Kommunikationsnetze

1. Vermittlungsverfahren

1.1. Grundgrößen

Sendezeit	$T_S = \frac{L}{R}$
Ausbreitungsverzögerung	$T_P = \frac{s}{c}$
Verarbeitungsverzögerung	Zeit zur Header-Analyse und Routingentschei
Warteschlangenverzögerung	Zeit, die ein Paket im Puffer wartet
Round Trip Time	$RTT = 2(T_P + T_V) + T_A$

1.2. Durchschalte-, Sendungs- und Paketvermittlung

Gegeben: $L_M, L_H, L_D, R, h, T_{P,ges}, T_{V,ges}$

Circuit Switching	$T_{CS} = \frac{L_M}{R} + T_{P,ges}$
Message Switching	$T_{MS} = h \cdot \frac{L_M + L_H}{R} + T_{P,ges} + T_{V,ges}$
Packet Switching	$T_{PS} = \frac{N(L_D + L_H)}{R} + (h-1)\frac{L_D + L_H}{R} + T_P$

Anzahl Pakete:
$N = \left\lceil \frac{L_M}{L_D} \right\rceil$
Für Optimierung:
$N \approx \frac{L_M}{L_D}$
Alternative Darstellung:

$$T_{PS} = \frac{L_M}{R} + \frac{L_M \cdot L_H}{L_D \cdot R} + (h-1)\frac{L_D + L_H}{R} + T_{P,ges} + T_{V,ges}$$

1.3. Optimale Paketgröße

$$\boxed{L_D^* = \sqrt{\frac{L_M \cdot L_H}{h-1}}}$$

Optimale MTU:
$MTU^* = L_D^* + L_H$

1.4. Store-and-Forward Relay (2 Hops)

Ein Paket wird vollständig empfangen, dann weitergeleitet.

$$T_{e2e} = (T_{S1} + T_{P1}) + (T_{S2} + T_{P2})$$

Falls ACKs/Rückkanal: analoge Sende- und Ausbreitungszeiten addieren.

2. ARQ und Stop-and-Wait

2.1. Wirkungsgrad

Begriffe

1. ARQ (Automatic Repeat reQuest): Fehlerkontrolle durch Wiederholung verlorener Pakete.
2. Stop-and-Wait: Sender sendet genau ein Paket und wartet auf ACK.

$$\boxed{\rho = \frac{T_S}{T_S + RTT}}$$

Mit $T_S = \frac{L}{R}$:

$$\rho = \frac{L}{L + R \cdot RTT}$$

Grenzfall $RTT \ll T_S$:

$$\rho \to 1$$

2.2. Stop-and-Wait Ablauf

Zeitachsen-Rezept

1. Sender: Paket senden (T_S), dann warten.
2. Ausbreitung: T_P hin, Empfänger empfängt komplett.
3. ACK zurück: T_P zurück (plus ACK-Sendezeit falls gefordert).
4. Nächstes Paket erst nach ACK.

2.3. Durchsatz und Goodput

Durchsatz	$T_P = \rho \cdot R$
Goodput	$G_P = \frac{L_D}{L_D + L_H} \cdot T_P$

2.4. Fehlerwahrscheinlichkeiten

Bitfehler $\to$ Paketfehler (L in bit):

$$p_P = 1 - (1 - p_B)^L$$

Mittlere Wiederholungen (geometrisch):

$$\frac{K}{N} = \frac{p_V}{1 - p_V}$$

Wirkungsgrad mit Fehlern:

$$\rho = \frac{T_S}{T_S + RTT + \frac{K}{N}(T_S + T_T)}$$

ARQ mit max. Wiederholungen Z_{max}:
Sei $p := p_P$ (Paketfehlerwahrscheinlichkeit pro Versuch).

$$P(\text{verworfen}) = p^{Z_{max}+1}$$

3. Fensterprotokolle

3.1. Maximale effektive Fenstergröße

Begriffe

1. Sliding Window: Mehrere unbestätigte Pakete dürfen gleichzeitig im Kanal sein.

$$W = \left\lceil \frac{T_S + RTT}{T_S} \right\rceil = \left\lceil \frac{RTT}{T_S} + 1 \right\rceil$$

3.2. Mittlere Übertragungszeit

$$T_U = \frac{T_S + RTT}{\min(w_{Sm}, W)}$$

Falls $w_{Sm} \geq W$:
$T_U = T_S$

3.3. Maximales Fenster bei Fehlerfall

Go-Back-N	$W^* = \left\lceil \frac{T_S + T_T}{T_S} \right\rceil$
Selective Repeat	$W^* = \left\lceil \frac{T_S + RTT + T_T}{T_S} \right\rceil$

3.4. Bandwidth-Delay-Product

$$\boxed{BDP = R \cdot RTT}$$

Benötigte Fenstergröße:

$$W \approx \frac{R \cdot RTT}{L} + 1$$

Vollauslastung, falls:

$$w_{Sm} \geq \frac{RTT}{T_S} + 1$$

Kanalauslastungsformel:

$$\rho_N = \frac{T_S W_S}{T_{ges}} \cdot \frac{L_N}{L}$$

Maximale Auslastung:

$$\rho_{n,max} = \frac{L_N}{L}$$

4. Web Cache

4.1. Auslastung

$$\rho = \frac{L \cdot a}{R}$$

4.2. Durchschnittliche Verzögerung

$$\boxed{T = h \cdot T_{Cache} + (1-h)(T_{Cache} + T_{Server})}$$

M/M/1-Näherung:

$$T_{Cache} = \frac{T_S}{1 - \rho}$$

5. HTTP Verzögerung

Begriffe

1. Nicht persistent: Für jedes Objekt neue TCP-Verbindung.
2. Persistent: Mehrere Objekte über eine TCP-Verbindung.
3. Pipelining: Mehrere Requests ohne auf Antwort zu warten.

DNS-Zeit:

$$T_{DNS} = \sum_{i=1}^{n} RTT_i$$

TCP 3-Way Handshake (Zeitbaustein):

$$T_{HS} \approx RTT$$

zzgl. Sendezeiten der Kontrollpakete (falls gefordert).
Mit Sendezeiten:

$$T_{HS} = T_S(\text{SYN}) + T_P + T_S(\text{SYN+ACK}) + T_P + T_S(\text{ACK}) + T_P$$

Ein Objekt:

$$T = T_{DNS} + T_{HS} + RTT_0 + T_S(\text{Request}) + T_S(\text{Response})$$

Allgemein - Nicht persistent (m Objekte):

$$T_{ges} = T_{DNS} + m(T_{HS} + RTT_0 + T_S(\text{Req}) + T_S(\text{Resp}))$$

Persistent ohne Pipelining (m Objekte):

$$T_{ges} = T_{DNS} + T_{HS} + m(RTT_0 + T_S(\text{Req}) + T_S(\text{Resp}))$$

Persistent mit Pipelining (m Objekte):

$$T_{ges} = T_{DNS} + T_{HS} + RTT_0 + \sum_{i=1}^{m} T_S(\text{Req}_i) + \sum_{i=1}^{m} T_S(\text{Resp}_i)$$

6. TCP

6.1. Grundlagen: Congestion / Flow Control

Begriffe

1. Congestion Window (cwnd): Senderseitige Begrenzung unbestätigter Bytes.
2. Congestion Control: Überlaufsteuerung im Netz.
3. Flow Control: Überlaufsteuerung am Sender (Empfängerfenster).
4. Jede TCP-Verbindung besitzt ein eigenes cwnd.

$$CongWin \geq LastByteSent - LastByteAcked$$

Mit Flow Control:

$$\min\{RecvWin, CongWin\} \geq LastByteSent - LastByteAcked$$

Maximale Datei ohne Wraparound:
2^{32} Byte ≈ 4.29 GB

6.2. Slow Start

Exponentielles Wachstum:
$cwnd_k = 2^k$
Gesendete Segmente bis Phase k:

$$\sum_{i=0}^{k} 2^i = 2^{k+1} - 1$$

6.3. Reno

Triple Duplicate ACK:

$$ssthresh = \frac{cwnd}{2}$$

$$cwnd = ssthresh + 3 \cdot MSS$$

Timeout:

$$ssthresh = \frac{cwnd}{2}$$

$$cwnd = 1 \cdot MSS$$

Congestion Avoidance (Additive Increase):

$$cwnd = cwnd + 1 \cdot MSS \text{ pro RTT}$$

6.4. Tahoe

Triple Duplicate ACK:

$$ssthresh = \frac{cwnd}{2}$$

$$cwnd = 1 \cdot MSS$$

Danach Slow Start bis $ssthresh$.

6.5. Congestion Window

Maximales Fenster	$cwnd_{max} = \frac{RTT \cdot R}{MSS}$
Durchschnittliches Fenster	$cwnd_{avg} = 0.75 \cdot cwnd_{max}$
Durchschnittlicher Durchsatz	$D = \frac{cwnd_{avg}}{RTT}$

Hinweis: $cwnd$ in Segmenten, für Bytes: $cwnd \cdot MSS$

6.6. Highspeed

Benötigte Segmente zur Linkauslastung	$\#Seg = \frac{RTT \cdot R}{MSS}$
TCP-Durchsatzformel	$D = \frac{1.22 \cdot MSS}{RTT\sqrt{L}}$
Nach Verlustwahrscheinlichkeit	$L = \left(\frac{1.22 \cdot MSS}{RTT \cdot D}\right)^2$

Volle Auslastung: $D = R$
RTT-Näherung: $RTT \approx 2T_p$

7. Verkehrstheorie und Wartesysteme

7.1. Kendall-Notation

A/B/N/K/M

A	Ankunftsprozess (M = exponential, D = deterministisch, G = allgemei
B	Bedienprozess
N	Anzahl Server
K	Systemkapazität
M	Quellenanzahl

Beispiele:
$M/M/N/0/\infty$: Verlustsystem
$M/M/1/\infty/\infty$: Wartesystem
$G/G/K/0/\infty$: allgemeines Verlustsystem
Systemkapazität K: maximale Anzahl Kunden im System
$K = N + s$ mit N = Anzahl Server, s = Anzahl Pufferplätze
Anzahl Zustände: $K + 1$ (Zustände $0, 1, \ldots, K$)

7.2. Zufallsverkehr (Poisson)

Poisson-Prozess (Anzahl Ankünfte in Zeit t):

$$P_k(t) = \frac{(\lambda t)^k}{k!} e^{-\lambda t}$$

Erwartungswert:

$$E[K(t)] = \lambda t$$

7.3. Grundgrößen

λ	Ankunftsrate	$\frac{1}{E[T_A]}$
ε (oder μ)	Bedienrate	$\frac{1}{E[T_H]}$
A	Verkehrsangebot	$\frac{\lambda}{\varepsilon}$ (Erlang)

ρ	$\frac{A}{N}$ (Auslastung)

Definition:
1 Erlang = 1 dauerhaft belegter Kanal
Aus Beobachtungsdiagramm (1 Server):

$$\hat{\rho} = \frac{t_{belegt}}{t_{gesamt}}, \quad A \approx \rho$$

Stabilitätsbedingung:
$A < N$
Statisches Gleichgewicht (Flussbalance):

$$\lambda_n p_n = \mu_{n+1} p_{n+1}$$

7.4. Erlang-B (M/M/N/0)

p_x	$\frac{A^x}{x!} p_0$
Rekursion	$p_x = p_{x-1} \cdot \frac{A}{x}$
p_0	$\frac{1}{\sum_{k=0}^{N} \frac{A^k}{k!}}$
Blockierwahrscheinlichkeit	$B = p_N = \frac{\frac{A^N}{N!}}{\sum_{k=0}^{N} \frac{A^k}{k!}}$
Effektive Ankunftsrate	$\lambda_{eff} = \lambda(1 - B)$
Verarbeiteter Verkehr	$Y = A(1 - B)$
Verlustverkehr	$A_{verl} = A \cdot B$

Geburts-Tod-Raten (M/M/K/0):

$$\lambda_n = \lambda$$

$$\mu_n = n\mu \quad (n \leq K)$$

Rekursion:

$$p_{n+1} = p_n \frac{\lambda}{(n+1)\mu}$$

7.5. M/M/1 Wartesystem

ρ	$\frac{\lambda}{\varepsilon}$
Stabilität	$\rho < 1$
Zustandswahrscheinlichkeit	$p_x = (1 - \rho)\rho^x$
Mittlere Warteschlangenlänge	$\Omega = \frac{\rho^2}{1-\rho}$
Mittlere Systemlänge	$L = \frac{\rho}{1-\rho}$

Little:
$\lambda T_W = \Omega$
$T_W = \frac{\Omega}{\lambda}$
Mittlere Durchlaufzeit:
$E[T_D] = T_W + h$

7.6. M/M/N Wartesystem

$\rho = \frac{A}{N}$
Wartewahrscheinlichkeit (Erlang-C):

$$P_W = \frac{\frac{A^N}{N!} \cdot \frac{N}{N-A}}{\sum_{k=0}^{N-1} \frac{A^k}{k!} + \frac{A^N}{N!} \cdot \frac{N}{N-A}}$$

Mittlere Warteschlangenlänge:

$$\Omega = P_W \cdot \frac{A}{N - A}$$

Mittlere Wartezeit (Gesetz von Little):

$$T_W = \frac{\Omega}{\lambda}$$

Latenzzeit (Durchlaufzeit):

$$T_D = T_W + T_H = T_W + \frac{1}{\varepsilon}$$

Wahrscheinlichkeit keine Wartezeit:

$$P(\text{keine Wartezeit}) = 1 - P_W$$

Wenn $A \geq N$:
$P_W \to 1$
$\Omega \to \infty$

7.7. Endliche Kapazität M/M/N/s

M/M/1/s (1 Server, s Pufferplätze):
$K = 1 + s$ (Systemkapazität), Zustände: $0, 1, \ldots, K$ (also $K + 1 = s + 2$ Zustände)

$$p_n = p_0 \rho^n, \quad \rho = \frac{\lambda}{\mu}, \quad n = 0, \ldots, K$$

$$p_0 = \frac{1 - \rho}{1 - \rho^{K+1}} = \frac{1 - \rho}{1 - \rho^{s+2}} \quad (\rho \neq 1)$$

Blockierung:

$$B = p_K = p_{s+1}$$

M/M/N/s (N Server, s Pufferplätze, Geburts-Tod):
$K = N + s$ (Systemkapazität), Zustände: $0, 1, \ldots, K$ (also $K + 1 = N + s + 1$ Zustände)

$$p_{n+1} = p_n \cdot \frac{\lambda}{\mu \min(n+1, N)} \quad (n \leq K - 1 = N + s - 1)$$

Normierung:

$$p_0 = \left(\sum_{n=0}^{K} \prod_{i=0}^{n-1} \frac{\lambda}{\mu \min(i+1, N)} \right)^{-1} = \left(\sum_{n=0}^{N+s} \prod_{i=0}^{n-1} \frac{\lambda}{\mu \min(i+1, N}$$

Blockierung:

$$B = p_K = p_{N+s}$$

Wahrscheinlichkeit keine Wartezeit:

$$P(\text{keine Wartezeit}) = \sum_{n=0}^{N-1} p_n \text{ (PASTA)}$$

8. Graphentheorie

Grad eines Knotens	$deg(v)$
Handshaking-Lemma	$\sum_{v \in G} deg(v) = 2K$
Baum	$K = M - 1$
Adjazenzmatrix (ungerichtet)	$a_{ij} = a_{ji}$

Ein Graph mit $M - 1$ Kanten ist genau dann ein Baum, wenn er zusammenhängend ist.
Euler-Kreis:
Alle Knoten haben geraden Grad.
Hamilton-Kreis:
Jeder Knoten wird genau einmal besucht.
Anzahl ungerader Knoten ist immer gerade.
Vollständiger Graph: Anzahl Gerüste

$$K_m : m^{m-2}$$

8.1. Minimum Spanning Tree

MST enthält:
$M - 1$ Kanten
Prim:
Wähle jeweils kleinste angrenzende Kante.

8.2. Shortest Path

8.2.1. Dijkstra

Initialisierung:
$D[s] = 0$
$D[v] = \infty$
Relaxation:
$D[u] = \min(D[u], D[v] + c(v, u))$
Nur gültig für $c(v, u) \geq 0$

8.2.2. Bellman-Ford

Relaxiere alle Kanten:
$D[v] = \min(D[v], D[u] + c(u,v))$
Wiederhole:
$M-1$ Iterationen
Negative Zyklen, wenn nach $M-1$ Iterationen noch Verbesserung möglich.

8.3. Maximaler Fluss

Max-Flow-Min-Cut:
F_{max} = Kapazität des minimalen Schnitts

Flusserhaltung (innere Knoten):

$$\sum f_{in} = \sum f_{out}$$

Kapazitätsbedingung:

$$0 \leq f_{ij} \leq c_{ij}$$

9. Routing

9.1. Distance Vector

Routingtabelle	$RT = N \cdot L$ Byte
Nachrichten im Netz	$\approx 2KNL$ (Tabelle pro Kante)

Update-Gleichung (Bellman-Ford):

$$D_x(y) = \min_v \{c(x,v) + D_v(y)\}$$

Next Hop:

$$v^* = \arg\min_v \{c(x,v) + D_v(y)\}$$

Count-to-Infinity Problem möglich.
Gegenmaßnahmen: Split Horizon, Poison Reverse.

9.2. Link State

LSA-Größe	$LSA(k) = deg(k) \cdot L$
Flutung	Nachrichten: $N \cdot K$
Bytes gesamt	$\approx 4K^2L$ (bidirektional)

LSR verwendet Dijkstra auf der vollständigen Topologie.
LSA-Inhalt (typisch): Router-ID, Nachbarn + Linkkosten, Seq#, Age.
Fluten mit Hop-Count: mindestens Graphdurchmesser (in Hops).

10. IP und Subnetting

Anzahl IP-Adressen	2^{32-n}
Hostbits	$h = 32 - n$
Verwendbare Hosts	$2^h - 2$
Anzahl /24 Netze in /n	2^{24-n}

Klassische Klassen:

Class A	/8
Class B	/16
Class C	/24

Benötigte Hostbits:

$$2^h \geq \text{benötigte Hosts} + 2$$

CIDR Zerlegung:
Netzadresse: alle Hostbits = 0
Broadcast: alle Hostbits = 1
Longest Prefix Matching:
Wähle größtes n.

10.1. NAT

NAT-Eintrag:
(interne IP, interner Port) ↔ (externe IP, externer Port)
Port Address Translation erlaubt mehrere interne Hosts mit gleicher externer IP.
Eindeutigkeit: Externe IP + externer Port müssen eindeutig sein.

11. Allgemeine Begriffe

Mobilität
1. Home Agent, care-of-address, Triangle Routing.

Kommunikationskette:

Quelle → Sender → Kanal → Empfänger → Senke

Adressarten	Public (global) vs. Private (lokal)
QoS Bits	DSCP/ToS im IP-Header

Kommunikationsarten:

Verbindungsorientiert	z.B. TCP
Verbindungslos	z.B. UDP
Durchschaltung	Circuit Switching
Paketvermittlung	Packet Switching

Traffic Matrix:

$$TM_{ij} = \text{Verkehr von Quelle } i \text{ nach Ziel } j$$

12. Zuverlässigkeit und Verfügbarkeit

λ	$\frac{1}{MTTF}$
μ	$\frac{1}{MTTR}$
Verfügbarkeit	$a = \frac{MTTF}{MTTF+MTTR}$
Nichtverfügbarkeit	$e = 1 - a$
Serienschaltung	$a_{ges} = \prod_i a_i$
Parallelschaltung	$a = 1 - \prod_i (1 - a_i)$

Falls alle Komponenten gleich:

$$a_{ges} = a^n$$

Jährliche Ausfallzeit:
$t_{down} = e \cdot 365 \cdot 24 \cdot 60$