

Propädeutik der Kommunikation

Wir versenden eine Email



<https://iuk.one/RNDS-PROP.pdf>

Clemens H. **Cap**
ORCID: 0000-0003-3958-6136

Department of Computer Science
University of **Rostock**
Rostock, Germany
clemens.cap@uni-rostock.de

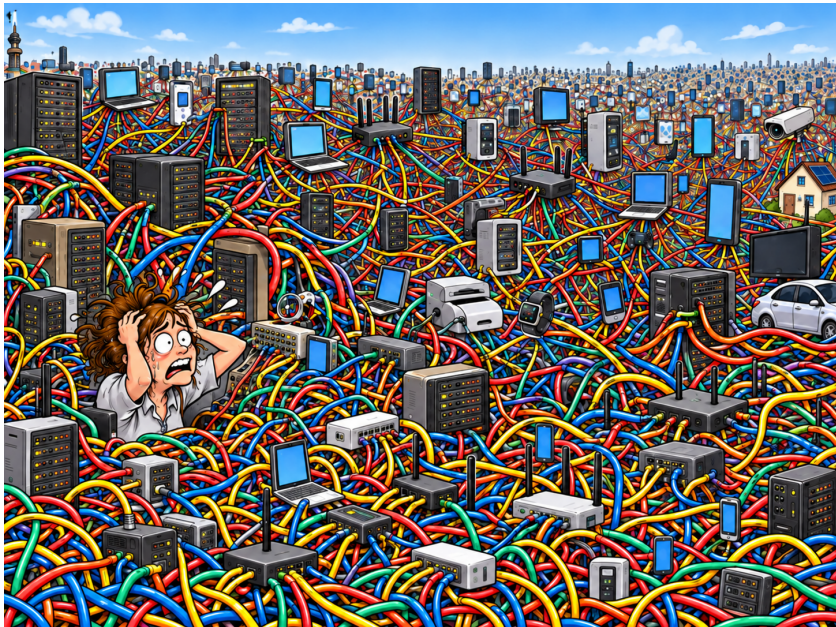
2026-10-11 20:43



Eine scheinbar einfache Aufgabe...

Wir senden eine Email von Rostock nach Wien.

An diesem Beispiel erklären wir die wichtigsten Begriffe.



1. Anwendungen: Eine Email
2. Darstellung: In Bits
3. Datenformate & MIME
4. Datenpakete & Netzdienste
5. Adressen, Teilnehmer & Namen
6. Routing
7. Sitzungen

Die erste naive Frage

Frage: Was ist eine Email?

Eine typischerweise eher kurze Nachricht die von einem Sender an einen oder mehrere Empfänger gesendet wird und dabei mehrere vermittelnde Zwischenknoten durchläuft. Eigentlich ist es viel komplizierter, aber für den Beginn darf uns das genügen.

Wichtiger als die Antwort selbst ist diese Frage:

Und wo finde ich eine Antwort auf diese Frage?

Suchmaschinen und KI können mir die Antwort geben.

Besser ist es, diese dafür zu verwenden, mir zu sagen, wo ich die Antwort finde.

1. Anwendungen: Eine Email

Auf Anwendungsebene erklären wir, was die Anwendung genau umfaßt (Teile, Aufgaben usw.)

Im **ISO-OSI Modell** wird diese Aufgabe auf dem **Application Layer (Layer 7, Anwendungs-Schicht)** beschrieben.

1. Anwendungen: Eine Email

2. Darstellung: In Bits

3. Datenformate & MIME

4. Datenpakete & Netzdienste

5. Adressen, Teilnehmer & Namen

6. Routing

7. Sitzungen

1. Anwendungen: Eine Email

Aus welchen Teilen besteht eine Email? (1)



Abb. 2: Bestandteile einer Email [Rechte s. Anhang.](#)

1. Anwendungen: Eine Email

Aus welchen Teilen besteht eine Email? (2)

Eine E-Mail besteht aus

- ① **Envelope** Wohin die Email gehen soll.
- ② **Header** Weitere Meta-Daten oder Steuerdaten der Email.
 - `From: Alice <alice@example.org>` Zieladresse
 - `Subject: Test` Subject (Titel)
 - `X-Spam-Status: Yes` Spam Filter Bewertung
 “unterwegs” hinzugefügt
- ③ **Body** Texte, Bilder, Word-Dokumente
 - Frage 1: Wie voneinander abgetrennt?
 - Frage 2: Wie als Text / Bild / Word erkannt?
 - Frage 3: Wie wird ein Bild richtig als Bild dargestellt?

1. Anwendungen: Eine Email

Wo finden wir die Antwort?

Es gibt verschiedene **Referenz-Modelle**:

- **Internet Modell**: Beschreibt den konkreten Aufbau des Internet.
- **ISO-OSI Modell**: Ist ein konzeptuelles, didaktisches Modell.

Im **Internet Modell** geregelt in den **RFCs**: Request For Comments

- ① RFC 5321: Simple Mail Transfer Protocol [[Kle08](#)]
- ② RFC 5322: Internet Message Format [[Res08](#)]
- ③ Liste aller RFCs

Im **ISO-OSI Modell** geregelt in den Dokumenten:

- ① Zur Anwendungs-Schichte: [ISO/IEC 9545:1994\(en\) Application Layer Structure](#)
- ② Allgemein: [ISO/IEC 7498-1: Open Systems Interconnection: Basic Reference Model](#)

Empfehlung: Sehen Sie sich diese RFCs einmal kurz an. 10 Minuten genügen...

Hinweis: Die farbigen Texte sind Links...

2. Darstellung: In Bits

Anwendungsteile müssen in Symbole und diese in einzelne Bits (0/1) heruntergebrochen werden, die dann übertragen werden.

Im **ISO-OSI Modell** wird diese Aufgabe auf dem **Presentation Layer (Layer 6, Darstellungsschicht)** beschrieben.

1. Anwendungen: Eine Email

2. Darstellung: In Bits

3. Datenformate & MIME

4. Datenpakete & Netzdienste

5. Adressen, Teilnehmer & Namen

6. Routing

7. Sitzungen

2. Darstellung: In Bits

Darstellung von Texten

Buchstaben werden durch Bit-Ketten dargestellt.

1 Bit: 0 oder 1

1 Byte: 8 bits

Wichtige **Standards:**

- **ASCII** American Standard Code for Information Interchange
ASCII Tabelle
- **ISO 8859-1** Greift nationale Sonderzeichen auf: Bsp: **dégoûtèrent**
ISO 8859-1 Tabelle
- **Unicode** Unicode Standard
Modernster Standard, der fast alles abdeckt.
∀ ∃ ∫ ∂ bis Koreanisch, Japanisch, Arabisch und Emoji.

2. Darstellung: In Bits

Bestandteile einer Email



Abb. 3: Bestandteile einer Email: Wie erfolgt die Darstellung der einzelnen Teile? Rechte s. Anhang.

Trennung von Header und Body

Header und Body werden getrennt durch: Leerzeile (CRLF CRLF)

CRLF steht für CR LF

CR steht für Carriage Return

LF steht für Line Feed

Das erste CRLF für das Ende der letzten Header-Zeile.

Das zweite CRLF unmittelbar danach ohne weitere Header-Zeile.

Das bedeutet: Alle Header sind zu Ende, der Body beginnt.

2. Darstellung: In Bits

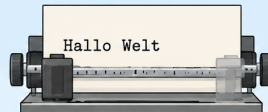
Warum heißen CR und LF so?

CR – Carriage Return

Der Schreibkopf (**Carriage**) wird an den Anfang der aktuellen Zeile zurückbewegt.



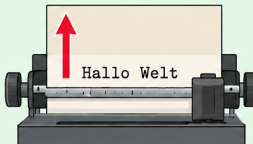
- 1 **Carriage Return (CR):**
Schreibkopf fährt nach links an den Zeilenanfang.



- 2 Der Schreibkopf steht wieder am Anfang der gleichen Zeile.

LF – Line Feed

Das Papier wird um eine Zeile nach oben bewegt, sodass eine neue Zeile entsteht.



- 1 **Line Feed (LF):**
Papier wird um eine Zeile nach oben bewegt.

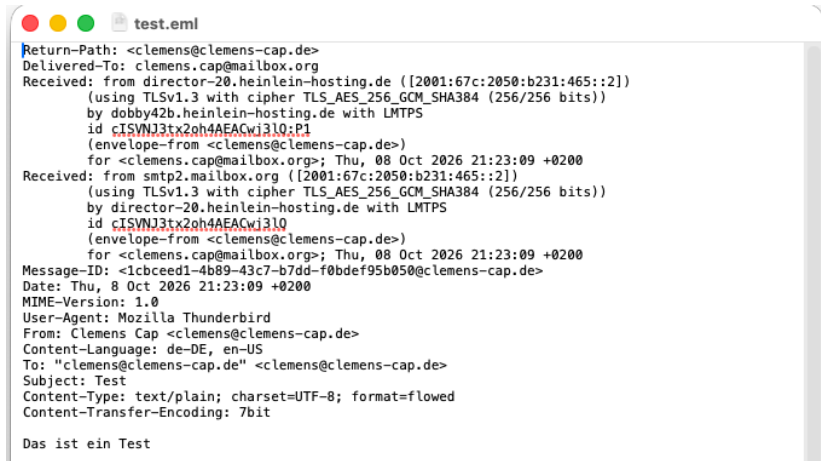


- 2 Der Schreibkopf steht in der neuen Zeile (gleiche horizontale Position wie zuvor).

Abb. 4: Erklärung der Namen von Carriage Return und Line Feed [Rechte s. Anhang.](#)

2. Darstellung: In Bits

Wir testen das jetzt selber! (1)



```
test.eml
Return-Path: <clemens@clemens-cap.de>
Delivered-To: clemens.cap@mailbox.org
Received: from director-20.heinlein-hosting.de ([2001:67c:2050:b231:465::2])
    (using TLSv1.3 with cipher TLS_AES_256_GCM_SHA384 (256/256 bits))
    by dooby42b.heinlein-hosting.de with LMTPS
    id cISVNj3tx2oh4AEACwi3l0:P1
    (envelope-from <clemens@clemens-cap.de>)
    for <clemens.cap@mailbox.org>; Thu, 08 Oct 2026 21:23:09 +0200
Received: from smtp2.mailbox.org ([2001:67c:2050:b231:465::2])
    (using TLSv1.3 with cipher TLS_AES_256_GCM_SHA384 (256/256 bits))
    by director-20.heinlein-hosting.de with LMTPS
    id cISVNj3tx2oh4AEACwi3l0
    (envelope-from <clemens@clemens-cap.de>)
    for <clemens.cap@mailbox.org>; Thu, 08 Oct 2026 21:23:09 +0200
Message-ID: <1cbceed1-4b89-43c7-b7dd-f0bdef95b050@clemens-cap.de>
Date: Thu, 8 Oct 2026 21:23:09 +0200
MIME-Version: 1.0
User-Agent: Mozilla Thunderbird
From: Clemens Cap <clemens@clemens-cap.de>
Content-Language: de-DE, en-US
To: "clemens@clemens-cap.de" <clemens@clemens-cap.de>
Subject: Test
Content-Type: text/plain; charset=UTF-8; format=flowed
Content-Transfer-Encoding: 7bit

Das ist ein Test
```

Abb. 5: Email senden, empfangen, als Datei speichern, im Texteditor öffnen. Zwischen Header und Body sehen wir eine Leerzeile. [Rechte s. Anhang.](#)

2. Darstellung: In Bits

Wir testen das jetzt selber! (2)

```
0002100  6e 74 2d 54 72 61 6e 73 66 65 72 2d 45 6e 63 6f
          n  t  -  T  r  a  n  s  f  e  r  -  E  n  c  o
0002120  64 69 6e 67 3a 20 37 62 69 74 0d 0a 0d 0a 44 61
          d  i  n  g  :          7  b  i  t  \r \n \r \n  D  a
0002140  73 20 69 73 74 20 65 69 6e 20 54 65 73 74 0d 0a
          s          i  s  t          e  i  n          T  e  s  t  \r \n
0002160
cap@eurydice-2 tmp %
```

Abb. 6: Mit dem [octal dump Werkzeug od](#) finden wir vor dem Body ein `\r\n\r\n` bzw. hexadezimal `0d 0a 0d 0a`. Laut [ASCII Tabelle](#) ist das genau CR LF! Im Texteditor wurde das nicht angezeigt. [Rechte s. Anhang](#).

Aufgabe: E-Mail Header

- ① Wie trennen E-Mail Header **ein** key-value Paar vom **nächsten** key-value Paar.
Dh.: Wo endet die Information?
Was ist X-Custom:foo:bar:baz?
- ② Wie **trennt ein key value** Paar den key vom value?
Was ist: X-Custom: foo:bar?
key X-Custom:foo mit value bar oder key X-Custom mit value foo:bar?
- ③ Wie geht E-Mail vor, wenn ein CRLF **Teil** eines value im Header sein soll.
Beispiel: Wir wollen Zeilenumbruch im Subject haben?
- ④ Kann in einer E-Mail ein key-value Paar **zu lang** sein?
- ⑤ Sind die key X-Spam und x-sPam gleich? Sind die keys **case-sensitive**?

Und wenn Sie diese Antworten jetzt vermutlich mit einer KI gefunden haben:

- Wo genau sind die Antworten denn definitiv **festgelegt**?
- **Stimmen** die Antworten der KI? (Man muß immer die Quellen überprüfen!)

2. Darstellung: In Bits

Weitere Themen

Zustellung: Unterschiedliche Funktionen in Envelope und Header.

- RCPT TO:, MAIL FROM: aus Envelope verschieden von To: From: aus Header.
- **Envelope:** Email wird erhalten.
- **Header:** Mailsystem zeigt anderen Absender oder Empfänger an.

Klassifikation: Header X-Spam: Gibt Spam-Rating an.

- Header X-Spam: stammt nicht vom Absender!
- Zwischenknoten analysiert Mail ("*Make money now*") und fügt Header hinzu.
- Zwischenknoten können Mails lesen und verändern (sic!).

Thema 1: Vertrauen in Zwischenknoten

Vorlesung Datensicherheit beschreibt Verschlüsselung und digitale Unterschrift.

Thema 2: Exaktes Funktionieren nach Spezifikation

- **Application Layer 7:** regelt Teile der Anwendung und ihre Bedeutung.
- **Presentation Layer 6:** regelt (unter anderem) wie Teile codiert/extrahiert werden.
- Jeweils spezifizierte Funktionen sind Teile eines Architekturmodells.

3. Datenformate & MIME

Wir bringen wir Bilder in die Email?

Welche weiteren Datenformate gibt es?

Wie werden Datenformate erkannt?

Im **ISO-OSI Modell** gehört diese Aufgabe ebenso in den Presentation Layer (Layer 6, Darstellungsschicht).

1. Anwendungen: Eine Email

2. Darstellung: In Bits

3. Datenformate & MIME

4. Datenpakete & Netzdienste

5. Adressen, Teilnehmer & Namen

6. Routing

7. Sitzungen

3. Datenformate & MIME

MIME: Multipurpose Internet Mail Extensions

Aufbau: Typ/Subtyp Typ Art der Daten Subtyp Spezifisches Format

- Geben Art der Daten an und wählen Verhalten im empfangenden Programm aus.
- Nutzung erst für Email, später für weitere Protokolle (speziell: HTTP).

Beispiele:

text/plain

Reiner Text ohne Formatierung

text/html

Text, der als HTML zu interpretieren ist.

text/css

Text, als Web Style Sheet zu interpretieren

image/png

Bild im Format PNG Portable Network Graphics

image/jpeg

Bild im Format JPEG Joint Photographics Experts Group

application/octet-stream

Binärdatei (kann ausführbar sein!)

Liste **aller** offiziell registrierten MIME-Types

Liste der **wichtigsten** MIME-Types in Browsern

Folgen falscher MIME-Types

Folge: Falsche MIME-Types führen zum Aufruf falscher Renderer im Browser.

```
<h3>Hallo Prof. Cap</H3> <br/>  
Heute komme ich <b>nicht</b> zur Vorlesung.
```

Abb. 7: Darstellung einer text/html Email als Text [Rechte s. Anhang.](#)

Hallo Prof. Cap

Heute komme ich **nicht** zur Vorlesung

Abb. 8: Darstellung einer text/html Email als HTML [Rechte s. Anhang.](#)

Folgen falscher MIME-Type Regeln (1)

Situation:

Manche Programme lassen die Konfiguration von Verhalten für bestimmte MIME-Types zu.

- Bsp: Mail-Programme (Bsp: Mail-Filter, Mail-Bots)
- Bsp: Web-Browser (via Extensions, native bridges)
- Bsp: Filesystem-Browser (Bsp: Explorer, Finder)

Entwicklung: Über die Jahre ist das immer komplizierter geworden.

- **Erst:** Browser-Settings
- **Später:** Registry & besondere, versteckte Browser-Flags
- **Grund:** Fehlkonfiguration kann Sicherheitslücken öffnen.
- **Daher:** Endbenutzer sollen das besser nicht anfassen (können).

Folgen falscher MIME-Type Regeln (2)

How do I associate a custom MIME-type to my local application browsers?

Asked 15 years, 4 months ago Modified 7 years, 11 months ago Viewed 17k times



14

I want to invent a new mime-type and associate it to a custom application in the browser to enable users to launch my app from a web page. The users of my secure web site are in a closed-environment, meaning this is not a general-purpose, mainstream application - I can configure their browser ahead of time.

Abb. 9: Der **Wunsch**, bei Laden einer bestimmten MIME-Type aus dem Netz automatisch eine konfigurierbare Anwendung zu starten ist alt und erscheint durchaus gerechtfertigt. [Rechte s. Anhang.](#)

3. Datenformate & MIME

Folgen falscher MIME-Type Regeln (3)



Abb. 10: Die sicher falsche Reaktion des Web-Browsers, wenn der Server eine Datei mit `application/octet-stream` Mime-Type sendet, ist die Ausführung als Binärdatei. Rechte s. Anhang.

Folgen falscher MIME-Type Regeln (4)

Experiments

154.0.8037.58

Available

Unavailable

Enable chrome.**mime**Handler extension API

Enables the chrome.mimeHandler extension API for extensions with a mime_types_handler manifest entry. The API is currently limited to a small set of MIME types (e.g. application/pdf). – Mac, Windows, Linux, ChromeOS

[#api-mime-handler](#)

Default



Abb. 11: Daher wurde die freie Konfigurierbarkeit durch die Browser-Hersteller immer komplizierter gestaltet. Hier in Chrome unter <chrome://flags>. (Automatische Link-Öffnung funktioniert nicht unter jedem PDF-Reader, was eine sinnvolle Einschränkung darstellt.) [Rechte s. Anhang.](#)

Folgen falscher MIME-Type Regeln (5)

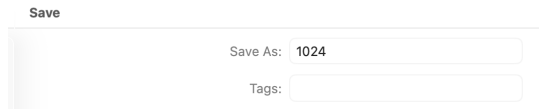


Abb. 12: Die richtige Reaktion des Web-Browsers, wenn der Server eine Datei mit `application/octet-stream` Mime-Type sendet, *kann* ein Download-Angebot sein und ein Datei-Namen-Vorschlag, der das nicht als .EXE Datei anbietet. **Der Benutzer sollte die Datei dann nicht ohne weitere Sicherheitsanalyse in eine .EXE umbenennen oder ausführen.** [Rechte s. Anhang.](#)

Multipart MIME Types (1)

Frage: Wie bringt man in einer Email mehrere Teile unter?

Beispiel:

- 1 Anschreiben, das in `text/html` und `text/plain` vorgehalten ist und auf beide Arten dargestellt werden kann – je nach Einstellung des Client.
- 2 3 eingebettete Bilder mit jeweils verschiedenen Codierungen (`bmp`, `png`, `webp`).
- 3 4 weitere Dateien als Attachment.

Antwort: Eine weitere, eigene MIME Type: `multipart`

3. Datenformate & MIME

Multipart MIME Types (2)

```
From: Nathaniel Borenstein <nsb@bellcore.com>  
To: Ned Freed <ned@innosoft.com>  
Subject: Sample message  
MIME-Version: 1.0  
Content-type: multipart/mixed; boundary="simple  
boundary"
```

```
This is the preamble. It is to be ignored, though it  
is a handy place for mail composers to include an  
explanatory note to non-MIME compliant readers.  
--simple boundary
```

```
This is implicitly typed plain ASCII text.  
It does NOT end with a linebreak.  
--simple boundary  
Content-type: text/plain; charset=us-ascii
```

```
This is explicitly typed plain ASCII text.  
It DOES end with a linebreak.
```

```
--simple boundary--  
This is the epilogue. It is also to be ignored.
```

Abb. 13: Beispiel aus der Spezifikation der Multipart MIME Types, [RFC 1341](#). Rechte s. Anhang.

```
From: Nathaniel Borenstein <nsb@bellcore.com>
To: Ned Freed <ned@innosoft.com>
Subject: Sample message
MIME-Version: 1.0
Content-type: multipart/mixed; boundary="simple
boundary"

This is the preamble. It is to be ignored, though it
is a handy place for mail composers to include an
explanatory note to non-MIME compliant readers.
--simple boundary

This is implicitly typed plain ASCII text.
It does NOT end with a linebreak.
--simple boundary
Content-type: text/plain; charset=us-ascii

This is explicitly typed plain ASCII text.
It DOES end with a linebreak.

--simple boundary--
This is the epilogue. It is also to be ignored.
```

Abb. 14: Beispiel aus der Spezifikation der Multipart MIME Types, [RFC 1341](#). Rechte s. Anhang.

Wie werden hier die einzelnen Teile voneinander abgetrennt?

Wie stellt man in der Praxis sicher, daß die boundary nicht Teil der Nachricht ist?

Wie macht man das bei streaming Inhalten?

3. Datenformate & MIME

Wie erkennen Systeme Datentypen? (1) MIME Header

```
Content-Type: image/png  
Content-Disposition: attachment; filename="bild.png"  
Content-Transfer-Encoding: base64
```

Abb. 15: (1) MIME Header, hier in einer Email [Rechte s. Anhang.](#)

```
HTTP/1.1 200 OK  
Content-Type: application/pdf  
Content-Length: 12345
```

Abb. 16: MIME Header, hier in einer http Nachricht an einen Web-Browser [Rechte s. Anhang.](#)

3. Datenformate & MIME

Wie erkennen Systeme Datentypen? (2) Magic Bytes

Dateityp	Typische Signatur (hexadezimal)
PDF	25 50 44 46 2D (%PDF-)
PNG	89 50 4E 47 0D 0A 1A 0A
JPEG	FF D8 FF
GIF	47 49 46 38 (GIF8)
ZIP	50 4B 03 04 (häufig)
GZIP	1F 8B
ELF	7F 45 4C 46
SQLite	53 51 4C 69 74 65 20 66 6F 72 6D 61 74 20 33 00

Abb. 17: (2) Magic Bytes, die zu Beginn einer Datei oder an anderen, vordefinierten Positionen stehen.

Rechte s. Anhang.

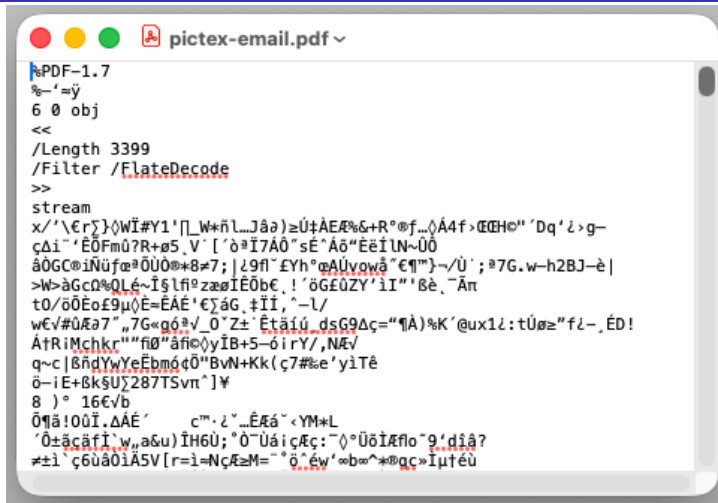
Wie erkennen Systeme Datentypen? (3) Dateinamen & Extension

Dateiname	Bedeutung
program.ts	Typescript Programm
bild.png	image/png
backup.tar.gz	application/gzip, gzip komprimierte Tar-Archiv Datei Verschachtelung kann MIME Spezifikation nicht optimal handhaben
Dockerfile	Docker Image Beschreibung, mutmaßlich text/plain
Makefile	Anweisung für make Werkzeug, mutmaßlich text/plain
README	Mutmaßlich text/plain
README.md	text/markdown

Tab. 1: (3) Dateinamen oder seiner sogenannten Extension.

3. Datenformate & MIME

Wie erkennen Systeme Datentypen? (4) Magic Words



```
%PDF-1.7
%—=ÿ
6 0 obj
<<
/Length 3399
/Filter /FlateDecode
>>
stream
x/''\€rΣ}◊Wİ#Y1'[_W*ñl...Jâa)≥Ú±ÀÆÆ%&+R°®f...◊Á4f>ÆEH©''Dq'¿>g-
çΔi''ÉÖFmû?R+ø5_V'['òªİ7ÁÔ"sÉ^Áô"ÊëİLN~ÜÖ
âÖG©iñüfæªÖÜÖ®*8≠7;|¿9fl~£Yh°æAÜvowâ"€¶"}~÷Ü';ª7G.w-h2BJ-è|
>W>àGcΩ%QLÉ~İ§lfiºzæøİÉÖb€,! 'öGfÜZY'İI''Bè_~Äπ
t0/öÖÈo£9µ◊È=ÊÁÊ'€ΣáG_‡İİ,^~l/
w€√#öÆa7"„7G«qóª√_0°Z±'Êtäİú_dsg9Δç="¶İÀ)%K'@ux1¿:tÜø≥"f¿-_ÉD!
Á†RiMchkr""ñø"âfi©◊yİB+5—óirY/,NÆ√
q~c|BñdYwYeĒbmóqÖ"BvN+Kk(ç7#%e'yİTê
ö-iE+ßk$UΣ287TSvn^]¥
8 )° 16€√b
Ö¶ā!0Üİ.ΔÁÊ' c™·¿°...ÊÆá~<YM*L
'Ö±āçāfi`w„a&u)İH6Ü;°Ö~ÜáicÆç:-◊°ÜöİÆflo`9'diâ?
≠±İ`ç6üâöİÄSV[r=İ=NçÆ≥M="°ö^éw'°bæ^*®qc>İµtêü
```

Abb. 18: (4) Magic Words, die zu Beginn einer Datei stehen. Hier das Beispiel einer PDF Datei, die in einem normalen Texteditor geöffnet wurde. Rechte s. Anhang.

3. Datenformate & MIME

Wie erkennen Systeme Datentypen? (5) Viele weitere

Verfahren	Informationsquelle	Typische Verwendung
Dateiendung	.pdf, .png, .html	Betriebssysteme, Webserver
Dateiname und Muster	Glob, reguläre Ausdrücke	MIME-Datenbanken
Magic Bytes	Signaturen im Dateiinhalt	file, libmagic
Strukturprüfung	Interner Dateiaufbau	Parser, Formatbibliotheken
Containerinspektion	ZIP, RIFF, ISO-BMFF usw.	Office, Audio, Video
Syntaxanalyse	XML, JSON, HTML, SVG usw.	Dokumentenverarbeitung
Zeichensatz-/Textanalyse	UTF-8, Steuerzeichen usw.	Text/Binär-Unterscheidung
Heuristische Analyse	Statistische Eigenschaften	Dateityperkennung
Maschinelles Lernen	Gelernte Inhaltsmerkmale	Forensik, Klassifikation
HTTP-Header	Content-Type	Browser, Downloads
MIME-Nachrichtenheader	Content-Type eines MIME-Parts	E-Mail
Dateisystem-Metadaten	Extended Attributes, Typcodes	Betriebssysteme
Betriebssystem-Typdatenbank	UTI/UTType, MIME-Registry	macOS, Linux
Anwendungs-Metadaten	Dokumenttyp beim Speichern	Office, Editor, Exporter
Protokoll-Metadaten	Multipart, Netzwerkprotokolle	Uploads, APIs
Externe Metadaten	Datenbank, Manifest, Sidecar	Cloud, Archive
Benutzerangabe	Manuelle Typauswahl	Importdialoge
Anwendungskontext	Bekannter Dateityp eines Workflows	Spezialisierte Software

Abb. 19: Lange Liste weiterer Techniken. Quelle: ChatGPT. Rechte s. Anhang.

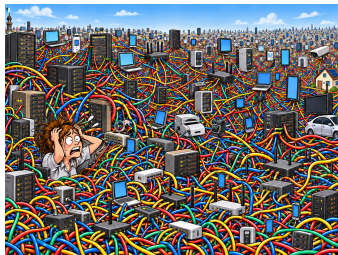


Abb. 20: Komplexität des Internet [Rechte s. Anhang.](#)

Komplexitätsreduktion: Aufteilung in Funktionsblöcke (sog. Schichten oder Layer)

Zwei Arten, die Funktionsblöcke zu kapseln:

- ISO-OSI Modell: Schwerfällig, vollständig, konzeptuell & terminologisch gut ausgearbeitet
- Internet-Modell: Leichtgewichtig, praxisnah, im Internet realisiert

Wir bisher: Anwendungs-Schichte (application layer 7), Darstellungs-Schichte (presentation layer 6)

Unser erstes Schichtenmodell

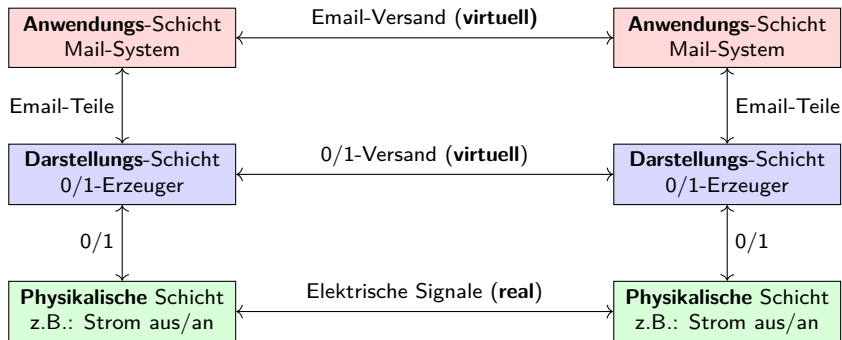


Abb. 21: Unser erstes (naives, didaktisches, eingeschränktes, in der Praxis so weder genutztes noch so bezeichnetes) Schichtenmodell besteht bisher aus **Anwendungs-Schicht** und **Darstellungs-Schicht**. Zur tatsächlichen, **realen** Übertragung werden (hier noch nicht gezeigte) weitere Schichten benötigt, insbesondere die **physikalische** Schichte, welche die Daten nun *wirklich* überträgt. Aöoe anderen Übertragungen sind **virtuell** (scheinbar). Rechte s. Anhang.

4. Datenpakete & Netzdienste

4.1. Datenpakete

4.2. Versenden der Datenpakete

4.3. Netzdienste

Wie werden diese 0-en und 1-en und komplexere Symbole jetzt versendet?

Im **ISO-OSI Modell** wird diese Aufgabe auf weiteren **Layern (1 bis 4)** behandelt.

1. Anwendungen: Eine Email

2. Darstellung: In Bits

3. Datenformate & MIME

4. Datenpakete & Netzdienste

5. Adressen, Teilnehmer & Namen

6. Routing

7. Sitzungen

4.1 Datenpakete

Aufteilung in Pakete

Wir machen aus der Email 0-en und 1-en.

Wir senden die 0-en und 1-en von Rostock nach Wien.

Optionen, die bestehen:

- ① Wir senden die 0-en und 1-en **jeweils einzeln**.
- ② Wir senden alle 0-en und 1-en **in einem einzigen Block**.
- ③ Wir senden die 0-en und 1-en **in einzelnen Paketen**.

Frage: Welche Option wählen wir und vorallem: **Warum?**

Überlege:

- Was wäre, wenn wir die Bits einzeln senden?
- Was wäre, wenn wir alle Bits in einem Block senden?



<https://twbk.de/2b56>

Lösung anzeigen (Adobe PDF Reader nötig!)

6 Gründe für Datenpakete

Antwort zeigen

4.1 Datenpakete

Wie kann ich mir diese Pakete ansehen?

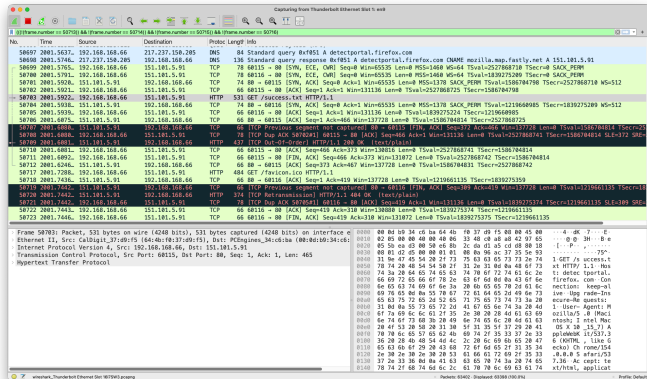


Abb. 23: Packet Tracer Wireshark im Einsatz. Hier bei der Anzeige der Webseite <http://detectportal.firefox.com/success.txt>. Rechte s. Anhang.

Capturing from Thunderbolt Ethernet Slot 1: en0

(((frame.number == 50713)) && ((frame.number == 50714)) && ((frame.number == 50715)) && ((frame.number == 50716)))

No.	Time	Source	Destination	Protocol	Length	Info
50697	2001.5637...	192.168.168.66	217.237.150.205	DNS	84	Standard query 0xf051 A detectportal.firefox.com
50698	2001.5746...	217.237.150.205	192.168.168.66	DNS	136	Standard query response 0xf051 A detectportal.firefox.com CNAME mozilla.map.fastly.net A 151.101.5.91
50699	2001.5765...	192.168.168.66	151.101.5.91	TCP	78	60115 → 80 [SYN, ECE, CWR] Seq=0 Win=65535 Len=0 MSS=1460 WS=64 TSval=2527868710 TSecr=0 SACK_PERM
50700	2001.5791...	192.168.168.66	151.101.5.91	TCP	78	60116 → 80 [SYN, ECE, CWR] Seq=0 Win=65535 Len=0 MSS=1460 WS=64 TSval=1839275209 TSecr=0 SACK_PERM
50701	2001.5920...	151.101.5.91	192.168.168.66	TCP	74	80 → 60115 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1378 SACK_PERM TSval=1586704798 TSecr=2527868710 WS=512
50702	2001.5921...	192.168.168.66	151.101.5.91	TCP	66	60115 → 80 [ACK] Seq=1 Ack=1 Win=131136 Len=0 TSval=2527868725 TSecr=1586704798
50703	2001.5922...	192.168.168.66	151.101.5.91	HTTP	531	GET /success.txt HTTP/1.1
50704	2001.5938...	151.101.5.91	192.168.168.66	TCP	74	80 → 60116 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1378 SACK_PERM TSval=1219660985 TSecr=1839275209 WS=512
50705	2001.5939...	192.168.168.66	151.101.5.91	TCP	66	60116 → 80 [ACK] Seq=1 Ack=1 Win=131136 Len=0 TSval=1839275224 TSecr=1219660985
50706	2001.6075...	151.101.5.91	192.168.168.66	TCP	66	80 → 60115 [ACK] Seq=1 Ack=466 Win=137728 Len=0 TSval=1586704814 TSecr=2527868725
50707	2001.6080...	151.101.5.91	192.168.168.66	TCP	66	[TCP Previous segment not captured] 80 → 60115 [FIN, ACK] Seq=372 Ack=466 Win=137728 Len=0 TSval=1586704814 TSecr=25
50708	2001.6080...	192.168.168.66	151.101.5.91	TCP	78	[TCP Dup ACK 50702#1] 60115 → 80 [ACK] Seq=466 Ack=1 Win=131136 Len=0 TSval=2527868741 TSecr=1586704814 SLE=372 SRE=
50709	2001.6081...	151.101.5.91	192.168.168.66	HTTP	437	[TCP Out-of-Order] HTTP/1.1 200 OK (text/plain)
50710	2001.6081...	192.168.168.66	151.101.5.91	TCP	66	60115 → 80 [ACK] Seq=466 Ack=373 Win=130816 Len=0 TSval=2527868741 TSecr=1586704814
50711	2001.6092...	192.168.168.66	151.101.5.91	TCP	66	60115 → 80 [FIN, ACK] Seq=466 Ack=373 Win=131072 Len=0 TSval=2527868742 TSecr=1586704814
50712	2001.6246...	151.101.5.91	192.168.168.66	TCP	66	80 → 60115 [ACK] Seq=373 Ack=467 Win=137728 Len=0 TSval=1586704831 TSecr=2527868742
50717	2001.7288...	192.168.168.66	151.101.5.91	HTTP	484	GET /favicon.ico HTTP/1.1
50718	2001.7436...	151.101.5.91	192.168.168.66	TCP	66	80 → 60116 [ACK] Seq=1 Ack=419 Win=137728 Len=0 TSval=1219661135 TSecr=1839275359
50719	2001.7442...	151.101.5.91	192.168.168.66	TCP	66	[TCP Previous segment not captured] 80 → 60116 [FIN, ACK] Seq=309 Ack=419 Win=137728 Len=0 TSval=1219661135 TSecr=18
50720	2001.7442...	151.101.5.91	192.168.168.66	HTTP	374	[TCP Retransmission] HTTP/1.1 404 OK (text/plain)
50721	2001.7442...	192.168.168.66	151.101.5.91	TCP	78	[TCP Dup ACK 50705#1] 60116 → 80 [ACK] Seq=419 Ack=1 Win=131136 Len=0 TSval=1839275374 TSecr=1219661135 SLE=309 SRE=
50722	2001.7443...	192.168.168.66	151.101.5.91	TCP	66	60116 → 80 [ACK] Seq=419 Ack=310 Win=130880 Len=0 TSval=1839275374 TSecr=1219661135
50723	2001.7446...	192.168.168.66	151.101.5.91	TCP	66	60116 → 80 [FIN, ACK] Seq=419 Ack=310 Win=131072 Len=0 TSval=1839275375 TSecr=1219661135

> Frame 50703: Packet, 531 bytes on wire (4248 bits), 531 bytes captured (4248 bits) on interface en0
 > Ethernet II, Src: CalDigit_37:d9:f5 (64:4b:f0:37:d9:f5), Dst: PCEngines_34:c6:ba (00:0d:b9:34:c6:ba)
 > Internet Protocol Version 4, Src: 192.168.168.66, Dst: 151.101.5.91
 > Transmission Control Protocol, Src Port: 60115, Dst Port: 80, Seq: 1, Ack: 1, Len: 465
 > Hypertext Transfer Protocol

0000 00 0d b9 34 c6 ba 64 4b f0 37 d9 f5 08 00 45 00 ...4...dK 7....E-
 0010 02 05 00 0d 00 00 00 06 33 48 c0 a8 a8 42 97 65 ...@.@ 3H...B e
 0020 05 5b ea d3 00 50 e6 8b 2c da d1 a5 cd d8 00 18 [...P...
 0030 08 01 d2 d5 00 00 01 01 08 0a 96 ac 37 35 5e 9375-
 0040 31 9e 47 45 54 20 2f 73 75 63 65 73 73 2e 74 1-GET /s success-
 0050 78 74 20 48 54 50 2f 31 2e 31 0d 0a 48 6f 73 xt HTTP/1.1: Hos
 0060 74 3a 20 64 65 74 65 63 74 70 6f 72 74 61 6c 2e tect: tportal.
 0070 66 69 72 65 66 6f 78 2e 63 6f 6d 0d 43 6f 6e firefox.com: Con
 0080 6e 65 63 74 69 6f 6e 3a 20 6b 65 65 70 2d 61 6c nection: upg-
 0090 69 76 65 0d 0a 55 70 67 72 61 64 65 2d 49 6e 73 ivate: Upgrade-Ins
 00a0 65 63 75 72 65 2d 52 65 71 75 65 73 74 73 3a 20 equire-Requests:
 00b0 31 0d 0a 55 73 65 72 2d 41 67 65 6e 74 3a 20 4d 1-User-Agent: M
 00c0 6f 7a 69 6c 6c 61 2f 35 2e 30 20 28 4d 61 63 69 ozilla/5.0 (Maci
 00d0 6e 74 6f 73 68 3b 20 49 6f 74 65 6c 20 4d 61 63 ntel Mac
 00e0 20 4f 53 20 58 20 31 30 5f 31 35 5f 37 29 20 41 OS X 10_15.7) A
 00f0 70 70 6c 65 57 65 62 4b 69 74 2f 35 33 67 2e 33 ppleWebKit/537.3
 0100 36 20 28 4b 48 54 4d 4c 2c 20 6c 69 6b 65 20 47 6 (KHTML, like G
 0110 65 63 6b 6f 29 20 43 68 72 6f 6d 65 2f 31 35 34 ecko) Chrome/154
 0120 2e 30 2e 30 2e 30 20 53 61 66 61 72 69 2f 35 33 .0.0.0 Safari/53
 0130 37 2e 33 36 0d 0a 41 63 63 65 70 74 3a 20 74 65 7.36- Accept: te
 0140 78 74 2f 68 74 6d 6c 2c 61 70 70 6c 69 63 61 74 xt/html, applicat

wireshark_Thunderbolt Ethernet Slot 16/7SW3.pcapng
 Packets: 63402 - Displayed: 63398 (100.0%)
 Profile: Default

“**Schweizer Taschenmesser**” für den Informatiker in Kommunikation.

- Greift (fast) alle Pakete auf Betriebssystem-Ebene ab.
- Filtert, sortiert & stellt diese nach verschiedensten Kriterien dar.
- Beherrscht **über 3000 Protokolle!**

Problem: Die allermeisten Verbindungen sind heutzutage verschlüsselt.

Kein Problem:

- Client soll die Daten nutzen, kann Sie daher (meist) (irgendwie) (wie?) entschlüsseln.
- Dazu gibt es gute Unterstützung in Wireshark.
- Das ist je nach Protokoll immer etwas anders.

Aufgabe

Aufgabe für alle: (Dringend empfohlen!)

- Video ansehen: [Wireshark verstehen](#)
- Wireshark installieren und einige Pakete ansehen.

Vorsicht, Warnung!

Mitschneiden von Daten Dritter und Nutzen nicht für einen selbst bestimmter Daten (Passwörter!) wird in Deutschland strafrechtlich verfolgt.

Daher: Nur im eigenen Netz oder spezifischen Laborumgebungen durchführen.

Bonus-Aufgabe: Weitere Tutorials für Wireshark

- [Wireshark Tutorial](#), [Wireshark Tutorial](#)

Doppel-Bonus-Aufgabe für Infos & ITTIs: Wireshark bei Verschlüsselung

- [Text 1](#), [Text 2](#), [Video 1](#), [Video 2](#)

4.1 Datenpakete

Packet Trace einer http-Verbindung (1)

50697	2001.5637...	192.168.168.66	217.237.150.205	DNS	84	Standard query 0xf051 A detectportal.firefox.com
50698	2001.5746...	217.237.150.205	192.168.168.66	DNS	136	Standard query response 0xf051 A detectportal.firefox.com CNAME mozilla.map.fastly.net A 151.101.5.91
50699	2001.5765...	192.168.168.66	151.101.5.91	TCP	78	60115 → 80 [SYN, ECE, CWR] Seq=0 Win=65535 Len=0 MSS=1460 WS=64 TSval=2527868710 TSecr=0 SACK_
50700	2001.5791...	192.168.168.66	151.101.5.91	TCP	78	60116 → 80 [SYN, ECE, CWR] Seq=0 Win=65535 Len=0 MSS=1460 WS=64 TSval=1839275209 TSecr=0 SACK_
50701	2001.5920...	151.101.5.91	192.168.168.66	TCP	74	80 → 60115 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1378 SACK_PERM TSval=1586704798 TSecr=25
50702	2001.5921...	192.168.168.66	151.101.5.91	TCP	66	60115 → 80 [ACK] Seq=1 Ack=1 Win=131136 Len=0 TSval=2527868725 TSecr=1586704798
50703	2001.5922...	192.168.168.66	151.101.5.91	HTTP	531	GET /success.txt HTTP/1.1
50704	2001.5938...	151.101.5.91	192.168.168.66	TCP	74	80 → 60116 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1378 SACK_PERM TSval=1219660985 TSecr=18
50705	2001.5939...	192.168.168.66	151.101.5.91	TCP	66	60116 → 80 [ACK] Seq=1 Ack=1 Win=131136 Len=0 TSval=1839275224 TSecr=1219660985
50706	2001.6075...	151.101.5.91	192.168.168.66	TCP	66	80 → 60115 [ACK] Seq=1 Ack=466 Win=137728 Len=0 TSval=1586704814 TSecr=2527868725
50707	2001.6080...	151.101.5.91	192.168.168.66	TCP	66	[TCP Previous segment not captured] 80 → 60115 [FIN, ACK] Seq=372 Ack=466 Win=137728 Len=0 TSv
50708	2001.6080...	192.168.168.66	151.101.5.91	TCP	78	[TCP Dup ACK 50702#1] 60115 → 80 [ACK] Seq=466 Ack=1 Win=131136 Len=0 TSval=2527868741 TSecr=1
50709	2001.6081...	151.101.5.91	192.168.168.66	HTTP	437	[TCP Out-Of-Order] HTTP/1.1 200 OK (text/plain)
50710	2001.6081...	192.168.168.66	151.101.5.91	TCP	66	60115 → 80 [ACK] Seq=466 Ack=373 Win=130816 Len=0 TSval=2527868741 TSecr=1586704814
50711	2001.6092...	192.168.168.66	151.101.5.91	TCP	66	60115 → 80 [FIN, ACK] Seq=466 Ack=373 Win=131072 Len=0 TSval=2527868742 TSecr=1586704814
50712	2001.6246...	151.101.5.91	192.168.168.66	TCP	66	80 → 60115 [ACK] Seq=373 Ack=467 Win=137728 Len=0 TSval=1586704831 TSecr=2527868742
50717	2001.7288...	192.168.168.66	151.101.5.91	HTTP	484	GET /favicon.ico HTTP/1.1
50718	2001.7436...	151.101.5.91	192.168.168.66	TCP	66	80 → 60116 [ACK] Seq=1 Ack=419 Win=137728 Len=0 TSval=1219661135 TSecr=1839275359
50719	2001.7442...	151.101.5.91	192.168.168.66	TCP	66	[TCP Previous segment not captured] 80 → 60116 [FIN, ACK] Seq=309 Ack=419 Win=137728 Len=0 TSv
50720	2001.7442...	151.101.5.91	192.168.168.66	HTTP	374	[TCP Retransmission] HTTP/1.1 404 OK (text/plain)
50721	2001.7442...	192.168.168.66	151.101.5.91	TCP	78	[TCP Dup ACK 50705#1] 60116 → 80 [ACK] Seq=419 Ack=1 Win=131136 Len=0 TSval=1839275374 TSecr=1
50722	2001.7443...	192.168.168.66	151.101.5.91	TCP	66	60116 → 80 [ACK] Seq=419 Ack=310 Win=130880 Len=0 TSval=1839275374 TSecr=1219661135
50723	2001.7446...	192.168.168.66	151.101.5.91	TCP	66	60116 → 80 [FIN, ACK] Seq=419 Ack=310 Win=131072 Len=0 TSval=1839275375 TSecr=1219661135

Abb. 24: Zeitlicher Ablauf beim Abruf der Webseite <http://detectportal.firefox.com/success.txt>. (1) DNS-Anfrage übersetzt Rechnernamen detectportal.firefox.com in IP Adresse. (2) Aufbau TCP-Verbindung (syn, syn ack, ack). (3) Abholen einzelner Inhalte mit HTTP. (4) Fehler, Duplikate, Reihenfolgenvertauschung. (5) Schließen der TCP Verbindung (ack, fin ack). Rechte s. Anhang.

Packet Trace einer http-Verbindung (2)

```
> Frame 50703: Packet, 531 bytes on wire (4248 bits), 531 bytes captured (4248 bits) on interface e
> Ethernet II, Src: CalDigit_37:d9:f5 (64:4b:f0:37:d9:f5), Dst: PCEngines_34:c6:ba (00:0d:b9:34:c6:
> Internet Protocol Version 4, Src: 192.168.168.66, Dst: 151.101.5.91
> Transmission Control Protocol, Src Port: 60115, Dst Port: 80, Seq: 1, Ack: 1, Len: 465
> Hypertext Transfer Protocol
```

Abb. 25: Einzelne Abstraktionsschichten, hier weder nach ISO-OSI noch nach Internet sortiert sondern nach allen für Wireshark sinnvoll ersichtlichen Schichten. [Rechte s. Anhang.](#)

4.1 Datenpakete

Packet Trace einer http-Verbindung (3)

```
> Frame 50703: Packet, 531 bytes on wire (4248 bits), 531 bytes captured (4248 bits) on interface en9, id 0
> Ethernet II, Src: CalDigit_37:d9:f5 (64:4b:f0:37:d9:f5), Dst: PCEngines_34:c6:ba (00:0d:b9:34:c6:ba)
> Internet Protocol Version 4, Src: 192.168.168.66, Dst: 151.101.5.91
> Transmission Control Protocol, Src Port: 60115, Dst Port: 80, Seq: 1, Ack: 1, Len: 465
    Source Port: 60115
    Destination Port: 80
    [Stream index: 298]
    [Stream Packet Number: 4]
    > [Conversation completeness: Complete, WITH_DATA (31)]
    [TCP Segment Len: 465]
    Sequence Number: 1 (relative sequence number)
    Sequence Number (raw): 3867880666
    [Next Sequence Number: 466 (relative sequence number)]
    Acknowledgment Number: 1 (relative ack number)
    Acknowledgment number (raw): 3517304280
    1000 ... = Header Length: 32 bytes (8)
    > Flags: 0x018 (PSH, ACK)
    Window: 2049
    [Calculated window size: 131136]
    [Window size scaling factor: 64]
    Checksum: 0xd2d5 [unverified]
    [Checksum Status: Unverified]
    Urgent Pointer: 0
    > Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
    > [Timestamps]
    > [SEQ/ACK analysis]
    [Client Contiguous Streams: 1]
    [Server Contiguous Streams: 1]
    TCP payload (465 bytes)
> Hypertext Transfer Protocol
```

Abb. 26: Einzelne Abstraktionsschichten, hier mit der Transport-Schichte (TCP) in allen Details angezeigt.

Rechte s. Anhang.

4.1 Datenpakete

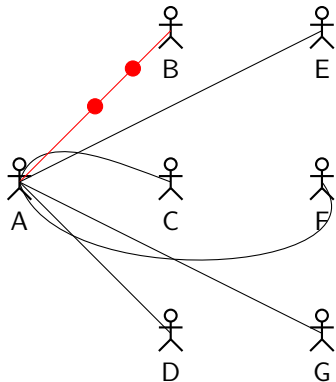
Packet Trace einer http-Verbindung (4)

0000	00 0d b9 34 c6 ba 64 4b f0 37 d9 f5 08 00 45 00	...4...dK...7...E.
0010	02 05 00 00 40 00 40 06 33 48 c0 a8 a8 42 97 65	@.@ 3H...B.e
0020	05 5b ea d3 00 50 e6 8b 2c da d1 a5 cd d8 80 18	.[...P... ,.....
0030	08 01 d2 d5 00 00 01 01 08 0a 96 ac 37 35 5e 93	 75^.
0040	31 9e 47 45 54 20 2f 73 75 63 63 65 73 73 2e 74	1·GET /s uccess.t
0050	78 74 20 48 54 54 50 2f 31 2e 31 0d 0a 48 6f 73	xt HTTP/ 1.1·Hos
0060	74 3a 20 64 65 74 65 63 74 70 6f 72 74 61 6c 2e	t: detec tportal.
0070	66 69 72 65 66 6f 78 2e 63 6f 6d 0d 0a 43 6f 6e	firefox. com·Con
0080	6e 65 63 74 69 6f 6e 3a 20 6b 65 65 70 2d 61 6c	nection: keep-al
0090	69 76 65 0d 0a 55 70 67 72 61 64 65 2d 49 6e 73	ive·Upg rade-Ins
00a0	65 63 75 72 65 2d 52 65 71 75 65 73 74 73 3a 20	ecure-Re quests:
00b0	31 0d 0a 55 73 65 72 2d 41 67 65 6e 74 3a 20 4d	1·User- Agent: M
00c0	6f 7a 69 6c 6c 61 2f 35 2e 30 20 28 4d 61 63 69	ozilla/5 .0 (Maci
00d0	6e 74 6f 73 68 3b 20 49 6e 74 65 6c 20 4d 61 63	ntosh; I ntel Mac
00e0	20 4f 53 20 58 20 31 30 5f 31 35 5f 37 29 20 41	OS X 10 _15_7) A
00f0	70 70 6c 65 57 65 62 4b 69 74 2f 35 33 37 2e 33	ppleWebK it/537.3
0100	36 20 28 4b 48 54 4d 4c 2c 20 6c 69 6b 65 20 47	6 (KHTML , like G
0110	65 63 6b 6f 29 20 43 68 72 6f 6d 65 2f 31 35 34	ecko) Ch rome/154
0120	2e 30 2e 30 2e 30 20 53 61 66 61 72 69 2f 35 33	.0.0.0 S afari/53
0130	37 2e 33 36 0d 0a 41 63 63 65 70 74 3a 20 74 65	7.36·Ac cept: te
0140	78 74 2f 68 74 6d 6c 2c 61 70 70 6c 69 63 61 74	xt/html, applicat

Abb. 27: ASCII und hexadezimale Darstellung. Diese ist bei Protokollbestandteilen wenig aussagekräftig, hilft aber beim Lesen von Texten und der eigenen detaillierten Analyse. [Rechte s. Anhang.](#)

4.2 Versenden der Datenpakete

Leitungsvermittlung



Leitungsvermittlung:

- Sender hat zu jedem Empfänger seine eigene Verbindung.
- Sehr hoher Leitungsaufwand!
- Senden bedeutet Auswahl der richtigen direkten Leitung.
- Pakete können sich nicht gegenseitig überholen.

Abb. 28: Leitungsvermittelltes Netz [Rechte s. Anhang](#).

4.2 Versenden der Datenpakete

Leitungsvermittlung im Internet?



Abb. 29: Warum Leitungsvermittlung im Internet nicht geht. Rechte s. Anhang.

4.2 Versenden der Datenpakete

Paketvermittlung

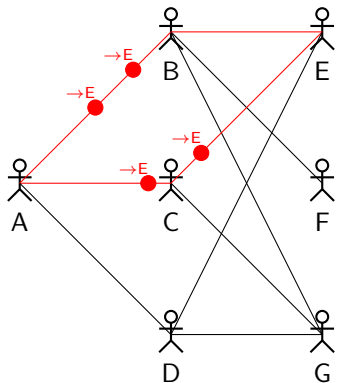


Abb. 30: Paketvermitteltes Netz. Rechte s. Anhang.

Paketvermittlung:

- Pakete suchen sich auf **vielen Wegen** durch das Netz ihr Ziel.
- Pakete werden durch **Zwischenknoten** weitervermittelt.
- Pakete können sich dabei **gegenseitig überholen**.
- Bei hohen Verlustraten können Kopien auf mehreren Wegen versendet werden.
- **Mehrfachauslieferung** ist möglich kann durch Protokolle verhindert werden.

4.2 Versenden der Datenpakete

Simulierte Leitungsvermittlung

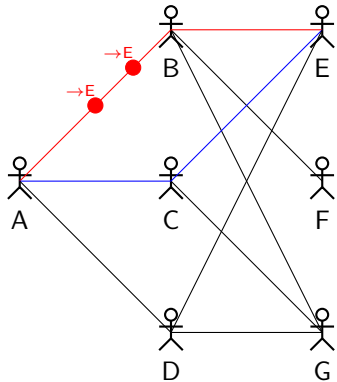


Abb. 31: Simulierte Leitungsvermittlung. Rechte s. Anhang.

Simulierte Leitungsvermittlung:

- Vor dem Senden wird auf einem paketvermittelten Netz eine **virtuelle Verbindung** (virtual circuit) bestimmt und “durchgeschaltet”.
- Die virtuelle Verbindung wird wie bei Leitungsvermittlung genutzt.
- Bricht Verbindung zusammen, kann auf eine **andere** bereits vorgehaltene gewechselt werden, (sog. **alternative routing**) oder neue dynamisch bestimmt werden (sog. **adaptive routing**).
- Sucht Vorteile beider Welten, Leitungs- und Paketvermittlung.

4.2 Versenden der Datenpakete

Tweedback

Bewahren paketvermittelte Netzdienste die Reihenfolge?

Sind paketvermittelte Netzdienste Verlust-frei?



<https://twbk.de/2b56>

Unabhängig davon, wie das Netz tatsächlich vermittelt,
können Kommunikationsdienste mit verschiedenen Garantien verfügbar sein.

Verbindungslose Netzdienste (Connectionless Network Services)

Vorstellung: Einzelne Datenpakete auf ihrem Weg durch das Netz.

Individueller Versand einzelner Pakete an ihre Empfänger.

Typisch für paketvermittelte Kommunikation.

Verbindungsorientierte Netzdienste (Connection Oriented Network Services)

Vorstellung: Eine Leitung verbindet zwei Kommunikationspartner.

Erfordert in paketvermittelten Netzen den Aufbau virtueller Verbindungen.

Frage: Was sind die wesentlichen Eigenschaften dieser Netzdienste?

5 Eigenschaften von Netzdiensten (1)

Integrität:

- Wenn Daten ankommen, dann kommen sie ohne Fehler an.
- **Mögliches Problem:** Störungen in Bits

Reihenfolge erhalten:

- Die Reihenfolge der Daten bleibt erhalten.
- **Mögliches Problem:** Daten nehmen verschiedene Wege durch das Netz.
Früher abgesandte Daten können später ankommen als später abgesandte.

Verlust-frei:

- Alle Daten kommen an.
- **Mögliches Problem:** Daten gehen wegen technischer Fehler verloren oder werden wegen Überlastung bewußt fallengelassen.

5 Eigenschaften von Netzdiensten (2)

Verdoppelungs-frei:

- Einmal abgesandte Daten kommen auch nur einmal an.
- **Mögliches Problem:** Aus Gründen der Redundanz, Fehlerbehebung oder Effizienz werden Daten manchmal mehrfach versendet und können dann mehrfach ankommen.

Kommata:

- Daten kommen mit jenen Grenzsetzungen (Pausen oder "Kommata") an, mit denen sie abgeschickt wurden.
- **Mögliches Problem:** Beschränkte Puffergrößen lassen es nicht zu, eine Datenmenge von 10 [TByte] als eine einzige Einheit zu übertragen. Die Datenmenge wird aufgeteilt. Zwischen den Teilen treten Wartezeiten auf, wodurch zusätzliche Grenzen entstehen.

Eselsbrücke für die 5 Eigenschaften

Integre

Reihenfolgen

Verlieren

Verdoppelte

Kommata



Abb. 32: Eselsbrücke. Rechte s. Anhang.

Typische Transport-Dienste

Die 2 wichtigsten Netzdienste im Internet sind:

TCP: Transmission Control Protocol.

Steht hier für den **typischen verbindungsorientierten** Transportdienst.

UDP: User Datagram Protocol.

Steht hier für den **typischen verbindungslosen** Transportdienst.

Vorsicht, Falle!

TCP und UDP gehören zur Schicht 4 (Transport Layer) des ISO-OSI Modells, weil sie Teilnehmer adressieren und nicht Netzwerk-Knoten.

Eine Zuordnung zu Schicht 3 (Network Layer, Netzwerk-Schichte) geschieht öfters, ist aber falsch.

Wir bezeichnen hier TCP und UDP (auch) (etwas unscharf) als Netzdienste, weil hier die Trennung Schicht 3 / 4 nicht im Vordergrund steht.

Kerneigenschaften typischer Transportdienste

Eigenschaft	TCP	UDP
Integrität erhalten	Ja	Ja
Reihenfolge erhalten	Ja	Nein
Verlust-frei	Ja	Nein
Verdoppelungs-frei	Ja	Nein
Kommata erhalten	Nein	Nein

Tab. 2: Kerneigenschaften typischer Transportdienste

Integritätsprüfung: Kann bei UDP kann auf Treiberebene abgeschaltet werden.

Kommata: Können wegen endlicher Puffer in gar keinem Protokoll erhalten werden.

Welche Bedeutung hat das?

Welches Protokoll nutze ich für **Email**?

- UDP
- TCP

Welches Protokoll nutze ich für einen **Zeitdienst**?

- UDP
- TCP

Welches Protokoll nutze ich für eine **Videokonferenz**?

- UDP
- TCP

Und vorallem: **Warum?**



<https://twbk.de/2b56>

Ein abschließender didaktischer Hinweis zu den Kommata.

Kommata sind hier meine Begrifflichkeit, sie werden das kaum in Büchern finden.

Da Kommata von keinem Protokoll eingehalten werden, buffern viele Bibliotheksfunktionen automatisch.

Niedrigere Funktionen (etwa: Socket API des Betriebssystems) buffern nicht, übertragen nicht immer alles und melden zurück, wie viel übertragen wurde.

Das hat dann diese Folge:

- ① Sie programmieren ein System mit der Socket API und denken nicht an diesen Effekt.
- ② Sie testen ein System im lokalen oder regionalen Netz – und dort funktioniert es.
- ③ Nach dem Ausrollen in das globale Internet werden gelegentlich Daten verschluckt, weil die größeren Latenzzeiten und geringeren Bandbreiten bewirken, daß nicht immer alles übertragen wird.

Wenn Sie um diesen Effekt wissen, dann haben Sie die Socket API entsprechend selber gebuffert oder gute Bibliotheksfunktionen genutzt.

Und damit Sie nicht in diese Falle laufen, bringe ich das in der Vorlesung.

Und damit Sie sich das leichter merken, gebe ich dem Ding einen Namen.

Note that a successful `write()` may transfer fewer than *count* bytes. Such partial writes can occur for various reasons; for example, because there was insufficient space on the disk device to write all of the requested bytes, or because a blocked `write()` to a socket, pipe, or similar was interrupted by a signal handler after it had transferred some, but before it had transferred all of the requested bytes. In the event of a partial write, the caller can make another `write()` call to transfer the remaining bytes. The subsequent call will either transfer further bytes or may result in an error (e.g., if the disk is now full).

Abb. 33: Die `man page` des Linux `write` API-Aufruf macht *eigentlich* deutlich auf das Problem mit den Kommata aufmerksam. Genau dieser Abschnitt beschreibt das Problem. [Rechte s. Anhang.](#)

5. Adressen, Teilnehmer & Namen

5.1. Konzepte

5.2. MAC, IPv4 und IPv6 Adressen

Adressen und Namen **identifizieren** die an einer Kommunikation teilnehmenden **Komponenten**.

1. Anwendungen: Eine Email
2. Darstellung: In Bits
3. Datenformate & MIME
4. Datenpakete & Netzdienste
5. Adressen, Teilnehmer & Namen
6. Routing
7. Sitzungen

5.1 Konzepte

PHY Adressen

Auf fast jeder Schichte des ISO-OSI Modells¹ gibt es Adressen bzw. Adress-Konzepte.

Physikalische Adresse (Layer 1: Physical Layer)

Identifiziert eine Komponente anhand des physikalischen Übertragungswegs oder der spezifischen Hardware-Schnittstelle

Didaktisches Beispiel: Nummer des Kabels in der Wand.

Technisches Beispiel: Index einer NetzwerkSchnittstelle.

Predictable Network Interface Names

¹...das wir noch näher kennen lernen werden...

Datalink-Layer Adresse (Layer 2: Datalink-Layer)

Identifiziert ein **Endgerät innerhalb** eines (“lokalen”) Netzes einer Technologie.

Bsp: Ethernet, Bluetooth, Wifi, ...

MAC (Media Access Control) Adresse auf dem Medium einer bestimmten Technologie.

Internet Adresse (Layer 3: Network-Layer)

Identifiziert ein **Endgerät netzübergreifend** im Internet (Netz der Netze).

IP (Internet Protokoll) Adresse im Internet.

Vorsicht, Falle

Auf dem Network-Layer nutzen wir Internet Adressen.

Bereits auf dem Datalink-Layer können wir in einem (lokalen) Netz kommunizieren.

Der Network-Layer verbindet mehrere Netzwerke und ist daher eigentlich ein Inter-Network-Layer.

5.1 Konzepte

Beispiel für Adressen auf einem System

```
|cap@eurydice-2 tmp % ifconfig en0
en0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
    options=6460<TSO4,TSO6,CHANNEL_IO,PARTIAL_CSUM,ZEROINVERT_CSUM>
    ether fe:ed:8c:bb:de:05
    inet6 fe80::48a:24f8:c9d1:dba0%en0 prefixlen 64 secured scopeid 0xf
    inet 192.168.2.29 netmask 0xffffffff00 broadcast 192.168.2.255
    inet6 fdc0:2ee9:5047:0:1437:5c33:eea8:1d77 prefixlen 64 autoconf secured
    inet6 2003:e7:571e:7500:84:c3f6:21f1:4e4 prefixlen 64 autoconf secured
    inet6 2003:e7:571e:7500:45c0:9f8e:5c2f:7562 prefixlen 64 autoconf temporary
    nd6 options=201<PERFORMNUD,DAD>
    media: autoselect
    status: active
```

Abb. 34: Auf diesem Rechner existiert eine Ethernet-Schnittstelle mit Layer 1 **PHY Adresse** en0. Der Befehl **ifconfig** zeigt Details ihrer Konfiguration: Sie hat eine Layer 2 (ether) **MAC Adresse** fe:ed:8c:bb:de:05 und ist für Layer 3 auf die **IPv4 Adresse** 192.168.2.29 konfiguriert und (unter anderem) auf die **IPv6 Adresse** 2003:e7:571e:7500:84:c3f6:21f1:4e4. [Rechte s. Anhang.](#)

ifconfig: Was macht der Kommandozeilen-Befehl `ifconfig`?

Musterlösung: <https://en.wikipedia.org/wiki/Ifconfig>

Andere: Welche anderen, ähnlichen Befehle gibt es?

Musterlösung: ChatGPT / Claude / Grok / Gemini fragen.

Ausgaben: Was bedeuten die Ausgaben dieser Webseite: <https://ifconfig.me/>

Interpretation: Was bedeutet in der obigen Ausgabe dieser Zeile:

```
inet 192.168.2.29 netmask 0xffffffff00 broadcast 192.168.2.255
```

- Was genau bedeutet die `netmask` Ausgabe?
- Was genau bedeutet die `broadcast` Ausgabe?
- Was bedeutet die Angabe `192.168.2.29/24`?
- Bonusaufgabe: Kopieren Sie den Output von `ifconfig` (eigener Rechner oder vorherige Folie) in einen ChatGPT Prompt, bitten Sie um eine Erklärung aller Felder.

Windows: Hat oft kein `ifconfig`. Was nutze ich denn da?

Zur Vermeidung von Mißverständnissen

Die gelegentlichen Aufgaben dienen nicht nur der Illustration und Auflockerung. Sie sollen Sie tatsächlich auch bearbeiten.

Bonusaufgaben dienen der Ergänzung und sind *optional*, aber empfohlen.

Begründung:

- Sie sollen dabei praktisch lernen, wie man recherchiert.
Google, Duck Duck Go, Wikipedia, ChatGPT, Claude, Gemini, Grok
- In den konkreten Aufgaben erwerben Sie sich auch wichtige Grundlagen die Bestandteil einer Prüfung sein kann.
- Bei Problemen können Sie auch in den Übungen nachfragen und diskutieren.

Eine mögliche Prüfungsaufgabe

Ermittle die netmask in hexadezimaler Form für die Adreßangabe 192.168.2.13/16.
Man muß das häufig im Kopf konvertieren beim Konfigurieren.

Musterlösung: 0xffff0000

Teilnehmer Adresse (Layer 4: Transport-Layer)

Identifiziert einen **Teilnehmer** (statt nur ein Endgerät)

Bsp: Port-Nummer (bzw: IP –Adresse plus Port-Nummer)

Sitzungs-Adresse (Layer 5: Session-Layer)

Identifiziert eine **Sitzung** eines Teilnehmers.

Bsp: Session-Id in einem Cookie oder im URL-query Teil.

Anwendungs-Adresse (Layer 7: Application-Layer)

Identifiziert ein **Element der Anwendung**.

Bsp: Email Adresse (bei SMTP, POP-3, IMAP-4 Protokoll).

Bsp: URL (bei HTTP, FTP Protokoll).

Eine Adresse kann auch **mehrere Kommunikations-Partner** identifizieren.

Bsp: MAC-Broadcast

Adressiert **alle** Endgeräte eines (OSI-2) Netz(bereichs).

Bsp: IP-Multicast

Adressiert alle Endgeräte einer Gruppe.

Anmeldung als Mitglied an einer Gruppen-Adresse erforderlich.

Bsp: Anycast

Adressiert einen Partner aus einer Gruppe, denen alle dieselbe Adresse zugeordnet ist.

Typischerweise erfolgt Kommunikation mit dem nächstgelegenen Partner.

Bsp: Unicast

Bezeichnet keine Gruppen-Adresse sondern die klassische Situation, in der eine Adresse *genau einen* Kommunikations-Partner identifiziert.

Inhaltsdeskriptoren:

Jeder Teilnehmer sendet Inhalte an alle anderen Teilnehmer.

Die Inhalte haben Deskriptoren der Nutzlast.

Teilnehmer betrachten nur jene Informationen an denen sie aufgrund der Deskriptoren interessiert sind.

Beispiel: Engine-Speed bei einer Maschine.

Bsp: CAN-Bus:

Je nach Version 2¹¹ oder 2²⁹ verschiedene Deskriptoren möglich.

MAC-Adresse: Aufbau und Darstellung

Hier: Ethernet MAC Adressen.

- 6 Oktette (Bytes), durch Doppelpunkte abgetrennt
- **Beispiel:** 00:1A:2B:3C:4D:5E
- **Länge:** 48 Bit = $6 \cdot 8$ Bytes
- **Adreßraum:** $2^{48} \approx 2,8 \cdot 10^{14}$
- Oktette 1–3: **OUI** Organizationally Unique Identifier
vergeben durch IEEE Registration Authority
- Oktette 4–6: Vergabe durch den Hersteller

Muß im lokalen Netz eindeutig sein. (Diskussion: Warum eigentlich?)

IPv4 und IPv6: Was ist das?

IPv4: Erste stabile Version des Internet Protokolls.

IPv6: Aktuellste Version, 1998 eingeführt.

Wichtigster Grund: Adreßmangel bei IPv4.

Derzeit: Koexistenz IPv4 und IPv6,
da in beiden Versionen viele Systeme bestehen.

Neuerungen in IPv6:

- Deutliche Vergrößerung des Adreßraums von 32 bit auf 128 bit.
- Signifikante Verbesserungen in Skalierbarkeit, Sicherheit, Effizienz.
- Bessere Unterstützung für Dienstqualität und Echtzeit-Kommunikation.
- Implementierung kommt zusammen mit Sicherheits-Mechanismen (IPsec).

IPv4-Adresse: Aufbau und Darstellung

IPv4 Internet Adressen:

- 4 Oktette (Bytes), durch Punkte abgetrennt
- **Beispiel** 192.168.10.37 = 11000000.10101000.00001010.00100101
- **Länge:** 32 Bit = 4 · 8 Bytes
- **Adreßraum:** $2^{32} \approx 4,3 \cdot 10^9$

Netzaufteilung in Adreßbereiche

- **Vor 1993:** Jede Firma verwaltet eigenes Netz, 3 Standardgrößen (A,B,C)
- **Nach 1993:** CIDR Notation erlaubt flexiblere Aufteilung.
- **Netzanteil:** Präfix, dient der Identifikation des Netzwerks
- **Hostanteil:** Suffix, dient der Identifikation des Rechners im Netz
- **Maske:** Subnetmaske markiert Bits, die für Netzwerk zuständig sind
- **Beispiel:** 192.168.10.1/24 Die ersten 24 Bit dienen als Präfix
 11111111.11111111.11111111.00000000 = ffffffff00 = /24
- **Aufgabe:** Im **Subnetz Rechner** praktisch anschauen!

IPv4-Adresse: Spezielle Adreßbereiche

Adreßblock	Verwendung	Quelle
0.0.0.0/32	Hat noch keine Adresse	RFC1122
10.0.0.0/8	Privater Adreßraum	RFC1918
172.16.0.0/12	Privater Adreßraum	RFC1918
192.168.0.0/16	Privater Adreßraum	RFC1918
127.0.0.0/8	Loopback	RFC1122, Localhost
169.254.0.0/16	Link-Local	RFC3927
224.0.0.0/4	Multicast	RFC1112
255.255.255.255/32	Broadcast	RFC919

Tab. 3: IPv4 Adressen müssen **im globalen Internet eindeutig** sein (**Warum eigentlich?**). **Ausnahmen:** (1) Private Adressen ohne Weiterleitung im öffentlichen Internet. (2) Lokale adressen für den eigenen Host oder den lokalen Link. (3) Gruppenadressen. Weitere Spezialbereiche: **Special-Purpose Address Space**, **Multicast Address Space**.

IPv6-Adresse: Aufbau und Darstellung

IPv6 Internet Adressen:

- **Notation:** 8 Gruppen zu 16 Bit (4 Hexadezimalziffern), durch Doppelpunkte abgetrennt
- **Beispiel:** 2001:0db8:85a3:0000:0000:8a2e:0370:7334.
- **Länge:** 128 Bit = $8 \cdot 16\text{Bit}$
- **Adreßraum:** $2^{128} \approx 3,4 \cdot 10^{38}$
340 Sextillionen Adressen (340.282.366.920.938.463.374.607.431.768.211.456).
Jeder Mensch: 42 Oktillionen Adressen (42.000.000.000.000.000.000.000.000.000).
Jeder Quadratmillimeter Erdoberfläche: Mehrere Billionen Adressen.
- **Aufbau:** 64 Bit **Präfix** + 64 Bit **Interface-Identifikator** (IID)

Kürzungsregeln

- ① Entfernung führender Nullen in jeder Gruppe
- ② Ersatz der längsten Folge von mindestens zwei Nullgruppen durch :: einmal pro Adresse, im Zweifel die erste Folge
- ③ 2001:0db8:0000:0000:0000:ff00:0042:8329 → 2001:db8::ff00:42:8329

IPv6-Adresse: Spezielle Adreßbereiche

Präfix	Adreßtyp	Quelle
::/128	Hat noch keine Adresse	RFC4291
fc00::/7	Privater Adreßraum	RFC4291
::1/128	Loopback	RFC4291
fe80::/10	Link-Local	RFC4291
ff00::/8	Multicast	RFC4291

Tab. 4: Wichtige IPv6-Adreßtypen. Die Slash-Notation gibt die Länge des Präfixes für den entsprechenden Bereich an. Weitere spezialbereiche: [IANA IPv6 Address Space](#), [IANA IPv6 Special-Purpose Address Space](#).

Konzept des Teilnehmers

Kommunikationspartner auf Layer 3: Endgerät.

- Jedes Gerät hat (oft) (nur) eine Adresse.
- Rechner, Router, Drucker, Lichtschalter, IoT-Gerät.
- Ausnahme 1: Geräte mit mehr als einer Netzwerk-Karte.
- Ausnahme 2: Netzwerk-Karte mit mehr als einer Adresse einer Art konfiguriert.
- Ausnahme 3: Netzwerk-Karte mit IPv4 und IPv6 Adresse konfiguriert.

Kommunikationspartner auf Layer 4: Teilnehmer.

- Bedürfnis nach Untergliederung des Adreßraums im Endgerät.

Beispiele für Teilnehmer

Telekommunikations-Welt: Aufteilung eines Endgeräts in mehrere Funktionen:
Neben "Telefon" treten "Fax, Anrufbeantworter, Kurznachricht".

Zunächst sind es historisch alles einzelne Geräte.

Neues Konzept: Hauptnummer mit Durchwahl/Nachwahl-Nummern.

Schließlich werden es Dienste auf einem Gerät.

Server-Welt: Mehrere Services auf einem Server.

Notwendigkeit, die einzelnen Services anzusprechen,
obwohl Server nur eine Server-Adresse hat.

Neues Konzept: Port-Nummer oder auch Port-Adresse.

Server an IP-Adresse 192.168.0.1 bietet Dienste unter Ports an:

- 192.168.0.1:80 Web-Service (http),
- 192.168.0.1:443 Web-Service mit secure socket layer (https)
- 192.168.0.1:25 Outgoing Email-Service, unverschlüsselt
- 192.168.0.1:111 Portmapper: Konvertiert Dienstnamen in Port Nummer
- Kurze Liste der wichtigsten Port Nummern, längere Liste

Fortsetzung folgt.

1. Anwendungen: Eine Email
2. Darstellung: In Bits
3. Datenformate & MIME
4. Datenpakete & Netzdienste
5. Adressen, Teilnehmer & Namen
- 6. Routing**
7. Sitzungen

6. Routing

7. Sitzungen

1. Anwendungen: Eine Email
2. Darstellung: In Bits
3. Datenformate & MIME
4. Datenpakete & Netzdienste
5. Adressen, Teilnehmer & Namen
6. Routing
7. Sitzungen

Appendix

Übersicht		 84
Literaturverzeichnis		 85
Verzeichnis aller Abbildungen	Abb	 86
Verzeichnis aller Tabellen	Tab	 90
Rechtsnachweise	©	 91
Rechtliche Hinweise	§	 94
Zitierweise dieses Dokuments	→	 96
Verzeichnis aller Folien		 97

- [Kle08] John Klensin. *Simple Mail Transfer Protocol*. RFC Editor. Oct. 2008. DOI: [10.17487/RFC5321](https://doi.org/10.17487/RFC5321). URL: <https://rfc-editor.org> (cit. on p. 9).
- [Res08] Peter W. Resnick. *Internet Message Format*. RFC Editor. Oct. 2008. DOI: [10.17487/RFC5322](https://doi.org/10.17487/RFC5322). URL: <https://rfc-editor.org> (cit. on p. 9).

Verzeichnis aller Abbildungen (1/4)

1	Komplexität des Internet	3
2	Bestandteile einer Email	7
3	Bestandteile einer Email	12
4	Carriage Return und Line Feed	14
5	Email im Texteditor	15
6	Octal dump einer Email	16
7	text/html als Text	21
8	text/html als HTML	21
9	Konfiguration von MIME-Type Verhalten	23

10	Falsche Reaktion im Download	24
11	chrome.mimeHandler Flag	25
12	Richtige Reaktion im Download	26
13	Mutlipart MIME Type Beispiel	28
14	Mutlipart MIME Type Beispiel	29
15	MIME Header	30
16	MIME Header	30
17	Magic Bytes	31
18	PDF Datei im Texteditor	33
19	Erkennen von Datentypen	34
20	Komplexität des Internet	35

21	Naives Schichtenmodell	36
22	Selbstbetrug	38
23	Packet Tracer Wireshark	40
24	Packet Trace einer http-Verbindung (1)	45
25	Packet Trace einer http-Verbindung (2)	46
26	Packet Trace einer http-Verbindung (3)	47
27	Packet Trace einer http-Verbindung (4)	48
28	Leitungsvermitteltes Netz	49
29	Leitungsvermittlung	50
30	Paketvermitteltes Netz	51
31	Simulierte Leitungsvermittlung	52

32	Eselsbrücke.....	57
33	man page für write	62
34	Analyse einer physikalischen Adresse.....	66

1	Dateinamen und Extensions	32
2	Kerneigenschaften typischer Transportdienste	59
3	Spezielle IPv4-Adreßbereiche	75
4	Spezielle IPv6-Adreßbereiche	77

Rechtsnachweise (1/3)

- Abb. 1 Eigenes Bild, Visualisierung durch ChatGPT, 
- Abb. 2 Eigenes Bild, Visualisierung durch ChatGPT, 
- Abb. 3 Eigenes Bild, Visualisierung durch ChatGPT, 
- Abb. 4 Eigenes Bild, Visualisierung durch ChatGPT, 
- Abb. 5 Eigener Screenshot
- Abb. 6 Eigener Screenshot
- Abb. 7 Eigene Abbildung
- Abb. 8 Eigene Abbildung
- Abb. 9 Eigener Screenshot von [Stackoverflow](#)
- Abb. 10 Eigenes Bild, Visualisierung durch ChatGPT, 
- Abb. 11 Eigener Screenshot von [chrome://flags/](#) unter Version 154.0.8037.58
- Abb. 12 Eigener Screenshot.
- Abb. 13 Eigener Screenshot
- Abb. 14 Eigener Screenshot

- Abb. 15 Eigene Abbildung
- Abb. 16 Eigene Abbildung
- Abb. 17 Screenshot von Eigenes Bild, Visualisierung durch ChatGPT, 
- Abb. 18 Eigener Screenshot
- Abb. 19 Screenshot von Eigenes Bild, Visualisierung durch ChatGPT, 
- Abb. 20 Eigenes Bild, Visualisierung durch ChatGPT, 
- Abb. 21 Eigene Darstellung in Tikz
- Abb. 22 Eigenes Bild, Visualisierung durch ChatGPT, 
- Abb. 23 Eigener Screenshot
- Abb. 24 Eigener Screenshot
- Abb. 25 Eigener Screenshot
- Abb. 26 Eigener Screenshot
- Abb. 27 Eigener Screenshot
- Abb. 28 Eigene Abbildung
- Abb. 29 Eigenes Bild, Visualisierung durch ChatGPT, 
- Abb. 30 Eigene Darstellung

Abb. 31 Eigene Darstellung

Abb. 32 Eigenes Bild, Visualisierung durch ChatGPT, 

Abb. 33 Eigener Screenshot vom der Linux man page <https://man.archlinux.org/man/write.2.en>.

Abb. 34 Eigener Screenshot.

Die hier angebotenen Inhalte unterliegen deutschem Urheberrecht. Inhalte Dritter werden unter Nennung der Rechtsgrundlage ihrer Nutzung und der geltenden Lizenzbestimmungen hier angeführt. Auf das Literaturverzeichnis wird verwiesen. Das **Zitatrecht** in dem für wissenschaftliche Werke üblichen Ausmaß wird beansprucht. Wenn Sie eine Urheberrechtsverletzung erkennen, so bitten wir um Hinweis an den auf der Titelseite genannten Autor und werden entsprechende Inhalte sofort entfernen oder fehlende Rechtsnennungen nachholen. Bei Produkt- und Firmennamen können Markenrechte Dritter bestehen. Verweise und Verlinkungen wurden zum Zeitpunkt des Setzens der Verweise überprüft; sie dienen der Information des Lesers. Der Autor macht sich die Inhalte, auch in der Form, wie sie zum Zeitpunkt des Setzens des Verweises vorlagen, nicht zu eigen und kann diese nicht laufend auf Veränderungen überprüfen.

Alle sonstigen, hier nicht angeführten Inhalte unterliegen dem Copyright des Autors, Prof. Dr. Clemens Cap, ©2020. Wenn Sie diese Inhalte nützlich finden, können Sie darauf verlinken oder sie zitieren. Jede weitere Verbreitung, Speicherung, Vervielfältigung oder sonstige Verwertung außerhalb der Grenzen des Urheberrechts bedarf der schriftlichen Zustimmung des Rechteinhabers. Dieses dient der Sicherung der Aktualität der Inhalte und soll dem Autor auch die Einhaltung urheberrechtlicher Einschränkungen wie beispielsweise **Par 60a UrhG** ermöglichen.

Die Bereitstellung der Inhalte erfolgt hier zur persönlichen Information des Lesers. Eine Haftung für mittelbare oder unmittelbare Schäden wird im maximal rechtlich zulässigen Ausmaß ausgeschlossen, mit Ausnahme von Vorsatz und grober Fahrlässigkeit. Eine Garantie für den Fortbestand dieses Informationsangebots wird nicht gegeben.

Die Anfertigung einer persönlichen Sicherungskopie für die private, nicht gewerbliche und nicht öffentliche Nutzung ist zulässig, sofern sie nicht von einer offensichtlich rechtswidrig hergestellten oder zugänglich gemachten Vorlage stammt.

Use of Logos and Trademark Symbols: The logos and trademark symbols used here are the property of their respective owners. The YouTube logo is used according to brand request 2-9753000030769 granted on November 30, 2020. The GitHub logo is property of GitHub Inc. and is used in accordance to the GitHub logo usage conditions <https://github.com/logos> to link to a GitHub account. The Tweedback logo is property of Tweedback GmbH and here is used in accordance to a cooperation contract.

Disclaimer: Die sich immer wieder ändernde Rechtslage für digitale Urheberrechte erzeugt ein nicht unerhebliches Risiko bei der Einbindung von Materialien, deren Status nicht oder nur mit unverhältnismäßig hohem Aufwand abzuklären ist. Ebenso kann den Rechteinhabern nicht auf sinnvolle oder einfache Weise ein Honorar zukommen, obwohl deren Leistungen genutzt werden.

Daher binde ich gelegentlich Inhalte nur als Link und nicht durch Framing ein. Lt EuGH Urteil 13.02.2014, C-466/12 ([Pressemitteilung](#), [Blog-Beitrag](#), [Urteilstext](#)). ist das unbedenklich, da die benutzten Links ohne Umgehung technischer Sperren auf im Internet frei verfügbare Inhalte verweisen. Wenn Sie diese Rechtslage stört, dann setzen Sie sich für eine Modernisierung des völlig veralteten Vergütungs- und Anreizsystems für urheberrechtliche Leistungen ein. Bis dahin klicken Sie bitte auf die angegebenen Links und denken Sie darüber nach, warum wir keine für das digitale Zeitalter sinnvoll angepaßte Vergütungs- und Anreizsysteme digital erbrachter Leistungen haben. Zu Risiken und Nebenwirkungen fragen Sie Ihren Rechtsanwalt oder Gesetzgeber. Weitere Hinweise finden Sie im Netz [hier](#) und [hier](#) oder [hier](#).

Zitierweise dieses Dokuments

Wenn Sie Inhalte aus diesem Werk nutzen oder darauf verweisen wollen, zitieren Sie es bitte wie folgt:

Clemens H. Cap: Propädeutik der Kommunikation. Electronic document, 11. 10. 2026

<https://iuk.one/RNDS-PROP.pdf>

Bibtex Information: <https://iuk.one/RNDS-PROP.bib>

```
@misc{doc:RNDS-PROP,  
  author      = {Clemens H. Cap},  
  title       = {Propädeutik der Kommunikation},  
  year        = {2026},  
  month       = {10},  
  howpublished = {Electronic document},  
  url         = {https://iuk.one/RNDS-PROP.pdf }  
}
```

Typographic Information:

Typeset on 2026-10-11 at 20:43
This is pdfTeX, Version 3.141592653-2.6-1.40.29 (TeX Live 2026) kpahsea version 6.4.2
This is pgf in version 3.1.11a
This is preamble-slides.tex ©C.H.Cap
Jobtype flag=foreground
Active modes (if any): slides, biblio
Keywords: keywords
Subject: subject

Verzeichnis aller Folien

Titelseite	1
Eine scheinbar einfache Aufgabe...	2
○	3
Übersicht	4
Die erste naive Frage	5
1. Anwendungen: Eine Email.....	6
Aus welchen Teilen besteht eine Email? (1)	7
Aus welchen Teilen besteht eine Email? (2)	8
Wo finden wir die Antwort?	9
2. Darstellung: In Bits.....	10
Darstellung von Texten	11
Bestandteile einer Email	12
Trennung von Header und Body	13
Warum heißen CR und LF so?	14
Wir testen das jetzt selber! (1)	15
Wir testen das jetzt selber! (2)	16
Aufgabe: E-Mail Header	17
Weitere Themen	18

3. Datenformate & MIME.....19

MIME: Multipurpose Internet Mail Extensions	20
Folgen falscher MIME-Types	21
Folgen falscher MIME-Type Regeln (1)	22
Folgen falscher MIME-Type Regeln (2)	23
Folgen falscher MIME-Type Regeln (3)	24
Folgen falscher MIME-Type Regeln (4)	25
Folgen falscher MIME-Type Regeln (5)	26
Multipart MIME Types (1)	27
Multipart MIME Types (2)	28
Diskussion	29
Wie erkennen Systeme Datentypen? (1) MIME Header	30
Wie erkennen Systeme Datentypen? (2) Magic Bytes	31
Wie erkennen Systeme Datentypen? (3) Dateinamen & Extension	32
Wie erkennen Systeme Datentypen? (4) Magic Words	33
Wie erkennen Systeme Datentypen? (5) Viele weitere	34
Gesamtbild bisher	35
Unser erstes Schichtenmodell	36

4. Datenpakete & Netzdienste.....37

4.1. Datenpakete	
Aufteilung in Pakete	38
6 Gründe für Datenpakete	39
Wie kann ich mir diese Pakete ansehen?	40
○	41
○	42
Packet Tracer Wireshark	43
Aufgabe	44
Packet Trace einer http-Verbindung (1)	45
Packet Trace einer http-Verbindung (2)	46
Packet Trace einer http-Verbindung (3)	47
Packet Trace einer http-Verbindung (4)	48
4.2. Versenden der Datenpakete	
Leitungsvermittlung	49
Leitungsvermittlung im Internet?	50
Paketvermittlung	51
Simulierte Leitungsvermittlung	52
Tweedback	53
4.3. Netzdienste	
Netzdienste	54
5 Eigenschaften von Netzdiensten (1)	55
5 Eigenschaften von Netzdiensten (2)	56
Eselsbrücke für die 5 Eigenschaften	57
Typische Transport-Dienste	58
Kerneigenschaften typischer Transportdienste	59
Welche Bedeutung hat das?	60
Hinweis	61
man page für write	62

5. Adressen, Teilnehmer & Namen 63

5.1. Konzepte	
PHY Adressen	64
MAC und IP Adressen	65
Beispiel für Adressen auf einem System	66
Aufgabe	67
Zur Vermeidung von Mißverständnissen	68
Adressen auf weiteren Schichten	69
Gruppen-Adressen	70
Adreß-freie Kommunikation	71
5.2. MAC, IPv4 und IPv6 Adressen	
MAC-Adresse: Aufbau und Darstellung	72
IPv4 und IPv6: Was ist das?	73
IPv4-Adresse: Aufbau und Darstellung	74
IPv4-Adresse: Spezielle Adreßbereiche	75
IPv6-Adresse: Aufbau und Darstellung	76
IPv6-Adresse: Spezielle Adreßbereiche	77
Konzept des Teilnehmers	78
Beispiele für Teilnehmer	79

6. Routing 80

7. Sitzungen 81

Legende:

-  Fortsetzungsseite
-  Seite ohne Überschrift
-  Bildseite