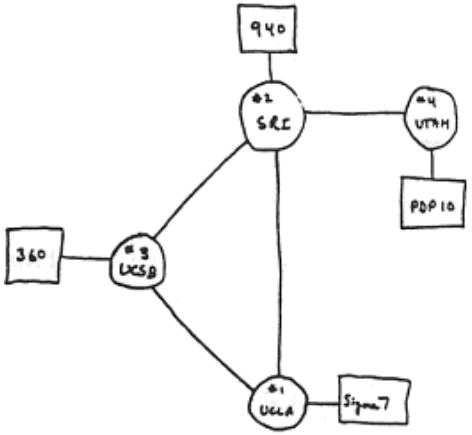
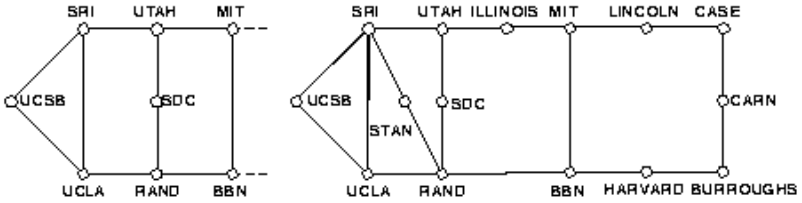
Internet-Geschichte: Pionierzeit

- 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



- 1971: Betriebsaufnahme des ARPAnet

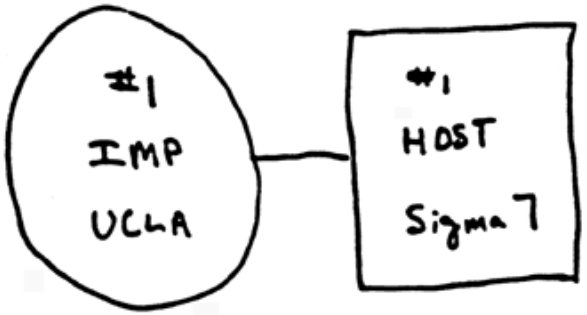


Juli 1970 März 1971

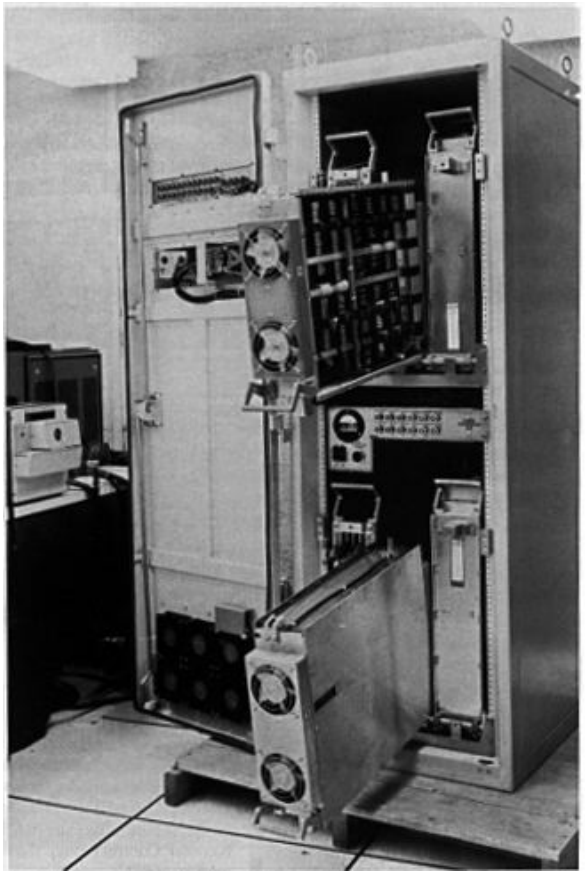
- 1971: Experiment zum Einloggen auf entfernten Rechnern; erstmalig Nutzung von E-Mail

-1972: Erste öffentliche Demonstration des Netzes

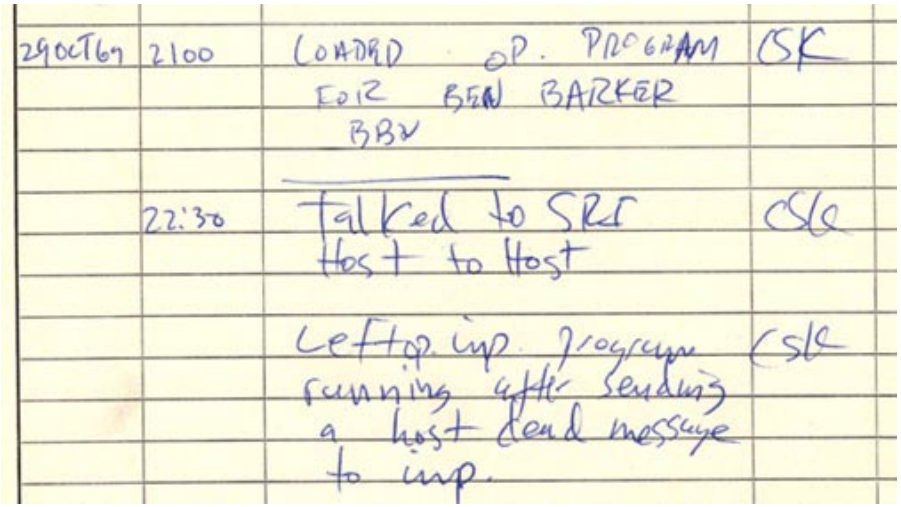
-1974: Neue Protokollsuite: TCP/IP



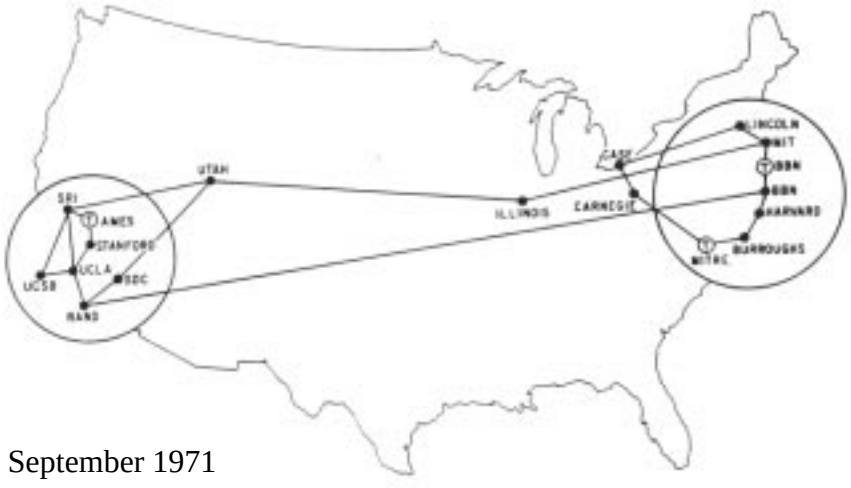
Der erste Knoten (Skizze vom September 1969)



Ein ARPANET-IMP (Interface Message Processor)

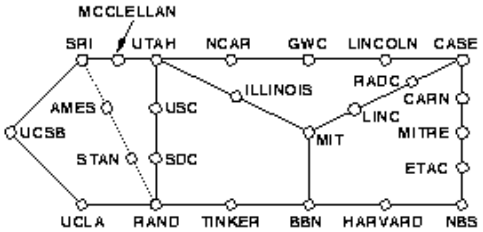


UCLA-Logbucheintrag der ersten Verbindung zu SRI (29. Okt. 1969)

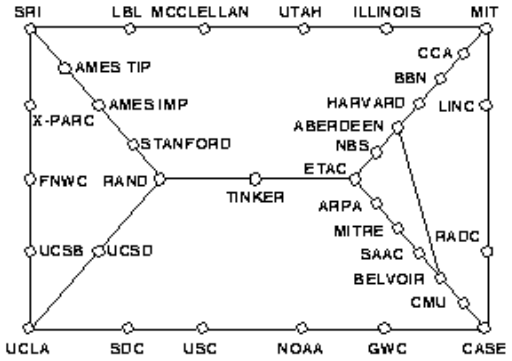


September 1971

Internet-Geschichte: Wachstum



April 1972



September 1972

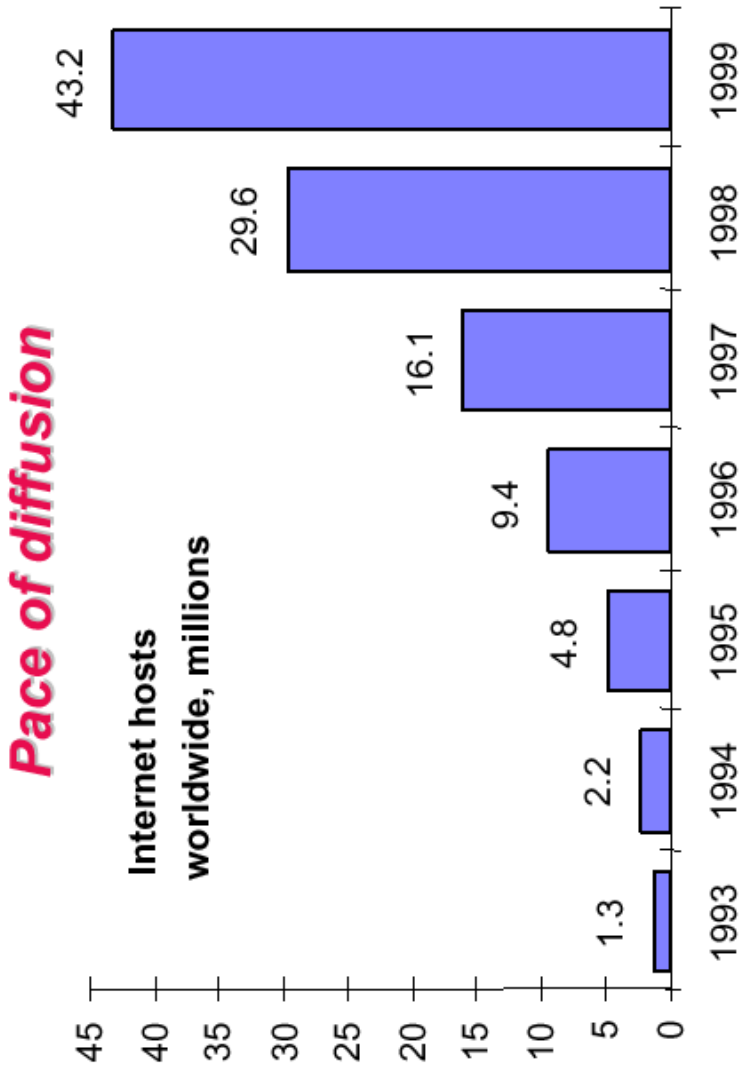
- Netz für Computer Science und andere Wissenschaften

- 1980: Integration der TCP/TP-Protokolle in UNIX
- 1981: Gründung des CSnet (Computer Science Network) durch NSF (National Science Foundation)
- 1986: NSFnet als zweites Backbone
- 1988: IP-Verbindung zum Internet aus Deutschland
- 1991: EBONE: Europäisches Backbone

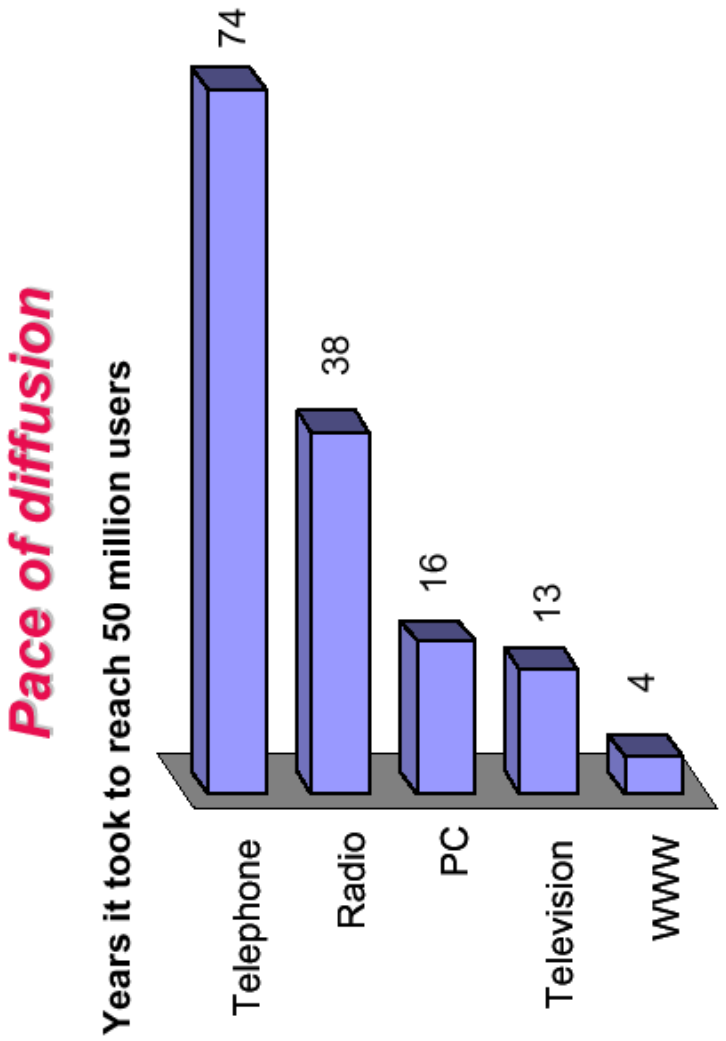
- Kommerzialisierung und Popularisierung

- 1991: Internet Exchange CIX (“Commercial Internet Exchange”)
- 1997: Streit der deutschen kommerziellen Internetprovider mit dem deutschen Wissenschaftsnetz des DFN-Vereins
- Dez. 1999: 293 996 Rechner in der Schweiz, 1 635 067 in Deutschland

http://www.nic.de/DENICdb/stats/index.html
http://navigators.com/statall.gif
http://www.isc.org/ds/
http://www.nw.com/zone/WWW/
http://www.ripe.net/

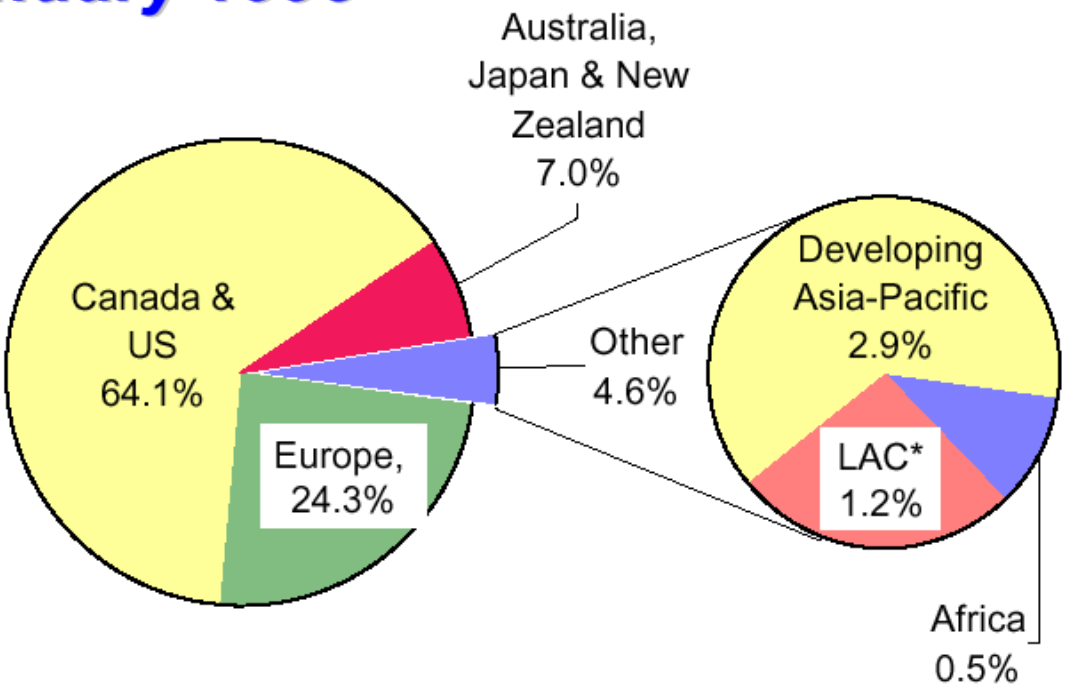


Quelle: Laura Männistö, Ben Petrazzini, Strategic
Planning Unit, ITU:Challenges to the Network 1999
2 nd World Telecommunication Indicators Meeting, Geneva, 29 - 31 March 1999
http://www.itu.int/ti/WTIM99/Documents.html



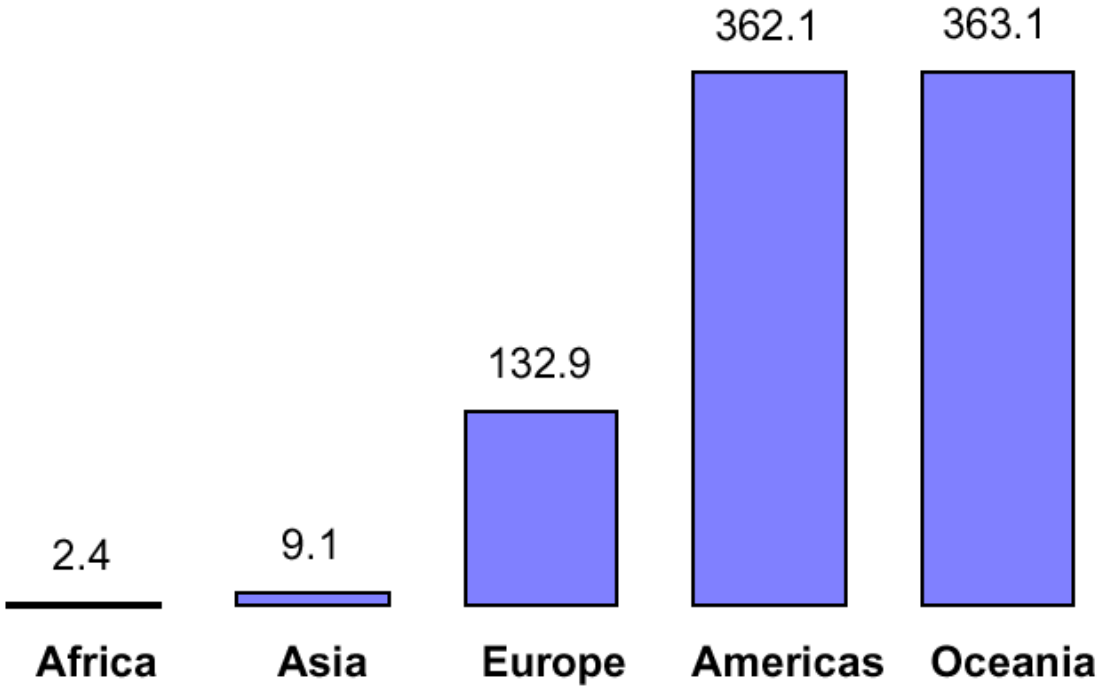
Quelle: LAURA MÄNNISTÖ, BEN PETRAZZINI, STRATEGIC
PLANNING UNIT, ITU:CHALLENGES TO THE NETWORK 1999
2 nd World Telecommunication Indicators Meeting, Geneva, 29 - 31 March 1999
http://www.itu.int/ti/WTIM99/Documents.html

Distribution of Internet hosts, January 1998



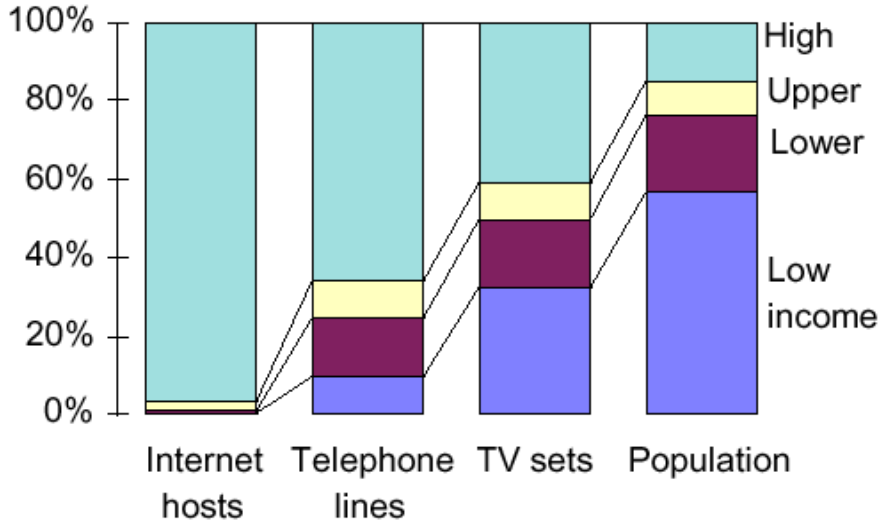
Source: ITU "Challenges to the Network: Internet for development, 1999".

Internet host density by region, January 1999, Per 10'000 inhabitants



Source: ITU "Challenges to the Network: Internet for Development, 1999", Network Wizards.

Persistence of inequalities in service take up: Distribution in percentages



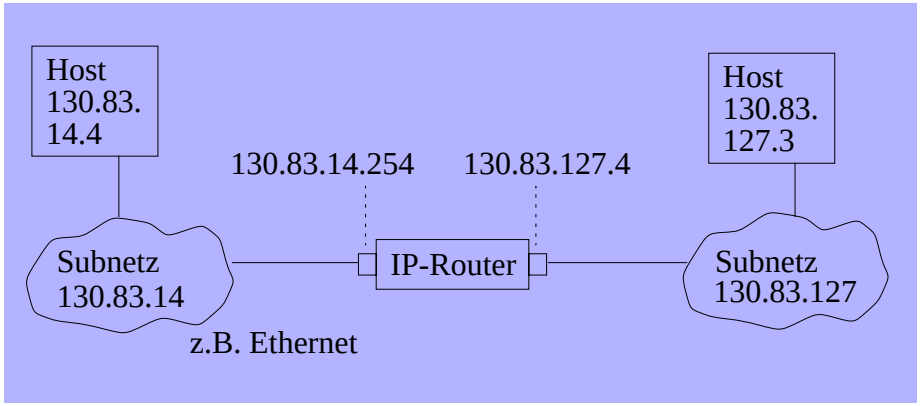
Source: ITU "Challenges to the Network: Telecoms and the Internet", 1997. Network Wizards (www.nw.com).

http://www.itu.int/ti/papers/ISS97/23sep97.pdf

Internet: Namen und Adressen

- Symbolische Namen von Maschinen
  - z.B. www.inf.ethz.ch
  - gut zu merken; relativ unabhängig von spezifischer Maschine
  - muss bei Verwendung in eine IP-Adresse umgesetzt werden!
  - dazu verteilte Namensdienste (DNS; nslookup etc.)

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



- Zuordnung IP-Adresse zur 6-gliedrigen Adresse der MAC-Ebene (z.B. Ethernet) eines Subnetzes durch ARP ("Address Resolution Protocol"; RFC 826)
  - Broadcast: "Wer hat die IP-Adresse 1.2.3.4?" z.B. im Subnetz 1.2.3
  - jede Maschine prüft ihre IP-Adresse
  - eine antwortet mit ihrer Ethernet-Adresse

IP, UDP und TCP

Ebene 4 (Transport)	Transmission Control Protocol **TCP**	User Datagram Protocol **UDP**
Ebene 3 (Network)	Internet Protocol **IP**	

- Internet Protokoll (IP)
  - verbindungsloses, ungesichertes Protokoll
  - sorgt für Routing der Pakete
  - "best effort": Pakete können u.U. verloren gehen oder verfälscht werden

- UDP

  logische Kommunikationsendpunkte, durch 16-Bit-Zahlen identifiziert

  - i.w. Hinzufügen von Port-Nummern ("Transportadressen") zu IP
  - damit eine Kommunikationsbeziehung von einem Sendeport eines Rechners zu einem Empfangsport eines anderen Rechners
  - Ports können von (den Prozessen) einer Anwendung benutzt werden
  - ungesichert bzgl. Paketverlust
  - broadcast- und multicastfähig

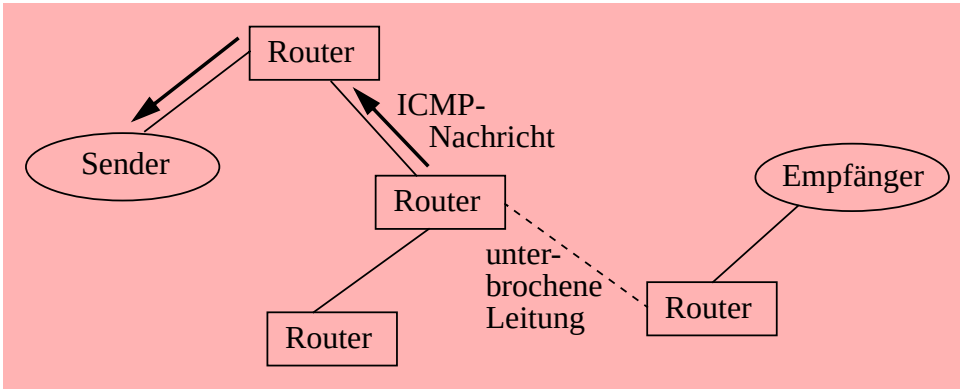
- TCP
  - aufwendiger als UDP: gesicherte Verbindungen ("byte stream")
  - Verbindung besteht aus Sendeportnummer, Sende-IP-Nummer, Empfangsportnummer, Empfangs-IP-Nummer
  - Sequenznummern und acknowledge / retransmit
  - Timeouts und Fenstergrössen werden dynamisch angepasst
  - ebenfalls Ports (16-Bit-Nummern) als Kommunikationsendpunkte

- Auf TCP/IP (bzw. UDP/IP) bauen weitere Internet-Protokolle höherer Ebene auf, z.B.: telnet, ftp, smtp, http
  - ferner Systeme wie NFS (verteiltes Dateisystem) etc.

ICMP

- Hilfsprotokoll ICMP (Internet Control Message Protocol) auf IP-Ebene
  - schwerwiegende Probleme (z.B. Unterbrechung einer Leitung) werden zur Vermeidung von Folgefehlern mittels ICMP den Kommunikationspartnern mitgeteilt
  - ICMP unterstützt den Austausch von Statusanfragen und Zustandsinformation zur Kontrolle und Fehlersuche im Netz (Test von Routen; Messen von Verzögerungen etc.)
  - ICMP nutzt IP selbst als "Transportdienst"
  - ICMP meldet keine Fehler bzgl. ICMP-Pakete

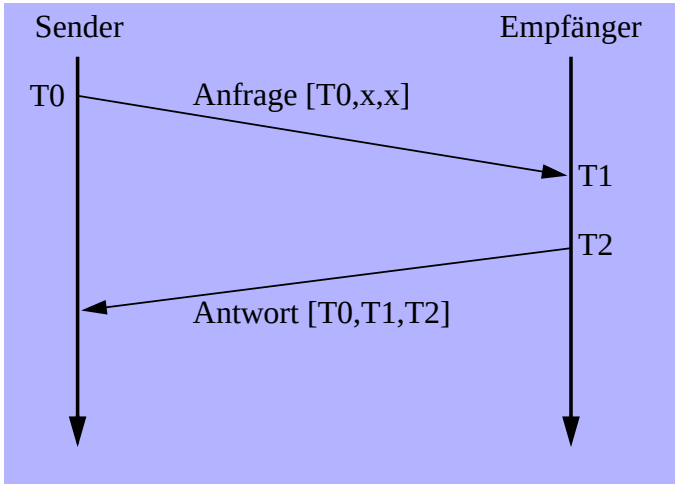
- Einige Meldungen von ICMP:
  - *Zieladresse nicht erreichbar* (destination unreachable): Ein Datenpaket konnte nicht zugestellt werden.
  - *Zeit abgelaufen* (time exceeded): Ein Datenpaket wurde wegen Ablauf seiner Lebenszeit von einem Router verworfen.
  - *Quellendämpfung* (source quench): Ein überlastetes Kommunikationssystem fordert den Sender auf, die Übertragungsrate zu senken.



ICMP (2)

- Statusanfragen mit ICMP

  - *Echo* und Echoantwort (echo request 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 request and timestamp reply): Dient der Bestimmung von Paketumlaufzeiten. Die Meldungen umfassen mehrere Felder zur Aufnahme von Zeitstempeln, an Hand derer die Paketbearbeitungszeiten beim Empfänger und die Verzögerung im Netzwerk bestimmt werden können.



- ICMP wird z.B. von “ping” benutzt:

  - *Ping utilizes the ICMP protocol’s ECHO_REQUEST datagram to elicit an ICMP ECHO_RESPONSE from the specified host or network gateway. Ping computes round trip times and packet loss statistics; it displays a summary of this information ...*

Netzstatistik auf IP-Ebene

- Beispiel für einen ping-Aufruf:

```
ping  www.altavista..com
PING altavista.digital.com: 56 data bytes
64 bytes from altavista.com (204.123.2.69): icmp_seq=0. time=282. ms
64 bytes from altavista.com (204.123.2.69): icmp_seq=2. time=358. ms
64 bytes from altavista.com (204.123.2.69): icmp_seq=4. time=290. ms
64 bytes from altavista.com (204.123.2.69): icmp_seq=5. time=304. ms
64 bytes from altavista.com (204.123.2.69): icmp_seq=6. time=265. ms
64 bytes from altavista.com (204.123.2.69): icmp_seq=8. time=456. ms
64 bytes from altavista.com (204.123.2.69): icmp_seq=10. time=251. ms
----altavista.com PING Statistics----
15 packets transmitted, 9 packets received, 40% packet loss
round-trip (ms)  min/avg/max = 251/309/456
```

- Explizites Setzen des time-to-live-Parameters

```
ping -t 1 www.altavista.com
ICMP Time exceeded in transit from rou-ifw-inf-vs.ethz.ch (129.132.13.65)
ping -t 3
ICMP Time exceeded in transit from swiez1-eth-switch-fast.ethz.ch (192.33.92.87)
ping -t 5
ICMP Time exceeded in transit from swiNY1-A5-0-0-200.switch.ch (130.59.33.202)
ping -t 7
ICMP Time exceeded in transit from ATM2-0.XR2.NYC4.ALTER.NET (152.63.22.14)
ping -t 10
ICMP Time exceeded in transit fromt ATM6-0.XR2.PAO1.ALTER.NET (152.63.49.33)
```

Dies lässt sich mit dem Kommando “traceroute” automatisieren...

Traceroute

NAME

traceroute - print the route packets take to network host

DESCRIPTION

The Internet is a large and complex aggregation of network hardware, connected together by gateways. Tracking the route one's packets follow (or finding the miscreant gateway that's discarding your packets) can be difficult...

This program attempts to trace the route an IP packet would follow to some internet host by launching UDP probe packets with a small ttl (time to live) then listening for an ICMP "time exceeded" reply from a gateway. We start our probes with a ttl of one and increase by one... Three probes are sent at each ttl setting and a line is printed showing the ttl, address of the gateway and round trip time of each probe.

AUTHOR

Implemented by Van Jacobson...

Washington - Moskau

1 **www1.whitehouse.gov** (198.137.240.91)
2 198.137.240.65 (198.137.240.65)
3 198.137.240.34 (198.137.240.34)
4 **ip1.ci3.herndon.va.us.psi.net** (38.25.11.1)
5 **sc.southeast.us.psi.net** (38.1.25.1)
6 38.1.4.6 (38.1.4.6)
7 **tip1-mae-east.cwix.net** (192.41.177.182)
8 **tip-7513-2-f11-0.cwix.net** (207.124.104.98)
9 **blb-7513-1-h3-0.cwix.net** (207.124.105.78)
10 **phy-7513-1-h9-0.cwix.net** (207.124.117.5)
11 **nyd-7513-2-h4-0.cwix.net** (207.124.108.41)
12 **lon-7513-2-a10-0-1.cwix.net** (207.124.108.62)
13 207.124.116.70 (207.124.116.70)
14 194.186.157.69 (194.186.157.69)
15 194.186.157.73 (194.186.157.73)
16 **cisco1.Moscow.ST.NET** (194.67.0.246)
17 **MSK-M9-1-S1-0-1.iip.net** (195.178.192.65)
18 **kremlin.fr.iip.net** (195.178.192.73)
19 195.178.196.70 (195.178.196.70)

http://www.itu.int/ti/WTIM99/Presentations/paltridge_e.pdf

Traceroute to Microsoft

traceroute to **www.microsoft.com** (198.105.232.6)
1 swiEZ2.switch.ch (130.59.1.202) 3 ms 2 ms 2 ms
2 swiEZ6.switch.ch (130.59.20.206) 4 ms 2 ms 2 ms
3 swiBT2.switch.ch (130.59.34.1) 13 ms 7 ms 9 ms
4 swiBT1.switch.ch (130.59.37.1) 15 ms 9 ms 15 ms
5 CH-s0.dante.bt.net (194.72.26.129) 13 ms 13 ms 15 ms
6 CH-f0-0.eurocore.bt.net (194.72.24.65) 9 ms 10 ms 11 ms
7 194.72.24.241 (194.72.24.241) 40 ms 59 ms 39 ms
8 **UK**-f0.dante.bt.net (194.72.7.5) 39 ms 44 ms 37 ms
9 **New-York**2.dante.net (194.72.26.210) 253 ms 293 ms 279 ms
10 mf-1.cnss32.New-York.t3.ans.net (204.149.4.5) * 255 ms 294 ms
11 t3-0.cnss48.**Hartford**.t3.ans.net (140.222.48.1) * 235 ms 271 ms
12 t3-2.cnss43.**Cleveland**.t3.ans.net (140.222.43.3) 300 ms * 248 ms
13 t3-1.cnss27.**Chicago**.t3.ans.net (140.222.27.3) 278 ms 307 ms 271 ms
14 t3-1.cnss96.**Denver**.t3.ans.net (140.222.96.2) 279 ms 290 ms *
15 t3-1.cnss8.**San-Francisco**.t3.ans.net (140.222.8.2) 309 ms 330 ms 279 ms
16 mf-0.cnss11.San-Francisco.t3.ans.net (140.222.8.195) * 341 ms 390 ms
17 enss257-F.ans.net (198.32.128.65) 318 ms 322 ms 365 ms
18 enss456-H.ans.net (198.32.128.226) 334 ms 315 ms 361 ms
19 pb-F1.MCI.net (198.32.128.197) 341 ms * 316 ms
20 borderx1-hssi3-0.SanFrancisco.mci.net (204.70.158.105) 308 ms 335 ms 291 ms
21 core2-fddi-0.SanFrancisco.mci.net (204.70.158.49) 325 ms 317 ms *
22 core1-hssi-2.**Sacramento**.mci.net (204.70.1.146) 349 ms * 356 ms
23 core-hssi-3.**Seattle**.mci.net (204.70.1.150) 379 ms * 319 ms
24 border1-fddi-0.Seattle.mci.net (204.70.2.146) * 451 ms 525 ms
25 nwnet.Seattle.mci.net (204.70.52.6) 353 ms * 373 ms
26 seabr1-gw.nwnet.net (192.147.179.5) * * 355 ms
27 **microsoft**-t3-gw.nwnet.net (198.104.192.9) 402 ms 354 ms *
28 131.107.249.3 (131.107.249.3) 346 ms 331 ms 345 ms
29 www.microsoft.com (198.105.232.6) 409 ms 359 ms 392 ms

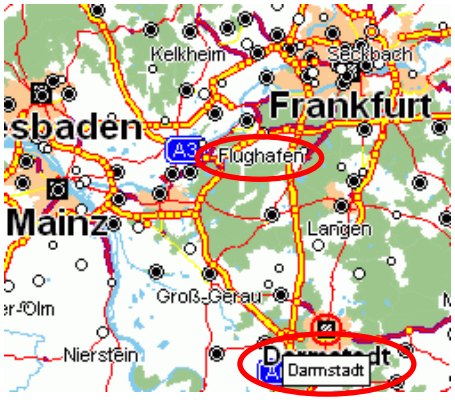
So sah es jedenfalls vor 1999 aus - man probiere selbst, ob inzwischen die Verbindung kürzer und schneller geworden ist und ob man wirklich noch bis ans Ende kommt (oder ob irgend ein Router die “Ping-Pakete” herausfiltert aus Furcht vor Missbrauch durch Hacker)!

Übung: Man verfolge einmal den Weg zu www.cs.pku.edu.cn (162.105.203.94)

Der schnellste Weg von Darmstadt nach Frankfurt?

3 TU-Darmstadt2.WiN-IP.DFN.DE (188.1.11.17) 2 ms
4 ZR-Frankfurt1.WiN-IP.DFN.DE (188.1.11.113) 4 ms
5 IR-Perryman1.WiN-IP.DFN.DE (188.1.15.2) 114 ms
6 bordercore3-hssi0-0.Washington.mci.net (166.48.41.249) 130 ms
7 core3.Washington.mci.net (204.70.4.29) 134 ms
8 * * f3-1.t56-2.Washington-DC.t3.ans.net (140.222.56.122) 133 ms
9 h4-1.t32-1.New-York.t3.ans.net (140.223.33.21) 250 ms
10 f0-0.cnss38.New-York.t3.ans.net (140.222.32.198) 517 ms
11 enss567.t3.ans.net (199.222.50.14) 282 ms
12 204.151.208.166 (204.151.208.166) 375 ms
13 Hamburg1-s4-0.is-bone.net (195.180.0.5) 289 ms
14 Frankfurt-BB-h5-0.is-bone.net (195.180.0.222) 268 ms
15 Frankfurt5-e0.is-bone.net (195.180.3.14) 284 ms
16 195.180.3.154 (195.180.3.154) 279 ms
17 www.frankfurt-airport.de (194.195.240.85) **268 ms**

Seitdem es den Commercial Internet Exchange in Frankfurt gibt, geht es besser:

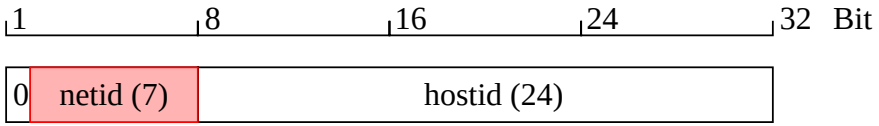


3 TU-Darmstadt2.WiN-IP.DFN.DE (188.1.11.17) 4 ms
4 ZR-Frankfurt1.WiN-IP.DFN.DE (188.1.11.113) 4 ms
5 cix-frankfurt1.WiN-IP.DFN.DE (188.1.164.14) 4 ms
6 DECIX.maz.net (194.31.232.14) 4 ms
7 Frankfurt-BB-h0-1-0.is-bone.net (195.180.0.29) 5 ms
8 Frankfurt5-fe1.is-bone.net (195.180.3.14) 5 ms
9 195.180.3.154 (195.180.3.154) 5 ms
10 www.frankfurt-airport.de (194.195.240.85) **6 ms**

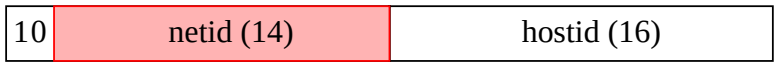
IP-Adressformat

- Es gibt 3 verschiedene Adressformate (“Adressklassen”)

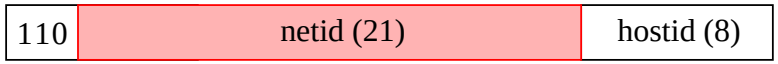
- *Class A:* 126 Netze mit jeweils bis zu ca. 4 Mio. Rechnern
  - Netze zu praktisch 100% aufgebraucht



- *Class B:* 16382 Netze mit jeweils bis zu 65534 Rechnern
  - Okt. 95: Netze zu 62% aufgebraucht

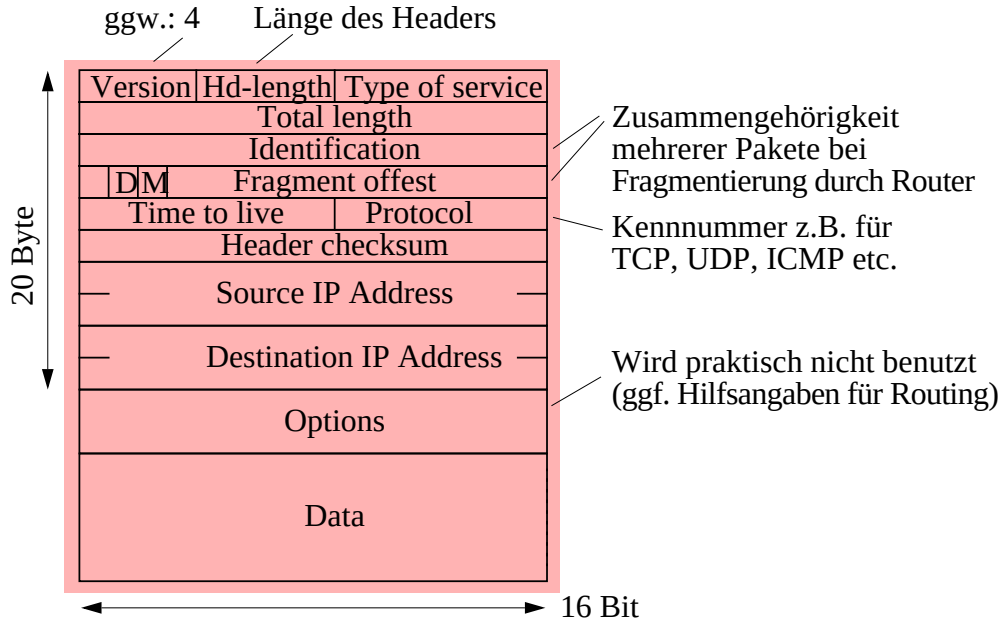


- *Class C:* ca. 2 Mio. Netze mit jeweils bis zu 254 Rechnern
  - Okt. 95: Netze zu 31% aufgebraucht



- Es gibt einige spezielle Adressen, z.B.
  - 0.0.0.0 : “this host”; bzw. 255.255.255.255 : broadcast auf dem LAN
  - 1110... sind Multicast-Adressen (Class D),
    z.B. 224.0.0.2 : alle Router im Subnetz
  - es gibt spezielle Multicast-Adressen, z.B. 224.0.1.7 für MBone, AudioNews...

IP-Datagramme



- Type of service: Priorität + Quality of Service
  - Idee: bei Dateitransfer ist reliability wichtiger als low delay
  - bei Audio ist low delay wichtiger als reliability
  - Router könnte z.B. zwischen Glasfaser und Satellit entscheiden
  - wird in der Praxis aber praktisch nicht genutzt

- Paketlänge max. 64k
  - in der Praxis meist < 1500 (IP-Pakete in Ethernet-Rahmen!)

- Fragmentierung
  - D=1: Don’t fragment: Router darf Paket nicht fragmentieren
  - M=1: More fragments (nur letztes Fragment hat dieses Bit = 0)

- Time to live: bei jedem Router um 1 vermindert
  - falls 0: Paket vernichten und Fehlermeldung an Absender (ICMP)

- Header checksum wird in jedem Router neu berechnet!

IPv6 - die neue IP-Version

- Probleme mit dem klassischen IP ("IPv4"):
  - zu wenig Adressen (sollte nicht am besten jeder TV-Settop, jeder Stromzähler und jeder Toaster seine eigene IP-Adresse bekommen?)
  - Anzahl der am Internet angeschlossenen Netze verdoppelt sich z.Z. mindestens jährlich; weiteres Wachstum weniger im Computer-Bereich als vielmehr bei "networked entertainment", der Fernsteuerung von Geräten in Gebäuden und Haushalten etc.
  - schlecht für "Hochgeschwindigkeit" (zu viel Aufwand in den Routern, z.B. Prüfsumme berechnen, Längenfeld evaluieren...)
  - schlecht geeignet für neuere Anforderungen wie Realzeitfähigkeit, Multicasting, Sicherheit...

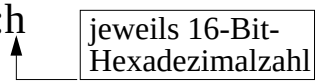
- Wesentliche Eigenschaften von IPv6:
  - 128-Bit-Adressen
  - Multicast-Möglichkeit
  - realzeitfähig
  - optionale Authentifizierung und Verschlüsselung
  - vereinfachter Header mit weniger Feldern (--> Beschleunigung)
  - optionale Erweiterungsmöglichkeiten (--> "offenes Design")
  - Unterstützung zur Autokonfiguration von Netzadressen neuer Geräte

- Koexistenz (über viele Jahre) mit IPv4
  - Versionsnummer (4 bzw. 6) steht am Anfang beider Header
  - nach und nach Router, die IPv4 und IPv6 können ("dual stack")
  - ggf. Tunneln von IPv6 über länger IPv4-Strecken und Teilnetze

IPv6: Adressformat

- 128 Bit (=16 Byte) lange Adressen (statt 32 Bit in IPv4)
  - Adressraumgrösse $2^{128} \approx 10^{38}$ --> ca. 10^{24} Adressen pro m^2 der Erde
  - 2^{96} Mal mehr als bei IPv4
  - letzten 48 Bits könnten u.U. die MAC-Adresse sein

- Notation: a:b:c:d:e:f:g:h



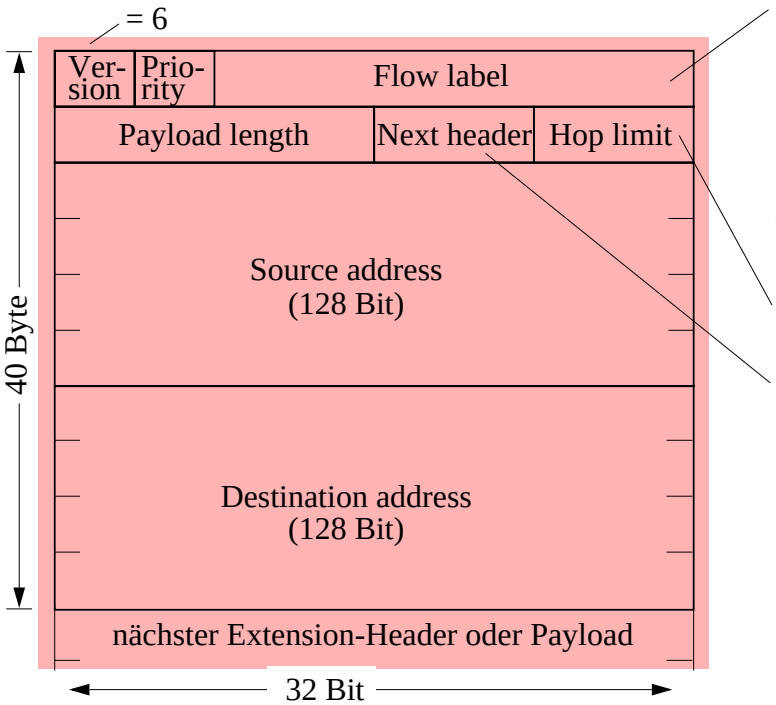
  - klassische IP-Adresse ("IPv4") jetzt so: ::128.83.5.72

- Jedes Kommunikationsinterface eines Knoten (Rechner, Router,...) kann *mehrere* IPv6-Adressen haben

- Lange Adressen mit hierarchischer Gliederung erlaubt in Routern die Aggregation von Adressbereichen
  - z.B.: "alles, was den Präfix xxx hat zu Knoten y"
  - dadurch kleinere Routingtabellen

- Neuer Vorschlag: *Anycast-Adressen*
  - Zieladresse spezifiziert das Ziel nur "vage"; die Router suchen sich dann einen besten / nächsten Knoten, der zu dieser Adresse passt (z.B. für Auskunftsdienste, mobile Stationen, lokale Filiale von McDonalds, ein Router eines Subnetzes...)

IPv6: Header



für Folge von Paketen zu einem Empfänger, für die der Sender eine spezielle Behandlung bei den Routern wünscht ("quality of service")

entspricht time to live bei IPv4

für zusätzliche Angaben in einem (oder mehreren) Headern am Ende dieses Headers; letzter Header nennt hier das Transportprotokoll (z.B. TCP oder UDP)

- Priority (4 Bits); zwei unabhängige Klassen:
  - 8-15 für Realzeit; z.B. Audio oder Video (Sender kann nicht gebremst werden, notfalls Datenpaket wegwerfen)
  - 1-7 falls Sender mittels Laststeuerung gebremst werden kann (z.B. 1 für news, 2 für Email, 4 für ftp / http, 6 für telnet, 7 für Routinginfo)
- Hop limit: reichen 255 hops für alle Zukunft aus?
  - was ist z.B. mit langen Nachsendeketten bei mobilen Stationen?
- Payload length: Genügt eine Maximallänge von 2^{16}?
  - ein Supercomputer möchte z.B. nicht alle 65535 Bits unterbrochen werden, wenn sehr grosse Datenmengen schnell zu versenden sind
  - hierfür ist bereits eine Lösung vorgesehen: "Jumbogramme"

IPv6: Flow Label

- Beispiel Multimedia-Konferenz (typischerweise mit Multicast-Adresse): getrennte flows für
  - Audio
  - Video
  - Graphik

  } unterschiedliche Anforderungen an Bandbreite, Delay, Varianz...
- Aus Sicht eines Routers haben alle Pakete eines flows identische Anforderungen z.B. hinsichtlich
  - maximale Verzögerung
  - Priorität
  - accounting
  - Sicherheit
  - Routing-Weg

  } Router kann Ressourcen für den flow einmalig reservieren (Puffer, Bandbreite, Service-Qualität von Subnetzen etc.) und braucht nicht immer alle Felder aller (Extension-) Header auszuwerten
- Quelle wählt eine flow-Nummer zufällig aus dem Bereich 1...2^{24}-1
  - natürlich verschieden von anderen flows der gleichen Quelle
  - keine "Kollisionen", da durch Quelladresse eindeutig

IPv6: Extension-Header

- Optionale weitere Header (angehängt an den Haupt-Header) für verschiedene Zwecke, z.B.:

- für Datenpakete > 64k (“Jumbogram” bis 2^{32} Byte)

- für Routing-Unterstützung (z.B. Source-Routing)

- Authentifizierung

  - Idee: Felder im Header, die sich unterwegs nicht ändern, mit einem Geheimschlüssel (nur Sender und Empfänger bekannt) konkatenieren und darauf kryptographische Prüfsumme (z.B. MD 5) anwenden

-Verschlüsselung

  - Felder für Schlüsselkennung etc. vorgesehen, Verschlüsselungsverfahren selbst ist aber nicht festgelegt (Default: DES mit cipher block chaining)

  - Verschlüsselung wurde bewusst getrennt von der Authentifizierung gehalten, da es bzgl. Verschlüsselung ggf. politische Vorbehalte gibt (Verbot der Anwendung bzw. des Exports), die bzgl. Authentifizierung i.a. nicht bestehen

-Destinations options

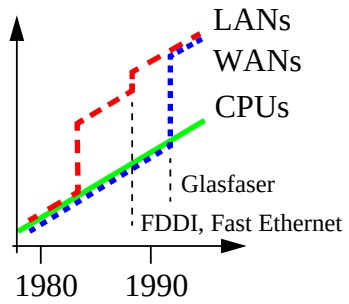
  - Für Informationen, die nur den Empfänger betreffen (brauchen die Router nicht zu betrachten)

IPv6: Routing-Effizienz

- Routing soll effizient sein

  - Routen werden tendenziell eher länger (viele Provider...)
  - Bandbreite von Netzen wächst schneller als cpu-Leistung

Leistungszunahme im qualitativen Vergleich



- Header hat konstante Grösse (40 Byte)

  - kein Längenfeld, das ausgewertet werden muss

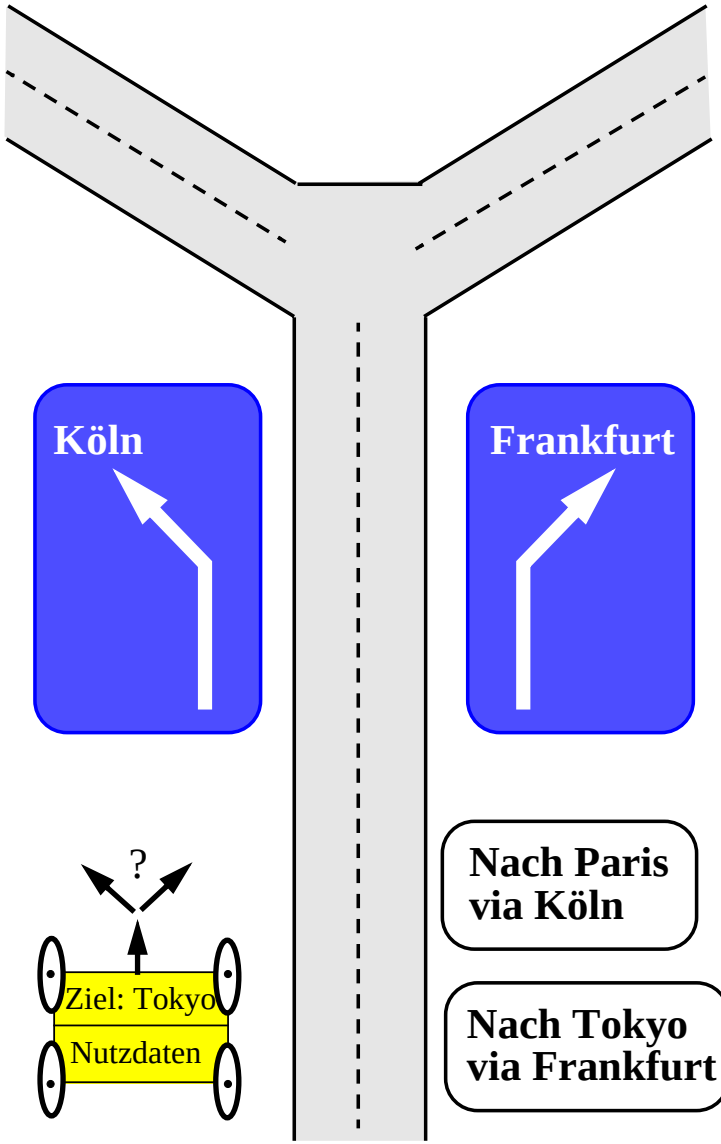
- Keine Berechnung von Prüfsummen (wie in IPv4)

  - wesentlicher Zeitgewinn!
  - Kritik: Entfernen der Bremsen macht ein Auto auch leichter und schneller
  - Rechtfertigung: Prüfsummen werden sowohl auf der tieferen als auch der höheren Ebene (“end-to-end-Argument”) verwendet

---

- Zu IPv6 lese man folgenden Artikel: *Robert M. Hinden: IP Next-Generation. Commun. of the ACM, Vol 39 No 6 (Jun. 1996), 61-71*
- Mehr zu IPv6 auch im Internet (auf den “offiziellen” Seiten der Arbeitsgruppen)

Routing (1)



Ein Knoten-punkt der Datenautobahn

Routingtabelle eines Netz-knotens

Problem: Wer stellt die Wegweiser auf?

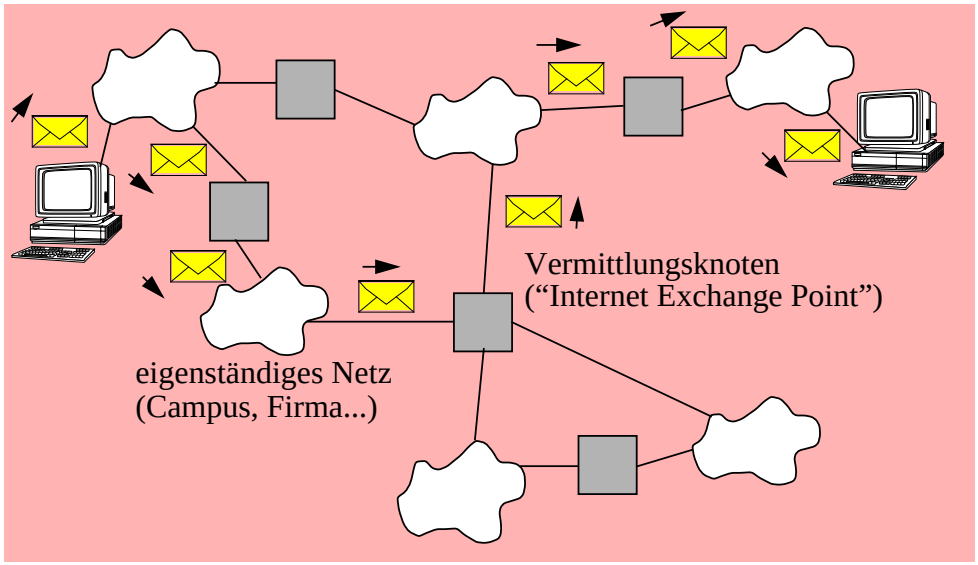
Routing (2)

- Jede Nachricht hat eine *Zieladresse* und muss ihren Weg durch das vermaschte Netz finden

Nachricht als Datenpaket

Daten (Bits)	Zieladresse

  - Problem: zyklenfreier, kurzer, schneller, billiger Weg



- Wegbestimmung meist mit *Optimierungszielen*, z.B.:
  - gleichmässige Auslastung der Leitungen
  - kurze Wartezeiten
  - hoher Durchsatz
  - Wege mit geringer Knotenzahl
  - Robustheit (ggf. alternative Wege bei Störungen)
  - geringe Kosten
  - geringe Fehlerrate

Routing (3)

- Unterscheide strenggenommen:
  - Ermittlung von Routing-Tabellen (eigentliches Routing)
  - Weiterleitung einer Nachricht mittels Routing-Tabellen (“forwarding”)

- Unterscheide:
  - Routing-Entscheidung wird beim Sender getroffen (“Source-Routing”)
  - Routing-Entscheidung wird bei den Knoten unterwegs getroffen

- Unterscheide:
  - “virtual circuit”-Verbindungen auf dem network layer
    --> Routing nur bei Aufbau der Verbindung
  - “connection-less”-Verbindungen
    --> dediziertes Routing (eigentlich “forwarding”) für jedes einzelne Paket (“Datagram”)
    --> Pakete können auf “parallelen” Pfaden laufen (Reihenfolge?!)

- Routing kann bei kleinen Fehlern zu Disastern führen!
  - Bsp.: Bedienungs- bzw. Konfigurationsfehler führt dazu, dass ein Knoten meint, jeden anderen Knoten (im Internet) mit Kosten 0 erreichen zu können
  - er erhält damit eine grosse Menge von Datenpaketen
  - kann mit diesen (und dieser Menge) aber nichts anfangen und ist überlastet
  - viele Datenpakete verschwinden in diesem “attraktiven” schwarzen Loch