



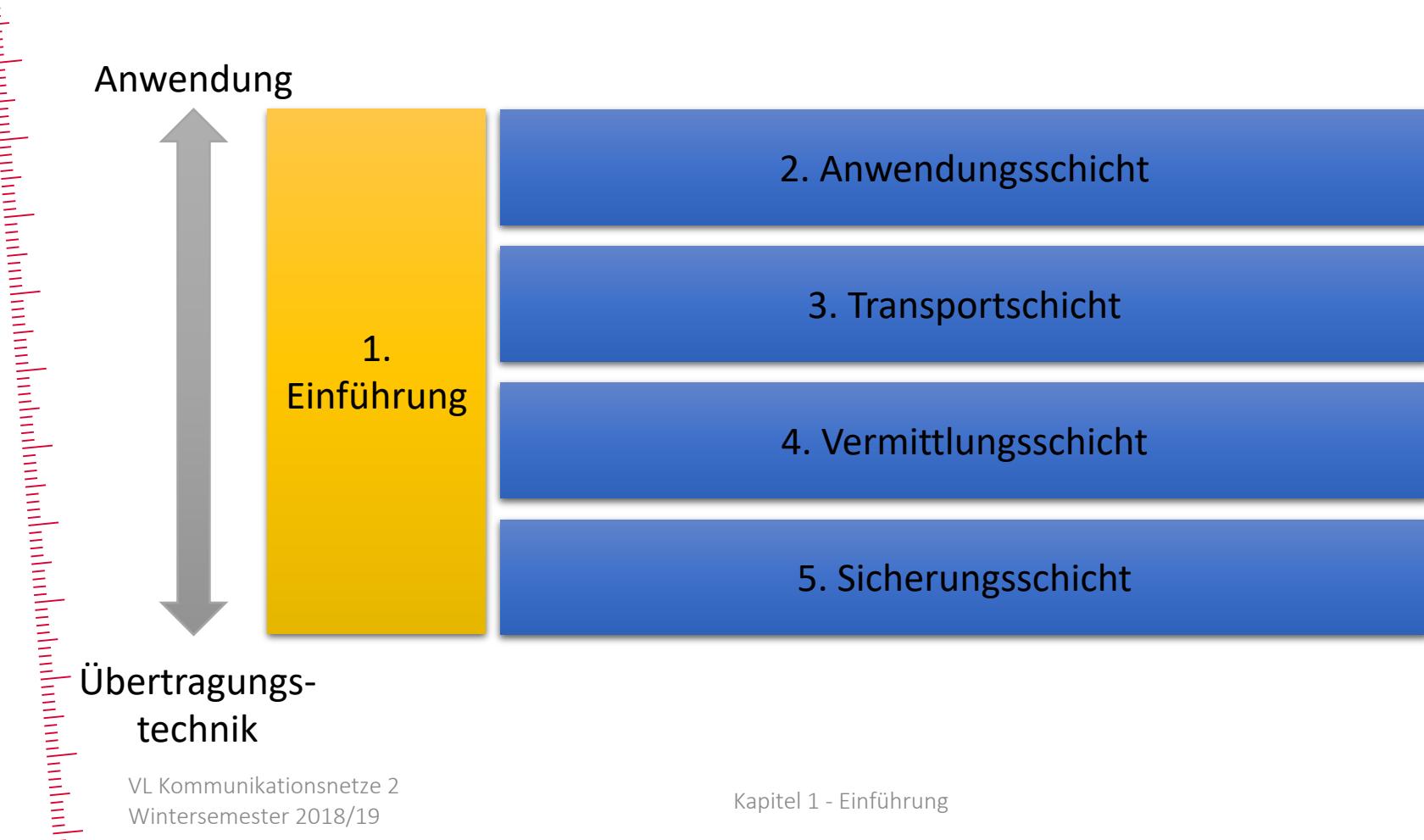
Kapitel 1 Einführung

Vorlesung Kommunikationsnetze 2
Wintersemester 2018/19

Oliver P. Waldhorst

(Basierend auf Materialien von J. Kurose und K. Ross © 1996-2018)

Gliederung der Vorlesung



Das Internet – Dienste und Protokolle

Das Internet Dienste und Protokolle

Internet: Infrastruktur die den Anwendungen (Kommunikations-)Dienste zur Verfügung stellt

- Anwendungen: Web, VoIP, Email, Spiele, E-Commerce, Soziale Netze, ...
- Das Internet bietet **Programmierschnittstellen (Application Programming Interfaces, API)** zum übertragen von Daten zwischen Programmen

Internet: „Netz von Netzen“

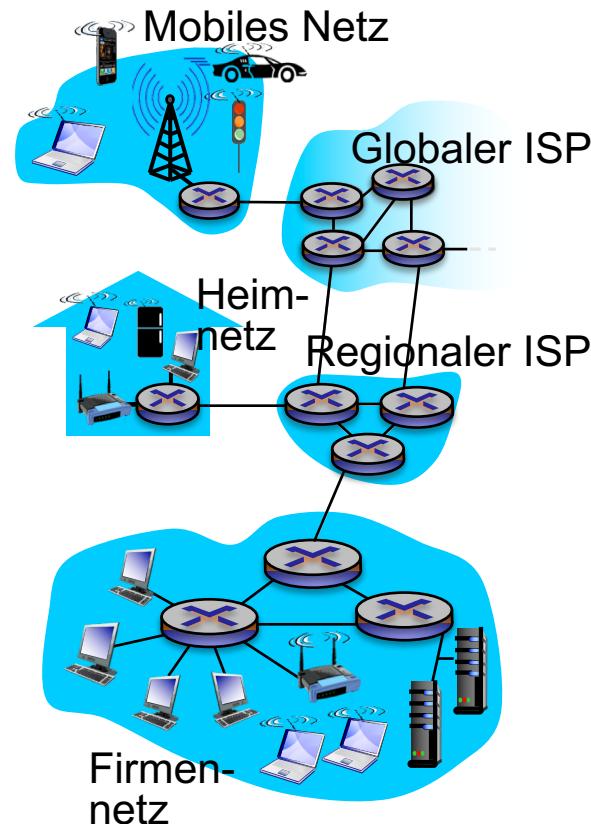
- Miteinander verbundene **Internet Service Provider (ISPs)**

Protokolle kontrollieren das Senden und Empfangen von Nachrichten

- Z.B. TCP, IP, HTTP, Skype, 802.11

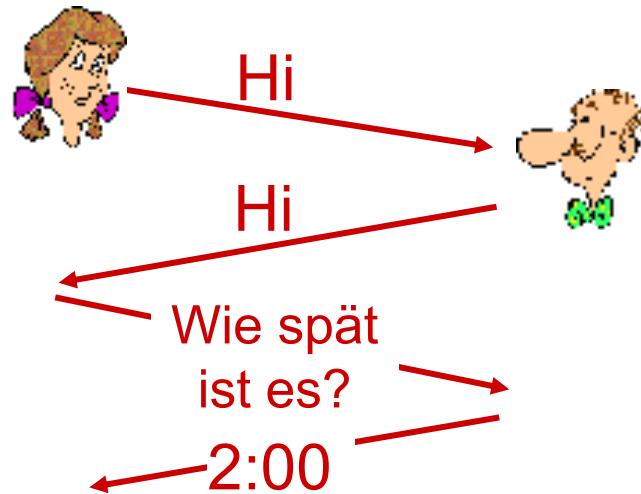
Internet Standards

- Werden von der Internet Engineering Task Force (IETF) festgelegt
- Typischerweise durch **Requests for Comments (RFCs)** beschrieben



Was ist ein Protokoll?

Ein zwischenmenschliches Protokoll und ein Netzprotokoll



Internet Standards: Request for Comments (RFC)

- RFCs haben eine eindeutige Nummer (z.B. ist „Hypertext Transfer Protocol -- HTTP/1.1“ das RFC 2616)
- Können über <https://www.rfc-editor.org/search> gefunden werden
- Historisch: Textdokumente in Festbreitenschrift
 - Heute auch andere Formate verfügbar

RFC Search Detail

Sicher https://www.rfc-editor.org/search/rfc_search_detail.php?rfc=2616&pubstatus%5B%...

Apps Webmail LEO Scholar HsKA KIT Tools

RFC Editor

About this page

RFC Number (or Subseries 2616 Number):

Title/Keyword:

Show Abstract Show Keywords

Additional Criteria ≈

Search Clear all

1 result

Number	Files	Title	Authors	Date	More Info	Status
RFC 2616	ASCII, PS, PDF, PDF with Images	Hypertext Transfer Protocol -- HTTP/1.1	R. Fielding, J. Gettys, R. Mogul, H. Frystyk, L. Masinter, P. Leach, T. Berners-Lee	June 1999	Obsoletes RFC 2069, Obsoleted by RFC 7230, RFC 7231, RFC 7232, RFC 7233, RFC 7234, RFC 7235, Updated by RFC 2817, RFC 5785, RFC 6266, RFC 6585, Errata	Draft Standard

IAB • IANA • IETF • IRTF • ISE • ISOC Reports • Site Map • Contact Us

Network Working Group
Request for Comments: 2616
Obsoletes: 2068
Category: Standards Track

R. Fielding
 UC Irvine
 J. Gettys
 Compaq/W3C
 J. Mogul
 Compaq
 H. Frystyk
 W3C/MIT
 L. Masinter
 Xerox
 P. Leach
 Microsoft
 T. Berners-Lee
 W3C/MIT
 June 1999

Hypertext Transfer Protocol -- HTTP/1.1

Status of this Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Copyright Notice

Copyright (C) The Internet Society (1999). All Rights Reserved.

Abstract

ISO/OSI Referenzmodell

Standardisiert von der **International Organization for Standards (ISO)** als **Open Systems Interconnection Model (OSI)**

- **Anwendungsschicht:** Unterstützung für Netzanwendungen
 - Beispiele: FTP, SMTP, HTTP
- **Darstellungsschicht:** Ermöglicht Anwendungen Bedeutung von Daten zu interpretieren, z.B. Verschlüsselung, Kompression, Konvertierung
- **Sitzungsschicht:** Synchronisation, Checkpoints, Wiederherstellung von Übertragungen
- **Transportschicht:** Datenübertragung zwischen Prozessen
 - Beispiele: TCP, UDP
- **Vermittlungsschicht:** Routen von Nachrichten von Quelle zu Ziel
 - Beispiele: IP, Routing Protokolle
- **Sicherungsschicht:** Datenübertragung zwischen benachbarten Netzelementen
 - Beispiele: Ethernet, 802.11, PPP
- **Bitübertragungsschicht:** Bits „auf der Leitung“
 - In dieser Vorlesung nicht behandelt!



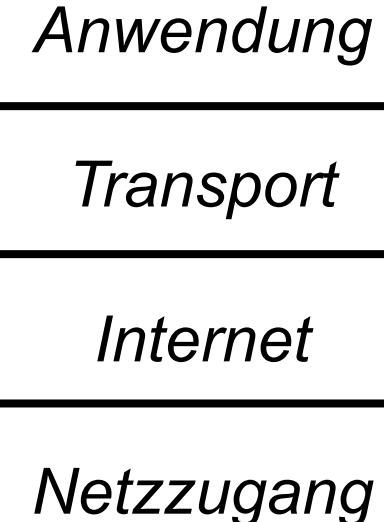
Der Internet-Protokollstapel

Weniger Schichten:

- Anwendungsschicht: OSI-Schicht 7 (+6, +5)
- Transportschicht: OSI-Schicht 4
- Internet-Schicht: OSI-Schicht 3
- Netzzugangsschicht: OSI-Schicht 1+2

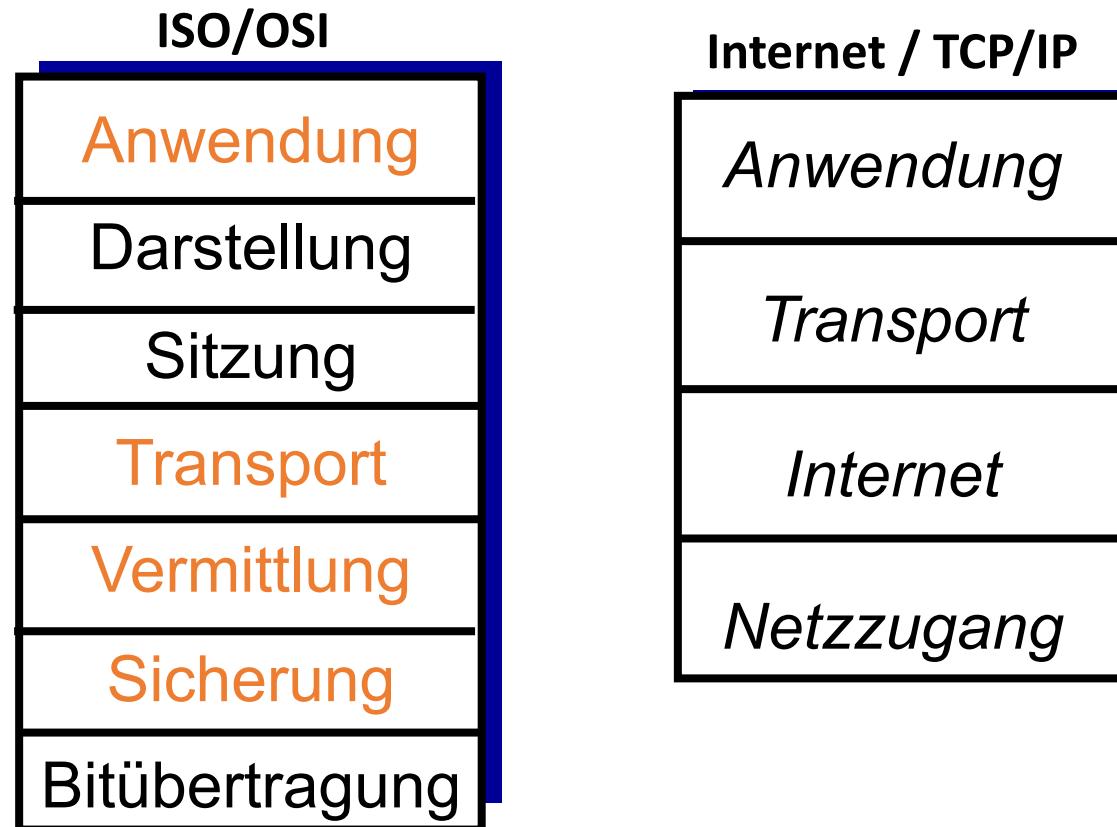
Darstellung- und Sitzungsschicht fehlen im Internet-Protokollstapel!

- Müssen bei Bedarf von Anwendung implementiert werden
- Werden sie wirklich benötigt?

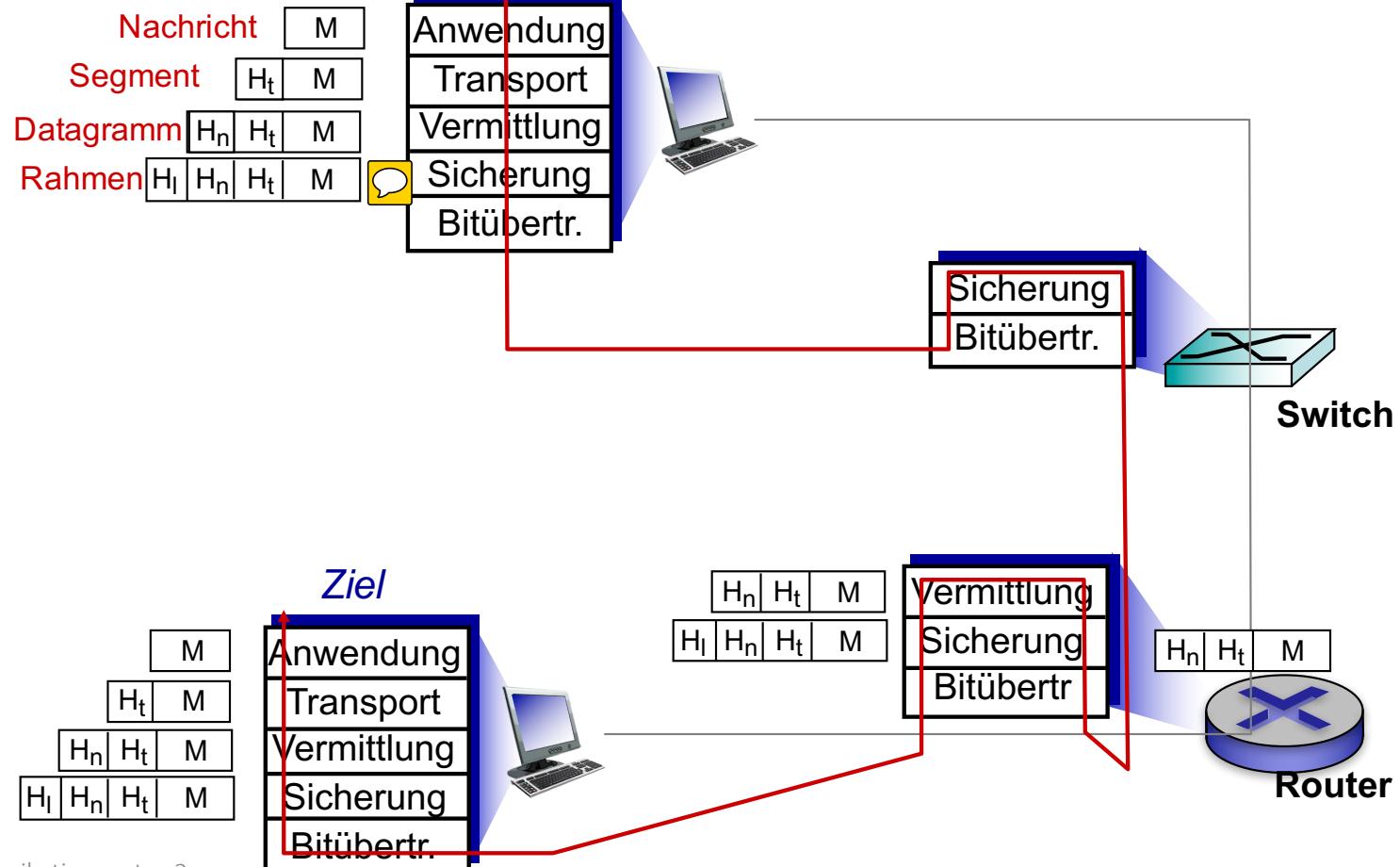


Wir betrachten im weiteren die ISO/OSI Schichten 7, 4, 3, 2!

Wir betrachten die ISO/OSI Schichten 7, 4, 3, 2!



Kapselung



Wiederholung: Ein Tag im Leben einer Web-Seiten-Anfrage

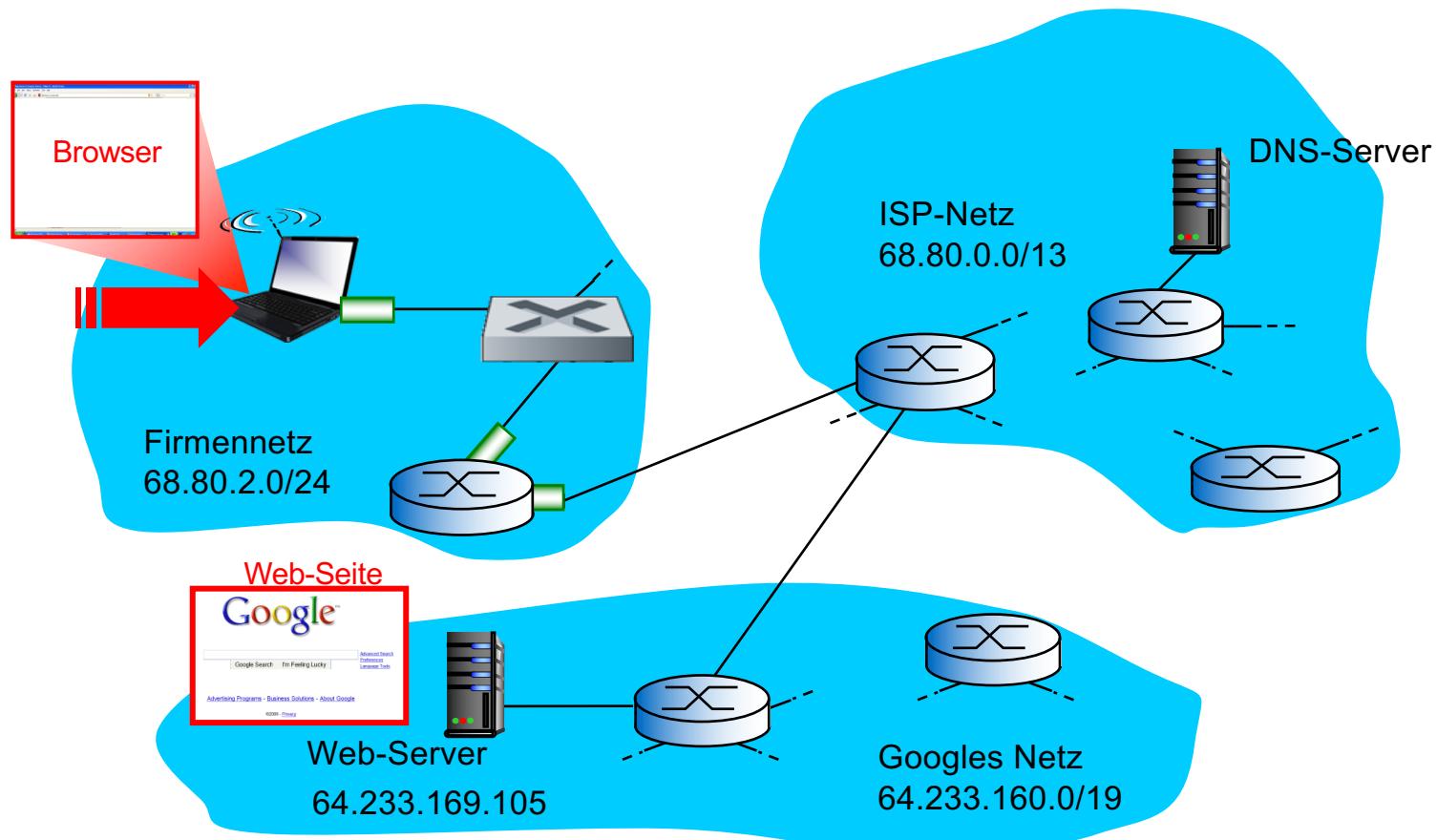
Wir haben in K-Netze 1 den Internet-Protokollstapel kennengelernt!

- Anwendungs-, Transport-, Vermittlungs- und Sicherungsschicht

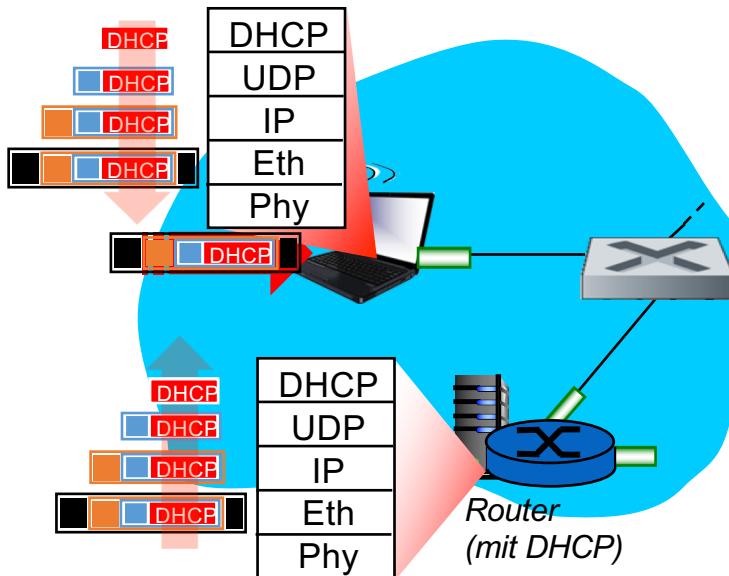
Hier wird nochmal Zusammenspiel der Schichten verdeutlicht

- Ziel: Identifikation der benötigten Protokolle und Demonstration von deren Zusammenspiel in allen Schichten
- Einfaches Beispiel: Nutzer verbindet Laptop mit einem Firmennetz und fordert www.google.com an

Scenario



Internet-Verbindung herstellen



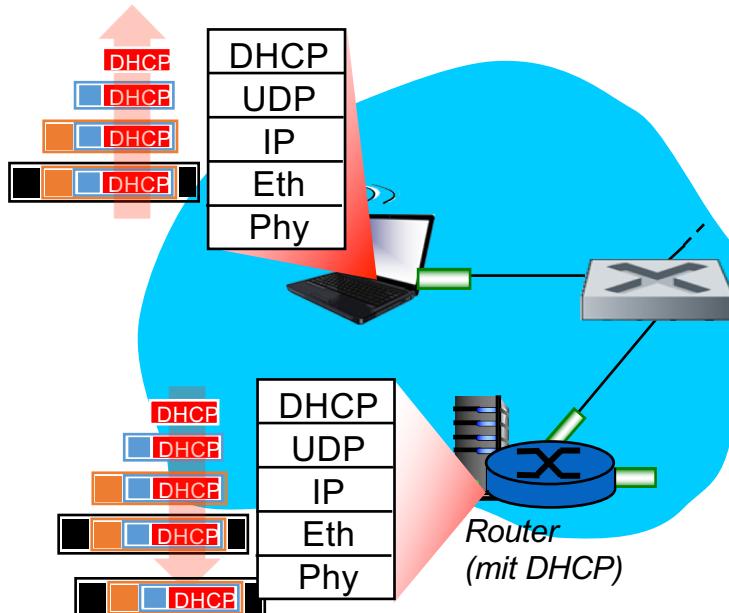
Das Notebook benötigt

- Eigene IP-Adresse
- Adresse des Gateway-Routers
- Adresse des DNS-Servers

→ Einsatz von **DHCP**

- **DHCP-Request** (Anwendungsschicht)
 - Gekapselt in **UDP-Segment**
 - Gekapselt in **IP-Datagramm**
 - Gekapselt in **Ethernet-Rahmen**
- Broadcast des Ethernet-Rahmens (Ziel: FF-FF-FF-FF-FF-FF) im lokalen Netz, wird von DHCP-Server empfangen
- **Demultiplexing**
 - Ethernet zu IP
 - IP zu UDP
 - UDP zu DHCP

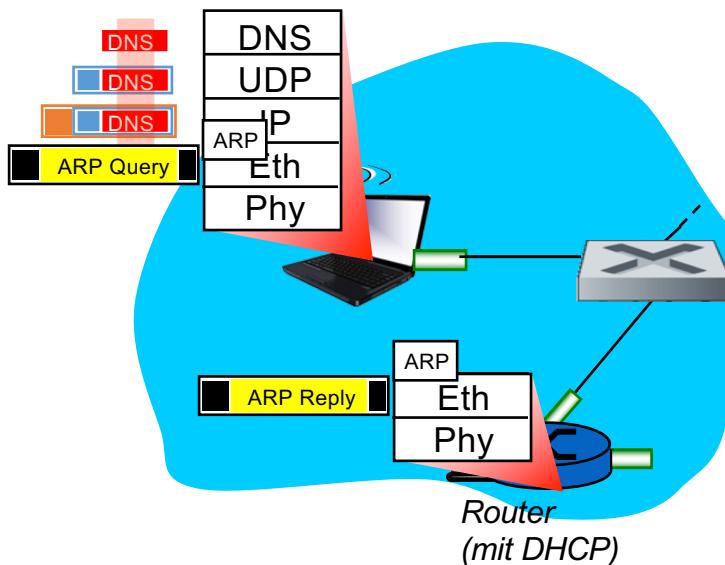
Internet-Verbindung herstellen (Forts.)



- DHCP-Server konstruiert **DHCP-ACK** mit
 - IP-Adresse des Notebooks
 - IP-Adresse des Gateway-Routers
 - IP-Adresse des DNS-Servers
- Kapselung des DHCP-ACK (UDP, IP, Ethernet), Weiterleitung durch lokales Netz, Demultiplexing
- DHCP Client Prozess auf Notebook empfängt DHCP-ACK

Notebook hat nun eine IP-Adresse und kennt die IP-Adressen von DNS-Server und Gateway Router

ARP (vor DNS, vor HTTP)



Für **HTTP-Anfrage** an www.google.com wird dessen IP-Adresse benötigt: **DNS**

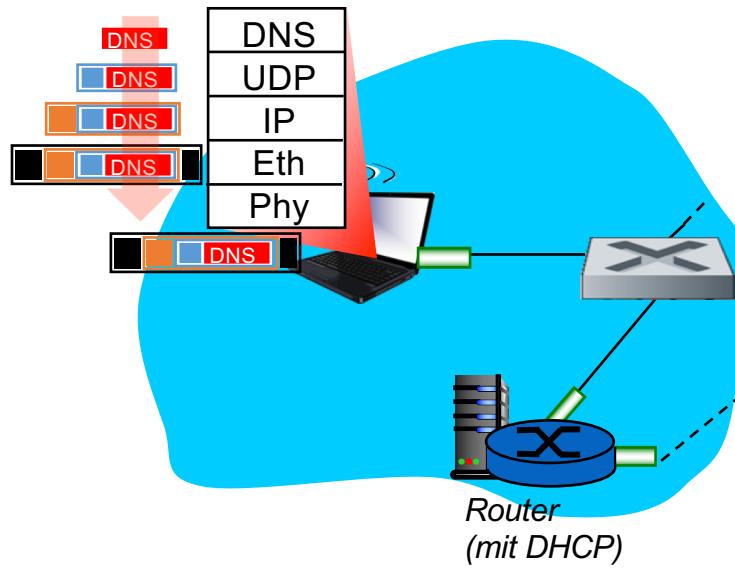
- Notebook erzeugt DNS-Anfrage
- Wird gekapselt in UDP, IP, Ethernet

Zum Senden wird **MAC-Adresse** des Gateway-Routers benötigt: **ARP**

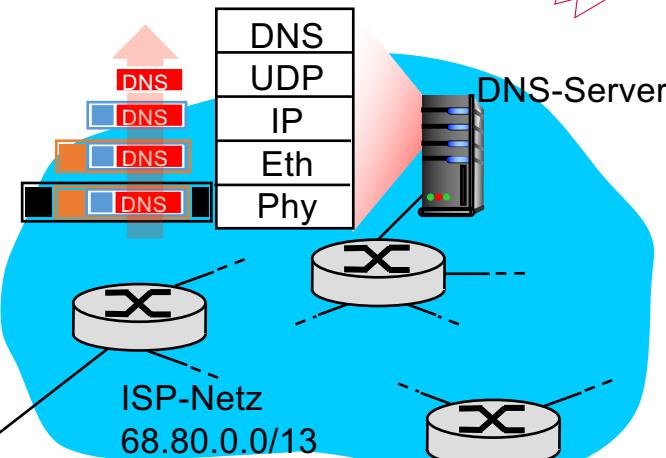
- **ARP-Query** für IP-Adresse des Gateway-Routers wird per **Broadcast** an MAC-Adresse **FF-FF-FF-FF-FF-FF** versendet
- Wird von Router empfangen, dieser antwortet mit **ARP-Reply** und seiner MAC-Adresse per **Unicast** an MAC-Adresse des Notebook
- Switch „**lernt**“ beide MAC-Adressen / Schnittstellen

Notebook kennt nun MAC-Adresse des Gateway-Routers, also kann Ethernet-Rahmen mit DNS-Anfrage versendet werden!

DNS

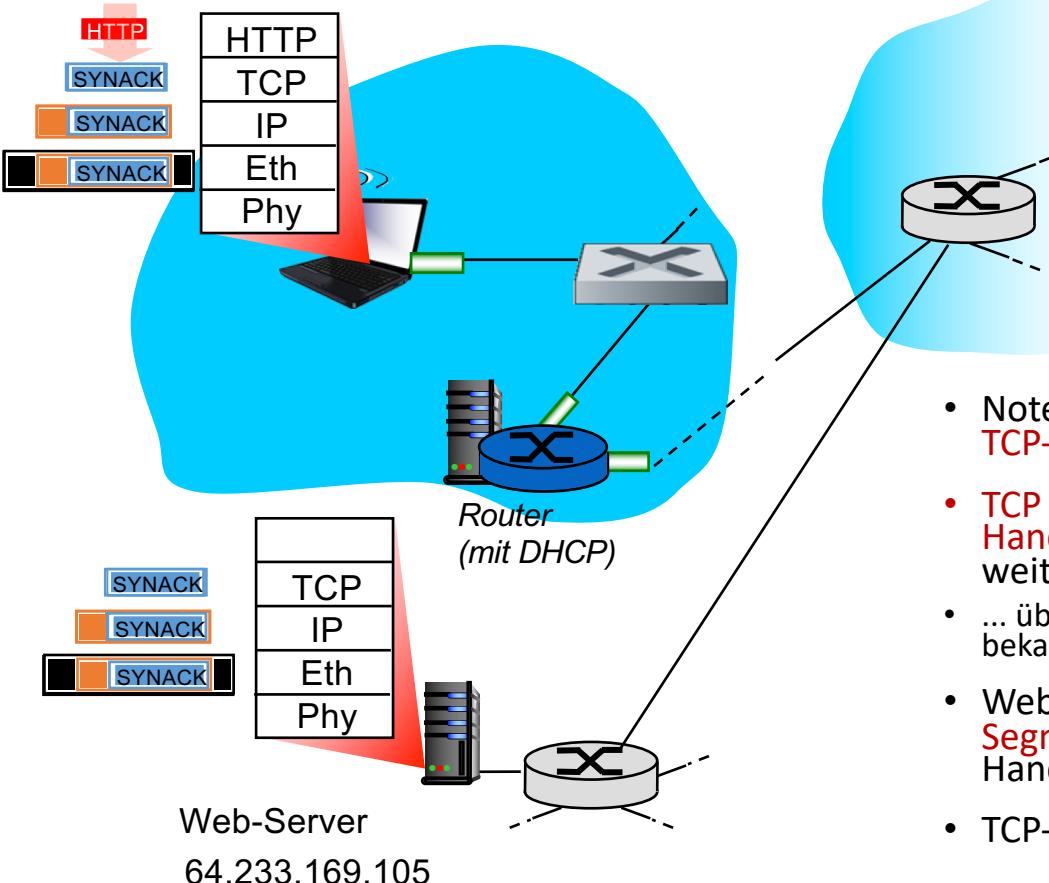


- **IP-Datagramm** mit DNS-Anfrage wird über lokales Netz an Gateway-Router weitergeleitet



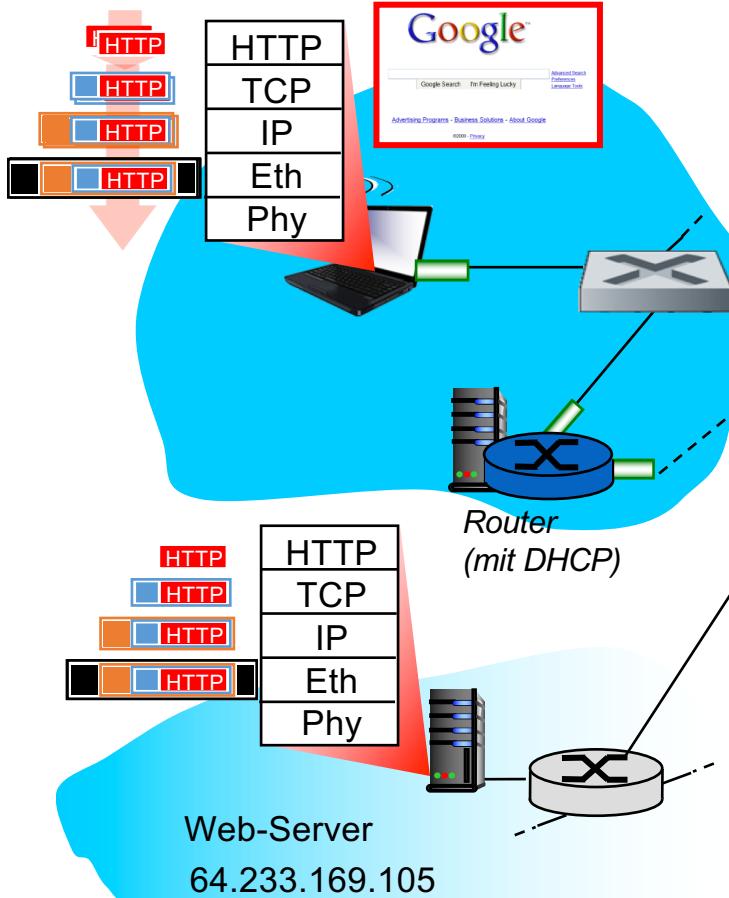
- IP-Datagramm wird in Telekom-Netz weitergeleitet und dort zu DNS-Server geroutet
 - Routing-Tabellen können von **RIP**, **OSPF** und/oder **BGP** stammen
 - Demultiplexing auf DNS-Server
 - DNS-Server antwortet an Notebook mit der IP-Adresse von www.google.com

TCP-Verbindung für HTTP



- Notebook öffnet für HTTP-Anfrage ein **TCP-Socket** zum Webserver
- **TCP SYN Segment** (1. Schritt 3-Wege-Handshake) wird zum Webserver weitergeleitet
- ... über Gateway-Router, dessen MAC-Adresse bekannt ist
- Web-Server antwortet mit **TCP SYNACK Segment** (Schritt 2 in 3-Wege-Handshake)
- TCP-Verbindung (fast) aufgebaut!

HTTP-Anfrage / -Antwort



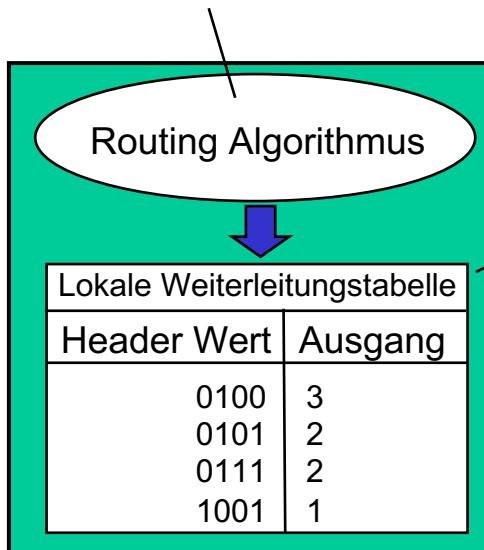
- **HTTP-Anfrage** wird über TCP-Socket gesendet
 - I.d.R. zusammen mit ACK für SYNACK (Schritt 3 in 3-Wege-Handshake)
- IP-Datagramm mit HTTP-Request wird zu www.google.com geroutet
- Web-Server sendet **HTTP-Antwort** (enthält Web-Seite)
- IP-Datagramm mit HTTP-Antwort wird zurück zu Notebook geroutet
- (ggf. Verbindungsabbau durch Notebook oder Web-Server)

Weiterleitung, Verzögerung und Paketverluste

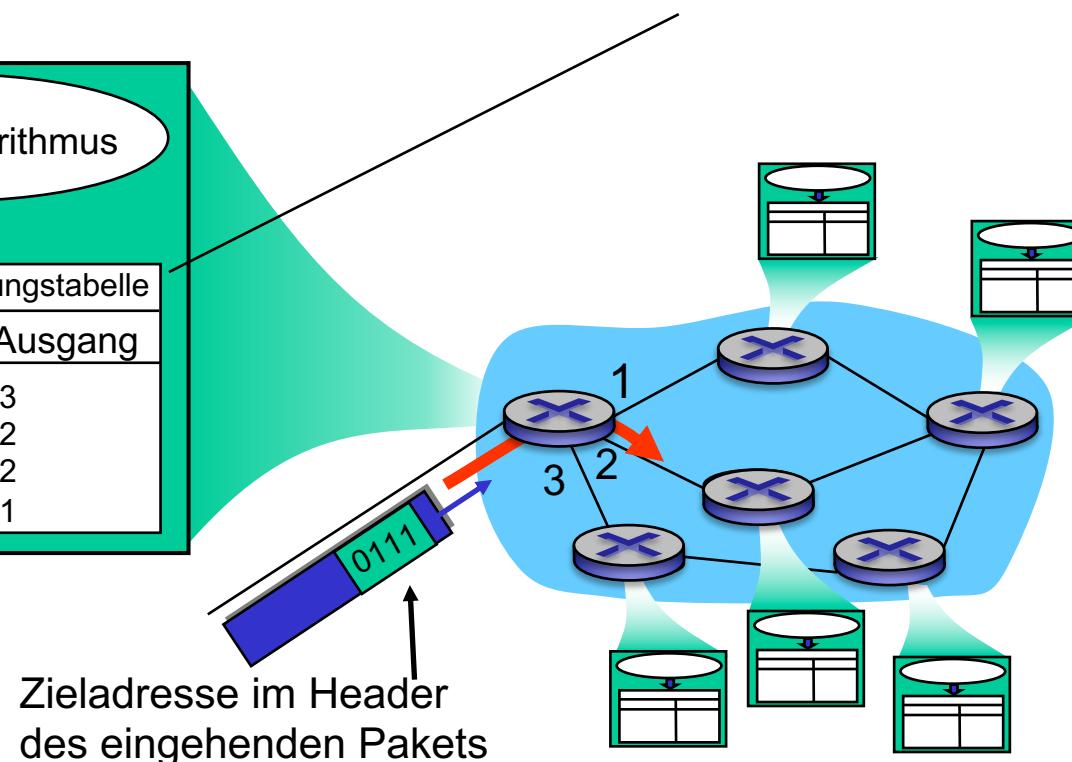
Paketvermittlung im Netzinneren

Routing: bestimmt Pfad eines Pakets zwischen Quelle und Ziel

- **Routing Algorithmus**



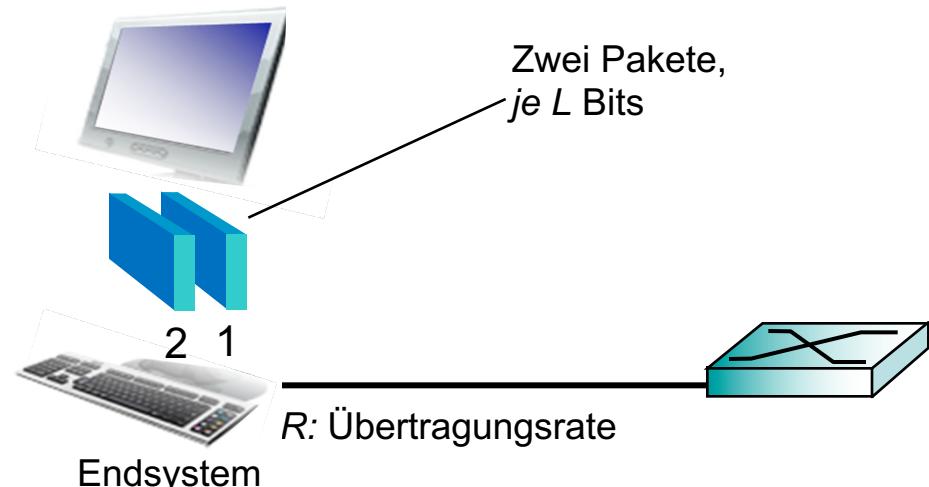
Forwarding: Weiterleiten eines Pakets zwischen Router-Eingang und passendem Ausgang



Senden von Daten in Paketen

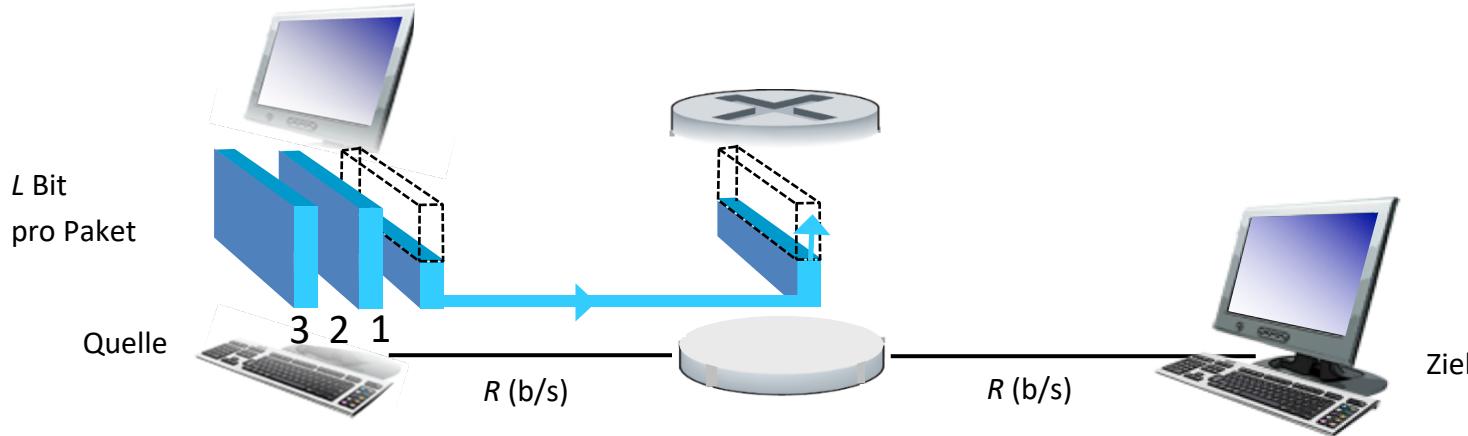
Sendefunktion auf einem Endsystem:

- Annehmen von Nachrichten der Anwendung
- Aufteilen in kleinere Einheiten (**Pakete** der Länge L Bit)
- Übertragen des Pakets in das Zugangsnetz mit **Übertragungsrate R**
 - Übertragungsrate der Verbindung, auch **Verbindungskapazität** oder **Verbindungsbandbreite**



Paket- übertragungs- verzögerung	$= \frac{\text{Zeit um } L\text{-bit}}{\text{zu übertragen}}$	$\frac{L \text{ (b)}}{R \text{ (b/s)}}$
--	---	---

Paketvermittlung: Store-and-Forward



- **Store and Forward** (Speichern und Weiterleiten): Das komplette Paket muss am Router ankommen, bevor es über den nächsten Link übertragen wird
- Übertragung eines L -Bit Pakets über R (b/s) Verbindung dauert L/R
- Ende-zu-Ende-Verzögerung im Beispiel $2L/R$ (bei Vernachlässigung der Ausbreitungsverzögerung)

Beispiel bei zwei Hops:

- $L = 7,5 \text{ Mb}$
- $R = 1,5 \text{ Mb/s}$

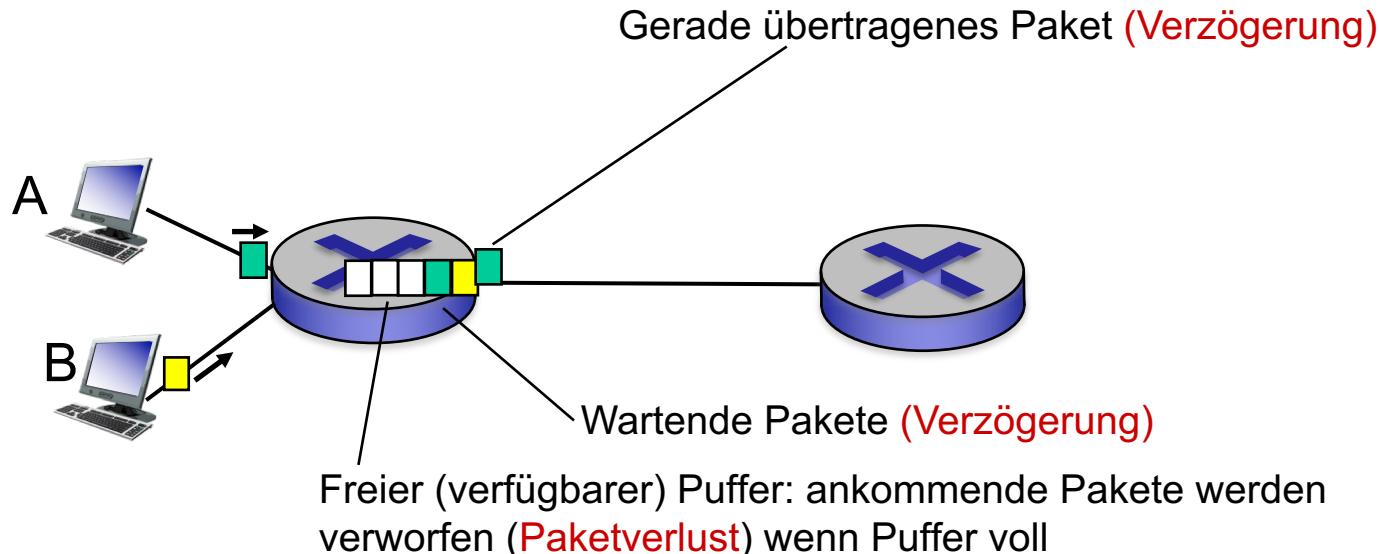
! Zwei-Hop Verzögerung?

Dieses Modell ist stark vereinfacht!

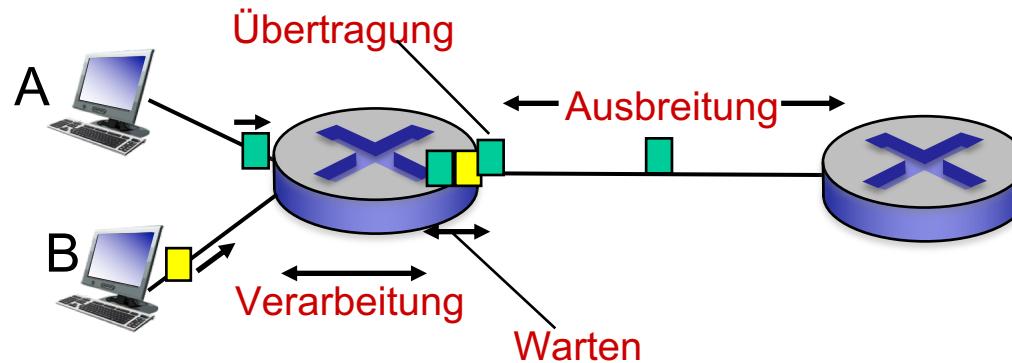
Wie entstehen Verzögerung und Paketverluste?

Paketpuffer sind als **Warteschlangen (Queues)** realisiert

- Paketankunftsrate kann (temporär) größer als Kapazität des Ausgangslinks sein
- Pakete warten in Warteschlange, bis sie gesendet werden können



Vier Ursachen für Paketverzögerungen (1)



$$d_{\text{nodal}} = d_{\text{proc}} + d_{\text{queue}} + d_{\text{trans}} + d_{\text{prop}}$$

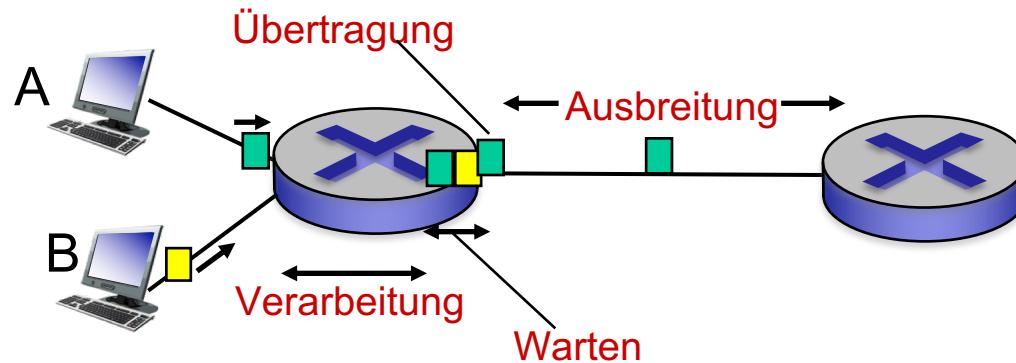
d_{proc} : Verarbeitungsverzögerung

- (engl. Nodal Processing Delay)
- Fehlerprüfung (Bitfehler)
- Bestimmung des Ausgangsports
- Typischerweise < Millisekunden

d_{queue} : Wartezeitverzögerung 

- (engl. Queueing Delay)
- Warten auf Übertragung an ausgehender Verbindung
- Abhängig von Stausituation am Router

Vier Ursachen für Paketverzögerungen (2)



$$d_{\text{nodal}} = d_{\text{proc}} + d_{\text{queue}} + d_{\text{trans}} + d_{\text{prop}}$$

d_{trans} : Übertragungsverzögerung:

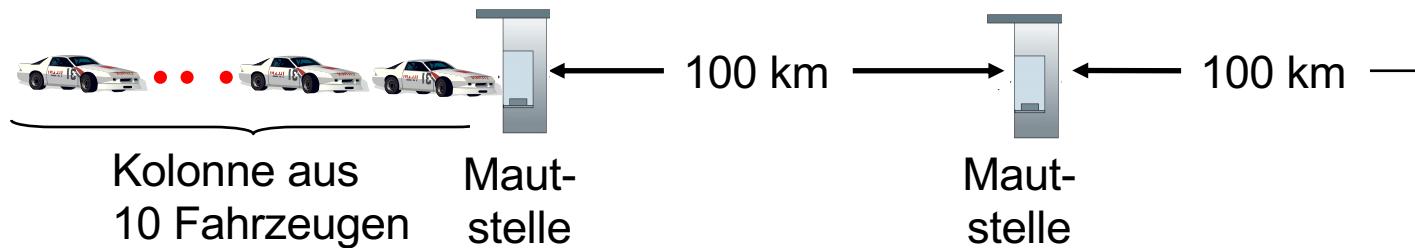
- (engl. Transmission Delay)
- L : Paketgröße (Bit)
- R : Bandbreite der Verbindung (b/s)
- $d_{\text{trans}} = L/R$

d_{trans} und d_{prop}
verschieden (!)

d_{prop} : Ausbreitungsverzögerung:

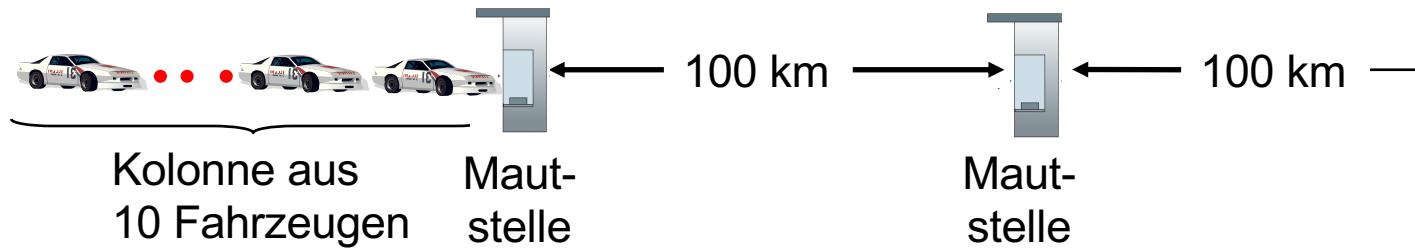
- (engl. Propagation Delay)
- d : Länge der physischen Verbindung
- s : Ausbreitungsgeschwindigkeit ($\sim 2 \times 10^8 \text{ m/s}$)
- $d_{\text{prop}} = d/s$

Analogie: Fahrzeugkolonne



- Fahrzeuge fahren 100 km/h (Ausbreitungsgeschwindigkeit)
 - Mautstelle braucht 12 s zur Abfertigung eines Fahrzeugs (Übertragungsverzögerung)
 - Fahrzeug ~ Bit; Fahrzeugkolonne ~ Paket; Mautstelle ~ Router
- ! Wie lange dauert es, bis letztes Fahrzeug (~ komplettes Paket) zweite Mautstelle erreicht hat?

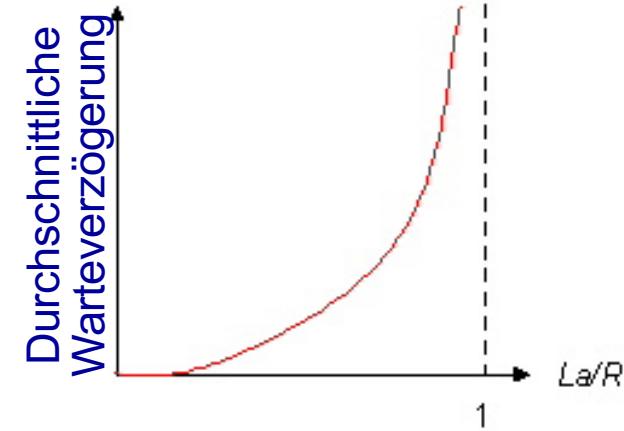
Analogie: Fahrzeugkolonne (2)



- Fahrzeuge fahren jetzt mit 1000 km/h
 - Mautstelle braucht 1min zur Abfertigung eines Fahrzeugs?
- !** Sind Fahrzeuge an der 2. Mautstelle, bevor alle Fahrzeuge an der ersten Mautstelle abgefertigt sind?

Details zur Warteverzögerung

- R: Verbindungsbandbreite (b/s)
- L: Paketgröße (b)
- a: Durchschnittliche Ankunftsrate ($1/s$)
- **Verkehrsintensität** La/R
- **La/R nahe 0:** Durchschnittliche Warteverzögerung klein
- **La/R nahe 1:** Durchschnittliche Warteverzögerung groß
- **La/R größer 1:** Es kommt mehr Last an, als verarbeitet werden kann, durchschnittliche Warteverzögerung unendlich



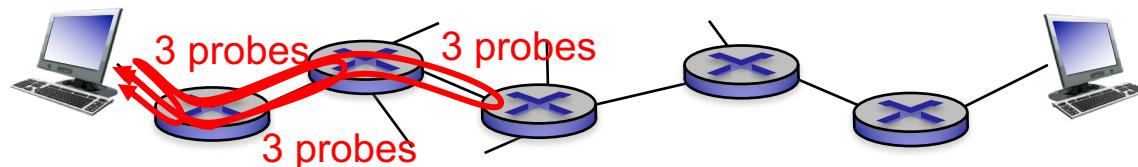
$La/R \sim 0$



$La/R \rightarrow 1$

Verzögerungen (und Routen) im „echten“ Internet

- Programm traceroute: Messung der Verzögerung von Quelle zu jedem Router auf dem Pfad zu gegebenem Ziel
- Für jeden Router i:
 - Sende drei Pakete, die den Router i auf dem Pfad Richtung Ziel erreichen, mit TTL i. 
 - Router i sendet ICMP-Nachricht an Quelle (Typ 0, Code 0 „TTL expired“)
 - Quelle misst Verzögerung zwischen Senden und Empfangen der Pakete



Verzögerungen (und Routen) im „echten“ Internet

3 Verzögerungsmessungen von
vsrv41980.customer.vlinux.de zu 31.15.64.1

```
waldhorst@vsrv41980:~$ traceroute gaia.cs.umass.edu
traceroute to gaia.cs.umass.edu (128.119.245.12), 30 hops max, 60 byte packets
 1  31.15.64.1 (31.15.64.1)  1.444 ms  0.763 ms  1.400 ms
 2  77.172.rev.synaix.de (80.87.172.77)  0.284 ms  0.373 ms  0.445 ms
 3  ddf-b2-link.telia.net (213.248.69.122)  7.823 ms  7.732 ms  7.699 ms
 4  ffm-bb3-link.telia.net (62.115.112.60)  11.682 ms  11.692 ms  11.726 ms
 5  ffm-b1-link.telia.net (62.115.121.5)  18.097 ms  ffm-b1-link.telia.net
(62.115.137.165)  11.204 ms  ffm-b1-link.telia.net (62.115.141.239)  15.455 ms
 6  * * *
 7  * * *
 8  UNIVERSITY.ear3.NewYork1.Level3.net (4.71.230.234)  92.653 ms  92.618
ms  92.813 ms
 9  core1-rt-et-8-3-0.gw.umass.edu (192.80.83.109)  91.959 ms  core2-rt-et-8-3-
0.gw.umass.edu (192.80.83.113)  98.711 ms  98.678 ms
10  n5-rt-1-1-et-0-0-0.gw.umass.edu (128.119.0.8)  99.215 ms  98.718 ms  98.770
ms
11  cics-rt-xe-0-0-0.gw.umass.edu (128.119.3.32)  92.779 ms  92.786 ms  92.846 ms
12  nscls1bbs1.cs.umass.edu (128.119.240.253)  101.917 ms  100.547 ms  100.546 ms
13  gaia.cs.umass.edu (128.119.245.12)  92.721 ms !X  92.673 ms !X  92.687 ms !X
```

* Keine Antwort (Paket verloren, Router antwortet nicht)

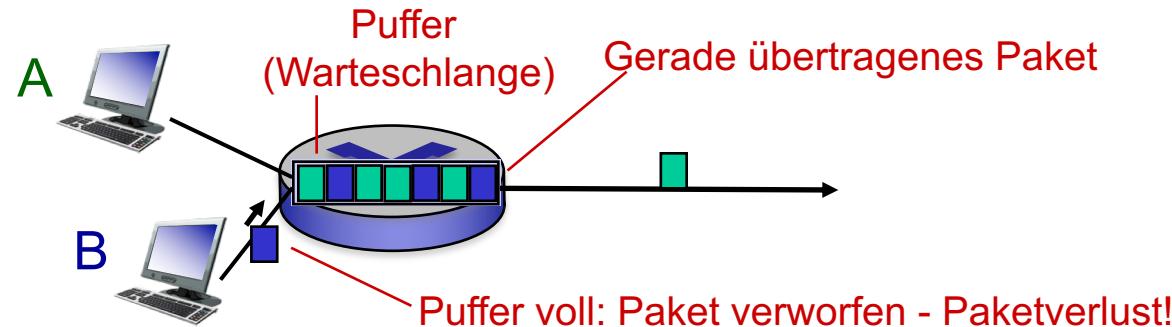
Anmerkung zu Verzögerungen

Protokolle können Verzögerungen deutlich erhöhen

- Beispiel: 3-Wege-Handshake von TCP beeinflusst maßgeblich Übertragungsverzögerung von HTTP

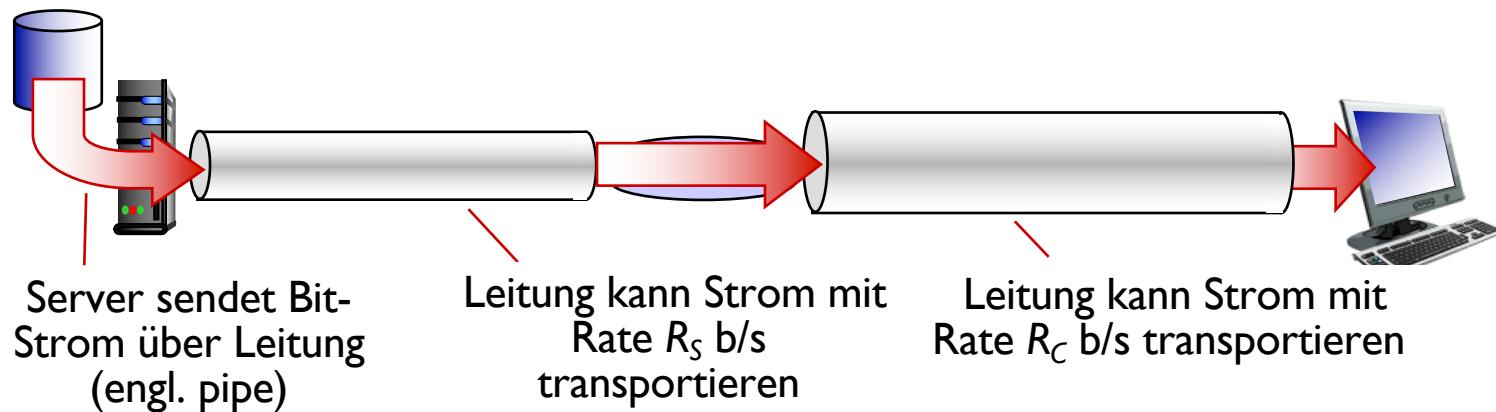
Paketverluste

- Puffer / Warteschlange am Ausgangsport hat begrenzte Kapazität
- Bei vollem Puffer ankommende Pakete werden verworfen (Paketverlust!)
- Verlorene Pakete können wiederholt werden oder nicht



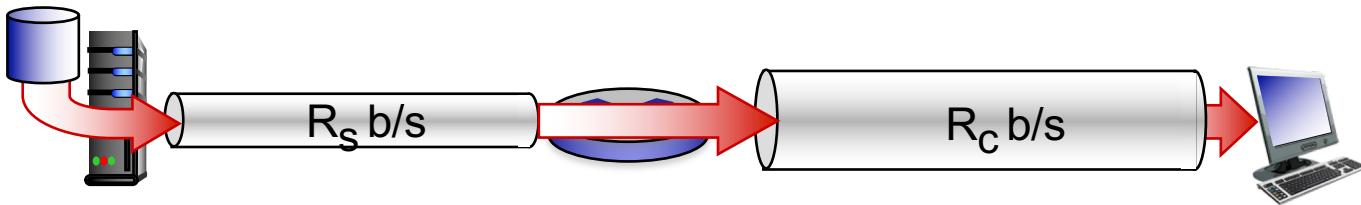
Durchsatz

- **Durchsatz:** Rate (Bit pro Zeiteinheit) mit der Bits übertragen zwischen Sender und Empfänger übertragen werden
 - **Momentaner Durchsatz:** Rate zu einem gegebenen Zeitpunkt
 - **Durchschnittlicher Durchsatz:** Rate über einen längeren Zeitraum

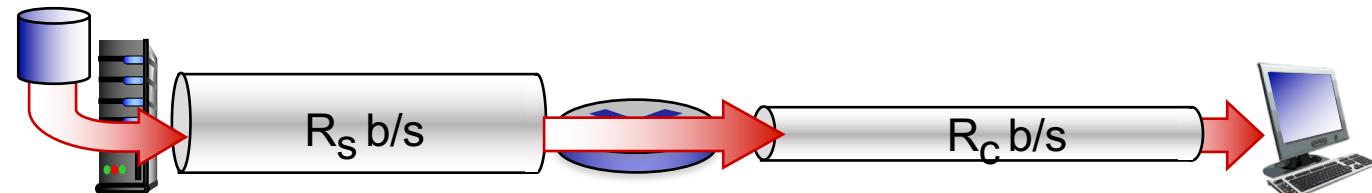


Durchsatz (2)

! $R_s < R_c$: Was ist der durchschnittliche Durchsatz?



! $R_s > R_c$: Was ist der durchschnittliche Durchsatz?

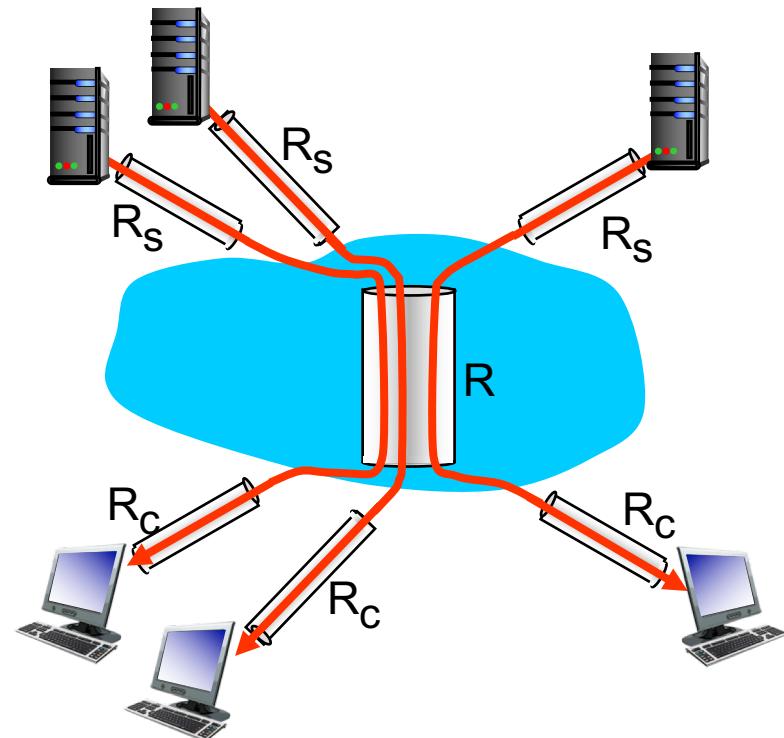


Flaschenhals (Bottleneck Link)

Verbindung auf dem Pfad, die Durchsatz beschränkt

Durchsatz im Internet

- Durchsatz pro Verbindung:
 $\min(R_C, R_S, R/3)$
- In der Praxis ist oft R_C oder R_S der Flaschenhals



3 Verbindungen teilen (fair) die Backbone-Verbindung mit R b/s

Sicherheit in Netzen

Netzsicherheit

Fragestellungen der Netzsicherheit

- Wie können böswillige Teilnehmer Netze angreifen?
- Wie können wir Netze gegen Angriffe verteidigen?
- Wie können wir Netzarchitekturen entwerfen, die immun gegen Angriffe sind

Sicherheit wurde beim Internet-Design zunächst vernachlässigt

- Annahme: Alle mit dem Netz verbundenen Benutzer trauen sich gegenseitig
- Protokoll-Designer liefern sich Wettlauf mit Angreifern
- Sicherheit muss auf allen Schichten berücksichtigt werden!

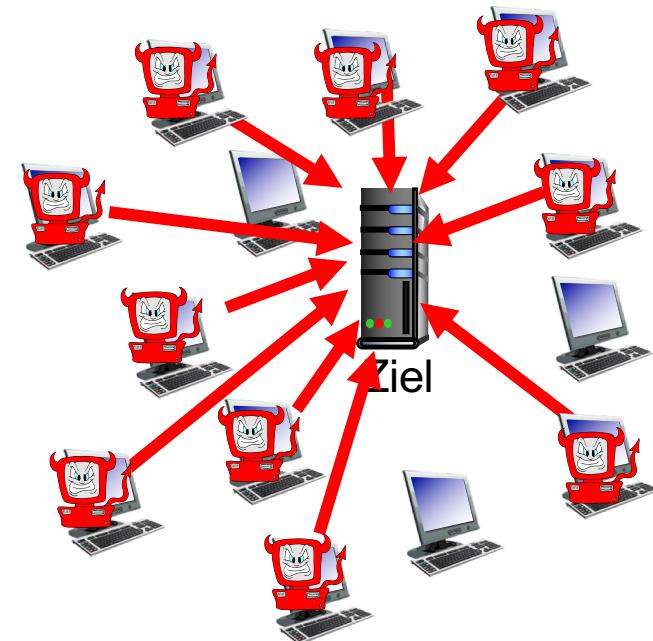
Angriff: Rechner über Internet mit Schadprogramm infizieren

- **Schadprogramm** kann auf Rechner kommen als
 - **Virus**: Selbst-replizierendes Programm, das empfangen und ausgeführt wird (z.B. Email-Attachment)
 - **Wurm**: Selbst-replizierendes Programm, das sich selber ausführt
- Schadprogramm kann als **Spionagesoftware** Tastaturanschläge, besuchte Webseiten, ... aufzeichnen und an Sammelstelle im Internet übertragen
- Infizierte Rechner können als **Bot-Netz** Spam versenden, DoS-Attacken ausführen, ...

Angriff: Blockieren von Server und Netzinfrastruktur

Denial of Service (DoS): Angreifer macht Ressourcen (Server, Bandbreite) unnutzbar für legitime Nutzung, z.B. durch Überschwemmen mit gefälschtem Verkehr

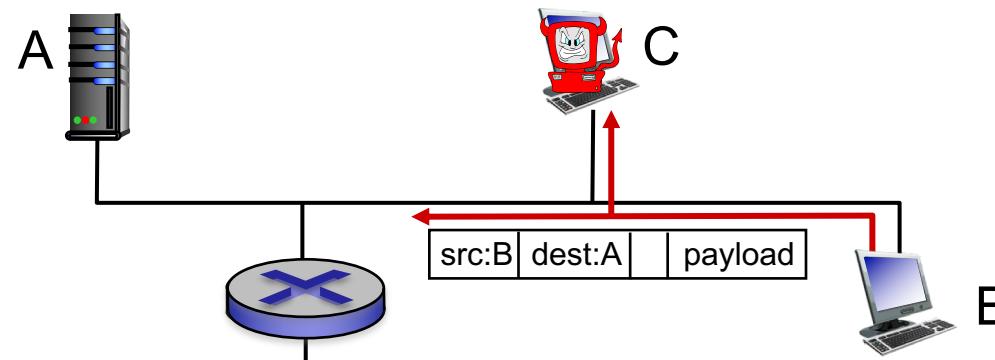
1. Ziel auswählen
2. Viele Rechner im Internet mit Schadprogramm infizieren (Bot-Netz)
3. Pakete von infizierten Rechnern zum Ziel schicken



Angriff: Pakete mitlesen

„Packet-Sniffing“

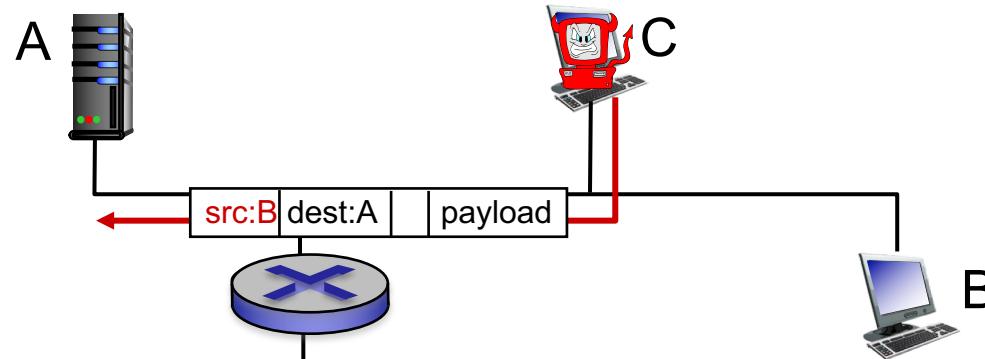
- Broadcast-Medium (geteiltes Ethernet, Funk, ...)
- Netzwerkkarte im „Promiscuous Mode“ kann alle Pakete lesen / aufzeichnen 
- Z.B. Wireshark ist ein Packet-Sniffer!



Angriff: Adressen fälschen

„IP-Spoofing“

- Paket mit falscher IP-Quelladresse senden



Zusammenfassung Kapitel 1

Entwicklung von Internet-Anwendungen und -Protokollen ist und bleibt eine Herausforderung

- Anforderungen an Leistung (Verzögerung, Durchsatz, ...) und Sicherheit steigen ständig
- Protokolle kommen und gehen, aber Problemstellungen und Lösungsansätze bleiben

→ Diese Vorlesung vermittelt Verständnis für Probleme und Lösungsansätze auf allen Schichten des Protokollstapels