

# Internettechnologien I

## Anwendungsprotokolle - Anforderungen

# Inhaltsverzeichnis

## Schichtenmodell

## Architekturen

- Client-Server

- Peer-to-Peer

## Anwendungsprotokolle

- Network Byte Order

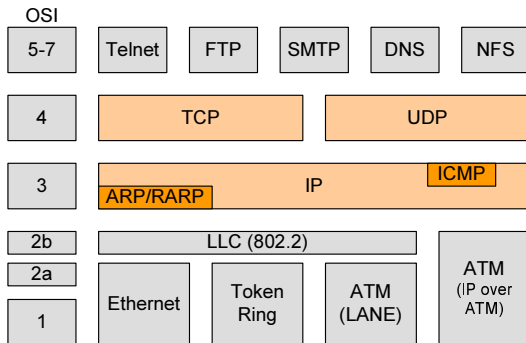
- Transportdienst

- Anforderungen

- Beispiele

# Internet-Schichtenmodell

- ▶ Anwendungen sitzen an den Endpunkten des Netzwerks!
- ▶ Neuer Dienst einfach möglich durch Anschluss eines Hosts mit einem höherschichtigen Protokoll (z.B. HTTP)
- ▶ Adressierung der Anwendungen/Dienste über Portnummern (Portnummern  $< 1024$  für Standarddienste, well-known Ports)
- ▶ z. B. TCP: 22, 25, 53, 80, 443 UDP: 53

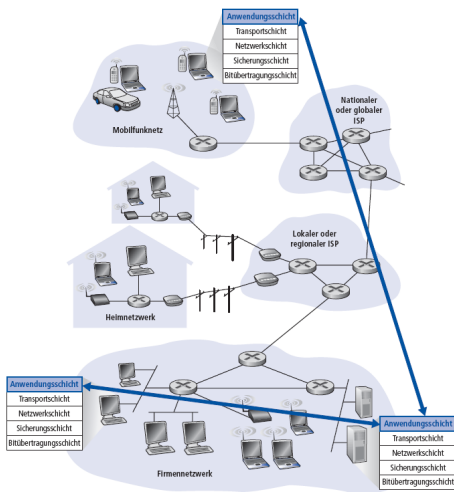


# Einige Netzwerkanwendungen

- ▶ E-Mail
- ▶ Web
- ▶ Instant Messaging
- ▶ Terminalfernzugriff
- ▶ P2P-Filesharing
- ▶ Netzwerkspiele
- ▶ Streaming von Videoclips
- ▶ Voice over IP (VoIP)
- ▶ Videokonferenzen
- ▶ Internet-TV
- ▶ Google Maps/Streetview
- ▶ Onlineanwendungen
- ▶ Onlinespeicher
  - ▶ z.B. Dropbox
- ▶ Grid Computing

# Internet-Schichtenmodell

- ▶ Zugangsnetz, Backbone, Hosts
- ▶ Im Inneren des Netzwerkes werden keine Anwend. ausgeführt!



# Architekturen

- ▶ Client-Server
- ▶ Peer-to-Peer (P2P)
- ▶ Kombination aus Client-Server und P2P

# Client-Server-Architektur

## Server:

- ▶ Immer eingeschaltet
- ▶ Feste IP-Adresse
- ▶ Serverfarmen, um zu skalieren

## Clients:

- ▶ Kommunizieren mit Servern
- ▶ Sporadisch angeschlossen
- ▶ Können dynamische IP-Adressen haben
- ▶ Kommunizieren nicht direkt miteinander

## Beispiele:

- ▶ Webanwendungen z. B. Google-Dienste

# Peer-to-Peer-Architektur

- ▶ Keine zentralen Server
- ▶ Beliebige Endsysteme kommunizieren direkt miteinander
- ▶ Peers sind nur sporadisch angeschlossen und wechseln ihre IP-Adresse
- ▶ Gut skalierbar
- ▶ U.u. schwierig zu warten
- ▶ Beispiele:
- ▶ Bitcoin-Netzwerk



# Kombination von Client-Server und P2P

- ▶ Zentraler Server: z. B. Adresse anderer Clients finden
- ▶ Verbindung zwischen Clients: direkt (nicht über einen Server)

Beispiel:

- ▶ Skype (P2P-Anwendung für Voice-over-IP)

# Kommunizierende Prozesse

**Client-Prozess** Prozess, der die Kommunikation beginnt

**Server-Prozess** Prozess, der darauf wartet, kontaktiert zu werden

- ▶ Anwendungen mit einer P2P-Architektur haben Client- und Server-Prozesse
- ▶ Prozesse auf verschiedenen Hosts kommunizieren, indem sie Nachrichten über ein Netzwerk über Sockets austauschen

# Anwendungsprotokolle

Bestimmung von:

- ▶ Arten von Nachrichten
    - ▶ z.B. Request, Response
  - ▶ Syntax der Nachrichten
    - ▶ Welche Felder sind vorhanden und wie werden diese voneinander getrennt?
  - ▶ Semantik der Nachrichten
    - ▶ Bedeutung der Informationen in den Feldern
  - ▶ Regeln für das Senden von und Antworten auf Nachrichten
- ▶ Öffentliche Protokolle:
    - ▶ Definiert in RFCs
    - ▶ Erlauben Interoperabilität
    - ▶ z. B. HTTP, SMTP
  - ▶ Proprietäre Protokolle:
    - ▶ z. B. Skype

# Network Byte Order

Bei Netzwerkprotokollen ist für fehlerfreien Datenaustausch immer die Byte-Reihenfolge festgeschrieben (versch. Plattformen).

**Big-Endian** Zuerst die höherwertigen Bytes, z. B. Motorola 68000

**Little-Endian** Zuerst niederwertige Bytes, z. B. x86

**Middle-Endian** Mischform zwischen Big- und Little-Endian

**Bi-Endian** Reihenfolge umschaltbar, z. B. ARM-Prozessoren

Beispiel: HEX-Zahl: 0x01020304

- ▶ Protokolle im Internet: Big-Endian-Format
- ▶ Bei bitserieller Übertragung muss auch die Bitreihenfolge spezifiziert sein (Übertragungsprotokoll)

**Big-Endian** Zuerst MSB, z. B. I<sup>2</sup>C

**Little-Endian** Zuerst LSB, z. B. USB, RS232, Ethernet

# Wahl des Transportdienstes

## Datenverlust

- ▶ Toleranz von Datenverlust
- ▶ zuverlässige Übertragung
- ▶ VoIP <-> Dateitransfer

## Zeitanforderungen

- ▶ Toleranz nur geringer Verzögerungen
- ▶ unkritische Zeitanforderungen
- ▶ VoIP, Spiele <-> Dateitransfer

## Datenrate

- ▶ Mindestdatenrate, um zu funktionieren
- ▶ Nutzung der verfügbaren Datenrate (datenratenelastische Anwendungen)
- ▶ Streaming <-> Dateitransfer

# Anforderungen an die Transportschicht

Anwendung	Datenverlust	Datenrate	Echtzeit
Dateitransfer	nein	elastisch	nein
E-Mail	nein	elastisch	nein
Videokonferenz	ja	fest	ja < 150 ms
Gespeichertes Video	ja	fest	ja, einige Sek.
Interaktive Spiele	ja	verschieden	ja, wendige ms

# Protokollbeispiele

Anwendung	Anwendungsprotokoll	Transportprotokoll
E-Mail	SMTP (RFC2821)	TCP
WWW	HTTP (RFC2616)	TCP
Streaming	HTTP, RTP	TCP oder UDP
VoIP	SIP, RTP oder proprietär	i.d.R. UDP

# Zusammenfassung

- ▶ Anwendungsschicht läuft nur auf den Endknoten im Netz
- ▶ Anwendungen nutzen das Socketinterface für UDP/TCP
- ▶ Client- / Server oder P2P-Architektur möglich
- ▶ Kommunikation erfolgt zwischen dem Client- und dem Serverprozess über Nachrichtenaustausch (Protokolle)
- ▶ Frei verfügbare oder proprietären Protokolle möglich
- ▶ HTTP ist ein zentrales (Anwendungs)Protokoll zum Datenaustausch und basiert auf TCP



# Literatur

- ▶ Kurose, Ross „Computernetzwerke“, Person
- ▶ Tanenbaum „Computernetzwerke“, Person