

Grundlagen Rechnernetze und Verteilte Systeme (GRNVS)

IN0010 – SoSe 2026

Prof. Dr.-Ing. Georg Carle

Lorenz Lehle, Christian Dietze, Stefan Lachnit, Manuel Simon, Markus Sosnowski

Lehrstuhl für Netzarchitekturen und Netzdienste
School of Computation, Information and Technology
Technische Universität München

Organisatorisches zur Vorlesung

Zusammenfassung der einzelnen Kapitel

Entstehung des Internets

Schichtenmodelle

Literaturangaben

Kapitel 0: Organisatorisches und Einführung

Organisatorisches zur Vorlesung

Vorlesungsbetrieb

Übungsbetrieb

Programmier- und Praxisaufgaben

Midterm-Quiz

Modulprüfung

Bonusregelung

Vorlesungsunterlagen

Zusammenfassung der einzelnen Kapitel

Entstehung des Internets

Schichtenmodelle

Literaturangaben



- Prof. Dr.-Ing. Georg Carle
- Email: carle@tum.de
- Raum: MI 03.05.054
- Sprechstunde: Nach Vereinbarung



Name Lorenz Lehle
Room MI 03.05.042



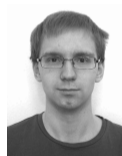
Name Manuel Simon
Room MI 03.05.044



Name Markus Sosnowski
Room MI 03.05.037



Name Stefan Lachnit
Room MI 03.05.041



Name Christian Dietze
Room MI 03.05.053

Fragen an die Übungsleitung: grnvs@net.cit.tum.de

Vorlesungsbetrieb

Termine

- Mo 10:15 – 11:45 Uhr (Audimax Galileo)
- Di 10:15 – 11:45 Uhr (Audimax Galileo)

Fragen

- Zum interaktiven Beantworten von Fragen werden wir Tweedback einsetzen, und dafür in der jeweiligen Vorlesung einen aktuellen Link bekannt geben

Vorlesungsunterlagen und Aufzeichnung

- Vorlesungsunterlagen und Übungsmaterialien werden über <https://grnvs.net> bereitgestellt. (Näheres siehe Folie 0-18)
- Die Vorlesung wird als Präsenzveranstaltung ohne Live Streaming und ohne Aufzeichnung angeboten
- Die Aufzeichnungen vom SoSe 2021 werden auf <https://live.rbg.tum.de> angeboten (Anmeldung zur Vorlesung erforderlich)
- Wir weisen darauf hin, dass sich die Inhalte der diesjährigen Vorlesung leicht unterscheiden können
- Wir empfehlen, regelmäßig an der Vorlesung teilzunehmen

Zentralübung

- Es ist keine regelmäßige Zentralübung geplant
- Gelegentlich finden anstelle von Vorlesungen Fragestunden zur Klausurvorbereitung statt
- Die wöchentlichen Übungsblätter werden Montags in der Vorlesung vor den Übungen vorgestellt

Tutorübungen

- Wöchentliche Übungsblätter, die jeweils am Montag am Ende der Vorlesung vorgestellt werden
- Bearbeitung während der Tutorübung in Kleingruppen
- Keine Anwesenheitspflicht / Hausaufgaben
- Bitte machen Sie sich vor der Übung mit dem Inhalt des jeweiligen Übungsblatts vertraut

Anmeldung zu den Tutorübungen

- Die **Anmeldung erfolgt über TUMonline**
- Das **Matching auf Tutorgruppen erfolgt täglich um 23:00 Uhr**
- Die **Übungen beginnen ab Montag, den 20. April**

Abhaltung der Übungen

- Alle Gruppen finden in Präsenz statt.
- Bitte beachten Sie, dass Gruppen bei Bedarf bzw. bei Teilnehmermangel auch verschoben oder im Modus geändert werden können. Angemeldete Studierende erhalten in diesem Fall eine Email.

Es ist geplant, 29 Tutorgruppen anzubieten:

Tag	Start	Ende	# Gruppen
Montag	12:00	14:00	2
Montag	16:00	18:00	2
Dienstag	08:00	10:00	1
Dienstag	12:00	14:00	2
Dienstag	14:00	16:00	3
Dienstag	16:00	18:00	2
Mittwoch	08:00	10:00	1
Mittwoch	10:00	12:00	1
Mittwoch	12:00	14:00	2
Mittwoch	14:00	16:00	2
Mittwoch	16:00	18:00	1

Tag	Start	Ende	# Gruppen
Donnerstag	08:00	10:00	1
Donnerstag	10:00	12:00	2
Donnerstag	12:00	14:00	2
Donnerstag	14:00	16:00	1
Donnerstag	16:00	18:00	1
Freitag	10:00	12:00	1
Freitag	12:00	14:00	1
Freitag	14:00	16:00	1

- Sollten sich Änderungen an den Gruppen ergeben, wird dies über Moodle bekanntgegeben
- Eine aktuelle Liste mit den stattfindenden Gruppen finden Sie auf <https://grnvs.net/tutorgruppen.txt>
- Es steht aktuell noch nicht fest, welche Gruppe von welchem Tutor gehalten wird

Bitte verteilen Sie sich auf gleichzeitig stattfindende Parallelgruppen!

Programmier- und Praxisaufgaben

Es gibt semesterbegleitend voraussichtlich 4 Programmieraufgaben:

- Einzelabgaben
- Anmeldung über <https://grnvs.net/connect> (jeweils ab Bekanntgabe der Aufgabe)
- Geplante Aufgaben:
 1. Einführungsaufgabe zum Umgang mit Git und dem Kennenlernen der Testumgebung
 2. Statistiken über empfangene Rahmentypen
 3. Traceroute
 4. TCP Sockets
- Rahmenprogramme werden in Python, C und Java zur Verfügung gestellt
- Abgabe in anderen Programmiersprachen ist erlaubt
- Abgabe der Programmieraufgaben über das LRZ Gitlab¹
- Automatische Tests der Abgaben (→ Details folgen)

Die Teilnahme an den Programmieraufgaben ist

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

¹ Wir möchten Sie im Rahmen der Programmieraufgaben auch an für Ihre zukünftige Arbeit wichtige Konzepte und Tools wie Versionsverwaltung mittels Git und die Arbeit auf der Kommandozeile vorbereiten. Diese Dinge sind nicht Bestandteil von Vorlesungen und können nur durch selbstständiges Arbeiten erlernt werden.

Es gibt semesterbegleitend voraussichtlich 6 praktische Aufgaben mit einem virtuellen Testbed:

- Einzelbearbeitung
- Zugriff auf vier virtuelle Maschinen
- Live-Bearbeitung und Bewertung
- In den Aufgaben konfigurieren Sie beispielsweise:
 - Netzwerk-Interfaces
 - Routing zwischen mehreren Netzen
 - verschiedene Dienste wie z.B. NAT

Die Teilnahme an den Praxisaufgaben ist ebenso

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

¹ Wir möchten Sie im Rahmen der Praxisaufgaben auch an Systeme heranführen, die in der Forschungspraxis am Lehrstuhl aktiv verwendet werden.

Sprechstunden zu den Programmier- und Praxisaufgaben

Infolge des unterschiedlichen Vorwissens bieten wir **zusätzlich** 3 spezielle Sprechstunden an:

Tag	Start	Ende	Raum
Dienstag	16:00	18:00	03.07.023
Donnerstag	10:00	12:00	03.07.023
Freitag	10:00	12:00	03.07.023

- Keine Anmeldung notwendig
- **Konkrete Fragen / Probleme zu den Aufgaben**
- **Kein** festes Programm und keine Besprechung der Übungsblätter der regulären Tutorien
- Die Sprechstunden finden **nur dann** statt, wenn auch gerade eine Programmier- oder Praxisaufgabe zu bearbeiten ist

Anerkennung der Programmieraufgaben aus Vorjahren

Ergebnisse werden nicht direkt anerkannt, aber **eigener** Code vom Vorjahr darf wieder abgegeben werden sofern

- ein deutlicher Hinweis auf die Abgabe vom Vorjahr enthalten ist und
- der Code der Aufgabenstellung und den Abgeberichtlinien entspricht.

Es wird in der Mitte des Semesters voraussichtlich ein **Midterm-Quiz** stattfinden:

- Closed-book
- Montag oder Dienstag während Vorlesungsslot
- voraussichtlich primär Multiple Choice
- Genauer Termin TBA, aber rechtzeitige Ankündigung

Die Teilnahme am Midterm-Quiz ist ebenso

- [freiwillig](#)
- aber [Bestandteil der Bonusregelung](#). (→ Details folgen gleich)

Endterm vsl. am Freitag, den 31.07.2026 08:00 Uhr

Retake vsl. am Montag, den 05.10.2026 14:00 Uhr

Modus der Modulprüfung:

- Präsenz, 90 Minuten, voraussichtlich 90 Punkte
- Closed-book
- vsl. Hilfsmittel:
 - das mit der Prüfung ausgeteilte Cheatsheet
 - Wörterbuch Muttersprache ↔ Deutsch ohne Anmerkungen
 - ein nicht-programmierbarer Taschenrechner
- Multiple Choice Multiple Answer möglich (max. 20%)
- Rechtzeitige Anmeldung über TUMonline zwingend erforderlich

Juli

		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

Oktober

			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

Beschreibung des Bonusverfahrens

- In den Programmieraufgaben können **insgesamt bis zu 6 Bonuspunkte** erreicht werden (p_{Prog})
- In den Praxisaufgaben können **insgesamt bis zu 6 Bonuspunkte** erreicht werden (p_{Praxis})
- Im Midterm-Quiz können **insgesamt bis zu 6 Bonuspunkte** erreicht werden ($p_{Midterm}$, skaliert von Midterm-Quiz Punktzahl)
- Die Gesamtzahl der **möglichen Bonuspunkte beträgt demnach 18**, wovon jedoch **max. 15 Bonuspunkte angerechnet werden**, d. h. sofern Sie mehr als 15 Bonuspunkte erzielen, werden dennoch nur 15 angerechnet
- Der Bonus wird auf das Ergebnis der Modulprüfung addiert, sofern diese **ohne Bonus mindestens mit der Note 4,0 bestanden** wurde
- Der Bonus wird auch auf die Retake angerechnet

$note : [0; 105] \rightarrow \text{Note}$ Note gemäß Notenschlüssel

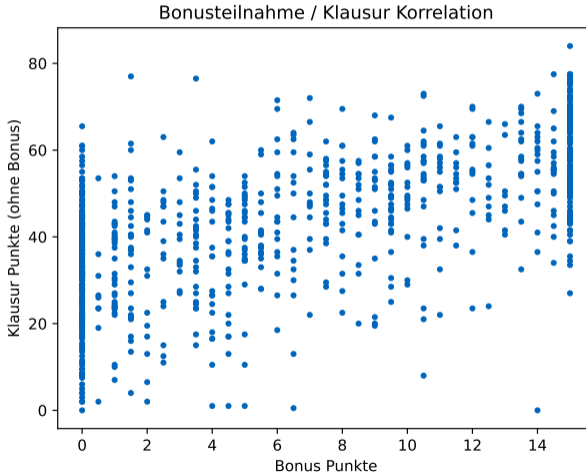
$$p_{Bonus} = \left[\min \left(p_{Prog} + p_{Praxis} + p_{Midterm}, 15 \right) \right]_{1/2} \quad \text{mit } p_{Prog}, p_{Praxis}, p_{Midterm} \in [0; 6]$$

$$note_{Bonus} = \begin{cases} note(p_{Klausur}) & \text{wenn } note(p_{Klausur}) > 4.0 \\ note(p_{Klausur} + p_{Bonus}) & \text{sonst} \end{cases}$$

Warum werden max. 15 Punkte angerechnet?

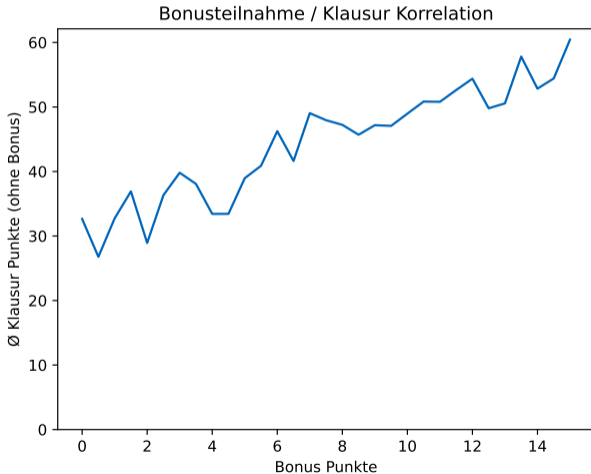
Wir möchten es auch dann ermöglichen, den maximal erzielbaren Bonus zu erhalten, wenn man nicht an allen Aufgaben teilnehmen kann.

Daten aus Endterm 2024



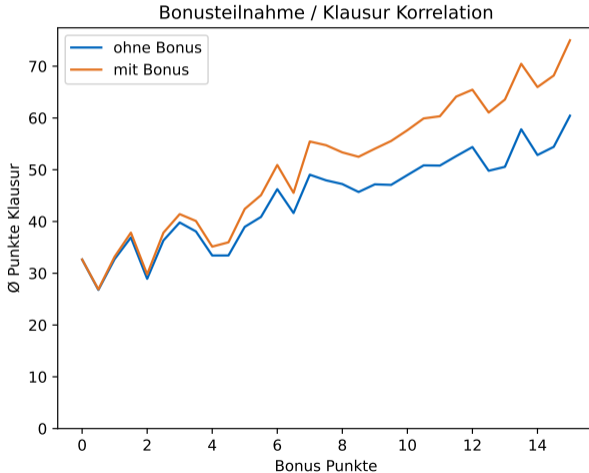
- Frühzeitige und vertiefte Wiederholung des Stoffes während des Semesters erhöht den Lernerfolg
 - Es ist eine Korrelation zwischen erreichten Bonuspunkten und den erreichten Punkten in der Klausur sichtbar
 - Dieser Effekt verstärkt sich durch die Bonuspunkte nochmals zusätzlich
- ⇒ Eine Teilnahme an der Bonusregelung ist deshalb sehr zu empfehlen!

Daten aus Endterm 2024



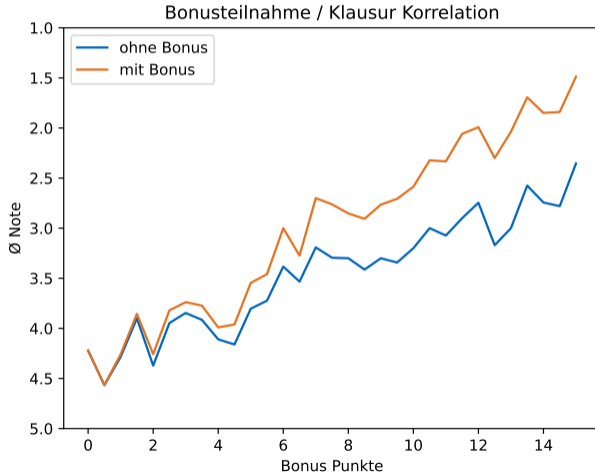
- Frühzeitige und vertiefte Wiederholung des Stoffes während des Semesters erhöht den Lernerfolg
 - Es ist eine Korrelation zwischen erreichten Bonuspunkten und den erreichten Punkten in der Klausur sichtbar
 - Dieser Effekt verstärkt sich durch die Bonuspunkte nochmals zusätzlich
- ⇒ Eine Teilnahme an der Bonusregelung ist deshalb sehr zu empfehlen!

Daten aus Endterm 2024



- Frühzeitige und vertiefte Wiederholung des Stoffes während des Semesters erhöht den Lernerfolg
 - Es ist eine Korrelation zwischen erreichten Bonuspunkten und den erreichten Punkten in der Klausur sichtbar
 - Dieser Effekt verstärkt sich durch die Bonuspunkte nochmals zusätzlich
- ⇒ Eine Teilnahme an der Bonusregelung ist deshalb sehr zu empfehlen!

Daten aus Endterm 2024



- Frühzeitige und vertiefte Wiederholung des Stoffes während des Semesters erhöht den Lernerfolg
 - Es ist eine Korrelation zwischen erreichten Bonuspunkten und den erreichten Punkten in der Klausur sichtbar
 - Dieser Effekt verstärkt sich durch die Bonuspunkte nochmals zusätzlich
- ⇒ Eine Teilnahme an der Bonusregelung ist deshalb sehr zu empfehlen!

Wissenschaftliches Fehlverhalten, Unterschleif und Plagiate

- Programmier- und Praxisaufgaben erfolgen in Einzelarbeit
 - Diskussion von Konzepten und Ansätzen in Gruppe möglich
 - Coding selbst muss alleine erfolgen
- Es werden Plagiatschecks durchgeführt
- Eindeutige Hinweise auf Plagiate führen zum Ausschluss aller Beteiligten aus dem Bonussystem
 - alle Programmier- und Praxisaufgaben und das Midterm-Quiz werden mit 0 Punkten bewertet
- Kopien und Adaptionen externer Quellen wie z. B. StackOverflow sind eindeutig zu kennzeichnen
- Umbenennungen und Umsortierungen stellen **keine** signifikante Eigenleistung dar
- Täuschung und Ordnungsverstöße in einer Prüfung werden gemäß APSO gemeldet
 - Gesamter Kurs nicht bestanden
 - Eine solche Prüfung kann maximal einmal wiederholt werden (§24)
 - Täuschung kann den Ausschluss aus dem Studiengang nach sich ziehen (§22)

- Vorlesungsunterlagen und Übungsmaterialien werden über HTTP und Git¹ bereitgestellt:

<https://grnvs.net>

- Alle Unterlagen, die einem gesonderten Urheberrecht unterliegen (beispielsweise Scans von Auszügen aus Büchern), sind nur nach Authentifizierung über Git zu erreichen
- Sollten Sie nicht an der TUM studieren (beispielsweise LMU- oder Schülerstudenten), wenden Sie sich bitte an die Übungsleitung: **grnvs@net.cit.tum.de**
- Für den Zugriff auf die Git-Repositories ist eine (ohne Probleme mehrfache) Anmeldung unter **<https://grnvs.net/connect>** nötig
- Für den Zugriff auf Moodle ist die Vorlesungsanmeldung in TUMonline erforderlich
- Vorlesungsaufzeichnungen vom SoSe 2021 sind auf <https://live.rbg.tum.de> zu finden
- Die aktuelle Vorlesung wird als Präsenzveranstaltung ohne Live Streaming und ohne Aufzeichnung angeboten

Hinweis: Aktuelle Informationen zur Vorlesung erhalten Sie ausschließlich über Moodle. Dies betrifft insbesondere Informationen zum Stand der Vorlesung oder Updates zu Klausuren. Moodle benachrichtigt Sie bei neuen Posts via Email, sofern Sie die Benachrichtigung nicht deaktiviert haben.

¹ Zum Klonen muss man sich vorher in TUMonline zum Kurs einschreiben und sich im Rahmen der Anmeldung zu den Programmieraufgaben auf <https://grnvs.net/connect> registrieren

Vorlesungsunterlagen

Hinweise zum Urheberrecht

Es sollte klar sein, dass **alle Vorlesungsunterlagen einschließlich Übungsaufgaben, Klausuren und deren Lösungsvorschläge** dem Urheberrecht unterliegen.

Sie dürfen diese Unterlagen

- ✓ zur Vorbereitung auf die Prüfung,
- ✓ als Nachschlagewerk sowie
- ✓ Inhalte daraus (z. B. Abbildung) mit **entsprechender Quellenangabe** für eigene, nicht kommerzielle Arbeiten verwenden.

Sie dürfen insbesondere **nicht** die Unterlagen

- × vervielfältigen oder verbreiten,
- × im Internet teilen (auch nicht auf anmeldepflichtigen Plattformen, z. B. Studoku),
- × ohne Quellenangabe für eigene Projekte nutzen,
- × für kommerzielle Zwecke nutzen,
- × auf irgendeine andere als oben genannte Art verwenden.

Organisatorisches zur Vorlesung

Zusammenfassung der einzelnen Kapitel

Entstehung des Internets

Schichtenmodelle

Literaturangaben

Kapitel 1: Physikalische Schicht

1. Signale, Information und deren Bedeutung

- Was sind Signale?
- Information und Entropie

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

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

Kapitel 3: Vermittlungsschicht

1. Vermittlungsarten

- Leitungsvermittlung
- Nachrichtenvermittlung
- Paketvermittlung

2. Adressierung im Internet

- Internet Protocol version 4 (IPv4)
 - Adressauflösung (ARP)
 - Internet Control Message Protocol (ICMP)
 - Adressklassen (für Classful Routing)
 - Subnetting und Präfixe (für Classless Routing)
- Internet Protocol version 6 (IPv6)
 - Stateless Addresss Autoconfiguration (SLAAC)
 - Internet Control Message Protocol v6 (ICMPv6)
 - Neighbor Discovery Protocol (NDP)

3. Routing

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

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)

1. Schichten

- Vor- und Nachteile verschiedener Schichtenmodelle

2. Sitzungsschicht

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

3. Darstellungsschicht

- Datenkompression (Huffman Code)

4. Anwendungsschicht

- Namensauflösung im Internet (DNS)
- HTTP
- SMTP

Organisatorisches zur Vorlesung

Zusammenfassung der einzelnen Kapitel

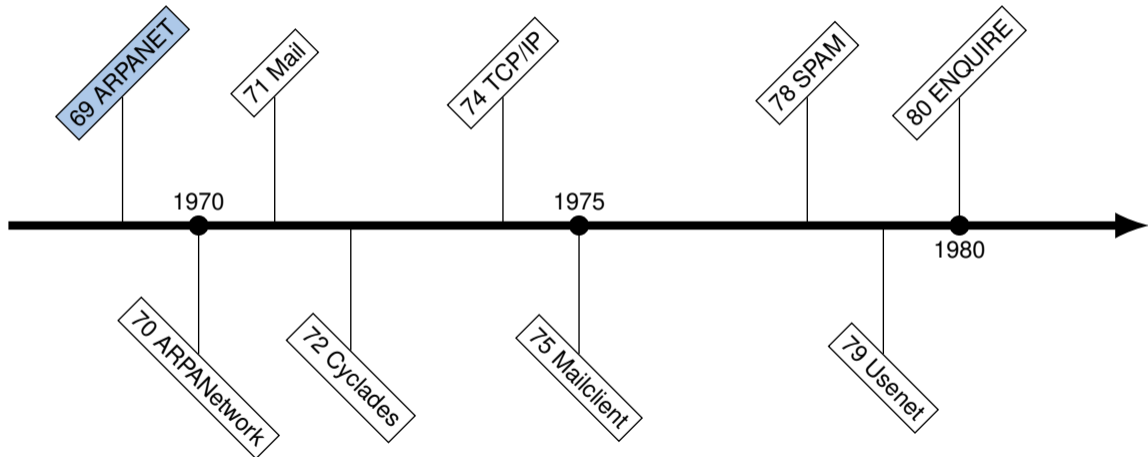
Entstehung des Internets

Schichtenmodelle

Literaturangaben

Entstehung des Internets

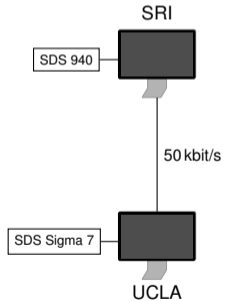
Übersicht bis 1980



Entstehung des Internets

ARPANET mit den ersten 4 Knoten

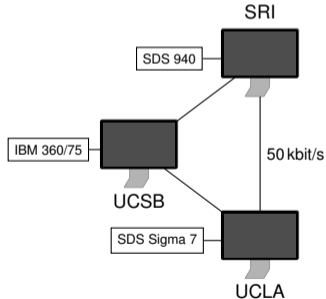
- University of California, Los Angeles (UCLA) Sept. 1969
- Stanford Research Institute (SRI) Okt. 1969



Entstehung des Internets

ARPANET mit den ersten 4 Knoten

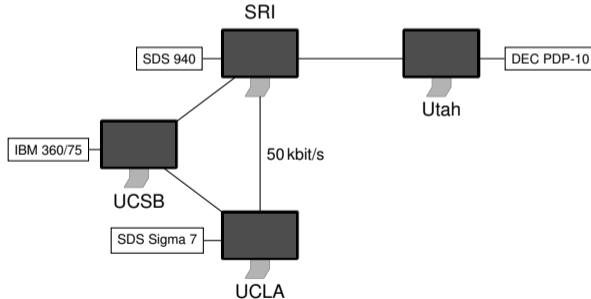
- University of California, Los Angeles (UCLA) Sept. 1969
- Stanford Research Institute (SRI) Okt. 1969
- UC Santa Barbara (UCSB) Nov. 1969



Entstehung des Internets

ARPANET mit den ersten 4 Knoten

- University of California, Los Angeles (UCLA) Sept. 1969
- Stanford Research Institute (SRI) Okt. 1969
- UC Santa Barbara (UCSB) Nov. 1969
- University of Utah Dez. 1969

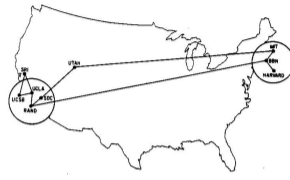


Entstehung des Internets

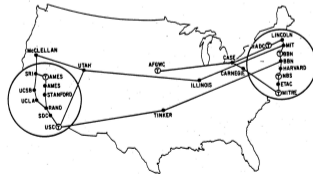
ARPANET von 1969 bis 1977



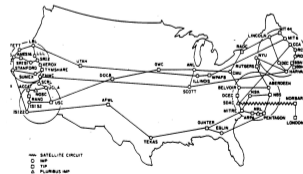
ARPANET 1969, 4 Knoten



ARPANET 1970, 9 Knoten



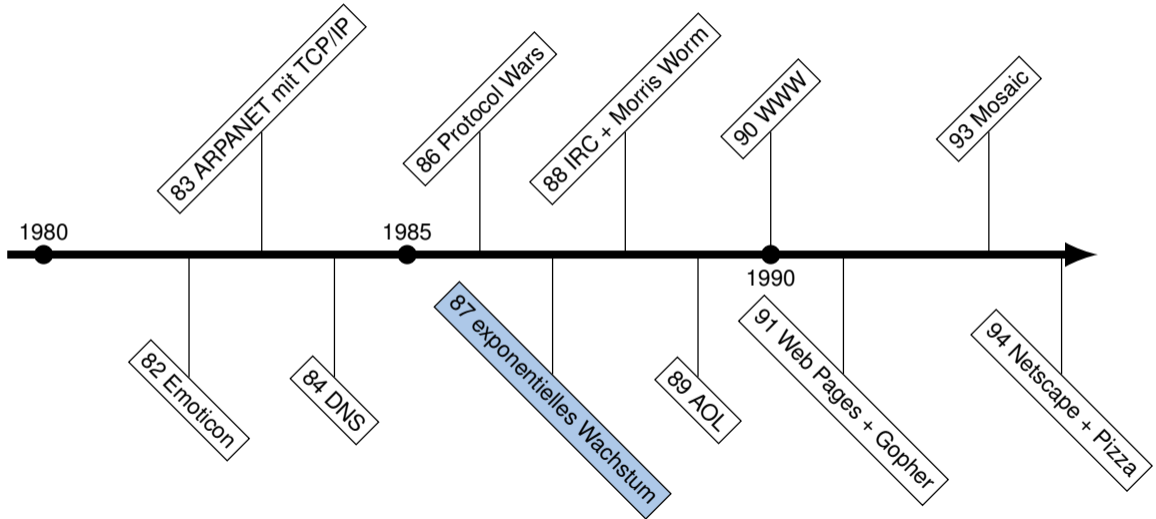
ARPANET 1972, 25 Knoten



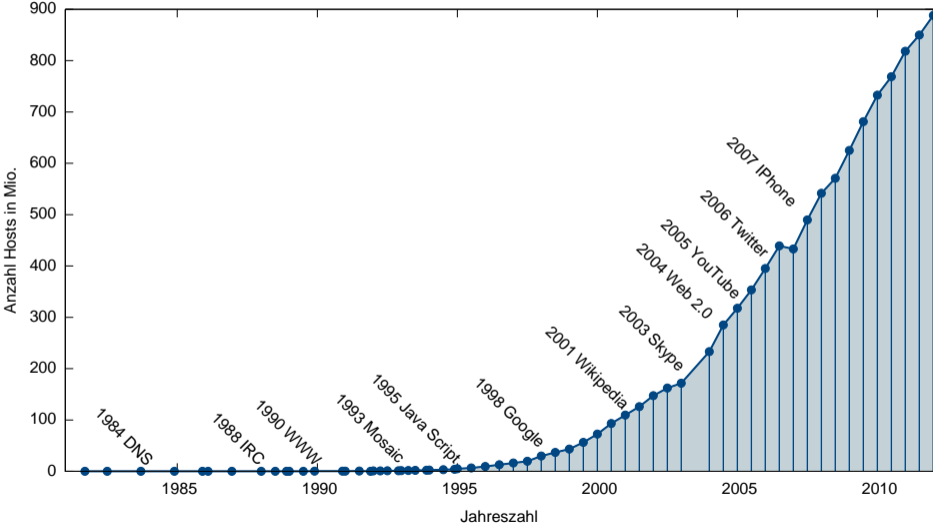
ARPANET 1977, 58 Knoten

Entstehung des Internets

Übersicht von 1980 bis 1994

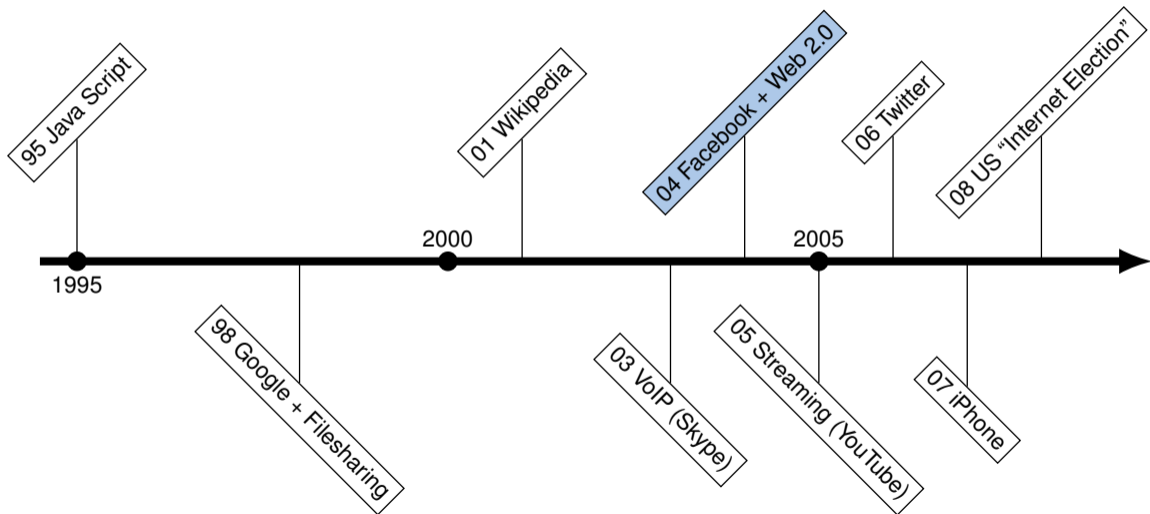


Anzahl der Hosts von 1981 bis 2012



Entstehung des Internets

Übersicht ab 1994



Strategic Positioning: The Web as platform

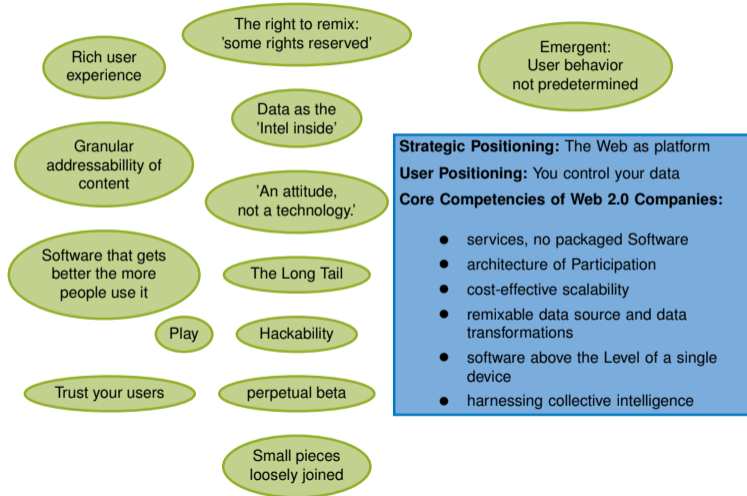
User Positioning: You control your data

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

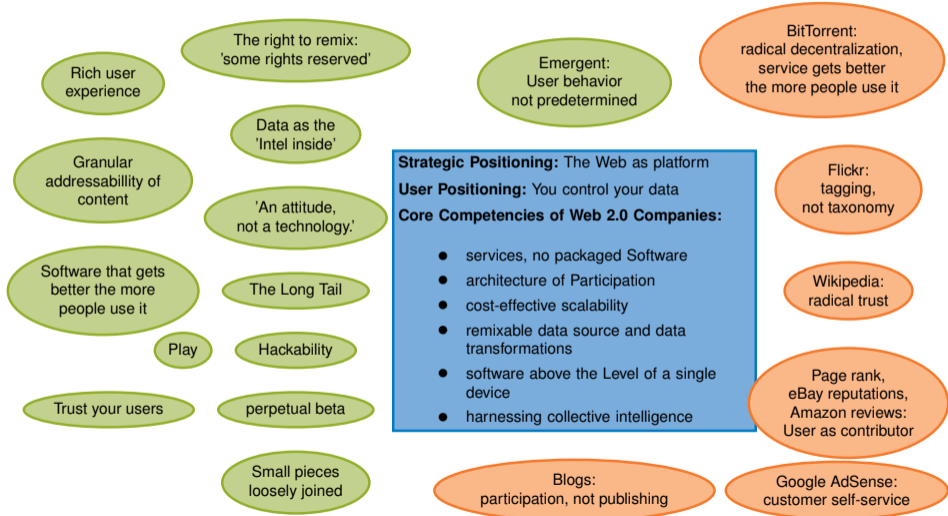
Entstehung des Internets

Web 2.0 Meme Map, by Tim O'Reilly 2005 [2]



Entstehung des Internets

Web 2.0 Meme Map, by Tim O'Reilly 2005 [2]



Entstehung des Internets

Das Internet heute



² Verbindungen zwischen Autonomen System im Internet (2007) [1]

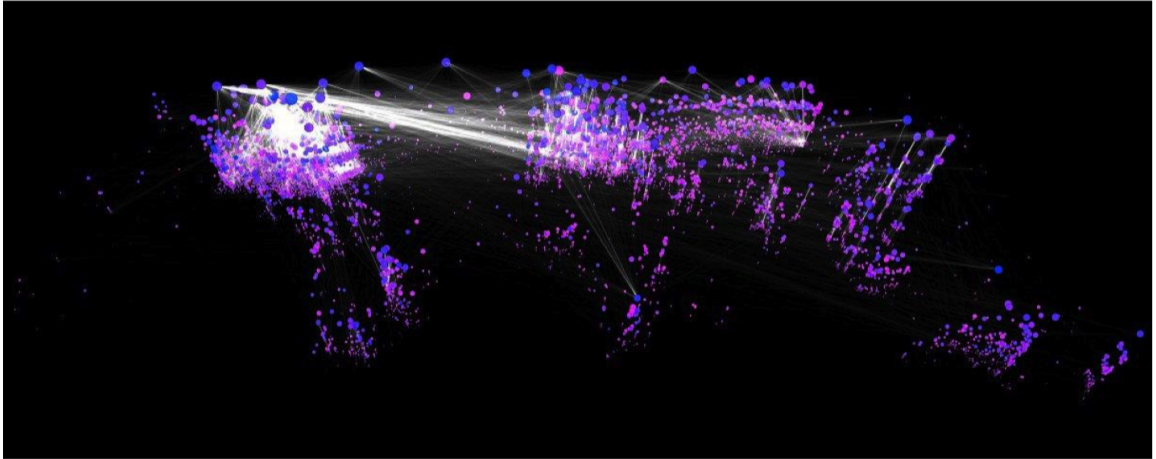


Abbildung 1: Lange existierende Autonome Systeme blau hervorgehoben [1]

Kapitel 0: Organisatorisches und Einführung

Organisatorisches zur Vorlesung

Zusammenfassung der einzelnen Kapitel

Entstehung des Internets

Schichtenmodelle

Was sind Schichtenmodelle?

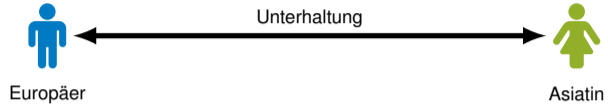
Wozu sind Schichtenmodelle gut?

Das ISO/OSI-Modell

Literaturangaben

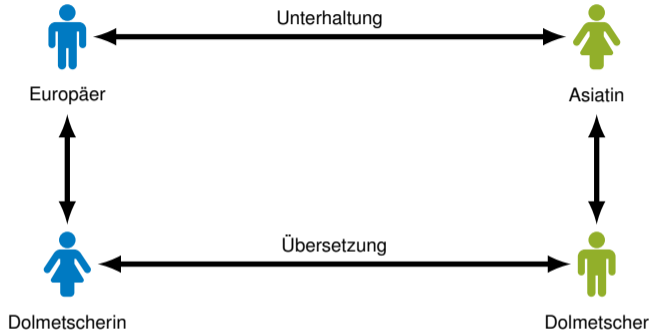
Was sind Schichtenmodelle?

Ein einfaches Beispiel



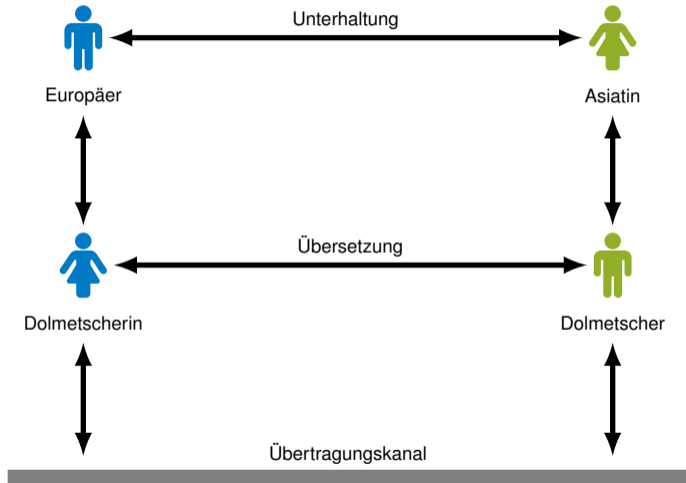
Was sind Schichtenmodelle?

Ein einfaches Beispiel



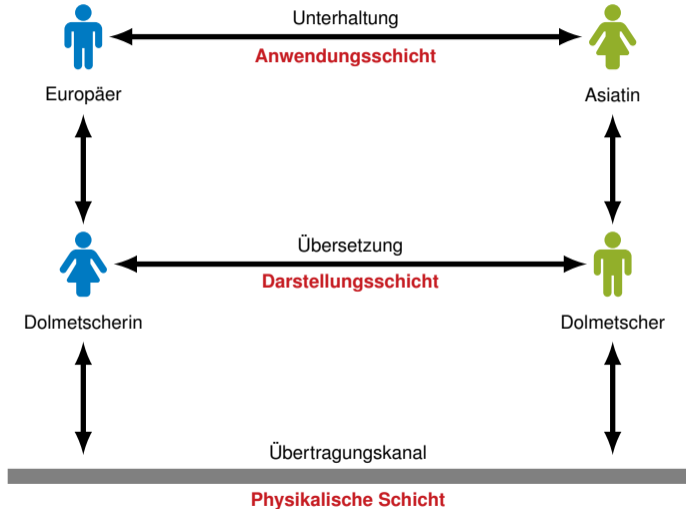
Was sind Schichtenmodelle?

Ein einfaches Beispiel



Was sind Schichtenmodelle?

Ein einfaches Beispiel



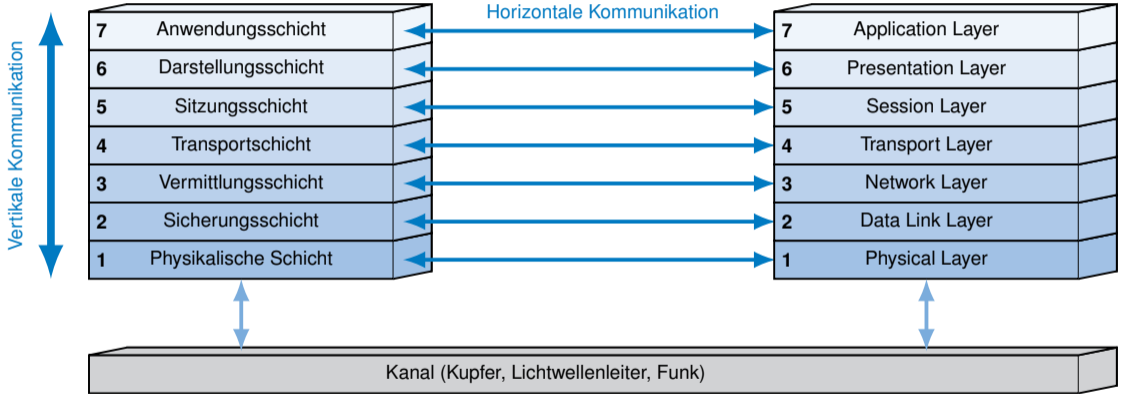
Wozu sind Schichtenmodelle gut?

- 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

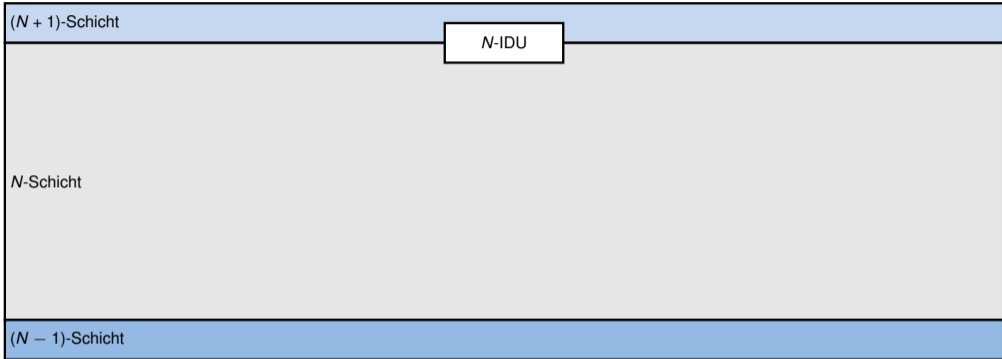
- Entwickelt zwischen 1979 und 1983 von der [International Organization for Standardization \(ISO\)](#)
- OSI model = [Open Systems Interconnection model](#)
- Unterteilt die Architektur des Systems, das die Kommunikationsfunktionalität erbringt, in insgesamt **7 Schichten**
- Jede Schicht erbringt bestimmte Dienste (z. B. Aufteilen einer Nachricht in kleinere Pakete)
- Keine Aussage, wie diese Dienste zu erbringen sind

Das ISO/OSI-Modell

Schematische Darstellung des OSI-Modells

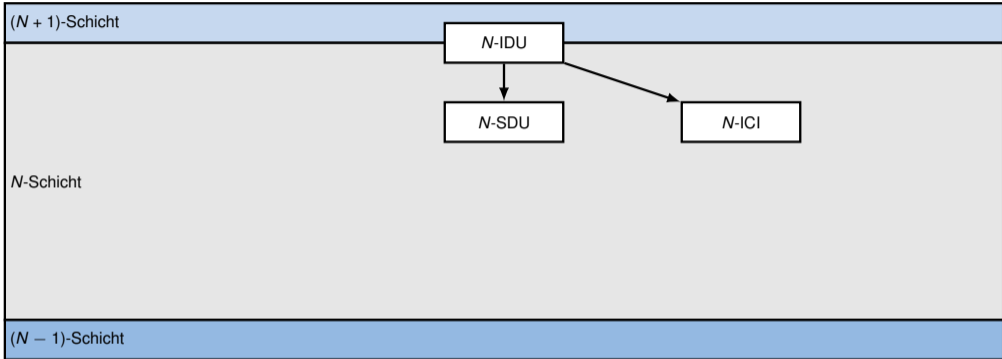






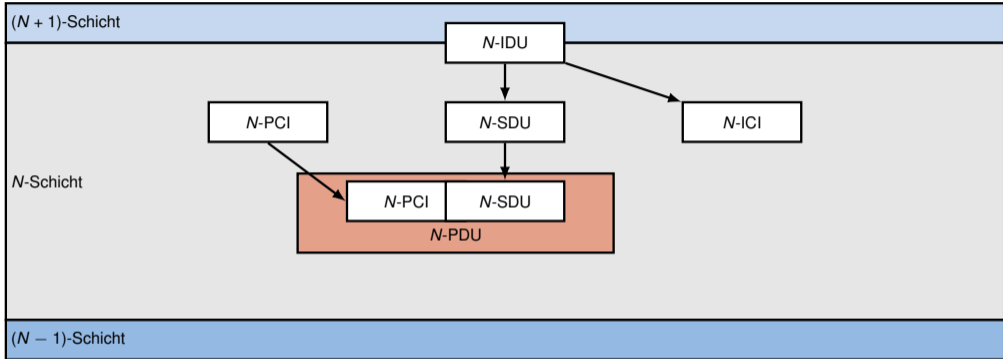
Die (N + 1)-Schicht nimmt Dienste der N-Schicht in Anspruch:

- Die N-Schicht erhält eine [Interface Data Unit \(IDU\)](#) von der (N + 1)-Schicht.



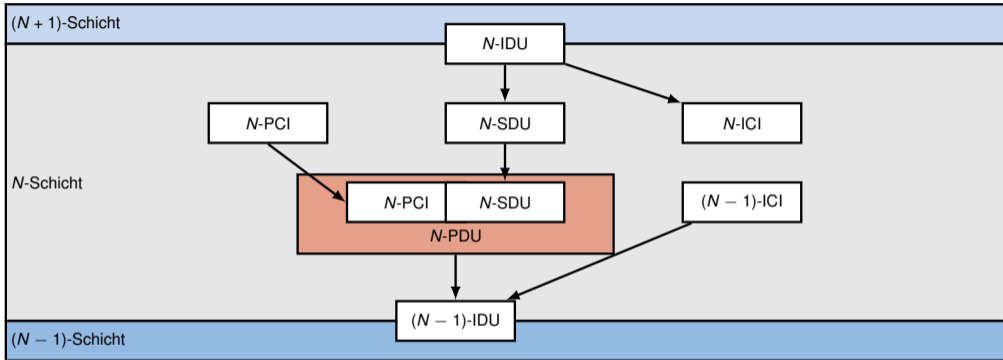
N-IDU enthält aus Sicht der N-Schicht

- Nutzdaten ([Service Data Unit \(SDU\)](#)) und
- Kontrollinformationen ([Interface Control Information \(ICI\)](#)), welche zum Erbringen des Dienstes notwendig sind (z. B. Länge der SDU oder Adressinformationen).



Die N -Schicht

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



Die N -Schicht nutzt den Dienst der $(N - 1)$ -Schicht.

- Sie erzeugt eine $(N - 1)$ -ICI, und
- übergibt diese zusammen mit der N -PDU als $(N - 1)$ -IDU der nächst niedrigeren Schicht

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

- die (ggf. bearbeiteten) Nutzdaten der $(N + 1)$ -Schicht, d. h. die SDU, sowie
- Protokollsteuerungsinformationen (Protocol Control Information - 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

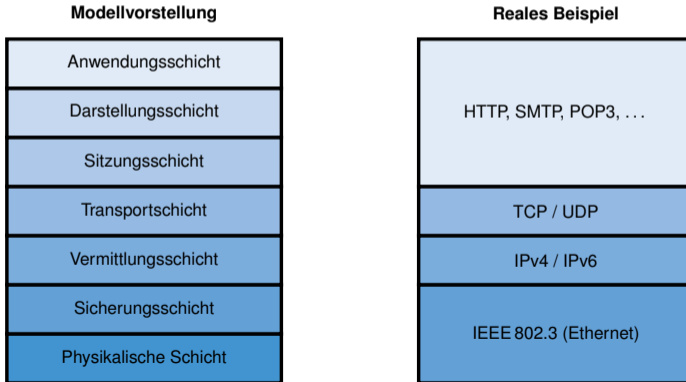
- **Segmenten** auf der Transportschicht,
- **Paketen** auf der Vermittlungsschicht bzw.
- **Rahmen** (engl. **Frames**) auf der Sicherungsschicht.

Diese Unterscheidungen ermöglichen es, implizit die gerade betrachtete Schicht anzugeben. Die Verwendung der Begriffe in der Literatur ist allerdings nicht immer einheitlich.

Das ISO/OSI-Modell

Schwächen des ISO/OSI-Modells

- Die Trennung der Schichten widerspricht manchmal anderen Interessen (z. B. der Effizienz)
- Einige Protokollmechanismen sind nicht klar einer bestimmten Schicht zuzuordnen, bzw. arbeiten sogar auf mehreren Schichten (**Cross Layer**)
- Die Zuordnung von Protokollen auf einzelne Schichten kann vom konkreten Einsatz der Protokolle abhängen



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

Organisatorisches zur Vorlesung

Zusammenfassung der einzelnen Kapitel

Entstehung des Internets

Schichtenmodelle

Literaturangaben

- [1] C. Harrison.
World City-to-City Connections.
<http://www.chrisharrison.net/index.php/Visualizations/InternetMap>.
- [2] T. O'Reilly.
O'Reilly Network: What Is Web 2.0, Sept. 2005.
- [3] E. Stein.
Taschenbuch Rechnernetze und Internet, chapter Das OSI-Modell, pages 22–28.
Fachbuchverlag Leipzig, 2. edition, 2004.

Literaturangaben

Weiterführende Literatur zur Internet-Architektur und Entwurfsprinzipien

Konzeptionelle Vorarbeiten zum ARPANET

- Paul Baran, On Distributed Communication Networks, RAND Corporation, 1962-1964
<https://www.rand.org/pubs/papers/P2626.html>
<https://web.archive.org/web/20101228070851/http://www.rand.org:80/about/history/baran-list.html>

Veröffentlichungen zu Internet-Entwurfsprinzipien

- J.H. Saltzer, D.P.Reed, D.D.Clark, End-To-End Arguments in System Design, ACM TOCS, Vol 2, Number 4, November 1984
- David D. Clark, The Design Philosophy of the DARPA Internet Protocols, Proc SIGCOMM 88, ACM CCR Vol 18, Number 4, August 1988

RFCs zur Internet-Architektur

- Brian Carpenter, Ed., Architectural principles of the Internet, RFC 1958, June 1996.
- Randy Bush, David Meyer, Internet Architectural Guidelines, RFC 3439, December 2002.