

Schichtenmodelle:

TCP/IP - Referenzmodell:

| Schicht | Aufgabe | Protokolle |
|---------|------------|----------------------------|
| 1 | Anwendung | HTTP, FTP, SMTP, POP3, SSH |
| 2 | Transport | TCP, UDP |
| 3 | Internet | IPv4/v6, ICMP, IPsec |
| 4 | Netzzugang | Ethernet, WLAN, ATM, PPP |

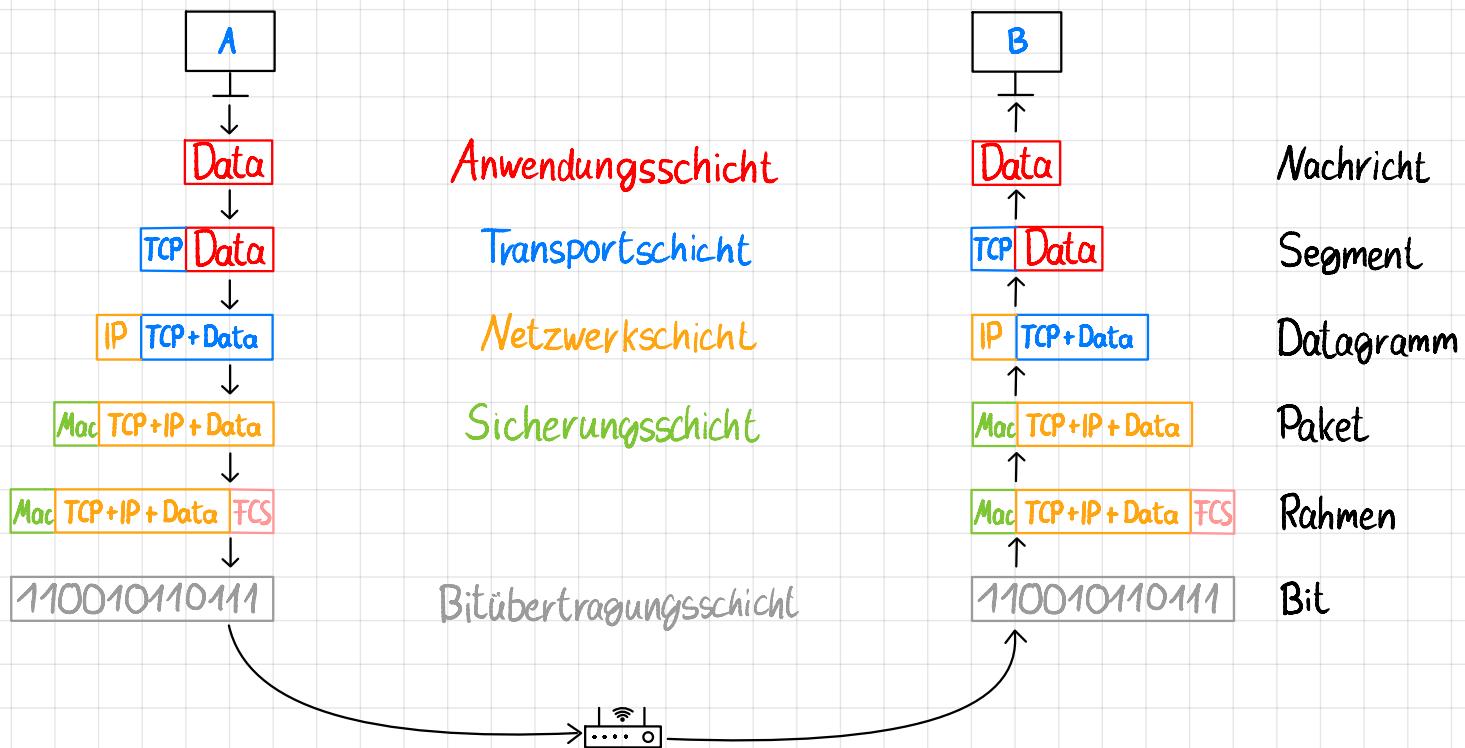
Hybride Referenzschicht:



Vollständiges ISO/OSI - Modell:

| TCP/IP | OSI - Modell: |
|-----------|--|
| Anwendung | Anwendungsschicht Präsentationsschicht Sitzungsschicht |
| Transport | Transportschicht |
| Internet | Netzwerkschicht |
| Netzwerk | Sicherungsschicht Bitübertragungsschicht |

ISO/OSI Kapselung:



Standardisierung:

1. International Organization for Standardization (ISO):

- ↪ Normung von Telekommunikationssystemen
- ↪ "ISO/OSI" - Modell

2. International Telecommunications Union (ITU):

- ↪ Weltweite Koordination von Telekommunikation
- ↪ ITU Radio Communication Sector Standards (ATM, ISDN, V.24)

3. Institute of Electrical and Electronics Engineers (IEEE):

- ↪ Wichtiges Standardisierungsgremium für Elektronik und Informatik
- ↪ WLAN Standardisierung

4. Internet Engineering Task Force (IETF):

- ↪ Definition von Internet standards
- ↪ RFCs

OSI Layer 7-5: Anwendung:

Anwendungsschicht:
- Enthält Funktionen für Anwendungen
- Dateiein- und ausgabe erfolgt in dieser Schicht

HTTP:

1. Nichtpersistentes HTTP: - Ein Objekt über TCP-Verbindung
- HTTP/1.0 verwendet nicht persistentes HTTP

Aufruf: example.de

TCP-Verbindung: example.de:80

Hörcht auf: example.de:80
Benachrichtigt: Client

Schickt: Request /home.html über den TCP-Socket

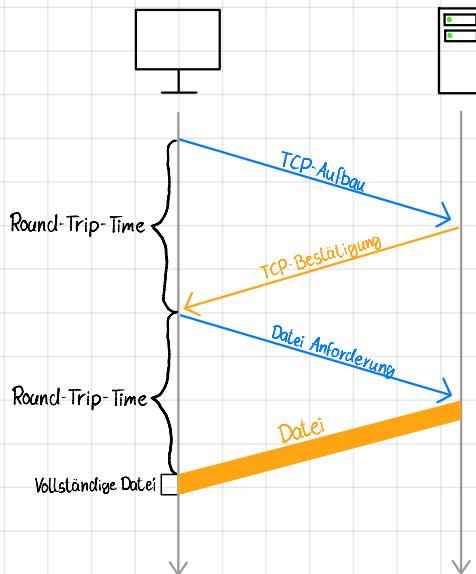
Schickt: HTTP-Response mit gewünschter Ressource

Empfängt: Ende TCP-Verbindung
Prüft: Auf neue empfangbare Objekte

Schließt: TCP-Verbindung

Dieser Prozess wird nun für jedes referenzierte Objekt ausgeführt.

1.1 Round-Trip-Time:



Die Round Trip Time ist die Verzögerung von einem Paket, welches vom Client zum Server und zurück gesendet wird.

Für nicht persistentes HTTP gilt folgende Verzögerung:

- + 1x Round Trip Time für TCP Aufbau
- + 1x Round Trip Time für Request/Response
- + 1x Übertragungsverzögerung der Ressource

Probleme nichtpersistentes HTTP:

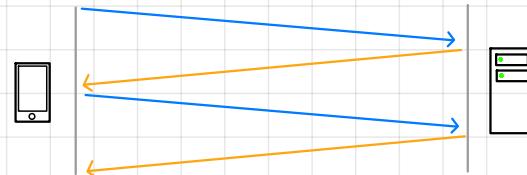
- 2 Round Trip Times pro Objekt
- Aufwand im Betriebssystem pro TCP-Verbindung

2. Persistentes HTTP:

- Mehrere Objekte über TCP-Verbindung
- HTTP/1.1 verwendet persistentes HTTP standardmäßig
- Server lässt TCP-Verbindung offen

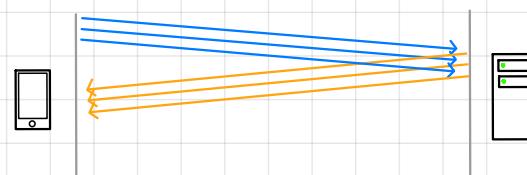
Persistent ohne Pipelining:

- Client schickt erst nach einer Response die nächste Request
- Eine Round Trip Time pro referenziertes Objekt



Persistent mit Pipelining:

- Client schickt bei jeder Referenz sofort eine Request
- Standard in HTTP/1.1



Request-Zeile {
 Methode Zieldatei Protokoll
 Get /page.html HTTP/1.1
 Header Zeilen {
 Host: example.de
 User-Agent: Mozilla/4.0
 Connection: close
 Accept-language: de
}

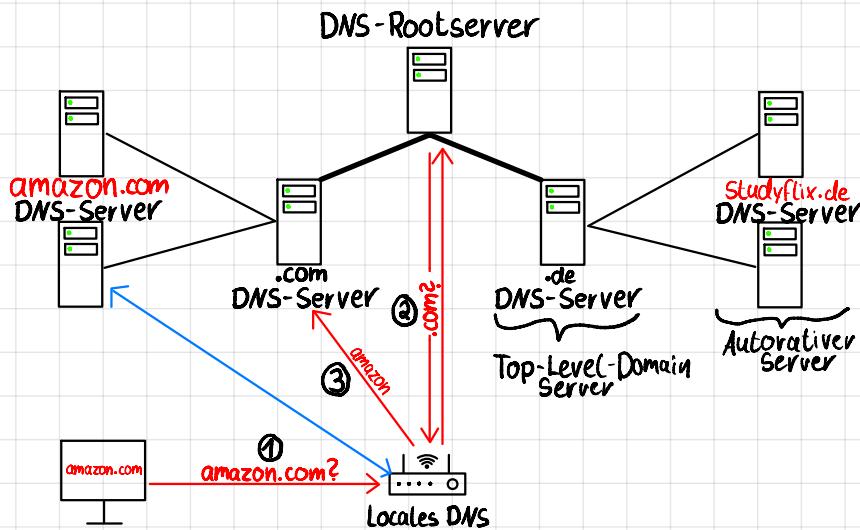
| Protokoll | Statuscode |
|------------------------------------|------------|
| HTTP/1.1 | 200 OK |
| Header Zeilen { | |
| Connection Close | |
| Date: Thu, 6 Oct 2011 11:01:15 GMT | |
| Server: Apache/1.0.3 (Unix) | |
| Content-type: text/html | |
| Daten { | |
| Lorem ipsum dolor | |

Netznahe Dienste:

- Dienste, die:
- Notwendig für ein Netz sind (DHCP)
 - Nutzern erheblich helfen (DNS)

DNS: Das Domain Name System löst Domain-Namen in IP-Adressen auf

- ↪ A/NS-Records: Hostnamen zu IP-Adressen (google.de → 216.58.207.174)
- ↪ CName: Aliasnamen zu Hostnamen (maps.google.de → google.de/maps)
- ↪ MX-Record: Aliasnamen für Mailserver (@gmail.de)



Zonendatei: Beinhaltet Ressource Records mit:

- A: IPv4 eines Hosts
- AAAA: IPv6 eines Hosts
- SOA: Verwaltunginfos (Admin Name und Email)
- CName: Alias Domain
- MX: SMTP-Mailserver
- PTR: Zuweisung von IP zu Hosts

DHCP: Das Dynamic Host Configuration Protocol verteilt IPs an Endgeräte

- ↪ Keine manuelle Vergabe notwendig



IP Lease: Eintrag vergebene IP
Lease-Time: Verfallsdatum eines Eintrags
 ↪ Verlängerung durch Client

Telnet: Zeichenorientierter Datenaustausch über Kommandozeile

- ↪ Wird zur Fehlersuche verwendet
- ↪ Verbindung zu beliebigen Ports (**unverschlüsselt**)

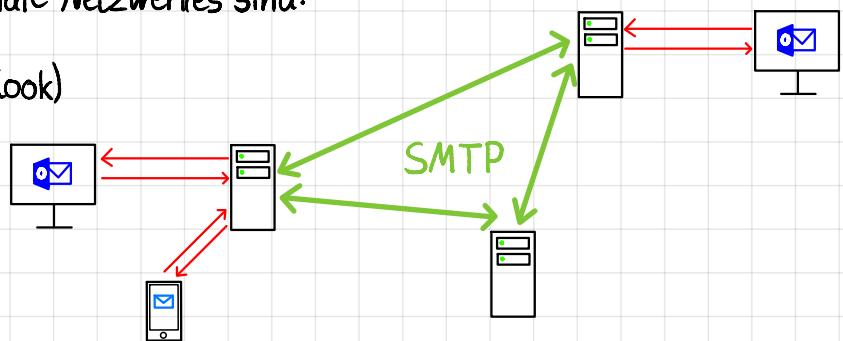
SSH: Die Secure Shell ist eine sichere Alternative zu Telnet

- ↪ Fernzugriff auf andere Dienste

Emailing:

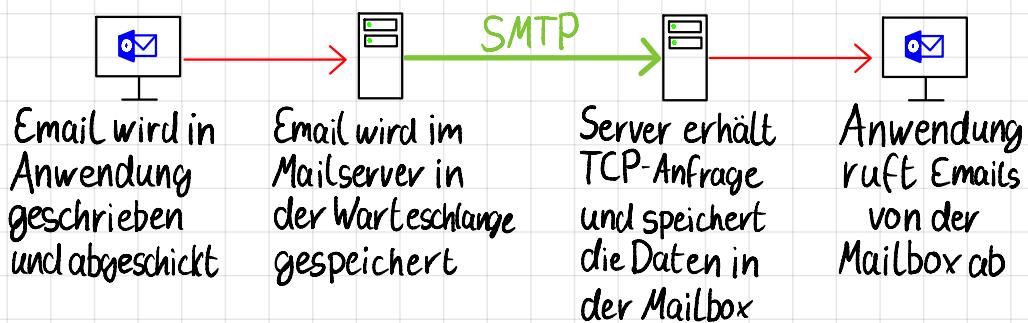
Die 3 Hauptbestandteile eines Email Netzwerkes sind:

1. Anwendungsprogramm (z.B. Outlook)
2. Mailserver
3. SMTP

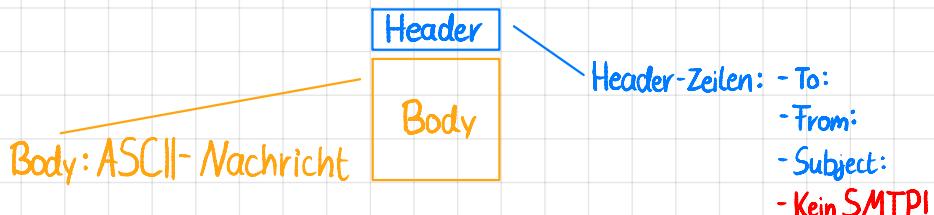


SMTP: Das simple mail transfer protocol dient zum versenden von Emails

- ↳ 3 Phasen: Handshaking, Transfer, Closure



Email:



Mailzugriff: - Pop3: Zustandsloses Zugriffsprotokoll

- ↳ "Download-and-Delete"-Modus: Mails werden nach Zugriff gelöscht
- ↳ "Download-and-Keep"-Modus: Mails werden nach Zugriff behalten

- Imap: Zustands Zugriffsprotokoll

- ↳ Alle Daten bleiben auf dem Server
- ↳ Verwaltung von Ordnern möglich
- ↳ Bewahrt Zustand: Namen und Zuordnungen bleiben

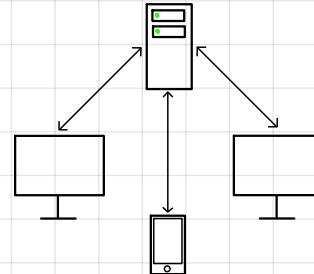
OSI Layer 4: Transport:

Transportsschicht:

- Logische Kommunikation zwischen Endprozessen
- Adressierung der Prozesse mittels Ports
- Unterteilung in Datensegmente

Client-Server-Architektur:

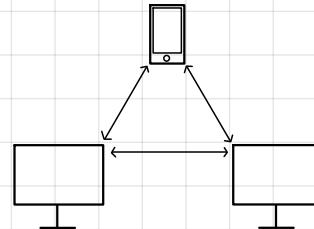
1. Server:
 - Feste Adresse (IP)
 - Durch Serverfarmen skalierbar
 - Hostserver muss online sein



2. Clients:
 - Temporäre Verbindung mit Host
 - Keine Kommunikation untereinander

Peer-to-Peer-Architektur:

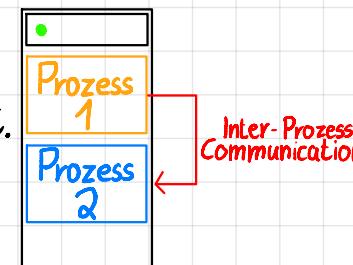
1. Server:
 - Keine Server



2. Clients:
 - Beliebige Clients kommunizieren
 - Peers nur sporadisch
 - Temporäre Adressen

Prozesskommunikation:

1. Prozess: Programm, welches auf einem Host läuft.

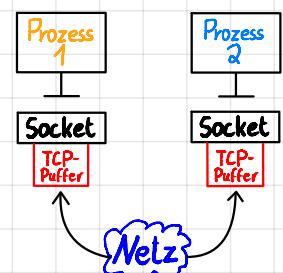


2. Client-Prozess: Prozess, der die Kommunikation beginnt.

3. Server-Prozess: Prozess, der darauf wartet, kontaktiert zu werden.

4. Sockets: Prozesse senden/empfangen Nachrichten über einen Socket.

- Sockets bestehen aus IP-Adresse und Portnummer
- Stream Sockets für TCP-Verbindungen
- Datagramm Sockets für UDP-Verbindungen



5. Protokolle:

1. Art: Request, Response
2. Syntax: Inhalt, Trennung
3. Semantik: Logische Informationen

Transportdienst: Einen solchen Dienst wählt man mit folgenden Kriterien:

| | Datenverlust | Zeitanforderung | Bandbreite |
|--------------------------|----------------------------|---------------------|------------|
| - Toleranz bei Datenver. | - Toleranz bei Verzögerung | - Keine Vorgaben | |
| - Lückenloser Transfer | - Sehr geringe Latenz | - Mindestbandbreite | |
| Dateitransfer | lückenlos | Toleranz | Toleranz |
| E-Mail | lückenlos | Toleranz | Toleranz |
| Web | lückenlos | Toleranz | Toleranz |
| Internettelefonie | Toleranz | geringe Latenz | >1 MBPS |
| Streaming | Toleranz | wenig Latenz | >5 MBPS |
| Multiplayer | Toleranz | geringe Latenz | >10 KBPS |
| Instant Messaging | lückenlos | wenig Latenz | Toleranz |

TCP/UDP:

| TCP | UDP |
|--|--|
| <ul style="list-style-type: none"> - Verbindungsorientiert - Zuverlässiger Transport - Überlastungsschutz (Flusskontrolle) - Keine Bandbreitengarantien - Keine Zeitgarantien | <ul style="list-style-type: none"> - Verbindunglos - Besserer Throughput - Keine Flusskontrolle - Keine Bandbreitengarantien - Keine Zeitgarantien |

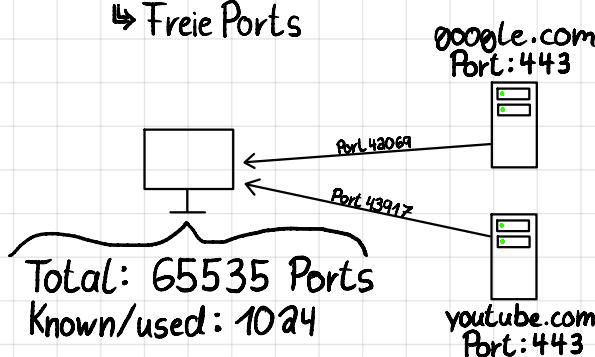
Eigenschaften:

- Garantierte Datenübertragung
- Korrekte Reihenfolge von Daten
- Transport beliebig großer Daten
- Kontrolle des eigenen Datenflusses $\hat{=}$ Flusskontrolle
- Kontrolle des Sendeverhaltens durch Empfänger $\hat{=}$ Überlastkontrolle

Ports:

- Anwendungen mit TCP/UDP haben eine Portnummer
- Portnummer ist 16Bit groß
- Freie Portwahl möglich
- Mittlerweile "quasi-standard":
 - Port: 0-1023 (well known ports)
 - ↳ Feste Zuordnung
 - Port: 1024-49151 (registered ports)
 - ↳ Reservierbare Ports
 - Port: 49152-65535 (private ports)
 - ↳ Freie Ports

| Well Known Ports | |
|------------------|--------|
| 20 | FTP |
| 21 | SFTP |
| 22 | SSH |
| 23 | Telnet |
| 25 | SMTP |
| 53 | DNS |
| 80 | HTTP |
| 443 | HTTPS |



Vermittlungskonzepte:

1. Leitungsbasierte Vermittlung

- ↳ Pro Verbindung eine Leitung
- ↳ Beispiel: Telefonnetz

Leitungsbasiert

- + Zuverlässig
- + Abbruchsfrei
- Ineffizient

Packetbasiert

- + Mehrere Verbindungen
- + Volle Bandbreite
- Überlastung möglich

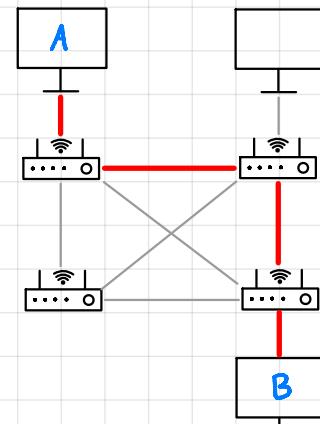
2. Packetbasierte Vermittlung

- ↳ Datenpakete teilen sich eine Leitung
- ↳ Beispiel: Postnetz

Leitungsvermittlung:

Endressourcen (A/B) werden für eine Verbindung reserviert.

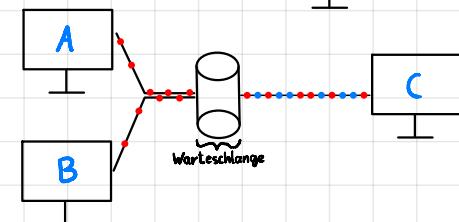
- ⇒ Voraussetzung zur Datenübertragung
- ⇒ Eine Verbindung verwendet alle Ressourcen



Paketvermittlung:

Aufteilung von Datenströme in Pakete.

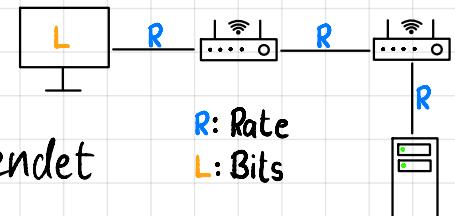
- ⇒ Pakete verwenden volle Bandbreite
- ⇒ Ressourcen nach Bedarf verwendbar
- ⇒ Wettbewerb um Ressourcen
 - ↳ Nachfrage > Angebot $\hat{=}$ Überlast



Store and Forward:

Pakete werden Zwischengespeichert und bei verfügbaren Ressourcen weitergeleitet

- ⇒ Pakete werden ausschließlich komplett gesendet
- ⇒ Übertragungsdauer pro Paket: L/R



Fazit: Paketvermittlung für unregelmäßigen Datenverkehr
 ↳ Bei Überlast: Verzögerung und Verlust von Paketen

Kommunikationsmodi:

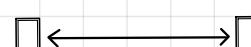
1. Simplex: Einseitige Übertragung



2. Half-Duplex: Beidseitige Übertragung



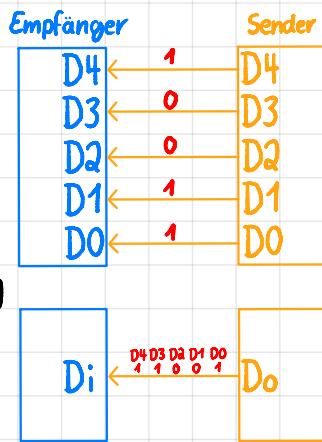
3. Duplex: Gleichzeitige beidseitige Übertragung



Datenübertragung:

1. Parallel: Übertragung über mehrere Datenleitungen

- ↳ Hoher Datendurchsatz
- ↳ Hohe Ressourcenkosten

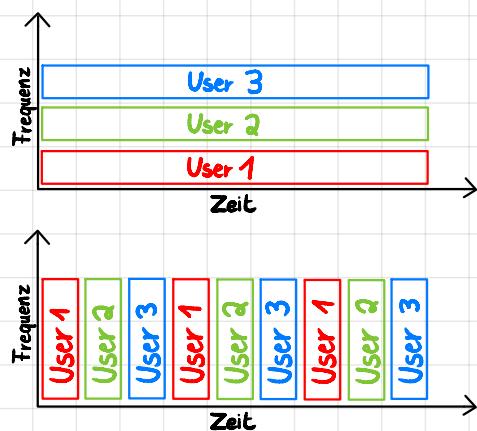


2. Seriell: Übertragung nacheinander über eine Datenleitung

- ↳ Niedriger Datendurchsatz
- ↳ Niedrige Ressourcenkosten

Multiplexing:

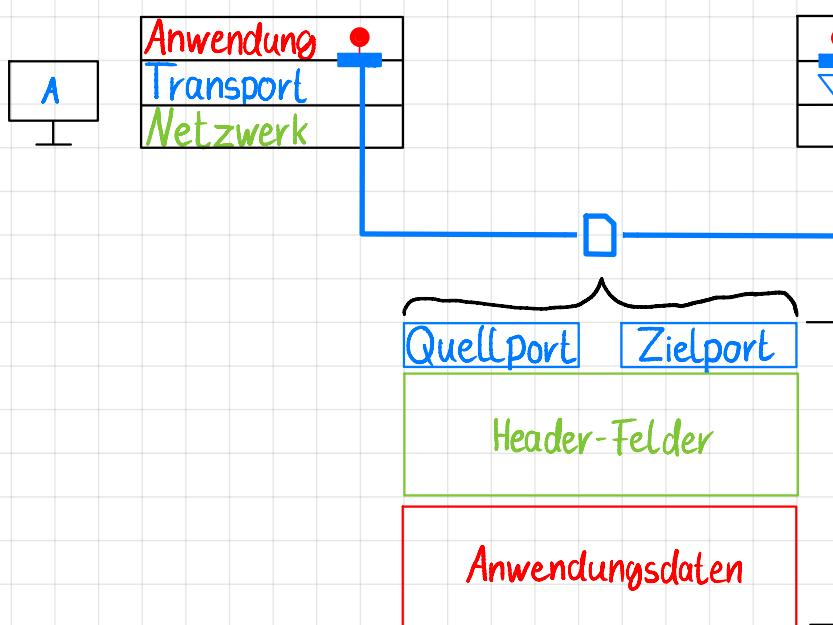
Aufteilung der Bandbreite durch zeitliche / physische Sequenzierung.



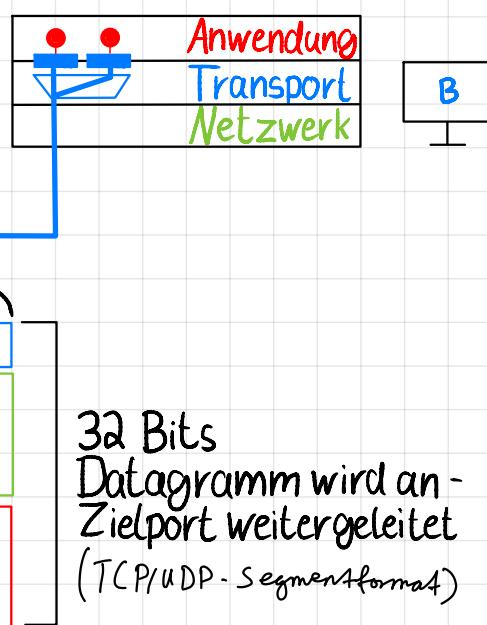
Aufteilung in verschiedene Frequenzen (FDM)

Aufteilung in Zeitabschnitte (TDM)

Multiplexing beim Sender



Demultiplexing beim Empfänger

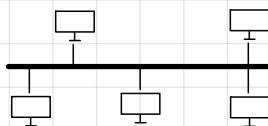


32 Bits
Datagramm wird an-
Zielport weitergeleitet
(TCP/UDP-Segmentformat)

- TCP-Socket durch 4-Tupel identifiziert: Absender-IP/Port und Empfänger-IP/Port
- Geräte können viele TCP-Sockets gleichzeitig offen haben
- Webserver haben verschiedene Sockets für jeden einzelnen Client

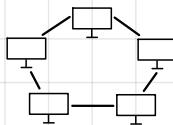
Netzwerktopologien:

1. Bus-Netz:



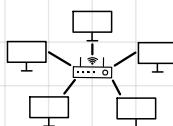
- Alle Hosts über gemeinsamen Bus verbunden
- Alle Hosts haben Zugriff auf den Bus
- + Geringe Kosten; Knotenunabhängig
- Fehleranfällig; Teilen der Bandbreite; Nur ein Host kann senden;

2. Ring-Netz:



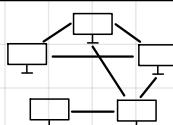
- Alle Hosts über geschlossene Kabelstrecke verbunden
- Verbindung in eine Richtung
- + Verstärkung des Signales; Regeneration möglich;
- Jeder Host essenziell; Teilen der Bandbreite;

3. Stern-Netz:



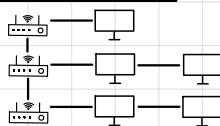
- Alle Hosts über einen gemeinsamen Knoten verbunden
- In Heimnetzwerken mit Router / Switch realisiert
- + Übersichtlich;
- Abhängigkeit durch eine Komponente; Überlastung;

4. Maschen-Netz:



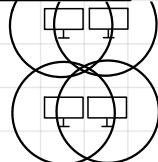
- Vollvermaschung: Jeder Host ist miteinander verbunden
- Teilvermaschung: Niedrig ausgelastete
- + Versorgungssicherheit; Geringe Verluste;
- -

5. Baum-Netz:



- Eine Wurzelverbindung und viele Äste
- Verbindung von vermaschten Netzen
- + Geringe Kosten;
- Teilen der Bandbreite; Ausfall bei Unterbrechung;

6. Zell-Netz:



- Drahtlose Datenübertragung
- Jede Zelle erweitert das Basisnetzwerk
- + Keine Leitungen; Keine Störung bei Endgerätausfall;
- Reichweite begrenzt;

Veraltete Netze:

- Bus-Netze: Thick-Ethernet
- Ring-Netze: Token-Ring

Verwendete Systeme:

- Teil- und Vollvermaschte Netze: Funknetze / Logische Verbindung von Routern
- Zell-Netze: Funknetz
- Stern-/Baum-Netze: Ethernet-Standard

Ausdehnen von Netzen:

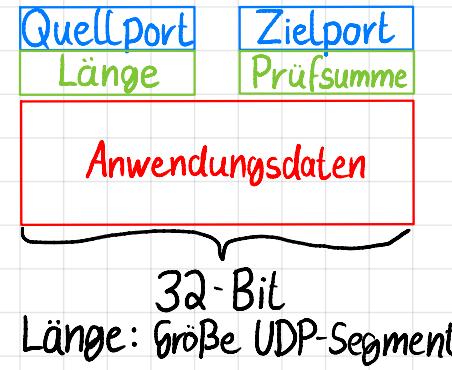
1. Personal Area Network (PAN): 100m Ausdehnung (Bluetooth)
2. Local Area Network (LAN): 10 km Ausdehnung (In Gebäuden)
3. Metropolitan Area Network (MAN): 50km Ausdehnung (Stadt)
4. Wide Area Network (WAN): >10km (Campus-Netze)
5. Global Area Network (GAN): Interkontinental (Internet)

Verbindungsloser Transport:

- Minimales Internet - Transportprotokoll
- "Best-Effort" Dienst:
 - Segmente können verloren gehen
 - Es kann eine falsche Reihenfolge entstehen

- Verbindungslos:
 - Kein Handshake notwendig
 - Jedes Segment unabhängig

- Vorteile:
 - Kein Verbindungsaufbau
 - Kein Verbindungszustand
 - Kleiner Header
 - Keine Überlastkontrolle \Rightarrow keine Begrenzung



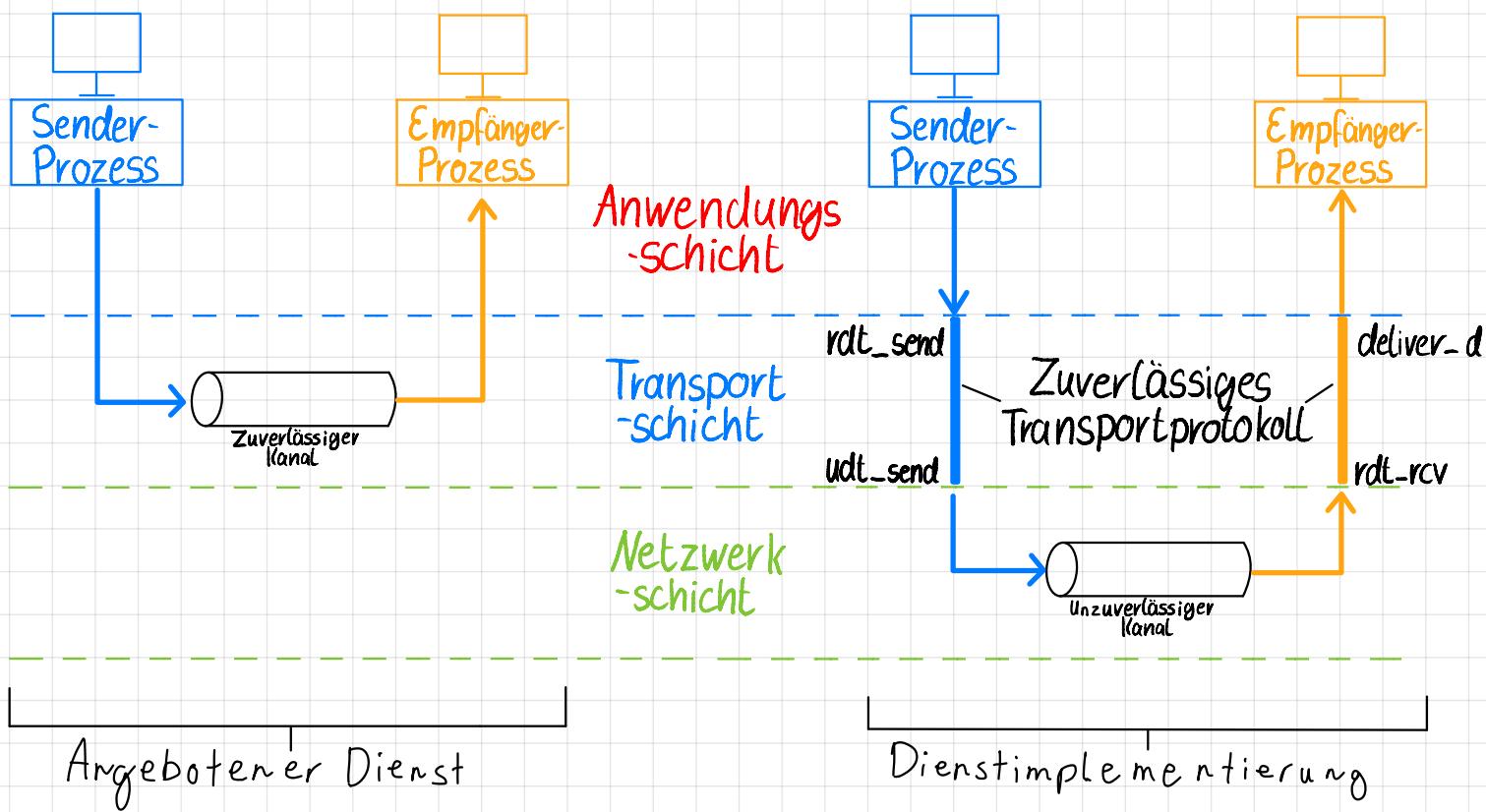
UDP-Prüfsumme: Bitfehler erkennen

- ↪ Summe aller 16-Bit-Wörter (Überlauf zum 1. Bit addiert)
- ↪ Bitsumme + Prüfsumme = 16×1

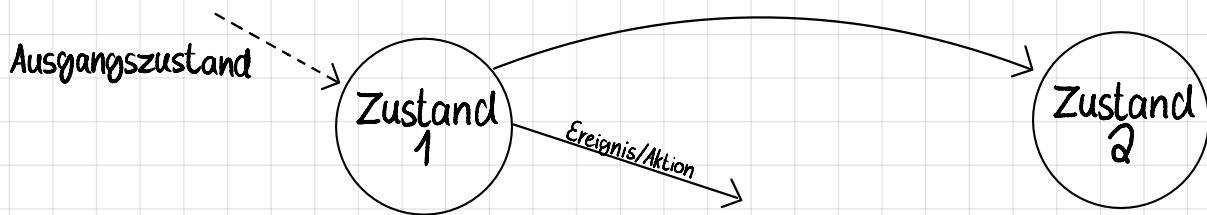
| | |
|-----------|---------------------------------------|
| Übertrag | 1 1 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 |
| Summe | 1 1 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 |
| Prüfsumme | 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1 1 |

Zuverlässige Datenübertragung:

- Wichtig für Anwendungs-, Kommunikations- und Sicherungsschicht
- Eigenschaften des unzuverlässigen Kanals bestimmen Komplexität (rdt)



Schnittstelle:

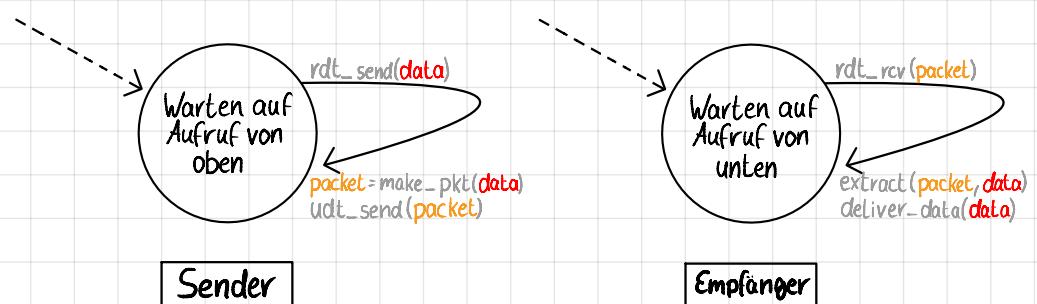


Perfekter zuverlässiger Kanal (rdt 1.0):

- Keine Verfälschung von Bits
- Keine Verlust ganzer Rahmen/Pakete
- Sender übergibt Daten an den zuverlässigen Kanal
- Endliche Automaten von Sender und Empfänger sind unabhängig
- Für jeden Zustand wird auslösendes Ereignis und die ausgelösten Aktionen aufgeschrieben

Sender: Ereignis → akzeptiert Daten der darüberliegenden Kanal (`rdt(data)`)
 Aktion → erzeugt Paket mit den Daten und sendet es in den Kanal (`packet = make(packet); udt_send(packet)`)

Empfänger: Ereignis → erhält Paket von darunterliegenden Kanal (`rdt_send(data)`)
 Aktion → entfernt die Daten aus dem Paket und reicht sie an die darüberliegenden Schicht weiter (`extract(packet, data); deliver_data(data)`)



| RDT 2.0 | RDT 2.1 | RDT 2.2 |
|--|--|--|
| <ul style="list-style-type: none"> ⇒ Gegen Bitfehler + Fehlererkennung mit Prüfsumme + Kontrollnachrichten mit ARQ-Protokoll (Automatic Repeat) reQuest | <ul style="list-style-type: none"> ⇒ Gegen verfälschte Ack's + Sequenznummern <ul style="list-style-type: none"> Stop-and-wait: 2 Zustände Bei anderen Methoden wie z.B. Pipelining beliebig viele Zustände | <ul style="list-style-type: none"> ⇒ Eliminierung von Naks - Naks + Nur korrekt empfangene Packete ⇒ Ack + Fehlerhafte Packete ⇒ Keine Rückmeldung |

Zuverlässiger Datentransfer (RDT 3.0):

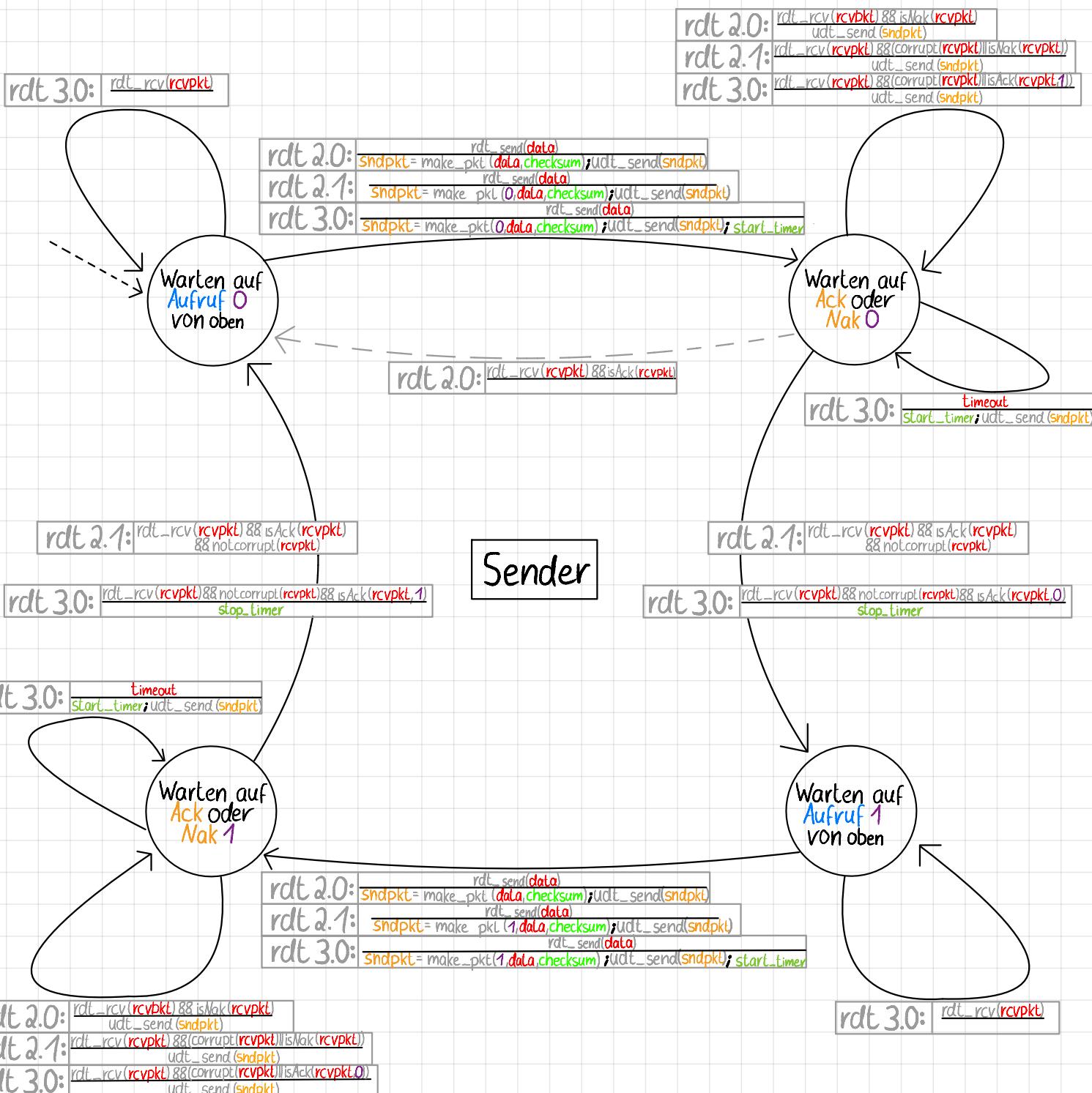
- Besitzt alle vorherige Funktionen:
 - 2.0: - Fehlererkennung
 - Kontrollnachrichten
 - 2.1: - Sequenznummern
 - 2.2: - Nur Ack's

+ Prüfung auf vollständig verlorene Pakete

Kein Ack: Wiederholung der Übertragung

Ack verzögert: Duplicat (gleiche Sequenznummern)

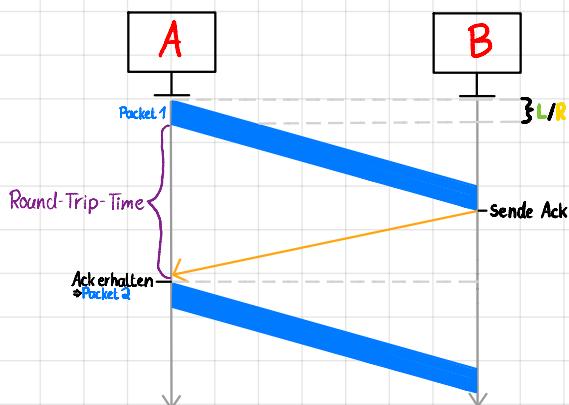
⇒ Timer benötigt



Pipelining:

- Fall: System mit 1Gbit/s Link, 15ms Verzögerung, 8000Bit Pakete
- R: Übertragungsrate (Bit/s)** **Round-Trip-Time** **L: Paketgröße (Bit)**

Stop-and-wait



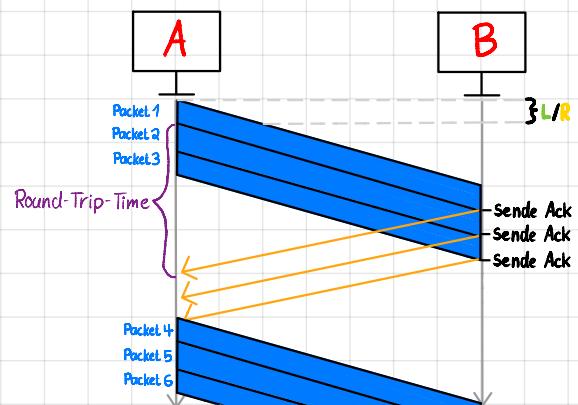
$$\text{Übertragung: } T_A = \frac{L}{R}$$

$$T_A = \frac{8000 \text{ Bit}}{10^9 \text{ Bit/s}} = 0,008 \text{ ms}$$

$$\text{Auslastung: } U_A = \frac{L/R}{RTT+L/R}$$

$$U_A = \frac{0,008 \text{ ms}}{30,008 \text{ ms}} = 0,00027 \text{ ms}$$

Pipelining



$$T_A = 3 \cdot \frac{8000 \text{ Bit}}{10^9 \text{ Bit/s}} = 0,024 \text{ ms}$$

$$U_A = \frac{0,024 \text{ ms}}{30,008 \text{ ms}} = 0,0008 \text{ ms}$$

Go-Back-N:

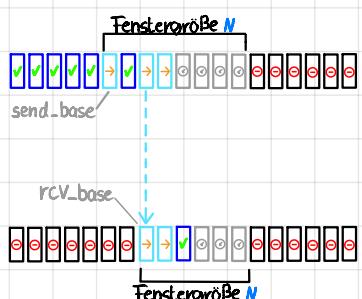
- Sender darf bis zu **N** Pakete ohne Bestätigung (Ack) senden



- Fenstergröße **N**: Anzahl der zugelassenen unbestätigten Pakete
- Timer: Bei jedem unbestätigten Packet
- Timeout: Alle verlorenen Pakete werden neu übertragen
- Empfänger:

Selective Repeat:

- Empfänger bestätigt jedes Packet einzeln. Sender schickt nur einzelne Pakete wiederholt.

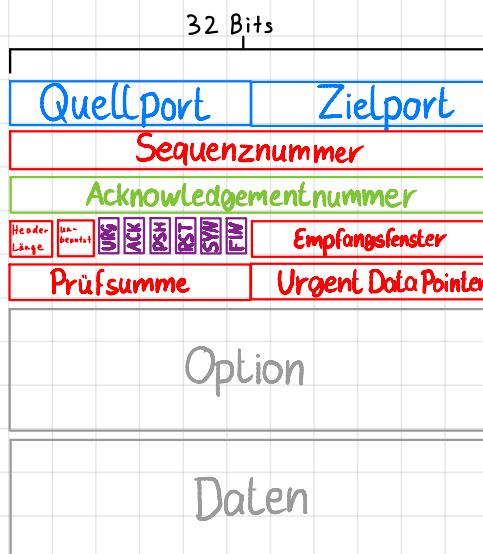


- Sender:**
 - Nächste Sequenznummer: Packet senden, Timer
 - Timeout: Wiederholte Übertragung

- Empfänger:**
 - Bestätigt jedes einzelne Packet
 - Puffert korrekt empfangene Daten
 - Liefert Pakete (wenn möglich) in der richtigen Reihenfolge

Transmission Control Protocol (TCP):

- Verbindungsorientiert: „Datenübertragung aushandeln“
- Ausschließlich auf Endsystemen
- Nur zwischen 2 Geräten möglich



Header-Länge & unbenutzt je 8 Bits

URG: Urgent Data (nicht benutzt)

PSH: Push - Daten nach oben (nicht benutzt)

RST/SYN/FIN: Befehle zum Verbindungsaufl/abbau

Empfangsfenster: Akzeptierte Anzahl an Bytes

Seq/Ack: Zählen in Bytes

Pro Bit

Zeitraum bis Timeout: - Zu kurz: Unnötige Wiederholungen
- Zu lang: Langsame Reaktion

$$\text{Estimated RTT} = (1-\alpha) \cdot \text{Estimated RTT} + \alpha \cdot \text{SampleRTT}$$



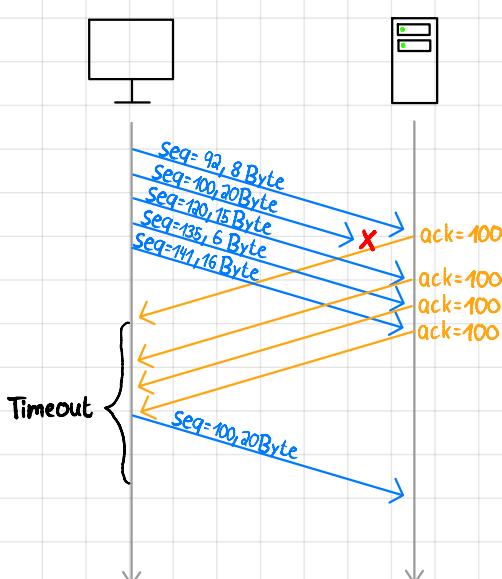
SampleRTT: Gemessene Zeit von SYN bis Ack
 α : meist 0,125

$$\text{DevRTT} = (1-\beta) \cdot \text{DevRTT} + \beta \cdot |\text{SampleRTT} - \text{EstimatedRTT}|$$

β : meist 0,25

$$\text{Timeout Interval} = \text{EstimatedRTT} + 4 \cdot \text{DevRTT}$$

TCP-Fast Retransmit:



- Timeoutintervall sehr lang
- Packetverlust: durch viele doppelte Ack's

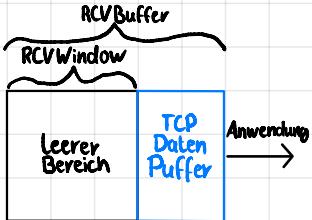
Bei 3 Duplikaten → Segment verloren

⇒ Fast Retransmit: Segment erneut schicken, bevor der Timer ausläuft

Flusskontrolle:

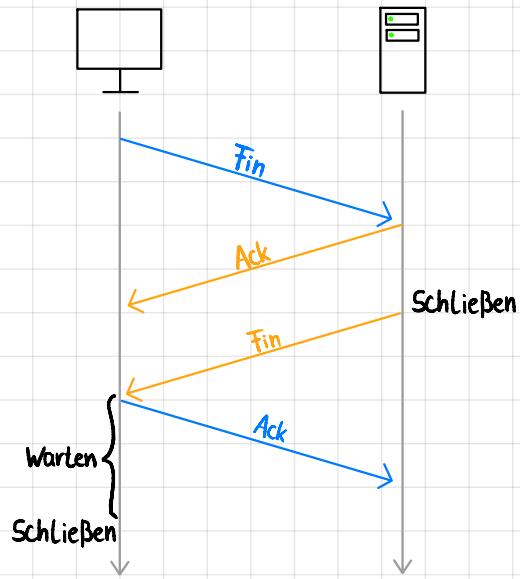
- Empfänger besitzt Empfangspuffer
- Sender sendet nicht mehr, als der Puffer speichern kann
- Senderrate gleicht sich an Verarbeitungsrate an
- Empfänger kündigt Daten auf RcvWindow im Header an
- Sender begrenzt unbestätigte Daten

Daten von IP



Close-Aufruf:

- Client und Server schicken jeweils ein „Fin“
- Das „Fin“ wird mit einem „Ack“ bestätigt



Überlastkontrolle:

- Problem: Sender ignorieren Netzauslastung \Rightarrow Überlast
- \Rightarrow Ende-zu-Ende-Ansatz: Überlast durch Packetverlust / erhöhte Verzögerung erkennbar (TCP im Internet)
- \Rightarrow Netzwerkunterstützter-Ansatz:
 1. Direkte Rückmeldung vom Routern (z.B. Choke-Packet)
 2. Empfänger informiert über Bit im Header (SNA, ATM)

OSI Layer 5: Netzwerk:

Funktionen:

1. Forwarding: Eingangsleitung → Router nimmt Packet entgegen
Ausgangsleitung → Router bestimmt Ausgang anhand von lokalen Informationen (z.B. Routing Tabellen)
2. Routing: - Router kommunizieren, um geeignete Wege im Netzwerk zu finden
- Infos über Erreichbarkeiten werden gesammelt
3. Verbindungen in bestimmten Architekturen:
 - Virtuelle Verbindung zwischen Sender und Empfänger zur Datenübertragung
 - Verbindung in Transport- (2 Prozesse) und Netzsicht (2 Hosts)
 - Anforderungsbasierte Dienstmodelle:
 1. Garantierte Zustellung
 2. Garantierte Zustellung in 40ms
 3. Reihenfolgengehaltende Zustellung
 4. Garantierte minimale Datenrate

Netzwerkschicht = Pfad zwischen Sender und Empfänger:

1. Aufbau der Verbindung
2. Datentransport
3. Abbau der Verbindung

- Ein Datagrammnetzwerk verwendet eine verbindungslose Netzwerkschicht
- Ein Netzwerk mit virtuellen Leitung ⇒ verbindungsorientiert
- In höheren Schichten kann verbindungsorientiertes TCP über eine verbindungslose Vermittlungsschicht (IP) realisiert werden

Virtual Channel Identifier:

- Jedes Packet besitzt einen VC-Identifier
- Jede Leitung besitzt:
 1. einen Pfad vom Sender zum Empfänger
 2. VC-Identifier: Für jeden Link des Pfades
- Packete mit VC-Identifier besitzen keine Zieladresse

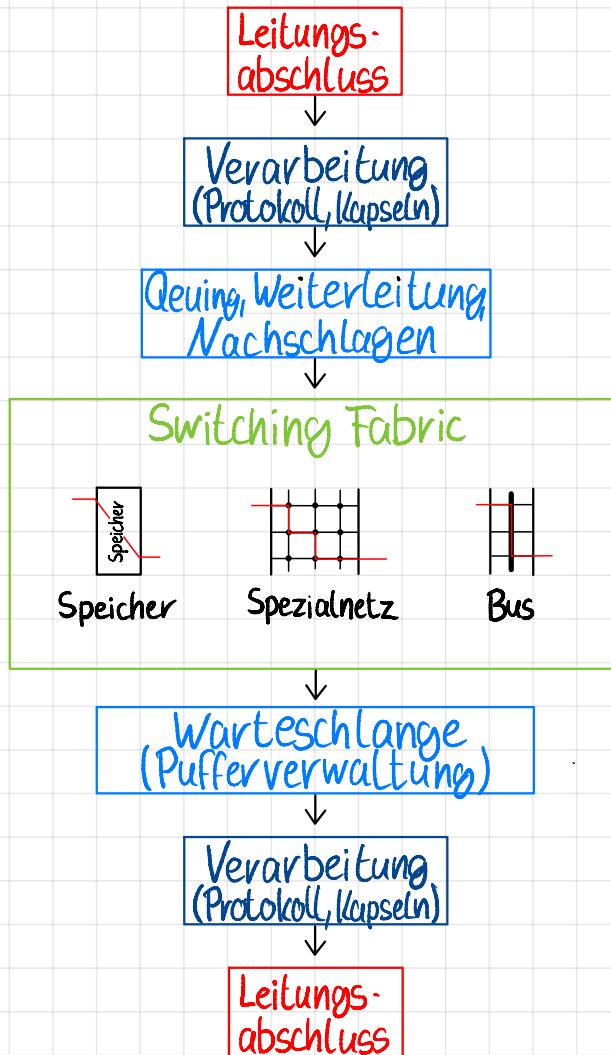
Internet:

- Datenaustausch zwischen Computern
- Last bei Endsystemen: Fehlerbehebung ⇒ einfaches Netzwerk
- Vielzahl von Links mit unterschiedlichen Eigenschaften
⇒ einheitlicher Dienst schwierig

Asynchronous Transfer Mode (ATM):

- Stammt von Telefontechnologie ab
- Menschliche Kommunikation \rightarrow Hohe Anforderungen an Echtzeit/Zuverlässigkeit
- Last im Netzwerk: Einfache Endsysteme

Router: 1. Ausführen von Routing-Algorithmen / -Protokollen (RIP, OSPF, BGP)
2. Weiterleiten von Datagrammen



Layer 1 - Bitübertragung: Bits empfangen

Layer 2 - Sicherung: z.B. Ethernet

Layer 3 - Netzwerk:

- Suche nach geeignetem Port
- Dezentrale Routing-Tabelle
- Schnellstmögliche Verarbeitung
- Puffern, wenn Switch bewegt

Layer 4 - Transport:

- Routing zum Ausgangsport

Layer 3 - Netzwerk:

- Puffert Überlast an Paketen
- Bei Überlauf: verwerfen

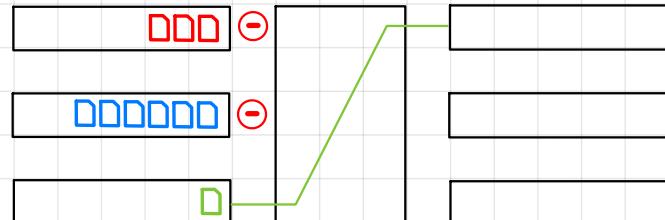
Layer 2 - Sicherung:

- Verkapselung

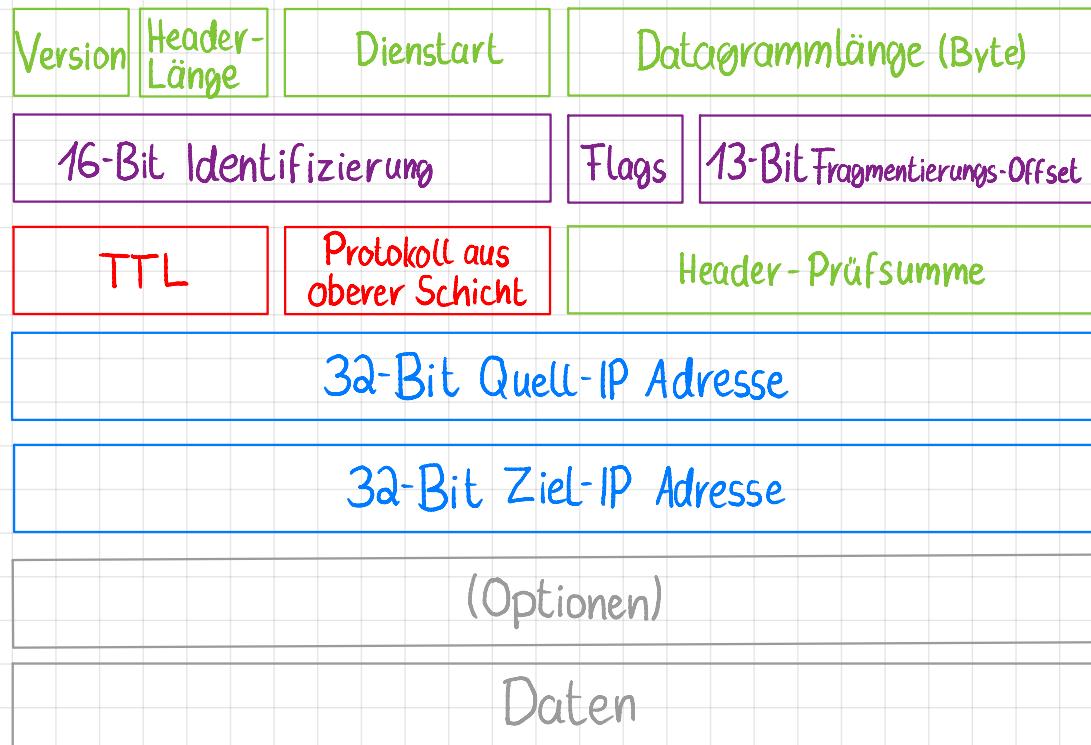
Layer 1 - Bitübertragung: Bits senden

- Puffergröße: $\frac{\text{Round-Trip-Time} \cdot \text{Datenrate} (c)}{\text{Datenfluss} (m)}$

- Head-of-Line Blocking: Effiziente Pufferung



IPv4:



Version: IP-Protokoll

TTL: Anzahl der zu durchlaufenden Router

Protokoll: An welches Datagramme ausgeliefert werden

Optionen: Zeitstempel; Router verlauf...

IPv6:



Nutzdatenlänge: Länge der Ipv6-Payload in Oktetten

Nächster Header: Typ des folgenden Headers

Hop-Limit: Maximale Anzahl der zu durchlaufenden Knoten

Anwendungsschicht

- Daten und Informationen für Anwendungen
- Anwendungsspezifische Protokolle

DNS; DHCP; SMTP; SSH

Präsentationsschicht:

- Systemunabhängige Darstellung von Daten
- Datenkompression / Verschlüsselung

Jpeg; MIDI; Mpeg

Sitzungsschicht:

- Auf-/Abbau und Überwachung einer Sitzung
- Dialogkontrolle
- Synchronisierter Datenaustausch (Kontrollpunkte)

SQL

Transportschicht:

- Logische Kommunikation zwischen Anwendungen verschiedener Hosts
- Verpackt Daten in Segmente und sendet diese über die Vermittlungsschicht

 TCP / UDP
Gateway

Netzwerk / Vermittlungsschicht:

- Implementiert Host zu Host Verbindung
- Verpackt Daten aus höherer Schicht in Datagramme
- Transport über das Netzwerk

 IP; RIP; ICMP
Router

Sicherungsschicht:

- Transport von Datagrammen mittels Pakete
- Prüfung auf Bitübertragungsfehlern
- Protokolländerungen

 PPP; Token Ring
Switch

Bitübertragungsschicht:

- Physische Übertragung von Daten

 Ethernet
Repeater