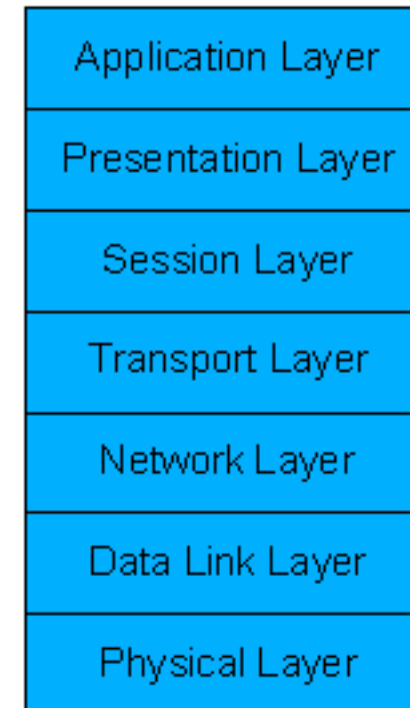




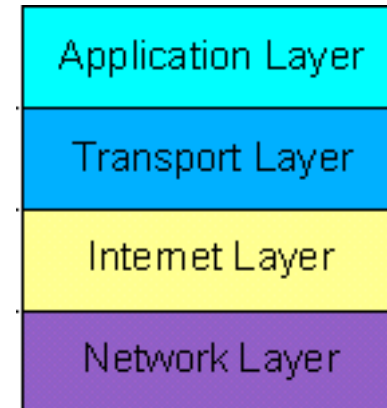
- Computernetzwerke sind sehr komplexe Gebilde
- Beschreibung des Gesamtsystems praktisch unmöglich
- daher: Aufteilung in Module
- Module werden unabhängig voneinander beschrieben
- Schnittstellen zwischen Modulen einzige Verknüpfung
- Um darüber zu reden: Referenzmodelle
- Übersicht, wo man sich gerade im Gesamtsystem befindet

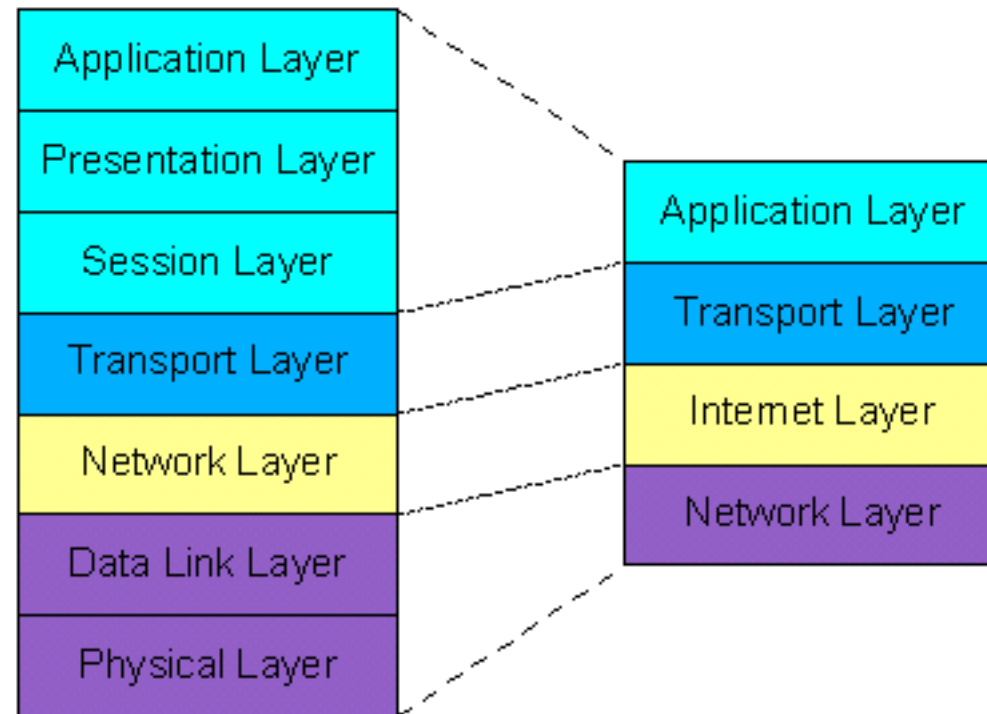
- Theoretisches Modell
- Legt fest, *welche* Aufgaben auf einer Schicht bearbeitet werden
- Beschreibt *nicht, wie* Aufgaben bearbeitet werden
- Wird zum Bezugnehmen verwendet



- Benannt nach den beiden primären Protokollen TCP und IP
- Beruht auf den Vorschlägen, die bei der Fortentwicklung des ARPANETs gemacht wurden
- Zeitlich vor dem OSI-Referenzmodell entstanden
- Erfahrungen des TCP/IP- Modells sind mit in die OSI-Standardisierung eingeflossen
- Besteht aus vier Schichten

- Unabhängigkeit von der verwendeten Netzwerk-Technologie.
- Unabhängigkeit von der Architektur der Hostrechner.
- Universelle Verbindungsmöglichkeiten im gesamten Netzwerk.
- Ende-zu-Ende-Quittungen.
- Standardisierte Anwendungsprotokolle.





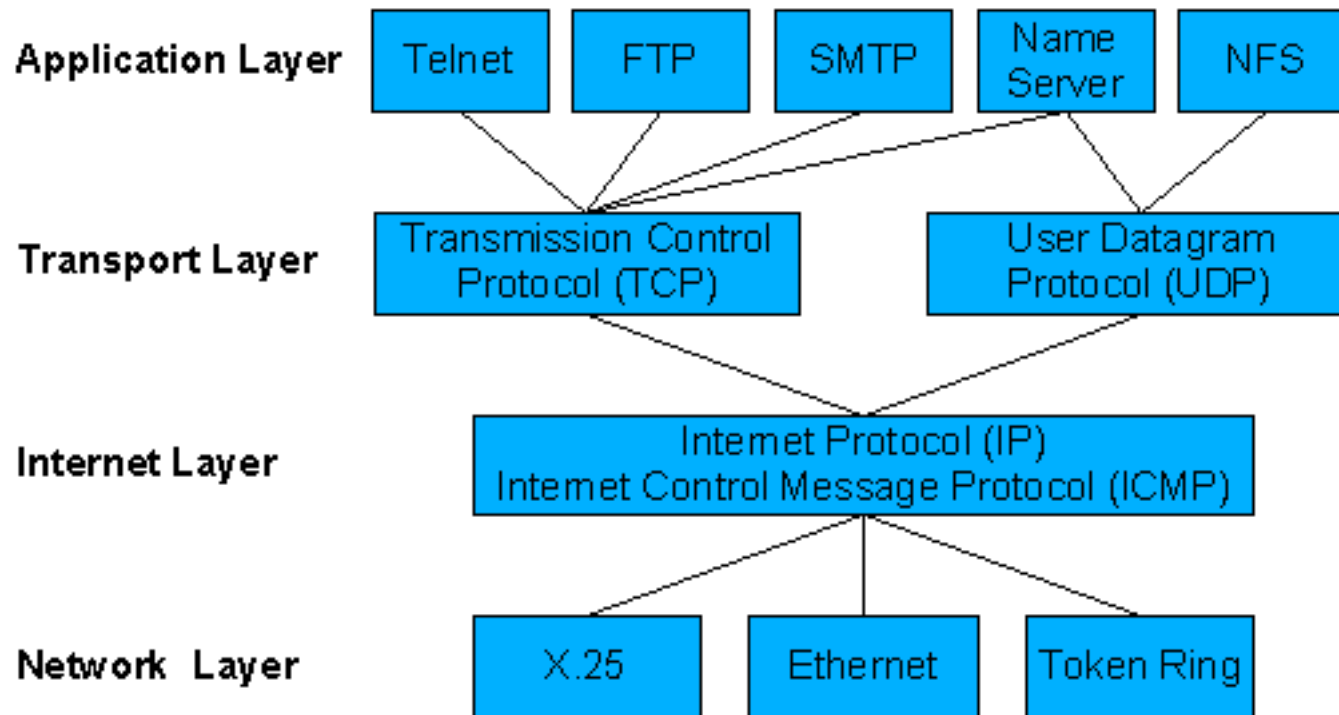
- Umfaßt alle höherschichtigen Protokolle des TCP/IP-Modells
- Wird auch Verarbeitungsschicht oder Anwendungsschicht genannt
- Erste Protokolle dieser Schicht:
  - TELNET (für virtuelle Terminals)
  - FTP (Dateitransfer)
  - SMTP (zur Übertragung von E-Mail)
- Weitere Protokolle dieser Schicht:
  - DNS (Domain Name Service)
  - HTTP (Hypertext Transfer Protocol)
  - ...

- Ermöglicht die Kommunikation zwischen den Quell- und Zielrechnern
- Zwei Ende-zu-Ende-Protokolle:
  - Transmission Control Protocol (TCP)
    - \* Zuverlässig
    - \* Verbindungsorientiert
    - \* Datenstrom/Bytestrom wird fehlerfrei an einen anderen Rechner übermittelt
  - User Datagram Protocol (UDP)
    - \* Unzuverlässig
    - \* Verbindungslos
    - \* Z.B. für Abfragen und Anwendungen in Client/ Server-Umgebungen
    - \* Zwar keine genaue, aber dafür schnelle Datenübermittlung



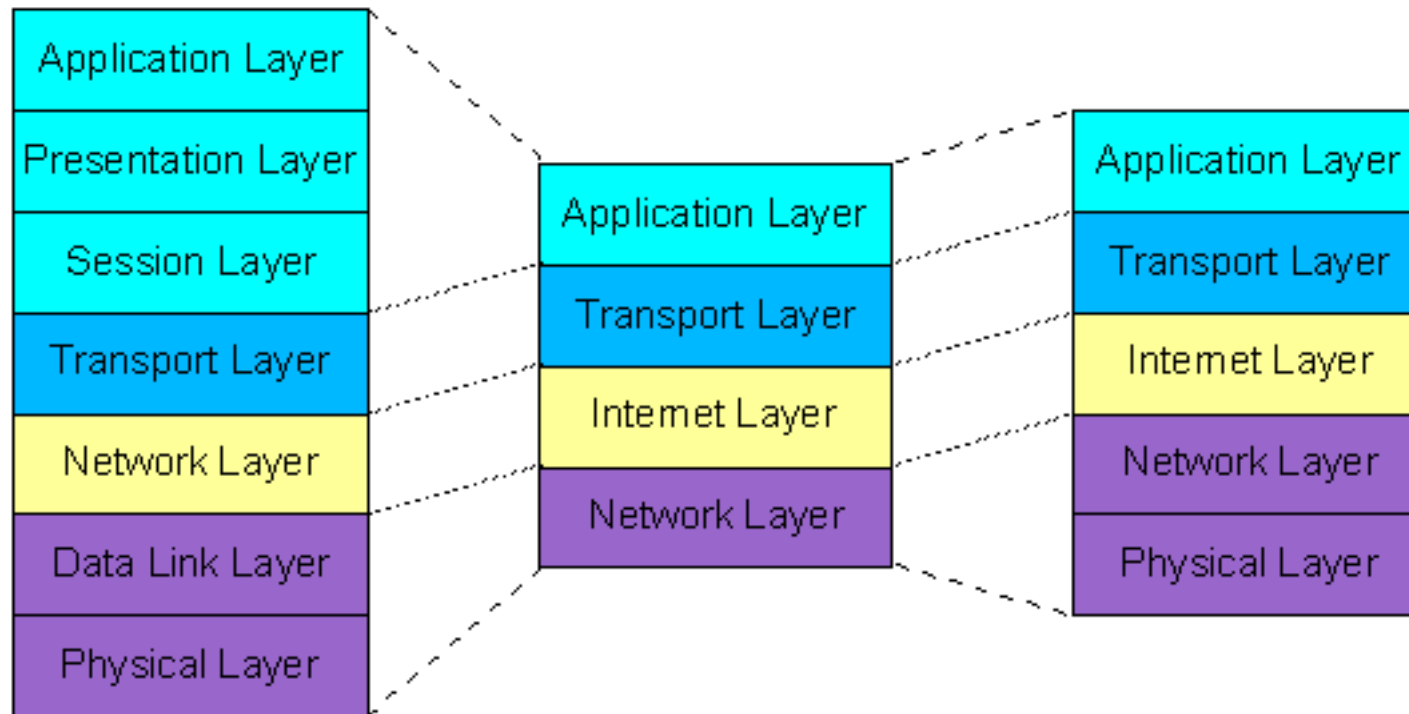
- Nur ein Protokoll: Internet Protokoll (IP)
- Kann von allen am Netzwerk beteiligten Rechnern verstanden werden
- Aufgabe der Schicht: IP-Pakete richtig zustellen
- Dafür nötig: Routing der Pakete
- Teilprotokoll: Internet Control Message Protocol (ICMP)
- ICMP dient zur Übertragung von Diagnose- und Fehlerinformationen für das Internet Protocol.

- Definitionslücke
- Festgelegt ist lediglich, daß zur Übermittlung von IP-Paketen ein Host über ein bestimmtes Protokoll an ein Netz angeschlossen werden muß
- Dieses Protokoll ist nicht weiter definiert
- Es weicht von Netz zu Netz und Rechner zu Rechner ab
- Gängige Protokolle:
  - Ethernet (IEEE 802.3)
  - Serial Line IP (SLIP)
  - ...



Andrew S. Tanenbaum:

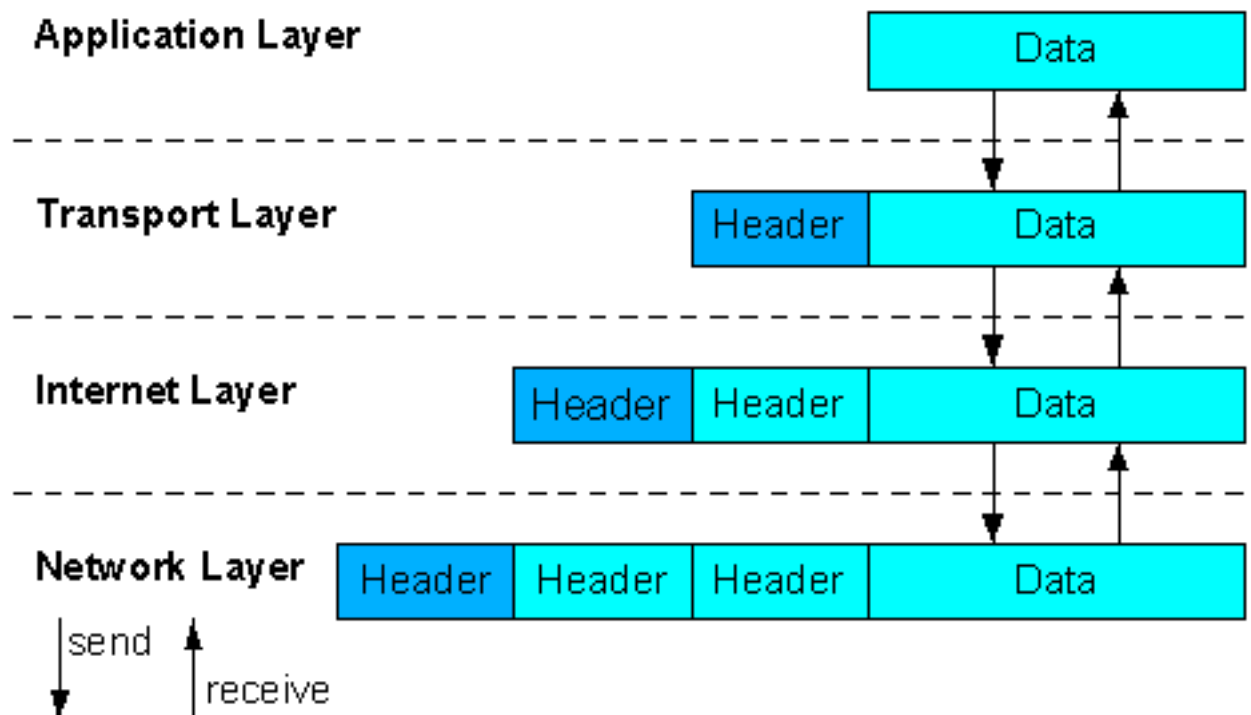
- „(...) Viertens unterscheidet das TCP/IP-Modell nicht zwischen den Bitübertragungs- und Sicherungsschichten (erwähnt sie nicht einmal).
- Diese Schichten sind völlig unterschiedlich.
- Die Bitübertragungsschicht hat mit den Übertragungsmerkmalen von Kupferdarht, Glasfaser und drahtlosen Kommunikationsmedien zu tun.
- Die Sicherungsschicht ist darauf beschränkt, den Anfang und das Ende von Rahmen abzugrenzen und sie mit der gewünschten Zuverlässigkeit von einem Ende zum anderen zu befördern.
- Ein korrektes Modell sollte beides als separate Schichten beinhalten. Das TCP/IP-Modell tut das nicht.”

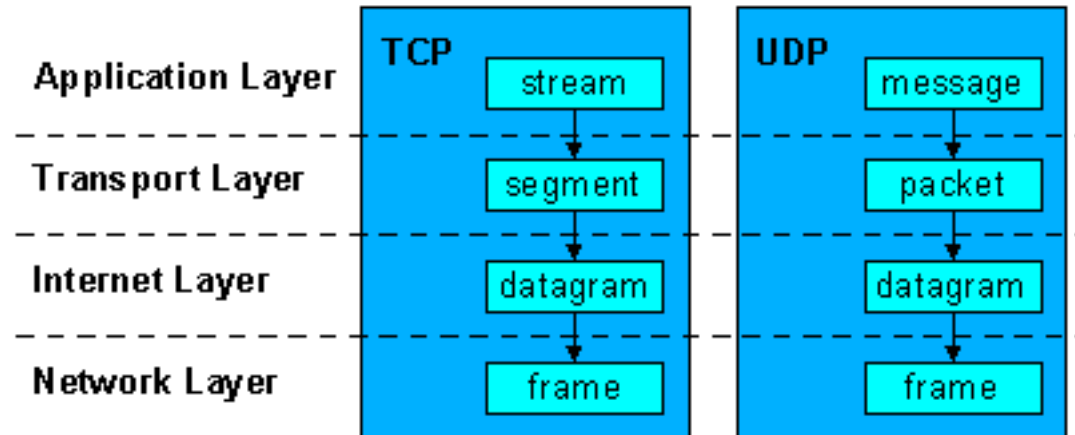


Daten durchlaufen den TCP/IP-Protokollstapel

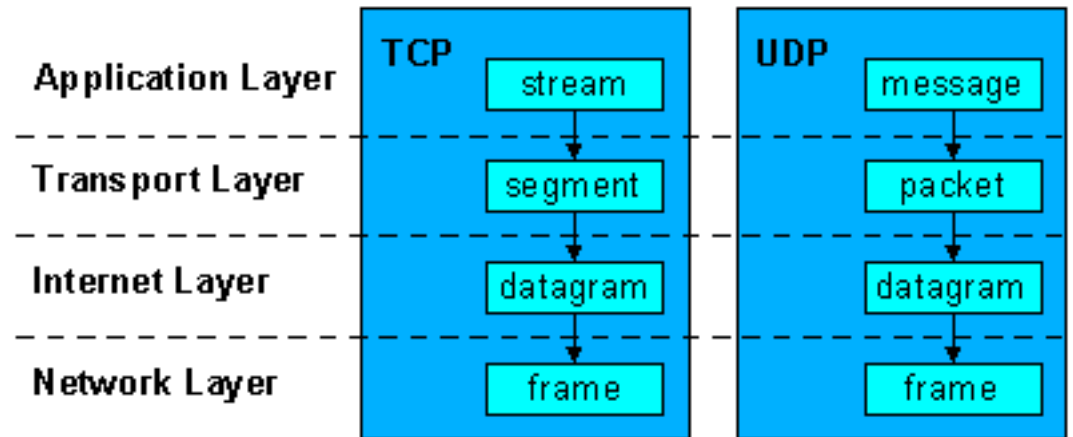
- von der Applikationsschicht zur Netzwerkschicht (beim Senden)
- von der Netzwerkschicht zur Applikationsschicht (beim Empfangen)

Dabei: Anwendung des Prinzips der Datenkapselung von Kontrollinformationen





- Innerhalb der Schichten werden Daten mit verschiedenen Termini benannt
- Jede Schicht hat ihre eigenen Datenstrukturen
- Applikationen, die TCP benutzen, bezeichnen Daten als *Strom (stream)*
- Applikationen, die UDP verwenden, bezeichnen Daten als *Nachricht (message)*

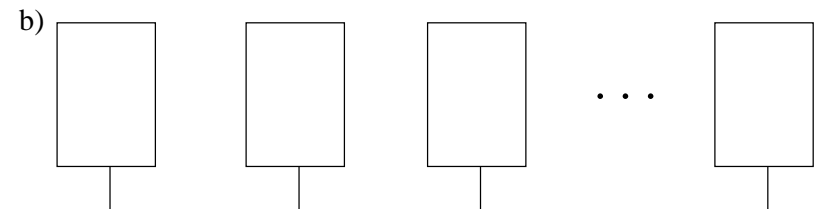
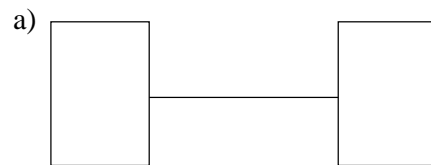


- Auf Transportschicht:
  - TCP: Daten werden als *Segment (segment)* bezeichnet
  - UDP: Daten werden als *Paket (packet)* bezeichnet
- Auf Internetschicht: Daten als *Datagramm (datagram)* bezeichnet
- Auf Netzwerkebene: *Pakete* oder *Rahmen (frames)*

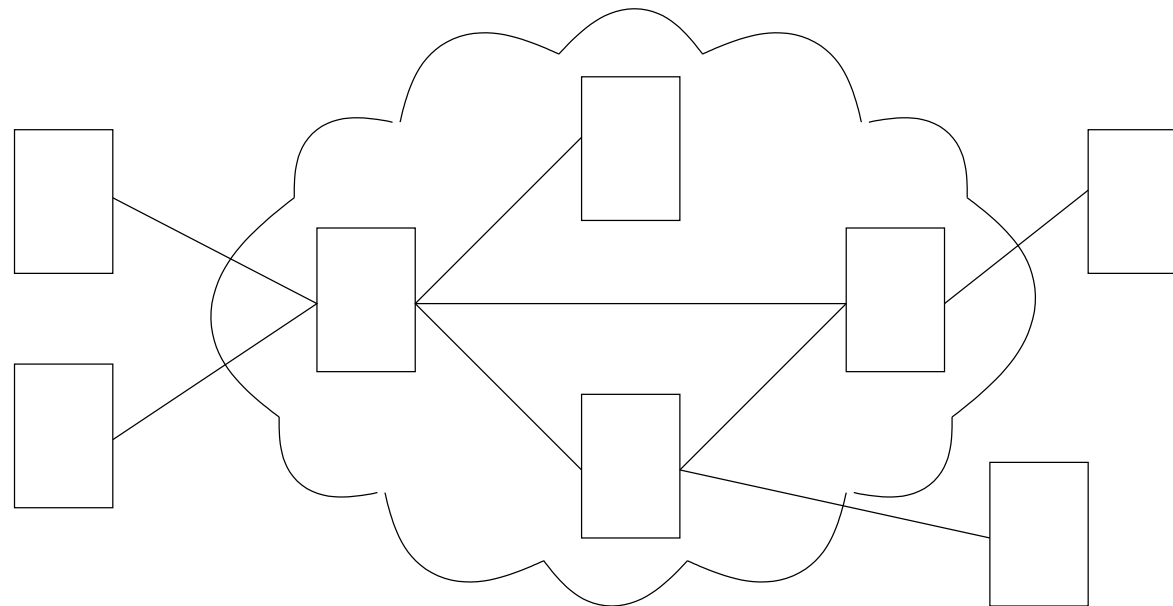


## Anforderungen:

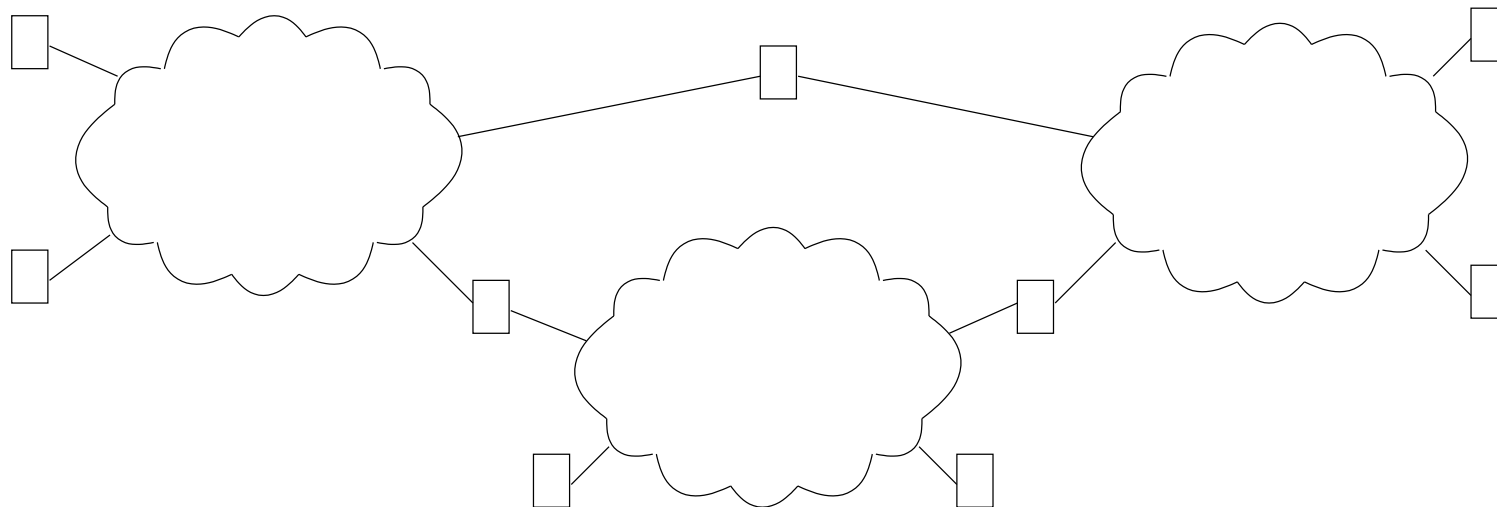
- Kommunikation soll möglich sein
- Verbindungen von Computer zu Computer (*direct link*)
  - *Punkt-zu-Punkt (point-to-point) (a)*
  - *Mehrfachanschluß (multiple access) (b)*



- Computer oder spezielle Hardware (z.B. Router) oft als *Knoten (node)* bezeichnet
- Netzwerke oft als *Wolken (clouds)* dargestellt, wenn keine Netzwerkdetails nötig
- Verbindungen von Knoten über andere Knoten sind indirekt (*indirect link*)
- Ein Knoten, der eine Verknüpfung durchführt wird häufig *Switch* genannt.



- Netzwerke werden miteinander über Verbindungsknoten verbunden.
- Ein Knoten, die mit mehr als einem Netzwerk verbunden ist, wird *Router* oder *Gateway* genannt.
- Die Aufgabe ist ähnlich der eines Switches innerhalb eines Netzwerks.
- Knoten, die ein Netzwerk nur benutzen, werden häufig *Host* genannt.



Netzwerke lassen sich nach ihrer Ausdehnung einteilen

- SAN: System Area Network / Storage Area Network
- LAN: Local Area Network
- MAN: Metropolitan Area Network
- WAN: Wide Area Network

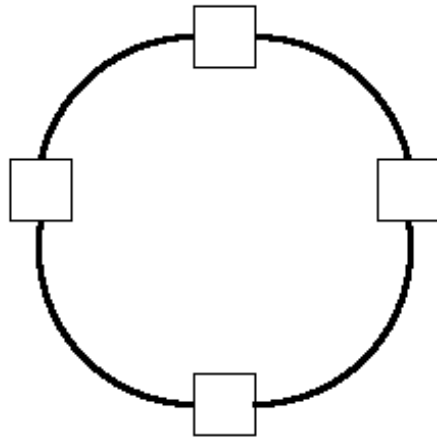
Interprocessor distance	Processors located in same	Example
0.1 m	Circuit board	Data flow machine
1 m	System	Multicomputer
10 m	Room	} Local area network
100 m	Building	
1 km	Campus	
10 km	City	Metropolitan area network
100 km	Country	} Wide area network
1,000 km	Continent	
10,000 km	Planet	The internet

Netzwerke können nach einer bestimmten Topologie aufgebaut sein

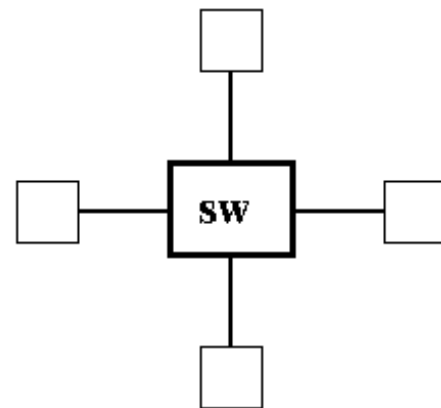
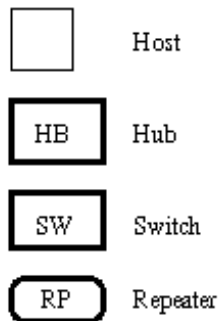
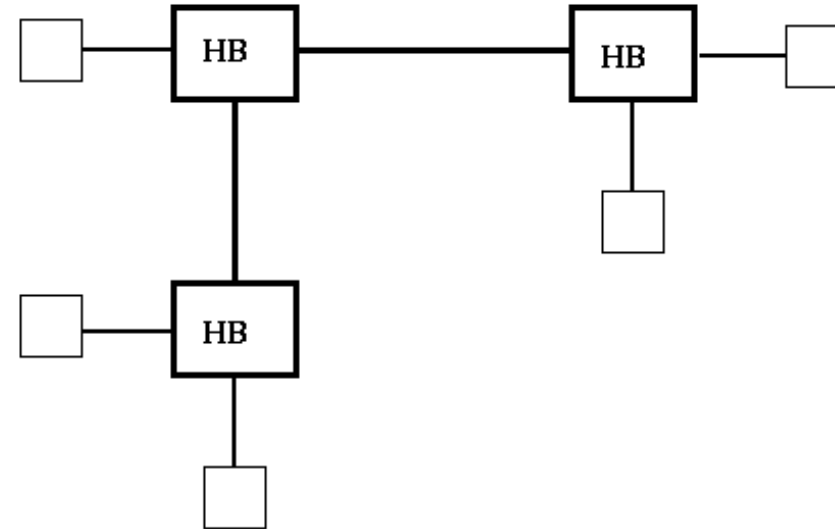
- Ring
- Stern
- Bus
- Baum

Auch Kombinationen davon sind möglich

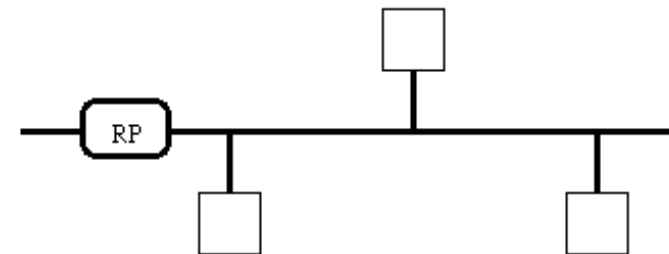
**Ring-Topologie**



**Baum-Topologie**



**Stern-Topologie**



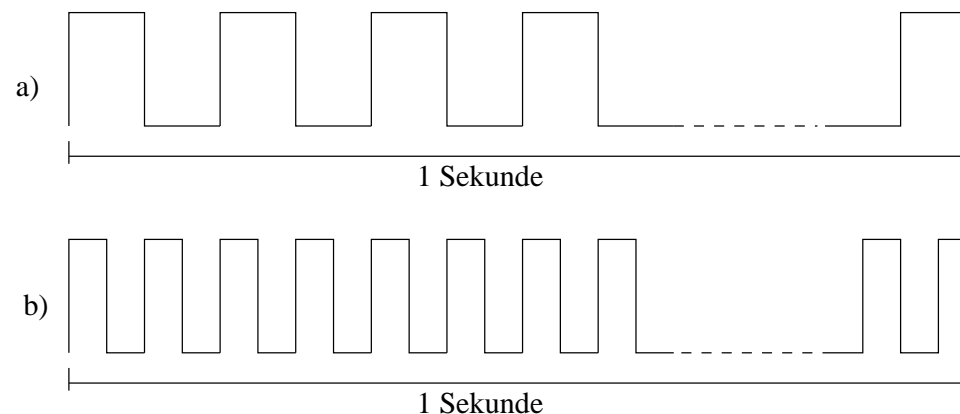
**Bus-Topologie**

- Begriff *Bandbreite (bandwidth)* mehrfach in Gebrauch
- Breite eines Frequenzbandes in *Hertz (Hz)*
- z.B. Telefonleitung: 300..3300Hz → 3000Hz Bandbreite
- Bandbreite einer Kommunikationsverbindung
- Anzahl von *Bits pro Sekunde (bits per second, bps)*
- z.B. 10Mbps für 10base2 Ethernet
- Hingegen bezeichnet *Durchsatz (throughput)* die *gemessene Leistung*, z.B. stehen einer Anwendung gerade 2Mbps zur Verfügung
- Eine Anwendung kann eine bestimmte *Bandbreitenanforderung* besitzen

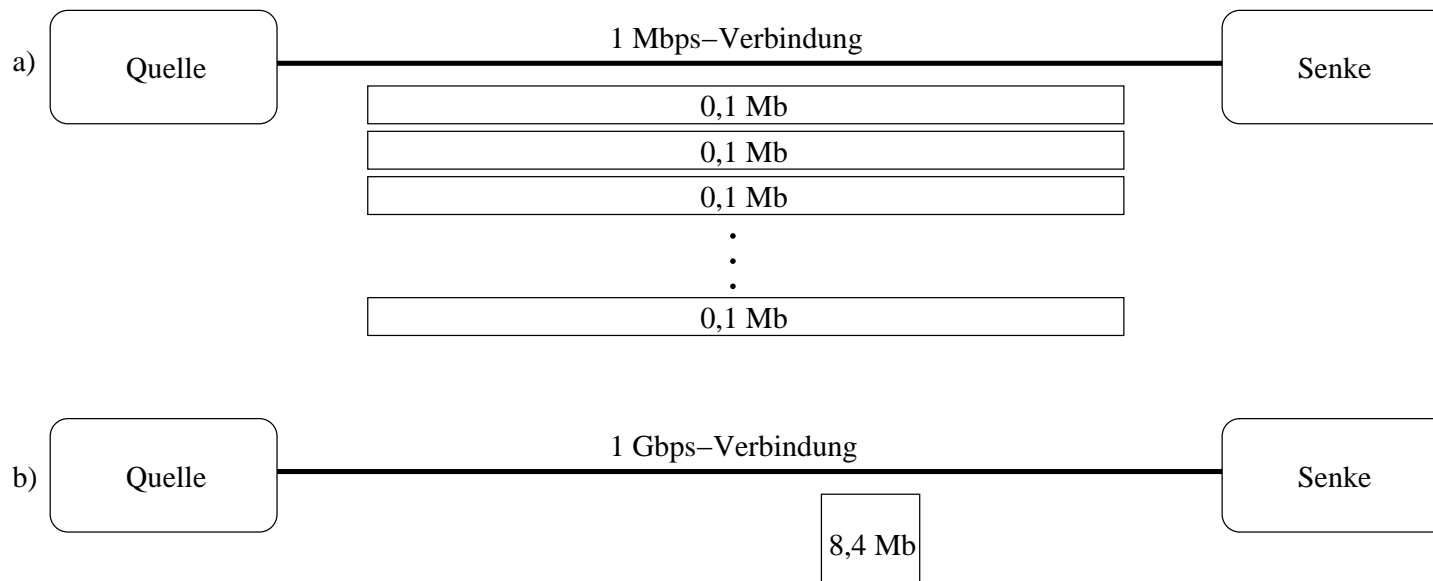


Leistungsmaß: *Bandbreite (bandwidth) / Durchsatz (throughput)*

- in Bits/Sekunde angegeben oder
- in (Mikro-)Sekunden (wie lange dauert es ein Bit zu transferieren)  
z.B. 10Mb/s  $\equiv$  0.1  $\mu$ s/b oder
- „Bitlänge“ in (Mikro-)Sekunden  
z.B. ist ein Bit bei (a) 1Mbps 1  $\mu$ s, bei (b) 2Mbps 0,5  $\mu$ s „lang“



- $\text{Durchsatz} = \text{Nachrichtengröße} / \text{Transferzeit}$
- $\text{Transferzeit} = \text{RTT} + 1/\text{Bandbreite} \times \text{Nachrichtengröße}$



Beziehung zwischen Bandbreite und Latenz: Eine 8.4Mb-Datei,  
(a) bei einer 1Mbps-Verbindung:  $84 \times 0.1\text{Mb}$   
(b) bei einer 1Gbps-Verbindung: nur 1/12 „gefüllt“

Leistungsmaß: *Latenz (latency) / Verzögerung (delay)*

- Wie lange benötigt eine Nachricht von einem zum anderen Ende?
- Gemessen in Zeit: (Milli-)Sekunden
- Auch üblich: Dauer, die eine Nachricht hin und ihre Antwort zurück benötigt
- die so genannte *Round-Trip-Time (RTT)*
- Formel:
  - Latenz = Ausbreitungszeit + Sendedauer + Wartezeit
  - Ausbreitungszeit = Entfernung / Lichtgeschwindigkeit im Medium
  - Sendedauer = Nachrichtengröße / Bandbreite

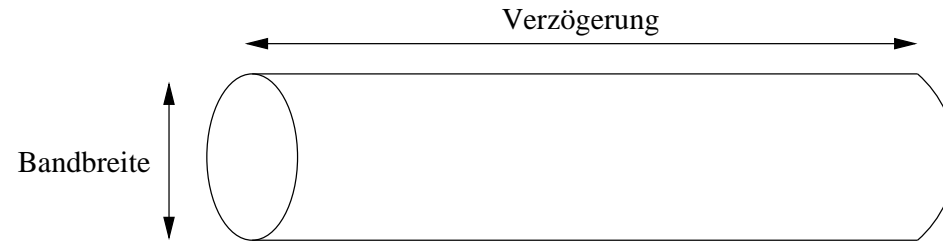
Die RTT läßt sich grob mit dem Programm „ping“ messen:

```
jh@sadewa:~> ping matrix
PING matrix.rvs.uni-bielefeld.de (129.70.123.40) 56(84) bytes of data.
64 bytes from matrix.rvs.uni-bielefeld.de (129.70.123.40): icmp_seq=1 ttl=255 time=2.90 ms
64 bytes from matrix.rvs.uni-bielefeld.de (129.70.123.40): icmp_seq=2 ttl=255 time=0.211 ms
64 bytes from matrix.rvs.uni-bielefeld.de (129.70.123.40): icmp_seq=3 ttl=255 time=0.201 ms
64 bytes from matrix.rvs.uni-bielefeld.de (129.70.123.40): icmp_seq=4 ttl=255 time=0.209 ms
64 bytes from matrix.rvs.uni-bielefeld.de (129.70.123.40): icmp_seq=5 ttl=255 time=0.207 ms
64 bytes from matrix.rvs.uni-bielefeld.de (129.70.123.40): icmp_seq=6 ttl=255 time=0.205 ms

--- matrix.rvs.uni-bielefeld.de ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5000ms
rtt min/avg/max/mdev = 0.201/0.656/2.903/1.004 ms
```

Für das Messen des Durchsatzes reicht es oft den Transfer einer Datei zu stoppen.

So wird der Overhead aller involvierten Protokolle mitgemessen.



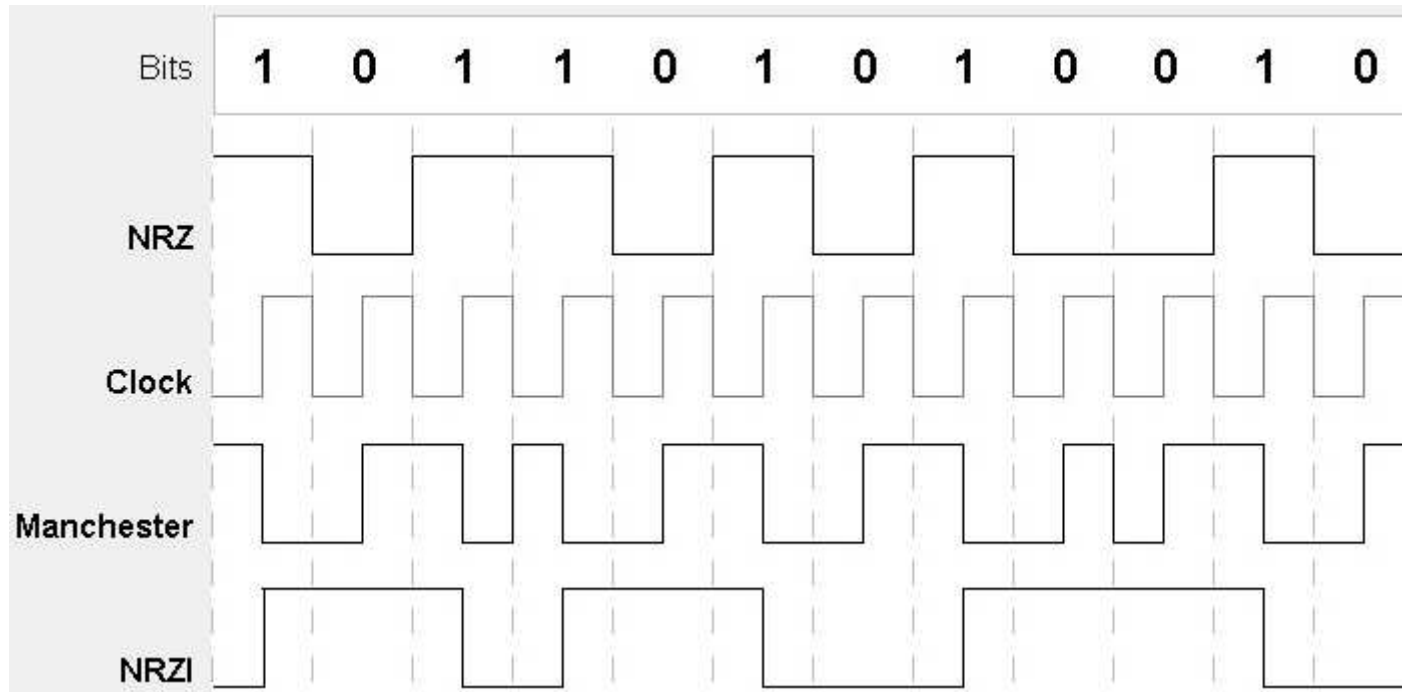
- Produkt aus Bandbreite und Verzögerung
- Ergibt Volumen einer fiktiven Netzwerkröhre
- Aussage darüber, wieviele Bits bereits abgesendet wurden, bevor das erste Bit ankommt
- Weiteres Maß: Instruktionen pro km (oder pro Meile)
- Wieviele Instruktionen könnte ein Computer während der Übertragungszeit durchführen

- Die binär vorliegenden Daten müssen für eine Übertragung in Signalzustände kodiert werden.
- Abstraktion: Es gebe zwei diskrete Signalzustände bzw. Signalpegel: High und Low.
- Einfachste Möglichkeit: korrespondierende Bit- und Signalzustände (1 → High, 0 → Low)
- Non-Return to Zero (NRZ)-Kodierung macht dies.
  - 1. Problem: „Grundlinienwanderung“ (*baseline-wander*)
  - 2. Problem: „Takterkennung“ (*clock recovery*)

- Non-Return to Zero Inverted (NRZI)
- 1 → Pegelwechsel
- 0 → kein Pegelwechsel
- Löst das Problem mit aufeinander folgenden Einsen
- Jedoch nicht das aufeinanderfolgender Nullen

- Manchester-Kodierung benutzt Verknüpfung mit Taktsignal
- Exklusives Oder von Taktsignal und NRZ-Kodierten Daten
- Problem: Verdopplung der Datenrate
- Signalwechselrate: *Baud-Rate*
- nicht verwechseln mit *Bit-Rate*
- Manchester: 50% Effizienz





- Versuch, effizienter als Manchester zu sein
- Jedoch sollen die damit gelösten Probleme gelöst bleiben
- Fügt Extra-Bits in den Datenstrom ein, um lang anhaltende Signalpegel zu vermeiden
- Alle 4 Bits werden durch 5 Bits kodiert
- Diese 5 Bits werden NZRI-kodiert verschickt
- Derart gestaltet, daß nie mehr als eine führende Null und zwei Endnullen

- Um Daten zu übertragen müssen neben den Daten auch Kontrollinformationen übertragen werden.
- Viele Netzwerke verwenden dazu keinen separaten Kanal
- Vielmehr werden Kontrolldaten und Daten gemischt
- Problem: Auseinanderhalten, wo Daten und wo Kontrolldaten anfangen bzw. enden