Internet-kommunikation

Hinweis: Im Fach 'Internetkommunikation' ist in der Klausur keine Formelsammlung zugelassen. Daher kann diese Formelsammlung lediglich zur Prüfungsvorbereitung dienen.

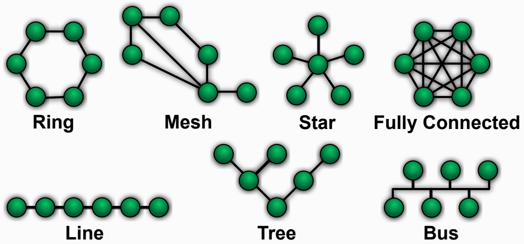
1. Das Internet

„Ein Netz aus Netzen" – Öffentliches Internet verbindet private Intranets.

1.1. Begriffe

User	Teilnehmer
Terminal	Endgerät
Router	
Nodes	
Links	Abschnitte
M2M	Maschine zu Maschine
MSS	Maximum Segment Size
RTT	Round Trip Time

1.2. Netzstrukturen



1.3. Architekturen

Client/Server: Ständige Verfügbarkeit

Peer to Peer(P2P): Hohe Skalierbarkeit, da jeder Besitzer auch Anbieter einer Datei ist.

Paketvermittlung (100 – 1000 Byte) Pakete teilen sich die Netzressourcen, statistisches Multiplexen. Ermöglicht varaible Übertragungsraten

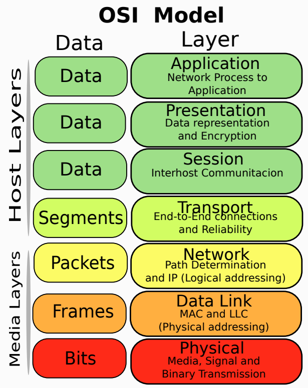
Routing bestimmt den Weg von Quelle zur Senke.

Forwarding Eingehende Pakete werden an den richtigen Ausgang geleitet.

2. Protokolle (Regeln)

Protokolle definieren das Format und die Reihenfolge in der Nachrichten im System gesendet und empfangen werden.

2.1. OSI-ISO Schichtenmodell



2.2. Internet Layers

T_s	Übertragungsverzögerung

Application (HTTP:) Regelt Syntax und Semantik von Nachrichten
Transport (TCP): Ende-zu-Ende Datentransfer zwischen Prozessen.
Network (IP): Routing der Packets durchs Netz mit Quell- und Zieladresse.
Link (MAC,LLC): Verteilung des Medienzugangs (MAC) und Sicherung der Übetragung durch Flusssteuerung (LLC) mit Prüfsummen.
Physical (PHY): Bits auf der Leitung/el. magn. Welle

Kenngrößen eines Netzwerks:

Delay (Verzögerung): und Jitter, d.h. Schwankung der Verzögerung

$$T_{\text{ges}} = \underbrace{T_{\text{VK}}}_{\text{Verarbeitung}} + \underbrace{T_{\text{VQ}}}_{\text{Wartezeit Puffer}} + \underbrace{T_{\text{S}}}_{\text{Übertragung}} + \underbrace{T_{\text{P}}}_{\text{Ausbreitung}}$$

$$T_{\text{S}} = \frac{L}{R} = \frac{\text{Packetgröße (Bit)}}{\text{Bandbreite einer Leitung (Bit/s)}}$$

$$T_{\text{P}} = \frac{d}{s} = \frac{\text{Leitungslänge}}{\text{Ausbreitungsgeschwindigkeit}}$$

Loss (Verlust:) Verluste von Paketen

Throughput (Durchsatz:) $R \cdot \rho$ Goodput: $R \cdot \rho \frac{L_{\text{D}}}{L_{\text{H}} + L_{\text{D}}}$

Leitung mit kleinstem R begrenzt den Throughput (Engpass)

Verkehrsauslastung: $\eta =$ Anforderungsrate · Verzögerung $= a \cdot T_{\text{S}}$

2.3. Kendall Notation für Warteschlangen

Kendall Notation: X / Y / N / s / q

X	Art des Ankuftsprozesses (M für Markov)
Y	Art des Bedienprozesses (M für Markov)
N	Anzahl der Bedieneinheiten (Server)
s	Kapazität (Plätze) der Warteschlange
	Verlustsystem $s = 0$, Wartesystem $s \to \infty$
	Warteverlustsystem $0 < s < \infty$
q	Zahl der Quellen
λ_X	Geburtsrate
μ_X	Sterberate

Wartewahrscheinlichkeit P_W dafür, dass ein ankommendes Paket warten muss, also falls Pakete $> N$

Mittlere Warteschlangenlänge $\Omega = p_N \frac{\frac{A}{N}}{\left(1 - \frac{A}{N}\right)^2}$

Mittlere Wartezeit $T_{\text{W}} = \frac{\Omega}{\lambda}$

Angebot: $A = \frac{\lambda}{\mu}$ Ausnutzung:$\rho = \frac{A}{N} = 1 - p_0$

Protokoll Wirkungsgrad $\rho = \frac{T_{\text{S}}}{T_{\text{S}} + 2T_{\text{P}}}$

Nachrichtensendedauer T_{S}; Kanallaufzeit $T_{\text{P}} = \frac{l}{c}$

2.3.1 Poisson-Prozess

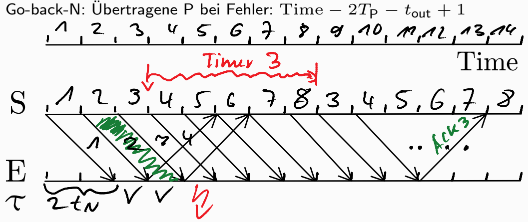
Geburtsrate $\lambda = \frac{1}{E[A]}$

2.3.2 Neg-Exponentialverteilung

mittlere Bearbeitungszeit (Sterberate) $\mu = \frac{1}{\text{Bedienzeit}}$

2.4. Flusssteuerung

Go-back-N: Übertragene P bei Fehler: $\text{Time} - 2T_{\text{P}} - t_{\text{out}} + 1$



3. Protokolle (Beispiele)

3.1. HTTP

3.1.1 Allgemein

Non-Persistent: Neue Anfrage für jedes Objekt (2 RTT pro Objekt)
Persistent: Server lässt Verbindung offen (1 RTT pro Objekt)
Pipelining: Parallele Object Requests möglich (≈ 1 RTT für alles)
DNS Anfrage: $T_{\text{DNS}} = RTT$, TCP Aufbau: $T_{\text{TCP}} = RTT$

Gesamtladen $T_{\text{HTTP}} = 2RTT + 4T_{\text{Si}}$
Round-Trip-Time $RTT = 2(T_{\text{p}} + T_{\text{v}})$
$T_{\text{Si}} = T + T + T$

3.1.2 Coockies

1. set-cookie: Kopfzeile in der HTTP Response Nachricht
2. Cookie Kopfzeile in jeder HTTP Request Nachricht
3. Cookie auf dem Rechner des Anwenders
4. Cookie in der Datenbank des Servers

3.1.3 Web-Cache

Benutzer definiert Webzugriff über Cache (Proxy-Server)
Browser sendet alle HTTP-Requests an den Cache
Im Cache: Proxy sendet Seite aus dem Cache an den Client
Nicht im Cache: Proxy läd Seite in den Cache und leitet sie weiter

3.2. FTP – File Transfer Protocol

Client kontaktiert Server:
Login: user: ¡name¿, pass: ¡password¿
Dateizugriff: list, retr ¡file¿, stor ¡file¿

3.3. SMTP – Simple Mail Transfer Protocol

Header: To: ¡adress¿, From: ¡adress¿, Subject: ¡subject¿
Zugriffsprotokolle: POP3 [RFC 1939], IMAP [RFC 1730], HTTP

3.4. DNS – Domain Name System

Übersetzung von Hostnamen zu IP Adressen
Kanonische Namen (Originalname) oder Aliase (Alternativnamen)
Baumstruktur: DNS-Rootserver speichert TLD-DNS-Server (org, com)
Iterativ: Client → Lokaler DNS → Rest der Reihe nach probieren
Rekursiv: Client ⇄ Lokaler DNS ⇄ RootDNS ⇄ TLD-DNS

4. Transportschicht

Ende-zu-Ende Transport zwischen Prozessen auf hosts

- 2 Tupel → UDP
- 4 Tupel → TCP

zuverlässige Übertragung

4.1. Fensterprotokoll

Kanalausnutzung $\rho_n = \frac{L_N}{L} \frac{W_s t_n}{t_R}$

4.2. IP – Internet Protocol

4.2.1 UDP – User Datagram Protocol [RFC 768]
Unzuverlässiger Transport
Minimales, best-effort
2 Tupel: Empfänger IP, Empfänger Port
— Bild UDP mit Sockets —

4.2.2 TCP – Transfer Control Protocol Reno [RFC 793]
Zuverlässige Datenübertragung
— Bild TCP mit Sockets —
4 Tupel: Empfänger IP, Empfänger Port, Sender IP, Sender Port

Zustände:
Server: `CLOSED, LISTEN, SYN_RECV, ESTB, CLOSE_WAIT, LAST_ACK`
Client: `CLOSED, SYN_SENT, ESTB, FIN_WAIT1, FIN_WAIT2, TIME_WAIT`
Header:
Max-Sequenznummer: 2^{32} Bit
Three-Way-Handshake
FlowControl: Sender schickt nicht mehr Daten als Empf. puffern kann.
ACK: Empfänger bestätigt jedes Packet mit einem `ACK` + `seq` vom nächsten erwarteten Paket. Ist das nächste anders → duplicated `ACK`
Fast-Retransmit: Falls 3 `Ack` mit selber `seq` → resend before timeout
Three-Way-Handshake: `SYN` → `SYN+ACK` → `ACK`

Fehlererkennung Timer (wird bei Empfang eines `ACK`s zurückgesetzt) oder 3 gleiche `ACK`s → fast retransmit
Congestion Control: Verhindert Überlastung: `Rate` $= \frac{\texttt{CongWin}}{\texttt{RTT}}$
Slowstart: Exp. Wachstum (Verdopplung pro `RTT`) von `CongWin`, ab `thresh` lineares Wachstum. (+1`MSS` pro `RTT`)
Bei 3 doppelten `Acks`: `threshNew` = `CongWinNew` = `CongWin`/2 (Reno)
Bei Timeout/Anfang: `CongWinNew` = 1, `threshNew` = `CongWin`/2

LastByteSent - LastByteAcked ≤ CongWin

L	Fehlerrate
W	max. Fenstergröße

Durchsatz: $D = \frac{\text{Daten}}{\text{Zeit}} = \frac{1.22 \cdot \texttt{MSS}}{\texttt{RTT} \cdot \sqrt{L}} = \frac{0,75 \cdot N \cdot \texttt{MSS}}{\texttt{RTT}} = \frac{0,75 \cdot W}{\texttt{RTT}}$
MSS (Max. Seg. Size.) = MTU - Header IP - Header TCP $\approx 1500\text{Byte}$
$R = \frac{W \cdot MSS}{RTT}$

durchschnittliche Fenstergröße mit cong. control: $\frac{W + \frac{W}{2}}{2} = 0.75W$
Anzahl der Segmente die einen Link voll ausnutzen:

4.3. Protokolle auf verlustbehafteten Kanälen
Wirkungsgrad (Utilization):
$U = \frac{\text{Nachrichten Sendedauer}}{\text{Zeit bis nächste Nachricht gesendet werden kann}} = \frac{T_s}{RTT + T_s}$

4.4. Protokolle mit Pipelining
Prinzip: Sende mehrere unbestätigte Pakete und warte dann auf ACKs.
Puffern der unbestätigten Pakete notwendig, höhere Sequenznummern.

4.4.1 Go-Back-N
— Bild Go-Back-N (aus Folie) —
Schiebe Sendefenster bei entspr. `ACK` um eins über die Pakete. Nur ein Timer für das gesamte Fenster. Wenn Timer abläuft, sende gesamtes Fenster nochmals.

4.4.2 Selective Repeat
Wiederhole nur Pakete für die keine ACKs empfangen wurden.

4.5. Anwendungsschicht
— Bild —

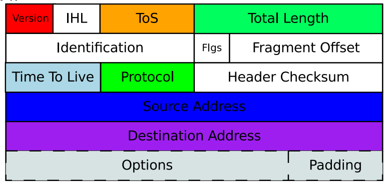
Prinzipien	Architektur:	Client-Server / P2P
	Overlay:	virtuelle Netzstruktur
	Adressierung	Host(IP) + Port
	Transportdienstgüte:	Delay / Loss / Durchsatz
	Info-Abruf:	pull/push
	Nachrichtenformat	
	Zustand	stateless / stateful
	Verbindungsverwaltung	persistent / non-persistent

5. Netzschicht
Protokolle: IP, ATM
Aufgaben: Routing und Forwarding
IP: BestEffort, Reihenfolge egal, keine Zuverlässigkeit

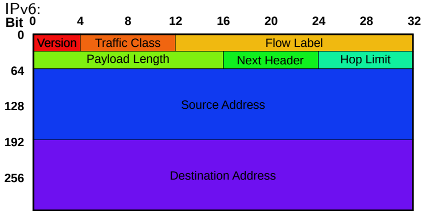
5.1. Router
Routing: Memory, Bus, Crossbar

Pufferdimensionierung bei N Datenflüssen und Link-Datenrate R
Puffergröße $P = \frac{\text{RTT} \cdot R}{\sqrt{N}}$

5.2. IP – Internet Protocol [RFC]
IPv4:



IPv6:



20 Byte Header Seit 1993: Klassenlose Adressierung

5.2.1 Abbildung auf MAC-Adressen

Version	Protocol	Table
IPv4	Address Resolution Protocol ARP	ARP-Table
IPv6	Neighbor Discovery Protocol NDP	NDP-Table

5.2.2 DHCP – Dynamic Host Configuration Protocol [RFC 2131]
Automatische Vergabe von Adressen und Parametern über UDP ohne manuelle Konfiguration

5.2.3 NAT – Network Adress Translation
Abbildung von verschiedenen LAN Adressen auf eine WAN Adresse mit unterschiedlichen Ports.

5.2.4 Subnetze
200.56.168.0/21 bedeutet von 32bit sind 21bit fest und $32 - 21 = 11\text{bit}$ variabel

Anzahl der Class C Subnetze (/24): $24 - 21 = 3 \quad 2^3 = 8$
Anzahl der IP-Adressen: $32 - 21 = 11 \quad 2^{11} = 2048$

5.3. IPv6
32 Bit Adresse, 40 Byte Header, Verkehrsklassen für QoS
Migration mit Tunneling (IPv6 im Datenteil von IPv4)

5.4. ICMP – Internet Control Message Protocol
Für Pings

5.5. Routingverfahren
Dijkstra Algorithmus (erfordert globales Wissen)
globale Infornation: Link-State-Routing
jeder Knoten berechnet den kürzesten Weg zu jedem anderen Router

dezentrale (lokale) Information: Distance-Vector-Routing
Jeder Knoten schätzt den kürzesten Weg zu jedem anderen Router

Hirachisches Routing Autonomer Systeme AS:
Inter-AS-Routing: BGP – Border Gateway Protocol [RFC 4271]
BGP hält das Internet zusammen!
Intra-AS-Routing: RIP, OSPF, IGRP

6. Quality of Service

6.1. Prinzipien
1. Markieren(Verkehrsklassen) und differenziert weiterleiten:
   Scheduling Algorithmen: FIFO, Priority, Round Robin, WFQ

2. Isolation = Eingrenzung und Überwachen: Policing/Shaping → *Token Bucket* begrenzt Paketfluss auf vorgegebene Burst-Größe und vorgegebene durchschnittliche Rate.
   max. Pakete $\leq R \cdot t$ und $\leq b + r \cdot t$
3. Ressourcen vollständig ausnutzen
4. *Admission Control*: Datenfluss meldet Bedarf und muss ggf. abgewiesen werden

DiffServ verschiedene Klassen priorisieren den Verkehr (Feld in IP)
IntServ Reservierung auf dem gesamten Pfad (Garantie mit RSVP)
Weighted Fair Queuing: $B_i = \frac{W_i}{\sum_{j=1}^{n} W_j} \cdot R$
Delay Weighted Fair Queuing + Token Bucket: $d_{\text{max,WFQ}} = \frac{b_i}{R w_i / \sum w_j}$
$d_{\text{max,TB}} = \frac{b_i (R - S_i)}{S_i (R - r_i) w_i / \sum w_j}$

w_i: Prioritätsgewicht des Kanals, R : Gesamtrate, b : max Anzahl der Tokens r : Tokenrate, S : Warteschlangenrate

7. Link Layer and Medium Access Control
Transportiert *frames* von einem Knoten über einen *Link* zum Nachbarknoten. Regelt Zugriff auf das Medium (MAC)
Protokolle: Ethernet, WLAN, PPP
Frame: In header(MAC-Adresse) und trailer(Checksum) verpacktes Datagramm
Fehler durch Signalabfall und Rauschen

7.1. Medium Access Control
Aufteilung des Mediums (Partitioning): Datenrate aufteilen in Zeit/Frequenz
Wahlfreier Zugriff (Random Access): Kollisionenserkennung: Nur senden wenn Kanal frei
Abwechselnder Zugriff
Exponential Backoff: Zufällige Wartezeit aus einem exponentiell steigenden Zeitbereich ARP um MAC Adresse über Broadcast ermitteln.
Hub: Verbindet bloß alle Kabel, keine Pufferung
Switch: Pufferung, gezielte Weiterleitung

8. Sonstiges

Port	Service	Protocols	Description
20	ftp-data	TCP/UDP	File Transfer Data
21	ftp	TCP/UDP	File Transfer Control
22	ssh	TCP	SSH Remote Login Protocol
23	telnet	TCP	Telnet
25	smtp	TCP	Simpler Mail Transfer Protocol
53	domain	TCP/UDP	Domain Name Server
80	http	TCP	Hypertext Transfer Protocol
110	pop3	TCP	Post Office Protocol 3
143	imap4	TCP	Interner Message Access Protocol 4
443	ssl	TCP	HTTPS: HTTP over TLS/SSL