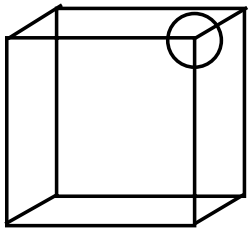
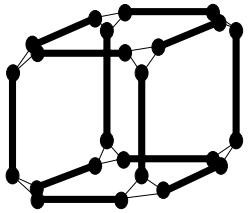
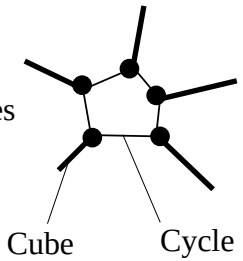
Cube Connected Cycles (CCC)

d-dimensionaler Hypercube mit aufgeschnittenen Ecken, die durch Gruppen von d ringförmig verbundenen Knoten ersetzt sind.





"Ecke" eines CCC (d=5)



Beachte: Jeder Knoten hat immer *drei* Anschlüsse!

Bei Dimension d: $n = d\ 2^d$

Maximale / mittlerer Weglänge? ← Denkübung! (wieso < 2d ?)

Anzahl der Verbindungen = 3 n / 2
(statt O(n log n) wie beim Hypercube)

Es gibt viele weitere Verbindungstopologien...
(z.B. de Brujn Graphen)

Zufallstopologien (mit max. Grad = 4)

Verfahren A:

1) Starte mit einem Graphen ohne Kanten.
2) Wähle zwei zufällige Knoten, die noch weniger als 4 Kanten haben, und verbinde diese.
3) Wiederhole Schritt 2 solange wie möglich.
4) Falls ein unzusammenhängender Graph entsteht, beginne von vorne.

Verfahren B ("greedy graphs"): ← Motivation: Kurze Zyklen vermeiden

2) Wähle zwei bel. Knoten *mit maximaler Entfernung* (einschliesslich ∝), die noch weniger als 4 Kanten haben, und verbinde diese.

Wie gut sind diese Zufallsgraphen bzgl. mittlerer Knotenentfernung und Routingbelastung?

Verzögerungszeiten, Routingoverhead

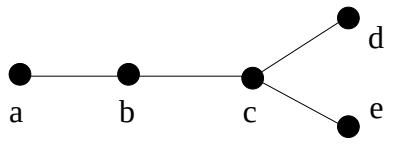
Bottleneck

Bem.: Explizites Routing nötig!

Routing-Belastung eines Knotens:
Zahl von Routen, die durch den Knoten gehen

Routing-Belastung einer Verbindung:
Zahl von Routen, die durch die Verbindung gehen

Beispiel:

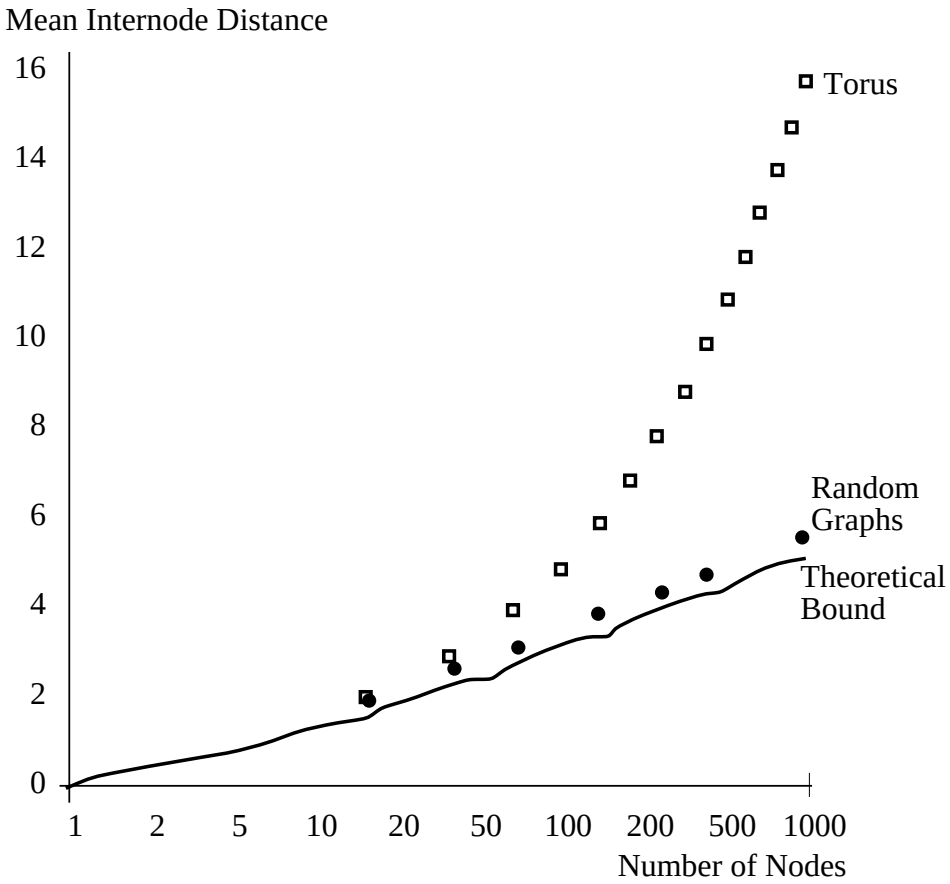


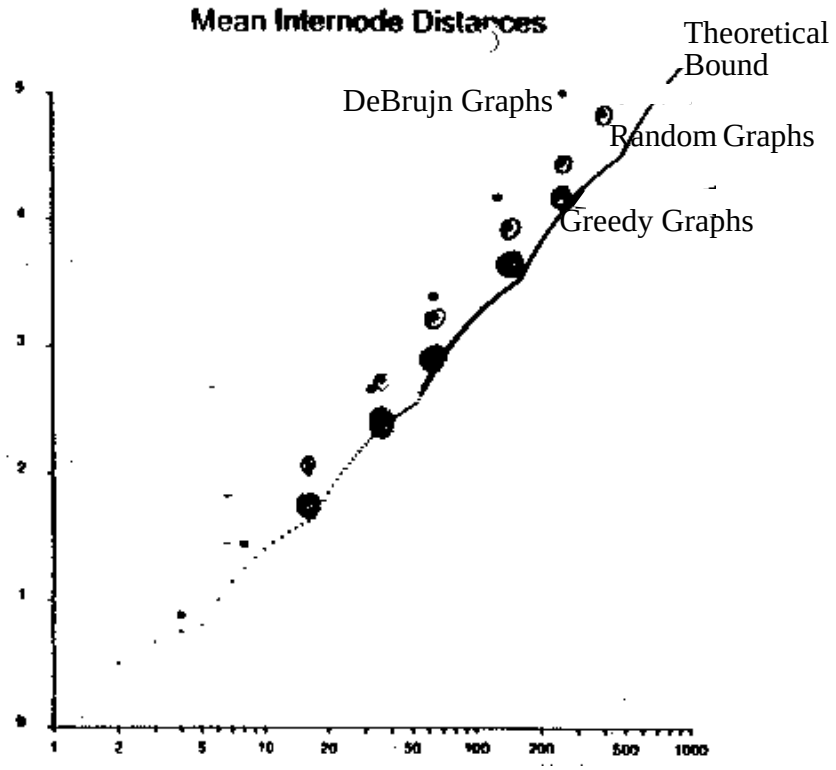
10 Routen von 20 gehen durch c

12 Routen von 20 gehen durch bc

Mittlerer Knotenabstand

- Für Knoten mit Grad 4
- Untersucht von D. Prior, ECSP (Edinburg)





- Ergebnis: Die Greedy-Strategie ist erwartungs-gemäss noch etwas besser.

Man lese zu den Untersuchungen zu Zufallstopologien folgenden Artikel:

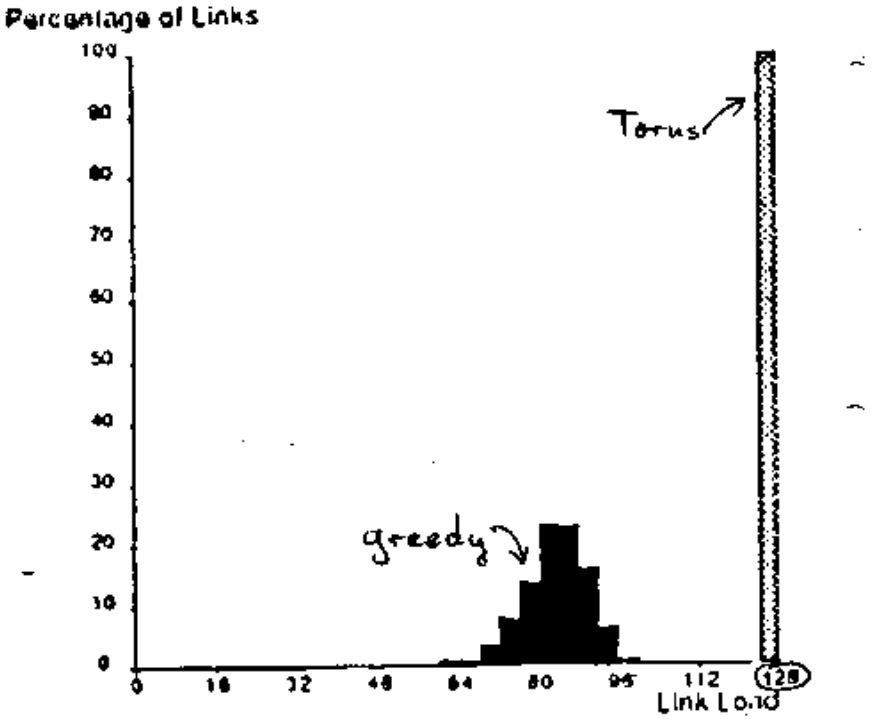
D. M. N. Prior, M. G. Norman, N. J. Radcliffe, L. J. Clarke: *What Price Regularity*?, Concurrency, Practice and Experience, Vol 2 No 1, pp. 55-78, 1990

Link loads for 100 processors

- Wenn jeder mit jedem anderen kommuniziert:
  ca. 100^2 Nachrichten (auf kürzesten Pfaden)

- 10×10-Torus: mittlere Entfernung = 5

  --> Jede Nachricht benutzt im Mittel 5 links.
  Bei 100^2 Nachrichten, 400 links --> 125 Nachrichten / link

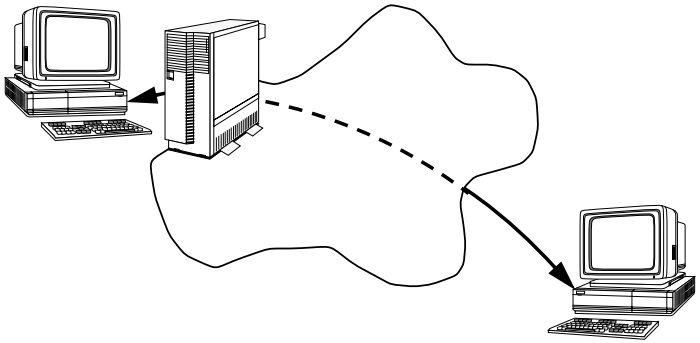


Rechnernetze

Rechnernetze

mehr dazu in speziellen Vorlesungen!

- Verbindung von autonomen Rechnern zum Zweck der Datenkommunikation



- Man unterscheidet:

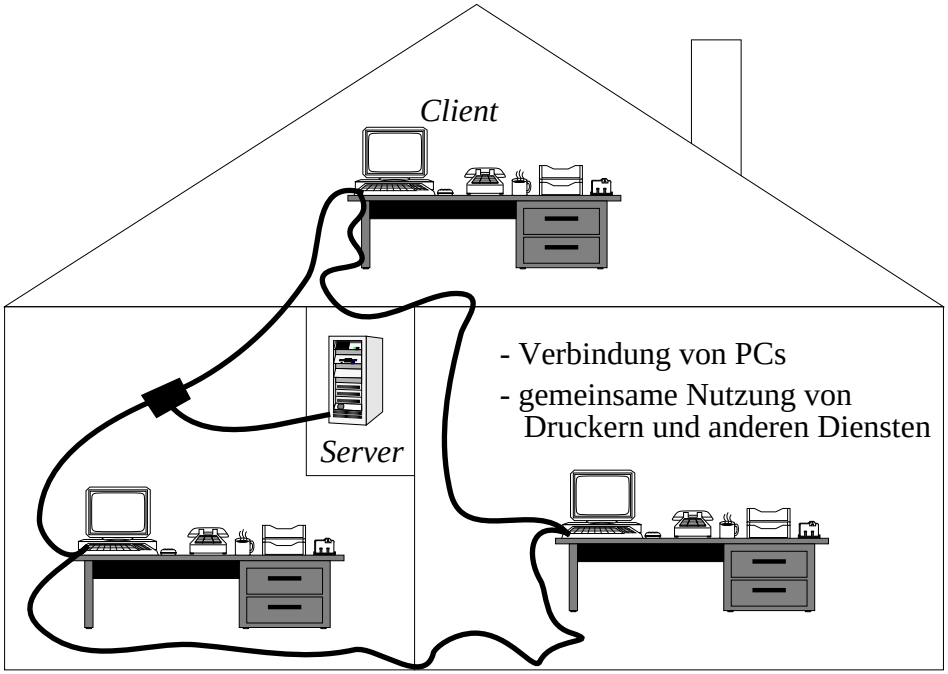
  - *Lokale Netze* (LAN, local area network):

    Rechner einer Abteilung oder einer Firma werden miteinander verbunden. Ausdehnung: einige 100 Meter.

  - *Weitverkehrsnetze* (WAN, long haul networks):

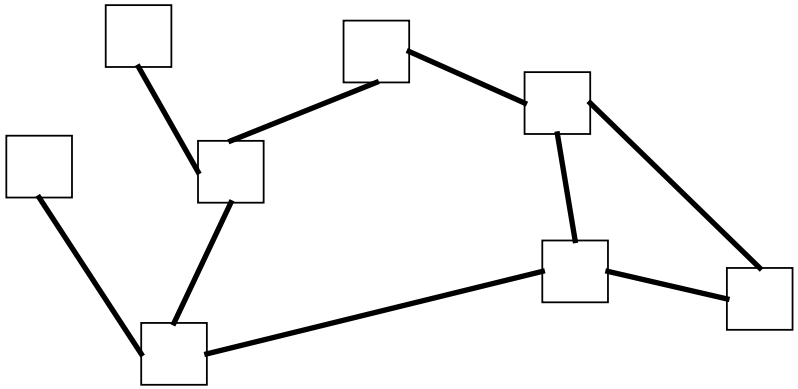
    Weltweite Verbindung über Kommunikationsnetze (spezielle Datennetze) bzw. private oder kommerzielle Satellitenverbindungen etc.

Lokale Rechnernetze



- Ausdehnung wenige 100 Meter.
- Hohe Übertragungsrate (typ. 10 - 1000 M Bit / s).
- Typischerweise innerhalb eines Gebäudes.
- Oft: Möglichkeit zu Broadcast.
- Zusammenfassung von mehreren “Subnetzen” zu einem “logisch” einheitlichen lokalen Netz.

Weitverkehrsnetze



- Verbindungen (z.B. einige 100 oder 1000 km) über Standleitungen, Mikrowellen, Satelliten...

- Insgesamt schwach “vermascht” (Kosten!)

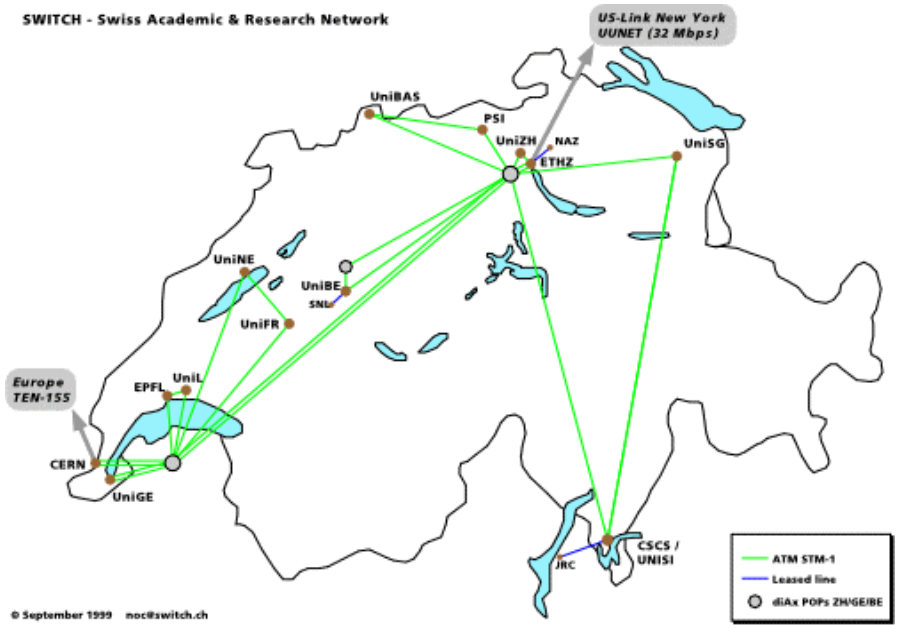
- Dezentral organisiert

- Nachrichten werden i.a. über andere Rechner geleitet (Routing; “store and forward”)

- Oft geringere Übertragungsrate als LANs
  - typisch: 50 k Bit/s (z.B. Modem) bis 2 M Bit/s (ggf. auch höher)

Beispiel: SWITCH (1999)

Swiss Academic & Research Network



http://www.switch.ch/lan/national.html

The new SWITCHlan consists of an ATM core network which is built on top of diAx’s national SDH infrastructure. STM-1 links are used to construct a double star ATM topology with hubs in Le Lignon (GE) and Altstetten (ZH). Every site on the ATM core is connected by two different STM-1 links which use physically diverse paths on the SDH rings. The local loops between the customer’s site and the nearest diAx PoP are realized as dark fibers.

Access from ETHZ/SWITCH to WorldCom’s Zurich PoP is realized through the City Ring Zurich. Two STM-1 links to Amsterdam and Frankfurt connect Switzerland to WorldCom’sATM Backbone.
On November 8, 1999, SWITCH has once more upgraded the bandwidth on the US link. From Zurich to New York, the new connection uses two circuits which follow different paths through Europe and across the Atlantic ocean (geograpical diversity). The combined bandwidth, equally spread over the two circuits, is 40Mbps ATM, corresponding to roughly 34Mbps on the IP level. The next upgrade to 2 x 28 Mbps ATM is scheduled for January 2000.

Beispiel: Neues Transatlantik-Kabel

A consortium of telecom operators on Wednesday signed a **$1.5 billion** project for a new fibre-optic cable link between Europe and the U.S. in order to cut waiting time on the World Wide Web.

More than 50 telecommunications operators signed contracts for the **TAT-14** cable which will have a capacity of **640 Gb/s** and could carry about 7.7 million simultaneous telephone calls.

Some **80 percent** of its capacity will be allocated to **Internet** and multimedia traffic.

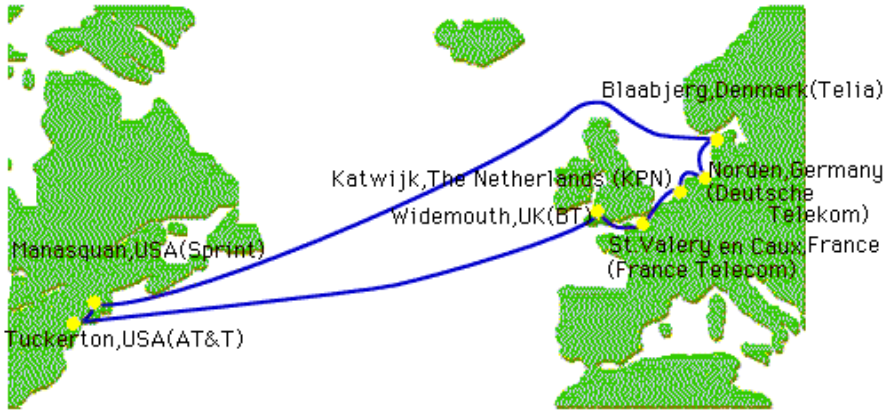
The TAT-14 cable network is a 1997 initiative of **11 carriers** - AT&T, BT, Cable & Wireless, Deutsche Telekom, France Telecom, KPN MCII, Pacific gateway Exchange, Sprint, Swisscom and Telia.

The TAT-14 cable will link five European countries - Germany, England, Denmark, France and the Netherlands - with the United States.

It will span 15,000 kilometres and is expected to be completed and in service by the **end of 2000**.

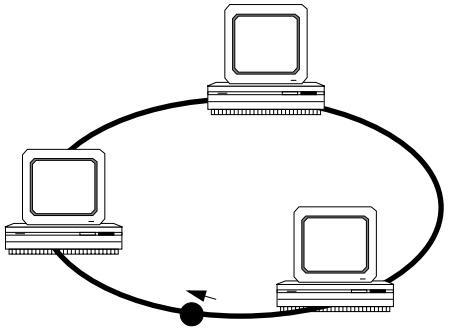
The new system, which is a ring network, will utilise the latest advances in Synchronous Digital Hierarchy (SDH), Wavelength Division Multiplexing (WDM; 16 wavelengths of 9.953 Gb/s) technology and will consist of **four pairs of optical fibre cable**.

The new system will represent **64 times the capacity of the original TAT-12/TAT-13** cable network which was put into service in September 1996.



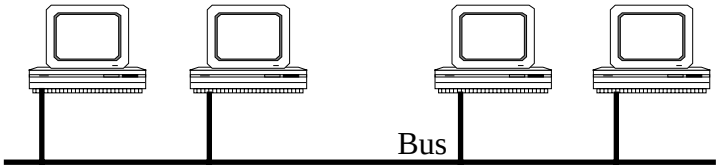
Topologien lokaler Rechnernetze

- *Token-Ring*



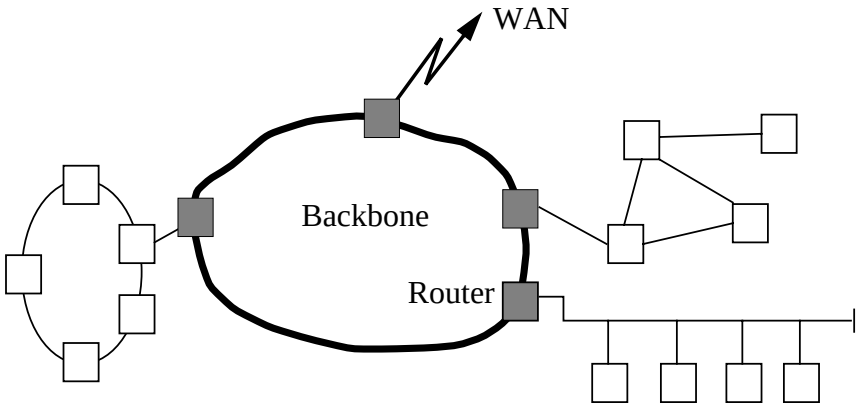
- Nachricht wird solange weitergereicht, bis die Zieladresse mit der angegebenen übereinstimmt.
- Es kreist ein sogenanntes “Token” (spezielles Paket).
- Ein Rechner darf nur dann senden, wenn er das Token besitzt.

- *Bus-Topologie* (z.B. *Ethernet*)



- Jeder Rechner “horcht” am Bus und empfängt die Nachrichten, die seine Adresse tragen.
- Senden, wenn Bus frei.
- Kollisionen von mehreren Sendern sind ein Problem.
- CSMA/CD (Carrier Sense Multiple Access / Collision Detection)

Backbone-Netze



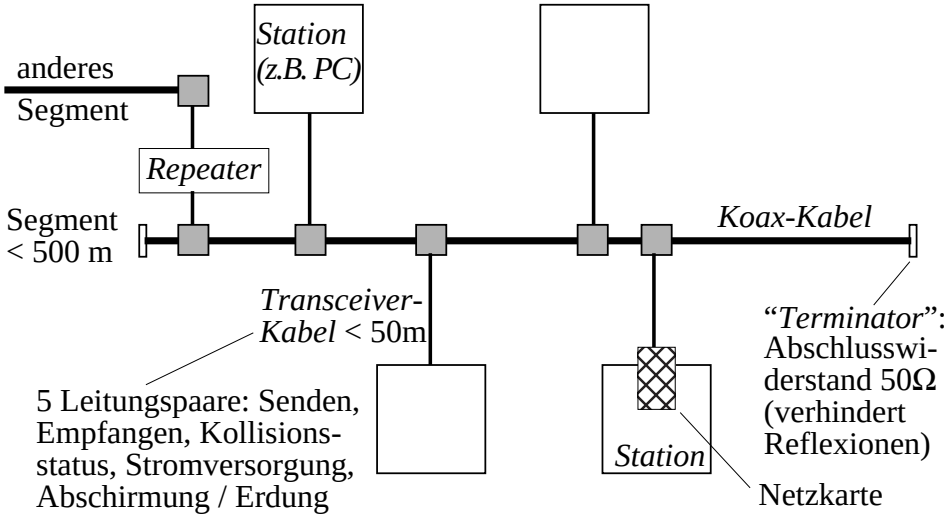
- Dient (nur) dazu, andere Netze (LANs) zu verbinden
- Leistungsfähiger als normale LANs
  - z.B. 1 Gb/s Ethernet oder ATM-Verbindung bei 10 Mb/s in den angeschlossenen LANs
- Räumliche Ausdehnung typischerweise Campus, Firma, grosses Gebäude...

Ethernet

"Äther"
--> gemeinsames Medium
--> broadcast der Nachrichten

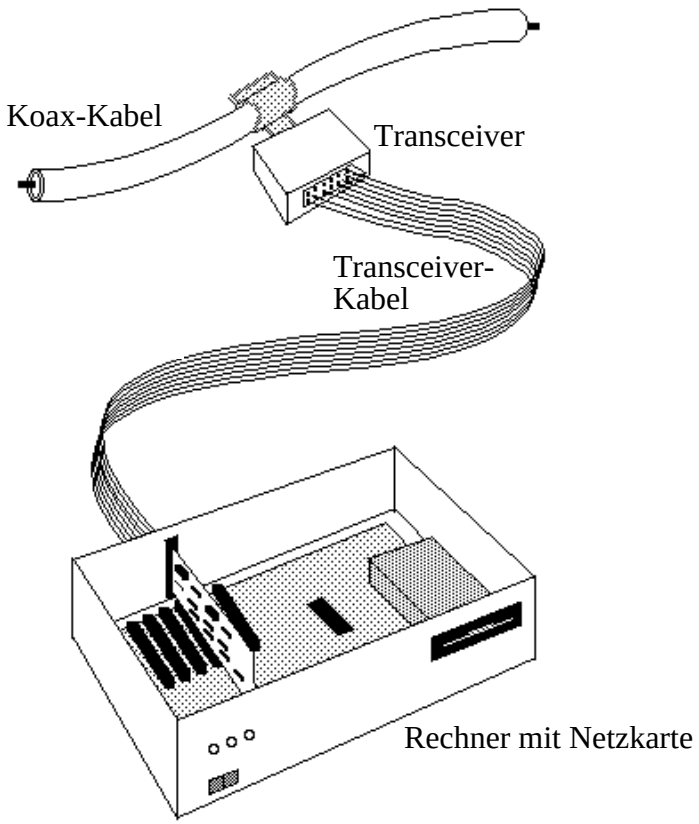
- Bus mit dezentraler Zugangskontrolle
- Industriestandard seit 1980
- Genormt durch IEEE 802.3
- Relativ unempfindlich gegenüber Ausfall einer Station
- Preiswert, einfach ausbaufähig, wenig Administration

Klassische Konfiguration:



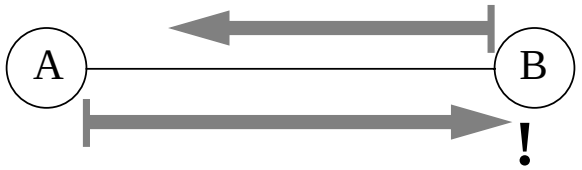
- Verbinden zweier Segmente durch *Repeater* (bidirektionale Signalverstärker) möglich
- Gesamtausdehnung max. ca. 2500 m

Klassisches Ethernet: Anschluss einer Station

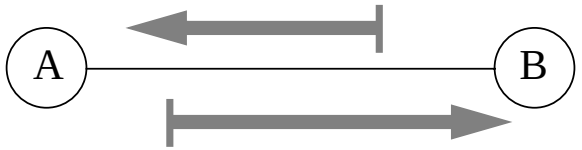


Kollisionen

A und B fangen “fast” gleichzeitig an zu senden.



Etwas später merkt B, dass A auch sendet.

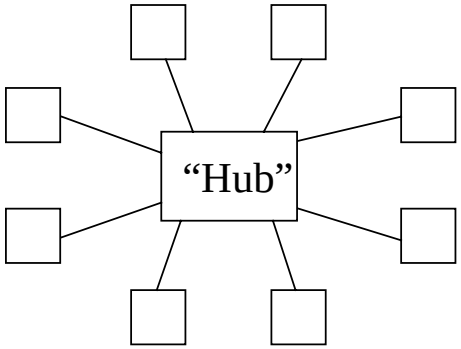


Das darf nicht auftreten: Keine Station merkt (während sie sendet!), dass eine andere ebenfalls sendet.

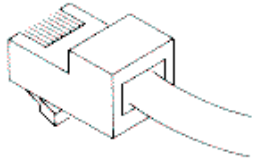
==> - Maximalentfernung zweier Stationen
- Mindestlänge eines Nachrichtenpakets
(--> Mindestdauer eines Sendevorgangs)

Twisted-Pair-Ethernet

- “10 Base T”;
  i.a. sternförmig



RJ-45-Anschluss:



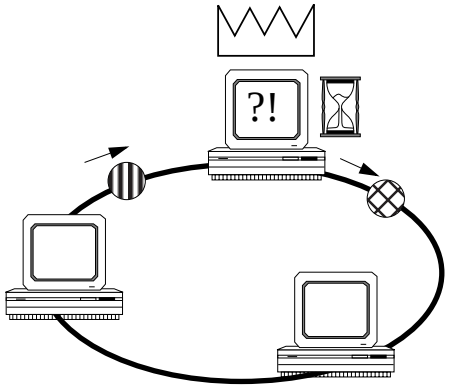
- Einfacher Hub: “Collabsed Backbone” in einem Gerät
  - generiert Kollisionssignal, wenn mehrere Stationen gleichzeitig senden

- Kabelstörungen betreffen i.a. nur eine Station
  - Hub entfernt im Fehlerfall betroffene Station logisch

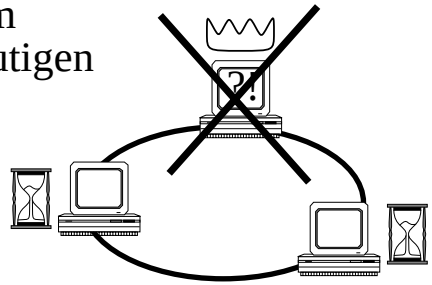
- “Switch” statt “Hub”
  - stellt jedem Endgerät die volle Bandbreite zur Verfügung
  - schaltet dedizierte Verbindungen zwischen je zwei Ports
  - simuliert auf jeder Leitung eigenes Ethernet (--> weniger Kollisionen)
  - bei einem belegten Zielport muss ein Paket ggf. (kurz!) gepuffert werden, oder es wird dem Sender ein Kollisionssignal geschickt

Token-Ring: der Monitor

- Eine Station übernimmt die Rolle des *Monitors*.
  --> Erkennung und Behebung von Fehlersituationen
  - Timeout, um verlorene Token zu erkennen (--> regenerieren)
  - Stempelt Nachrichten, um mehrfaches Zirkulieren zu erkennen.
  - Absorbiert alte Token und fehlerhafte Nachrichtenreste.



- *Monitorausfall* wird von anderen Stationen durch einen “grossen” Timeout (kein Token gesehen) erkannt.
  --> Election-Protokoll, um *dezentral* einen eindeutigen Monitor zu wählen.

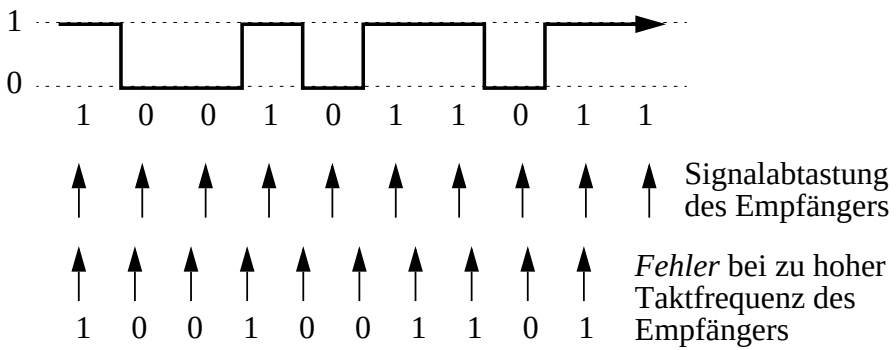


Election-Protokoll wird auch benötigt, damit die Stationen nach “erstmaliger” Konfiguration einen Monitor auswählen.

Taktsynchronisation; Datenpakete

-“Gleichlauf” von Sender und Empfänger notwendig

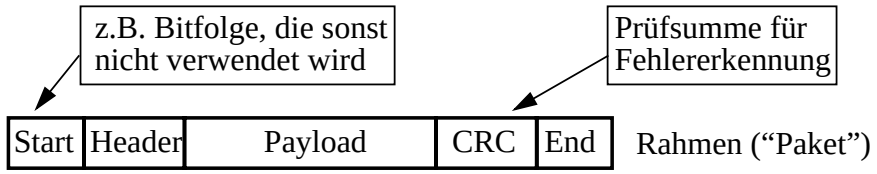
- *Taktsynchronisation* notwendig zur korrekten Interpretation übertragener Signale (Zeitpunkt zum Abtasten des Signals)



- hierzu verschiedene Techniken: Taktsignal über getrennte Leitung oder Regenerierung des Taktes aus dem Übertragungssignal (dann genügend viele Wertänderungen im Signalstrom notwendig!) und punktuelle Resynchronisation eines unabhängigen Taktgenerators

- Typischerweise werden mehrere Zeichen zu einem Datenpaket verbunden und als Einheit übertragen

- als Einheit zur Fehlerprüfung, Quittierung, Wiederholung etc.



- Länge von Datenpaketen z.B. 53 Byte bei ATM, 64-1500 Byte bei Ethernet, max. 64 kByte bei IP

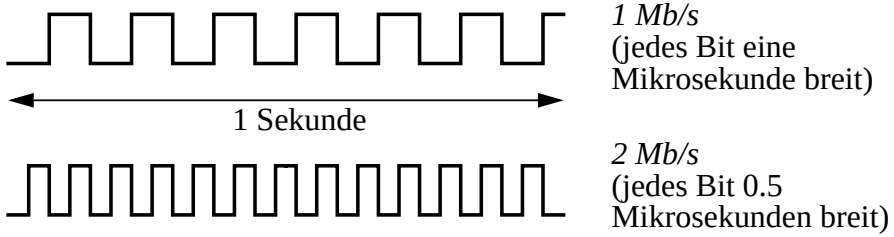
- Header enthält typischerweise die Länge des Pakets, Adressinformation und Steuerinformation (Flags etc.)

Bandbreite und Delay

- *Bandbreite*: Übertragene Datenmenge pro Zeiteinheit

- Beispiel Ethernet: 10 Mb/s (“klassisch”) bzw. 100 Mb/s (“fast”)

- Bandbreite veranschaulicht durch “Bitbreite”:

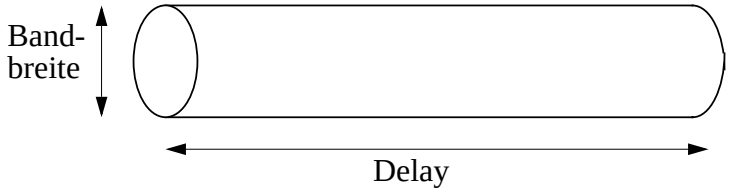


- *Delay*: Zeit, um eine Nachricht von A nach B zu senden

- Für kurze Nachrichten (“Null-Message”) ist Delay entscheidend; Bandbreite ist vernachlässigbar.

- Für grossvolumige Daten (z.B. Bilder) ist die Bandbreite der dominierende Faktor.

- Bandbreite-Delay-Produkt oft interessant:



- Beispiel: 100 ms Delay und 45 Mb/s Bandbreite
  --> 562 KB Daten, die unterwegs sind, bevor erstes Bit ankommt

- Speichervermögen eines Kommunikationskanals

Hochgeschwindigkeitsnetze

ca. 100 Mbit/s - 1 Gbit/s

- “Fast Ethernet” (100 Mb/s) in LANs
  - mit Twisted-pair-Kabel und i.a. über Switch (sternförmig)
  - Switch hat i.a. Ports, die sowohl 10 Mb/s als auch 100 Mb/s können

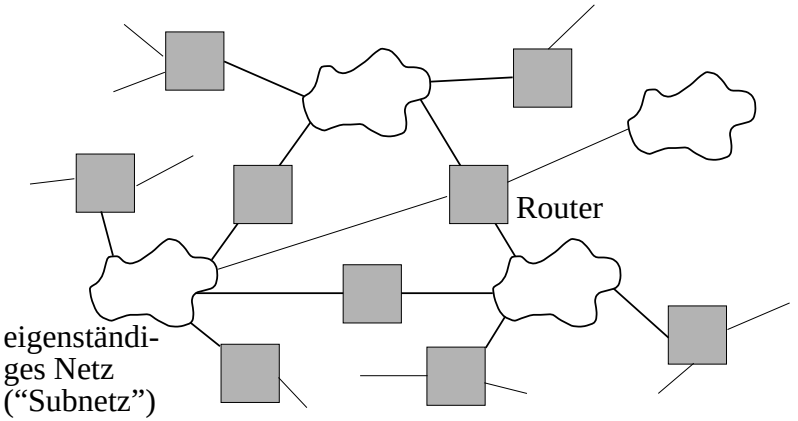
- “Gigabit-Ethernet” (1 000 000 000 b/s) in LANs
  - über Switch (sternförmig) und typw. Glasfaser
  - z.Z. noch wenig verbreitet (neu, teuer)

- ATM (Asynchronous Transfer Mode)
  - für WAN und LAN (in LANs aber teurer als Ethernet)
  - Grundlage für Breitband-ISDN
  - Pakete (“Zellen”) von konstant 48 Byte mit 5 Byte Header
  - kleine Zellen: Zelle ist schnell voll und kann gleich verschickt werden
  - typische Bandbreiten: 155 Mb/s; ca. 600 Mb/s
  - für Audio, Video und Realzeitdaten gut geeignet (“Dienstgüte”)

- Engpass ist oft nicht mehr die “Leitung”, sondern die Geschwindigkeit der Kommunikationssoftware

- “Hochgeschwindigkeit” flächendeckend in die Haushalte?
  - Kabel-TV-Infrastruktur aufrüsten (“Kabelmodem” beim Kunden und Anbindung des “Headends” über Glasfaser an die Internet-Infrastruktur)
  - Telefon-Infrastruktur aufrüsten (z.B.: xDSL mit ca. 1-8 Mb/s)
  - Glasfaser in die Haushalte (teuer)
  - noch Science Fiction: tieffliegende Satelliten, spezielle Flugzeuge etc.

Inter-Netzwerke



- Koppelung von LANs, Intranets, Campusnetze, Firmennetze etc. über “Gateways”
  - ggf. mit Protokollumsetzung oder -anpassung

- “Internet”: Verwendung einheitlicher Protokolle
  - basierend auf IP = Internet Protocol
  - Abschottung lokaler Netze durch Firewalls

- Routing (“Leitwegbestimmung”)
  - Ermittlung von Routing-Tabellen
  - Weiterleitung einer Nachricht mittels Routing-Tabellen (“Forwarding”)
  - es gibt diverse Routing-Algorithmen (oft: dezentral)

Kommunikationsprotokolle

- Kommunikation in verteilten Systemen geschieht ausschliesslich über *Nachrichten*.

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

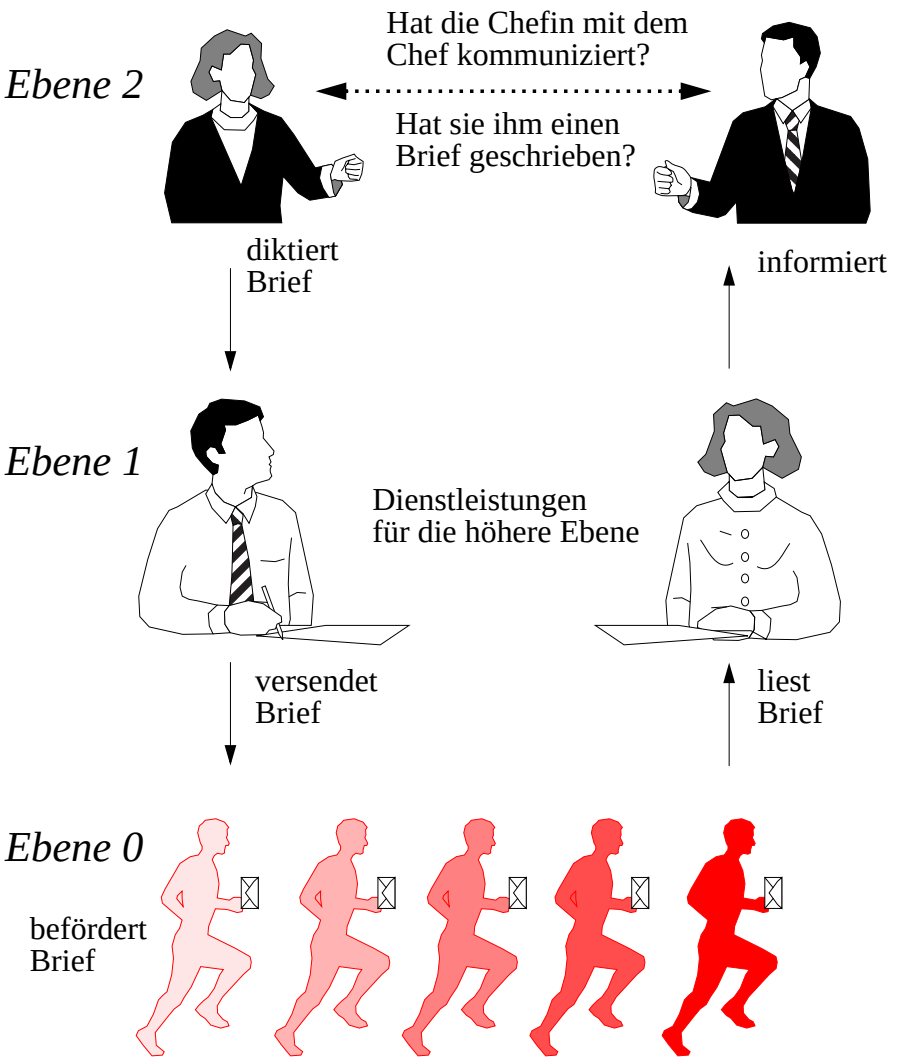
- *Protokoll* = Festlegung der Regeln und des algorithmischen Ablaufs bei der Kommunikation zwischen 2 oder mehr Partnern.

- 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”) und 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

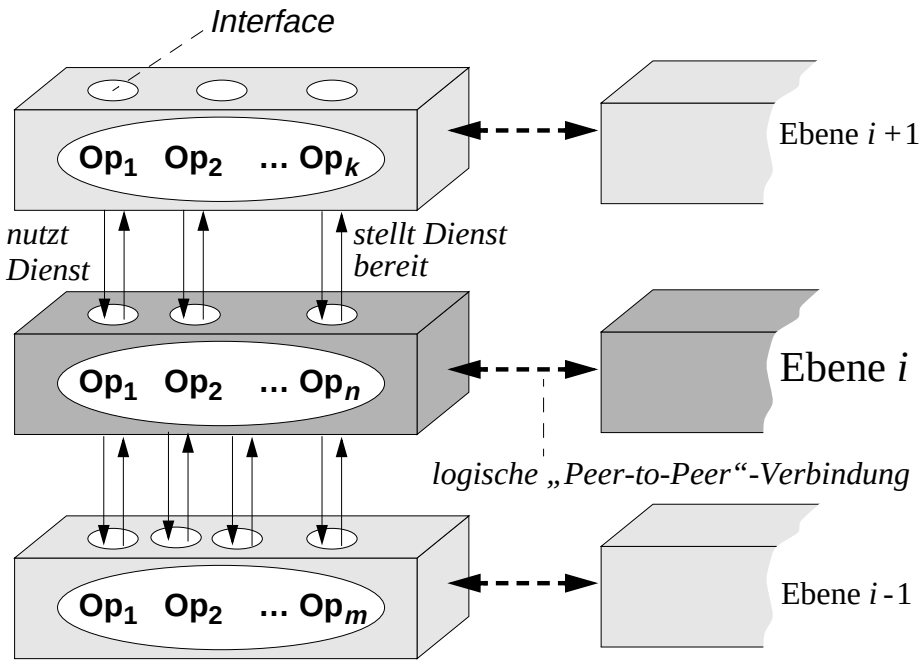
- Ebene 0 kann ausgetauscht werden (z.B. Fax statt gelbe Post), ohne dass sich auf höherer Ebene etwas ändern muss (--> Transparenz)

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

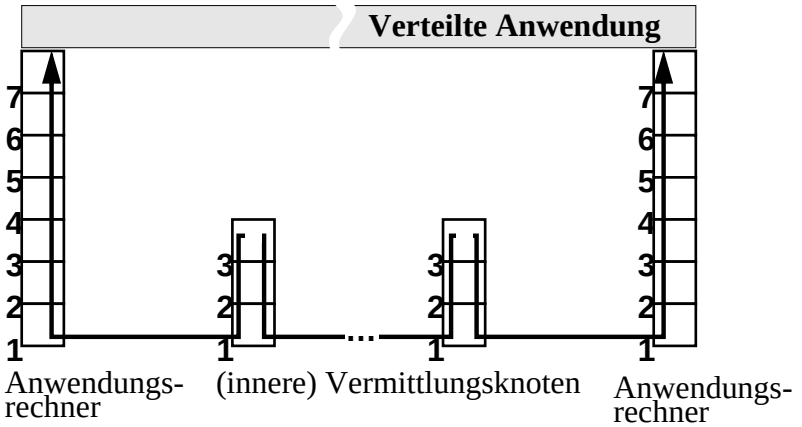
Vgl. auch Speditionsfirma, die verscheidene Transportdienste in Anspruch nehmen kann:

- Auf jeder Ebene gelten jeweils eigene Regeln (StVO...: Rechtsverkehr...), die aber nicht nach oben durchschlagen
- Leistungsfähigkeit eines Dienstes (Kosten, Geschwindigkeit, Güte...) sind allerdings spürbar
- Verschiedene Transportdienste bieten i.w. den gleichen Service --> sind austauschbar

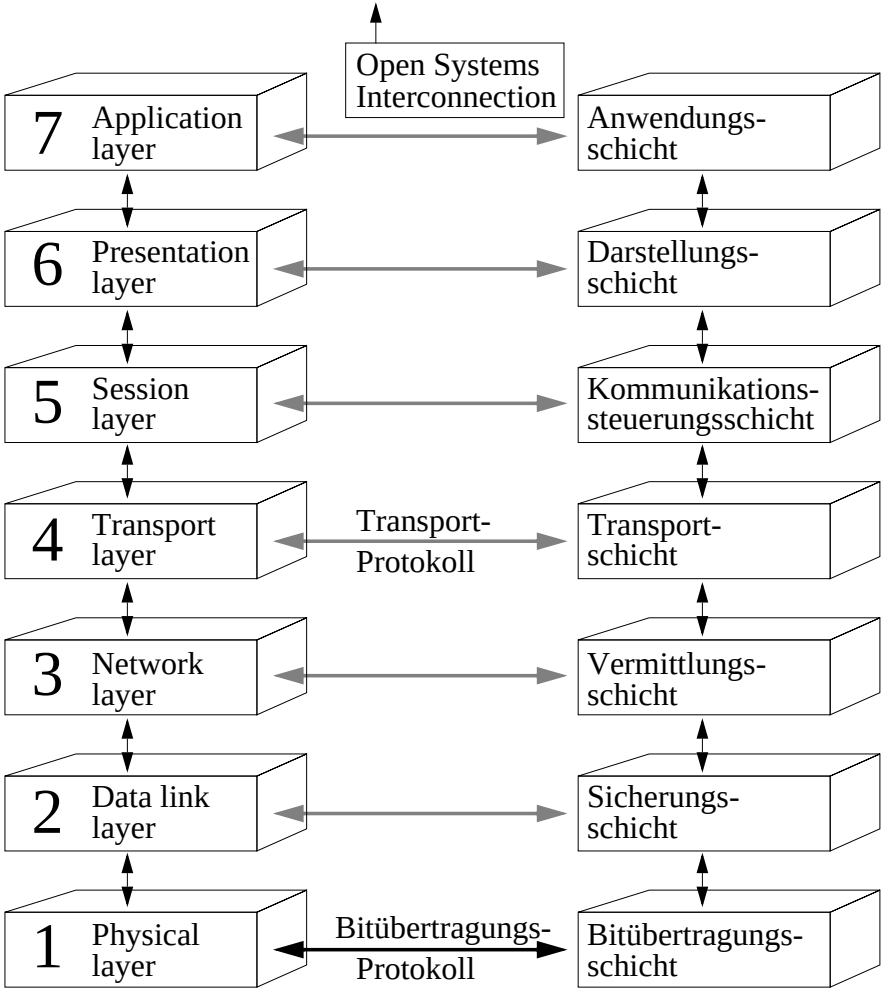
Prinzip der Protokollschichtung



- Transport von Nachrichten über Vermittlungsknoten:



Das ISO-OSI-Referenzmodell



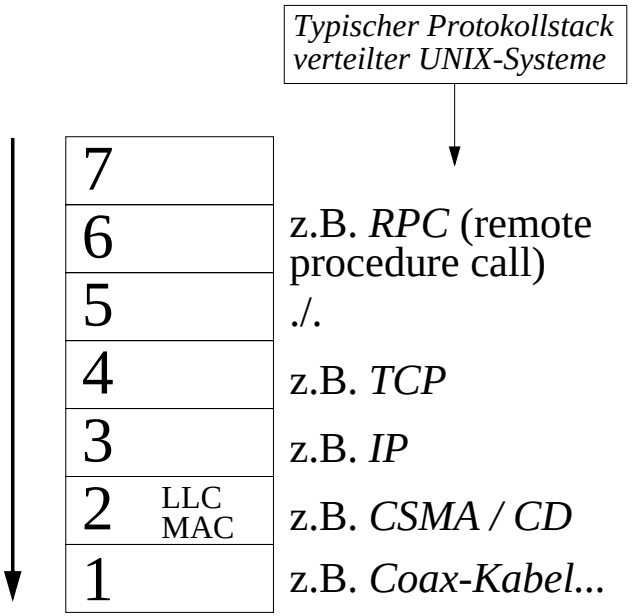
- Beachte: Dies ist ein “Referenzmodell”!
  - Strukturelle Basis (“Architekturmodell”) für Protokolle und deren Normierung (einheitliches Vokabular; Orientierung etc.)
- Wieso gerade 7 Schichten?
  - In einigen Protokollen sind einige Schichten fast “leer”
  - In einigen Protokollen werden einige Schichten nochmals unterteilt

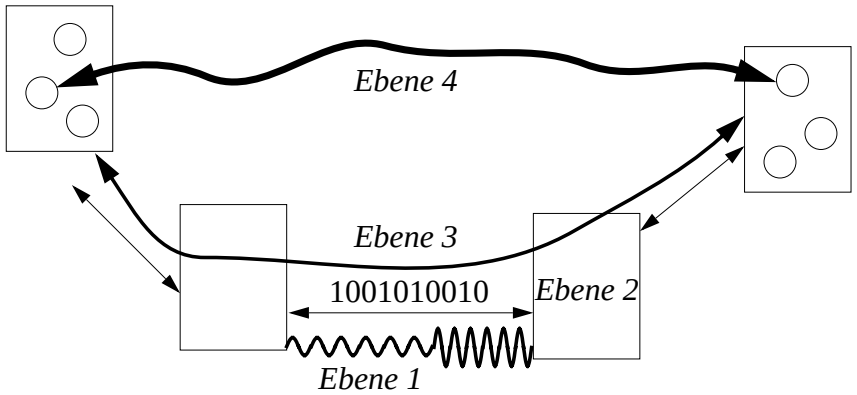
Wichtigste Aufgaben der 7 Schichten

1. *Physical Layer*
   - Normung von Steckern und Kabeleigenschaften
   - Kodierung von Bitfolgen mit physikalischen Signalen
2. *Data Link Layer*
   - Flusssteuerung (“flow control”): Sender ggf. bremsen
   - Fehlerfreie Übertragung von Bit Frames zwischen zwei Nachbarknoten
3. *Network Layer*
   - Übertragung von Paketen zwischen beliebigen Endknoten
   - Routing
   - Multiplexen von Verbindungen
4. *Transport Layer*
   - Logische Verbindung zwischen (adressierten!) Prozessen (bzw. ports, sockets...) statt Rechnern
   - Bitströme bzw. Nachrichten beliebiger Länge werden in Pakete aufgeteilt
5. *Session Layer*
   - Sitzungsverwaltung über Phasen (z.B. Login / Logout) hinweg
   - Kopplung mehrerer Transportverbindungen (z.B. Audio + Video) zu einer Sitzung
6. *Presentation Layer*
   - Kodierung komplexer Daten (Typ, Wertebereich, Struktur...)
7. *Application Layer*
   - z.B. WWW (http-Protokoll) oder ftp

Protokoll-Stack

- Menge der in einem gegebenen Fall verwendeten spezifischen Protokolle





Die Internet-Protokollhierarchie

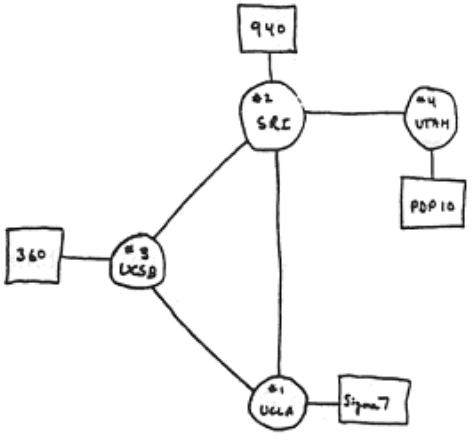
	Application Layer	Gegenüber OSI sind die drei obersten Schichten zusammengefasst
4	Transport Layer	TCP (bzw. UDP)
3	Internet Layer	Verfahren, um mehrere verschiedene Netze zu traversieren; Wegewahl; IP
		In LANs Unterteilung in zwei Teilschichten:
2	Network Access Layer: Logical Link Control (LLC)	Fehlerfreie Übertragung;
	Network Access Layer: Medium Access Control (MAC)	Zugriff auf das Medium
1	Physical Layer (PHY)	Bitübertragung

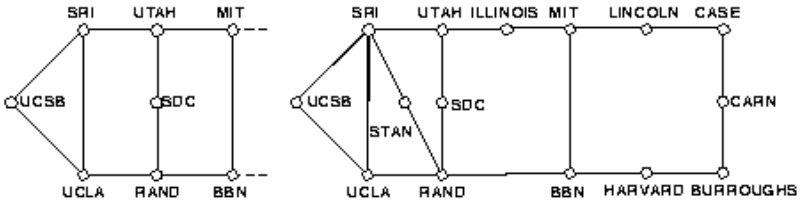
- Im Internet gehört alles oberhalb der TCP-Ebene zur “Anwendungsebene”
  - Dienstprogramme wie dns oder ftp
  - Echte verteilte Anwendungen
  - “Middleware”, die eine eigene Infrastruktur für grosse verteilte Anwendungssysteme bereitstellt

Internet-Geschichte: Pionierzeit

- 1967: ARPA (Advanced Research Project Agency) des DoD vergibt Auftrag “Projektstudie ausfalltolerantes Paketnetz” an SRI (Stanford Research Institute)

- 1969: Erstes “Internet” aus 4 Knoten: University of California at Los Angeles (UCLA), Stanford Research Institute (SRI), University of California at Santa Barbara (UCSB), University of Utah





Juli 1970 März 1971

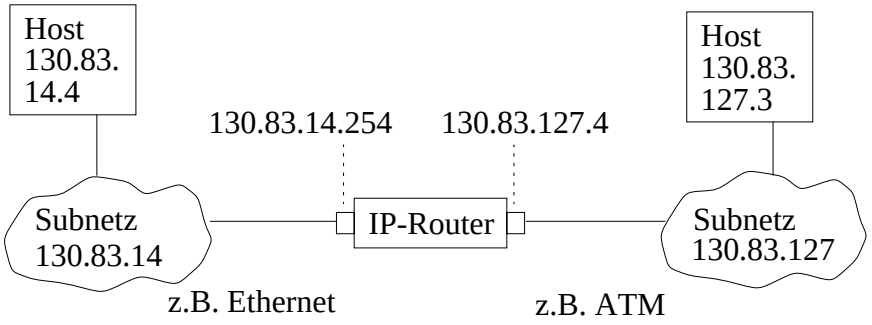
- 1971: Betriebsaufnahme des ARPAnet; Experiment zum Einloggen auf entfernten Rechnern; erstmalig Nutzung von E-Mail
- 1972: Erste öffentliche Demonstration des Netzes
- 1974: Neue Protokollsuite: TCP/IP
- 1980: Integration der TCP/TP-Protokolle in UNIX

Das Internet

- Geschichte
  - Pionierzeit (USA): 70er Jahre des 20. Jahrhunderts
  - Wissenschaftsnetz (getrieben durch die Informatik): 80er Jahre
  - Kommerzialisierung und Popularisierung: 90er Jahre

- Technologie
  - globales Addressierungsschema (“IP-Adressen”)
  - TCP/IP-Protokolle + Routing-Verfahren + Hilfsdienste
  - Anwendungsprotokolle: E-mail, ftp, news, telnet, http,...

- Jede Maschine am Internet hat eine IP-Adresse
  - 32 Bit lang (zukünftige IPv6-Version: 128 Bit)
  - typischerweise als 4 Dezimalzahlen geschrieben
  - Bsp.: 192.130.10.121 (= 11000000.10000010.00001010.01111001)

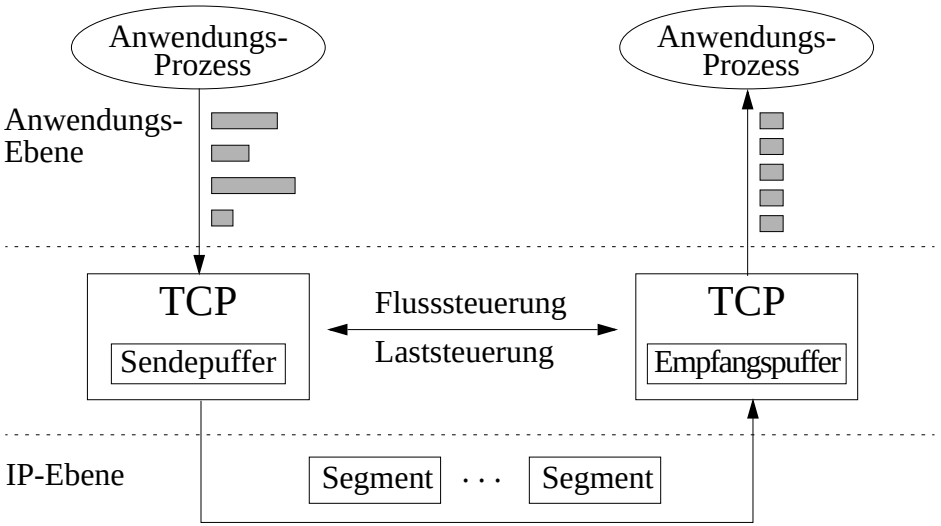


- Adressen werden in einen Netz- und Host-Teil aufgespalten
  - z.B. Class B: 16382 Netze mit jeweils bis zu 65534 Rechnern

TCP (Transmission Control Protocol)

- TCP bietet einen zuverlässigen Kanal für einen Bytestrom
  - zwischen zwei Ports (=Kommunikationsendpunkte für Programme)
  - in beide Richtungen (“voll duplex”)
  - unstrukturiert (nur Bytes; Nachrichtengrenzen gehen verloren)

- Verbindungsaufbau vor Datenübertragung notwendig
  - Verbindungsabbau danach



- Ein spezielles Protokoll (“sliding window”) sorgt für eine gegen Übertragungsfehler gesicherte Kommunikation sowie die Fluss- und Laststeuerung
  - *Flusssteuerung*: Sender soll *Empfänger* nicht überlasten
  - *Laststeuerung* : Sender soll das *Netz* nicht verstopfen

Kommunikation

Kommunikation

- Prozesse sollen kooperieren, daher untereinander Information austauschen können - über
  - *gemeinsame Daten* in einem globalen Speicher (dieser kann physisch oder ggf. nur logisch vorhanden sein: “virtual shared memory”)
  - oder *Nachrichtenaustausch*: Kopie der Daten an eine entfernte Stelle

- Notwendig, damit die Kommunikation klappt ist jedenfalls:

  1) dazwischenliegendes *physikalisches Medium*
     - z.B. elektrische Signale in Kupferkabeln

  2) einheitliche *Verhaltensregeln*
     - Kommunikationsprotokolle

  3) gemeinsame *Sprache* und gemeinsame *Semantik*
     - gleiches Verständnis der Bedeutung von Kommunikationskonstrukten und -Regeln

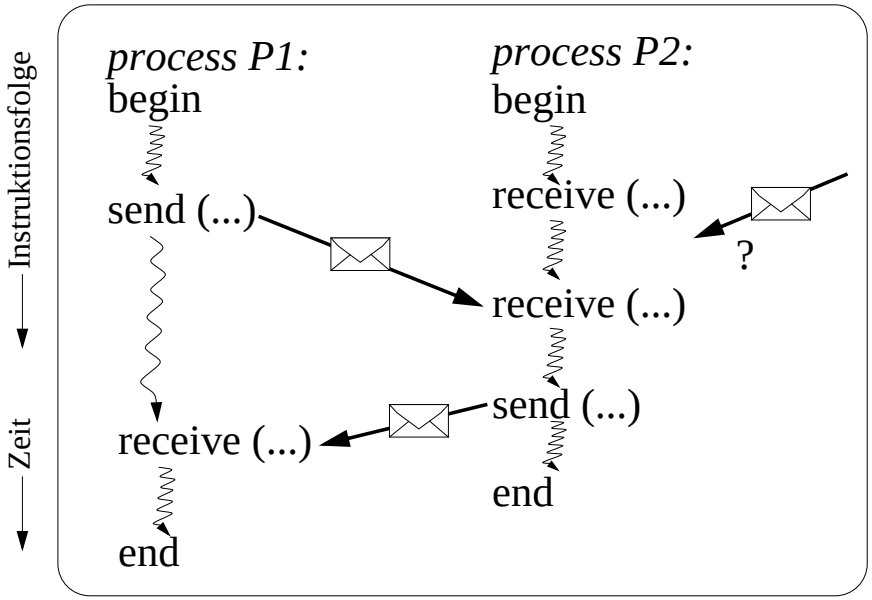
Also trotz Verteiltheit gewisse gemeinsame “Dinge”!

- Wir betrachten im folgenden den dritten Punkt genauer
  - Punkte 1) und 2) sind eher Themen einer Vorlesung über “Rechnernetze”

Sprachkonstrukte zur Kommunikation und deren Wirkung

Nachrichtenbasierte Kommunikation

- “Austausch” von Nachrichten: send --> receive
- Implizite Synchronisation: Senden *vor* Empfangen
  Empfänger erfährt, wie weit der Sender mindestens ist
- Nachrichten sind dynamische Betriebsmittel: gegründet beim Senden, verbraucht beim Empfangen
- Interprozesskommunikation - naive Sicht:



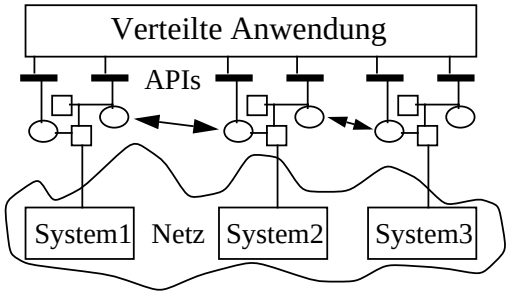
- Welche Kommunikationsanweisungen “matchen”?
- Empfangsbereitschaft, aber keine Nachricht? (verlorene Nachricht?)
- Nachricht, aber keine Empfangsbereitschaft?
- Sprachliche Ausprägung der Kommunikation?
- Wie wird adressiert?

==> Vielfältige Aspekte, Varianten, Probleme...

Message Passing System

einfache Form von "Middleware" (--> später)

- Organisiert den Nachrichtentransport
- Bietet Kommunikationsprimitive (APIs) an
  - z.B. send (...) bzw. receive (...)
  - ggf. auch ganze Bibliothek unterschiedlicher Kommunikationsdienste
  - Verwendbar in gängigen Programmiersprachen (oft zumindest C)



- Besteht aus Hilfsprozessen, Pufferobjekten...
- Verbirgt Details des zugrundeliegenden Netzes
- Verwendet vorhandene Protokolle; implementiert ggf. (damit) neue Protokolle
- Garantiert (abhängig von der "Semantik") gewisse Eigenschaften
  - z.B. Reihenfolgeerhalt
- Abstrahiert von Implementierungsdetails
  - wie z.B. Puffer, Low-level-Adressen etc.
- Maskiert gewisse Fehler
  - z.B. durch automatische Wiederholung nach einem timeout
- Verbirgt Heterogenität unterschiedlicher Rechner- bzw. Betriebssystemplattformen

===> Portabilität!

Nachrichtenkommunikation - pragmatische Aspekte

Vollständige Transparenz lässt sich kaum oder nur sehr teuer realisieren; gelegentlich schlagen Eigenschaften von tieferen Protokollschichten oder der Einsatzumgebung durch, dies betrifft z.B.:

- *Nachrichtenlänge*
  - fest
  - variabel aber begrenzt
  - (prinzipiell) unbegrenzt

  } dann muss man mit solchen Einschränkungen leben und darum herumprogrammieren

- *Zuverlässigkeitsgrad*: Nachrichtenverlust
  - nicht bemerkt
  - gemeldet
  - vermieden

  qualitatives Merkmal

- *Zuverlässigkeitsgrad*: Nachrichtenverfälschung
  - nicht bemerkt
  - erkannt und gemeldet
  - automatisch korrigiert

(*Techniken zur Erhöhung des Zuverlässigkeitsgrades:* Timeouts, Quittungen, Sequenznummern, Wiederholungen, Prüfsummen, fehlerkorrigierende Codes,...)

Derartige pragmatische Aspekte müssen in der Praxis neben der eigentlichen Kommunikationssemantik ebenfalls beachtet werden!

Prioritäten von Nachrichten?

- Vgl. dies mit Prioritätsebenen von Unterbrechungen

- *Semantik* zunächst nicht ganz klar:
  - Soll (kann?) Transportsystem Nachrichten höherer Priorität bevorzugt (=?) befördern?
  - Sollen (z.B. bei fehlender Pufferkapazität) Nachrichten niedrigerer Priorität überschrieben werden?
  - Wieviele Prioritätsstufen gibt es?
  - Sollen beim Empfang zunächst Nachrichten mit höherer Priorität angeboten werden?

- Mögliche *Anwendungen*:
  - Unterbrechen laufender Aktionen (--> Interrupt)
  - Aufbrechen von Blockaden
  - Out-of-band-Signalisierung

  } Durchbrechung der FIFO-Reihenfolge!

(Vgl. auch Service-Klassen in *Rechnernetzen*: bei Rückstaus bei den Routern soll z.B. interaktiver Verkehr bevorzugt werden vor ftp etc.)

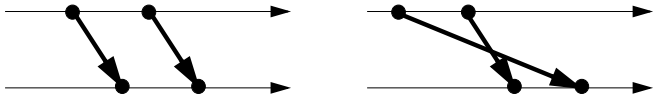
Vorsicht bei der Anwendung: Nur bei klarer Semantik verwenden; löst oft ein Problem nicht grundsätzlich!

- *Inwiefern* ist denn eine (faule) Implementierung, bei der "eilige" Nachrichten (insgeheim) wie normale Nachrichten realisiert werden, nicht korrekt?

Ordnungserhalt

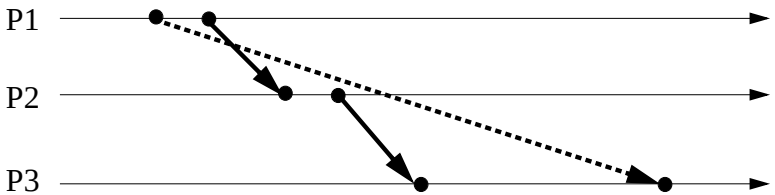
- Oft werden vom Kommunikationssystem keine Garantien bzgl. Nachrichtenreihenfolgen gegeben

- Eine mögliche Garantie stellt FIFO (First-In-First-Out) dar: Nachrichten zwischen zwei Prozessen überholen sich nicht: Sendereihenfolge = Empfangsreihenfolge



FIFO kein FIFO

- FIFO garantiert allerdings nicht, dass Nachrichten nicht indirekt (über eine Kette anderer Nachrichten) überholt werden



nicht kausal geordnet

- Möchte man auch dies haben, so muss die Kommunikation "kausal geordnet" sein (Anwendungszweck?)
  - entspricht einer "Globalisierung" von FIFO auf mehrere Prozesse
  - "Dreiecksungleichung": Keine Information erreicht Empfänger auf Umwegen schneller als auf direktem Wege
  - Denkübung: wie garantiert (d.h. implementiert) man kausale Ordnung auf einem System ohne Ordnungsgarantie?