

# Dienste und Protokolle im Internet

## *E-Mail-Protokolle*

---

iMedia 2018

**9.5.2018**

Sebastian Kapp  
AG Didaktik der Physik  
TU Kaiserslautern

# Inhalt

---

- Überblick über das Thema
- Einordnung in die Netzwerkschichten
- Einordnung in den Lehrplan
- Auswahl der verwendeten Protokolle
- Request for Comments (RFCs)
- Umsetzung der Aufgaben und Übungen
- Inhalt des [inf-schule.de](http://inf-schule.de) Kapitels

# Dienste und Protokolle des Internets

## *E-Mail-Protokolle*

---

- Kapitel im inf-schule.de Schulbuch
- **Inhalt:** Ein Protokoll auf Socket-Ebene untersuchen und anwenden
- **Ziel:** Umsetzung der im Lehrplan vorgegebenen Inhalte
- **Verwendete Protokolle:** SMTP/POP3
- **Zusätzlicher Inhalt:**
  - Einführung von RFCs als Standarddefinition
  - Umsetzung mit Filius als Simulationsumgebung

# Einordnung in die Netzwerkschichten

---

<b>Anwendung</b>	<u>Protokolle von Internetanwendungen</u>	<u>E-Mail-Protokolle</u>	<u>Netzwerkspiel - Zahlenraten</u>	
<b>Transport</b>	<u>Datentransport zwischen Anwendungsprozessen</u>		<u>Grundlagen der Socketprogrammierung</u>	<u>Simulation mit Filius</u>
<b>Vermittlung</b>	<u>Vermittlung von Datenpaketen</u>			
<b>Sicherung</b>	<u>Bitübertragung über die serielle Schnittstelle</u>		<u>Bitübertragung mit dem Raspi</u>	
<b>Bitübertragung</b>				

Quelle: <http://inf-schule.de/kommunikation/netze/module>

# Einordnung in den Lehrplan

## Kapitel: Kommunikation in Rechnernetzen

**Abschnitt:** Kommunikation in Rechnernetzen erläutern und am Beispiel des Internet verdeutlichen

### Grundkurs:

Dienste und Protokolle des Internet	<ul style="list-style-type: none"><li>➤ Kommunikationsvorgänge mit der Client-Server-Struktur beschreiben.</li><li>➤ Einen Überblick über aktuelle Dienste und Protokolle des Internet verschaffen</li><li>➤ Exemplarisch ein Protokoll (z.B. HTTP, POP, SMTP, FTP) mit Hilfe eines Terminalprogramms und mit Hilfe von RFC-Auszügen untersuchen.</li><li>➤ Optional: Mit Hilfe von vorhandenen Klassen, die eine TCP/IP-Verbindung herstellen (Sockets), eine Anwendung programmieren.</li></ul>
-------------------------------------	---

### Leistungskurs:

Dienste und Protokolle des Internet	<ul style="list-style-type: none"><li>➤ Kommunikationsvorgänge mit der Client-Server-Struktur beschreiben.</li><li>➤ Einen Überblick über aktuelle Dienste und Protokolle des Internet verschaffen.</li><li>➤ Protokolle (z.B. HTTP, POP, SMTP, FTP) mit Hilfe eines Terminalprogramms und mit Hilfe von RFC-Auszügen untersuchen.</li><li>➤ Mit Hilfe von vorhandenen Klassen, die eine TCP/IP-Verbindung herstellen (Sockets), eine Anwendung programmieren.</li></ul>
-------------------------------------	--

*Quelle: Lehrplan Informatik, Grund- und Leistungsfach Einführungsphase Und Qualifikationsphase der gymnasialen Oberstufe (Mainzer Studienstufe)*  
[https://static.bildung-rp.de/lehrplaene/gymnasium/Informatik\\_LP\\_SekII.pdf](https://static.bildung-rp.de/lehrplaene/gymnasium/Informatik_LP_SekII.pdf)

# Auswahl der verwendeten Protokolle

---

**Voraussetzung:** Protokoll wird aktiv im Internet verwendet und ist den Schülern „bekannt“.

**Unter anderem in Frage kommende Protokolle:**

- HTTP
- DNS
- SMTP
- POP3
- IMAP
- ...

# HTTP/1.1

---

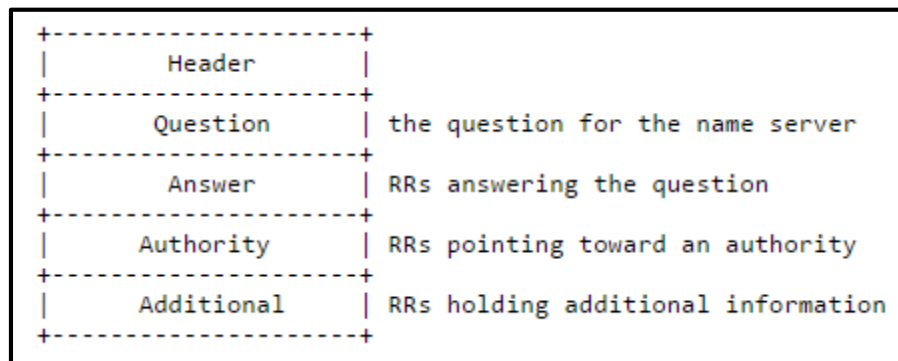
- + Sehr einfacher Grundbefehl
- + Lässt sich gegen echte Server testen
- Keine „echte“ Interaktion mit Server (nur Antwort auf eine einzelne Anfrage)
- Ergebnisse von realen Servern zu groß und nicht einfach lesbar
- **Praktische Einschränkung:** Timeouts bei realen Servern!

```
GET / HTTP/1.1  
HOST: checkip.dyndns.org
```

# DNS

---

- + Sehr einfacher Gedanke
- + Sehr einfache Inhalte
- + Gut in einen Kontext zu bringen
- + Gegen echte Server testbar
- **Binäres Protokoll! Nicht Textbasiert über einen Socket ansprechbar**





# SMTP

---

- + Einfaches Protokoll
- + Interaktion mit Server notwendig (nicht nur Formulierung einer Anfrage)
- + Enthält viele Aspekte von Datenübertragung (z.B. Ende einer Nachricht)
- ± Theoretisch im Internet einsetzbar
- **Aber:** Im Internet Authentifizierung notwendig die nicht unverschlüsselt stattfinden sollte!
- Offene Server zum Testen sind wegen Spamgefahr nicht möglich

# SMTP

---

```
Verbindung hergestellt
>>220 Willkommen bei example.com
<<HELO alice.example.com
>>250 Hello alice.example.com
<<MAIL FROM:<alice@example.com>
>>250 Sender OK
<<RCPT TO:<bob@example.com>
>>250 Recipient OK
<<DATA
>>354 End data with <CR><LF>.<CR><LF>
<<Subject: Testnachricht

Dies ist ein Test!

.
>>250 Mail queued for delivery
<<QUIT
>>221 Server beendet Verbindung.
```

# IMAP

---

- + Aktuelles Protokoll
- + Grundsätzlich ebenfalls einfache Befehle
- Höhere Komplexität in der Bedienung da Ordnerstruktur zu navigieren
- Langes RFC (77 Seiten)

# IMAP

```
S: * OK IMAP4 Service Ready
C: a001 login mrc secret
S: a001 OK LOGIN completed
C: a002 select inbox
S: * 18 EXISTS
S: * FLAGS (\Answered \Flagged \Deleted \Seen \Draft)
S: * 2 RECENT
S: * OK [UNSEEN 17] Message 17 is the first unseen
    message
S: * OK [UIDVALIDITY 3857529045] UIDs valid
S: a002 OK [READ-WRITE] SELECT completed
C: a003 fetch 12 full
S: * 12 FETCH (FLAGS (\Seen) INTERNALDATE "14-Jul-1993
    02:44:25 -0700"
    RFC822.SIZE 4282 ENVELOPE ("Wed, 14 Jul 1993
    02:23:25 -0700 (PDT)"
    "IMAP4 WG mtg summary and minutes"
    (("Terry Gray" NIL "gray"
    "cac.washington.edu"))
    (("Terry Gray" NIL "gray"
    "cac.washington.edu"))
    (("Terry Gray" NIL "gray"
    "cac.washington.edu"))
    ((NIL NIL "imap" "cac.washington.edu"))
    ((NIL NIL "minutes" "CNRI.Reston.VA.US")
    ("John Klensin" NIL "KLENSIN"
    "INFOODS.MIT.EDU")) NIL NIL
    "<B27397-0100000@cac.washington.edu>")
    BODY ("TEXT" "PLAIN" ("CHARSET" "US-ASCII")
    NIL NIL "7BIT" 3028 92))
S: a003 OK FETCH completed
C: a004 fetch 12 rfc822.header
S: * 12 FETCH (RFC822.HEADER {346}
S: Date: Wed, 14 Jul 1993 02:23:25 -0700 (PDT)
S: From: Terry Gray <gray@cac.washington.edu>
S: Subject: IMAP4 WG mtg summary and minutes
S: To: imap@cac.washington.edu
S: cc: minutes@CNRI.Reston.VA.US, John Klensin
    <KLENSIN@INFOODS.MIT.EDU>
S: Message-Id: <B27397-0100000@cac.washington.edu>
S: MIME-Version: 1.0
S: Content-Type: TEXT/PLAIN; CHARSET=US-ASCII
S: )
S: a004 OK FETCH completed
C: a005 store 12 +flags \deleted
S: * 12 FETCH (FLAGS (\Seen \Deleted))
S: a005 OK +FLAGS completed
C: a006 logout
S: * BYE IMAP4 server terminating connection
S: a006 OK LOGOUT completed
```

# POP3

---

- + Einfaches Protokoll
- + Interaktion mit Server
- + Sehr kurzes (erstes) RFC (16 Seiten)
  
- Ebenfalls Verschlüsselungsproblem da Authentifizierung notwendig
- „Altes“ Protokoll, IMAP als moderne Alternative

# POP3

---

```
Verbindung hergestellt
>>+OK POP3 server ready
<<USER Bob
>>+OK enter password
<<PASS bob
>>+OK Mailbox locked and ready
<<LIST
>>+OK 1 38
0 38

<<RETR 0
>>+OK message follows
Subject: Testnachricht

Dies ist ein Test!
<<DELE 0
>>+OK message marked for delete
<<QUIT
>>+OK
```

# RFC (Request for Comments) [1/3]

---

- Entstanden 1969 im ARPANET um die evolutionär entstandene Netzwerkstruktur zu dokumentieren.
- Zunächst mit der Schreibmaschine geschriebene Briefe welche per Post verschickt wurden.
- Keine Standarddefinitionen sondern reine Ideensammlungen für Protokolle o.ä.
- „Echte“ bitte um Kommentare.
- Jeder konnte ein RFC versenden und seine Ideen teilen.
- Zugriff und Verwendung der Ideen war frei.
- Auch einfache Ideen wurden als RFC geteilt.

# RFC (Request for Comments) [2/3]

---

- Entscheidungen wurden nach “rough consensus and running code” gefällt.  
Funktionierte ein RFC und wurde von vielen genutzt galt es automatisch als Standard.
  - Nach und nach wurden die vorgeschlagenen Inhalte ausgefeilter und von größeren Gruppen entworfen.
  - Heute Veröffentlichung von RFCs eingeschränkt.
  - Bevor ein RFC veröffentlicht wird finden bereits umfassende Überarbeitungen statt.
- Damit kein “echter” RFC mehr sondern ausgefeilter Vorschlag für einen neuen Standard.



# RFC (Request for Comments) [3/3]

---

- Struktur:
  - Jeder RFC besitzt eine eindeutige Nummer.
  - Formalisierte Schreibweise mit festen Definitionen und festem Aufbau.
- Die meisten Standards im Internet sind in Form eines RFC definiert.

# Wichtige und Bekannte RFCs

---

Nummer	Protokoll
768	UDP
791	IPv4
793	TCP
821	SMTP
1034	DNS
1738	URL
1939	POP3
1945	HTTP 1.0
2460	IPv6
...	

# Umsetzung der Aufgaben und Übungen

---

## **Verwendung von Filius zur Simulation eines abgeschlossenen E-Mail Netzes.**

- + Wohl definierte Umgebung
- + Auch offline nutzbar
- + Auf allen Plattformen verfügbar
  
- ± Primär zum Aufbau von Netzwerktopologien gedacht
  
- Dadurch hohe unnötige Komplexität (durch vordefinierte Umgebung reduzierbar)
- Kein „echtes“ Internet sondern intern simulierte Server und E-Mails

# Inhalt des inf-schule.de Kapitels

---