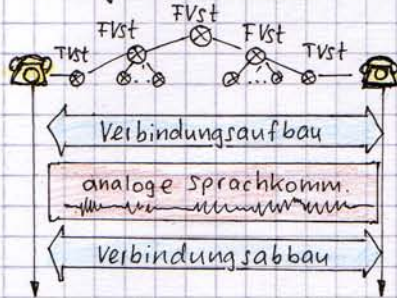


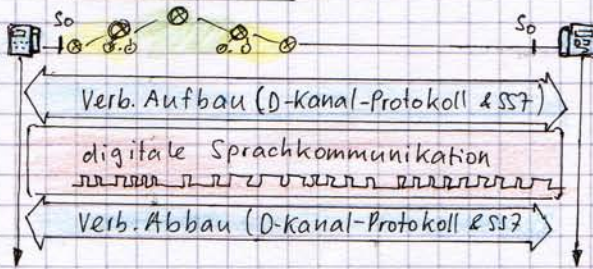
Analoge Telephonie → Buch S. 4



- Verbindung muss auf- und wieder abgebaut werden
- Verbindung ist privat, d.h. zuverlässig, sicher
- Sprache und Daten über die selbe Leitung → Inband-Signalisierung
- Baumartige / Hierarchische Struktur

öffentliches Telephonnetz: PSTN (Public Switched Telephone Network)

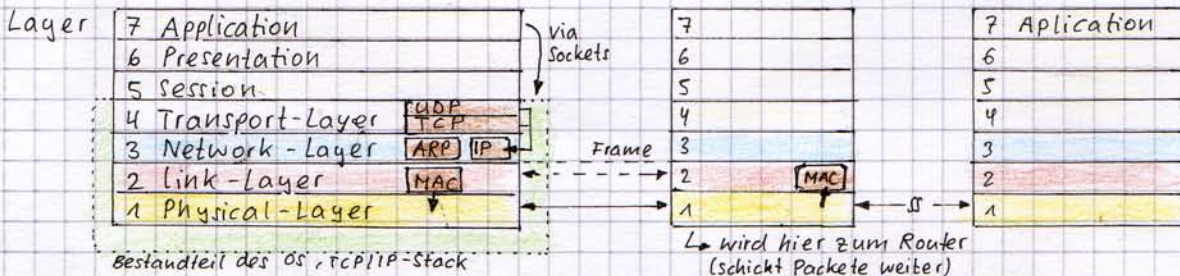
ISDN-Telephonie → Buch S. 5



- alle ISDN Endgeräte besitzen S0 Schnittstelle mit:
  - 2 Nutzkanäle (B-Kanäle) à je 64 kbits/s
  - 1 Kanal für Signalisierung (D-Kanal) à 16 kbits/s
- Verbindung via D-Kanal-Protokoll

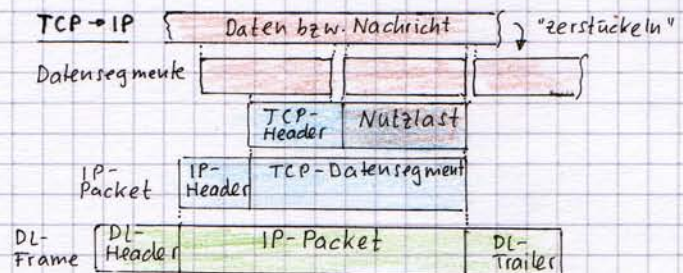
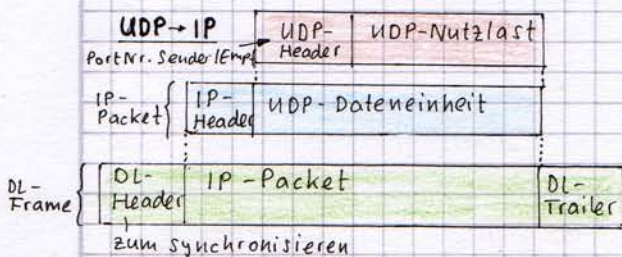
■ typisch Q.931 Protokoll ■ typisch SS7 (Pakete!)  
 Aufbau: Setup → Setup Ack  
 Abbau: Dis → REL → REL COM

Protokollfamilie TCP/IP → Buch S. 68

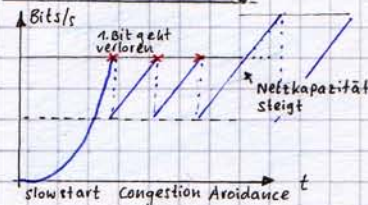


- HUB: nur auf dem physical-Layer
- SWITCH: auf physical und link-Layer
- Router: auf phys. / link / Network-Layer
- ARP: Adress Resolution Protokoll → löschen: Arp -d  
→ sucht physikalische Rechner Adresse im Netzwerk
- TCP: Transmission Control Protokoll → stellt virtuelle Verbindungen her, ist verbindungsorientiert  
→ sichere Datenübermittlung Bsp: FTP, HTTP, SMTP, H.323, DNS
- UDP: User Datagram Protokoll → verbindungslos ohne virtuelle Verbindung → nicht sicher Bsp: DHCP, RTP, SIP, DNS  
DNS einträge löschen: flushdns
- Physical-Layer: zuständig für das Schicken/Empfangen von Bits
- Link-Layer: -schicht lempfängt Frames (mehrere Bits) -ist zuständig, dass Device senden lempfängen kann (einen slot kriegt → MAC)
- Network-Layer: ist zuständig für den Datenverkehr ausserhalb des eigenen sub-Netzes

Bildung von IP-Paket → Buch S. 71



TCP-Verbindung



QoS (Quality of Service), Anforderungen → S. 99

- Zeitabstände zwischen aufeinander folgenden Packete muss auf Sende- und Empfangsseite identisch sein.  
 → Isochronität (isochrone Kommunikation) → kein Jitter!  
 garantieren von: - geforderter Bandbreite  
 - max. Übermittlungszeit  $t_u$   
 bieten von: - möglichst wenig Übermittlungsfehler (Packet loss Rate)  
 - geringe Schwankung von  $t_u$  (Jitter)  
 - kleine End-zu-End Verzögerung  $T_{EE}$

- Konzepte: - Diffserve (Differentiated Services)  
 - Management von Warteschlangen  
 - RSVP (Resource reSerVation Protocol)  
 → IP-Sprachpackete bevorzugen!

End-zu-End Verzögerung  $T_{EE}$  → S. 101

Entsteht durch Zwischenspeicherung der IP-Pakete in den Routern (Warteschlangen).

- ITU-T: •  $T_{EE} < 150ms$ : akzeptabel für alle Benutzer  
 •  $150ms < T_{EE} < 300ms$ : eingeschränkt akzeptabel  
 •  $T_{EE} > 300ms$ : nicht akzeptabel

'Budget' von  $T_{EE}$ : max. zulässiger Wert für  $T_{EE}$

$T_{EE}$  = Zeit vom Aussprechen des Signals bis zum Hören

Verzögerungen, die  $T_{EE}$  bestimmen

$T_{seg}$ : Der Encoder muss warten, bis er Sprachsegmente der Länge  $T_{seg}$  vorliegen hat (z.B. 20ms)

$T_{look-ahead}$ : Benötigte Zeit, um das Segment zu codieren und daraus ein Packet zu bilden.

TCP:  $T_{cp} = T_{seg} + T_{look-ahead}$  (codierungs- und Packetsierungszeit)

$D_0$ : Zeit um ein Packet zu serialisieren  
 $D_0 = \text{Packetgrösse [Bit]} / \text{Übertragung [Bit/s]}$

$T_0$ : Zeit, um Packet zum Router zu schicken.

$\tau_i$ : zufällige Wartezeit in Warteschlange  $i$

$D_1$ : Zeitverlust beim schicken durch den Router

$T_1$ : Zeit, um Packet vom Router zum Ziel zu schicken.

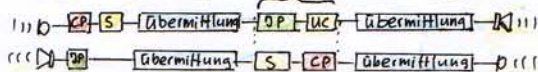
$t_u$ : Übermittlungszeit,  $t_u = \sum \tau_i + \sum D_j + \sum T_k$

wobei  $\tau_i$ : Zeiten bei Zwischenspeicherung  
 $D_j$ : Zeiten bei Serialisierung (auch im Router)  
 $T_k$ : Zeiten der Übertragung (Laufzeit: 5µs/km)

Tip: Zeit im Jitter-Ausgleichsbuffer (oft:  $T_{jp} = 2 \cdot T_{seg}$ )

RSVP (Resource reSerVation Protocol) → S. 126

- 'Path' Nachricht wird ausgeschildet und jeder Router trägt seine IP-Adresse im Header ein.
  - Am Ziel wird 'Path' kopiert, und als 'Resv' in der umgekehrten Reihenfolge zurückgeschickt → Reservierung
  - jeder Router checkt, ob er die in 'R' geforderte Bandbreite garantieren kann. ggf. muss er 'R' anpassen. Danach reserviert er die Bandbreite.
- Die 'optimale' Route kann sich ständig ändern.  
 Darum wird die Reservierung zyklisch neu gestartet.  
 → RSVP wird wenig benutzt.



Kommunikation von Computer mit (IP) Telefon.

CP: Codierung und Packetsierung

JP: Jitter-Ausgleichsbuffer

S: Sender

UC: Umcodierer

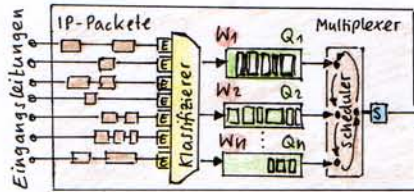
Bei reiner Computer zu Computer Kommunikation kann das obige Bild auf den linken Teil reduziert werden.

Queue-Management → S. 114

Priority Queueing → S. 116

- eintreffende Packete werden i.d.R. in drei Warteschlangen verschiedenen Prioritäten eingereiht.
- Scheduler bedient zuerst die höher priorisierten Warteschlangen, und erst wenn diese leer sind die nächst wichtige.  
 → Sprachdaten z.B. wird schneller behandelt als HTML-Daten.  
 → es kann sein, dass Warteschlangen tiefer Priorität nie bedient werden!

Custom Queueing → S. 117



→ hauptsächlich in Systemen von Cisco  
 → Scheduler arbeitet nach dem Round Robin-Prinzip.

$Q_1 = 0.5$   
 $Q_2 = 0.25$   
 $Q_3 = 0.25$

E: Empfänger Q: Warteschlange  
 S: Sender W: Gewichtung [%]

- jeder Warteschlange  $Q_i$  wird ein Gewicht  $W_i$  zugeordnet.
- Scheduler verschickt bei jedem Durchlauf von jeder Warteschlange eine von  $W_i$  abhängige Anzahl Pakete  $P_i$
- $W_i$  ist der Anteil der geforderten Bandbreite in %

mittlere Packetlänge in allen Warteschlangen gleich:

- $P_i$  berechnen:  $P_i = Q_i / \min(Q_i)$  (durch kleinstes  $Q_i$  teilen)
  - $Q_1: 0.5 / 0.25 = 2 = P_1$
  - $Q_2: 0.25 / 0.25 = 1 = P_2$
  - $Q_3: 0.25 / 0.25 = 1 = P_3$
- Anzahl der zu verschickende Packet pro Durchgang.

mittlere Packetlänge in Warteschlangen (klassen) verschieden:

- $L_1 = 500$  Bytes / Packet
  - $L_2 = 300$  Bytes / Packet
  - $L_3 = 100$  Bytes / Packet
- 1.)  $\max(L_i)$  teilen durch  $L_i$   
 $x_1 = 500 / 500 = 1$   
 $x_2 = 500 / 300 = 1.65$   
 $x_3 = 500 / 100 = 5$

2.) mit geforderter Bandbreite multiplizieren

- $y_1 = 0.5 \cdot 1 = 0.5$
- $y_2 = 0.25 \cdot 1.65 = 0.4175$
- $y_3 = 0.25 \cdot 5 = 1.25$

3.) normalisieren:  $y_i / \sum y_i$ , aufrunden.

- $P_1 = 0.5 / 0.4175 = 1.19 = 2$
- $P_2 = 0.4175 / 0.4175 = 1$
- $P_3 = 1.25 / 0.4175 = 2.99 = 3$

Bei gewissen Routern muss die Anzahl der Bytes angegeben werden, die pro Durchgang aus den einzelnen Klassen gesendet werden muss.

Fair Queueing → S. 120

- für jeden Packetstrom wird eine separate Warteschlange organisiert. Eintreffende Pakete werden aufgrund ihrer Eingangspriorität sortiert.
- Der Scheduler entnimmt nun pro Durchlauf von jeder Reihe ein Paket.
- Die Sortierung (Einreihung der Pakete) ist auch nach IP-Adr, Applikation o.ä. möglich.

Weighted Fair Queueing → S. 122

Wie Fair Queueing, aber Warteschlangen können gewichtet werden. Der Scheduler benutzt die Variable FT (finish time) um die Reihenfolge der Pakete zu bestimmen.

Class-based Weighted Fair Queueing → S. 124

Wie WFQ, aber IP-Pakete werden durch Angaben im Header (ToS, DS-Feld) differenziert / sortiert.

- Voice Daten z.B. hat anderer Header als HTML-Daten
- CBWFQ garantiert max. Verzögerung im Netzwerk
- CBWFQ ist sehr komplex und rechenintensiv  
 → extrem langsam, hat sich nicht durchgesetzt.

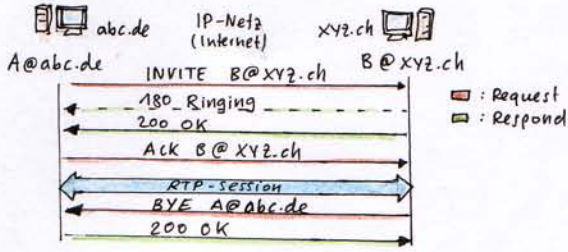
Strategien bei Engpässen

tail drop Moto: 'first in first out'

head drop ältestes Packet am wenigsten wichtig

RED (Random Early Drop) (Random Early Deletion) Router löscht mit wachsendem Buffer zufällige Pakete. Je grösser der Buffer, desto mehr Pakete werden gelöscht → verhindert "Pumpen"

SIP (Session Initiation Protocol)



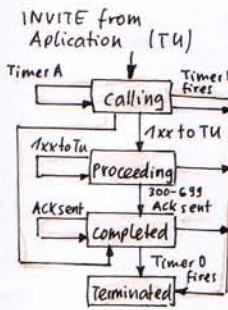
typische Messages (1xx sind optional)

- 100: Trying
- 200: OK
- 404: Not Found
- 180: Ringing
- 30x: Redirection
- 50x: Errors

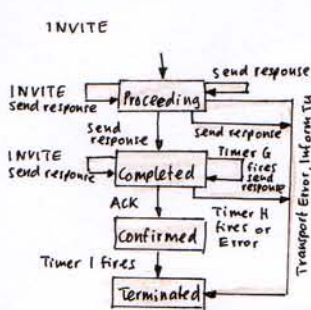
Transaktionstypen

- Client Invite (Client will Verbindung starten)
- Client non-Invite (Client will z.B. Gespräch beenden)
- Server Invite (Server erhält Invite Nachricht)
- Server non-Invite (Server erhält non-Invite Message)

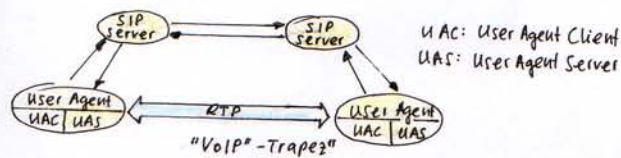
Client Invite



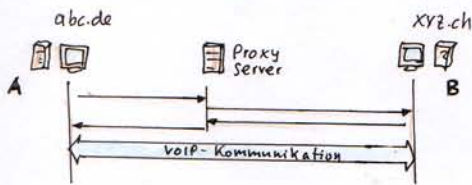
Server Invite



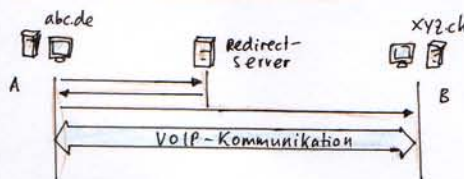
Folgt auf ein Request nicht innerhalb einer gewissen Zeit ein Respond (z.B. 100, 180) so wird Request neu verschickt und Timer neu gestartet (aber mit doppelter Länge) -> client wird weniger aggressiv.



Einsatz eines Proxy-Servers -> Buch S. 252



Einsatz eines Redirect-Servers -> Buch S. 253



Anrufweiterleitung mit SIP -> Buch S. 254

-> Teilnehmer B kann sich irgendwo auf der Welt anmelden, ist immer erreichbar

Anrufverzweigung -> Buch S. 257

-> Wenn Teilnehmer nicht unter standard-Adresse erreichbar ist, so umleiten auf z.B. PDA, Computer, ...

Struktur von SIP-Request -> Buch S. 262

Method Request-URI SIP-Version (Request-Line als Startzeile)  
 Header-Feld 1  
 ...  
 Header-Feld n } Message Header  
 (Leerzeile)  
 [Message Body]

INVITE sip: A@abc.de SIP/2.0 (Start-Zeile)  
 Via: SIP/2.0/UDP B@xy2.ch (Protokoll und Info des letzten Absenders -> wird immer größer!)  
 To: Teilnehmer B <sip: B@abc.de> (Benutzer als Ziel des Request)  
 From: Teilnehmer A <sip: A@abc.de> (Benutzer als Initiator des Request)  
 Call-ID: 1877352@abc.de (Sequenz Nr., aufsteigend)  
 Cseq: 1 INVITE  
 Max-Forwards: 10  
 Contact: <sip: A@abc.de>; expires=360 (Kontakt zum Initiator -> direkte Rückwahl-Adresse)  
 Content-Type: application/SDP (Typ des Body-Inhalt)  
 Content-Length: 152 (Länge des Body)  
 [ ] bleiben bei jedem Request/Response gleich!

SDP (Session description protocol) -> Buch S. 265

nicht so wichtig {  
 v=0  
 o=sonne 2890846521 2890846843 IN IP4 126.17.54.6  
 s=herzliche Grüsse  
 t=0 0  
 Port-Nr.  
 Payload Types  
 m=audio 3456 RTP/AVP 0 2  
 a=rtpmap: 0 PCM/8000 (1. Wahl)  
 a=rtpmap: 2 G721/8000 (2. Wahl)

anderes Bsp: m=video 4000 RTP/AVP 31 (31 = H.261)

Einige Payload Typen (PT)

0	PCMU	A (Audio)	12	G.722	A
8	PCMA	A			
26	JPEG	V (Video)	34	H.263	V
31	H.261	V			

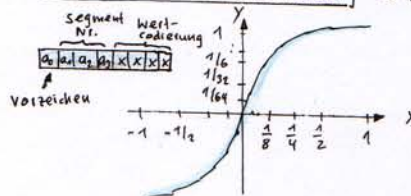


Digitale Sprachcodierung -> Buch S. 145

- Abtastwert-orientierte Komprimierung: jedes Sample wird einzeln komprimiert
- Segment-orientierte Komprimierung: Samples werden zusammengefasst und dann komprimiert -> effizienter!

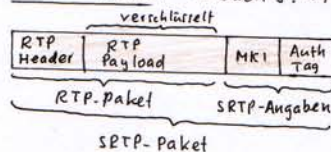
Sprache: 300..3400 Hz -> Shannon: 8000 Samples/s. à 8 Bits -> 64 kbit/s (PCM)

Nichtlineare Quantisierung -> Buch S. 139



- A-law: Europa, 13 segmente
- M-law: Amerika/Japan, 15 segmente
- A-law: Abtastwertorientiert, 64 kbit/s
- M-law: Abtastwertorientiert, 64 kbit/s
- G.729: Segmentorientiert, 8 kbit/s
- G.711: ISDN-Kompression

SRTP-Paket -> Buch S. 176

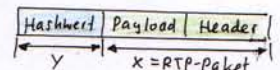
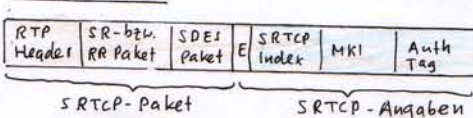


Authentifizierungstag: erste 32..80 bits (of 160) von Hashwert

Hashfunktion: HMAC-SHA1

f(.) -> gibt immer 160 Bits zurück

SRTPC-Paket



Z = HMAC(X, SA) SA: Schlüssel zur Authentifizierung

Z = Y\* => X stammt vom wahren Absender, nicht manipuliert

Z != Y\* => X verfälscht, manipuliert

Integritätsprüfung