

Brandenburgische Technische Universität Cottbus

Lehrstuhl Rechnernetze und  
Kommunikationssysteme

b-tu

---

# Rechnernetze

## Eine (kurze) Einführung

Cluj, Wintersemester 2019/20

**Prof. Dr.-Ing. habil. Hartmut König**

---

## II.4

# Schichtenarchitekturen



König 4.1  
Tanenbaum/Wetherall 1.3.1



---

# Schichtenarchitekturen

In Rechnernetzen und Kommunikationssystemen werden aufeinander abgestimmte Schichtenarchitekturen verwendet, in die die Kommunikationsprotokolle eingebettet werden.

## ● Schichtenarchitektur:

- Definiert die Funktionalität der Schichten und legt die Prinzipien der Interaktion zwischen ihnen fest
- In der Regel durch Standardisierungsgremien (z.B. IETF, ISO) oder Firmenkonsortien
  - ↳ langwieriger Prozess
    - auch durch Firmenstrategien und politische Kompromisse geprägt
    - meist nicht von Protokollentwickler direkt beeinflusst

---

# Arten von Schichtenarchitekturen

Schichtenarchitekturen werden bzgl. der Art der Integration neuer Rechnersysteme in zwei Arten unterschieden:

- **Geschlossene Architekturen**

- ↳ geschlossene Systeme

- **Offene Architekturen**

- ↳ offene Systeme

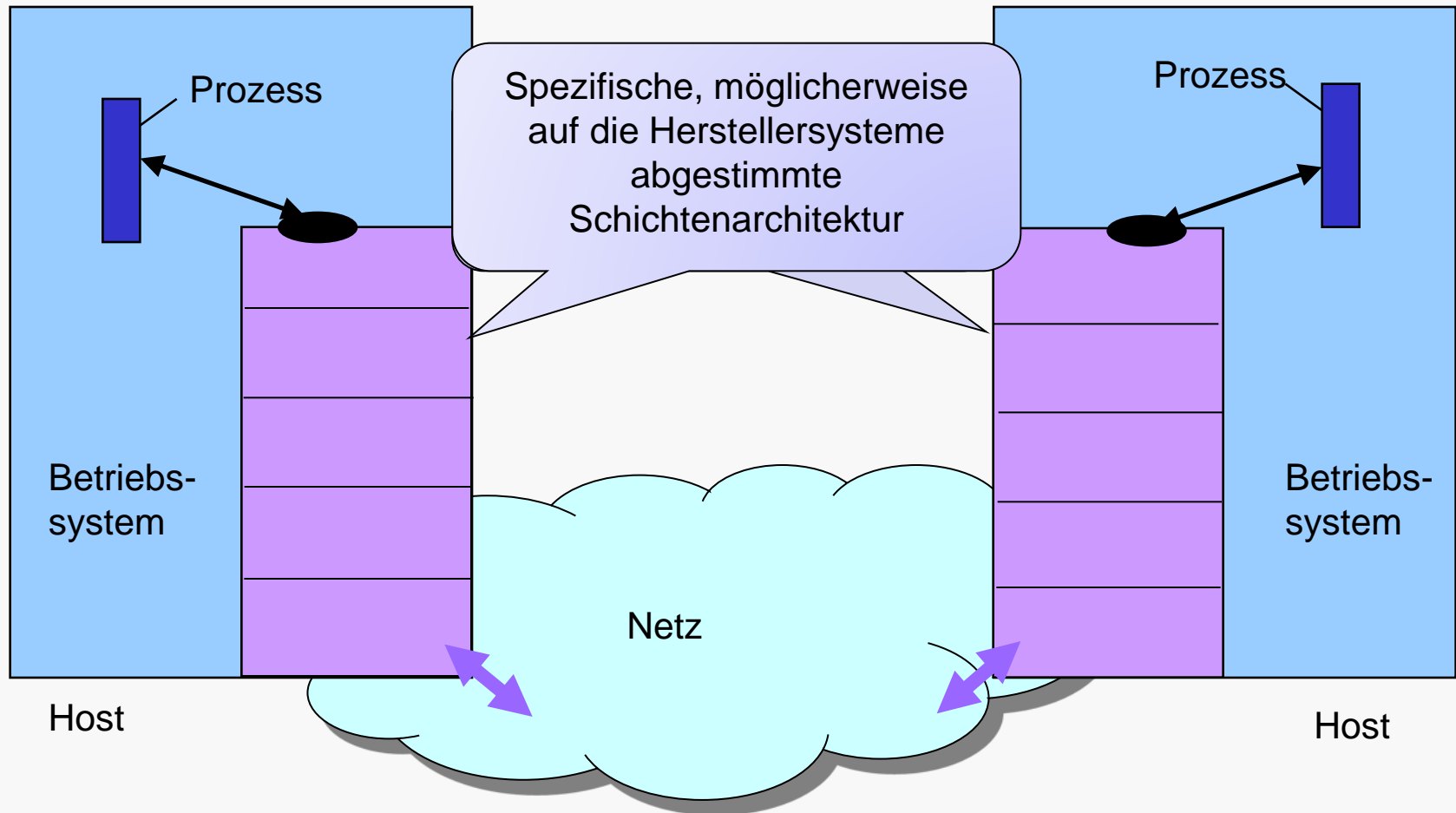
---

## II.4.1

# Geschlossene Schichtenarchitekturen



# Geschlossene Schichtenarchitekturen



---

# Geschlossene Architekturen

**Geschlossene Schichtenarchitekturen** sind auf ein bestimmtes Anwendungsgebiet ausgerichtet. Sie berücksichtigen die spezifischen Anforderungen des jeweiligen Einsatzbereichs oder eines bestimmten Produzenten.

## ● Proprietäre Architekturen

↳ auch *Herstellerorientierte geschlossene Architekturen*

- Systeme, die die Hard- und Softwareprodukte eines Herstellers verbinden, oder für einen speziellen Einsatzfall entwickelt wurden.
- **Beispiele:**
  - SNA (*System Network Architecture*) (IBM) (1970/80 Jahre)
  - DNA (*Digital Network Architecture*) (DEC) (1970/80 Jahre)

---

# Proprietäre Architekturen

## ● Vorteile:

- effizient, da auf die Herstellersysteme ausgerichtet
- gute Pflege
- kontinuierliche Weiterentwicklung
- schnelle Reaktion auf neue Markterfordernisse

## ● Nachteile:

- Einbindung von Systemen anderer Hersteller ist aufwendig



**Im Internetzeitalter haben geschlossene Architekturen ihre Bedeutung verloren !!!**



Bleiben auf spezielle Einsatzfälle beschränkt !!!



Existierende proprietäre Architekturen haben sich der Internet-Architektur geöffnet !!!



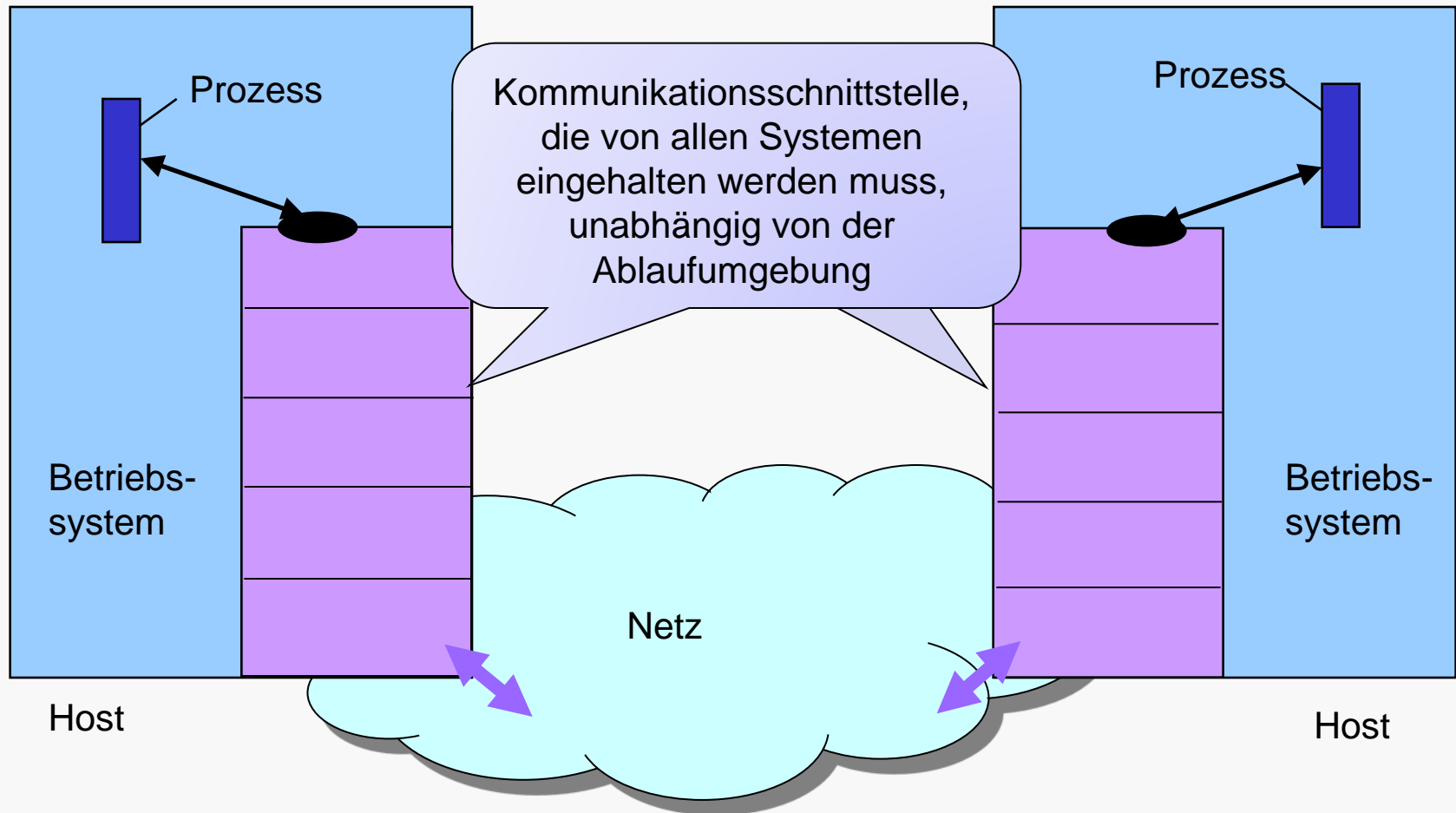
---

## II.4.2

# Offene Schichtenarchitekturen



# Offene Schichtenarchitekturen



---

# Offene Architekturen (1)

Offene Schichtenarchitekturen definieren eine einheitliche Kommunikationsschnittstelle (Protokolle), die von allen Systemen einzuhalten sind.

● **Offen:** Jedes Rechnersystem, das diese Schnittstelle einhält, kann in das Netz integriert werden

☞ **Keine Implementierungsvorschrift !!!**

↪ Kommunikationssoftware kann in jedem System unterschiedlich implementiert sein !!!

■ Die Protokollschnittstelle muss gleich sein !!!

---

# Offene Architekturen (2)

- Offene Architekturen sind die Voraussetzung für den Aufbau
  - heterogener Netze
  - Internationaler/firmenübergreifender Netze
  - des Internets
  
- **Beispiele:** - OSI-Kommunikationsarchitektur  
- Internet-Architektur

---

# Standardisierte Protokolle

Das Ziel einer offenen Kommunikation kann nur erreicht werden, wenn die Protokolle, die in den einzelnen Schichten eingesetzt werden, standardisiert werden.

## ● OSI-Architektur

- Standardisierung durch ISO und CCITT/ITU-T

## ● Internet-Architektur

- Standardisierung durch IETF
  - ↳ Request for Comments (RFCs)

## ● Lokale Netze

- Standardisierung durch IEEE

☞ **Die Standardisierung ist die Grundlage für die mehrfache Implementierung von Protokollen !!!**

---

# II.4.3

## Protokollarchitekturen



---

# Protokollarchitekturen

Ursprünglich wurden Schichtenarchitekturen durch Kommunikationsarchitekturen (OSI-Modell) beschrieben. Heutzutage haben sich in der Praxis so genannte Protokollarchitekturen etabliert.

## ● Protokollarchitekturen:

- Hierarchische Protokollstrukturen in denen die Schnittstellen zwischen den Protokollen nicht mehr durch definierte Architekturelemente, sondern durch die Protokollschnittstellen bestimmt werden.
  - ↪ enthalten keine vorgegebenen Architekturelemente wie Schichten, SAPs usw.
  - ↪ Protokolle können in verschiedenen Protokollarchitekturen auftreten
- Anpassung an verschiedene Protokollschnittstellen erforderlich

## ● Anwendung

- häufig zur Definition einer Protokollhierarchie für eine bestimmte Anwendung
  - ☞ Internet-Architektur ist eine Protokollarchitektur



---

# Protokollstack

- Als **Protokollstack** oder *Protokollstapel* wird allgemein eine Protokollhierarchie in einer Kommunikations- oder einer Protokollarchitektur bezeichnet.

 **umgangssprachlicher Begriff !!!**



---

## II.4.4

# Ausgewählte Schichtenarchitekturen



---

# II.4.4.1

## OSI-Referenzmodell



König 4.3.1  
Tanenbaum/Wetherall 1.4.1

---

# OSI-Referenzmodell (1)

Das **OSI-Referenzmodell** galt lange Zeit als das wichtigste Referenzmodell für den Aufbau offener heterogener Rechnernetze.

↪ OSI/RM - *Open Systems Interconnection / Reference Model*

● **Entwickler:** ISO (*International Standardisation Organisation*)

↪ genauer: ISO/TC97/SC16

● **Entwicklungszeitraum:** 1978 -1984

● 1984 internationaler Standard (ISO IS 7498)

● später ergänzt durch ein Netzmanagement-Modell und eine Sicherheitsarchitektur

● **Ursprünglich konzipiert für flächendeckende Rechnernetze (WAN)**

↪ Grundlage: Vermaschte Topologie



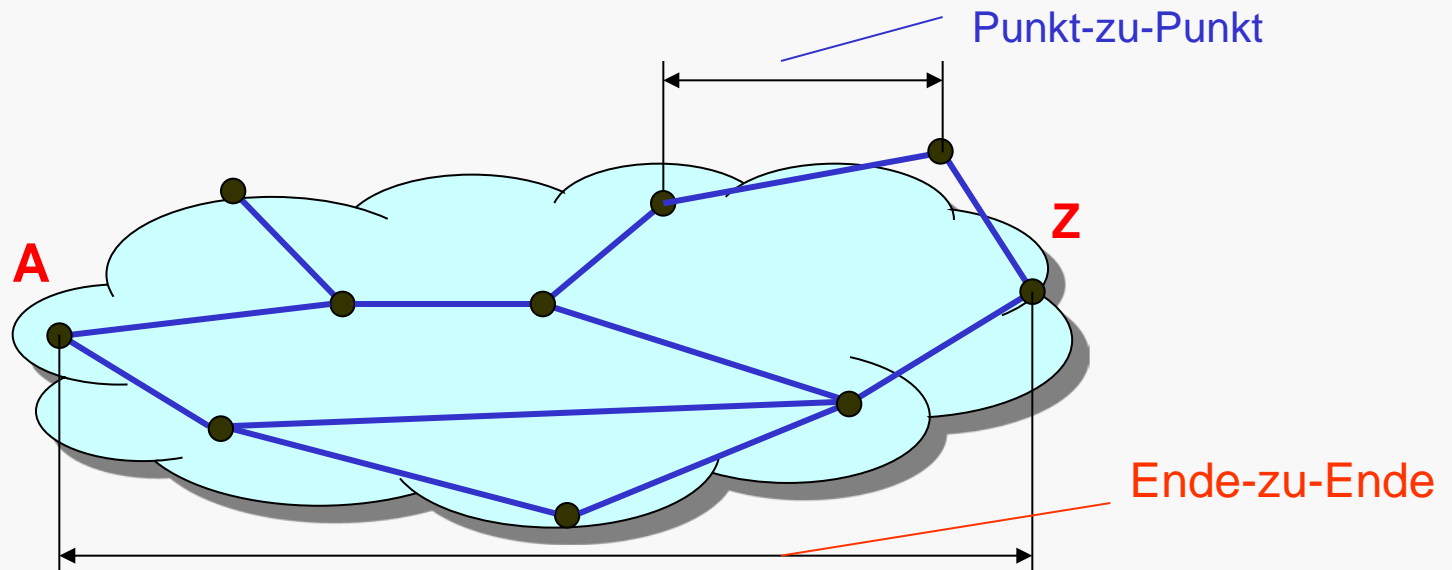
# Vermaschte Topologie

- **Topologie**

Art und Weise, wie die Netzknoten angeordnet und miteinander verbunden sind

- **Vermaschte Topologie**

Beliebige Verteilung der Netzknoten über einen möglicherweise großen Raum über Punkt-zu-Punkt-Verbindungen

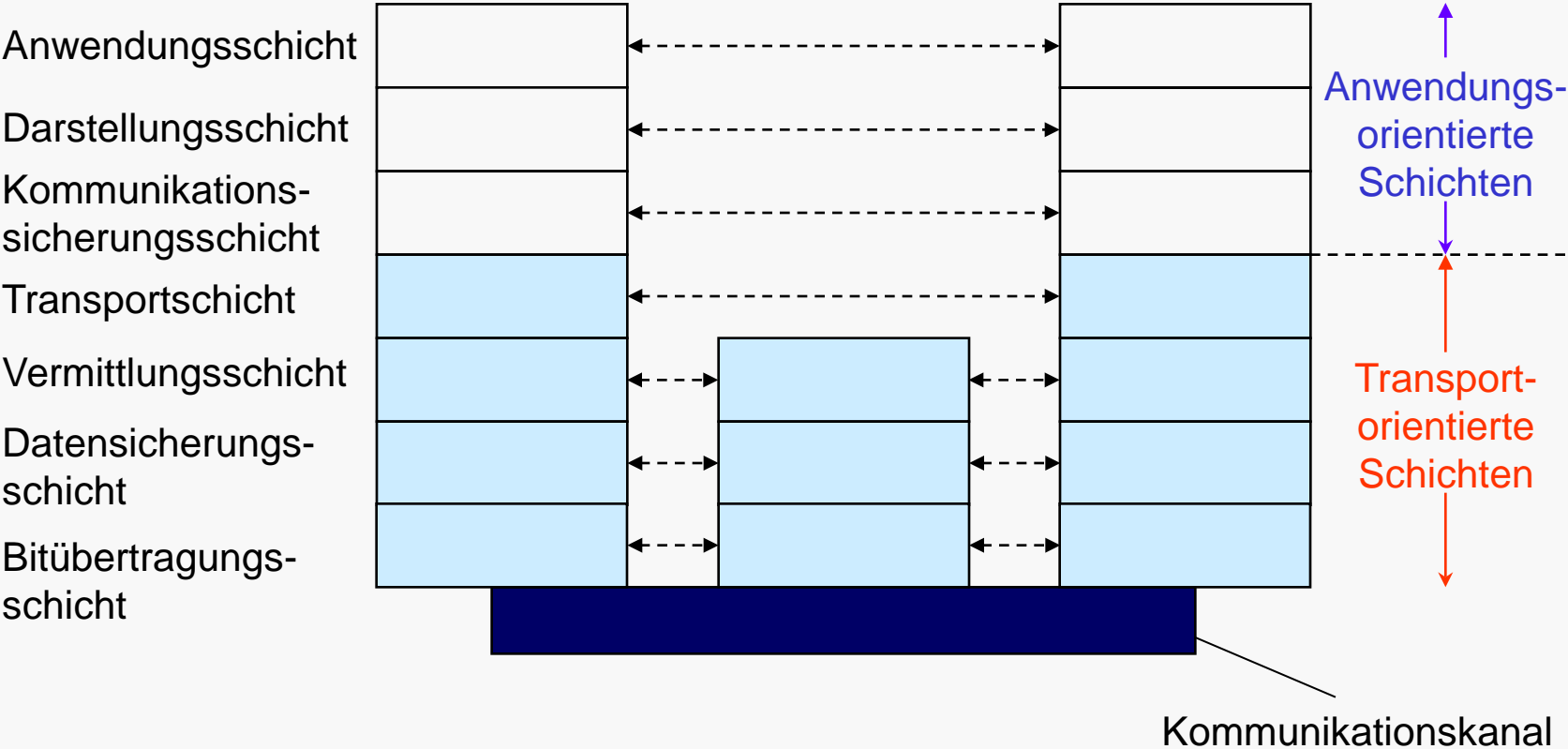


---

# OSI-Referenzmodell (2)

- Besteht aus 7 Schichten
  - Auswahl auf der Grundlage spezifischer Kriterien
- **Schichten 1- 4:** Transport-orientierte Schichten
  - Realisieren Übertragungsfunktionen
- **Schichten 5-7:** Anwendungs-orientierte Schichten
  - Unterstützen die Anwendung

# OSI-Referenzmodell



---

# A) Bedeutung der OSI-Schichten



# OSI-Schichten 1+2

## (1) Bitübertragungsschicht

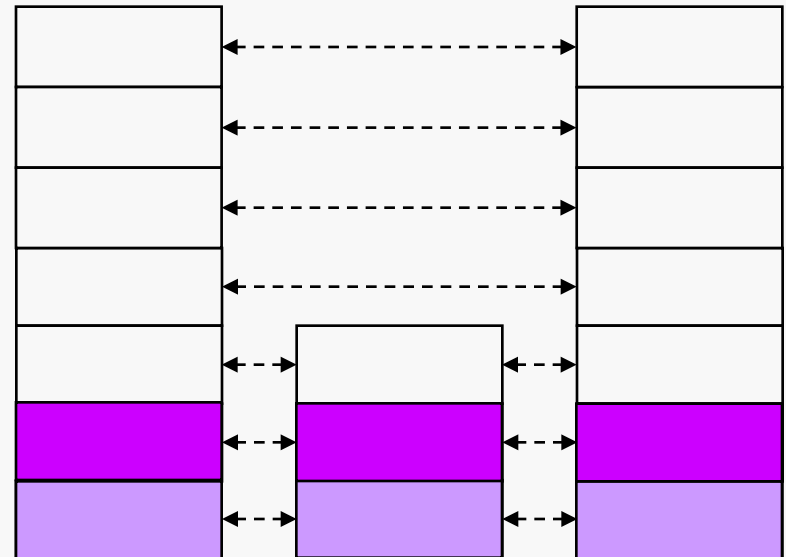
↳ *physical layer, PH*

- Übertragung der Daten in Form von Signalen über Kommunikationskanal
- Definiert die mechanischen, elektrischen, funktionalen sowie prozeduralen Anforderungen für die Datenübertragung

## (2) Datensicherungsschicht

↳ *data link layer, DL*

- Sichert die korrekte Übertragung zusammenhängender Bitfolgen (*Rahmen*) zwischen 2 Netzknoten





# OSI-Schichten 3+4

## (3) Vermittlungsschicht

↪ *network layer, N*

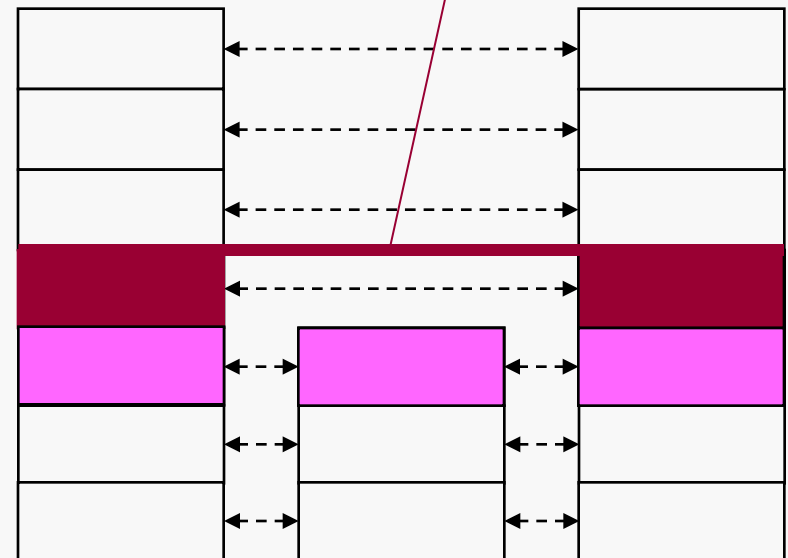
- Übernimmt die Vermittlung und Weiterleitung der Daten im Netz
- Wegewahl (*Routing*)
- Realisiert auch den Verbund von Netzen (*internetworking*)

## (4) Transportschicht

↪ *transport layer, T*

- Realisiert direkte Kommunikation zwischen den Endsystemen
- **Ende-zu-Ende Kommunikation**
- Verdeckt Netzstruktur

Transportschnittstelle



# OSI-Schichten 5+6

## (5) Kommunikationssicherungsschicht

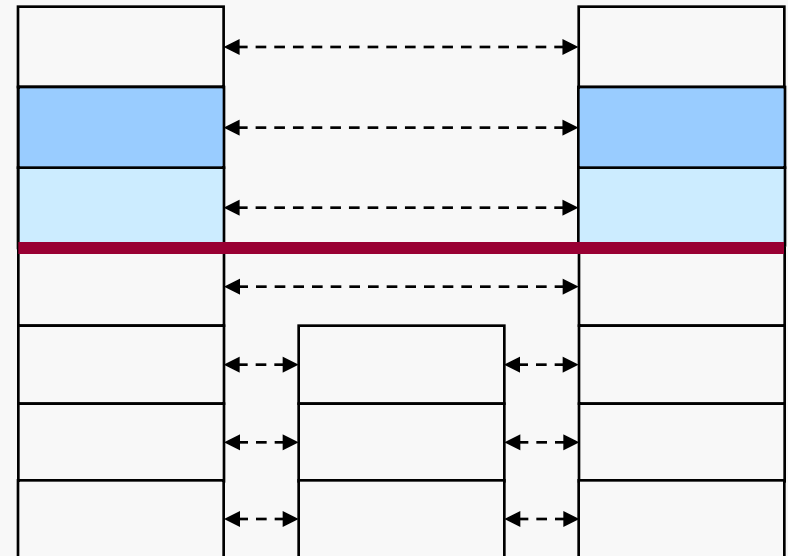
↪ *session layer, S*

- Dient der inhaltlichen Synchronisation der Kommunikation
  - ↪ Resynchronisation
  - ↪ Wiederaufsetzen der Kommunikation
  - ↪ Vergabe von Senderechten

## (6) Darstellungsschicht

↪ *presentation layer, P*

- Sorgt für eine einheitliche Interpretation der Daten unabhängig von der Darstellung in den einzelnen Rechnern
  - ↪ Darstellungskontext & Transfersyntax

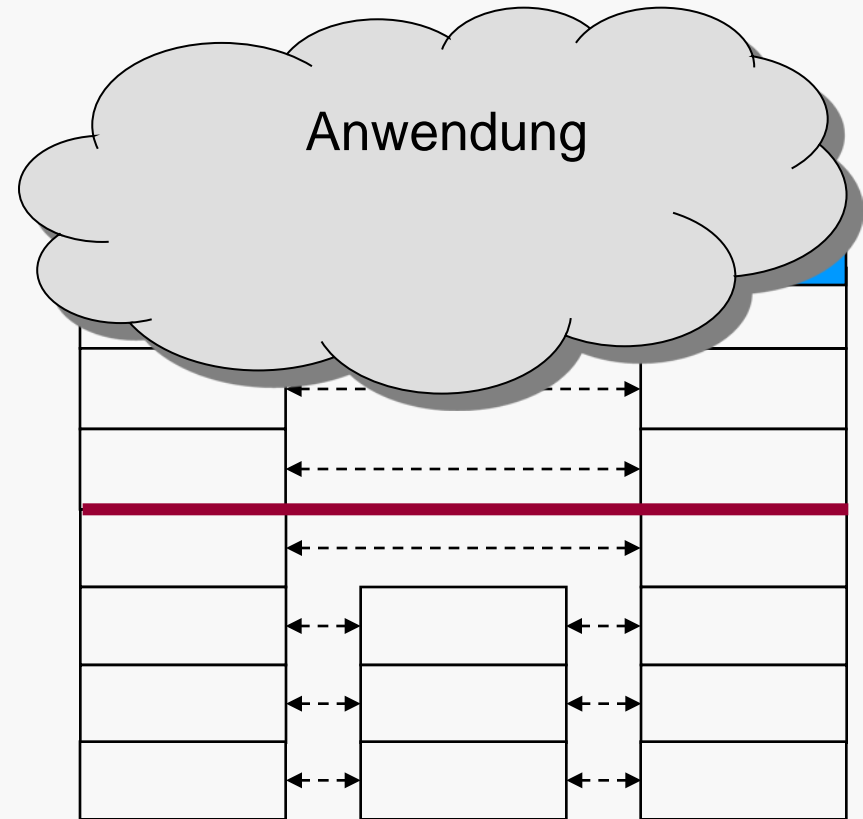


# OSI-Schicht 7

## (7) Anwendungsschicht

↪ *application layer, A*

- Stellt dem Nutzer die Dienste zur Verfügung, die er in seine Anwendung einbindet
- Für die Anwendungsschicht wurde ein **eigenes Modell** definiert, das sich von dem der anderen Schichten unterscheidet.
  - ↪ fließender Übergang zur Anwendung
  - ↪ z. B. keine Dienstzugangspunkte
- ☞ **Hat sich nicht durchgesetzt !!!**



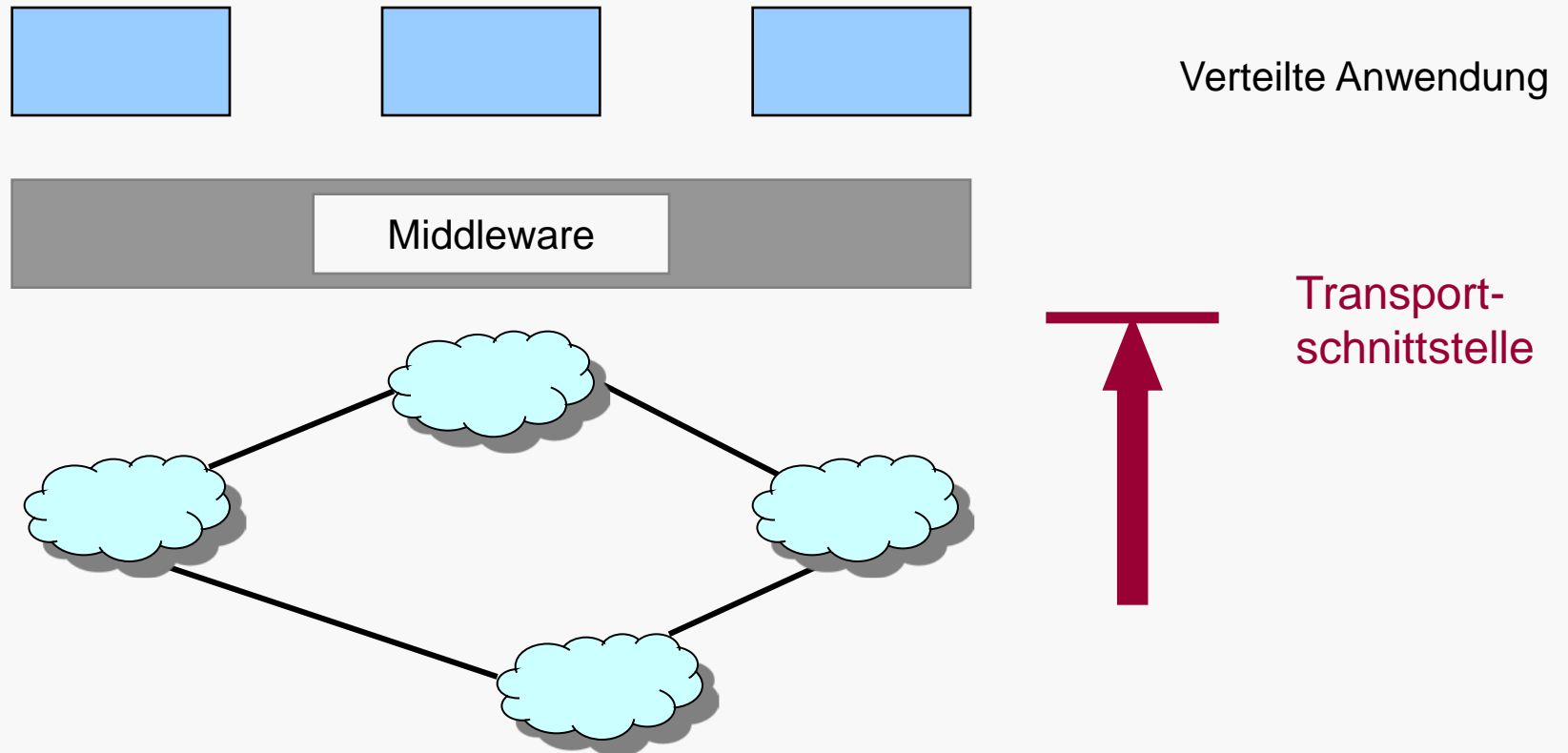
---

# Middleware

Es hat sich gezeigt, dass die Funktionen der Schichten 5-7 in ihrer ganzen Komplexität nicht immer benötigt werden.

- Schichten wurden von vielen Anwendungen nicht genutzt
- Verlagerung der Funktionalität in die so genannte **Middleware-Plattformen**
  - ↪ weniger komplex
  - ↪ stärker anwendungsbezogen
- **Beispiele:**
  - DCE (*Distributed Computing Environment*)
  - CORBA
  - Web Services

# Middleware-Plattformen



---

## B) Wertung des OSI-Ansatzes



---

# Wertung des OSI/RM (1)

Das OSI/RM in seiner ursprünglichen Form hat sich nicht durchgesetzt !!!

## ● Gründe:

- Modell in sich zu komplex
  - ↳ z.B. haben sich die höheren Schichten als nicht zweckmäßig erwiesen
- späte Protokollentwicklung
  - ↳ nach der Modelldefinition
- lange Standardisierung
  - ↳ späte Verfügbarkeit kommerzieller Produkte
- neue Anforderungen durch neue Entwicklungen
  - ↳ Hochleistungskommunikation
  - ↳ Middleware-Plattformen

---

# Wertung des OSI/RM (2)

## ● Was bleibt ???

- Systematik des Ansatzes
- Vielzahl konzeptioneller Beiträge
  - ↪ z.B. Dienstzugangspunkte, Verbindungen, Zeitablaufdiagramme u.v.a.
- Architekturkonzepte
  - ↪ Dienste
  - ↪ Protokolle
  - ↪ Schichten u. a.

 **OSI/RM wird als methodisches Bezugsmodell genutzt !!!**



---

# II.4.4.2

## TCP/IP-Architektur



König 4.3.2  
Tanenbaum/Wetherall 1.4.2+4

---

# TCP/IP-Architektur

Die TCP/IP-Architektur ist die Schichtenarchitektur des Internet.



**Kein Referenzmodell !!!**

- Beschreibt weniger ein allgemeingültiges Rahmenwerk für die Gestaltung von Schichtenarchitekturen, sondern vielmehr das konkrete Zusammenwirken der Kernprotokolle mit den anderen Protokollen im Internet
- Keine Architekturfestlegungen
- Anwendungs- bzw. Netzzugangsschicht sind keine richtigen Schichten !!!



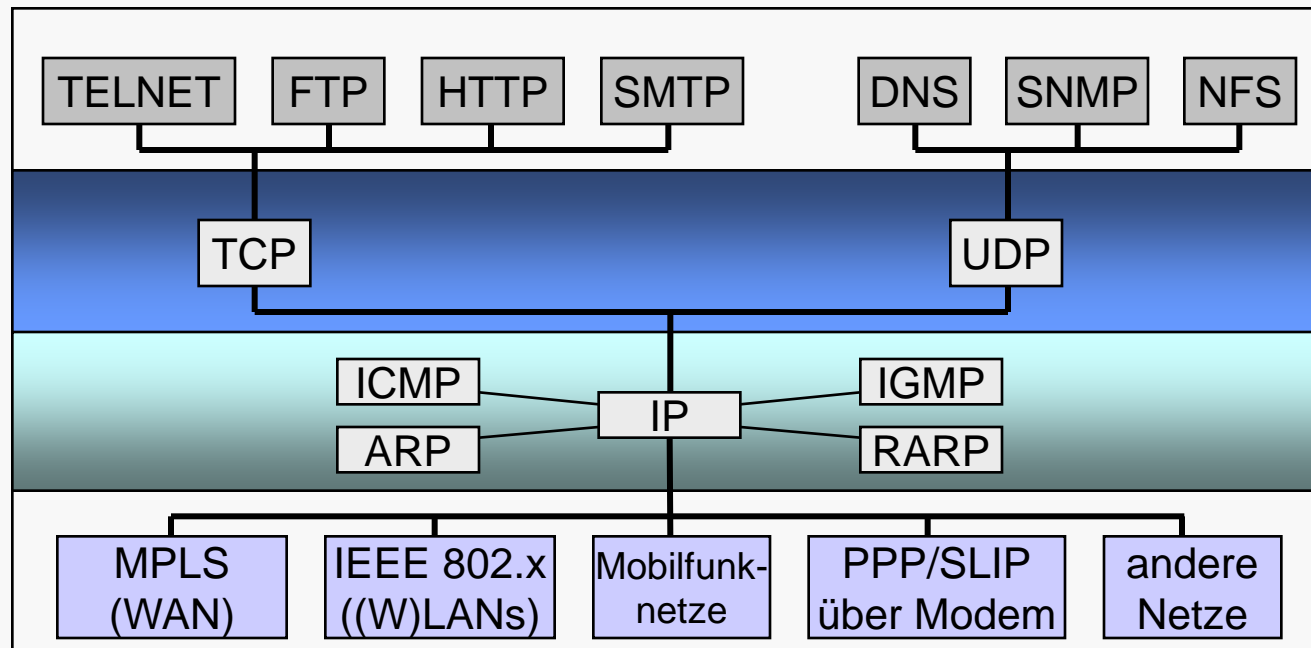
**Primär eine Protokollarchitektur !!!**



Protokolle TCP und IP waren Ausgangspunkt der Entwicklung

# TCP/IP-Architektur

(im Vergleich mit dem OSI-Referenzmodell)



## OSI/RM

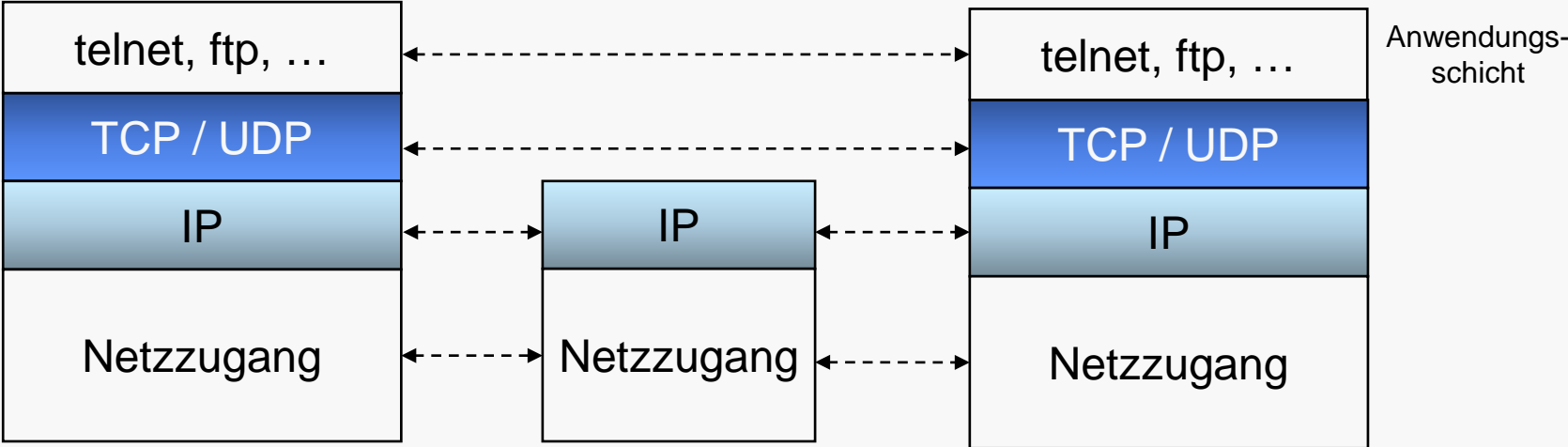
Anwendungs-  
schicht

Transport-  
schicht

Vermittlungs-  
schicht

Datensicherungs-/  
physikalische  
Schicht

# TCP/IP-Architektur



---

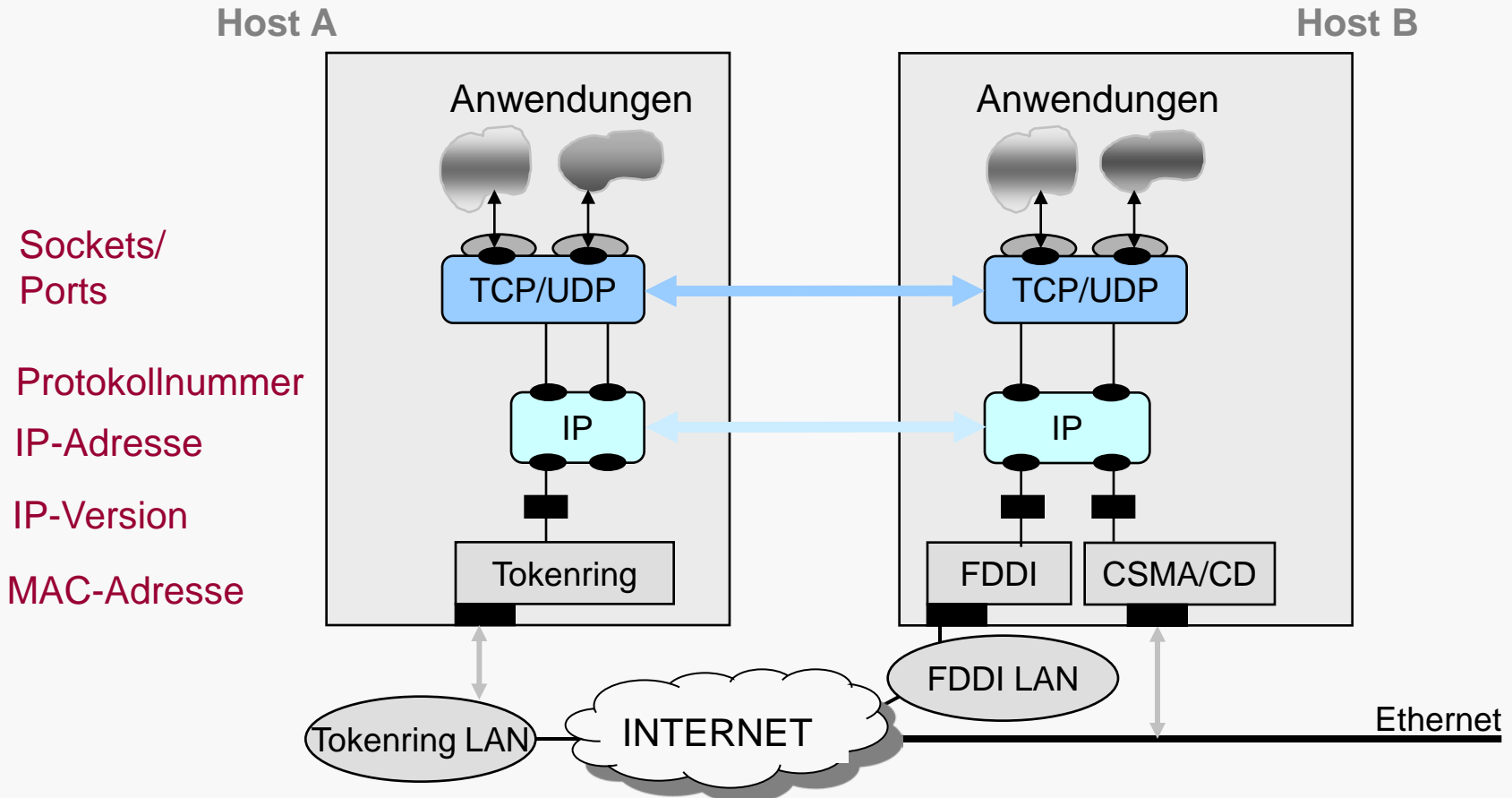
# Beispiel: Interaktionspunkte

Als Beispiel, um den Unterschied zwischen einer Kommunikationsarchitektur und einer Protokollarchitektur zu erläutern, sei hier die Gestaltung der Interaktionspunkte betrachtet. Die TCP/IP-Architektur kennt das Konzept des Dienstzugangspunkts wie im OSI nicht.

- Vielzahl unterschiedlicher Formen von Interaktionspunkten

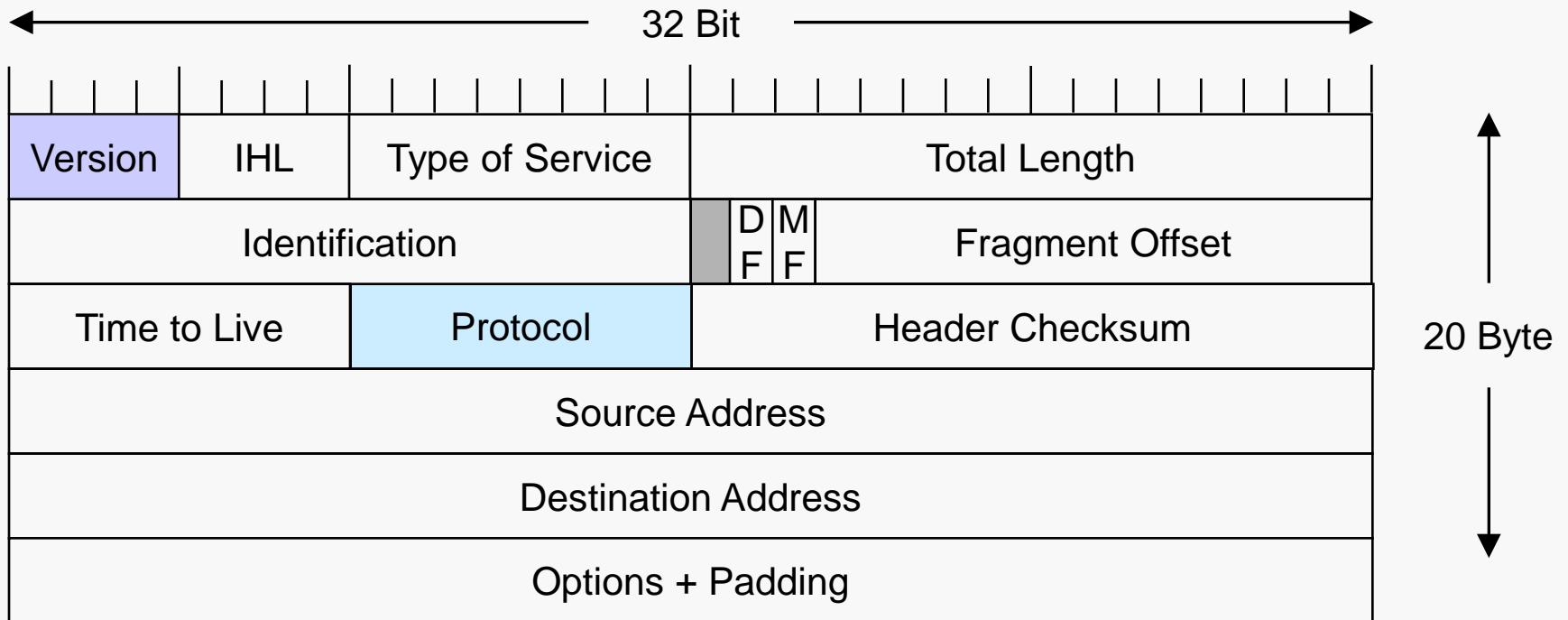
- **Sockets + Ports:** Zugangspunkte der Transportschnittstelle für Anwendung
- **Protokollnummer:** Zuweisung des „IP-Dienstnutzers“
- **Versionsnummer:** Auswahl des Protokollstacks
- **IP-Adresse:** Identifikation des Hosts (Zugang IP-Schicht)  
(mehrere Adressen möglich)
- **MAC-Adresse:** LAN-Zugang

# Interaktionspunkte in der TCP/IP-Architektur<sup>1</sup>



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

# Struktur des IPv4-Headers



---

# Unterschiede zur OSI-Architektur

- Kein Dienstbegriff
- Kein starrer Schichtenbegriff
  - ↪ vorrangig nur ordnend
- Zugriff auf tiefere Schichten möglich
- Mehrere Protokolle in einer Schicht möglich
- Verbindungslose und verbindungsorientierte Kommunikation
- Separate Protokolle für Datenübertragung und Verbindungsverwaltung möglich
  - ↪ *out of band*-Signalisierung
- Unterstützung Internetworking