

Rechnersysteme und -netze

Kapitel 9

Rechnernetze

Bastian Goldlücke

Universität Konstanz
WS 2020/21

Inhalt

1 Das Internet

- 1.1 Struktur des Internet: Wirte, Verbindungen, Vermittlungsstellen
- 1.2 Kernstruktur und Zugriffsnetze
- 1.3 Paketvermittlung
- 1.4 Leitungsvermittlung, Frequenz- und Zeit-Multiplexverfahren
- 1.5 Prinzipien der Paketvermittlung, Speichern und Weiterleiten
- 1.6 Warteschlangen, Paketverlust und Verzögerungen

2 Kommunikation im Internet

- 2.1 Beispiel Webseitenaufruf
- 2.2 Schichtenmodell
- 2.3 Routing
- 2.4 Server und Datenfluss

3 Einige Protokolle

- 3.1 Anwendungsschicht: HTTP, SMTP, FTP
- 3.2 Transportschicht: Transmission Control Protocol (TCP)
- 3.3 Netzwerkschicht: Internet Protocol (IP)
- 3.4 Verbindungsschicht: z.B. Ethernet
- 3.5 Physische Schicht: z.B. Manchester-Kode

Inhalt

1 Das Internet

- 1.1 Struktur des Internet: Wirte, Verbindungen, Vermittlungsstellen
- 1.2 Kernstruktur und Zugriffsnetze
- 1.3 Paketvermittlung
- 1.4 Leitungsvermittlung, Frequenz- und Zeit-Multiplexverfahren
- 1.5 Prinzipien der Paketvermittlung, Speichern und Weiterleiten
- 1.6 Warteschlangen, Paketverlust und Verzögerungen

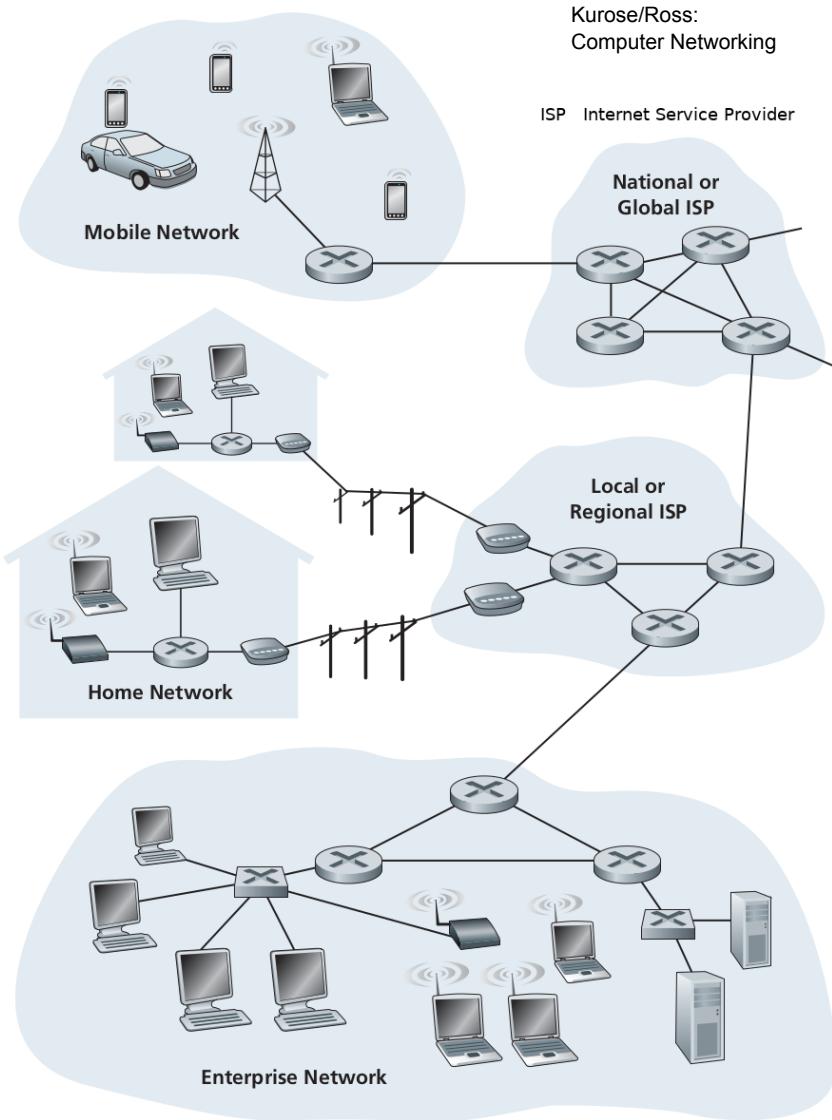
2 Kommunikation im Internet

- 2.1 Beispiel Webseitenaufruf
- 2.2 Schichtenmodell
- 2.3 Routing
- 2.4 Server und Datenfluss

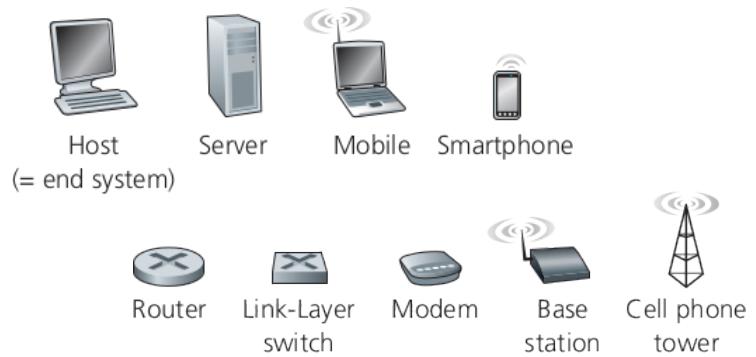
3 Einige Protokolle

- 3.1 Anwendungsschicht: HTTP, SMTP, FTP
- 3.2 Transportschicht: Transmission Control Protocol (TCP)
- 3.3 Netzwerkschicht: Internet Protocol (IP)
- 3.4 Verbindungsschicht: z.B. Ethernet
- 3.5 Physische Schicht: z.B. Manchester-Kode

Bestandteile des Internet

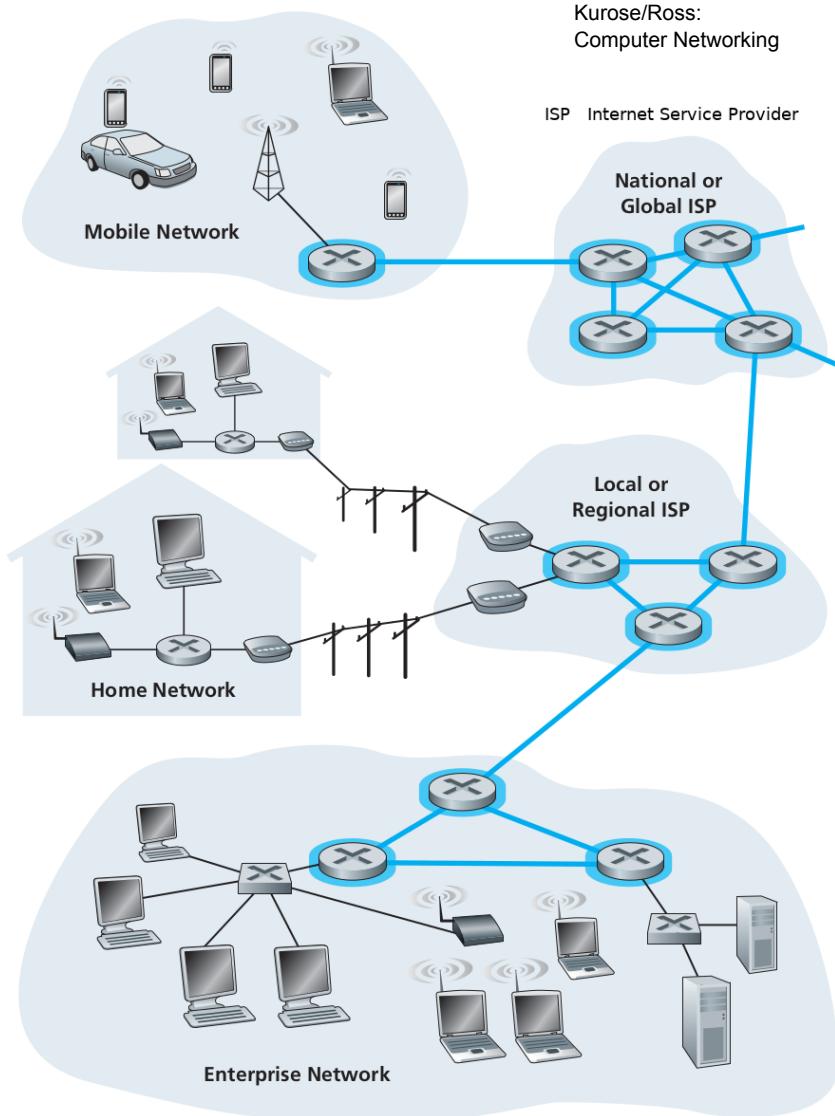


- Das Internet ist ein Rechnernetzwerk, das etwa eine Milliarde Rechner (Laptops, Mobiltelefone u.ä. nicht gezählt) rund um die Erde miteinander verbindet.
- Diese Rechner nennt man **Wirte** (hosts) oder **Endsysteme** (end systems).
- Sie sind durch ein Netz von **Kommunikationsverbindungen** (communication links) und **Paketvermittlungen** (packet switches) verbunden.



Internet: Kernstruktur

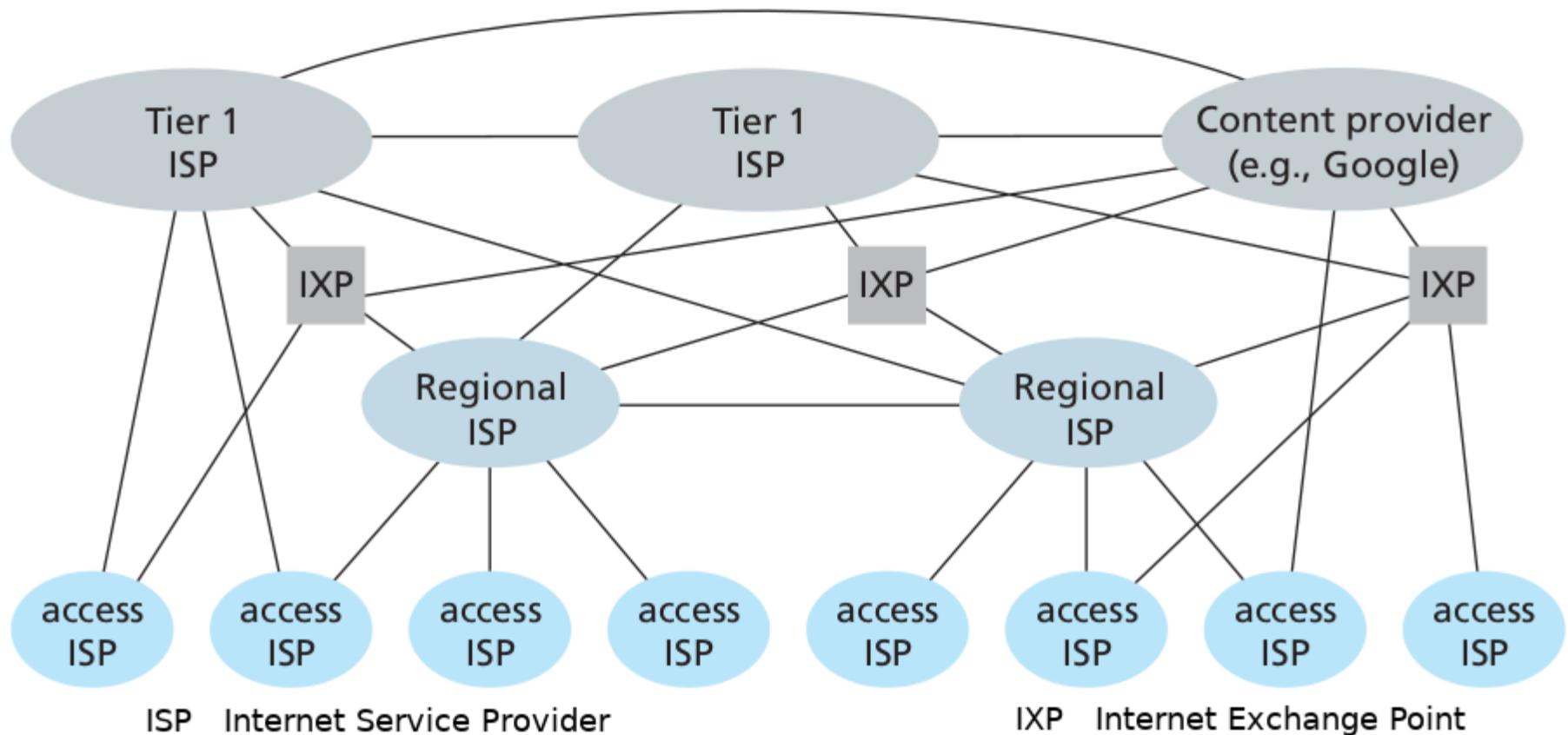
Kurose/Ross:
Computer Networking



- Die Kernstruktur des Internets ist ein Netz von miteinander verbundenen **Vermittlungsstellen** (routers), über die Nachrichten (messages) in Form von Datenpaketen (packets) zwischen Endsystemen übertragen werden.
- Diese Kernstruktur des Internet wird von vielen verschiedenen **Internet-Dienstleistungsanbietern** (internet service providers, ISPs) bereitgestellt.

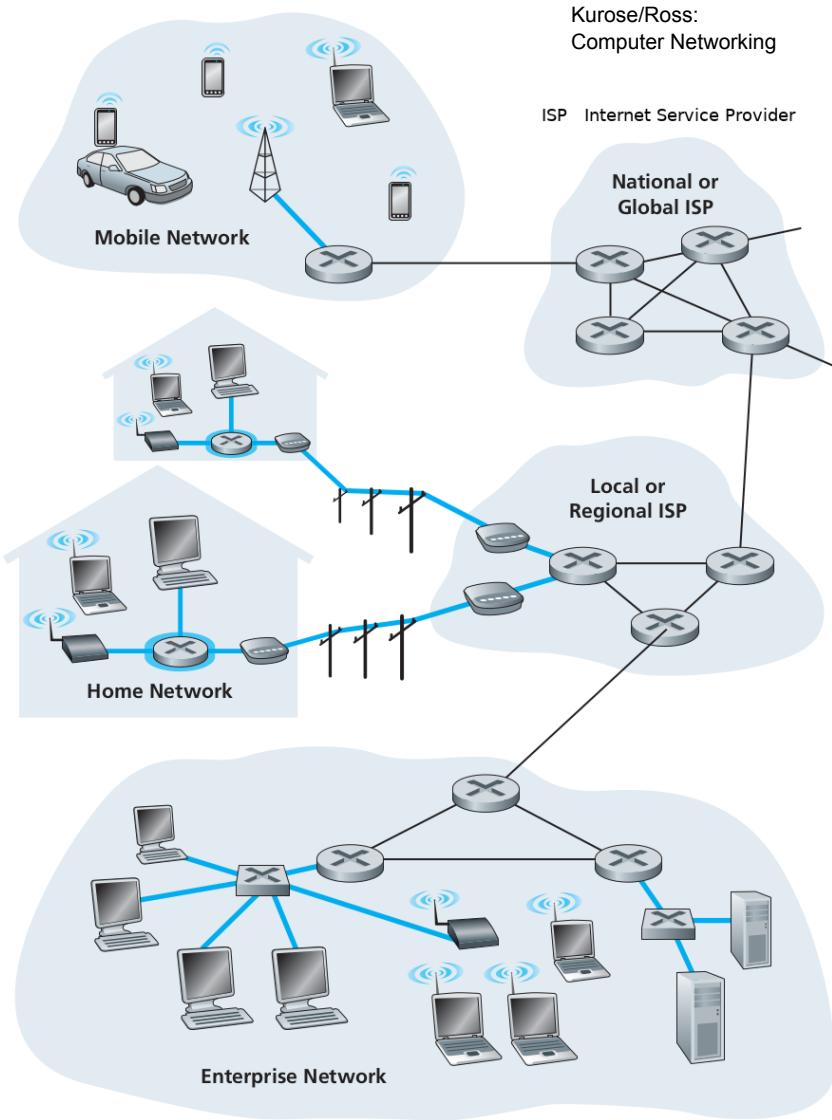


Internet: Kernstruktur



Das **Internet** ist ein System miteinander verbundener Netze („Internet“ ist zusammengezogen aus „interconnected networks“), das von vielen verschiedenen Internet-Dienstleistungsanbietern (internet service providers, ISPs) betrieben wird, die direkt oder über Austauschpunkte (internet exchange points, IXPs) verbunden sind.

Internet: Zugriffsnetze

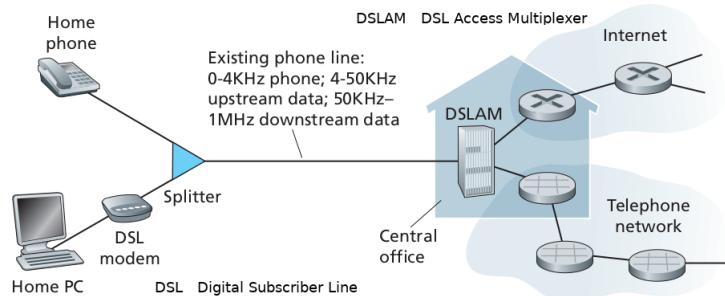


- Netze, die Endsysteme mit der ersten Vermittlungsstelle (router) verbinden, heißen **Zugriffsnetze (access networks)**.
- Sie können von sehr verschiedener Art sein, z.B. digital betriebene Telefonleitungen (digital subscriber line, DSL), Koaxialkabel für Kabelfernsehen, Glasfaserkabel, Drahtlosverbindungen z.B. über das Mobiltelefonnetz, Ethernetkabel etc.

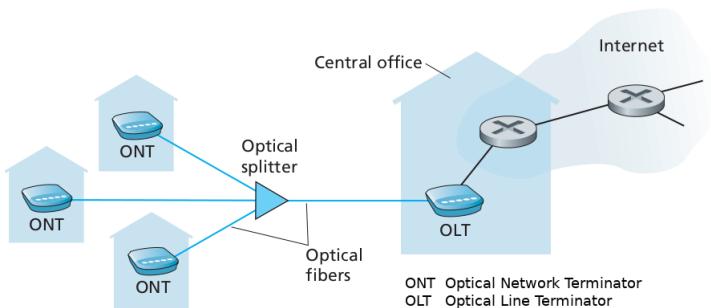


Internet: Zugriffsnetze

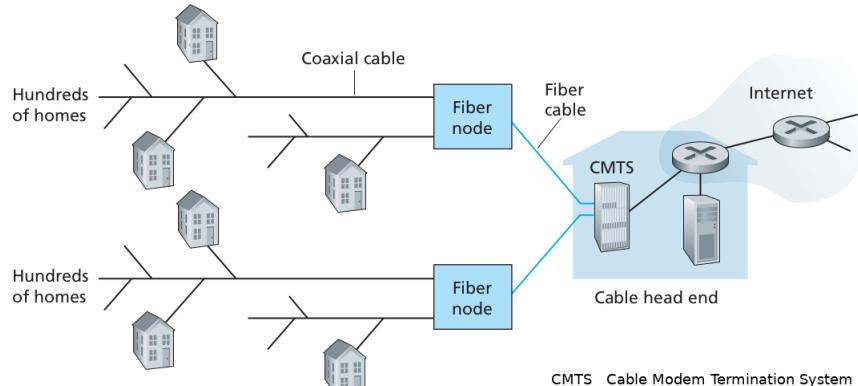
Zugriff über Telefonleitungen



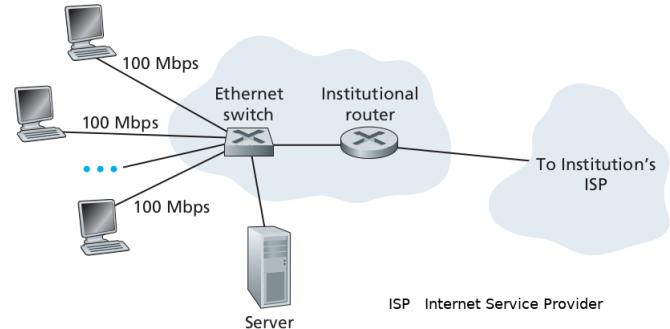
Zugriff über Glasfaserkabel



Zugriff über Fernsehkabel

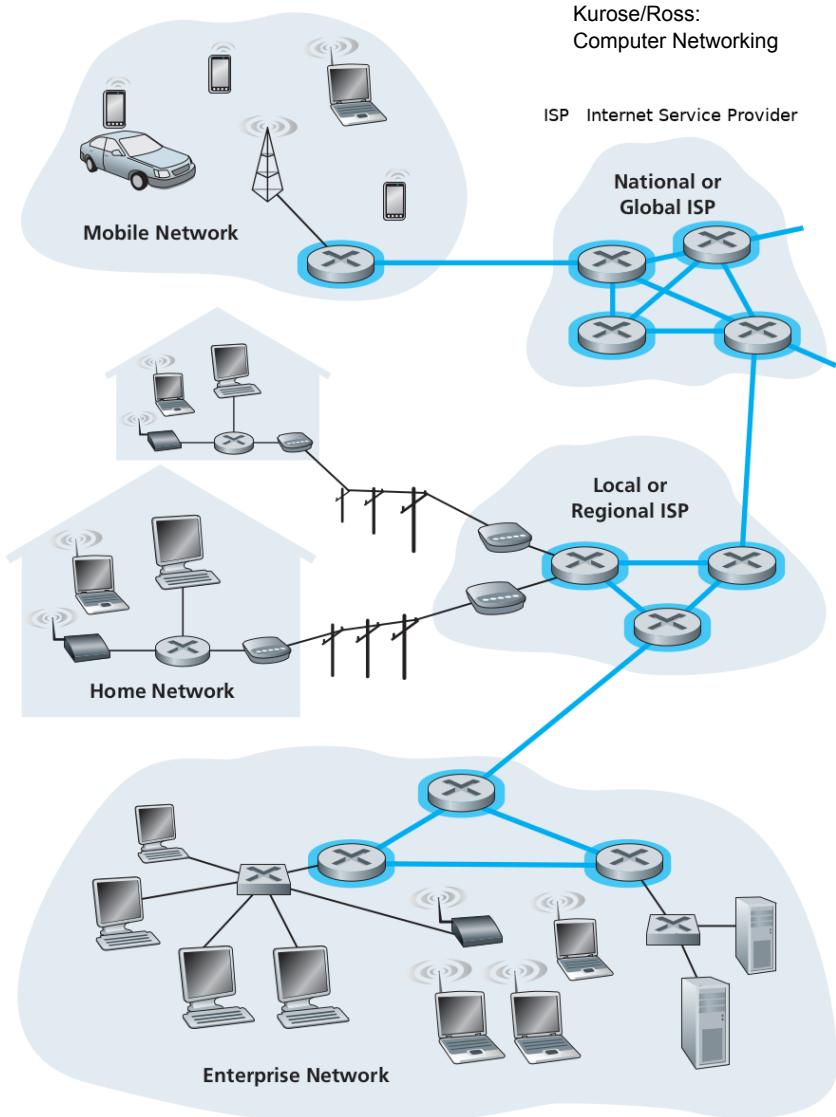


Zugriff über Ethernetkabel



Ethernet-Verbindungen werden innerhalb einer Firma / Institution oder auch eines Haushalts eingesetzt.

Datenübertragung in Rechnernetzen

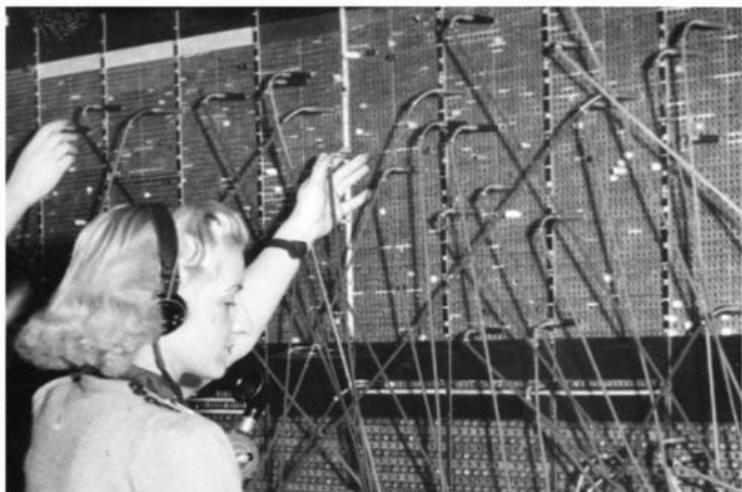


- Ein **Rechnernetz** ist ein Netz von miteinander verbundenen **Vermittlungsstellen** (routers), über die Nachrichten (messages) in Form von Datenpaketen (packets) übertragen werden.
- Warum wurde für die Datenübertragung in Rechnernetzen die Paketform gewählt?
- Welche Alternativen hätte es gegeben und warum wurden diese verworfen?

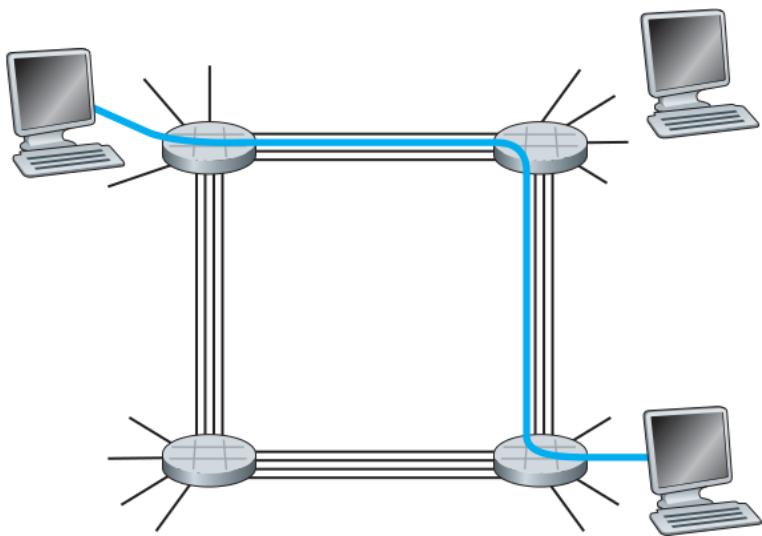


Leitungsvermittlung

Technoseum

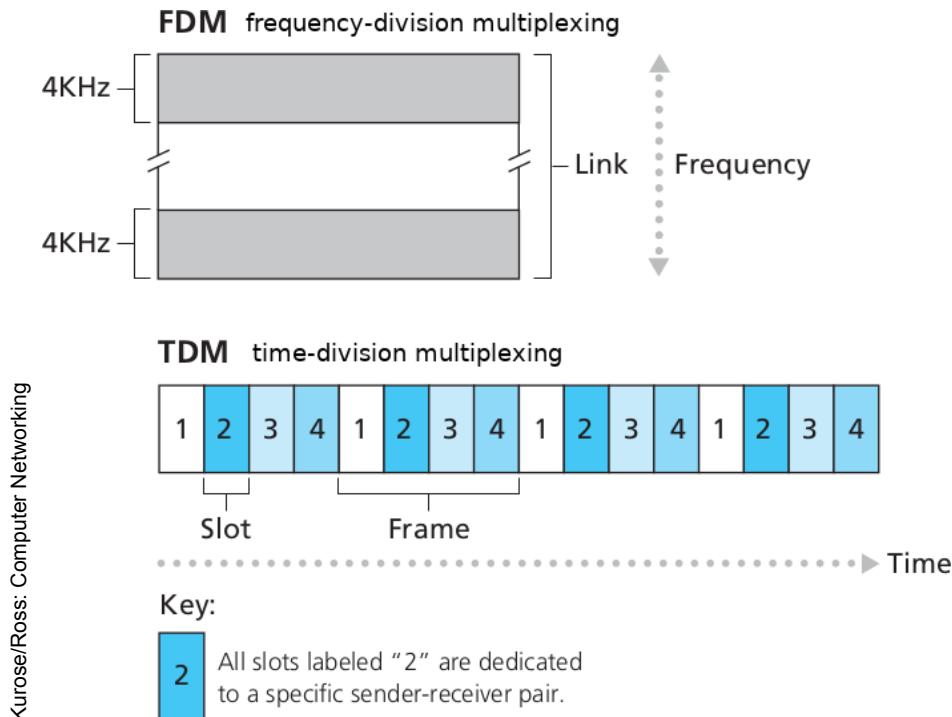


Kurose/Ross: Computer Networking



- Die klassische Methode von Telefonnetzen, die im Prinzip auch für Rechnernetze genutzt werden könnte, ist die **Leitungsvermittlung**.
- Die Vermittlungsstellen sind durch Leitungen verbunden, von denen jeweils eine dem zu vermittelnden Telefongespräch zugeordnet wird.
- Nachteil: Die volle Übertragungskapazität der Leitung ist für das Gespräch reserviert, auch wenn sie nicht genutzt wird (besonders bei Schweigepausen in einem Gespräch).
- Außerdem ist die Zahl der gleichzeitig möglichen Telefongespräche durch die Leitungszahl beschränkt.

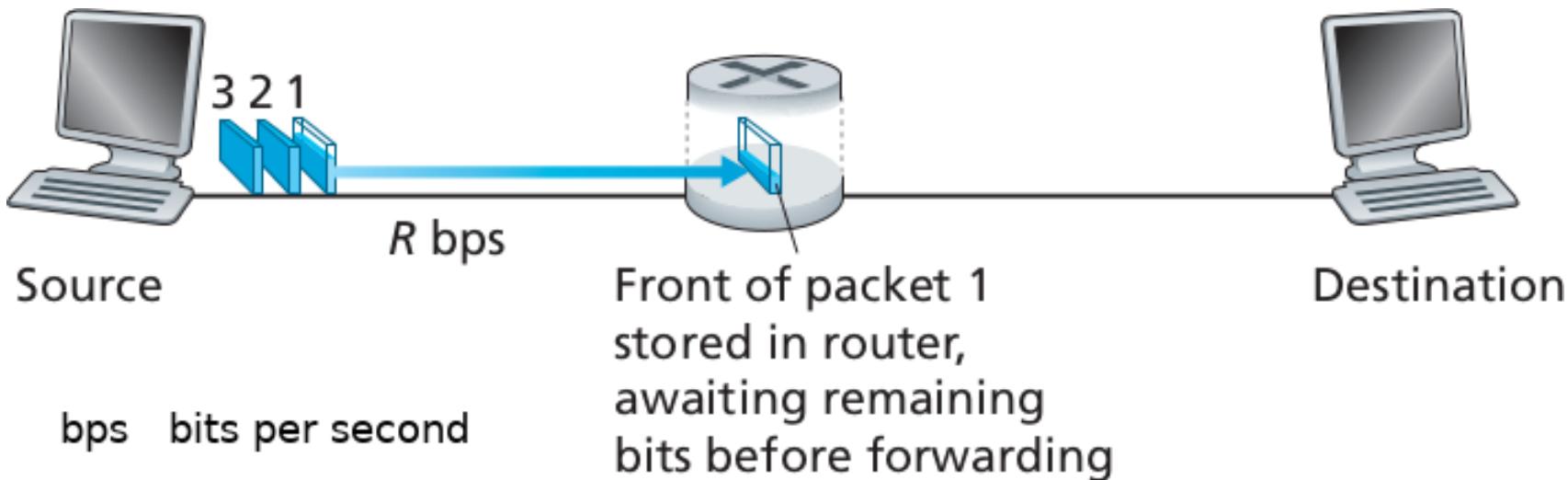
Frequenz- und Zeit-Multiplexverfahren



- Durch **Multiplexverfahren** lässt sich die Bandbreite einer Leitung besser ausnutzen.
- **Frequenzmultiplex** (frequency division multiplexing, FDM) nutzt aus, daß ein Telefongespräch nur ca. 4kHz benötigt, eine Leitung (link) aber ein viel größeres Spektrum übertragen kann.
⇒ Aufteilung in Bänder

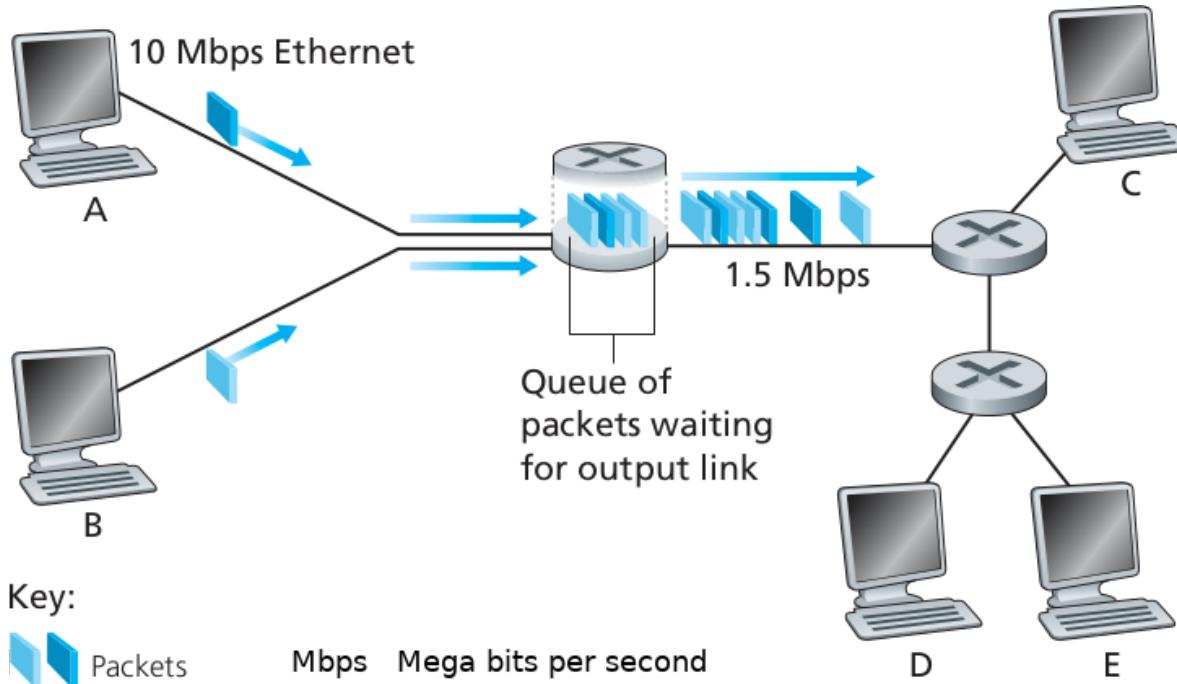
Paketvermittlung: Speichern und Weiterleiten

- Noch besser lassen sich Übertragungskapazitäten von Leitungen durch eine zeitliche **Zuordnung nach Bedarf und Verfügbarkeit** ausnutzen.
- Grundprinzip: Die zu übertragenden Daten werden in **Datenpakete** aufgeteilt. Ein Datenpaket wird übertragen, wenn die Leitung verfügbar ist.
- Es muß sichergestellt werden, daß nicht nur Datenpakete einer Verbindung übertragen werden. Meist reicht: „Wer zuerst kommt, mahlt zuerst.“ auf Paketebene.
- **Speichern und Weiterleiten:** In einer Vermittlungsstelle wird ein Datenpaket empfangen und zwischengespeichert. Erst wenn das gesamte Paket empfangen wurde, wird es weitergeleitet (wenn die Ausgangsleitung frei ist).



Paketvermittlung: Warteschlangen

- Von verschiedenen Quellen an einer Vermittlungsstelle (router) ankommende Datenpakete werden in eine **Warteschlange** eingereiht.
- Das Paket am Kopf der Warteschlange wird als nächstes über die Ausgangsleitung übertragen (Prinzip: first in, first out, FIFO).
- Prinzipiell ähnlich zu Zeitmultiplex, aber ohne feste Zuordnung der Zeitabschnitte.



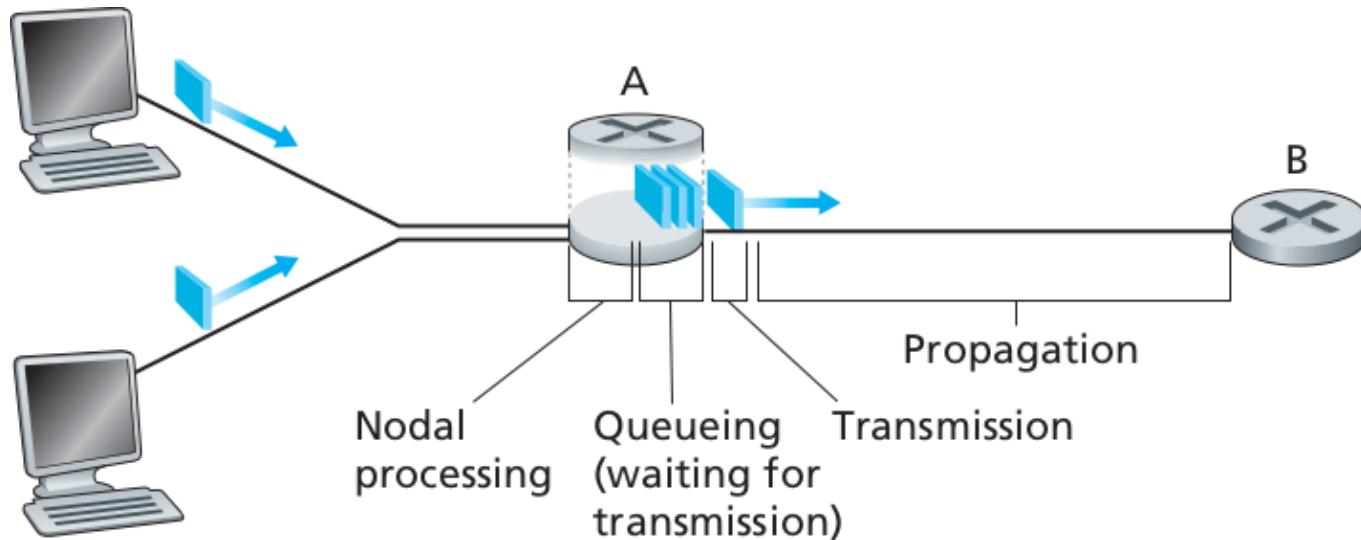
Kurose/Ross: Computer Networking

Paketvermittlung: Übertragungskapazität

- Eine Signaländerung am Anfang einer (elektrischen) Leitung breitet sich entlang der Leitung etwa mit Lichtgeschwindigkeit aus.
- Die reine **Signallaufzeit** oder **Laufzeitverzögerung** (propagation delay) ist die Zeit, die vergeht, bis ein Signal am Leitungsende ankommt. Sie ist daher die Leitungslänge geteilt durch die Lichtgeschwindigkeit.
- Aber Signaländerungen können nicht beliebig schnell aufeinander folgen, wenn am Ende der Leitung ein hinreichend rauschfreies Signal ankommen soll.
- Eine Leitung hat daher eine endliche **Übertragungskapazität** oder **Leitungskapazität**, die in Bits pro Sekunde (bits per second, bps) angegeben wird.
- Sie hängt von der Kodierung der zu übertragenden Bits ab und davon, wie hohe Frequenzen auf der Leitung hinreichend rauschfrei übertragen werden können.
- Typische Übertragungskapazitäten liegen heute in der Größenordnung von Megabit (10^6) oder Gigabit (10^9) pro Sekunde (Mbps bzw. Gbps).

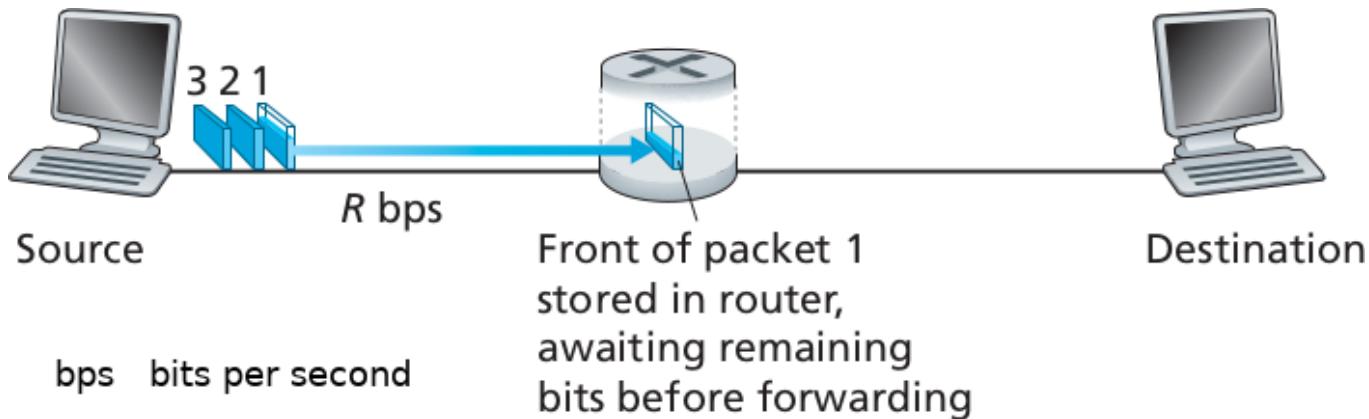
Paketvermittlung: Verzögerungen

- Bei Leitungsvermittlung hängt die Zeit, die vergeht, bis gesendete Daten bei einem Empfänger ankommen, nur von der **Laufzeitverzögerung** (propagation delay) und der **Übertragungskapazität** (transmission capacity) der Leitung ab.



- Bei Paketvermittlung kommen weitere Verzögerungen hinzu:
 - **Übertragungsverzögerung** (transmission delay)
 - **Wartezeit in Warteschlange** (queueing delay)
 - **Verarbeitungszeit** (nodal processing delay)

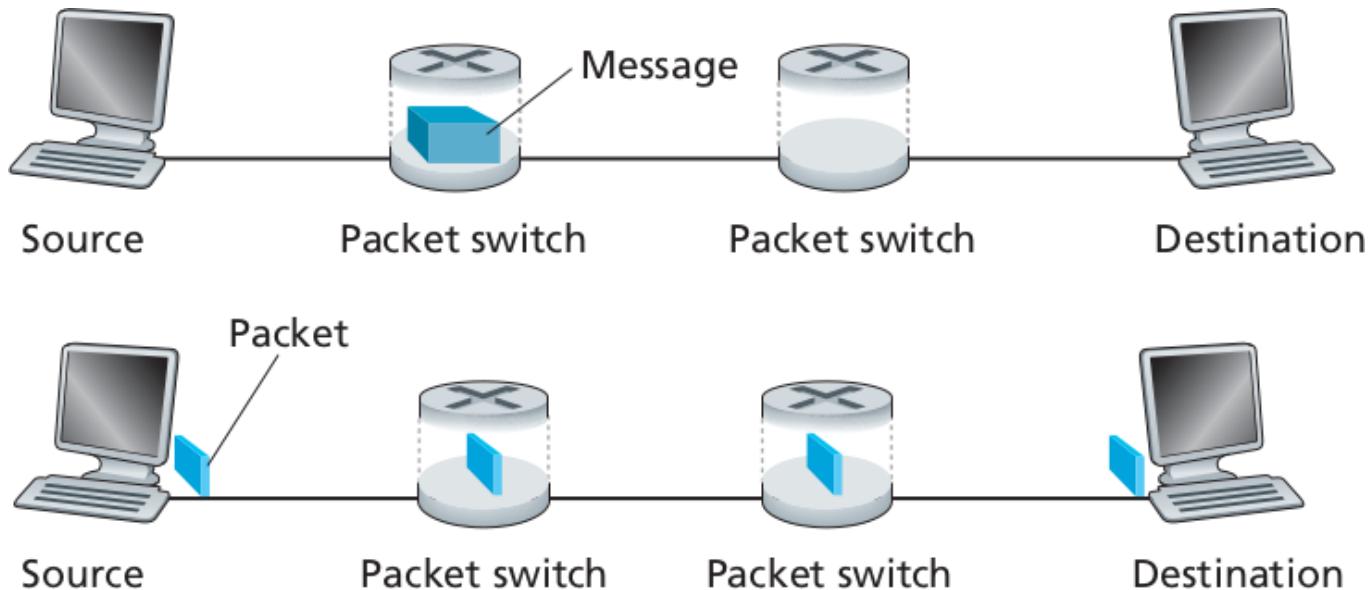
Paketvermittlung: Übertragungsverzögerung



Kurose/Ross:
Computer Networking

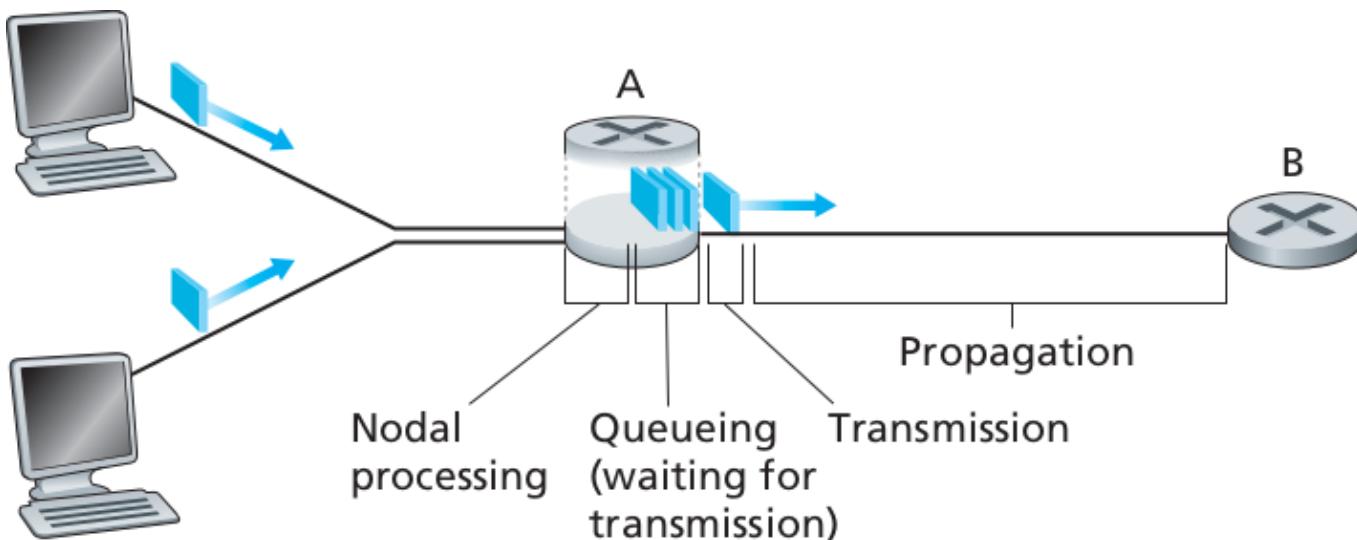
- Die **Übertragungsverzögerung** (transmission delay) entsteht dadurch, daß ein Datenpaket in einer Vermittlungsstelle (router) erst vollständig empfangen wird, ehe es weitergeleitet wird (Speichern und Weiterleiten).
- Diese Verzögerung ist die Zeit, die vergeht zwischen
 - dem Senden (bzw. Empfangen) des ersten Bits eines Pakets und
 - dem Senden (bzw. Empfangen) des letzten Bits dieses Pakets.
- Diese Verzögerung hängt offenbar von der Übertragungskapazität und der Größe der Datenpakete ab (je kleiner die Pakete, desto geringer die Verzögerung). Sie entsteht außerdem in jeder durchlaufenen Vermittlungsstelle erneut.

Paketvermittlung: Übertragungsverzögerung



- Die **Übertragungsverzögerung** (*transmission delay*) entsteht dadurch, daß ein Datenpaket in einer Vermittlungsstelle (router) erst vollständig empfangen wird, ehe es weitergeleitet wird (Speichern und Weiterleiten). Sie entsteht in jeder durchlaufenden Vermittlungsstelle erneut.
- Ohne Segmentierung entsteht erhebliche Verzögerung.
Je kleiner die Pakete, desto geringer die Verzögerung.
Je größer die Pakete, desto größer auch der benötigte Speicher.

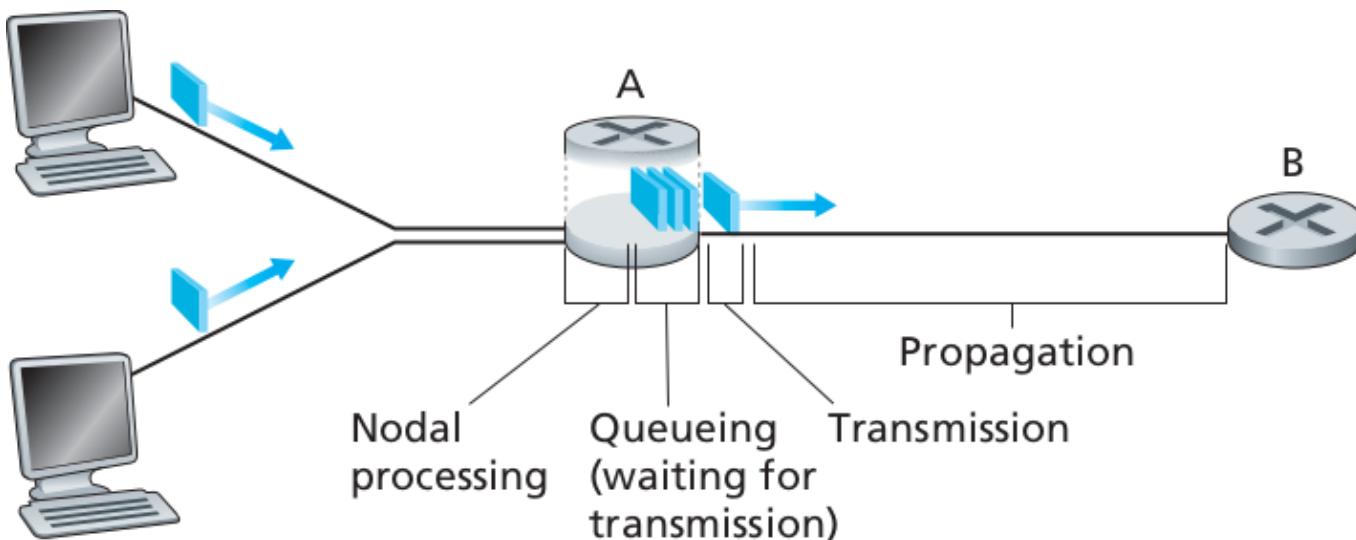
Paketvermittlung: Wartezeit in Warteschlange



Kurose/Ross: Computer Networking

- Die **Wartezeit in der Warteschlange** (queueing delay) entsteht, wenn ein empfangenes Paket nicht sofort weitergeleitet werden kann, weil die Ausgangsleitung belegt ist. (Es wird gerade ein früher empfangenes Datenpaket übertragen).
- Laufen je Zeiteinheit mehr Pakete ein, als auf der Ausgangsleitung übertragen werden können, wächst die Warteschlange, laufen weniger ein, schrumpft sie.
- Da die Warteschlange eine endliche maximale Länge hat (Speicherkapazität ist zwangsläufig begrenzt), kann sie überlaufen, wodurch es zu **Paketverlust** kommt.

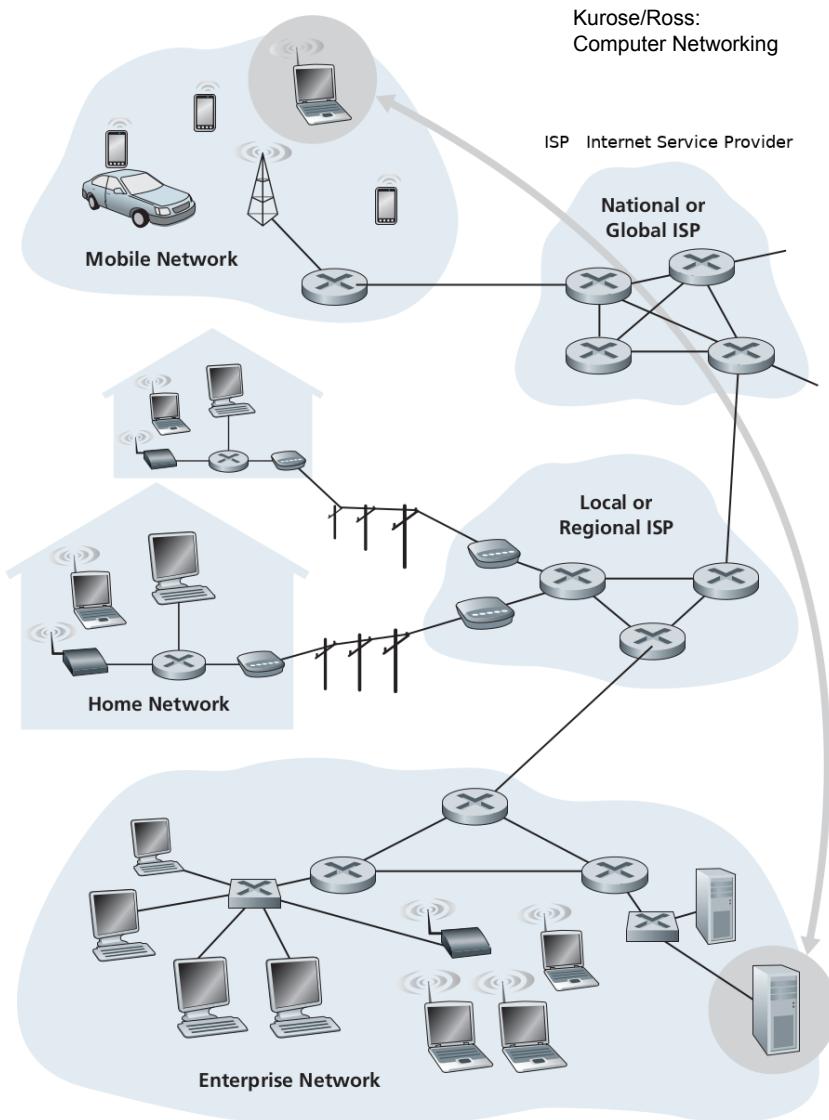
Paketvermittlung: Verarbeitungszeit



Kurose/Ross: Computer Networking

- Die **Verarbeitungszeit** (nodal processing delay) entsteht durch die rechentechnische Verarbeitung eines Paketes in einer Vermittlungsstelle.
- Z.B. muß ein ankommendes Paket einer Stelle in der Warteschlange zugeordnet und seine Daten dort abgelegt werden (dies erfordert ggf. ein Kopieren der Daten).
- Weiter muß ein Datenpaket, wenn zur Übertragung ansteht, vom Kopf der Warteschlange abgeholt werden (auch dies erfordert ggf. ein Kopieren der Daten).
- Diese Verzögerung entsteht in jeder Vermittlungsstelle, ist aber meist sehr gering.

Internet: Kommunikation zwischen Endsystemen



- Das Internet ermöglicht Kommunikation zwischen Endsystemen, wie sie in der ersten Vorlesung betrachtet wurde:
- Auf einem Rechner wird ein **Stöberer** (browser) geöffnet, um die Webseite der Universität Konstanz aufzurufen.
- Diese Webseite wird von einem Endsystem an der Universität Konstanz bereitgestellt, einem **Bediener** (server).



Rechnernetze: Webseitenaufruf

Einfaches Beispiel: Wir wollen die Webseite `www.uni-konstanz.de` aufrufen.

(Achtung: im folgenden stark vereinfachtes Modell!)

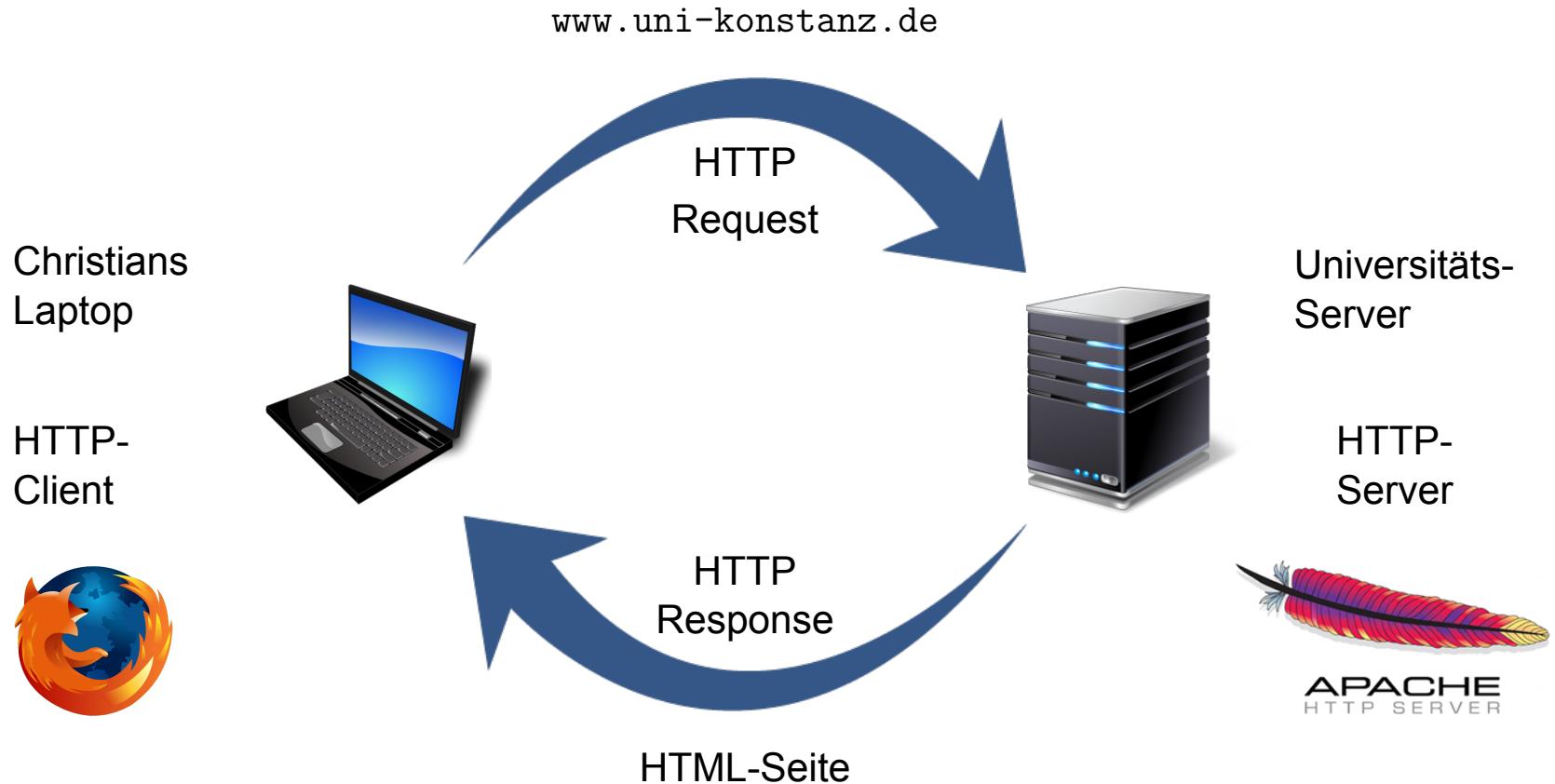


HTML: HyperText Markup Language (Beschreibungssprache hauptsächlich für Webseiten)

Rechnernetze: Webseitenaufruf

Einfaches Beispiel: Wir wollen die Webseite `www.uni-konstanz.de` aufrufen.

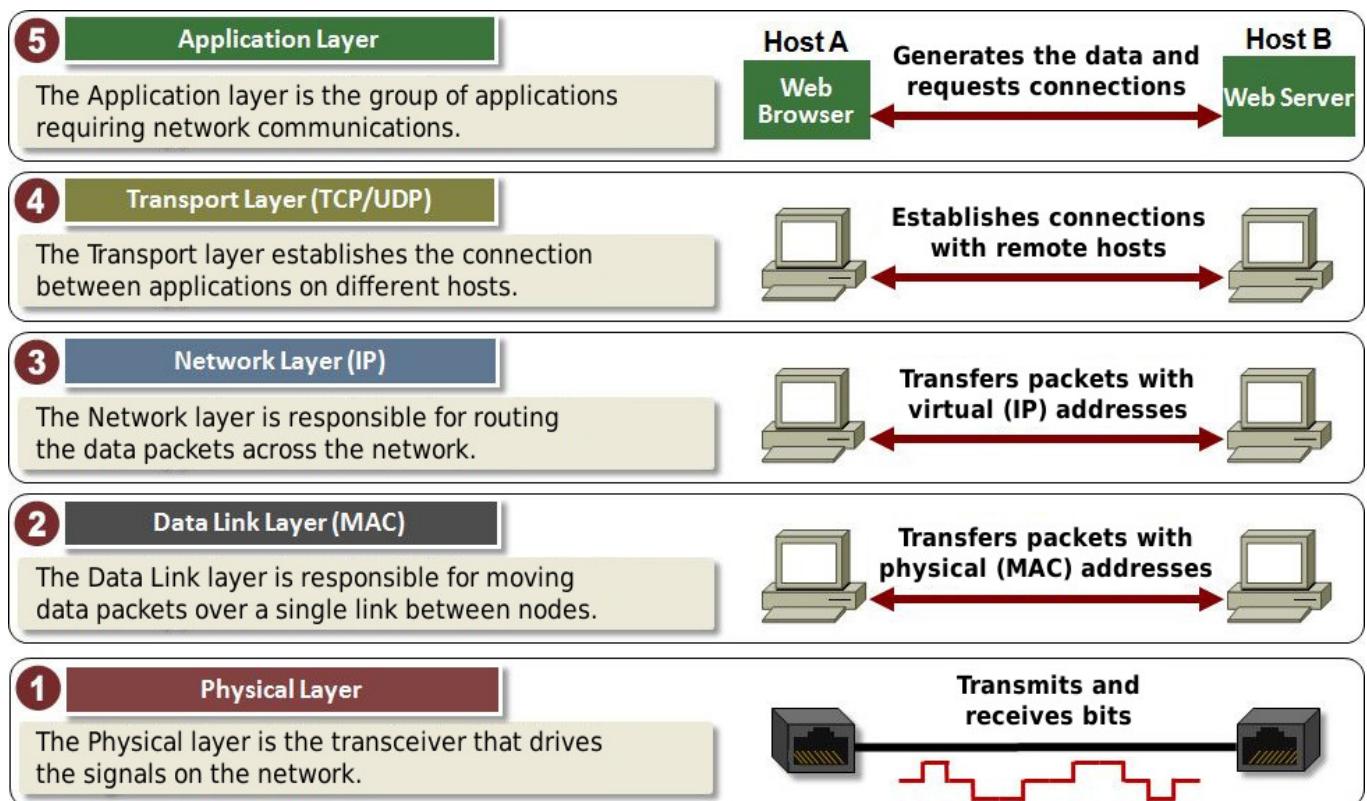
(Achtung: im folgenden stark vereinfachtes Modell!)



HTTP: HyperText Transfer Protocol (Übertragungsprotokoll hauptsächlich für Webseiten)

Rechnernetze: Webseitenaufruf

- Die Anfrage an den Universitäts-Server wird in mehreren Schritten bearbeitet und durch mehrere Abstraktionsschichten hindurchgeleitet.
- Jede dieser Schichten hat bestimmte, eng begrenzte Aufgaben, und auf jeder Schicht gibt es Netzwerkprotokolle mit klaren Schnittstellen.



Rechnernetze: Webseitenaufruf

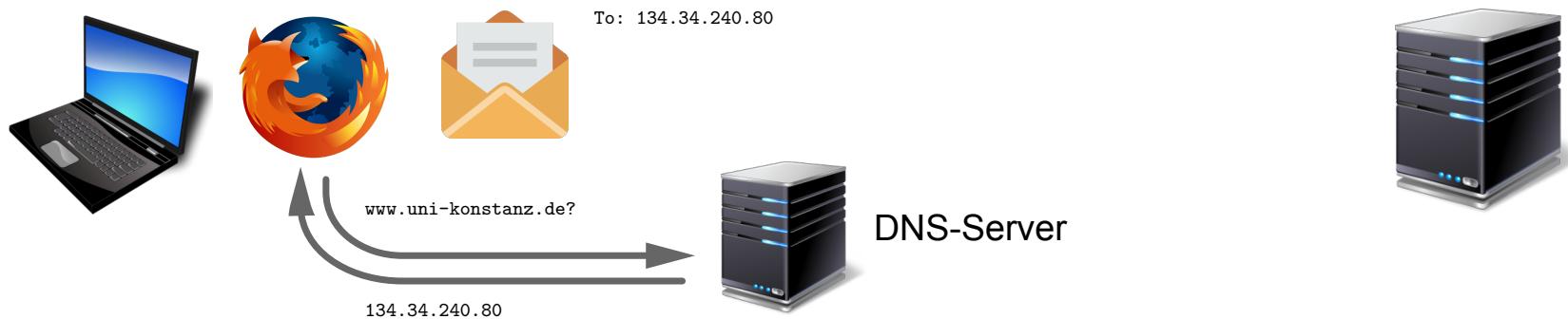


To: ???
www.uni-konstanz.de



- Der Internetbrowser Mozilla Firefox ist unser HTTP-Client.
Dies ist die **Anwendungsschicht** (application layer).
- Die HTTP-Anfrage wird z.B. durch Eintippen von `www.uni-konstanz.de` in die Adresszeile des Firefox-Browsers gestellt.
- Der Firefox-Browser kapselt die HTTP-Anfrage („steckt sie in einen Briefumschlag“) und gibt sie an die **Transportschicht** (transport layer) weiter.
- Aber: Die Transportschicht versteht die Adresse `www.uni-konstanz.de` nicht.
Adressen auf der Transportschicht sind **IP-Adressen** (IP: internet protocol).
Diese bestehen aus vier (IPv4) oder acht (IPv6) Zahlen, z.B. 134.34.240.80.

Rechnernetze: Domain Name System (DNS)



- Der HTTP-Client fragt einen DNS-Server (DNS: Domain Name System, gewissermaßen ein „Telefonbuch“ oder ein „Adressverzeichnis“) nach dem **Domain-Namen** `www.uni-konstanz.de`.
- Der DNS-Server übersetzt `www.uni-konstanz.de` in eine **IP-Adresse**, hier: `134.24.240.80` (IPv4) (IPv6 nicht verfügbar).
- An diese Adresse schickt nun der HTTP-Client (hier: Firefox-Browser) die Anfrage nach der Webseite der Universität Konstanz.
- An welchem „Briefkasten“ oder „Postfach“ (port) wird die Anfrage zugestellt?

Rechnernetze: Ports



To: 134.34.240.80:80

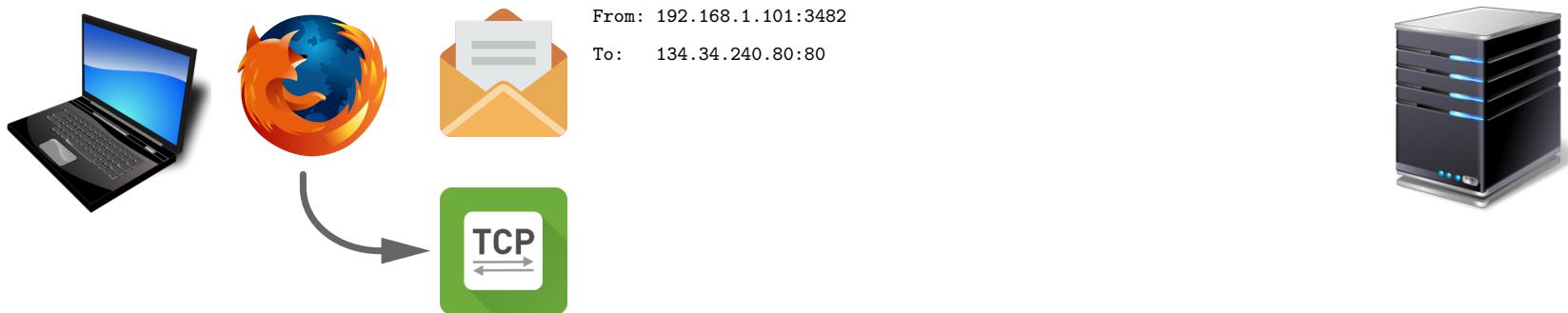
$\langle \text{ip-address} \rangle : \langle \text{port} \rangle$



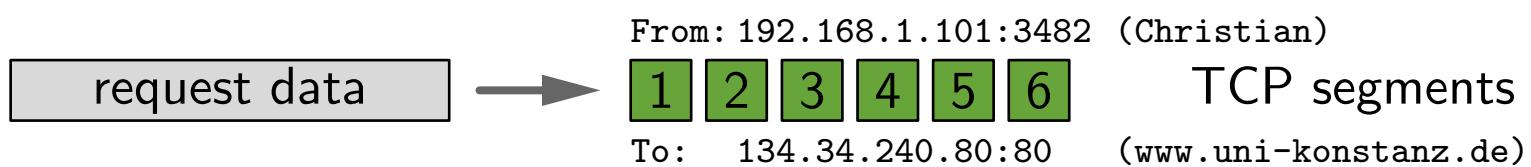
- **Ports** sind wie „Briefkästen“ oder „Postfächer“ für bestimmte Dienste (services). Analog hat z.B. der Lehrstuhl Bioinformatik und Information Mining das Postfach 712, um ihn innerhalb der Universität Konstanz zu identifizieren.
- Insgesamt gibt es $2^{16} = 65536$ Ports, von denen die unteren $2^{10} = 1024$ (also die Ports 0 bis 1023) für standardisierte (System-)Dienste vorgesehen sind, z.B.:

File Transfer Protocol (FTP)	20 (Daten) & 21 (Steuerung)
Secure SHell (SSH)	22
Simple Mail Transfer Protocol (SMTP)	25
Domain Name System (DNS)	53
HyperText Transfer Protocol (HTTP)	80

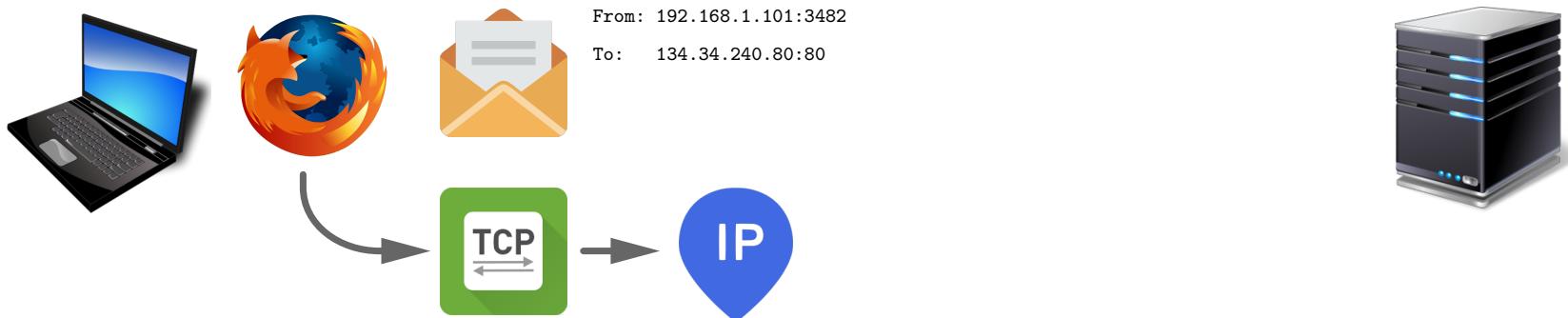
Rechnernetze: Transmission Control Protocol (TCP)



- Der HTTP-Client (Firefox) kapselt die HTTP-Anfrage und gibt sie an die **Transportschicht** weiter, die von einem **TCP-Client** verwaltet wird.
- Der TCP-Client baut die Verbindung zum Universitäts-Server auf (hier: zu Port 80 der IP-Adresse 134.34.240.80).
- Der TCP-Client segmentiert und numeriert die zu sendenden Daten, fügt Absender- und Empfängerinformationen und einige weitere Dinge hinzu.

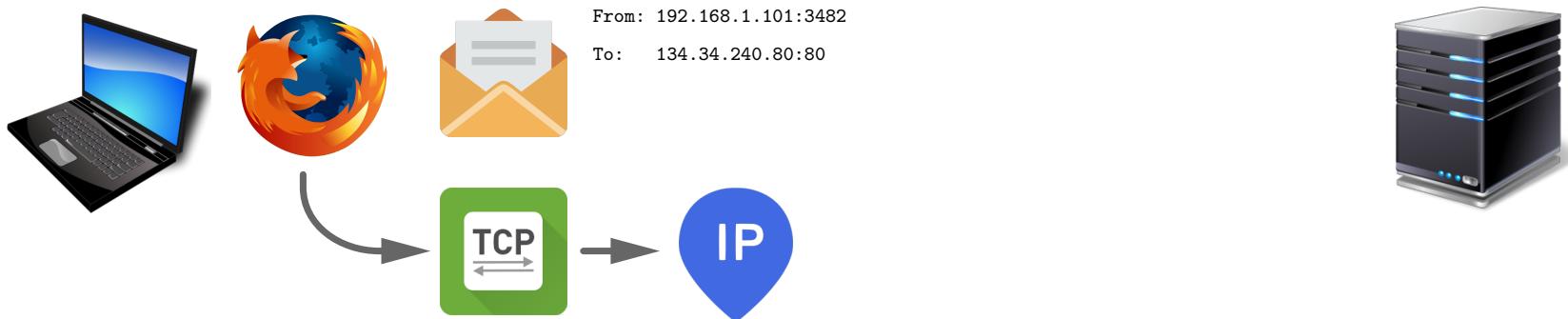


Rechnernetze: Internet Protocol (IP)

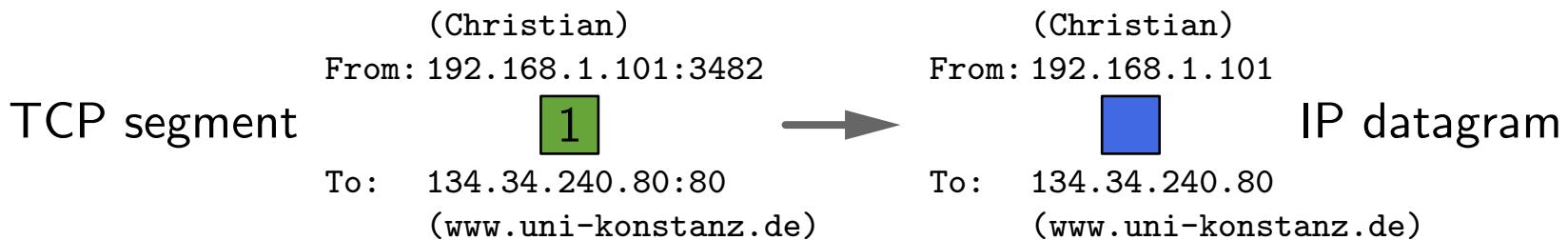


- Der TCP-Client übergibt die erzeugten Segmente/Pakete an das **IP-Modul** und beauftragt es, sie an die IP-Adresse 134.34.240.80 zu liefern.
- Auf dieser **Netzwerkschicht** ist eine direkte Auslieferung der Segmente/Pakete nur an Zieladressen **im gleichen Netzwerk** möglich.
- `www.uni-konstanz.de` ist nicht im gleichen Netzwerk wie Christians Laptop. (Christians Laptop ist in einem lokalen Netzwerk, local area network, LAN.)
- Das IP-Modul erzeugt daher ein **IP-Datagramm/IP-Paket** und sendet es an den **Standard-Netzübergang** (default gateway).

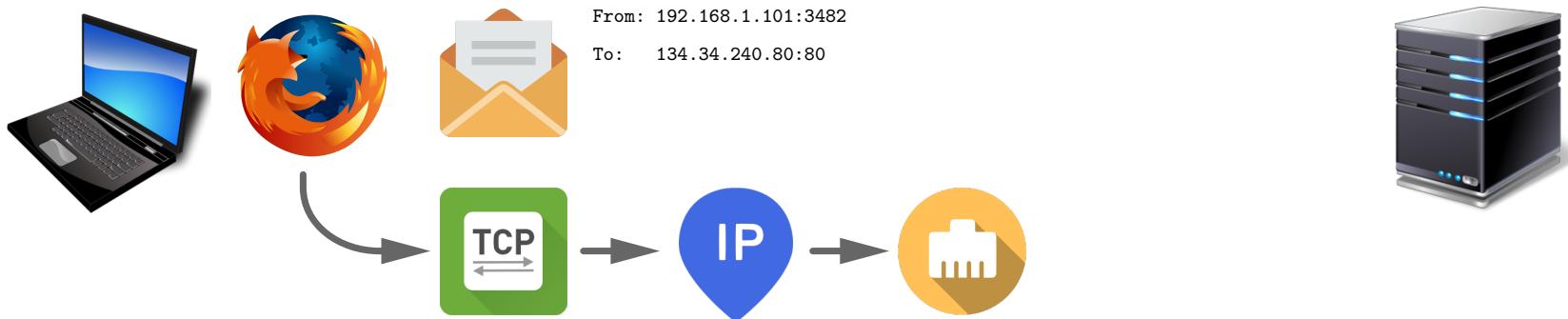
Rechnernetze: Internet Protocol (IP)



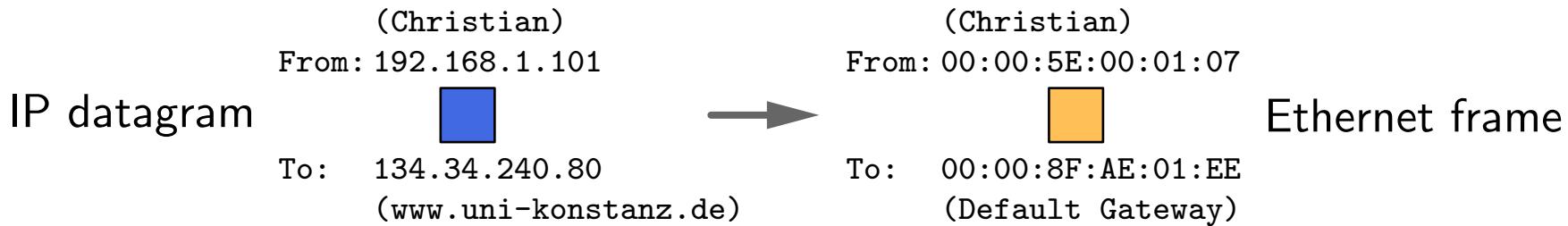
- Der TCP-Client übergibt die erzeugten Segmente/Pakete an das **IP-Modul** und beauftragt es, sie an die IP-Adresse 134.34.240.80 zu liefern.
- Das IP-Modul erzeugt daher ein **IP-Datagramm/IP-Paket** und sendet es an den **Standard-Netzübergang** (default gateway).



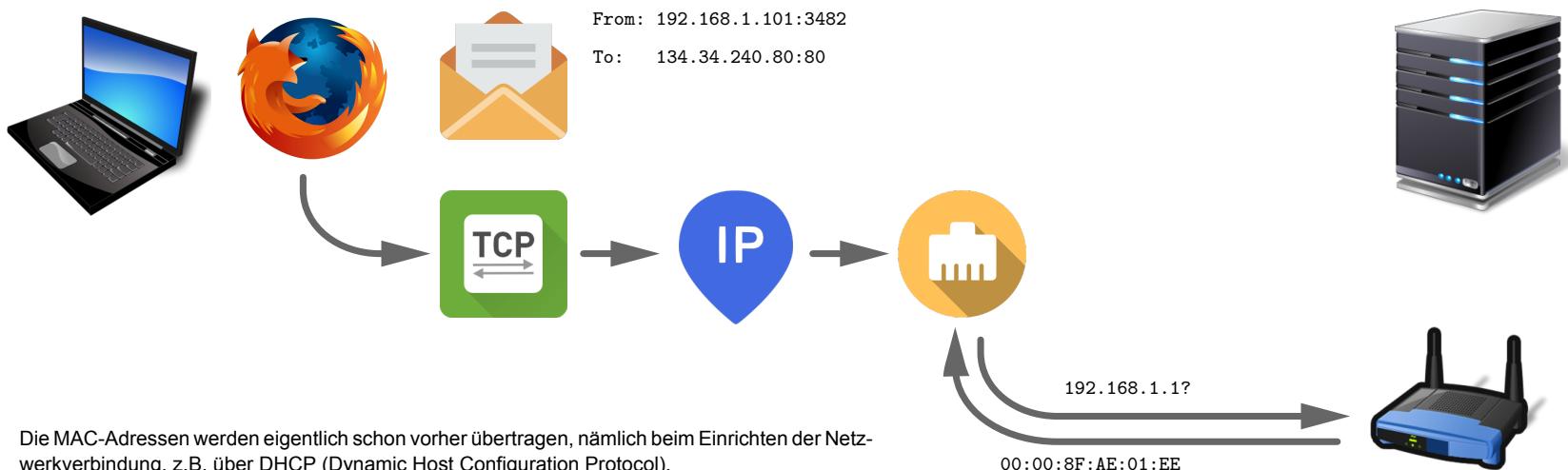
Rechnernetze: Internet Protocol (IP)



- Um es an den **Standard-Netzübergang** (default gateway) zu senden, muß das **IP-Datagramm** in einen **Ethernet-Datenrahmen** (Ethernet (data) frame) gepackt werden. Dies ist die **Verbindungsschicht**.
- Der Ethernet-Treiber versteht weder IP-Adressen noch Domain-Namen, sondern nur **MAC-Adressen** (MAC: Media Access Control).



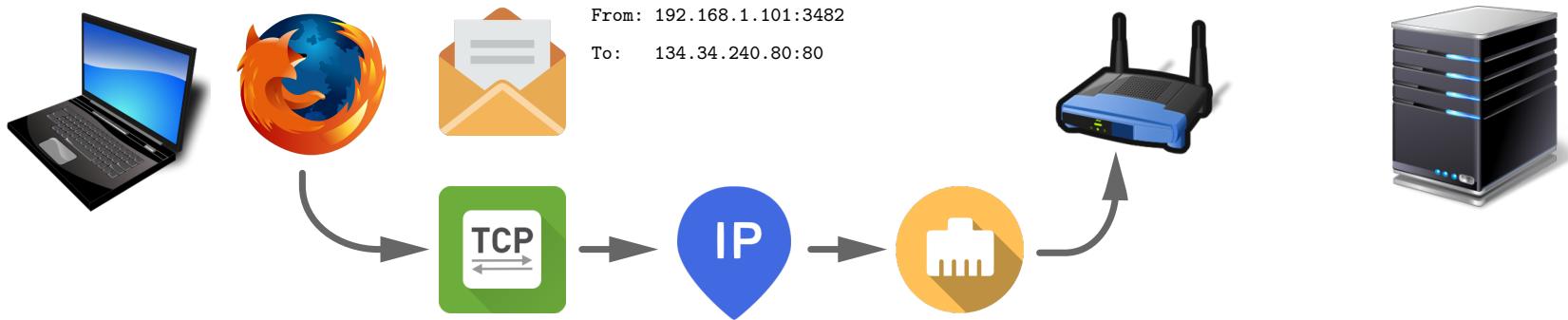
Rechnernetze: Ethernet



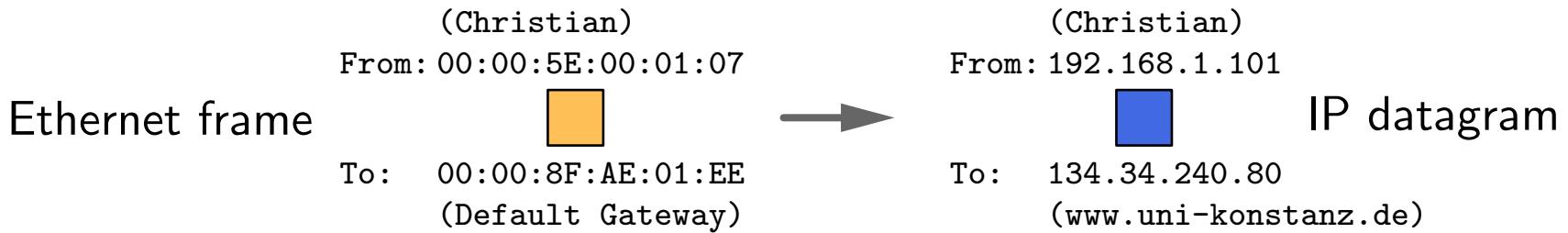
- Der Ethernet-Treiber versteht weder IP-Adressen noch Domain-Namen, sondern nur **MAC-Adressen** ([Media Access Control addresses](#)).
- Die Übersetzung wird über sogenannte **ARP-Tabellen** vorgenommen ([Address Resolution Protocol](#)), die z.B. im Rechner oder Router abgelegt sind.
- Der **Ethernet-Datenrahmen** ([Ethernet frame](#)) gelangt nun an den Router / Netzwerkübergang.



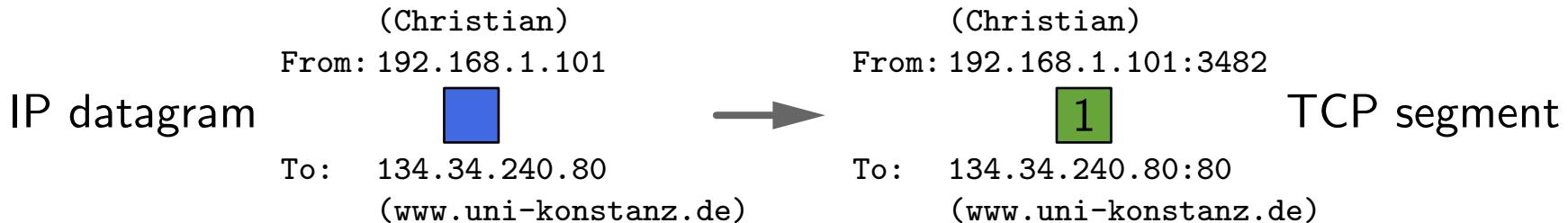
Rechnernetze: Router



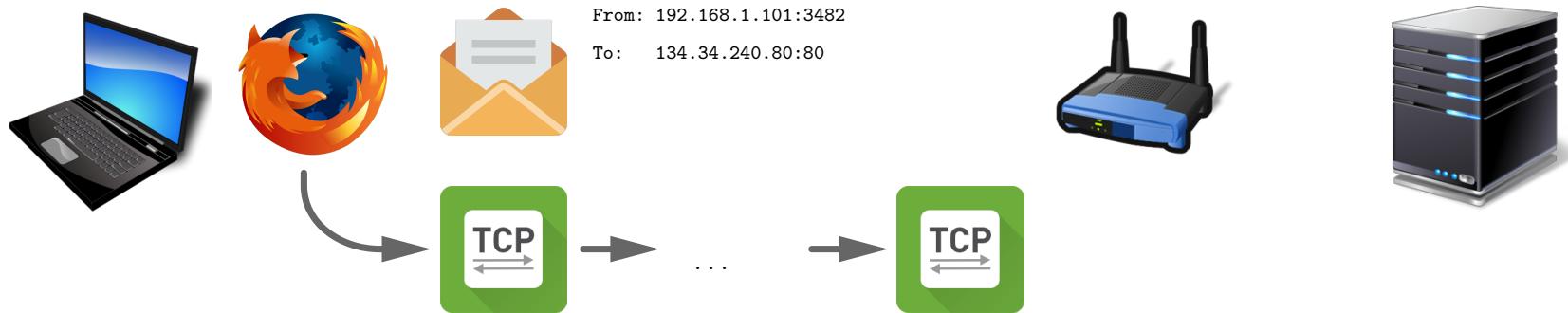
- Der Router entpackt den **Ethernet-Datenrahmen** (ethernet frame) ...



- ... und das **IP-Datagramm** bis zum **TCP-Segment**.



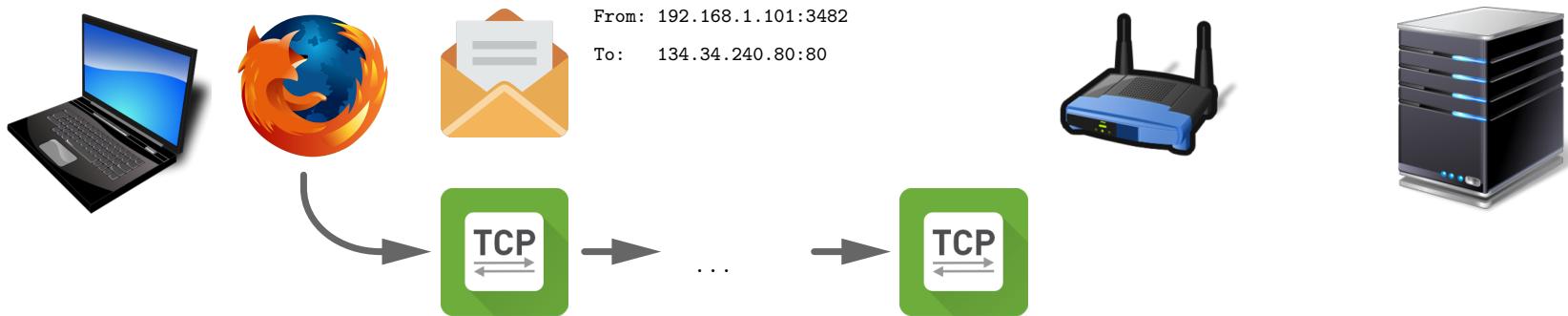
Rechnernetze: Router



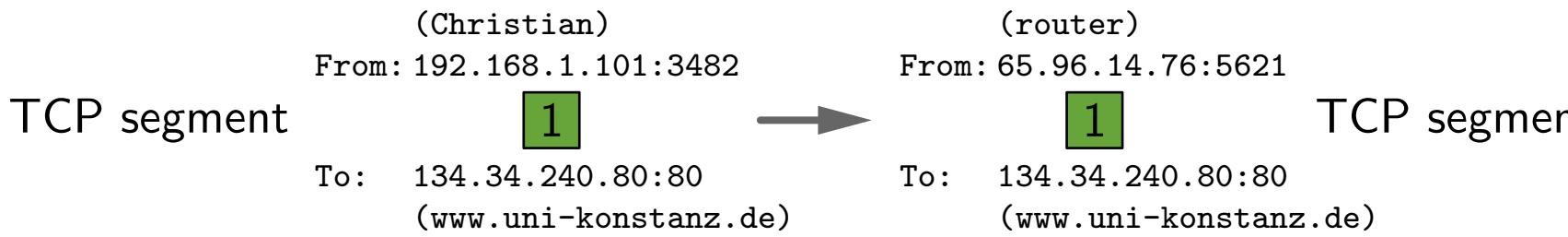
- Die Ziel-IP-Adresse des IP-Datagramms wird vom Router benutzt, um das nächste Zwischenziel auf dem Weg zum Web-Server zu bestimmen.
- Hierbei helfen **Wegetabellen** (*routing table, routing information base*). Wegetabellen enthalten Angaben zu möglichen Wegen, zu “optimalen” Wegen, zum Status, zur Metrik (z.B. Anzahl “Hops”), und zum Alter.
- Durch Wegetabellen werden Ziel-IP-Adressen mit Richtungsangaben in Form des Folgerouters und der zu verwendenden Schnittstelle verknüpft.

Wegeinformationen können statisch sein oder dynamisch gelernt werden.

Rechnernetze: Network Address Translation (NAT)



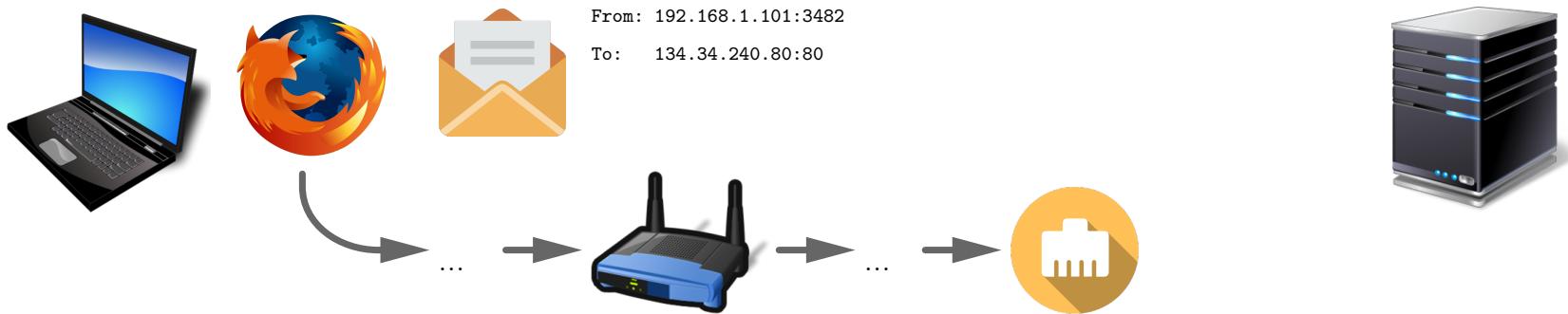
- Im TCP-Segment wird die **Quell-IP-Adresse** ersetzt durch die **Router-IP-Adresse** und einen spezifischen **Port**.



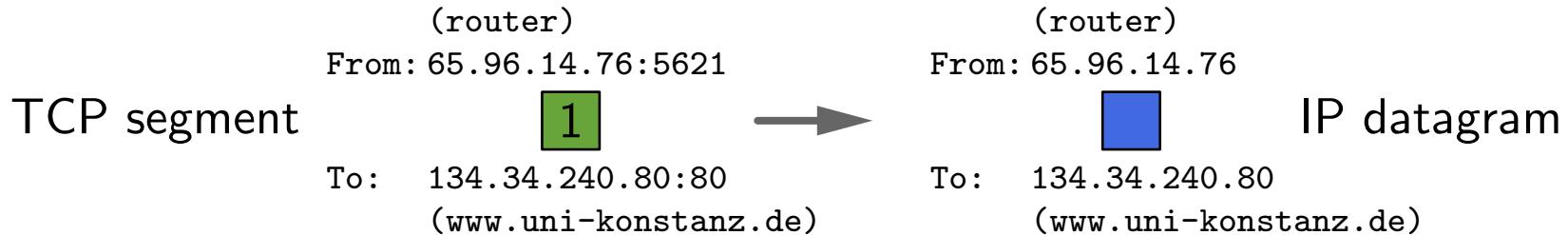
- In die **NAT-Tabelle** (NAT: Network Address Translation) des Routers wird die entsprechende Zuordnung eingetragen:

$$192.168.1.101:3482 \Leftrightarrow 65.96.14.76:5621$$

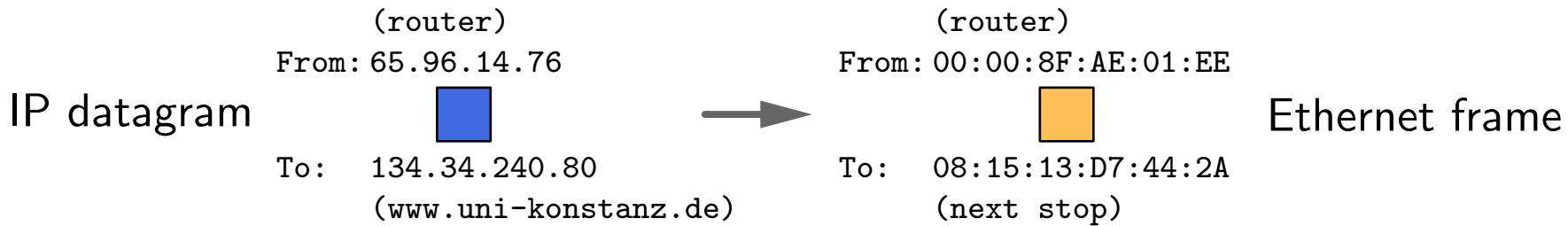
Rechnernetze: Network Address Translation (NAT)



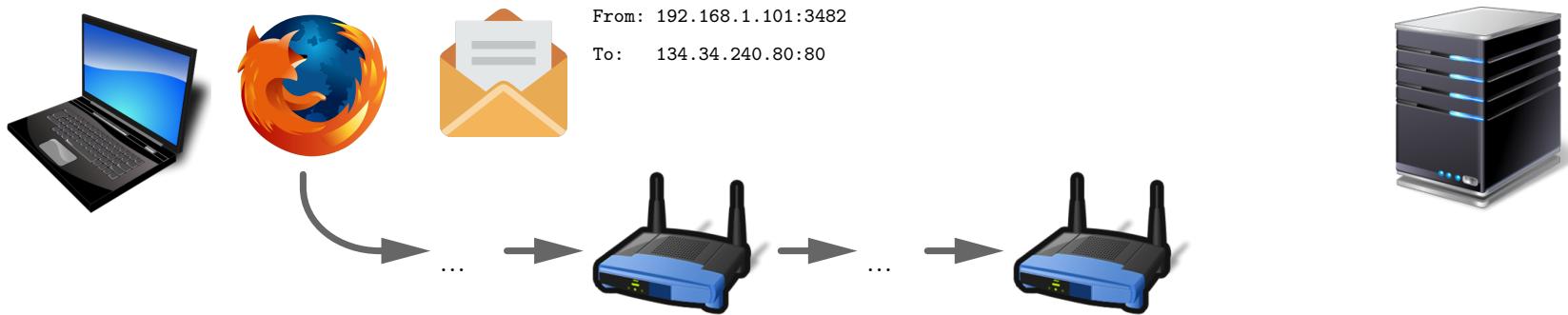
- Das modifizierte **TCP-Segment** wird in ein **IP-Datagramm** umgewandelt:



- Das **IP-Datagramm** wird in einen **Ethernet-Datenrahmen** umgewandelt:

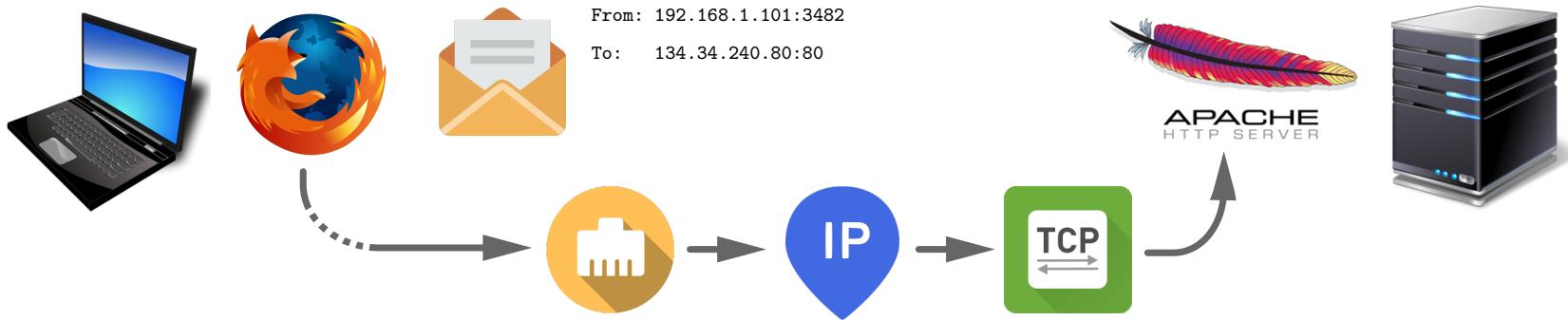


Rechnernetze: Network Address Translation (NAT)

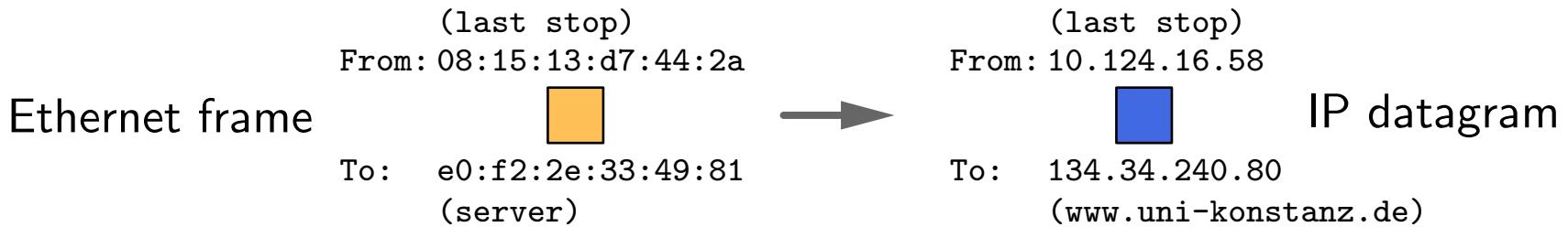


- Am nächsten Router wiederholt sich der Ablauf:
 - Der Ethernet-Datenrahmen wird zum IP-Datagramm und dieses zum TCP-Segment ausgepackt.
 - Das TCP-Segment bekommt eine neue Quelladresse für Antworten.
 - Das modifizierte TCP-Segment wird wieder zum IP-Datagramm und dieses zum Ethernet-Datenrahmen eingepackt.
 - Der Ethernet-Datenrahmen wird zum nächsten Router geschickt...
- ... bis schließlich der Zielrechner (hier: Universitätsserver) erreicht wird.

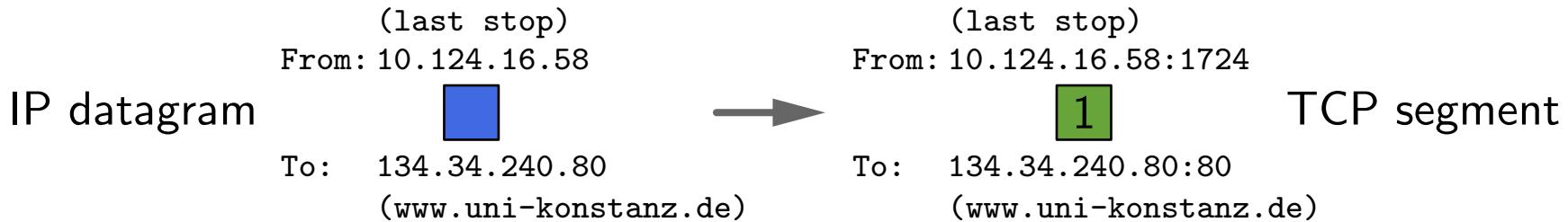
Rechnernetze: Server



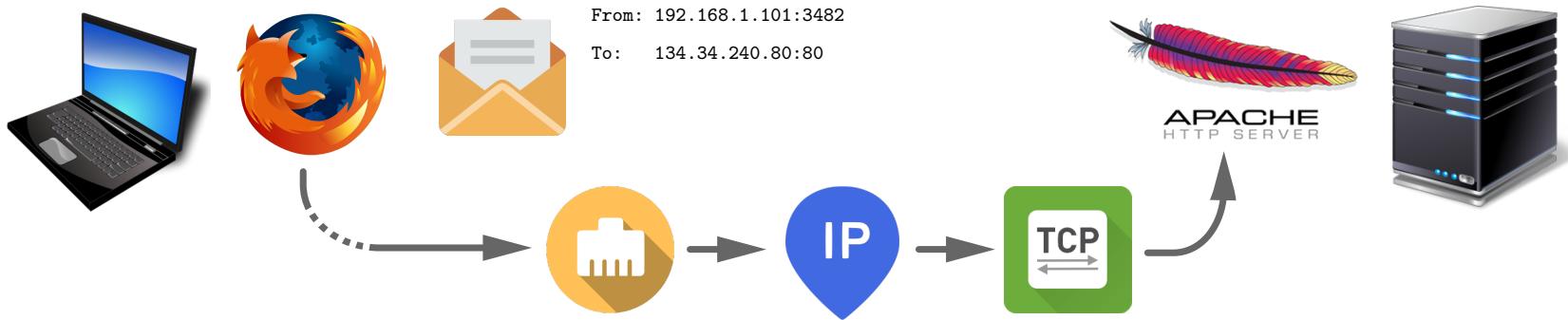
- Der Server entpackt den **Ethernet-Datenrahmen** (ethernet frame) ...



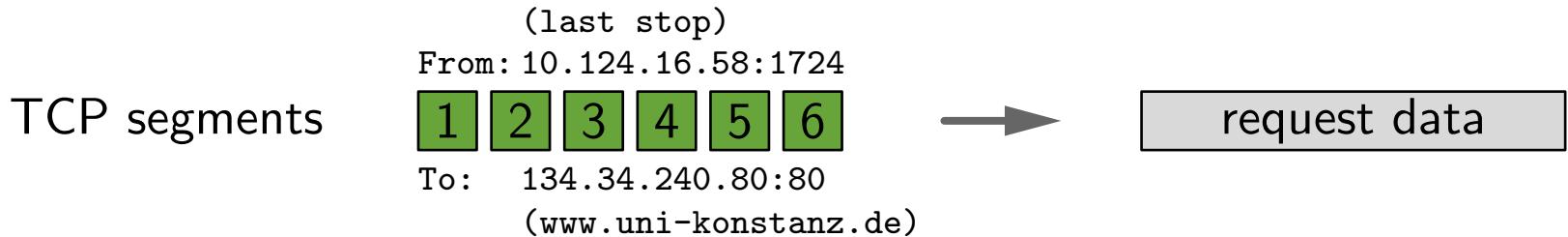
- ... und das **IP-Datagramm** bis zum **TCP-Segment**.



Rechnernetze: Server

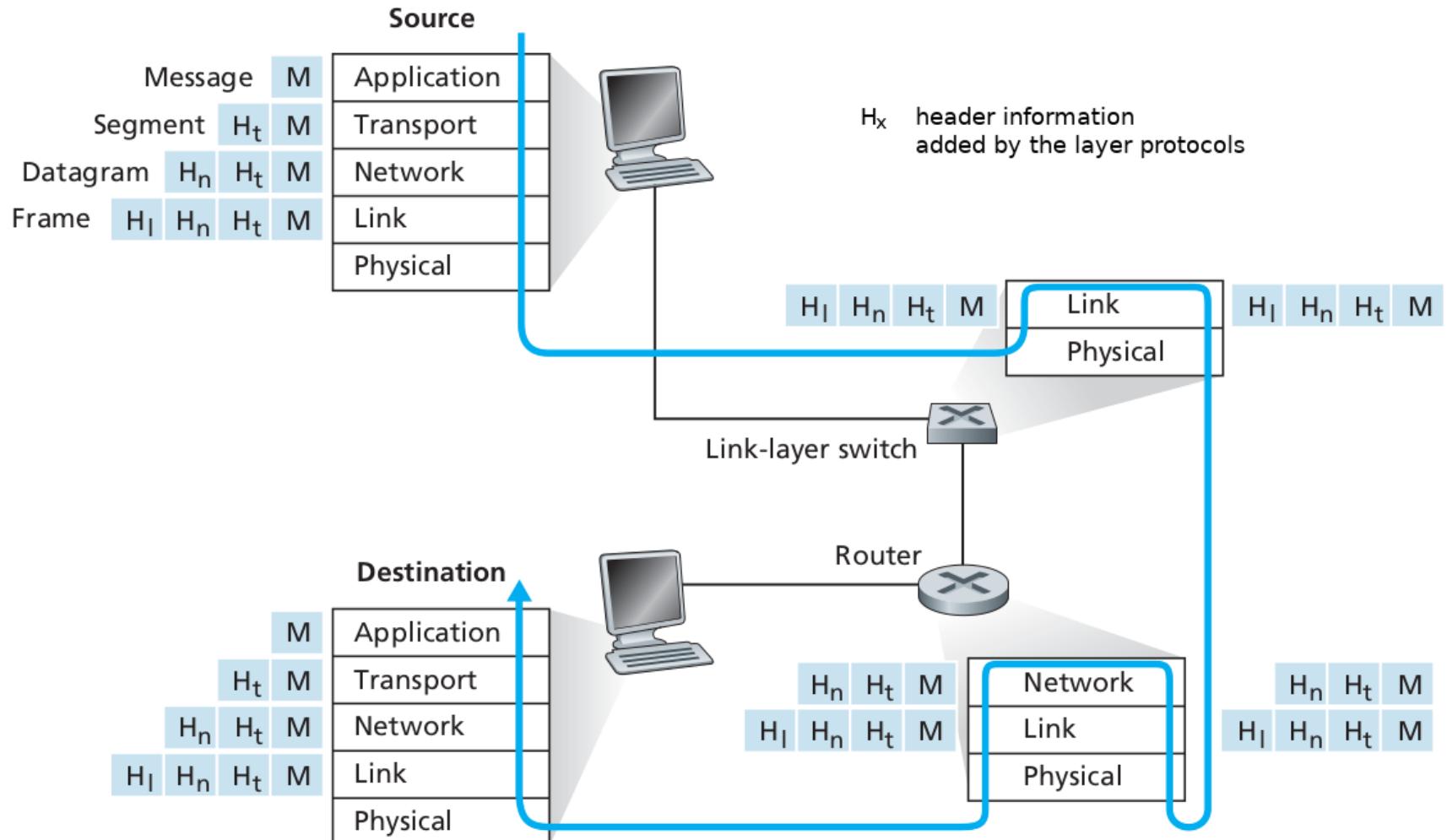


- Die einzelnen **TCP-Segmente** werden wieder zusammengesetzt:



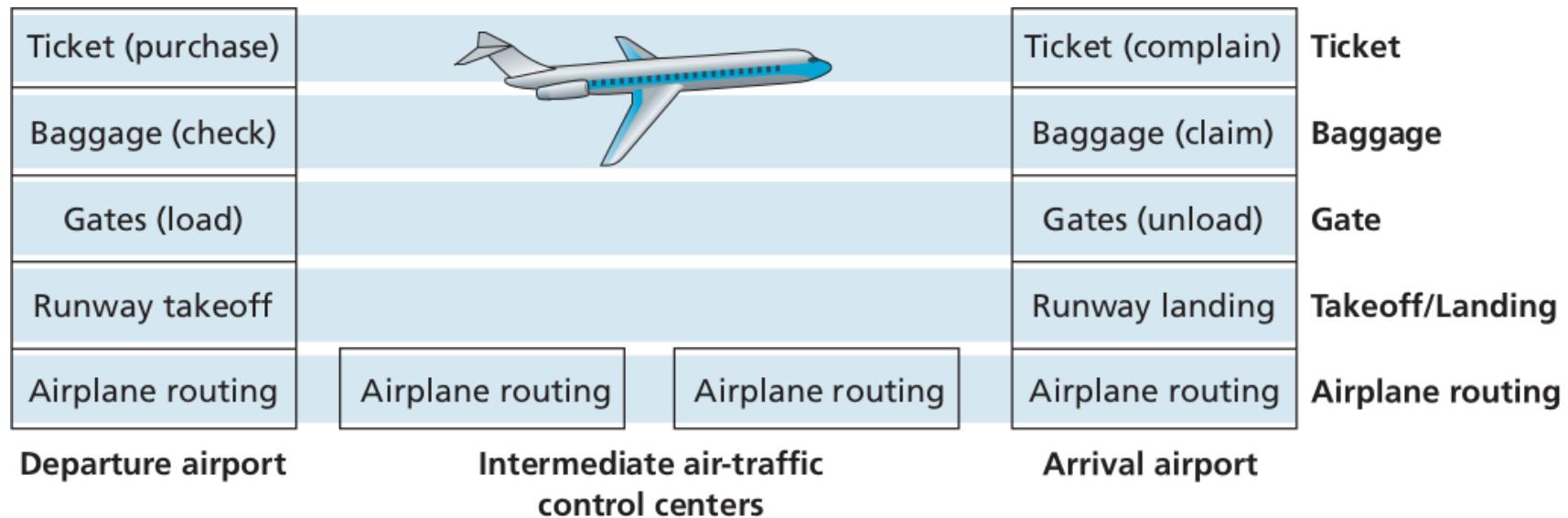
- Dem **HTTP-Server** auf dem Universitätsserver liegt nun die Anfrage nach der (Haupt-)Webseite der Universität Konstanz vor.
- Die Daten dieser Webseite werden nun auf gleiche Weise zurückgeschickt: Aufteilung in TCP-Segmente etc.

Kommunikation zwischen Endsystemen: Datenfluß



Kurose/Ross: Computer Networking

Steuerung der Übertragung: Abstraktionsschichten



- Aufteilung der Funktionalität in (abstrakte) **Schichten**.
- Vereinfacht Organisation, Wartung, Verbesserung etc. des Systems.
- Jede Schicht implementiert einen bestimmten Dienst mit Hilfe
 - eigener, schicht-interner Aktionen und
 - der von tieferen Schichten bereitgestellten Dienste.

Inhalt

1 Das Internet

- 1.1 Struktur des Internet: Wirte, Verbindungen, Vermittlungsstellen
- 1.2 Kernstruktur und Zugriffsnetze
- 1.3 Paketvermittlung
- 1.4 Leitungsvermittlung, Frequenz- und Zeit-Multiplexverfahren
- 1.5 Prinzipien der Paketvermittlung, Speichern und Weiterleiten
- 1.6 Warteschlangen, Paketverlust und Verzögerungen

2 Kommunikation im Internet

- 2.1 Beispiel Webseitenaufruf
- 2.2 Schichtenmodell
- 2.3 Routing
- 2.4 Server und Datenfluss

3 Einige Protokolle

- 3.1 Anwendungsschicht: HTTP, SMTP, FTP
- 3.2 Transportschicht: Transmission Control Protocol (TCP)
- 3.3 Netzwerkschicht: Internet Protocol (IP)
- 3.4 Verbindungsschicht: z.B. Ethernet
- 3.5 Physische Schicht: z.B. Manchester-Kode

Steuerung der Übertragung: Protokolle

Menschliche Protokolle

Quelle: Wikipedia (modifiziert)

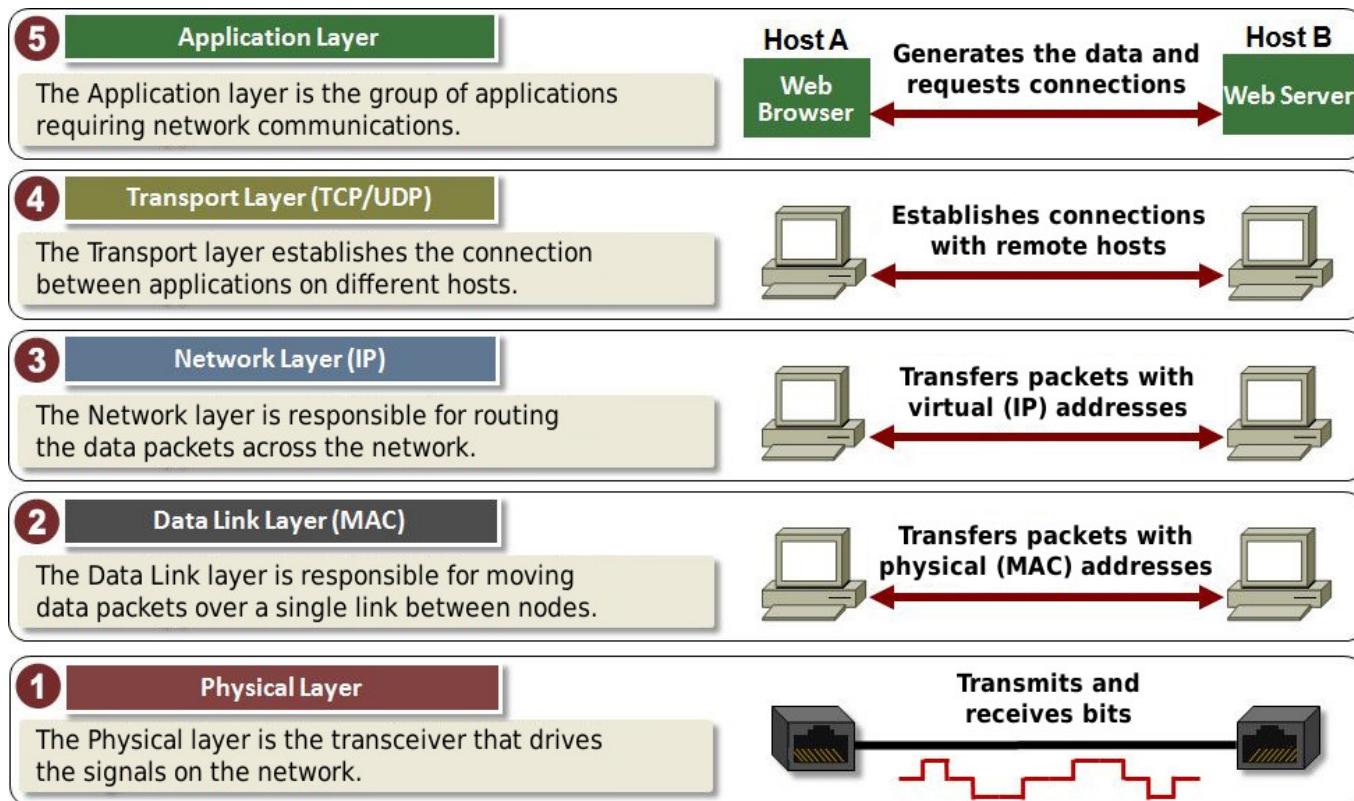
- Ein Protokoll zeichnet auf oder schreibt vor, zu welchem Zeitpunkt oder in welcher Reihenfolge welcher Vorgang durch wen oder durch was veranlasst wurde/wird.
- Das Protokollieren ermöglicht es, Vorgänge zu rekonstruieren oder zu planen, um Fehler bzw. Fehlfunktionen zu orten bzw. zu vermeiden.
- Hofprotokolle und diplomatische Protokolle legen z.B. Rangfolgen, Abläufe, Kleidervorschriften, Sitzordnungen und Verhalten fest.
- Auch viele Alltagssituationen folgen informellen Protokollen (z.B. Begrüßung).

Protokolle für Rechnernetze definieren

- das **Format**, den **Inhalt** und die **Reihenfolge** von Nachrichten, die von Netzwerknoten gesendet und empfangen werden und
- die **Aktionen**, die durchgeführt werden, wenn Nachrichten übertragen und entgegengenommen werden.

Kommunikation zwischen Rechnern

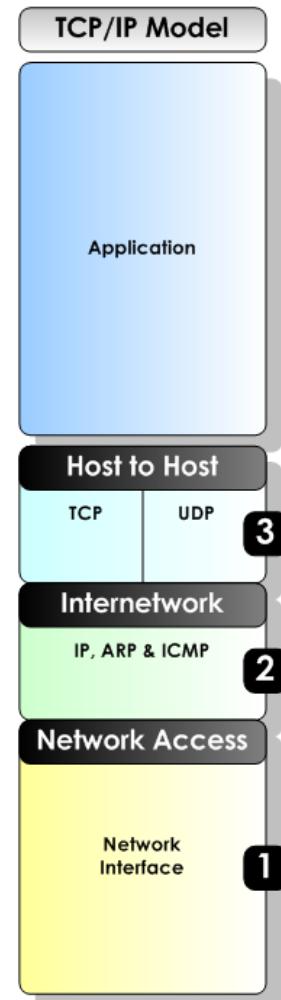
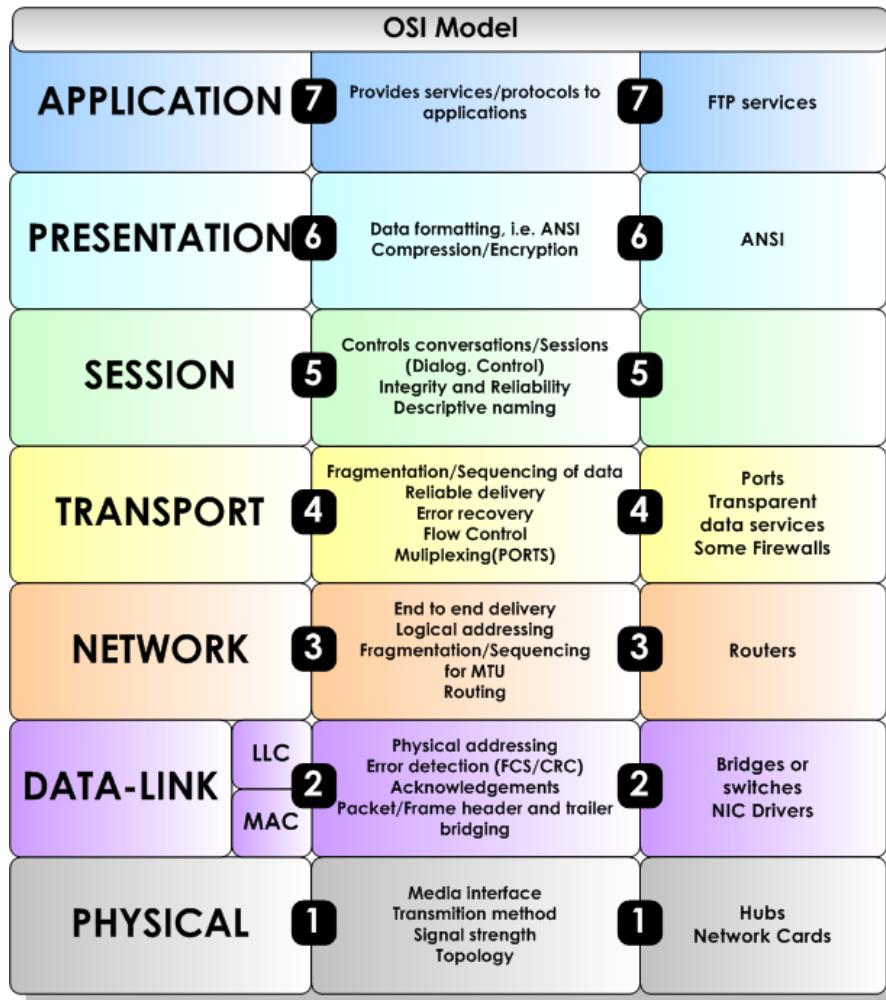
- Der Bereich der Rechnernetze beschäftigt sich mit der abstrakten Beschreibung der Kommunikation zwischen Rechnern (Endsystemen) durch Paketvermittlung.
- Die Kommunikation wird üblicherweise durch ein Schichtenmodell beschrieben, z.B. das TCP/IP-5-Schichten-Modell oder das ISO/OSI-7-Schichten-Modell.



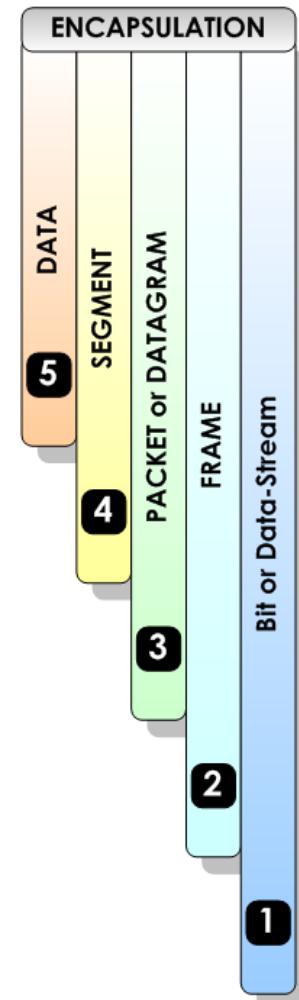
2.bp.blogspot.com (modifiziert)

Rechnernetze: ISO/OSI versus TCP/IP

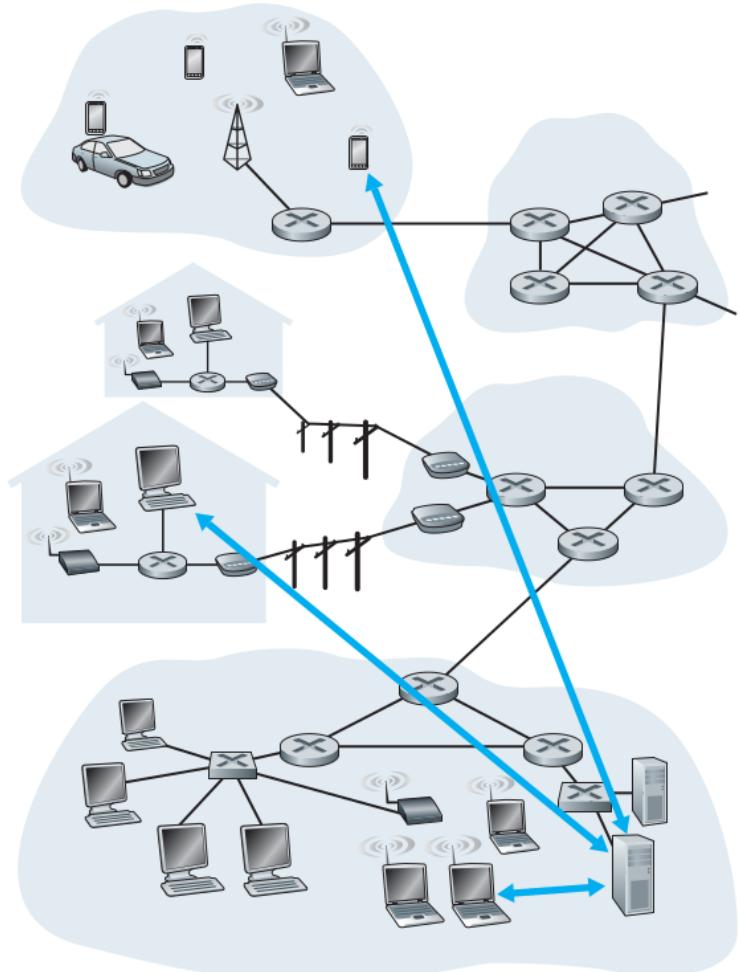
The OSI Model (Open Systems Interconnection)



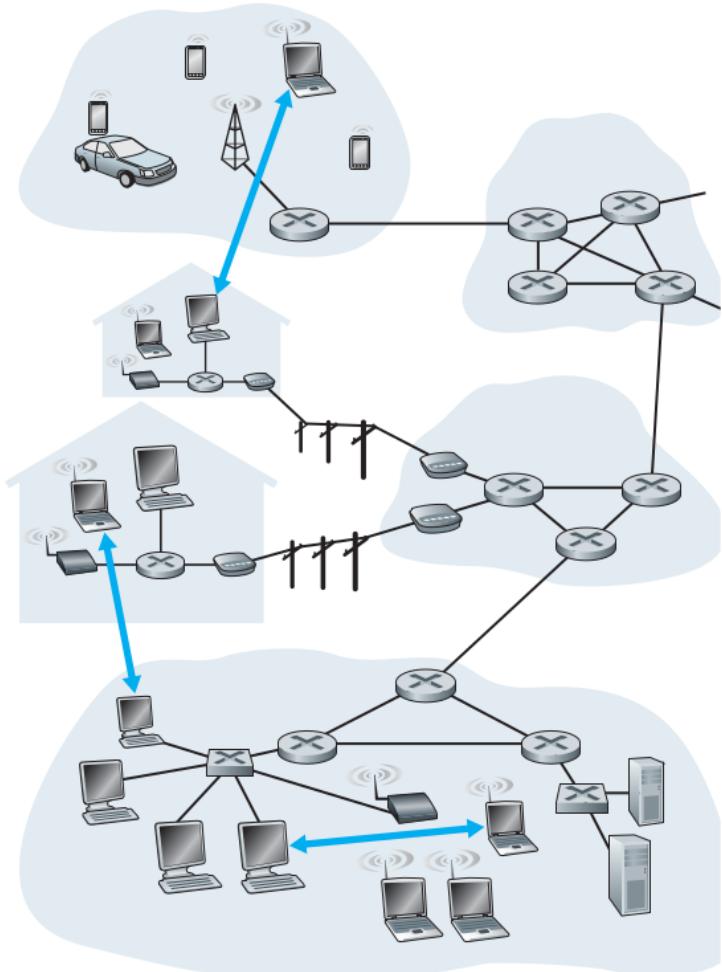
© Copyright 2008 Steven Iveson
www.networkstuff.eu



Anwendungsschicht: Client-Server und Peer-to-Peer



a. Client-server architecture



b. Peer-to-peer architecture

Anwendungsschicht: Client-Server und Peer-to-Peer

Client-Server-Architektur

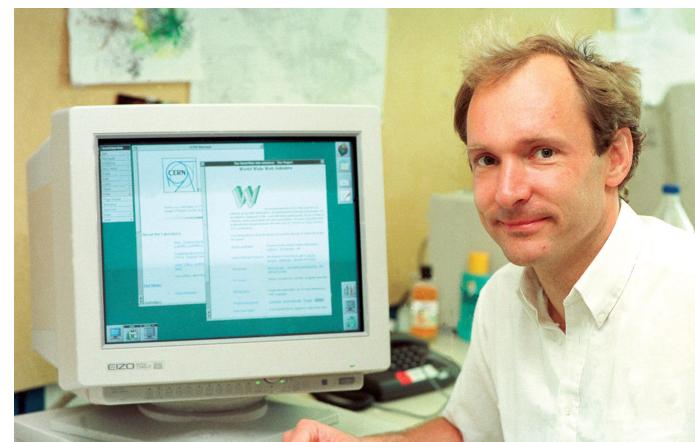
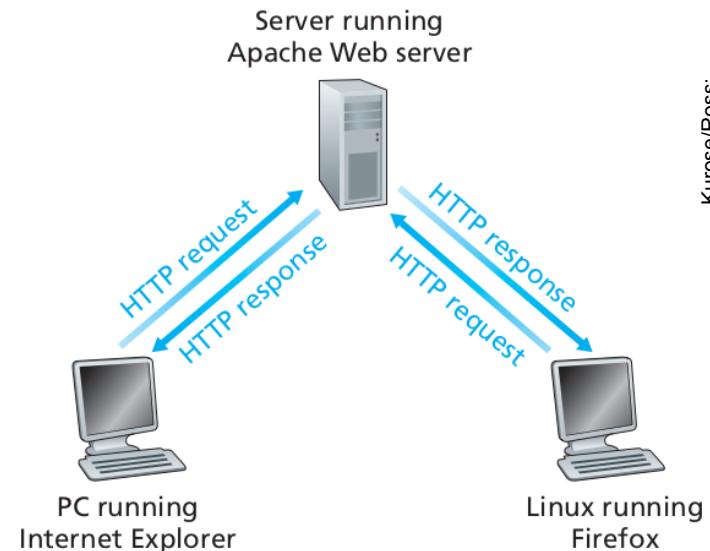
- Es gibt einen immer verfügbaren Netzbediener, den Server, der Anfragen von vielen anderen Wirten, den Clients, erhält.
- Typisches Beispiel: Webserver, der Anfragen nach Objekten, meist Webseiten, erhält und daraufhin das angefragte Objekt überträgt.
- Bei stark nachgefragten Bedienern (z.B. Google, Facebook etc.) werden mehrere Bediener in Datenzentren zu einem virtuellen Bediener zusammengeschaltet.

Peer-to-Peer-Architektur (P2P)

- Die Anwendung nutzt direkte Kommunikation zwischen Wirten, die temporär verbunden, ggf. auch nur temporär verfügbar sind.
- Typisches Beispiel: Das Dateiteilungsprogramm (file sharing program) BitTorrent.
- Angenehme Eigenschaft: Selbstskalierbarkeit, da jeder Empfänger sofort auch wieder als Sender auftreten kann.

Anwendungsschicht: HyperText Transfer Protocol (HTTP)

- Das **HyperText Transfer Protocol (HTTP)** ist das Herzstück des sogenannten **World Wide Web (WWW)**, ein über das Internet abrufbares System von elektronischen Hypertext-Dokumenten (Webseiten).
- Es ist in zwei Programmen implementiert: einem Server-Programm und einem Client-Programm.
- Diese beiden Programme, die auf verschiedenen Witten laufen, kommunizieren durch den Austausch von HTTP-Nachrichten.
- HTTP definiert die Struktur dieser Nachrichten und wie Client- und Server-Programm diese Nachrichten austauschen.
- HTTP wurde von Tim Berners-Lee definiert und in einer ersten Version implementiert.

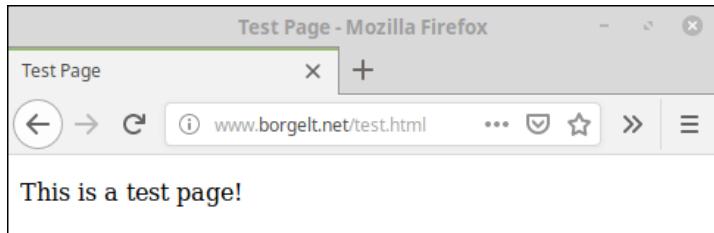


Anwendungsschicht: HyperText Transfer Protocol (HTTP)

- Das HyperText Transfer Protocol (HTTP) dient i.w. zum Übertragen von Webseiten in der HyperText Markup Language (HTML).
- Es spezifiziert z.B. ein Anfrage- (request) und ein Antwort- (response) -format für Daten.
- Die Spezifikationen dieser Formate sind rechts beispielhaft gezeigt, mit markierten Abschnitten und Beispielinhalten.

GET /index.html HTTP/1.1	Request Line	HTTP Request	
Date: Thu, 20 May 2004 21:12:55 GMT	General Headers		
Connection: close			
Host: www.myfavoriteamazingsite.com	Request Headers		
From: joebloe@somewebsitesomewhere.com			
Accept: text/html, text/plain			
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)	Entity Headers		
www.tcpipguide.com			
HTTP/1.1 200 OK	Status Line	HTTP Response	
Date: Thu, 20 May 2004 21:12:58 GMT	General Headers		
Connection: close			
Server: Apache/1.3.27	Response Headers		
Accept-Ranges: bytes			
Content-Type: text/html	Entity Headers		
Content-Length: 170			
Last-Modified: Tue, 18 May 2004 10:14:49 GMT			
<html>	Message Body		
<head>			
<title>Welcome to the Amazing Site!</title>			
</head>			
<body>			
<p>This site is under construction. Please come back later. Sorry!</p>			
</body>			
</html>			

Anwendungsschicht: HyperText Transfer Protocol (HTTP)



- Um zu sehen, was hinter den Kulissen eines Stöberers (browsers) abläuft, kann man z.B. das Programm telnet verwenden.
- Mit diesem Programm kann man z.B. eine HTTP-Anfrage stellen (Port 80) und beobachten, welche Antwort gesendet wird.
- Die rechts in rot gezeigten Eingaben rufen z.B. die oben gezeigte sehr einfache Webseite ab.

```
> telnet www.borgelt.net 80
Trying 134.0.30.202...
Connected to www.borgelt.net.
Escape character is ']'.
GET /test.html HTTP/1.1
Host: www.borgelt.net
Accept: text/html

HTTP/1.1 200 OK
Date: Wed, 09 Jan 2019 18:15:24
GMT
Server: Apache/2.4.26 (Unix)
OpenSSL/1.0.1t mod_fcgid/2.3.9
Vary: User-Agent
Last-Modified: Wed, 09 Jan 2019
18:13:31 GMT
ETag: "d9-57f0a6b66b70b"
Accept-Ranges: bytes
Content-Length: 217
Content-Type: text/html

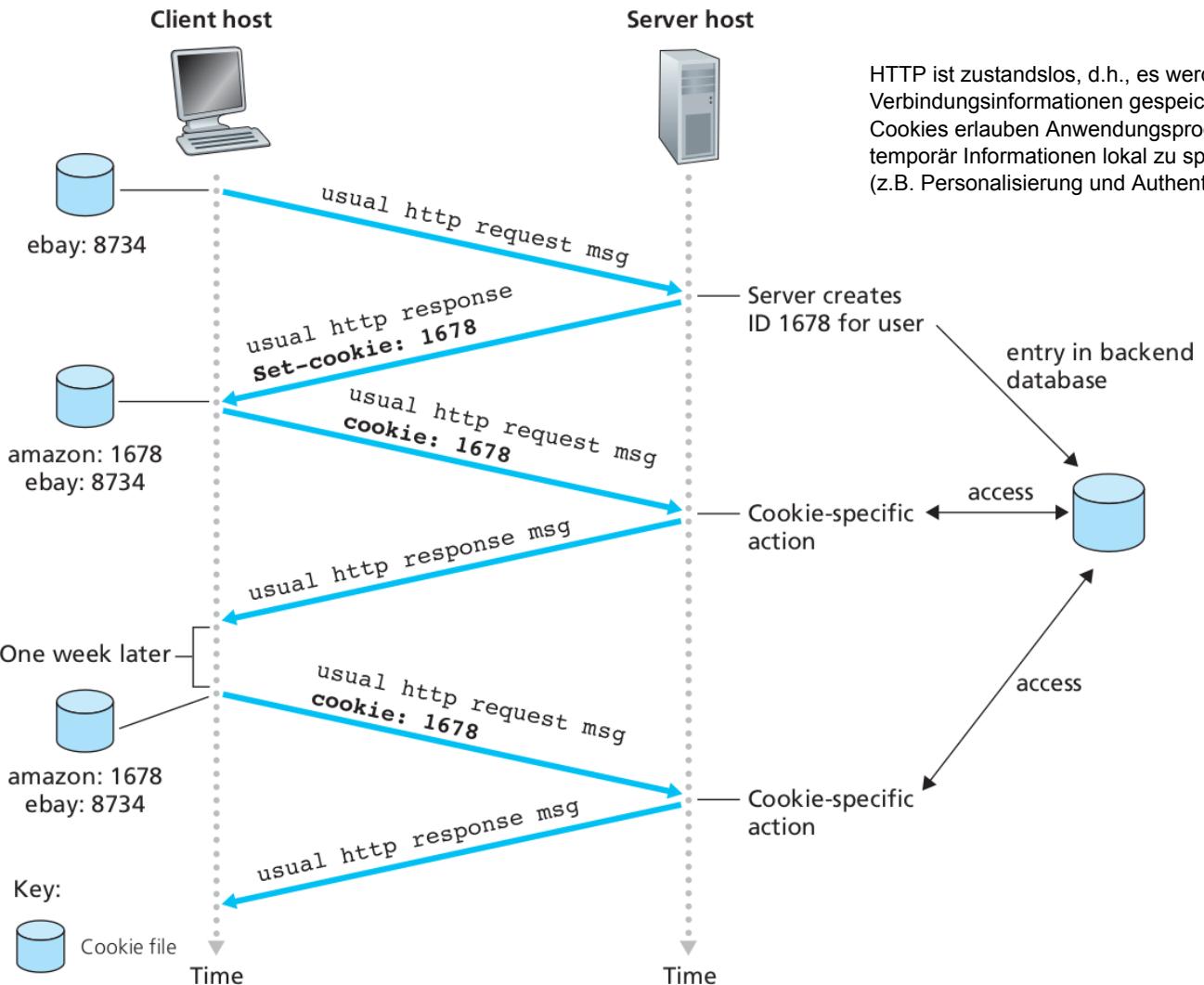
<!DOCTYPE HTML PUBLIC "-//W3C//DTD
HTML 4.01//EN">
<html>
<head>
<meta http-equiv="content-type"
content="text/html;
charset=utf-8">
<title>Test Page</title>
</head>
<body>
<p>This is a test page!</p>
</body>
</html>
Connection closed by remote host.
```

Anwendungsschicht: HyperText Transfer Protocol (HTTP)

Von einem Server wird auf jede HTTP-Anfrage ein **HTTP-Statuscode** als Antwort geliefert, der anzeigen, ob die Anfrage erfolgreich bearbeitet wurde.

- 200 OK
Die Anfrage wurde erfolgreich bearbeitet und das Ergebnis der Anfrage wird in der Antwort übertragen.
- 301 Moved Permanently
Die angeforderte Ressource steht unter einer neuen Adresse bereit (Location-Feld). Die alte Adresse ist nicht länger gültig.
- 400 Bad Request
Die Anfrage-Nachricht war fehlerhaft aufgebaut.
- 404 Not Found
Die angeforderte Ressource wurde nicht gefunden. Dieser Statuscode kann ebenfalls verwendet werden, um eine Anfrage ohne näheren Grund abzuweisen.
- 505 HTTP Version not supported
Die benutzte HTTP-Version (gemeint ist die Zahl nach dem Schrägstrich) wird vom Server nicht unterstützt oder abgelehnt.

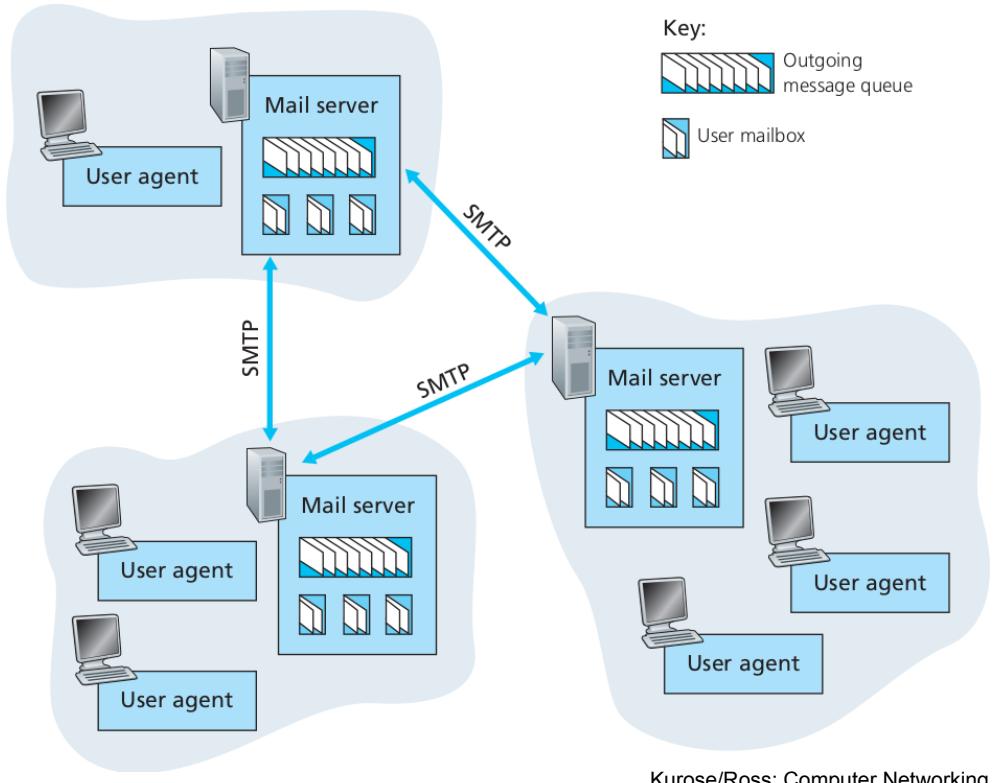
Anwendungsschicht: HyperText Transfer Protocol (HTTP)



HTTP ist zustandslos, d.h., es werden keine Verbindungsinformationen gespeichert. Cookies erlauben Anwendungsprogrammen temporär Informationen lokal zu speichern (z.B. Personalisierung und Authentifizierung).

Anwendungsschicht: Simple Mail Transfer Protocol (SMTP)

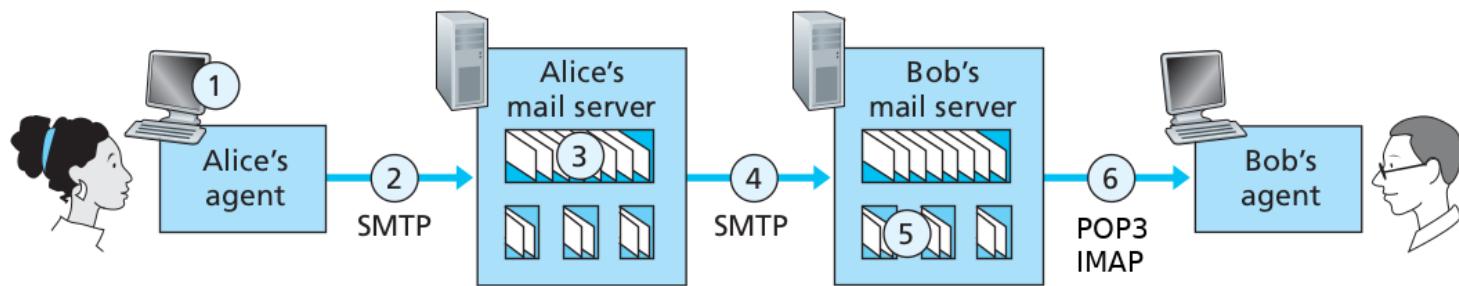
- Das **Simple Mail Transfer Protocol (SMTP)** dient der Zustellung von elektronischer Post.
- SMTP überträgt Nachrichten vom Email-Server des Senders zum Email-Server des Empfängers.
- SMTP ist viel älter als HTTP und leidet daher an einigen Einschränkungen (z.B. nur 7-Bit ASCII).



Kurose/Ross: Computer Networking

ASCII: American Standard Code for Information Interchange

Anwendungsschicht: Simple Mail Transfer Protocol (SMTP)



Kurose/Ross:
Computer Networking

- Alice startet ihr Email-Programm, gibt Bobs Email-Adresse an, schreibt den Nachrichtentext und weist das Email-Programm an, die Nachricht auszuliefern.
- Alices Email-Programm sendet die Nachricht an ihren Email-Server, wo sie in einer Nachrichten-Warteschlange abgelegt wird.
- Das Client-Programm auf Alices Email-Server sieht die Nachricht in der Warteschlange und öffnet eine Verbindung zum Server-Programm auf Bobs Email-Server.
- Das SMTP-Client-Programm sendet Alices Nachricht über die Verbindung.
- Die Nachricht kommt auf Bobs Email-Server an und das Server-Programm legt die Nachricht in Bobs elektronischen Briefkasten ab.
- Bob startet sein Email-Programm, um die Nachricht zu lesen.

Anwendungsschicht: Simple Mail Transfer Protocol (SMTP)

- Auch das Simple Mail Transfer Protocol lässt sich „roh“ über das Programm telnet ansprechen.
- Nach dem Aufbau der Verbindung werden eine Reihe von Befehlen abgesetzt, mit der man sich beim Mailserver anmeldet, Sender und Empfänger angibt, und die Email verfaßt.
- Im Gegensatz zu dem vorangehenden Beispiel (HTTP) und dem folgenden Beispiel (FTP) können die rechts stehenden Befehle leider nicht so ausgeführt werden, sondern zeigen lediglich die grundsätzliche Struktur.

```
> telnet mail.example.com 25
Trying xxx.xxx.xxx.xxx...
Connected to mail.example.com.
Escape character is '^]'.
220 service ready
HELO name.example.net
250 OK
MAIL FROM:<sender@example.org>
250 OK
RCPT TO:<receiver@example.com>
250 OK
DATA
354 start mail input
From: <sender@example.org>
To: <receiver@example.com>
Subject: Testmail
Date: Wed, 9 Jan 2019 18:20:43 GMT

Lorem ipsum dolor sit amet, consetetur sadipscing
elitr, sed diam nonumy eirmod tempor invidunt ut
labore et dolore magna aliquyam erat, sed diam
voluptua. At vero eos et accusam et justo duo
dolores et ea rebum. Stet clita kasd gubergren,
no sea takimata sanctus est.
.
=> Dieser alleinstehende Punkt beendet die Nachricht.
250 OK
QUIT
221 closing channel
Connection closed by remote host.
```

Anwendungsschicht: POP3 und IMAP

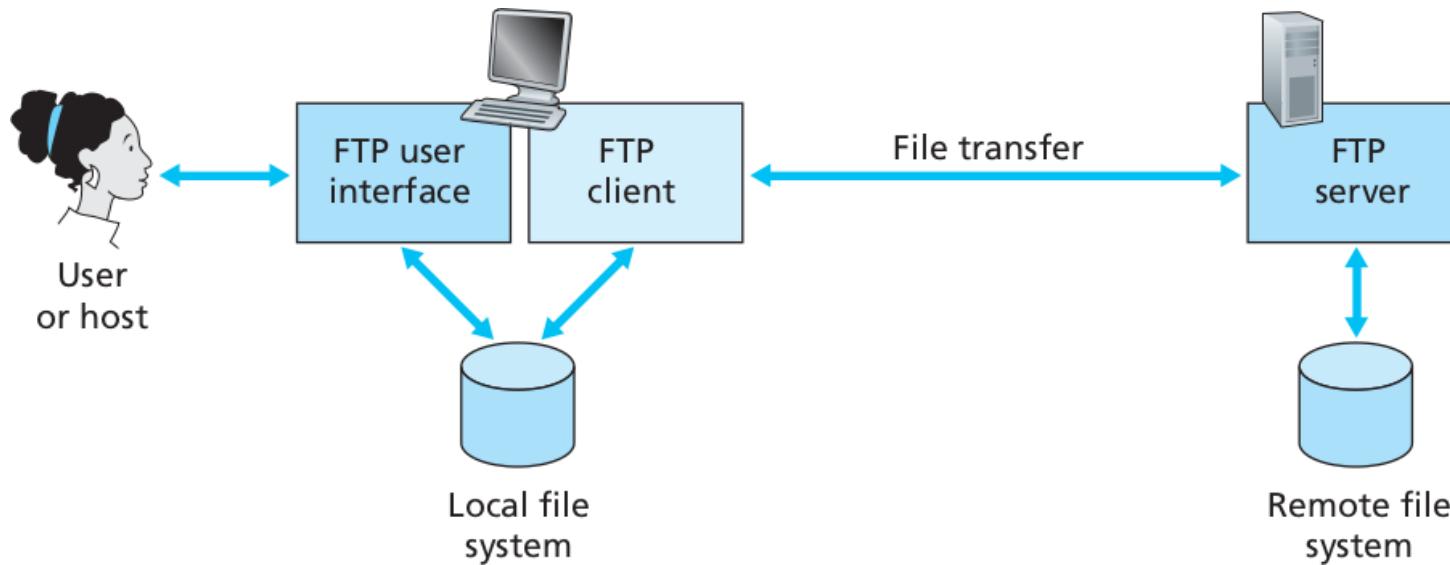
Post Office Protokoll, Version 3 (POP3)

- Ähnlich wie SMTP ist POP3 ein ASCII-Protokoll, bei dem die Steuerung der Datenübertragung durch Kommandos geschieht.
- Der POP3 standardmäßig zugeordnete Port ist 110 (SMTP: 25).
- POP3 ist sehr eingeschränkt und erlaubt nur das Auflisten, Abholen und Löschen von Nachrichten am Email-Server.

Internet Message Access Protocol (IMAP)

- Netzwerkprotokoll, das ein Netzwerkdateisystem für Emails bereitstellt.
- Erweitert die Funktionen von POP3 so, daß Benutzer Emails, Ordnerstrukturen und Einstellungen auf den Email-Servern belassen können.
- Das Simple Mail Access Protocol (SMAP) ist ein Ansatz, die Funktionalität von IMAP und SMTP zu vereinen.

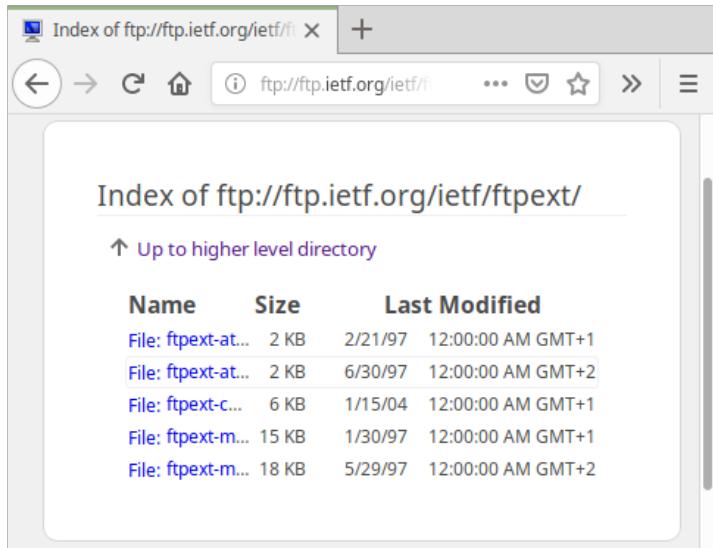
Anwendungsschicht: File Transfer Protocol (FTP)



Kurose/Ross:
Computer Networking

- Das **File Transfer Protocol (FTP)** dient zum Übertragen von Dateien von einem Wirt auf einen anderen.
- Ein Benutzer auf einer Rechner startet das Client-Programm und meldet sich darüber bei einem Server-Programm an, das auf einem anderen Wirt läuft.
- Der Benutzer muß sich durch einen Benutzernamen und ein Paßwort authentifizieren (eventuell möglich: anonyme Zugriff, anonymous) und kann anschließend eine oder mehrere Dateien übertragen.

Anwendungsschicht: File Transfer Protocol (FTP)



```
> telnet ftp.ietf.org 21
Trying 4.31.198.44...
Connected to ietf.org.
220 FTP server ready
USER anonymous
331 Anonymous login ok,
send your complete email address as your password
PASS user@example.com
230 Anonymous access granted, restrictions apply
CWD ietf/ftpext/
250 CWD command successful
PASV
227 Entering Passive Mode (4,31,198,44,242,84).
NLST
150 Opening ASCII mode data connection for file list
226 Transfer complete
QUIT
221 Goodbye.
Connection closed by foreign host.
```

Steuerverbindung

```
> telnet 4.31.198.44 63876
Trying 4.31.198.44...
Connected to 4.31.198.44.
ftpext-minutes-97apr.txt
ftpext-attendees-97apr.txt
ftpext-charter.txt
ftpext-attendees-96dec.txt
ftpext-minutes-96dec.txt
Connection closed by remote host.
```

Datenverbindung
(63876 = 242 · 256 + 84)

- Was bei einer FTP-Verbindung z.B. in einem Stöberer passiert, kann man sich auch mit dem Programm telnet anschauen.
- Mit den rechts rot gezeigten Anweisungen kann man ein Verzeichnis der Internet Engineering Task Force (IETF) auslesen.

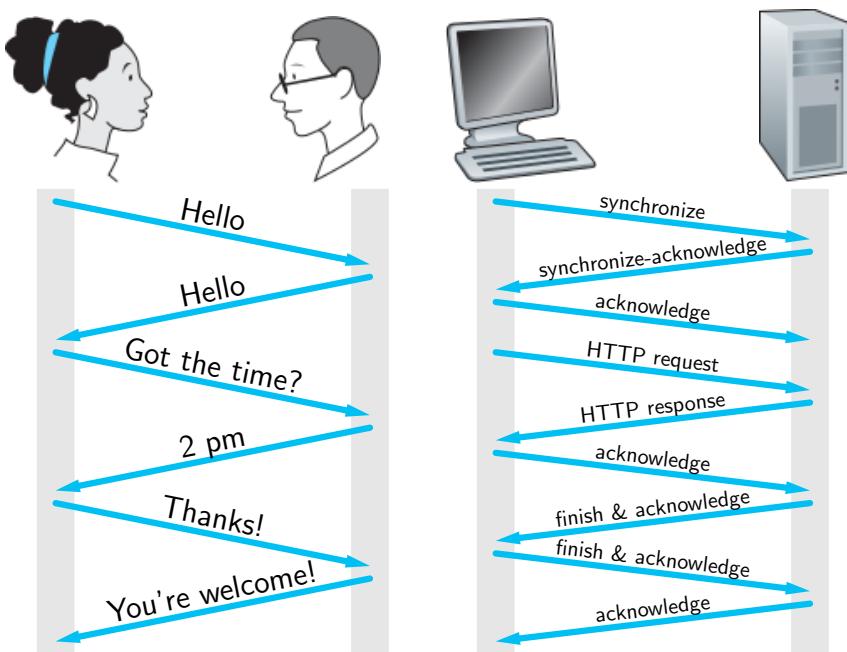
Transportschicht: Transmission Control Protocol (TCP)

Die Hauptfunktionen der Transportschicht sind:

- **Logische Kommunikation** (logical communication):
Für Anwendungen soll es so aussehen, als ob die kommunizierenden Wirte direkt verbunden wären. Details der physischen Infrastruktur werden verborgen.
- **Verlässliche Zustellung** (reliable delivery):
Einige Protokolle der Transportschicht garantieren eine verlässliche Zustellung; z.B. Erkennen von Paketverlusten und Veranlassung einer erneuten Übertragung.
- **Integritätsprüfung** (integrity check):
Durch den Datenpaketen hinzugefügte Prüfsummen werden Bitfehler erkannt und können u.U. sogar korrigiert werden.
- **Blockierungskontrolle** (congestion control):
Verhindert, daß eine Verbindung die Leitungen und Vermittlungsstellen zwischen kommunizierenden Wirten durch übermäßige Datenübertragung überfordert.
- **Multiplex und Demultiplex** (multiplexing and demultiplexing):
Erweiterung der Kommunikation von Wirten auf Prozesse (über sog. sockets).

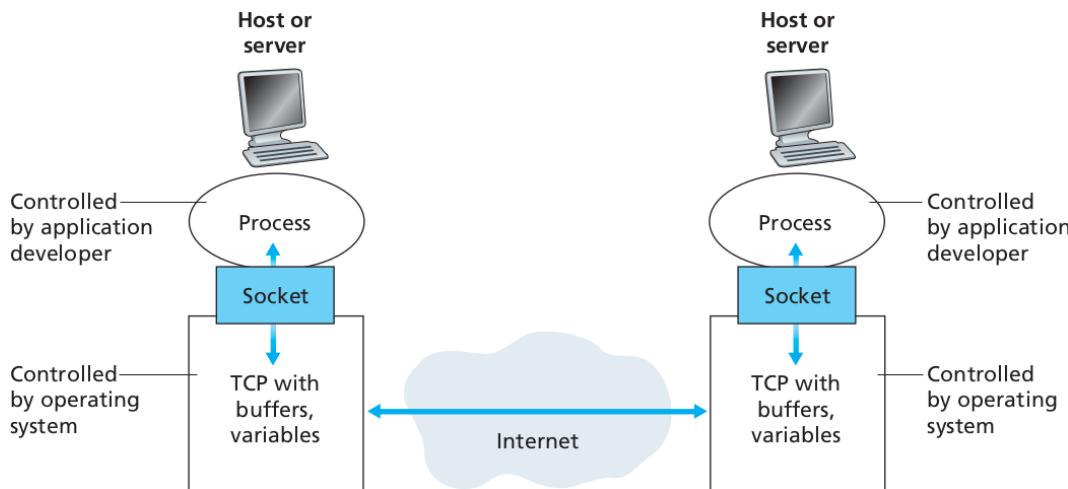
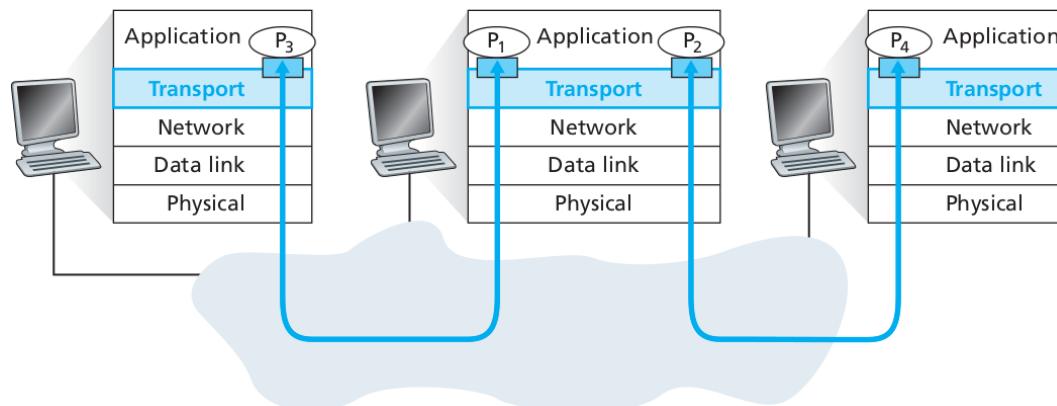
Transportschicht: Transmission Control Protocol (TCP)

Kurose/Ross: Computer Networking (modifiziert)



- Die Kommunikation zwischen Rechnern ist im Prinzip der Kommunikation zwischen Menschen nachempfunden.
 - Sie umfaßt die Schritte
 - **Verbindungsaufbau**,
 - **Datenübertragung**,
 - **Verbindungsabbruch**.
 - Für die Verlässlichkeit: mehr Bestätigungen.
-
- Die Protokolle der Transportschicht sind in den Wirten (hosts) und Endsystemen (end systems) implementiert, aber nicht in den Vermittlungsstellen (routers).
 - Die Nachrichten der Anwendungsschicht werden in **Segmente** zerlegt, z.B. TCP-Segmente (transmission control protocol segments).

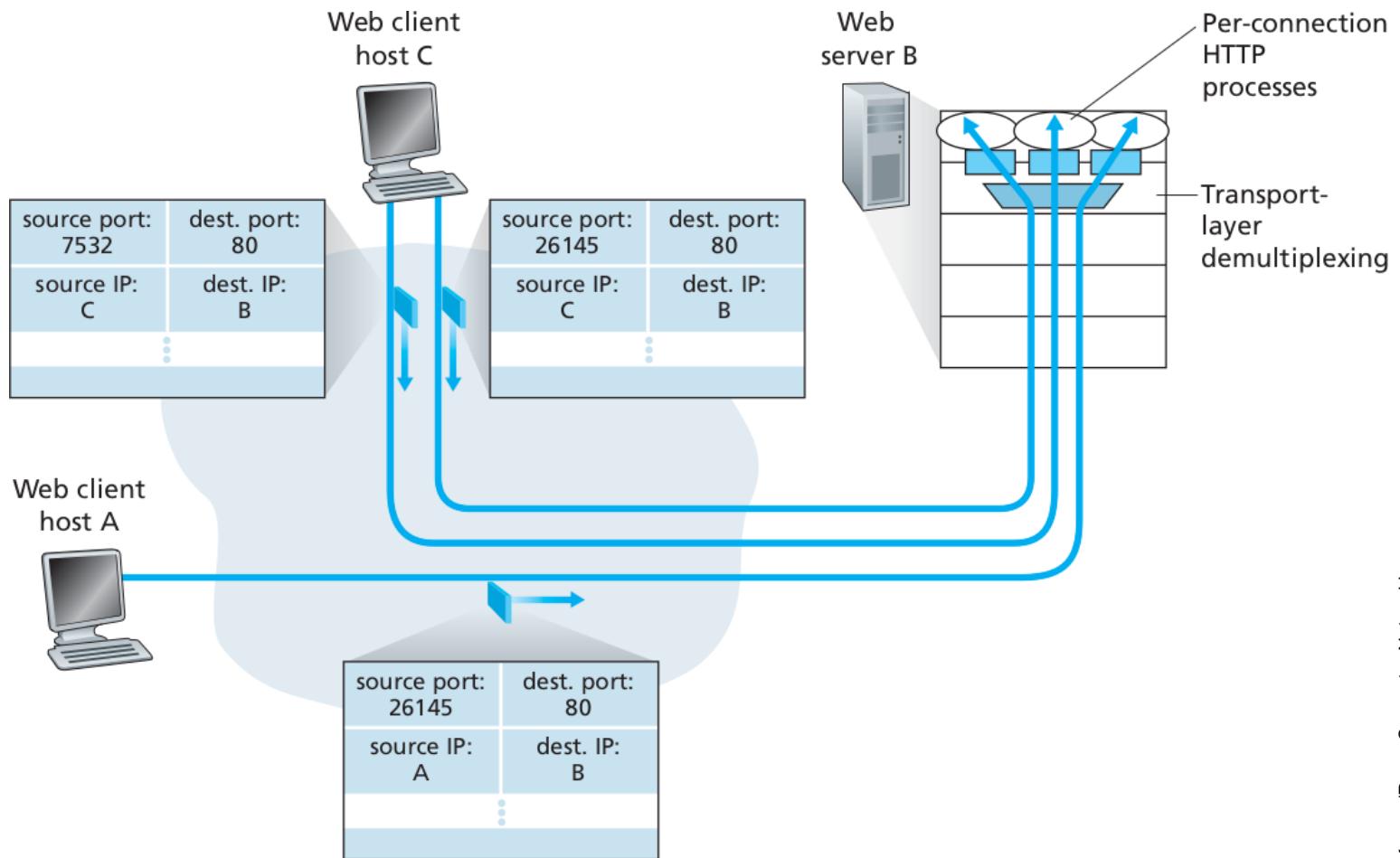
Transportschicht: Transmission Control Protocol (TCP)



Kurose/Ross: Computer Networking

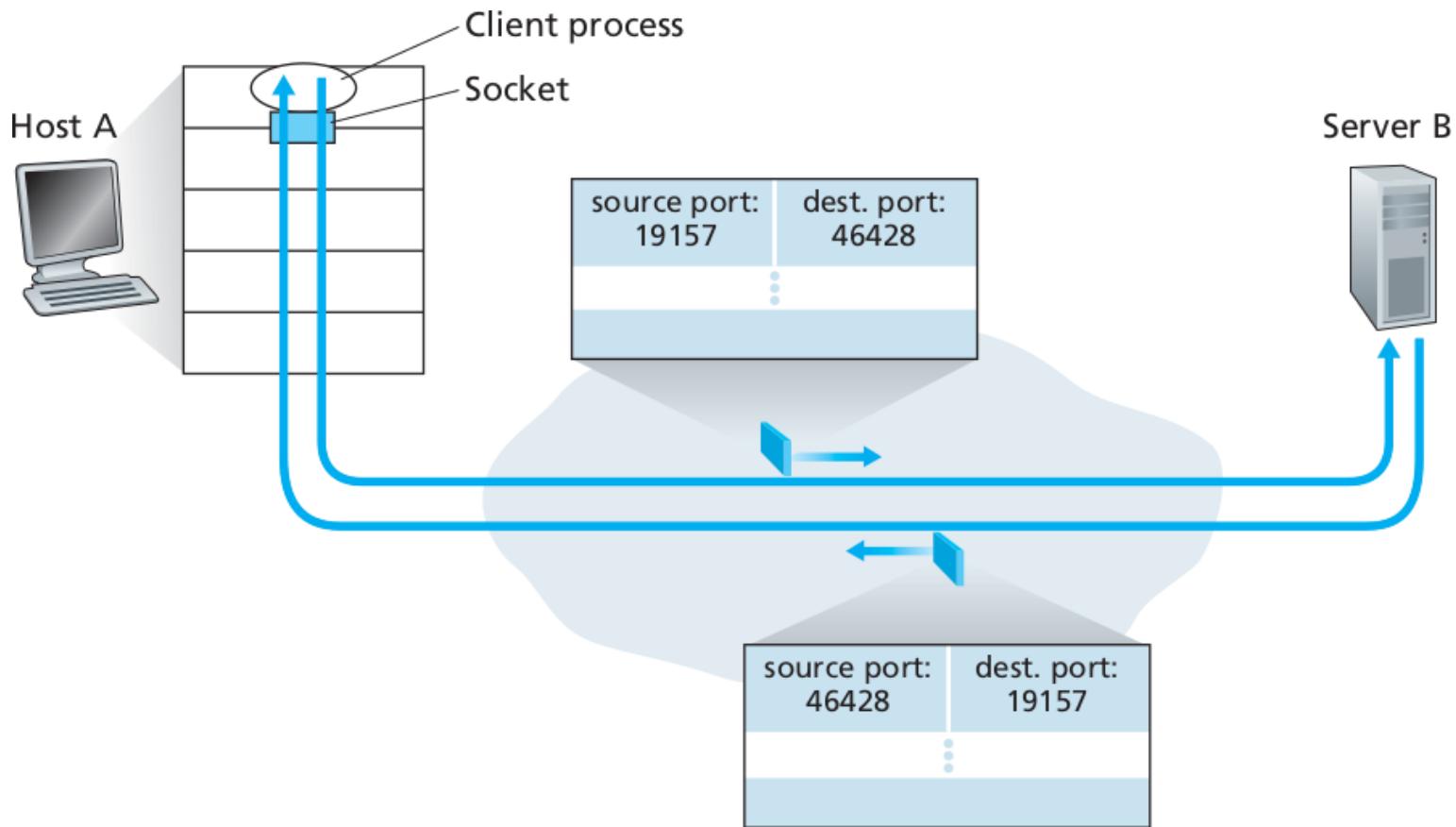
- Eigentlich gewünscht: Kommunikation zwischen Prozessen statt Wirten (Anwendungsprogramme kommunizieren).
- Dies wird erzielt durch **Sockel** (sockets), durch die ein Prozeß mit der Transportschicht kommuniziert.
- Im Netzwerk werden Sockel über **Ports** identifiziert.
- TCP-Segmente enthalten daher Quell- und Zielpunkt.

Transportschicht: Transmission Control Protocol (TCP)



- Mehrere Verbindungen zum gleichen Port, speziell bei Webservern, werden durch Transportschicht-Multiplexverfahren behandelt.

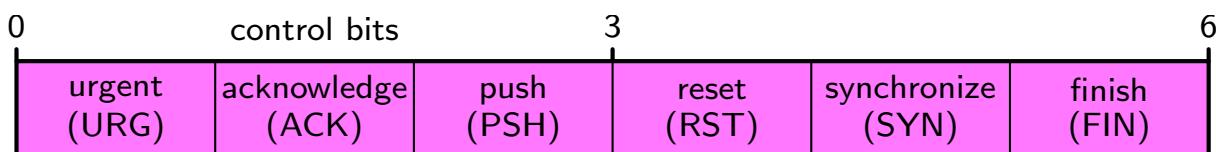
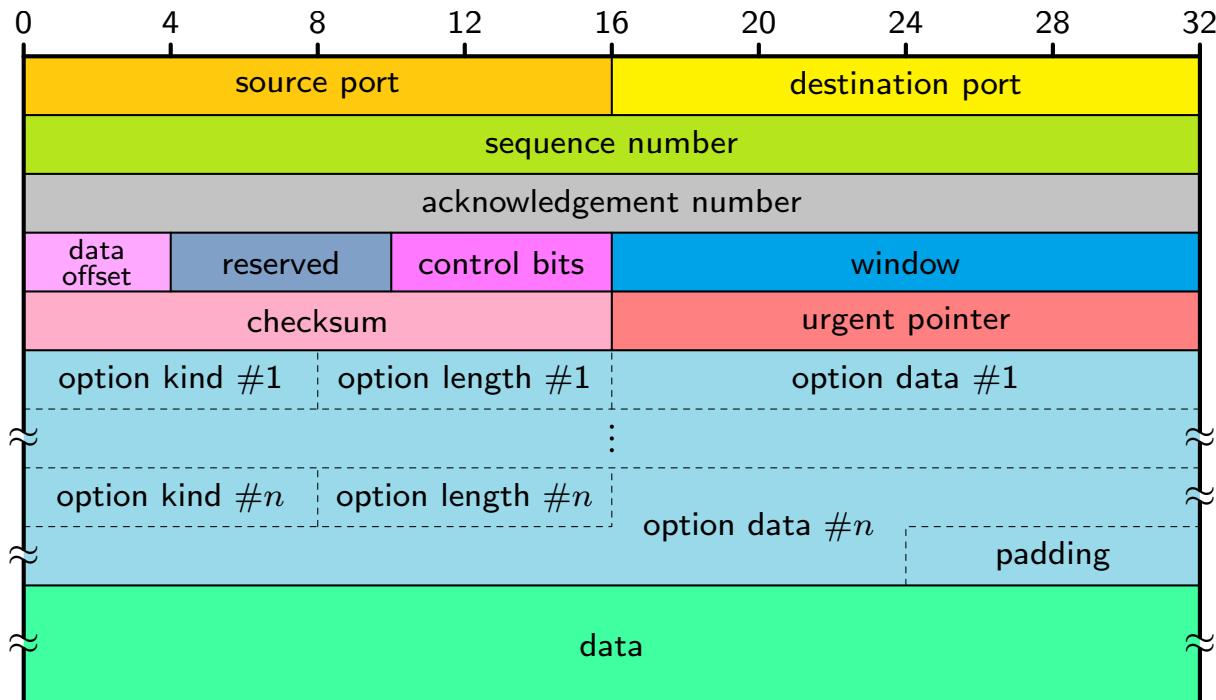
Transportschicht: Transmission Control Protocol (TCP)



Kurose/Ross: Computer Networking

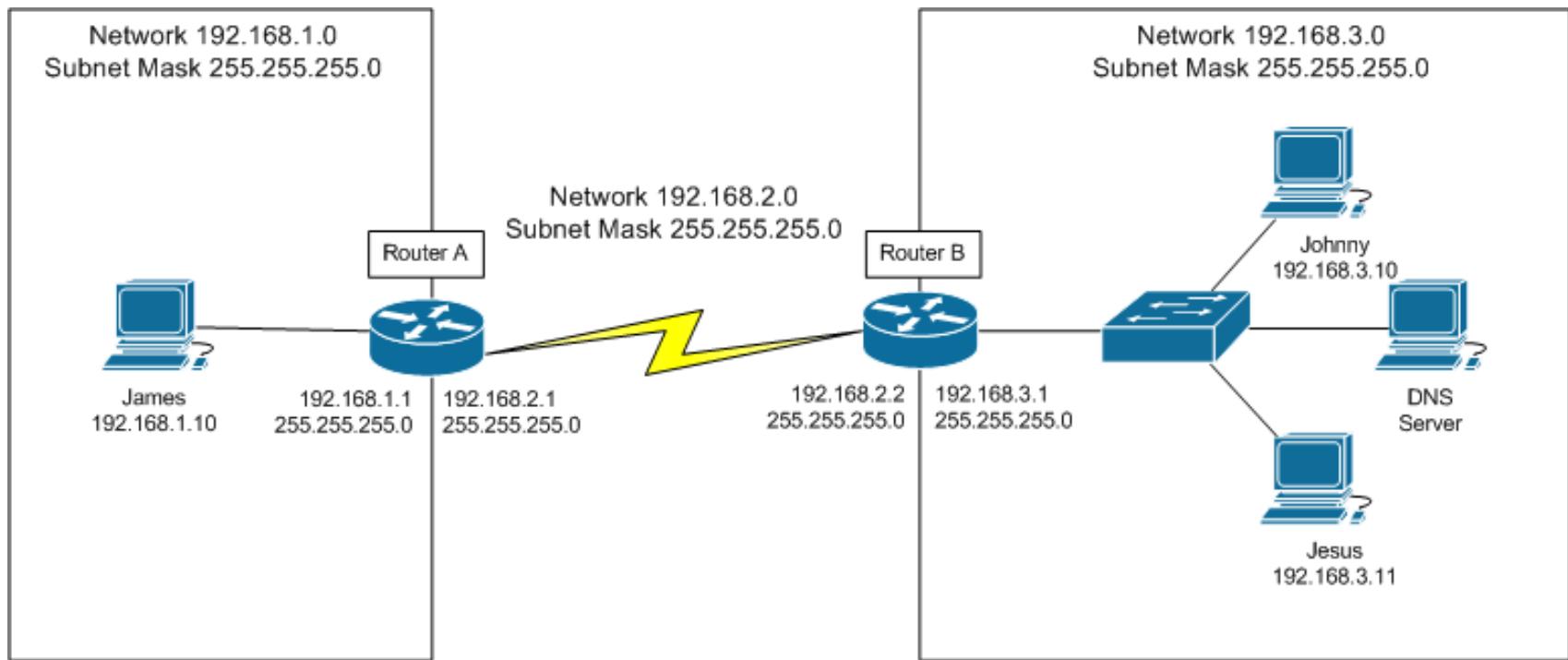
- Bei Antworten auf Anfragen werden Quell- und Zielpoert vertauscht.
- Außerdem müssen die auf dem Hinweg vorgenommenen Übersetzungen rückwärts ausgeführt werden (network address translation, NAT) \Rightarrow Netzwerkschicht.

Transportschicht: Transmission Control Protocol (TCP)



- Struktur eines **TCP-Segments** (Datenpaket der Transportschicht)
- Die Zahlen über dem Diagramm bezeichnen Bits.
- Quell- & Zielport identifizieren Sockel und dienen der Zustellung an einen Prozeß.
- Die control bits bestehen aus mehreren Feldern.

Netzwerkschicht: Internet Protocol (IP)



- Ein **IP-Netzwerk** kann in logische Teilnetze unterteilt werden.
- Die **Teilnetzmaske** (subnet mask) wird benutzt, um festzustellen, ob die Zieladresse im gleichen Teilnetzwerk ist.
- Kein Unterschied der Adressen unter der Maske \Rightarrow gleiches Teilnetz.

Netzwerkschicht: Internet Protocol (IP)

Die beiden Hauptfunktionen der Netzwerkschicht sind:

- **Weiterleiten** (forwarding):

Wenn ein Datenpaket an einer Vermittlungsstelle (router) ankommt, muß die Vermittlungsstelle es an die passende Ausgangsleitung (und damit an die nächste Vermittlungsstelle oder, falls möglich, direkt an den Empfänger) weiterleiten.

Das Weiterleiten ist ein Prozeß, der lokal in einzelnen Routern abläuft.

- **Wegeplanung** (routing):

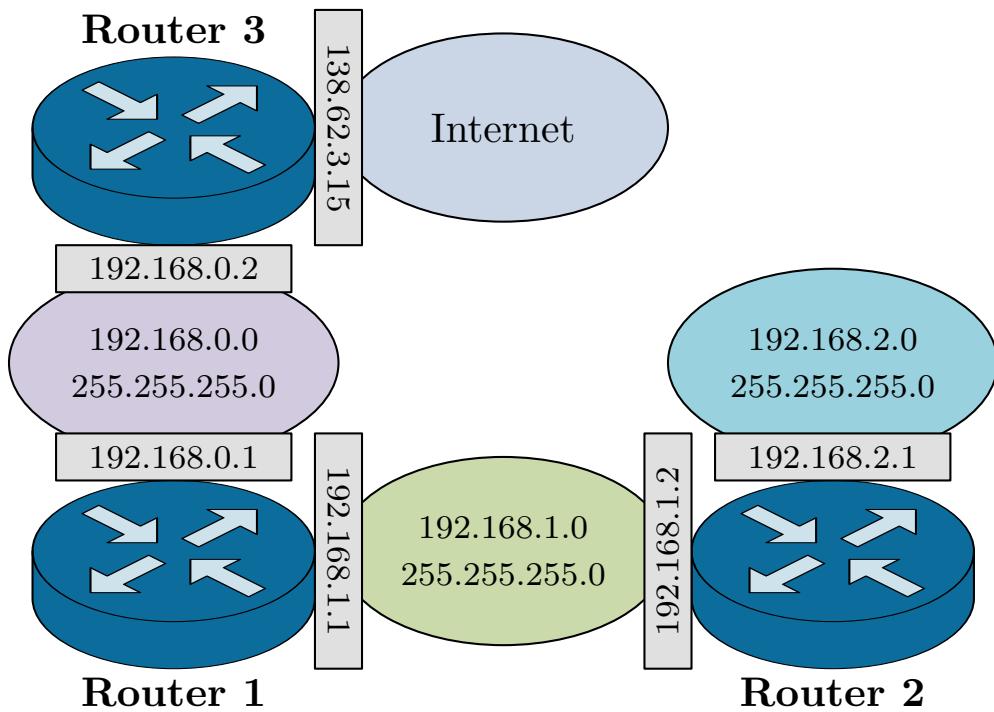
Die Netzwerkschicht bestimmt den Weg oder Pfad, den Pakete durch das Netzwerk nehmen, während sie vom Sender zum Empfänger übertragen werden.

Die Wegeplanung ist ein netzwerkweiter Prozeß, wenn auch lokal gesteuert.

Um diese Funktionen zu implementieren, verwalten die Vermittlungsstellen (router) **Weiterleitungstabellen** (forwarding tables).

destination	subnet mask	gateway /router	interface	metric (hops)
192.168.0.0	255.255.255.0	192.168.1.1	192.168.1.2	2
192.168.1.0	255.255.255.0	192.168.1.2	192.168.1.2	1
192.168.2.0	255.255.255.0	192.168.2.1	192.168.2.1	1
0.0.0.0	0.0.0.0	192.168.1.1	192.168.1.2	3

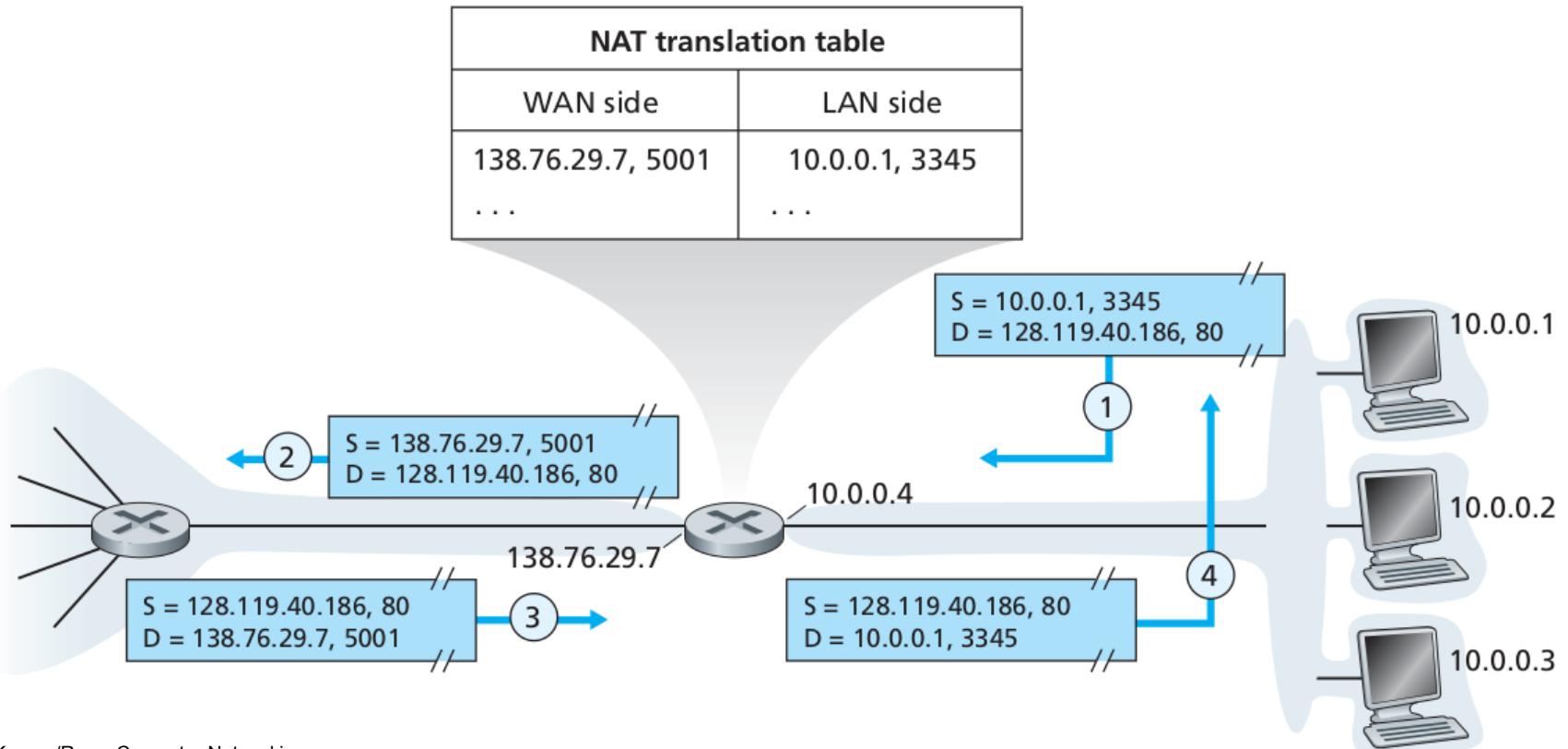
Netzwerkschicht: Internet Protocol (IP)



Forwarding table of router 2

- **Netzadresse (destination) & Teilnetzmaske (subnet mask)** Bestimmen zusammen das IP-Adressmuster eines Zielnetzes.
- **Netzübergang (gateway, router, next hop)** IP-Adresse für Weiterleitung; meist IP-Adresse einer Nachbar-Vermittlungsstelle.
- **Schnittstelle (interface)** Die Schnittstelle des Routers für die Weiterleitung.
- **Metrik (metric)** Information über die Gewichtung/Präferenz der Nutzung dieser Weginformation.

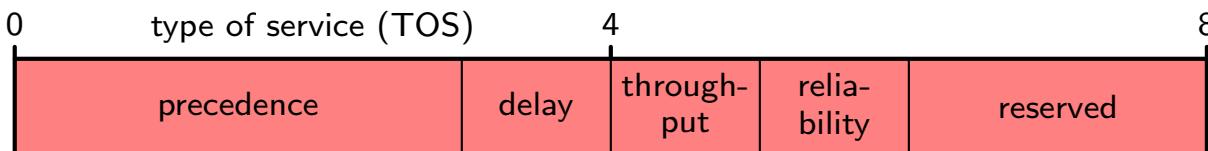
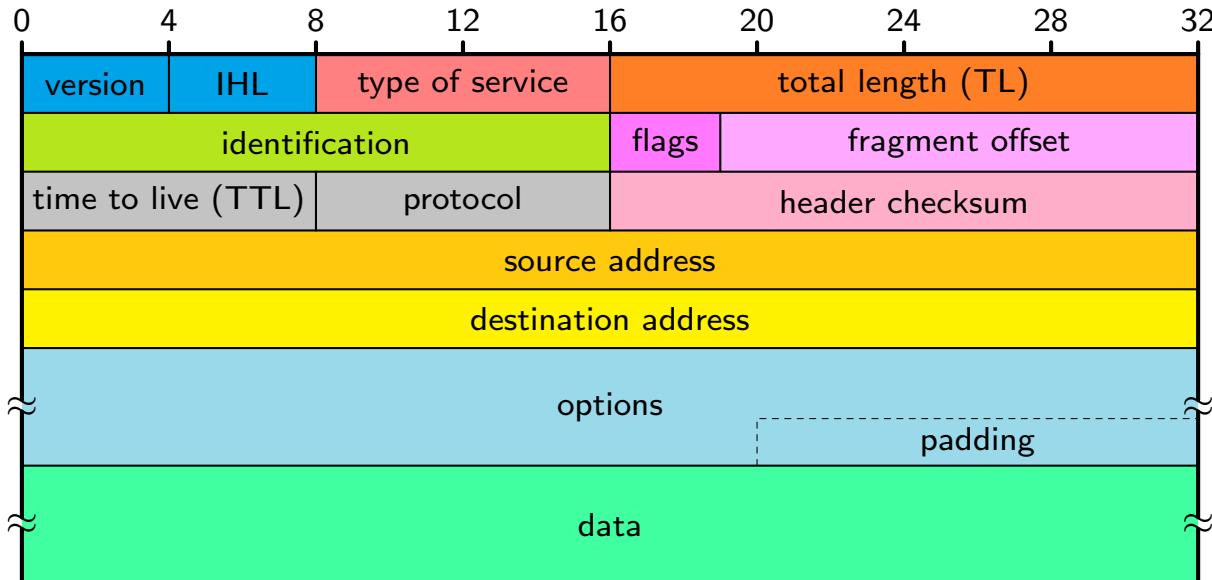
Netzwerkschicht: Internet Protocol (IP)



Kurose/Ross: Computer Networking

- Bei Antworten auf Anfragen werden Quell- und Zielport vertauscht.
- Außerdem müssen die auf dem Hinweg vorgenommenen Übersetzungen rückwärts ausgeführt werden (network address translation, NAT).

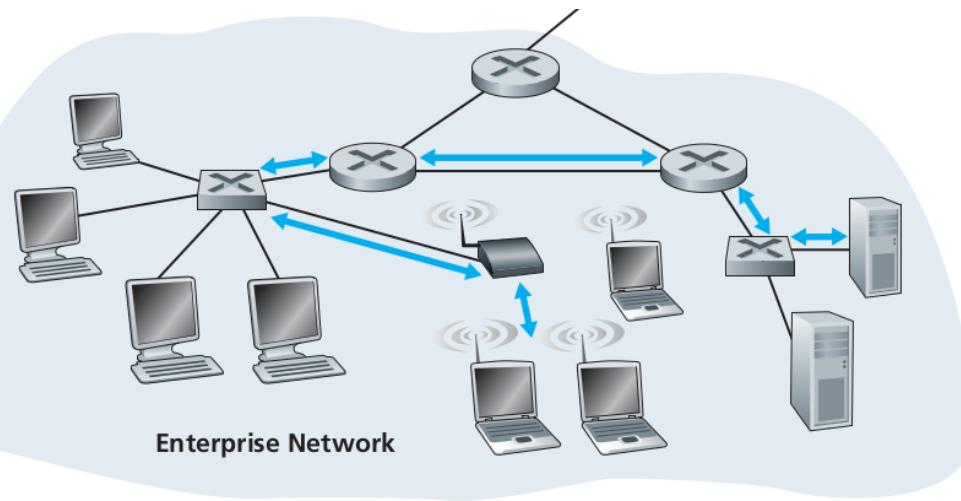
Netzwerkschicht: Internet Protocol (IP)



- Struktur eines **IP-Datagramms** (Datenpaket der Netzwerkschicht) für IPv4 (32-Bit-Adressen)
- Die Zahlen über dem Diagramm bezeichnen Bits.
- IHL: internet header length
- type of service (TOS) und flags bestehen aus mehreren Feldern.

Verbindungsschicht

- Geräte, die ein Protokol der **Verbindungsschicht** (link layer) implementieren, werden üblicherweise **Knoten** (nodes) genannt.
- Knoten sind z.B. Wirte (hosts), Vermittlungsstellen (router), Weichen (switches), Drahtloszugriffspunkte (WiFi access points) etc.
- Die Datenobjekte auf dieser Schicht heißen **Verbindungsschichtrahmen** (link layer frames), speziell z.B. Ethernet-Rahmen (Ethernet frames).



Die Verbindungsschicht verwaltet die einzelnen Verbindungen (links) zwischen Geräten auf dem Weg vom Sender zum Empfänger und sendet Datenrahmen über diese Verbindungen.

Verbindungsschicht

Die Verbindungsschicht stellt folgende Dienste zur Verfügung:

- **Einrahmen (framing):**
Einkapselung der Datagramme der Netzwerkschicht in Datenrahmen durch Hinzufügen von Kopffeldern (header fields) z.B. für MAC-Adressen (media access control addresses) und Signalsynchronisierung.
- **Verbindungszugriff (link access):**
Ein Protokol für den Medienzugriff (media access control protocol) spezifiziert, wie Datenrahmen auf der Verbindung übertragen werden. Dies umfaßt auch die Übersetzung von IP- in MAC-Adressen (address resolution protocol, ARP).
- **Verlässliche Zustellung (reliable delivery):**
Ein Verbindungsschichtprotokol kann ggf. verlässliche Übertragung garantieren, z.B. durch Empfangsbestätigung und ggf. Neuübertragung.
- **Fehlererkennung und -korrektur (error detection and correction):**
Durch einem Datenrahmen hinzugefügte Prüfbits bzw. Prüfsummen können Bitfehler erkannt und u.U. sogar korrigiert werden.

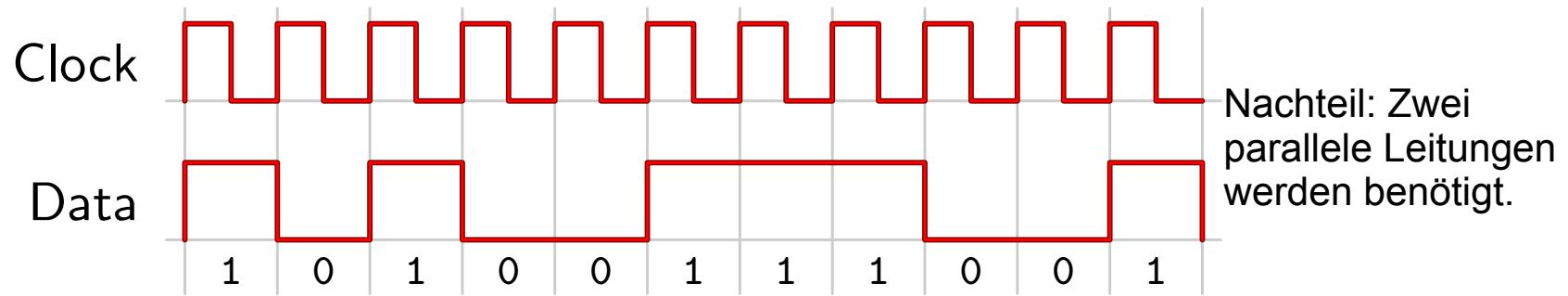
Verbindungsschicht: Ethernet IEEE 802.3

bit. seq. 10101010	bit. seq. 10101011	Ethernet frame 64 – 1518 byte									inter frame gap $9.6\mu s$
preamble 7 byte	SFD 1 byte	dst. addr. 6 byte	src. addr. 6 byte	length 2 byte	DSAP 1 byte	SSAP 1 byte	control 1 byte	data 43 – 1497 byte		FCS 4 byte	

- Präambel und Datenrahmenstartmarkierung (start of frame delimiter, SFD) dienen zur Takt synchronisierung mit dem Empfänger.
- Ziel- und Quelladresse sind die MAC-Adressen der verbundenen Geräte.
- Die Länge gibt an, aus wieviel Bytes (8-Bit-Werte) die Daten bestehen. Bei weniger als 43 Bytes Daten wird aufgefüllt.
- DSAP: destination service access point
SSAP: source service access point (hier nicht näher betrachtet)
- Control: definiert den “logical link” (LLC) des Protokolls (für Medienteilung).
- Eine Prüfsumme (frame check sequence, FCS) dient zur Fehlererkennung.

Physische Schicht: Übertragen von Bitströmen

- Auf der physischen Schicht werden binär kodierte Daten (also Bits) übermittelt.
- Darstellung durch Spannungspegel: eine elektrische Verbindungsleitung wird
 - für ein 1-Bit mit der Versorgungsspannung,
 - für ein 0-Bit mit Masse verbunden (für eine bestimmte Zeitspanne).
- Problem dieses Ansatzes: Wann hört ein Bit auf und fängt das nächste an?
(speziell bei langen Folgen von gleichen Bits)
- Eine einfache Zeitmessung ist unzureichend, da man nicht sicherstellen kann, daß die „Uhren“ von Sender und Empfänger exakt gleich schnell laufen.
- Mögliche Lösung: Parallele Übertragung eines Taktsignals zur Synchronisierung.

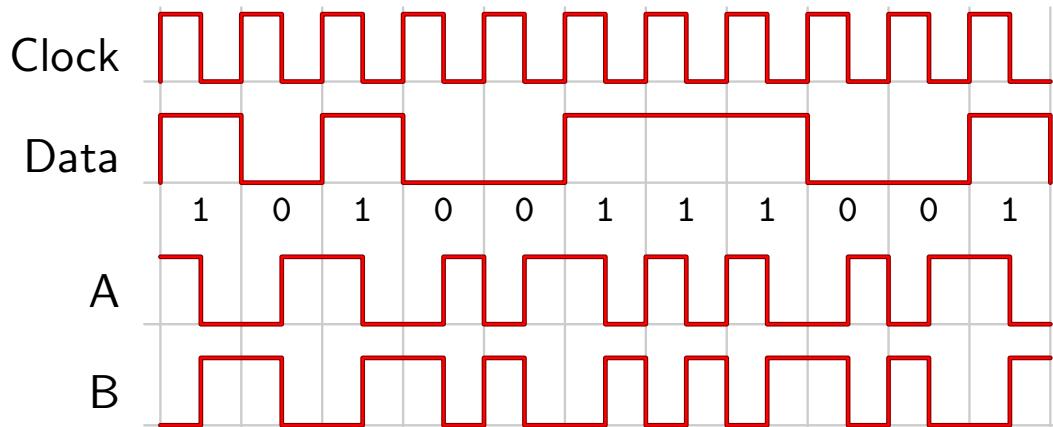
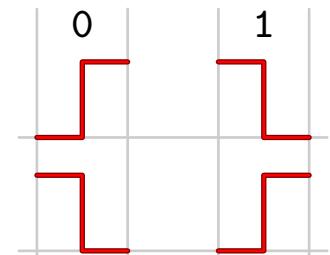


Physische Schicht: Manchester-Kode

- Alternative Lösung: Vereinigung von Taktsignal und Daten zu einem Signal.
- Beispiel: **Manchester-Kode** (z.B. 10Mbps Ethernet nach IEEE 802.3)
Die Flanken des Signals, bezogen auf das Taktsignal, beschreiben die Daten.
- Zwei mögliche, prinzipiell gleichwertige Definitionen:

A: G.E. Thomas: 0-Bit: steigende, 1-Bit: fallende Flanke

B: IEEE 802.3: 0-Bit: fallende, 1-Bit: steigende Flanke



Flanken in der Mitte eines Taktzyklus kodieren die Daten.

Flanken zu Beginn eines Taktzyklus stellen die nötigen Signalpegel her.

Netzwerksicherheit

Kernthemen der Netzwerksicherheit:

- Wie können Rechnernetze von böswilligen Personen angegriffen werden?
- Wie können Rechnernetze gegen solche Angriffe gesichert werden?
- Wie kann man Architekturen entwerfen, die gegen Angriffe immun sind?

Mangelnde Sicherheit ist ein Geburtsfehler der Rechnernetze.

- Startpunkt: Eine Gruppe einander vertrauernder Benutzer, die an ein transparentes Rechnernetz angeschlossen sind.
- Zugangskontrolle, Abhörsicherheit, Authentifizierung etc. anfangs vernachlässigt.
- Viele Sicherungsmechanismen wurden erst nachträglich „hineingeflickt“.
- Es fehlt ein schichtenübergreifendes, klares und konsistentes Sicherheitskonzept.
Dies begünstigt fehlerhafte und unvollständige Implementierungen selbst der Mechanismen, die man nachträglich hinzugefügt hat.

Netzwerksicherheit

- **Schadprogramme** (auch Schadsoftware, malware) sind Programme, die entwickelt wurden, um unerwünschte oder sogar schädliche Funktionen auszuführen.
- Einige Beispiele für Schadprogramme sind:
 - **Virus**: schreibt Kopien seiner selbst in andere Programme, Dokumente oder auf Datenträger; bedarf meist einer Aktion des Wirtes für seine Verbreitung.
 - **Wurm**: verbreitet sich direkt über Rechnernetze mit dem Ziel in Rechner einzudringen; bedarf keiner Aktion eines Wirtes für seine Verbreitung.
 - **Trojanisches Pferd**: Kombination eines (ggf. scheinbar) nützlichen Wirtsprogrammes mit einem versteckt arbeitenden, bösartigen Teil.
 - **Scareware**: verunsichert einen Nutzer, um ihn dazu zu verleiten, schädliche Software zu installieren oder für ein unnützes Produkt zu bezahlen (z.B. gefälschte Virenwarnmeldungen).
 - **Ransomware**: blockiert den Zugriff auf das Betriebssystem bzw. verschlüsselt Dateien und fordert den Benutzer zur Zahlung von Lösegeld (ransom) auf.

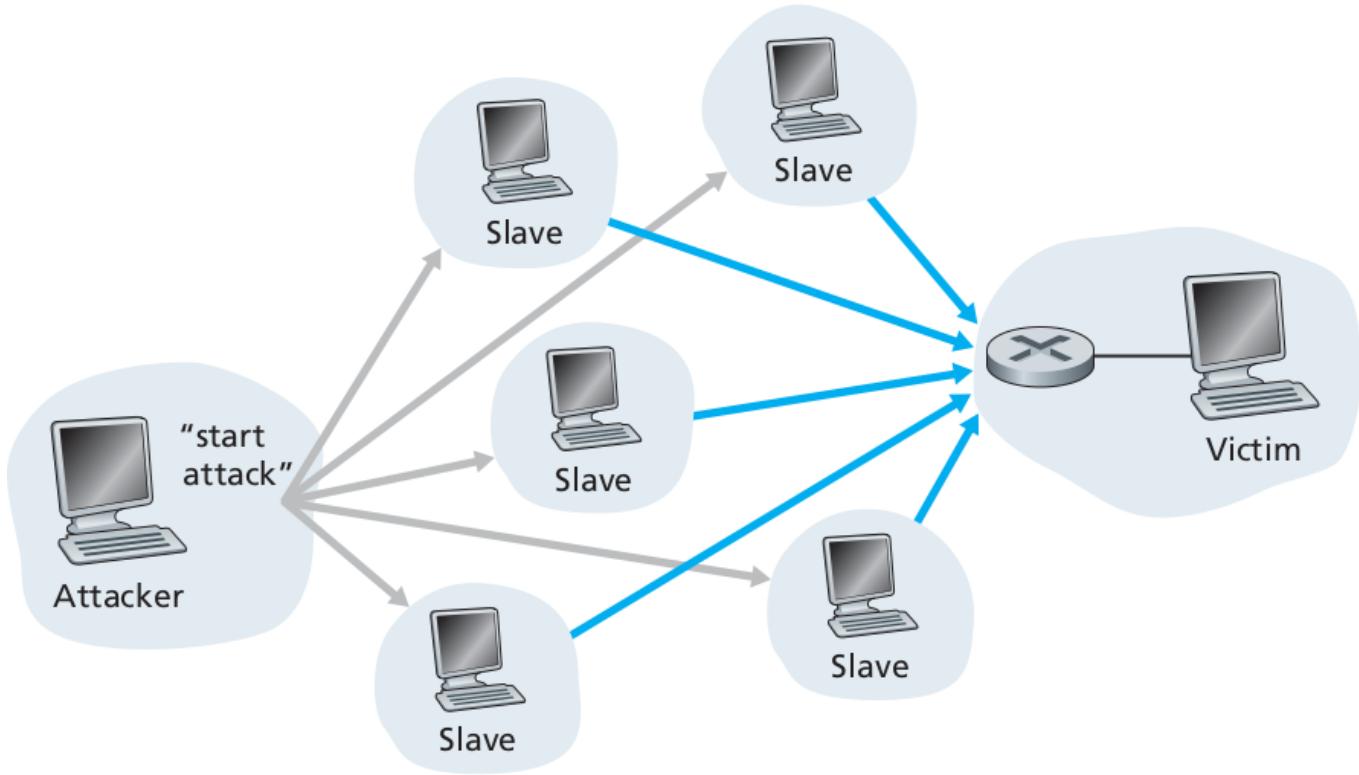
Netzwerksicherheit

- Einige Beispiele für Schadprogramme sind:
 - **Spyware:** forscht den Rechner und das Nutzerverhalten aus (z.B. Mitprotokollieren von Eingaben) und sendet die Daten an den Hersteller.
 - **Hintertür (Backdoor):** ermöglicht Außenstehenden einen unbefugten, aber versteckten Zugang zum Computer; werden z.B. durch Viren, Würmer etc. installiert; Sammlungen so kompromittierter Rechner können zu **Botnets** zusammengeschlossen werden.
- Angriffe auf Netzbediener und die Netzinfrastruktur
 - **Denial-of-Service (DoS):** Dienstverweigerung

Häufigster Grund für eine Dienstverweigerung in Rechnernetzen ist eine Überlastung des Rechnernetzes (durch zu viele Datenpakete).
Grund kann der konzertierte Angriff (z.B. durch ein Botnet) auf einen Netzbediener oder sonstige Komponenten des Netzes sein.
Sinnlose Anfragen und Datenübertragungen blockieren legitime Datenpakete.

Quelle: Wikipedia

Netzwerksicherheit



Kurose/Ross: Computer Networking

- Bei einem einfache Denial of Service (DoS) Angriff, kann das Netz im Prinzip den Rechner, von dem der Angriff ausgeht, identifizieren und dann ignorieren.
- In einem **distributed Denial of Service (DDoS)** Angriff werden vom Angreifer dagegen mehrere Quellen, z.B. ein Botnet, kontrolliert.

Netzwerksicherheit

- **Paketschnüffler (packet sniffer)**

Ürsprünglich Programme, die zur Netzwerkanalyse entwickelt wurden, um Auffälligkeiten im Datenverkehr eines Netzes zu erkennen.

Diese können aber auch von böswilligen Personen eingesetzt werden, um Datenverkehr anderer „mitzuhören“ (Form der Spyware).

- Im sogenannten non-promiscuous mode wird nur der Datenverkehr eines Rechners überwacht. Im promiscuous mode wird der gesamte Datenverkehr einer Netzwerkschnittstelle gesammelt und ausgewertet.

- Bei Netzwerkkomponenten mit quasi öffentlicher Übertragung (broadcast, z.B. Drahtlosnetzwerke, geteiltes Ethernet, Kabelfernsehen etc.) können auch andere als die eigentlichen Adressaten einer Übertragung alle Datenpakete sehen und auswerten.

Dies kann, bei fehlender Verschlüsselung, Paßwörter im Klartext beinhalten!

- Ein frei verfügbarer Paketschnüffler ist das Programm Wireshark („Kabelhai“).

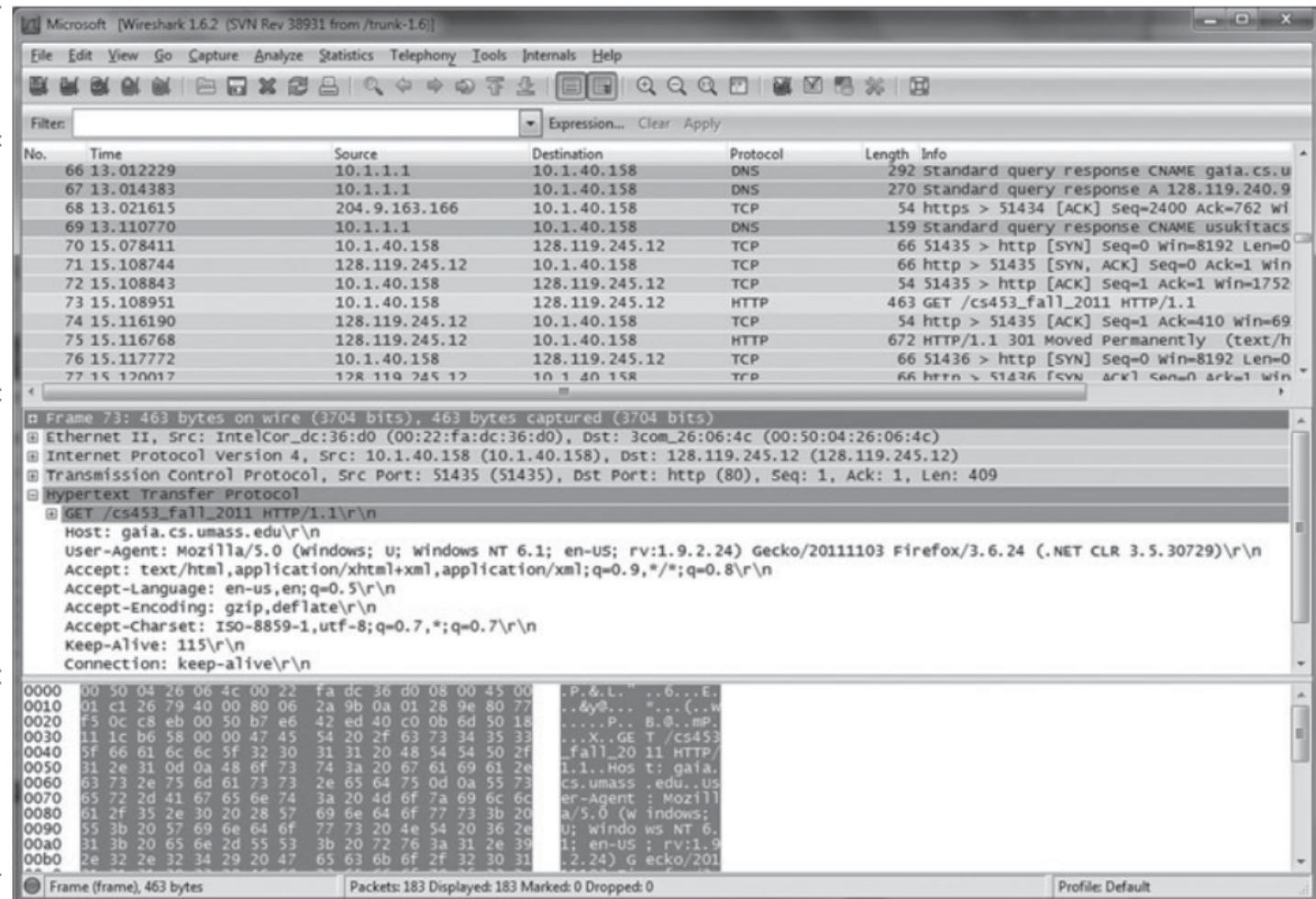
Netzwerksicherheit

Command menus

Listing of captured packets

Details of selected packet header

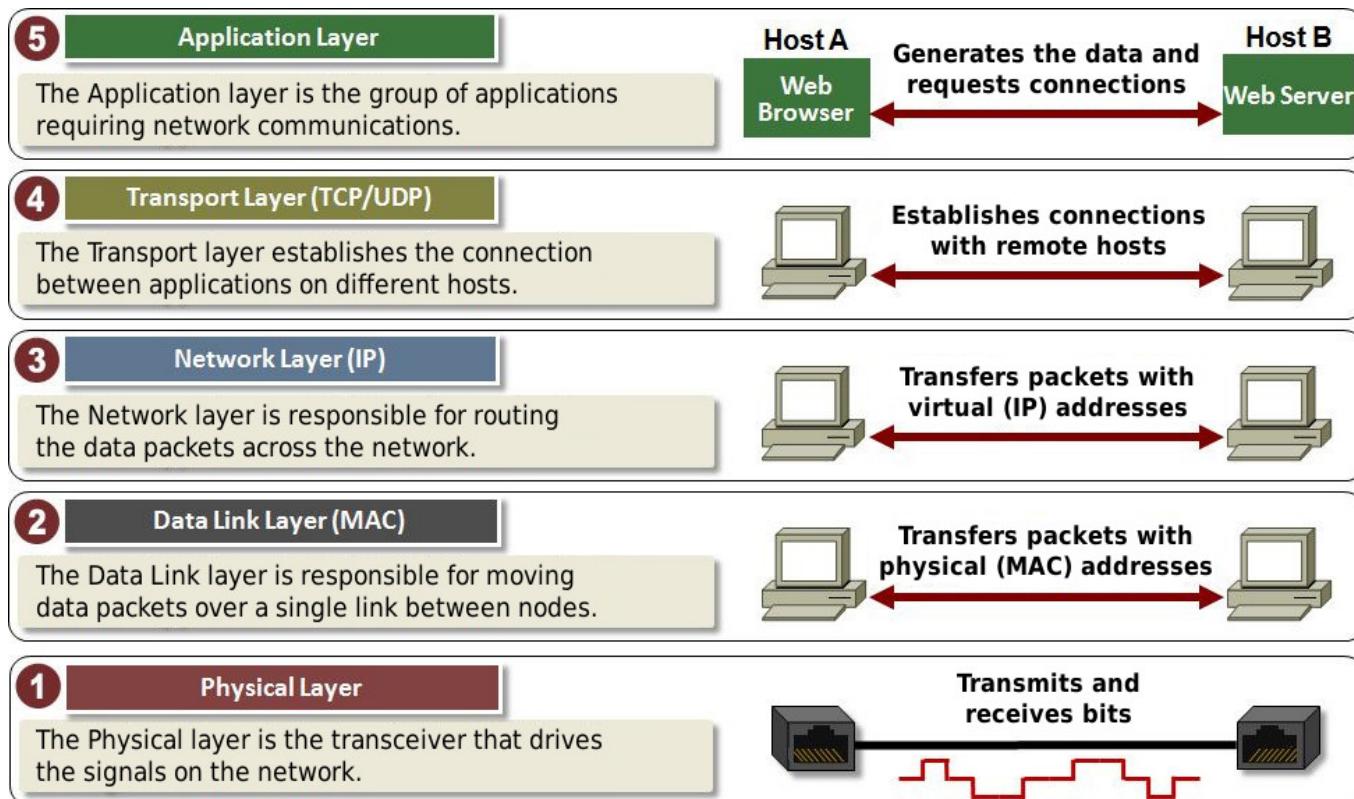
Packet contents in hexadecimal and ASCII



Bildschirmfoto des Paketschnüfflers Wireshark, das seine Funktionalität zeigt.

Rechnernetze: Zusammenfassung

- Durch Paketvermittlung wird die Übertragungskapazität physischer Verbindungen besser ausgenutzt als durch Leitungszuordnung oder Multiplexverfahren.
- Zu jeder Kommunikationsschicht gehören ein oder mehrere Protokolle, die festlegen, wie auf dieser Schicht Daten übertragen werden.



2.bp.blogspot.com (modifiziert)