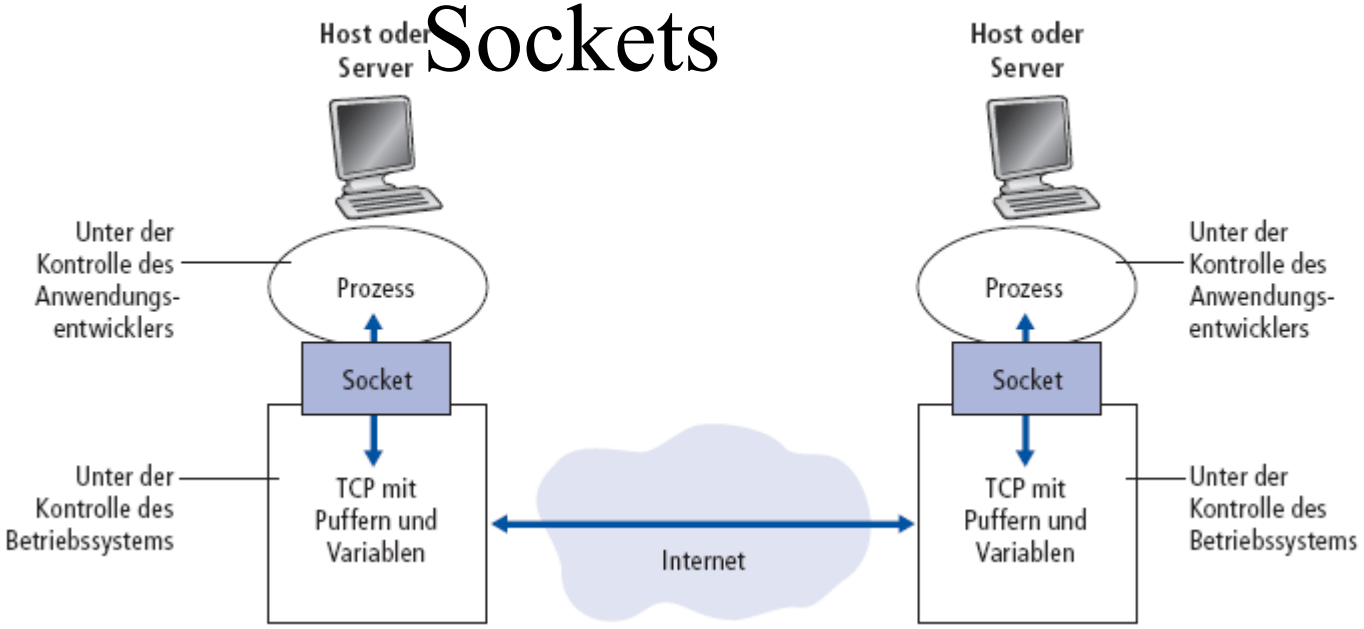
Protokollstapel TCP/IP

- Anwendungsschicht: Unterstützung von Netzwerkanwendungen
  - FTP, SMTP, HTTP
- Transportschicht: Datentransfer zwischen Prozessen
  - TCP, UDP
- Netzwerkschicht (auch Vermittlungsschicht): Weiterleiten der Daten von einem Sender zu einem Empfänger
  - IP, Routing-Protokolle
- Sicherungsschicht: Datentransfer zwischen benachbarten Netzwerksystemen
  - PPP, Ethernet
- Bitübertragungsschicht: Bits auf der Leitung

We reject kings, presidents and voting. We believe in rough consensus and running code (Dave Clark)

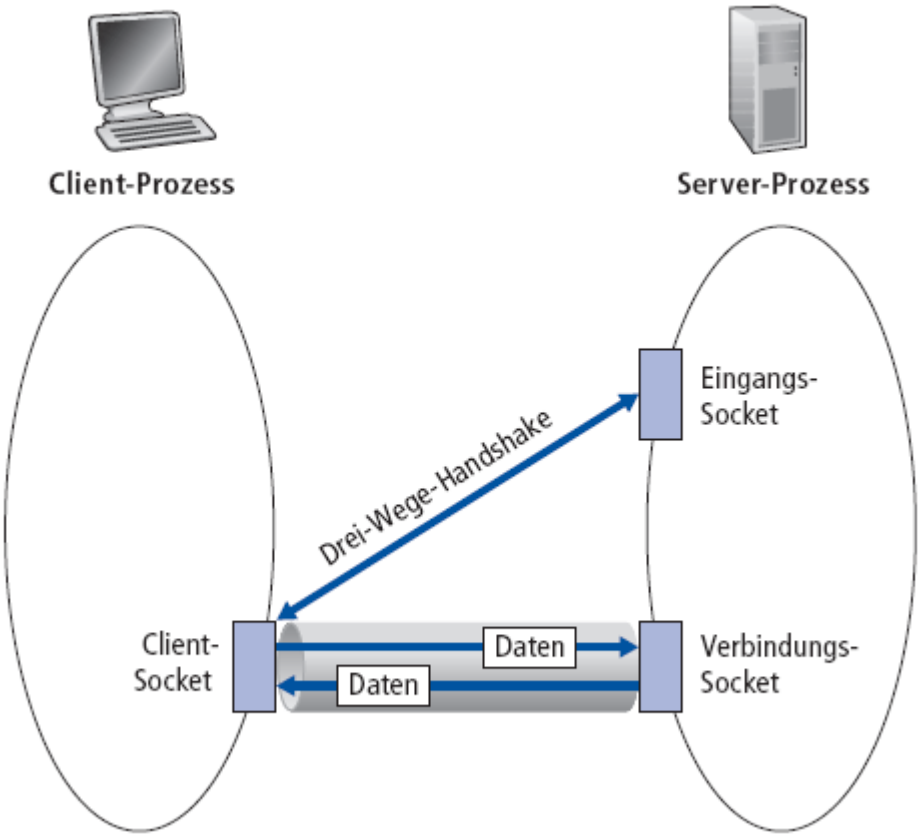
T-Shirt IETF

Sockets

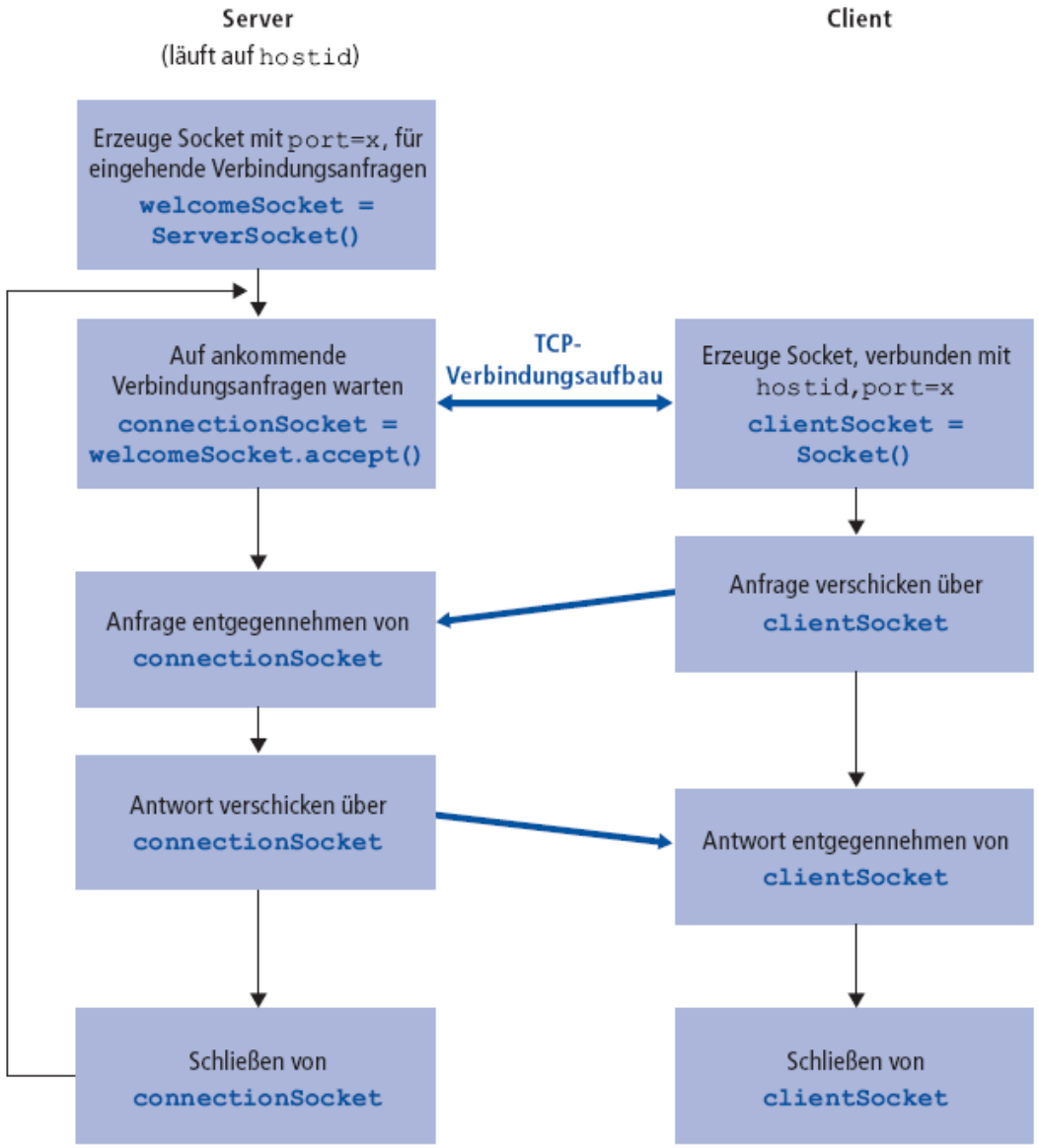


- Prozesse senden/empfangen Nachrichten über einen Socket
- Ein Socket lässt sich mit einer Tür vergleichen
  - Der sendende Prozess schiebt die Nachrichten durch die Tür
  - Der sendende Prozess verlässt sich auf die Transportinfrastruktur auf der anderen Seite der Tür, um die Nachricht zum Socket des empfangenden Prozesses zu bringen

Zusammenspiel der Sockets



Client/Server-Socket-Programmierung: TCP



Web und HTTP

Fakultät für Feinwerk- und Mikrotechnik, Physikalische Technik
HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN · FH MÜNCHEN

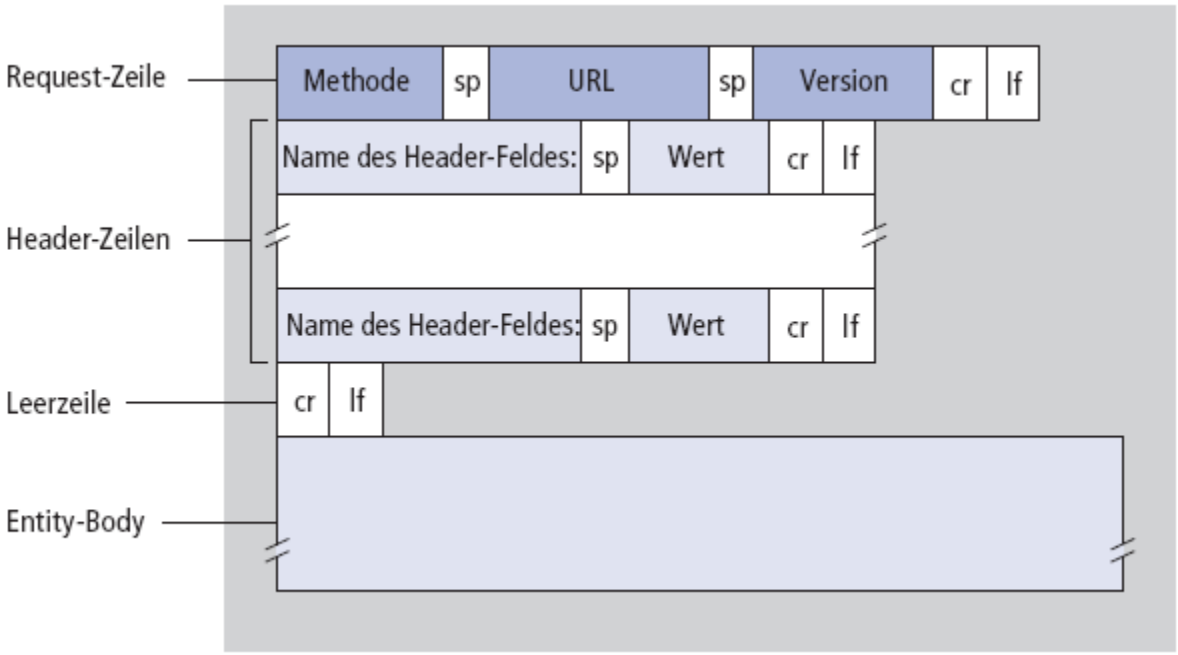
Einige Definitionen

- Eine Webseite besteht aus Objekten
- Objekte können sein: HTML-Dateien, JPEG-Bilder, Java-Applets, Audiodateien, …
- Eine Webseite hat eine Basis-HTML-Datei, die mehrere referenzierte Objekte beinhalten kann
- Jedes Objekt kann durch eine URL (Uniform Resource Locator) adressiert werden
- Beispiel für eine URL:

`www.hm.edu/someDept/pic.gif`

Hostname | Pfad

HTTP-Request-Nachricht: Format



Hochladen von Informationen

Post-Anweisung:

- Webseiten beinhalten häufig Formulare, in denen Eingaben erfolgen sollen
- Eingaben werden zum Server im Datenteil (entity body) der Post-Anweisung übertragen

Get-Anweisung:

- Eingabe wird als Bestandteil der URL übertragen:

```
www.somesite.com/animalsearch?monkeys&banana
```

GET und POST Methode(1)

Fakultät für Feinwerk- und Mikrotechnik, Physikalische Technik
HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN · FH MÜNCHEN

History:	Parameters remain in browser history because they are part of the URL	Parameters are not saved in browser history.
Bookmarked:	Can be bookmarked.	Can not be bookmarked.
BACK button/re-submit behaviour:	GET requests are re-executed.	The browser usually alerts the user that data will need to be re-submitted.
Encoding type (enctype attribute):	application/x-www-form-urlencoded	multipart/form-data or application/x-www-form-urlencoded Use multipart encoding for binary data.
Parameters:	can send but the parameter data is limited to what we can stuff into the request line (URL). Safest to use less than 2K of parameters, some servers handle up to 64K	Can send parameters, including uploading files, to the server.
Hacked:	Easier to hack for script kiddies	More difficult to hack
Restrictions on form data type:	Yes, only ASCII characters allowed.	No restrictions. Binary data is also allowed.
Security:	GET is less secure compared to POST because data sent is part of the URL. So it's saved in browser history and server logs in plaintext.	POST is a little safer than GET because the parameters are not stored in browser history or in web server logs.

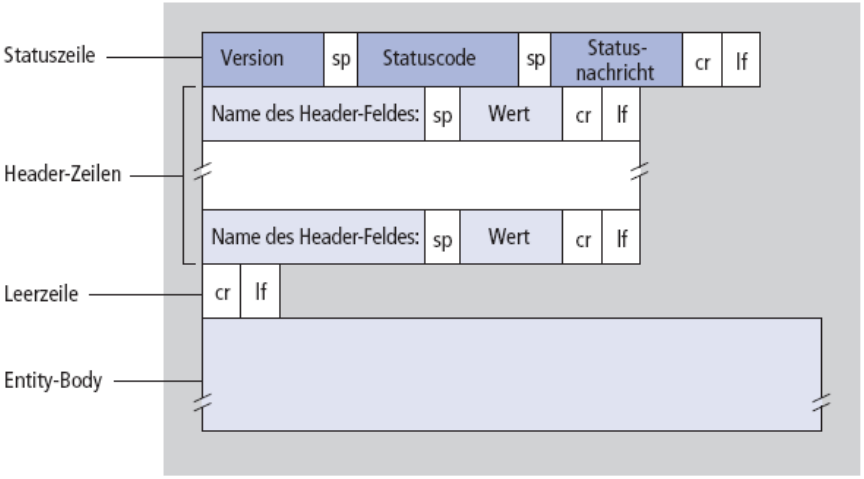
GET und POST Methode(2)

Restrictions on form data length:	Yes, since form data is in the URL and URL length is restricted	No restrictions
Usability:	GET method should not be used when sending passwords or other sensitive information.	POST method used when sending passwords or other sensitive information.
Visibility:	GET method is visible to everyone (it will be displayed in the browser's address bar) and has limits on the amount of information to send.	POST method variables are not displayed in the URL.
Cached:	Can be cached	Not cached
Large variable values:	7607 character maximum size.	8 Mb max size for the POST method.

http://www.diffen.com/difference/Get_vs_Post

HTTP-Response-Nachricht: Format

Fakultät für Feinwerk- und Mikrotechnik, Physikalische Technik
HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN · FH MÜNCHEN



Statuszeile (Statuscode, Statusnachricht)

```
HTTP/1.1 200 OK
Connection close
Date: Thu, 06 Aug 1998
12:00:15 GMT
Server: Apache/1.3.0 (Unix)
Last-Modified: Mon, 22 Jun
1998 …...
Content-Length: 6821
Content-Type: text/html

data data data data data ...
```

Header-Zeilen

Entity Body: Daten, z.B. die angefragte HTML-Datei

Ausprobieren von HTTP (Client)

1. Telnet zu einem Webserver:

```
telnet 141.39.241.216 80
```

Öffnet eine TCP-Verbindung zu Port 80 (Standard-HTTP-Server-Port) auf den Server
Alles, was jetzt eingegeben wird, wird an Port 80 dieses Servers geschickt

2. Eingeben eines HTTP-GET-Requests:

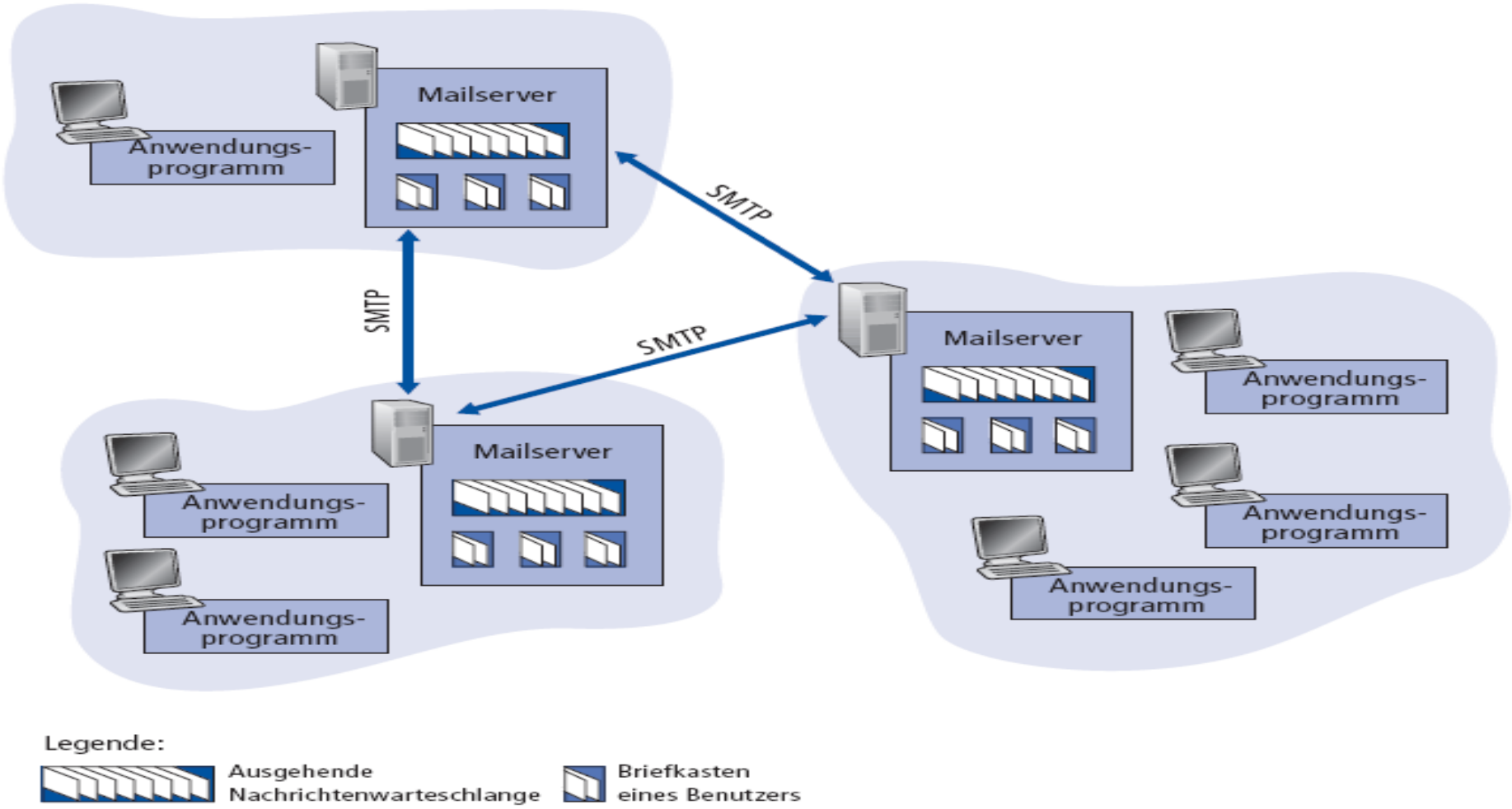
```
GET /hallo.html HTTP/1.1
Host: PC
```

Durch diese Eingabe (am Ende zweimal Return tippen!) wird ein minimaler (aber vollständiger) GET-Request an den HTTP-Server geschickt

3. Betrachten der Antwort!

Electronic Mail

Fakultät für Feinwerk- und Mikrotechnik, Physikalische Technik
HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN · FH MÜNCHEN



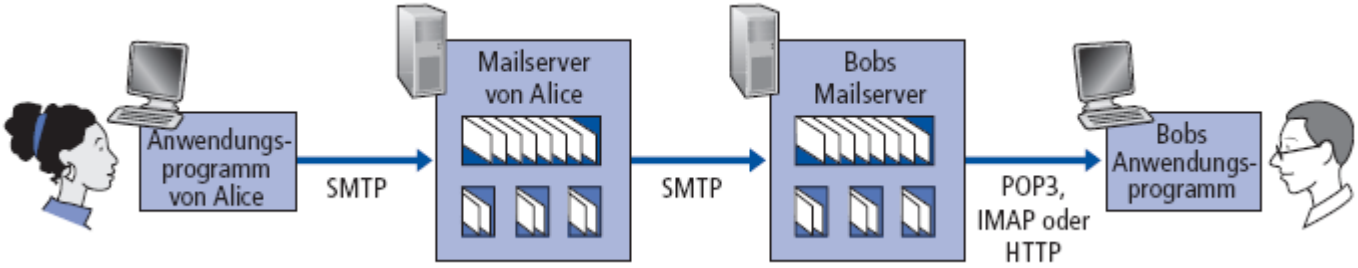
Electronic Mail: SMTP [RFC 2821]

- Ursprüngliche Version aus dem Jahr 1982
- TCP wird zum zuverlässigen Transport von E-Mail-Nachrichten vom Client zum Server (Port 25) verwendet
- Direkter Transport der Nachrichten: vom sendenden Server zum empfangenden Server
- Drei Phasen des Mail-Versands: analog zu einer Unterhaltung
  - Handshaking (Begrüßung)
  - Transfer of Messages (Austausch von Informationen)
  - Closure (Verabschiedung)
- Interaktion basiert auf dem Austausch von Befehlen (Commands) und Antworten (Responses)
  - Command: ASCII-Text
  - Response: Statuscode und Bezeichnung
- Nachrichten müssen in 7-Bit-ASCII kodiert sein

Beispiel für eine SMTP-Sitzung


```
S: 220 hamburger.edu
C: HELO crepes.fr
S: 250  Hello crepes.fr, pleased to meet you
C: MAIL FROM: <alice@crepes.fr>
S: 250 alice@crepes.fr... Sender ok
C: RCPT TO: <bob@hamburger.edu>
S: 250 bob@hamburger.edu ... Recipient ok
C: DATA
S: 354 Enter mail, end with "." on a line by itself
C: Do you like ketchup?
C: How about pickles?
C: .
S: 250 Message accepted for delivery
C: QUIT
S: 221 hamburger.edu closing connection
```

Mail-Zugriffsprotokolle



- SMTP: Zustellung/Speicherung auf dem Mailserver des Empfängers
- Zugriffsprotokoll: Protokolle zum Zugriff auf E-Mails
- Abruf vom Server
  - POP: Post Office Protocol [RFC 1939]
    - Autorisierung (Anwendung <--> Server) und Zugriff/Download
  - IMAP: Internet Mail Access Protocol [RFC 1730]
    - Größere Funktionalität (deutlich komplexer)
    - Manipulation der auf dem Server gespeicherten Nachrichten
  - HTTP: Hotmail, Yahoo!Mail etc.

POP3-Protokoll

Autorisierungsphase:

- Befehle des Clients:
  - **user:** Benutzername
  - **pass:** Passwort
- Antworten des Servers:
  - **+OK**
  - **-ERR**

Transaktionsphase:

- **list:** Nachrichten auflisten
- **retr:** Nachrichten herunterladen
- **dele:** Löschen von Nachrichten
- **Quit: Ende**

```
S: +OK POP3 server ready
C: user bob
S: +OK
C: pass hungry
S: +OK user successfully logged on
```

```
C: list
S: 1 498
S: 2 912
S: .
C: retr 1
S: <message 1 contents>
S: .
C: dele 1
C: retr 2
S: <message 1 contents>
S: .
C: dele 2
C: quit
S: +OK POP3 server signing off
```