

```
-code  
"100"  
"101"  
"200"  
"201"  
"202"  
"203"  
"204"  
"205"  
"206"  
"300"  
"301"  
"302"  
"303"  
"304"  
"305"  
"307"  
"400"  
"401"  
"402"  
"403"  
"404"  
"405"
```

[illegible]

```

1: Switch
2: OK
1: Created
2: Accepted
3: Non-Content
4: No Content
2.5: Reset Content
2.6: Partial Content
2.7: Multi-Status
3.1: Moved Permanently
3.2: Found
3.3: See Other
3.4: Not Modified
3.5: Use Proxy
0.4.1: BadRequest
0.4.2: Unauthorized
10.4.1: Forbidden
10.4.2: Not Found
10.4.3: Method Not Allowed
10.4.4: Not Acceptable
10.4.5: Not Found
10.4.6: Not Found
10.4.7: Not Found
10.4.8: Not Found
10.4.9: Not Found
10.4.10: Not Found
10.4.11: Not Found
10.4.12: Not Found
10.4.13: Not Found
10.4.14: Not Found
10.4.15: Not Found
10.4.16: Not Found
10.4.17: Not Found
10.4.18: Not Found
10.4.19: Not Found
10.4.20: Not Found
10.4.21: Not Found
10.4.22: Not Found
10.4.23: Not Found
10.4.24: Not Found
10.4.25: Not Found
10.4.26: Not Found
10.4.27: Not Found
10.4.28: Not Found
10.4.29: Not Found
10.4.30: Not Found
10.4.31: Not Found
10.4.32: Not Found
10.4.33: Not Found
10.4.34: Not Found
10.4.35: Not Found
10.4.36: Not Found
10.4.37: Not Found
10.4.38: Not Found
10.4.39: Not Found
10.4.40: Not Found
10.4.41: Not Found
10.4.42: Not Found
10.4.43: Not Found
10.4.44: Not Found
10.4.45: Not Found
10.4.46: Not Found
10.4.47: Not Found
10.4.48: Not Found
10.4.49: Not Found
10.4.50: Not Found
10.4.51: Not Found
10.4.52: Not Found
10.4.53: Not Found
10.4.54: Not Found
10.4.55: Not Found
10.4.56: Not Found
10.4.57: Not Found
10.4.58: Not Found
10.4.59: Not Found
10.4.60: Not Found
10.4.61: Not Found
10.4.62: Not Found
10.4.63: Not Found
10.4.64: Not Found
10.4.65: Not Found
10.4.66: Not Found
10.4.67: Not Found
10.4.68: Not Found
10.4.69: Not Found
10.4.70: Not Found
10.4.71: Not Found
10.4.72: Not Found
10.4.73: Not Found
10.4.74: Not Found
10.4.75: Not Found
10.4.76: Not Found
10.4.77: Not Found
10.4.78: Not Found
10.4.79: Not Found
10.4.80: Not Found
10.4.81: Not Found
10.4.82: Not Found
10.4.83: Not Found
10.4.84: Not Found
10.4.85: Not Found
10.4.86: Not Found
10.4.87: Not Found
10.4.88: Not Found
10.4.89: Not Found
10.4.90: Not Found
10.4.91: Not Found
10.4.92: Not Found
10.4.93: Not Found
10.4.94: Not Found
10.4.95: Not Found
10.4.96: Not Found
10.4.97: Not Found
10.4.98: Not Found
10.4.99: Not Found
10.5: Not Found
10.6: Not Found
10.7: Not Found
10.8: Not Found
10.9: Not Found
10.10: Not Found
10.11: Not Found
10.12: Not Found
10.13: Not Found
10.14: Not Found
10.15: Not Found
10.16: Not Found
10.17: Not Found
10.18: Not Found
10.19: Not Found
10.20: Not Found
10.21: Not Found
10.22: Not Found
10.23: Not Found
10.24: Not Found
10.25: Not Found
10.26: Not Found
10.27: Not Found
10.28: Not Found
10.29: Not Found
10.30: Not Found
10.31: Not Found
10.32: Not Found
10.33: Not Found
10.34: Not Found
10.35: Not Found
10.36: Not Found
10.37: Not Found
10.38: Not Found
10.39: Not Found
10.40: Not Found
10.41: Not Found
10.42: Not Found
10.43: Not Found
10.44: Not Found
10.45: Not Found
10.46: Not Found
10.47: Not Found
10.48: Not Found
10.49: Not Found
10.50: Not Found
10.51: Not Found
10.52: Not Found
10.53: Not Found
10.54: Not Found
10.55: Not Found
10.56: Not Found
10.57: Not Found
10.58: Not Found
10.59: Not Found
10.60: Not Found
10.61: Not Found
10.62: Not Found
10.63: Not Found
10.64: Not Found
10.65: Not Found
10.66: Not Found
10.67: Not Found
10.68: Not Found
10.69: Not Found
10.70: Not Found
10.71: Not Found
10.72: Not Found
10.73: Not Found
10.74: Not Found
10.75: Not Found
10.76: Not Found
10.77: Not Found
10.78: Not Found
10.79: Not Found
10.80: Not Found
10.81: Not Found
10.82: Not Found
10.83: Not Found
10.84: Not Found
10.85: Not Found
10.86: Not Found
10.87: Not Found
10.88: Not Found
10.89: Not Found
10.90: Not Found
10.91: Not Found
10.92: Not Found
10.93: Not Found
10.94: Not Found
10.95: Not Found
10.96: Not Found
10.97: Not Found
10.98: Not Found
10.99: Not Found
11: Not Found
12: Not Found
13: Not Found
14: Not Found
15: Not Found
16: Not Found
17: Not Found
18: Not Found
19: Not Found
20: Not Found
21: Not Found
22: Not Found
23: Not Found
24: Not Found
25: Not Found
26: Not Found
27: Not Found
28: Not Found
29: Not Found
30: Not Found
31: Not Found
32: Not Found
33: Not Found
34: Not Found
35: Not Found
36: Not Found
37: Not Found
38: Not Found
39: Not Found
40: Not Found
41: Not Found
42: Not Found
43: Not Found
44: Not Found
45: Not Found
46: Not Found
47: Not Found
48: Not Found
49: Not Found
50: Not Found
51: Not Found
52: Not Found
53: Not Found
54: Not Found
55: Not Found
56: Not Found
57: Not Found
58: Not Found
59: Not Found
60: Not Found
61: Not Found
62: Not Found
63: Not Found
64: Not Found
65: Not Found
66: Not Found
67: Not Found
68: Not Found
69: Not Found
70: Not Found
71: Not Found
72: Not Found
73: Not Found
74: Not Found
75: Not Found
76: Not Found
77: Not Found
78: Not Found
79: Not Found
80: Not Found
81: Not Found
82: Not Found
83: Not Found
84: Not Found
85: Not Found
86: Not Found
87: Not Found
88: Not Found
89: Not Found
90: Not Found
91: Not Found
92: Not Found
93: Not Found
94: Not Found
95: Not Found
96: Not Found
97: Not Found
98: Not Found
99: Not Found
100: Not Found

```

```

* The second const triggers an assembler bug on UTS 2.1.
* Another workaround is to move some code out of this file
* but this is easier. Dave Dykstra, 3/31/99
*/
static const char * status_lines[RESPONSE_CODES] =
else
static const char * const status_lines[RESPONSE_CODES] =
#endif

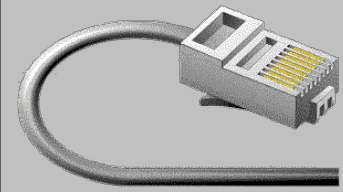
"100 Continue",
"101 Switching Protocols",
"102 Processing",
line LEVEL_200
"200 OK",
"201 Created",
"202 Accepted",
"203 Non-Authoritative Information",
"204 No Content",
"205 Reset Content",
"206 Partial Content",
"207 Multi-Status",
LEVEL_300
"300 Multiple Choices",
"301 Moved Permanently",
"302 Found",
"303 See Other",
"304 Not Modified",
"305 Use Proxy",
"306 (Unused)",
"307 Temporary Redirect",
"400 Bad Request",
"401 Unauthorized",
"402 Payment Required",
"403 Forbidden",
"404 Not Found",
"405 Method Not Allowed",
"406 Not Acceptable",
"407 Proxy Authentication Required",

```

Protokolle Dienste

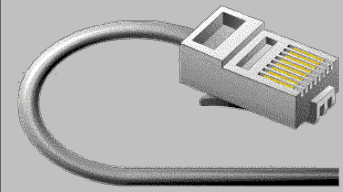
Kapitel 2

© 2006 ... 2011, u. heuer



Ziele

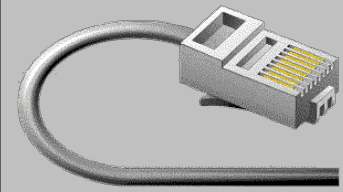
- Einsatzbereich von Netzwerken kennen
- Unterschiede von Protokollen und Diensten kennen
- Einteilung von Protokollen kennen
- Einteilung von Diensten kennen



Bereich von Netzwerken

Die Ausdehnung von Netzwerken kann in 4 Bereiche unterteilt werden:

- **WAN**
- **MAN**
- **LAN**
- **PAN**



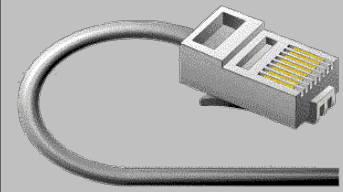
Begriffe: WAN ...

WAN: Wide Area Network

Ausdehnung: Weltweit

Technologie: Serielle Links, ATM, Ethernet

Medium: Glas, Funk



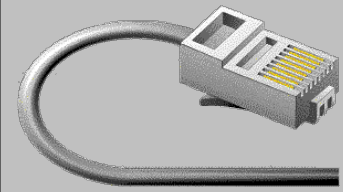
Begriffe: MAN ...

MAN: Metrapolitan Area Network

Ausdehnung: Campus, City, Region

Technologie: Ethernet, ATM, Richtfunk

Medium: Kupfer, Glas, Funk, Laser



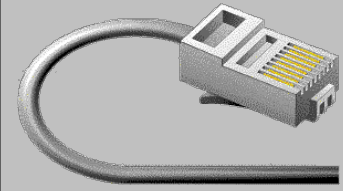
Begriffe: LAN ...

LAN: Local Area Network

Ausdehnung: Raum, Gebäude

Technologie: Ethernet, Wireless

Medium: Glas, Kupfer, Funk, Infrarot



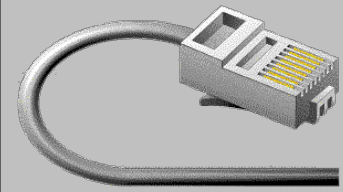
Begriffe: PAN ...

PAN: Personal Area Network

Ausdehnung: Rund um eine Person

Technologie: BlueTooth, RFID, USB, Firewire

Medium: Kupfer, Funk, Infrarot



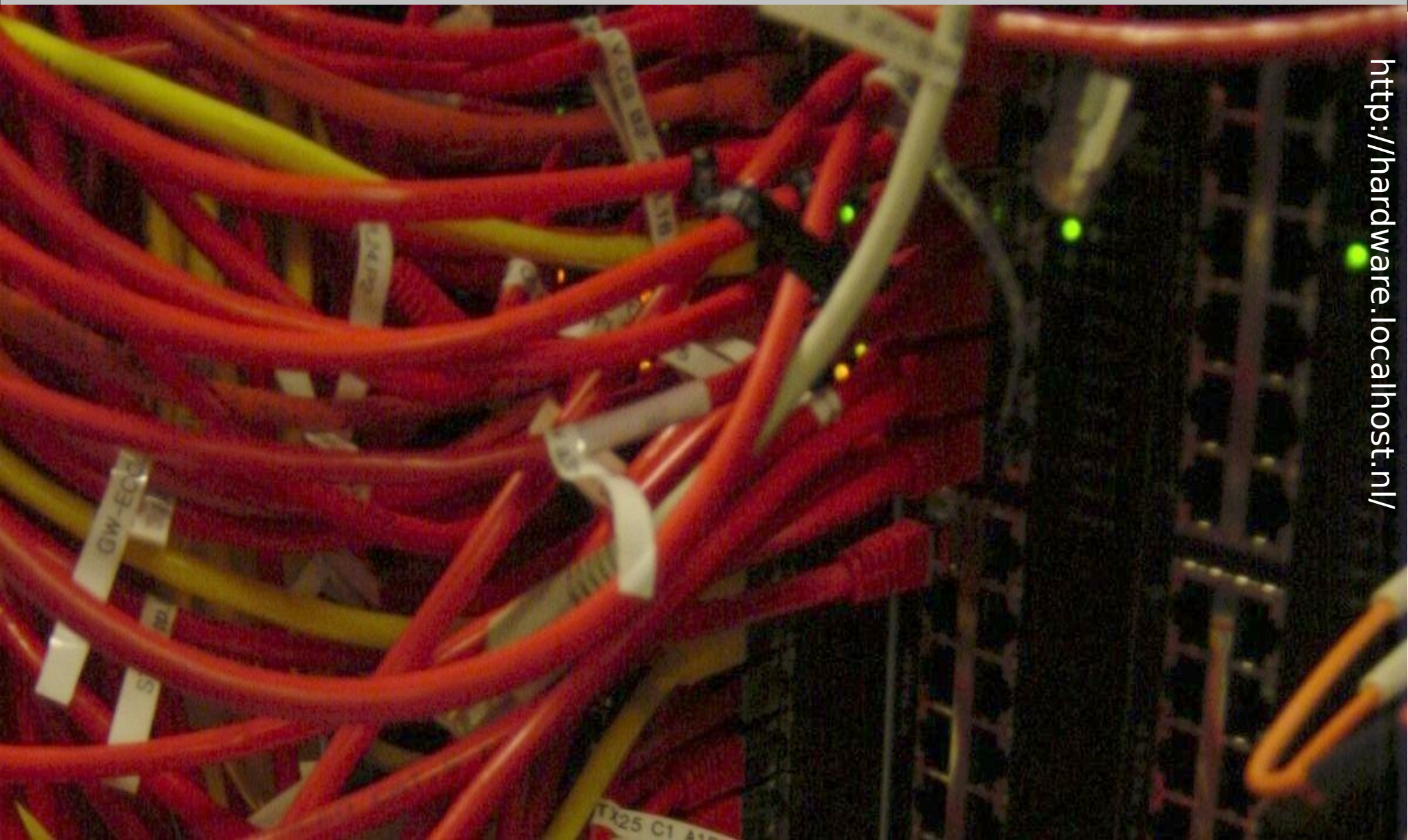
Begriffe: [WMLP]AN ...

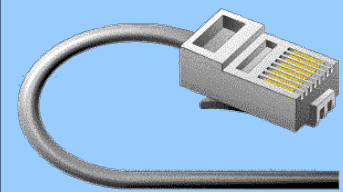
Die Abgrenzung zwischen den einzelnen Bereichen ist nicht immer klar gegeben.

Beispielsweise kann Bluetooth Class 1 bis zu 100m überbrücken - was deutlich mehr ist, als was die Definition 'rund um eine Person' beschreibt.



Fragen ?





Aufgabe

Ordnen sie die folgenden Technologien dem Einsatzgebiet - entspreche den Definition - zu. Verwenden sie Wikipedia, Google, ... um ihnen unbekannte Abkürzungen nach zuschlagen

VDSL

ISDN

GSM

DECT

SDH (Synchronous Digital Hierarchy)

Ethernet

Metro-Ethernet

Richtfunk

POTS

ADSL Annex B





Protokolle und Dienste

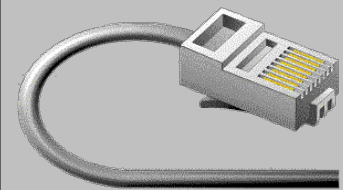
Protokolle:

Definieren wie die Kommunikation zwischen zwei Geräten erfolgt.

Dienste:

Dienste implementieren die Protokolle.

Dies kann in Software oder auch in Hardware erfolgen!



Protokoll eines Mailservers

Der Client öffnet die Verbindung zum Server:

> **Client**
< **Server**

* Connecting to SMTP server: mail.heuer.org ...

[16:38:38] SMTP< 220 largo.maillink.ch ESMTP

[16:38:38] ESMTP> EHLO guybrush.maillink.ch

[16:38:38] ESMTP< 250-largo.maillink.ch

...

[16:38:38] ESMTP< 250 AUTH LOGIN PLAIN

Begrüssung,
Der Server
sagt, was er
kann

[16:38:38] ESMTP> STARTTLS

[16:38:38] ESMTP< 220 ready for tls

Starten der
Verschlüsselung

[16:38:38] ESMTP> EHLO guybrush.maillink.ch

...

[16:38:38] ESMTP< 250 AUTH LOGIN PLAIN

[16:38:38] ESMTP> AUTH LOGIN

[16:38:38] ESMTP< 334 VXNlcm5hbWU6

Begrüssung (2)

Authentication

...

[16:38:38] ESMTP< 235 ok, go ahead (#2.0.0)



Protokoll eines Mailservers

> Client
< Server

[16:38:38] ESMTP> MAIL FROM:<ueli@heuer.org>

[16:38:38] SMTP< 250 ok

[16:38:38] SMTP> RCPT TO:<ueli@heuer.org>

[16:38:38] SMTP< 250 ok

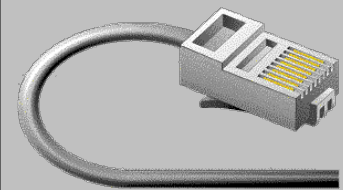
[16:38:38] SMTP> DATA

[16:38:38] SMTP< 354 go ahead

[16:38:38] SMTP> . (EOM)

[16:38:38] SMTP< 250 ok 1298475518 qp 10413

* Mail sent successfully.



Protokolle

Protokolle:

Beschreiben wie sich die Partner unterhalten.

Protokolle sind auf allen Schichten des ISO/OSI Modells zu finden.



Protokolle

- Der Webbrowser und Webserver müssen sich gegenseitig verstehen. (Layer 7)
Beide 'sprechen' das Protokoll http
- Der Rechner wo der Webbrowser installiert ist, muss sich mit dem Rechner wo der Webserver läuft "unterhalten" können. (Layer 3)
Beide 'sprechen' das Protokoll IP



Protokolle

Der Benutzer will beispielsweise die Webseite **<http://www.heuer.org/>** anschauen.

Der Browser übersetzt diese Anfrage und sendet diese an den Web-Server

```
GET / HTTP/1.1
```

```
Host: www.heuer.org
```

```
User-Agent: Mozilla/5.0 (X11; U; Linux ...
```

```
Accept: text/xml,application/xml ...
```

```
Accept-Language: en-us,en;q=0.5
```

```
Accept-Encoding: gzip,deflate
```

```
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*...
```

...



Protokolle

Der Rechner, wo der Browser installiert ist, sendet diese Anfrage via TCP/IP an den Webserver.

In der Regel sind noch weitere Protokolle notwendig: Bevor dir Anfrage gesendet werden kann, muss der Rechner noch die IP-Adresse des Webservers kennen.

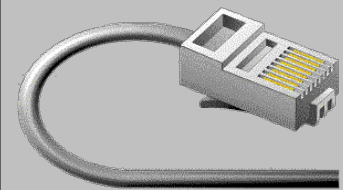
Dazu verwendet er den Domain-Name Server, den er via IP/UDP anspricht.



Protokolle

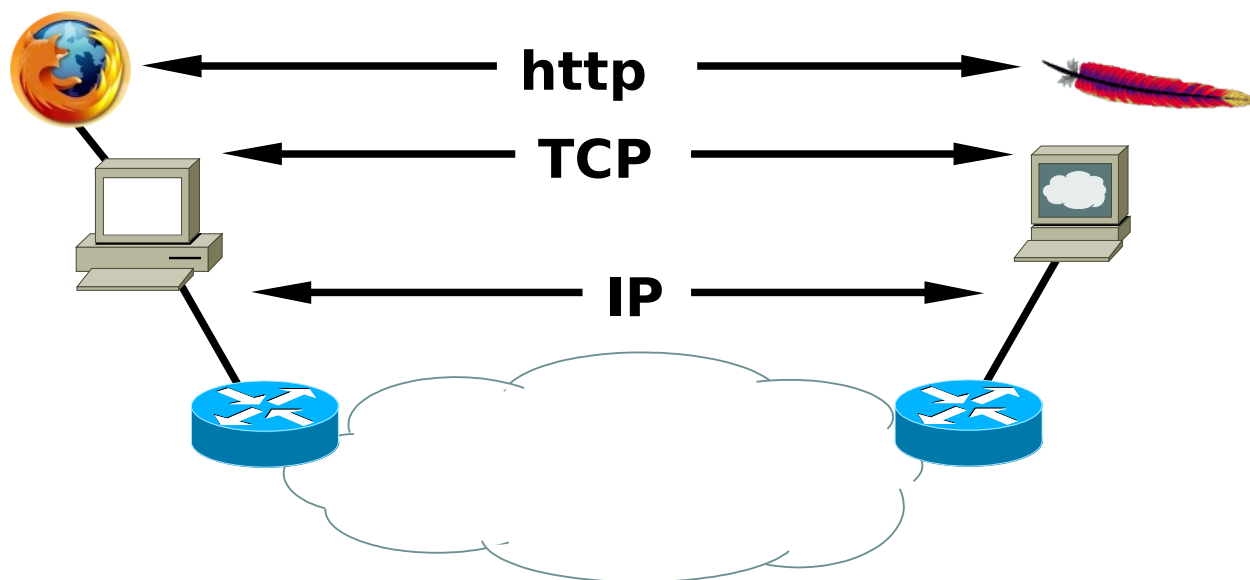
Der Webserver antwortet dem Browser:

```
HTTP/1.x 200 OK
Date: Wed, 04 Oct 2006 17:44:19 GMT
Server: Apache/2.0.59 (Unix)
Content-Location: index.html
Last-Modified: Mon, 05 Dec 2005 23:29:04 GMT
Accept-Ranges: bytes
Content-Length: 2761
Keep-Alive: timeout=15, max=100
Connection: Keep-Alive
Content-Type: text/html; charset=ISO-8859-1
```



Protokolle

Um erfolgreich zu kommunizieren sind verschieden Protokolle beteiligt.



Jedes dieser Protokoll deckt einen spezifischen Teil der Kommunikation ab.

► Es sind nicht alle notwendigen Protokolle aufgeführt.

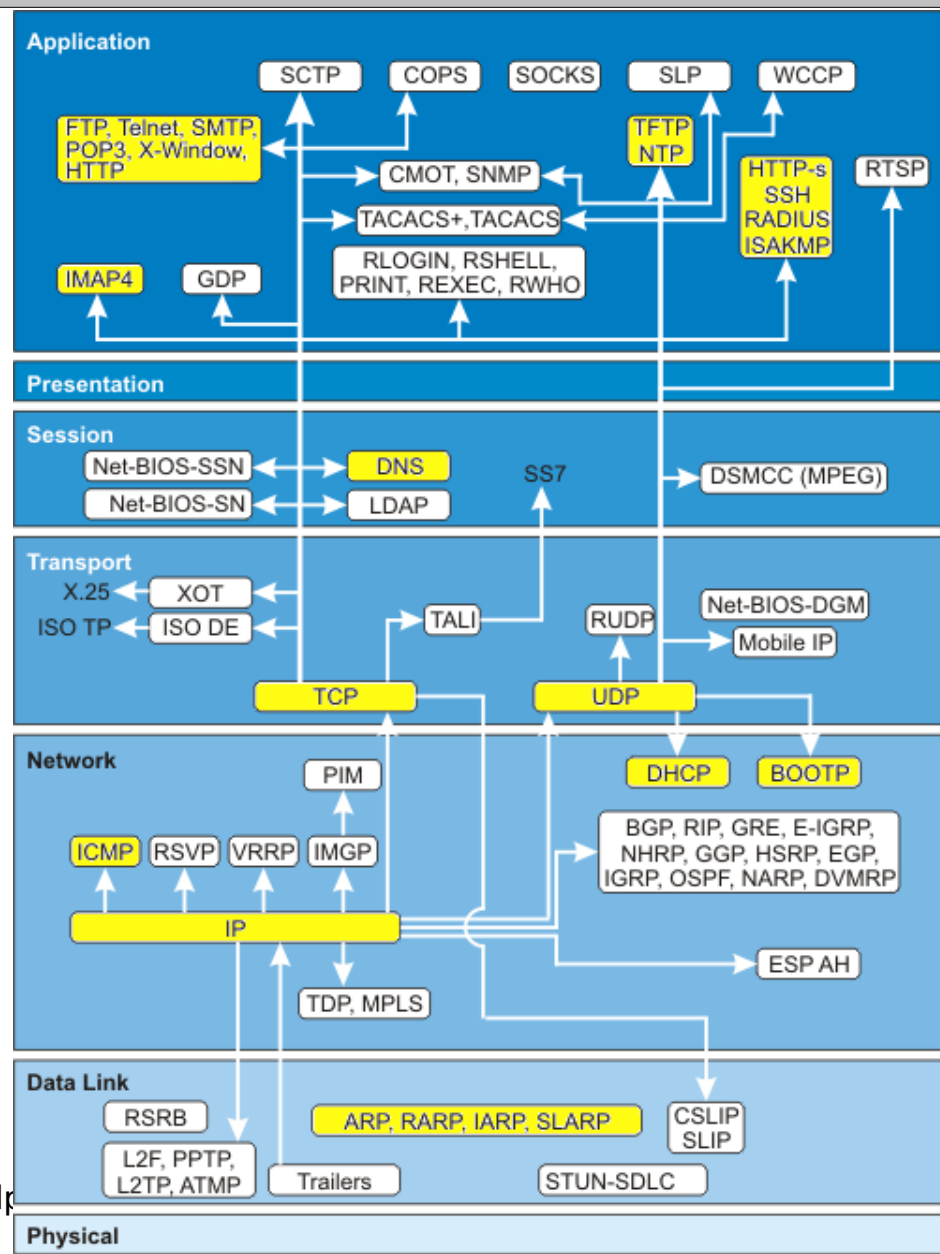


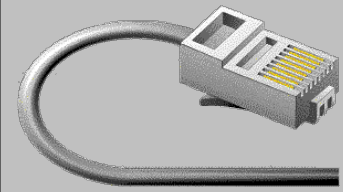
Protokolle in der TCP/IP Welt

Es sind sehr viele Protokolle definiert.

Jeder, der sich im Internet bewegt, verwendet diese, ohne dass er es bewusst bemerkt.

<http://www.protocols.com/pbook/tcpip1.htm>

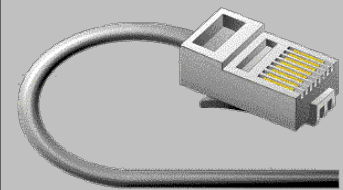




Protokolle Einteilung

Protokolle können auf verschiedene Arten eingeteilt werden.

- vom OSI-Layer (mehr im nächsten Kapitel)
- vom Einsatzgebiet WAN,MAN,LAN,PAN, ...
- der Aufgabe Bridge, Routing, PPP, Tunnel, ...



Beispiele von Protokollen

Physische Datenübertragung

10Gigabit-, Gigabit-, Fast-Ethernet, ADSL
Annex A, V.90, RS232, ...

Routing Protokolle

RIPv3, OSPF, EIGRP, ISIS, BGP, ...

Tunnel Protokolle

QinQ, L2TP, L2TPv3, IPinIP, GRE, ...

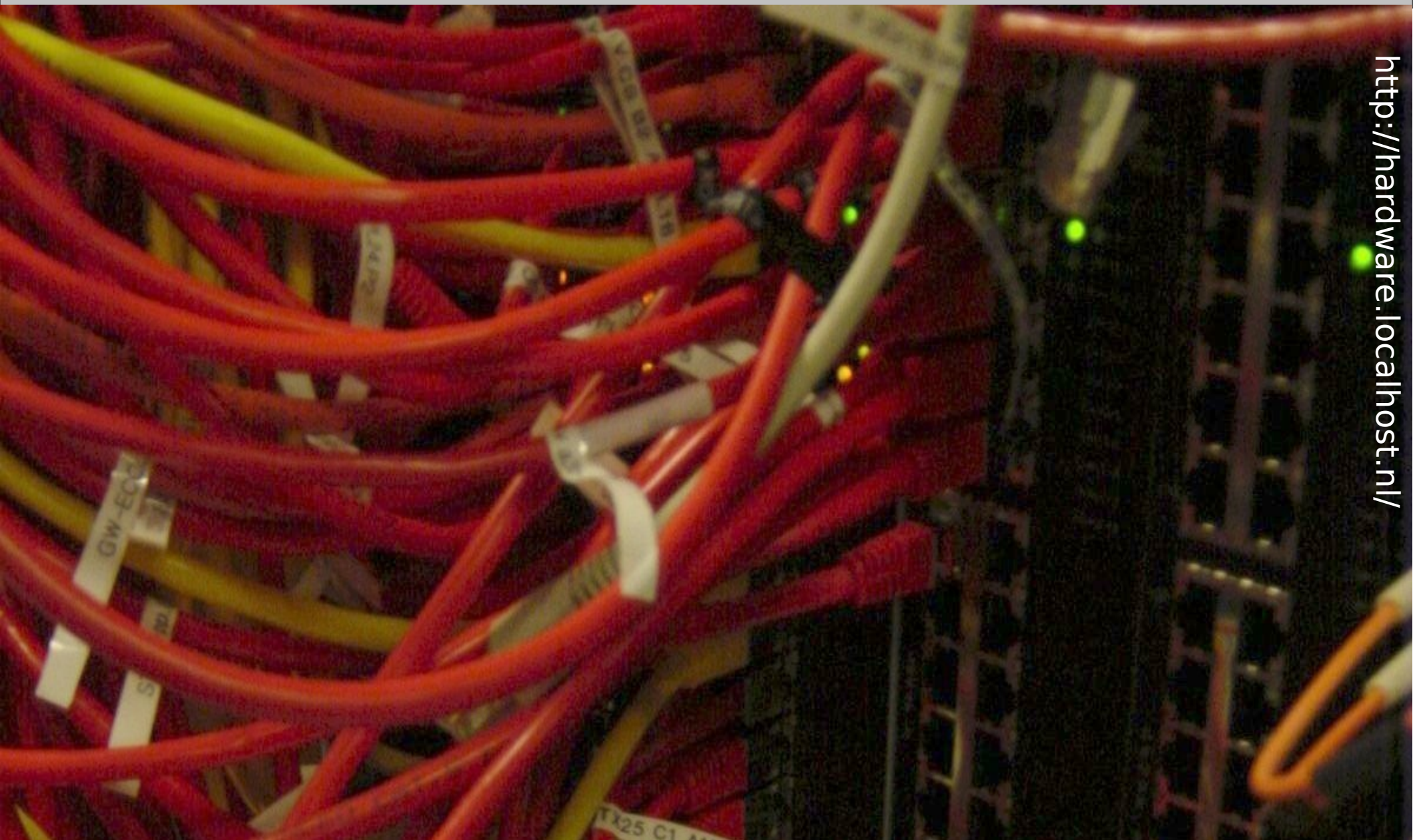
Mail Protokolle

SMTP, POP, IMAP, ...

Diese Aufzählung
ist nicht abschlies-
send!



Fragen ?





Protokolle ↔ Dienste

- **Protokolle** beschreiben die Art und Weise wie zwei Partner mit einander kommunizieren.
- **Dienste** implementieren ein Protokoll, so wie es in der Protokoll-Definition definiert ist.

Dienste implementieren je nach Anwendung nur die Client-, Server- oder auch Client- und Server-Seite des Protokolls.

Dienste, die die Server-Seite eines Protokolls implementieren werden oft als **Daemon** bezeichnet.

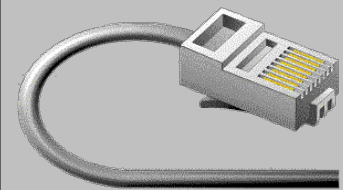


Dienste

Beispielsweise ermöglicht der Webserver **Apache**, dass Webseiten mit dem **Firefox** angesehen werden können.

Apache implementiert den Server-, **Firefox** den Client-Teil vom Protokoll HTTP.

Einzelne Programme können auch weitere Protokolle implementieren. (Beispielsweise kann der Firefox auch FTP-Client sein)



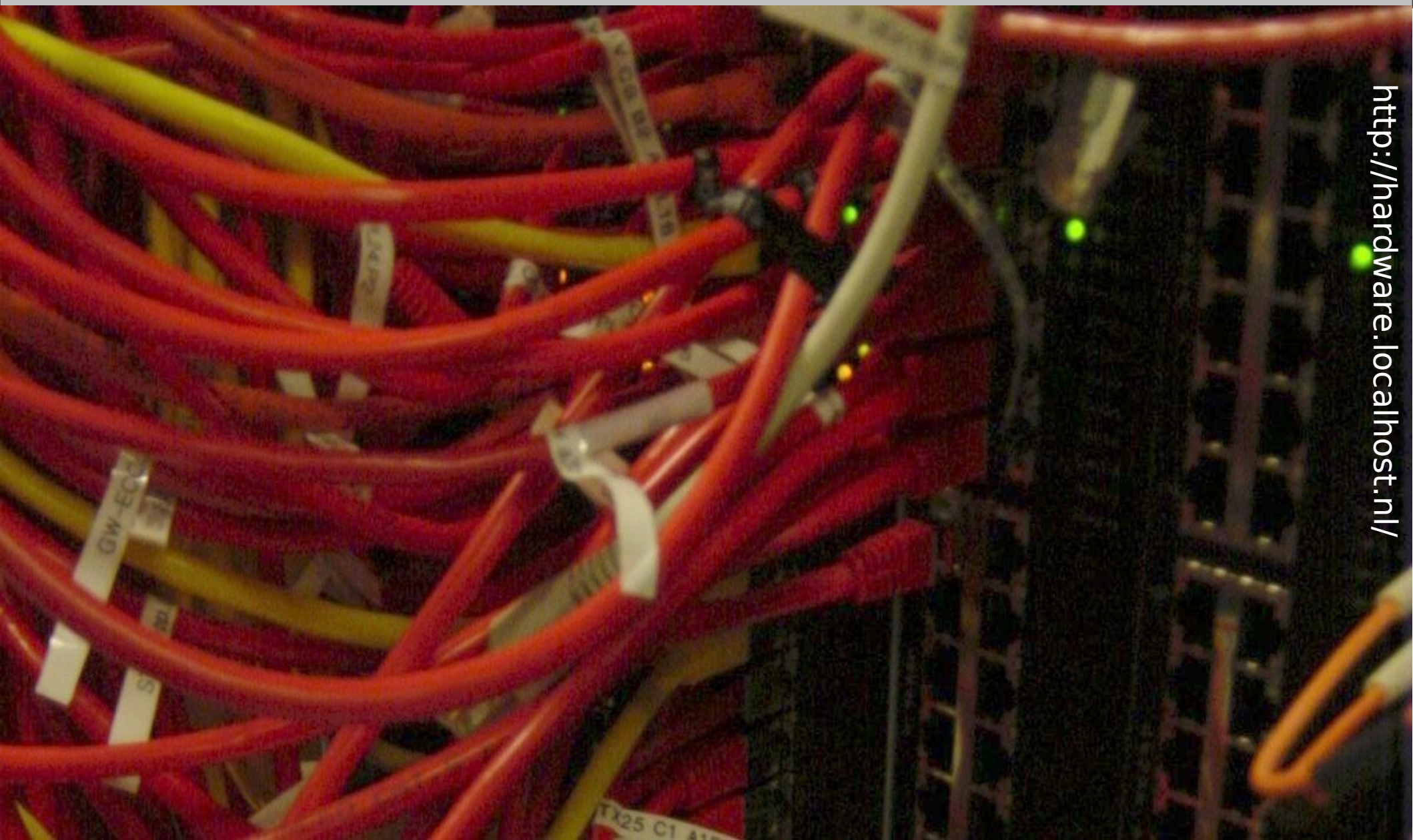
Dienste

Dienste können - gleich wie Protokolle - in verschiedene Arten unterteilt werden:

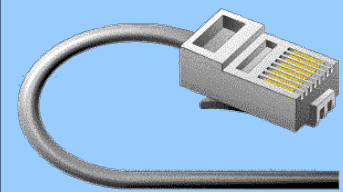
- **Namensauflösende** Dienste
(DNS, LDAP, WINS, ...)
- **Adressenauflösende** Dienste
(ARP, BOOTP/DHCP, Portmapper, ...)
- **Server** Dienste
(Samba, NFS, FTP, HTTP, NTP, ...)



Fragen ?



<http://hardware.localhost.nl/>



Aufgabe

- 2) Suchen sie auf ihrem PC nach Diensten.
- 3) Versuchen sie herauszufinden welche Protokolle die gefunden Dienste implementieren.
- 4) Versuchen sie die Protokolle der beiden Programme herauszufinden.
 - Web-Browser
 - Email-Client

