

1. Geschichte
2. Schichten
3. Netzzugang
4. Internet
5. Transport
6. Anwendung
7. DNS
8. eMail

## V. Internet

- Internet: globales extrem komplexe Kommunikationssystem
- Schlüssel zur Beherrschung: Abstraktion
  - Werkzeug: ISO/OSI 7 Schichtenmodell
  - **I**nternational **O**rganization for **S**tandardization 1983:  
**O**pen **S**ystems **I**nterconnection Reference Model
  - jede Schicht sichert genau definierte Aufgaben
- vielfältigste Technologien mit unterschiedlichsten Merkmalen
- viele Organisationen und Unternehmen: viele Technologien, Standards
  - **nicht** alle Standards **kompatibel**
  - Produkte von Herstellern entsprechen nicht vollständig Standards
  - es gibt keine einheitliche Grundlagentheorie
  - es gibt keine einheitliche Terminologie
- **Zusammenschalten von Netzen mit den unterschiedlichen Technologien**

1. **Geschichte**
2. Schichten
3. Netzzugang
4. Internet
5. Transport
6. Anwendung
7. DNS
8. eMail

## V.1 Internet: Geschichte

- „Schuld“ an dem Internet war der UdSSR Sputnik 1957
- ARPA: Advanced Research Projects Agency
  - Mangel an leistungsfähigen Computern
  - statt Neukauf - Untersuchung von Datennetzwerken – Vernetzungsprojekt
- **ARPANET ab 1962**
  - im Auftrag der US-Luftwaffe unter Leitung des MIT
- kein primär militärischer Einsatz, Verbindung von Universitäten
- **29.10.1969: 1. Datenpaket** von LA nach Stanford (San Francisco)
  - totaler Systemabsturz bei g von LOGIN
  - Haupt-Knoten hatte 12 KB Speicher, Verbindung 50 kbps
- **basierte auf Telefonleitungen**
- 1971: 23 Unis, im gleichen Jahr erste eMail

1. **Geschichte**
2. Schichten
3. Netzzugang
4. Internet
5. Transport
6. Anwendung
7. DNS
8. eMail

## V.1 Internet: Geschichte

- ab 1973 London und Norwegen
- **1974 Geburtsstunde des heutigen Internet:**  
Diplomarbeit von Vinton Cerf und Robert Kahn mit  
Vorschlag zur Standardisierung des Datenverkehrs im Netz
- **1982 TCP und IP Standard** des US-Verteidigungsministeriums
- 1989 Anschluss von Deutschland
- Tim Berners-Lee am CERN Hypertext-Prinzip
  - erstes Anzeige Programm **1989 WorldWideWeb**, 1993 Mosaic
- **ab 1991 Lockerung der Restriktionen für kommerziellen Nutzung**
  - bis 1987 ausschließlich wissenschaftliche Nutzung

1. Geschichte
2. **Schichten**
3. Netzzugang
4. Internet
5. Transport
6. Anwendung
7. DNS
8. eMail

## V.2 Schichtenmodell

- Schlüssel für Beherrschung der Komplexität: TCP/IP-Schichtenmodell
  - einfache, mächtige Schnittstellen notwendig
  - keine direkte Kommunikation zwischen NW-HW und Anwendungen
  - alle Beteiligten: Regeln für Kommunikation
- Zerlegung in Teilaufgaben: spezifische Protokolle
  - lösen ein bestimmtes Problem – gemeinsame Datenstrukturen, Infos
  - Gesamtkonzept: Protokollreihen (Suites) oder Familien
- zu behandelnde NW-Problem
  - Übertragungsprobleme (CRC), Reihenfolge (doppelte, verzögerte, verlorene Pakete), Flusskontrolle (Überlastung, Stau, Datenüberlauf)
  - Adressierung und Weiterleitung
  - Zugang zur Anwendung

1. Geschichte
2. **Schichten**
3. Netzzugang
4. Internet
5. Transport
6. Anwendung
7. DNS
8. eMail

## V.2 Schichtenmodell

### TCP/IP- Schichtenmodell

**1. Netzzugang:** **physikalische Kodierung** von Daten

Daten in **logischen Paketen**: Rahmen, Fehlerkorrektur

**2. Vermittlung:** Adressierung,

Weiterleitung **Sende → Empfangs-Knoten**

**3. Transport:** zuverlässige Übertragung **zwischen Anwendungen**

**4. Verarbeitung:** Anwendung

- jede Schicht unabhängig

- jede Umwandlung, die ein Protokoll vor dem Versenden auf einen Rahmen anwendet, muss beim Empfang des Rahmens **vollständig** umgekehrt werden.

- Zusatzinfos an Paket anhängen



1. Geschichte
2. Schichten
3. **Netzzugang**
4. Internet
5. Transport
6. Anwendung
7. DNS
8. eMail

- **LAN**
- Abstraktion
- NIC
- WAN
- WAN Standards
- ISDN
- DSL
- Alternativen
- WAN Technologien
- WAN Routing

## V.3.1 Netzzugangs Schicht: LAN

- Lokale Netzwerke: Zugriff auf gemeinsames Medium
  - sehr effektiv, da geringe Kosten
  - Aufteilung der Daten in kleine Blöcke: Pakete
- **Paketkonzept: grundlegendes Konzept für Vernetzung**
  - **gemeinsame Ressource**: Koordination von Sender und Empfänger
  - **gleichberechtigter Zugang**: kleine Zugriffszeit
- Lokalisierungsprinzip: grundlegendes Prinzip in Informatik
  - 1. Zeitliches Lokalisierungsprinzip**
    - 2 Rechner immer wieder miteinander**
  - 2. Örtliches Lokalisierungsprinzip**
    - benachbarte Computer häufiger als entfernte**
- Realisierung von Paketen in technologieabhängigen Rahmen

1. Geschichte
2. Schichten
3. Netzzugang
4. Internet
5. Transport
6. Anwendung
7. DNS
8. eMail

- LAN
- Abstraktion
- NIC
- WAN
- WAN Standards
- ISDN
- DSL
- Alternativen
- WAN Technologien
- WAN Routing

## V.3.1 Netzzugangs Schicht: Abstraktion

- Abstraktion Netzzugangs Schicht
  - auf dem Draht Signal: Änderung von physikalischen Größen: Symbole
  - Nyquist-Theorem:  $D_{\max} = 2 * B * \log_2 K$ 
    - B: **Bandbreite** in Hz = 1/s
    - K: Anzahl der **Symbole** (Anzahl der Zustände, Daten kodieren)
    - D: max. **Datenrate** in Bit/s
  - logische Größen: Zuordnung von Symbolen
  - Gruppierung von logischen Größe zu Daten (Zeichen) durch HW-Rahmen
  - Zusammenfassen von Zeichenblöcken zu Paketen
  - Realisierung von Paketen in technologiespezifischen Rahmen (Frames)
    - Start of Header (soh), End of Transmission (eot)
    - Sicherung der Rahmen durch Prüfsumme, CRC

1. Geschichte
2. Schichten
3. Netzzugang
4. Internet
5. Transport
6. Anwendung
7. DNS
8. eMail

- LAN
- Abstraktion
- NIC
- WAN
- WAN Standards
- ISDN
- DSL
- Alternativen
- WAN Technologien
- WAN Routing

## V.3.1 Netzzugangs Schicht: NIC

- Netzwerkschnittstellenkarte (Leiterkarte, Schaltkreis)
  - Entkoppelung von Netzwerk und Rechner
  - NIC – digitale Schaltung, LAN – analoge Signale der Technologie
- Bedienung des Netzanschlusses: technologieabhängig
  - Senden, Empfangen von Rahmen unabhängig von CPU
  - Entscheidung über Annahme, Ablehnung
  - Erzeugung, Entpacken der Rahmen
  - Vorverarbeitung: CRC, Präambel
- Ein-, Ausgabegerät des BS gesteuert durch Treiber
  - Verbindung zum Rechner: Bus, DMA
- Filterung HW-Adressen (MAC): Uni-, Multi- Broadcast
- Anschluss an LAN: Medientypen, z.B. Ethernet TP, Glasfaser, Geschwindigkeiten, Kodierung (Manchester, 4B5B-Code,...)



1. Geschichte
2. Schichten
3. Netzzugang
4. Internet
5. Transport
6. Anwendung
7. DNS
8. eMail

- LAN
- Abstraktion
- NIC
- WAN
- WAN Standards
- ISDN
- DSL
- Alternativen
- WAN Technologien
- WAN Routing

## V.3.1 Netzzugangs Schicht: Erweiterung

- Verkabelung abhängig von Technologie
  - Topologie: Bus, Stern, Ring
  - Größe der Segmente: Signallaufzeit, phys. Eigenschaften, Koordination
  - Anpassung an verschiedene Medientypen: el., opt., drahtlos
- Erweiterung: Verbindung mehrerer Netzwerk-Segmente
  - Repeater: Verstärker
  - Hub: Multiport-Repeater, simuliert gemeinsames Medium
  - Bridge: 2 NW-Segmente verschiedener Technologie, Rahmenfilterung
  - Switch: Vermittlung, simuliert ein Segment für jeden Knoten
    - Multiport-Bridge für eine Technologie
- Kollisionsdomäne: gemeinsames NW-Segment, Zugriff koordiniert

1. Geschichte
2. Schichten
3. **Netzzugang**
4. Internet
5. Transport
6. Anwendung
7. DNS
8. eMail

- LAN
- Abstraktion
- NIC
- **WAN**
- WAN Standards
- ISDN
- DSL
- Alternativen
- WAN Technologien
- WAN Routing

## V.3.2 Netzzugangs Schicht: WAN

- **LAN Einschränkung:** Größe (Anzahl), Reichweite (Standorte)
- **Telefonnetz:** weltweit flächendeckend, für Sprachübertragung
  - analoge → digitale Telephonie → Datenverkehr
  - **synchrones** Netz, konstante Datenrate
  - Verzögerung digitales Sprachsignal: starke Verzerrung
- **Datennetz asynchron**, Verzögerung kann aufgeholt werden
- **Standards für Telefonnetz - unabhängig von Computer-Standards**
- 1. **Teilnehmeranschlüsse für hohe Geschwindigkeiten**
  - Teilnehmeranschlüsse (lokal loop)
- 2. **digitale Punkt zu Punkt Verbindungen für große Entfernungen**
- **WAN: große paketvermittelte Netzwerksysteme**
  - **HW-Einheit: Speichervermittlung, kein gemeinsames Medium**

1. Geschichte
2. Schichten
3. **Netzzugang**
4. Internet
5. Transport
6. Anwendung
7. DNS
8. eMail

- LAN
- Abstraktion
- NIC
- WAN
- **WAN Standards**
- ISDN
- DSL
- Alternativen
- WAN Technologien
- WAN Routing

## V.3.2 Netzzugangs Schicht: WAN - Standards

- **T(E,J)-Standards: Kapazität des Trägersystem**
  - bekanntester Standard: T1 (1968 USA), **Europa E1: 2 Mbit/s (Standleitung)**
- **DS-Standards: effektive Datenrate** (Digital Signal Level Standard)
  - Synchronous Transport System: digitales Leitungs-IF
  - STS-1: 51,84 Mbit/s → 810 Sprachleitungen
  - **STS-3072: 160,0 Gbit/s → 2,5 10<sup>6</sup> Sprachleitungen**
- Standards für **digitale Multiplex-Technologien**
  - **SONET** (USA): Synchronous Optical Net-work
  - **SDH** (Europa): Synchronous Digital Hierarchy
- **Local Loop: Netzanbieter – Hausanschluss**
  - POTS: Plain Old Telephone Service (analog)

	Mbit/s	#
DS0	0,064	1
T1	1,544	24
T3	44,736	672 (28 T1)
<b>E1</b>	<b>2,048</b>	<b>30</b>
E3	34,368	480 (16 E1)

1. Geschichte
2. Schichten
3. **Netzzugang**
4. Internet
5. Transport
6. Anwendung
7. DNS
8. eMail

- LAN
- Abstraktion
- NIC
- WAN
- WAN Standards
- **ISDN**
- DSL
- Alternativen
- WAN Technologien
- WAN Routing

## V.3.2 Netzzugangs Schicht: WAN - ISDN

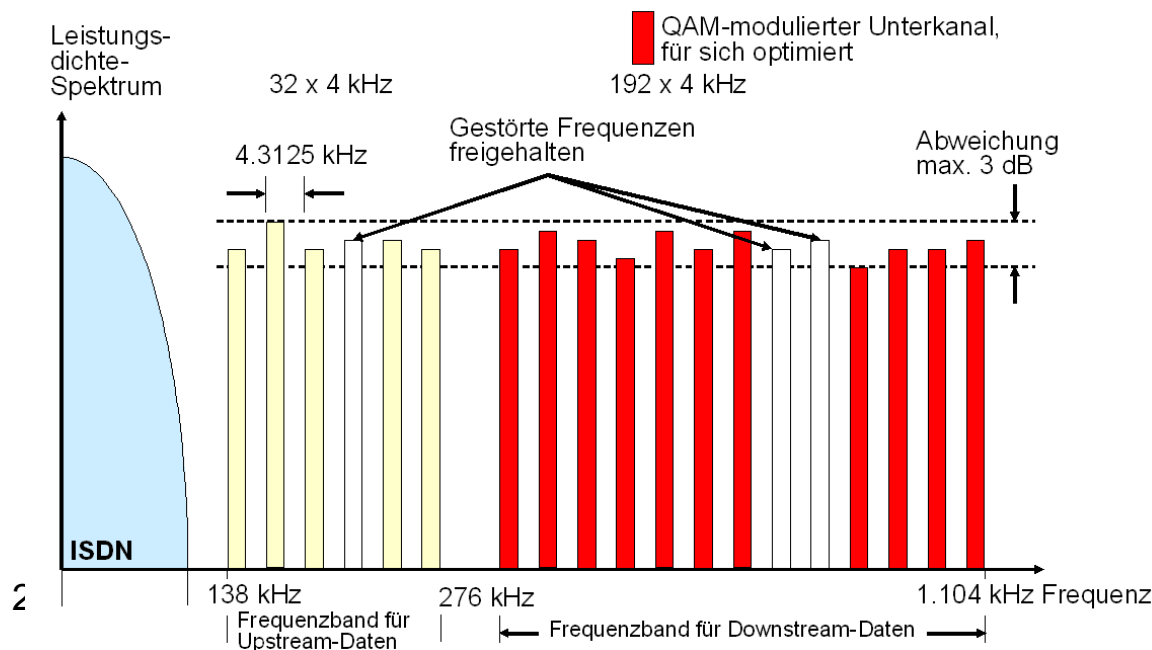
- Integrated Services Digital Network
- Integration Sprach- und Datendienste (leitungsvermittelt)
  - Telefongesellschaften der Welt: veraltet bei Verabschiedung (Euro-ISDN 12/1993)
- **S<sub>0</sub>-Basisanschluss**: 3 Kanäle Zeitmultiplex, Synchronisation
  - 2x B: 64 kbit/s bearer channel – Träger- (Nutz-) kanal
  - 1x D: 16 kbit/s data channel – Signalisierung, Steuerung
  - Rahmen 48 Bit je 250 µs: B1 2x 8b; B2 2x 8b; D 4x 1b
- **Primärmultiplexanschluss (E1)**: Brutto 2.048 kbit/s
  - 30 B-Kanäle, 1 D-Kanal 64 kbit/s, 1 Synchronisation 64 kbit/s
- **ATM: Asynchronous Transfer Mode**
  - Schlüsseltechnologie für Breitband-ISDN
  - LAN und WAN: synchrone 53 Byte Pakete (Slots) fester Länge – Zellen
  - **Breitband WissenschaftsNetz**: 34/155 Mbps (Jena 1996 34 Mbps)
  - **Universaldienst für Daten und Sprache**: abnehmende Akzeptanz (**gescheitert**)

1. Geschichte
2. Schichten
3. **Netzzugang**
4. Internet
5. Transport
6. Anwendung
7. DNS
8. eMail

- LAN
- Abstraktion
- NIC
- WAN
- WAN Standards
- ISDN
- **DSL**
- Alternativen
- WAN Technologien
- WAN Routing

## V.3.2 Netzzugangs Schicht: WAN - DSL

- **ADSL-Standard 1995**: Cu-TP 256 Kanäle 4,3125 kHz
  - adaptives Streuspektrum QAM 2-15 b/s
  - ISDN (POTS) (32 Kanäle)
  - $< 32 \times 4 \text{ kS/s Up} * 4 \text{ Bit/S} = 512 \text{ kbit/s}$
  - $< 192 \times 4 \text{ kS/s Down} * 2 \dots 15 \text{ Bit/S} < 11,4 \text{ Mbit/s}$
- 1-32: Telefon, ISDN
  - 33 – 64: Up
  - 65 – 256 Down
  - 16, 64 Pilotsignal



1. Geschichte
2. Schichten
3. **Netzzugang**
4. Internet
5. Transport
6. Anwendung
7. DNS
8. eMail

- LAN
- Abstraktion
- NIC
- WAN
- WAN Standards
- ISDN
- DSL
- **Alternativen**
- WAN Technologien
- WAN Routing

## V32.2 Netzzugangs Schicht: Alternative Lokal Loop

- Hauptproblem:
  - **Kabelinfrastruktur** analoges (Cu-)Telefonnetz
  - keine reinen Datenanschlüsse (POTS)
- Änderung der Kabelinfrastruktur: Up- und Down
  - **Fiber to the Corb** (FTTC, Bordstein), Teilnehmer TP
  - **IP-basierte Telefonanschlüsse**: Nachteil
    - Abhängigkeit von Status des Internet, Verzögerung, Verzerrung, Echo
    - Zuverlässigkeit, Stabilität, Stromversorgung
- **Digitaler Mobilfunk**: GSM, UMTS, LTE
  - Netzabdeckung erforderlich, Übertragungskapazität, Preis

1. Geschichte
2. Schichten
3. **Netzzugang**
4. Internet
5. Transport
6. Anwendung
7. DNS
8. eMail

- LAN
- Abstraktion
- NIC
- WAN
- WAN Standards
- ISDN
- DSL
- Alternativen
- **WAN Technologien**
- WAN Routing

## V.3.2 Netzzugangs Schicht: Beispiel Technologien

### 1. ARPANET

- grundlegende Konzepte, Algorithmen, Terminologie
- unabhängig, gleichzeitig Arpanet, UNIX, C

### 2. X.25: paketvermitteltes NW für analoge Telefonleitungen

- bidirektionale ASCII-Terminal Anbindung (erWiN Jena 1991)

### 3. Frame-Relay: Verbindung von 2 Standorten

- Anbindung der GSM-Basisstationen an Festnetz

### 4. SMDS: Switched Multimegabit Data Service

- Hochgeschwindigkeitsdienst von Weitnetzbetreibern: DATEX-M

### 5. **10 Gigabit Ethernet**: homogene IP, Ethernet Technologie

- **Konkurrenz zu SONET/OC-192, SDH/STM-64**
- DFN X-Win: Jena 3 Kernnetzfasern, 10x 10GE Verbindungen

1. Geschichte
2. Schichten
3. **Netzzugang**
4. Internet
5. Transport
6. Anwendung
7. DNS
8. eMail

- LAN
- Abstraktion
- NIC
- WAN
- WAN Standards
- ISDN
- DSL
- Alternativen
- WAN Technologien
- **WAN Routing**

## V.3.2 Netzzugangs Schicht: WAN – Routing

- **kein gemeinsames Medium** → Speichervermittlung
- **Speichervermittlungen**: grundlegende HW-Einheit
  - Umrechnung Rahmenformat für verschiedene Technologien
  - Unabhängigkeit von der Quelle
  - Teilstreckenvermittlung
- dynamisches Routing: Redundanzen, Störungen
  - verteilte Berechnung → Selbstanpassung durch Austausch
- **Dijkstra-Algorithmus**: Knoten – Switch; Kanten – Gewicht
  - Weiterleitungstabellen für jeden Weg
  - Distance-Vector-Routing: periodische Nachricht der besten Wege
  - **Link-State-Routing (SPF)**: Verteilung der Infos, Netzwerk-Graph



1. Geschichte
2. Schichten
3. Netzzugang
4. **Internet**
5. Transport
6. Anwendung
7. DNS
8. eMail

- Adressen
- Bindung
- Datagramm
- Weiterleitung
- Header
- MTU
- ICMP
- IPv6

## V.4 Internet Schicht

- bisher: physische Netzwerke, Netzzugangs-Schicht
  - LAN: gemeinsames Medium
  - WAN: kein gemeinsames Medium, Paketvermittlung
  - physische Adressierung, abhängig von Technologie
- **Abstraktion Internet**
  - **Verbindung einzelner heterogener physischer Netze durch Router**
  - kein physisches Netzwerk – virtuelles Netzwerk
  - verbindet beliebige Knoten im Internet
  - einheitliche, technologieunabhängige IP-Adressen, Datenpakete
- **globales Internet**: Internet Assigned Number Authority (IANA)
  - Europa, Naher Osten, Zentralasien: RIPE NCC
  - Abfrage Datenbank: `whois -h whois.denic.de -T dn uni-jena.de`

1. Geschichte
2. Schichten
3. Netzzugang
4. **Internet**
5. Transport
6. Anwendung
7. DNS
8. eMail

- **Adressen**
- Bindung
- Datagramm
- Weiterleitung
- Header
- MTU
- ICMP
- IPv6

## V.4 Internet Schicht Adress-Schema

- **Internet Protokoll-Adressen:** 32 Bit Binärzahl unabhängig von HW
  - ein in einem Internet eindeutig
  - **Präfix:** Identifikation **physisches Netzwerk**, Koordinierung global
  - **Suffix:** Identifikation **Host im physischen NW**, Koordinierung lokal
- Subnetz-Adressierung, **Adressmasken:** Ausnutzung Adressraum
  - Bitwert **1: Netz-Präfix**, Bitwert **0: Host-Suffix**
  - Punkt-Dezimal-Notation:           141.35.14.22           255.255.252.0
  - CCIDR-Notation:                   141.35.14.22           /22
- **spezielle Adressen**
  - Netzwerk-Adresse: Host-Adresse 0 Bsp: 141.35.14.0
  - Begrenzte Broadcast-Adresse: alles 1 Bsp: 255.255.255.255
  - Adresse „This Computer“: Präfix, Suffix 0 Bsp: 0.0.0.0
  - Schleifenadresse: Präfix 127, Suffix egal Bsp: 127.0.0.1

1. Geschichte
2. Schichten
3. Netzzugang
4. **Internet**
5. Transport
6. Anwendung
7. DNS
8. eMail

- Adressen
- **Bindung**
- Datagramm
- Weiterleitung
- Header
- MTU
- ICMP
- IPv6

## V.4 Internet Schicht – Adressbindung

- Übersetzung von IP-Adressen in HW-Adressen: Adressauflösung
  1. Netzzugangsschicht und physische Netze
  2. Abstraktion durch IP-Adressen
- **Quell- und Ziel-Adresse Protokoll-Adressen**
  - Abstraktion: ein einziges Internet
- **Rahmen-Adressen: HW-Adressen**
  - in physischen Netzen nur HW-Adressen lokal bekannt
- **Verbindung zwischen Netzzugangs- und Internet-Schicht**
- **Adressauflösungstechniken technologieabhängig**
  1. Tabellensuche: kleine Netzwerke
  2. Nachrichtenaustausch: Auflösung durch Server, alle Hosts (Broadcast)
    - Adress-Auflösungs-Protokoll, ARP, Neighbor Discovery Protocol (NDP)

1. Geschichte
2. Schichten
3. Netzzugang
4. **Internet**
5. Transport
6. Anwendung
7. DNS
8. eMail

- Adressen
- Bindung
- **Datagramm**
- Weiterleitung
- Header
- MTU
- ICMP
- IPv6

## V.4 Internet Schicht – Datagramme

- **grundlegender, allgemeiner Datenübertragungsdienst: universelle virtuelle Pakete → IP-Datagramme**
  - IP-Pakete von HW unabhängig
  - Erstellung, Verarbeitung von SW, nicht HW
  - werden in technologieabhängige HW-Rahmen eingepackt
- Datagramm-Kopf: Empfänger und Sender IP-Adresse
  - **Datagramm-Kopf: IP-Adresse**
  - **Rahmen-Kopf: HW-Adresse**
- Nutzdatenmenge sehr flexibel
  - von Anwendung abhängig, auch null
  - IPv4: 64 kB, einschließlich Kopf; IPv6 bis ca. 4GiB
- **Übertragung von Router zu Router**
  - Empfang, Entnahme IP Zieladresse, Ermittlung nächste Teilstrecke
  - Ermittlung nächste HW-Adresse, Einpacken in neuen Rahmen, Versenden

1. Geschichte
2. Schichten
3. Netzzugang
4. **Internet**
5. Transport
6. Anwendung
7. DNS
8. eMail

- Adressen
- Bindung
- Datagramm
- **Weiterleitung**
- Header
- MTU
- ICMP
- IPv6

## V.4 Internet Schicht – Weiterleitungstabellen

- Übertragung von Router zu Router: Ermittlung nächste Teilstrecke
- Informationen in **Weiterleitungs- (Routing-) Tabelle**
  - Initialisierung beim Start, Aktualisierung bei Veränderungen
  - je Zeile: das Zielnetzwerk oder nächste Teilstrecke (Router)
  - im Zielnetzwerk: direkte Zustellung
- Algorithmus: IP-Weiterleitungs-NW | Maske | next Hop | IF
  - Ziel-IP & Maske [Zeile i] = Ziel-NW [Zeile i] → Weiterleitung an Hop [i]
    - sonst nächste Zeile
  - Ermittlung HW-Adresse für Hop [i], senden über IF [i]
- Inhalt der Weiterleitungstabelle: → Routing Protokolle
- **Abgrenzung zwischen Netzwerk- und IP-Schicht**
  - im HW-Paket (Rahmen) immer HW-Adresse nächster Hop
  - im IP-Paket (Datagramm) immer Ziel-IP

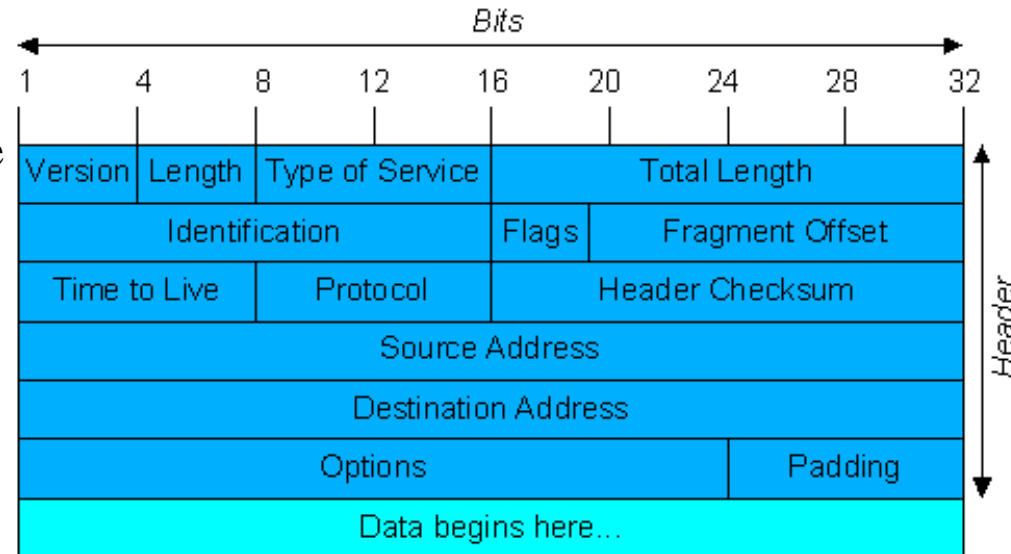
1. Geschichte
2. Schichten
3. Netzzugang
4. **Internet**
5. Transport
6. Anwendung
7. DNS
8. eMail

- Adressen
- Bindung
- Datagramm
- Weiterleitung
- **Header**
- MTU
- ICMP
- IPv6

## V.4 Internet Schicht – Header-Format

- **Version: IPv4**

- Sende-, Empfangs-IP-Adresse
- Gesamtlänge 16 Bit: 64 KiB
- TTL: verhindert Schleifen
- Fragmentierung
  - Identification
  - Offset
  - Flags



- **Version: IPv6:**

- Basis-Kopf + 6 mögliche Zusatz-Header
- Hop-by-Hop, Destination, Routing, Fragment, Authentication, Encapsulating Security Payload, No Next Header

1. Geschichte
2. Schichten
3. Netzzugang
4. **Internet**
5. Transport
6. Anwendung
7. DNS
8. eMail

- Adressen
- Bindung
- Datagramm
- Weiterleitung
- Header
- **MTU**
- ICMP
- IPv6

## V.4 Internet Schicht – MTU

- **MTU: maximale Nutz-Datenmenge eines HW-Rahmens**
  - **Maximum Transmission Unit** (RFC 1122)
  - jede HW-Technologie andere max. Nutz-Datenmenge
- Router verbindet Netzwerke
  - verschiedene Technologie → verschiedene MTU
  - Datagramm größer als MTU
- **Aufteilung** von Datagrammen **in Fragmente**
  - jedes Fragment übliches Datagramm
    - mit gleicher ID, verschiedenem Offset
  - IPv6: Fragment erfolgt durch Quelle, nicht durch Router
- **Pfad-MTU**: kleinste MTU auf dem Pfad zum Ziel (Path MTU)
  - Folge von großen Test-Datagrammen mit Flag „Don't Fragment“

Hyperchannel	65.535 B
Ethernet Jumboframes	9.000 B
FDDI	4.352 B
Ethernet	1.500 B
ISDN	576 B

1. Geschichte
2. Schichten
3. Netzzugang
4. **Internet**
5. Transport
6. Anwendung
7. DNS
8. eMail

- Adressen
- Bindung
- Datagramm
- Weiterleitung
- Header
- MTU
- **ICMP**
- IPv6

## V.4 Internet Schicht – ICMP

- IP: keine Fehlerbehandlung; **Best-Effort Semantik**
- **Internet Control Message Protocol** - RFC 792 (1981)
  - **Übertragung** von **Informationen** und **Fehlermeldungen**
  - **Gewinn von Informationen**: Datagramme, die Fehlermeldungen erzeugen
  - Bestandteil des IP-Protokolls, nicht für Datenübertragung
  - ICMPv6, Neighbor Discovery Protocol (NDP) – Erweiterung für ARP
- 39 ICMP-Meldungen z.B.:
  - Destination Unreachable (Type 3 – Ziel nicht erreichbar)
  - Source Quench (Type 4 – Puffer voll, Datagramme werden verworfen)
  - Echo (Type 8)
  - Time Exceeded (Type 11 – Zeit verstrichen)
  - IPv6 Where-Are-You (Type 33)
  - IPv6 I-Am-Here (Type 34)



1. Geschichte
2. Schichten
3. Netzzugang
4. **Internet**
5. Transport
6. Anwendung
7. DNS
8. eMail

- Adressen
- Bindung
- Datagramm
- Weiterleitung
- Header
- MTU
- ICMP
- **IPv6**

## V.4 Internet Schicht – IPv6

- **Internet Protokoll Version 4 hat sich allgemein bewährt**
  - durch exponentielle Entwicklung: Adresskrise 2011
  - **am 3.02.2011 letzter IPv4 Adressblock von IANA, 8.06.11 Welt IPv6 Tag**
- **Neue Eigenschaften**
  1. **Adressgröße 128 Bit:**  $3,4 \cdot 10^{38}$  Adressen (**155  $10^6$  IPv4-Räume je  $\text{mm}^2$** )
  2. **Header-Format:** 5 64 bit Worte: 1 Info, 2 Quelle, 2 Ziel
  3. **mehrere Header:** Basis-Header + mehrere Zusatz-Header
  4. **Echtzeitübertragung** Mechanismen für Video-, Audiounterstützung
  5. **Protokollerweiterung:** Schema für zusätzliche, nicht vordefinierte Infos
    - flexibel und offen für künftige Entwicklungen
- Adress-Notation: Doppelpunkt-Hexadezimal-Notation mit Null-Kompression
  - **2001:638:906:2:a00:20ff:feed:812d/64**
- Adressbereiche: Fehlende Adresse ::/128; localhost ::1/128
  - Unique Local Unicast: fc00::/7 private Adresse nur im gleichen Subnetz
  - Global Unicast: 2001:xxx::/32 an Provider vergeben

1. Geschichte
2. Schichten
3. Netzzugang
4. Internet
5. **Transport**
6. Anwendung
7. DNS
8. eMail

- Zuverlässigkeit
- Quittung
- Neuübertragung
- Flusskontrolle
- Überlast
- Format
- Routing
- Protokolle
- Multicast

## V.5 Transport Schicht

- nächste Abstraktionsebene: **Transportschicht**
  - Semantik für Anwendungen
  - zuverlässige Übertragung
- Transportschicht **verbindet Anwendungen** auf entfernten Computern
  - **ein Endpunkt** – **Anwendung** auf einem Computer
  - **zweite Endpunkt** – **Anwendung** auf entferntem Computer
  - Internetschicht verbindet Knoten (Rechner) –  
Transportschicht verbindet Anwendungen auf Knoten
- **Verbindungsorientierung:**
  - UDP minimalistisch, aber für Anwendungen: verbindungslos, ungesichert
  - TCP verbindungsorientiert, zuverlässig
- Transport Nachrichten: **Segmente**, werden in IP-Datagramm gekapselt

1. Geschichte
2. Schichten
3. Netzzugang
4. Internet
5. **Transport**
6. Anwendung
7. DNS
8. eMail

- **Zuverlässigkeit**
- Quittung
- Neuübertragung
- Flusskontrolle
- Überlast
- Format
- Routing
- Protokolle
- Multicast

## V.5 Transport Schicht - Zuverlässigkeit

**zuverlässiger Datenaustausch** über **unzuverlässige IP-Datagramme**

- sicherer Verbindungsauf- und -abbau, Fluss-, Überlastkontrolle
- Paketverlust, -doppelung, -verzögerung, -reihenfolge

### **1. Sequenzierung: Paketreihenfolge, -doppelung**

- Sequenznummer 32 Bit Zufallszahl
  - für jede Verbindung neue
    - keine Interferenz mit früheren Verbindungen
- **Eindeutigkeit der Nachrichten**
- jedes Paket wird bestätigt: Acknowledgement, ACK
  - eindeutige Reihenfolge
- doppelte Pakete: verwerfen

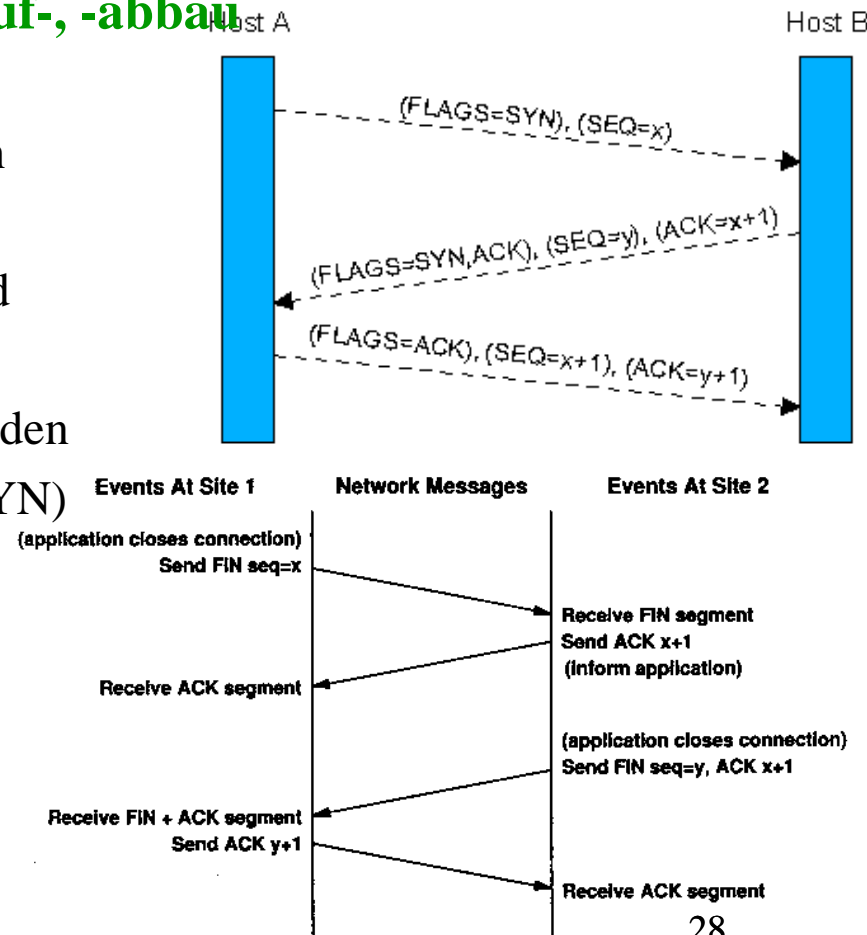
1. Geschichte
2. Schichten
3. Netzzugang
4. Internet
5. **Transport**
6. Anwendung
7. DNS
8. eMail

- Zuverlässigkeit
- **Quittung**
- Neuübertragung
- Flusskontrolle
- Überlast
- Format
- Routing
- Protokolle
- Multicast

## V.5 Transport Schicht – Drei-Wege-Quittung

### 2. Drei-Wege-Quittung: Verbindungsauf-, -abbau

- 3-way-handshake
- dreifacher Austausch von Nachrichten
  - Beweis: notwendig, hinreichend
  - trotz Paketverlust, Duplikaten und Verzögerungen
  - jedes Segment muss bestätigt werden
- Aufbau: Synchronisationssegment (SYN)
- Abbau: Endsegment (FIN)
- Garantie für sichere Verbindung



1. Geschichte
2. Schichten
3. Netzzugang
4. Internet
5. **Transport**
6. Anwendung
7. DNS
8. eMail

- Zuverlässigkeit
- Quittung
- **Neuübertragung**
- Flusskontrolle
- Überlast
- Format
- Routing
- Protokolle
- Multicast

## V.5 Transport Schicht – Neuübertragung

### **3. Neuübertragung:** wichtigste Technik für Zuverlässigkeit

#### – **Paketverlust, -verzögerung**

- wenn keine Bestätigung, dann Verlust → Neuübertragung
- Antwortzeit sehr unterschiedlich, schnell veränderlich
- **Überwachung der Antwortzeiten, Anpassung der Zähler**
- Zeit zwischen Sendung und Bestätigung (**Round-Trip Time**)
- **Linearkombination** aus geschätztem **Wert und Abweichung**
$$RTT_t = RTS + a (RTT_{t-1} - RTS), \quad a = [0,1)$$
  - $RTT_t$ : geschätzte RTT;     RTS: gemessene RTT (sampled)
  - $a \sim 1$ : kurzzeitige Änderungen haben wenig Auswirkung
  - $a \sim 0$ : schnelle Anpassung an Änderungen
- Timeout =  $b * RTT$ ,  $b = (1,2)$ ; **verworfen Pakete**: Timeout Verdoppelung

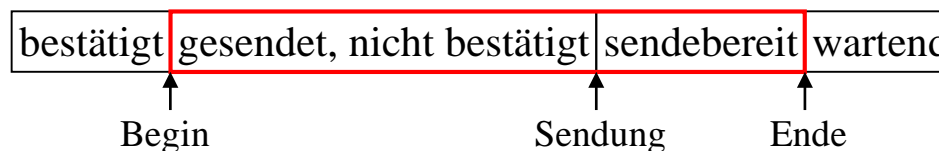
1. Geschichte
2. Schichten
3. Netzzugang
4. Internet
5. **Transport**
6. Anwendung
7. DNS
8. eMail

- Zuverlässigkeit
- Quittung
- Neuübertragung
- **Flusskontrolle**
- Überlast
- Format
- Routing
- Protokolle
- Multicast

## V.5 Transport Schicht – Flusskontrolle

### 4. Fenstermechanismus: Fluss-, Überlastkontrolle

- Austausch anfängliche Fenstergröße (Window-Advertisement)
- auf jeder Seite Puffer für ankommende Daten
- freier Pufferplatz ist Fenster
- Datenpakete senden, bis sie nicht mehr in Puffer passen
  - Null-Fenstergröße (Zero Window): **Sender muss warten**
  - Fenstergröße in der Bestätigung größer als nächstes Paket: **weiter senden**
- **gleitendes Fenster** (sliding window) mit 3 Zeigern
  - Begin: trennt bestätigte von nicht bestätigten bzw. sendebereiten
  - Ende: trennt nicht bestätigte bzw. sendebereite von wartende
  - im Fenster: trennt sendebereite von gesendeten



1. Geschichte
2. Schichten
3. Netzzugang
4. Internet
5. **Transport**
6. Anwendung
7. DNS
8. eMail

- Zuverlässigkeit
- Quittung
- Neuübertragung
- Flusskontrolle
- **Überlast**
- Format
- Routing
- Protokolle
- Multicast

## V.5 Transport Schicht – Überlastkontrolle

### **5. Fenstermechanismus: Überlastkontrolle**

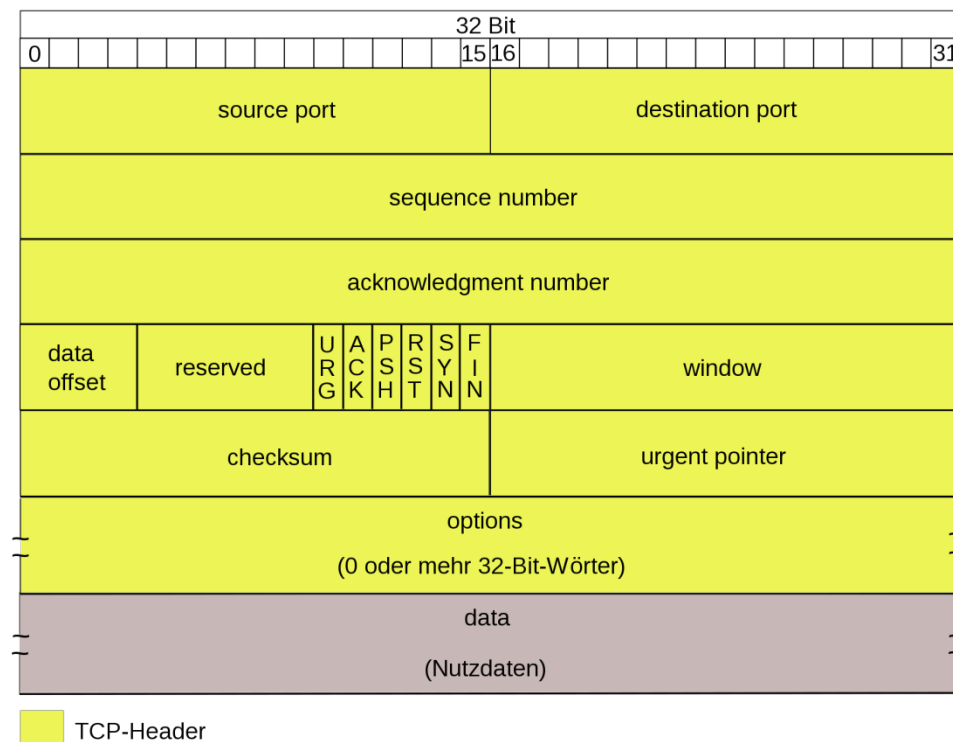
- Austausch anfängliche Fenstergröße
- **Paketverlust als Messeinheit für Überlast**
  - Neuübertragungen verschärfen Überlastung
  - erzeugen zusätzliche Kopien einer Nachricht
  - Kollaps des Gesamtsystems
- Absenken der Rate von Neuübertragungen
  - Fenstergröße wird halbiert
  - Wartezeit bis zur Neuübertragung wird verdoppelt
- nach Überlastung langsame Steigerung der Rate

1. Geschichte
2. Schichten
3. Netzzugang
4. Internet
5. **Transport**
6. Anwendung
7. DNS
8. eMail

- Zuverlässigkeit
- Quittung
- Neuübertragung
- Flusskontrolle
- Überlast
- **Format**
- Routing
- Protokolle
- Multicast

## V.5 Transport Schicht – TCP-Segment Format

- TCP-Nachricht ist Segment
  1. Identifikation der Anwendung: **Port** – keine IP-Adresse !
  2. Sequenznummer: Reihenfolge
  3. Bestätigungsnummer
  4. Fensteranzeige
  5. Flags – Bestätigung
    - Verbindung
  6. Daten Offset: Beginn Nutzdaten
  7. Optionen: MSS
    - Maximum Segment Size
    - in Nutzdaten
    - MTU





1. Geschichte
2. Schichten
3. Netzzugang
4. Internet
5. **Transport**
6. Anwendung
7. DNS
8. eMail

- Zuverlässigkeit
- Quittung
- Neuübertragung
- Flusskontrolle
- Überlast
- Format
- **Routing**
- Protokolle
- Multicast

## V.5 Transport Schicht – Routing

- Routing: wichtiger Bestandteil des globalen Internet
  - statisch: meisten Rechner, in lokalen NW, default Route
  - dynamisches Routing: Router, Info-Austausch, Routing Protokolle
- Routing **skaliert nicht** für gesamtes Internet: Hierarchie
- **Autonomes System (AS)**: Gruppierung von Netzwerken und Routern
  - unter Kontrolle einer administrativen Einheit (Uni, ISP, Unternehmen)
  - Router innerhalb einer Gruppe: tauschen Infos aus
  - **wenige Router** fassen Infos zusammen, **tauschen mit anderen Gruppen aus**
  - keine Festlegungen über Protokolle, Gruppengröße, Datendarstellung
  - **Transit-, Stub-, Multihome-AS-Systeme**
- AS hat eindeutige 32 Bit Nummer (Autonomous System Number ASN)
  - Feb. 2015 fast 50.000 ASN vergeben, Bsp: DTAG AS3320, DFN AS680, Google AS15169

1. Geschichte
2. Schichten
3. Netzzugang
4. Internet
5. **Transport**
6. Anwendung
7. DNS
8. eMail

- Zuverlässigkeit
- Quittung
- Neuübertragung
- Flusskontrolle
- Überlast
- Format
- Routing
- **Protokolle**
- Multicast

## V.5 Transport Schicht – Routing-Protokolle

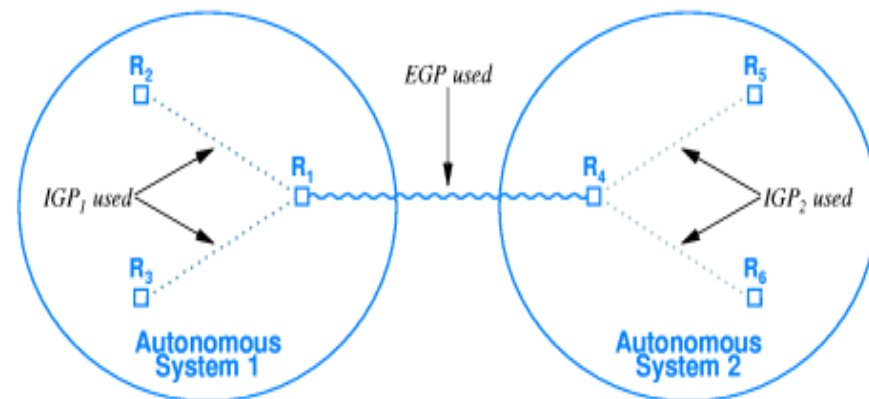
- Routing: wichtiger Bestandteil des globalen Internet

### 1. Interior Gateway Protocols (IGP)

- Austausch innerhalb eines AS
- Routing Metriken: Hop#, admin
- Routing Information Protocol (RIP)
- OSPF – Open Shortest Path First

### 2. Exterior Gateway Protocols (EGP)

- Austausch zwischen AS
- Zusammenfassung aller Routing-Infos vor Übertragung
- Policy-Constraints: Festlegung, welche Infos nach Außen gehen; keine Metrik
- Border Gateway Protocol (BGP-4): Folge von AS: AS 17, 2



1. Geschichte
2. Schichten
3. Netzzugang
4. Internet
5. **Transport**
6. Anwendung
7. DNS
8. eMail

- Zuverlässigkeit
- Quittung
- Neuübertragung
- Flusskontrolle
- Überlast
- Format
- Routing
- Protokolle
- **Multicast**

## V.5 Transport Schicht – Multicast-Routing

- Multicast: dynamische Gruppenmitgliedschaft
  - jederzeit Bei- und Austritt einer Anwendung zu einer Gruppe
  - Information eines nahegelegenen Routers
  - Mitgliedschaft in Gruppe definiert nur eine Reihe von Empfängern
  - Sender muss einer Gruppe nicht beitreten
- Internet Group Multicast Protocol: zw. Host und Router
  - Router muss Pfad zur Gruppe aufbauen
- Ansätze für Multicast-Weiterleitung
  1. **Flood and Prune**: **kleine Gruppen** in fortlaufenden LANs (eine Firma)
  2. **Configuration and Tunneling**: geographisch **weit verstreute** Mitglieder
  3. **Core Based Discovery**: **skaliert** von kleinen, lokalen zu verstreuten Gruppen

1. Geschichte
2. Schichten
3. Netzzugang
4. Internet
5. Transport
6. **Anwendung**
7. DNS
8. eMail

- Socket
- Prozeduren
- Client-Server
- Dienst
- Identifikation
- Beispiel
- Ablauf

## V.6 Anwendung Schicht

- **höchste Abstraktionsebene: Anwendungs-Schicht**
- **Nutzung der Abstraktion Internet**
  - auf Physischen- und Netzzugangs-Schicht erfolgt **realer Datentransport**
  - durch IP-Schicht wird die **Abstraktion Internet** geschaffen
  - Transport-Schicht stellt **Datenaustausch für Anwendungen** sicher
- ✓ **Protokoll-SW ist allgemeine Kommunikations-Infrastruktur, ist passiv**
  - **Protokoll-SW kann Anwendung nicht über Kommunikation informieren**
  - **Anwendung muss Protokoll-SW über Kommunikation informieren**
- ✓ **Anwendungs-SW: stellt Verbindung her, nutzt Kommunikation**
  - jede Kommunikation: mind. 2 Anwendungsprogramme teilnehmen
  - jede Anwendung muss Protokoll-SW genau über Kommunikationsart informieren
  - eine Anwendung muss aktiv einleiten, die andere passiv warten
  - nur, wenn Nachricht genau zu angekündigter Kommunikation passt  
leitet Protokoll-SW Nachricht an Anwendung

1. Geschichte
2. Schichten
3. Netzzugang
4. Internet
5. Transport
6. **Anwendung**
7. DNS
8. eMail

- **Socket**
- Prozeduren
- Client-Server
- Dienst
- Identifikation
- Beispiel
- Ablauf

## V.6 Anwendungs Schicht – Socket-API

- Verbindung zwischen Protokoll-SW und Anwendungen
  - **Netzwerk: Operationen** für Interaktion mit Transport-Schicht
  - **Anwendung: API** – Application Program Interface
- **Sammlung von Prozeduren**
  - Verbindungsaufbau, Datenübertragung (Senden, Empfangen)
  - meist für jede Operation mindestens eine Prozedur
  - Bereitstellung durch BS, Festlegung der Namen, Parameter
  - Nutzung durch Anwendungsprogramme
- **Implementierung durch BS**
  - entstanden als Teil von BSD-Unix (Berkeley SW Distribution)
  - native Prozeduren im BS-Adressraum
  - andere BS: Bibliotheken im Anwender-Adressraum
  - Socket-Bibliotheken: Prozeduren mit gleichen Bezeichnungen, Argumenten

1. Geschichte
2. Schichten
3. Netzzugang
4. Internet
5. Transport
6. **Anwendung**
7. DNS
8. eMail

## V.6 Anwendungs Schicht – Socket-Prozeduren

- Socket
- **Prozeduren**
- Client-Server
- Dienst
- Identifikation
- Beispiel
- Ablauf

- Prozeduren für Verbindungsaufbau
  - open, close, socket : erzeugt bzw. schließt Socket, liefert Deskriptor
  - bind (Port für Server), listen (Warteschlange), accept (neue Socket Verbindung)
  - connect (Verbindungsanforderung)
- Prozeduren für Datenaustausch:
  - send, recv, write, read
- **Socket-API für Parallelbetrieb ausgelegt**
  - Socket hat Referenzzähler: Anzahl der Threads, die Socket benutzen
  - Thread bei Erzeugung:
    - erbt **Liste mit allen Sockets** des Programms
    - erhöht **Referenzzähler der** (geerbten) **Sockets**
  - close Socket: senkt Referenzzähler, entfernt Socket von Liste
- **Grundlage für die Client-Server Programmierung**
  - **Server-Socket** definiert **lokale Adresse** (IP, Port) **für eingehende Anfragen**
  - **Client-Socket** definiert **Ziel-Adresse** (IP, Port) **für entfernten Server**

1. Geschichte
2. Schichten
3. Netzzugang
4. Internet
5. Transport
6. **Anwendung**
7. DNS
8. eMail

- Socket
- Prozeduren
- **Client-Server**
- Dienst
- Identifikation
- Beispiel
- Ablauf

## V.6 Anwendungs Schicht – Client-Server

- **wichtigste Programmier-Modell für Netzwerk-Anwendungen**
  - **Client**: Anwendung leitet Kommunikation **aktiv** ein – **stellt Anfrage**
  - **Server**: Anwendung **wartet** auf Kontaktaufnahme – **beantwortet Anfrage**
  - Modell passt genau zur Funktionsweise von NW-Protokoll-SW
- Merkmale Client-SW:
  - **lokales Anwendungsprogramm**, auf lokalem Computer für entfernten Zugriff
  - wird direkt **von Benutzer aktiviert** und nur für eine Sitzung ausgeführt
  - leitet Kontakt mit jeweils einem Server **aktiv** ein
- Merkmale Server-SW:
  - Programm nur für den einzigen Zweck: **Bereitstellung eines einzigen Dienstes**
  - wird automatisch bei Systemstart aktiviert, **nicht von Benutzer abhängig**
  - **wartet passiv auf Verbindungsaufnahme** durch entfernten Client
  - kann **gleichzeitig mehrere** entfernte **Clients** bedienen
  - nimmt **Kommunikation von beliebigen Clients** entgegen, nur ein einziger Dienst

1. Geschichte
2. Schichten
3. Netzzugang
4. Internet
5. Transport
6. **Anwendung**
7. DNS
8. eMail

- Socket
- Prozeduren
- Client-Server
- **Dienst**
- Identifikation
- Beispiel
- Ablauf

## V.6 Anwendungs Schicht – Server-Dienst

- jeder Dienst: eindeutiger Bezeichner – Beispiel: TCP
  - **16 Bit Ganzzahlen** als Bezeichner, **Protokoll Port Nummer**
  - bei Start des Servers (**Dienstes**), **Registrierung bei Protokoll-SW**
  - Client gibt den **Bezeichner bei Kontaktaufnahme** an
  - Protokoll-SW ermittelt durch Protokoll Port Nummer Server-Programm
- Server muss mehrere Clients gleichzeitig bedienen können
  - dynamische Erzeugung von Kopien des Servers
- Serverprogramm meist aus 2 Teilen
  1. Teil nimmt Anfragen entgegen und erzeugt Bearbeitungs-Thread
  2. Teil bearbeitet die einzelne Anfrage
- **Haupt-Thread hält Server reaktionsbereit**
  - nach Erzeugung Arbeiter-Thread, warten auf nächste Anfrage
- **Arbeiter-Thread bedient Anfrage** und beendet sich



1. Geschichte
2. Schichten
3. Netzzugang
4. Internet
5. Transport
6. **Anwendung**
7. DNS
8. eMail

- Socket
- Prozeduren
- Client-Server
- Dienst
- **Identifikation**
- Beispiel
- Ablauf

## V.6 Anwendungs Schicht – Dienstidentifikation

- Identifikation der Programme: Mechanismus Transportprotokoll
  - IP und Port
- Beispiel TCP – Client
  - Auswahl eines lokalen Ports, keine Dienst-Nummer: source port
  - Quellen: eigene IP, eigenes Port
  - Ziel: Server IP, Dienst-Port
- Beispiel TCP – Server
  - **Kombination Quell- und Ziel-Ports und IPs identifizieren Kommunikation**
  - gleiche Dienst-Anfrage: Ziel-Port und -IP gleich, aber Quellen verschieden
- **Verschiedene Instanzen eines Servers unterschieden durch Quell-Port und –IP**
  - wenn Server und Client auf einem Rechner: Crash wenn Client- = Server-Portnummer
- Dienste: verbindungslos oder –orientiert oder beides, mehrere Protokolle
  - oft: jede Anfrage, jede Antwort in getrennten Nachrichten

1. Geschichte
2. Schichten
3. Netzzugang
4. Internet
5. Transport
6. **Anwendung**
7. DNS
8. eMail

- Socket
- Prozeduren
- Client-Server
- Dienst
- Identifikation
- **Beispiel**
- Ablauf

## V.6 Anwendungs Schicht – Beispiel

- einfache Client Server Kommunikation über Sockets
  1. **Server zählt Anzahl der Client-Kontakte**, meldet Zählerstand als ASCII-Nachricht, beendet Verbindung
  2. Client baut Verbindung auf, wartet auf Nachricht, zeigt sie an und beendet sich  
Bsp: „**Dieser Server wurde 10 mal kontaktiert.**“
- Server: **server [ port ]**
  - Protokoll-Port **Nummer für Anfragen**
  - Default: 5193 (AmericaOnline3), beliebig aber kein Konflikt erzeugen
- Client: **client [ host [ port ] ]**
  - Hostname, Protokoll-Port Nummer des Servers für Anfragen
  - **Default: 5193 und localhost** (Alias-Name für Client-Host-Name)

1. Geschichte
2. Schichten
3. Netzzugang
4. Internet
5. Transport
6. **Anwendung**
7. DNS
8. eMail

- Socket
- Prozeduren
- Client-Server
- Dienst
- Identifikation
- Beispiel
- **Ablauf**

## V.6 Anwendungs Schicht – Beispiel - Ablauf

### C **gethostbyname: Umwandlung in Server-IP**

- **getprotobyname: interne Darstellung**
- **socket: erzeugt Socket**

### S **bind: lokales Port**

### S **listen: Warteschlange einrichten**

### C **connect: Anforderung der Verbindung**

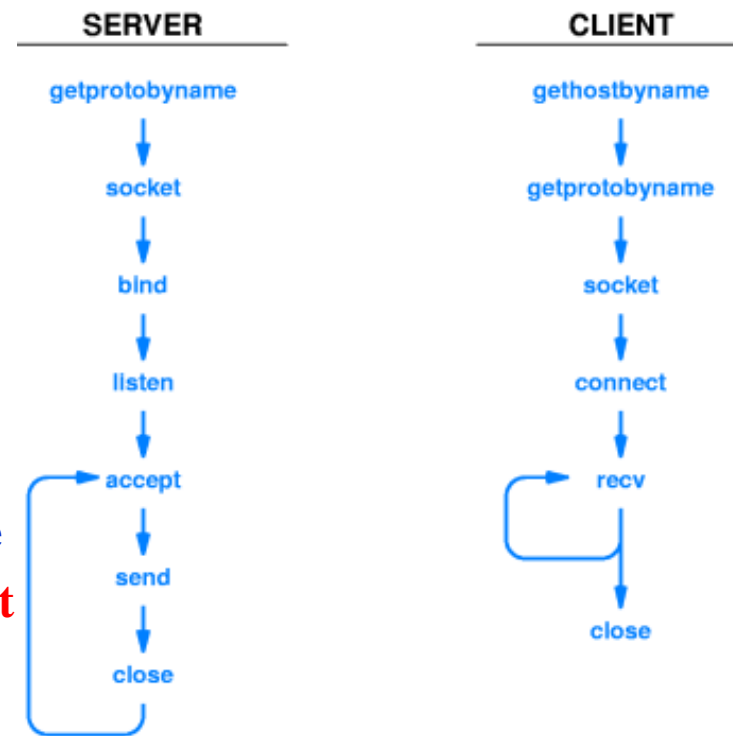
### S **accept: erzeugt neuen Socket für Verbindung**

### S **send: sendet Daten**

### C **recv: empfängt Daten, bis Rückkehr mit 0 Byte**

### S **schließt „Arbeiter“-Socket und ist wieder bereit für eine neue Verbindung**

### C **schließt Client-Socket, wenn Verbindungsende**



1. Geschichte
2. Schichten
3. Netzzugang
4. Internet
5. Transport
6. Anwendung
7. **DNS**
8. eMail

## V.7 Anwendung DNS

- **Domain Name System**
  - Transport-Schicht: Port + IP (binär, kompakt)
  - Anwendungs- (Benutzer-)Schicht: **symbolische Namen**
  - Namen müssen übersetzt werden, transparent für Benutzer
- **Namensdatenbank: weltweit verteilte riesige Datenbank**
- Client-Server Anwendung des Namensystems
  - Client: Anfrage an Namensserver
  - Namensserver sucht IP-Adresse und antwortet
  - Namensserver wird Client eines anderen Servers, wenn kein Treffer
  - Iteration bis Anfrage beantwortet

1. Geschichte
2. Schichten
3. Netzzugang
4. Internet
5. Transport
6. Anwendung
7. **DNS**
8. eMail

## V.7 DNS - Benennungsschema

- symbolischer Computer-Name: **alphanumerische Zeichenkette**
  - **durch Punkte getrennte** Segmente
  - Bsp: isun01.inf.uni-jena.de, anubis.cs.uni-magdeburg.de
- **Anzahl** der Segmente **nicht vorgegeben**
  - Segment ganz links: Host-Name, Segment ganz rechts: Landeskennung
  - 2. Segment von rechts: Domain-Inhaber (Uni Jena bzw. Magdeburg)
- **Domain-Namen hierarchisch aufgebaut**
  - Verwaltung ICAN: Internet Corporation for Assigned Names und Numbers
  - oberste Ebene: TLD – **Top Level Domain standardisiert**
    - com, edu, net, org, int, ca. 250 Landescodes z.B. de, viele weitere
  - Organisationen müssen ihren **Namen registrieren, unterhalb TLD**
    - z.B: uni-jena.de, uni-magdeburg.de, xilinx.com
- **Domain-Namen sind eindeutig**

1. Geschichte
2. Schichten
3. Netzzugang
4. Internet
5. Transport
6. Anwendung
7. **DNS**
8. eMail

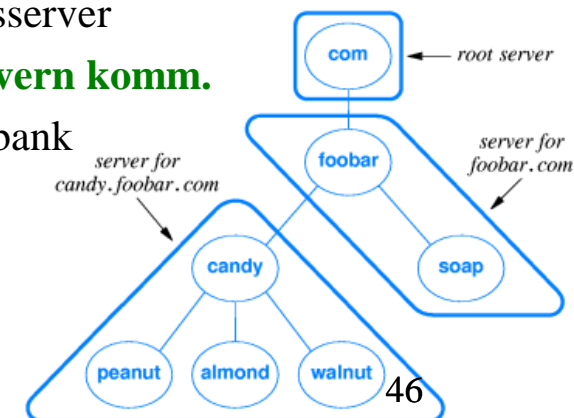
## V.7 DNS – Serverhierarchie

Serverhierarchie entspricht Namenshierarchie

- jeder Server: Autorität für Teil der Namenshierarchie
- an Spitze **13 Root-DNS-Server: Autorität für oberste Ebene (TLD)**
  - 10 Root-Server in USA, auch A-Root, Buchstaben geographisch verteilt
  - Angriffe: 2002 1,8 Mpkts/s, 2006, 2007
- Root-Server: nur Infos wie andere DNS-Server erreicht werden, keine über Hosts
  - **Ende 2006: 123 Root-Server mit Anycast-Instanzen für 6 DNS-Roots**
  - darunter **6 DNS-Server für TLD: de**
  - K-Root-Server seit 2004 in Frankfurt

- **Client-Server Modell macht Autonomie möglich**

- viele Unternehmen mit eigenen Domänen: eigenen Namensserver
- **Namen-Server hat Information, wie er mit anderen Servern komm.**
- insgesamt bilden Server große koordinierte verteilte Datenbank



1. Geschichte
2. Schichten
3. Netzzugang
4. Internet
5. Transport
6. Anwendung
7. **DNS**
8. eMail

## V.7 DNS – Resolver

- Verknüpfung der Namensserver untereinander
  - **jeder Server kennt alle Namensserver der nächst niedrigeren Ebene**
  - **jeder Server kennt mindestens einen Root-Server**
  - **jede Zone (Zweig) hat mindestens einen autoritativen Namensserver**
- SW: Name Resolver, Bibliotheks-Funktion gethostbyname
- **autoritative Antwort**, wenn Name im Zuständigkeitsbereich des Servers
- **rekursive Namensauflösung** (Recursive Query Resolution)
  - DNS-Server holt Auflösung von anderem Server (nicht autoritativ)
  - Antwort an Anwendung: gesuchte IP oder Name existiert nicht
- **iterative Namensauflösung: nur DNS-Server** (nicht autoritativ)
  - Rückgabe Verweis auf andere DNS-Server: Hierarchie schrittweise durchlaufen
- **für DNS gilt Lokalitäts-Prinzip (locality of reference)**
  - **zeitlich**: wiederholt die selben Namensauflösungen
  - **räumlich**: Namen von lokalen öfter als von entfernten Hosts

1. Geschichte
2. Schichten
3. Netzzugang
4. Internet
5. Transport
6. Anwendung
7. **DNS**
8. eMail

## V.7 DNS – Optimierung

- Verfahren hoffnungslos ineffizient, skaliert nicht
  - **Flaschenhals: Root-Server**
  - **Lokalitäts-Prinzip:** häufig gleiche Anfragen

### 1. Replikation

- Anycast: von einem Root-Server existieren viele Exemplare auf ganzer Welt
- schnellste Antwort, meist geographisch nächstliegend (Lokalität)

### 2. Caching: hat größere Bedeutung (zeitliche Lokalität)

- jede Namensauflösung wird in Cache kopiert (nicht autoritativ)
- jeder Eintrag hat Verfallsdatum, wird von autoritativen Server geliefert

### • Datenbankeinträge: Resource Record

- wesentliche Teile: Domain-Name, Verfallszeit, Daten-Typ, Wert
  - A                      Adresstyp IP                      üblich: FTP, Ping, Web
  - MX                    Mail eXchanger                      eMail
  - CNAME              Alias                                      weist auf kanonischen Name für einen Host
  - SOA, NS, PTR



1. Geschichte
2. Schichten
3. Netzzugang
4. Internet
5. Transport
6. Anwendung
7. **DNS**
8. eMail

## V.7 Zusammenfassung DNS

- **Domain Name Service (DNS) wandelt Host-Namen in IP-Adressen**
- Host-Namen durch Punkt getrennte Segmente der Namens-Hierarchie
- Anzahl der Hierarchie-Ebenen nicht standardisiert
- nur Suffix wird durch jeweilige Organisation festgelegt
- Namensauflösung erfolgt durch DNS-Server
- **DNS-Server miteinander verknüpft, weltweite verteilte Datenbank**
- Resolver-SW: Anfragen für Anwendungen als Client an Server
- Server antwortet direkt oder gibt zuständigen DNS-Server an Client
- Replikation von DNS-Servern zur Leistungssteigerung
- **Caching** zur Ausnutzung des **Lokalitäts-Prinzips**
- **Einträge in Namens-Datenbank haben verschiedenen Typ**