

Brandenburgische Technische Universität Cottbus

Lehrstuhl Rechnernetze und
Kommunikationssysteme

b-tu

Rechnernetze

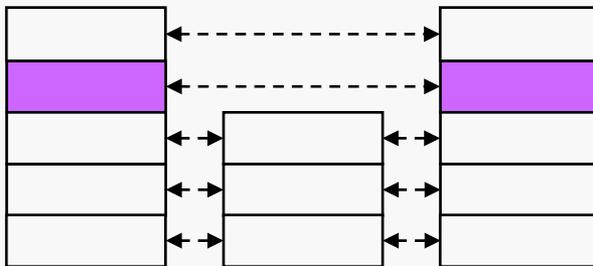
Eine (kurze) Einführung

Cluj, Wintersemester 2019/20

Prof. Dr.-Ing. habil. Hartmut König

III.6

Transportschicht (*Transport Layer*)



Stallings 20
Tanenbaum / Wetherall 6

Transportschicht

Die Transportschicht stellt eine Ende-zu-Ende-Beziehung zwischen dem Quell- und Zielknoten – den Endsystemen – her. Ab der Schicht 4 aufwärts gibt es nur noch eine Ende-zu-Ende Kommunikation. Die Transportschnittstelle verdeckt, die konkrete Netztopologie.

Wichtigste Protokolle

● **TCP** (*Transmission Control Protocol*)

- zuverlässiger, verbindungsorientierter Datenübertragungsdienst
 - ↪ Duplex-Übertragung
- beschleunigter Übertragungsdienst
 - ↪ z.B. für Abbruch von Prozessen auf Partner-Rechner

● **UDP** (*User Datagram Protocol*)

- unzuverlässiger Datenübertragungsdienst

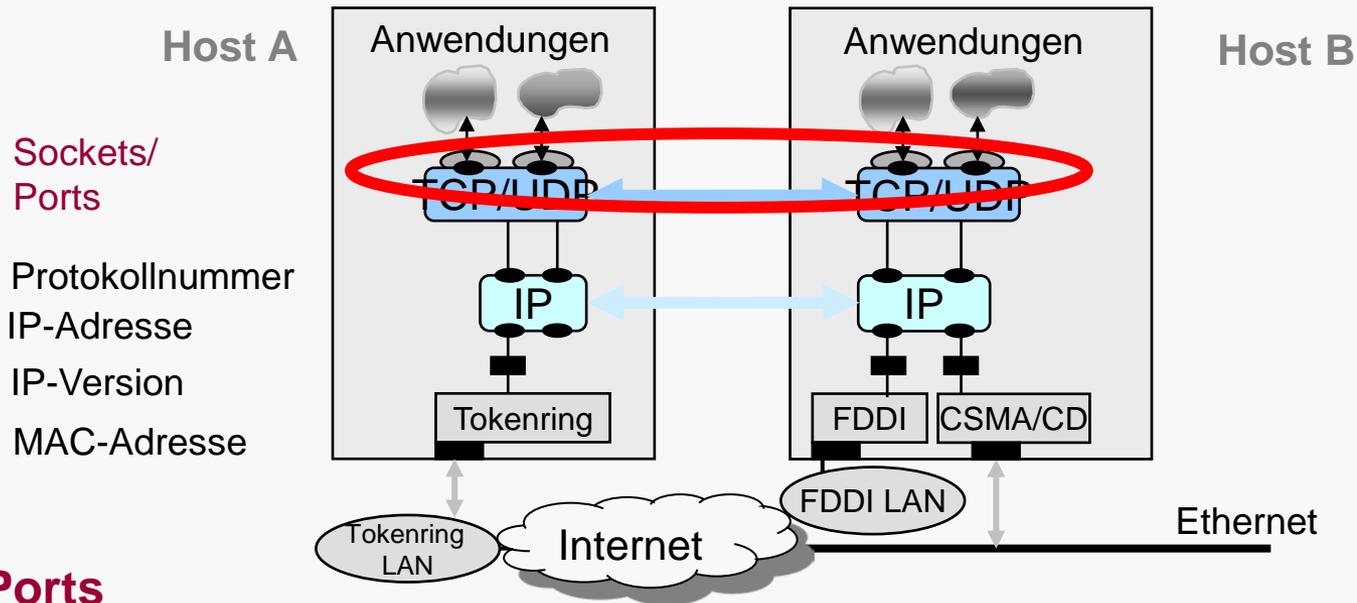
 **Hier nicht behandelt !!!**



Einschub: Sockets und Ports



TCP/IP-Dienstzugangspunkte (1) ¹



Ports

- Dienstzugangspunkte der Anwendung(sdienste)
 - ↳ entspricht den den TSAP im OSI/RM
- 16 Bit-Adresse
- *well-known*/ephemeral Ports
- Zuweisung *well-known* Ports durch ICANN (vorher IANA)

1) nach Stainov, R.: IPng Das Internet-Protokoll der nächsten Generation. Thompson Publishing, 1997

Ports

● Well-known Ports

- Anwendungen fest zugewiesene Ports

- **Beispiele:**

20	-	FTP (data)
21	-	FTP (control)
23	-	TELNET
24	-	SMTP
53	-	DNS
80	-	WWW
102	-	ISO TSAP
103	-	X.400
137 - 39	-	NETBIOS

● Ephemeral (kurzlebige) Ports

- dynamische beschränkte Zuweisung an Anwendung (Clients)
- >1024



TCP/IP-Dienstzugangspunkte (2)

● Sockets

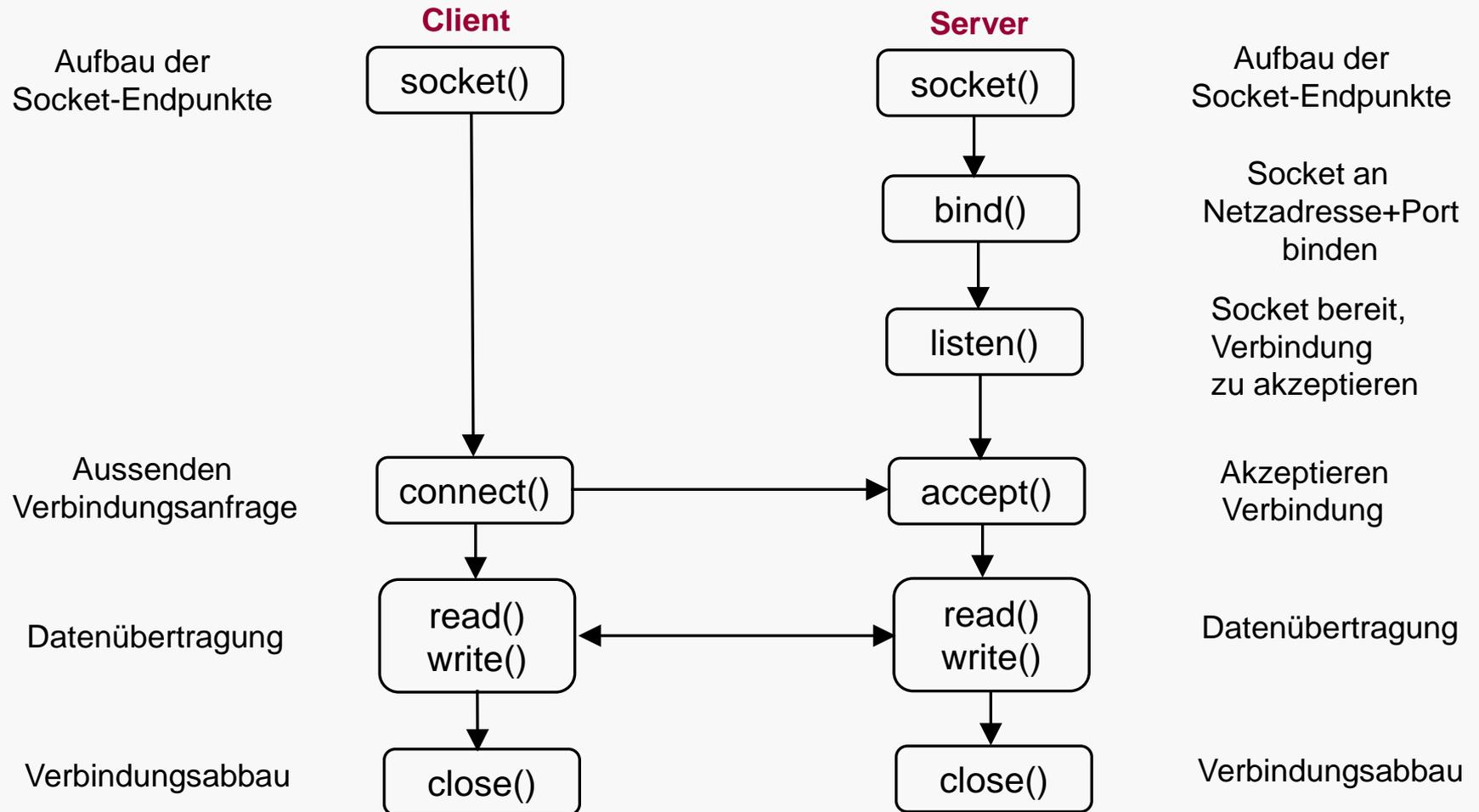
- *Application Programming Interface (API)* der Anwendung
 - ↳ programmierbar
 - **Socket-Programmierung**
- **Socket-Adresse:** IP-Adresse + Portnummer
 - Stream-Sockets
 - TCP
 - Datagram-Sockets
 - UDP

☞ **Die Socket-Programmierung ist eine Realisierungsform von Dienstprimitiven !!!**

Socket-Prozeduren (Berkeley)

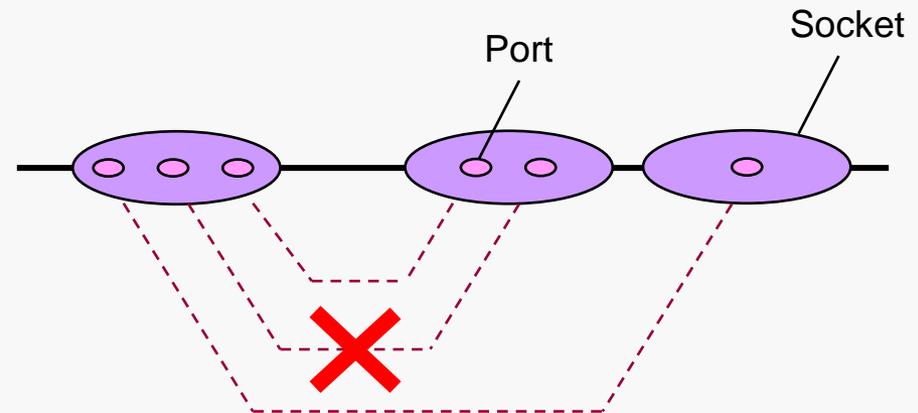
- *Socket* - Erzeugung eines neuen Sockets
- *Bind* - Binden einer lokalen Adresse an einen Socket
- *Listen* - Anzeige der Bereitschaft, eine Verbindung zu akzeptieren
- *Connect* - Aussenden einer Verbindungsanfrage
- *Accept* - Akzeptieren einer Verbindungsanfrage
- *Send* - Senden von Daten über die Verbindung. Dahinter verbirgt sich das Eintragen der Daten in den TCP-Sendepuffer
 - ↪ Durch Setzen der URG- und PSH-Flags kann ein sofortiges Absenden der Daten bewirkt werden
- *Receive* - Empfangen von Daten
- *Close* - Abbau einer Verbindung
- *Status* - Informationen über den Status der Verbindung, z. B. Socket-Adressen, Zustand der Verbindung, Größe der Fenster

Typisches TCP-Socket Szenario (Berkeley)



Verbindungen in TCP

- Expliziter Verbindungsaufbau
- von Socket zu Socket
 - ↪ **Verbindungsidentifikation:**
($socket_1, socket_2$)
→ keine Verbindungsreferenzen !!!
- mehrere Verbindungen an einem Socket möglich
 - ↪ aber nicht mit demselben Socket !!!
- Kollision beim Verbindungsaufbau führt zum Aufbau einer Verbindung



Erlaubte und nichterlaubte Verbindungen in TCP

Einschub: Protokollmechanismen zur Absicherung einer zuverlässigen Übertragung



Wichtige Protokollmechanismen

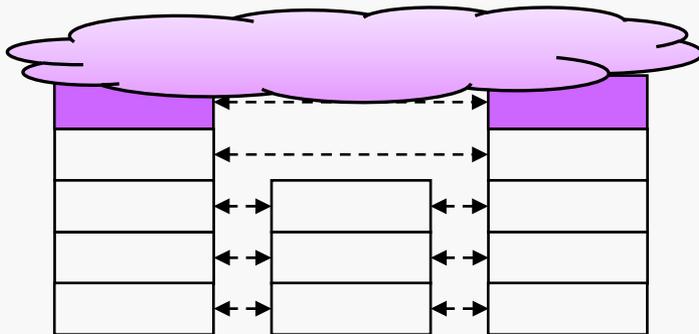
TCP verwendet zur Absicherung der zuverlässigen Übertragung eine Reihe von Protokollmechanismen, die im Abschnitt 2.5 eingeführt wurden. Obwohl TCP ausführlich erst in der VL Internet behandelt wird, sollen einige dieser Mechanismen hier genannt werden.

- Drei-Wege Handshake Verbindungsaufbau
 - Duplex-Übertragung
- Bestätigung jeder PDU
 - TCP: Segment
 - Timerüberwachung / Neuübertragung
- Sequenznummern mit Zeitstempel
- ARQ
 - Go back N/selektive Wiederholung
 - Stop and Wait (Fast Retransmit)
- Flusskontrolle mit Windows-Update
- Raten-basierte Flusssteuerung
 - Slow-Start
 - Überlaststeuerung
- Bestimmung Timeout-Werte durch Messung Round-Trip-Time
- Aktivitätstimer
- 2-Wege Handshake Verbindungsabbau



III.7

Anwendungsschicht



Tanenbaum / Wetherall 7
Stallings 22, 23
Kurose / Ross 2

Dienste und Protokolle in der Anwendungsschicht

In der Anwendungsschicht werden die vom Nutzer wahrnehmbaren Dienste eines Netzes bereitgestellt, die durch das Zusammenwirken der Protokolle aller Schichten erbracht werden. Sie werden als Anwendungs- oder Netz-dienste bezeichnet.

● Dienste

- häufig asymmetrische Dienste
- Nutzung
 - ↪ direkt (z.B. telnet, ftp, eMail, DNS)
 - ↪ Einbindung in eine Anwendung

● Protokolle der Anwendungsschicht

- i.d.R. einfache Struktur
 - ↪ z. B. Request/Response-Protokolle
 - ↪ text-orientiert (ASCII-Zeichen)
 - Reduzierung Kodierungs/Dekodierungsaufwand
 - siehe HTTP (Abschnitt III.6.2)



III.7.1

Gestaltung von Anwendungen



Gestaltung von Anwendungen (1)

Für die Gestaltung von Anwendungen oberhalb der Transportschicht gibt es unterschiedliche Möglichkeiten:

● Netzdienste

- z. B. telnet, ftp, eMail u.a.

● Webapplikationen

- World Wide Web
- Basisprotokoll: HTTP

● Anwendungen

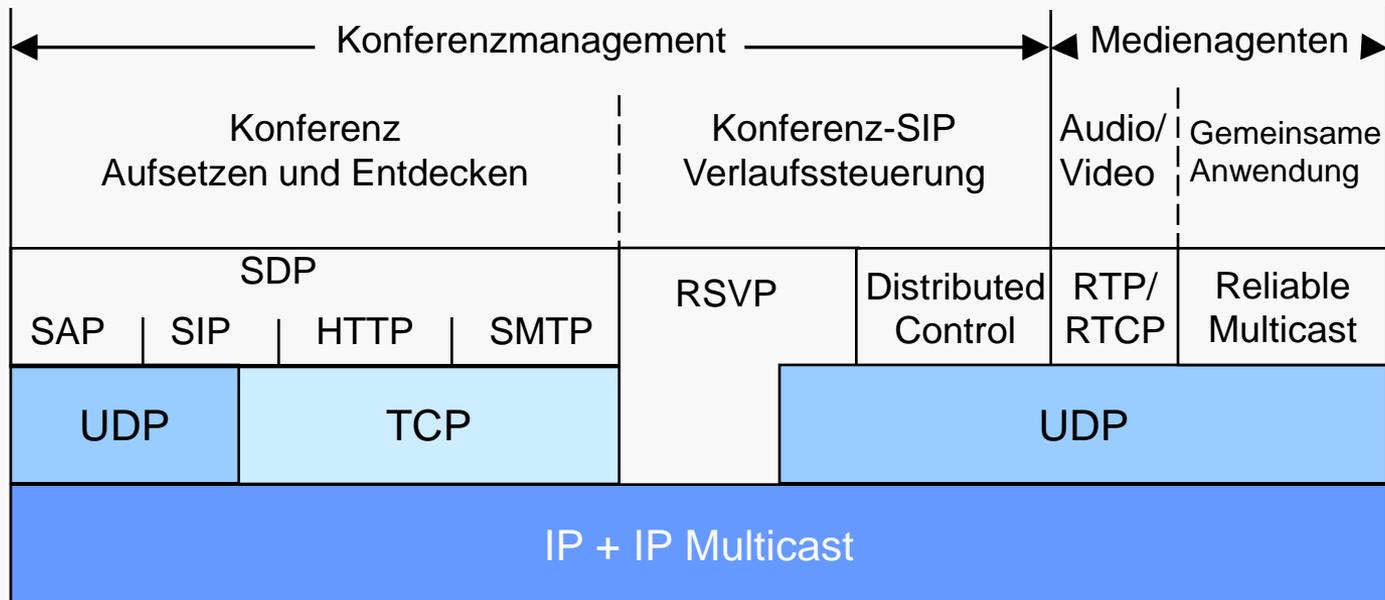
- anwendungsspezifische Software/Dienste
- können unmittelbar auf Transportschnittstelle aufsetzen
- häufig weitere Protokolle oder Middleware benötigt

↳ z. B. multimediale Anwendungen: Protokolle RTP + SIP

■ siehe VL Internet



MMUSIC-Protokollarchitektur für Voice over IP

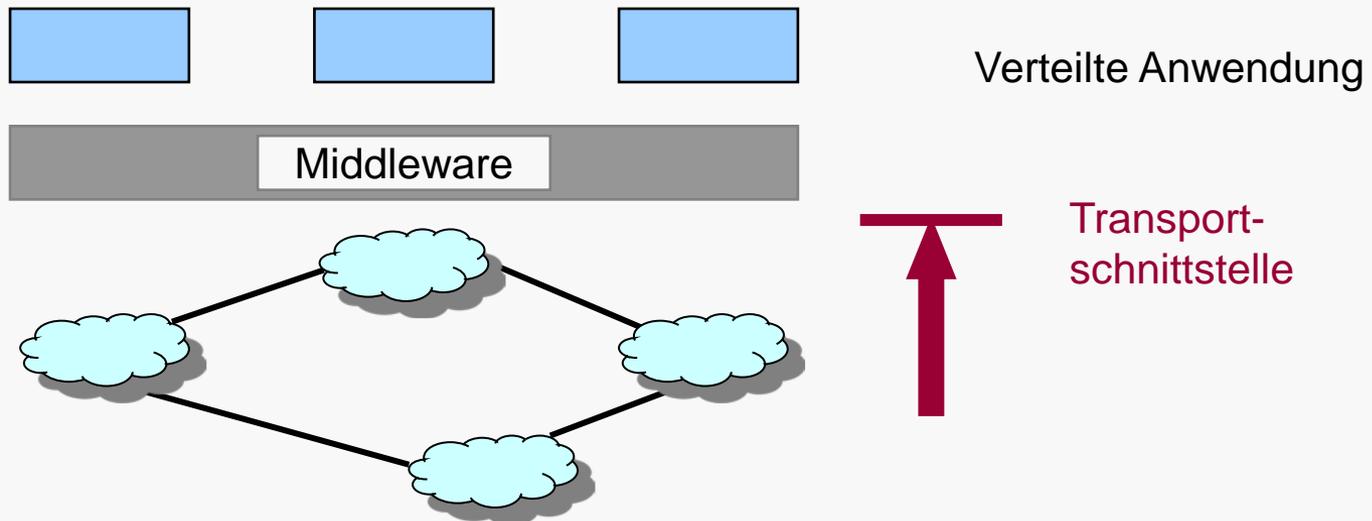


Gestaltung von Anwendungen (2)

● Client/Server-Anwendungen

- i.d.R. unterstützt durch spezifische Middleware

↪ z.B. CORBA, Web Services



Gestaltung von Anwendungen (3)

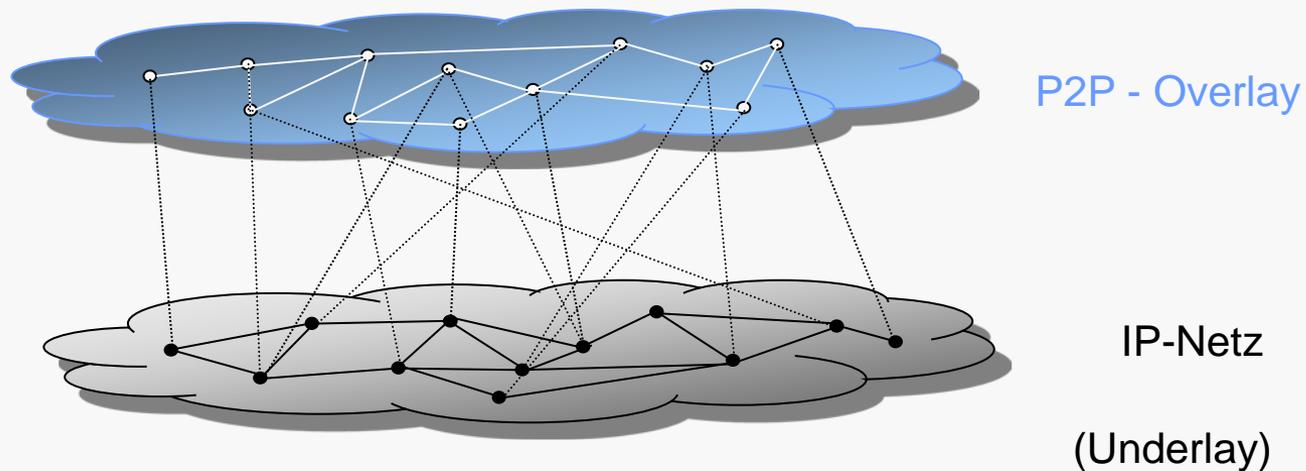
● Peer-to-Peer-Anwendungen

● gleichberechtigte Systeme

↳ keine Unterscheidung Client/Server

● spezifische Middleware möglich

↳ z.B. JXTA



III.7.2

Zwei Anwendungsprotokolle



III.7.2.1

FTP

(File Transfer Protocol)

RFC 959



FTP (1)

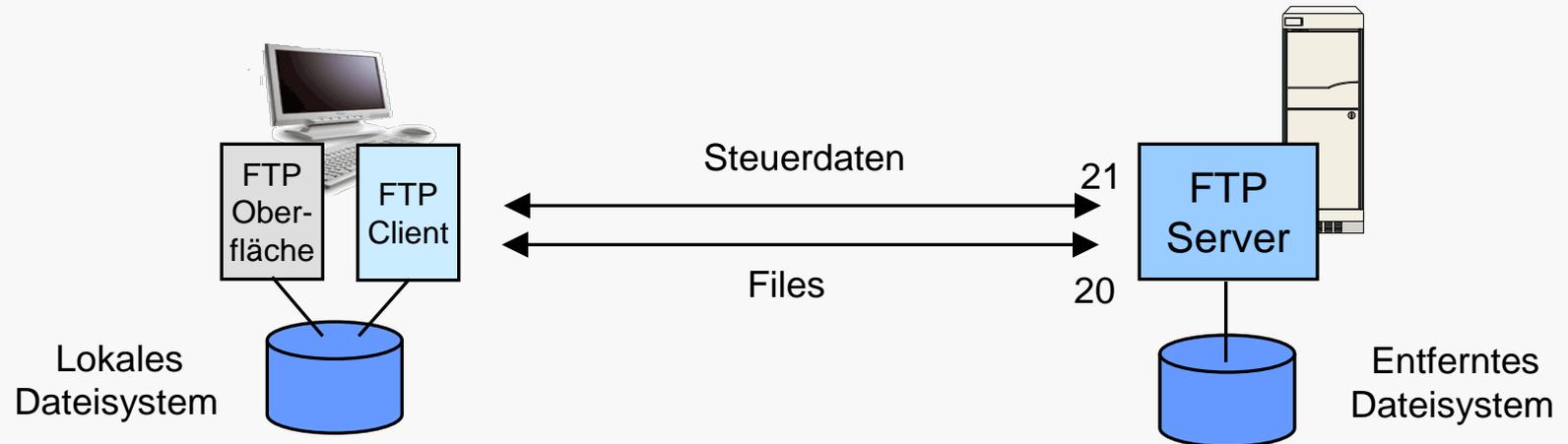
FTP ist ein Protokoll für die Übertragung von Dateien zwischen Hosts.

- Anfrage/Antwort Protokoll

↪ über TCP

- Out-of-band Signalisierung

↪ serverseitig: Port 20: Datenübertragung, Port 21: Steuerdaten



FTP (2)

● Kommandos (Auswahl)

- USER *username* - Benutzeridentifizierung zum Server
- PASS *password*
- LIST - Liste der Dateien im aktuellen entfernten Verzeichnis
- PORT - Angabe des Client-Ports
- RETR *filename* - Übertragung Datei aus aktuellem Fileverzeichnis des entfernten Host
- STOR *filename* - Übertragung Datei in das aktuelle Verzeichnis des entfernten Hosts

● Antworten (Auswahl)

- 331 Username OK, password required
- 125 Data connection already open; transfer starting
- 425 Can't open data connection
- 452 Error writing file

FTP (3)

- Separate TCP-Verbindungen für Übertragung Kommandos und Daten
- **Aktives FTP**
 - Client wählt einen zufälligen Port (>1035) und teilt diesen mit der IP-Adresse dem Server mit mittels des PORT-Kommandos
 - danach kann File-Transfer beginnen
- **Passives FTP**
 - Client sendet PASV-Kommando
 - Server öffnet Port und sendet Port-Nummer mit IP-Adresse an Client
 - ↳ Verwendung, wenn Server wegen NAT-Box keine Verbindung zum Client aufbauen kann
- **Anonymous FTP**
 - öffentliche FTP-Server
 - Account: anonymous oder ftp
 - Passwort: e-Mail-Adresse



III.7.2.2

HTTP

(HyperText Transfer Protocol)

(RFC 1945, 2616)



Tanenbaum / Wetherall 7.3.4
Stallings 23.2
Kurose / Ross 2.2

World Wide Web (WWW)

Das **WWW** ist ein Architektur-Rahmenwerk für den Zugriff auf verlinkte Dokumente im Internet.

- Basis-Idee: Tim Berners Lee (1989)
 - CERN (Schweiz)
 - ursprüngliches Ziel: weltweite Bereitstellung von Messergebnissen physikalischer Experimente
- **W3C** (<http://www.w3.org>)
 - World Wide Web Consortium
 - Dachorganisation des WWW
 - regelt alle das WWW betreffenden Fragen, z. B. Adressräume
 - unterstützt Forschung/Standardisierung
 - ↳ Entwicklung des Web 2.0 und des semantischen Webs

Unterschied Web/Internet

Es existiert eine synonyme Verwendung des Begriffs Internet in den Medien.
Damit ist i.d.R. das Web gemeint!

↪ fachlich nicht korrekt!

● Internet

- weltumspannender Netzverbund
 - ↪ Kommunikationsinfrastruktur
 - ↪ Ursprung des Namens

● Web

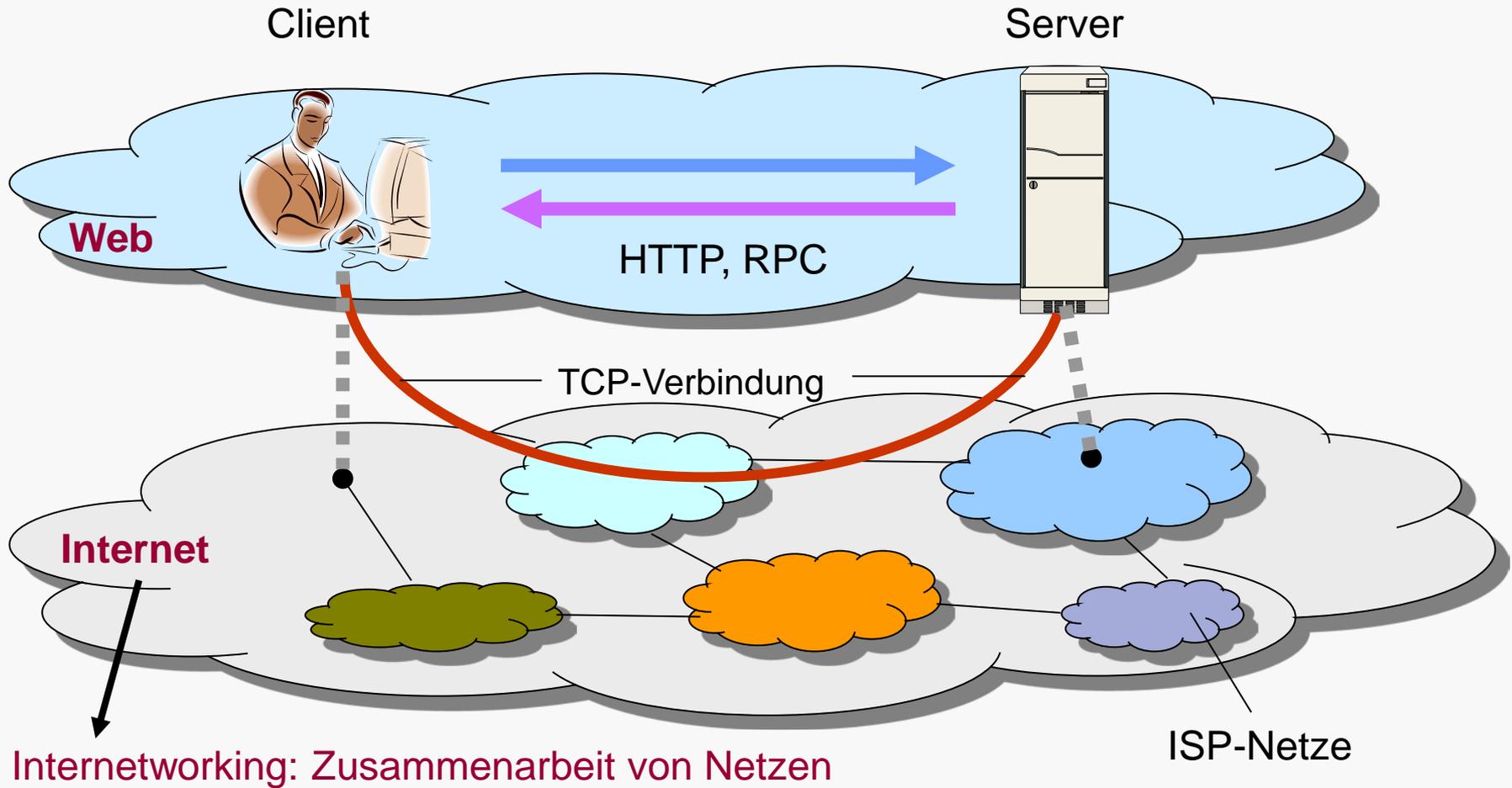
- ein Dienst des Internets
 - ↪ Client/Server-Anwendung
- weltumspannender Informationsraum

● Web-Technologien

- anwendungsunterstützende Dienste/Techniken, die auf dem WWW aufbauen
 - ↪ dienen der Unterstützung/Entwicklung von Anwendungslösungen



Web und Internet



ISP – Internet Service Provider

HTTP

HTTP ist Standardprotokoll für Kommunikation zwischen Web-Client (Web-Browser) und Web-Server, das hauptsächlich dazu eingesetzt wird, Webseiten anzufragen und herunter zu laden.

- über TCP, Port 80
- ab HTTP v1.1: dauerhafte TCP-Verbindungen (*persistent connections*)
 - ↳ mehrere HTTP-Anfragen über eine TCP-Verbindung
- Anfrage/Antwort-Protokoll (*Request/Response Protocol*)
 - Anfrage: Methoden
 - Antwort: Statusmeldung
 - 👉 so genanntes „zustandsloses“ Protokoll
 - ↳ Antwort/Response ist unabhängig von vorherigen Anfragen
- Text-orientiertes Protokoll
 - Nachrichten werden zeichenweise kodiert, **NICHT** bit-orientiert
 - ↳ einfachere Kodierung/Dekodierung



HTTP/2

Umfassende Überarbeitung des HTTP 1.1 Protokolls.

- RFC 7540, 7541
- Vollständig abwärtskompatibel zu HTTP/1.1
- **Neue Features**
 - Zusammenfassen (Multiplex) mehrerer Anfragen über einer Verbindung
 - Weitergehende Datenkompressionsmöglichkeiten einschl. Headerdaten
 - Binär kodierte Übertragung von Inhalten
 - Server-initiierte Datenübertragungen (push-Verfahren)
- Keine TLS-Verschlüsselung



HTTP-Methoden

GET	Anfordern einer Webseite
HEAD	Anfordern des Headers einer Webseite
PUT	Speichern einer Webseite auf einem Server
POST	Anhängen eines Datenblocks
DELETE	Entfernen einer Webseite
TRACE	Rücksenden einer Anforderung (Debugging)
CONNECT	für Proxy-Server, die SSI-Tunnel unterstützen
OPTIONS	Abfrage von unterstützten Methoden/Features

HTTP-Statuscodes

Code	Bedeutung	Beispiel
1xx	Information	100 Continue, 102 Processing
2xx	Erfolg	200 OK, 204 No Content
3xx	Umleitung	301 Moved Permanently
4xx	Client-Fehler	403 Forbidden, 404 Not Found
5xx	Server-Fehler	503 Service Unavailable

Nachrichten-Header

Nach den Methoden sowie nach den Antworten können weitere Zeilen mit Informationen folgen.

- Header
 - ↳ Request Header/Response Header
- mehrere Header-Typen

● Beispiel:

```
GET /btu/Spezial:Search?search=Studium&go=Text HTTP/1.1
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, */*
Referer: http://www.tu-cottbus.de/cms/
Accept-Language: de Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows 98; QXW0339c)
Host: www.tu-cottbus.de
Connection: Keep-Alive
```

} Anfrage-Header

HTTP: Beispiel

```
ao@banquo:/home/ao/temp 222 > telnet portia
80
```

```
Trying 141.43.3.129...
```

```
Connected to portia.Informatik.TU-Cottbus.DE.
```

```
Escape character is '^['.
```

```
GET /de/heading.html HTTP/1.1
```

```
Host: www-rnks.informatik.tu-cottbus.de
```

```
HTTP/1.1 200 OK
```

```
Date: Tue, 02 Dec 2003 17:58:38 GMT
```

```
Server: Apache/1.3.26 (Unix)
```

```
Last-Modified: Thu, 10 Aug 2000 18:19:34 GMT
```

```
ETag: "6e42cf-418-3992f236"
```

```
Accept-Ranges: bytes
```

```
Content-Length: 1048
```

```
Content-Type: text/html
```

Antwort-Header

(HTTP-Header)

```
<html>
```

Fortsetzung

```
<head>
```

```
<meta http-equiv="Content-Type"
content="text/html; charset=iso-8859-1">
```

```
<title>Lehrstuhl Rechnernetze und
Kommunikationssysteme</title>
```

```
<link rel="stylesheet" type="text/css"
href=" ../headingsstyles.css">
```

```
</head>
```

```
<body>
```

```
<table width="100%" height="100%">
```

```
<tr>
```

```
<td align="center" valign="middle">
```

```
...
```

(Inhalt der HTTP-Nachricht: HTML-Seite)



HTTP: Beispiel

(Erläuterung zum HTTP-Antwort-Header)

HTTP/1.1 200 OK
Date: Tue, 02 Dec 2003 17:58:38 GMT
Server: Apache/1.3.26 (Unix)
Last-Modified: Thu, 10 Aug 2013 18:19:34 GMT
ETag: "6e42cf-418-3992f236"
Accept-Ranges: bytes
Content-Length: 1048
Content-Type: text/html

Statusmeldung 200 → erfolgreicher Abruf der Seite

Info über Web-Server

Modifikationsdatum der HTML-Seite

eindeutige ID des Objekts (der HTML-Seite)
(bei HTTP/1.1 soll diese ID für Vergleich gecachter
Objekte verwendet werden)

es handelt sich um HTML-Seite
(und nicht um JPEG-Datei)

Web-Server erlaubt das Abrufen
bestimmter Ausschnitte der Datei
(wichtig für Wiederaufsetzen nach
abgebrochener Verbindung)

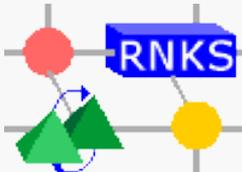


III.8

Tunneln



Tanenbaum / Wetherall 5.5.3



Tunneln

Tunneln ist eine Technik, um Netze mit gleicher Kommunikationsarchitektur über ein Netz zu verbinden, das über eine andere Protokollstruktur verfügt.

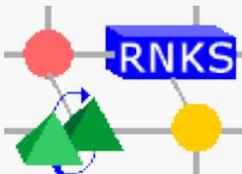
● Anwendungen

● Internet

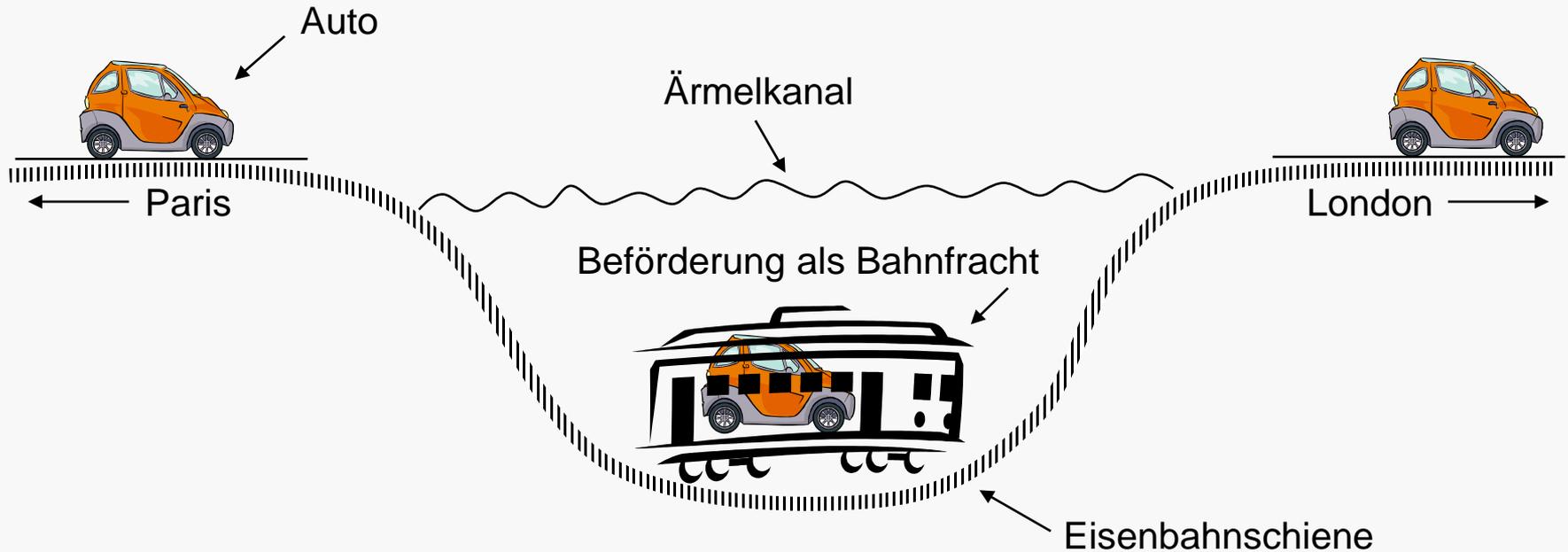
- ↳ Überwindung von Netzen, die nicht die TCP/IP-Architektur nutzen
→ z.B. Verbinden von IPv6-Inseln über IPv4-Tunnel

● Verbund von LAN über WAN

- ↳ entfernte Brücken

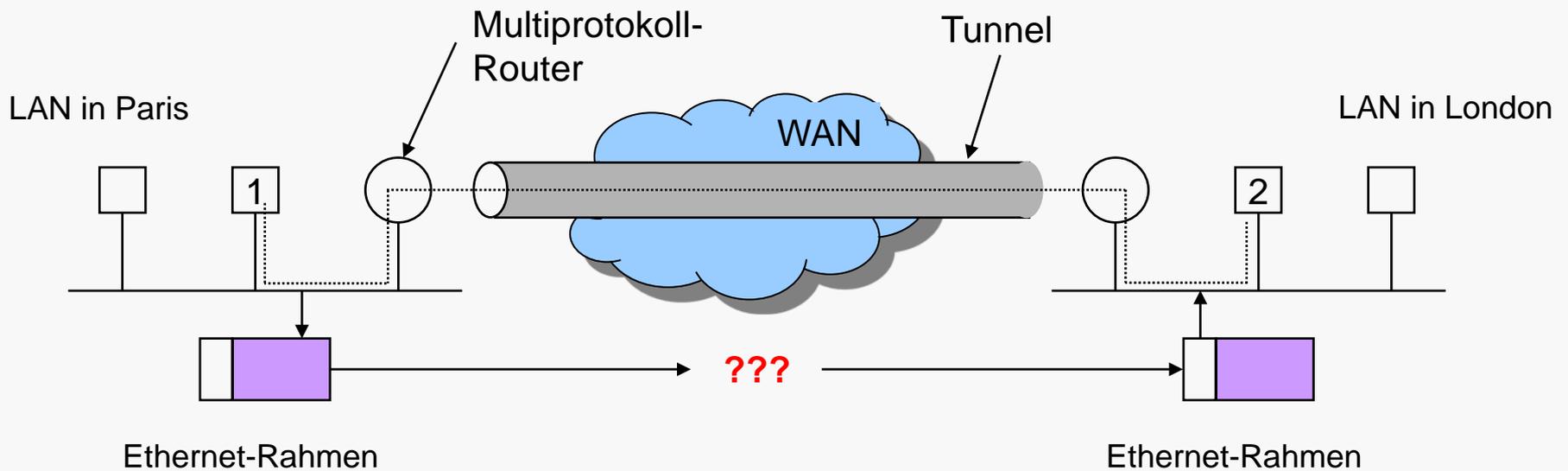


Beförderung eines PKWs von Frankreich nach England¹



1. entnommen: Tanenbaum, A. S., Wetherall, D.J. : Computernetzwerke, Pearson Studium, 5. Auflage, 2012.

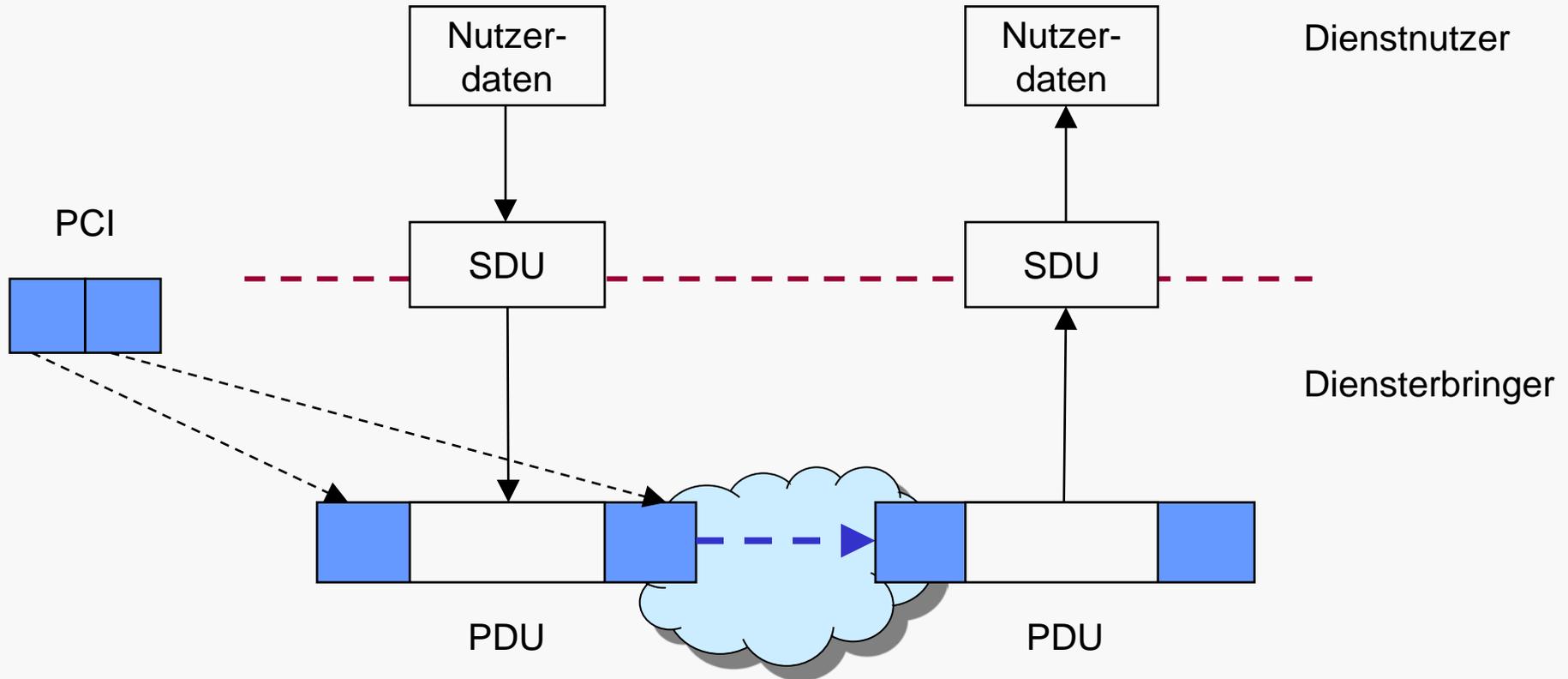
Tunneln eines Ethernet-Frames von Paris nach London¹



1. nach: Tanenbaum, A. S., Wetherall, D.J. : Computernetzwerke, Pearson Studium, 5. Auflage, 2012.

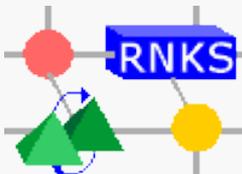
Prinzip der Transparenz

(Wiederholung)

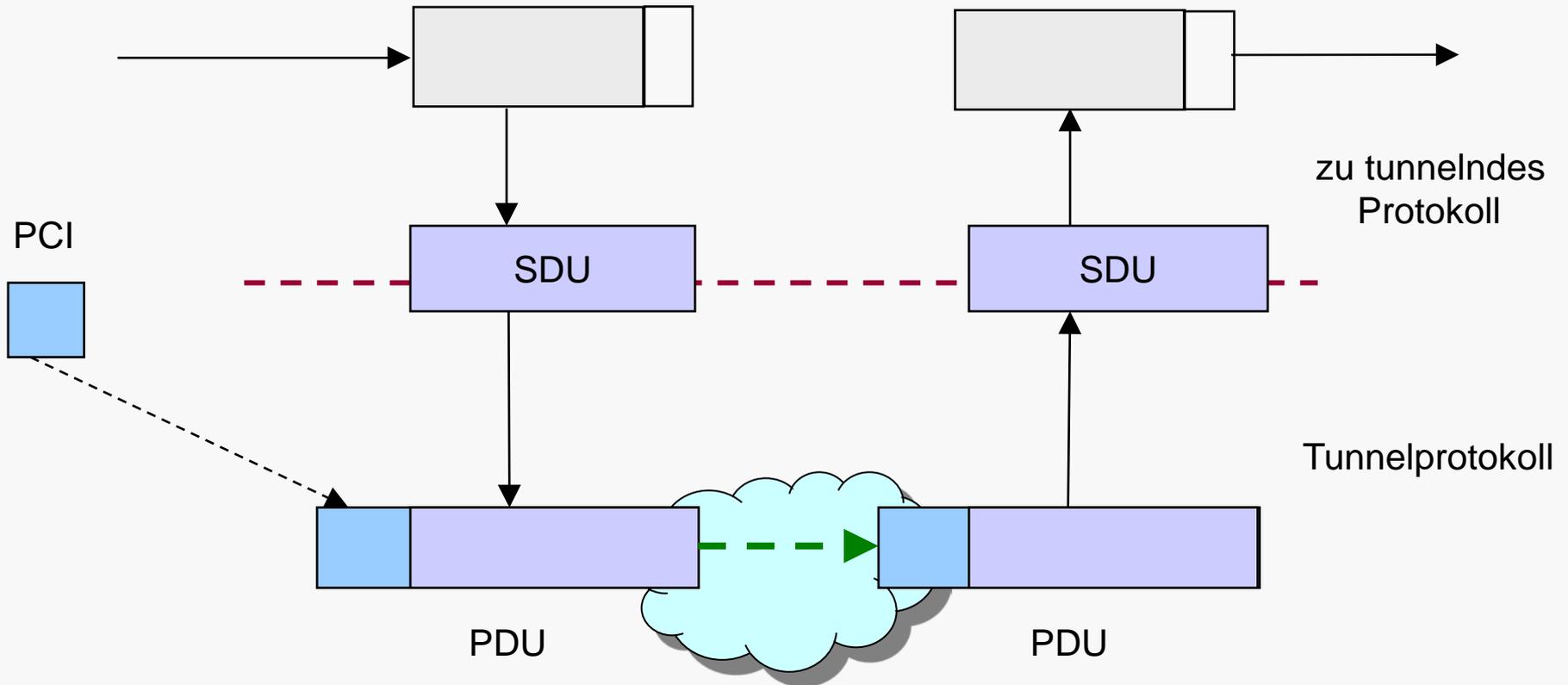


Legende:

PCI – Protocol Control Information
PDU – Protocol Data Unit
SDU – Service Data Unit



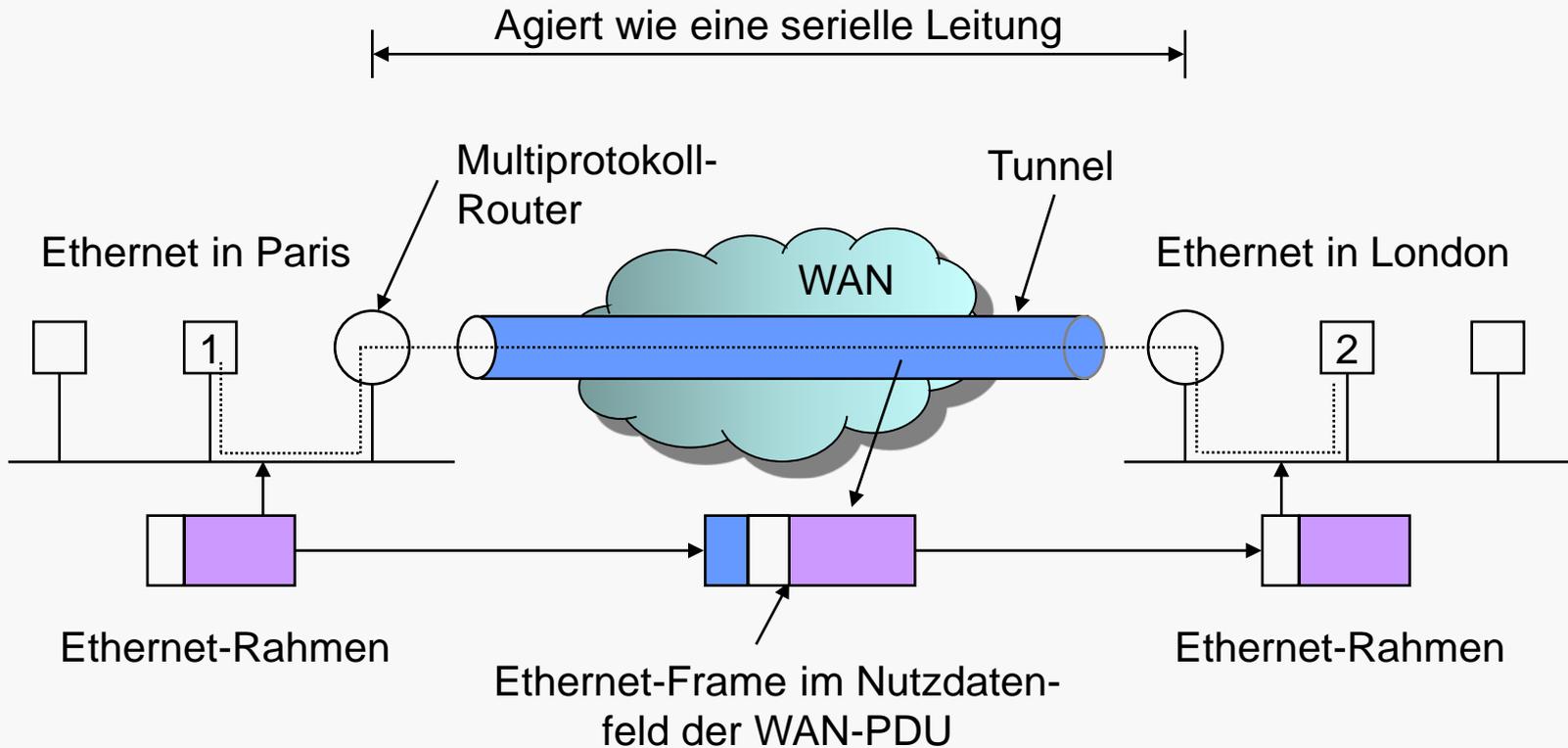
Prinzip des Tunnels



Legende:

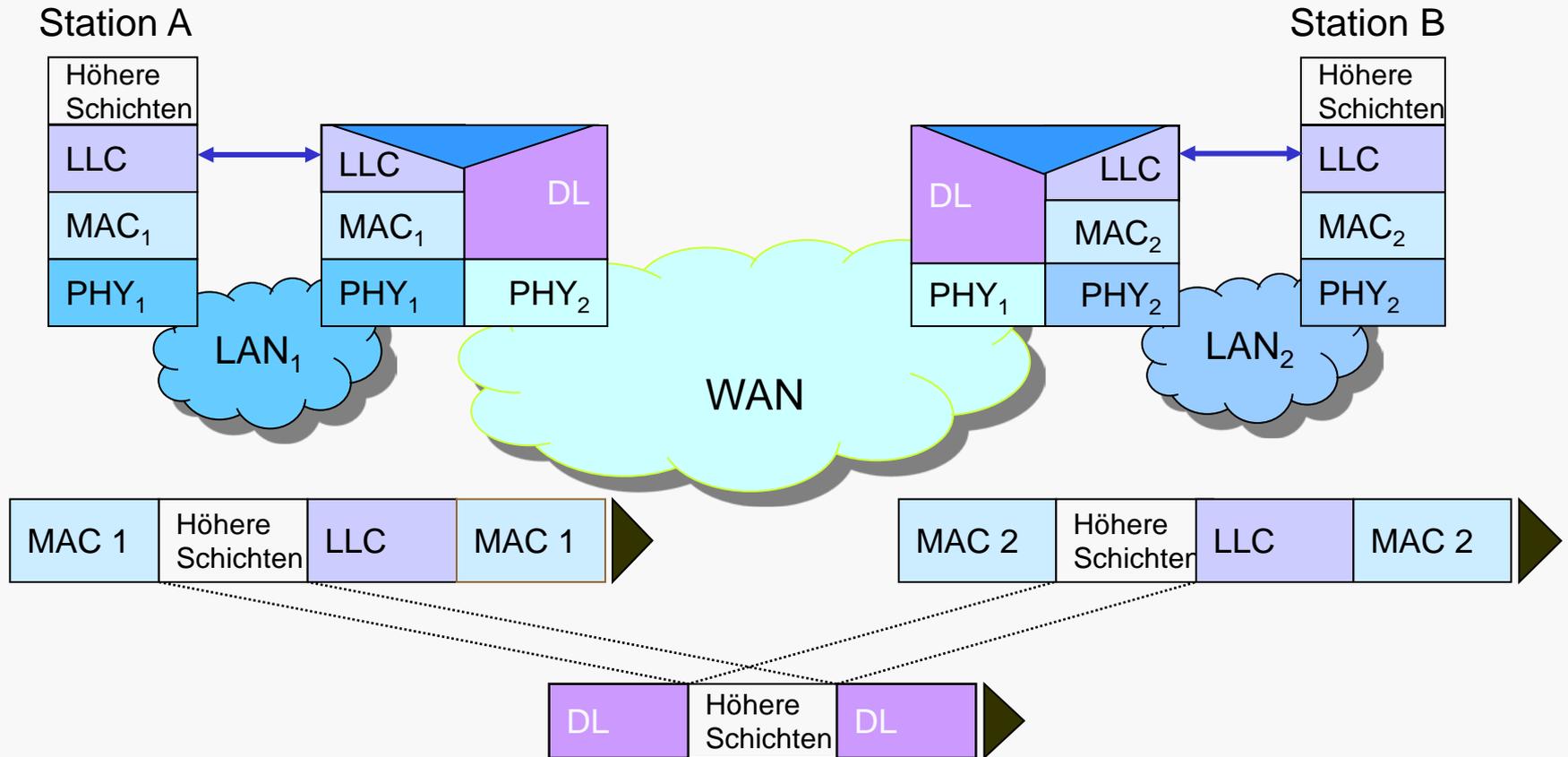
PCI – Protocol Control Information
PDU – Protocol Data Unit
SDU – Service Data Unit

Tunneling eines Ethernet-Frames von Paris nach London¹



1. nach: Tanenbaum, A. S., Wetherall, D.J. : Computernetzwerke, Pearson Studium, 5. Auflage, 2012.

Struktur einer entfernten LLC- Brücke¹



1. nach: Badach, A.; Hoffmann, E.; Knauer, O.: High Speed Internetworking. Addison-Wesley, 1994.