

# Grundlagen der Informatik für Ingenieure I

## Background

### 7. Rechnernetze - Einführung

- 7.1 Geschichte
- 7.2 Standardprotokolle
- 7.3 Struktur und Funktionsweise
  - 7.3.1 Hardware
  - 7.3.2 Struktur
  - 7.3.3 Client-Server-Modell
  - 7.3.4 Adressierung und Routing
- 7.4 Protocol-Layers
- 7.5 Dienste / Höhere Protokolle etc.
- 7.6 WWW
- 7.7 Sonstiges

## 7.1 Geschichte

### 1 Entwicklung

- Initiative 1968/69:  
Forschungsbehörde im US-Verteidigungsministerium - ARPA  
(Defense) Advanced Research Project Agency
  - Universitäten entwickeln ein Computer-Netz (ARPANET)
  - 1970: 4 Computer verbunden  
1971: 13, 1972: 23, ..., 1976: 63
- 80er-Jahre:
  - Aufteilung in militärisches und wissenschaftliches Netz
  - neue Technik um mehrere Computer-Netze miteinander zu verbinden  
→ Internet
  - neue Netze von Forschungsorganisationen, NASA, ...
  - Netze in Europa
  - 1981: 213 Computer, 1984: 1024, 1989: 159.000

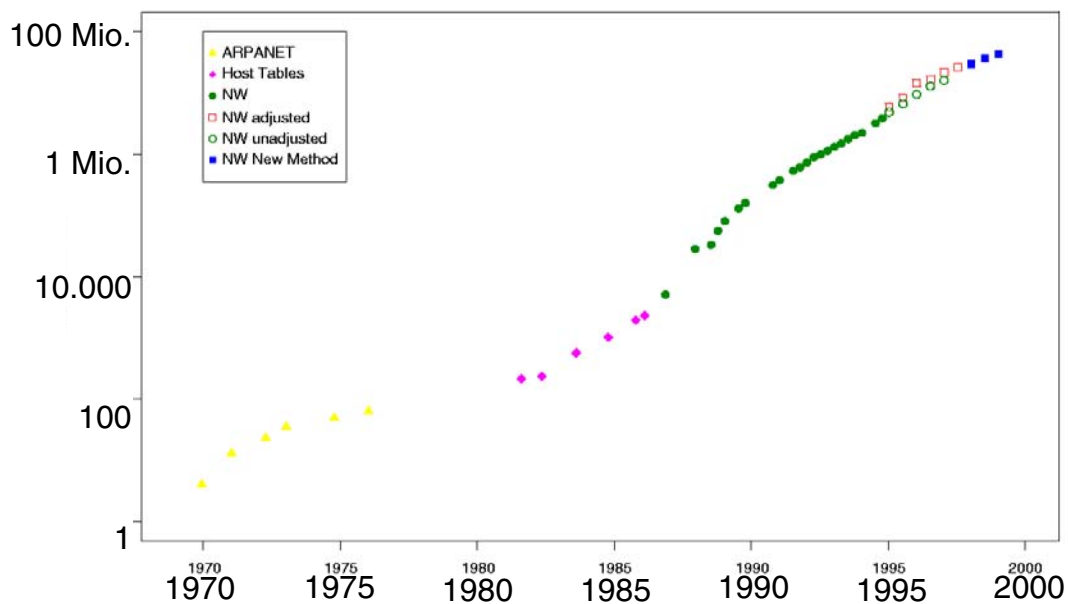
# 1 Entwicklung

## ■ 90er-Jahre

- Internet bislang nur zu Forschungszwecken
- Basis-Software für das World Wide Web wird entwickelt
- erste kommerzielle Netzwerke mit Internet-Technologie
- Unternehmen schließen sich an
- große Zahl kommerzieller Netz-Provider kommt auf den Markt
- zunehmend für Privathaushalte interessant
- Zusammenwachsen von Telefon, Rundfunk, Fernsehen und Computernetzen beginnt
- neue Anwendungsbereiche für Internet-Technologie (z. B. Haustechnik)
- 1990: 300.000 Computer, 1995: 6 Mio., 2000: 100 Mio. ?

# 2 Wachstum

Rechner im Internet 1969 - 1999



◆ 1 Milliarde WWW-Seiten durch Suchmaschinen indiziert (geschätzt: 550 Mrd ex. tatsächlich)

### 3 Geschwindigkeit

#### ■ typische Geschwindigkeit von Internetverbindungen

|       | Weitverkehr | lokale Netze |
|-------|-------------|--------------|
| 1969: | 10 kBit/s   |              |
| 1975: | 50 kBit/s   | 100 kBit/s   |
| 1980: | 250 kBit/s  | 10 MBit/s    |
| 1985: | 1 MBit/s    | 34 MBit/s    |
| 1990: | 2 MBit/s    | 100 MBit/s   |
| 1999: | 6 MBit/s    | 1 GBit/s     |
| 2003: | 1 GBit      | ?            |

### 4 Anwendungen

- 70er-Jahre: Electronic Mail  
Dateitransfer  
interaktiver Zugang zu Rechnern
- 80er-Jahre: News (schwarze Bretter, Diskussionsforen)  
frei zugängliche Dateiarhive
- 1990/1991: Software zur automatischen Suche in Dateiarhiven  
Elektronische Bibliothekskataloge
- 1992/1993: Software für World Wide Web entsteht
- 1994: WWW-Datenmenge wächst um Faktor 1.600
- ab 1995: Übertragung von Bildern und Ton  
Videokonferenzen

## 7.2 Standardprotokolle

- Spezifiziert in RFCs (Request for Comment)  
(<http://www.faqs.org/rfcs/rfc-index.html>)
  - **UDP** (User Datagram Protokoll) RFC 768 August 1980 4 Seiten
  - **IP** (Internet Protokoll) RFC 791 September 1981 51 Seiten
  - **ICMP** (Itn. Contr. Message Prot.) RFC 792 September 1981 22 Seiten
  - **TCP** (Transmission Control Prot.) RFC 793 September 1981 85 Seiten
  - **ARP** (Adress Resolution Prot.) RFC 826 November 1982 10 Seiten
  - **RARP** (Reversed Adress ...) RFC 903 Juni 1984 4 Seiten

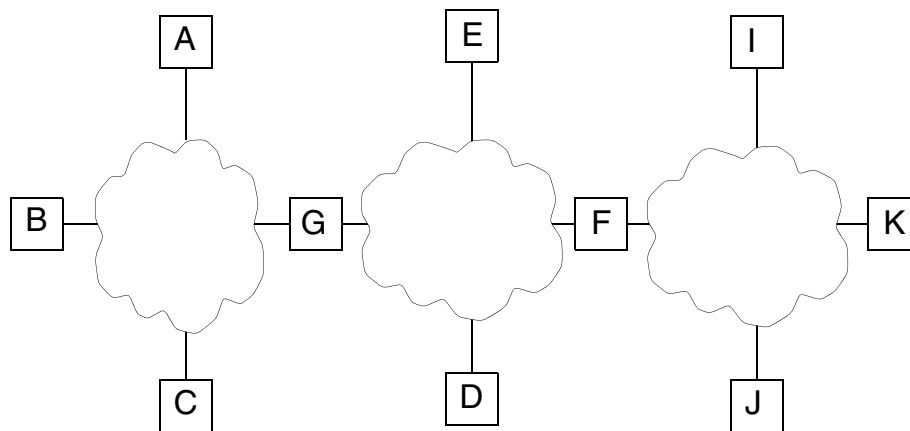
## 7.3 Struktur und Funktionsweise

### 1 Hardware

- Medien:
  - **Ethernet** (10/100 MBit/s, 50-500m)  
Bus-System mit CSMA/CD  
(Carrier Sense Multiple Access/Collision Detection)  
Gigabit-Ethernet über Glasfaser
  - **Token Ring** (4/16 MBit/s, 150-300m zw. Repeatern); (IBM)
  - **FDDI** (100 MBit/s, ca. 100km; Fiber Distributed-Data Interface)
  - **ATM** (155/622/+ MBit/s, variabel; Asynchronous Transfer Mode)
- Verbindung von Netzen:
  - Gateways / Router / Switches

## 2 Struktur

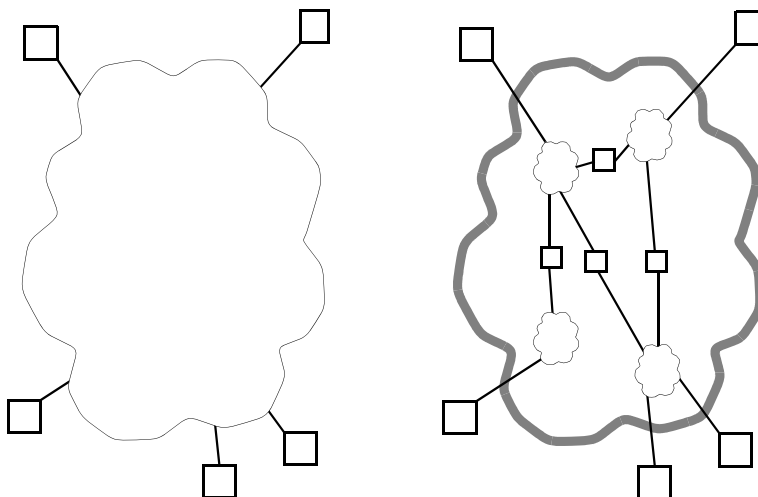
- **Internet** = Verbund vieler Einzelnetze zu einem virtuellen Gesamtnetz
- Beispiel: Verknüpfung von drei Netzwerken



- G und F sind Gateways

## 2 Struktur

- **Benutzersicht** **Systemsicht**



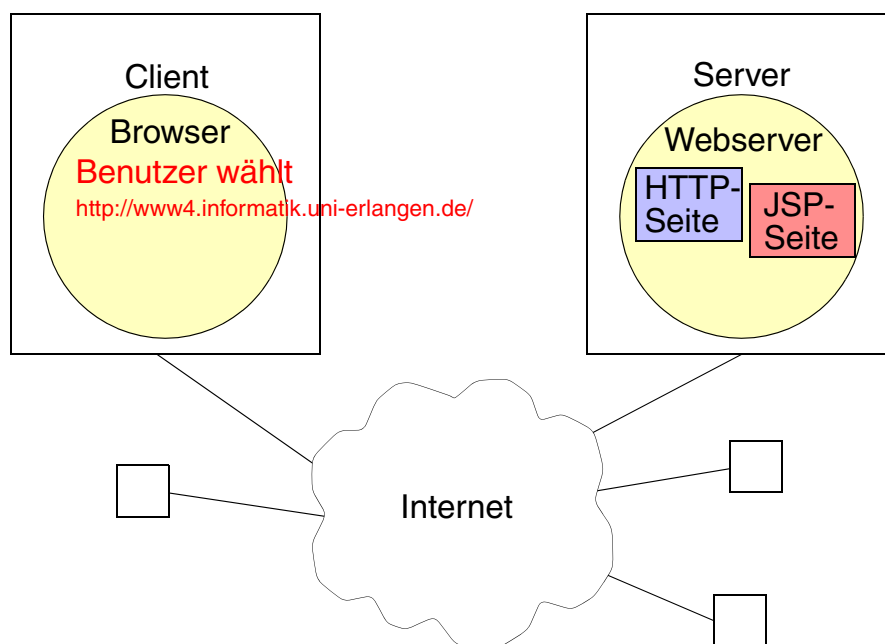
- Für "normale" Anwendungsprogrammierung reicht Benutzersicht
- Verstehen von Sicherheitsproblematiken erfordert Detailwissen.

### 3 Client-Server-Modell

- **Server:** Programm, das einen Dienst (Service) anbietet, der über einen Kommunikationsmechanismus (z.B. Internet) erreichbar ist
- **Server**
  - Akzeptieren Anforderungen anderer Rechner
  - Führen angebotenen Dienst aus
  - Schicken das Ergebnis zurück zum Sender der Anforderung
  - Server sind normalerweise als normale Benutzerprozesse realisiert
- **Client:**
  - Ein Programm wird ein Client, sobald es
    - eine Anforderung an einen Server schickt und
    - auf eine Antwort wartet
- Client-Server-Modell im Internet weit verbreitet

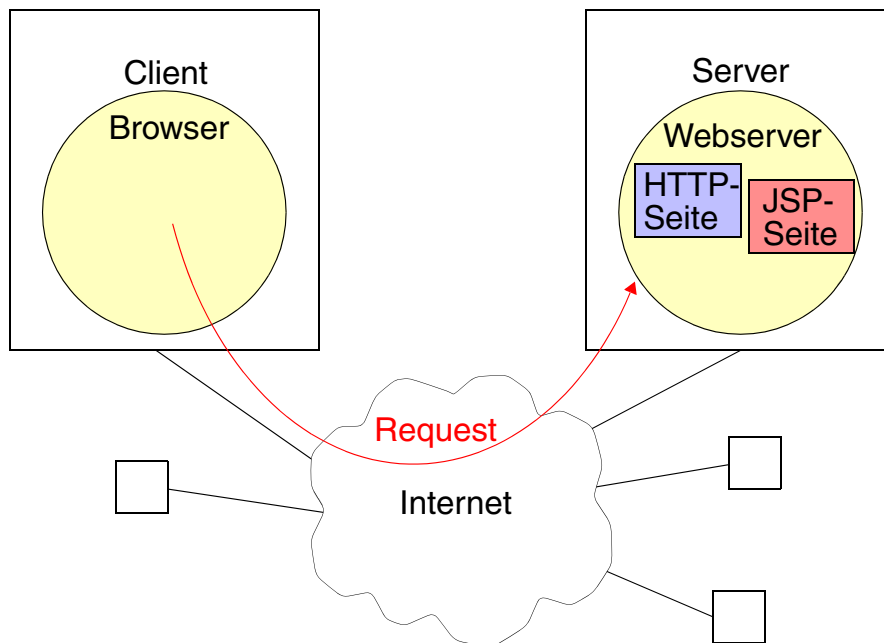
### 3 Client-Server-Modell

- **Beispiel: Webserver**



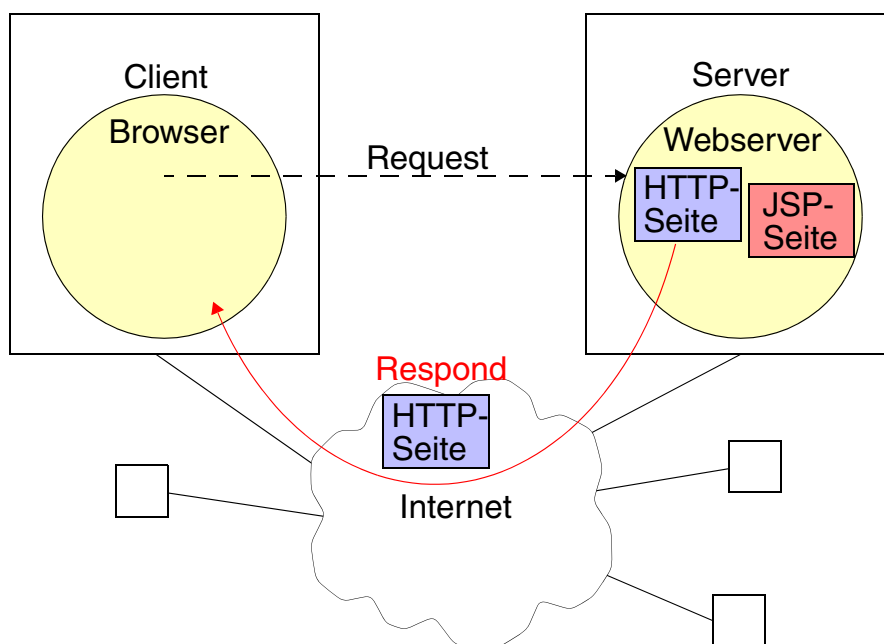
### 3 Client-Server-Modell

#### ■ Beispiel: Webserver



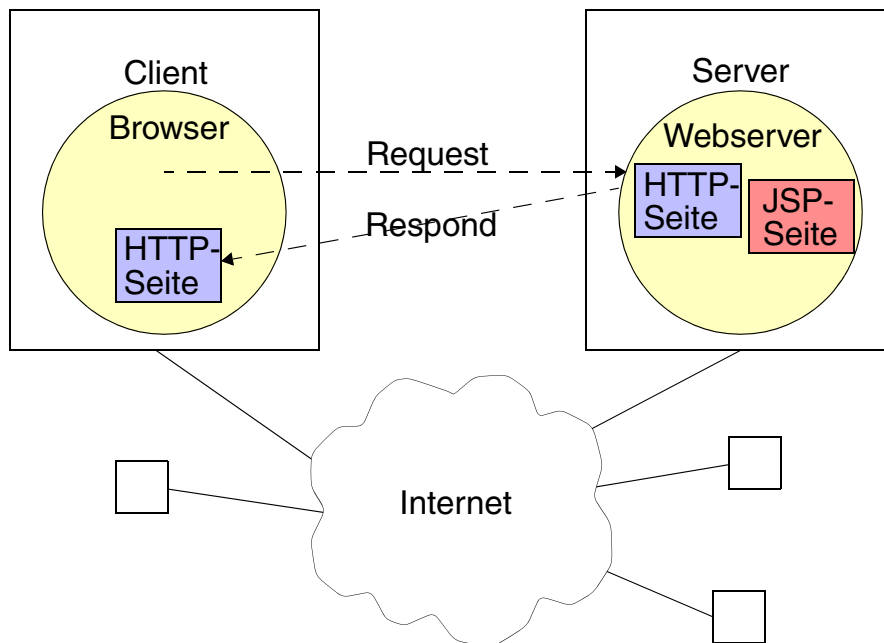
### 3 Client-Server-Modell

#### ■ Beispiel: Webserver



### 3 Client-Server-Modell

#### ■ Beispiel: Webserver

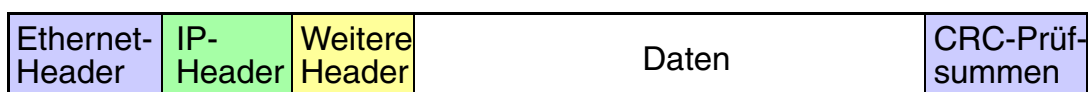


### 4 Adressierung und Routing

#### ■ Internet Protocol – IP

- Unabhängig von verwendeter Hardware
- Ermöglicht einheitliche Adressierung aller Rechner
- Bildet virtuelles Netzwerk auf der Basis vieler physischer Netzwerke
- Paketorientiert: IP-Datagramm
- Unzuverlässige Datenübertragung
- Routing-Konzepte (mittels ICMP)

#### ■ Realisierung:





## 4 Adressierung und Routing

### ■ Aufbau von Ethernet- und IP-Header

|                                 |                                  |                        |
|---------------------------------|----------------------------------|------------------------|
| 0                               | 16                               | 31                     |
| Präambel (0 - 31)               |                                  |                        |
| Präambel (32 - 63)              |                                  |                        |
| Ziel-Adresse (0 - 31)           |                                  |                        |
| Ziel-Adresse (32 - 47)          |                                  | Quell-Adresse (0 - 15) |
| Quell-Adresse (16 - 47)         |                                  |                        |
| Paket-Typ                       | Version (0x40),<br>Kopflänge (5) | Diensttyp              |
| Paketlänge (in Bytes)           | Fragmentidentifikator            |                        |
| Fragmentadresse                 | Max. Zahl<br>an 'hops'           | Protokoll              |
| Prüfsumme                       | IP-Adresse der Quelle (0 - 15)   |                        |
| IP-Adresse der Quelle (16 - 31) | IP-Adresse des Ziels (0 - 15)    |                        |
| IP-Adresse des Ziels (16 - 31)  | Optionen                         |                        |
| Optionen                        | Füllbyte                         | Daten (Bytes 0 - 1)    |
| Daten (Bytes 2 - 5)             |                                  |                        |
| ▪ ▪ ▪                           |                                  |                        |
| CRC-Prüfsumme                   |                                  |                        |

- IP-Paket ist in Ethernet-Frame eingebettet

## 4 Adressierung und Routing

### ■ IP-Adressen

- 4 Bytes: **a.b.c.d** (unterteilt in Netzwerk-/Rechneradresse)
  - a < 128: **a** = Netzwerk, **b.c.d** = Rechner (Class A Net)  
große Organisationen
  - 127 < a < 192: **a.b** = Netzwerk, **c.d** = Rechner (Class B Net)  
Organisationen mittlerer Größe
  - a > 191: **a.b.c** = Netzwerk, **d** = Rechner (Class C Net)  
kleinere Organisationen
- Spezielle Adressen
  - 127.0.0.1                      Localhost
  - 255 als Rechneradresse      Broadcast
  - a >= 224                      Multicast
- Problem: max ca. 4 Mrd Adressen möglich
  - Adressraum wird allmählich knapp
  - IPv6 (128 Bit Adressen)

## 4 Adressierung und Routing

### ■ DNS – Domain Name System

- Abbildung von Rechnernamen auf IP-Adressen
- Erlaubt Benutzerfreundliche Namen:

– Beispiele: `www.config.de`

`fai40.informatik.uni-erlangen.de`

## 4 Adressierung und Routing

### ■ Routing:

- Finden eines Weges für ein IP-Paket von der Quelle zum Ziel
- Verwendung von Routing-Tabellen

- Abbildung von Ziel-Netzwerk auf nächsten Rechner
- Default Route

- 2 Arten:

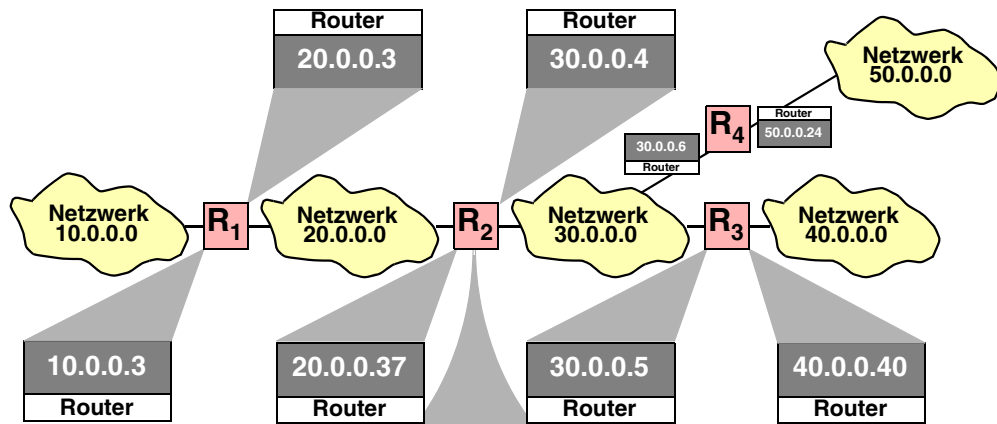
- direktes Routing

Quelle und Ziel auf demselben physikalischen Netz => direkte Zustellung des Pakets: MAC entspr. IP

- indirektes Routing

Ziel in anderem Netz => Paket wird zu einem Router geschickt; MAC vom Router, IP vom Zielrechner

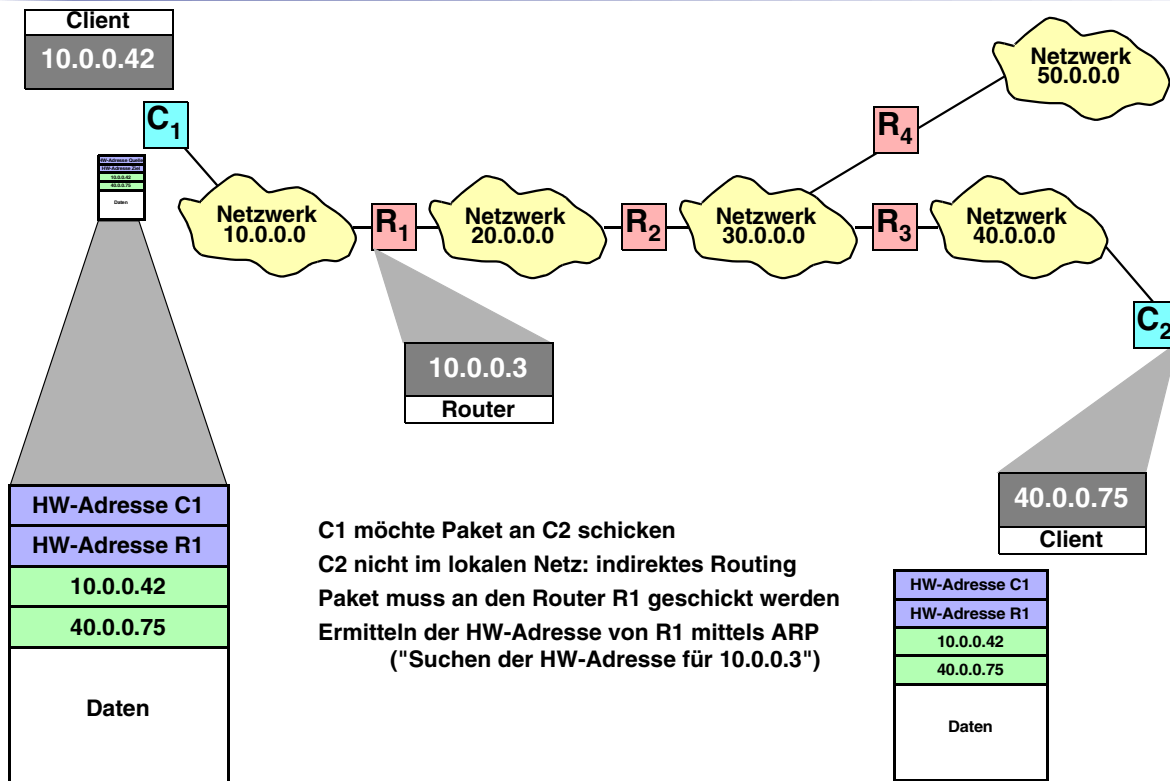
# 4 Adressierung und Routing



Router und Routing-Tabellen

| Routing-Tabelle Router R <sub>2</sub> |                |
|---------------------------------------|----------------|
| Ziel-Netzwerk                         | Route          |
| 10.0.0.0                              | 20.0.0.3       |
| 20.0.0.0                              | Deliver Direct |
| 30.0.0.0                              | Deliver Direct |
| 40.0.0.0                              | 30.0.0.5       |
| 50.0.0.0                              | 30.0.0.6       |

# 4 Adressierung und Routing



# 4 Adressierung und Routing

| Client    |
|-----------|
| 10.0.0.42 |

C<sub>1</sub>

Router 20.0.0.3

Netzwerk 10.0.0.0 — R<sub>1</sub> — Netzwerk 20.0.0.0 — R<sub>2</sub> — Netzwerk 30.0.0.0 — R<sub>3</sub> — Netzwerk 40.0.0.0 — R<sub>4</sub> — Netzwerk 50.0.0.0

Router 10.0.0.3      Router 20.0.0.37

Client 40.0.0.75

C<sub>2</sub>

| HW-Adresse R1 |
|---------------|
| HW-Adresse R2 |
| 10.0.0.42     |
| 40.0.0.75     |
| Daten         |

| Routing-Tabelle Router R <sub>1</sub> |                |
|---------------------------------------|----------------|
| Ziel-Netzwerk                         | Route          |
| 10.0.0.0                              | Deliver Direct |
| 20.0.0.0                              | Deliver Direct |
| 30.0.0.0                              | 20.0.0.37      |
| 40.0.0.0                              | 20.0.0.37      |
| 50.0.0.0                              | 20.0.0.37      |

R1 erhält Paket von C1  
C2 nicht im lokalen Netz  
Paket muss an Router R2 geschickt werden (siehe Routing-Tabelle von R1)  
Eintrag der entsprechenden HW-Adressen

Hier wäre ein Default-Route nach R2 sinnvoll!

# 4 Adressierung und Routing

| Client    |
|-----------|
| 10.0.0.42 |

C<sub>1</sub>

Router 20.0.0.3      Router 30.0.0.4

Netzwerk 10.0.0.0 — R<sub>1</sub> — Netzwerk 20.0.0.0 — R<sub>2</sub> — Netzwerk 30.0.0.0 — R<sub>3</sub> — Netzwerk 40.0.0.0 — R<sub>4</sub> — Netzwerk 50.0.0.0

Router 20.0.0.37      Router 30.0.0.5

Router 30.0.0.6

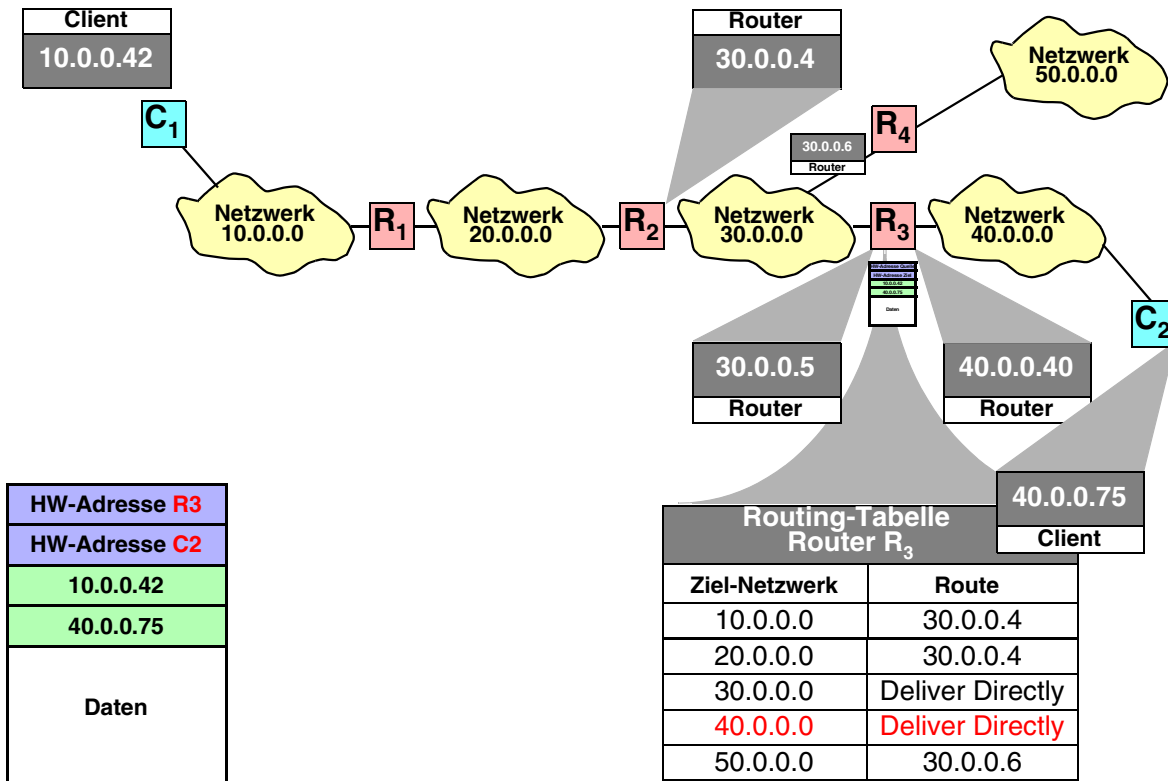
Client 40.0.0.75

C<sub>2</sub>

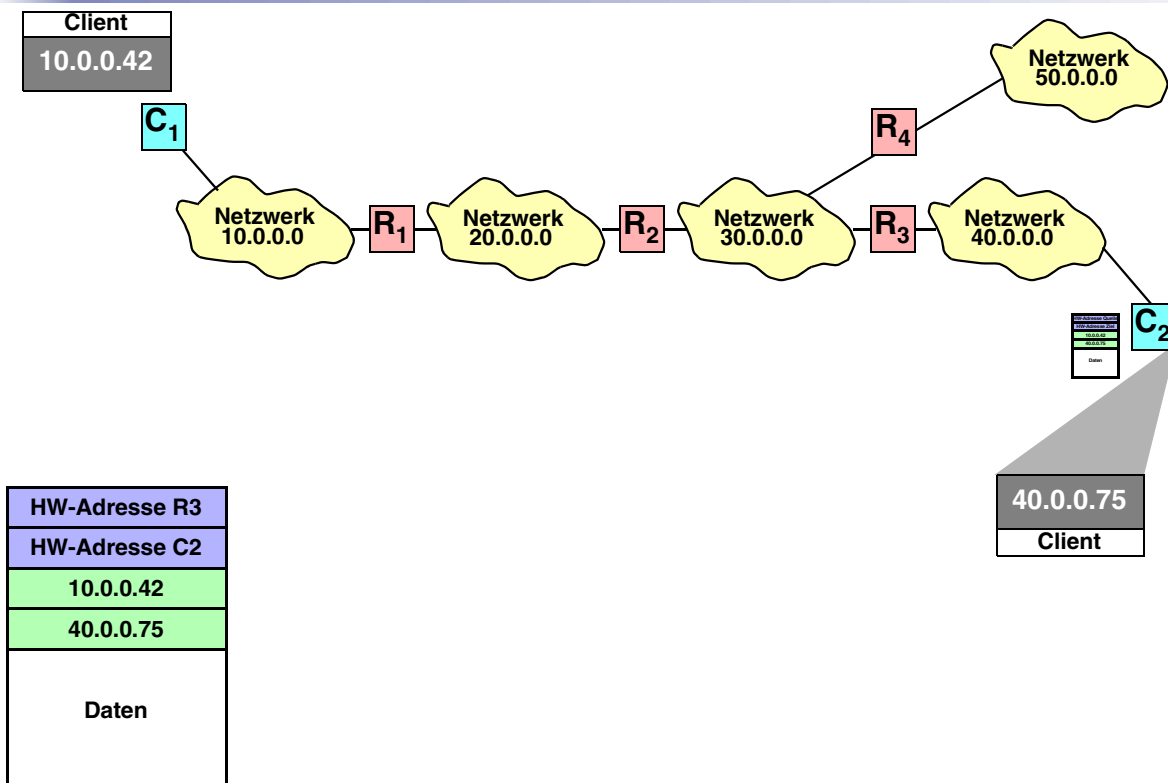
| HW-Adresse R2 |
|---------------|
| HW-Adresse R3 |
| 10.0.0.42     |
| 40.0.0.75     |
| Daten         |

| Routing-Tabelle Router R <sub>2</sub> |                |
|---------------------------------------|----------------|
| Ziel-Netzwerk                         | Route          |
| 10.0.0.0                              | 20.0.0.3       |
| 20.0.0.0                              | Deliver Direct |
| 30.0.0.0                              | Deliver Direct |
| 40.0.0.0                              | 30.0.0.5       |
| 50.0.0.0                              | 30.0.0.6       |

# 4 Adressierung und Routing



## 7.3.4.0 Adressierung und Routing



## 4 Adressierung und Routing

### Route ansehen (UNIX):

```

traceroute www4.informatik
traceroute to www4.informatik.uni-erlangen.de (131.188.34.200), 30 hops max, 40 byte packets
 1  reliant.gate.uni-erlangen.de (131.188.30.8)  0.904 ms  0.981 ms  0.568 ms
 2  faui40 (131.188.30.40)  0.911 ms *  0.551 ms

traceroute www.sun.com
traceroute to www.sun.com (64.124.140.181), 30 hops max, 40 byte packets
 1  reliant.gate.uni-erlangen.de (131.188.34.8)  0.718 ms  0.477 ms  0.44 ms
 2  suedstern.gate.uni-erlangen.de (131.188.21.65)  0.617 ms  0.487 ms  0.42 ms
 3  excelsior.gate.uni-erlangen.de (131.188.5.1)  3.296 ms  4.665 ms  3.265 ms
 4  ar-erlangen1.g-win.dfn.de (188.1.36.1)  0.273 ms  0.401 ms  9.783 ms
 5  cr-erlangen1.g-win.dfn.de (188.1.72.1)  0.423 ms  0.304 ms  0.326 ms
 6  cr-essen1.g-win.dfn.de (188.1.18.106)  41.563 ms  25.146 ms  24.231 ms
 7  cr-frankfurt1.g-win.dfn.de (188.1.18.89)  24.096 ms  24.031 ms  24.032 ms
 8  ir-frankfurt2.g-win.dfn.de (188.1.80.38)  41.244 ms  24.186 ms  24.298 ms
 9  ge9-0.pr1.fra1.de.mfnx.net (216.200.116.97)  24.076 ms  24.181 ms  24.063 ms
10  so-0-1-0.crl.fra1.de.mfnx.net (216.200.116.213)  24.18 ms  24.204 ms  24.209 ms
11  pos9-0.crl.cdg2.fr.mfnx.net (64.125.31.161)  33.873 ms  33.924 ms  33.845 ms
12  so-5-0-0.crl.lhr3.uk.mfnx.net (64.125.31.154)  43.539 ms  43.507 ms  43.452 ms
13  so-7-0-0.crl.dca2.us.mfnx.net (64.125.31.186)  115.417 ms  115.504 ms  115.309 ms
14  so-3-0-0.mpr3.sjc2.us.mfnx.net (208.184.233.133)  181.442 ms  181.473 ms  181.46 ms
15  so-0-0-0.mpr4.sjc2.us.mfnx.net (64.125.30.2)  181.498 ms  181.497 ms  181.504 ms
16  so-1-0-0.cr2.sjc3.us.mfnx.net (208.184.233.50)  181.604 ms  181.66 ms  181.672 ms
17  pos1-0.er2a.sjc3.us.mfnx.net (208.185.175.198)  181.648 ms  181.643 ms  181.567 ms

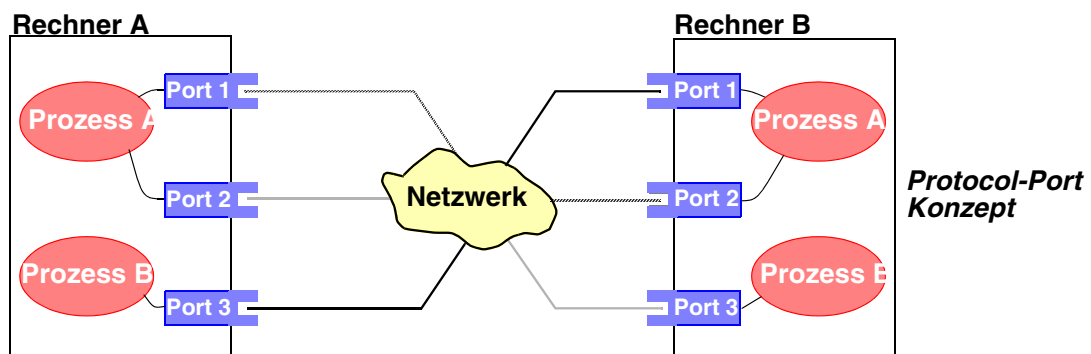
```

- 17 "Hops" bis zum Ziel (d.h. ein Paket geht durch 16 Router)!
- Routen können sich dynamisch ändern

## 4 Adressierung und Routing

### Adressierung von Diensten

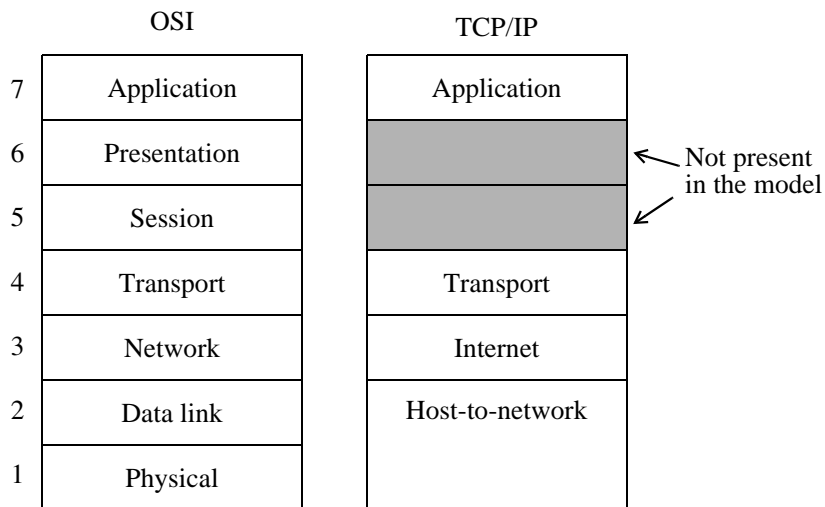
- IP: nur Adressierung von Rechnern
- Dienste / Services können über **Ports** angesprochen werden
  - Dienste verwenden normalerweise UDP oder TCP
  - UDP/TCP-Adresse = IP-Adresse + Portnummer des Dienstes



- Für viele Standard-Dienste sind Portnummern festgelegt (/etc/services)

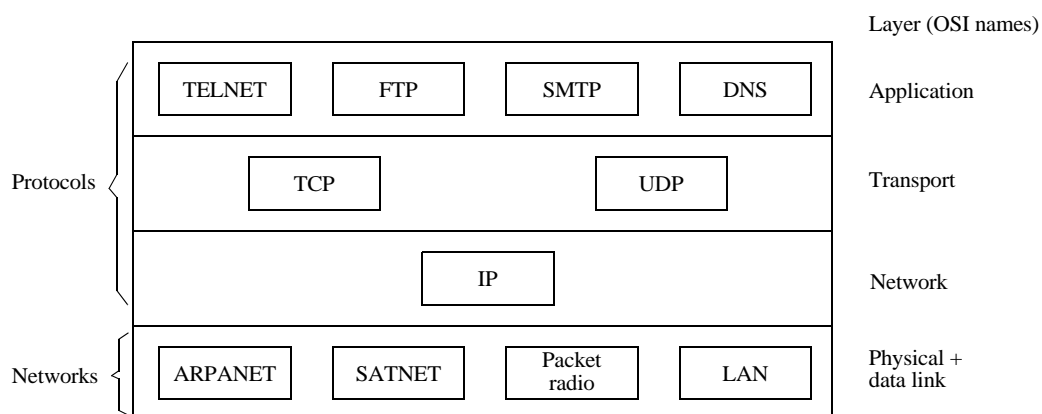
## 7.4 Protocol-Layers

### 1 OSI - TCP/IP - Reference-Modell



The TCP/IP reference model (nach Tanenbaum)

### 2 TCP/IP - Protocol - Stack



Protocols and network in the TCP/IP model initially (nach Tanenbaum)

## 7.5 Dienste / Höhere Protokolle etc.

### 1 Grundlegende Protokolle (IP basiert)

- **UDP** (User Datagram Protocol)
  - Paketorientiert / Verbindungslos
  - Keine Fehlerbehandlungsmaßnahmen
  - Geeignet für Broadcast / Multicast
  - Kommunikation zwischen Ports (Kommunikationsendpunkte)
  
- **TCP** (Transmission Control Protocol)
  - Stromorientiert / Verbindungsorientiert
  - Garantiert korrekte Datenübertragung
  - Kommunikation zwischen Ports (Sockets, Mailboxen, Prozesse)
  
- Diese Protokolle bilden die Grundlage fast aller Internet-Kommunikation

### 2 UDP-basiert Dienste

- **NTP** (Network Time Protocol)
  - Weltweite Synchronisation der Rechneruhren
    - LAN: im Bereich 1ms
    - WAN: im Bereich 10 bis 100ms
  - Updates ca. im 10-Minuten-Takt
  - Entfernung vom Zeit-Empfänger
    - Stratum-0: Empfänger selbst
    - Stratum-1: Rechner, der direkt mit Empfänger verbunden ist
    - Stratum-2: mit Stratum-1 Rechner verbunden, etc.
  
- **TFTP** (Trivial File Transfer Protocol; benutzt z.B. zum Booten)



### 3 TCP-basierte Dienste

- **SMTP** (Simple Mail Transfer Protocol)
  - Transportieren von E-Mails
- **FTP** (File Transfer Protocol)
  - Austauschen von Dateien zwischen Rechnern
- **HTTP** (Hypertext Transfer Protocol)
  - Übertragen von Web-Inhalten
- **NFS** (Network File System)
  - Transparenter Dateizugriff auf andere Rechner
  - Verwendung des RPC
- **telnet** (Terminal Emulation for TCP/IP Networks)
  - Einloggen in andere Rechner
  - Direkte Verbindung zu Diensten

### 3 TCP-basierte Dienste

- **rlogin, rsh, rcp** ( r = remote...)
  - Einloggen in andere Rechner
  - Kopieren von Daten von/zu anderen Rechnern
- **SSH** (Secure Shell)
  - Ersatz für rlogin, rsh, rcp
  - Starke Authentisierung
  - Verschlüsselung aller übertragenen Daten
  - Verschiedenen Algorithmen (IDEA, DES, RSA) wählbar
  - Verfügbar für Windows, Unix, Macintosh, and OS/2
- ...

## 7.6 WWW

---

- Ursprünglich statisch
  - **HTML** (Hypertext Markup Language)
    - = Sprache zum Beschreiben von Hypertexten inklusive ihres Layouts; Grundlage sind Tags: Textauszeichnungen
  - **HTTP** (Hypertext Transfer Protocol)
    - = über TCP/IP
  
- Client-Side Scripting:
  - JavaScript
  - Applets

## 7.7 WWW

---

- Server-Side Scripting:
  - **SHTML** (Server-Side Include)
  - **CGI** (Common Gateway Interface)
    - Oft zum Bearbeiten von Forms
    - Programmierung in C, C++, Perl, Java, ...

## 7.7 WWW

---

### ■ Server-Side Scripting (cont.):

- **PHP** (Hypertext Preprocessor)
  - Mischung aus Java, C, Perl + Erweiterungen
  - Einfaches Erstellen dynamisch erzeugter Web-Inhalte
- **ASP** (Active Server Page)
  - Microsoft-Spezifisch
  - Programmierung in VBScript or JScript
  - Einbettung direkt in HTML-Code
- **Servlets, JSP** (Java Server Page)
  - Programmierung in Java
  - Einbettung direkt in HTML-Code