

Grundlagen Rechnernetze und Verteilte Systeme

SoSe 2013 – vorläufige Version des Handouts

Kapitel 0: Überblick und Einführung

Prof. Dr.-Ing. Georg Carle

Dipl.-Ing. Stephan M. Günther, M.Sc.

Nadine Herold, M.Sc.

Dipl.-Inf. Stephan Posselt

Fakultät für Informatik

Lehrstuhl für Netzarchitekturen und Netzdienste

Technische Universität München

Kapitel 0: Organisatorisches, Überblick und Schichtenmodelle

- 1 Die Vorlesung im Überblick
 - Organisatorisches zur Vorlesung
 - Zusammenfassung der einzelnen Kapitel

- 2 Geschichte des Internets
 - Von der Entstehung bis zum heutigen Internet
 - Bedeutung des Internets für die Gesellschaft

- 3 Schichtenmodelle
 - Was sind Schichtenmodelle?
 - Wozu sind Schichtenmodelle gut?
 - Das ISO/OSI-Modell

Organisatorisches zu Vorlesung und Übung

- ▶ Prof. Dr.-Ing. Georg Carle
- ▶ Email: carle@net.in.tum.de
- ▶ Raum: MI 03.05.054
- ▶ Sprechstunde: Mo 18:00 – 18:30 Uhr bzw. nach Vereinbarung



Übungsleitung

- ▶ Dipl.-Ing. Stephan M. Günther, M.Sc.
- ▶ Email: guenther@tum.de
- ▶ Raum: MI 03.05.057
- ▶ Sprechstunde: jederzeit

- ▶ Nadine Herold, M.Sc.
- ▶ Email: herold@net.in.tum.de
- ▶ Raum: MI 03.05.044
- ▶ Sprechstunde: nach Vereinbarung



- ▶ Dipl.-Inf. Stephan-A. Posselt
- ▶ Email: posselt@net.in.tum.de
- ▶ Raum: MI 03.05.044
- ▶ Sprechstunde: nach Vereinbarung

Vorlesungsbetrieb

Termine

- ▶ Mo 14:15 – 15:45, MI HS 1 mit Videoübertragung im Interims-HS 1
- ▶ Di 10:15 – 11:45, MI HS 1 mit Videoübertragung im MI HS 2

Zentralübung

- ▶ Nach Ankündigung findet anstelle einer Vorlesung eine Zentralübung statt
- ▶ Inhalte sind Programmieraufgaben sowie Klausurvorbereitung

Übungsbetrieb

Tutorübungen

- ▶ Wöchentliche Übungsblätter
- ▶ Bearbeitung während der Tutorübung in Kleingruppen
- ▶ Keine Anwesenheitspflicht / Hausaufgaben

Termine und Anmeldung zu den Tutorübungen

- ▶ Anmeldung über TUMOnline
- ▶ Anmeldung freigeschaltet ab **Dienstag, 16. April, 19:00 Uhr**
(einige Gruppen erst ab 19:30 Uhr ← Beschränkung von TUMOnline)
- ▶ Übungsbeginn in der 2. Vorlesungswoche (ab 22. April)

Tutoren und (vorläufige) Gruppen

- ▶ Florian Baumann
- ▶ Kilian Batzner
- ▶ Tobias Betz
- ▶ Thomas Blocher
- ▶ Hieu T. Dao
- ▶ Adrian Diemer
- ▶ Michael Eiler
- ▶ Pablo Gómez
- ▶ Mathias Kanzler
- ▶ Felix Kuperjans
- ▶ Philipp Mager
- ▶ Julius Michaelis
- ▶ Frank Schmidt
- ▶ Alexander Schubert
- ▶ Dominik Spreuer
- ▶ Tim Wiese
- ▶ Alexander Winkler

Tag	Start	Ende	Raum	#
Montag	8:00	10:00	03.07.023	1
Montag	8:00	10:00	00.08.053	20
Montag	12:00	14:00	MI HS 2	2
Montag	12:00	14:00	00.08.036	3
Montag	16:00	18:00	00.08.036	4
Dienstag	8:00	10:00	02.08.011	5
Dienstag	8:00	10:00	03.07.023	14
Dienstag	12:00	14:00	00.08.036	6
Dienstag	16:00	18:00	00.08.059	8
Dienstag	16:00	18:00	00.08.036	17
Mittwoch	8:00	10:00	03.07.023	9
Mittwoch	8:00	10:00	00.08.038	23
Mittwoch	10:00	12:00	00.13.054	7
Mittwoch	10:00	12:00	03.07.023	10
Mittwoch	12:00	14:00	03.07.023	11
Mittwoch	14:00	16:00	00.08.036	24
Mittwoch	16:00	18:00	00.08.036	26
Mittwoch	16:00	18:00	00.08.053	27
Donnerstag	8:00	10:00	03.07.023	21
Donnerstag	8:00	10:00	00.08.053	25
Donnerstag	10:00	12:00	00.08.059	12
Donnerstag	10:00	12:00	00.08.036	18
Donnerstag	12:00	14:00	00.08.059	13
Donnerstag	12:00	14:00	00.08.036	22
Freitag	8:00	10:00	03.07.023	28
Freitag	10:00	12:00	00.08.059	15
Freitag	10:00	12:00	00.08.055	19
Freitag	14:00	16:00	00.08.036	16
Freitag	14:00	16:00	00.08.038	29

Programmieraufgaben

Es gibt semesterbegleitend voraussichtlich 3 Programmieraufgaben:

- ▶ Bearbeitung in Gruppen von **maximal 2 Personen**
- ▶ Anmeldung der Teams
 - ▶ bis spätestens 21. April
 - ▶ über diesen **Link**.
 - ▶ bzw. **Homepage zur Vorlesung (www.net.in.tum.de)** → Lehre → SoSe 13 → Vorlesungen → Informationen des Lehrstuhls → Formular
- ▶ Abgabe der Aufgaben über Subversion (Versionsverwaltung; Details folgen)
- ▶ Geplante Aufgaben:
 - 1 Implementierung von ARP über RAW-Sockets
 - 2 Implementierung von Traceroute
 - 3 Implementierung eines Client-/Server-Programms
- ▶ Bearbeitung der Programmieraufgaben in mehreren Sprachen möglich (C, Java, Python, etc.)
- ▶ Abgabe der Programmieraufgaben über Subversion

Die Teilnahme an den Programmieraufgaben

- ▶ ist **freiwillig**
- ▶ aber **Bestandteil der Bonusregelung**. (→ Details folgen gleich)

Tutorgruppen zu den Programmieraufgaben

Infolge des unterschiedlichen Vorwissens der einzelnen Vorlesungsteilnehmer bieten wir **zusätzlich** 4 spezielle Tutorgruppen an:

Tag	Raum	Start	Ende	#
Montag	16:00	18:00	00.08.059	30
Dienstag	12:00	14:00	00.08.059	31
Mittwoch	12:00	14:00	00.08.053	33
Mittwoch	14:00	16:00	00.08.053	32

- ▶ Keine Anmeldung notwendig
- ▶ **Konkrete Fragen / Probleme zu den Programmieraufgaben**
- ▶ **Kein** regulärer Tutorbetrieb (keine Tutoraufgaben)
- ▶ Die Übungen finden **nur dann** statt, wenn auch gerade eine Programmieraufgabe zu bearbeiten ist

Modulprüfung

- ▶ schriftlich, 90 Minuten, voraussichtlich 85 Punkte
- ▶ zugelassene Hilfsmittel:
 - ▶ 1 beidseitig **handschriftlich** beliebig beschriebenes A4-Blatt (Kopien sind nicht zulässig)
 - ▶ nicht-programmierbarer Taschenrechner
 - ▶ Wörterbuch für Muttersprache ↔ Deutsch
- ▶ voraussichtlich am 09.08.2013 (Wiederholung voraussichtlich am 23.09.2013)
- ▶ Anmeldung über TUMOnline

Bonusregelung

- ▶ Erfolgreiche Bearbeitung der Programmieraufgaben
 - ▶ In jeder Aufgabe können 0 – 2 Punkte erreicht werden
 - ▶ Die Programmieraufgaben gelten als bestanden, wenn mind. 3/6 Punkten erreicht wurden
- ▶ Teilname an der Midterm-Prüfung
 - ▶ 45 Minuten, 15 Bonus-Punkte
 - ▶ Hilfsmittel wie in der Modulprüfung
 - ▶ voraussichtlich in der 1. Juniwoche (tba)
 - ▶ Anmeldung über TUMOnline
- ▶ Die in der Midterm erzielten Punkte werden **nur dann auf das Ergebnis der Endterm angerechnet, wenn in den Programmieraufgaben mindestens 3/6 Punkte** erzielt wurden.
- ▶ Der Bonus wird auch auf die Wiederholung angerechnet

Zusammenfassung der einzelnen Kapitel (Stand 15. April 2013)

Kapitel 1: Physikalische Schicht

- 1 Signale, Information und deren Bedeutung
 - ▶ Was sind Signale?
 - ▶ Entropie und Information
- 2 Klassifizierung von Signalen
 - ▶ Zeit- und Frequenzbereich
 - ▶ Abtastung, Rekonstruktion und Quantisierung
- 3 Übertragungskanal
 - ▶ Einflüsse des Übertragungskanals auf Signale
 - ▶ Kapazität eines Übertragungskanals (Modell)
- 4 Nachrichtenübertragung
 - ▶ Quellen- und Kanalkodierung
 - ▶ Impulsformung
 - ▶ Modulation
- 5 Übertragungsmedien
 - ▶ Elektromagnetisches Spektrum
 - ▶ Koaxialleiter
 - ▶ Twisted-Pair-Kabel
 - ▶ Lichtwellenleiter

Zusammenfassung der einzelnen Kapitel

Kapitel 2: Sicherungsschicht

1 Darstellung von Netzwerken als Graphen

- ▶ Netztopologien
- ▶ Adjazenz- und Distanzmatrix
- ▶ Shortest Path Tree und Minimum Spanning Tree

2 Verbindungscharakterisierung, Mehrfachzugriff und Medienzugriffskontrolle

- ▶ Serialisierungs- und Ausbreitungsverzögerungen
- ▶ Nachrichtenflussdiagramme
- ▶ ALOHA und Slotted ALOHA
- ▶ CSMA, CSMA/CD und CSMA/CA
- ▶ Token Passing

3 Rahmenbildung, Adressierung und Fehlerkennung

- ▶ Erkennung von Rahmengrenzen und Codetransparenz
- ▶ Adressierung und Fehlererkennung
- ▶ Fallstudie: IEEE 802.3u (FastEthernet)
- ▶ Fallstudie: IEEE 802.11 a/b/g/n (Wireless LAN)

4 Verbindungen auf Schicht 1 und 2

- ▶ Hubs, Bridges und Switches
- ▶ Collision und Broadcast Domains

Zusammenfassung der einzelnen Kapitel

Kapitel 3: Vermittlungsschicht

1 Vermittlungsarten

- ▶ Leitungsvermittlung
- ▶ Nachrichtenvermittlung
- ▶ Paketvermittlung

2 Adressierung im Internet

- ▶ Internet Protocol (IP)
- ▶ Adressauflösung (ARP)
- ▶ Internet Control Message Protocol (ICMP)
- ▶ Adressklassen (für Classful Routing)
- ▶ Subnetting und Präfixe (für Classless Routing)

3 Routing

- ▶ Statisches Routing
- ▶ Longest Prefix Matching
- ▶ Dynamisches Routing
- ▶ Algorithmen von Bellman-Ford und Dijkstra
- ▶ Routingprotokolle (Distance Vector und Link State)
- ▶ Autonome Systeme

4 Nachfolge von IPv4: IPv6

Zusammenfassung der einzelnen Kapitel

Kapitel 4: Transportschicht

- 1 Aufgaben der Transportschicht
- 2 Multiplexing durch Port-Nummern
- 3 Verbindungslose Übertragung: UDP
 - ▶ Case-Study: UDP
 - ▶ Code-Study: `SOCK_DGRAM` (C)
- 4 Verbindungsorientierte Übertragung: TCP
 - ▶ Sliding-Window-Protokolle (Go-Back-N und Selective Repeat)
 - ▶ Case-Study: TCP (Fluss- und Staukontrolle)
 - ▶ Code-Study: `SOCK_STREAM` (C)
- 5 Network Address Translation (NAT)

Zusammenfassung der einzelnen Kapitel

Kapitel 5: Die Schichten 5 – 7

1 Sitzungsschicht

- ▶ Dienste
- ▶ Funktionseinheiten
- ▶ Synchronisation
- ▶ Quality of Service
- ▶ Performance Parameter

2 Kryptographie und Netzsicherheit

- ▶ Symmetrische und asymmetrische Verfahren
- ▶ RC4 und DH76
- ▶ Zertifikate
- ▶ Transport Layer Security (TLS)

3 Darstellungsschicht

- ▶ Datenkompression (Huffman Code)
- ▶ Einheitliche Syntax (ASN.1, BER)

4 Anwendungsschicht

- ▶ Namensauflösung im Internet (DNS)
- ▶ Demos: DNS, HTTP

Zusammenfassung der einzelnen Kapitel

Kapitel 6: Verteilte Systeme

1 Homogene, skalierbare Paradigmen

- ▶ Message Passing Interface (MPI)
- ▶ MapReduce
- ▶ Pipes, netcat, DUP

2 Remote Procedure Call

- ▶ Funktionsaufrufe und Parameterkodierung
- ▶ Stubs, IDL, Binding
- ▶ Java RMI
- ▶ RPC/RMI

3 Shared Memory

- ▶ NUMA (Non-Uniform Memory Access)
- ▶ Virtueller Speicher
- ▶ Auslagerung
- ▶ Distributed Shared Memory
- ▶ Konsistenz in parallelen Programmen

4 Einbettung in Programmiersprachen

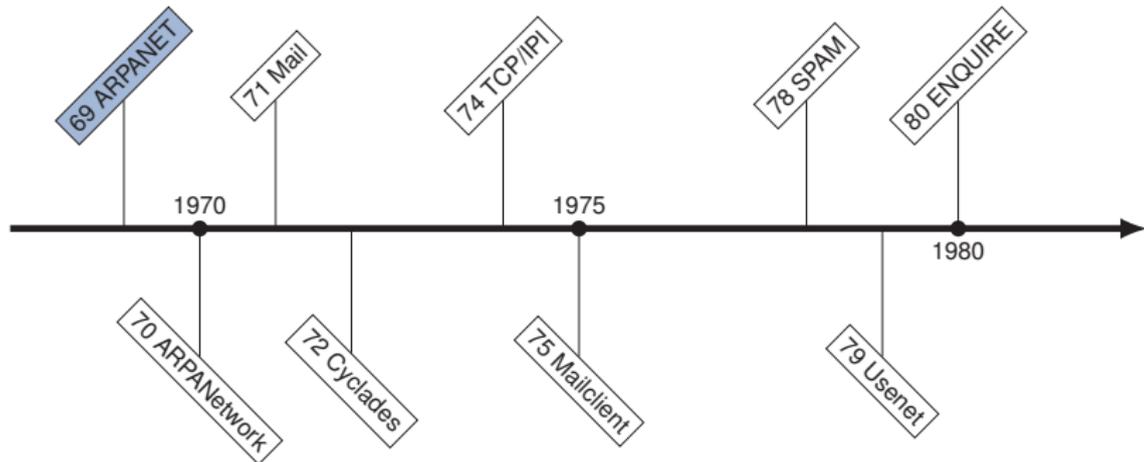
- ▶ Erlang
- ▶ Actor Model

Zurück zu Kapitel 0:

- 1 Die Vorlesung im Überblick
 - Organisatorisches zur Vorlesung
 - Zusammenfassung der einzelnen Kapitel
- 2 Geschichte des Internets
 - Von der Entstehung bis zum heutigen Internet
 - Bedeutung des Internets für die Gesellschaft
- 3 Schichtenmodelle
 - Was sind Schichtenmodelle?
 - Wozu sind Schichtenmodelle gut?
 - Das ISO/OSI-Modell

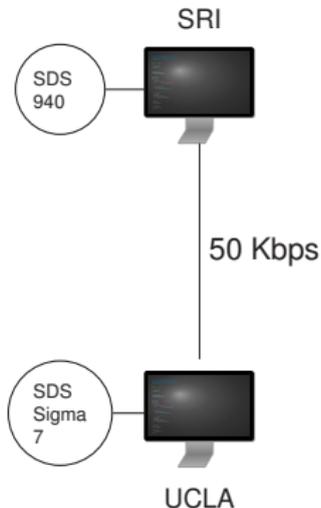
Von der Entstehung bis zum heutigen Internet

Geschichte des Internets: Übersicht bis 1980



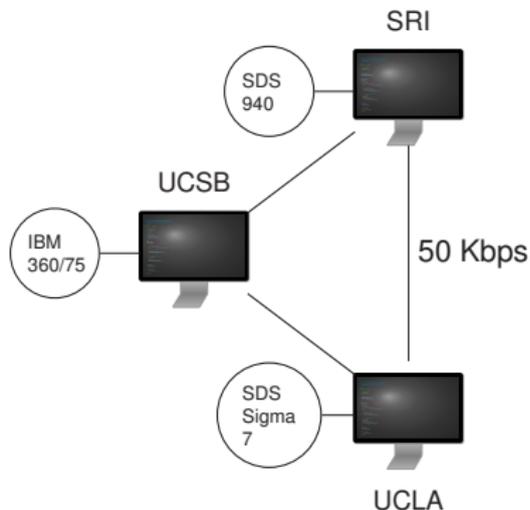
ARPANET mit den ersten 4 Knoten

- ▶ University of California, Los Angeles (UCLA) 1.9.1969
- ▶ Stanford Research Institute (SRI) 1.10.1969



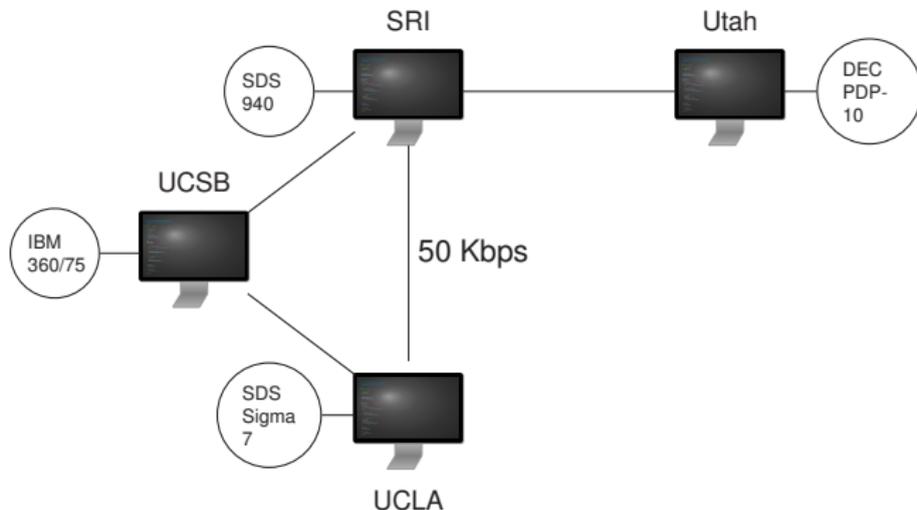
ARPANET mit den ersten 4 Knoten

- ▶ University of California, Los Angeles (UCLA) 1.9.1969
- ▶ Stanford Research Institute (SRI) 1.10.1969
- ▶ UC Santa Barbara (UCSB) 1.11.1969



ARPANET mit den ersten 4 Knoten

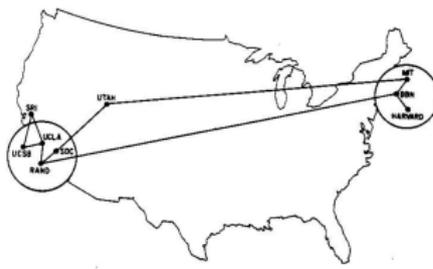
- ▶ University of California, Los Angeles (UCLA) 1.9.1969
- ▶ Stanford Research Institute (SRI) 1.10.1969
- ▶ UC Santa Barbara (UCSB) 1.11.1969
- ▶ University of Utah 12.1969



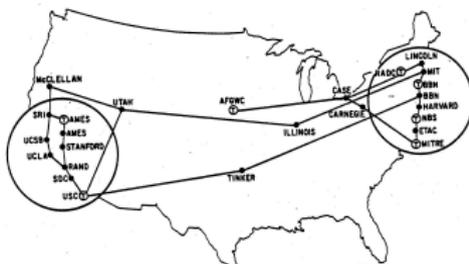
ARPANET von 1969 bis 1977



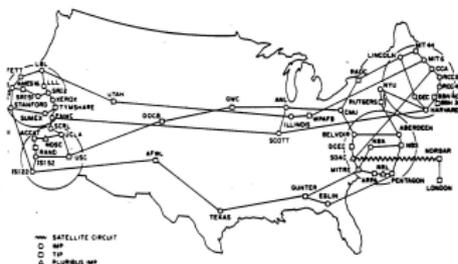
ARPANET 1969, 4 Knoten



ARPANET 1970, 9 Knoten

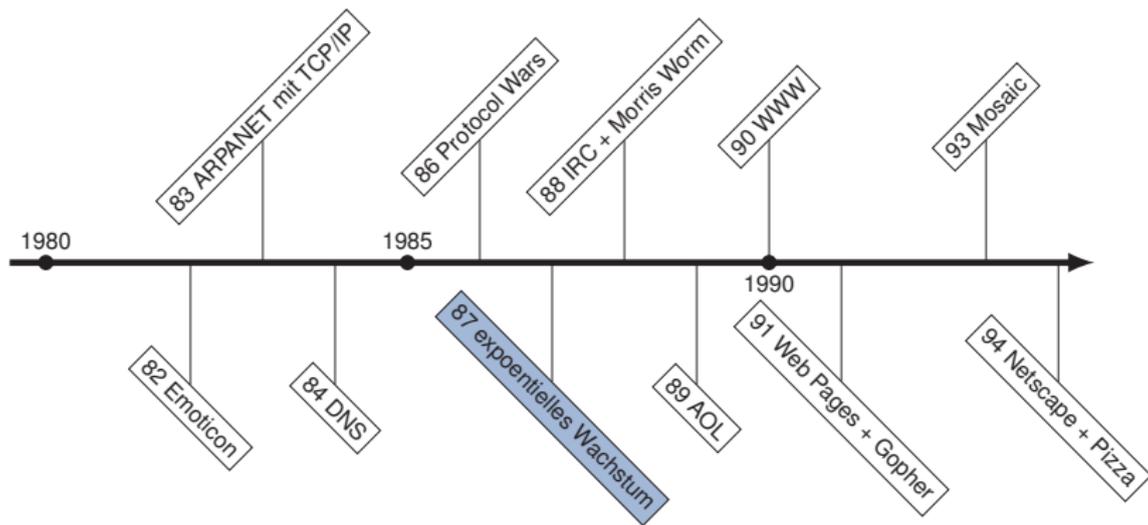


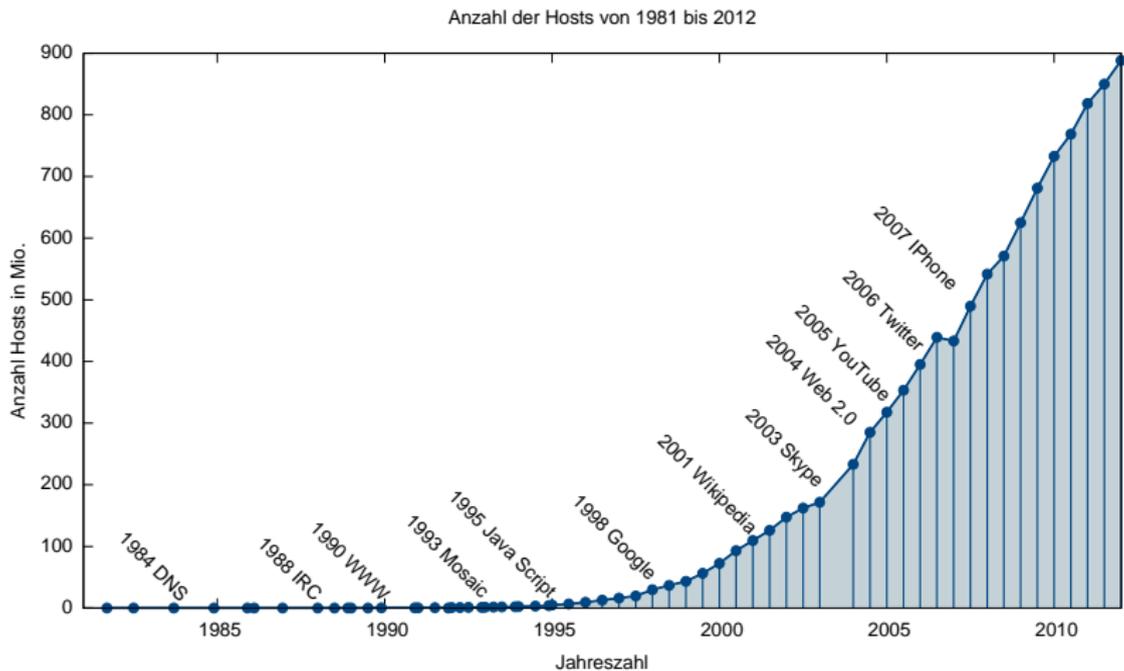
ARPANET 1972, 25 Knoten



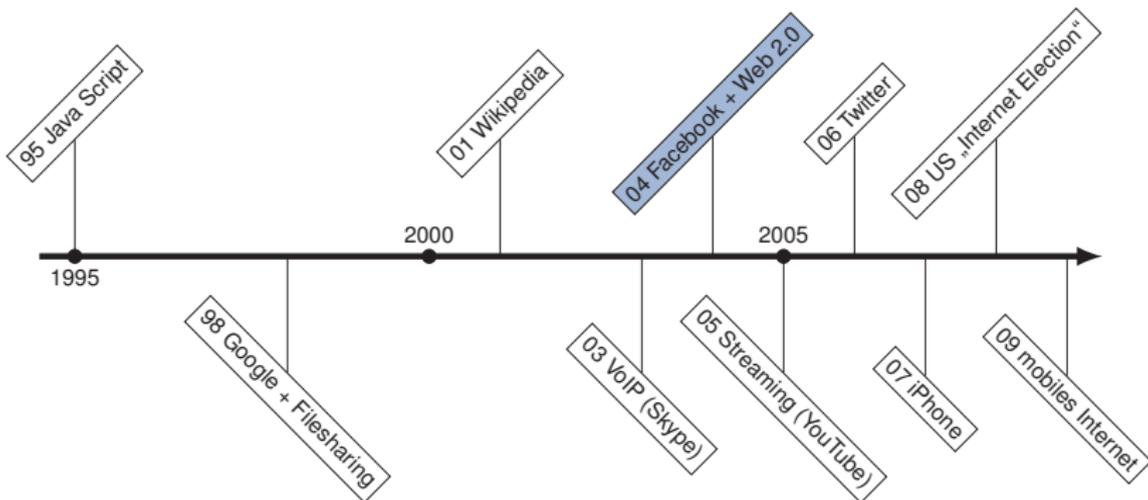
ARPANET 1977, 58 Knoten

Geschichte des Internets: Übersicht von 1980 bis 1994





Geschichte des Internets: Übersicht ab 1994



Web 2.0 Meme Map [2]

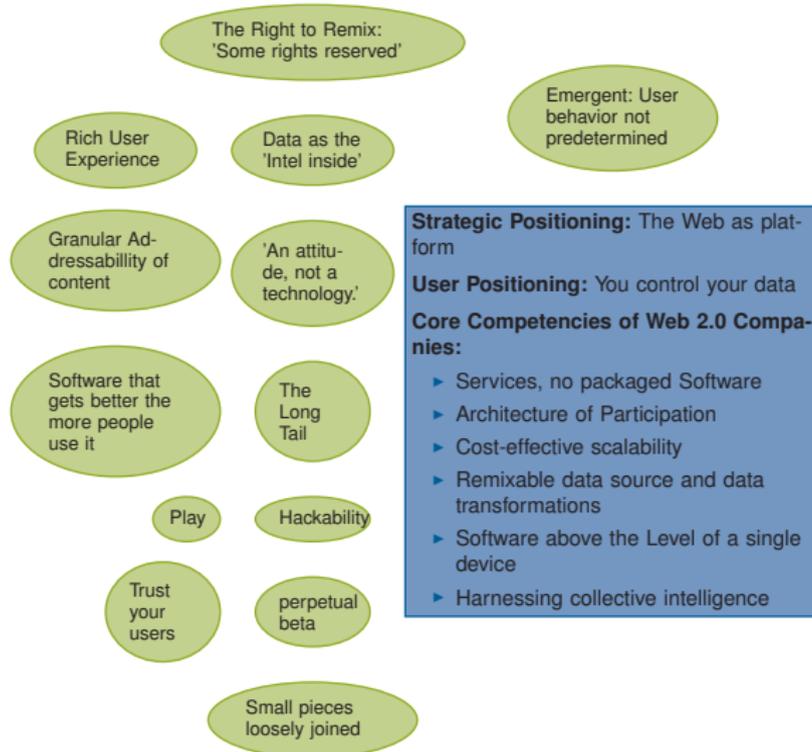
Strategic Positioning: The Web as platform

User Positioning: You control your data

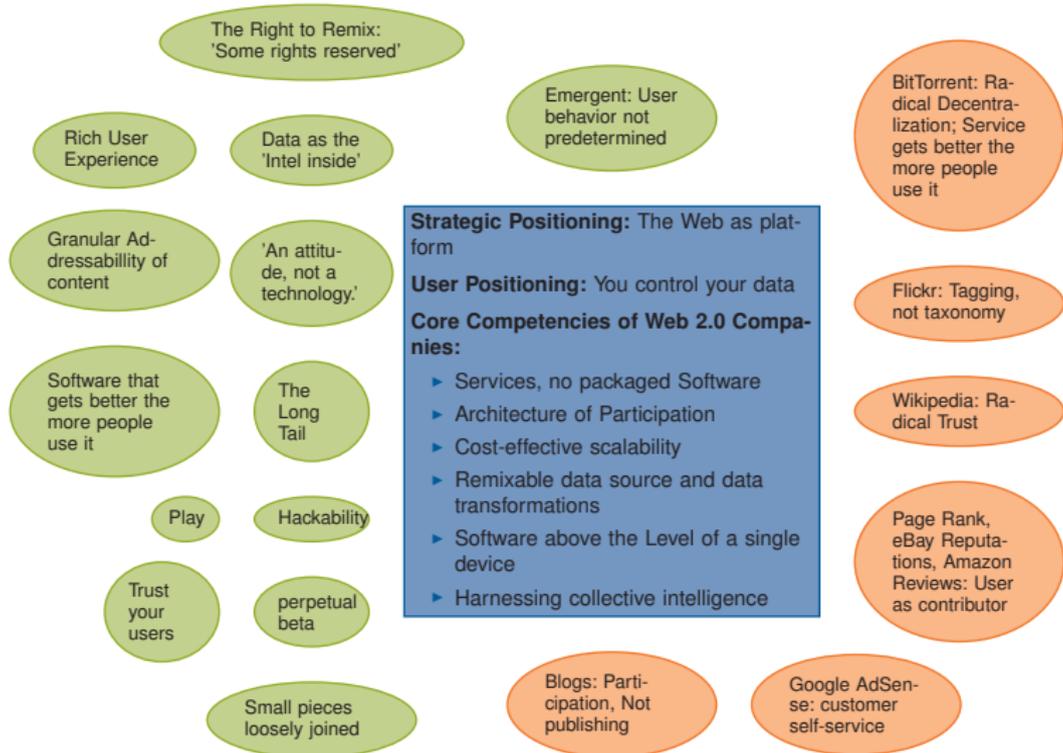
Core Competencies of Web 2.0 Companies:

- ▶ Services, no packaged Software
- ▶ Architecture of Participation
- ▶ Cost-effective scalability
- ▶ Remixable data source and data transformations
- ▶ Software above the Level of a single device
- ▶ Harnessing collective intelligence

Web 2.0 Meme Map [2]



Web 2.0 Meme Map [2]



Das Internet Heute



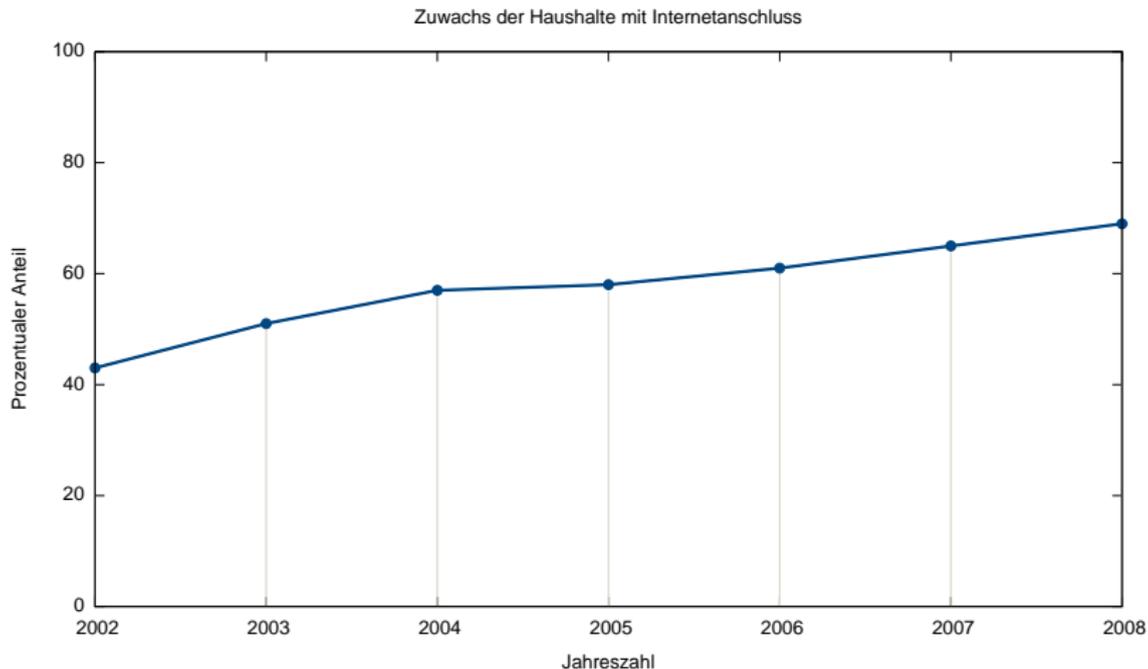
[1], 2007

Das Internet Heute

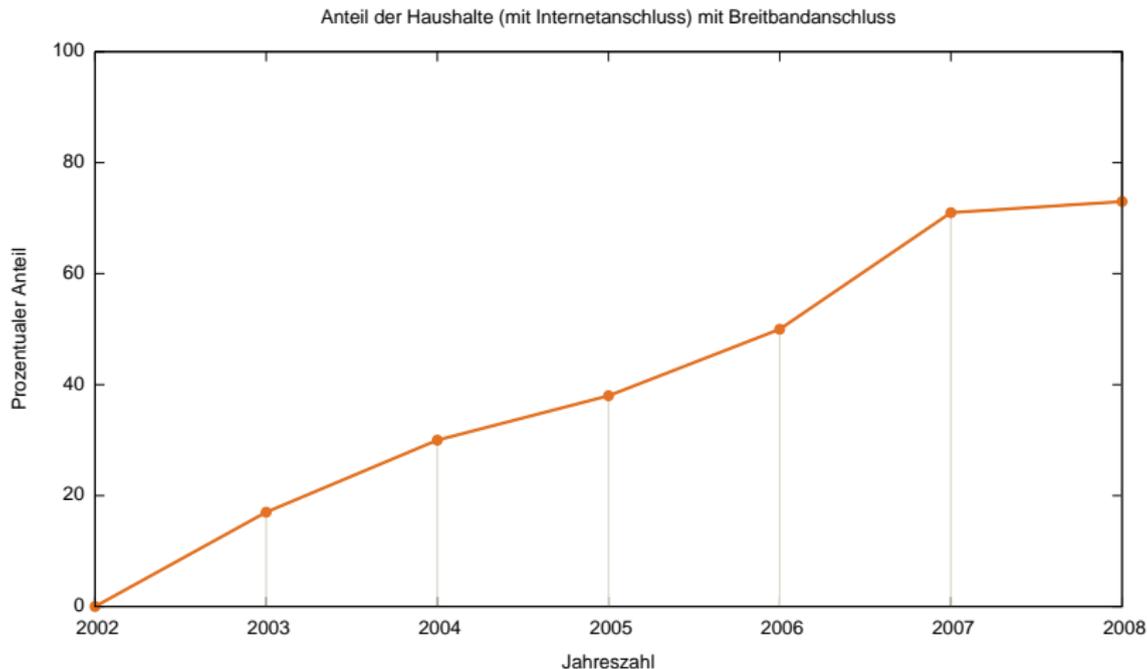


Blau: seit langem existierende Autonome Systeme

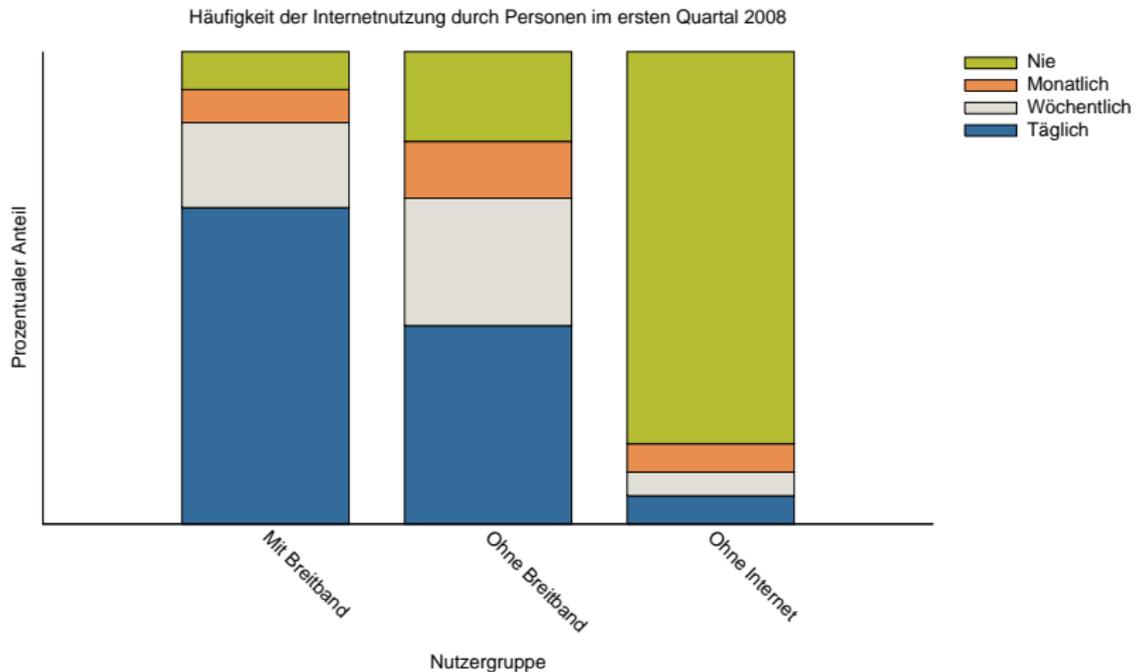
Bedeutung des Internets für die Gesellschaft



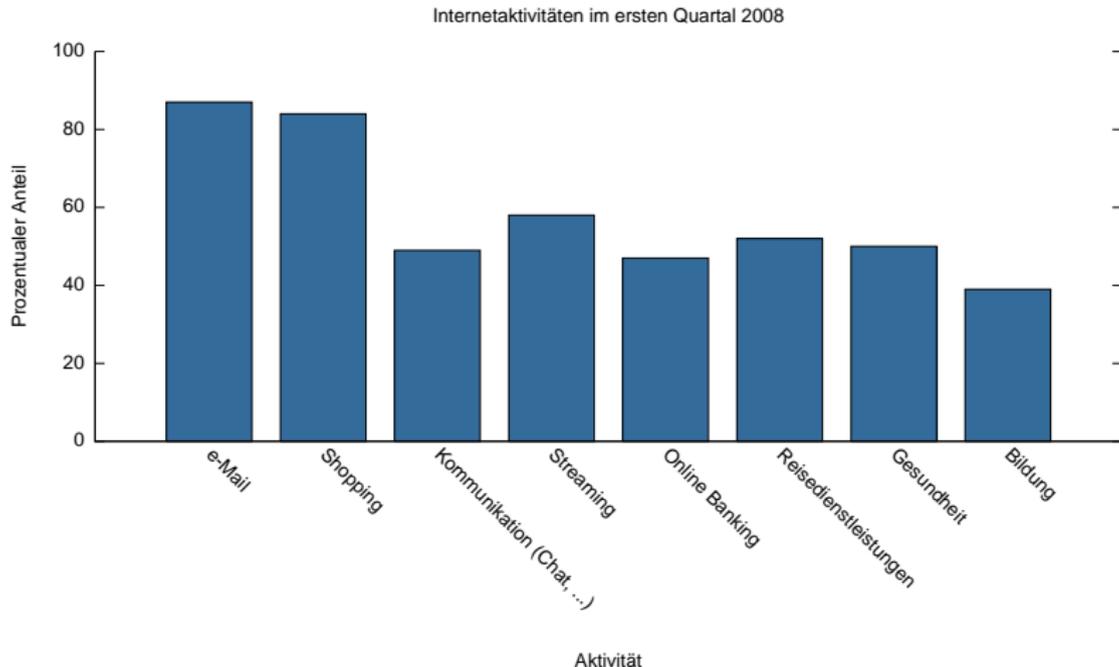
Bedeutung des Internets für die Gesellschaft



Bedeutung des Internets für die Gesellschaft



Bedeutung des Internets für die Gesellschaft

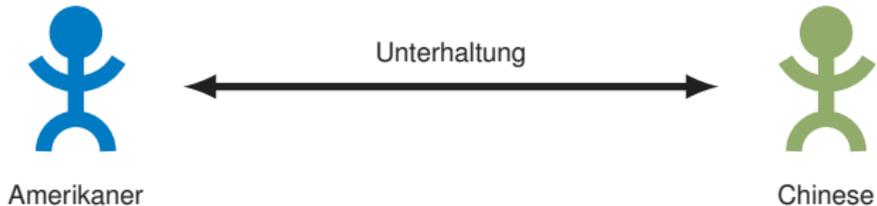


Inhalt

- 1** Die Vorlesung im Überblick
 - Organisatorisches zur Vorlesung
 - Zusammenfassung der einzelnen Kapitel
- 2** Geschichte des Internets
 - Von der Entstehung bis zum heutigen Internet
 - Bedeutung des Internets für die Gesellschaft
- 3** Schichtenmodelle
 - Was sind Schichtenmodelle?
 - Wozu sind Schichtenmodelle gut?
 - Das ISO/OSI-Modell

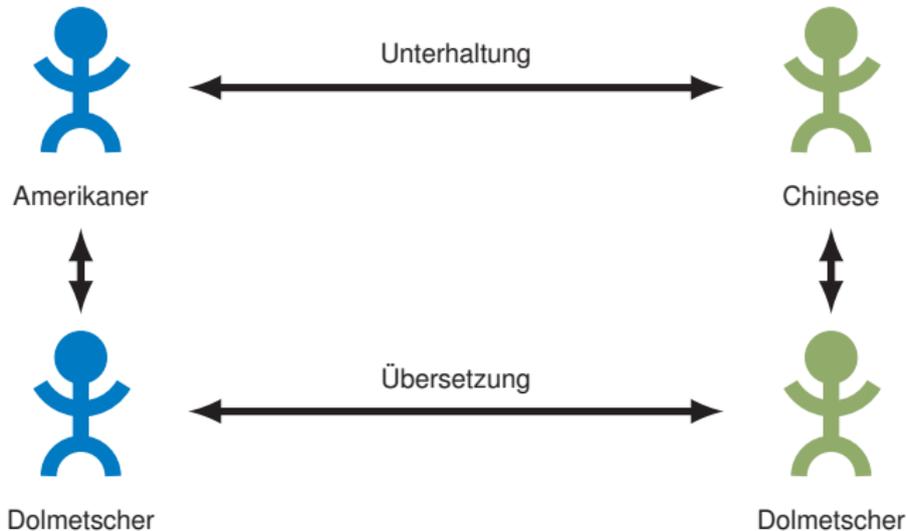
Was sind Schichtenmodelle?

Ein einfaches Beispiel:



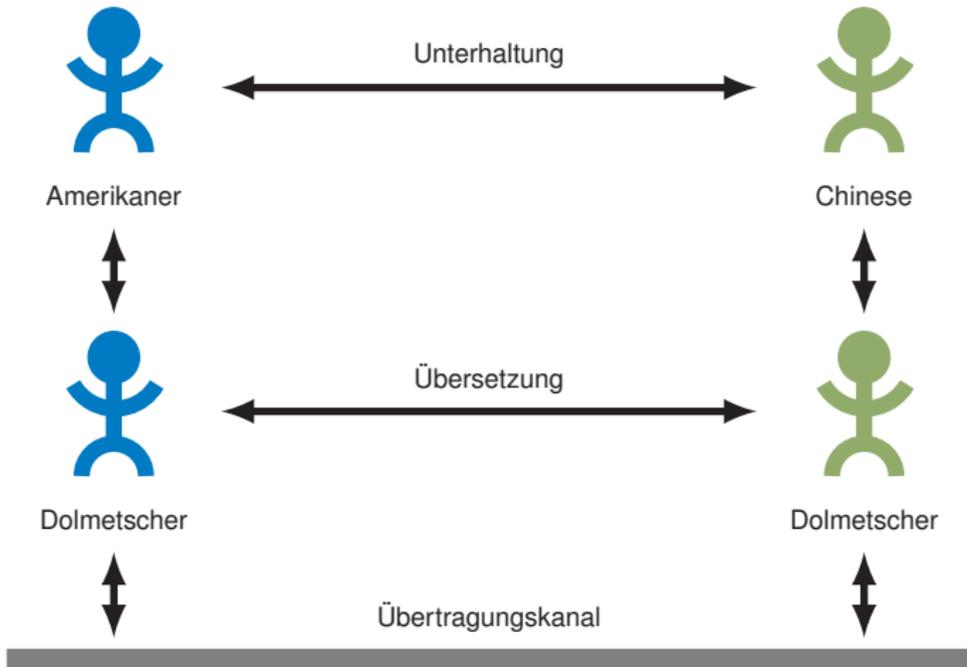
Was sind Schichtenmodelle?

Ein einfaches Beispiel:



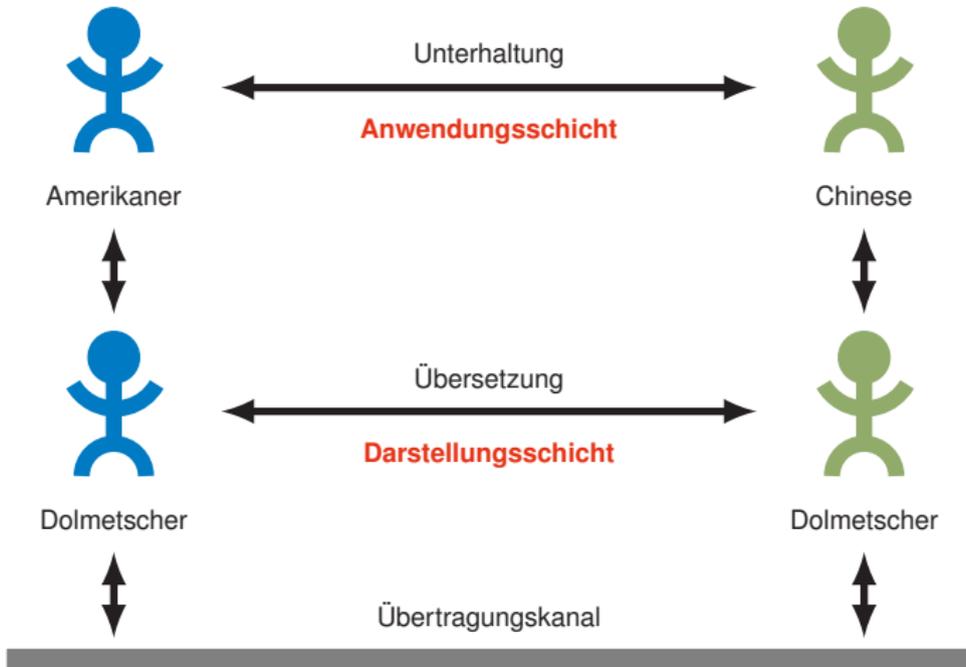
Was sind Schichtenmodelle?

Ein einfaches Beispiel:



Was sind Schichtenmodelle?

Ein einfaches Beispiel:



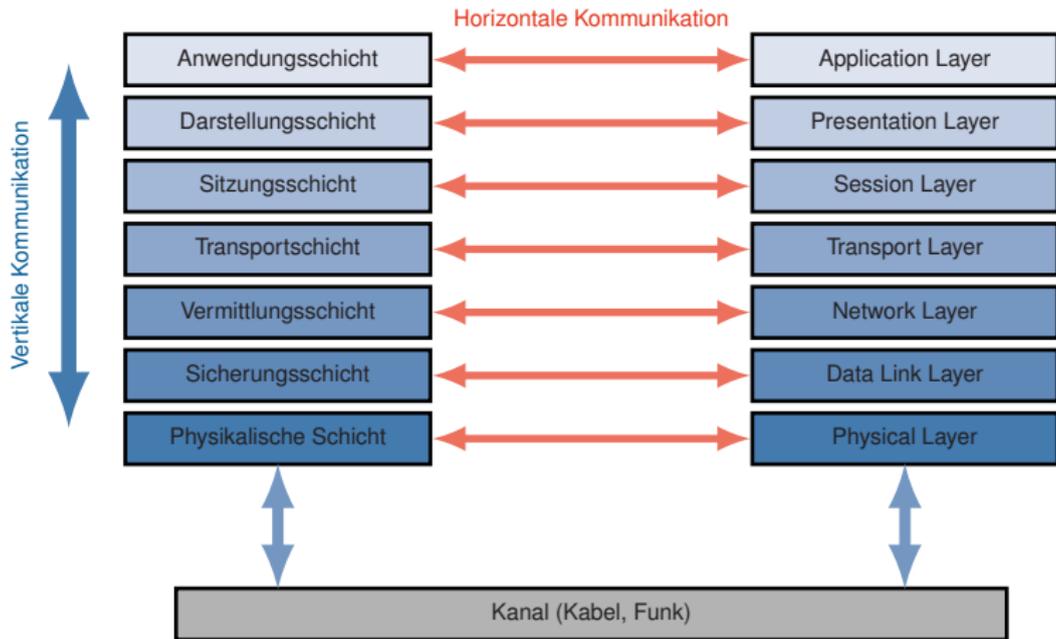
Wozu sind Schichtenmodelle gut?

- ▶ Unterteilung des komplexen Kommunikationsvorgangs
 - ▶ Niedrigere Schichten **bieten** höheren Schichten **Dienste an**
 - ▶ Höhere Schichten **nehmen Dienste** der jeweils niedrigeren Schicht **in Anspruch**
- ▶ Abstraktion von der Implementierung einer Schicht
 - ▶ Festlegung, **welche** Dienste angeboten werden, aber **nicht wie** sie erfüllt werden
 - ▶ Austauschbarkeit einzelner Implementierungen
- ▶ Anwendbar auf beliebige Kommunikationsvorgänge

Das ISO/OSI-Modell

- ▶ Entwickelt zwischen 1979 und 1983 von der *International Organization for Standardization (ISO)*
- ▶ OSI = *Open Systems Interconnect*
- ▶ Unterteilt den Kommunikationsvorgang in **7 Schichten**
- ▶ Jede Schicht erbringt bestimmte Dienste (z. B. Aufteilen einer Nachricht in kleinere Pakete)
- ▶ Keine Aussage, wie diese Dienste zu erbringen sind

Schematische Darstellung des OSI-Modells:



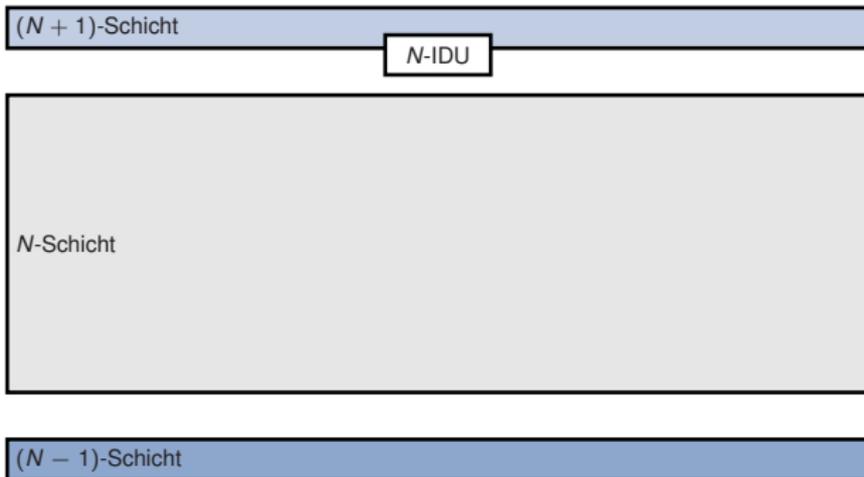
Datenaustausch zwischen Schichten

$(N + 1)$ -Schicht

N -Schicht

$(N - 1)$ -Schicht

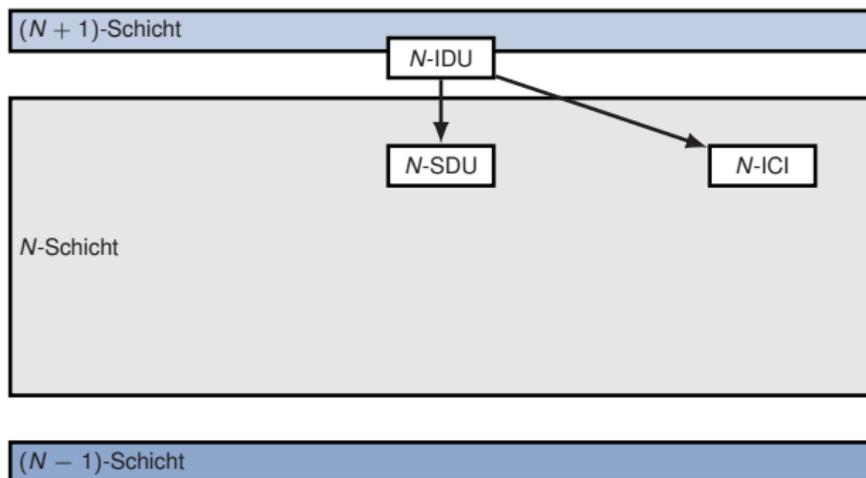
Datenaustausch zwischen Schichten



$(N + 1)$ -Schicht nimmt Dienste der N -Schicht in Anspruch:

- ▶ N -Schicht erhält eine **Interface Data Unit (IDU)** von der $(N + 1)$ -Schicht.

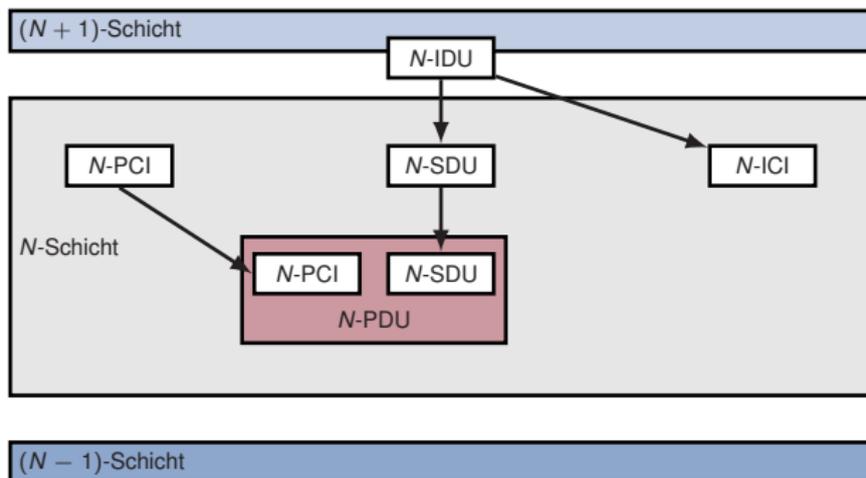
Datenaustausch zwischen Schichten



N -IDU enthält aus Sicht der N -Schicht

- ▶ Nutzdaten (**Service Data Unit (SDU)**) und
- ▶ Kontrollinformationen (**Interface Control Information (ICI)**), welche zum Erbringen des Dienstes notwendig sind (z. B. Adressinformationen).

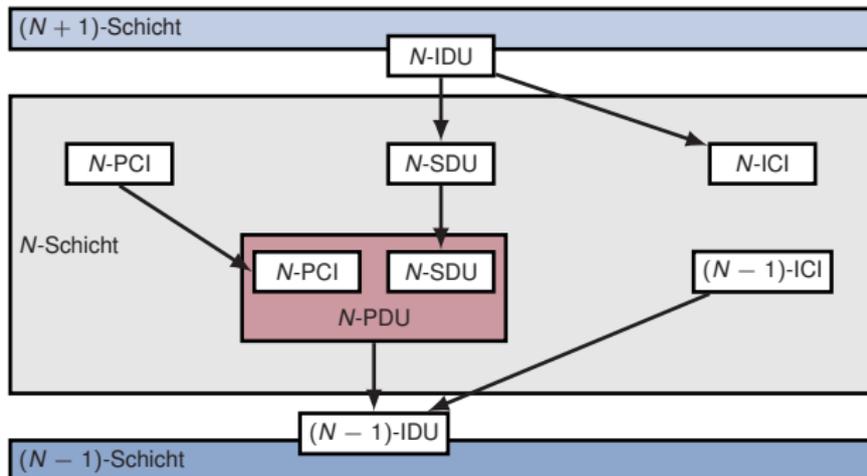
Datenaustausch zwischen Schichten



N -Schicht

- ▶ erbringt auf der N -SDU die angeforderten Dienste,
- ▶ fügt sog. **Protocol Control Information (PCI)** für die N -Schicht der Gegenseite hinzu und
- ▶ erzeugt so aus PCI und SDU die **Protocol Data Unit (PDU)**.

Datenaustausch zwischen Schichten



N -Schicht fordert nun Dienste der $(N-1)$ -Schicht an indem

- ▶ $(N-1)$ -ICI erzeugt und
- ▶ zusammen mit der N -PDU

als $(N-1)$ -IDU der nächst niedrigeren Schicht übergeben werden.

Üblich ist der Begriff **Protocol Data Unit (PDU)**, welcher auf der N -Schicht

- ▶ die (ggf. bearbeiteten) Nutzdaten der $(N - 1)$ -Schicht sowie
- ▶ protokollspezifische Informationen (PCI) der N -Schicht

bezeichnet. Die PCI wird dabei häufig in Form eines **Headers** den Nutzdaten vorangestellt.

PDU's einiger Schichten haben eigene Bezeichnungen. Man spricht von

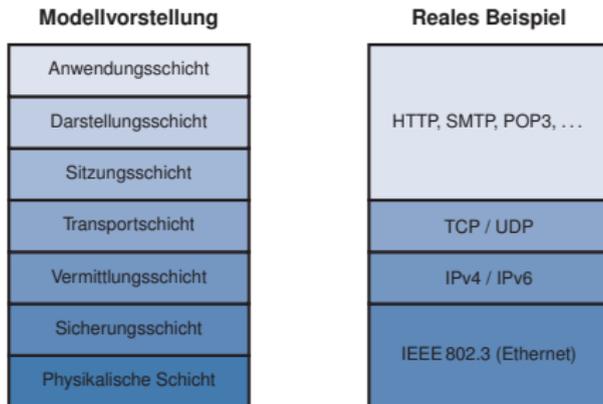
- ▶ **Segmenten** auf der Transportschicht,
- ▶ **Paketen** auf der Vermittlungsschicht bzw.
- ▶ **Rahmen** (engl. **Frames**) auf der Sicherungsschicht.

Die Unterscheidung ermöglicht es, implizit die gerade betrachtete Schicht anzugeben.

Insbesondere die Unterscheidung zwischen Rahmen und Paketen ist in der Literatur häufig fließend.

Schwächen des ISO/OSI-Modells

- ▶ Die Trennung der Schichten widerspricht manchmal anderen Interessen (z. B. der Effizienz)
- ▶ Einige Protokolle sind daher nicht klar einer bestimmten Schicht zuzuordnen oder arbeiten sogar auf mehreren Schichten ([Cross Layer](#))
- ▶ Die Zuordnung von Protokollen auf einzelne Schichten hängt häufig von der Betrachtungsweise ab



Eine kurze Übersicht zum ISO/OSI-Modell finden Sie in [3].

Literaturhinweise und Quellenangaben

- [1] Harrison, C.: World City-to-City Connections.
<http://www.chrisharrison.net/index.php/Visualizations/InternetMap>.
- [2] O'Reilly, Tim: O'Reilly Network: What Is Web 2.0, September 2005.
<http://www.oreillynet.com/lpt/a/6228>.
- [3] Stein, E.: Taschenbuch Rechnernetze und Internet, Kapitel Das OSI-Modell, Seiten 22–28.
Fachbuchverlag Leipzig, 2. Auflage, 2004.
Auszug s. Moodle/SVN.