

# Grundlagen

-

DNS, UDP, TCP, HTTP

# Domain Name System

- IP-Adressen schlecht zu merken – schon gar bei IPv6
  - Einfacher: „sprechende“ Namen (symbolische Adressen)
  - Gesucht: Dienst, der die „Auflösung“ der gegebenen symbolischen Adressen in IP-Adressen und umgekehrt (Rückwärtsauflösung) leistet

www.vs.eecs.uni-kassel.de  141.51.110.124

- Eigenschaften
  - Namen (symbolische Adressen) bestehen aus mit Punkten getrennten textuellen Teilen (a.b.c.d...x)
  - Erster Bestandteil bezeichnet den **Rechner**, Rest bezeichnet die **Domäne**
  - Hierarchisches Domänen-Konzept: vs.eecs.uni-kassel.de.
    - Domänen können Rechner und/oder Domänen beinhalten
  - Verteiltes System zur Abbildung der Namen auf IP-Adressen

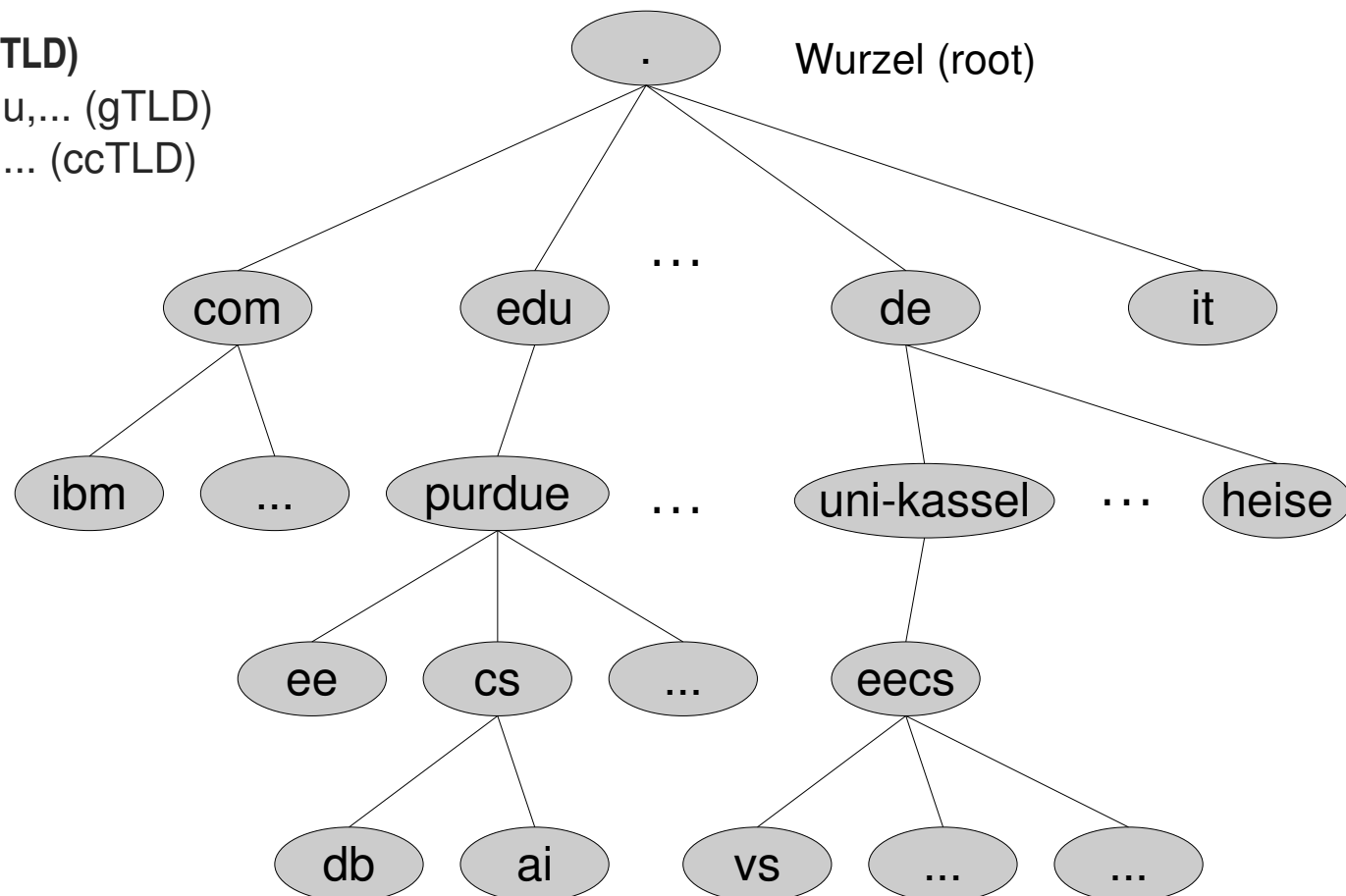
# Namensraum

## Top-Level-Domänen (TLD)

generisch: com, edu,... (gTLD)

geografisch: de, it, ... (ccTLD)

speziell: arpa



# Neue gTLDs

- Einführung neuer gTLDs zu den traditionellen
  - com, net, org sehr auf amerikanische Lokationen konzentriert
  - neue gTLDs
    - **biz**: ohne Sponsor, eingeschränkt: nur für kommerzielle Stellen
    - **info**: ohne Sponsor, uneingeschränkt
    - **name**: ohne Sponsor, eingeschränkt: nur für Individuen
    - **museum**: Sponsor „Museum Domain Management Association“, für Museen
    - **coop**: Sponsor „DotCooperation LLC“ (USA), vorgesehen für internationale Verbände
    - **aero**: Sponsor „Societe International de Telecommunications Aeronautiques“ (Belgien), für Flugverkehr
    - **pro**: ohne Sponsor, eingeschränkt: nur für Berufsgruppen, insbesondere Juristen, Steuerberater, Anwälte
    - **mobi** (Mobiltelefone) gerade in „Sunrise“-Phase (Erste Verteilung von Domännennamen)
    - **xxx** in einigen Ländern noch umstritten

# Namenserver

- Namenserver als verteiltes System
- Replikation der Namenserver zur Steigerung der Zuverlässigkeit
- Klienten des Namensdienstes stellen Anfrage per UDP an Port 53
- Jeder Namen-Server kennt mindestens einen Vorgänger
  - Die „Primary Name Server“ kennen die „Root Name Server“
- Zwischenspeichern der Namensinformation im Namensdienst
  - Einträge im Cache: Name / IP-Adresse / von welchem NS
  - Möglichkeit veralteter Information wird mitgeteilt
  - Bereinigung des Caches mit TTL (time-to-live) für Einträge

# Rückwärtsauflösung

- Diesmal IP-Adresse bekannt, aber nicht der Rechnername
  - Beispiel: Welcher Rechner heißt 193.99.144.85?
  - Wohin die Anfrage richten?
  - Wie kann man auch hier eine Hierarchie einführen?
- Problem: DNS-Adressen sind von rechts nach links aufgebaut (Rechner links), aber IP-Adressen von links nach rechts (Rechner rechts)
- Abbildung in den DNS-Namensraum **in-addr.arpa.:**
  - a.b.c.d → d.c.b.a.in-addr.arpa.
- Jede Institution, die eine Domäne verwaltet, bekommt auch die Hoheit über die entsprechende **in-addr.arpa.-**Domäne
  - Beispiel: uni-kassel.de = **141.51.x.x**
  - Damit verwaltet die Uni Kassel auch den Namensraum **x.x.51.141.in-addr.arpa**

# E-Mail-Versand

- E-Mail-Adressen: benutzer@maildomäne
- „maildomäne“ bezeichnet in der Regel **keinen** Rechner!

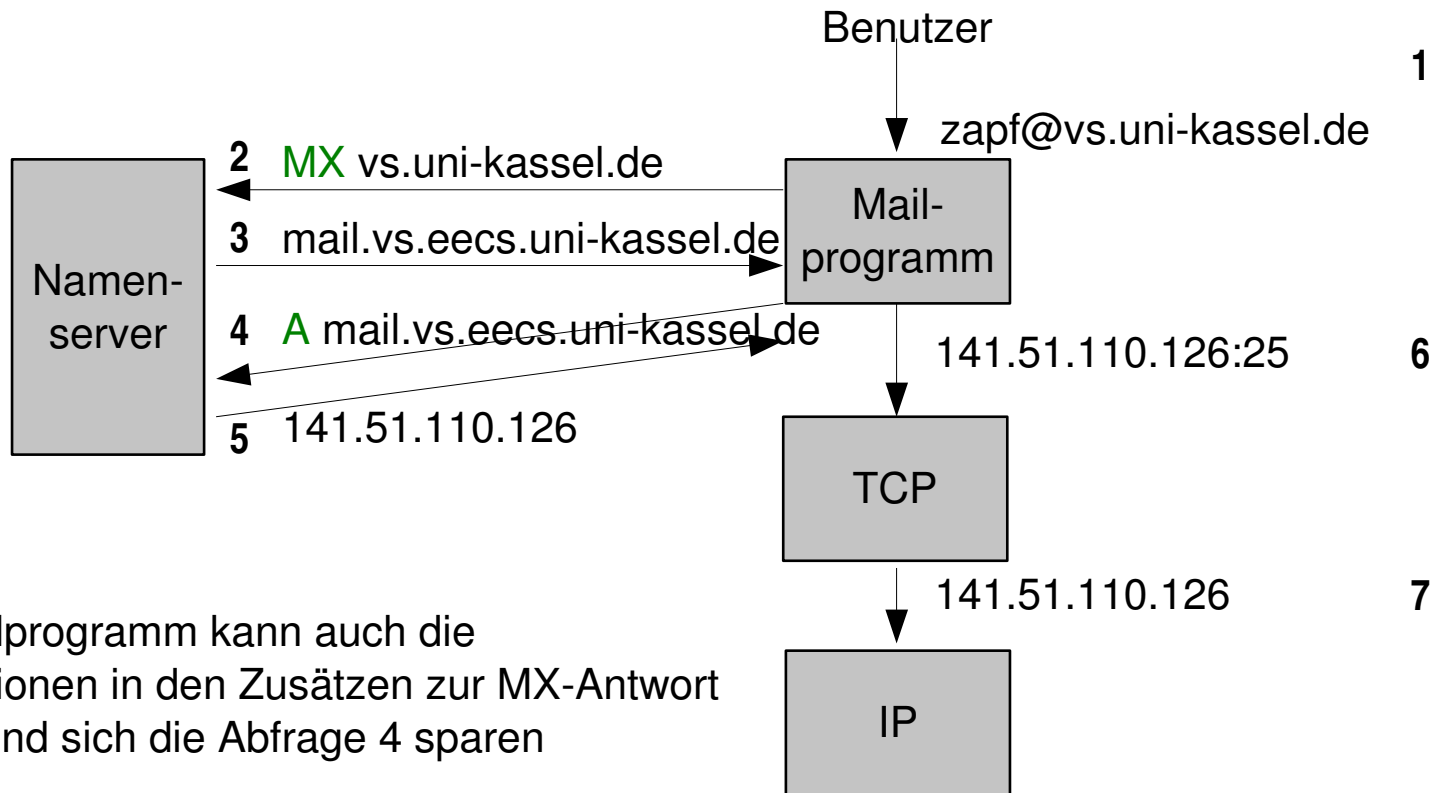
**\$ ping vs.uni-kassel.de**

**ping: unknown host vs.uni-kassel.de**

- Sendmail (verbreitetes E-Mail-Versendeprogramm) muss herausfinden, an welchen Rechner die Mail weitergereicht werden soll
  - Anfrage eines so genannten MX record (Mail exchange)

# E-Mail-Versand

- Beispiel: E-Mail an „zapf@vs.uni-kassel.de“



Das Mailprogramm kann auch die Informationen in den Zusätzen zur MX-Antwort nutzen und sich die Abfrage 4 sparen



# Unterstützung von IPv6

- Rückwärtsauflösung
  - Abfrage jetzt auf „ip6.arpa.“: IP-Adresse wieder umgekehrt, aber auf Halbbyte-Basis

4321:0:1:2:3:4:567:89ab

→ b.a.9.8.7.6.5.0.4.0.0.0.3.0.0.0.2.0.0.0.1.0.0.0.0.0.0.1.2.3.4.ip6.arpa.

# TCP und UDP

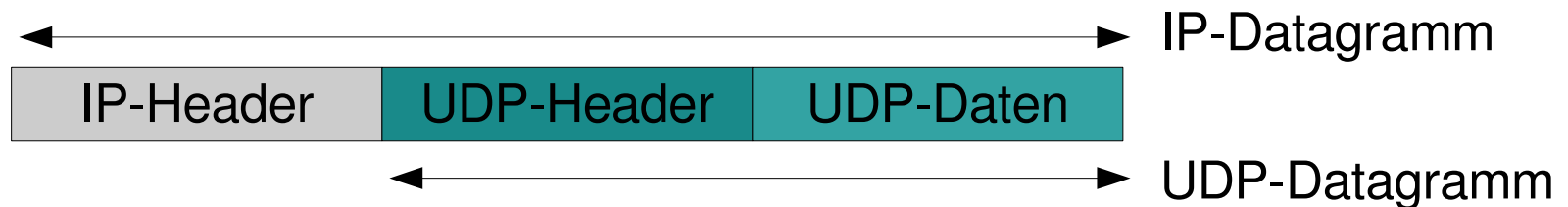
# Transportschicht

- IP: Vermittlungsschicht
  - Nicht zugänglich für Anwendungen!
- Transportschicht als Schnittstelle zum Netz
  - bietet verbindungsorientierten, verlässlichen Dienst
  - bietet verbindungslosen, nicht verlässlichen Dienst
  - Abstraktion von Details der Kommunikation

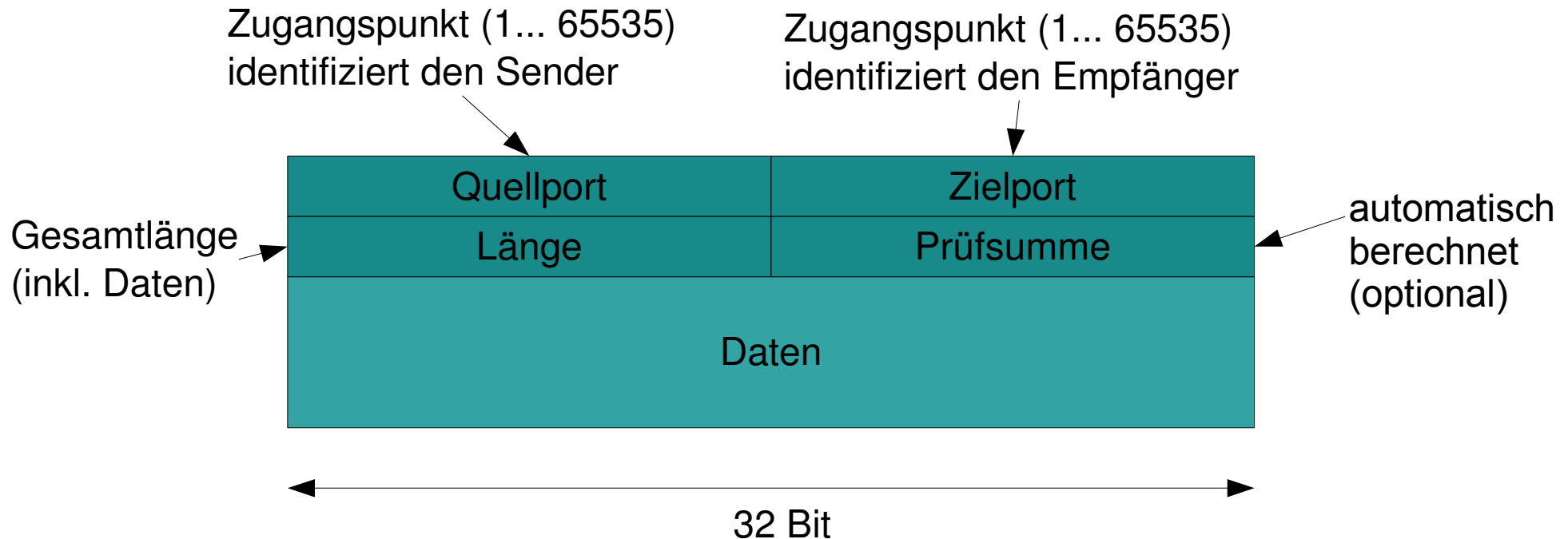
**Ende-zu-Ende-Sicht:** Die Transportschicht erlaubt den höheren Schichten, die gesamte Routingkomplexität zu ignorieren. Es gibt nur ein Ziel, den gewählten Empfänger. Die Router sind „unsichtbar“.

# UDP

- User Datagram Protocol (UDP) [RFC 768]
  - Einfachstes Transportprotokoll: verbindungsloser Transportdienst
  - Nicht verlässlich: Pakete können verloren gehen
  - Überlässt der Anwendung die Fehlerbehandlung und Flusssteuerung
  - Unterstützt Multicast und Broadcast (mit IP Multicast / Broadcast)



# Aufbau eines UDP-Pakets



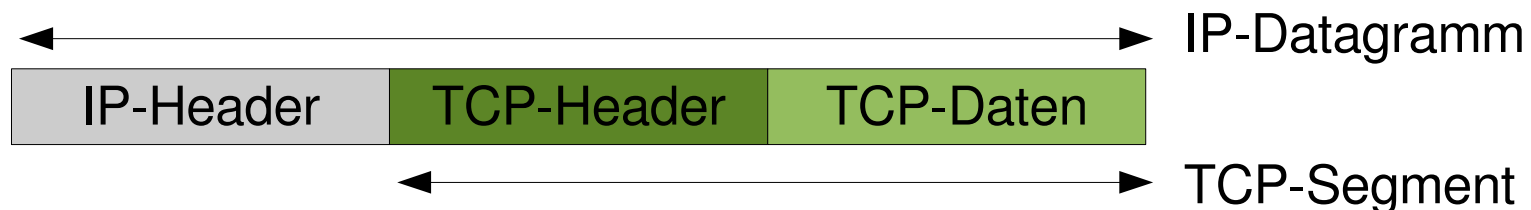
- UDP-Adresse: <IP-Adresse, UDP-Portnummer>
  - Sender und Empfänger so zu identifizieren: Jeder Port kann nur von einem Prozess (einer Anwendung) zur gleichen Zeit verwendet werden.
  - auch „UDP-Socket“ genannt

# UDP

- Anwendungen
  - DNS (Domain Name System)
    - Daten passen meist in ein Datagramm. Verwendet auch TCP.
  - NFS (Network File System)
    - Performanzgründe, heutige Versionen verwenden TCP
  - SNMP (Simple Network Management Protocol)
    - Möglichst geringe Netzlast erzeugen
  - RIP (Routing Information Protocol, dynamische Routingtabellenanpassung)
    - Ebenfalls: Kurze Nachrichten, Netzlast minimieren
  - Multimedia-Streaming
    - geringer Aufwand, keine Notwendigkeit für Verlässlichkeit

# TCP

- Transmission Control Protocol (TCP) [RFC 793, 1122, 1323]
- Bereitstellung eines zuverlässigen Ende-zu-Ende-Bytestroms in einem unzuverlässigen Netz (verbindungsorientierter Transportdienst)
- Verlässlicher Datenstrom
  - Fehlerfrei, keine Verluste, keine Duplikate, Reihenfolge intakt
  - reiner Bytestrom, d.h., Nachrichtengrenzen werden nicht erhalten, z.B.  $\text{send}(170 \text{ Bytes}) + \text{send}(230) = \text{receive}(400)$
  - Teilt Anwendungsdaten in Blöcke (Segmente)  $\leq 64 \text{ KiBytes}$  (meistens ca. 1500 Bytes); jedes Segment wird als ein IP-Datagramm geschickt
- TCP-Adresse =  $\langle \text{IP-Adresse}, \text{TCP-Portnummer} \rangle$  („TCP-Socket“)

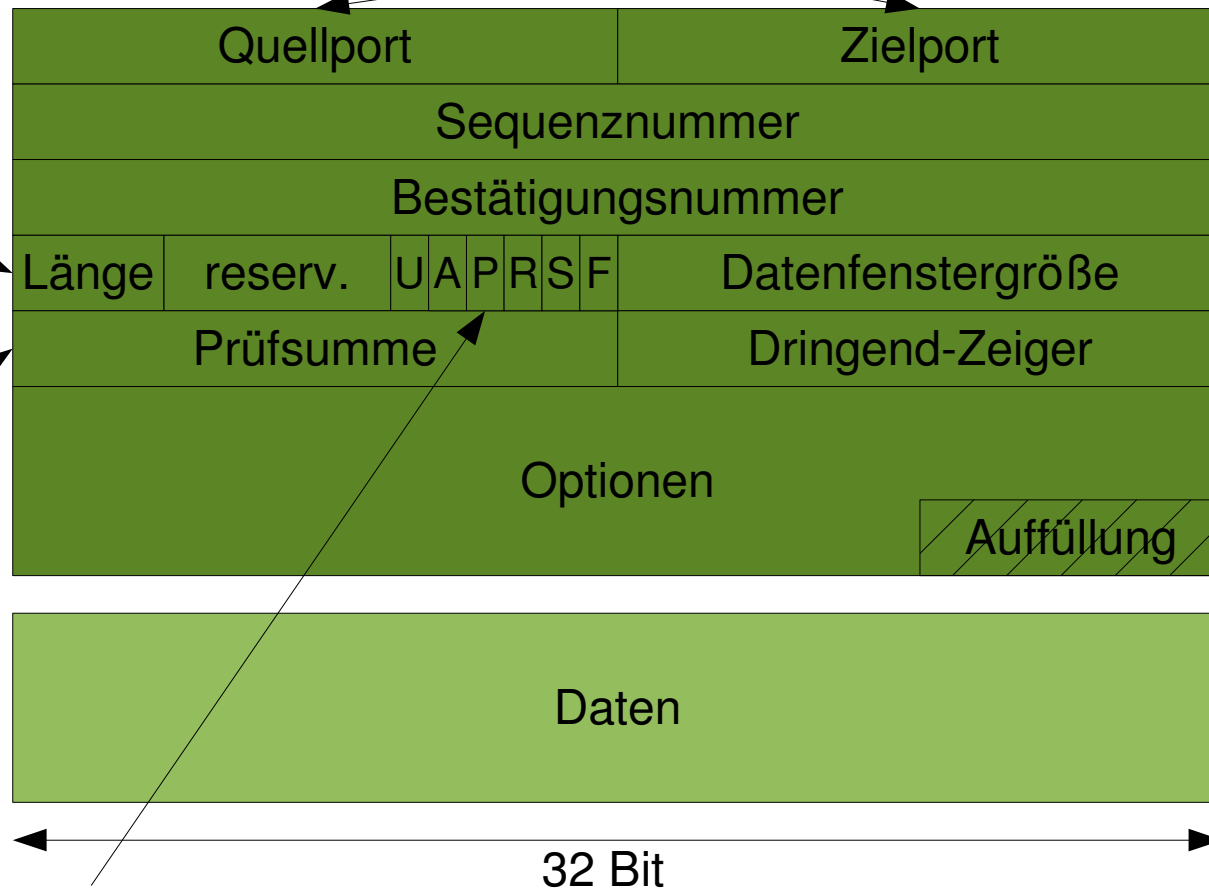


# Aufbau eines TCP-Segments

Identifizieren die Verbindung

Kopflänge als Anzahl 32-Bit-Wörter

Umschließt Kopf, Daten sowie IP-Adresse



verwendet Byte-Nummerierung

Größe des Schiebefensters (Flusssteuerung)

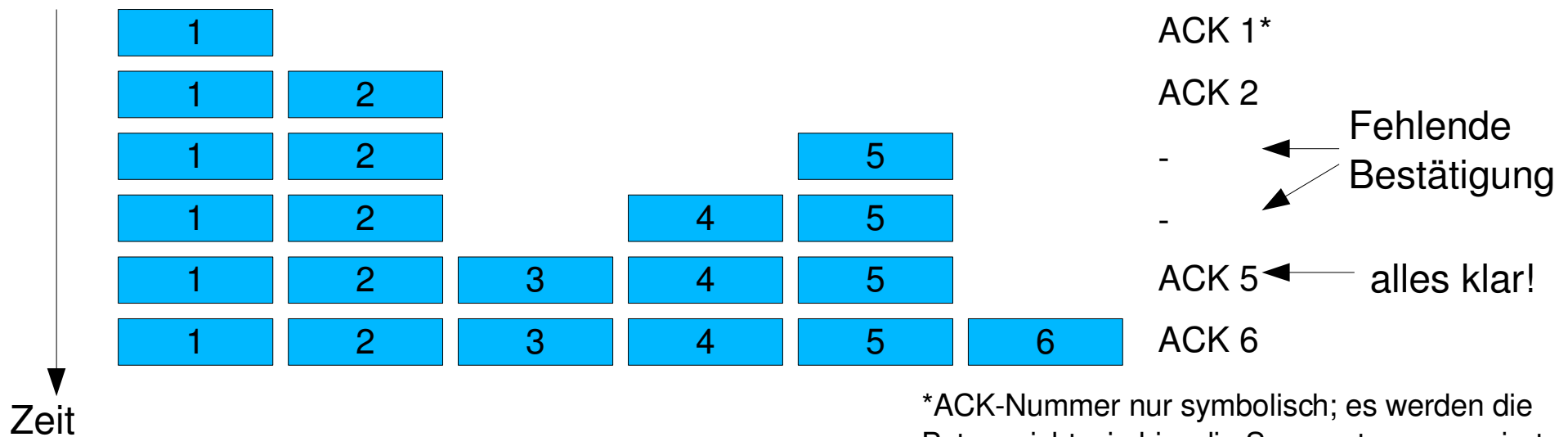
Zeigt auf dringliche Daten

**Flags:** URG, ACK, PSH, RST, SYN, FIN  
(dienen der Verbindungssteuerung)



# Verlässliche Kommunikation

- Segmente werden bestätigt
  - Empfänger schickt Bestätigung (ACK) mit Folgennummer an den Absender
  - Es wird nur bestätigt, wenn alle bisherigen Segmente eingetroffen sind
- Beispiel: sechs Segmente, der Reihe nach verschickt (1, 2, 3, 4, 5, 6), Paket 3 und 4 verspäten sich in der Ankunft



\*ACK-Nummer nur symbolisch; es werden die Bytes, nicht wie hier die Segmente nummeriert

# TCP- und UDP-Dienste

- Je Port kann nur ein Dienst laufen
- Aber: Dienst „von außen“ nicht identifizierbar
  - Man müsste sich verbinden, um zu erfahren, welcher Dienst an diesem Port läuft
  - Daher wird für bestimmte Ports („Well-Known“-Ports, [ RFC 1700 ]) ein zugehöriger Dienst angenommen.

## • TCP

- FTP: 20 und 21  
(Dateitransfer)
- SSH: 22 (Anmeldung)
- SMTP: 25 (E-Mail)
- HTTP: 80 (WWW)

## • UDP

- DNS: 53  
(Domännennamen)
- NFS: 111  
(Netzdateisystem)
- SNMP: 161 und 162  
(Management)

# Socket

- TCP / UDP – Programmierschnittstelle in UNIX
  - Socket-Abstraktion ( = Steckdose )
    - angelehnt an das Datei-Modell der UNIX-Ein- und -Ausgabe
    - erstmals in BSD UNIX, heute allgemein gebräuchlich
  - Erste Implementierung in C, heute Implementierungen für verschiedene Programmiersprachen (z.B. C++, Java)
  - TCP: Server-Socket vs. Client-Socket
    - Server-Socket: wartet passiv auf Verbindungsanfrage auf einem Port; eigentliche Verbindung läuft über neu vergebenen Port
    - Client-Socket stellt aktiv Verbindungsanfrage an einen Port
- Näheres siehe Übung

# Protokolle der Anwendungsschicht

# Das World Wide Web

- Verteiltes Hypermedia-System
  - dezentrale Speicherung von Dokumenten
  - Verbindungen (Hyperlinks) zwischen den Dokumenten
  - Dokumente mit Text, Grafik, Audio, Video, ...
  - einfache Navigation von Dokument zu Dokument
- Standards für die Beschreibung der Dokumentenstruktur und für das Protokoll des Dokumentenzugriffs
- Einheitliche Schnittstelle für beliebige Dokumentarten
- Integration anderer Internet-Anwendungen im WWW-Browser
- Web-Inhalte von WWW-Servern angeboten
  - über Well-known-Port 80

# Das World Wide Web

- Zeilenorientierte WWW-Browser
  - reine Textdarstellung ohne Bilder, Audio, Video; Bedienung nur über Tastatur
  - für Verbindungen mit geringer Übertragungsrate oder Systeme mit begrenzten Ressourcen

Lynx

```
# World Wide Web Consortium (p1 of 15)

#meta Technologies | News | Search | Contents | Offices | Systems |
Donors | Footnotes | RSS

The World Wide Web Consortium (W3C)

Leading the Web to Its Full Potential...

Skip to Technologies | Activities | Technical Reports | Site Index |
New Visitors | About W3C | Join W3C | Contact W3C

The World Wide Web Consortium (W3C) develops interoperable
technologies (specifications, guidelines, software, and tools) to lead
the Web to its full potential. W3C is a forum for information,
commerce, communication, and collective understanding. On this page,
you'll find W3C news, links to W3C technologies and ways to get
involved. New visitors can find help in Finding Your Way at W3C. We
encourage organizations to learn more about W3C and about W3C
Membership.

-- Leertaste für nächste Seite --
Pfeile: Auf/Ab: andere Seite im Text. Rechts: Verweis folgen; Links: zurück.
H)ilfe O)ptionen P) Druck G)ehe zu M) Hauptseite Q) Beenden /=Suche <-=History
```

# Das World Wide Web

- Grafische WWW-Browser
  - direkt darstellbar sind Dokumente mit HTML, ASCII, Grafiken in den Formaten GIF, XBM, XPM, JPEG, PNG, ...
  - Einbindung externer Betrachter für z.B. Postscript, PDF, spezielle Grafikformate (TIFF, PICT, ...), Audio (WAV, MP3, ...), Video (MPEG, QuickTime, ...)
  - Beispiele: Firefox, Opera, Konqueror, Internet Explorer

# Grafische Ausgabe

World Wide Web Consortium - Konqueror

Dokument Bearbeiten Ansicht Gehe zu Lesezeichen Extras Einstellungen Fenster Hilfe


Adresse: <http://www.w3.org/>

**W3C** WORLD WIDE WEB  
consortium

Leading the Web to Its Full Potential...

[Activities](#) | [Technical Reports](#) | [Site Index](#) | [New Visitors](#) | [About W3C](#) | [Join W3C](#) | [Contact W3C](#)

The World Wide Web Consortium (W3C) develops interoperable technologies (specifications, guidelines, software, and tools) to lead the Web to its full potential. W3C is a forum for information, commerce, communication, and collective understanding. On this page, you'll find [W3C news](#), links to [W3C technologies](#) and ways to [get involved](#). New visitors can find help in [Finding Your Way at W3C](#). We encourage organizations to learn more [about W3C](#) and [about W3C Membership](#).

W3C A to Z	News	Search
<ul style="list-style-type: none"><li><a href="#">Accessibility</a></li><li><a href="#">Amaya</a></li><li><a href="#">Annotea</a></li><li><a href="#">Binary XML</a></li><li><a href="#">CC/PP</a></li><li><a href="#">Compound Document Formats</a></li><li><a href="#">CSS</a></li><li><a href="#">CSS Validator</a></li><li><a href="#">Device Independence</a></li><li><a href="#">DOM</a></li><li><a href="#">HTML</a></li><li><a href="#">HTML Tidy</a></li><li><a href="#">HTML Validator</a></li><li><a href="#">HTTP</a></li><li><a href="#">InkML</a></li><li><a href="#">Internationalization</a></li><li><a href="#">Jigsaw</a></li><li><a href="#">Libwww</a></li></ul>	<p>► <b>W3C Hosts Fifth Annual Technical Plenary Week</b></p>  <p>2004-02-28: W3C holds its <a href="#">Technical Plenary Week</a> from 28 February - 4 March in Boston, Massachusetts, USA where 30 W3C Working Groups and Interest Groups hold face-to-face meetings. Participants and invited guests attend plenary day for talks and demos on extensibility and versioning, XML, test suites, Web applications, Web site usability and design, multimodal interaction and voice. Read the <a href="#">minutes</a>, learn how to <a href="#">join W3C</a> and read <a href="#">about</a></p>	<p>Google</p> <p>Search W3C</p> <input type="text"/> <input type="button" value="Go"/>

Search W3C Mailing Lists

Members

**Oxford Brookes University**

OXFORD BROOKES UNIVERSITY

The Department of Computing undertakes research in web-based systems and teaches a specialist MSc in Web Technologies based on W3C technologies. The



# Hypertext Transfer Protocol

- kurz „HTTP“ genannt; RFC 2616
  - Aktuell: Version 1.1
- Dient meist dem Transfer von Dateien vom Server zum Klient, aber auch der Übertragung z.B. von Formulardaten
  - ist trotz des Namens nicht auf HTML-Dokumente beschränkt
  - Übertragung stets vom Klient initiiert
  - einfaches ASCII-Protokoll, benutzt Elemente von MIME

HTTP ist ein Protokoll der Anwendungsschicht. Es nutzt TCP als Transportprotokoll (könnte theoretisch auch UDP verwenden).

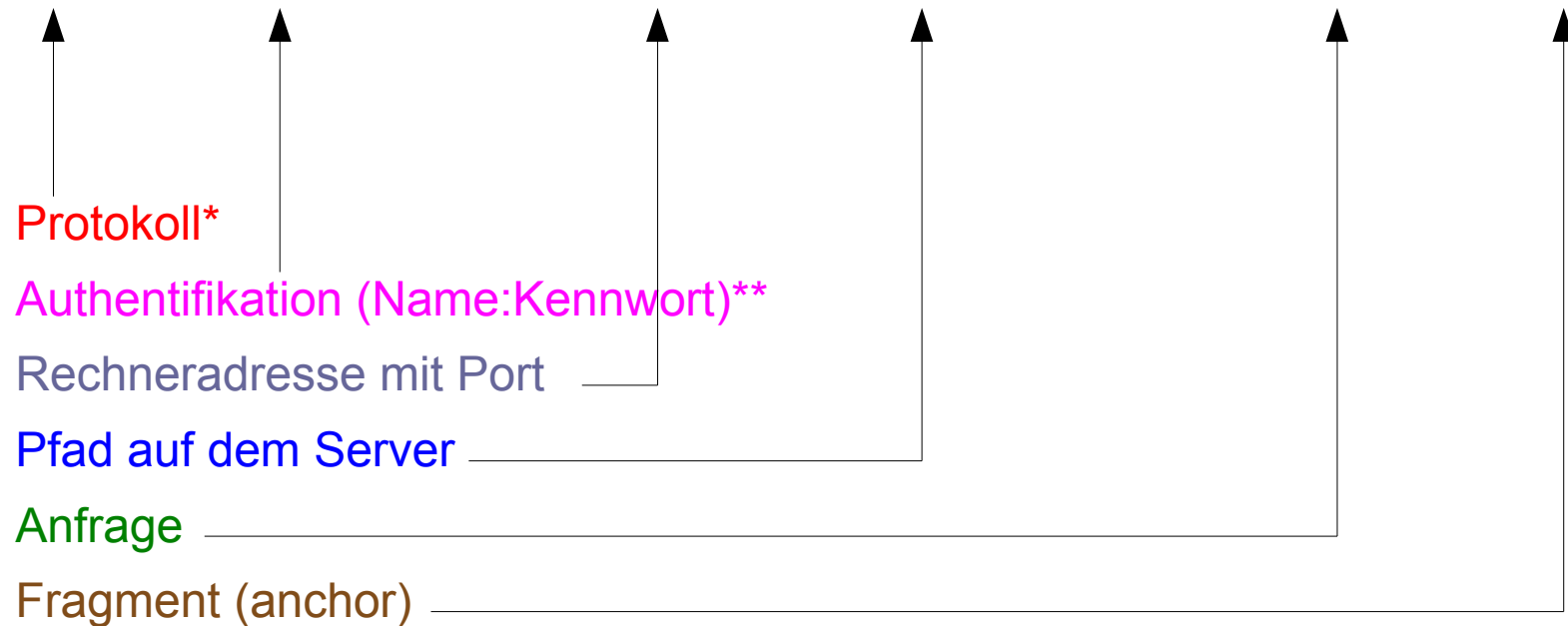
# Adressierung von Webinhalten

- Zugriff auf WWW-Seiten erfordert Klärung:
  - Wie lautet der lokale Name der Seite beim Server?
  - Wo liegt die Seite?
  - Wie kann auf die Seite zugegriffen werden?
- Adresse besteht aus Angaben zu Protokoll, Socket, Datei, Parameter
- Uniform Resource Locator
  - Aufgabe: Angabe der Position einer Ressource (meist Datei)
  - Geht davon aus, dass die Ressource über das Netz geholt werden muss
  - Wenn die Ressource nur einmal vorhanden ist, kann damit auch eine eindeutige Bezeichnung geliefert werden

Beispiel: <http://www.uni-dortmund.de/>

# Das URL-Format

<http://michael:pw1834@www.meinhost.de:80/docs/index.html?lang=de&sys=lx#glossar>



\* auch „Schema“ genannt

\*\* *Kennwort* soll nicht mehr angegeben werden, und Server müssen diese Angabe nicht mehr beachten (deprecated).

# Das URL-Format

- Bis auf die Rechneradresse (DNS-Name oder IP-Adresse) und das Protokoll ist alles optional
  - Protokoll kann vom Browser ergänzt werden (meist http)
  - Pfad wird als Server-Stammverzeichnis angenommen, wenn fehlend (entscheidet Server)
- Anfrage (Query)
  - Wenn vorhanden, mit „?“ angeschlossen
  - Format *Name=Wert*, mit „&“ getrennt
- Fragment (Anchor)
  - Obliegt dem Klienten, das übermittelte Dokument geeignet darzustellen (z.B. zu diesem Punkt herunterrollen)

# Uniform Resource Identifiers

- Verallgemeinerung des URL-Konzepts
- Nicht mehr nur Ort, sondern auch Identität bestimmend
- Format

*Schema:Hierarchische Angaben[?Anfrage][#Fragment]*

- Hierarchische Angaben: schemaspezifische Angaben, eventuell mehrgliedrig
- Jede URL ist eine URI
  - Protokoll wird generell als „Schema“ gelesen, z.B. das „http-Schema“
  - Hierarchische Angaben der URL: Rechner mit Port einschließlich Pfad (ggf. vorgehängte Authentifikation)

# Uniform Resource Names

- Uniform Resource Names (URNs) sind dauerhafte, ortsunabhängige Namen für WWW-Ressourcen
  - Es soll einfach sein, andere Namensräume auf URNs abzubilden
  - Schema: *urn:Namensraum:namensraumspezifische Angabe*
  - Beispiele
    - urn:[ietf:rfc:2141](#) // RFC 2141
    - urn:[ietf:std:50](#) // ...
    - urn:[isbn:3-8266-5042-5](#) // ...
  - URNs geben nicht die Quelle an; diese muss erst bestimmt werden
    - URN → URL
- Jede URN ist eine URI
  - Schema „urn“

# URI-Beispiele

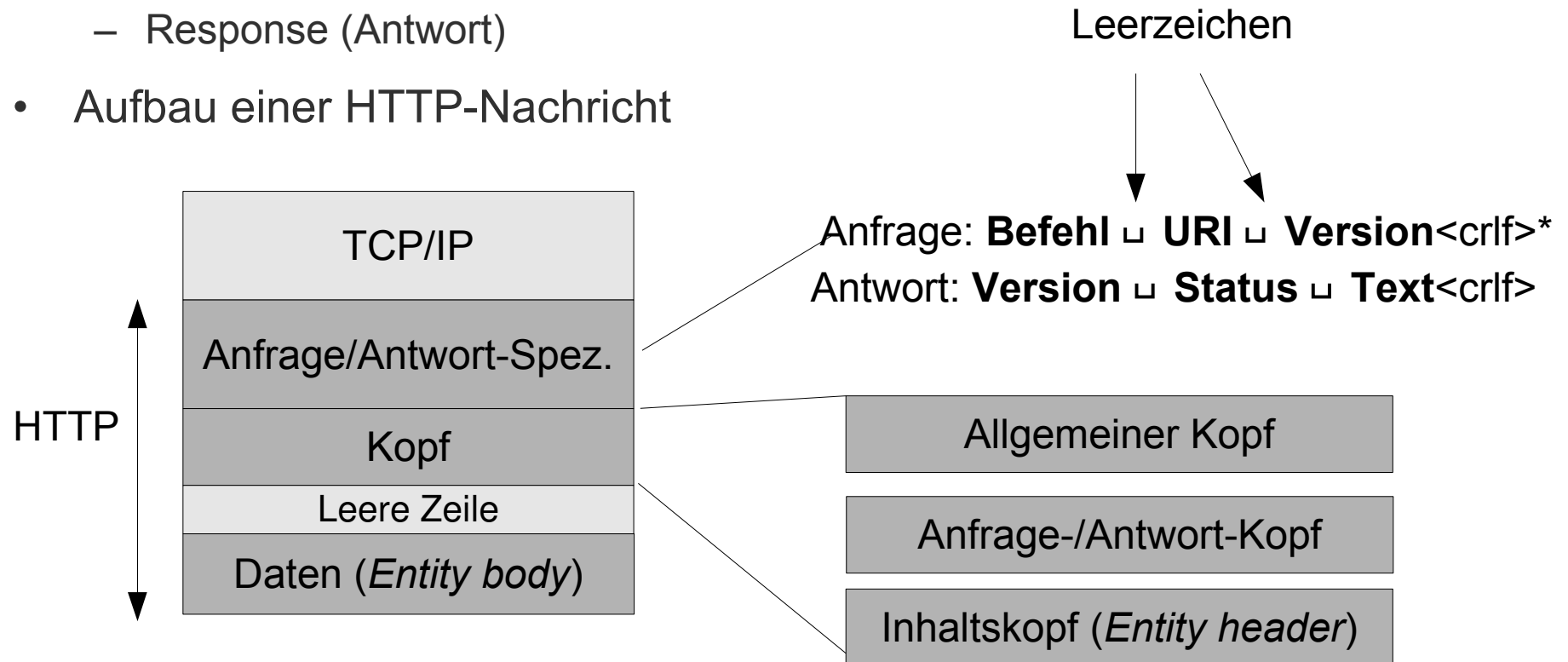
- <http://www.mit.edu:8001/people/dshapiro/macsites.html>
  - Webseite
- <file:///WWW/beispiel.html>
  - URI (mit fehlender Rechnerangabe), Datei im lokalen Dateisystem
- <ftp://ftp.cs.mit.edu/README>
  - Datei auf Server
- <mailto:majordomo@example.com?body=subscribe%20bamboo-l>
  - E-Mail an diese Adresse erstellen (RFC 2368)
- <news:comp.infosystems.www.users>
  - Diskussionsforum

Die frühere strikte Partitionierung (was nicht URL ist, ist URN) ist mittlerweile aufgegeben worden. Manche URIs passen weder gut zu den URLs, noch sind es URNs (z.B. „data:“, eingebettetes Datenelement)

In technischen Dokumenten wird generell nur noch der Begriff URI verwendet.

# HTTP

- Unterscheidung zweier Nachrichtentypen
  - Request (Anfrage)
  - Response (Antwort)
- Aufbau einer HTTP-Nachricht



\*<crlf> sind die beiden Zeichen mit ASCII-Code 13 und 10; dies ist unabhängig vom Betriebssystem von Server oder Klient so festgelegt



# Anfrage (Request)

- Befehle (auch Methoden genannt)
  - GET Lesen einer Seite (allg.: einer Ressource)
  - HEAD Lesen eines Seitenkopfs (insbesondere Angaben, wann die Seite zuletzt geändert wurde)
  - PUT Abspeichern einer Seite (Neuanlegen)
  - POST Anhängen von Daten (wird zum Übertragen von Formulardaten verwendet)
  - DELETE Löschen einer Seite (sofern erlaubt)
- Server kann entscheiden, worauf sich die URL tatsächlich bezieht
  - z.B. wird / auf eine spezielle Datei (index.html) umgelenkt
- Beispiel
  - **GET /index.html HTTP/1.1**

# HTTP-Kopfzeilen

- Jede Zeile besteht aus einer Angabe der Form **Name:Wert** und endet mit einem Zeilenvorschub
  - Wert kann eine Komma-separierte Liste sein

- Beispiel

## GET / HTTP/1.1

**Host:** www.vs.uni-kassel.de

**User-Agent:** Mozilla/5.0 (X11; U; Linux i686; de-DE; rv:1.7.6) Gecko/20050226 Firefox/1.0.1

**Accept:** text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,image/png,\*/\*;q=0.5

**Accept-Language:** de-de,de;q=0.8,en;q=0.5,en-us;q=0.3

**Accept-Encoding:** gzip,deflate

**Accept-Charset:** ISO-8859-15,utf-8;q=0.7,\*;q=0.7

**Keep-Alive:** 300

**Connection:** keep-alive

**Cookie:** wikidbUserID=3; wikidbToken=d9173a300eca8100; wikidbUserName=bla

**Cache-Control:** max-age=0

Zum Anfrage-Kopf gehörend  
Zum allgemeinen Kopf gehörend  
Zusatz

# HTTP-Header

- Beispiele
  - **Host:** Angabe des angefragten Webservers (es könnten mehrere Hostnamen der Adresse zugeordnet sein)
  - **Referer:** Angabe, auf welcher Webseite der angeklickte Verweis stand
  - **Accept-Language:** Die Webseite könnte in mehreren Sprachversionen vorliegen; hiermit wird die Präferenz festgelegt
  - **Content-Encoding:** Teil des Inhaltskopfs; wurden irgendwelche Filter auf den Inhalt angewendet (z.B. Kompression mit gzip)?
  - **Last-Modified:** Bei Dateien in der Regel der Zeitstempel; allgemein der Zeitpunkt, als sich zuletzt der Zustand änderte
- Kopfzeilen können beliebig angeordnet sein, sie müssen nicht nach Kategorien gruppiert sein.
  - Es wird jedoch empfohlen, erst allgemeine, dann anfragespezifische, dann inhaltspezifische Angaben zu nennen.

# Antwort (Response)

- Antwort (wieder reiner Text) besteht aus
  - Statuszeile
  - Kopf
  - Daten
- Statuszeile
  - Dreistelliger Zahlencode mit Klartexthinweis
  - Beispiele siehe rechts

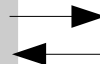
2xx	Success
200	OK
201	Created
202	Accepted
204	No content
3xx	Redirection
301	Moved permanently
307	Temporary redirect
4xx	Client error
400	Bad request
401	Unauthorized
402	Payment required
403	Forbidden
404	Not found
5xx	Server error
500	Internal server error
503	Service unavailable

# Beispiel

```
$ telnet www 80
Trying 141.51.110.124...
Connected to www.
Escape character is '^]'.
GET / HTTP/1.1
Host:www
```

```
Connection closed by
foreign host.
```

Klient



```
HTTP/1.1 200 OK
Date: Thu, 31 Mar 2005 08:41:00 GMT
Server: Apache/2.0.50 (Ubuntu) mod_ssl/2.0.50
OpenSSL/0.9.7d
Content-Length: 6758
Content-Type: text/html
```

```
<html>
  <head>
    <title>Distributed Systems -
    FB16 - Uni-Kassel</title>
```

```
...
```

```
</html>
```

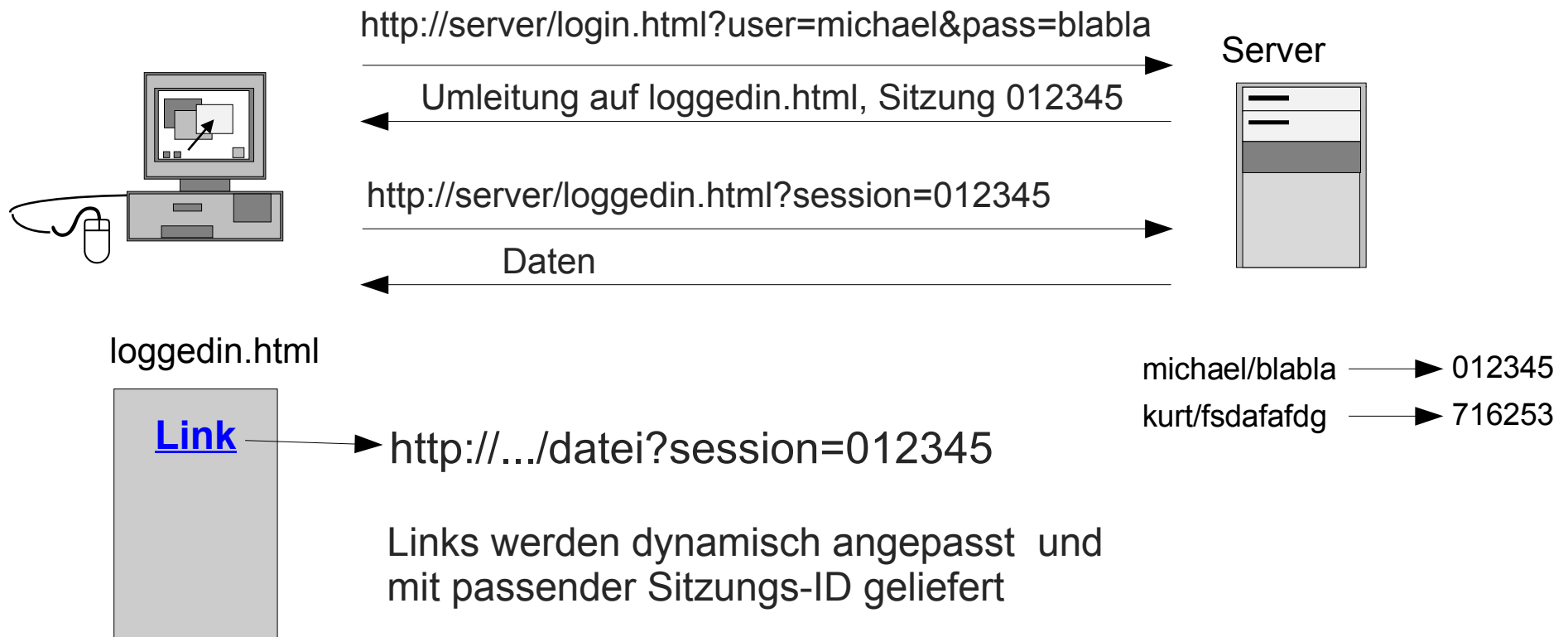
Server „www“

# HTTP und Sitzungen

- HTTP ist nicht sitzungorientiert
  - Jede HTTP-Abfrage ist eine eigene Verbindungsaufnahme
  - Möglich, im Kopf anzugeben, die Verbindung über ein paar Anfragen hinweg wiederzuverwenden (Keep-alive)
    - Jedoch ist jede Anfrage völlig eigenständig und hat nichts mit einer vorherigen Anfrage zu tun
- Problem bei geschützten Seiten
  - Wie merkt sich das System den Login?
- Erhaltung einer Sitzung auf Applikationsebene beim Klienten und Server
- Sitzungsreferenzen
  - über URL
  - über Cookies

# Sitzungen

- URL-Methode
  - Server erzeugt eine Sitzungs-ID und erzeugt passende Verweise, welche diese ID beinhalten




# Cookies

- Begriff vom Netscape geprägt
  - Offiziell: HTTP State Management Mechanism (RFC 2965)
- Spezieller Protokollkopf
  - vom Server in einer Antwort an den Klienten geschickt
  - *Set-Cookie: Cookie1,Cookie2,Cookie3,...* (veraltet)
  - Neuer: *Set-Cookie2: Cookie1,Cookie2,Cookie3,...*
  - Jedes Cookie ist ein Paar Name=Wert
  - Zusatzparameter je Cookie mit Semikolon getrennt
    - z.B. ist dies ein **einzelnes** Cookie: `lastvisit=1112277815;Path=/;Comment=Test`



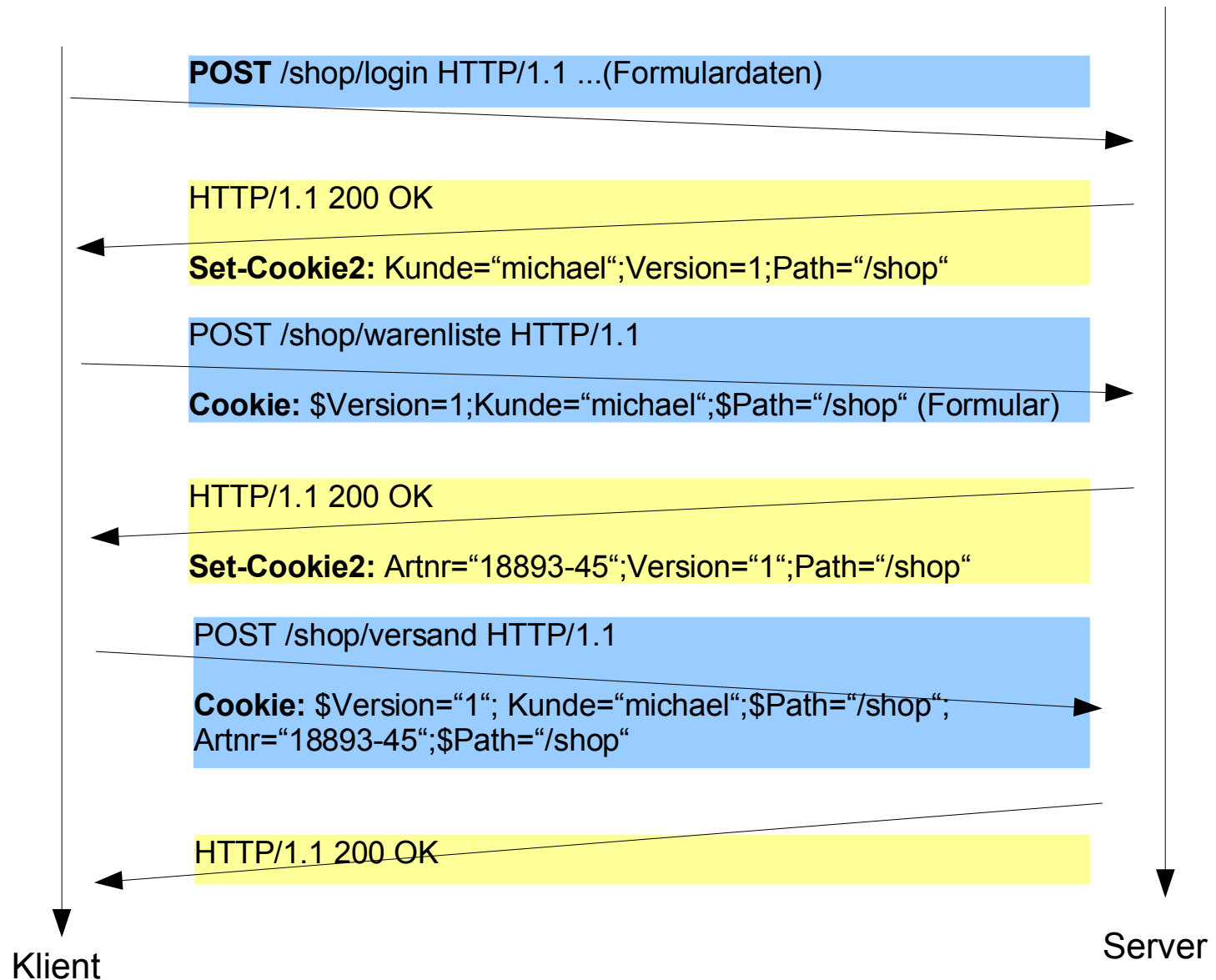
# Cookies

- Cookie wird vom Browser gespeichert
  - temporär: nur im Speicher (auch „Session Cookie“)
  - permanent: im Dateisystem (auch „Persistent Cookie“)
- Prinzip
  - Browser ist angehalten, das Cookie (als Kopfzeile) wieder an den Server zu schicken, wenn es zu Host/Port/URI ein passendes Cookie gibt (das nicht verfallen ist)
- Kopfzeile
  - **Cookie:** Version CookieSp1, CookieSp2, ...  
wobei Version=**\$Version**=Wert und *CookieSpX* so aussieht:  
  
Name=Wert[;**\$Path**=Pfad][;**\$Domain**=Domain][;**\$Port**=Port]  
  
mit den Werten vom Server



Nur diese 3 Parameter,  
nur diese Reihenfolge  
(trennt dadurch Cookies)

# Beispiel



# World Wide Web

- Weiteres zur Technik des World Wide Web in speziellen Folgen dieser Vorlesung
  - World Wide Web (1)
  - Web 2.0