

Rechnernetze

Organisation

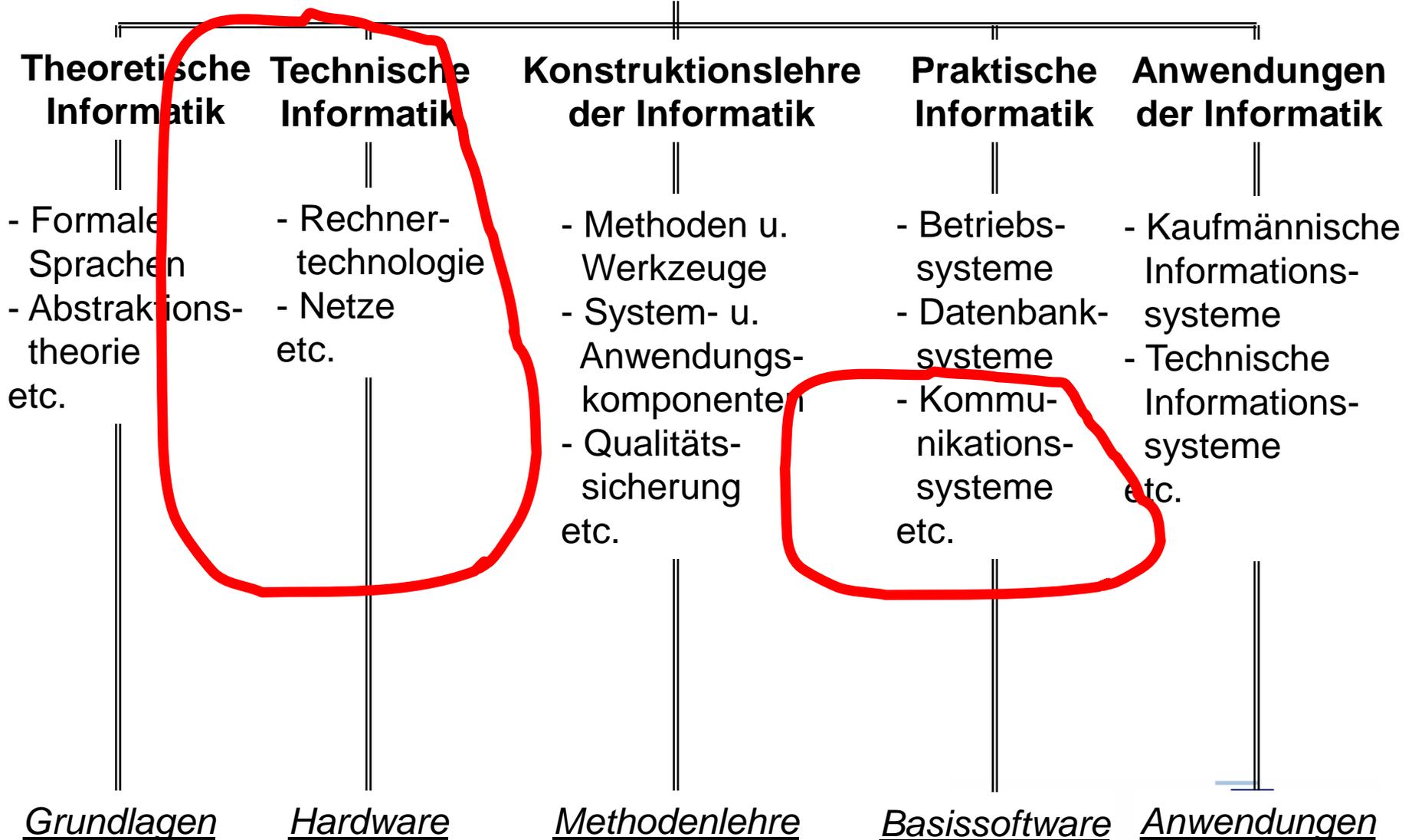
AG Internet-Technologien

Prof. Dr. Thomas Schmidt

t.schmidt@haw-hamburg.de



Informatik



Anforderungen

- Betriebssysteme (BS/BSP)
da in diesen Umgebungen die
Kommunikationssysteme arbeiten
- Programmierung, insbesondere C
- Algorithmen und Datenstrukturen



Vorlesungszyklus des Fachgebiets

Eigene
Werke

Bachelor Projekt

WPs
&
POs

Internet of Things

Webarchitekturen

Verteilte Software

Internettechnologien

Verteilte Systeme

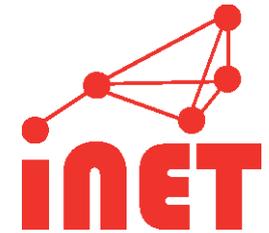
Grundlagen

Rechnernetze

Betriebssysteme



Arbeitsgruppe Internet-Technologien



o Website:

inet.haw-hamburg.de

o Sie finden uns in:

- Raum 580 & Räumen 480/1

o Messen & Ausstellungen:

- CeBIT, Emb. World, NdW

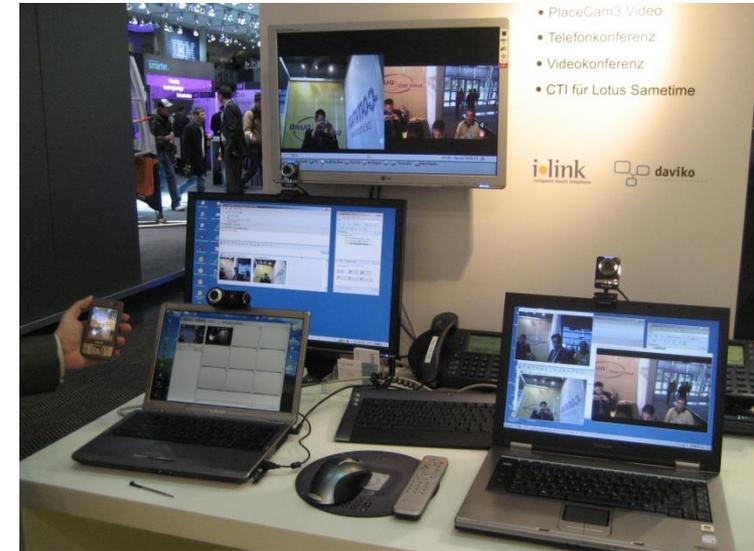
o Auslandskooperationen

- Wir vermitteln gerne



AG INET: Wer wir sind?

- ▶ ~ 20 aktive Forscher & Entwickler
- ▶ Arbeitsschwerpunkte
 - ▶ Internet of Things
 - ▶ Future Internet
 - ▶ Internet Sicherheit
 - ▶ Internet-Messungen
 - ▶ Skalierbare Software-Systeme
 - ▶ Open Source Software
- ▶ Langjährige Standardisierung (IETF)



Aufbau der Vorlesung

1. Einführung: Kommunikation, Protokolle, Modelle
2. Das Internet: Modell, Architektur, Dienste
3. Grundlagen der Netzwerktechnologien
4. Internet Protokoll: IPv4 & IPv6
5. Internet Routing
6. Transportprotokolle, Socket API
7. Wireless Networking
8. Multicast Gruppenkommunikation
9. Netzwerkmanagement
10. Netzwerksicherheit



Praktikum / PVL

► 4 Praktikumsaufgaben (**Bearbeitung in 2-er Gruppen**)

1. Pakete aufbauen und ersniffen
2. IP Routing
3. Socket Programmierung
4. Netzwerkmanagement

► **PVL-Bedingungen**

- Erfolgreiche Bearbeitung **aller** Aufgaben: siehe Aufgabenstellungen
- **Erfordert vorbereitende Arbeit** – Lösungsweg vorab planen!
- **Abgabeleistung bis Freitag nach dem Praktikum (Mail):**
 - **Protokoll:** Sprachlich ausgearbeitete Dokumentation (PDF), alle **Beobachtungen** müssen **erläutert** sein, alle **Behauptungen** müssen **begründet** sein, alle **Quellen** müssen **zitiert** sein.
 - **Code:** bei Programmieraufgaben



Anmerkung zu den Folien

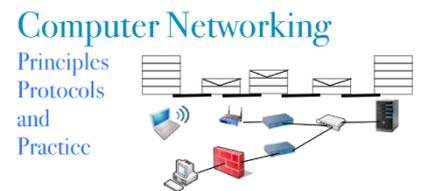
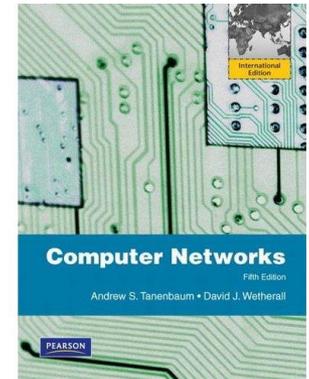
- ▶ Die Folien enthalten alle Themen und Gegenstände der Vorlesung
 - ▶ Themen sind nummeriert (in Übersicht und Folien)
- ▶ Aber sie sind kein Buch - Sie enthalten *nicht*:
 - ▶ ausführliche Herleitungen
 - ▶ die Gedanken & Diskussionen, die in der Vorlesung entstehen
- ▶ Deshalb sollten Sie:
 - ▶ eigene Notizen / Ergänzungen machen
 - ▶ in Bücher schauen.



Literatur zur Vorlesung

Diese Vorlesung orientiert sich unmittelbar an:

- ▶ Andrew Tanenbaum, David Wetherall: *Computer Networks*, 5th Ed., Pearson 2011 (auf Deutsch 2012 erschienen)
- ▶ Ch. Meinel, H. Sack: *Internetworking*, Springer 2012
- ▶ Olivier Bonaventure: *Computer Networking*, 2nd Edition, 2018
<http://cnp3book.info.ucl.ac.be/secondedition.html>



Weitere Literatur

- ▶ R. Stevens/(R.Wright): *TCP/IP Illustrated*, Vol 1-3, Addison-Wesley, 1994/1995. (*die 'alte Bibel'*)
- ▶ K. Fall/R. Stevens: *TCP/IP Illustrated*, Vol 1, Addison-Wesley, 2nd ed. 2011. (*Aktualisierte Bibel*)
- ▶ J. Kurose, K. Ross: *Computernetzwerke*, 7th ed., Pearson Education 2017.
- ▶ W. Stallings: *Data and Computer Communications*, 10th Pearson Education, 2013.



Nicht nur zum Scheine ...

... machen ist die Hochschule da.

- Selbständig Vor- und Nachbereiten
- Über Zusammenhänge nachdenken und diskutieren
- Aktiv in Projekten mitarbeiten

Auswendig gelerntes
Lexikonwissen hilft nicht weiter!



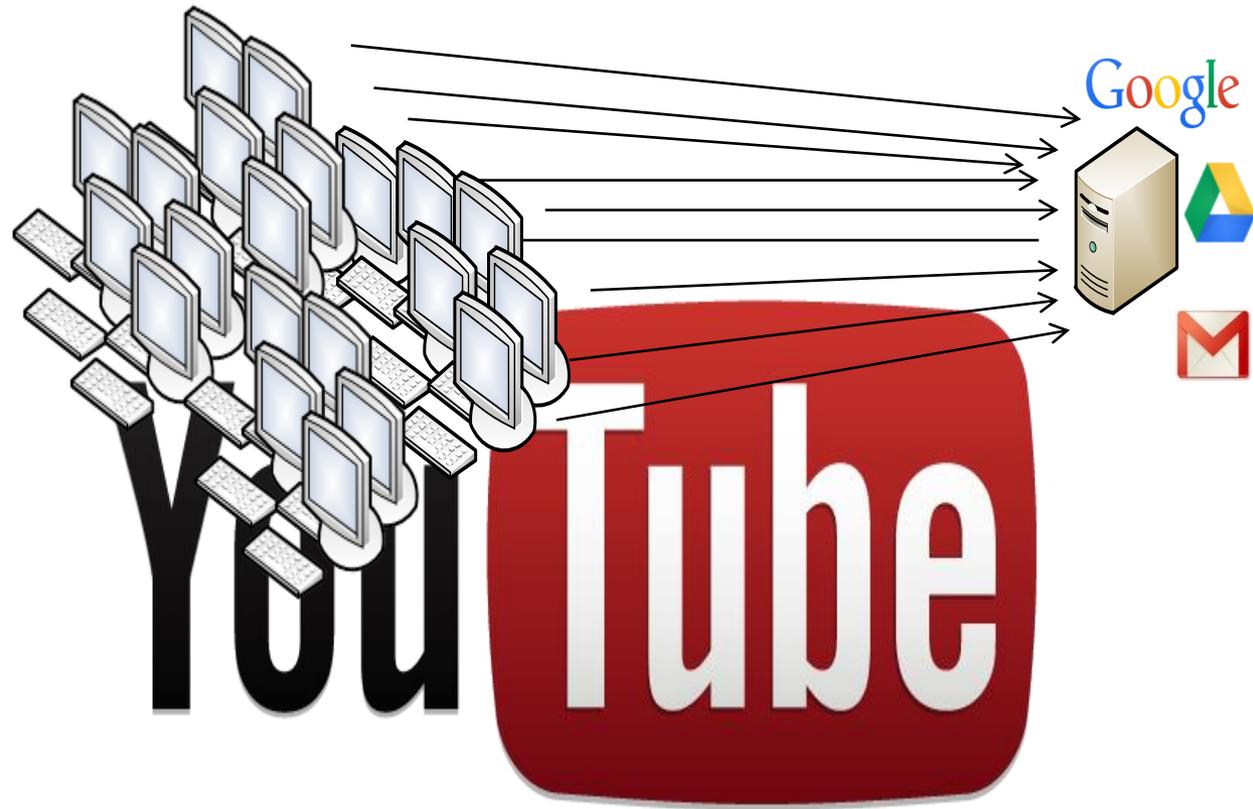
Das Nichtverstehen verstehen



Wir wollen mit Ihnen mehr erreichen



Motivation ?

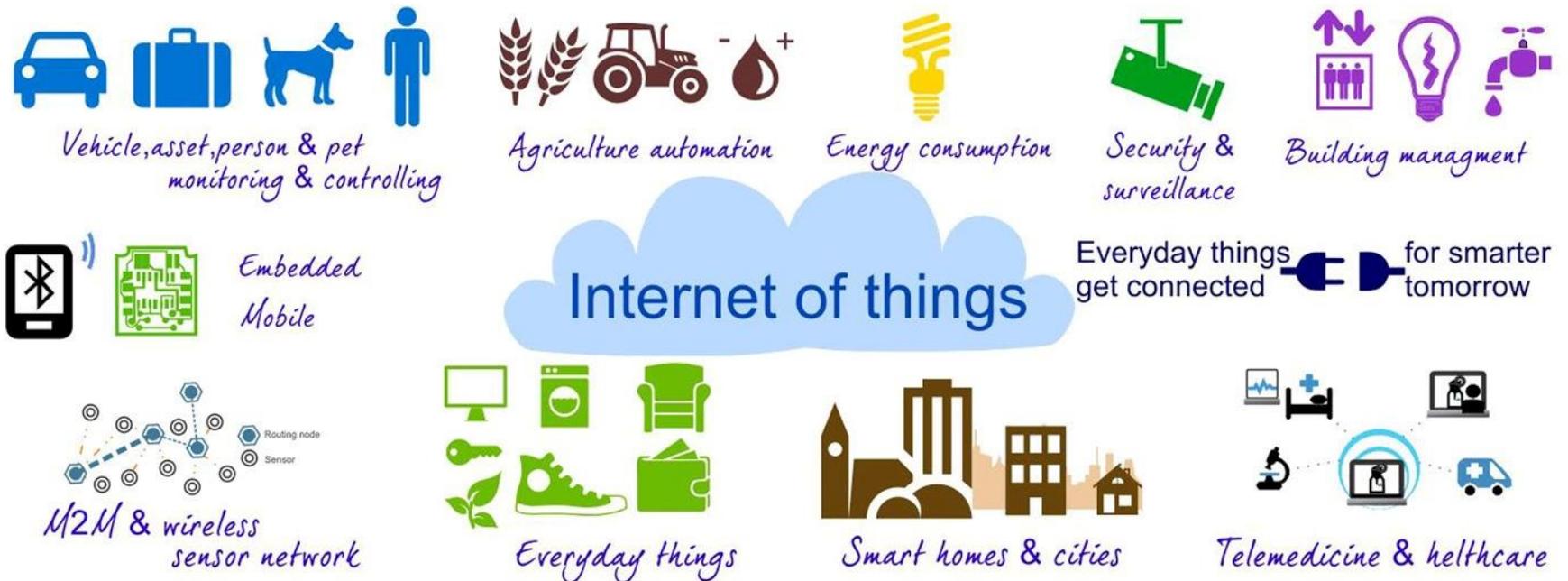


Vieles ist komplizierter als gedacht:

- Wie fließen die Daten?
- Wie greift man auf viele entfernte Dienste gleichzeitig zu?
- Wie macht man Systeme skalierbar?



Plötzlich eine neue Welt



Quelle: The Telecare Blog, thetelecareblog.blogspot.de, 24.10.14



Das IoT selbst entwickeln



- ▶ Das Internet of Things sind viele, unzuverlässige Knoten, die ‚schlecht sprechen und hören‘ können
- ▶ Ein neues Betriebssystem für das IoT: RIOT-OS.org
- ▶ Wie kann man hierfür verteilt programmieren?
Welche Programmiermodelle taugen?
- ▶ Reinschnuppern?
 - ▶ Hack`n`Ack an jedem letzten Dienstag im Monat
 - ▶ 17:00 Uhr in Raum 460



Schlüsselkompetenzen

Weltwirtschaftsgipfel Davos 2018:

1. Komplexe Problemlösungen
2. Kritisches Denken
3. Kreativität



Lernziele (klausurrelevant)

- Beschreibe Kernkonzepte in eigenen Worten
- Vergleiche und bewerte Verfahren begründet
- Wähle praktikable Lösungen aus und wende sie an, erkenne, was nicht funktioniert
- Analysiere das Verhalten von Zusammenhängen, identifiziere und isoliere Fehler
- Diskutiere Probleme und erschließe den Lösungsraum mit überprüfbaren Argumenten



Rechnernetze

Begriffe - Modelle - Protokolle

1. Aufgaben und Arten von Rechnernetzen
2. Grundeigenschaften der Kommunikation
3. Protokolle
4. OSI- und DoD-Modell
5. Standardisierung



Zum Inhalt

In diesem Kapitel lernen Sie die grundlegenden Aufgaben und Arten von Rechnernetzen kennen und sollen eine Vorstellung erhalten, wie Kommunikation im Netz stattfinden kann und wie sie mithilfe von Protokollen aufgebaut wird. Die (abstrakten) Modelle des modernen Protokollaufbaus begegnen Ihnen hier gemeinsam mit dem Kommunikationsablauf zwischen ihren Protokollschichten.

Das dazugehörige Kapitel im Tanenbaum ist 1, im Meinel/Sack sind es die Kapitel 1 und 2.



1. Aufgaben von Rechnernetzen

Lastverbund

- ▶ Verteilung von Aufgaben an mehrere Rechner
- ▶ Beseitigung von Engpässen / Nutzung freier Ressourcen

Leistungsverbund

- ▶ Zusammenarbeit von Rechnern verschiedener Funktionalitäten
- ▶ Virtuelle Universalmaschine

Verfügbarkeitsverbund

- ▶ Redundanz durch Systemdoppelung
- ▶ Problem: Datenredundanz



1. Aufgaben von Rechnernetzen (2)

Funktionsverbund

- ▶ Geteilte Ressourcennutzung (Massenspeicher, Software,...)
- ▶ Virtualisierte Funktionsumgebung (e.g. Walking User Support)

Datenverbund

- ▶ Gemeinsamer Zugriff auf Datenbestände (e.g. DB-Anwendungen)
- ▶ Redundanzfreie, ortsungebundene Datenhaltung

Nachrichtenverbund

- ▶ Austausch von Nachrichten / Kommunikation
- ▶ Ortsungebundene Erreichbarkeit

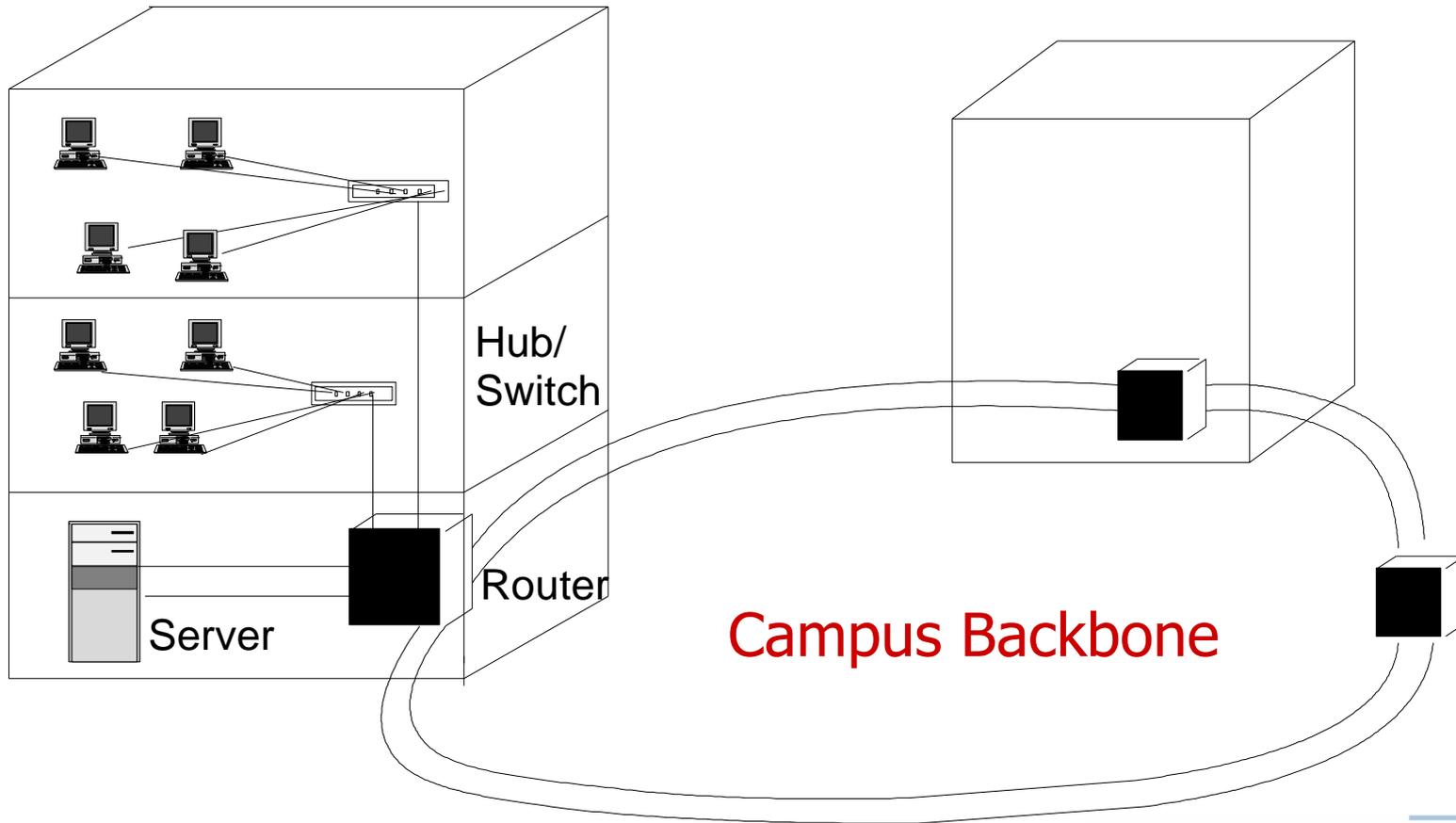


1. Klassifizierung von Rechnernetzen

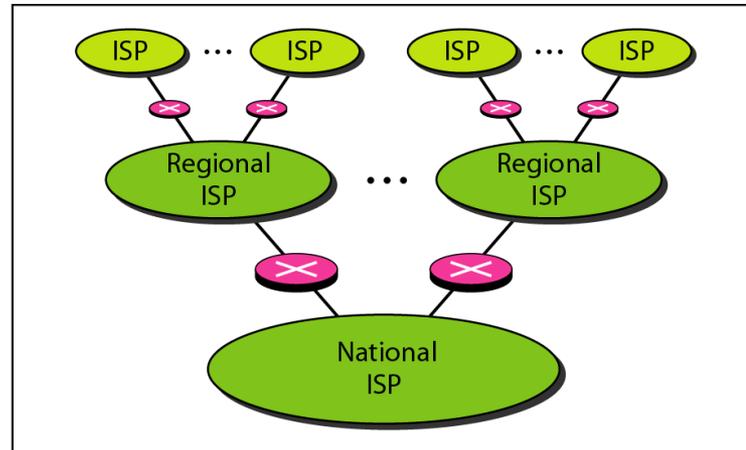
- GAN** Global Area Network
Weltweit ausgedehnt
- WAN** Wide Area Network
Rechnernetze etwa auf Landesebene
- MAN** Metropolitan Area Network
Netze einer Stadt wie etwa Hamburg
- LAN** Local Area Network
Netze im Nahbereich (Häuser, Häusergruppen)
Besonderheiten: viele Endgeräte, gemeinsam genutzte Übertragungsmedien
- PAN** Personal Area Network
Kurzreichweitig innerhalb eines Raumes



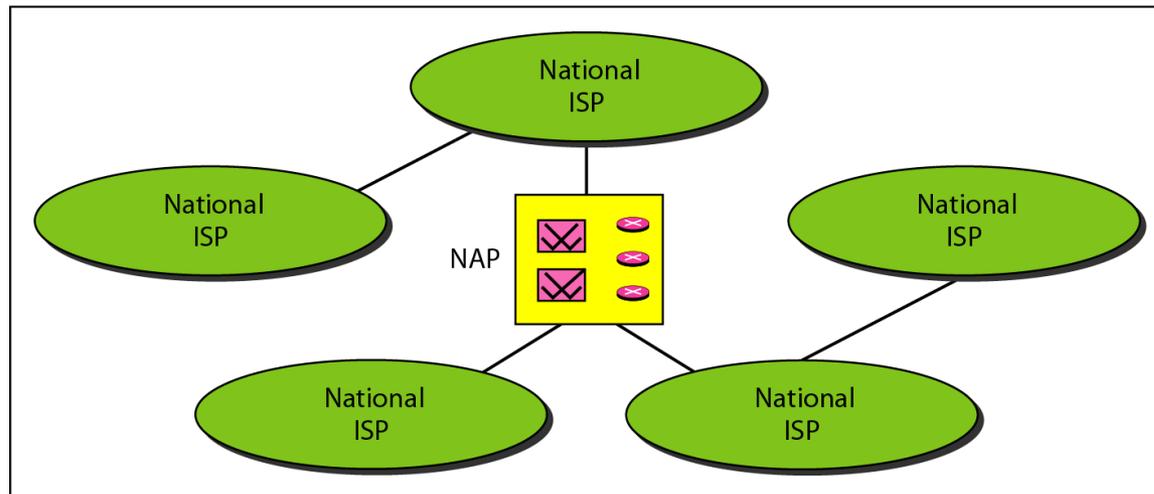
1. Lokale Rechnernetze in der Praxis



1. Globales Rechnernetz (Internet)



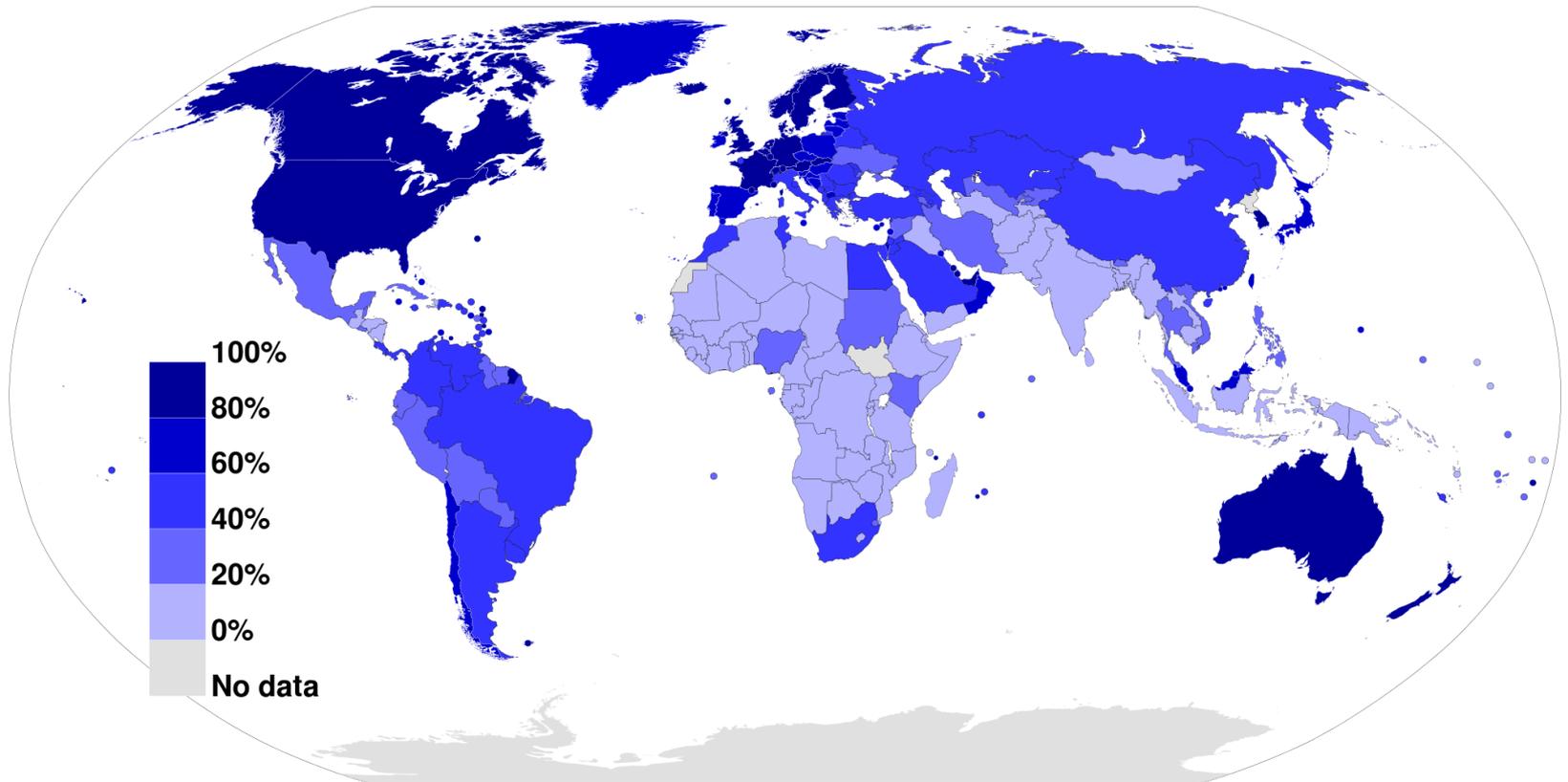
a. Structure of a national ISP



b. Interconnection of national ISPs



Das Internet ist (fast) überall



Internet Penetration Rate by Region (source ITU, Jeff Ogden)



Achtung!

Das Internet ist nicht das World Wide Web, facebook oder Google.
Das Internet ermöglicht Services wie das World Wide Web **on top**.

2. Kommunikation

- ▶ Die Aufgabe von Rechnernetzen ist es, Kommunikation zwischen den Teilnehmern zu ermöglichen
- ▶ Solche Kommunikation kann sehr vielfältig sein:
 - ▶ Prozesse können einander **Nachrichten schicken**
 - ▶ Rechner (Betriebssysteme) können **Aufträge** an einen Dienst **vergeben** und eine Antwort erwarten
 - ▶ Mitglieder einer **Gruppe** können ihren Informationsstand **synchronisieren**
 - ▶ ...



2. Kommunikationsmodi

Synchron

- ▶ Gemeinsame Aktion von Sender und Empfänger
- ▶ Erfordert (Warten auf) Kommunikationsbereitschaft aller Partner
- ▶ Bsp: Telefonie, Terminalsitzung, Videokonferenz

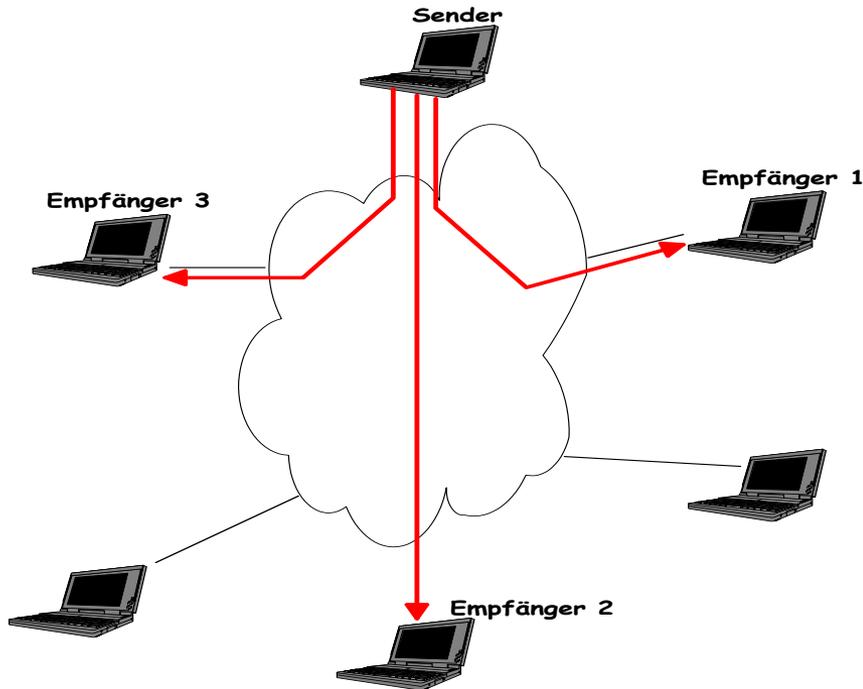
Asynchron

- ▶ Sender und Empfänger operieren losgelöst voneinander
- ▶ Erfordert Puffermechanismen
- ▶ Bsp: SMS, eMail, Instant Messaging

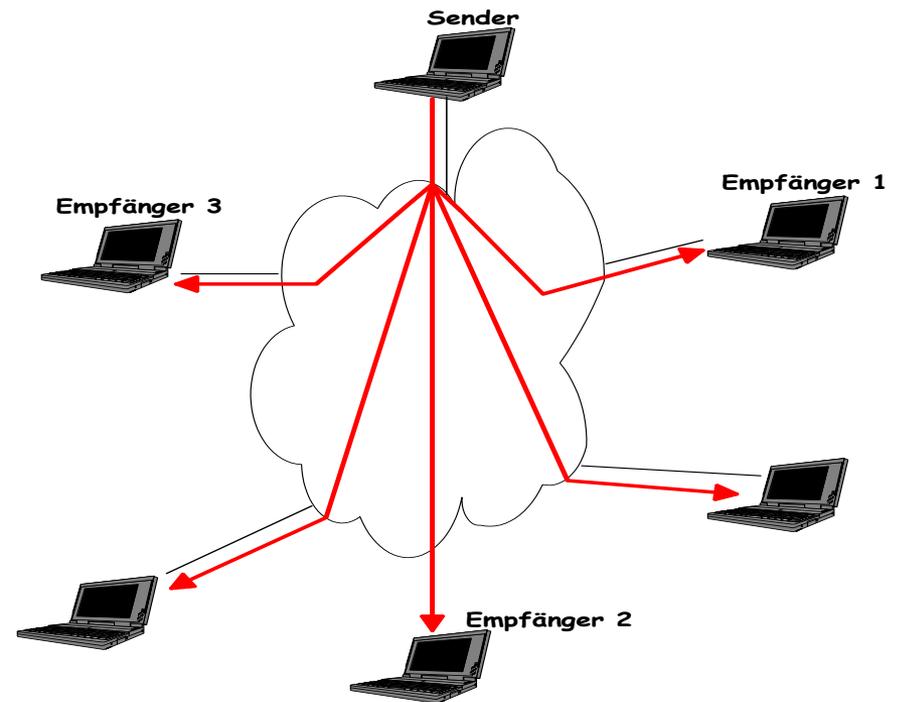


2. Kommunikationsformen

Unicast

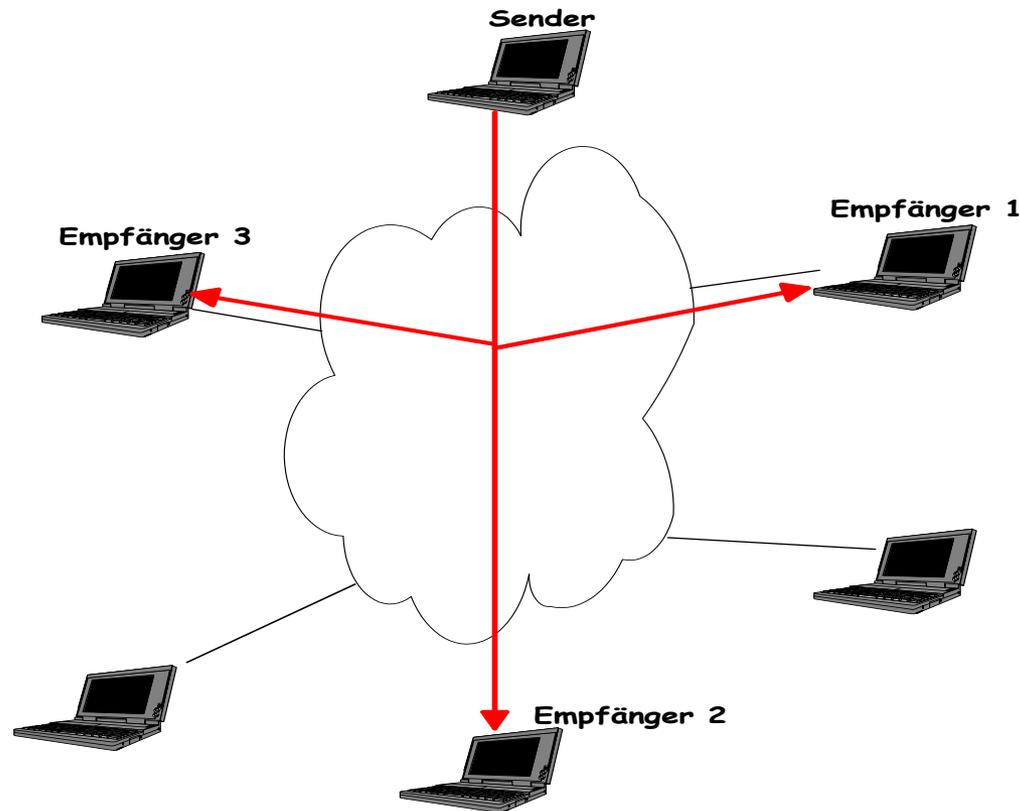


Broadcast



2. Spezifische Gruppenkommunikation

Multicast



Servergruppe: Anycast



2. Grundtypen von Netzwerken

➤ Punkt-zu-Punkt Netzwerke

Netzwerk zwischen zwei Vermittlungsknoten über dedizierte Leitungen

Bsp: Mietleitungen, Telefon, Richtfunk, serielle Leitung

➤ Broadcast Netzwerke

Netzwerk zwischen vielen Hosts/Gateways über geteilte Leitungen

Bsp: Typische LAN-Technologien, WLAN



2. Dienste

Wohldefinierte, allgemein benötigte Funktionen

- ▶ ausgelagertes Leistungspaket beim Dienstgeber (Server)
- ▶ Bestandteile: Dienstfunktion, Dienstprimitiven, Dienstprozeduren
- ▶ Inanspruchnahme durch Dienstnehmer (Client)

Dienstgüte

- ▶ Angemessenheit/Zugänglichkeit
- ▶ Technische Leistung: Antwortzeit, Genauigkeit, ...
- ▶ Kosten
- ▶ Zuverlässigkeit
- ▶ Sicherheit/Vertraulichkeit



2. Dienstmodelle

Client-Server Modell

- Rollenzuweisung: Server erbringt, Client erfragt einen Dienst
- Kommunikationsform: 1 Server : 1 + n Clients (einer mit vielen)
- Bsp: WWW, ftp, Mail (fast alle Internetdienste)

Peer-to-Peer Modell

- Aufgabenverteilung zwischen gleichartigen Teilnehmern
- Kommunikationsform: m : n (viele mit vielen)
- Bsp: Filesharing, Tauschbörsen, VCoIP



3. Protokolle

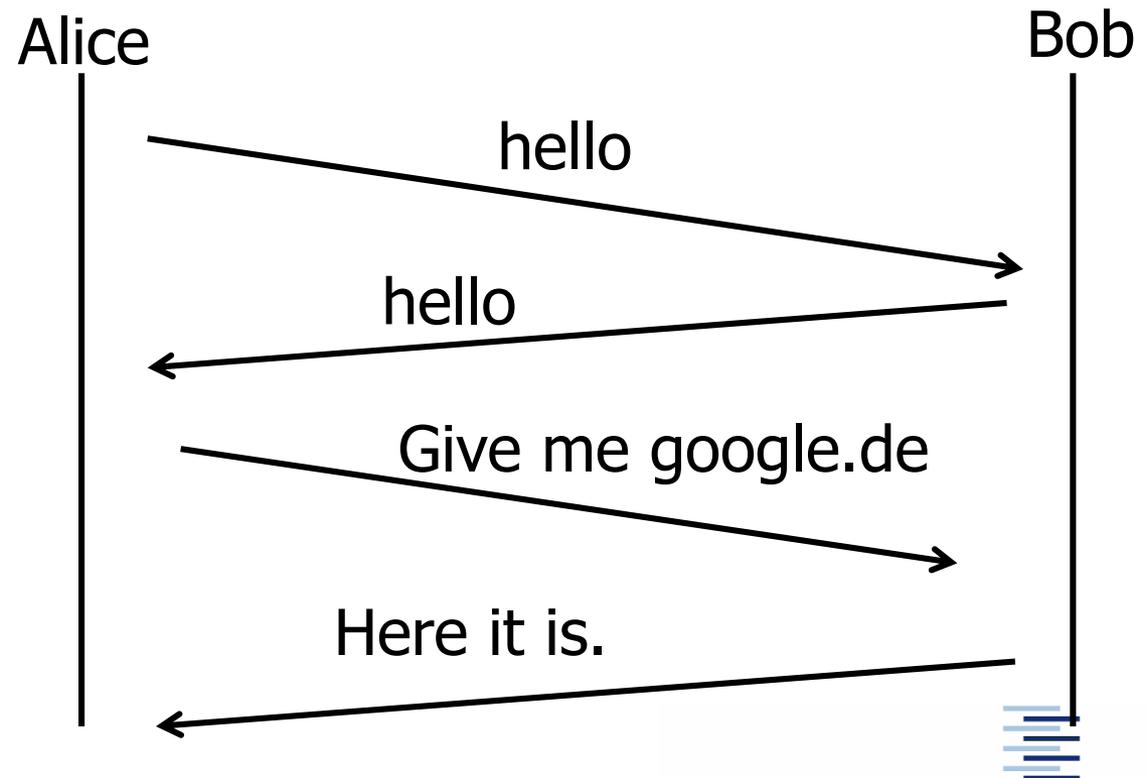
Zur Kommunikation untereinander benötigen Rechner gemeinsame ‚Sprachen‘, sogenannte Protokolle.

- Protokolle regeln den Datenaustausch zwischen Partnern
- Unterschiedliche Anforderungen/Kontexte führen zur Existenz vieler Protokolle
- Protokolle erbringen definierte Dienstleistungen gegenüber dem Nutzer / der übergeordneten Schicht
- Damit Protokolle universell einsetzbar sind, müssen sie in offenen Standards verabredet werden



Protokolle sind eine Gesprächskonvention

- Protokolle haben Syntax und Semantik



3. Aufgaben der Protokolle

Funktionsmechanismen höherer Kommunikations-Protokolle sind:

- Adressierung
- Einbettung von Daten (encapsulation)
- Segmentierung + Reassemblierung von Datenpaketen
- Fehlererkennung und -behebung
- Flußsteuerung (flow control)
- Verbindungskontrolle (connection control)



3. Verbindungskontrolle

Protokolle können Daten mit unterschiedlicher Zielsetzung übertragen. Deshalb sind Protokolle entweder

- **Verbindungsorientiert** (connection-oriented)
 - zustandsbehaftet, (gesichert)
 - drei ausgezeichnete Phasen zwischen Partnern:
Verbindungsaufbau - Datentransfer – Verbindungsabbau
 - Transfer zwischen beteiligten Partnern

oder

- **Verbindungslos** (connectionless)
 - zustandslos, ungesichert
 - Transfer zwischen unabhängigen Partnern



3. Zuverlässigkeit

Protokolle können unterschiedlich zuverlässig sein

► Zuverlässige Protokolle

- bieten Schutz vor Datenverlust/-zerstörung
- verifizieren Pakete nach Erhalt und quittieren
- haben Overhead, der verlangsamen kann

► Unzuverlässige Protokolle

- beachten Datenverluste nicht
- verifizieren und quittieren Pakete nicht
- Überprüfung und Korrektur kann in übergeordneten Schichten erfolgen

► k-zuverlässige Protokolle

- stellen sicher, dass k (aus $k+1$) Paketen zuverlässig ankommen



3. Flusskontrolle

Protokolle können den tatsächlichen Datenfluss an die Ressourcen von **Sender**, **Empfänger** und das **Netzwerk** anpassen, indem sie

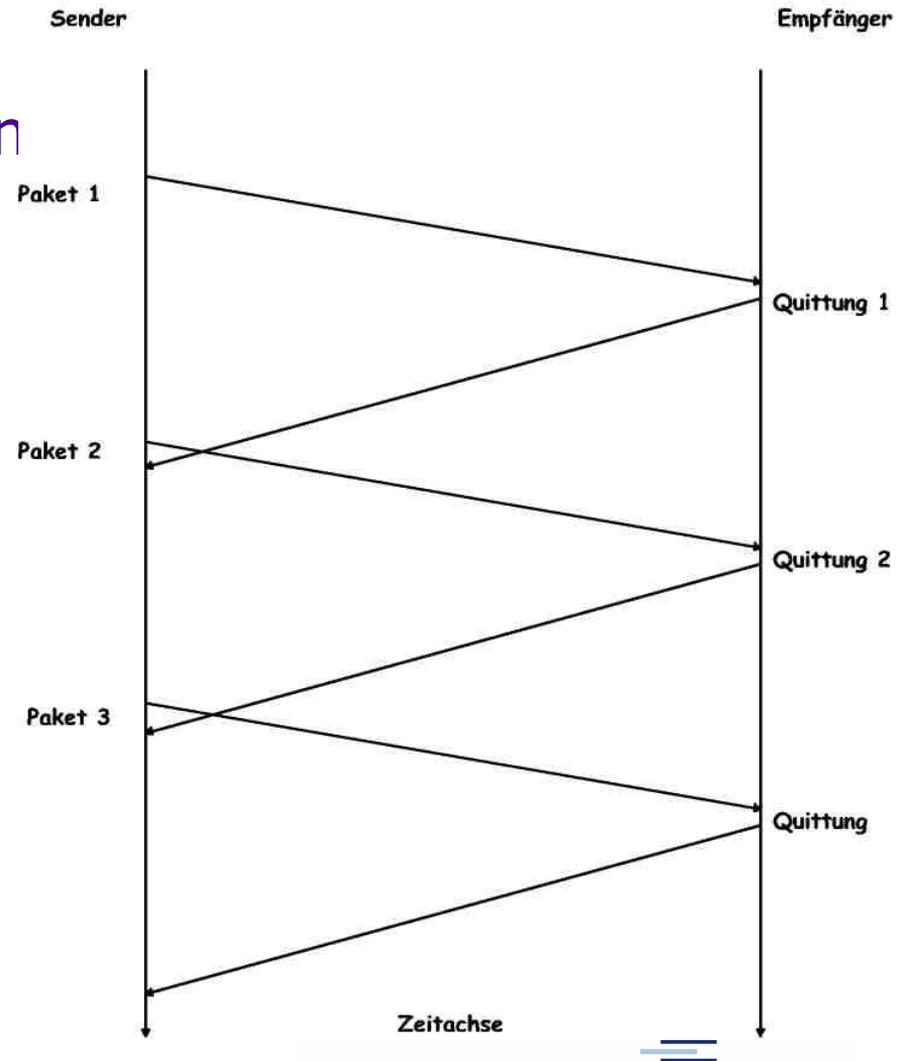
- Sende- und Empfangspuffer miteinander abgleichen
- Das Übertragungsverhalten im Netz messen und interpretieren
- Ihr Kommunikationsverhalten auf alle Leistungsgrößen einstellen



3. Realisierung in einer Verbindung

Empfänger sendet Quittungen

- Zustandsmeldungen (Verbindungskontrolle)
- Empfangsbestätigung (Sicherung)
- Bekanntgabe von Empfangspuffern (Flußkontrolle)



3. Das Kommunikationsproblem

- ▶ Heterogene Netzwerk-Infrastruktur
- ▶ Heterogene Rechnerarchitekturen
- ▶ Heterogene Anwendungslandschaft
- ▶ Verteilte Applikationen

Jeder 'Teilnehmer' des Netzes soll mit jedem anderen Teilnehmer sprechen können!



4. Der Lösungsansatz

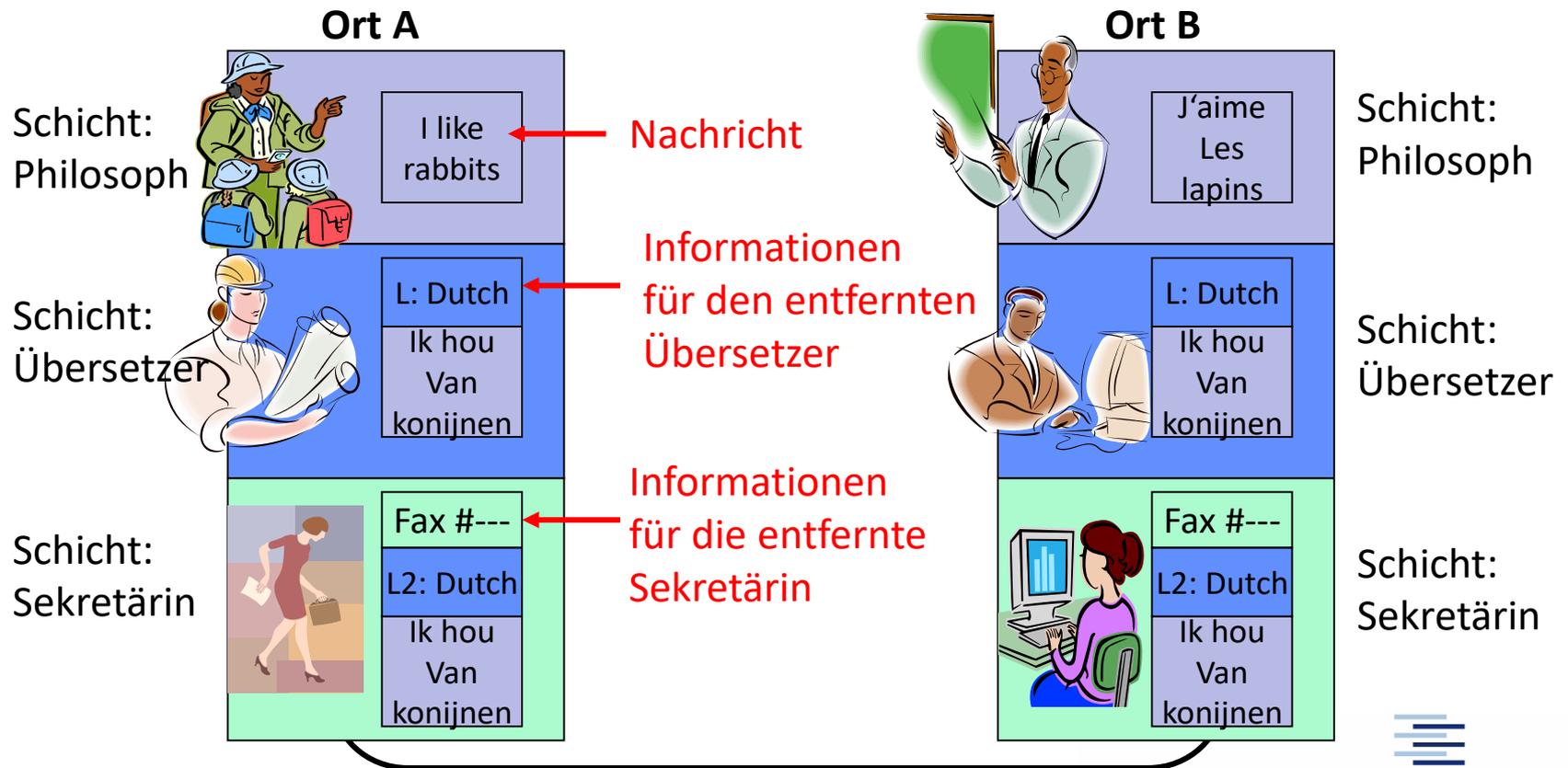
Für die Kommunikation in heterogenen, offenen Systemen ist eine konzeptionelle Gliederung der Funktionalitäten unerlässlich:

- Gliederung des Gesamtproblems in Teile (Ebenen)
- Jede Ebene löst einen Teil des Gesamtproblems
- Jede Ebene arbeitet mit den direkt benachbarten Ebenen zusammen
- Voraussetzung dafür sind kompatible Implementierungen (präzise definierte Schnittstellen)



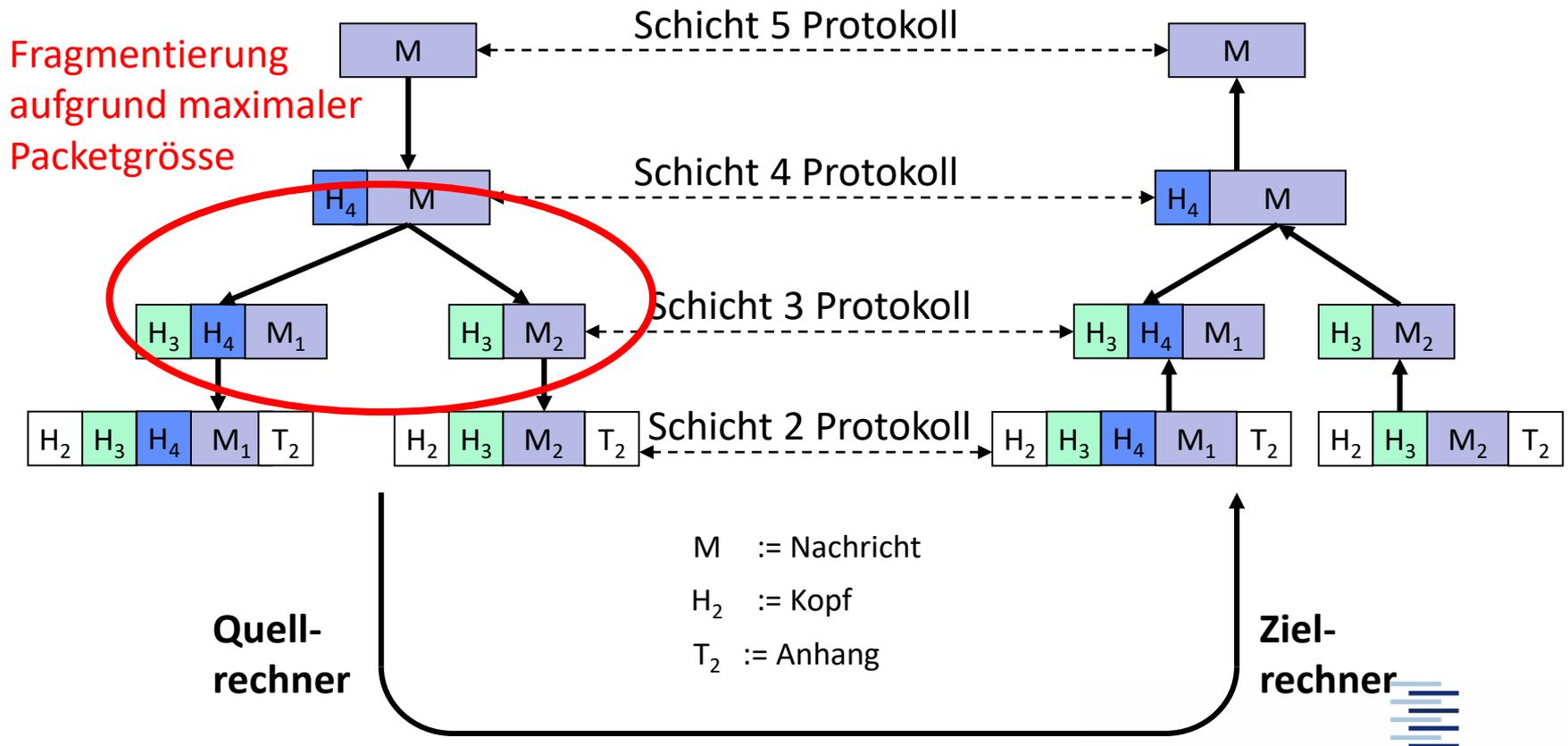
4. Protokollhierarchie – Schichtenmodell

Die “Philosoph-Übersetzer-Sekretärin“-Architektur



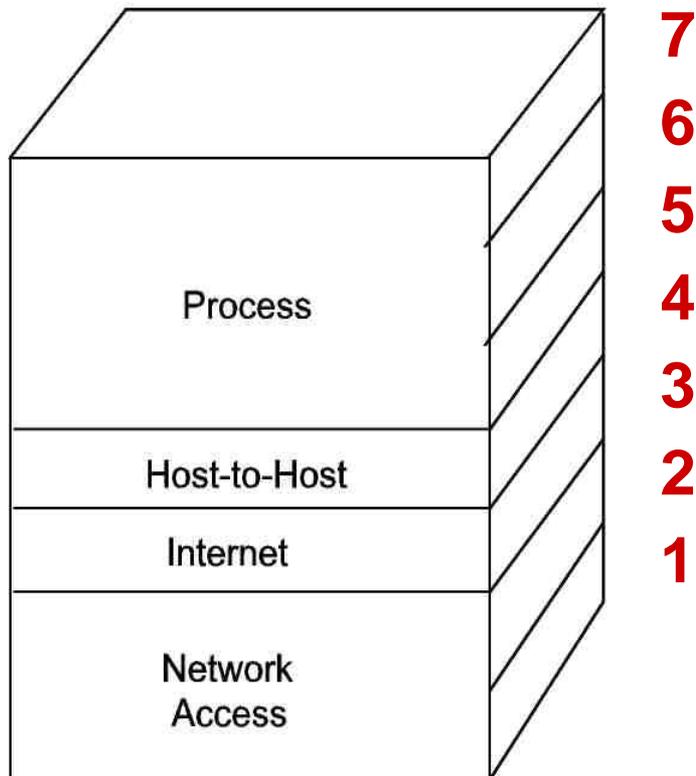
4. Protokollhierarchie – Schichtenmodell

Ein technischeres Beispiel...Header und Fragmentierung

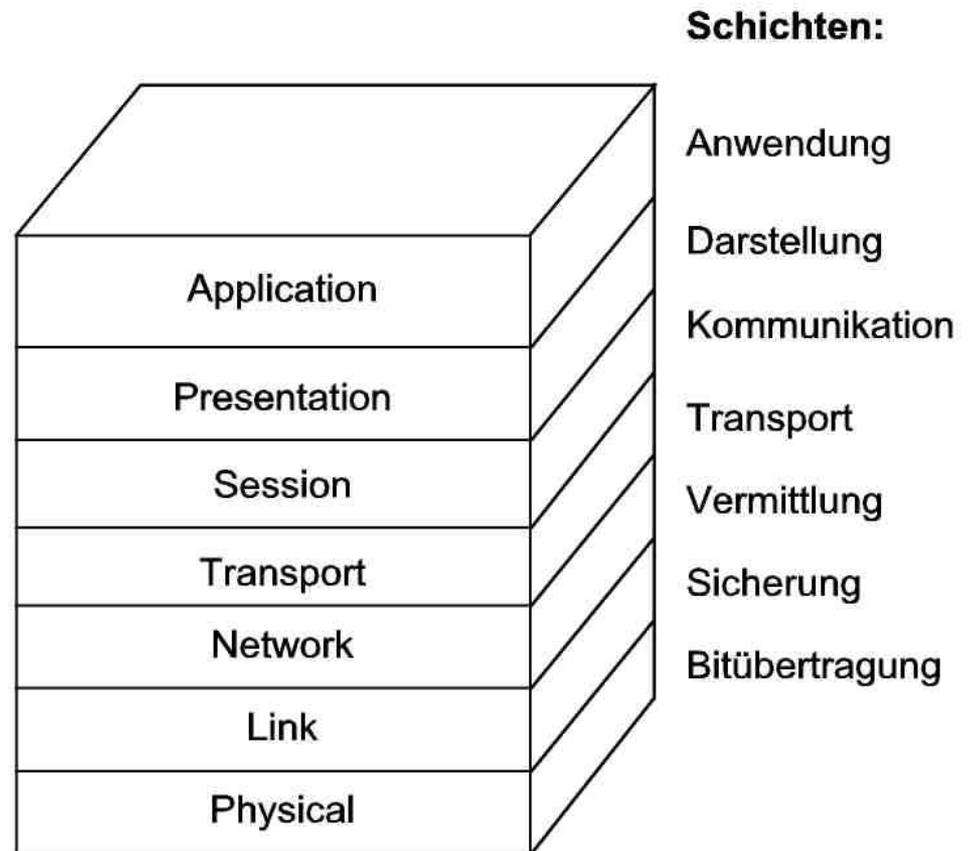


4. Referenzmodelle

DoD Internet Referenzmodell



ISO/OSI 7 Schichten Referenzmodell



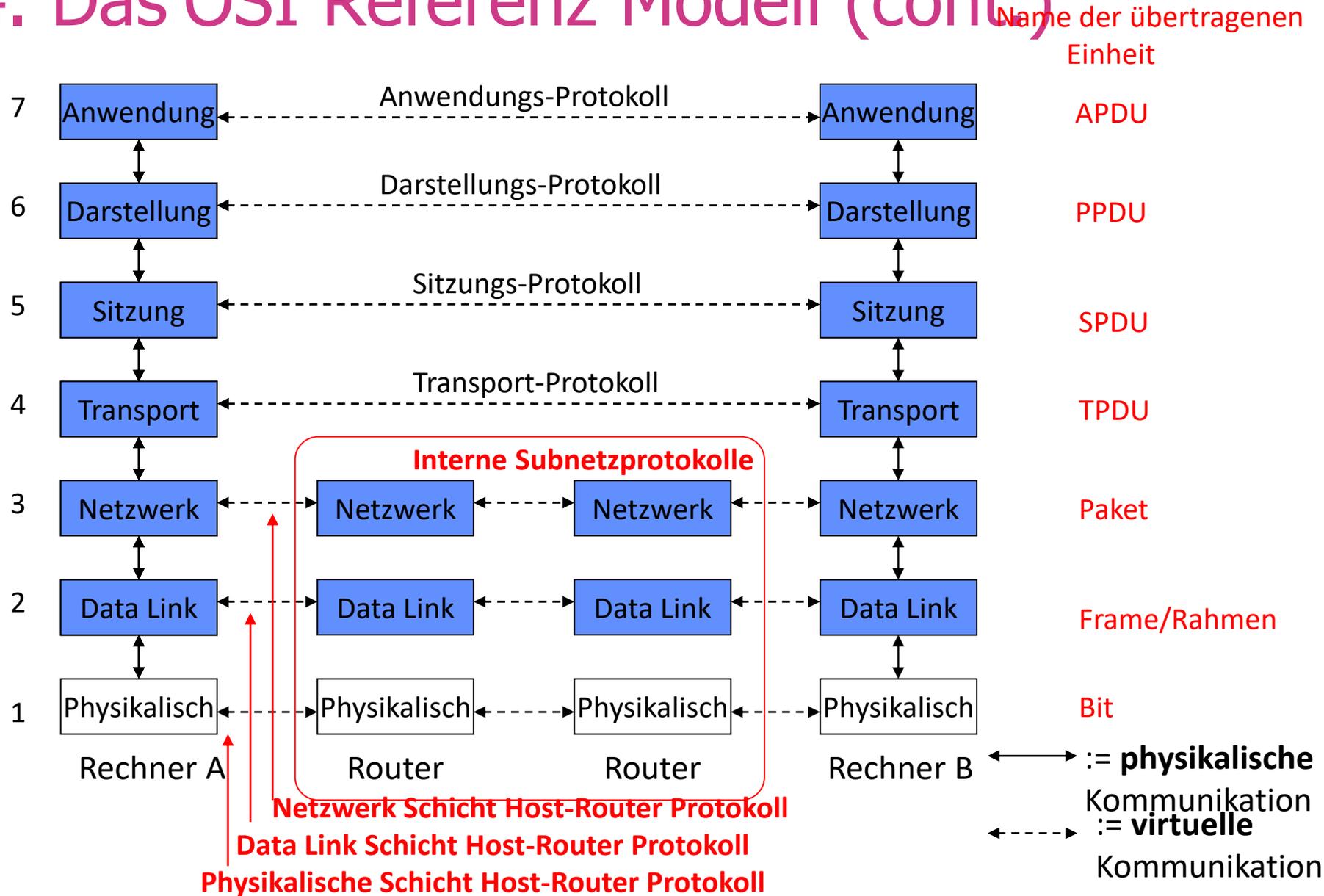
4. OSI-Modell

- ISO (International Organization of Standardization) beauftragte 1977 einen Unterausschuß mit der Entwicklung einer Kommunikationsarchitektur zwischen offenen Systemen
- Aufgabe des Modells:
 - Referenz zur Beschreibung von Protokollen und Funktionen
 - Standardisierungsgrundlage für OSI-Protokolle
 - Keine Spezifikation für Implementierungen

Standard-Konformität und Interoperabilität problematisch



4. Das OSI Referenz Modell (cont.)



4. DoD Internet-Modell

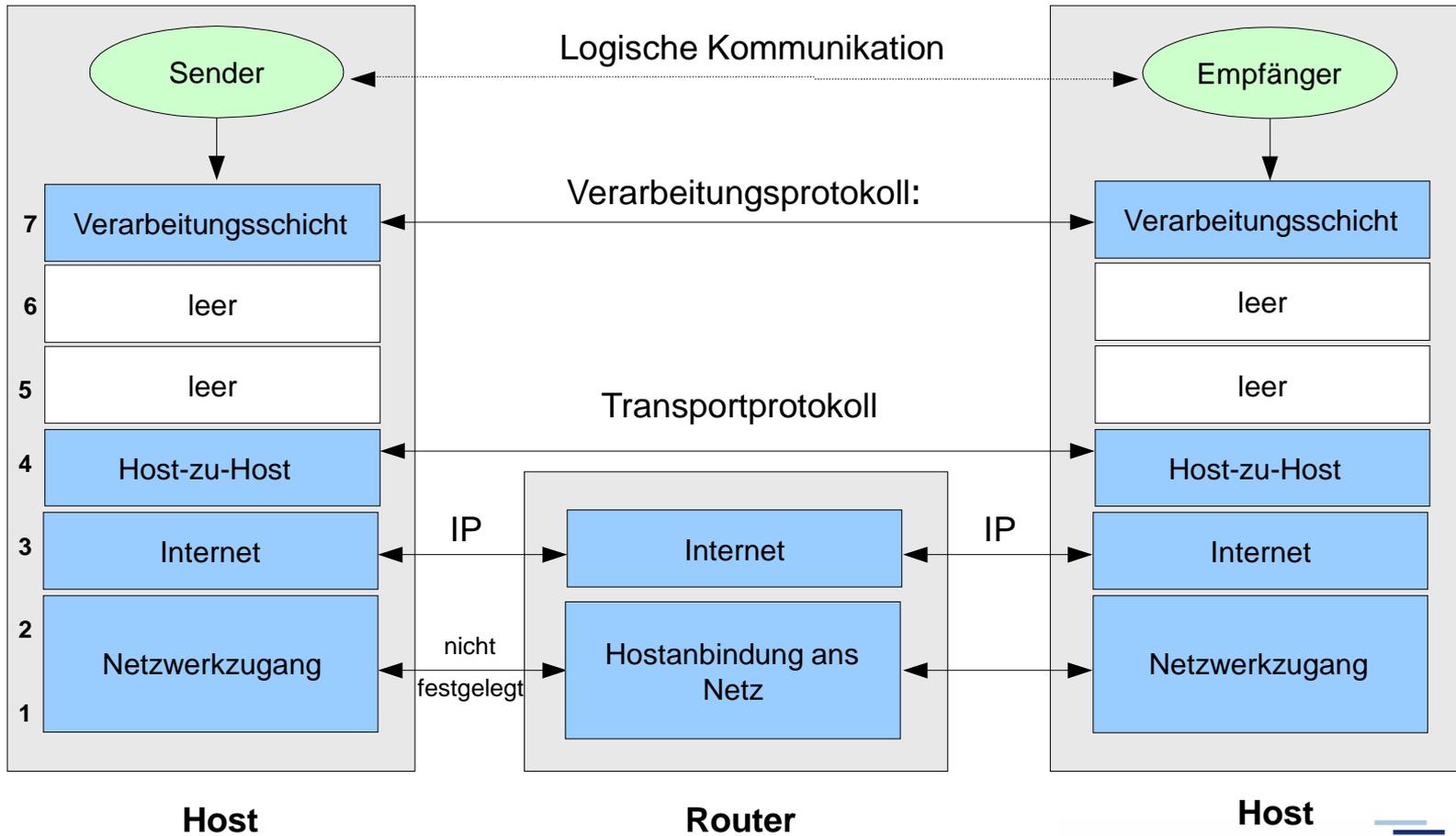
DoD (Department of Defense) - Kommunikationsarchitektur

➤ Bestandteile des Modells:

- **Process:** Implementiert durch Anwendungsprogramme
- **Host-to-Host:** Bietet die Ablaufumgebung für kommunizierende Prozesse
- **Internet:** Ermöglicht die Kommunikation/Vermittlung zwischen Rechnern (hosts)
- **Network Access:** Stellt Zugriff auf Übertragungsmedien bereit (10/100/1000 Base T, FDDI, etc.)



4. Kommunikation im DoD Internet Modell



5. Traditionelle Standardisierer

CCITT Comité Consultatif International de Télégraphique et Téléphonique

ISO International Organisation for Standardization

ITU International Telecommunication Union

ANSI American National Standards Institute

CEN Comité Européen de Normalisation

DIN Deutsches Institut für Normung

IEEE Institute of Electrical and Electronics Engineers

ETSI European Telecommunications Standards Institute

ECMA European Computer Manufacturers Association

Selbsteinschätzungsfragen

1. Weshalb ist eine umfassende Standardisierung im Bereich der Rechnernetze besonders wichtig?
2. Was sind Protokolle? Was tun sie? Welche grundlegenden Eigenschaften besitzen sie?
3. Welche Protokolleigenschaften sind für welche Kommunikationsform erforderlich?
4. Inwieweit unterscheiden sich die konzeptionellen Ansätze des OSI und des DoD Internet Modells?

