Netzwerktechnologien
3 VO

Dr. Ivan Gojmerac
ivan.gojmerac@univie.ac.at

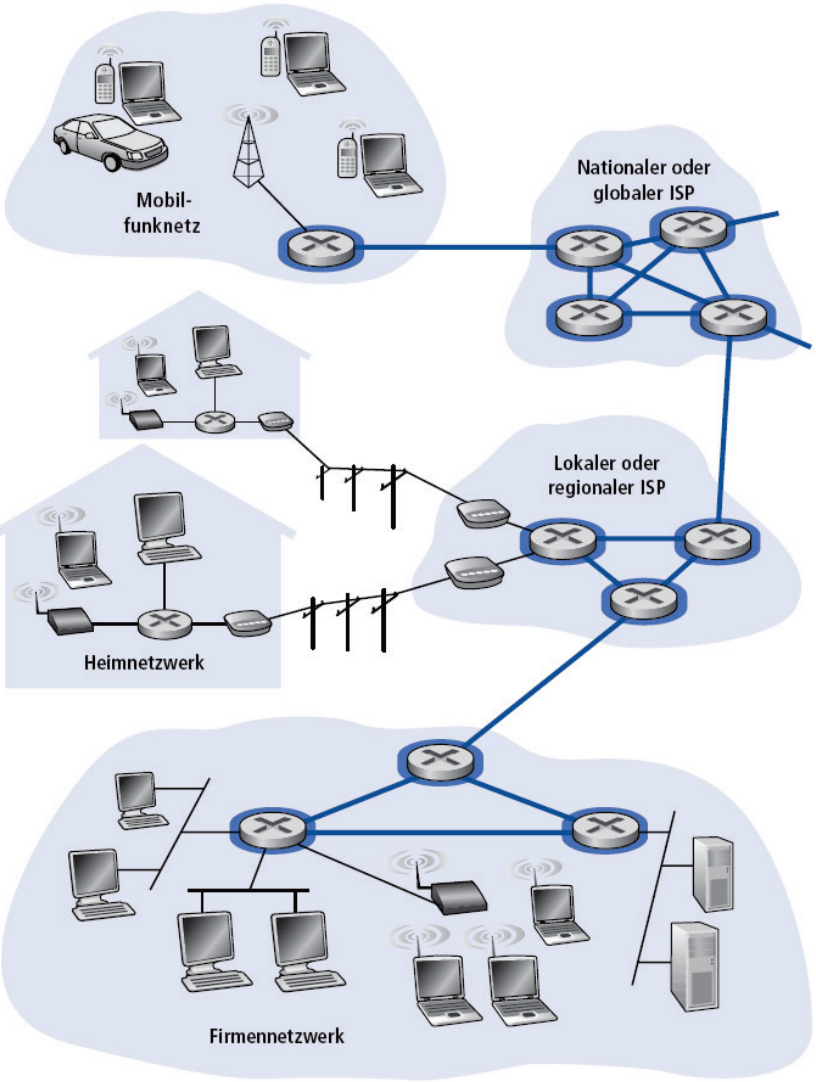
2. Vorlesungseinheit, 13. März 2013

Bachelorstudium Medieninformatik
SS 2013

1.3 Das Innere des Netzwerkes

1.3 Das Innere des Netzwerkes

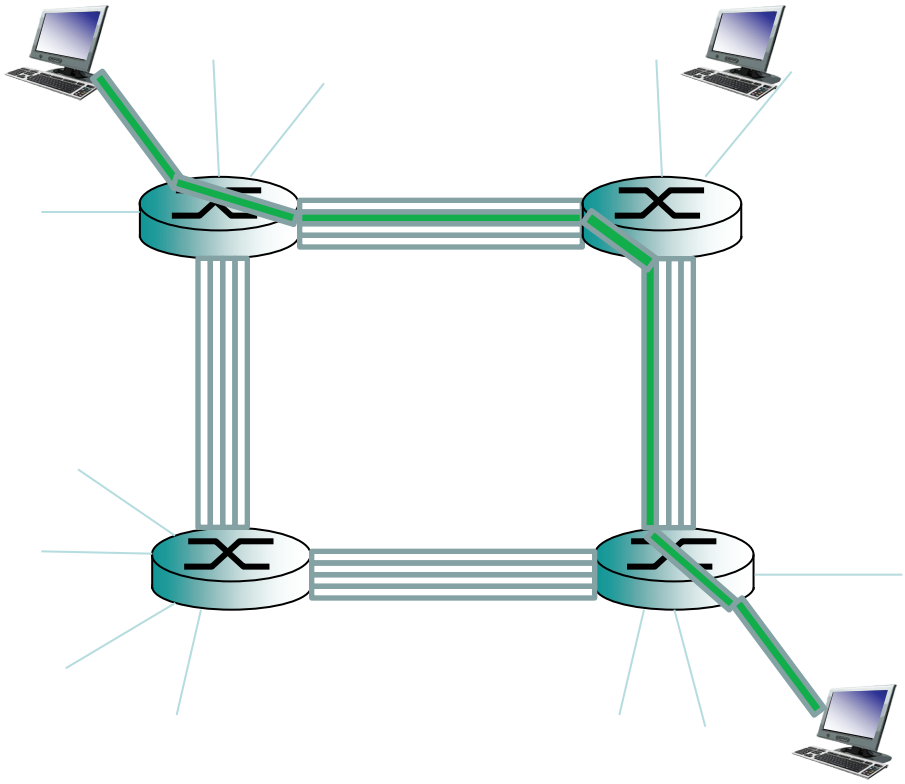
- Viele, untereinander verbundene Router
- Die zentrale Frage: Wie werden Daten durch das Netzwerk geleitet?
  - **Leitungsvermittlung**: eine dedizierte Leitung wird für jeden Ruf geschaltet
    → *Telefonnetz*
  - **Paketvermittlung**: Daten werden in diskreten Einheiten durch das Netzwerk geleitet
    → *Internet*



1.3.1 Leitungvermittlung

Leitungsvermittlung:

- Ende-zu-Ende-Ressourcen werden für einen Ruf reserviert:
  - Bandbreite auf Leitungen, Kapazität in Routern
  - Dedizierte Ressourcen: Keine gemeinsame Nutzung
  - Garantierte Dienstgüte wie beim “Durchschalten” einer physikalischen Verbindung
  - Vor dem Austausch von Daten müssen die notwendigen Ressourcen reserviert werden



1.3.1 Leitungsvermittlung

Wie teilt man die Bandbreite einer Leitung in Einheiten auf?

→ *Frequenzmultiplex*
(Frequency Division Multiplex, **FDM**)

→ *Zeitmultiplex*
(Time Division Multiplex**, TDM**)

<u>Bei Leitungsvermittlung:</u>

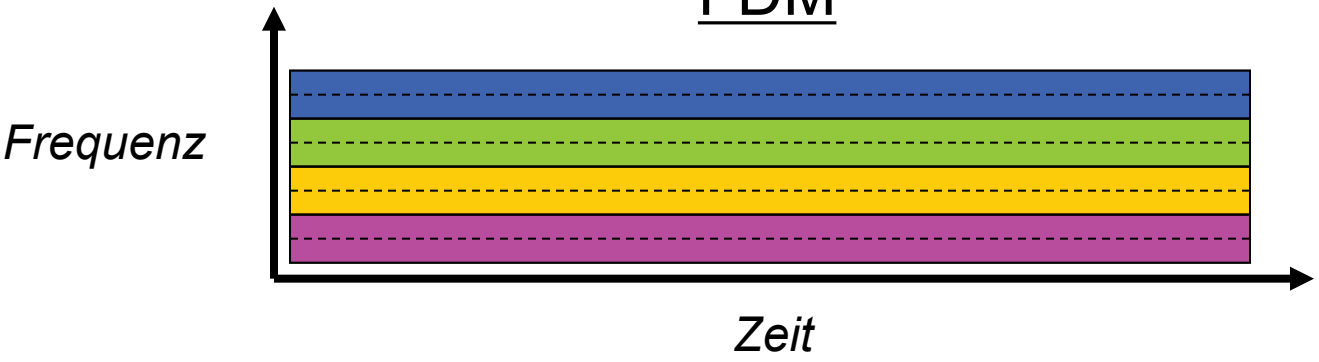
Netzwerkressourcen (z.B. Bandbreite) werden in Einheiten (Kanäle) aufgeteilt.

- Kanäle werden Rufen (Calls) zugewiesen
- Problem: **Kanäle bleiben ungenutzt**, wenn sie von ihrem Call nicht verwendet werden *(Keine gemeinsame Nutzung von Ressourcen)*

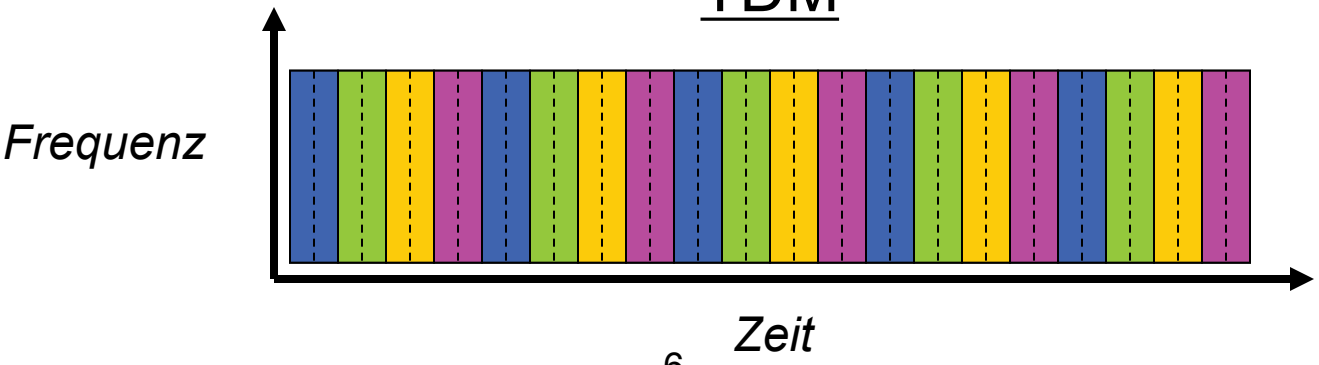
1.3.1 Frequency Division Multiplex und Time Division Multiplex

Beispiel: 4 Nutzer 

FDM



TDM



1.3.1 Leitungsvermittlung - Ein Beispiel

Wie lange dauert es, eine Datei mit 640.000 Bit von A nach B über ein leitungsvermitteltes Netzwerk zu übertragen?

- Alle Leitungen haben eine Bandbreite von 1.536 Mbit/s
- Alle Leitungen nutzen TDM mit 24 Zeitschlitzen/Sekunde
- 500 ms werden benötigt, um die Ende-zu-Ende-Leitung zu schalten

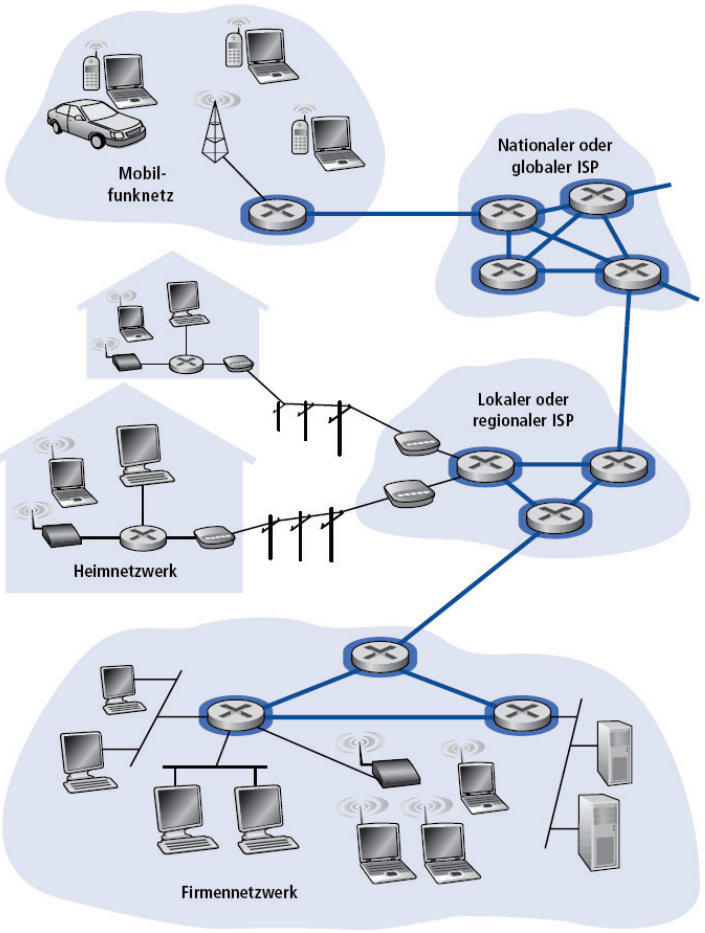
Antwort:

- 1.536.000 Bit/s / 24 = 64.000 Bit/s
- 640.000 Bit / 64.000 Bit/s = 10 s

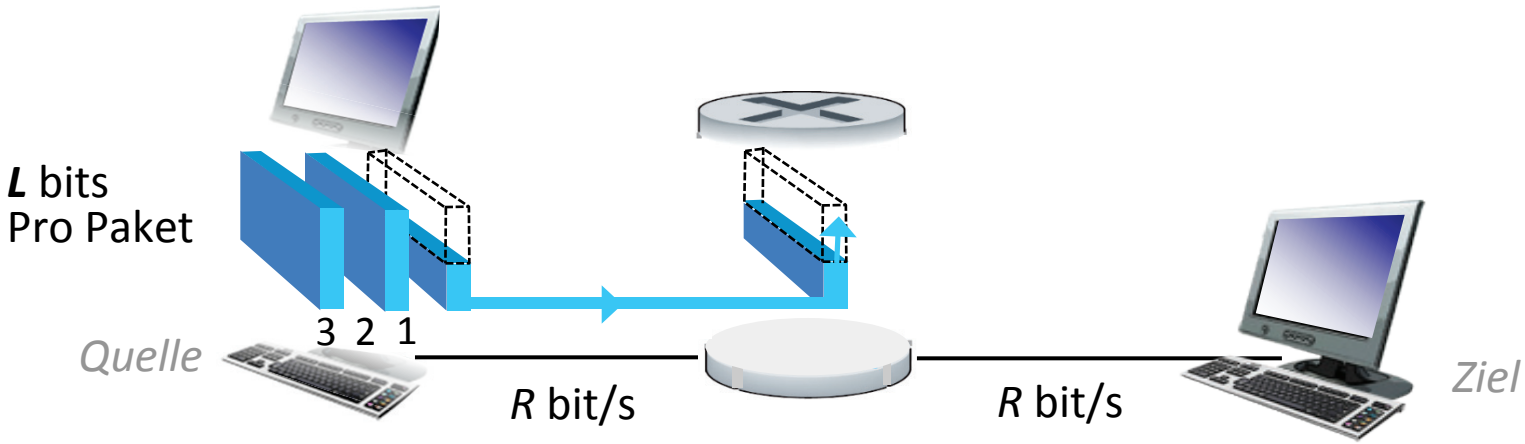
- Übertragungszeit = 10 + 0,5 = 10,5 s

1.3.1 Paketvermittlung

- Das Netzwerk besteht aus vielen untereinander verbundenen Routern
- Paketvermittlung:
  - Nachrichten des Application-Layers werden vom Host in Pakete aufgeteilt
  - Die Pakete werden über Links von einem Router zum nächsten weitergeleitet bis sie beim Ziel-Host ankommen
  - Die Übertragung eines Pakets nutzt die volle Kapazität des jeweiligen Links aus

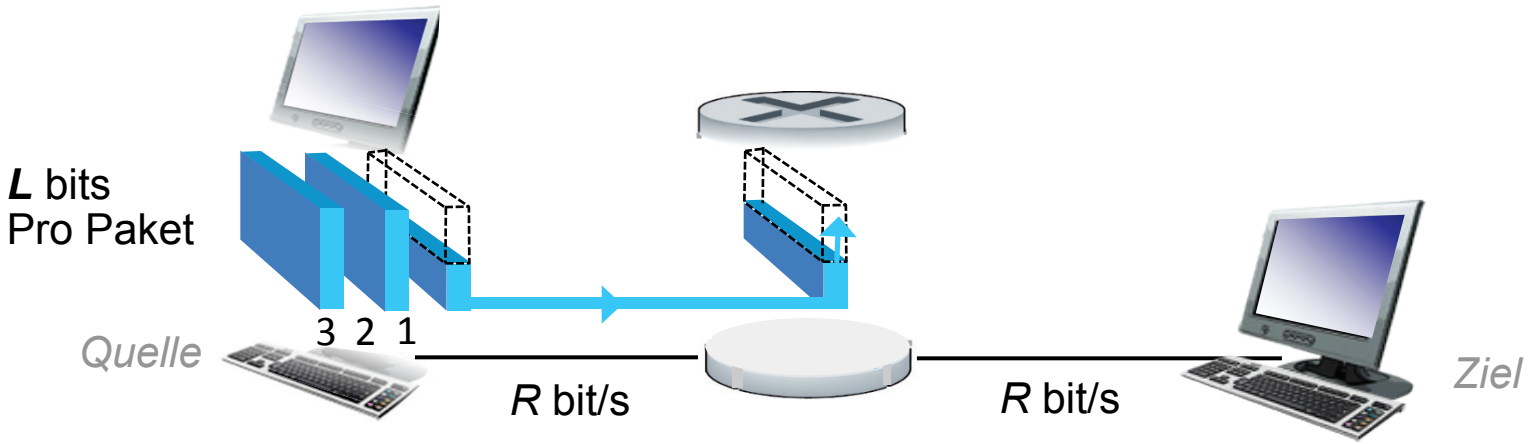


1.3.1 Paketvermittlung – Store and Forward



- Es dauert $\frac{L}{R}$ Sekunden um ein L Bit großes Paket über einen Link mit R bit/s zu übertragen
- **Store and Forward:** Das Paket muss erst vollständig beim Router angekommen sein bevor es über den nächsten Link übertragen werden kann
- Ende-zu-Ende Verzögerung: $2\frac{L}{R}$ (Ohne Ausbreitungsverzögerung)

1.3.1 Paketvermittlung – Store and Forward



Beispiel für 1 Hop:

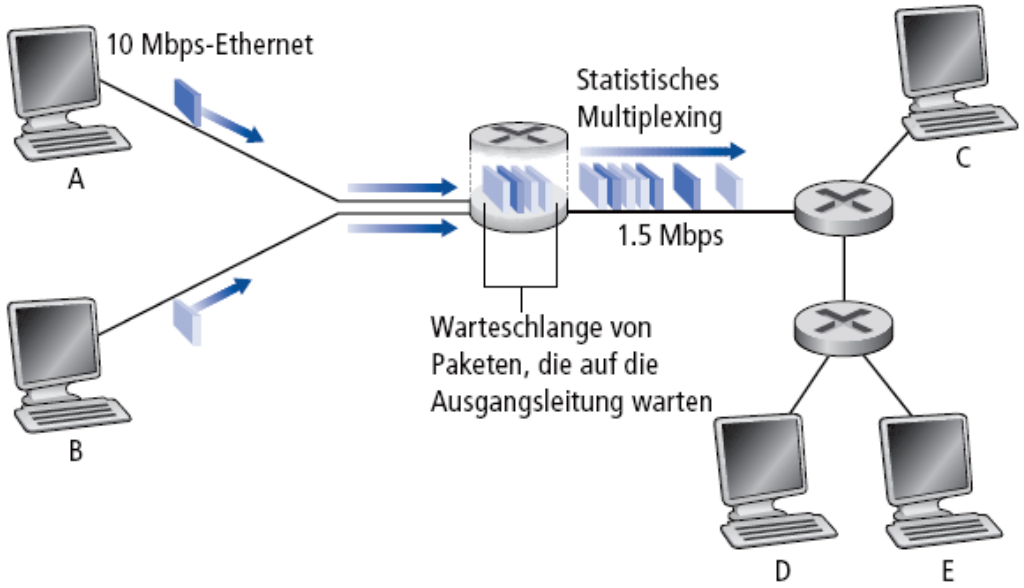
- L= 7,5 Mbits
- R= 1,5 Mbit/s

→ Übertragungsverzögerung = $\frac{L}{R}$ = 5s

Beispiel für n Hops:

→ Übertragungsverzögerung = n * $\frac{L}{R}$

1.3.1 Paketvermittlung – Queueing Delay, Paketverlust



Queuing und Paketverlust:

- Falls die Ankunftsrate (in Bits) der Pakete die Übertragungsrate des Links übersteigt:
  - Pakete werden gebuffert und warten auf Übertragung
  - Wenn Buffer voll ist werden ankommende Pakete verworfen

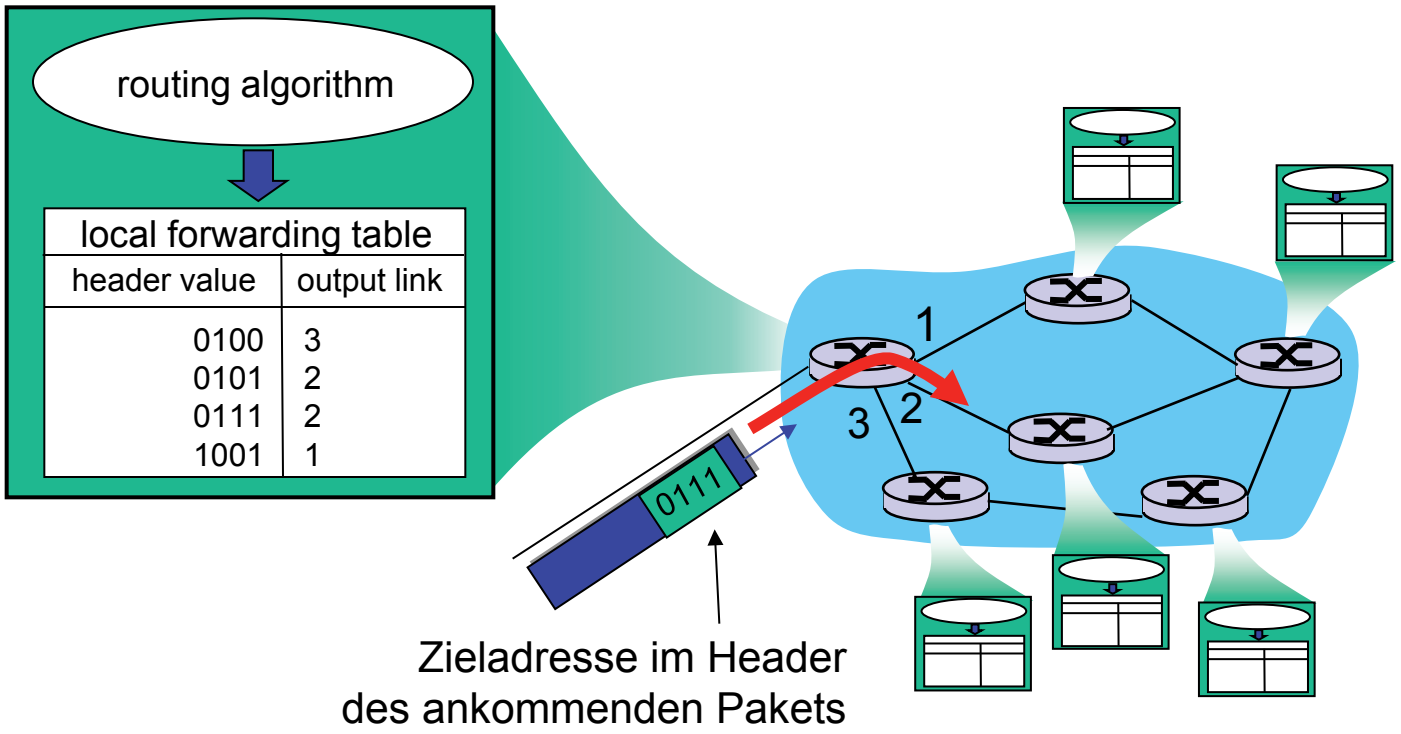
1.3.1 Statistisches Multiplexing

Die Folge von Paketen auf der Leitung hat kein festes Muster,
die Bandbreite wird nach Bedarf verteilt □***Statistisches Multiplexing***

(Nicht wie bei TDM, wo jede Verbindung immer den gleichen Zeitrahmen in einem sich wiederholenden Muster erhält.)

1.3.1 Paketvermittlung – Routing und Forwarding

- **Routing:** Legt fest welche Route Pakete vom Quell-Host zum Ziel-Host nehmen
- **Forwarding:** Leitet Pakete vom Router Eingang zum entsprechenden Router Ausgang weiter



1.3.1 Paketvermittlung vs. Leitungsvermittlung

Ist Paketvermittlung grundsätzlich besser?

- Sehr gut für unregelmäßigen Verkehr (*bursty traffic*)
  - Gemeinsame Verwendung von Ressourcen, bessere Auslastung
  - Keine Verschwendung von ungenutzten Ressourcen
  - Einfacher, keine Reservierungen
- Problem Überlast: Verzögerung und Verlust von Paketen
  - Protokolle für zuverlässigen Datentransfer und Überlastkontrolle werden benötigt
- Frage: Wie kann man leitungsähnliches Verhalten bereitstellen?
  - Bandbreitengarantien werden potentiell für Audio- und Videoanwendungen gebraucht → Quality of Service (QoS)

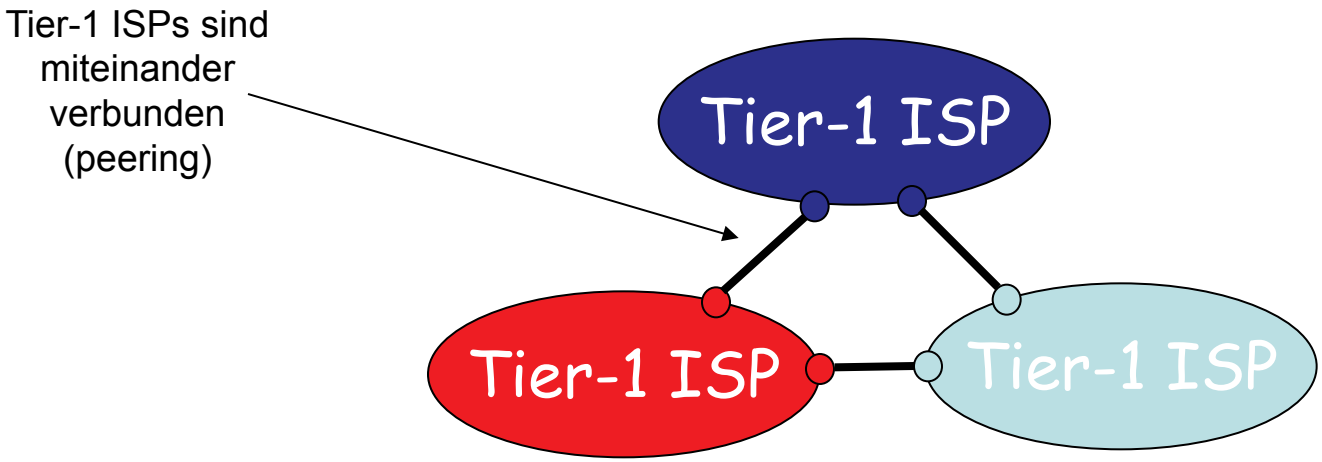
1.3.2 Wie gelangen Pakete zum Ziel?

1. Sender schreibt Zieladresse in den Paket-Header
2. Schickt es an den ersten Router
3. Router besitzt Weiterleitungstabellen
4. Lookup in der Tabelle
5. Wählt nächsten Router aus
6. Bis Paket beim Zielhost ankommt

Frage: wie werden diese Weiterleitungstabellen erstellt? (→ Kapitel 4)

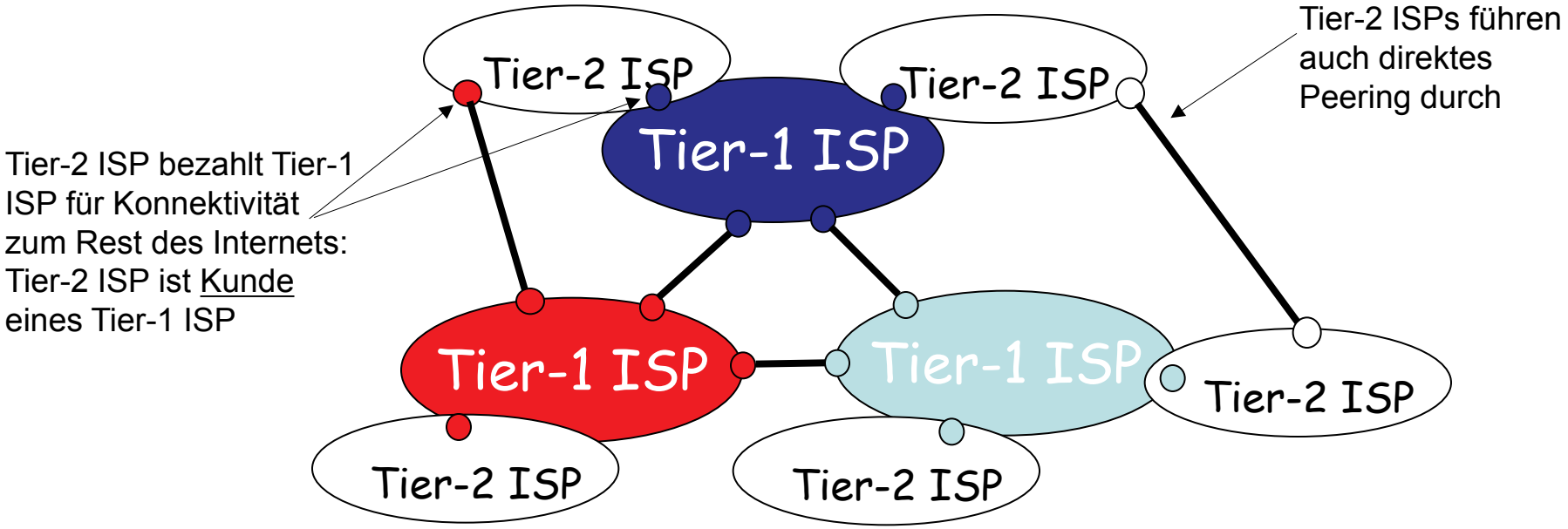
1.3.3 ISPs und Internet-Backbones

- Internet als „Netzwerk von Netzwerken“
- Grob hierarchisch
- Im Zentrum: “Tier-1” Internet Service Providers (ISPs)
  - „Internet Backbones“
  - Behandeln sich als gleichberechtigte Partner
  - Sind vollständig miteinander verbunden, sind mit vielen Tier-2 Netzen verbunden
  - Viele parallele Leitungen von 40 Gbit/s oder gar 100 Gbit/s
  - Sprint, Verizon, Quest, AT&T, Level3, usw.



1.3.3 ISPs und Internet-Backbones

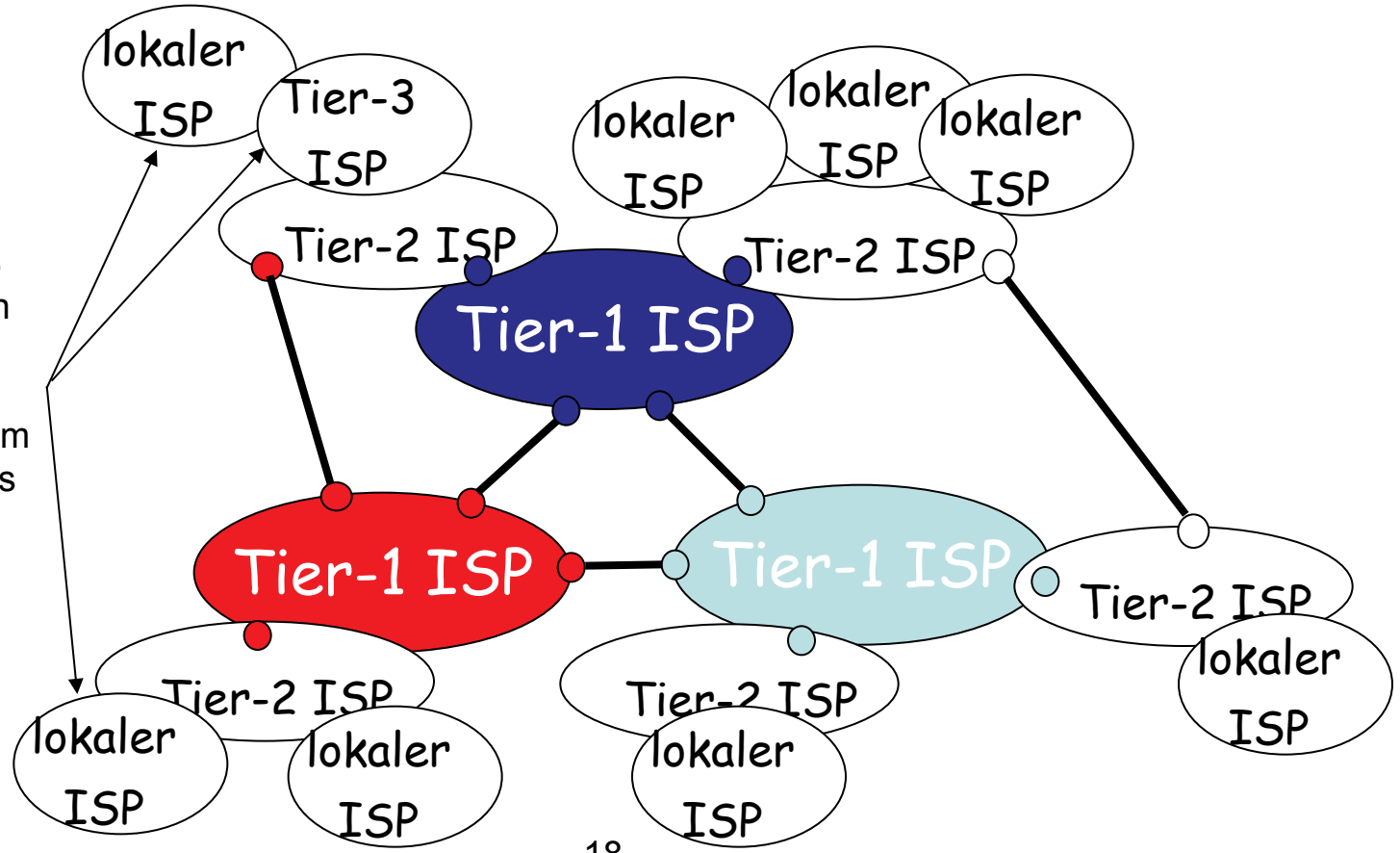
- "Tier-2" ISPs: kleinere, oft nationale oder regionale ISPs
  - Sind mit einem / mehreren Tier-1 ISPs verbunden, oft auch mit anderen Tier-2 ISPs
  - Sind **Kunden** von Tier-1 Netzen (zahlen für Leitung und Upstream Verkehr)
  - Manche Tier-2 ISPs sind auch Tier-1 ISPs (vertikale Integration)



1.3.3 ISPs und Internet-Backbones

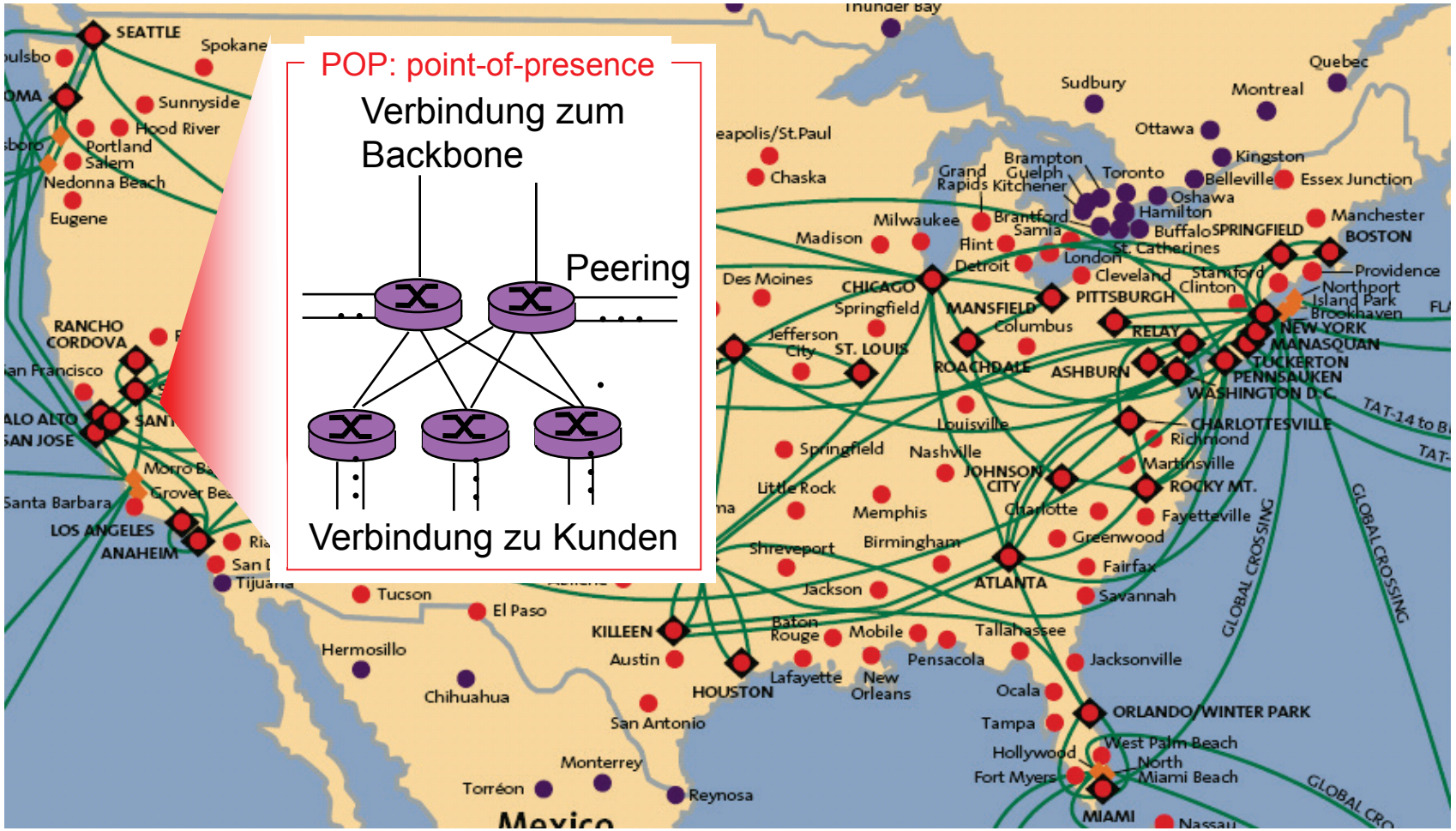
- "Tier-3" ISPs und lokale ISPs
  - Zugangsnetzwerke (last hop, access network)

Lokale und Tier-3 ISPs sind Kunden von Tier-1 und Tier-2 ISPs, welche sie mit dem Rest des Internets verbinden



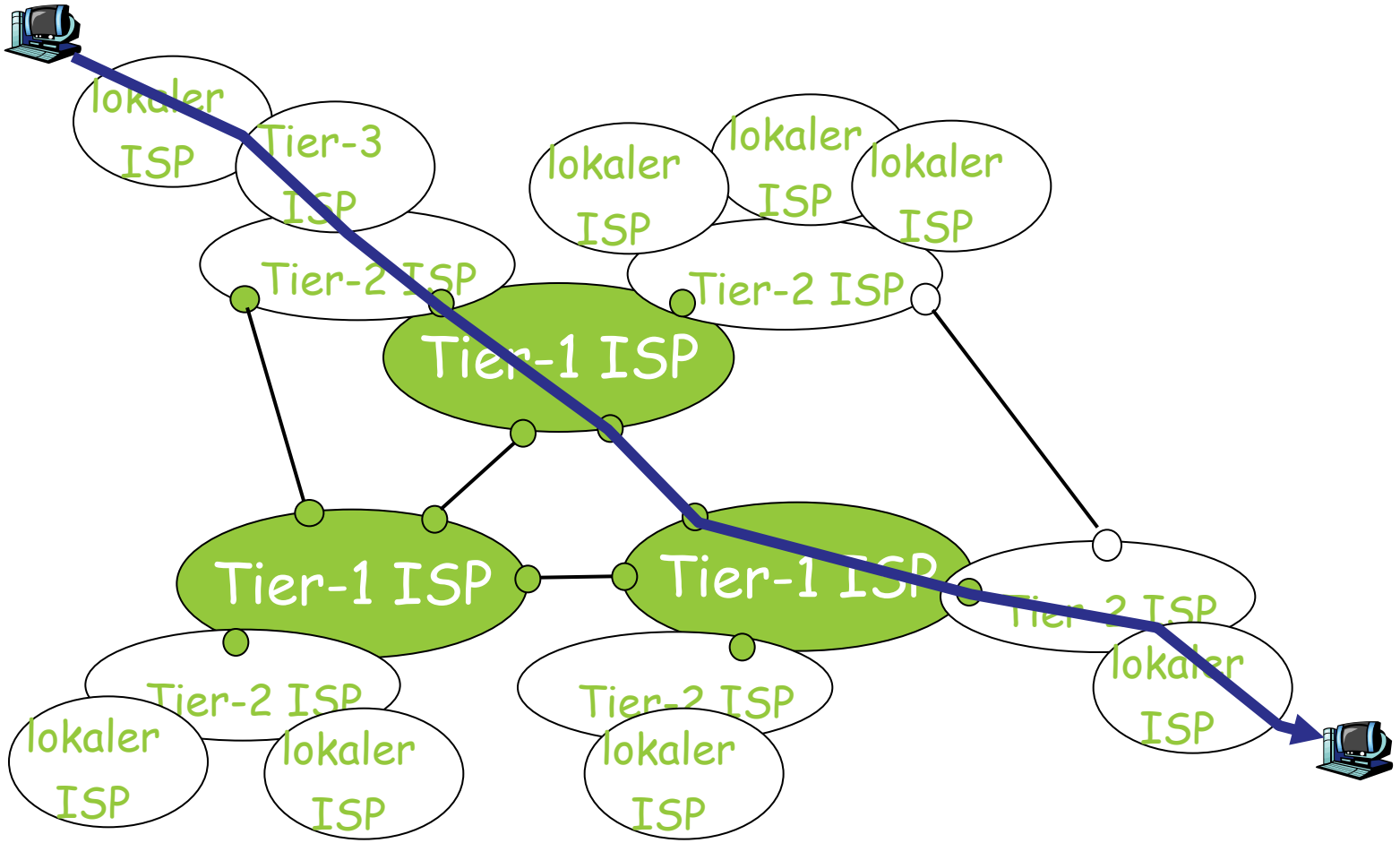
1.3.3 Point of Presence (POP)

- Ort an dem Router aufgestellt sind



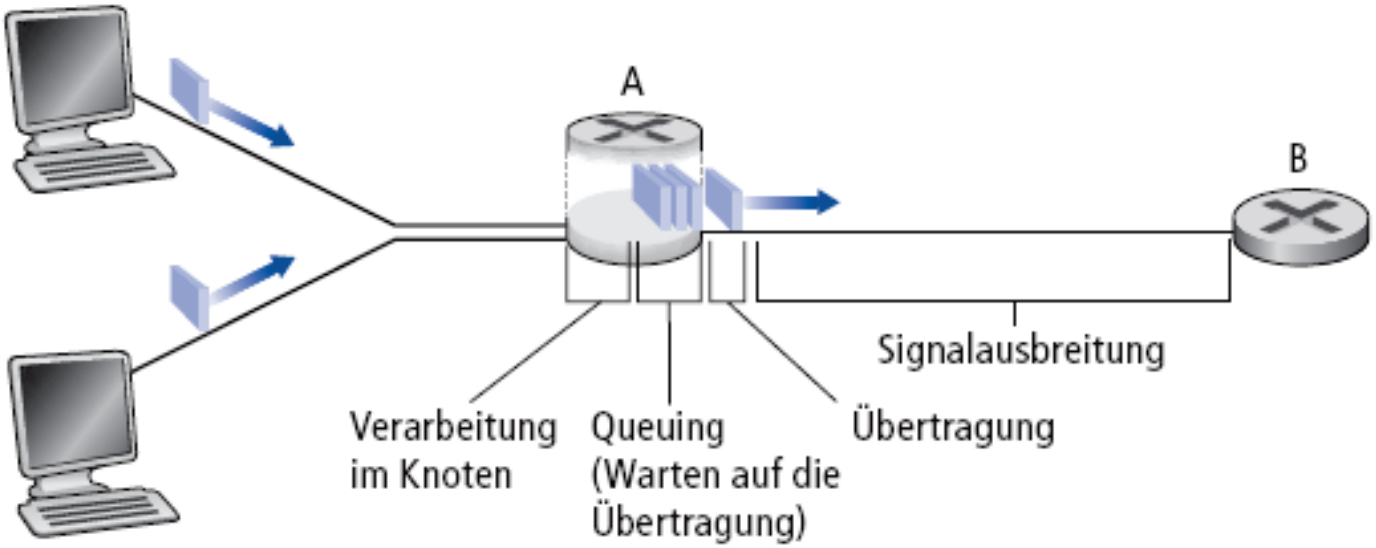
1.3.3 ISPs und Internet-Backbones

- Ein Paket durchquert viele Netzwerke!



1.4 Verzögerung, Verlust und Durchsatz in paketvermittelten Netzen

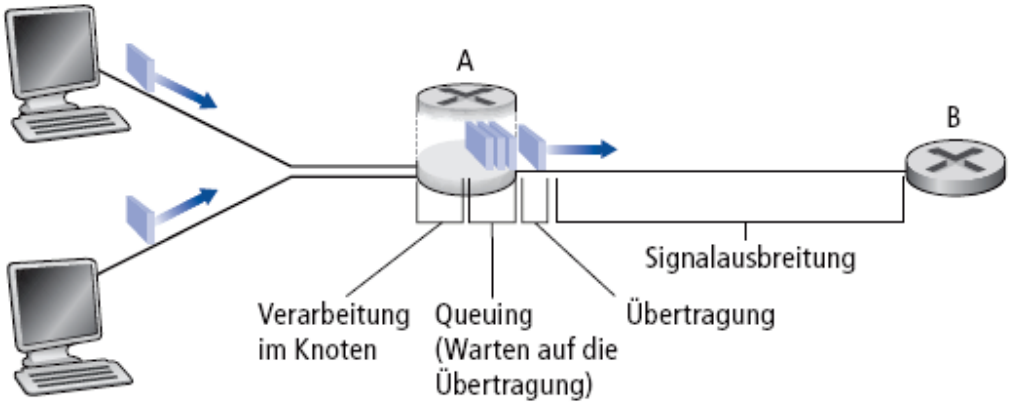
1.4 Verzögerung, Verlust und Durchsatz in paketvermittelten Netzen



Wie entstehen Paketverluste und Verzögerungen?
→ Pakete warten in den Puffern von Routern wenn die Ankunftsrate die Kapazität der Ausgangsleitungen übersteigt.

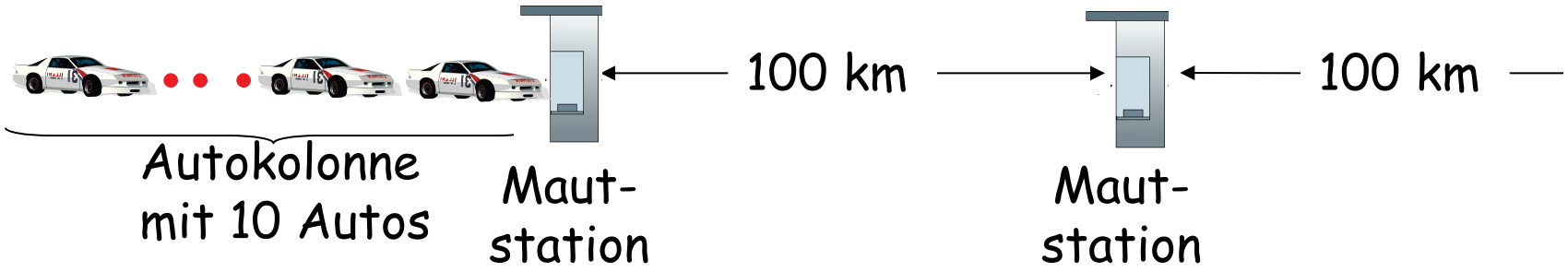
→ Ist die Warteschlange vor einer Leitung voll verwirft der Router ankommende Pakete (da er keinen Platz hat um sie zu speichern), d.h. sie gehen verloren.

1.4.1 Arten der Verzögerung



1. Verarbeitung im Knoten:
   - Auf Bitfehler prüfen
   - Wahl der ausgehenden Leitung
2. Warten auf die Übertragung:
   - Wartezeit, bis das Paket auf die Ausgangsleitung gelegt werden kann
   - Hängt von der Warteschlangenlänge auf (bzw. “vor”) der Ausgangsleitung ab
3. Übertragungsverzögerung:
   - Wenn R = Bandbreite einer Leitung (Bit/s) und L = Paketgröße (Bit), dann ist die Übertragungsverzögerung = L/R
4. Ausbreitungsverzögerung:
   - Wenn **d** = Länge der Leitung und **s** = Ausbreitungsgeschwindigkeit des Mediums (~2×10^8 m/s in optischen Glasfasern), dann ist die Ausbreitungsverzögerung **d/s**

1.4.1 Verhalten wie auf einer Autobahn



1.4.1 Verzögerung, Verlust und Durchsatz in paketvermittelten Netzen

Gesamtverzögerung:

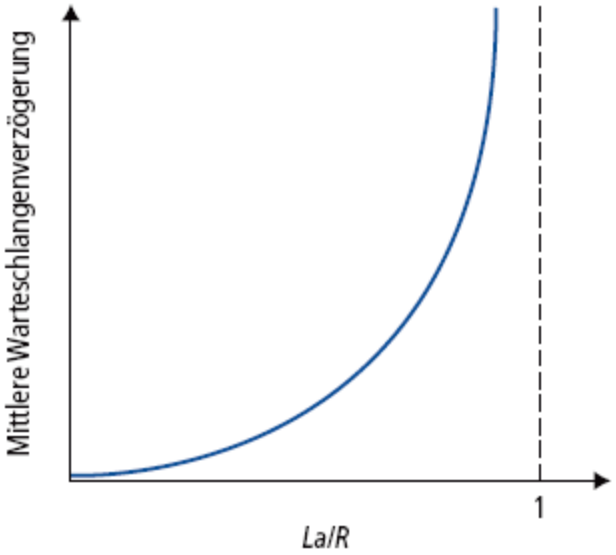
$$d_{\text{gesamt}} = d_{\text{Verarbeitung}} + d_{\text{Warten}} + d_{\text{Übertragung}} + d_{\text{Ausbreitung}}$$

- $d_{\text{Verarbeitung}}$ = Verarbeitungsverzögerung
  - Üblicherweise wenige Mikrosekunden oder weniger
- d_{Warten} = Wartezeit in Puffern
  - Abhängig von der aktuellen Überlastsituation
- $d_{\text{Übertragung}}$ = Übertragungsverzögerung
  - = L/R, signifikant wenn R klein ist
- $d_{\text{Ausbreitung}}$ = Ausbreitungsverzögerung
  - Wenige Mikrosekunden bis einige hundert Millisekunden

1.4.2 Warteschlangenverzögerung

- R = Bandbreite (Bit/s)
- L = Paketgröße (Bit)
- a = durchschnittliche Paketankunftsrate

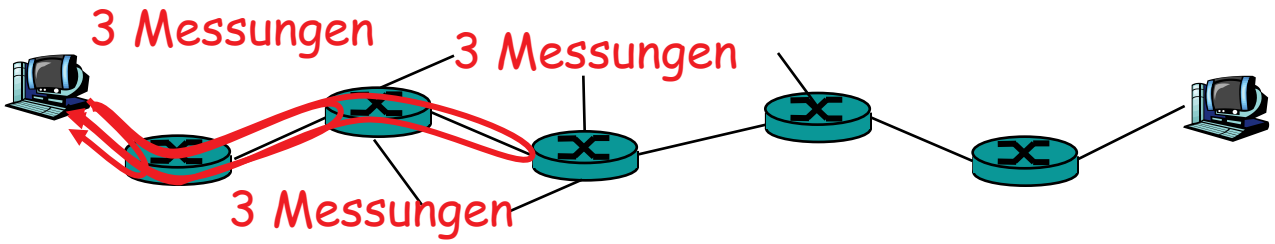
Verkehrswert (Last, Load) = L * a / R



- La/R ~ 0: Wartezeit gering
- La/R → 1: Wartezeit steigt stark an
- La/R > 1: durchschnittliche Wartezeit ist unendlich!

→ Warteschlangentheorie

1.4.3 Ende-zu-Ende-Verzögerung

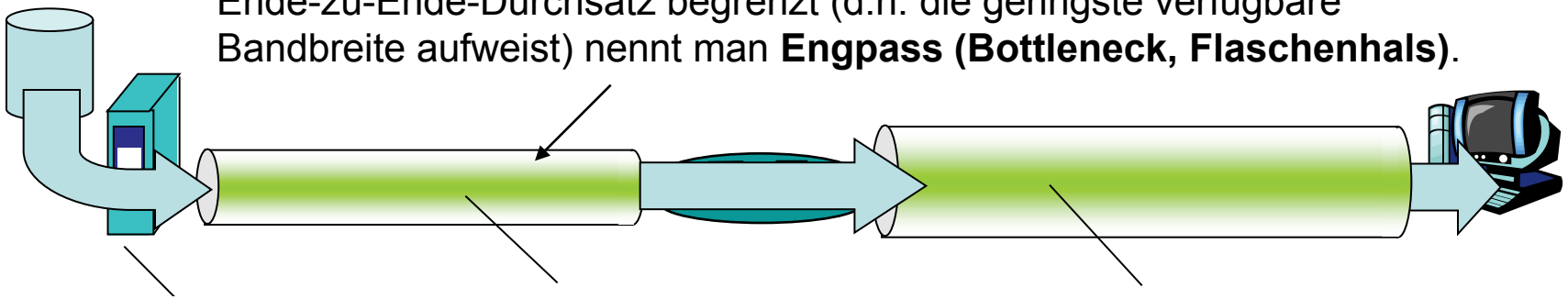


- **`Traceroute`**: Misst die Verzögerung von einer Quelle zu allen Routern auf dem Weg zu einem Ziel. Für alle Router *i*:
  - Sende drei Pakete, die *i* auf dem Pfad zum Ziel erreichen
  - Router *i* schickt als Reaktion Pakete an den Sender
  - Sender misst die Zeit zwischen Senden des eigenen Paketes und Empfang des Paketes vom Router

1.4.4 Durchsatz in Computernetzwerken

- *Durchsatz:* Rate (Bit/Zeiteinheit), mit der Daten zwischen Sender und Empfänger ausgetauscht werden
  - *Unmittelbar:* Rate zu einem gegebenen Zeitpunkt
  - *Durchschnittlich:* Rate über einen längeren Zeitraum
- Bits als Wasser vorstellbar und Kommunikationsleitungen als Rohre:

Die Leitung auf dem Ende-zu-Ende-Pfad, welche den Ende-zu-Ende-Durchsatz begrenzt (d.h. die geringste verfügbare Bandbreite aufweist) nennt man **Engpass (Bottleneck, Flaschenhals)**.



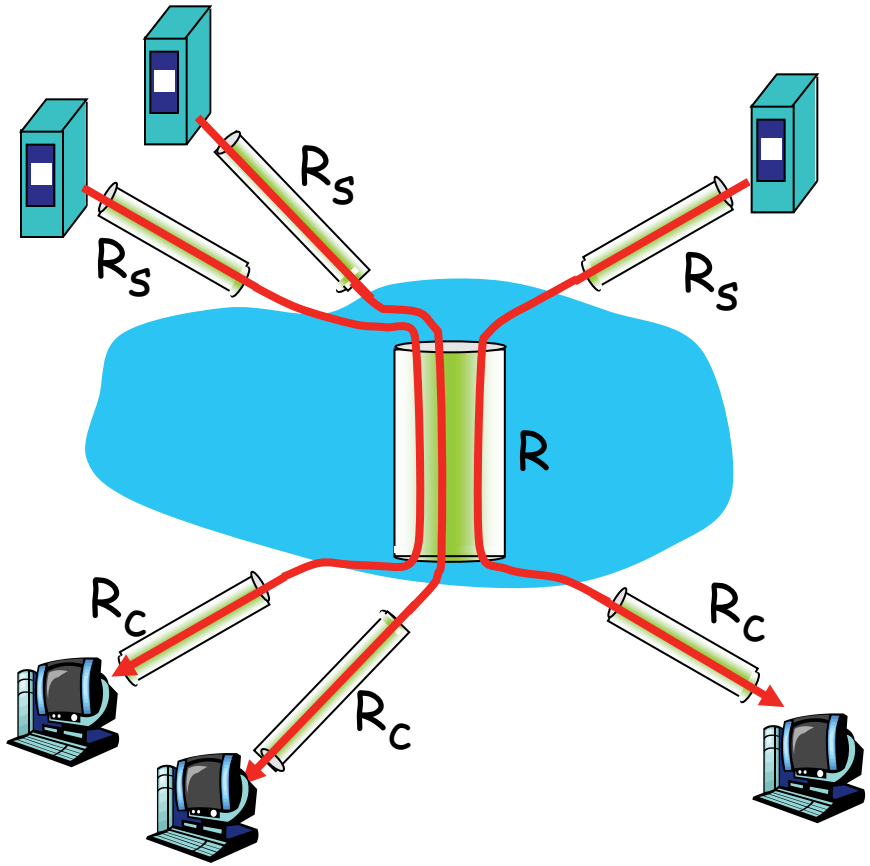
Server sendet Bits (Wasser) in die Leitung

Leitung, die Wasser mit der Rate R_s Bit/s weiterleiten kann

Leitung, die Wasser mit der Rate R_c Bit/s weiterleiten kann

1.4.4 Durchsatz in Computernetzwerken

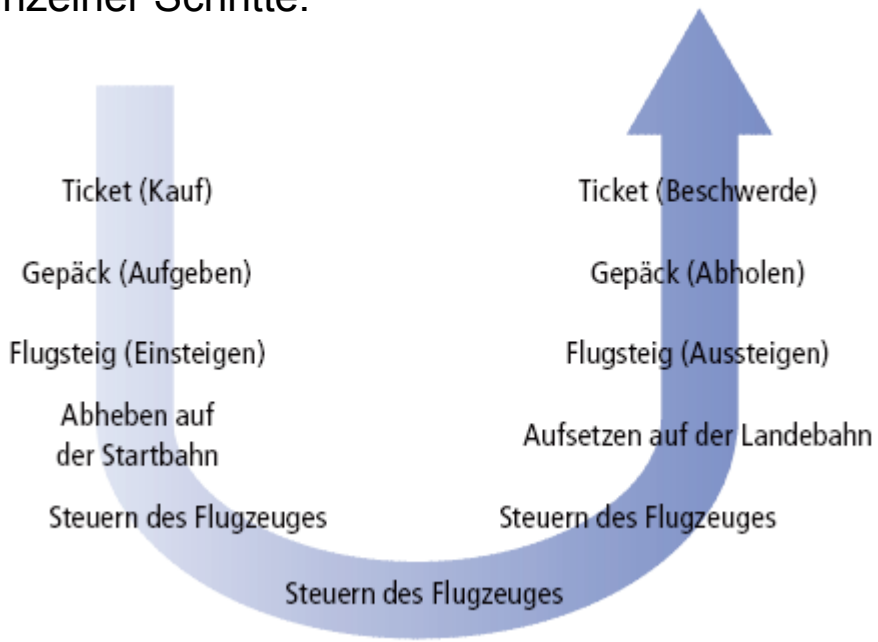
- Im Internet teilen sich Verbindungen den Engpass des Backbone-Netzwerkes
- Aber es hat sich gezeigt: Häufig sind R_c oder R_s die Engpässe!



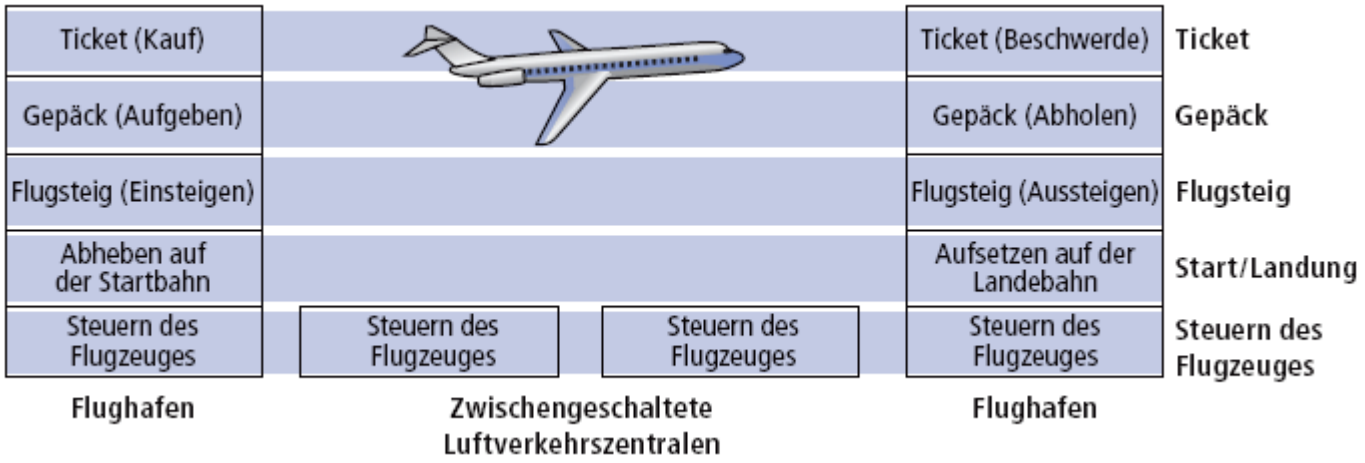
1.5 Protokollschichten und ihre Dienstmodelle

1.5 Protokollschichten und ihre Dienstmodelle

- Das Internet stellt ein äußerst komplexes System dar
  → Schwierig zu strukturieren
- Die Internetarchitektur wird in Schichten gegliedert
- Zum besseren Verständnis die Analogie Luftverkehrssystem
  → Eine Folge einzelner Schritte:



1.5 Protokollschichten und ihre Dienstmodelle



Schichten: Jede Schicht implementiert einen Dienst

- Mit Hilfe von schichtinternen Aktionen
- Unter Verwendung von Diensten der Schicht, die unter ihr liegt

1.5 Protokollschichten und ihre Dienstmodelle

Schichten bieten folgende Vorteile beim Umgang mit komplexen Systemen:

- Strukturierung ermöglicht die Identifikation und das Verständnis des Zusammenspiels einzelner Bestandteile des Systems
  - Referenzmodell für die Diskussion des Systems
- Modularisierung vereinfacht die Wartung und das Arbeiten mit dem System:
  - Änderungen an der Implementierung einer Schicht sind transparent für den Rest des Systems

    Beispiel: Eine Veränderung der Einsteigeprozedur am Gate beeinflusst nicht den Rest des Systems.

1.5 Internet Schichten

Protokollstapel des Internets:

- **Anwendungsschicht**: Unterstützung von Netzwerkanwendungen
  - FTP, SMTP, HTTP
- **Transportschicht**: Datentransfer zwischen Prozessen (auf Betriebssystemebene)
  - TCP, UDP
- **Netzwerkschicht** (auch Vermittlungsschicht): Weiterleiten der Daten von einem Sender zu einem Empfänger
  - IP, Routing-Protokolle
- **Sicherungsschicht**: Datentransfer zwischen benachbarten Netzwerksystemen
  - PPP, Ethernet, WLAN
- **Bitübertragungsschicht**: Bits auf der Leitung

1.5.1 ISO/OSI Referenzmodell

Zwei zusätzliche Schichten im ISO/OSI Modell:

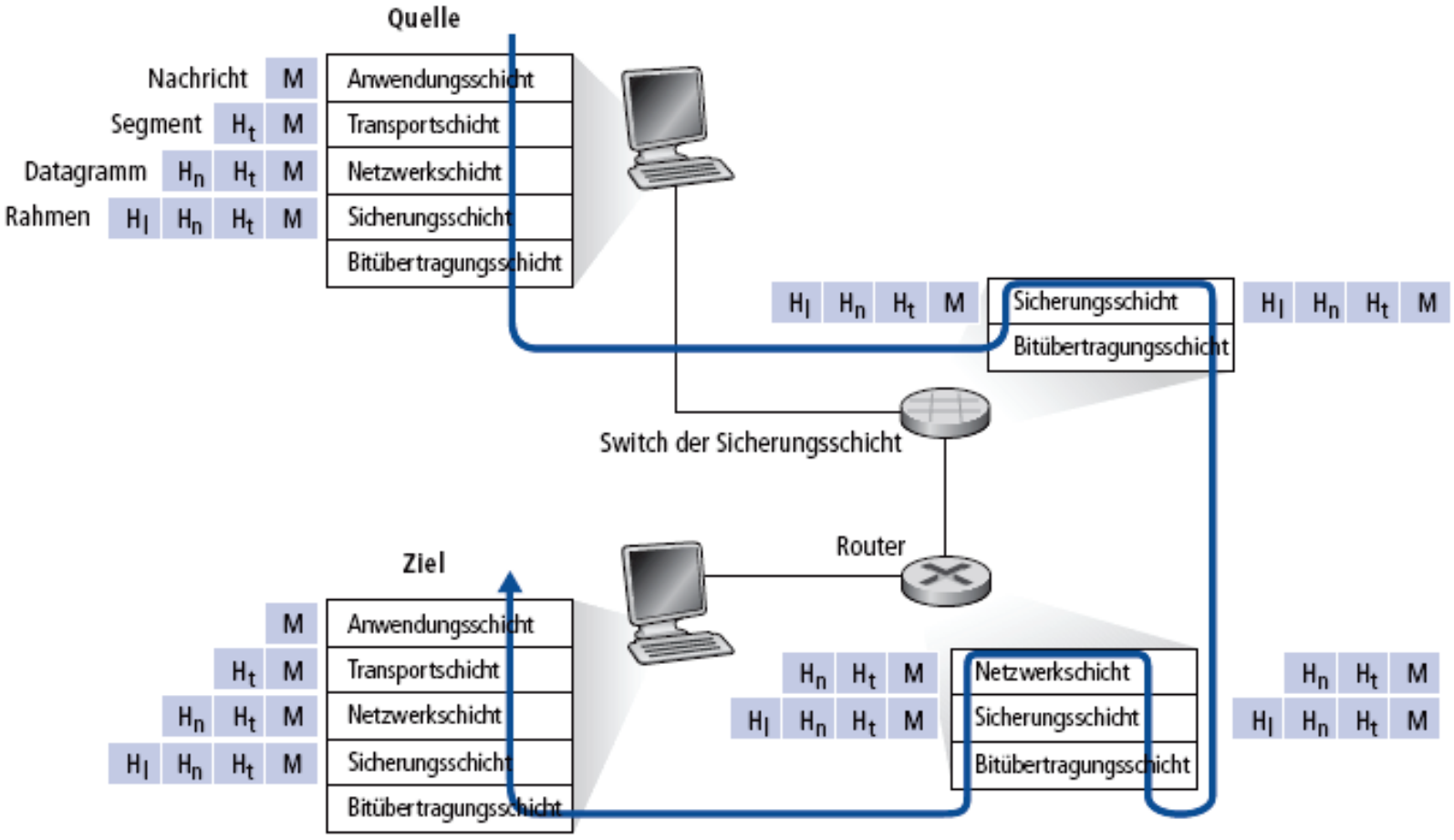
- **Darstellungsschicht**: Ermöglicht es Anwendungen, die Bedeutung von Daten zu interpretieren, z.B. Verschlüsselung, Kompression, Vermeidung systemspezifischer Datendarstellung
- **Kommunikationssteuerungsschicht***:* Synchronisation, Setzen von Wiederherstellungspunkten

Der Protokollstapel des Internets bietet diese Funktionalitäten nicht!

→ Wenn benötigt, müssen sie von der Anwendung implementiert werden

→ Aber werden sie wirklich benötigt?

1.5 Protokollschichten und ihre Dienstmodelle



Weg, den Daten durch den Protokollstapel eines sendenden Endsystems nehmen.

1.6 Sicherheit von Netzwerken

1.6 Sicherheit von Netzwerken

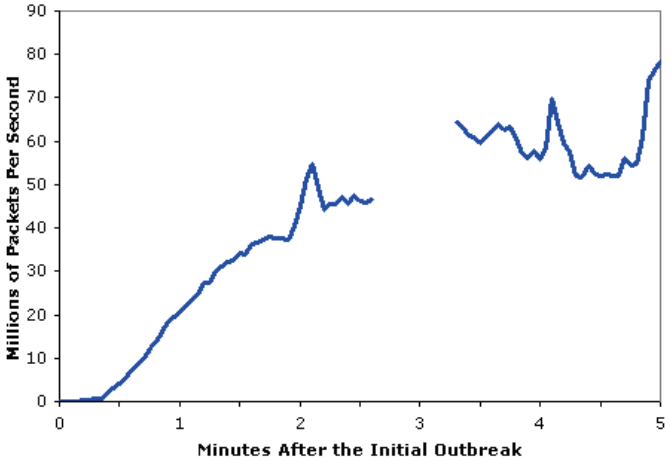
- *Auf welche Art sind Computernetzwerke angreifbar?*
- *Wie kann man Computernetzwerke gegen Angriffe schützen?*
- *Können Architekturen entwickelt werden, die gegen Angriffe immun sind?*

- Das Internet wurde nicht mit dem Ziel Sicherheit entworfen
  - Vision: *"Eine Gruppe von Benutzern, die sich gegenseitig vertrauen, sind über ein transparentes Netzwerk miteinander verbunden"*
  - Die Entwickler von Internetprotokollen versuchen Sicherheit **nachträglich** einzubauen
  - Inzwischen: Sicherheit wird in allen Protokollschichten untersucht!

1.6 Sicherheit von Netzwerken

- Angriffe auf die Infrastruktur des Internets
  - Kompromittieren/Angreifen von Endsystemen:
    z.B. *Malware, Spyware, Würmer, unberechtigter Zugriff (Diebstahl von Daten und Accounts)*
  - Denial of Service: Den Zugang zu Ressourcen verhindern

1.6 Sicherheit von Netzwerken



Sapphire-Wurm: scans/s in den ersten 5 Minuten des Ausbruchs (Daten von CAIDA, UWisc)

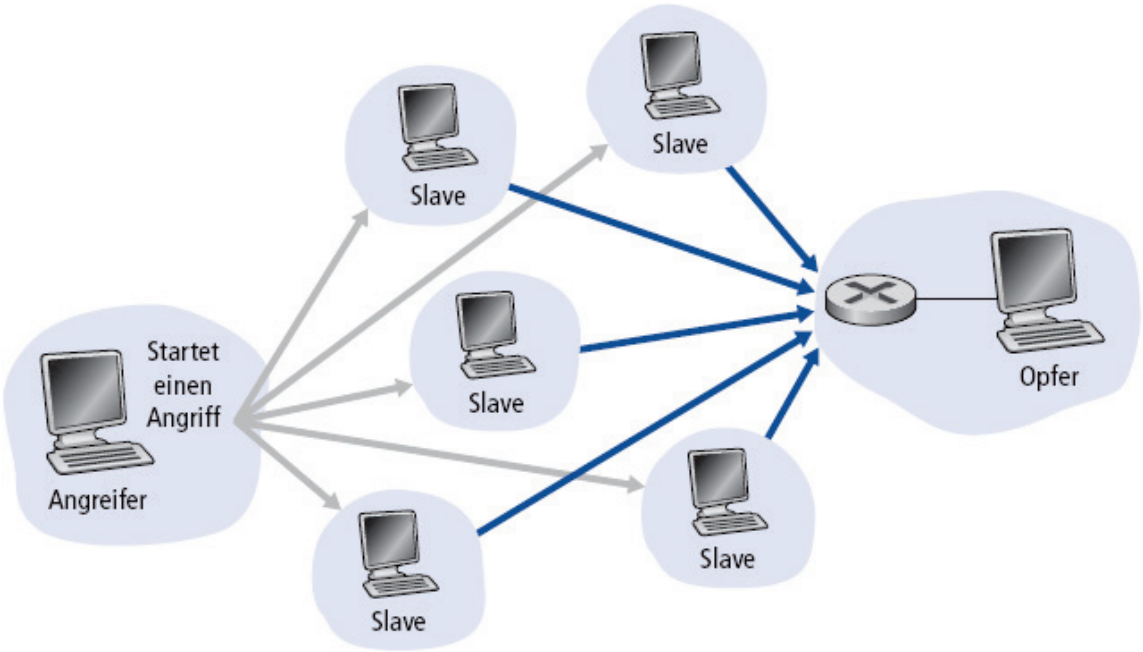
Malware:

- Virus
  - Infektionen über empfangene Objekte (z.B. per E-Mail), erfolgen aktiv
  - Selbst replizierende Viren verbreiten sich über weitere Endsysteme und Benutzer
- Würmer
  - Infektion durch Objekte, die ohne Benutzereingriff empfangen wurden
  - Selbst replizierende Würmer verbreiten sich über weitere Endsysteme und Benutzer
- Spyware
  - Infektion oft durch die Installation von anscheinend harmlosen Programmen
  - Aufzeichnen und Weitermelden von Tastenanschlägen, besuchten Websites, usw.

1.6 Sicherheit von Netzwerken

Denial-of-Service-Angriff (DoS Angriff):

- Angreifer verhindern den Zugriff von Benutzern auf Ressourcen (Server, Bandbreite), indem diese durch den Angreifer belegt werden



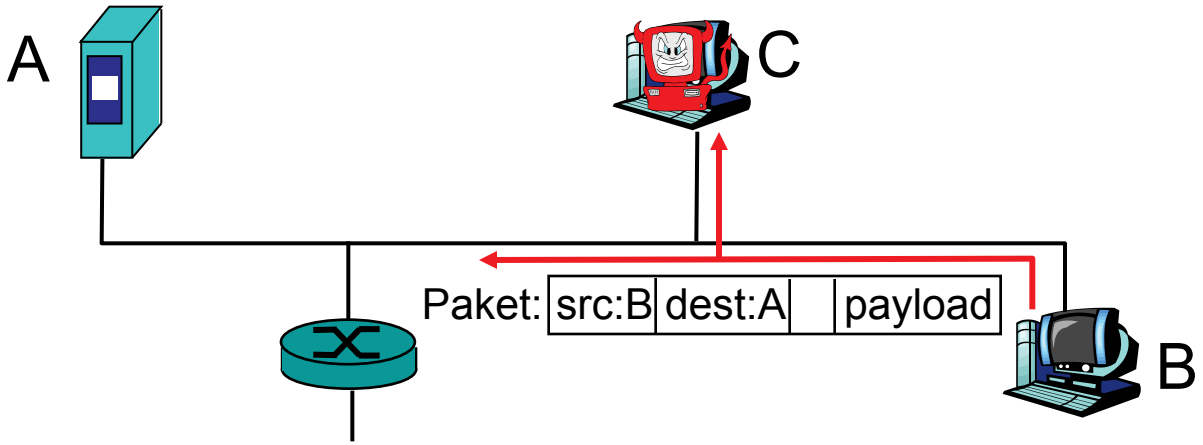
Vorgehensweise:

1. Wähle ein Ziel
2. Kompromittiere andere Systeme (z.B. durch Malware)
3. Sende eine sehr große Anzahl an Paketen von den kompromittierten Systemen an das Ziel

1.6 Sicherheit von Netzwerken

Mithören, Verändern und Löschen von Paketen:

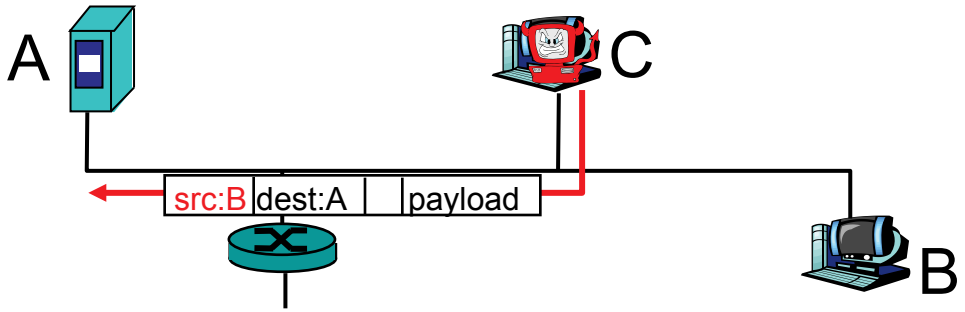
- **Mithören von Paketen, Man in the Middle:**
  - Broadcast-Medien (Ethernet, WLAN)
  - Netzwerkkarten im Promiscuous Mode zeichnen alle (!) Pakete auf, die sie hören können



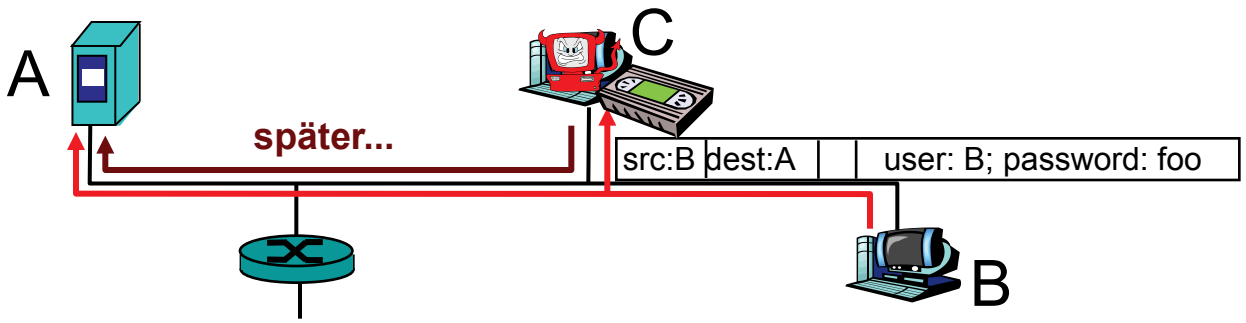
1.6 Sicherheit von Netzwerken

Eigene Identität fälschen:

- ***IP-Spoofing:*** Sende Pakete mit falscher Absenderadresse



- ***Aufzeichnen und Abspielen:*** Sicherheitsrelevante Informationen (z.B. Passwort) mithören und später verwenden
  - Das System hält jemanden, der das Passwort eines Benutzers kennt, für den Benutzer!

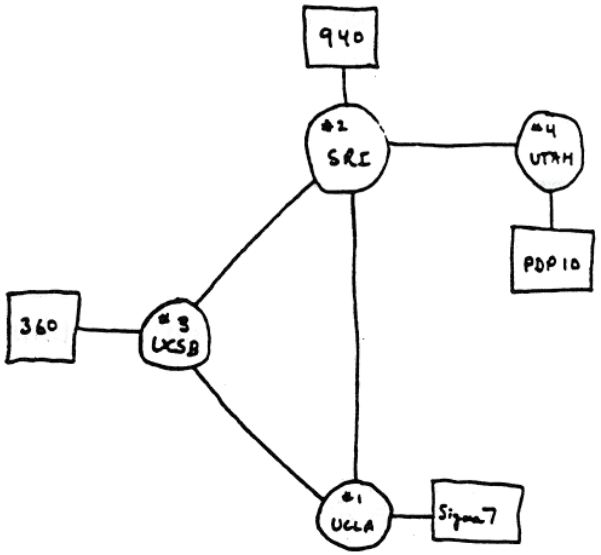


1.7 Geschichte der Computernetzwerke

1.7 Geschichte der Computernetzwerke

Geschichte des Internets: *1961–1972: Paketvermittlung*

- 1961: Kleinrock – Warteschlangentheorie zeigt die Effizienz der Paketvermittlung
- 1964: Baran – Paketvermittlung in Militärnetzen
- 1967: ARPAnet von der Advanced Research Projects Agency geplant
- 1969: erster ARPAnet-Knoten in Betrieb
- 1972:
  - ARPAnet, öffentliche Vorführung
  - NCP (Network Control Protocol), erstes Protokoll zwischen Hosts
  - Erstes E-Mail-Programm
  - ARPAnet hat 15 Knoten



THE ARPA NETWORK

1.7 Geschichte der Computernetzwerke

Geschichte des Internets: *1972–1980: Netzwerk von Netzwerken*

- 1970: ALOHAnet Satellitennetzwerk auf Hawaii
- 1974: Cerf und Kahn – Architektur für die Verbindung von Netzwerken
- 1976: Ethernet: Xerox PARC
- Späte 1970er: proprietäre Architekturen: DECnet, SNA, XNA
- Späte 1970er: Pakete fester Größe (später ATM)
- 1979: ARPAnet hat jetzt 200 Knoten

Cerf und Kahn: Prinzipien für die Verbindung von Netzen

- Minimalismus, Autonomie – keine internen Änderungen an den einzelnen Netzwerken
- Best-Effort-Dienst – keine Garantien
- Kein (Verbindungs-) Zustand in den Routern
- Dezentrale Kontrolle

→ Definition der aktuellen Internetarchitektur!

1.7 Geschichte der Computernetzwerke

Geschichte des Internets: *1980–1990: Neue Protokolle, Ausbreitung des Netzes*

- 1983: Einführung von TCP/IP
- 1982: Definition des SMTP-E-Mail-Protokolls
- 1983: Definition von DNS zur Übersetzung von Namen auf IP-Adressen
- 1985: Definition von ftp
- 1988: Überlastkontrolle in TCP

- Neue nationale Netzwerke: *Csnet, BITnet, NSFnet, Minitel*
- 100.000 Endsysteme sind an einen Verbund von Netzwerken angeschlossen

1.7 Geschichte der Computernetzwerke

Geschichte des Internets: *1990–20XX: Kommerzialisierung, WWW*

- Anfang 1990er Jahre: ARPAnet wird eingestellt
- 1991: NSF hebt die Einschränkungen bezüglich der kommerziellen Nutzung des NSFnet auf
- Anfang 1990er Jahre: Web
  - Hypertext [Bush 1945, Nelson 1960er]
  - HTML, HTTP: Berners-Lee
  - 1994: Mosaic, später Netscape
  - Späte 1990er Jahre: Kommerzialisierung des Web

Späte 1990er–20XX: Neue Anwendungen

- Mehr Killeranwendungen: Instant Messaging, P2P-Filesharing
- Netzwerksicherheit wird immer wichtiger
- Geschätzte 50 Millionen Endsysteme, 100 Millionen Anwender
- Backbone-Leitungen mit Gbit/s

1.7 Geschichte der Computernetzwerke

Geschichte des Internets: *2005-Heute*

- Inzwischen ~900 Millionen Hosts (inklusive Smartphones und Tablets)
- Breitband Zugang weit verbreitet
- Highspeed W-Lan weit verbreitet
- Aufkommen sozialer Netzwerke → Facebook hat über eine Milliarde Nutzer
- Service Provider (Google, Microsoft) erschaffen ihre eigenen Netzwerke
  - Ermöglicht dem Nutzer sofortigen Zugang zu Email, Suche etc.
- E-Commerce, Universitäten und Firmen nutzen Cloudservices (z.B. Amazon EC2)

Kapitel 2: Anwendungsschicht

2. Anwendungsschicht

- 2.1 Grundlagen der Netzwerkanwendungen
- 2.2 Das Web und HTTP
- 2.3 Dateitransfer: FTP
- 2.4 E-Mail im Internet
- 2.5 DNS – der Verzeichnisdienst des Internets
- 2.6 Peer-to-Peer-Anwendungen
- 2.7 Socket-Programmierung mit UDP und TCP

2.1 Grundlagen der Netzwerkanwendungen

2.1 Grundlagen der Netzwerkanwendungen

Entwicklung von Netzanwendungen

Programme, die

- auf mehreren (verschiedenen) Endsystemen laufen
- über das Netzwerk kommunizieren
  - Beispiel: Die Software eines Webservers kommuniziert mit dem Browser

Praktisch keine Software für das Innere des Netzwerkes

- Im Inneren des Netzwerkes werden keine Anwendungen ausgeführt
- Die Konzentration auf Endsysteme erlaubt eine schnelle Entwicklung und Verbreitung der Software

Beispiele für Netzanwendungen:

- E-Mail
- Web
- Instant Messaging
- Terminalfernzugriff
- P2P-Filesharing
- Netzwerkspiele
- Streaming von Videoclips
- Voice over IP (VoIP)
- Videokonferenzen
- Grid Computing

2.1.1 Architektur von Netzwerkanwendungen

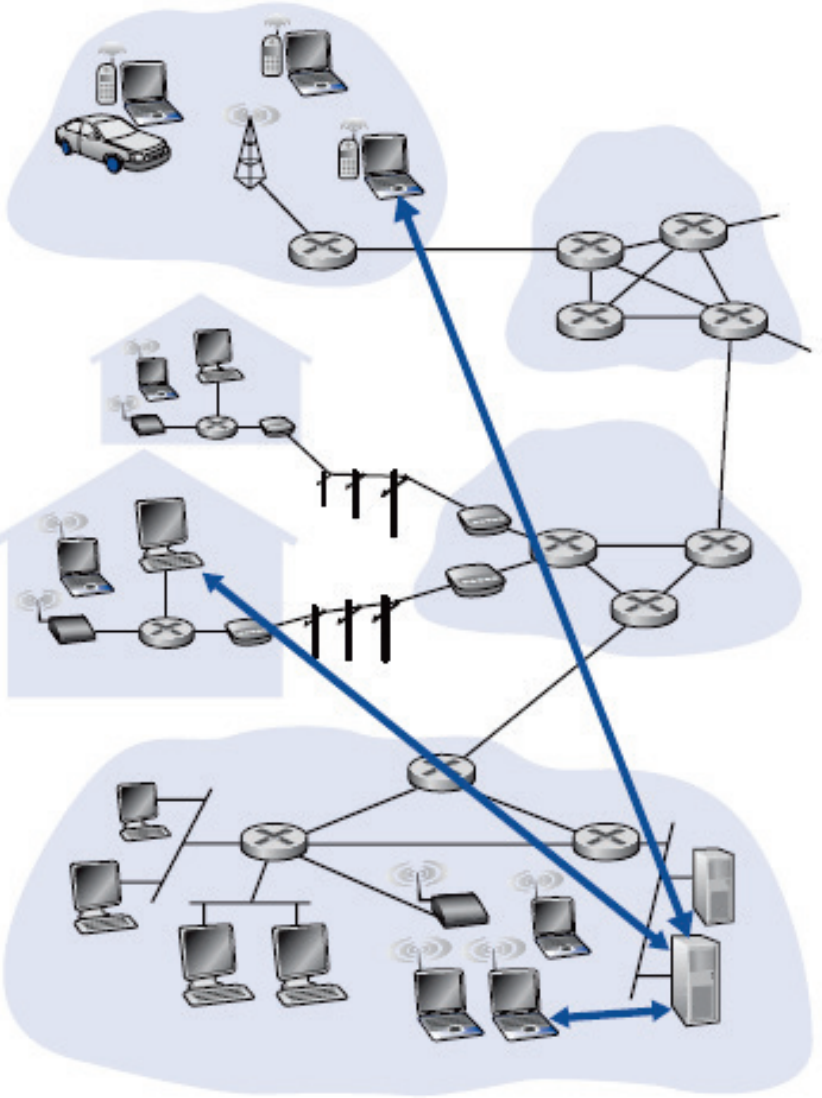
Client-Server-Architektur

Server
- Bearbeitet Anfragen von Clients
- Immer eingeschaltet
- Feste IP-Adresse
- Serverfarmen, um zu skalieren

Clients
- Kommunizieren mit Servern
- Permanent oder nur manchmal online
- Können dynamische IP-Adressen haben
- Kommunizieren nicht direkt miteinander

Beispiele für bekannte Anwendungen mit Client-Server-Architektur:

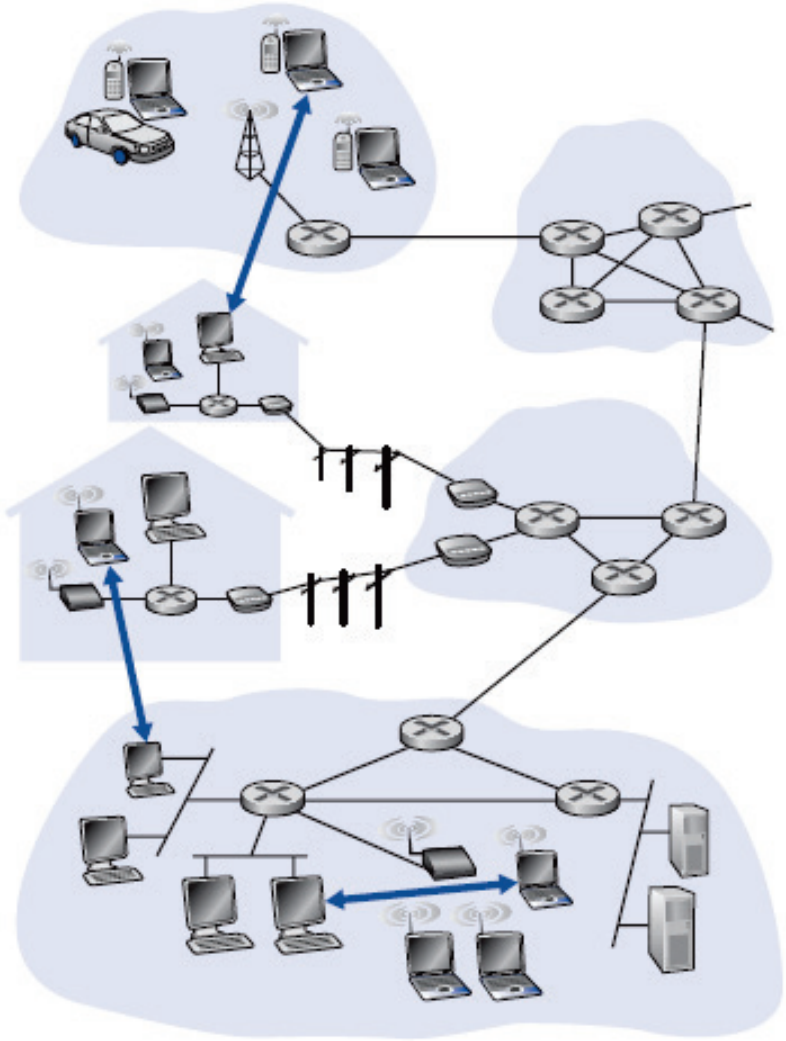
- Das Web
- FTP
- Telnet
- E-Mail



2.1.1 Architektur von Netzwerkanwendungen

Reine Peer-to-Peer-Architektur (P2P)

- **Keine** Server (kostengünstig)
- Beliebige Endsysteme kommunizieren direkt miteinander
- Peers sind nur sporadisch angeschlossen und wechseln ihre IP-Adresse
- Selbstskalierbarkeit (siehe Buch!), aber schwer zu warten und zu kontrollieren!



Beispiele für bekannte Anwendungen mit P2P-Architektur:

- File-Distribution (z.B. BitTorrent)
- Filesharing (z.B. eMule, LimeWire)
- IPTV (z.B. PPLive)

2.1.1 Architektur von Netzwerkanwendungen

Kombination von Client-Server und P2P

Skype - Voice-over-IP Anwendung

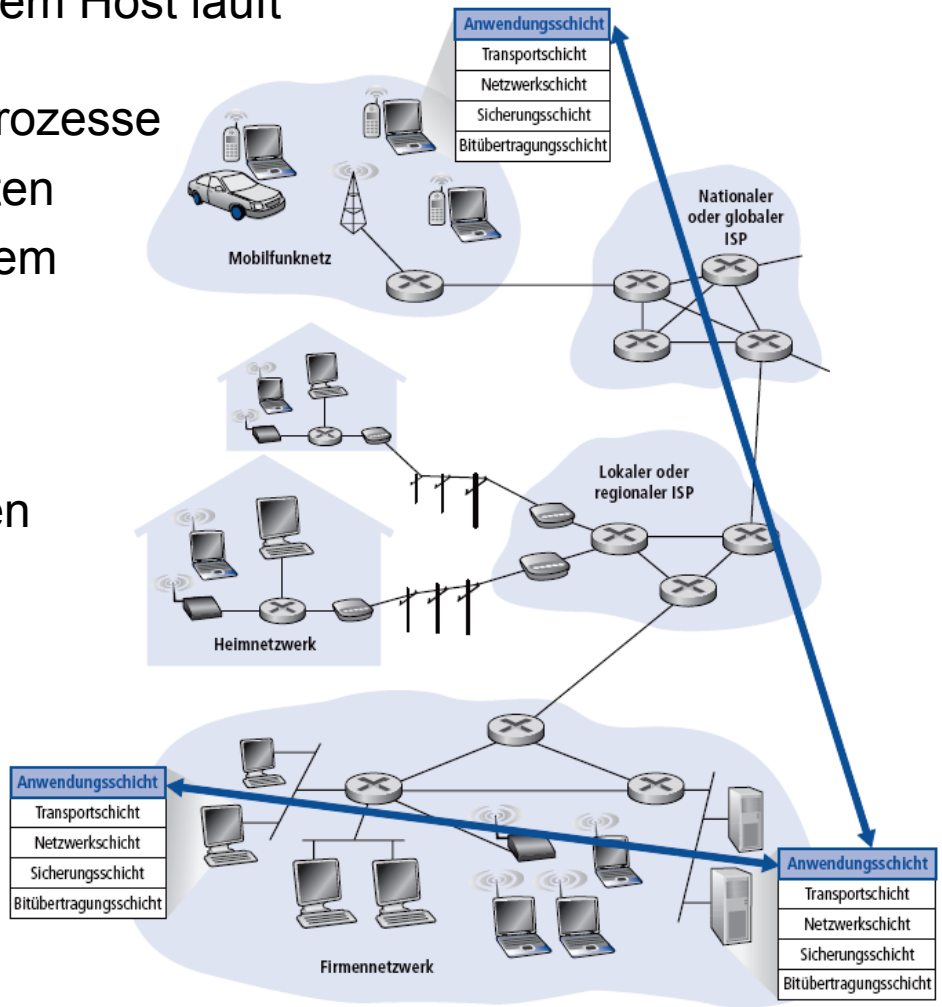
- Zentraler Server: Adresse des Kommunikationspartners finden
- Verbindung zwischen Clients: direkt (P2P)

Instant Messaging

- Chat zwischen zwei Benutzern: P2P
- Zentralisierte Dienste: Erkennen von Anwesenheit, Zustand, Aufenthaltsort eines Anwenders
- Benutzer registriert seine IP-Adresse beim Server, sobald er sich mit dem Netz verbindet
- Benutzer fragt beim Server nach Informationen über seine Freunde und Bekannten

2.1.2 Kommunikation zwischen Prozessen

- **Prozess:** Programm, welches auf einem Host läuft
- Innerhalb eines Hosts können zwei Prozesse mit Inter-Prozess-Kommunikation Daten austauschen (durch das Betriebssystem unterstützt)
- Prozesse auf verschiedenen Hosts kommunizieren, indem sie Nachrichten über ein Netzwerk austauschen



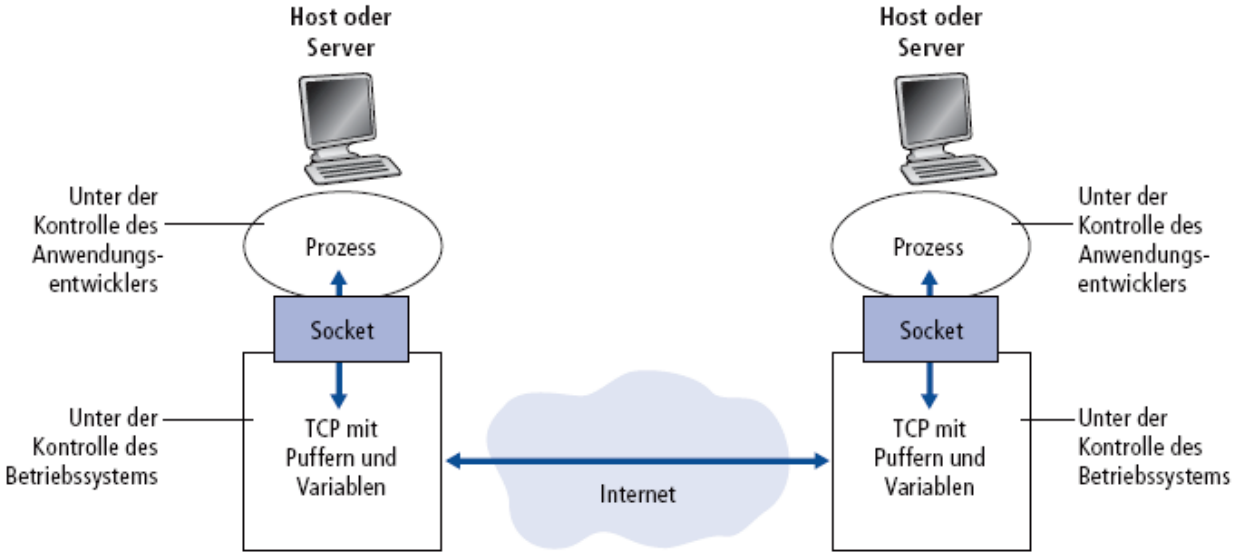
2.1.2 Client- und Server-Prozesse

Eine Netzanwendung besteht aus Prozesspaaren, die einander Nachrichten über ein Netzwerk zusenden:

- Client-Prozess: Definiert als Prozess, der die Kommunikation beginnt
- Server-Prozess: Definiert als Prozess, der darauf wartet, kontaktiert zu werden

- P2P: Enthält sowohl Client- als auch Server-Prozesse

2.1.2 Sockets



- Prozesse senden/empfangen Nachrichten über einen **Socket**
- **Schnittstelle** zwischen der Anwendungsschicht und der Transportschicht = Anwendungsprogrammierschnittstelle (API)
- Analogie *Tür zu einem Haus* (entspricht Socket zu einem Prozess):
  - Der sendende Prozess schiebt die Nachrichten durch seine Tür raus
  - Der sendende Prozess verlässt sich auf die Transportinfrastruktur auf der anderen Seite der Tür, um die Nachricht zum Socket des empfangenden Prozesses zu bringen
  - Sobald die Nachricht beim Zielhost ankommt, tritt sie durch die Tür des empfangenden Prozesses, der danach auf die Nachricht reagiert

2.1.2 Addressierung von Prozessen

- Um eine Nachricht empfangen zu können, muss ein Prozess identifiziert werden können

- Prozesse werden durch die (bei IPv4 32 Bit lange) IP-Adresse des Hosts auf dem sie laufen UND eine durch **16 Bit lange Portnummer** identifiziert
  - Beispiel-Portnummern:
    - HTTP-Server: 80
    - E-Mail-Server: 25

2.1.2 Anwendungsschicht

Anwendungsprotokolle bestimmen:

- Arten von Nachrichten, Syntax der Nachrichten, Semantik der Nachrichten, Regeln für das Senden von und Antworten auf Nachrichten

<u>Öffentlich verfügbare Protokolle:</u>

- Definiert in RFCs
- Ermöglichen Interoperabilität
- z.B. HTTP, SMTP

<u>Proprietäre Protokolle:</u>

- z.B. Skype

- Brauchen: **Transportprotokoll**

2.1.3 Transportdienste für Anwendungen

Kriterien für die Wahl des Transportdienstes:

- Datenverlust
  - Einige Anwendungen können Datenverlust tolerieren (z.B. Audioübertragungen)
  - Andere Anwendungen benötigen einen absolut zuverlässigen Datentransfer (z.B. Dateitransfer)
- Bandbreite
  - Einige Anwendungen (z.B. Multimedia-Streaming) brauchen eine Mindestbandbreite, um zu funktionieren (unelastisch, BB-empfindlich)
  - Andere Anwendungen verwenden einfach die verfügbare Bandbreite (bandbreitenelastische Anwendungen)
- Zeitanforderungen
  - Einige Anwendungen (z.B. Internettelefonie oder Netzwerkspiele) tolerieren nur eine sehr geringe Verzögerung
- Sicherheit
  - Verschlüsselung, Integrität der Daten

2.1.3 Beispiele für Anforderungen von Anwendungen

Anwendung	Datenverlust	Bandbreite	Echtzeit
Dateitransfer	Kein Verlust	Elastisch	Nein
E-Mail	Kein Verlust	Elastisch	Nein
Web	Kein Verlust	Elastisch (wenige Kbps)	Nein
Internettelefonie/ Bildkonferenz	Toleriert Verluste	Audio: wenige Kbps bis 1 Mbps Video: 10 Kbps bis 5 Mbps	Ja: einige Hundert ms
Gespeichertes Audio/Video	Toleriert Verluste	Wie oben	Ja: wenige Sekunden
Interaktive Spiele	Toleriert Verluste	Wenige Kbps bis 10 Kbps	Ja: einige Hundert ms
Instant Messaging	Kein Verlust	Elastisch	Ja und nein

2.1.4 Dienste der Transportprotokolle

TCP-Dienst:

- Verbindungsorientierung: Herstellen einer Verbindung zwischen Client und Server
- Zuverlässiger Transport zwischen sendendem und empfangendem Prozess
- Überlastkontrolle: Bremsen des Senders, wenn das Netzwerk überlastet ist

- **Nicht unterstützt:** Zeit- und Bandbreitengarantien, Verschlüsselung

UDP-Dienst:

- Unzuverlässiger Transport von Daten zwischen Sender und Empfänger

- **Nicht unterstützt:** Verbindungsorientierung, Zuverlässigkeit, Überlastkontrolle, Zeit- oder Bandbreitengarantien, Verschlüsselung

2.1.4 Beispiele für Anwendungsschicht- und Transportprotokolle im Internet

Anwendung	Anwendungsschichtprotokoll	Zugrunde liegendes Transportprotokoll
E-Mail-Dienst	SMTP [RFC 2821]	TCP
Remote-Terminalzugang	Telnet [RFC 854]	TCP
World Wide Web	HTTP [RFC 2616]	TCP
Dateitransfer	FTP [RFC 959]	TCP
Multimedia-Streaming	HTTP (z. B. YouTube), RTP	TCP oder UDP
Internettelefonie	SIP, RTP oder proprietär (z. B. Skype)	Normalerweise UDP