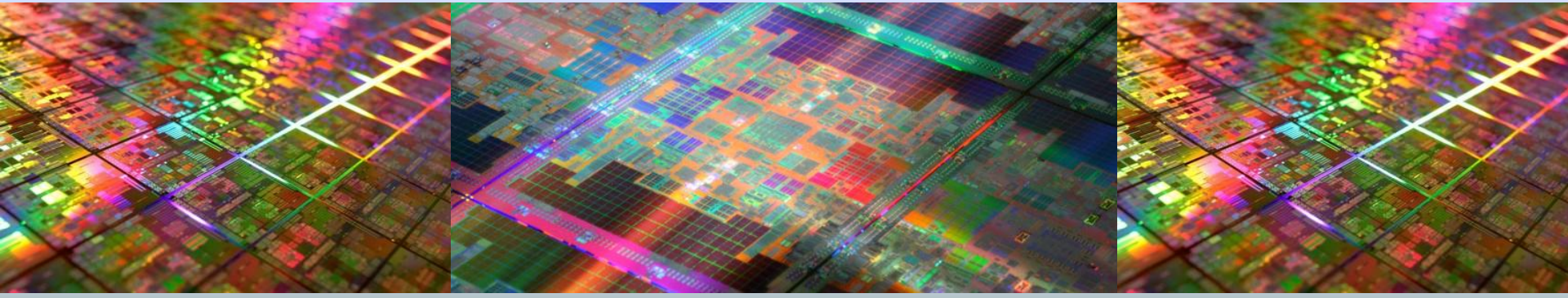
Rechnernetze

Technische Grundlagen der Informatik für Wirtschaftsinformatik

Stefan Podlipnig

TU Wien



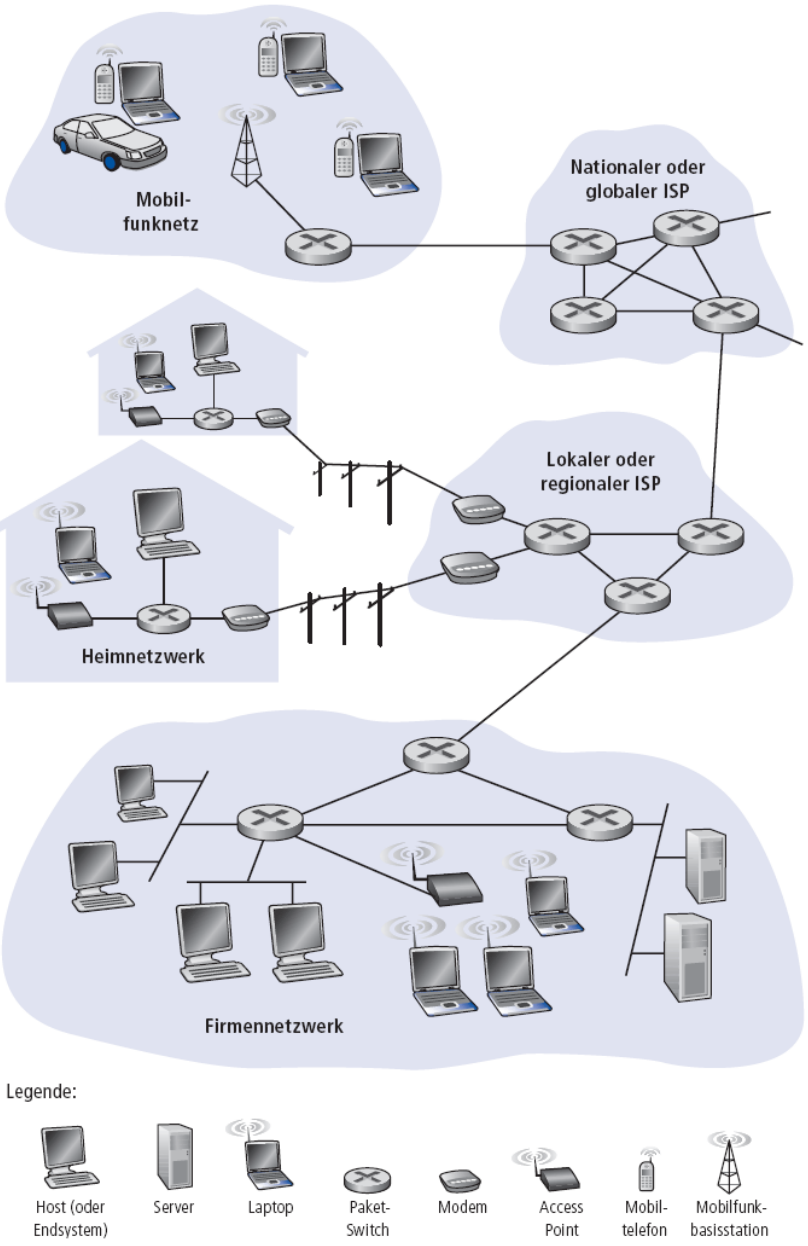
Lernziele

- Kennenlernen der grundlegenden Terminologie
- Kennenlernen einfacher technischer Grundlagen
- Verständnis für Prinzipien und Protokolle im Bereich Rechnernetze entwickeln
- Lösen einfacher Rechenbeispiele

INTERNET - ÜBERBLICK

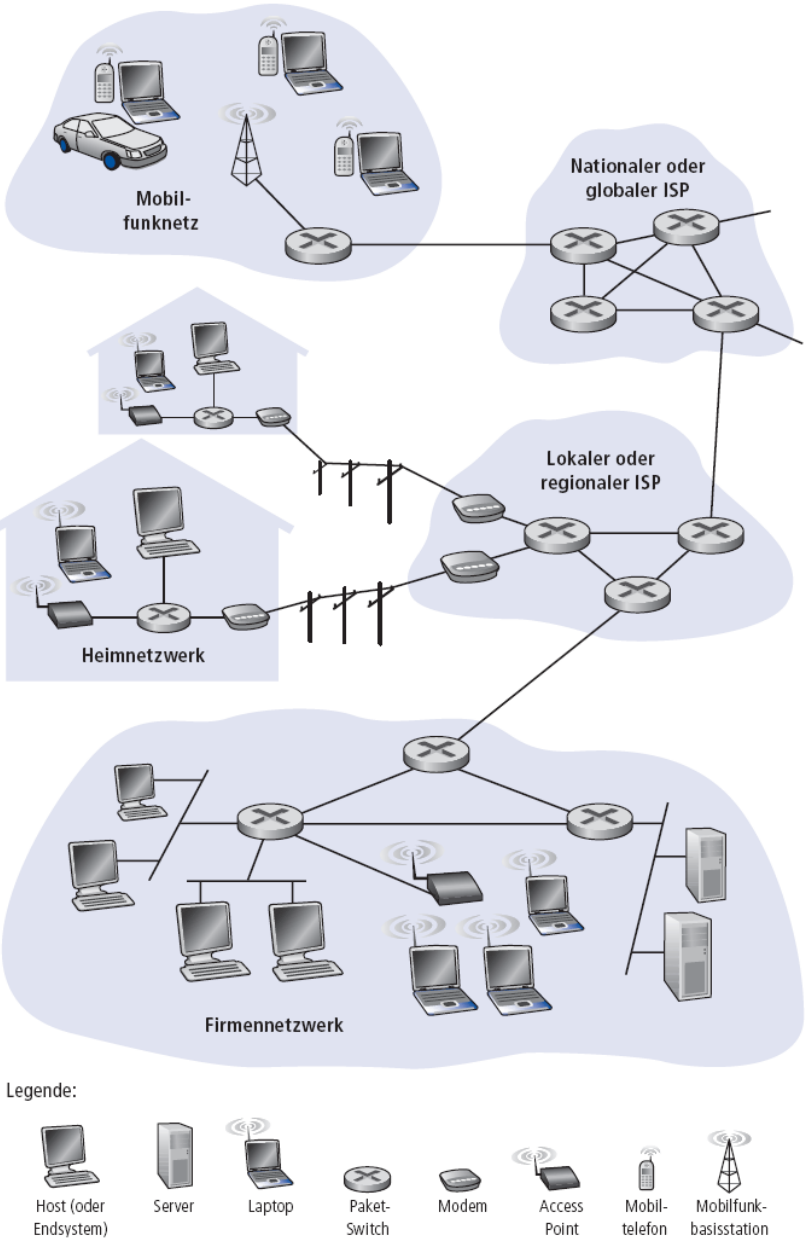
Internet – elementare Grundlagen (1)

- Millionen vernetzter Computer genannt Hosts = Endsysteme
  - Auf diesen Endsystemen laufen Netzwerkanwendungen
- Leitungen/Funkstrecken (*links*)
  - Glasfaser, Kupfer, Funk, Satellit
  - Datenübertragungsrate = Bandbreite
    - Bits pro Sekunde
    - Z.B. 1 Gbit/s = 1 Gbps = 10^9 Bit/s
- Router
  - Leiten Datenpakete weiter
  - Datenpakete sind Einheiten von Daten



Internet – elementare Grundlagen (2)

- Protokolle kontrollieren das Senden und Empfangen von Nachrichten
  - z.B. TCP, IP, HTTP, Skype, Ethernet
- Internet: “Netzwerk von Netzwerken”
  - Hierarchisch
    - ISPs (Internet Service Provider)
  - Öffentliches Internet und privates Intranet
- Internetstandards
  - RFC: Request For Comments
  - IETF: Internet Engineering Task Force



Was versteht man unter einem Protokoll?

Protokolle zwischen Menschen

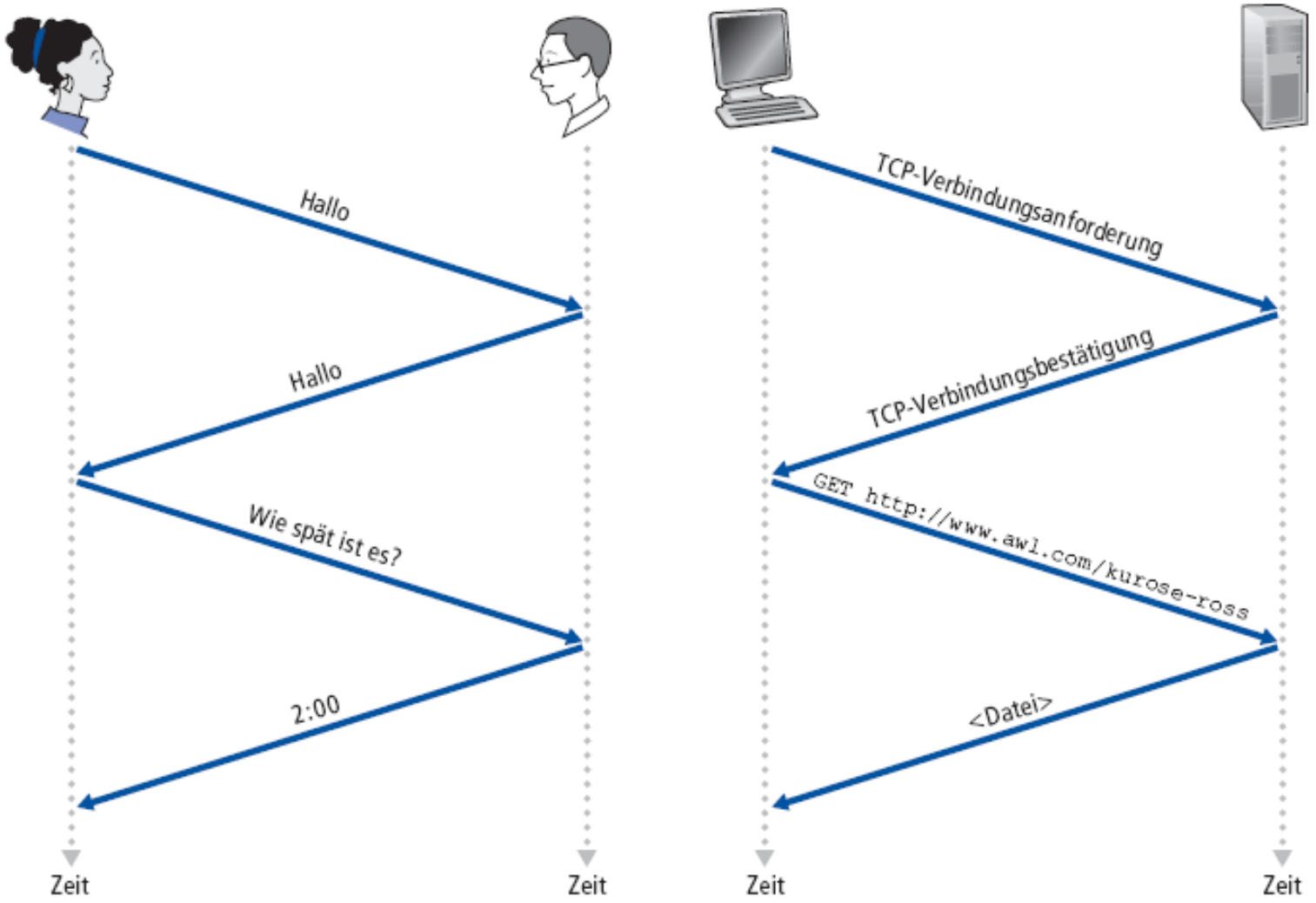
- Beispiele
  - “Wie viel Uhr ist es?”
  - “Ich habe eine Frage”
  - Gegenseitiges Vorstellen
- Bedeutet
  - … es werden „standardisierte“ Nachrichten übertragen
  - … durch den Empfang dieser Nachrichten werden „standardisierte“ Aktionen ausgelöst

Netzwerkprotokolle

- Maschinen statt Menschen
- Sämtliche Kommunikation im Internet wird durch Protokolle geregelt
- **Protokolle definieren das Format und die Reihenfolge, in der Nachrichten von Systemen im Netzwerk gesendet und empfangen werden, sowie die Aktionen, welche durch diese Nachrichten ausgelöst werden**

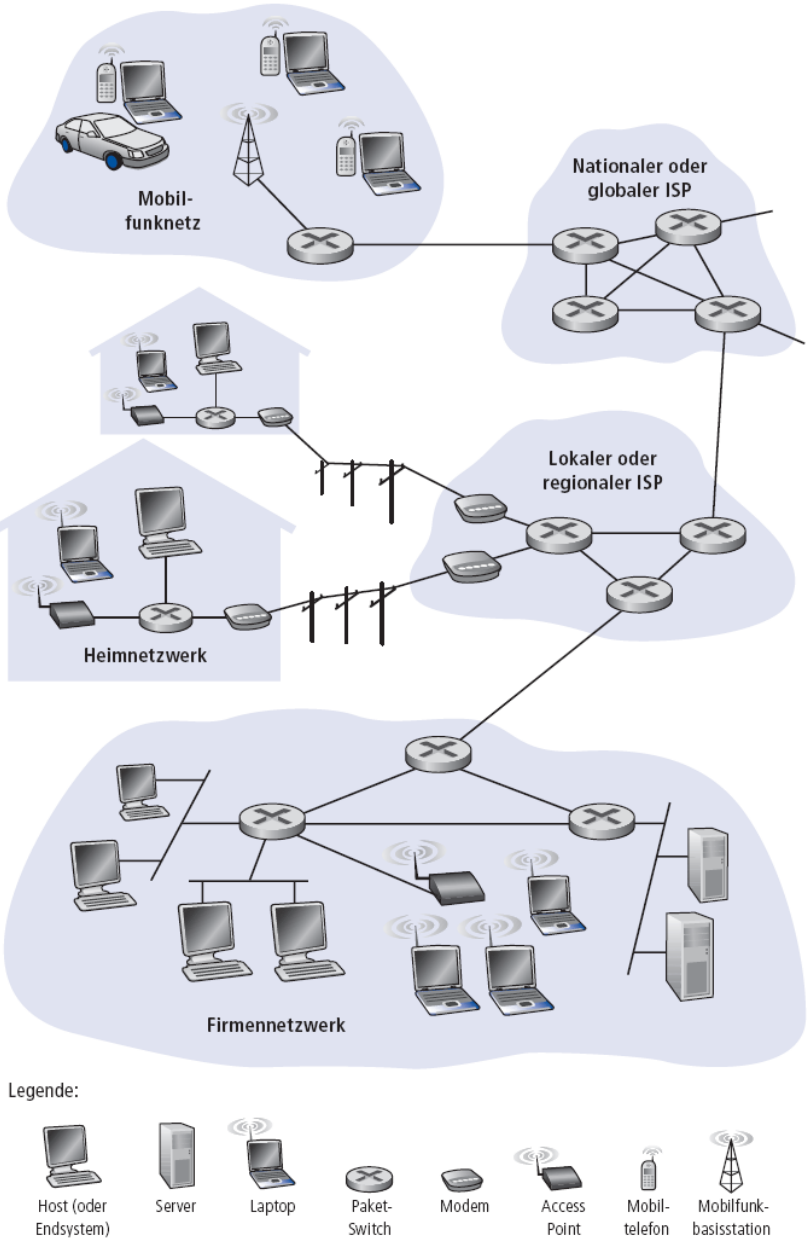
Was ist ein Protokoll?

- Vergleich von Protokollen



Internet – dienstorientierte Sichtweise

- Kommunikationsinfrastruktur, die verteilte Anwendungen ermöglicht
  - Web, VoIP, E-Mail, Spiele, eCommerce, File Sharing usw.
- Kommunikationsdienste, die den Anwendungen zur Verfügung gestellt werden
  - Zuverlässige Datenübertragung von einer Quelle zu einem Ziel
  - Unzuverlässige ("best effort") Datenübertragung



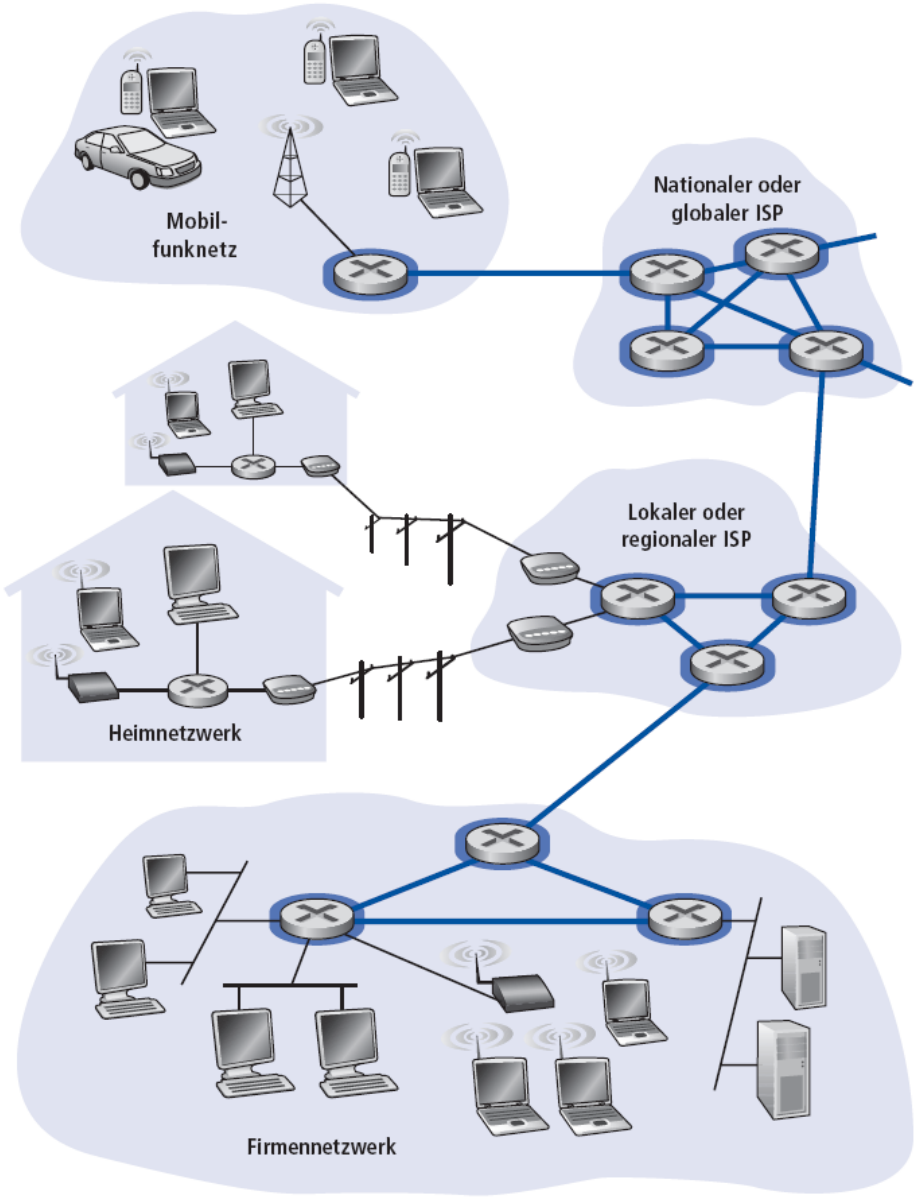
Netzwerkklassifizierung

- Klassifizierung nach Übertragungstechnik
  - Broadcast-Netze
    - Alle Rechner benutzen einen Übertragungskanal
  - Punkt-zu-Punkt-Netze
    - Verbindungen zwischen einzelnen Paaren von Rechnern
- Klassifizierung nach Ausdehnung

Entfernung der Prozessoren	Prozessoren im gleichen	Beispiel
1 m	Quadratmeter	Persönliches Netz (**PAN** – Personal Area Network)
10 m	Raum	Lokales Netz (**LAN** – Local Area Network)
100 m	Gebäude	
1 km	Gelände	
10 km	Stadt	Stadtnetz (**MAN** – Metropolitan Area Network)
100 km	Land	Fernnetz bzw. Weitverkehrsnetz (**WAN** – Wide Area Network)
1000 km	Kontinent	
10000 km	Planeten	**Internet**

Das Innere des Netzwerkes

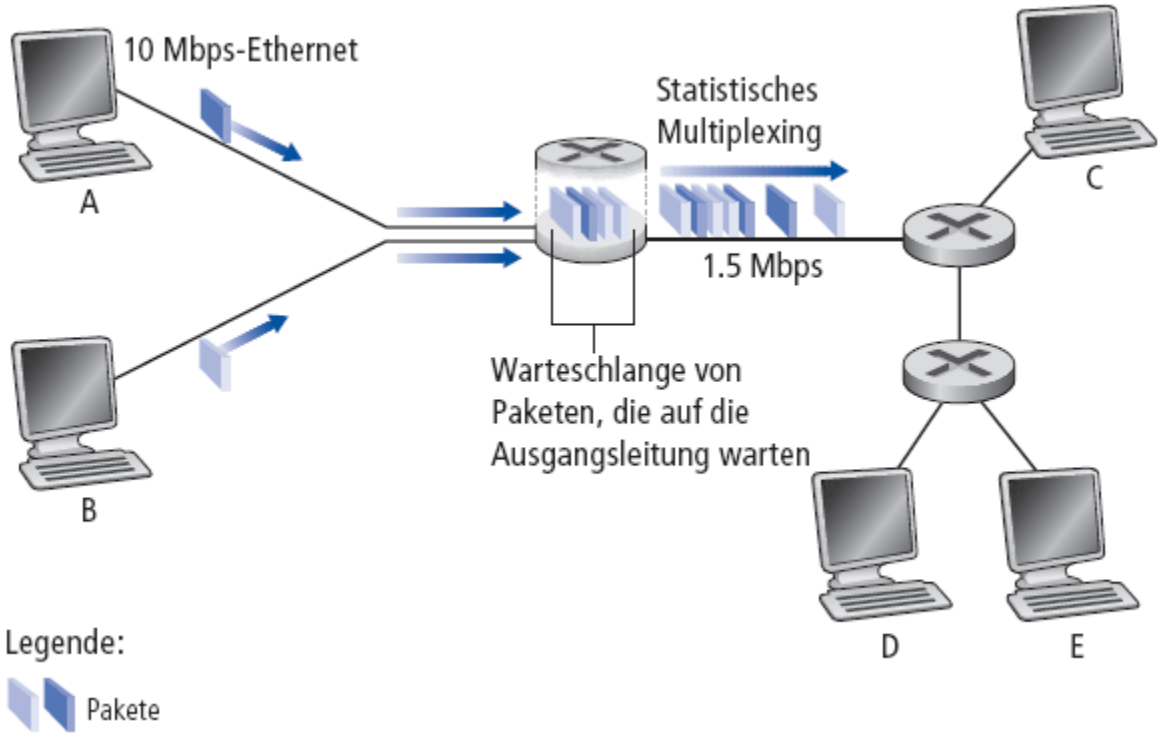
- Viele, untereinander verbundene Router
- Die zentrale Frage: Wie werden Daten durch das Netzwerk geleitet?
  - Leitungsvermittlung: eine dedizierte Leitung wird für jeden Ruf geschaltet - Telefonnetz
  - Paketvermittlung: Daten werden in diskreten Einheiten durch das Netzwerk geleitet - Internet



Paketvermittlung

- Jeder Ende-zu-Ende-Datenstrom wird in Pakete aufgeteilt
  - Die Pakete aller Nutzer teilen sich die Netzwerkressourcen
  - Jedes Paket nutzt die volle Bandbreite der Leitung
  - Ressourcen werden nach Bedarf verwendet
- Wettbewerb um Ressourcen
  - Die Nachfrage nach Ressourcen kann das Angebot übersteigen
  - Überlast: Pakete werden zwischengespeichert und warten darauf, eine Leitung benutzen zu können
  - Teilstreckenverfahren (Store and Forward)
    - Pakete durchqueren eine Leitung nach der anderen
    - Knoten (z.B. Router) empfangen ein komplettes Paket, bevor sie es weiterleiten
- Konsequenzen
  - Keine Aufteilung der Bandbreite in Einheiten
  - Keine dedizierte Zuweisung
  - Keine Reservierung von Ressourcen

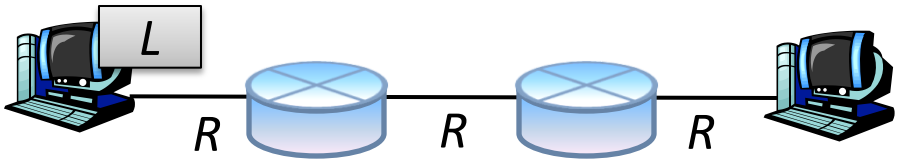
Paketvermittlung – Statistisches Multiplexing



- Die Folge von Paketen auf der Leitung hat kein festes Muster, d.h. die Bandbreite wird nach Bedarf verteilt - statistisches Multiplexing

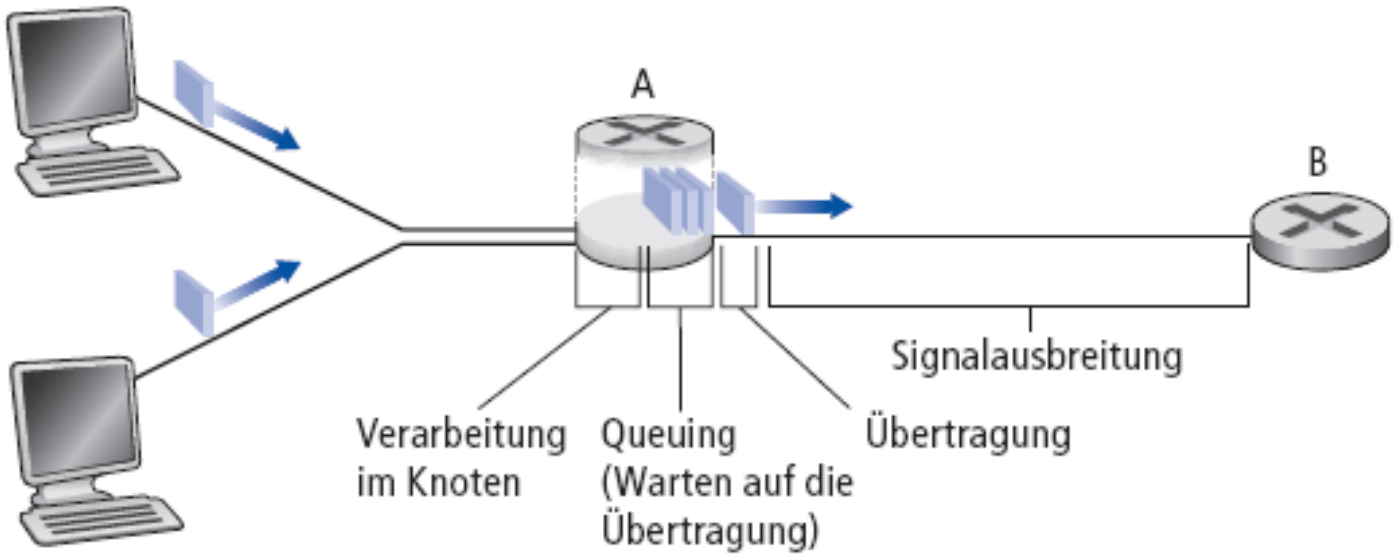
Paketvermittlung – Store and Forward

- Es dauert L/R Sekunden, um ein Paket von L Bits auf einer Leitung mit der Rate R bps zu übertragen
- Store and Forward
  - Das gesamte Paket muss bei einem Router angekommen sein, bevor es auf der nächsten Leitung übertragen werden kann
- Verzögerung = QL/R bei Q Links
  - Wenn die Ausbreitungsverzögerung vernachlässigt wird!
- Beispiel
  - L = 7,5 Mbits
  - R = 1,5 Mbit/s
  - Übertragungsverzögerung = 15 s (ungefähr, noch kein Overhead dabei)

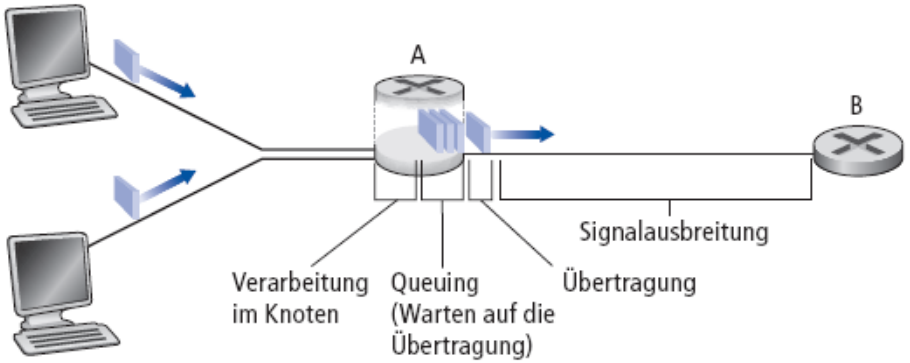


Wie entstehen Paketverluste und Verzögerungen?

- Pakete warten in den Puffern von Routern
- Verzögerung bzw. Verlust
  - Wenn die Ankunftsrate die Kapazität der Ausgangsleitungen übersteigt



Vier Quellen 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 Last auf der Ausgangsleitung ab
3. Übertragungsverzögerung
   - R = Bandbreite einer Leitung (Bit/s)
   - L = Paketgröße (Bit)
   - Übertragungsverzögerung = L/R
4. Ausbreitungsverzögerung
   - d = Länge der Leitung
   - s = Ausbreitungsgeschwindigkeit des Mediums (~2×10^8 m/s)
   - Ausbreitungsverzögerung = d/s

Wichtig

s und R sind sehr verschiedene Größen

Gesamtverzögerung für einen Knoten

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

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

Protokollschichten

- Netzwerke sind komplex!
- Viele Elemente
  - Hosts, Router
  - Leitungen auf Basis verschiedener Medien
  - Anwendungen
  - Protokolle
  - Hardware, Software
- Lösung
  - Aufteilung der Kommunikation in Netzwerken in mehrere Schichten
  - Strukturierung ermöglicht die Identifikation und das Verständnis des Zusammenspiels einzelner Bestandteile des Systems
  - Referenzmodell für die Diskussion des Systems
  - Änderungen an der Implementierung einer Schicht sind transparent für den Rest des Systems (Modularisierung)

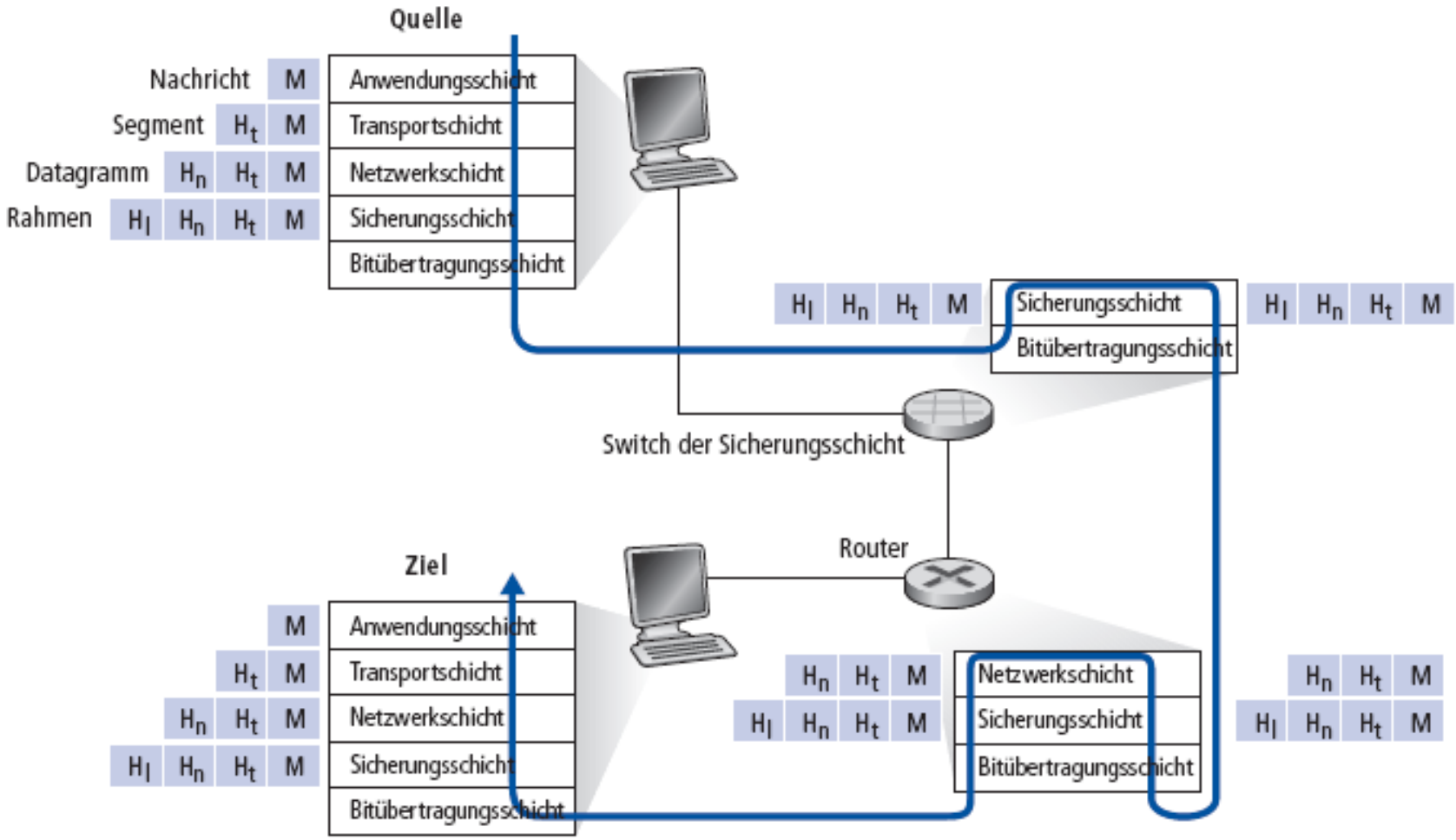
Protokollstapel des Internets

- Anwendungsschicht (*Application layer*)
  - Unterstützung von Netzwerkanwendungen
  - FTP, SMTP, HTTP
- Transportschicht (*Transport layer*)
  - Datentransfer zwischen Programmen
  - TCP, UDP
- Netzwerkschicht oder Vermittlungsschicht (*Network layer*)
  - Weiterleiten der Daten von einem Sender zu einem Empfänger
  - IP, Routing-Protokolle
- Sicherungsschicht (*Link layer*)
  - Datentransfer zwischen benachbarten Netzwerksystemen
  - PPP, Ethernet
- Bitübertragungsschicht (*Physical layer*)
  - Bits auf der Leitung

ISO-OSI Referenzmodell

- 2 zusätzliche Schichten
- Darstellungsschicht (*Presentation layer*)
  - Ermöglicht es Anwendungen, die Bedeutung von Daten zu interpretieren, z.B. Verschlüsselung, Kompression, Vermeidung systemspezifischer Datendarstellung
- Kommunikationssteuerungsschicht (*Session layer*)
  - Synchronisation
  - Setzen von Wiederherstellungspunkten
- Der Protokollstapel des Internets bietet diese Funktionalitäten nicht
  - Wenn benötigt, müssen sie von der Anwendung implementiert werden

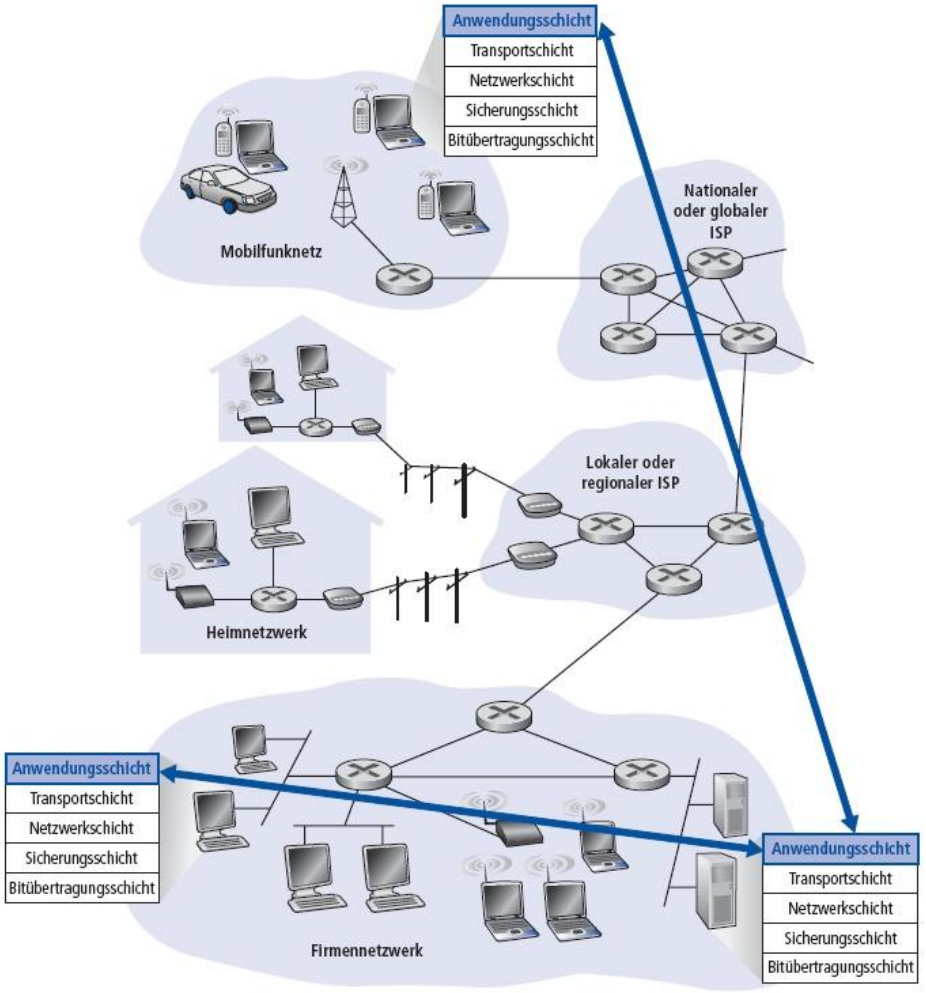
Nachrichten, Segmente, Datagramme, Rahmen



ANWENDUNGSSCHICHT

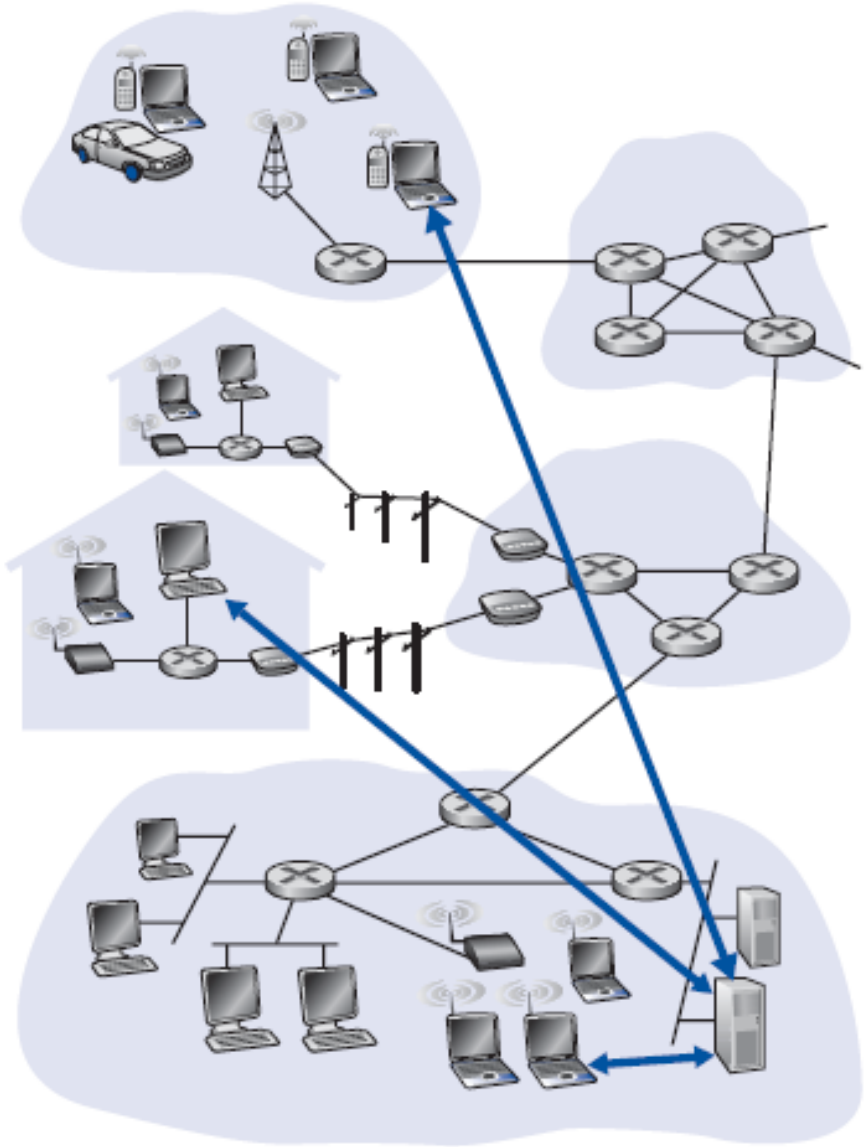
Entwickeln einer Netzwerkanwendung

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



Beispiel – Client-Server-Architektur

- Server
  - Immer eingeschaltet
  - Feste IP-Adresse (Internet-Adresse zum Auffinden)
- Clients
  - Kommunizieren mit Servern
  - Sporadisch angeschlossen
  - Können dynamische Internet-Adressen haben
  - Kommunizieren nicht direkt miteinander



Kommunizierende Prozesse

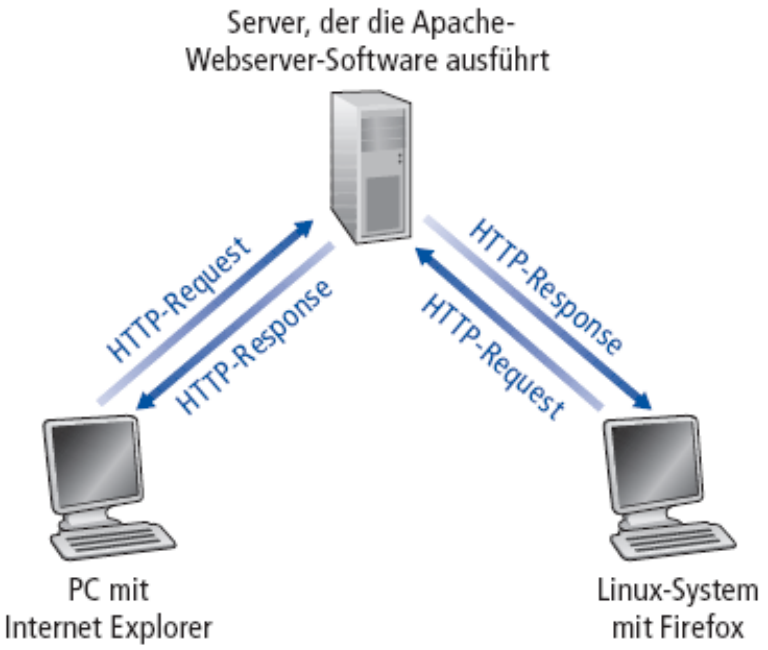
- Prozess
  - Programm, welches auf einem Host läuft
  - Prozesse auf verschiedenen Hosts kommunizieren, indem sie Nachrichten über ein Netzwerk austauschen
- Client-Prozess
  - Prozess, der die Kommunikation beginnt
- Server-Prozess
  - Prozess, der darauf wartet, kontaktiert zu werden

Adressierung von Prozessen

- Um eine Nachricht empfangen zu können, muss ein Prozess identifiziert werden können
- Ein Host besitzt eine eindeutige, 32 Bit lange IP-Adresse (bei IPv4)
- Reicht diese IP-Adresse, um einen Prozess auf diesem Host zu identifizieren?
  - Nein, denn viele Prozesse können auf demselben Host laufen!
  - Prozesse werden durch eine IP-Adresse UND eine Portnummer identifiziert
- Beispiel-Portnummern
  - HTTP-Server: `80`, E-Mail-Server: `25`
- Um an den Webserver `gaia.cs.umass.edu` eine HTTP-Nachricht zu schicken:
  - IP-Adresse: `128.119.245.12`
  - Portnummer: `80`

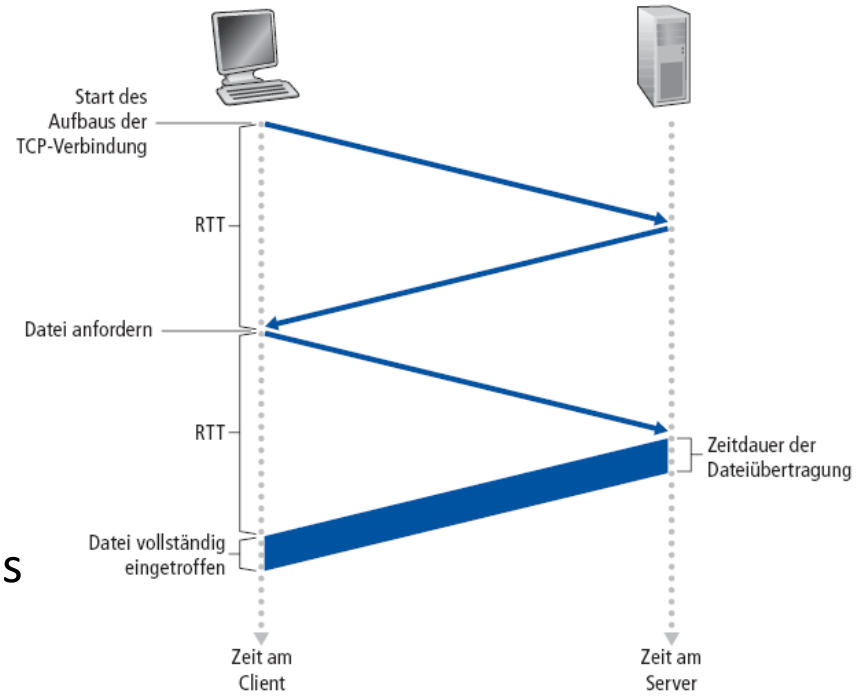
Beispiel – HTTP (1)

- HTTP (HyperText Transfer Protocol)
- Anwendungsprotokoll des Web
  - Verwendet TCP als Transportprotokoll
- Client/Server-Modell
  - Client: Browser, der Objekte anfragt, erhält und anzeigt
  - Server: Webserver verschickt Objekte auf Anfrage
- HTTP 1.0: [RFC 1945]
- HTTP 1.1: [RFC 2068]



Beispiel – HTTP (2)

- Einfachste Form (nichtpersistentes HTTP)
- RTT (Round Trip Time): Zeit, um ein kleines Paket vom Client zum Server und zurück zu schicken
- Verzögerung (idealisiert)
  - Eine RTT für den TCP-Verbindungsaufbau
  - Eine RTT für den HTTP-Request, bis das erste Byte der HTTP-Response beim Client ist
  - Zeit für das Übertragen der Daten auf der Leitung
- Zusammen = 2 RTT + Übertragungsverzögerung



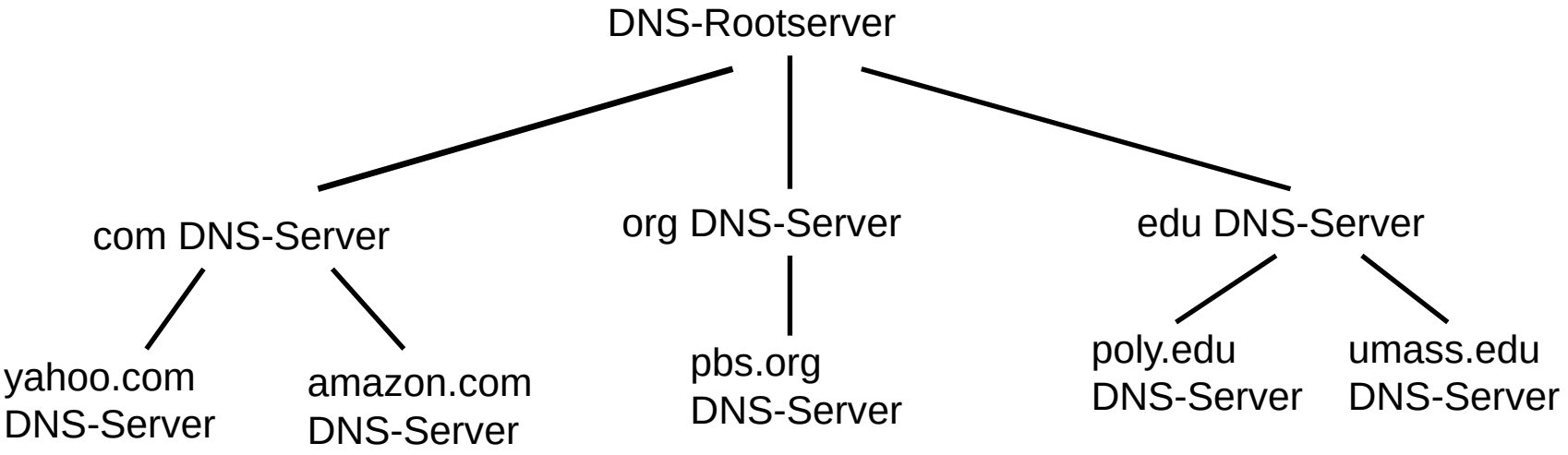
Beispiel – HTTP (3)

- Probleme mit nichtpersistentem HTTP (HTTP 1.0)
  - 2 RTTs pro Objekt
  - Aufwand im Betriebssystem für jede TCP-Verbindung
  - Browser öffnen oft mehrere parallele TCP-Verbindungen, um die referenzierten Objekte zu laden
- Persistentes HTTP (HTTP 1.1)
  - Server lässt die Verbindung nach dem Senden der Antwort offen
  - Nachfolgende HTTP-Nachrichten können über dieselbe Verbindung übertragen werden

Beispiel – DNS

- Internet-Hosts, Router
  - IP-Adresse (32 Bit) – für die Adressierung in Paketen
  - “Name”, z.B. `www.yahoo.com` – von Menschen verwendet
- Wie findet die Abbildung zwischen IP-Adressen und Namen statt?
- Domain Name System (DNS)
  - Verteilte Datenbank, implementiert eine Hierarchie von Nameservern
  - Wird von Hosts verwendet, um Namen aufzulösen
    - Zentrale Internetfunktion, implementiert als Protokoll der Anwendungsschicht
    - Grund: Komplexität nur am Rand des Netzwerkes!

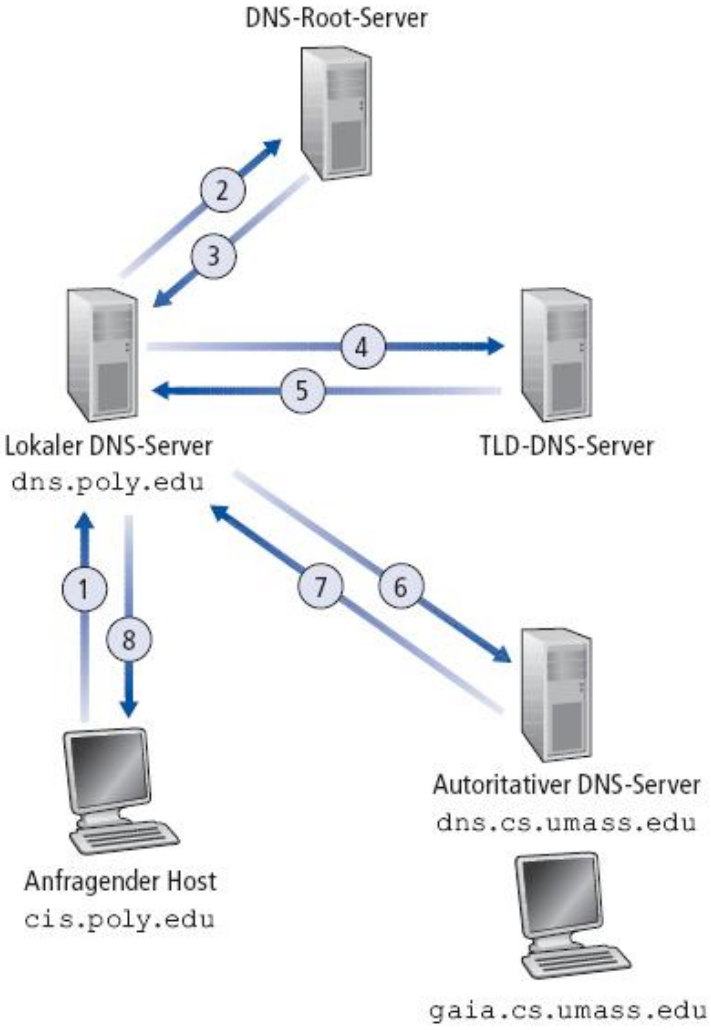
Verteilte, hierarchische Datenbank



- Root-Nameserver
  - Kennt die Adressen der Nameserver der Top-Level-Domains
- Top-Level-Domain (TLD)-Server
  - Verantwortlich für `com`, `org`, `net`, `edu` etc. sowie für alle Länder-Domains, z.B. `de`, `uk`, `fr`, `ca`, `jp`
- Autoritativer DNS-Server
  - DNS-Server einer Organisation, der eine autorisierte Abbildung der Namen dieser Organisation auf IP-Adressen anbietet

Namensauflösung mit DNS

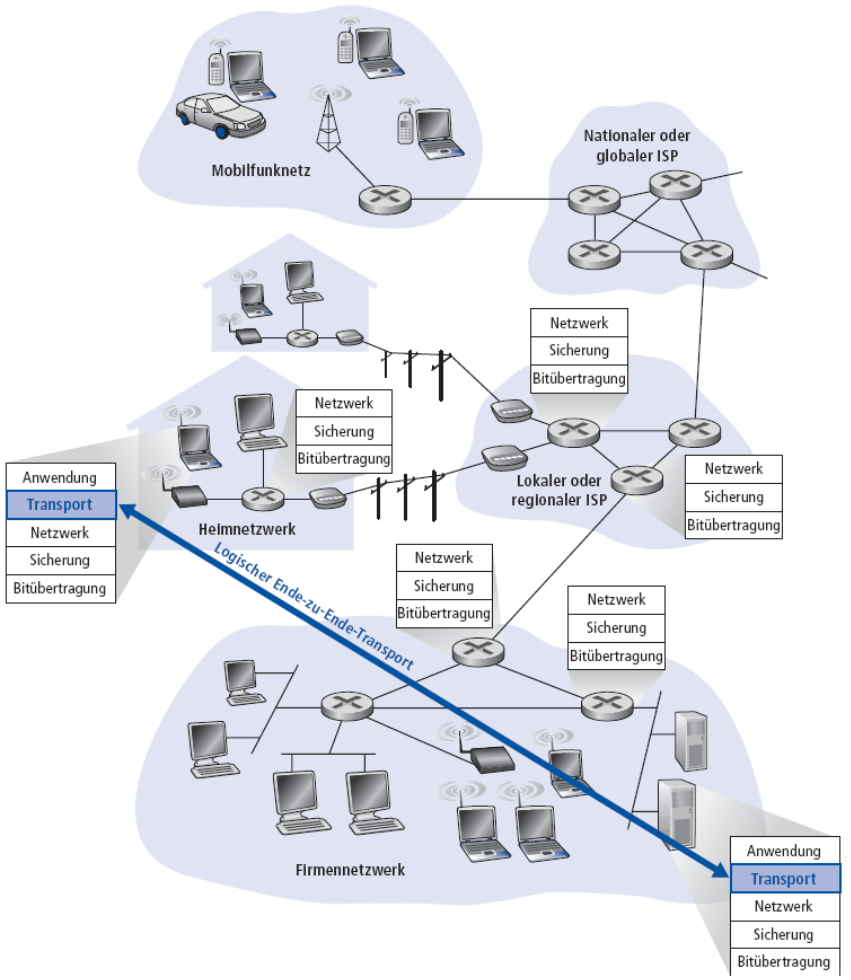
- Iterative Auflösung
- Caching
  - Sobald ein Nameserver eine Namensauflösung kennenlernt, merkt er sich diese in einem Cache
    - Die Einträge im Cache werden nach einer bestimmten Zeit wieder gelöscht
  - Die Adressen der TLD-Server werden üblicherweise von den lokalen Nameservern im Cache gespeichert
    - Root-Nameserver werden eher selten angesprochen



TRANSPORTSCHICHT

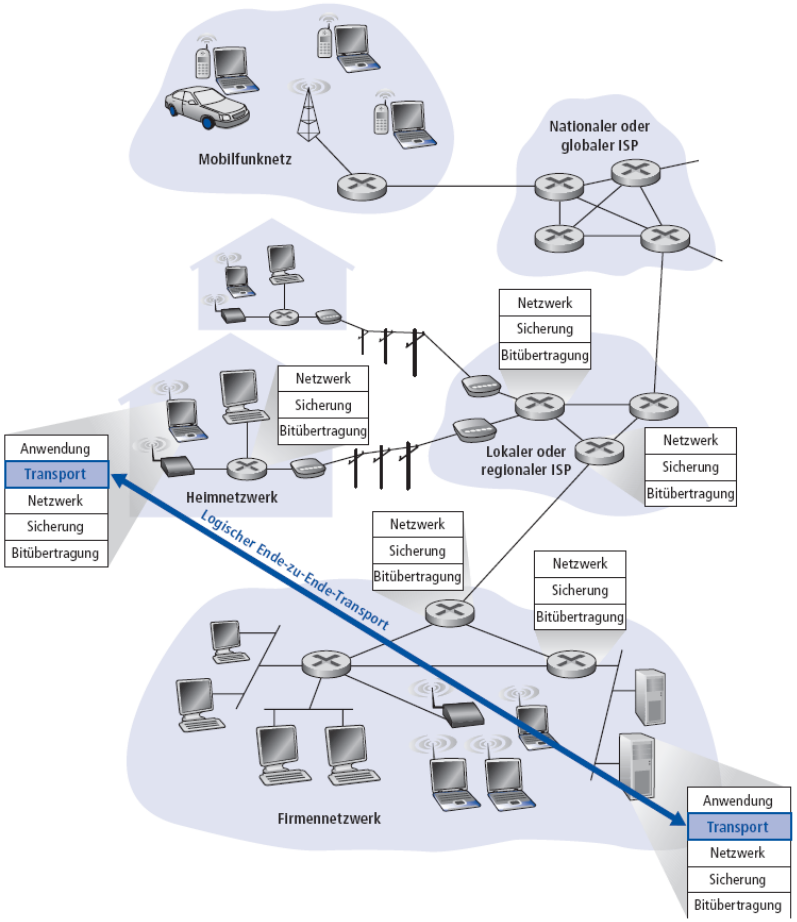
Transportdienste und -protokolle

- Stellen logische Kommunikation zwischen Anwendungsprogrammen auf verschiedenen Hosts zur Verfügung
- Transportprotokolle laufen auf Endsystemen
  - Sender teilt Anwendungsnachrichten in Segmente auf, gibt diese an die Netzwerkschicht weiter
  - Empfänger fügt Segmente wieder zu Anwendungsnachrichten zusammen, gibt diese an die Anwendungsschicht weiter
- Es existieren verschiedene Transportschichtprotokolle
  - Internet: TCP und UDP



Transportprotokolle im Internet

- Zuverlässige, reihenfolgeerhaltende Auslieferung (TCP)
  - Verbindungsmanagement
    - Verbindungen auf- und abbauen
  - Überlastkontrolle (für Netzwerk)
  - Flusskontrolle (bei Hosts)
- Unzuverlässige Datenübertragung ohne Reihenfolgeerhaltung (UDP)
  - Minimale Erweiterung der “Best-Effort”-Funktionalität von IP
- Dienste, die nicht zur Verfügung stehen
  - Garantien bezüglich Bandbreite oder Verzögerung



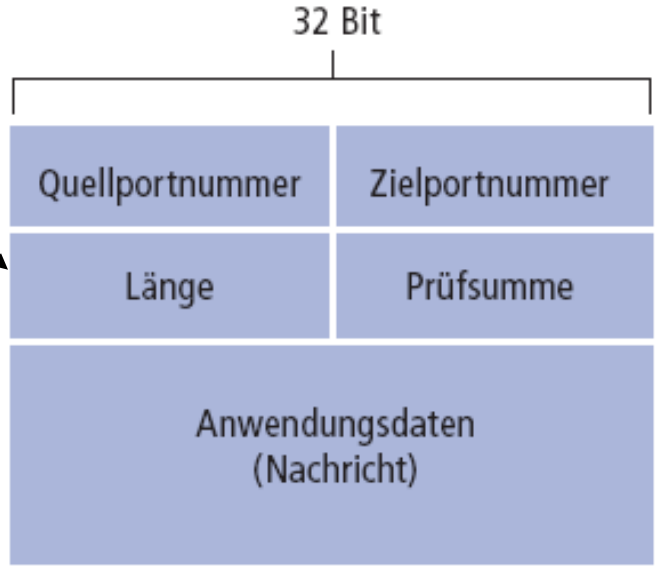
UDP – User Datagram Protocol

- Minimales Internet-Transportprotokoll
- “Best-Effort”-Dienst, UDP-Segmente können
  - verloren gehen
  - in der falschen Reihenfolge an die Anwendung ausgeliefert werden
- Verbindungslos
  - Kein Verbindungsaufbau
  - Jedes UDP-Segment wird unabhängig von allen anderen behandelt
- Warum gibt es UDP?
  - Kein Verbindungsaufbau (der zu Verzögerungen führen kann)
  - Einfach - kein Verbindungszustand im Sender oder Empfänger
  - Kleiner Header
  - Keine Überlastkontrolle - UDP kann so schnell wie von der Anwendung gewünscht senden

UDP – Fortsetzung

- Häufig für Anwendungen im Bereich Multimedia-Streaming eingesetzt
  - Verlusttolerant
  - Mindestrate
- Andere Einsatzgebiete
  - DNS
- Zuverlässiger Datentransfer über UDP
  - Zuverlässigkeit auf der Anwendungsschicht implementieren
  - Anwendungsspezifische Fehlerkorrektur

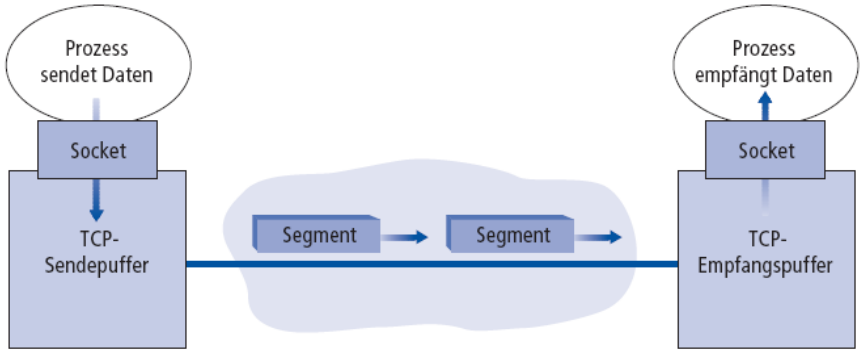
Länge (in Byte) des UPD-Segments, inklusive Header



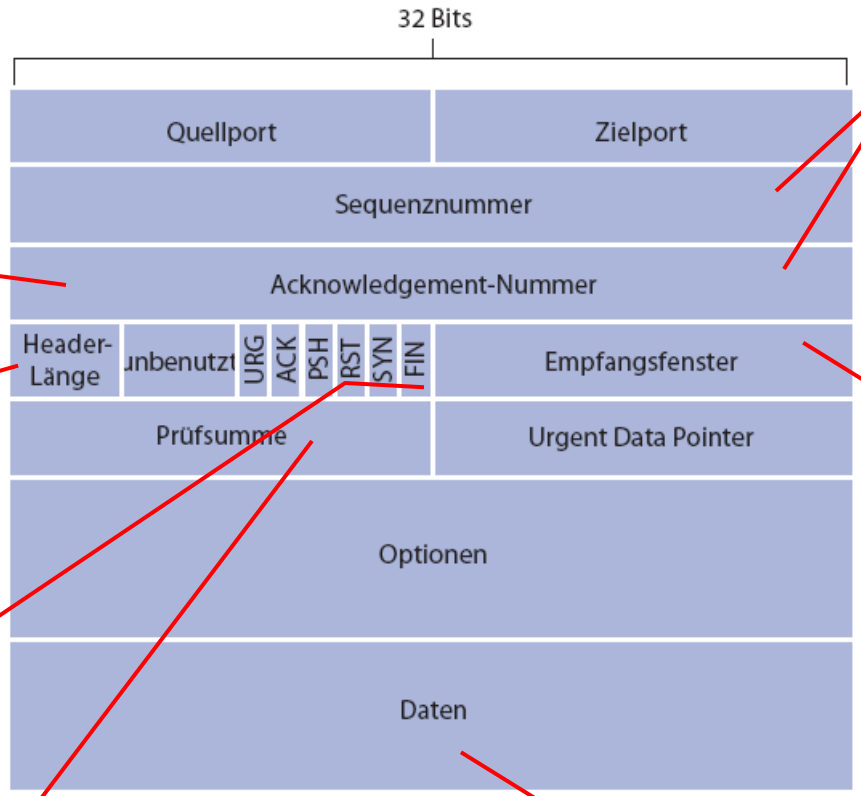
UDP-Segment-Format

TCP - Überblick

- Punkt-zu-Punkt
  - Ein Sender, ein Empfänger
- Zuverlässiger, reihenfolgeerhaltender Byte-Strom
  - Keine “Nachrichtengrenzen”
- Sender- & Empfängerfenster
- Vollduplex
- Verbindungsorientiert
  - Handshaking
  - Austausch von Kontrollnachrichten initialisiert den Zustand im Sender und Empfänger, bevor Daten ausgetauscht werden
- Flusskontrolle und Überlastkontrolle
  - Sender überfordert den Empfänger bzw. das Netzwerk nicht



TCP-Segmentaufbau



Zählen Bytes und nicht Segmente!

ACK: ACK-Nummer gültig

Header-Länge in 32-Bit-Blöcken

RST, SYN, FIN: Flags für Verbindungsauf- und -abbau

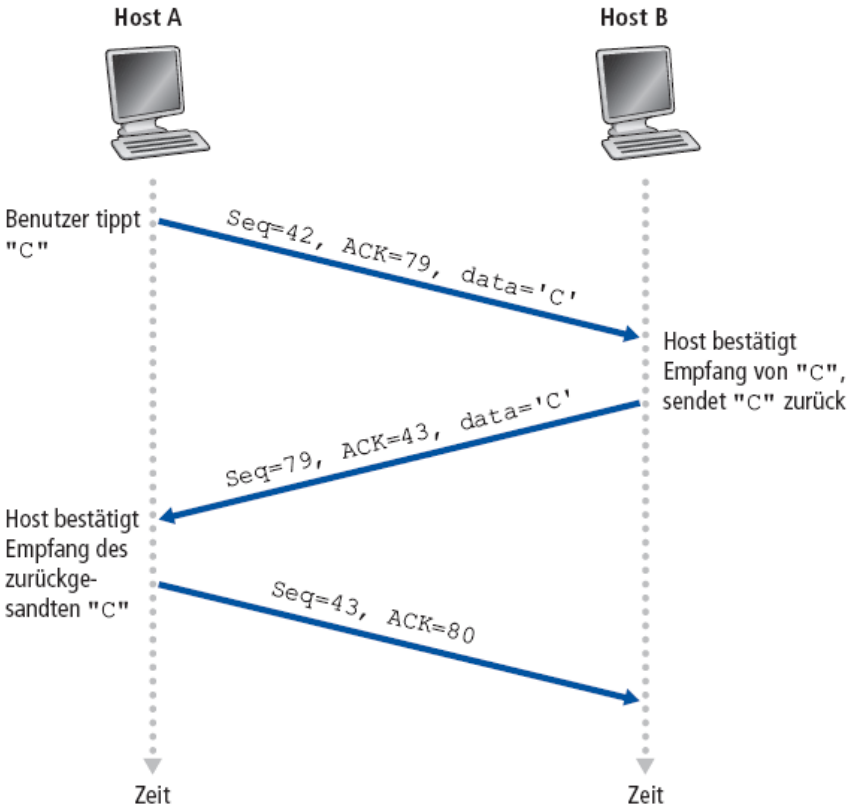
Anzahl an Bytes, die der Emfpänger bereit ist, entgegenzu-nehmen

Daten (auch als Nutzdaten bezeichnet)

Prüfsumme (wie bei UDP)

TCP-Sequenznummern und -ACKs

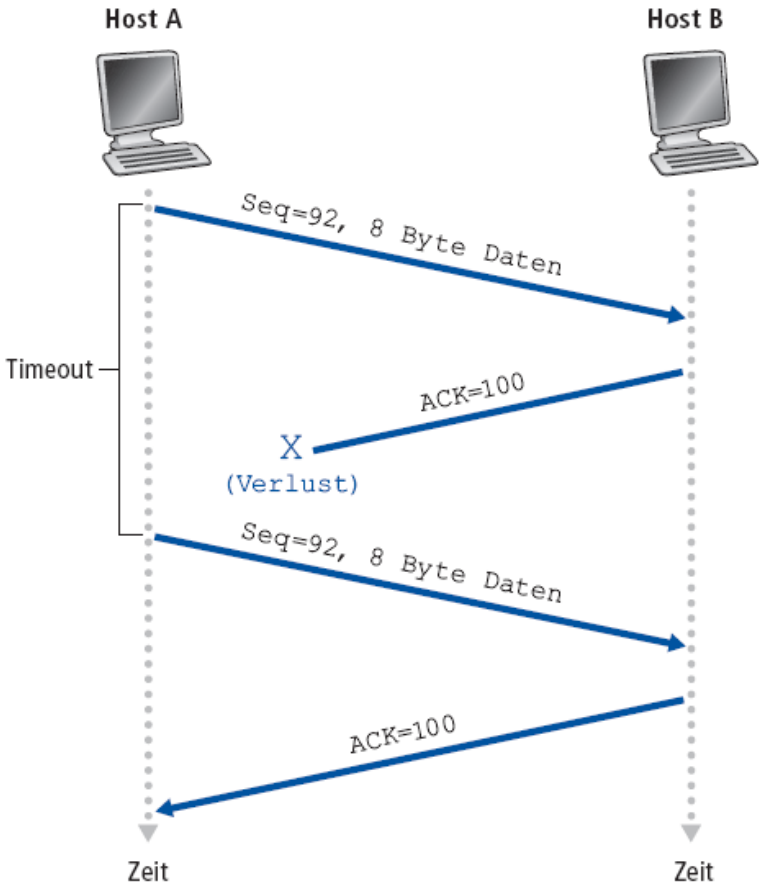
- Sequenznummern
  - Nummer des ersten Byte im Datenteil
- ACKs
  - Sequenznummer des nächsten Byte, das von der Gegenseite erwartet wird
- Wie werden Segmente behandelt, die außer der Reihe ankommen?
  - Wird von der TCP-Spezifikation nicht vorgeschrieben!
  - Bestimmt durch die Implementierung!

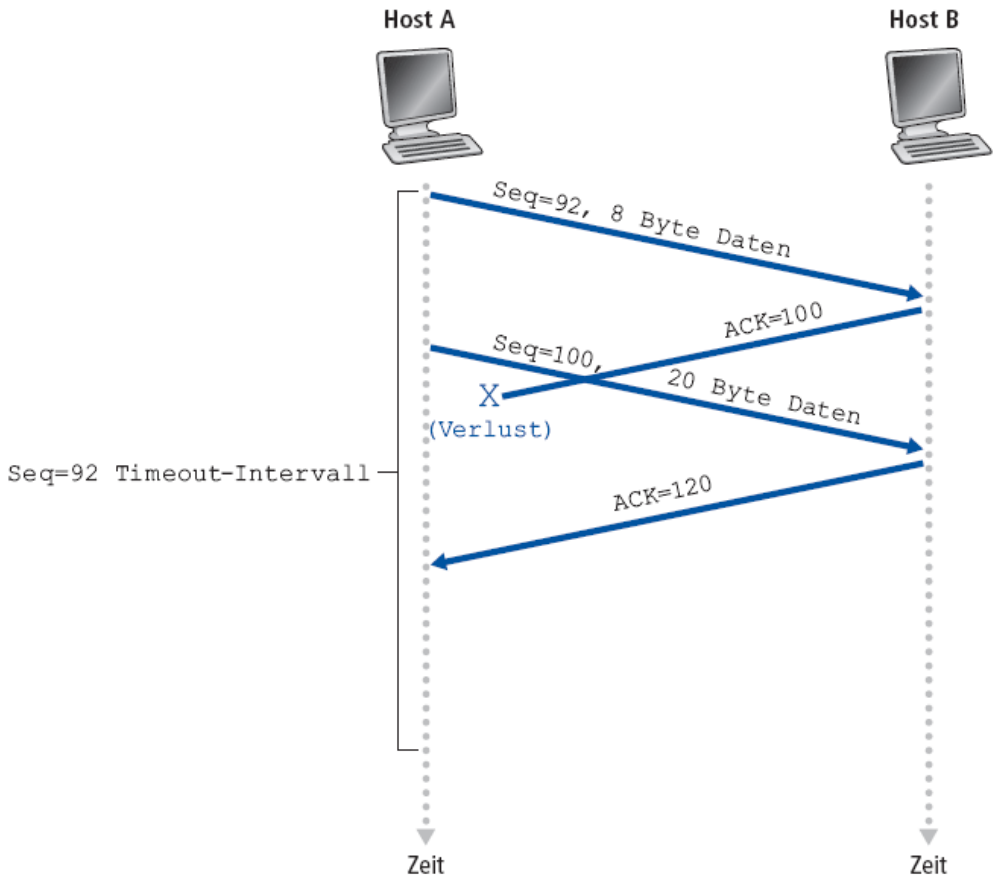


TCP – zuverlässiger Datentransfer

- TCP stellt einen zuverlässigen Datentransfer über den unzuverlässigen Datentransfer von IP zur Verfügung
- Pipelining von Segmenten
- TCP verwendet einen einzigen Timer für Übertragungswiederholungen
  - Abschätzen der RTT
- Übertragungswiederholungen werden ausgelöst durch
  - Timeout
  - Doppelte ACKs

Übertragungswiederholung – Beispiele

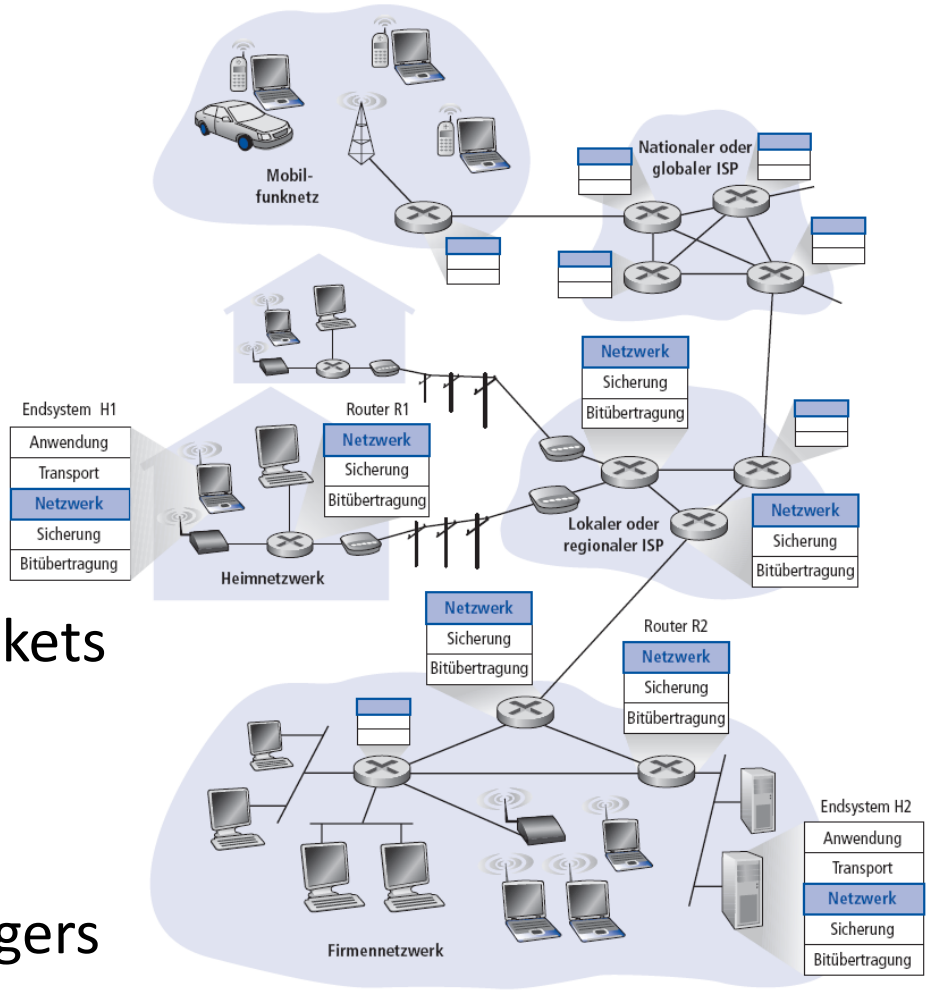




NETZWERKSCHICHT

Netzwerkschicht

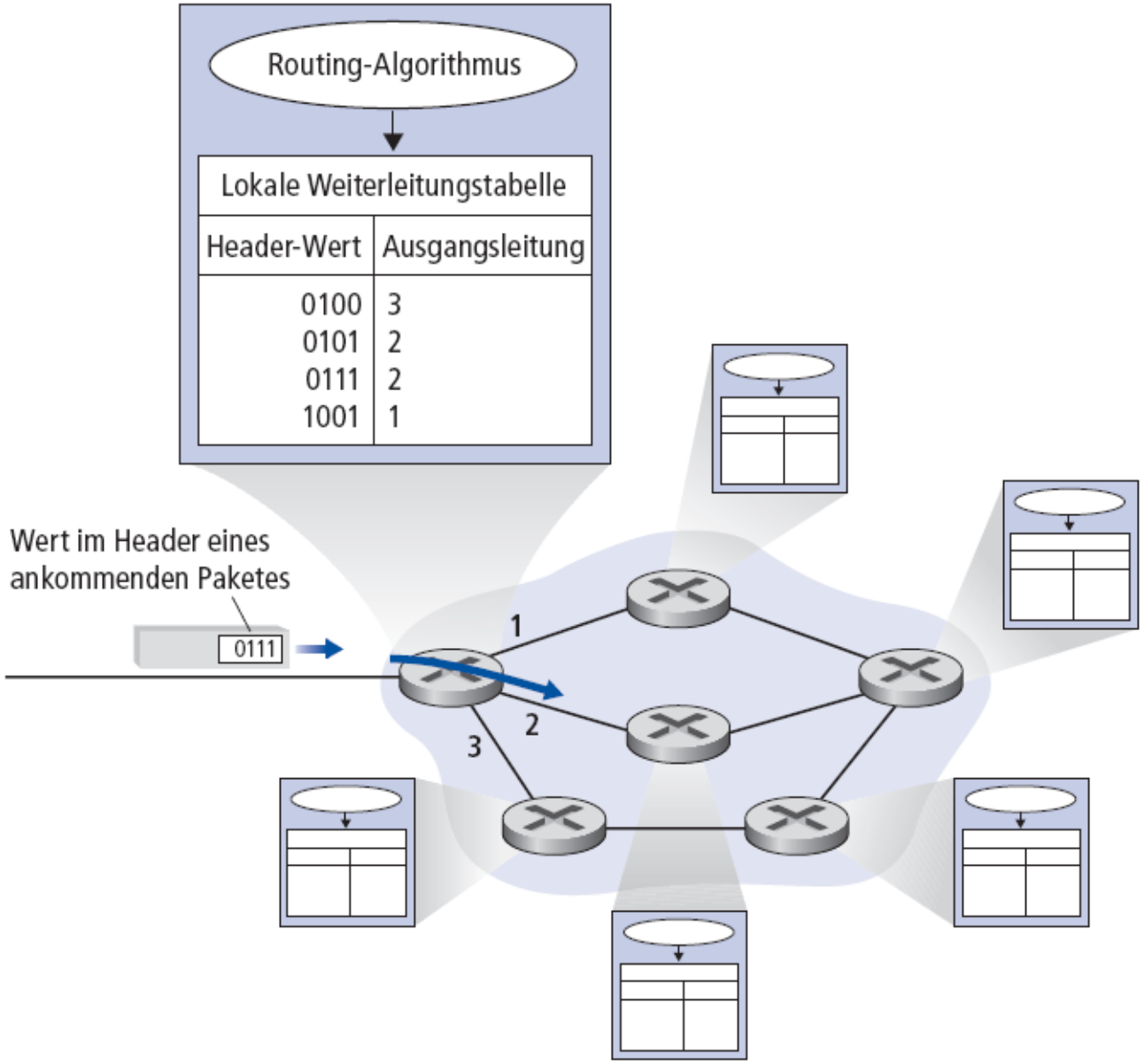
- Auch
  - Vermittlungsschicht
  - Network Layer
- Daten von der nächsthöheren Schicht (Transportschicht) des Senders entgegennehmen
- In Datagramme verpacken
- Durch das Netzwerk leiten
- Auspacken des Vermittlungspakets beim Empfänger
- Ausliefern der Daten an die nächsthöhere Schicht (Transportschicht) des Empfängers
- Netzwerkschicht existiert in jedem Host und Router!



Funktionen der Netzwerkschicht

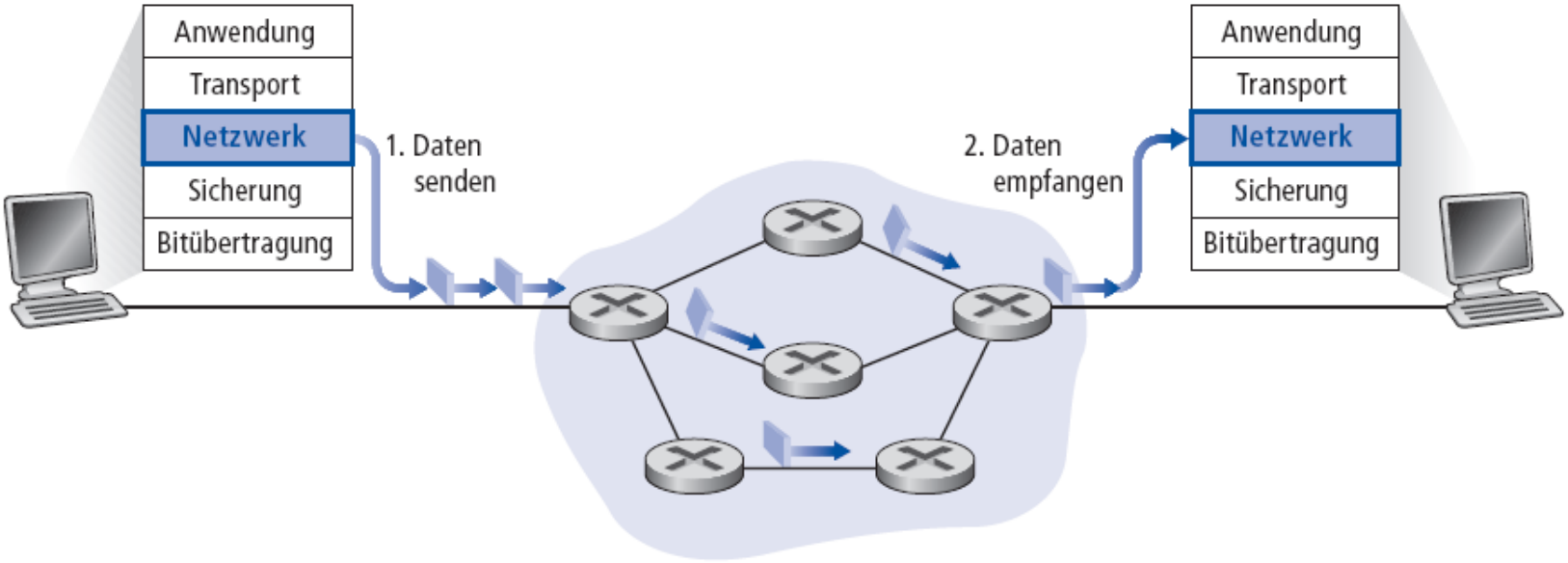
- Weiterleiten von Paketen (Forwarding)
  - Router nimmt Paket auf einer Eingangsleitung entgegen
  - Router bestimmt die Ausgangsleitung anhand lokaler Informationen (z.B. Routing-Tabelle)
  - Router legt das Paket auf die Ausgangsleitung
- Wegewahl (Routing)
  - Router kommunizieren miteinander, um geeignete Wege durch das Netzwerk zu bestimmen
  - Als Ergebnis erhalten sie Informationen, wie welches System im Netzwerk zu erreichen ist (z.B. wird eine Routingtabelle mit Einträgen gefüllt)
- Metapher
  - Routing = Planen einer Strecke für eine Autofahrt
  - Weiterleitung = Verhalten an einer Autobahnkreuzung

Weiterleiten und Wegewahl



Datagrammnetzwerke

- Kein Verbindungsaufbau auf der Netzwerkschicht!
- Router halten keinen Zustand für Ende-zu-Ende-Verbindungen
  - Auf Netzwerkebene gibt es nicht das Konzept einer “Verbindung”
- Pakete werden unter Verwendung einer Zieladresse weitergeleitet
  - Pakete für dasselbe Sender-Empfänger-Paar können unterschiedliche Pfade nehmen



Weiterleitungstabelle

Zieladressbereich	Schnittstelle
11001000 00010111 00010000 00000000	
bis	0
11001000 00010111 00010111 11111111	
11001000 00010111 00011000 00000000	
bis	1
11001000 00010111 00011000 11111111	
11001000 00010111 00011001 00000000	
bis	2
11001000 00010111 00011111 11111111	
sonst	3

Longest Prefix Matching

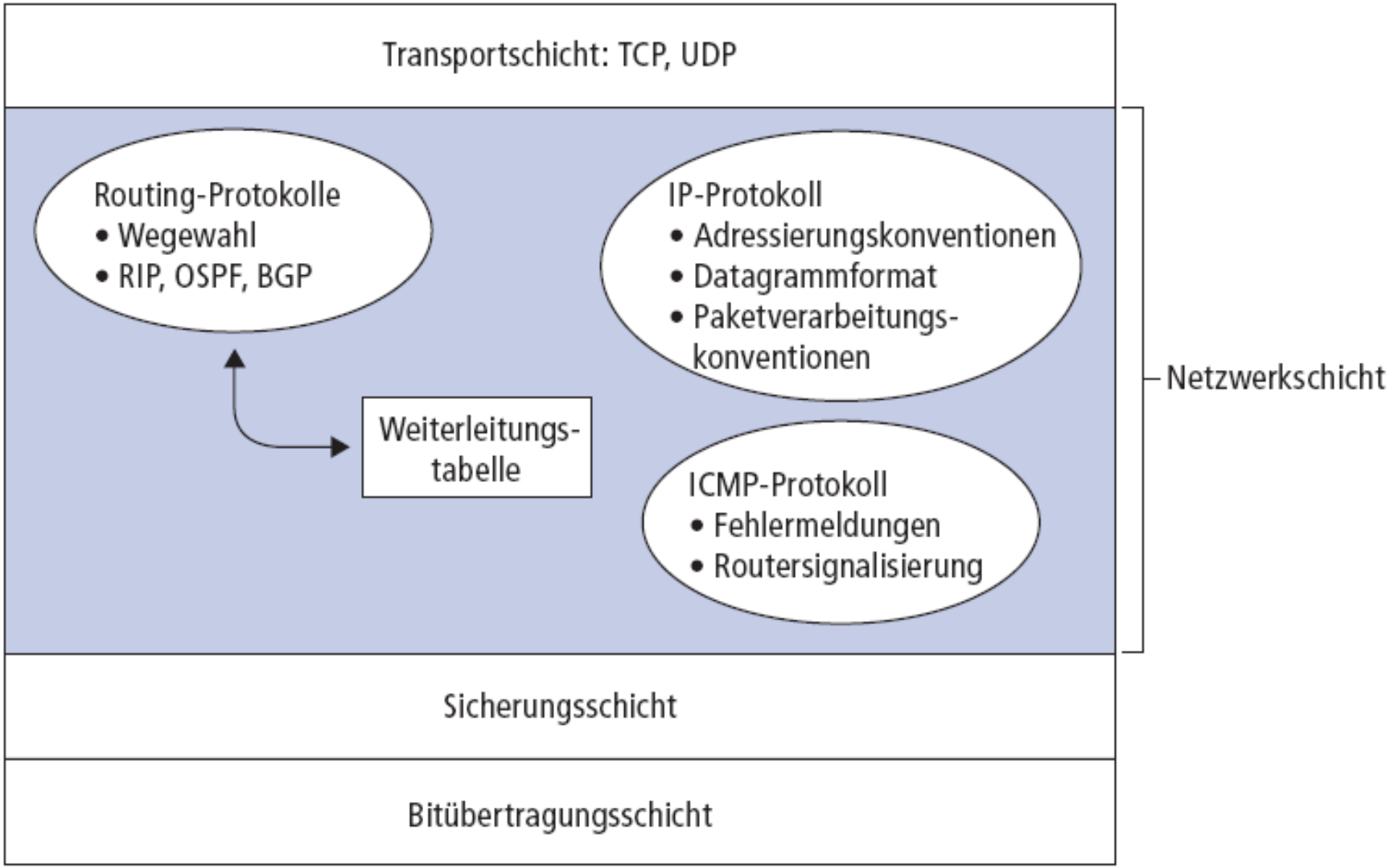
Passender Präfix	Schnittstelle
11001000 00010111 00010	0
11001000 0 0010111 00011000	1
11001000 00010111 00011	2
sonst	3

Beispiele

Adresse: 11001000 00010111 00010110 10100001 Welche Schnittstelle? 0!

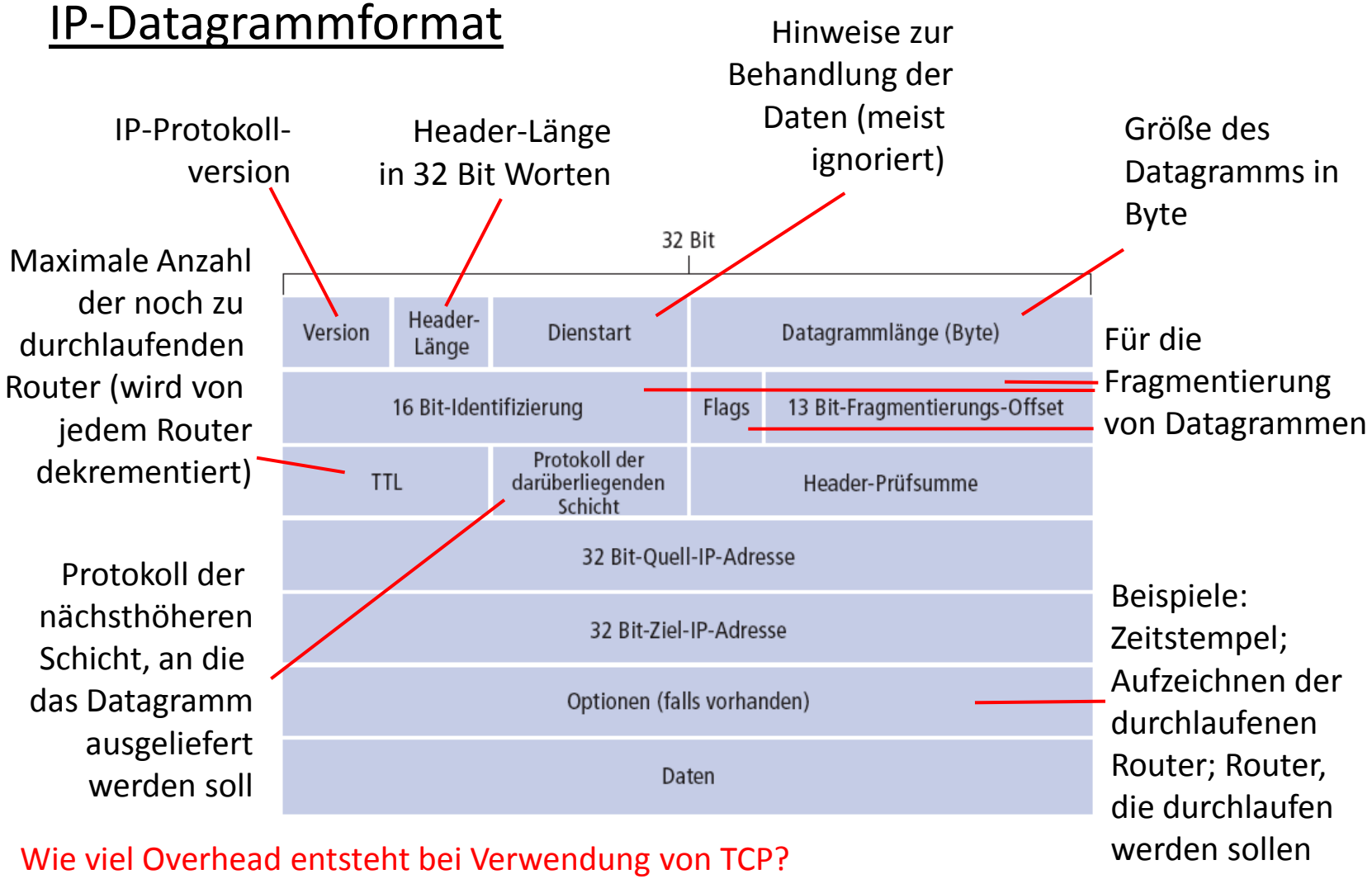
Adresse: 11001000 00010111 00011000 10101010 Welche Schnittstelle? 1!

Die Netzwerkschicht des Internets



IP-Datagrammformat

IP-Datagrammformat

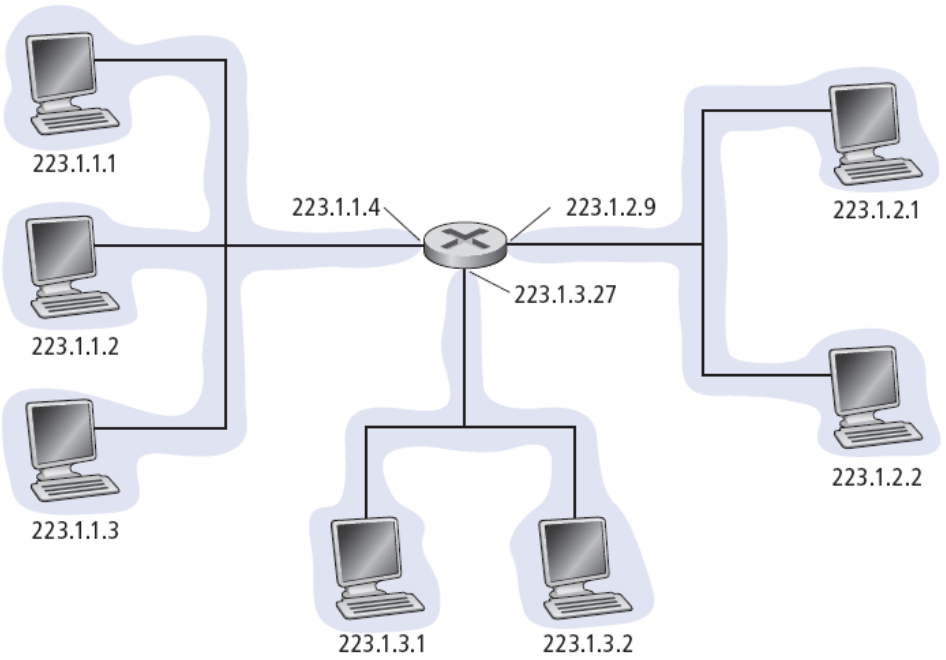


Wie viel Overhead entsteht bei Verwendung von TCP?

- 20 Byte für den TCP-Header, 20 Byte für den IP-Header
- = 40 Byte + Overhead auf der Sicherungsschicht und vielleicht Anwendungsschicht

IP-Adressierung – Grundlagen

- IP-Adresse
  - 32-Bit-Kennung für das Interface (Schnittstelle) eines Endsystems oder eines Routers
- Interface = Verbindung zwischen dem System und dem Link
  - Wird normalerweise durch eine Netzwerkkarte bereitgestellt
  - Router haben typischerweise mehrere Interfaces
  - Endsysteme können ebenfalls mehrere Interfaces haben
  - Jedes Interface besitzt eine IP-Adresse

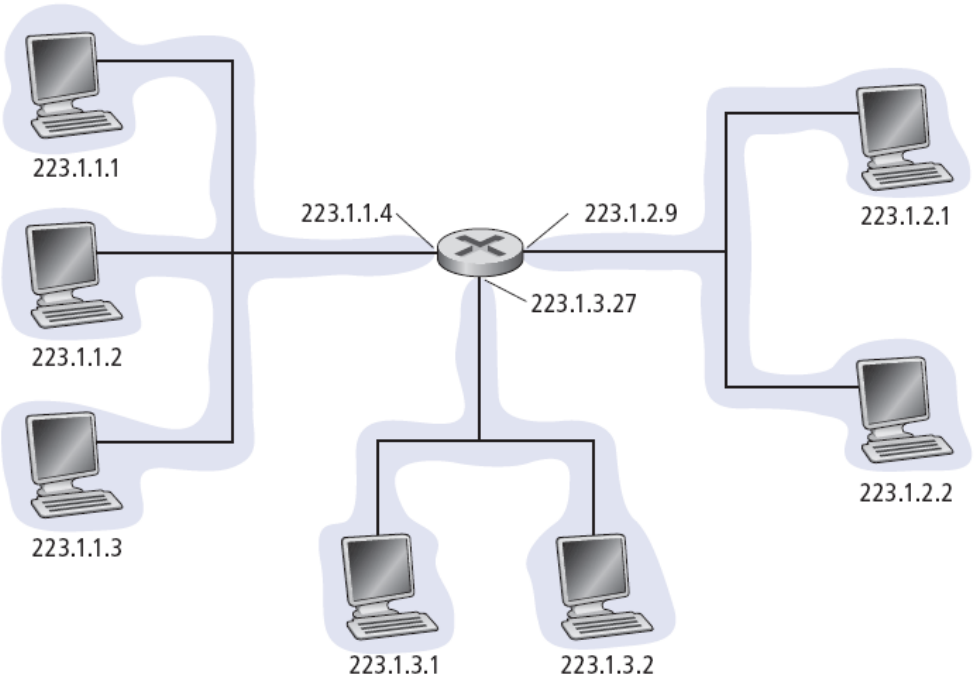


223.1.1.1 = 11011111 00000001 00000001 00000001

223 1 1 1

IP-Adressierung – Netzwerke

- IP-Adresse hat zwei Bestandteile
  - netid: Die oberen Bits der Adresse, identifiziert ein Netzwerk
  - hostid: Die unteren Bits der Adresse, identifiziert ein Interface eines Systems
- Was ist ein (Sub-)Netzwerk?
  - Alle Interfaces mit derselben netid formen ein Netzwerk
  - Alle Interfaces eines Netzwerkes können sich direkt (ohne einen Router zu durchqueren) erreichen



Drei IP-Netzwerke, die mit einem Router verbunden sind
Die netid steht hier in den oberen 24 Bit

Subnetzmaske: /24 oder 255.255.255.0

Subnetze

- Früher wurden IP-Adressen in Adressklassen aufgeteilt
  - Die Klasse bestimmte das Verhältnis der Längen netid/hostid
  - Kompliziertes Vorgehen bei kleineren Subnetzen (subnetid)
- Heute klassenlose Adressierung
  - Die Aufteilung nach netid/hostid wird immer explizit per Subnetzmaske durchgeführt
  - Keine explizite Unterscheidung zwischen netid und subnetid
- Schreibweise
  - a.b.c.d/x, wobei x die Länge der netid (hier auch Präfix oder englisch prefix) bestimmt
    - Alternative Schreibweise zur Subnetzmaske
  - Beispiel:
    - 192.48.96.0/23
    - 11000000.00110000.01100000.00000000 (netidhostid)

Subnetze – Beispiele

- Beispiel: 192.168.1.16 Subnetzmaske 255.255.255.0
- Erste IP (Netzwerkadresse) = 192.168.1.0 (IP AND Subnetzmaske)

	1	1	0	0	0	0	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0
&	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
=	1	1	0	0	0	0	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

- Letzter Teil (Hostnummer) = 16 (IP AND NOT Subnetzmaske)

	1	1	0	0	0	0	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0
&	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
=	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

- Letzte IP (Broadcast) = 192.168.1.255 (IP OR NOT Subnetzmaske)

	1	1	0	0	0	0	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0
OR	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
=	1	1	0	0	0	0	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1

Weitere Beispiele

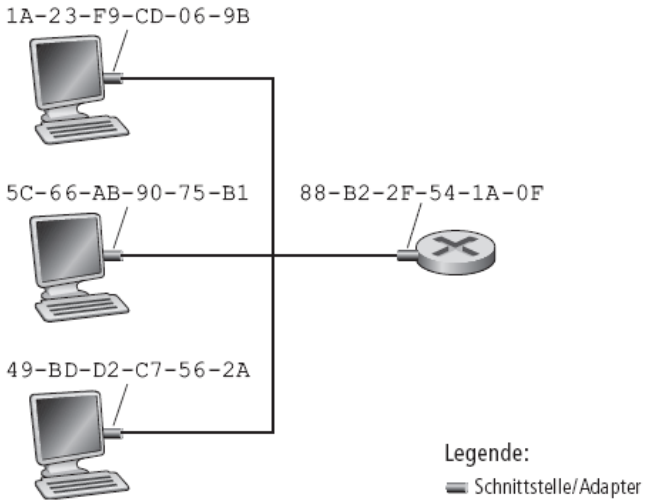
- Netzwerk 192.168.4.0/22
  - Netzmaske: 255.255.252.0 = 22 11111111.11111111.11111100.00000000
  - Netzwerkadresse: 192.168.4.0 11000000.10101000.00000100.00000000
  - Broadcast: 192.168.7.255 11000000.10101000.00000111.11111111
  - Hosts/Netzwerk: 1022 (ohne Netzwerk- und Broadcastadresse)
- 193.43.55.0/24 soll in zwei gleich große Subnetze unterteilt werden
  - Netzmaske: 255.255.255.128 = /25
  - Subnetz 1
    - Netzwerkadresse: 193.43.55.0 und Broadcastadresse: 193.43.55.127
  - Subnetz 2
    - Netzwerkadresse: 193.43.55.128 und Broadcastadresse: 193.43.55.255
  - Hosts/Netzwerk bei beiden Subnetzen: 126

Adressen – weitere Aspekte

- Wie bekommt ein Host seine IP-Adresse?
- Durch manuelle Konfiguration!
  - IP-Adresse
  - Subnetzmaske
  - Weitere Parameter
- DHCP (Dynamic Host Configuration Protocol)
  - Dynamisches Beziehen der Adresse von einem Server
  - “Plug-and-Play”

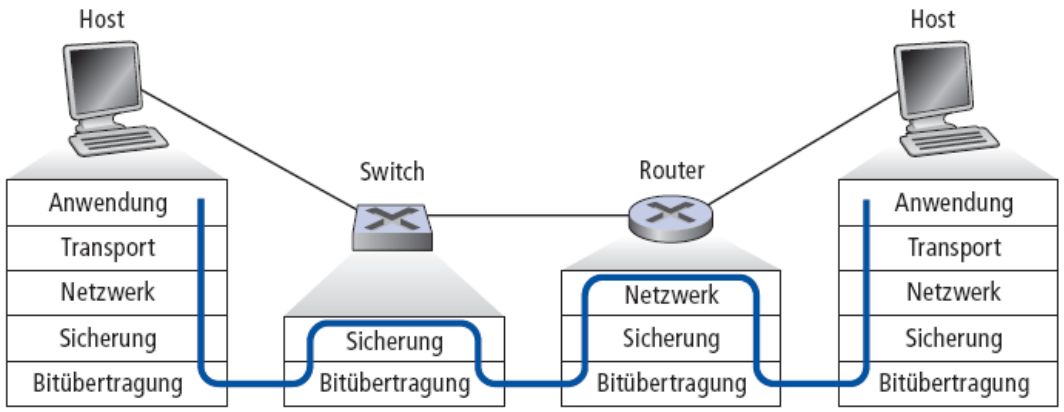
MAC-Adresse

- Neben einer IP-Adresse hat ein Host auch eine MAC-Adresse
- 48 Bit lange Adresse auf der Sicherungsschicht
- Wird verwendet, um einen Rahmen von einem Host zu einem benachbarten Host weiterzuleiten (im selben Netzwerk!)
  - Keine Ortsinformationen, muss nur im gegebenen Netzwerk eindeutig sein!
- Beispiel



Router und Switch

- Zwei wichtige Aufgaben eines Routers
  - Ausführen von Routing-Algorithmen und -Protokollen
    - RIP, OSPF (Dijkstra-Algorithmus für kürzeste Pfade), BGP
  - Weiterleiten von Datagrammen von einem eingehenden zu einem ausgehenden Link (basierend auf Weiterleitungstabelle)
- Vergleich Switch und Router
  - Beide speichern Pakete und leiten diese weiter
    - Router: Auf der Netzwerkebene (verwendet IP-Adressen)
    - Switch: Gehört zur Sicherungsschicht (verwendet MAC-Adressen)
  - Switch verwaltet eine Switch-Tabelle und ist selbst lernend

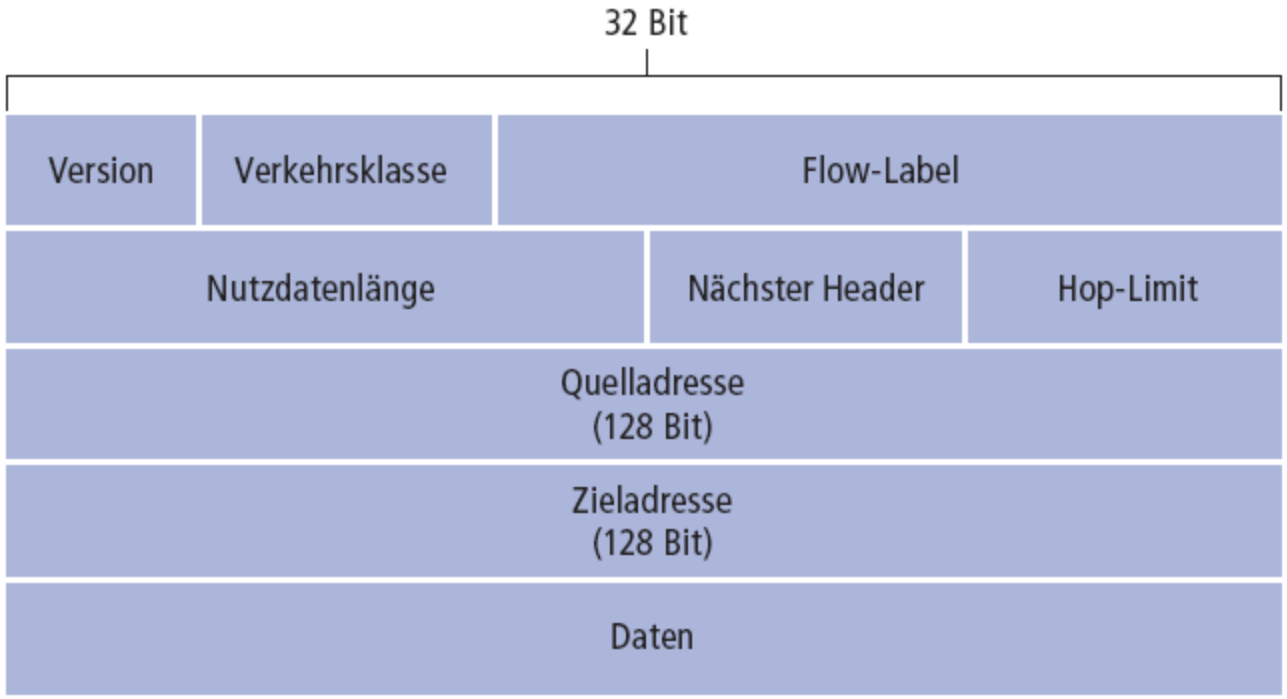


IPv6

- Ursprüngliche Motivation
  - Es gibt zu wenig 32-Bit-Adressen
  - Beispiel: Was passiert, wenn jedes Mobiltelefon eine feste IP-Adresse bekommen soll?
- Weitere Motivation
  - Vereinfachtes Header-Format für eine schnellere Verarbeitung in den Routern
  - Header soll Dienstgütemechanismen (Quality of Service, QoS) unterstützen
- IPv6-Datagrammformat
  - Header fester Länge (40 Byte)
  - Keine Fragmentierung in den Routern

IPv6-Header

- Verkehrsklasse: Priorisierung von Datagrammen
- Flow Label: Identifikation von zusammengehörigen Flüssen von Datagrammen (z.B. ein Voice-over-IP-Telefonat)
- Nächster Header: An welches Protokoll sollen die Daten im Datenteil übergeben werden? Beispiel: TCP



ZUSAMMENFASSUNG

Zusammenfassung

- Protokolle
- Pakete
- Transportprotokolle
- Netzwerkprotokolle
- IP-Adressen
- Subnetze

Literatur

- Grundliteratur
  - J. Kurose, K. Ross: **Computernetzwerke: Der Top-Down-Ansatz**, 6. Auflage, Pearson Studium, 2014