



## Zusammenfassung Transportschicht I

- Transportschicht macht unzuverlässige Netzwerke zuverlässig
- TCP: Verbingung, UDP: unzuverlässiges Paket
- Verbindungsabbau: symmetrisch vs. asymmetrisch
- Adresse + Port definieren Endpunkt
- 2 Endpunkte definieren eine Verbindung
- Sequenznummern dürfen innerhalb der maximalen Paketlebenszeit nicht wiederholt werden
- 3 Way Handshake beim Verbindungsaufbau
- 4 Way Verbindungsabbau (auch 3 Way, da FIN+ACK in einem Segment gesendet werden dürfen)
- Crash Recovery mit unterschiedlichen Client- und Serverstrategien





## Ziele heute

- Sie können die Struktur des E-Mail-Transports erklären
- Sie können SMTP, IMAP und MIME erkennen
- Sie können SMTP Umschlag (envelope) und Nachricht (message) unterscheiden
- Sie können die Kompressionsraten von Medien-Codecs einordnen
- Sie kennen übliche Optimierungen für Video-Codecs
- Sie können Unterschiede in den Anforderungen bei Video-on-Demand und Interaktiven Konferenzen erklären
- Sie verstehen den Nutzen von CDNs und Proxies
- Sie wissen, wodurch Peer-to-Peer-Netzwerke selbstskalierend sind





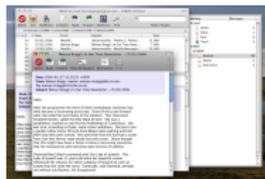
# E-Mail-Programme: Mail User Agent (MUA)

## Thunderbird



Tim Schulz  
[commons.wikimedia.org/wiki/File:Mozilla-Thunderbird-Windows.png](https://commons.wikimedia.org/wiki/File:Mozilla-Thunderbird-Windows.png)

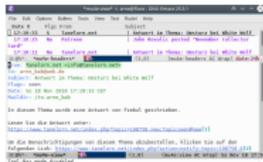
## Gnumail



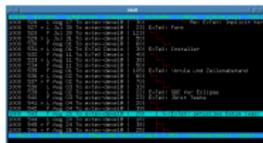
Russ h [com-](http://com-)

Draketo

## mu4e



## mutt



plaicy  
[commons.wikimedia.org/wiki/File:Mutt-Threaded-Modus.png](https://commons.wikimedia.org/wiki/File:Mutt-Threaded-Modus.png)

## webmail



The Horde Project  
[commons.wikimedia.org/wiki/File:Horde-turba.png](https://commons.wikimedia.org/wiki/File:Horde-turba.png)

## hg email

hg email --confirm --to USER@ANBIET  
Diese Patch-Serie besteht aus 1 Pat

Cc:

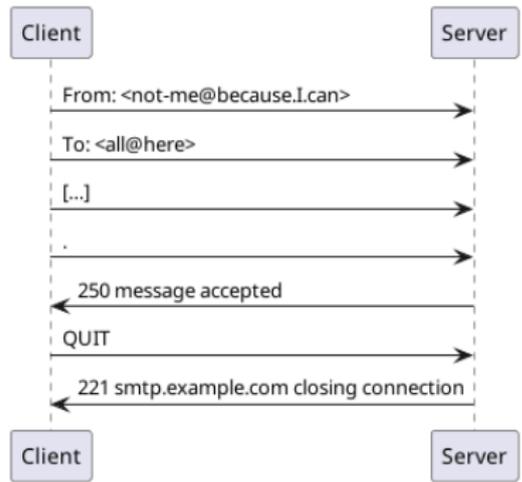
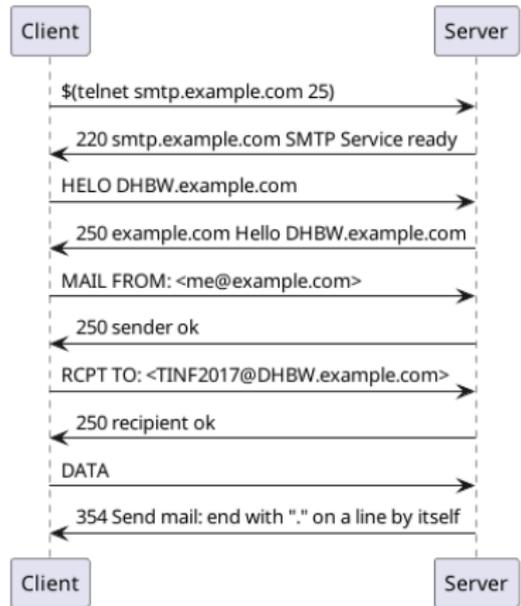
Ergültigte Zusammenfassung:

From: USER@ANBIETER.TLD  
To: USER@ANBIETER.TLD  
Subject: [PATCH] Struktur der Verm

# Mobile Clients?

- Kennen Sie gute? Frei? Offlinefähig?
- Hochleistung nötig!
  - 150k Nachrichten üblich
  - 15 GiB an Daten auch üblich
  - Interaktiver Zugriff nötig!
- Eudora immernoch unerreicht. [computerhistory.org/blog/the-eudora-email-client-source-code/](http://computerhistory.org/blog/the-eudora-email-client-source-code/)

# SMTP



example.com: explizit verbotene Domain.









## IMAP praktisch

*Ausgaben von Telnet mit .., Antworten mit Präfix –*

```
$ telnet imap.web.de 143
.. Trying 212.227.17.162...
.. Connected to imap.web.de.
.. Escape character is '^]'.
-- * OK [CAPABILITY IMAP4rev1 CHILDREN ENABLE ID IDLE LIST-
A0001 NOOP
-- A0001 OK NOOP completed
A0002 LOGOUT
-- * BYE Server logging out
-- A0002 OK LOGOUT completed
.. Connection closed by foreign host.
```



# Format von E-Mails

- Envelope
- Header
- Leerzeile
- Daten
  - MIME





## Format: Umschlag vs. Nachricht III

extensions (RFC 2047 and RFC 2231) define an algorithm for representing header values outside the US-ASCII repertoire, while still encoding them using the US-ASCII repertoire.

Kurz: Das einzig Verlässliche ist:

- **Envelope To.**

Wäre das falsch, hätte die E-Mail Sie nicht erreicht.

Envelope From ist im Umschlag, aber nicht gesichert.

Alles, das keine explizite Kodierung gesetzt hat, ist 7 Bit US-ASCII.

# Format Beispiel

```

Return-Path: <USER@ANBIETER.TLD>
Received: from pop3.web.de [212.227.17.177]
    by localhost with POP3 (fetchmail-6.3.26)
    for <arne@localhost> (single-drop); Sun, 18 Nov 2018 19:00:12 +0100 (CET)
...
Received: from fluss.draketo.de ([84.165.29.35]) by smtp.web.de (mrweb003
    [213.165.67.108]) with ESMTPSA (Nemesis) id 0Lylol-1fLj8s39GB-0167Yi for
    <USER@ANBIETER.TLD>; Sun, 18 Nov 2018 18:45:57 +0100
MIME-Version: 1.0
Content-Type: text/plain; charset="utf-8"
Content-Transfer-Encoding: base64
Subject: [PATCH] Struktur der Vermittlungsschicht
Message-Id: <201d4a8e31a16734b10b.1542563152@fluss.Baez>
X-Mercurial-Node: 201d4a8e31a16734b10b91a9da5bd52ddda5a06b
...
User-Agent: Mercurial-patchbomb/4.7.1
Date: Sun, 18 Nov 2018 18:45:52 +0100
From: USER@ANBIETER.TLD
To: USER@ANBIETER.TLD
X-Spam-Flag: NO
Envelope-To: <USER@ANBIETER.TLD>
X-Spam-Flag: NO
X-UI-Filterresults: notjunk...

IyBIRyBjaGFuZ2VzZXQgcGF0Y2gKIyBvc2VyIEFybUgQmFizW5oYXVzZXJoZWlkZSA8YmFiQGRy
YWtldG8uZGU+CiMgRGFOZSAxNTQyNTYyNjMzIC0zNjAwCiMgICAgICBtdW4gTm92IDE4IDE4OjM3
OjEzIDlwMTggKzAxMDAKIyB0b2RlIE1EIDlwMQ0YThlMzFhMTY3MzZrImtBb2RlOTFhOWhhNjktNTk
...

```

Draketo





# Format: MIME Beispiel II

-----BEGIN PGP SIGNATURE-----

iQIzBAEBCAAAFiEE801qEjXQSQPNItXAE++NRSQDw+sFAlvxsGEACgkQE++NRSQD  
w+sqnxAAtpfgtJKJlJrVf2WHIiDXB86TJHkyQ7uf17Ses3zRIi28RC00KfVYVe  
6CTq0d9NaECn8S/rc2FetHcJnUpW3NwehPJWYyZtMnqFgFjI217acwFwC1T465XO  
DW9xRXTPU3nJZSL2yk80GvQG6b6X+lgZn8wN7iuM+8Hd0ZF95fCA2iyi9/dkanS6U  
Uvgi9W+qY+YeLjnsCq7qZcvIQeViz0WrVa4jgfPibwenVn0Wx2WIeLGAGGhSUFc9  
cTicwJYrZL5rHWT1DsR2NmjNP+fzuKprkZwwTJYhMVzc8z7ExN6nYAwzFW02EFQa  
qpECJEr/hxx8B0i8gRdmBli+AAYouUaLpzap1K2yzQWsfXXpsh0o5DGzdJ3ejv6z  
Ts0+X4V2F+uiay9x21P/agPXJK5IvyQ31NAAPVY22eW/rToS0mltS/zf/uQxkQyw  
SMVrkXMrCdUn6eQ++AMzvGFxliN6NefJW3UviwDWE4U21C/PH2UVuf3xgCG5PNx+  
TmPvN69UxsLKKdagqqWUUAaBzym13Dcilnw26tDo/3LAT6H8MWRhASPRpHS5SWV  
axFMDRW0sFJsLLNrUCOBGGM6SnEECsZT/R72WHA101oIByMUTJaTESpl5nyTzo51  
3of0/Gifs0JtbZsu8IsZnHqMGVERB2Q381Gbkh8TIVNIqAZwZ/GIswQBAQgAHRyh  
BN0ovebZh1yrzkqLHdzPDbMLwQVIBQJb8bBhAAoJENzPDbMLwQVI4DID/373p3W0  
6YCAbd11t0BmH203vRL7/fX43CW9oRYej3RJ10vNME20EZmY+2G1DURDsA9mOXcL  
39n1C6UCN/q/7VJXu7fSdq0YMLZFXfAjBjDWTZegJcF2Ajs4gapgbkSnYjK6PYrS  
dwKVjHoHdiASjPPdbmgL/HWpr8BbahJiuR+mY  
=ZvaC

-----END PGP SIGNATURE-----

-----



# MIME-Typen

Kategorisierung von Inhalten. Beispiele:

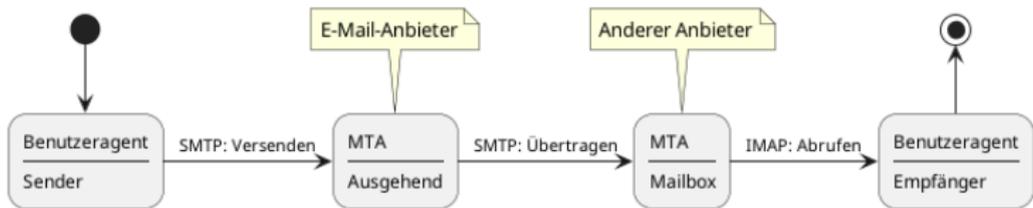
Typ	Endung	Bedeutung
application/octet-stream	-	Beliebige bytes
application/gzip	*.gz	GNU Zip
application/json	*.json	Javascript-Object-Notation
application/pdf	*.pdf	PDF-Dateien
audio/basic	*.au, *.snd	Sound-Dateien
audio/mpeg	*.mp3	mp3-audio
image/png	*.png	PNG-Bilder
multipart/mixed	-	multipart ohne Bezug
text/plain	*.txt	Text
text/css	*.css	Style-Sheets
text/html	*.html	HTML-Dateien (Webseiten)



## Weitere Themen

- Spam-Abwehr. Effektiv ungelöst: Serverside mit hohen Kosten. Spamassassin war mal gut. Spam-„Qualität“ wird immer besser. Kostenlose globale Sichtbarkeit ist schwierig.
  - WhatsApp entfernt das "kostenlos" durch den Zwang zur SIM-Karte.
  - Skalierende Lösungen durch Aufgabe globaler Erreichbarkeit möglich (eigene Arbeit).
    - [www.draketo.de/english/freenet/deterministic-load-decentralized-spam-filter](http://www.draketo.de/english/freenet/deterministic-load-decentralized-spam-filter)
    - Ende-zu-Ende Signaturen => mailvelope auf web.de/gmx
- Verschlüsselung: PGP → pEp (pretty easy privacy) + autocrypt: Automatische Verschlüsselung mit Sender-Identifikation und TOFU (trust on first use).
  - <https://gnupg.org>
  - <https://www.pep.security/> — pEp plugin
  - <https://autocrypt.org/level1.html>
  - <https://mailvelope.com/> — webmail

# Zusammenfassung



- MUA: E-Mail-Client.
- MTA: Mail Transfer Agent
- SMTP: MUA → MTA oder MTA → MTA. text via telnet
  - **Umschlag**: MAIL FROM (ich), RCPT TO (Sie).
  - Einzig verlässlicher Wert: **Envelope To**.
  - EHLO: Extended HELO.
- IMAP: MTA → MUA, Textprotokoll, Port 143.
  - Serverseitige Operationen!
- Format: Plain Text Header. 7Bit ASCII. Body MIME.
- MIME: Multipart mit Trenner (boundary=XXX).





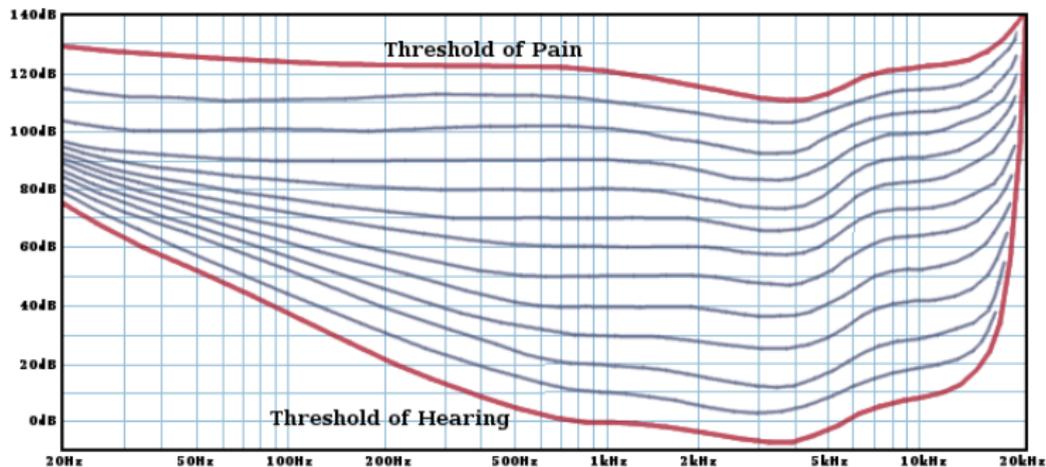








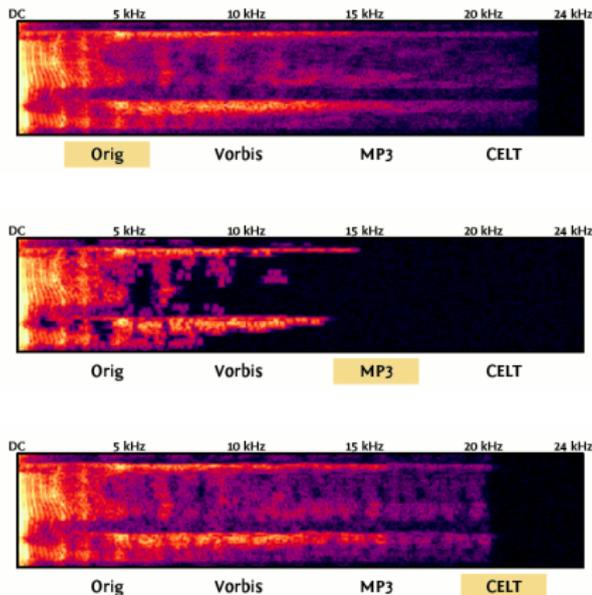
# Das Gehör verstehen



*„Neben“ lauten Tönen weniger!*

Dank Monty: [people.xiph.org/~xiphmont/demo/neil-young.html](http://people.xiph.org/~xiphmont/demo/neil-young.html)

# Moderne Optimierungen



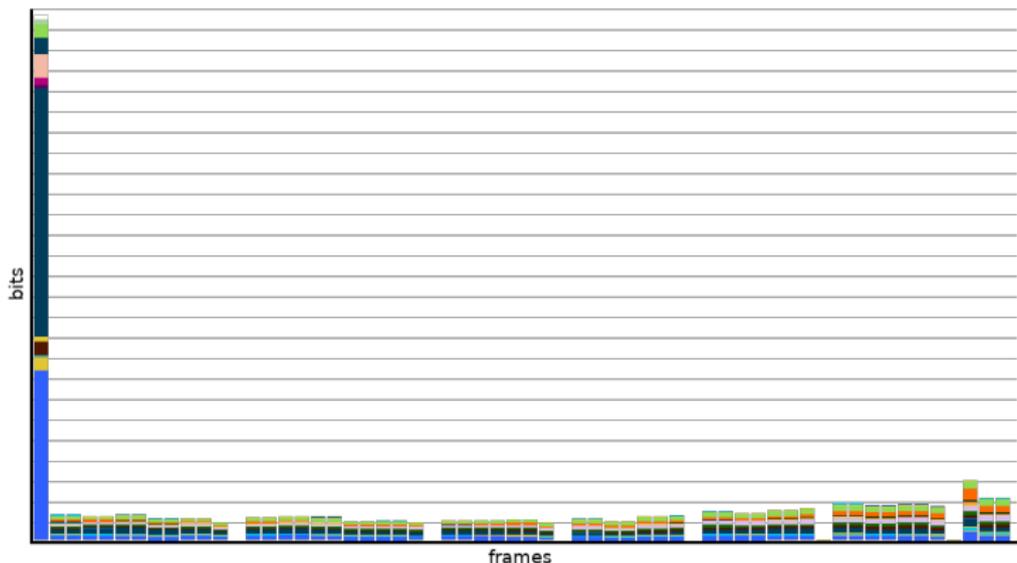
- orig: >1000 kbit/s
- mp3: 128 kbit/s
- CELT: 32 kbit/s
- SILK: 6-40 kbit/s
- LPCNet: 1,6 kbit/s  
(PDF)

*SILK für Opus Sprache genutzt,  
Celt für Opus Musik:*

[people.xiph.org/~xiphmont/demo/celt/demo.html](http://people.xiph.org/~xiphmont/demo/celt/demo.html)  
[people.xiph.org/~jm/opus/opus-1.3/](http://people.xiph.org/~jm/opus/opus-1.3/)



# Video-Codex: Keyframes + Inkrementelle Änderungen



*Vergangenheit oder Vergangenheit + Zukunft*

<https://people.xiph.org/~xiphmont/demo/av1/demo1.shtml>

Vergleich: 132 MiB h264 vs. 7 MiB vp9 q62 (min: 64) |  
h264, 132 MiB



Vergleich: 132 MiB h264 vs. 7 MiB vp9 q62 (min: 64) II  
vp9, 7 MiB

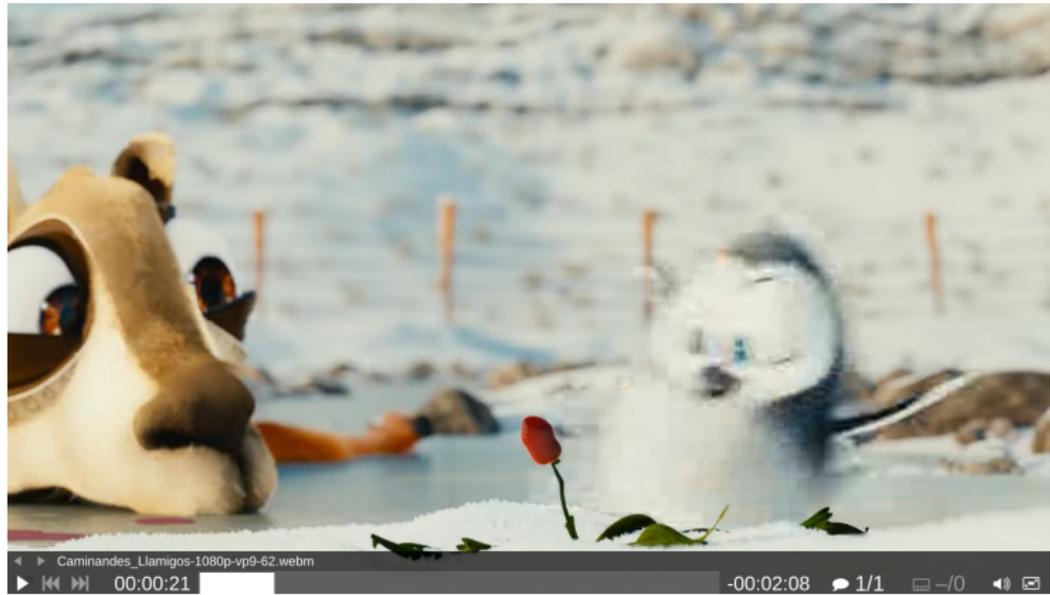


# Vergleich: 132 MiB h264 vs. 7 MiB vp9 q62 (min: 64) III

h264, 132 MiB



# Vergleich: 132 MiB h264 vs. 7 MiB vp9 q62 (min: 64) IV vp9, 7 MiB

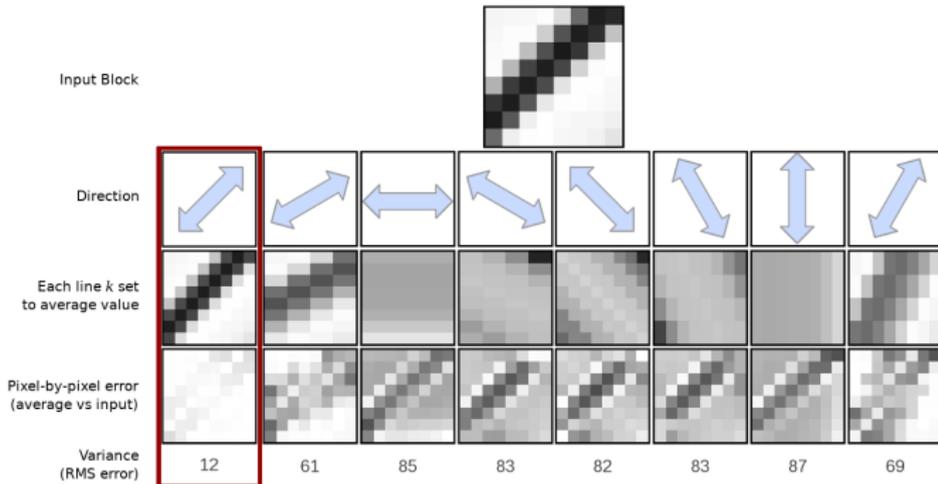




# Optimierungsbeispiel: Directional Paint, Konzept



# Optimierungsbeispiel: Directional Paint, Implementierung

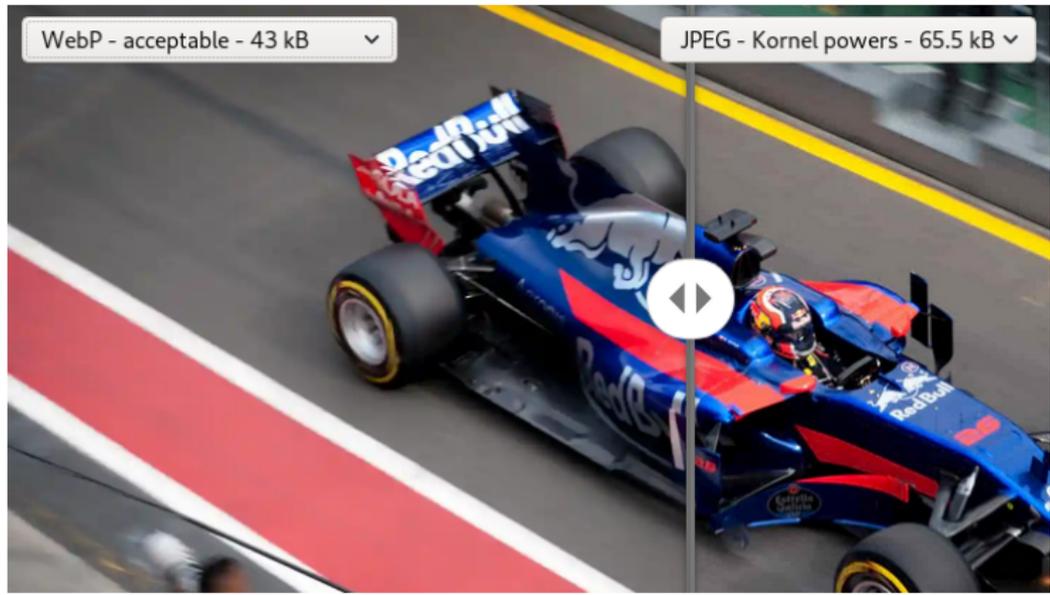


CDEF: <https://people.xiph.org/~xiphmont/demo/av1/demo2.shtml>

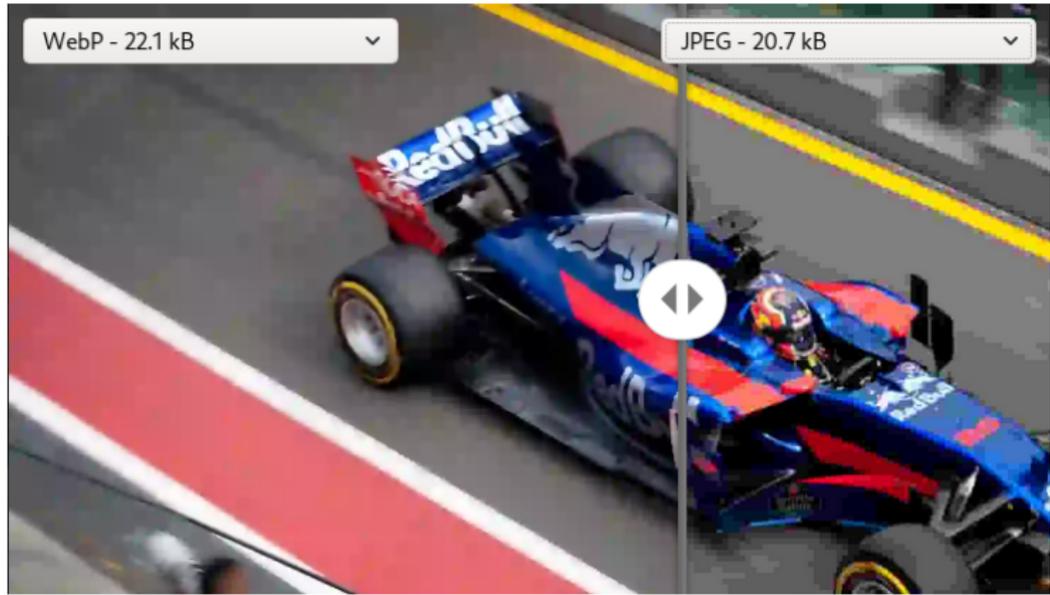




# Bild-Codects: Beyond JPEG: webp 43kB vs. JPEG 66kB



# Bild-Codects: Beyond JPEG: webp 22kB vs. jpeg 21kB



## Zwischen-Zusammenfassung: Codecs

### ■ Audio:

- CD: 1333 kib/s
- opus: 16 kib/s
- LPCnet: 1,6 kib/s (Sprache)
- **Faktor 85 – 850** → <https://caniuse.com/opus>

### ■ Video:

- dv uncompressed:  $\approx 150.000$  kib/s
- dv compressed:  $\approx 30.000$  kib/s
- av1: 100kib/s
- **Faktor: 1700** → <https://caniuse.com/av1>

### ■ Bild:

- Original: 3500kB; 1920x1080  $\Rightarrow$  13.5bit/pixel
- JPEG: 66kB  $\Rightarrow$  0.26 bit/pixel
- avif: 18kB  $\Rightarrow$  0.07 bit/pixel
- **Faktor: 200** → <https://caniuse.com/avif>



# Unterschiedliche Anforderungen und Randbedingungen

- Video-On-Demand (VOD)
- Live-Streaming
- Interaktive Medien

→ *An der Tafel sammeln*

## Anwendung: Video on-demand

- Startverzögerung minimieren
- Zwischenspeicher minimieren
- Cache glättet Jitter (aber Youtube hängt -, -)
- Geringe Anforderungen: TCP reicht
- <video>-Tag macht das einfach, mit dem richtigen MIME-Typ
- Teil-Anfragen über Range-Header
- Einmal enkodiert, oft dekodiert.

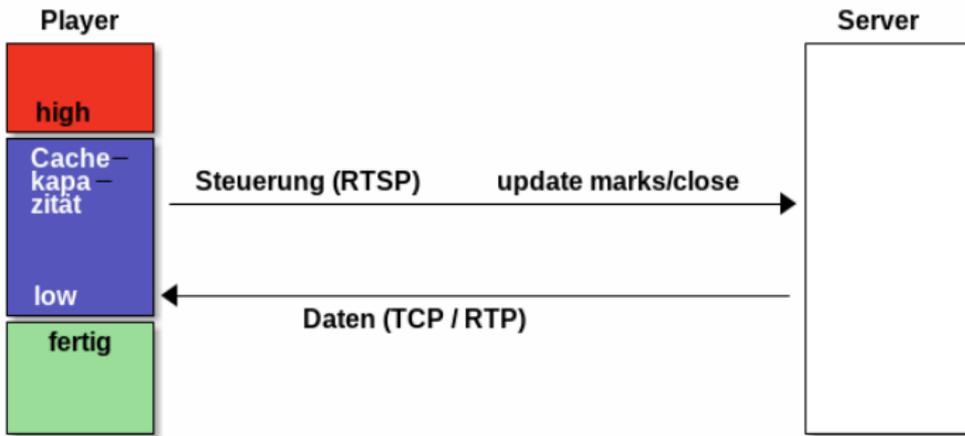
*Früher über Mediaplayer mit RTSP, heute macht der Browser alles.*

# Live-Streaming

- Keine Optimierung mit Daten aus der Zukunft
- Begrenzte Enkodierungs-Zeit
- Dafür: Multicast vom Provider
- Cache-Optimierung mit RTSP
- RTP: Real-Time Transport Protocol
  - UDP + Steuerpakete (RTCP: Real-Time Control Protocol)

# RTSP: Real-Time Streaming Protocol

- High-Water-Mark,
- Low-Water-Mark,
- Abbrechen
- ...





# Latenz Real

```
$ ping uni-hamburg.de
64 bytes from 134.100.36.5: icmp_seq=0 ttl=246 time=40,793 ms
64 bytes from 134.100.36.5: icmp_seq=1 ttl=246 time=81,462 ms
64 bytes from 134.100.36.5: icmp_seq=2 ttl=246 time=40,622 ms
64 bytes from 134.100.36.5: icmp_seq=3 ttl=246 time=40,096 ms
```

```
$ sudo traceroute uni-hamburg.de
 1  192.168.2.1  0,339ms  0,748ms  0,186ms
 2  62.155.245.143  22,924ms  17,829ms  17,195ms
 3  217.0.198.229  22,654ms  22,100ms  22,116ms
 4  217.0.198.229  22,590ms  22,112ms  21,399ms
 5  * * *
 6  4.69.142.209  41,394ms  41,031ms  41,059ms
 7  195.122.181.62  42,341ms  39,598ms  40,828ms
 8  188.1.231.82  73,916ms  47,432ms  47,750ms
 9  134.100.254.173  40,370ms  39,352ms  39,368ms
10  134.100.36.5  40,333ms  39,651ms  39,568ms
```

# Weitere Aspekte

## Protokolle für paketbasierte interaktive Kommunikation

- H.323
  - Schnittstelle zwischen Telefonnetz und Internet.
  - RTCP für Steuerung, UDP für Daten.
  - Gateway vermittelt zwischen Telefon und Internet.
- SIP: Session-Initiation-Protocol
  - Liefert Umleitungsserver und externe IP.
  - INVITE, ACK, BYE, CANCEL, REGISTER (Umleitungsserver)

# Streaming Media: Zusammenfassung

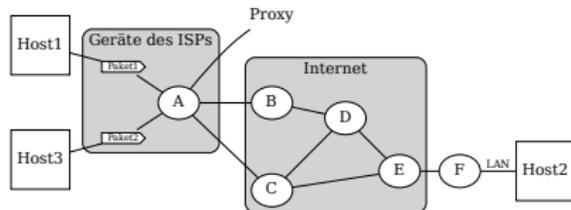
- Codecs reduzieren die Bandbreite um Faktor 100
- Video on Demand über TCP
- Interaktiv: RTSP zur Cache-Optimierung
- SIP zum Aufbau von Verbindungen (Sitzungs-Protokoll — OSI-Schicht 5 in Anwendungsschicht)

# Datenverteilung

- Häufigkeit der Zugriffe
- Proxy
- Content Delivery Network (CDN)
- peer-to-peer (p2p)

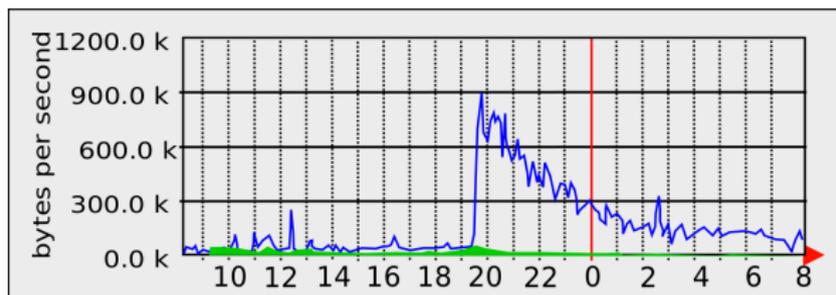


# Proxy



- Cache des Internetanbieters (ISP)
- Ruft statische Dateien nur einmal ab → spart Bandbreite
- Invalidierung: HTTP-Header für Lebenszeit (z.B. modified-since)
- Kontrolliert vom Internetanbieter

# Content-Delivery-Network (CDN)

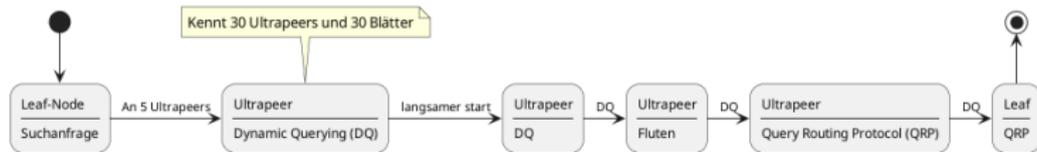


- Regional verteilte Server von Fremdanbietern
- Kontrolliert vom Anbieter. Beispiele: Akamai, Cloudflare, AWS
- Vertrag mit Webseitenbetreiber, Einfluss auf die Webseiten, unterstützt SSL.
- Über DNS zugewiesen
- Vermeidet den Slashdot-Effekt

# peer-to-peer (p2p)

- Nutzer bieten untereinander Dienste an
  - Verteilte Suche
  - Verteilter Index
  - Gemeinsame Inhaltsverteilung (swarming)
- Selbstskalierende Dienste
  - Kapazität steigt mit der Anzahl der Nutzer
  - Kosten steigen üblicherweise nur logarithmisch

## Beispiel für verteilte Suche: Gnutella 0.6 (50 mio Nutzer)



- Dynamic Querying
  - Einen nach dem anderen anfragen (alle binnen etwa 3 Sekunden)
  - bei ausreichend Antworten abbrechen
- Query Routing Protocol
  - Hash-Tabellen mit schwachem Hash auf Suchwort
  - Anfragen erreichen nur Knoten mit wahrscheinlichen Treffern
  - Inter-Ultrapeer QRP: Zusammenfassen der Tabellen.

*weiterlesen:*

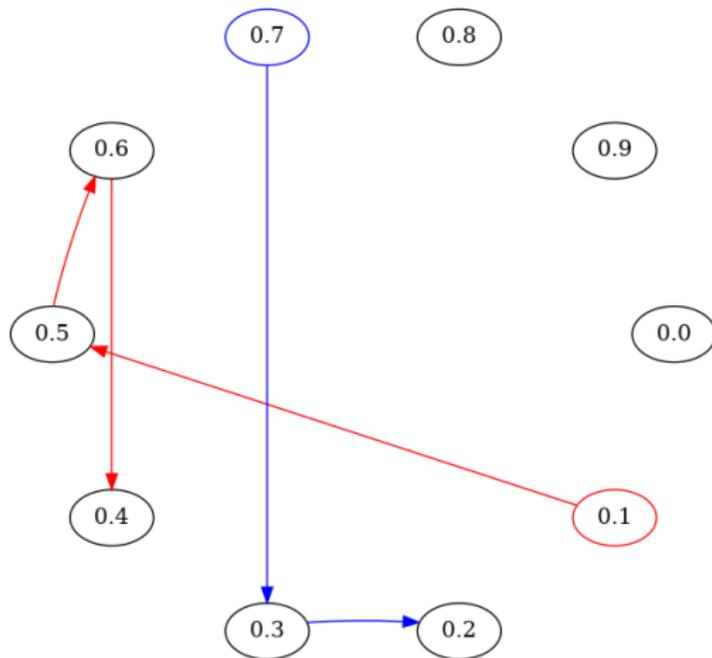
*<http://rfc-gnutella.sourceforge.net/src/grp.html>  
und <http://www.ic.unicamp.br/~celio/peer2peer/>*

## Verteilter Index: Kademlia

- Abstand: (sha1 Knoten-ID) XOR (sha1 Daten)  $\rightarrow$  1011 XOR 1000 = 0011
- Knotenlisten (k-Buckets): 160 Listen mit jeweils k Einträgen: IP-Adresse, UDP-Port und Node-ID. Least-recently-seen-queue.
- Bedingung: Jeder Eintrag in Liste n hat die ersten n bits gleich.
- Suche nach Daten: Anfrage an alle in der passenden Liste:
  - Gib mir die Besten Knoten für den Hash X.
  - Die besten Knoten behalten und wieder fragen.
  - Je näher die Knoten am Ziel sind, desto mehr der ihnen bekannte Knoten sind nahe am Ziel.
- Sucht exakten Hash.

*weiterlesen: <https://en.wikipedia.org/wiki/Kademlia> + <https://sarwiki.informatik.hu-berlin.de/Kademlia>*

# Verteilter Index: Freenet







# Zusammenfassung

## Proxy:

- Vom Internetanbieter betrieben
- Zwischenspeicher für Daten
- Spart Bandbreite

## CDN:

- Dienstleistung für Webseiten
- Bei Internetanbietern aufgestellt, aber von CDN-Betreiber Kontrolliert

## Peer-to-Peer:

- Selbstskalierend
- Fuzzy-Suche: Gnutella
- Hash-Suche: Kademlia
- Swarming: Dateien aus vielen Quellen

# Rückmeldung

- Was sollte ich beibehalten?
- Was sollte ich ändern?

# Zusammenfassung

- E-Mail ist ein verteiltes Protokoll zwischen verschiedenen Anbietern
  - E-Mails werden mit SMTP weitergeleitet
  - Nicht-Text-Inhalte können in MIME gekapselt werden
- Streaming verwendet hoch-optimierte Codecs
  - Video-on-Demand kann mehr Optimierungen nutzen als interaktive Konferenzen
- Mittel zur Datenverteilung sind
  - Proxies (beim Internetanbieter)
  - CDNs (vom Webseitenbetreiber)
  - Peer-to-Peer-Netze (bei den Nutzern)

# Fragen für die Prüfung?

## Ideensammlung:

- 36 GiB für 1h 720p Video, kann das ein moderner Codec sein?
- Nennen Sie 2 Beispiele für Informationen, die in den Envelope einer E-Mail gehören.
- Beschreiben Sie den Unterschied zwischen Envelope und Header

## Selbststudium diese Woche I

- Schreiben Sie einen Webserver, der auf die ersten drei SMTP-Anfragen antworten kann. Verbinden Sie sich mit telnet mit Ihrem Server und dokumentieren Sie die Interaktion. Sie brauchen **kein** STARTTLS zu implementieren. Wählen Sie dafür entweder die Sprache, in der Ihnen die Aufgabe **leichter** fällt. Die Sprachen erhalten Sie wieder von dem Sprachpaargenerator (sie sind noch gleich):  
<https://www.draketo.de/software/vorlesung-netztechnik#nummer-zu-sprache> (läuft clientseitig in Ihrem Browser).
- Zeigen Sie, wie weit sie mit TELNET mit dem wirklichen SMTP-Server ihres E-Mail-Providers sprechen können.



# Verweise I

Bilder: