. -behebung
- Flusssteuerung (zu schnellen Sender bremsen)
- Paketisierung grosser Nachrichten in Maximallänge des Dienstes der tieferen Schicht
- Multiplexen
- Komprimierung
- Verschlüsselung

Vorlesung “Information und Kommunikation”

Denkübung: Diskutieren, auf welchen Ebenen sinnvollerweise diese Aufgaben angesiedelt werden sollten!

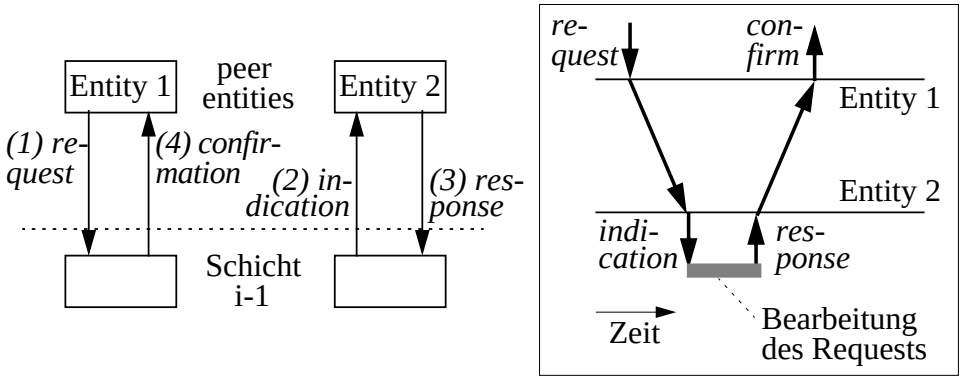
- Diese Aufgaben werden jedoch oft unterschiedlich (“schichtspezifisch”) gelöst

- Gelegentlich verzichtet man auf eine Aufgabe (z.B. Fehlererkennung), wenn dies von einer anderen Schicht quasi perfekt miterledigt wird
  - sogen. “End-zu-End-Argument”

Schichtenarchitekturen

- Schichtenbildung bei Protokollen:
  - Erleichtert Entwurf, Implementierung, Klassifikation
  - “Layering” ist gängiges Prinzip in der Informatik (Abstraktion; information hiding; Reduktion der Komplexität)

- “Kunst”, eine geeignete Schichtenarchitektur zu definieren:
  - Verschiedene Grundaufgaben sollen in verschiedenen Schichten liegen
  - Ähnliche Aufgaben in gleichen Schichten
  - Jeder Schicht sollte eine klare Gesamtaufgabe zukommen
  - Anzahl der Schichten sollte nicht zu gross werden
  - Eine höhere Schicht sollte von den Aufgaben tieferer Schichten abstrahieren
  - Informationsfluss zwischen den Schichten sollte gering sein

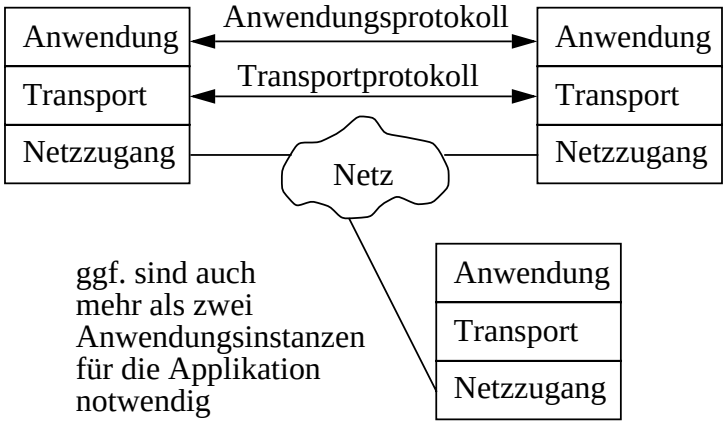


- Typen von Service-Primitiven zur Realisierung von Services (OSI-Modell):
  - *Request*: Entity auf Ebene i fordert Service von Entity auf Ebene i-1 an
  - *Indication*: Entity auf Ebene i-1 teilt Request einer i-Entity mit
  - *Response*: i-Entity antwortet auf Indication einer i-1 Entity
  - *Confirmation*: i-Entity antwortet auf Indication einer i-1 Entity

(Vgl. Herstellen einer Telefonverbindung: Nummer wählen, klingeln... Kritik an diesem Modell: Telefone klingeln; Rechner nicht.)

Ein 3-Ebenen-Modell

- In einem generellen Sinn gibt es drei Mitspieler bei der Kommunikation: *Anwendung*, *Rechner* und *Netz*
  - Daten gelangen aus dem *Netz* zum *Rechner*; dieser (bzw. die zugehörige Systemsoftware) reicht diese an die *Anwendung* weiter



- Spezifika des Netzes; Treiber der Netzkarte; Adressen der Rechner etc. werden in einer eigenen Ebene verborgen (*Netzzugangsebene)*

- Die sichere (korrekte) Datenübertragung ist Zweck der *Transportschicht*
  - ist eine Aufgabe, die damit für viele Anwendungen “gleichzeitig” erledigt wird
  - isoliert die Anwendungen von der Technologie, den Spezifika und den Unvollständigkeiten des verwendeten Netzes

- Die *Anwendungsschicht* enthält die (verteilten) Applikationen (z.B. Homebanking-Software)

ISO-OSI-Referenzmodell

International Standardisation Organisation

Open Systems Interconnection

- *Referenzmodell:*

  Kein Protokoll, sondern ein “Schema” für Protokolle und deren Normierung

  - Anzahl der Ebenen (Vorschlag: 7)
  - prinzipielle Aufgaben der verschiedenen Ebenen

--> Strukturelle Basis (“Architekturmodell”) für Protokolle und Standards zur Datenkommunikation

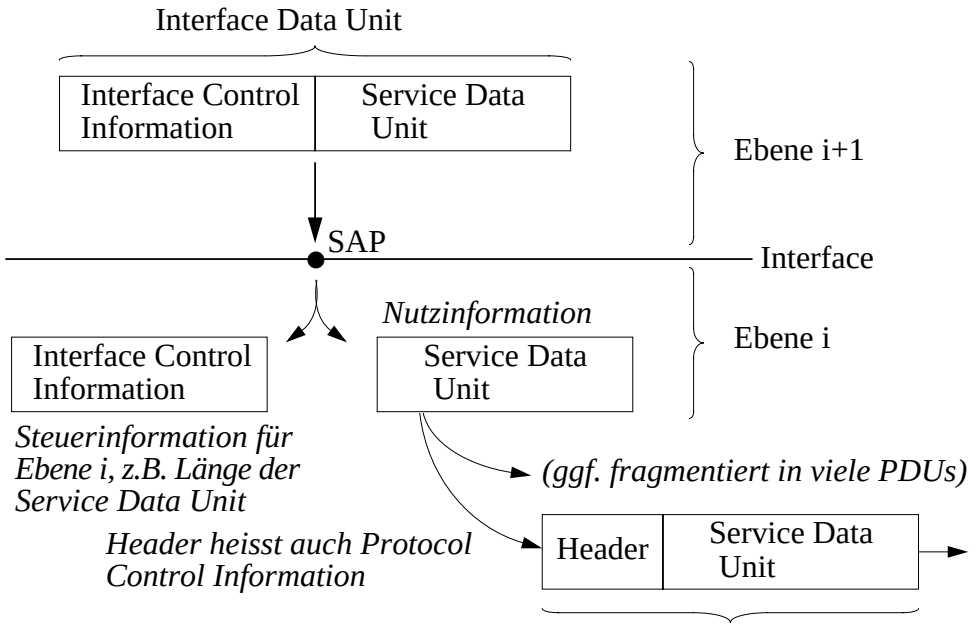
  - einheitliches Vokabular für Normierungszwecke

Offenes System:

- Festlegung spezifischer Standards entsprechend einer vereinbarten Architektur
- Jeder Anwenderprozess kann mit jedem anderen kommunizieren, sofern er sich an die vereinbarten Regeln hält (offengelegte Schnittstellen)

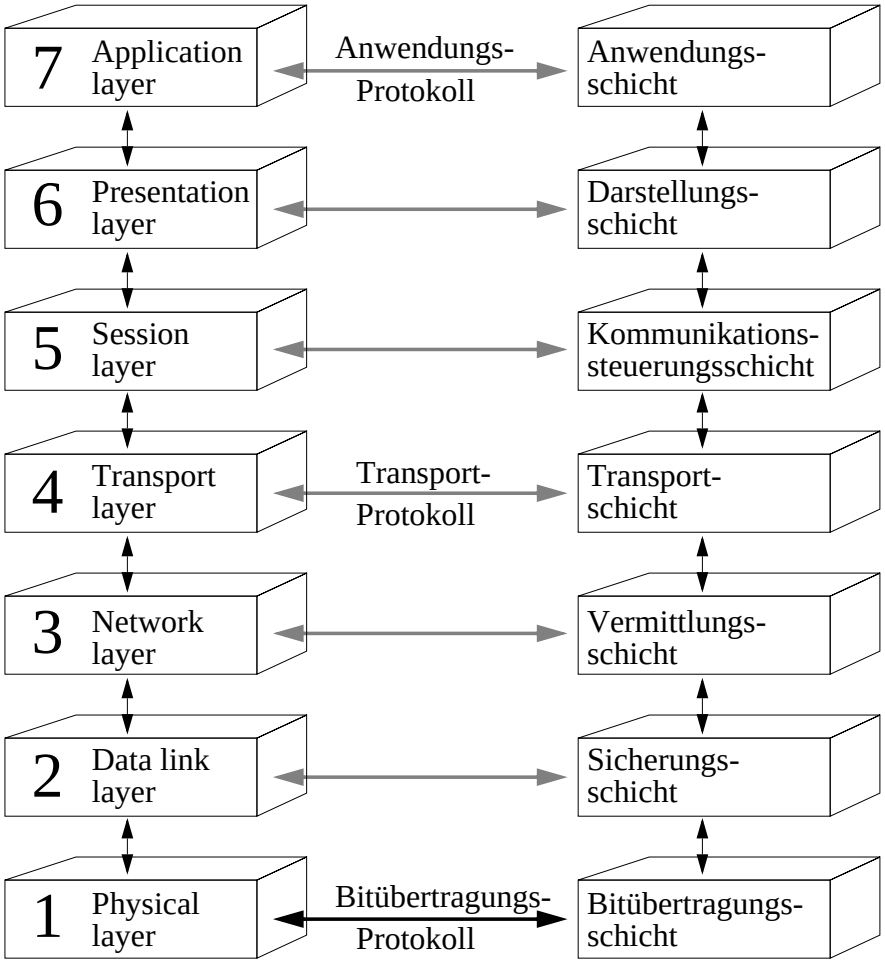
Service Access Point (SAP) und PDU

- OSI-Terminologie (ein ganz kleiner Einblick...)
- *Entity* = Aktive Einheit einer Schicht
  - realisiert in Hardware oder Software
- *Peer entities* = Instanz der gleichen Schicht auf verschiedenen Rechnern
- *SAP* (Service Access Point) = Stelle, wo der Service angeboten wird (identifiziert durch eine Adresse)



- Interface Data Unit geht über einen SAP von der Ebene i+1 zur Ebene i

Die OSI-Protokollhierarchie



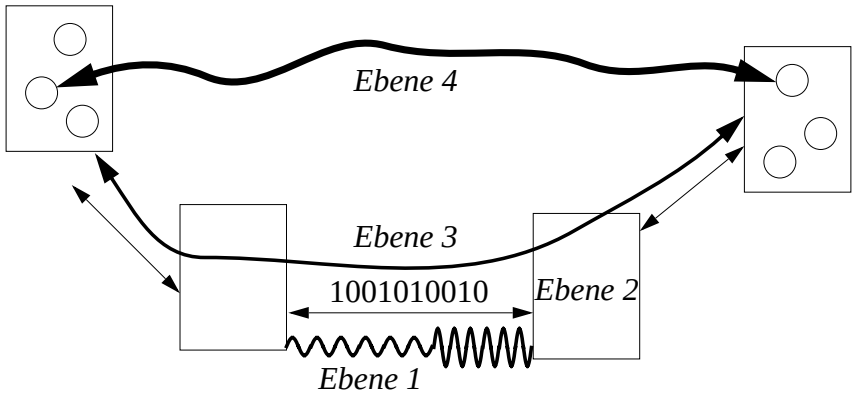
Wieso gerade 7 Schichten?

- in einigen Protokollen sind einige Schichten fast “leer”
- in einigen Protokollen werden einige Schichten nochmals unterteilt

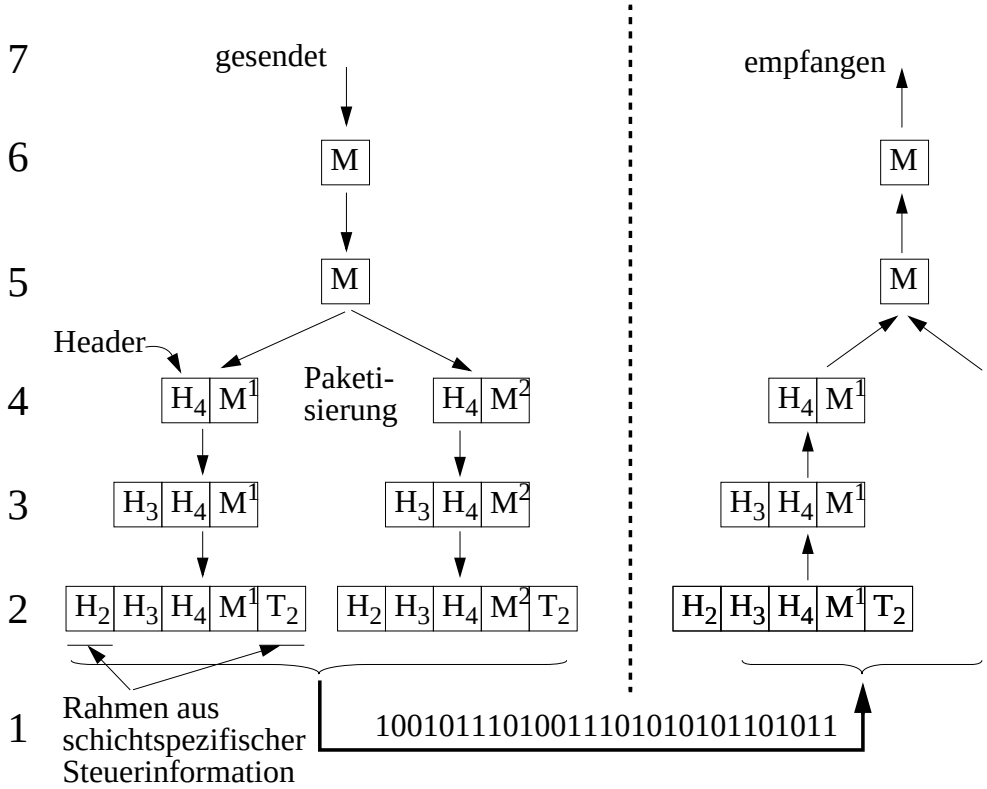
Was sind die Aufgaben der Schichten? -->...

Zweck der 7 Schichten

7	Kommunikationsabläufe der *Anwendung*
6	Systemunabhängige *Datendarstellung*
5	Verbindung über *längeren Zeitaum* aufrechterhalten
4	*Sichere* Verbindung zwischen *Prozessen* herstellen
3	*End-zu-End*-Verbindung zwischen *Rechnern*
2	Datenübertragung zwischen *benachbarten* Stationen
1	“Physikalische” Übertragung von *Signalen*



Informationsfluss und Nachrichtenformat



- Header H_i und Trailer T_i der Schicht i
  - Prüfbits
  - Sequenzzähler

  } und andere "Verwaltungsdaten" } hinzufügen bzw. entfernen
- Eigentliche Nutzdaten M
  - Gesamtnachricht der Schicht i = Nutzdaten der Schicht i-1
- Einige Schichten paketisieren lange Nachrichten
  - Nachrichtenlänge auf unterer Ebene durch Puffergrössen begrenzt
  - Zusammenbau ("assembly") auf der Empfangsseite

Aufgaben der Schichten 1 und 2

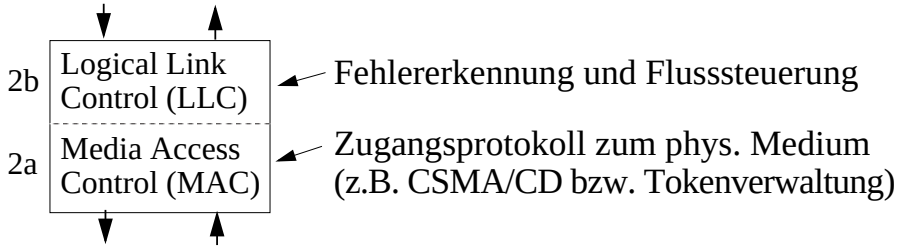
1. *Physical Layer*
   - Normung von Steckern und Kabeleigenschaften
   - Physikalische Darstellung der Bits
     - z.B. Strom / Spannung (z.B. "0" = 1V), oder
     - Licht (mit Vereinbarung der Wellenlänge etc.), oder ...
   - Kodierung von Bitfolgen und ggf. Taktsignale
     - Takt zwecks Bitsynchronisation von Sender und Empfänger
   - Übertragung unstrukturierter Bitfolgen über ein Medium
     - z.B. Telefonleitung, Lichtleiter, Funkkanal für Radiosignale...
   - Aktivierung / Deaktivierung von Leitungen
   - Beispiele: RS232-C oder X.21 der ITU

---

2. *Data Link Layer*

Dafür Aufteilung des Bitstroms in Pakete!

- Erkennung und Behebung von Übertragungsfehlern
  - z.B. mit Sequenznummern und Prüfsummen
  - Meldung nicht-behebbarer Fehler nach oben
- Bei LAN: Aufspaltung in zwei Teilschichten:

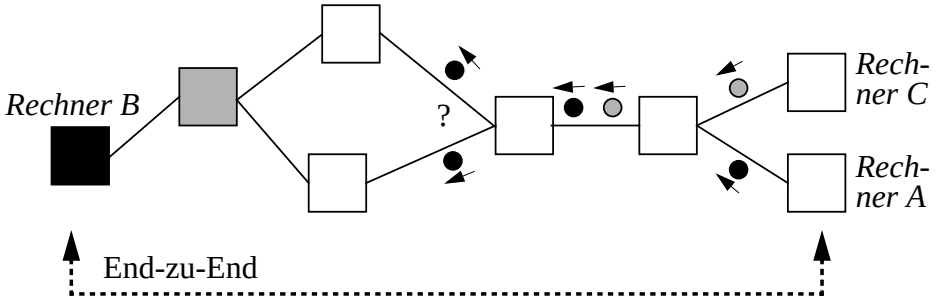


- Adapterkarten, die das Protokoll "in Hardware" abwickeln
- Bsp: HDLC-Protokoll (bei X.25)

Schicht 3 (Network Layer)

- Verknüpft Teilstreckenverbindung zu Endsystemverbindungen
- Wegewahl (Routing)
- Multiplexen von Verbindungen
- Ggf. Fehlerbehebung und ggf. Flusssteuerung (u.U. über mehrere Zwischensysteme hinweg)

Insbesondere bei Kommunikation von Rechnern unterschiedlicher Leistung (Rückkanal notwendig!)



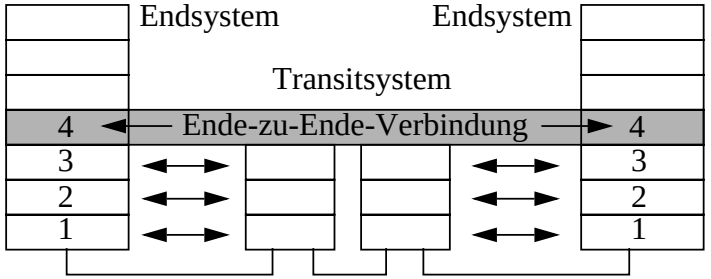
- Beachte: Meisten Aufgaben entfallen in LANs
- Man unterscheidet:
  1) *verbindungslos* (“packet switching”; Datagramm)
     - jedes Paket wird einzeln geroutet
  2) *verbindungsorientiert* (“circuit switching”)
     - Einrichtung einer virtuellen Verbindung
     - Wegewahl i.a. nur bei explizitem Verbindungsaufbau
     - expliziter Verbindungsabbau notwendig
- Beispiel: IP im Internet

Schicht 4 (Transport Layer)

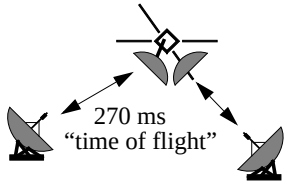
- Aufbau und Abbau
- Multiplexen mehrerer virtueller

“Benutzer” des Transportdienstes in den Endsystemen

- Logische Verbindung zwischen (adressierten!) Prozessen (bzw. ports, sockets...) statt Rechnern



- Reihenfolgeerhaltende, sichere End-zu-End-Verbindung
- Flusssteuerung (“flow control”) z.B. *Sliding window-Protokoll:*
  - Anzahl unbestätigter Pakete vereinbaren
  - kontinuierlicher Datenfluss auch bei langen Verzögerungen



- Abstrahiert von der Art und Natur des benutzten Netzes
  - Erbringung eines Dienstes mit vereinbarter (bzw. “ausgehandelter”) Dienstqualität wie z.B. Fehlerrate oder Schutz / Sicherheit unabhängig von den Leistungen der darunterliegenden Schicht
- Bietet daher *Transparenz* bzgl. Übertragungs- und Vermittlungstechnik sowie benutzten Teilnetzen
- Bsp: TCP im Internet
- Nachrichten beliebiger Länge werden in Pakete aufgeteilt; Adressen des network layer werden hinzugefügt
  - ausserdem: Sequenznummern, Prüfsumme (z.B. CRC) für Bitfehler und weitere Kontrollfelder für die Flusssteuerung etc.