

Datenkommunikation und Sicherheit

Kapitel 1: Einführung

Klaus Wehrle

Communication and Distributed Systems

Chair of Computer Science 4

RWTH Aachen University

<http://www.comsys.rwth-aachen.de>

10
Jahre
COMSYS
Lernen. Forschen. Grillen!



RWTHAACHEN
UNIVERSITY

I-1

Kapitel 1: Einführung

- **Einführung und Begriffe**

- ▶ Was ist Datenkommunikation?
- ▶ Information, Daten, Signale
- ▶ Netze

- **Allgemeine Grundlagen**

- ▶ Dienste
- ▶ Protokolle und Schichten
- ▶ Kommunikationsarchitekturen

Was ist Datenkommunikation?

• Datenkommunikation:

- ▶ Verarbeitung und Transport von digitalen Daten zwischen Computern und/oder anderen Geräten (i.A. über kleine oder größere Entfernungen)
- ▶ Oder: Datenkommunikation ist der Oberbegriff für jeden Datenaustausch über *immaterielle Träger* und *Entfernungen* zwischen Menschen und/oder Maschinen
 - Immaterielle Träger:
 - Energieflüsse, meist elektrische Ströme, elektromagnetische Wellen
 - Gegensatz: materieller Datentransport (z.B. Brief, Versand von USB-Sticks)
 - Noch ein Gegensatz: Datenbanken, Dateien:
Datentransport über die Zeit hinweg, der Ort bleibt gleich

Als Datenkommunikation bezeichnet man generell alle Methoden, die Nutzdaten von einer Quelle (Sender) hin zu einer Senke (Empfänger) übermitteln. Gemeinsam haben all diese Methoden, dass technisch gesehen der Sender zur Übermittlung der Daten eine physikalische Größe (z.B. elektrische Spannung) zeitlich variiert, und dass der Empfänger diese Variationen misst.

Datenaustausch zwischen Geräten kann auch z.B. über CDs oder USB-Stick erfolgen, aber dies ist keine Datenkommunikation mehr.

Der Begriff *Daten*

• Daten (universell)

- ▶ Darstellung von Sachverhalten (Fakten), Konzepten, Vorstellungen und Anweisungen *in formalisierter Weise*, die für die Kommunikation, Interpretation und die Verarbeitung durch Menschen und/oder technische Mittel geeignet ist
- ▶ Allgemeine Beispiele für Datendarstellungen:
 - Gesprochene Sprache
 - Zeichen-/Gebärden-Sprache
 - Geschriebene Sprache

Gegenstände des Denkens:
Fakten, Konzepte,
Vorstellungen, Modelle usw.
(Informationen)

Konventionen zur
Darstellung von
Denkinhalten

Daten als formalisierte
Darstellung von Denkinhalten

Modell zur Erzeugung von Daten
durch den Menschen

Was sind „Daten“ in der Datenkommunikation?

Ausgangspunkt bei der Datenkommunikation sind die unterschiedlichsten Gegenstände unseres Denkens. Dazu zählen Fakten, Konzepte, Vorstellungen und Modelle.

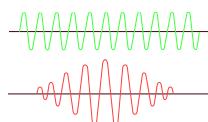
Bevor diese Gegenstände in irgendeiner Form übertragen werden können, müssen sie in einem ersten Schritt in eine formalisierte Darstellung transformiert werden, in die sogenannten *Daten*. Unter dem Begriff „Daten“ versteht man also die Darstellung von Sachverhalten (Fakten), Konzepten, Vorstellungen und Anweisungen in formalisierter Weise, die für die Kommunikation, Interpretation und die Verarbeitung durch Menschen und/oder technische Mittel geeignet ist. Beispiele für Daten sind z.B. Sätze in gesprochener Sprache, geschriebener Sprache, aber auch in Zeichen- und Gebärdensprache.

Der Begriff *Signal*

• Signal

- ▶ *Physikalische Darstellung (Repräsentation) von Daten durch charakteristische räumliche und/oder zeitliche Veränderungen der Werte physikalischer Größen*

- ▶ Reale physikalische Repräsentation abstrakter Darstellungen (der Daten)



Gegenstände unseres Denkens

Konventionen zur Darstellung von Denkinhalten

Abstrakte Welt: Daten als formalisierte Darstellung

Konventionen zur Darstellung von Daten

Physikalische Welt: Signale als reale Darstellung von Daten



I-5

Wichtig im Kontext der Vorlesung ist hingegen der Begriff „Signal“. Während Daten lediglich eine abstrakte Darstellung einer für den Menschen (meist) wesentlichen Information darstellen, ist es für die Datenkommunikation nötig, Daten über einen physikalischen Kanal zu transportieren (Kabel, Glasfaser, Funk, ...). Die Daten müssen dazu in eine physikalische Form gebracht werden, die einen Transport über den zu verwendenden Kanal erlauben. Am Beispiel Kupferkabel: Daten können in Form von Strompulsen über das Kupferkabel transportiert werden. Durch Veränderung z.B. der Stärke des Stroms können unterschiedliche Daten (0 oder 1) repräsentiert werden.

Oft erfolgt eine Verwechslung (oder unsaubere Trennung) der Begriffe „Daten“ und „Signal“, da Daten nur in objektivierter, physikalisch dargestellter Form, d.h. als Signale, erfassbar, speicherfähig, übertragbar und verarbeitbar sind. Jede konkrete Datendarstellung ist somit mit einer spezifischen Signalrepräsentation verbunden, weswegen die konzeptionelle Unterscheidung oft nicht unmittelbar sichtbar ist. Beispiele wären:

- Laute einer Sprache (Daten) beim Sprechen als akustische Schwingungen (Signale)
- Druckbuchstaben auf Papier als optische Signale abstrakter Schriftzeichen (Daten)
- Darstellung von Sprachlauten (Daten) durch elektrische Sprechströme (Signale)

Der Begriff *Information*

- **Information:**

- ▶ *Bedeutung, die ein Mensch (bzw. bestimmte Anwendungen) den Daten aufgrund von Vereinbarungen (Konventionen), die ihnen zugrundeliegenden, zuordnen kann*
- ▶ Informationsbegriff bezieht sich damit vorwiegend auf den Menschen
 - Enge Definition des Informationsbegriffs in der Vorlesung (verglichen mit der Alltagssprache)
- Verwendung des Begriffs „Information“ bei präziser Ausdrucksweise in der Datenkommunikation möglichst vermeiden

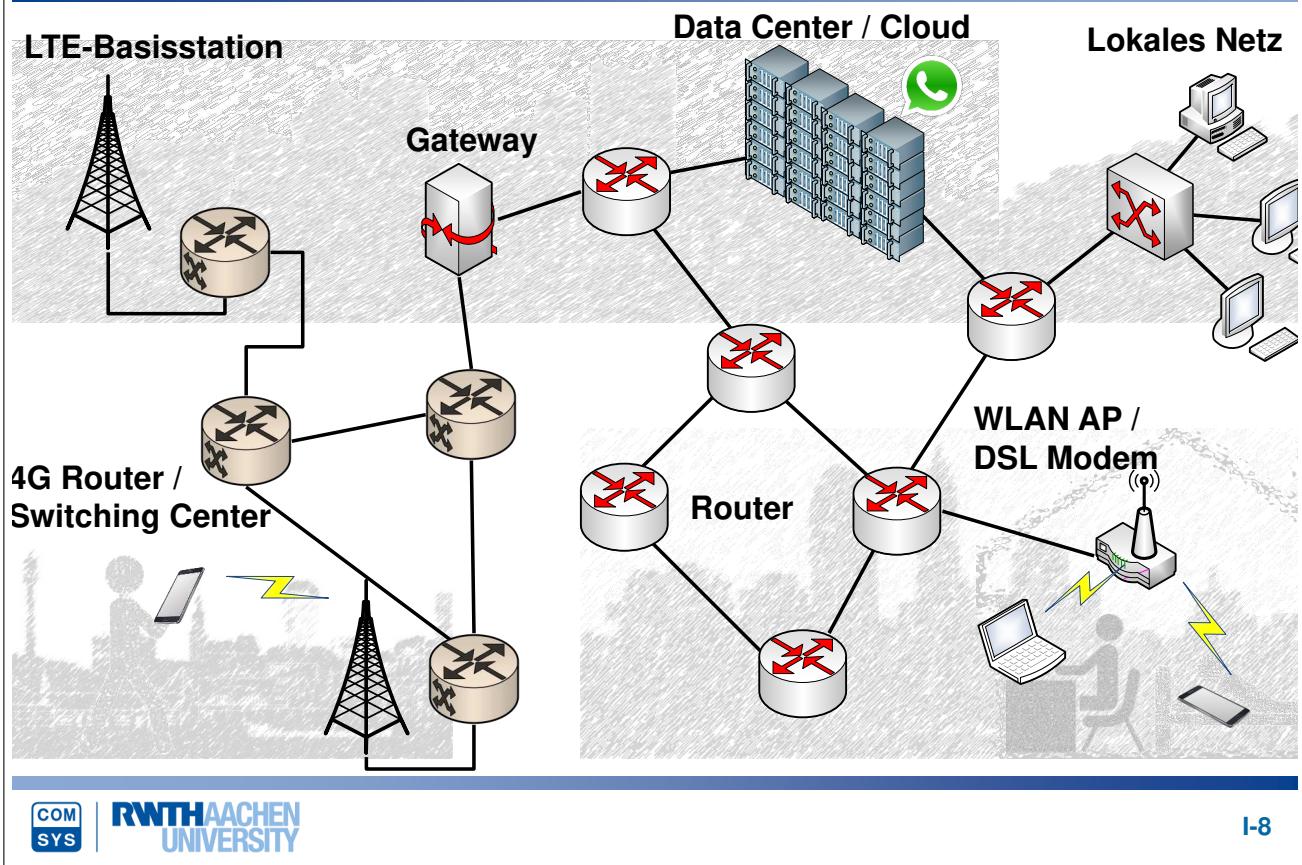
Es ist Aufgabe des Menschen, Daten so zu interpretieren, dass daraus die übertragene *Information* gewonnen wird. An dieser Stelle wird somit die Unterscheidung zwischen Syntax und Semantik sichtbar: Zur Darstellung der Daten genügt eine Syntax, d.h. Regeln, wie gewisse Sachverhalte darzustellen sind, während die Semantik, also die Bedeutung dieser dargestellten Sachverhalte, erst durch die Interpretation durch den Menschen zustande kommt.

Grundlage: vernetztes System



Ein alltägliches Szenario? Ermöglicht wird es durch die Datenkommunikation.

Grundlage: vernetztes System



Grundlage für die Datenkommunikation ist ein vernetztes System, das sich im wesentlichen aus Endsystemen und Vermittlungseinrichtungen (Router, Switch) zusammensetzt. Endsysteme bieten dabei sowohl Personen den interaktiven Zugang zur Datenkommunikation als auch Softwaresystemen und Geräten wie beispielsweise Datenbanken oder Sensoren.

Bezüglich der Vernetzung von Endsystemen und Vermittlungseinrichtungen gibt es keine Einschränkungen – Verbindungen können über Kabel, Funk oder andere Techniken hergestellt werden.

Auf dieser Folie ist ein kleiner Ausschnitt der Kommunikationsmöglichkeiten gezeigt. Ein Smartphone ist via LTE mit einem LTE-Provider verbunden, das andere in seinem lokalen Netz per Wi-Fi mit seinem DSL-Provider. Der LTE-Provider verfügt über sein eigenes Netz, einen sogenannten Backbone, der alle Funkzellen des Providernetzes miteinander verbindet und Daten bzw.

Telefongespräche vermitteln kann. Der DSL-Provider ist über einen Backbone an andere Datennetze („das Internet“) angeschlossen. Zusätzlich zu sehen ist ein kabelgebundenes lokales Netz, das z.B. innerhalb einer Firma alle Rechner und Peripheriegeräte verbindet.

Bestimmte Exchange Points (hier Gateway genannt) verbinden LTE- und Datennetze miteinander. Angeschlossen ist hier auch ein Data Center, in dem z.B. die WhatsApp-Services laufen.

Kategorisierung von Netzen nach Ausdehnung

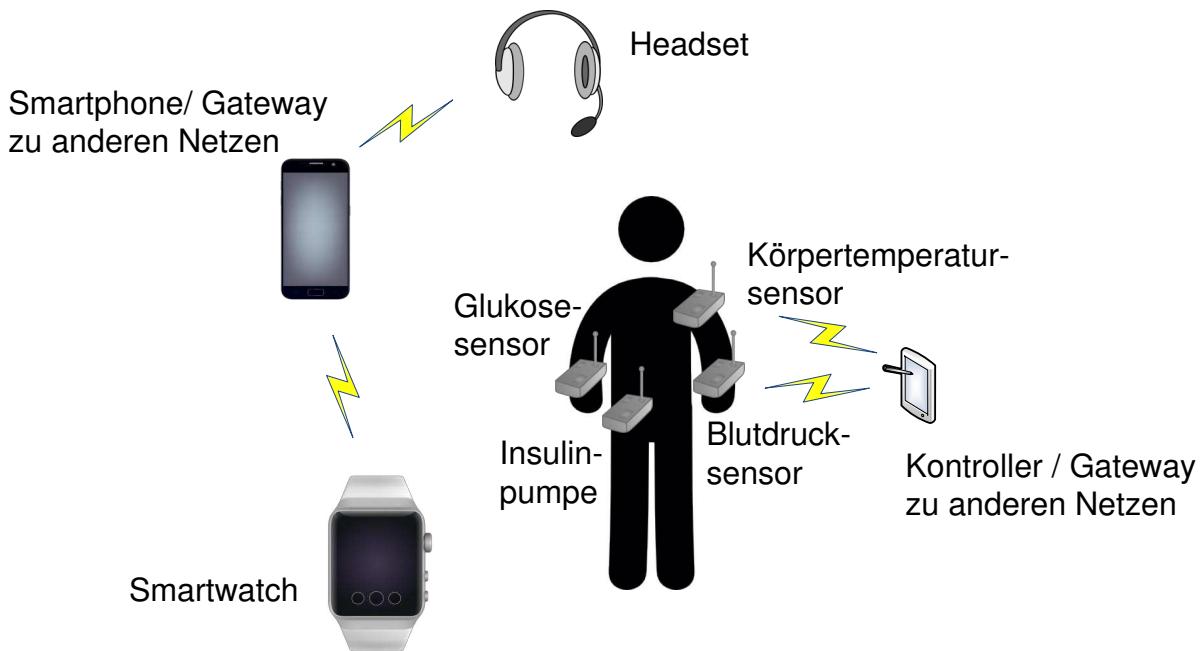
- **Personal Area Network (PAN)**
 - ▶ Vernetzung von persönlichen Geräten eines Benutzers
- **Local Area Network (LAN)**
 - ▶ Heimnetz, Firmennetz
- **Metropolitan Area Network (MAN)**
 - ▶ Campus, Stadtbereich
- **Wide Area Networks**
 - ▶ Länder, Kontinente
- **Internet**
 - ▶ Weltumfassender Zusammenschluss aller Netze

Das bekannteste Netz weltweit ist das Internet. Das Internet selbst ist allerdings kein eigenständiges Netz, sondern beschreibt die Kopplung der unterschiedlichsten Netztypen zu einem weltweiten Kommunikationsverbund. Eine erste Einteilung dieser verschiedenenartigen Netze lässt sich dabei anhand ihrer Ausdehnung vornehmen:

- PAN (*Personal Area Network*): Netz wenige Meter um den Benutzer herum. Beispiel Bluetooth – Kopplung eines drahtlosen Headsets mit dem Handy.
- LAN (*Local Area Network*) : lokales Netz einer Organisation, welches schnellen Datenaustausch innerhalb der Organisation und gemeinsame Nutzung von Ressourcen (Drucker, Dateiserver, ...) erlaubt. Beispiel: WLAN in der eigenen Wohnung.
- MAN (*Metropolitanean Area Network*): Stadtnetz, welches die lokalen Netze innerhalb eines geographischen Bereichs koppeln soll. Beispiel: RWTH-Backbone zur Kopplung aller lokalen Netze der Institute/Lehrstühle der Uni.
- WAN (*Wide Area Network*): Weitverkehrsnetz zur Kopplung lokaler und/oder Stadtnetze innerhalb eines ganzen Landes oder gar Kontinents. Beispiel: X-WIN des DFN (deutsches Forschungsnetz) zur Kopplung aller Universitäten und Forschungseinrichtungen.

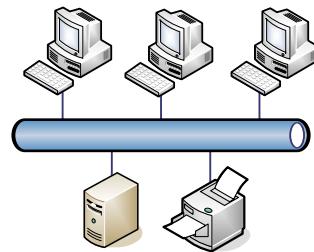
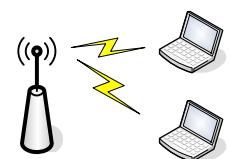
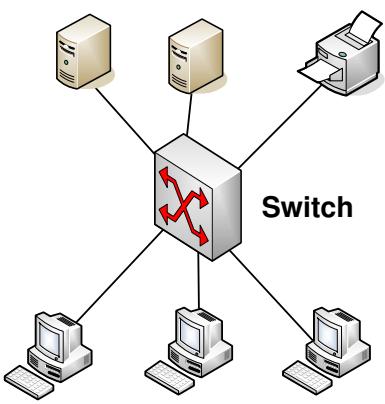
Wir werden später sehen, dass die Flexibilität, praktisch beliebige Netze und damit verknüpfte Übertragungstechniken in einem Netz, dem Internet, zusammenzufassen, dadurch erreicht wird, dass die Kopplung dieser Netze auf einer Protokollebene erfolgt, die oberhalb der Festlegungen von Übertragungsspezifika liegt.

Personal Area Network (und Body Area Network)



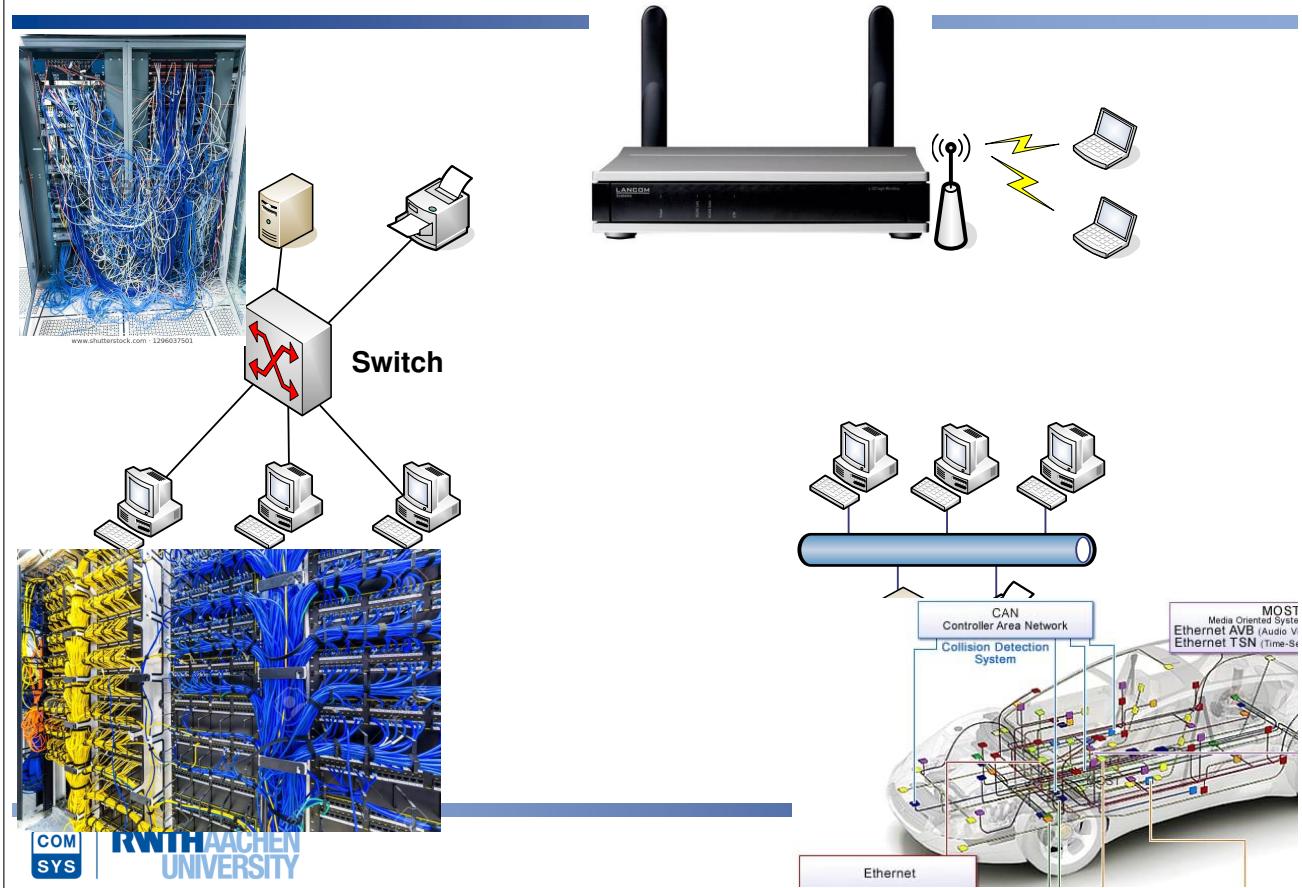
Spezialfall Body Area Network: Vernetzung verschiedener Geräte, die den menschlichen Körper überwachen/kontrollieren.

Local Area Network



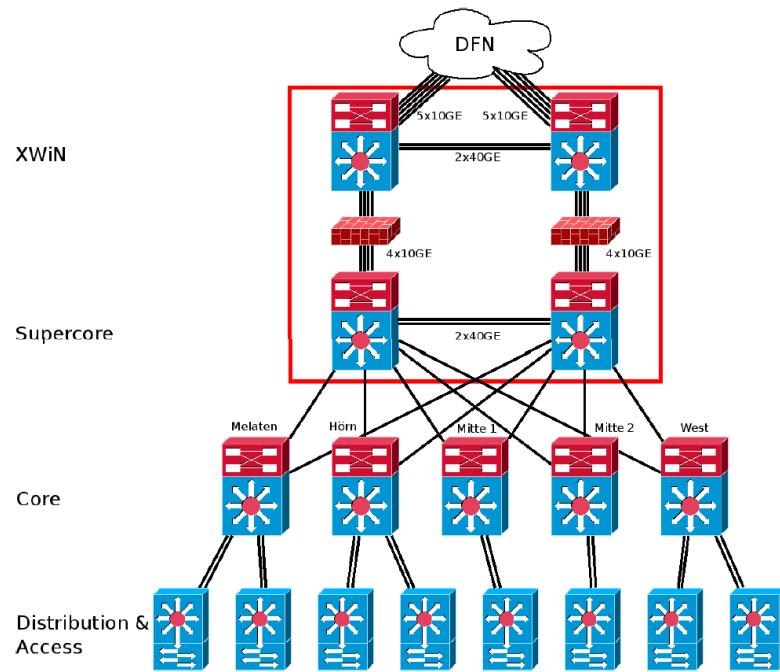
Es gibt unterschiedliche Möglichkeiten, Rechner per Kabel oder Funk zu einem lokalen Netz zu verbinden. Dies wird in Kapitel 3 näher betrachtet.

Local Area Network



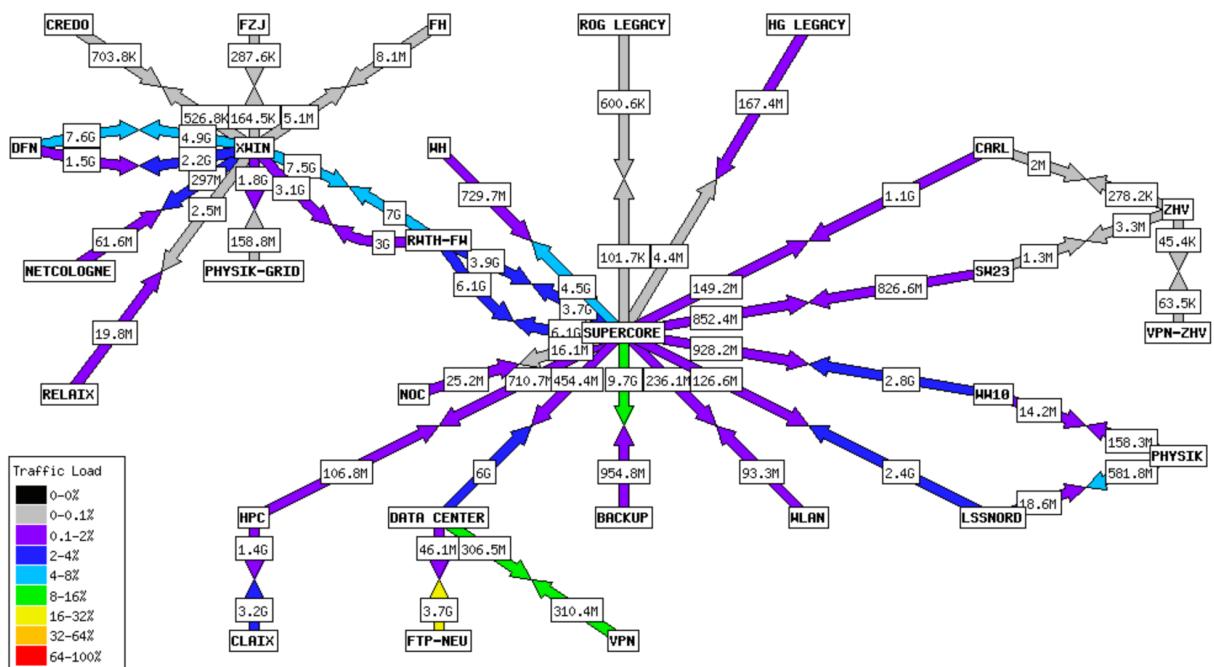
Es gibt unterschiedliche Möglichkeiten, Rechner per Kabel oder Funk zu einem lokalen Netz zu verbinden. Dies wird in Kapitel 3 näher betrachtet.

Metropolitan Area Network – RWTH-Netz



RWTH Weather Map

Created: Apr 09 2021 23:30:01



Deutsches Forschungsnetz: X-WIN

X-WiN-Topologie: Glasfasern

DFN

Glasfaser Bestand
Kernnetzknoten Bestand

Zentraler Knoten Frankfurt –
Verbindung zum Europäischen
Forschungsnetz



Stand: Oktober 2018



RWTH AACHEN
UNIVERSITY

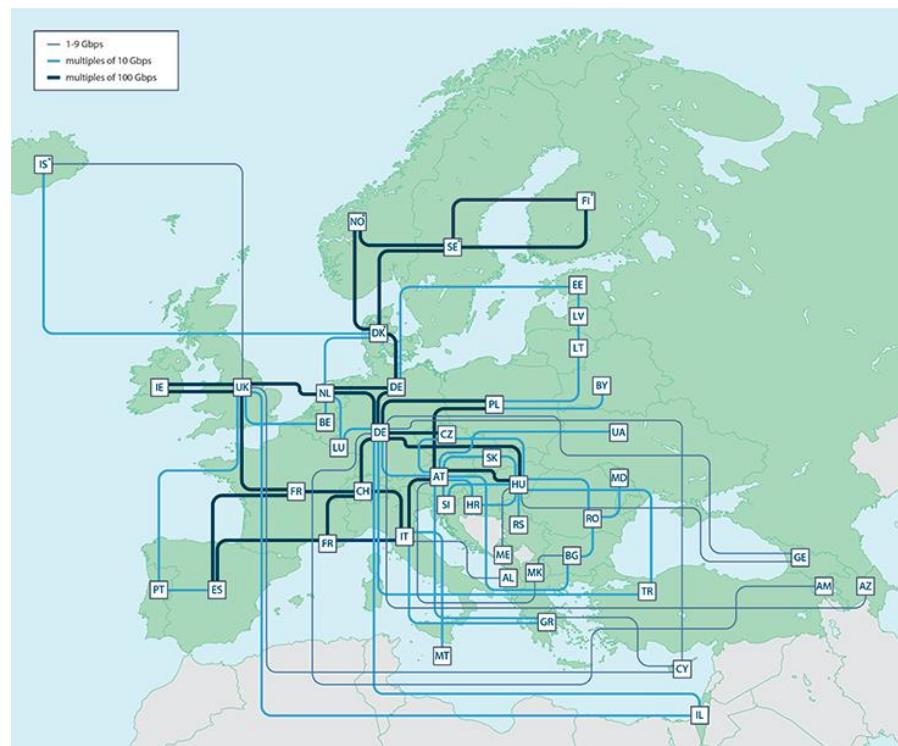
<https://www.dfn.de/xwin/>

I-15

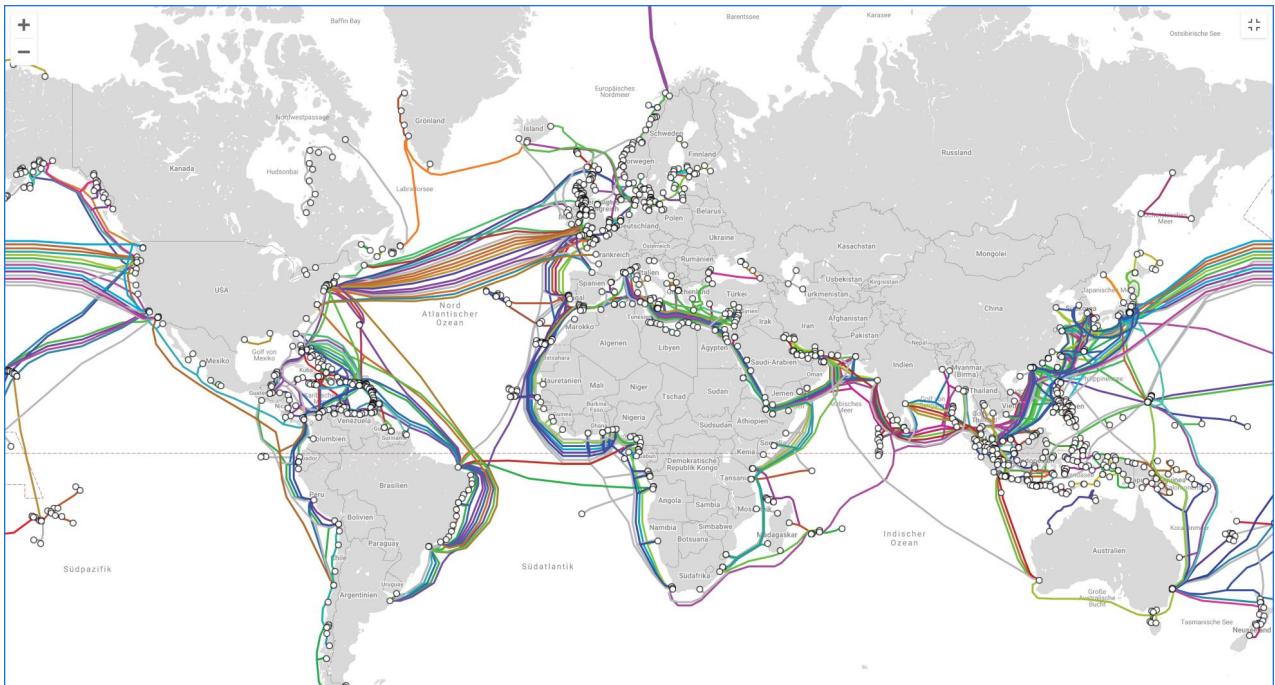
Netzwerke (Beispiel Géant)

Zentraler Knoten
Frankfurt –
Verbindung zum
Europäischen
Forschungsnetz.

In Frankfurt und
Hamburg befinden
sich die
interkontinentalen
Verbindungen.



Verbindungen weltweit



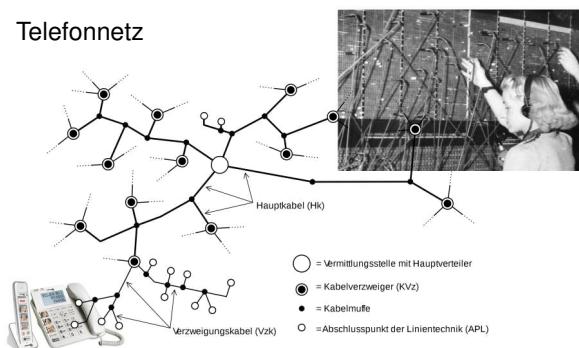
RWTH AACHEN
UNIVERSITY

<https://www.submarinecablemap.com/#/>

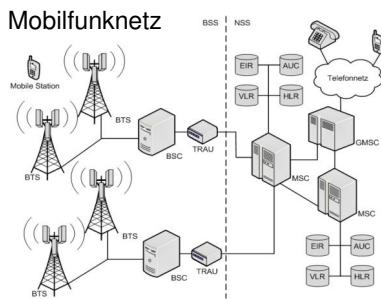
I-17

Weitere Netze...

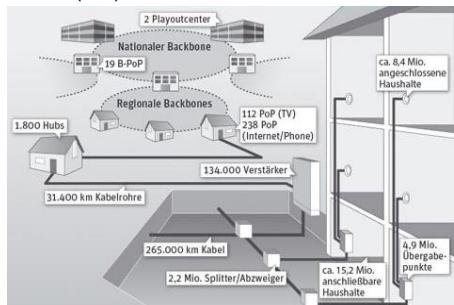
Telefonnetz



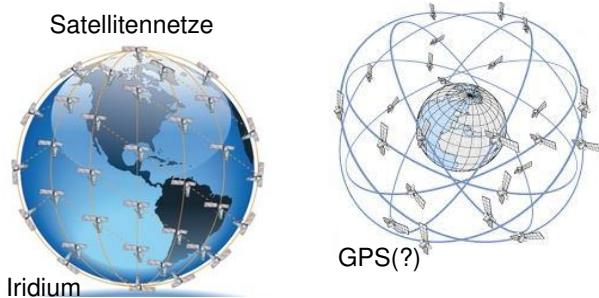
Mobilfunknetz



Kabel(TV)Netz



Satellitennetze



+ Radio, TV, TMC, Alarm/Warn/Melde/Leitsysteme (Straßen/Bahn/Flugverkehr, Industrie, Tsunamie, Pandemien (?!?), uvm)

Datenkommunikation = Rechnernetze?

- **Datenkommunikation**

- ▶ ... setzt die Vernetzung von Geräten voraus (Rechnernetze)
- ▶ ... behandelt die Übertragung von Daten als Signale
- ▶ ... ?

Zur Datenkommunikation ist es nötig, ein vernetztes System vorliegen zu haben. Abhängig von den verwendeten Kommunikationsmedien zur Verbindung der Geräte können die Daten zur Übertragung als konkrete physikalische Signale dargestellt werden.

Bei der Behandlung der Datenkommunikation müssen also Kommunikationsmedien und Signale behandelt werden. Aber: was umfasst die Datenkommunikation noch?

Aufgaben eines Kommunikationssystems

- **Ein Netzwerk ist mehr als nur das verbindende „Kabel“**

- ▶ Wie werden die Daten als Signal dargestellt?
- ▶ Welches angeschlossene System darf wann senden?
- ▶ Wie können Übertragungsfehler erkannt werden?
- ▶ Wie werden Endpunkte gefunden/adressiert?
- ▶ Wie können entfernte Systeme in anderen Netzen erreicht werden?
- ▶ Wie wird Datenstau in einem komplexen Netz vermieden?
- ▶ Wie können Ressourcen ohne zentrale Kontrolle fair genutzt werden?
- ▶ Wie werden die Daten bei den End-Systemen übermittelt?
- ▶ Welche Kodierungsregeln und Semantiken gibt es hierbei?
- ▶ Wie können Nachrichten sicher versendet werden?
- ▶ Was bedeutet „sicher“?

Bei genauerer Betrachtung fällt eine Vielzahl von Aufgaben an, die ein Kommunikationssystem erbringen muss. Es ist nicht damit getan, Daten in Signale zu konvertieren, diese zu übertragen und auf Empfängerseite wieder in Daten zurückzuübersetzen. Es ist eine Vielzahl weiterer Aufgaben notwendig, z.B. die Erkennung von Übertragungsfehlern, die das Medium verursachen kann, eine Zugriffsregelung auf das Medium, Entscheidungen über die Wegwahl zur Weiterleitung von Daten über Zwischenknoten hinweg usw.

Kapitel 1: Einführung

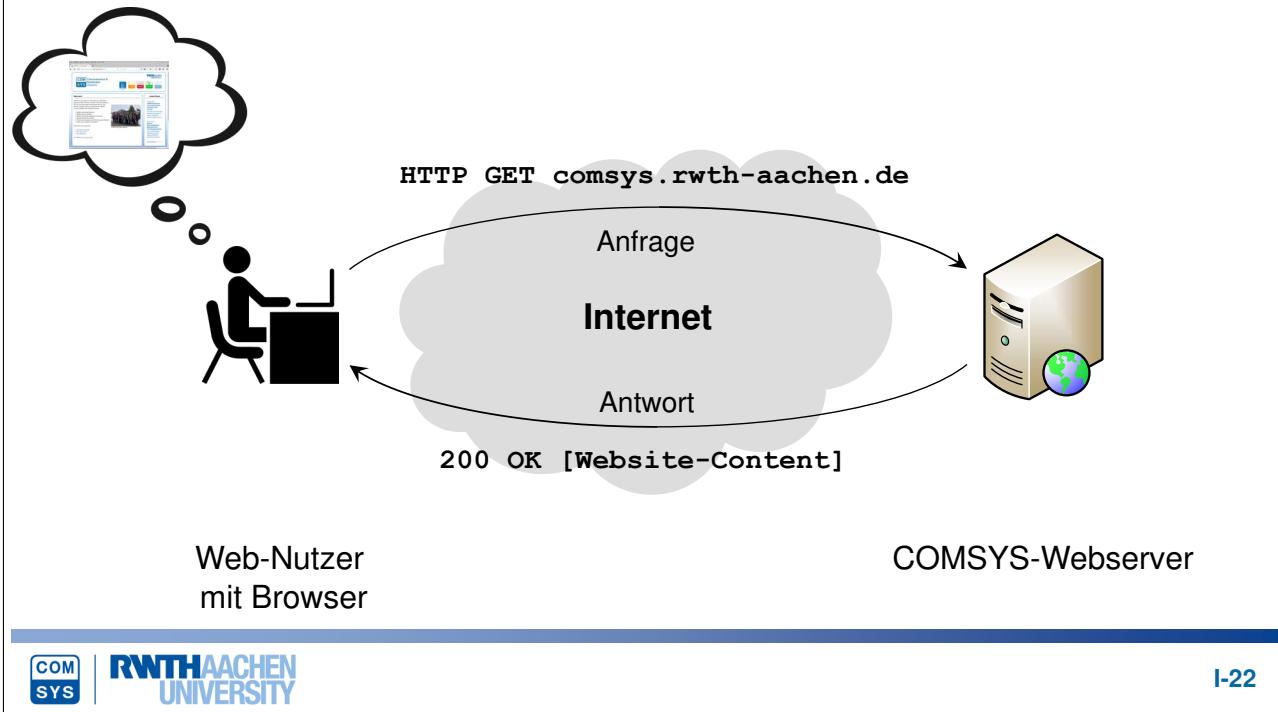
- **Einführung und Begriffe**

- ▶ Was ist Datenkommunikation?
- ▶ Information, Daten, Signale
- ▶ Netze

- **Allgemeine Grundlagen**

- ▶ Dienste
- ▶ Protokolle und Schichten
- ▶ Kommunikationsarchitekturen

Beispiel: Informationsabruf im WWW

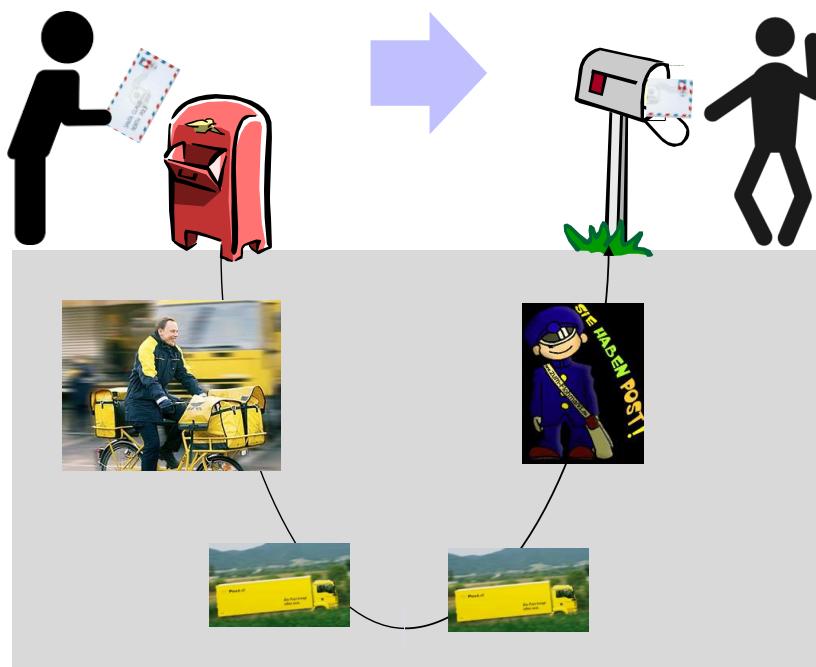


Generell stellen Kommunikationsnetze Dienste zur Verfügung – sie bieten den Dienst an, Daten zu übertragen. Alle, die die Datenübertragung in Anspruch nehmen, sind Dienstnutzer (auch: Dienstnehmer). Ein Beispiel ist der Zugriff auf die COMSYS-Webseite: der Benutzer (Browser) und der Webserver, der die Webseite hostet, sind Dienstnehmer.

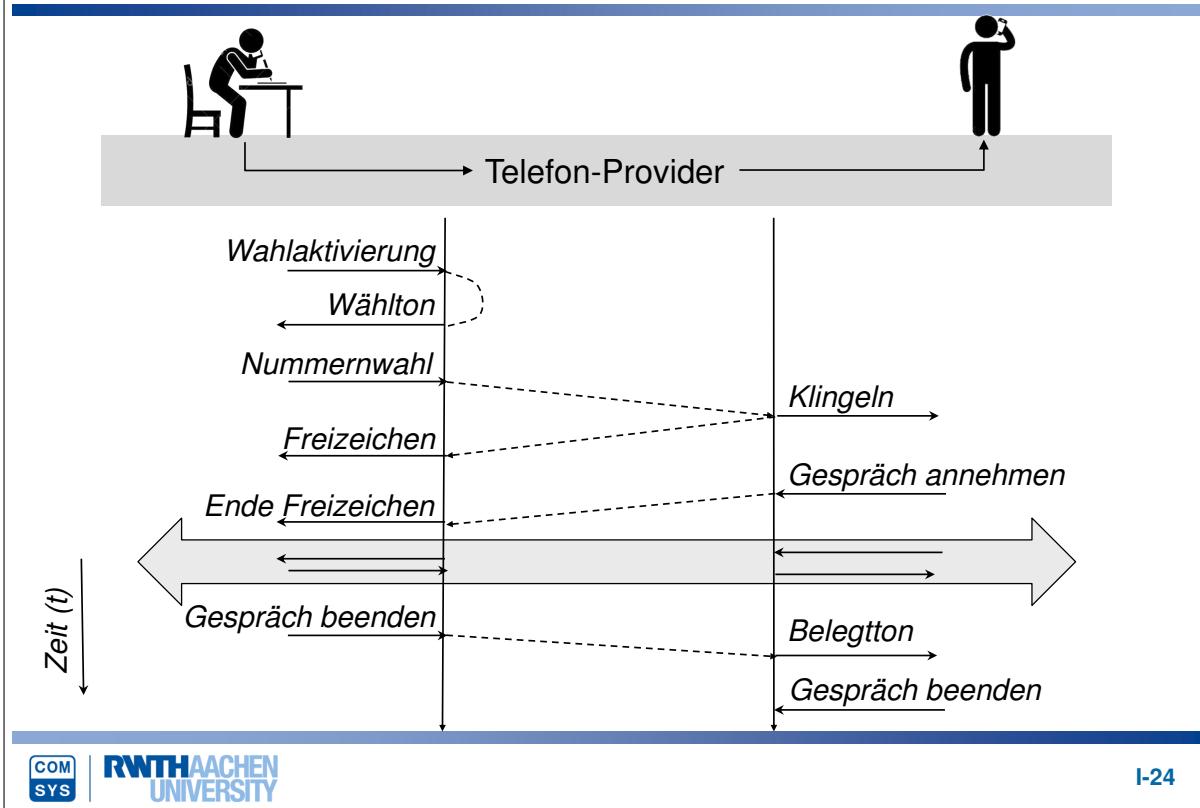
(Bitte nicht durcheinanderbringen: der Webserver selber bietet dem Web-Nutzer zwar auch einen Dienst an, aber dies ist in unserem Kontext nicht mit dem Begriff „Dienst“ gemeint. Wird im Bereich der Datenkommunikation von „Dienst“ gesprochen, ist stets ein Kommunikationsdienst gemeint, dem gegenüber auch der Webserver nur ein Dienstnehmer ist.)

Die Dienstnutzer „Web-Nutzer“ und „Webserver“ nehmen den Dienst eines Netzes (hier generell: Internet) in Anspruch, der die Anfrage vom Nutzer an den Server sowie die zugehörige Antwort zurück übermittelt. Der Dienst des „Internets“ sorgt dafür, dass Anfrage und Antwort jeweils am richtigen Ziel ankommen.

Beispiel: Post



Beispiel: Telefon



Gemeinsamkeiten der Beispiele

- Welche Ähnlichkeiten weisen diese Kommunikationssysteme auf?

- ▶ *Kommunikationsdienste* (Dienste eines Kommunikationssystems) zur Übertragung von Daten in verteilten Umgebungen
- ▶ Ablaufvorschriften / Mechanismen zur Koordination der Kommunikation (*Protokolle*)
- ▶ *Nachrichten* werden ausgetauscht (zur Koordination, zum Austausch der Inhalte)
- ▶ Kommunikation kann verschiedene Phasen aufweisen
- ▶ Interaktion der Nutzer mit System erfolgt nur lokal
- ▶ System kann hierarchisch strukturiert sein – mit verschiedenen Abstraktionsstufen – bleibt dem Nutzer in der Regel verborgen
- ▶ ...

Gemeinsamkeiten sind:

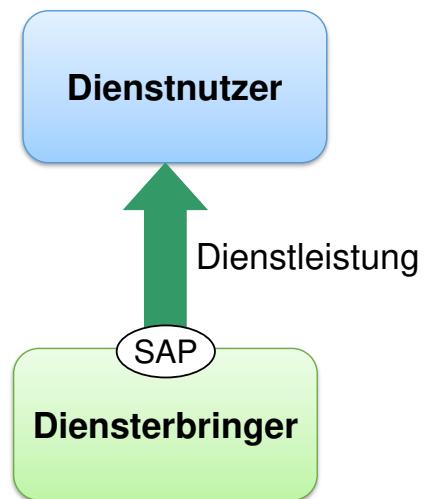
- Es werden Nachrichten ausgetauscht
- Wir haben in jedem Fall ein System dazwischen (= Kommunikationssystem, bezogen auf die Datenkommunikation)
- Das Kommunikationssystem kann aus mehreren, irgendwie strukturierten Komponenten bestehen

Letztendlich kann man sagen: zwei Kommunikationspartner nutzen die *Dienste* eines Kommunikationssystems, um eine Nachricht zu übertragen, wobei das Kommunikationssystem selbst noch weiter strukturiert sein kann.

Charakterisierung von Diensten

• Kommunikationsdienste

- ▶ **Dienstnutzer**: nimmt einen Dienst in Anspruch
- ▶ **Diensterbringer**: bietet einen Dienst an
- ▶ Ein Dienst wird im Rahmen einer **Dienstleistung** erbracht
 - Weist unterschiedliche Eigenschaften (z.B. bestätigt, unbestätigt) auf
 - Umfasst die Abwicklung eines Auftrags, welcher im Rahmen des Dienstes spezifiziert ist
- ▶ **Dienstzugangspunkt** (Service Access Point, SAP): Schnittstelle zur Dienstnutzung



Ein Kommunikationssystem soll eine Reihe allgemein brauchbarer, wohldefinierter und geregelter Funktionen anbieten. Eine derartige Funktionalität fassen wir unter dem Begriff des *Dienstes* zusammen. Allgemein sind Dienste Zusammenstellungen zusammengehöriger Funktionen, die wiederholt in ähnlicher Form benötigt werden und daher durch Instanzen bereitgestellt werden können, die sich auf diese Funktionen spezialisieren. Die Instanzen werden *Diensterbringer* (oder *Dienstgeber*) genannt.

Die Instanzen, die die Funktion nutzen, sind die *Dienstnutzer* (oder *Dienstnehmer*). Sowohl Sender als auch Empfänger von Daten sind Dienstnutzer. Eine Änderung der Implementierung des Diensterbringers ist möglich, ohne dass der Dienstnutzer es bemerkt.

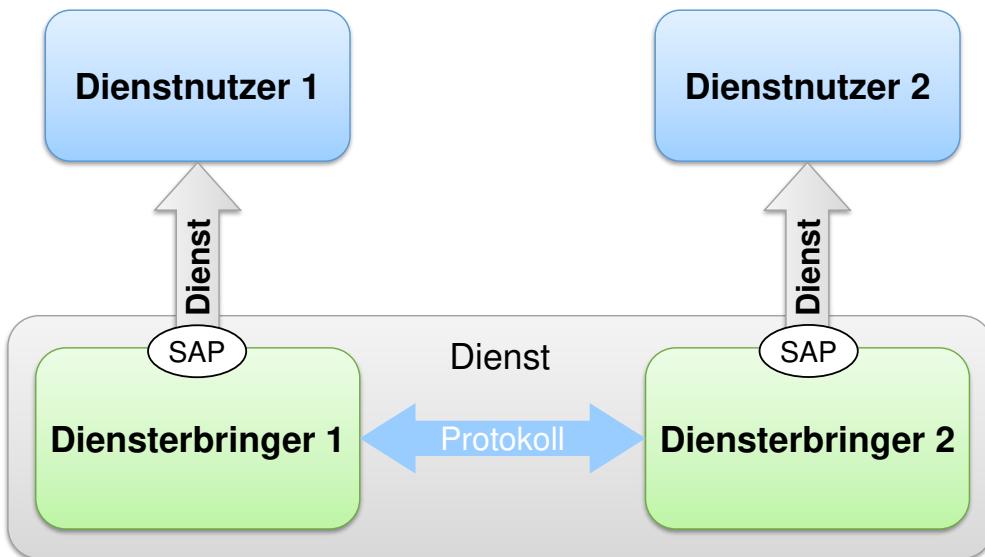
Beispiel: Kommunikationsdienst „Telefondienst“

- Kommunikationssystem: eine Sparte des Telefon-Providers
- Medium: verschiedene Kabel, Satellit, Funk
- Dienstnutzer: Kunde des Dienstes „Telefondienst“: Telefonierende Personen
- Dienstzugangspunkt: Telefon

Dienst und Protokoll – Übersicht

- **Grundlegendes Modell eines Kommunikationssystems**

- ▶ Basierend auf der Unterscheidung zwischen *Dienst* und *Protokoll*



Ein Dienst regelt zwar, welche Funktionalität ein Kommunikationssystem anbietet, aber die Details der Diensterbringung bleiben dem Dienstnutzer verborgen – er sieht nur festgelegte Schnittstellen zur Dienstnutzung. Allerdings muss auch genau definiert sein, wie der Dienst erbracht wird – auf jedem der beteiligten Rechner läuft lokal eine Instanz (z.B. als Modul des Betriebssystem-Kernels), die mit den Instanzen auf anderen Rechnern nach fest definierten Regeln interagieren muss, um global einen einheitlichen Kommunikationsdienst anbieten zu können.

Protokolle sind Verhaltenskonventionen, die die zeitliche Folge der Kommunikation zwischen den diensterbringenden Instanzen festlegen und das Format (Syntax und Semantik) der auszutauschenden Dateneinheiten bestimmen. Die Dateneinheiten werden Protocol Data Units (PDUs) genannt.

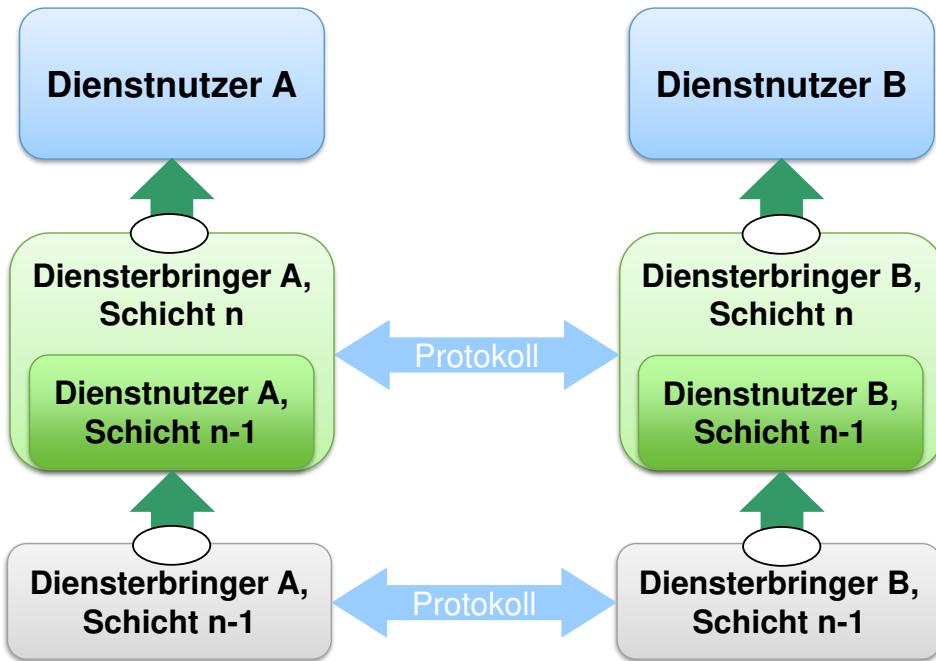
Kommunikationssysteme basieren daher auf

- Funktionalität = *Kommunikationsdienst* [engl. (Communication) Service]
- Realisierung der Funktionalität = Implementierung der Funktionalität als lokal ausführbare Instanzen der Diensterbringer. Auch „Protokollinstanz“ genannt.
- Regeln zur Erbringung der Funktionalität zwischen zwei räumlich verteilten Instanzen des Kommunikationssystems = *Kommunikationsprotokoll* [engl. (Communication) Protocol]

Dienstzugangspunkt: Eindeutig gekennzeichneter Zugangspunkt für die Inanspruchnahme eines Dienstes und die Wechselwirkungen/Interaktionen zwischen den Dienstnutzern. Diese eindeutige Kennzeichnung entspricht einer Adresse, unter der die jeweiligen Dienstnutzer erreicht werden können. Im Englischen: Service Access Point, SAP

Dienst und Protokoll – Übersicht

- Grundlegendes *Modell eines Kommunikationssystems*



Generell gibt es zwei Möglichkeiten, einen Diensterbringer zu implementieren.

1. Der Diensterbringer kann als ein monolithisches Stück Software entwickelt werden, welches alle Aufgaben des Kommunikationsdienstes erbringt. Das ist zwar effizient umzusetzen, bringt aber den üblichen Nachteil einer monolithischen Software mit sich: mangelnde Flexibilität (da die Funktionalität an einer einzelnen Anwendung ausgerichtet ist) oder Überladung (da alle möglichen Funktionalitäten umgesetzt werden – Kommunikationssysteme sind sehr komplexe Systeme).
2. Die in der Praxis verwendete Methode ist die Modularisierung – teile die gesamte Funktionalität in logische Bereiche auf, die unabhängig voneinander umgesetzt und (bestimmten Regeln folgend) beliebig kombiniert werden können. In der Praxis kommen Schichtenmodelle zum Einsatz: von sehr anwendungsnahen bis hin zu sehr hardwarenahen Funktionalitäten werden die Funktionalitäten in einige logische Schichten gruppiert. Jede Schicht stellt eine Teilfunktionalität bereit und agiert als Diensterbringer für die nächsthöhere Schicht. Diese nächsthöhere Schicht kann die Funktionalität der tieferen Schicht nutzen und agiert selbst als Erbringer weiterer Funktionalitäten für die wiederum nächsthöhere Schicht. Auf jeder Schicht können unterschiedliche Varianten implementiert werden und je nach Bedarf einer Anwendung kann die entsprechende Variante verwendet werden. Dadurch erzielt man eine höhere Flexibilität.

Betrachten wir nun die Partner einer Schicht:

- Sie bieten einen Dienst für eine höhere Schicht an
- Sie kennen nur den direkt unterliegenden Dienst
- Sie benutzen diesen unterliegenden Dienst (außer der untersten Schicht); generell kann man auch sagen, dass ein Medium genutzt wird, da die Partner einer Schicht den Dienst als „Medium“ sehen, welches die Daten zustellt. Ein tatsächliches Medium nutzt dabei nur die unterste Schicht, ansonsten hat man ein virtuelles Medium.
- Sie „unterhalten sich“ gemäß Regeln (= Protokollen) - z.B. „Telefon“-Schicht: wählen/klingeln/besetzt

Dienstprimitive

• Grundlegende Dienstprimitive nach OSI-Modell

► *Dienstfunktionen*

- CONNECT: Verbindungsaufbau
- DISCONNECT: Verbindungsabbau
- DATA: Datenaustausch
- ABORT: Abbruch durch Dienstnutzer oder Diensterbringer

► Typen bzw. Funktionen eines *Dienstprimitivs*:

- Request: Anfordern eines Dienstes
- Indication: Anzeige am Partner-Dienstzugangspunkt
- Response: Antwort des Partner-Dienstzugangspunkts
- Confirmation: Bestätigung der Diensterbringung am anfordernden Dienstzugangspunkt

Dienstprimitive stellen die – an einen zeitlichen Ablauf gebundene – abstrakte Beschreibung der Wechselwirkung an den Dienstzugangspunkten zur Nutzung eines Dienstes dar

Hier im Beispiel stelle ein Kommunikationsdienst vier Funktionen bereit:

- Connect: Herstellen einer Kommunikationsbeziehung zwischen den Kommunikationspartnern (z.B. Vereinbarung von Parametern)
- Disconnect: gegenseitige Einigung der Partner auf die Beendigung der Kommunikation
- Data: Austausch von Daten
- Abort: Beendigung der Kommunikationsbeziehung durch den Dienstanbieter oder einseitiger Abbruch durch einen der Dienstnutzer

Diese vier Dienstfunktionen werden durch die auf der Folie aufgeführten Primitive angeboten; im Kommunikationssystem erfolgt die Dienstnutzung durch Verwendung der Dienstprimitive.

Anmerkung: das sogenannte OSI-Modell wird später in diesem Kapitel noch behandelt. Hier soll dieses Modell (ohne genaueres Wissen, wie es aussieht) nur dazu dienen, eine konkrete Syntax für Dienstprimitive bereitzustellen.

Dienstprimitive – Beispiel

- Die Benennung eines Dienstprimitivs besteht aus folgenden Komponenten:

Name der Schicht/Anwendung	Dienstleistung	Dienstgrundtyp	Parameter
Physical (Ph)	Connect (Con)	Request (Req)	(entsprechend)
Data Link (DL)	Data (Dat)	Indication (Ind)	
Network (N)	Abort (Abo)	Response (Rsp)	
Transport (T)	Provider Abort (PAbo)	Confirmation (Cnf)	
HTTP	Disconnect (Dis)		
FTP	...		
...			

- Beispiel:

- ▶ T-Con.Req(Adressen) = Verbindungsaufbauanforderung an der Schnittstelle zum Transportdienst
- ▶ HTTP-Get[Dat.Req](URL) = Anforderung der Webseite, die durch URL identifiziert wird

Die Beschreibung der Dienstprimitive erfolgt hier gemäß dem Schema

<Schichtabkürzung> -<Dienstleistung>. <Diensttyp>

Das OSI-Modell definiert 7 Schichten. Hier kann zunächst verallgemeinernd davon ausgegangen werden, dass es sich dabei um Softwaremodule zur Datenübertragung handelt, die jeweils einen bestimmten Kommunikationsdienst erbringen. Der genauere Zweck der Schichten wird später noch erläutert. Jede der sieben OSI-Schichten hat eine Abkürzung, die sich an die englische Bezeichnung der Schicht anlehnt; so heißt z.B. die erste Schicht, die Bitübertragungsschicht, im Englischen Physical Layer mit der Abkürzung Ph.

Typische Dienstleistungen sind Aufbau, Datenübertragung, Rücksetzen, Abbruch. Diese fließen wie die Schichtbezeichnungen ebenfalls als Abkürzungen der entsprechenden englischen Begriffe in die Dienstprimitive ein, also z.B.

Verbindungsaubau : Connect = Con

Übertragung : Data = Dat

Der Diensttyp umfasst die bereits erwähnten vier Grundtypen von Ereignissen, die aus der Abwicklung der Dienstleistung hervorgehen

Anfrage : Request = Req

Anzeige : Indication = Ind

Antwort : Response = Rsp

Bestätigung : Confirmation = Cnf

In der Realität erfolgt die Umsetzung üblicherweise nicht sauber nach diesem Modell.

Dienste: Beispiele

- Beispiel: Post

- Dienstnutzer:

- Postkunden

- Diensterbringer:

- Post

- Zugangspunkte:

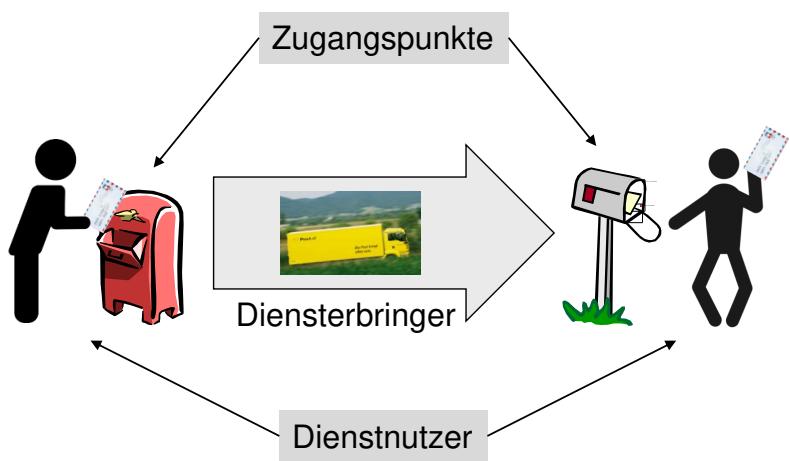
- öffentliche und private Briefkästen

- Dienstprimitive:

- Einwerfen eines adressierten und frankierten Briefes (DATA . Req),

- Postbote wirft Brief in Briefkasten des Empfängers (DATA . Ind)

- (in den USA: red flag als Indikator, dass Post ankam bzw. zum Abholen bereit ist)



Dienste: Beispiele

• Beispiel: Telefon

► Dienstnutzer:

Telefonkunden

► Dienstanbieter:

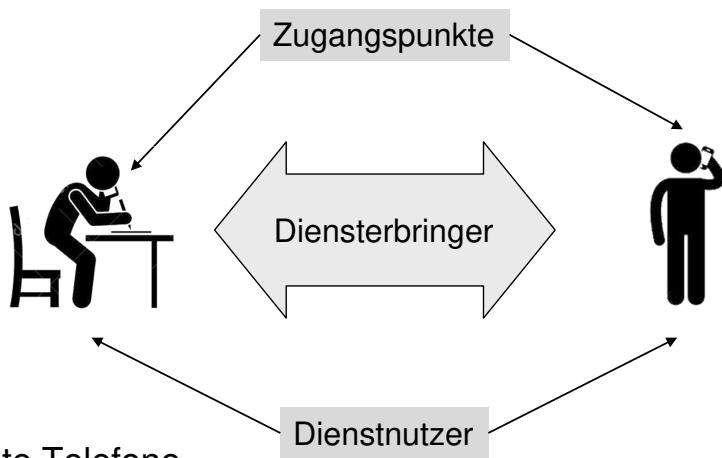
Telefon-Provider

► Zugangspunkte:

Öffentliche und private Telefone

► Dienstprimitive:

Nummer wählen (Con.Request), Klingeln (Con.Indication), ...



Dienstablauf

- **Abfolge von Dienstprimitiven ist nicht zufällig**
 - ▶ Abfolge muss spezifiziert werden können
- **Zeitablaufdiagramme (*Weg-Zeit-Diagramme*)**
 - ▶ Pro Dienstzugangspunkt eine Zeitachse (vertikal)
 - ▶ Dienstprimitive: Pfeile 
 - ▶ Weitere Zusammenhänge (neben Abfolge) durch Kommentierungen
 - Kausalitäten
 - Exakte Zeitbedingungen
 - Kommunikationsaktivitäten, z.B. Senden oder Verlust von Paketen
 - ▶ Zeitablaufdiagramme können nicht alle Zusammenhänge spezifizieren
- **Weitergehende Spezifikationstechniken**
 - ▶ Zustandstabellen und Zustandsübergangsdiagramme (Automaten), Spezifikationssprachen

Bislang wurden lediglich die statischen Beschreibungsaspekte, konkret die Dienstzugangspunkte und die Dienstprimitive, betrachtet. Interessant und komplex wird die Dienstbeschreibung, wenn zusätzlich noch dynamische Abläufe, d.h. das zeitliche Auftreten von Dienstprimitiven an den beteiligten SAPs, betrachtet werden.

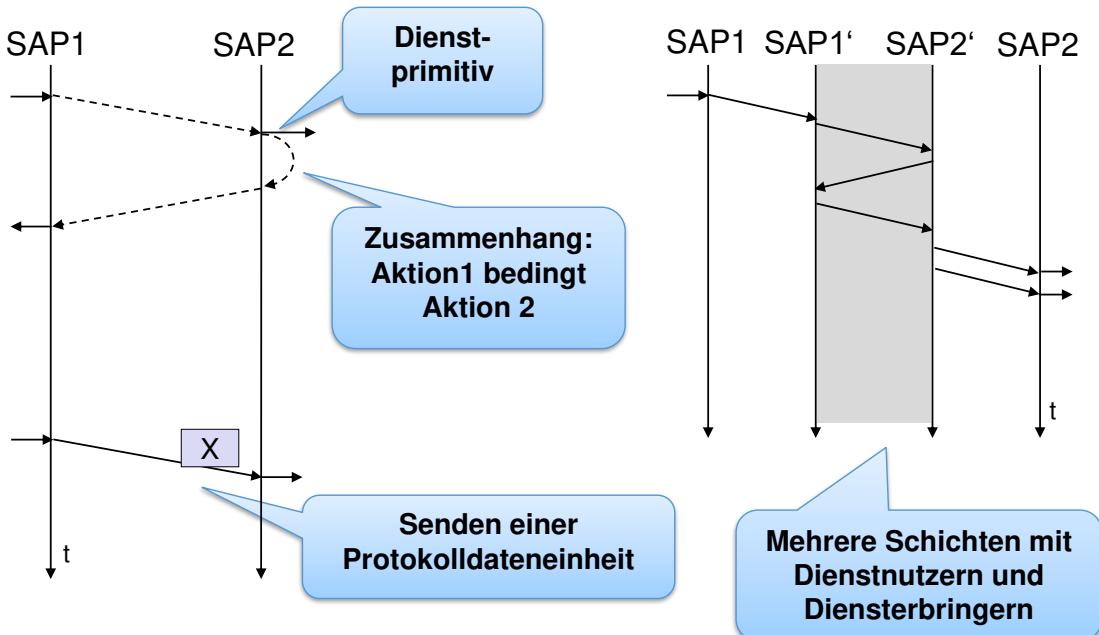
Eine einfache Möglichkeit zur Darstellung von Kommunikationsdiensten mit Betrachtung dynamischer Abläufe sind Weg-Zeit-Diagramme. Diese erlauben es, den Austausch von Dienstprimitiven zwischen zwei Dienstzugangspunkten einfach darzustellen. Solche Darstellungen können dann als Ausgangspunkt für die Implementierung der zugehörigen Protokolle verwendet werden.

Diese Diagramme werden hier knapp eingeführt und ab und zu zur Visualisierung verwendet; sie haben allerdings starke Nachteile, da die Visualisierung komplex wird, sobald ein dritter Dienstzugangspunkt verwendet wird; und auch einfache Verzweigungsmöglichkeiten im Dienstablauf (if-then-else) lassen sich nicht gut darstellen.

Zur tatsächlichen Protokollentwicklung bieten sich stattdessen mächtigere Spezifikationssprachen oder Automaten an – auch wenn die Protokollentwicklung oft auf eine vollständige formale Spezifikation verzichtet.

Dienstablauf

- Beschreibung durch Weg-Zeit-Diagramme



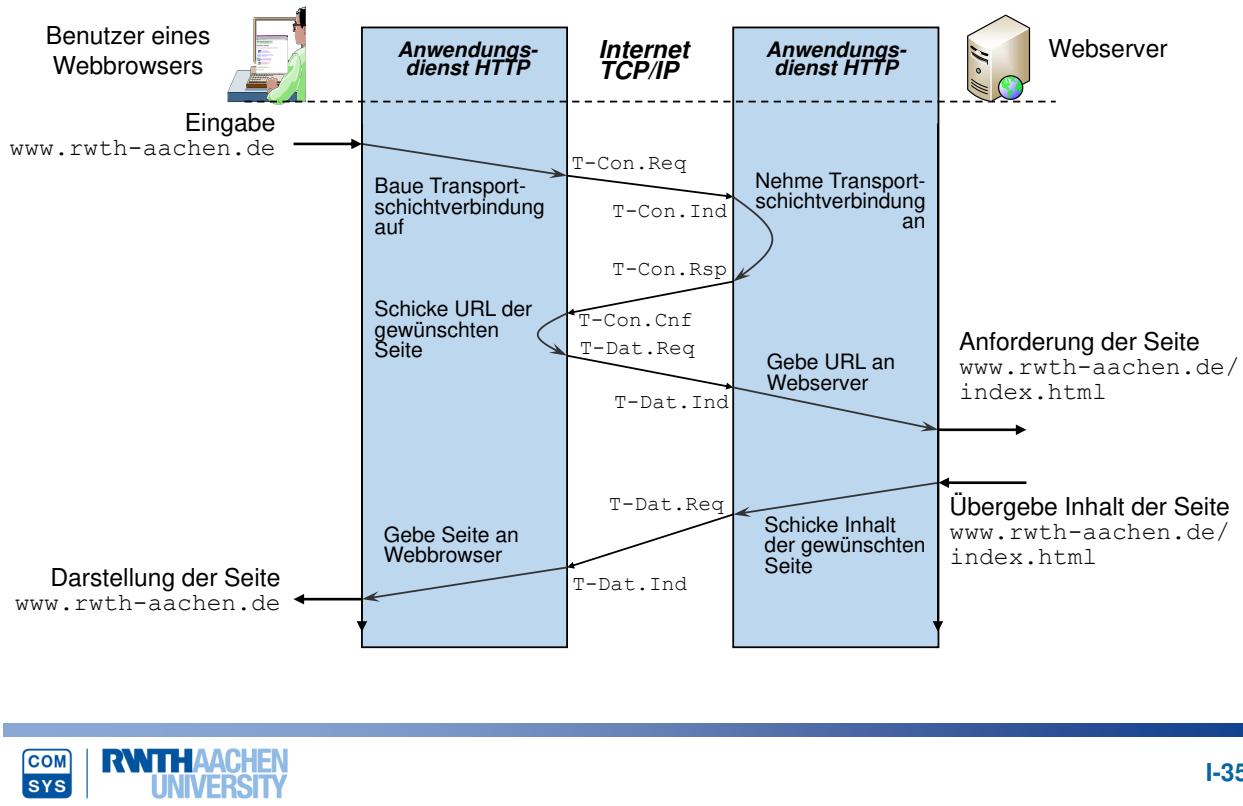
Horizontal werden die beteiligten SAPs angegeben, die vertikalen Achsen sind Zeitachsen, an denen die auftretenden Dienstprimitive aufgeführt sind. Die Ursache-Wirkung-Beziehungen zwischen den an getrennten SAPs auftretenden Dienstprimitive erfolgt durch Pfeile von einer vertikalen Zeitachse zu einer anderen.

Unterschied der Notation bei den Pfeilen: die gestrichelten Pfeile stellen eine Interaktion der Kommunikationsteilnehmer (mittels Dienstprimitive) dar, die nicht näher spezifiziert ist. Werden stattdessen (wie unten links dargestellt) durchgezogene Pfeile verwendet, ist uns die Implementierung bekannt (d.h.: wir wissen, welches Protokoll zur Interaktion eingesetzt wird und welche konkreten Nachrichten ausgetauscht werden).

Hinter den gestrichelten Pfeilen könnten sich auch noch komplexere Interaktionen verbergen. Z.B. könnte bei unserer Dienstimplementierung (verborgen vor dem Dienstnutzer) noch die Nutzung eines weiteren Dienstes nötig werden (siehe rechts auf der Folie) – unser Diensterbringer wäre hier nicht monolithisch implementiert, sondern nutzt selbst einen weiteren Diensterbringer über SAP1' zur Erbringung seines Dienstes. Gestrichelte Pfeile sollen also andeuten, dass uns die real notwendigen Zwischeninteraktionen nicht bekannt sind.

Zu jedem Dienstprimitive können außerdem Kontrollparameter angegeben werden. Dies sind z.B. Adressen von Sender und Empfänger oder Anforderungen an die Dienstqualität (beispielsweise: schnelle Übertragung notwendig, garantierte maximale Zeit bis zur Auslieferung der Daten).

Beispiel: Informationsabruf im Internet



Die betrachteten Partner-Protokollinstanzen sind in diesem Beispiel die HTTP-Instanzen im Webserver und im Webbrowser (Client). HTTP ist ein sogenanntes Anwendungsprotokoll (HyperText Transfer Protocol), das zum Austausch von Dokumenten im WWW entwickelt wurde. Die HTTP-Instanzen bieten kommunizierenden Anwendungen eine Dienstschnittstelle an, stellen also eine Implementierungsmöglichkeit der obersten Schicht eines Kommunikationssystems dar. Sie nutzen Dienste der sogenannten Transportschicht, um Anfrage und Antwort zuverlässig zwischen räumlich verteilten Rechnern zu übertragen.

Der Benutzer des Webrowsers gibt eine URL ein. Für den Web-Browser nicht sichtbar wird eine Verbindung zum Webserver aufgebaut, über die die Daten zuverlässig übertragen werden können. Der Verbindungsaufbau ist bestätigt, da die anfragende Instanz wissen muss, ob die Gegenseite die Verbindungsanfrage annimmt und wann sie zum Empfang von Daten bereit ist.

Die Anforderung der gewünschten Webseite hingegen erfolgt unbestätigt. Für die anfragende Instanz ist es nicht wichtig zu wissen, ob und wann die Anfrage auf der Gegenseite angekommen ist, da die unterliegende Schicht die Zustellung garantiert. Ebenso erfolgt die Übertragung der Antwort unbestätigt.

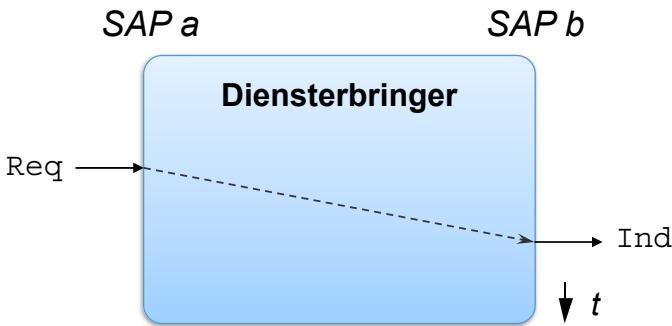
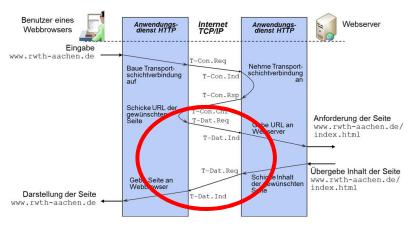
Wann bestätigte und wann unbestätigte Dienste eingesetzt werden, hängt also zum einen von den Anforderungen der Kommunikation selbst ab, zum anderen aber auch von den unterliegenden Diensten.

Dieses Beispiel beinhaltet mehrere Begriffe, deren Verständnis wichtig ist:

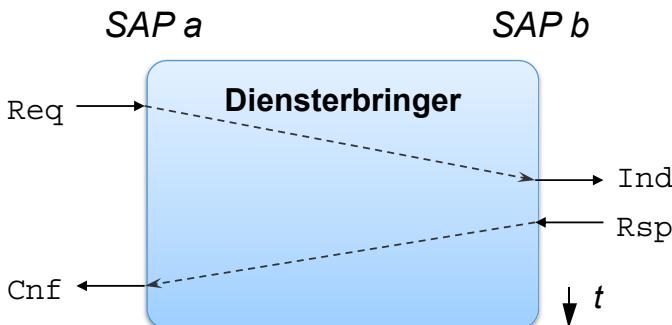
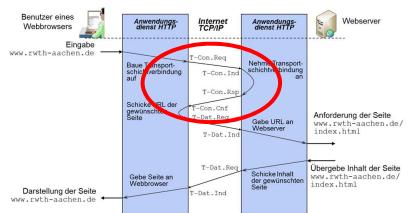
- Verbindungsorientierte Kommunikation (Verbindungsaufbau)
- Bestätigte/unbestätigte Kommunikation.

Bestätigter/unbestätigter Dienst

- Unbestätigter Dienst



- Bestätigter Dienst



Bei Nutzung eines unbestätigten Dienstes übergibt man dem Kommunikationsdienst an seinem SAP die Daten und diese werden – eventuell unter Zuhilfenahme weiterer Dienste – an den SAP des Empfängers ausgeliefert. Der Sender bekommt keine Rückmeldung, ob die Daten tatsächlich ausgeliefert bzw. angenommen wurden. Ob die Daten zuverlässig ausgeliefert werden oder nicht, hängt somit von der konkreten Dienstimplementierung ab (d.h. vom verwendeten Protokoll und den genutzten Diensten tieferer Schichten).

Beispiel: Versand einer E-Mail

Bei einem bestätigten Dienst hingegen bekommt der sendende Dienstnutzer eine Rückmeldung über die erfolgreiche Zustellung.

Beispiel: Anzeige, wann eine Nachricht über WhatsApp zugestellt wurde.

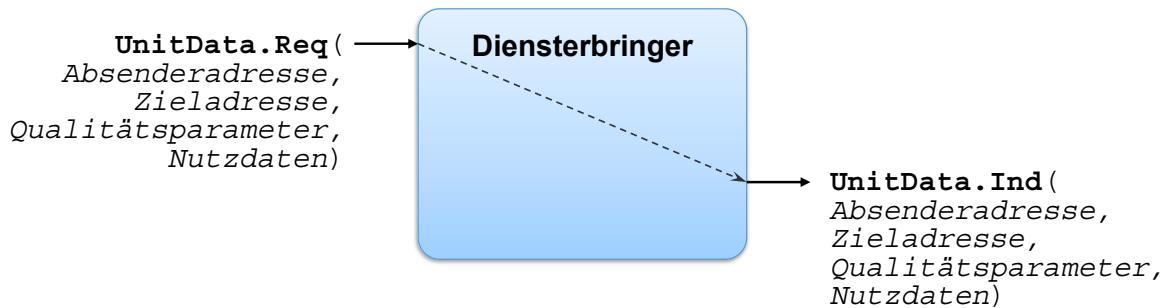
Bitte beachten: ob ein Dienst bestätigt oder unbestätigt ist, sagt noch nichts über die Zuverlässigkeit einer Datenübertragung aus – bei einer Übertragung im Internet kann es z.B. durch Netzüberlastung zu einem Verlust einzelner übertragener Dateneinheiten kommen, egal, ob der Dienst bestätigt ist.

Verwendet man einen bestätigten Dienst, erkennt man solche Verluste auf jeden Fall daran, dass keine Bestätigung erfolgt; der Dienstnutzer kann die Anfrage noch einmal stellen. Bei einem unbestätigten Dienst gibt es diese Möglichkeit nicht. Allerdings kann der Dienst intern (also im Protokoll) mit Bestätigungen arbeiten (die dem Dienstnutzer verborgen bleiben), so dass der Dienstnutzer auch bei Verzicht auf Bestätigungen davon ausgehen kann, dass die Daten zuverlässig übertragen werden.

Dienst und Verbindung

• Verbindungsloser Dienst

- ▶ Dienstleistung ohne Kontext
 - Jeder Datenaustausch wird isoliert betrachtet, ohne jegliche Berücksichtigung vorhergegangener Kommunikationsvorgänge
- ▶ Kontrollparameter als Teil der übergebenen Dateneinheit
- ▶ Keine feste Verbindung zwischen den Kommunikationspartnern



Eine weitere Unterscheidung von Diensten kann anhand des Begriffs der „Verbindung“ erfolgen: Dienste können verbindungslos oder verbindungsorientiert sein.

Verbindungslose Dienste sind die einfache Variante und werden daher zuerst betrachtet.

Das einzige Dienstprimitiv, das angeboten wird, ist UnitData, also ein „einzelnes Datum“. In diesem Dienstprimitiv müssen alle für die Übertragung relevanten Kontrollinformationen enthalten sein. Das sind zum Beispiel sämtliche Adressierungsinformation (Absender- und Zieladresse) und die Qualitätsparameter, in denen der Sender seine Wünsche bzgl. der Datenübertragung formuliert.

Verbindungslose Dienste sind normalerweise auch unbestätigt.

• Verbindungsorientierter Dienst

- ▶ Dienstleistung im Kontext, d.h. Diensterbringung abhängig von früher erbrachten Dienstleistungen
 - Vor dem Datenaustausch zwischen Dienstnutzern: *Verbindungsauftbau* durch die beteiligten Instanzen des Dienstanbieters
 - *Protokollabhängige Aushandlung von Übertragungsparametern*, z.B. Kommunikationspartner, Dienstqualität, Übertragungsweg
 - Datenaustausch innerhalb der Verbindung erfolgt unter Berücksichtigung des aktuellen Verbindungszustands
- ▶ Weniger Kontrolldaten notwendig, aber
 - Zeitaufwand für Herstellung des Kontextes (Verbindungsauftbau)
 - Aufwand für die Verwaltung des Kontextes

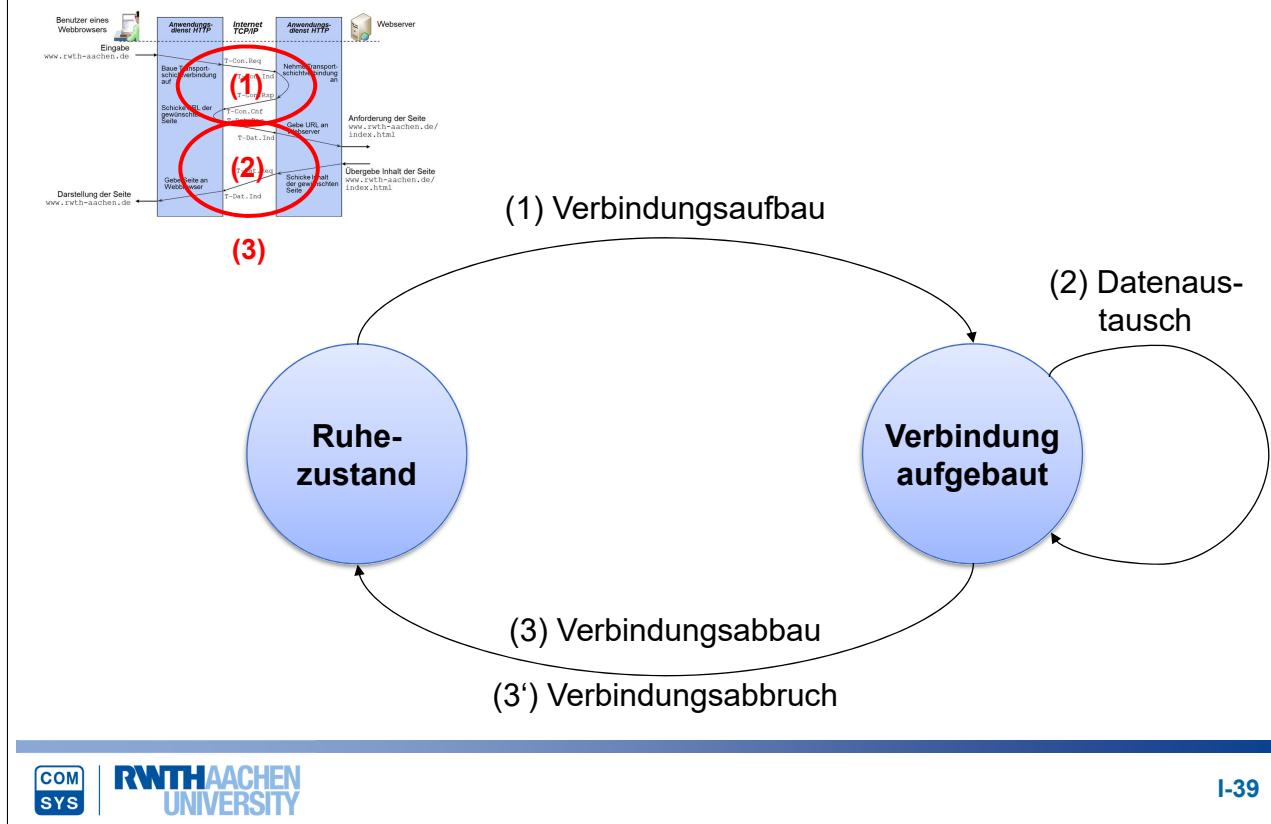
Der Kontext bei verbindungsorientierten Diensten entsteht durch früher erbrachte Dienstleistungen. Vor dem Datenaustausch muss bei einem verbindungsorientierten Dienst zuallererst eine Verbindung aufgebaut (d.h.: ein Kontext generiert) werden. Dies geschieht durch einen sogenannten Verbindungsauftbau, bei dem die Dienstnutzer einen Kontext aushandeln, d.h. Kontrollparameter definieren, die auf alle im Rahmen der Verbindung ausgetauschten Dienstprimitive angewendet werden.

Der Zustand einer Verbindung zum Zeitpunkt t setzt sich aus den Parametern zusammen, zusätzlich jedoch auch aus (theoretisch) allen Kommunikationsvorgängen, die von Zeitpunkt 0 (Verbindungsauftbau) bis Zeitpunkt t innerhalb dieser Verbindung stattgefunden haben. Wie lange dieses „Gedächtnis“ zurückreicht und wie viele/welche Daten hierbei gespeichert werden (wenn überhaupt), ist jedoch protokollspezifisch. Als Beispiel sei der (später im Detail besprochene) Protokollmechanismus der „Sequenznummern“ erwähnt, der jedem Datenpaket eine spezifische, ganzzahlige Nummer zuordnet. Zusammen mit entsprechenden Bestätigungen von Empfängerseite ermöglicht dies eine Erkennung von Paketverlusten - hierbei speichert der Sender Datenpakete nur solange zwischen, bis eine ordnungsgemäße Bestätigung eintrifft.

Durch Aufbau und Verwaltung eines Kontextes scheint alles komplexer und aufwändiger zu werden. Man hat aber den großen Vorteil, dass man sich die explizite Übertragung einiger Kontrollparameter in jedem Dienstprimitiv sparen kann, weil diese bereits im Kontext beschrieben sind. Zum Beispiel: Beim verbindungsorientierten Dienst kann beim Verbindungsauftbau der Weg durch ein komplexes Netzwerk zwischen den beteiligten Kommunikationspartnern ermittelt und als Kontext aufbewahrt werden. Bei einem verbindungslosen Dienst muss die Wegwahl bei jedem einzelnen Dienstauftruf erfolgen. Ein wesentlicher Vorteil verbindungsorientierter Dienste ist somit, die Tatsache, dass nach einem Verbindungsauftbau nicht mehr jedes Datenpaket einer Verbindung sämtliche (bereits ausgehandelte) Parameter beinhalten muss (z.B. Adresse, Dienstqualität etc.), was eine Reduzierung des Anteils der übertragenen Kontrollinformationen und somit letztendlich auch der Netzlast selbst bedeutet.

Beispiele von vorher: die Versendung von Briefen ist verbindungslos – jeder Brief muss alle Kontrollinformationen enthalten. Ein Telefongespräch ist verbindungsorientiert – das Wählen einer Nummer startet einen Verbindungsauflauf, alle Sprachdaten folgen einem festgelegten Weg durch das Telefonnetz.

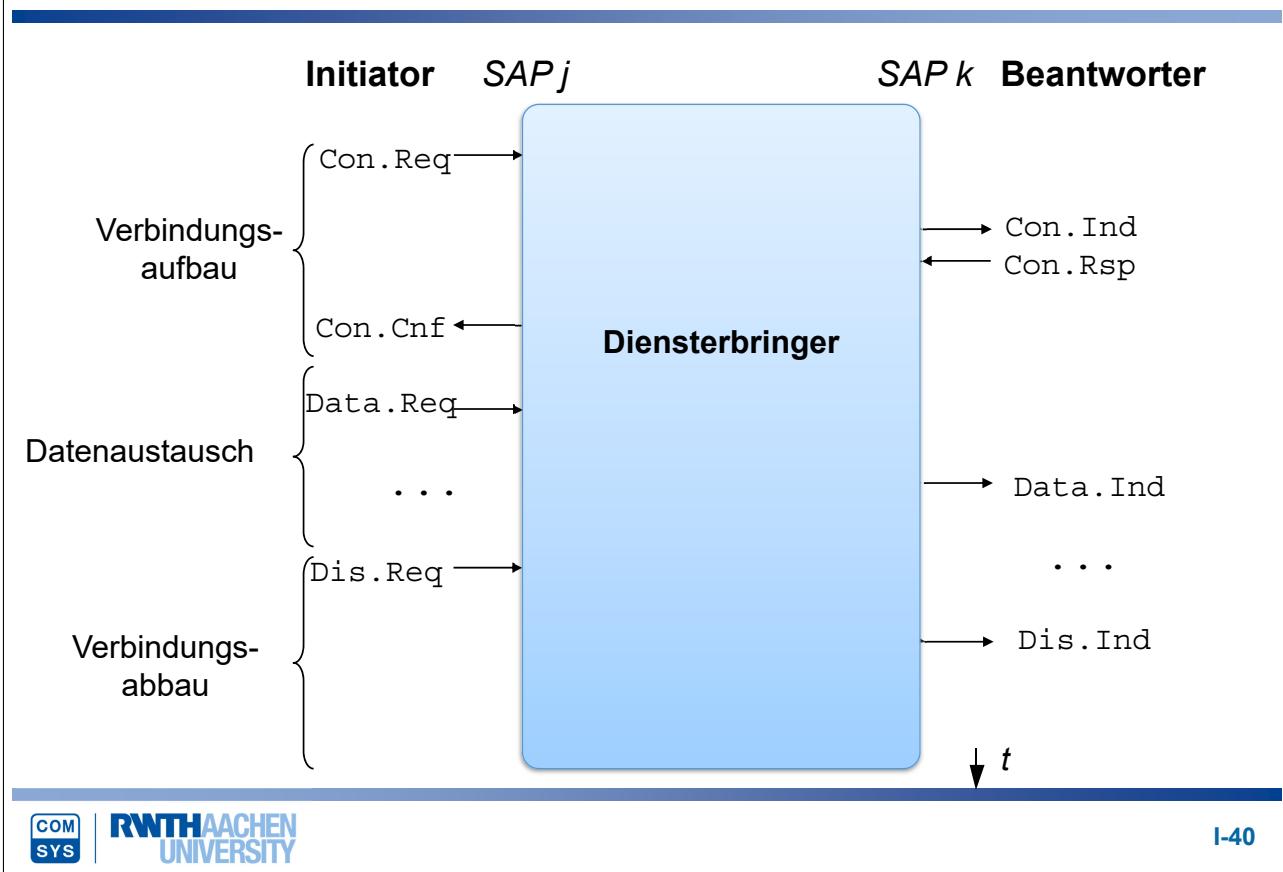
Verbindungsorientierte Dienste: Phasen



Die Klasse der verbindungsorientierten Dienste und das im Bereich der Datenkommunikation zentrale Konzept der Verbindung werden im Folgenden vertieft.

Ein verbindungsorientierter Dienst kann übersichtlich mithilfe eines endlichen Automaten in die vier Betriebsphasen „Verbindungsaufbau“, „Datenaustausch“, „Verbindungsabbau“ und „Verbindungsabbruch“ aufgeteilt werden.

Verbindungsorientierte Dienste im Weg/Zeit-Diagramm



(ausgeblendete Folie – wurde in der Vorlesung nicht behandelt)

Ein Verbindungsauftbau/-abbau hat im Weg/Zeit-Diagramm folgendes Aussehen:

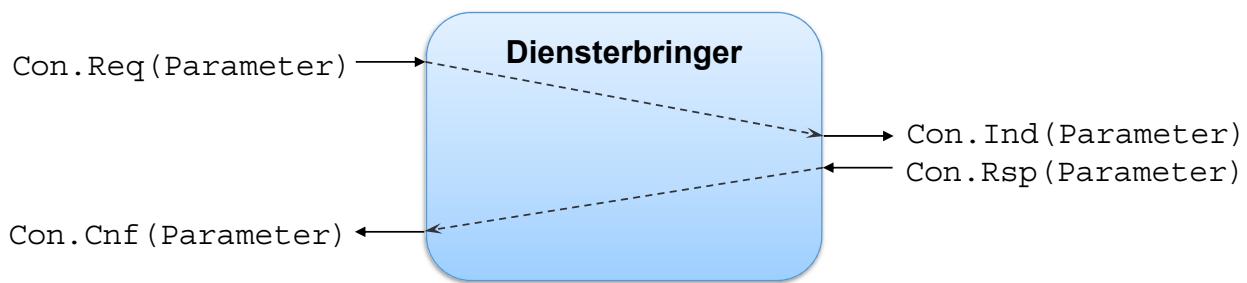
Phase 1: Verbindungsauftbau durch `Con.Req/Ind/Rsp/Cnf`

Phase 2: Datenaustausch durch `Data.Req/Ind`

Phase 3: Verbindungsabbau durch `Dis.Req/Ind` (unbestätigter Verbindungsabbau)

Üblicherweise ist der Verbindungsauftbau bestätigt, damit ein Dienstnutzer weiß, ab wann die Datenübertragung möglich ist. Datenaustausch und Verbindungsabbau können wie hier unbestätigt, oder auch bestätigt implementiert werden, abhängig von den unterliegenden Diensten.

Phase 1: Verbindungsauftbau



- **Parameter z.B.**

- ▶ Initiatoradresse
- ▶ Responderadresse
- ▶ Qualitätsparameter
- ▶ Wenige Nutzdaten

Im Folgenden werden die drei Phasen eines verbindungsorientierten Dienstes genauer behandelt, beginnend mit der Verbindungsauftaphase. Zunächst sollen die beim Verbindungsauftbau beteiligten Parameter näher betrachtet werden.

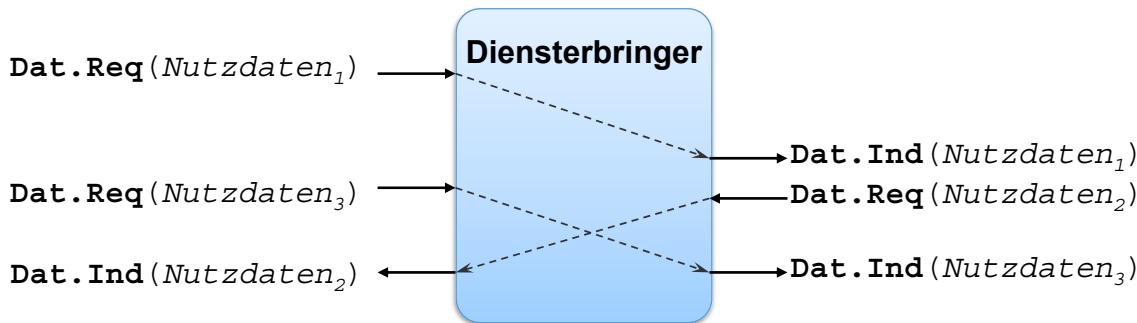
Folgende Parameter werden in den Dienstprimitiven zum Verbindungsauftbau genutzt:

- Initiator-/Responderadresse: SAP-Adressen
- Qualitätsparameter: weitere Kontrollparameter. Denkbare Qualitätszusicherungen eines Datenaustauschs sind verlustfreie Übertragung, reihenfolgetreue Auslieferung, hochpriorer Datenaustausch, ...
- Kurze Nutzdaten: Dienstnutzer kann während des Verbindungsauftbaus zusätzliche Daten austauschen. Die zusätzlichen Daten stehen i.d.R. in direktem Zusammenhang mit der aufzubauenden Verbindung. Sie können z.B. für die Angabe genutzt werden, für welche überliegenden Protokolle die Verbindung dienen soll (z.B. Transportverbindung zum Zweck des Dateitransfers). Der Responder hingegen kann die kurzen Nutzdaten dazu verwenden, einen Statuscode z.B. für den Grund der Ablehnung einer Verbindung anzugeben.

Ein interessanter Aspekt des Verbindungsauftbaus betrifft die Aushandlung der Qualitätsparameter. Der Initiator gibt gewisse Parameter, die er sich wünscht, vor. Diese werden vom Dienst bzw. vom Responder geprüft und können heruntergestuft werden.

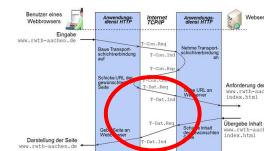
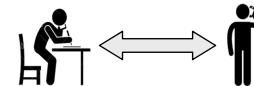
Konkretes Beispiel: Der Initiator wünscht sich einen Datendurchsatz von 200 MBit/s, der Dienst kann ihm aufgrund des zugrundeliegenden Mediums nur 100 MBit/s anbieten und der Responder reduziert letztendlich auf 50 MBit/s.

Phase 2: Unbestätigter Datenaustausch



• „Ungeregelter“ Datenaustausch

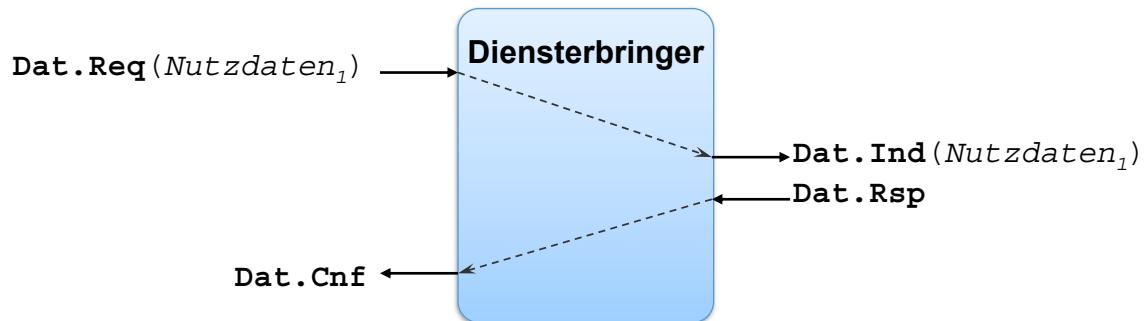
- ▶ Beide Seiten gleichberechtigt, können beide jederzeit senden
- ▶ Keine Rückmeldung an sendenden Dienstnutzer, ob die Nutzdaten zugestellt wurden



Die zweite Phase ist der Datenaustausch, also die eigentliche „Produktiv-Phase“, in der die Nutzdaten über die aufgebaute Verbindung ausgetauscht werden. Im Gegensatz zum bestätigten Datenaustausch besitzt der unbestätigte Datenaustausch dabei den Vorteil, dass Sender und Empfänger nur lose synchronisiert sind; das bedeutet, dass sich Sender und Empfänger nur wenig abstimmen müssen und sich entsprechend wenig im Fortschreiten ihrer Arbeit behindern.

Wie die Folie zeigt, beinhalten die beim unbestätigten Datentransfer verwendeten Dienstprimitive Dat.Req und Dat.Ind nur einen Parameter (die unbestätigt zu übertragenden Nutzdaten). Alle weiteren Parameter wurden bereits beim Verbindungsaufbau ausgetauscht/ausgetauscht. (Zumindest in diesem Beispiel – es können durchaus noch mehr Parameter vorhanden sein; der Phantasie der Protokollentwickler sind keine Grenzen gesetzt.)

Phase 2: Bestätigter Datenaustausch



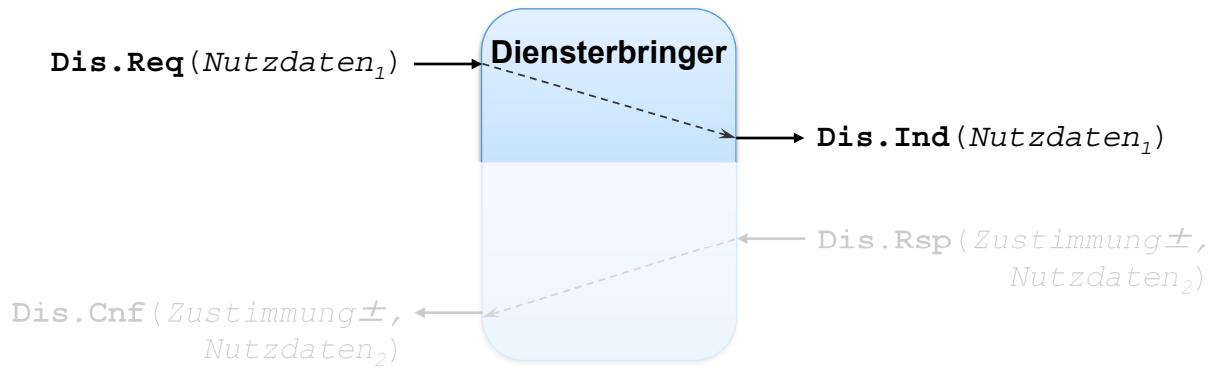
- „Geregelter“ Datenaustausch



- ▶ Sender Dienstnutzer erhält Rückmeldung
z.B. über erfolgreiche Zustellung oder
Ablehnung der Daten

Die Dienstprimitive **Dat.Rsp** und **Dat.Cnf** der bestätigten Variante beinhalten in diesem Beispiel gar keine Parameter mehr und sind z.B. nur eine Zustellungsbenachrichtigung

Phase 3: Verbindungsabbau



- **Geregelte Beendigung der Verbindung**

- ▶ Bestätigt oder unbestätigt
- ▶ Zur Freigabe von für die Kommunikation reservierten Ressourcen
- ▶ z.B. bei der Transportschichtverbindung im HTTP-Beispiel

Der Verbindungsabbau wird durch einen der Dienstnehmer durch die Inanspruchnahme des entsprechenden Dienstprimitivs „Disconnect“ (oder „Release“) eingeleitet.

Der wesentliche Unterschied zwischen dem bestätigten und dem unbestätigten Verbindungsabbau besteht darin, dass beim bestätigten Verbindungsabbau der zweite Teilnehmer dem Verbindungsabbau widersprechen kann. Diese Möglichkeit ist durch den Parameter „Zustimmung+/-“ in Rel.Rsp/Cnf gegeben. In den Nutzdaten könnte der Dienstnutzer dann angeben, warum er es ablehnt, die Verbindung abzubauen.

Phase 3': Verbindungsabbruch

- **Abbruch durch Diensterbringer (Provider Abort, PAbO)**

- ▶ z.B. Abbruch der LTE-Verbindung aufgrund überlasteter Zelle



- **Nutzerabbruch (User Abort, UAbO)**

- ▶ z.B. abrupte Beendigung einer TLS-Verbindung, wenn eine korrupte Dateneinheit empfangen wird



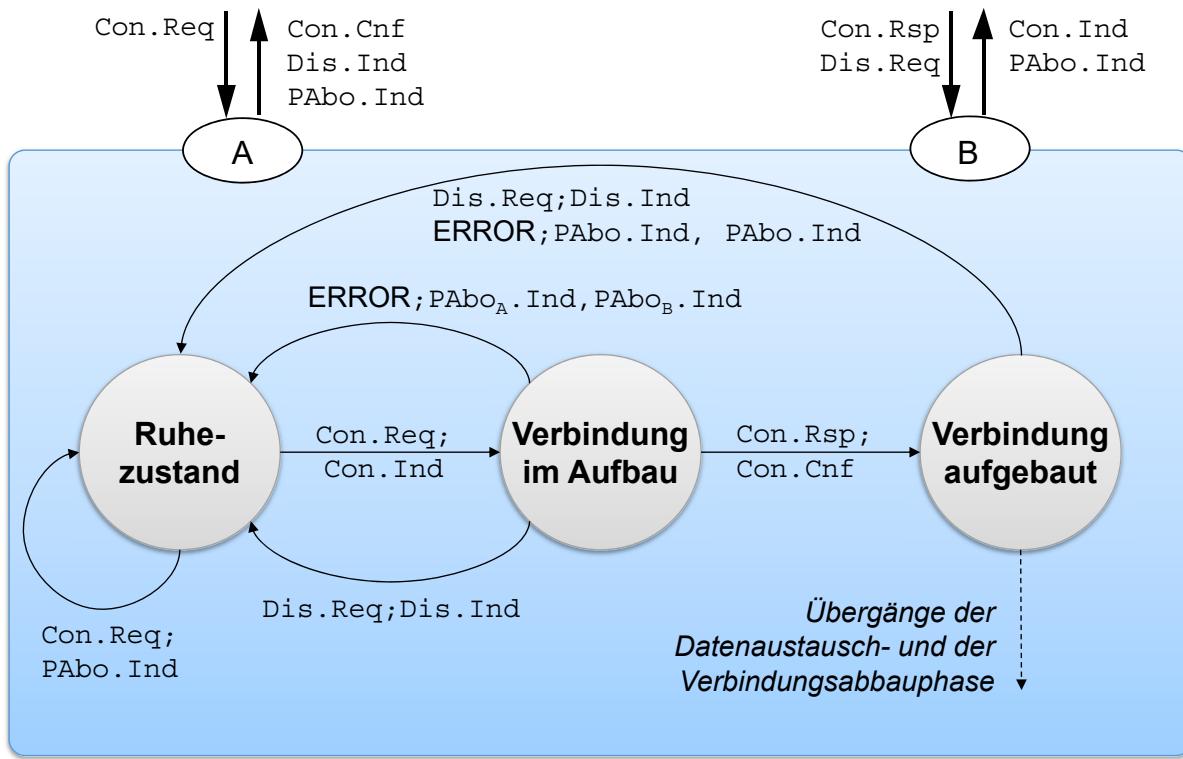
Abschließend soll auch die vierte und letzte Phase des verbindungsorientierten Dienstes behandelt werden, der Verbindungsabbruch. Wie der Begriff „Abbruch“ schon vermuten lässt, handelt es hier um einen ungewünschten Abbau der Verbindung aufgrund einer Ausnahmesituation. Diese Phase kommt also im regulären Betrieb nicht vor.

Die Ausnahmesituation kann in den unterliegenden Schichten aufgetreten sein, wofür das Dienstprimitiv PAbO zur Verfügung steht:

- Provider Abort (PAbO): werden durch den Dienst selbst oder die Diensterbringer tieferer Schichten verursacht
- User Abort (UAbO): Instanzen der über den Dienst kommunizierenden Dienstnutzer oder höhere Schichten

Der Dienst UAbO wird von der Instanz der betrachteten Schicht genutzt, wenn die Ausnahmesituation gerade in dieser Schicht auftritt und statt eines geordneten Verbindungsabbaus ein erzwungener Verbindungsabbruch notwendig ist.

Zustandsübergangsdiagramm: Verbindungsaufbau



Ein Weg-Zeit-Diagramm beschreibt das zeitliche Verhalten an den beteiligten Dienstzugangspunkten. Dieser von außen beobachtbare Ablauf ist aufgrund der Ursache-Wirkung-Beziehung von Dienstprimitiven gegeben, die sich geeignet durch sogenannte *Zustandsübergangsdiagramme* spezifizieren lässt (die Arbeitsweise eines solchen Diagramms entspricht im wesentlichen dem eines endlichen Automaten). Wie die Abbildung andeutet, wird damit das Verhalten des unterliegenden Mediums (Dienstes) beschrieben. Dieses befindet sich in einem gewissen Zustand – im Beispiel ist zwischen drei unterschiedlichen Zuständen zu unterscheiden. Gewisse eingehende Dienstprimitive werden in jedem Zustand akzeptiert und führen zu Zustandsübergängen: im Zustand „Verbindung in Aufbau“ existieren zu den Primitiven Con.Rsp und Dis.Req Übergänge; außerdem kann hier auch noch ein spontaner Übergang durch einen Abbruch stattfinden. Die Zustandsübergänge werden also durch das Dienstprimitiv ausgelöst, das vor dem „;“ angegeben ist. Nach dem Semikolon sind ein oder mehrere Dienstprimitive angegeben, die an den entsprechenden SAPs auszugeben sind. Die Beschriftung kann also als „Ereignis ; ausgelöste Aktion“ verstanden werden.

- Syntax: <auslösendes Dienstprimitiv ; resultierendes Dienstprimitiv>, z.B. Con.Req;Con.Ind

- Falls die erste Angabe fehlt, handelt es sich um einen spontanen Übergang.

Ein Weg-Zeit-Diagramm beschreibt einen konkreten Weg durch das Zustandsübergangsdiagramm, durch welches das Verhalten des Mediums dargestellt ist. Das Zustandsübergangsdiagramm verfügt also über größere Ausdrucksstärke.

Der gewünschte Normalweg führt vom Ruhezustand durch Con.Req zum „Verbindung in Aufbau“ und durch Con.Rsp zu „Verbindung aufgebaut“. Von hier geht es dann in die diversen anderen Dienstphasen des Datenaustauschs und des Verbindungsabbaus.

Zu bemerken ist noch, dass in obigem Beispiel die Initiative zum Verbindungsaufbau

immer vom Dienstnehmer am SAP A ausgehen muss, während der Dienstnehmer am SAP B eine Verbindung nur akzeptieren, ablehnen oder auch abbauen kann. Im Normalfall werden jedoch beiden Dienstnehmern sämtliche Dienstprimitive zur Verfügung stehen (d.h. alle Aktionen können sowohl von der einen wie auch von der anderen Seite ausgelöst werden). SAP A und B beschreiben also keine konkreten Geräte, sondern eher mögliche Rollen bei der Dienstnutzung.

Kapitel 1: Einführung

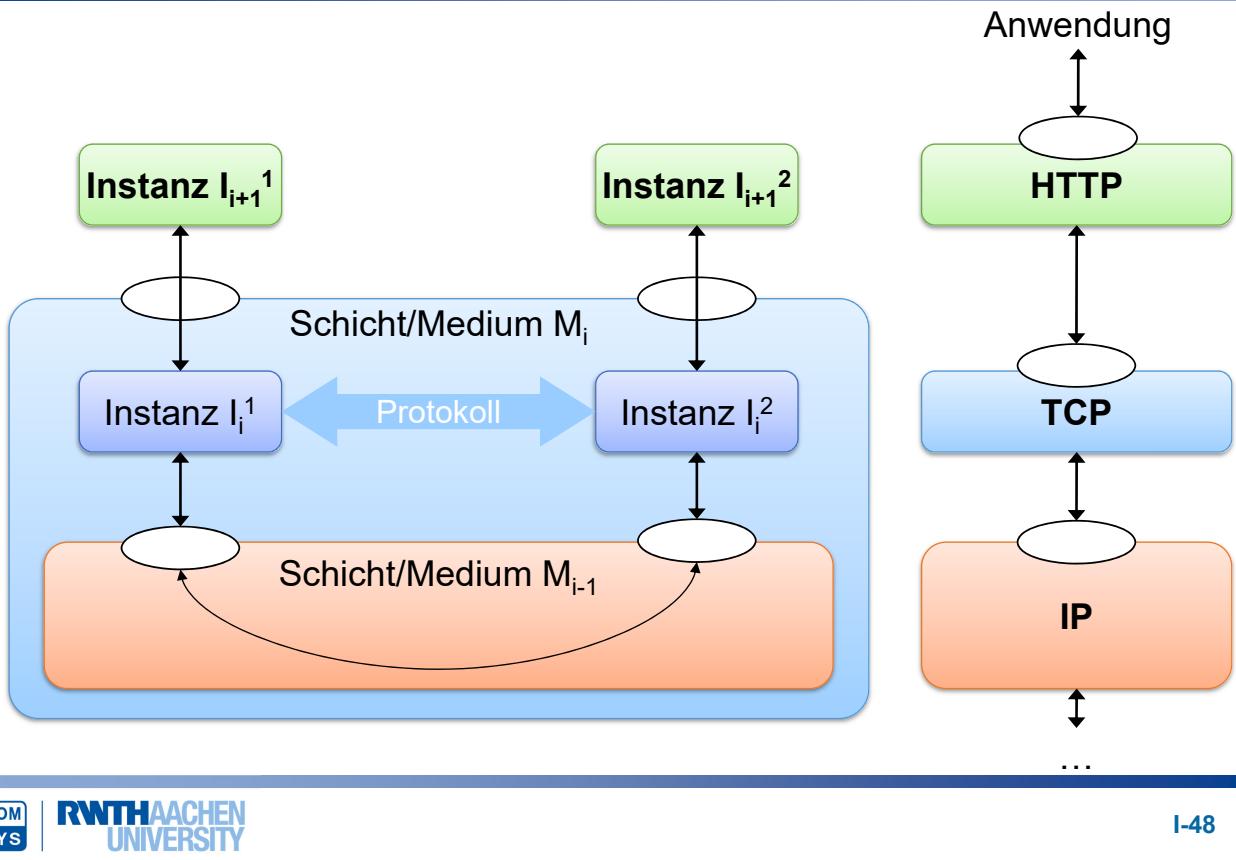
- **Einführung und Begriffe**

- ▶ Was ist Datenkommunikation?
- ▶ Information, Daten, Signale
- ▶ Netze

- **Allgemeine Grundlagen**

- ▶ Dienste
- ▶ **Protokolle und Schichten**
- ▶ Kommunikationsarchitekturen

Kommunikationsprotokolle



Bislang wurde behandelt, welche Kommunikationsfunktionalität in Form von Diensten an einer Schichtgrenze (der Dienstschnittstelle) angeboten wird. D.h.

- Der Kommunikationsdienst beschreibt, welche Funktionalität (was) angeboten wird.
- Das Kommunikationsprotokoll beschreibt, wie diese Funktionalität erbracht wird.

Zur Untersuchung der Protokolle ist es also notwendig, in die Schicht hineinzuschauen und die darin „arbeitenden“ Komponenten zu beleuchten. Im Kommunikationsbereich hat sich für diese Komponenten der Begriff der „Instanzen“ (engl. *entities*) durchgesetzt.

Ein Protokoll legt das Verhalten einer Instanz (*entity*) fest. Je zwei Partnerinstanzen (*peer entities*) realisieren das Protokoll und erbringen dadurch den entsprechenden Kommunikationsdienst.

Charakteristisch für ein Kommunikationsprotokoll ist, dass es zwischen zwei Protokollinstanzen abläuft. Im Folgenden soll der oben angedeutete Zusammenhang zwischen Protokoll und Dienst verdeutlicht werden.

Die Partnerinstanzen, zwischen denen das Protokoll abläuft, erbringen einen Dienst des Mediums M_i und setzen dazu auf einem Dienst des Mediums M_{i-1} auf.

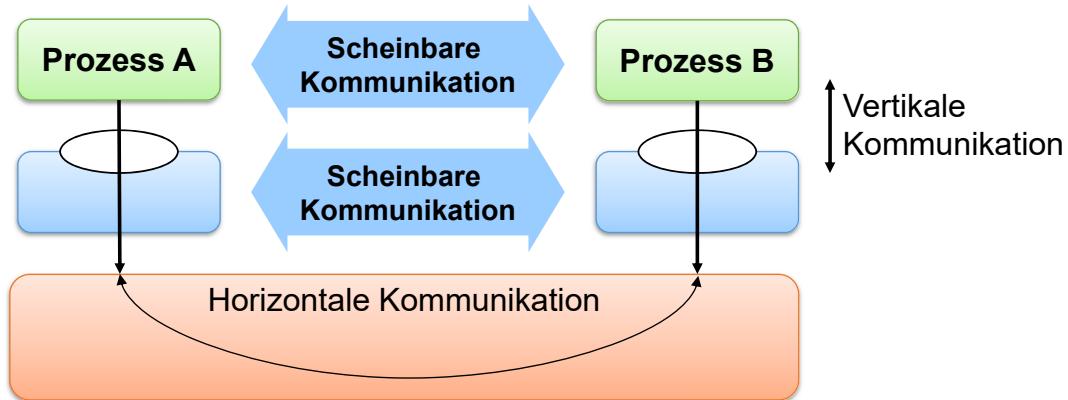
In der Abbildung auf der Folie wird gemäß dem zuvor eingeführten Dienstmodell von Medium gesprochen. Dieser Begriff ist gleichbedeutend mit dem Begriff der Schicht oder Ebene. In diesem Beispiel sind die Instanzen I_i^1 und I_i^2 die Partnerinstanzen. Sie erbringen für die darüberliegenden Instanzen

Schicht- i -Dienste und nutzen die vom darunterliegenden Medium angebotenen Medium_{i-1}-Dienste.

Scheinbare & tatsächliche Kommunikation

- Unterschied: Horizontale und vertikale Kommunikation

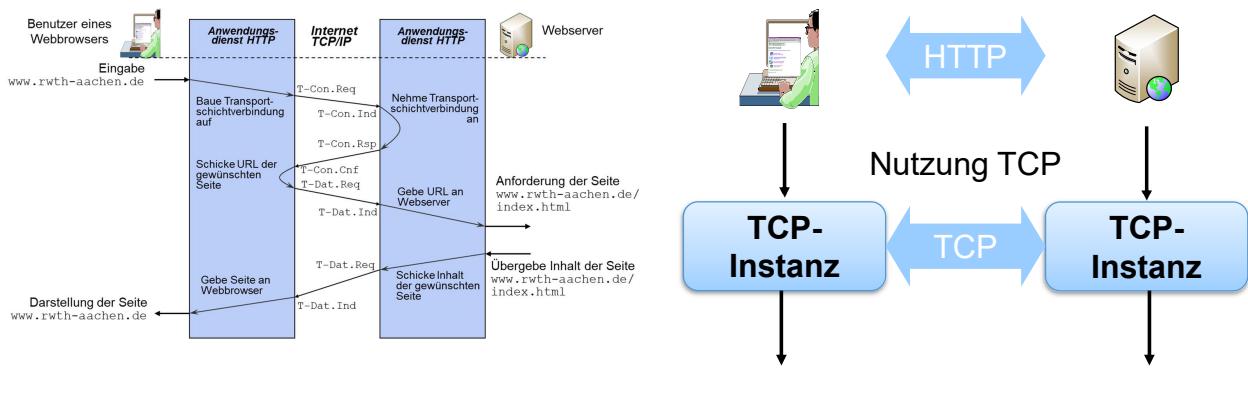
- ▶ **Horizontal:** Kommunikation zwischen Instanzen der Protokolle
 - Kommunikation scheinbar direkt zwischen Kommunikationspartnern
- ▶ **Vertikal:** tatsächliche Kommunikation innerhalb eines Protokollstapels
 - Tiefer liegenden Schichten erbringen den Kommunikationsdienst



Kommunikationsprotokolle regeln intern innerhalb einer Schicht den Datenaustausch zwischen Instanzen der jeweiligen Schicht. Die Kommunikation der Instanzen über das Protokoll findet scheinbar direkt zwischen den Instanzen statt. Dies wird auch horizontale Kommunikation genannt. Nur auf der untersten Schicht findet horizontale Kommunikation über das physikalische Medium tatsächlich statt; auf allen höheren Ebenen ist die horizontale Kommunikation nur virtuell.

Tatsächlich erbringen aber die tiefer liegenden Schichten den Kommunikationsdienst. Die Protokolldateneinheiten werden an die nächsttiefer liegende Schicht weitergereicht und durch diese zur Protokollinstanz des Kommunikationspartners gebracht. Da die nächsttiefe Schicht dadurch eventuell wieder auf tiefere Schichten zurückgreift, durchlaufen die zu versendenden Daten einen Stapel an Diensten, die durch eigene Protokolle realisiert werden. Man spricht hierbei von einem Protokollstapel (Protokoll-Stack). Durch diesen Protokoll-Stack werden die Daten vertikal weitergereicht.

Scheinbare & tatsächliche Kommunikation

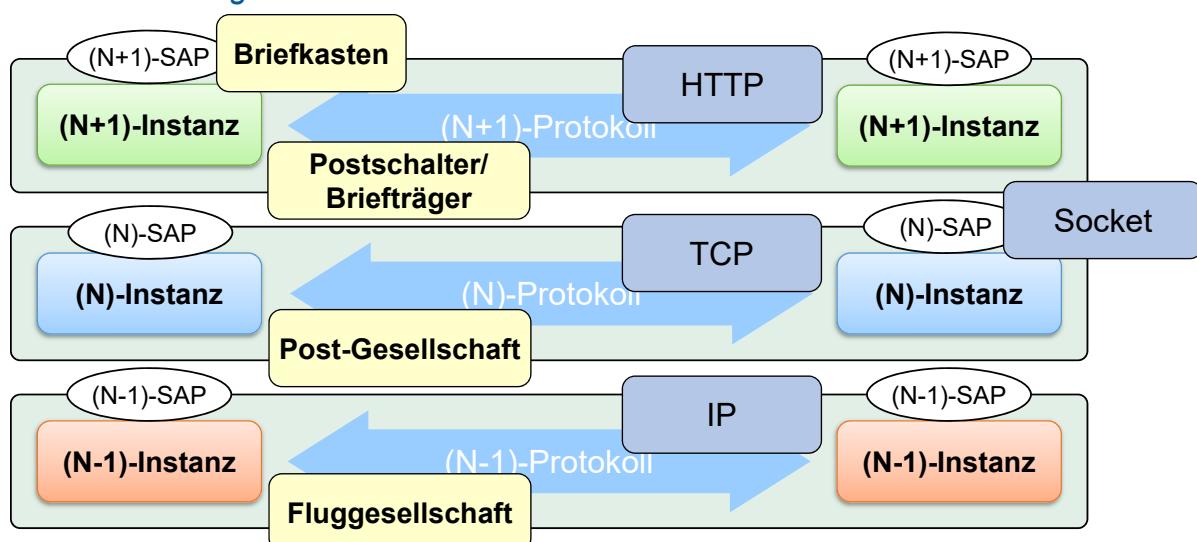


• Beispiel Webseite

- ▶ Browser und Webserver kommunizieren (scheinbar) horizontal
- ▶ Tatsächliche Kommunikation vertikal via TCP
- ▶ TCP-Instanzen kommunizieren ebenso (scheinbar) horizontal
- ▶ ...

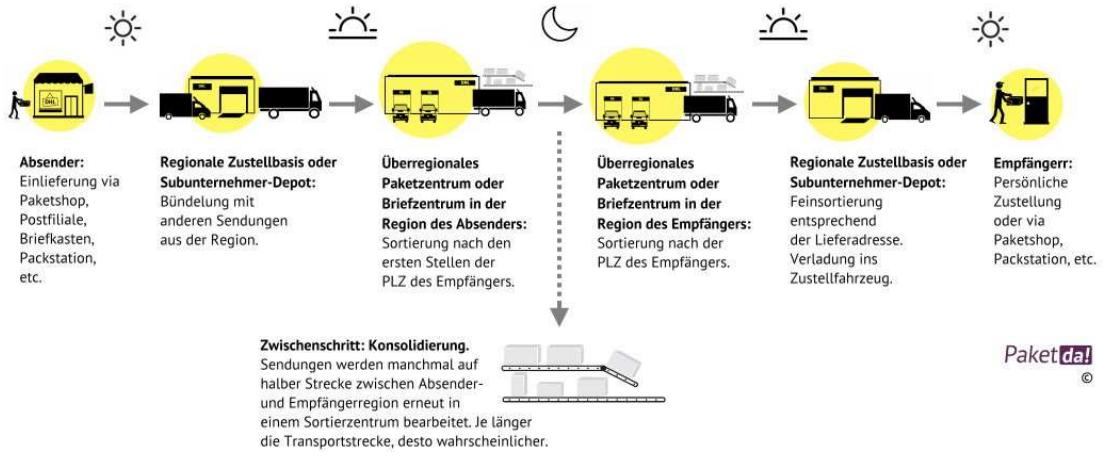
Protokolle und Protokollinstanzen

- **Protokolle:** Regeln/Formate für den Datenaustausch zwischen Rechnersystemen
- **Protokollinstanzen:** Protokollimplementierungen in den Rechnersystemen
 - **Schichtung** von Protokollinstanzen

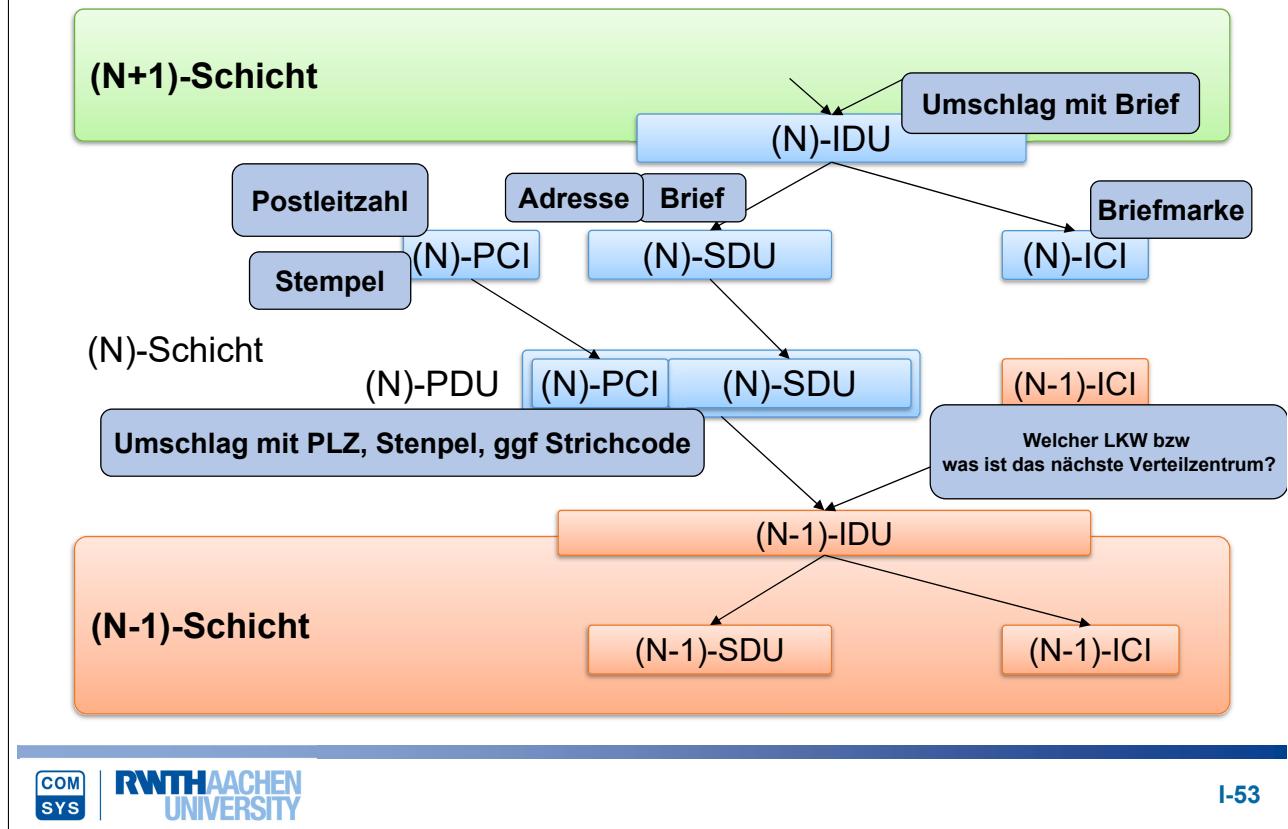


Der Dienst, den eine Schicht erbringt, wird durch das Zusammenwirken von Protokollinstanzen dieser Schicht (knapp auch Schichteninstanzen) auf den beteiligten Rechnersystemen erbracht – eine Schicht zieht sich also über ein gesamtes Kommunikationssystem hinweg. Diese Protokollinstanzen sind konkrete Implementierungen eines Protokolls auf einem Rechnersystem. Das Protokoll enthält alle Regeln, die für die Erbringung des Dienstes auf dieser Schicht erforderlich sind, sowie Formate, in denen die Daten zwischen den Protokollinstanzen ausgetauscht werden. Das Protokoll wird stets nur zwischen den Instanzen auf dieser Schicht abgearbeitet.

Auf einer Schicht existieren meist unterschiedliche Protokolle, die verschiedene Dienste erbringen können – wie beispielsweise ein Protokoll für verbindungsorientierte Kommunikation und ein Protokoll für verbindungslose Kommunikation, welche die vorher bereits dargestellten Dienstprimitive und Regeln zu ihrem Austausch definieren.



Dienst- und Protokolldateneinheiten: Vertikale Kommunikation im Detail



Im ISO/OSI-Referenzmodell (das später genauer erläutert wird) wird das Zusammenspiel von zwei Instanzen an der Dienstschnittstelle in einem Dienstmodell genau festgelegt. Die Zusammenarbeit erfolgt in den folgenden Schritten:

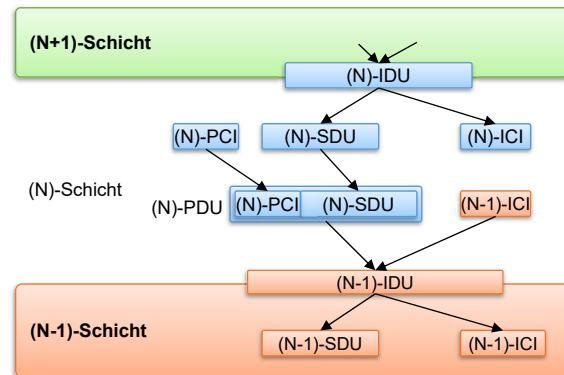
- (1) Die $(N+1)$ -Instanz über gibt an der Dienstschnittstelle eine (N) -Interface Data Unit (IDU).
 - (2) Die (N) -Instanz teilt die (N) -IDU in zwei Teile auf:
 - a) transparent zu übertragende Nutzdaten: (N) -SDU (Service Data Unit)
 - b) Steuerinformationen für die Dienstschnittstelle: (N) -ICI (Interface Control Information)
 - (3) Zur Übertragung der (N) -SDU ist gemäß dem vereinbarten Kommunikationsprotokoll eine (N) -PCI (Protocol Control Information) zu erzeugen, die gemeinsam mit der (N) -SDU die (N) -PDU bildet. Im Verlauf der Vorlesung werden die PCI als Header bezeichnet, da sie meist der SDU vorangestellt werden.
- Diese (N) -PDU wird transparent zwischen den Protokollinstanzen der Schicht N übertragen.
- (4) Zur Übertragung der PDU durch die darunterliegende Schicht ist entsprechende Kontrollinformation für die untere Schnittstelle zu erzeugen, die (N) -PDU und diese $(N-1)$ -ICI bilden die $(N-1)$ -IDU.

Zusammenfassung der Begriffe:

- **(N)-Schnittstellendateneinheit (IDU):** Dies ist die Dateneinheit, die die $(N+1)$ -Schicht an die (N) -Schicht über den Dienstzugangspunkt (Service Access Point - SAP) über gibt. Sie enthält die Nutzdaten (N) -SDU und Schnittstellenkontrollinformationen (N) -ICI.
- **(N)-Schnittstellenkontrollinformationen (ICI):** Diese Dateneinheit enthält Informationen über den auszuführenden Dienst, z.B: Parameter des Dienstes, Länge der SDU, ...
- **(N)-Dienstdateneinheit (SDU):** Daten der $(N+1)$ -Schicht ($(N+1)$ -PDU), die transparent zwischen (N) -SAPs übertragen werden.
- **(N)-Protokollkontrollinformationen (PCI):** Informationen, die (über vertikale Kommunikation) zwischen den einzelnen (N) -Instanzen ausgetauscht werden, um den Protokollablauf zu steuern.
- **(N)-Protokolldateneinheit (PDU):** Daten, die zwischen (N) -Partnerinstanzen ausgetauscht werden. Eine (N) -PDU setzt sich aus (N) -PCI und (N) -SDU zusammen.

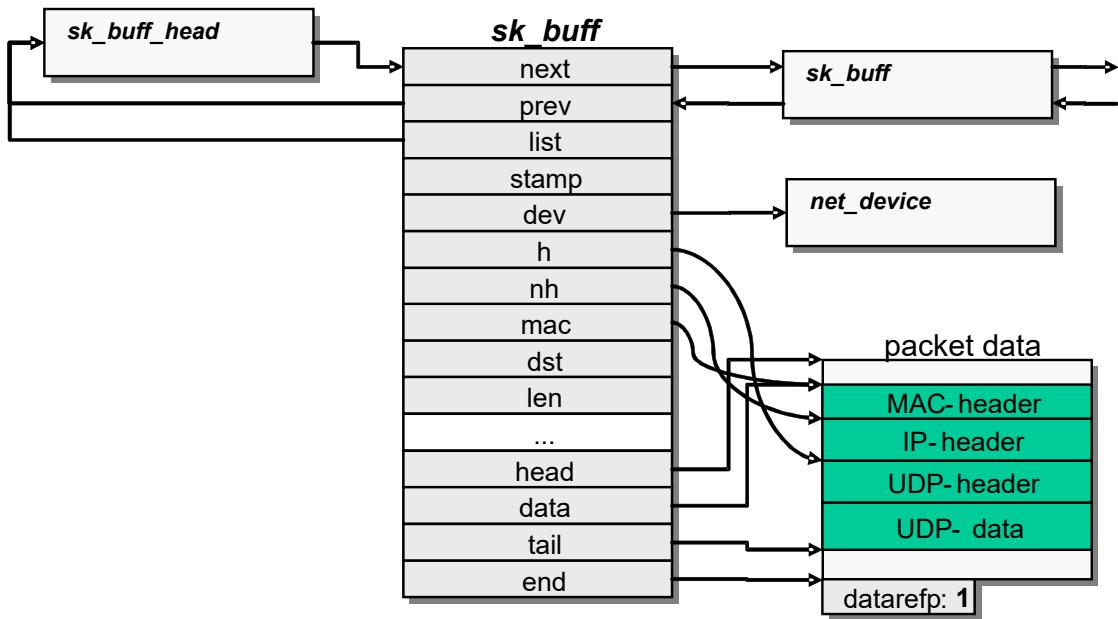
- Some questions:

- ▶ How are packets represented in the Kernel?
- ▶ How to realize SDU, PDU, IDU, ICI, etc.?
- ▶ Which variables do we need?
- ▶ One structure for all protocols
or individual structures for each protocol?
- ▶ ...

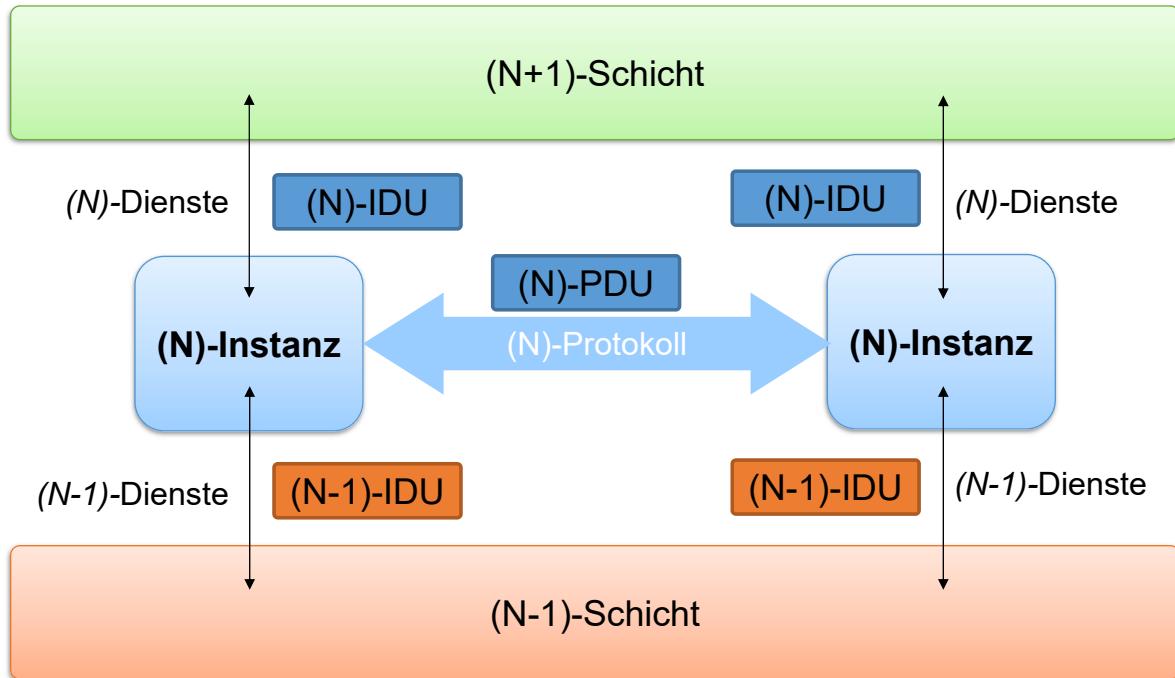


Socket Buffer: Represents Packets in the Linux Kernel

Auszug aus der Vorlesung
Communication Systems Engineering



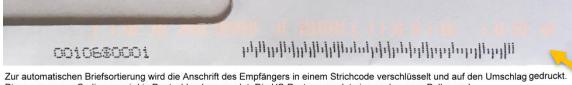
Kommunikation innerhalb/zwischen Systemen



Transparenz der Schichten

• Transparenz

→ PDU ist SDU (N+1) und in Schicht N als Nutzdaten betrachtet
Inhalts, *Transparenz* der Benutzerdaten (SDU)



Zur automatischen Briefsortierung wird die Anschrift des Empfängers in einem Strichcode verschlüsselt und auf den Umschlag gedruckt.

Die neon-orange Codierung wird in Deutschland verwendet. Die US-Post verwendet einen schwarzen Balkencode.

→ Einfluss auf die Anzahl der Schichten führt in den unteren Schichten zu einer ständigen Verdopplung des Anteils der PCIs am gesamten Datenumfang



(N+1)-PDU

Brief

N+1

Brief in Transportkiste
(N)-PCI (N+1)-PDU (N)-PCI

Brief in Transportkiste in Frachtcontainer
(N)-PDU (N-1)-PCI

N1

N-1



I-57

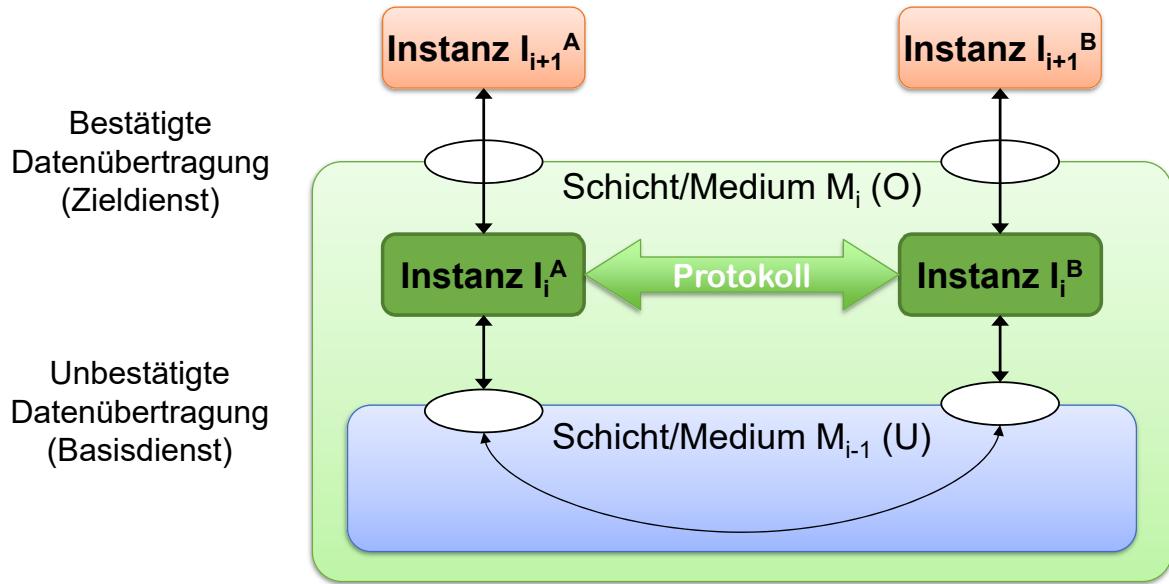
Jede Schicht soll unabhängig von allen anderen implementiert werden können, um Interoperabilität zwischen beliebigen Implementierungen auf allen Schichten zu gewährleisten. Dies hat aber auch einen Nachteil: die PDU einer Schicht, die an die nächsttiefer Schicht weitergereicht wird, wird dort als SDU aufgefasst – als Nutzdateneinheit, die unangetastet bleibt und mit neuen Kontrollinformationen versehen wird. Dadurch besteht eine PDU auf unterster Ebene aus den eigentlichen Nutzdaten selbst plus einer ganzen Kette an PCIs. Die zu übertragene Datenmenge wird umso mehr aufgebläht, je mehr Schichten verwendet werden. Dies kann einen enormen Overhead erzeugen – und das nicht nur bei der Menge der zu übertragenen Informationen, sondern auch bei der Verarbeitung der Daten durch den gesamten Protokoll-Stack.

Schichten und Protokolle

- **Beispiel: Alternating Bit Protocol**
 - ▶ Dienst vs. Protokoll
 - ▶ Interaktion zwischen Schichten

Im Folgenden werden die eingeführten Konzepte von Dienst, Protokoll und Schichtung anhand eines einfachen Beispiels im Zusammenhang dargestellt.

Beispiel: „Alternating Bit Protocol“



Als Beispiel soll ein Dienst betrachtet werden, den ein Medium „O“ erbringt. Dieses Medium bietet die Funktionalität „bestätigte Datenübertragung“. Dazu ist die Definition eines Protokolls zwischen den Instanzen des Diensterbringers notwendig. Das Protokoll erbringt die virtuell horizontale Kommunikation zwischen den Instanzen, nutzt zur Übertragung aber einen Basisdienst, den das Medium „U“ anbietet – dieser Basisdienst ist unbestätigt und unzuverlässig.

Alternating Bit Protocol: Informelle Beschreibung

- Dienst zur bestätigten Datenübertragung

- ▶ Auf Basis eines Dienstes zur unbestätigten Datenübertragung
- ▶ Nötig: *Quittungen*
 - Mit alternierenden Quittungsnummern 0 und 1
 - Nachdem eine quittierte Datenübertragung vollständig beendet ist (Quittung erhalten), geht die Instanz in einen Ruhezustand über, bis die nächste Übertragungsanforderung eintrifft
- ▶ Dienstprimitive
 - DtReq
 - DtInd
 - DtRsp
 - DtCnf

Als Zusammenfassung und Verdeutlichung der eingeführten Konzepte wird das Alternating Bit Protocol betrachtet. Es ist ein einfaches Protokoll, das eine elementare und wichtige Funktionalität innerhalb der Datenkommunikation erbringt, nämlich

- (i-1)-Dienst: unbestäigte und unzuverlässige Datenübertragung
- i-Dienst: bestäigte Datenübertragung

Um diese Dienstdifferenz von einem unbestätigten, unzuverlässigen zu einem bestätigten Dienst ausgleichen zu können, muss das Protokoll eine Quittierung einführen, durch die letztendlich sichergestellt wird, dass die übertragenen Daten auch beim Empfänger angekommen sind.

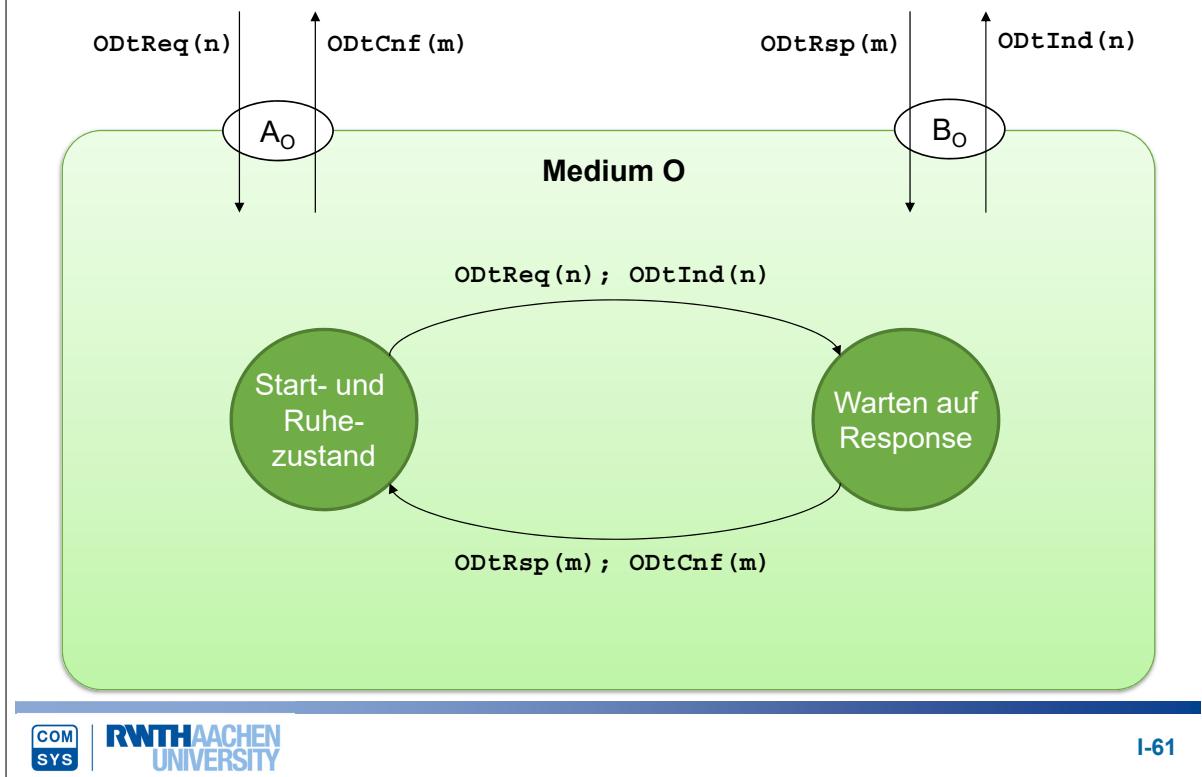
Protokollfunktionalität: Quittierung und Fehlerbehandlung

Die Beschreibung der Protokollfunktionalität umfasst drei Aspekte:

- Ablauffestlegungen innerhalb einer Instanz: Zustandsübergangsdiagramm (endlicher Automat - Protokollautomat)
- Bei den Übergängen zu erbringende Funktionalität: Erweiterung des endlichen Automaten um Aktionen auf interne Variablen
- Protokolldateneinheiten: Bitmuster oder Datenstruktur

Das Beispielprotokoll kommt aufgrund seiner Einfachheit noch ohne die Aktionen (Punkt 2) aus. Die beiden anderen Aspekte werden im folgenden anhand des Beispielprotokolls aufgezeigt.

Dienst O – Spezifikation des Dienstes O

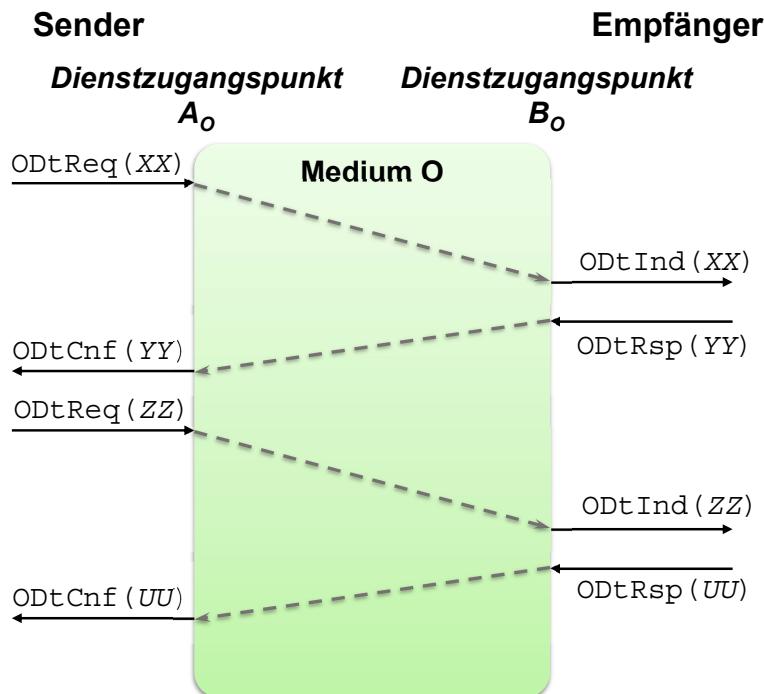


Wie das Diagramm (sehr grober Protokollautomat) für den zu realisierenden Dienst O aus dem Beispiel zeigt, handelt es sich hier um einen einfachen Dienst mit nur einer Dienstfunktion OD_t, die einen bestätigten Datentransfer realisiert.

Aus der Sicht des Dienstnutzers von O werden sämtliche Störungen, wie Bitfehler oder Verlust ganzer Dateneinheiten, vollständig durch das Protokoll des Dienstes verdeckt. (In der Realität geht das nur bis zu einem gewissen Grad, weshalb man i.d.R. immer noch Ausnahmesituationen bei einer solchen Dienstschnittstelle berücksichtigen sollte.)

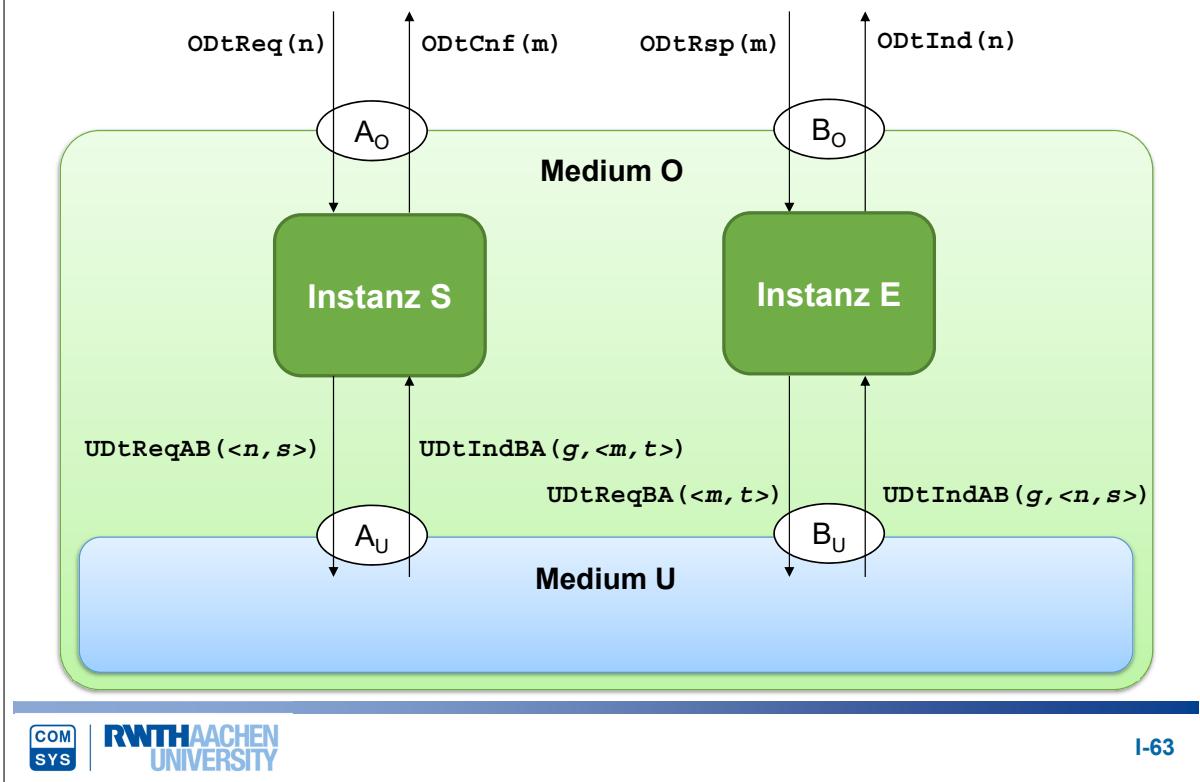
Die Initiierung der bestätigten Datenübertragung erfolgt über den Zugangspunkt A_O – dieser beschreibt die Rolle des sendenden Dienstnutzers. Zugangspunkt B_O beschreibt dementsprechend die Rolle des empfangenden Dienstnutzers. Über B_O werden zwar auch Dateneinheiten übertragen, die aber lediglich eine Bestätigung transportieren.

Dienstablauf in einem Weg-Zeit-Diagramm



Die Arbeitsweise eines Protokolls für O bzw. der Instanz, die dieses Protokoll ausführt, lässt sich am besten begreifen, wenn man typische Abläufe durch den Protokollautomaten nachvollzieht. Auf dieser Folie ist ein solcher Ablauf in einem Weg-Zeit-Diagramm dargestellt.

Zusammenspiel der Schichten O und U



Die Nutzdaten n werden zur bestätigten Übertragung durch ODtReq am SAP A_O bereitgestellt. Medium O realisiert nun das Alternating Bit Protocol. Die Protokollinstanz S (ender) benutzt den unbestätigten Übertragungsdienst am unteren Zugangspunkt A_U und überträgt neben den Nutzdaten n außerdem eine Sequenznummer s . Diese wird bei der ersten Übertragung auf 0 gesetzt. Über das Medium wird also $<n, 0>$ übertragen.

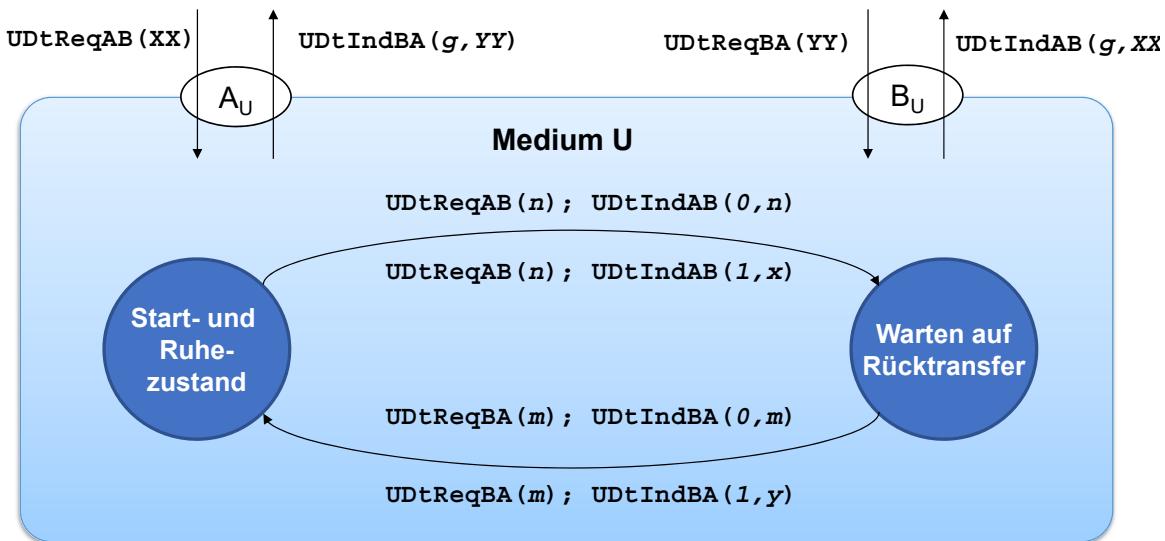
Bei einem störungsfreien Protokollablauf würden die Daten über B_U an die Protokollinstanz E (mpfänger) ausgeliefert (UDtIndAB), und über ODtInd an den Dienstnutzer des Mediums O weitergereicht. Der Dienstnutzer generiert eine Quittung m , welche durch E wiederum um einen Kontrollparameter t erweitert wird, bevor sie über UDtReqBA an S zurückgesendet wird. Der Parameter t dient zur Quittierung des vorherigen Requests und ist ebenfalls eine Sequenznummer. Ist der Wert gleich zu dem vorher verwendeten Wert für s , wird die Übertragung als erfolgreich angesehen und die enthaltene Bestätigung m an den Dienstnutzer des Mediums O weitergereicht.

Anhand des Beispielablaufs lässt sich ein wichtiger Aspekt, der bei einer Protokollbeschreibung auftritt, feststellen:

In einer Protokollbeschreibung ist neben dem Zusammenwirken der am oberen und unteren Zugangspunkt auftretenden Dienstprimitive auch genau zu definieren, welche Protokolldateneinheiten (Protocol Data Units, PDU) über das Medium auszutauschen sind. In diesem Fall ist es der Kontrollparameter „Sequenznummer“, der zusammen mit den zu übertragenden Nutzdaten die PDU bildet. Die Information, die in Form von solchen in ihrer Syntax genau festzulegenden PDUs beschrieben werden muss, wird im Beispiel immer in den spitzen Klammern $\langle \text{PDU-Inhalt} \rangle$ geschrieben.

Zusätzlich zur Sequenznummer wird ein Kontrollparameter g eingeführt, der allerdings kein Bestandteil der PDUs im Medium O mehr ist, sondern der Interaktion zwischen U und O innerhalb eines Protokoll-Stacks dient. Über diesen Parameter kann eine Instanz des Mediums U der Instanz des Mediums O im gleichen Protokoll-Stack signalisieren, ob Daten korrekt empfangen wurden oder ob sie verfälscht wurden. g ist also eine Interface Control Information, während die Sequenznummer eine Protocol Control Information ist.

Basisdienst U – Spezifikation des Dienstes U



- Parameter g : Signalisierung von U an O, dass die Daten während der Übertragung verfälscht wurden
 - $g=0$: korrekter Empfang; $g=1$: Daten verfälscht

Der Basisdienst U unterstützt eine unbestätigte Datenübertragung in beide Richtungen. Aus der Sicht des Dienstnehmers (in diesem Fall die Instanzen S(ender) und E(empfänger), die das Alternating Bit Protocol realisieren), ist diese Dienstschnittstelle also symmetrisch.

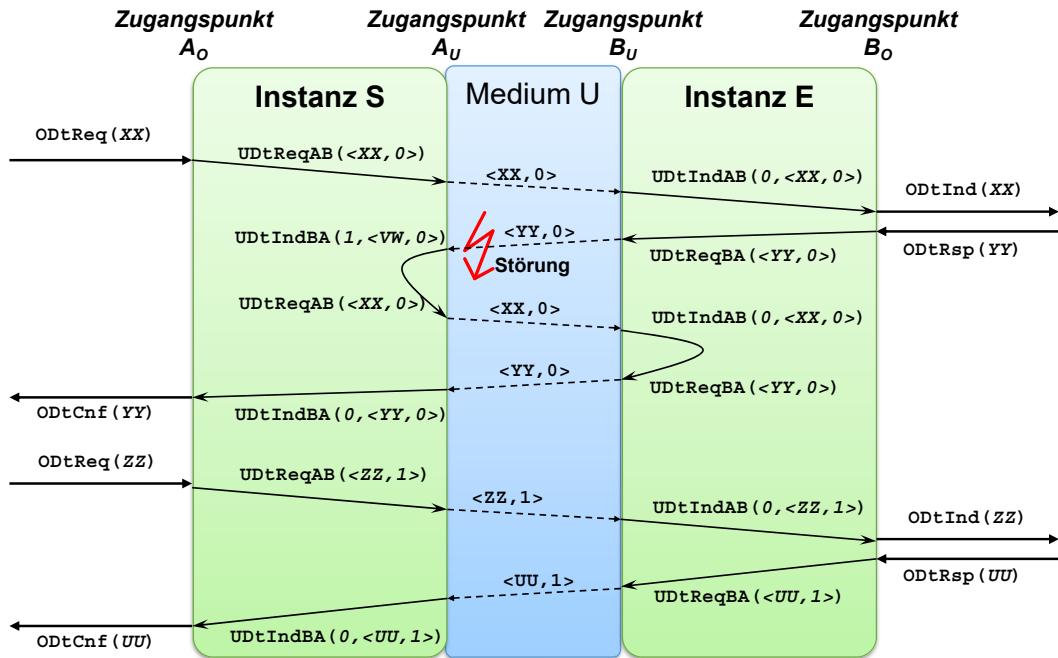
Betrachtet man jetzt aber das Zustandsübergangsdiagramm, erkennt man recht deutlich, dass zunächst eine Übertragung in der Richtung A → B erfolgen muss und daraufhin eine Übertragung in die entgegengesetzte Richtung erfolgt.

Der Basisdienst U berücksichtigt die Problematik, dass Störungen auftreten können, in folgender Weise:

Beim Basisdienst U werden dem Dienstnehmer medieninterne, nicht behebbare Fehler durch einen entsprechenden auf 1 gesetzten Kontrollparameter angezeigt.

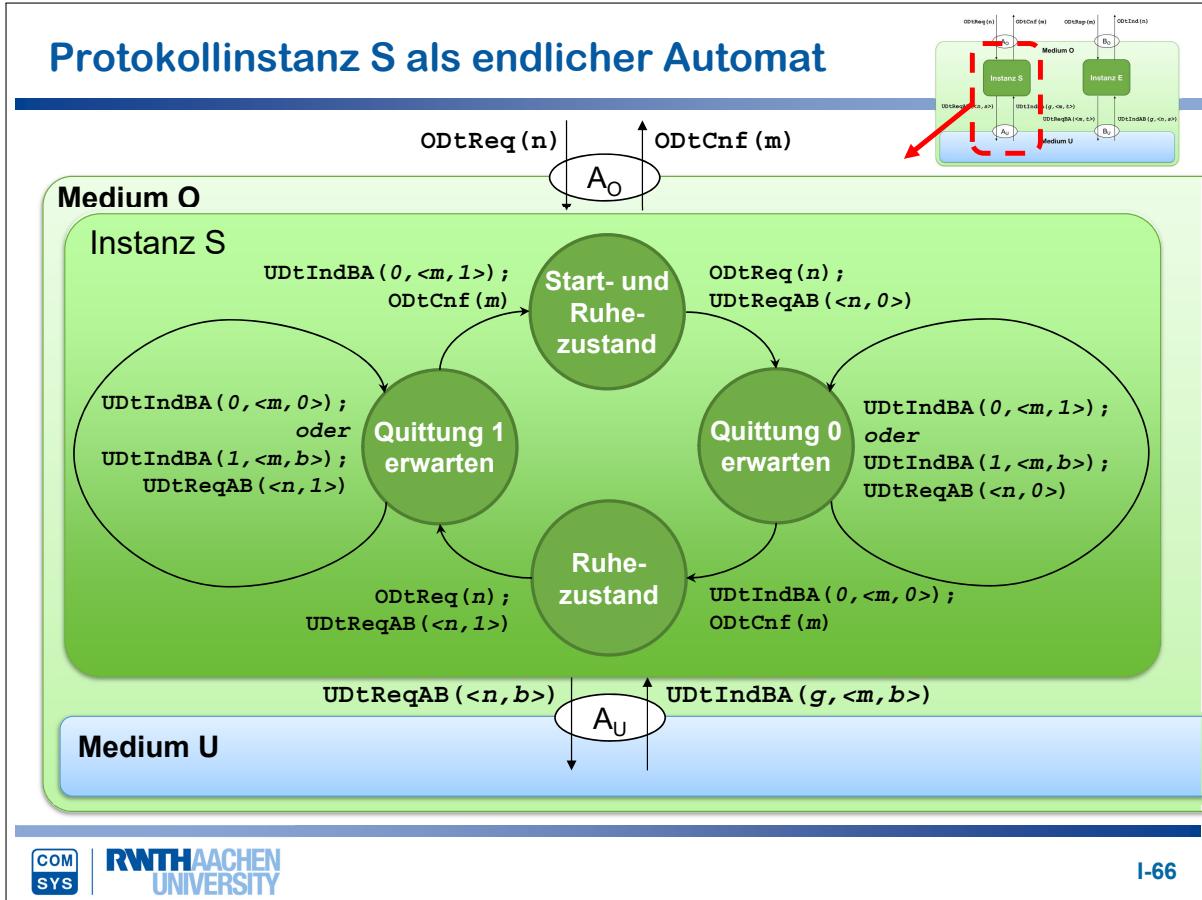
In den endlichen Automaten, welche das Verhalten der Instanzen S und E beschreiben, wird dieser Parameter ausgewertet. Bei der Anzeige eines Übertragungsfehlers muss eine Übertragungswiederholung erfolgen, da der Dienstnutzer des Mediums O erwartet, dass die Daten übertragen werden. Zusammen mit der Beschreibung der Dienste U und O bilden die Instanzen S und E eine vollständige Beschreibung des vorliegenden Kommunikationsszenarios.

Beispielablauf



Hiermit sind alle vier Komponenten, also die zwei Instanzen S und E sowie der Ziel- und Basisdienst O und U, beschrieben.

Protokollinstanz S als endlicher Automat



Schnittstellen des Protokollautomaten für S:

- ODReq(n): Anforderung der Dienstleistung „bestätigte Datenübertragung der Nutzdaten n“
- UDReqAB(<n,0>): Anforderung der Dienstleistung „unbestätigte Datenübertragung von einem Dienstzugangspunkt A_U nach B_U“ mit Nutzdaten n und Sequenznummer 0
- UDIndBA(g,<m,b>): Anzeige der Dienstleistung „unbestätigte Datenübertragung von einem Dienstzugangspunkt B_U nach A_U“ mit den Parametern
 - g=Verfälschung ja/nein (1/0),
 - n, m=Nutzdaten,
 - b=Sequenznummer (0 oder 1)

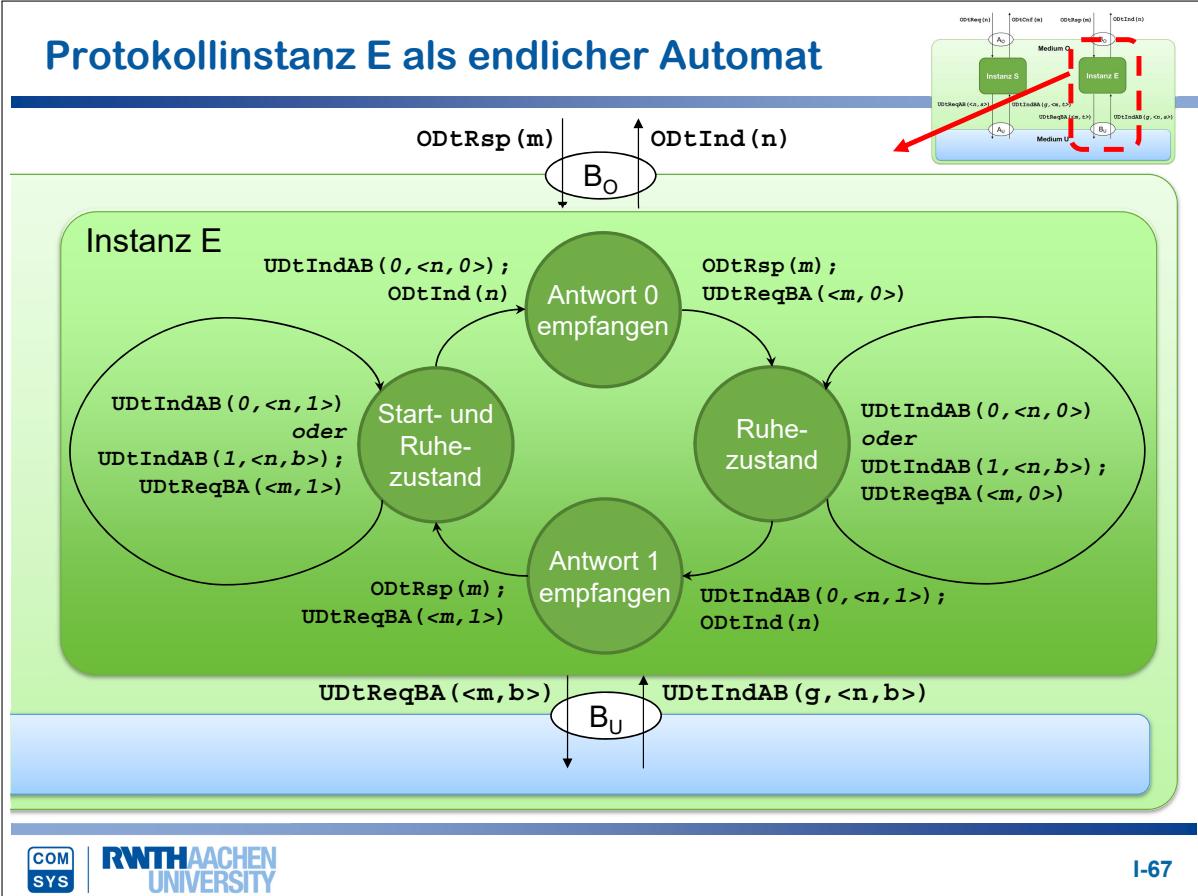
Aus diesen Beschreibungen lässt sich die Bedeutung der Dienstprimitive ableiten, die die Ausgaben des Automaten bilden: ODReq(m) ist die Bestätigung der Übertragung von Nutzdaten m, UDReqAB(<n,b>) ist die unbestätigte Übertragung der Nutzdaten n und der Sequenznummer b.

Der endliche Automat enthält insgesamt vier Zustände. Der Startzustand ist der oberste. Aus diesem kommt man in den Zustand „Quittung 0 erwarten“ durch die Eingabe ODReq(n) (eine Anfrage vom oberen Dienstzugangspunkt, Nutzdaten n bestätigt zu übertragen). Hierauf erfolgt die Ausgabe UDReqAB(<n,0>), d.h. es werden die Nutzdaten n und die Sequenznummer b=0 an den unteren Dienstzugangspunkt A_U zur unbestätigten Übertragung weitergegeben.

In diesem Zustand „Quittung 0 erwarten“ wartet die Instanz jetzt auf die Bestätigung, die durch eine entsprechende Anzeige UDInd am unteren Dienstzugangspunkt mit der Sequenznummer 0 erfolgt.

Das Alternating Bit Protocol kommt deshalb mit einer alternierenden, d.h. nur die Werte 0 und 1 annehmenden Sequenznummer aus, weil eine Instanz immer erst eine weitere Nachricht senden darf, nachdem die zuvor gesendete Nachricht quittiert wurde.

Protokollinstanz E als endlicher Automat

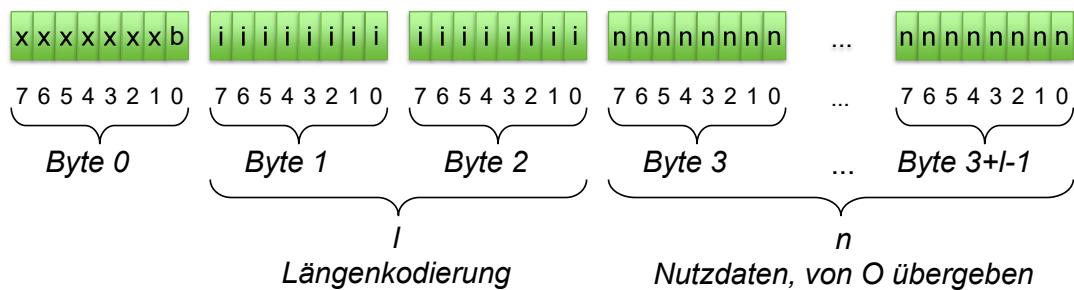


Die Auswertung des Verfälschungsparameters erfolgt je nachdem, wo man sich im alternierenden Ablauf gerade befindet, im "Start- und Ruhezustand" oder im "Ruhezustand" (selbiges für Instanz S in den anderen Zuständen). Die Reaktion der Instanz E auf eine Verfälschungsanzeige (also Wert = 1) besteht in der Übertragung von irgendwelchen Nutzdaten (hier: m) mit einem bewusst falsch gesetzten Sequenznummern-Wert. Im Beispiel: Instanz S erwartet anfangs die Sequenznummer 0, Instanz E antwortet bei einer Verfälschung aber mit der Sequenznummer 1.

Beim Alternating Bit Protocol wird eine negative Quittung und somit die Aufforderung zum nochmaligen Senden durch die Übertragung des gerade nicht-erwarteten Sequenznummern-Wertes erreicht.

Beschreibung von Protokolldateneinheiten

• Format



► Byte 0:

- Kodierung der Sequenznummer b im niederwertigsten Bit

► Byte 1+2:

- Kodierung der Länge l / der Nutzdaten n

► Folgebytes:

- Nutzdaten n

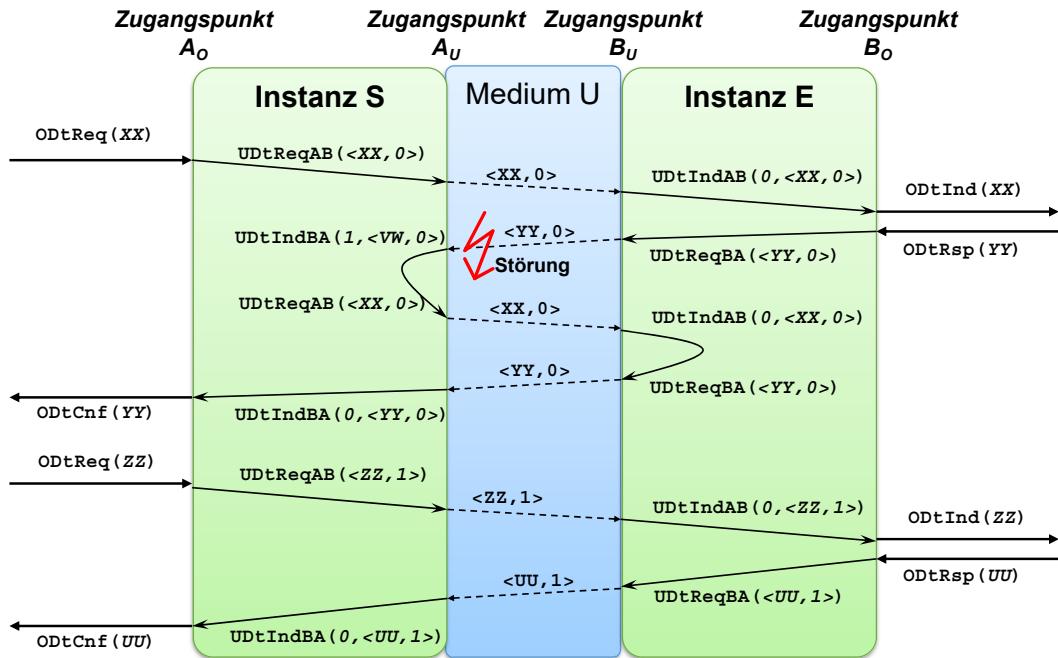
Die Protokolldateneinheiten unterscheiden sich von den Dienstprimitiven in einem ganz elementaren Punkt: während Dienstprimitive in einer Protokollnorm nur in ihrer Semantik beschrieben werden, muss zu jeder im Protokoll benutzten PDU auch der genaue syntaktische Aufbau festgeschrieben werden.

Der Grund hierfür liegt auf der Hand: während Dienstprimitive nur an internen Schnittstellen eines Kommunikationssystems auftreten, deren Realisierung implementierungsabhängig ist, werden PDUs tatsächlich zwischen den heterogenen Systemen ausgetauscht. Nur wenn einem System, welches eine PDU von einem entfernten System erhält, die genaue Struktur kennt, kann es diese auch verarbeiten.

Im Beispiel-Protokoll, dem Alternating Bit Protocol, kam u.a. eine PDU vor, in der die Nutzdaten n und die Sequenznummer b , notiert als $\langle n, b \rangle$, zu beschreiben ist. Für diese PDU ist nun in einer Protokollbeschreibung das genaue Bitmuster festzuschreiben. Hier ist jetzt genau anzugeben,

- in welcher Reihenfolge die nieder- und höherwertigen Bits angegeben sind; Entscheidung hier: höherwertiges Bit vor niederwertigem
- in welchen Bytes die zu kodierenden Protokollelemente enthalten sind; Entscheidung hier: Sequenznummer b im ersten Byte; Länge der Nutzdaten in den zwei folgenden Bytes; Nutzdaten in den übrigen Bytes.

Vollständiger Beispielablauf



Hiermit sind alle vier Komponenten, also die zwei Instanzen S und E sowie der Ziel- und Basisdienst O und U, beschrieben.

Kapitel 1: Einführung

- **Einführung und Begriffe**

- ▶ Was ist Datenkommunikation?
- ▶ Information, Daten, Signale
- ▶ Netze

- **Allgemeine Grundlagen**

- ▶ Dienste
- ▶ Protokolle und Schichten
- ▶ **Kommunikationsarchitekturen**

- **Architektur von Kommunikationssystemen**
 - ▶ Geschichtete Systeme
 - ▶ Notwendig: Standardisierung
 - Genaue Abgrenzung der Funktionalität der einzelnen Schichten
 - Definierte Schnittstellen zwischen den Schichten
 - ...
- **Relevante Architekturen**
 - ▶ ISO/OSI
 - ▶ TCP/IP (Internet)

Hier behandelt: ISO/OSI und TCP/IP – das erstere deshalb, da seine Terminologie sich bis heute gehalten hat und es eine gute Strukturierungshilfe bereitstellt; das zweite, da es das heute relevante Modell ist.

Andere Architekturen wie ISDN und ATM waren eine Zeit lang von Relevanz, wurden aber mittlerweile zum großen Teil von weiteren Entwicklungen abgelöst.

Standardisierungsgremien

- Nötig für den weltweiten Einsatz: Standardisierung
- International Standards Organization - ISO
 - ▶ Organisation, die auf freiwilliger Basis arbeitet (seit 1946)
 - ▶ Mitglieder: Standardisierungsorganisationen vieler Länder
 - ▶ Hat 200 Technical Committees (TC) für sehr weites Spektrum von Standards
 - TCs haben spezifischen Aufgaben, z.B. TC97 für Computer und Informationsverarbeitung
 - TCs haben Subkomitees, die wiederum in Arbeitsgruppen unterteilt sind
 - ▶ Bahnbrechende Leistung von ISO in der Datenkommunikation:
ISO/OSI-Referenzmodell als erstes Modell zur Strukturierung einer Kommunikation zwischen Rechnern (*OSI: Open Systems Interconnection*)
 - Bahnbrechendes Konzept, aber keine vernünftigen Produkte!



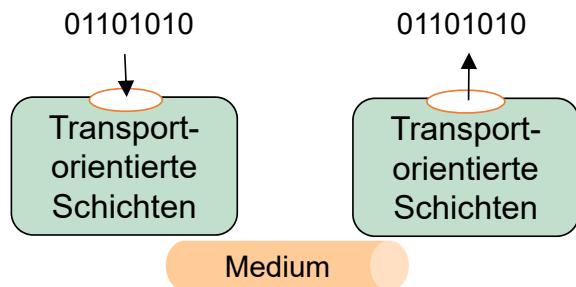
www.iso.ch

Ein sehr umfassendes Modell zur Umsetzung von Kommunikationssystemen ist das ISO/OSI-Referenzmodell (seit 1984 ein ISO-Standard). Es teilt den komplexen Kommunikationsvorgang in 7 Schichten auf, die jeweils logisch schlüssige und möglichst unabhängig behandelbare Teile eines Kommunikationsvorgangs umfassen und durch wohldefinierte Schnittstellen miteinander verbunden sind. Die hier gewählte Aufteilung hat sich als sinnvoll erwiesen; auch wenn das ISO/OSI-Modell in der Praxis nicht eingesetzt wird, haben sich andere Modelle im Wesentlichen an die hier definierte Aufteilung und Terminologie gehalten - zumindest ein Teil des OSI-RMs dient den standardisierten und auch herstellerspezifischen Protokollwelten aus dem Bereich der Rechnerkommunikation als konzeptionelle Basis.

Daher wird das ISO/OSI-Modell im weiteren Verlauf der Vorlesung zwar nur eine untergeordnete Rolle spielen, seine generelle Struktur bestimmt aber den Ablauf der Vorlesung.

- **Transportorientierte Schichten**

- ▶ Transparente Ende-zu-Ende-Übertragung von Daten
 - Inhalt (Semantik) der Daten transparent
 - Kein Bezug zu Kooperationsbeziehung der Dienstnutzer
 - Elementare Datenübertragung, Funktionsumfang abhängig vom bereitzustellenden Dienst



Generell lassen sich die im ISO/OSI-Modell definierte Schichten in zwei Gruppen aufteilen: transportorientierte und anwendungsorientierte Schichten.

Die transportorientierten Schichten dienen, wie der Name schon sagt, ausschließlich zum Transport der Daten. Weitere Aspekte, die z.B.

die Kooperation der Anwendungen oder die Inhalte der Daten selbst betreffen, werden nicht berücksichtigt.

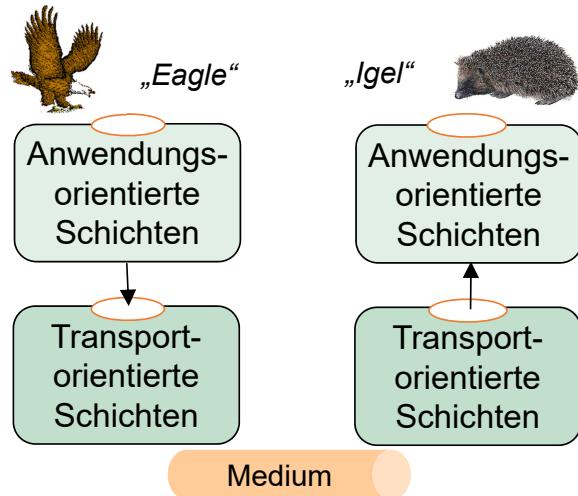
- Anwendungsorientierte Schichten

- ▶ Anwendungsbezogene Aspekte der Datenübertragung

- Semantik der Daten ist wichtig

- Kooperation der Teilnehmer unter formalen Gesichtspunkten (Strukturierung, Präsentation), z.B.

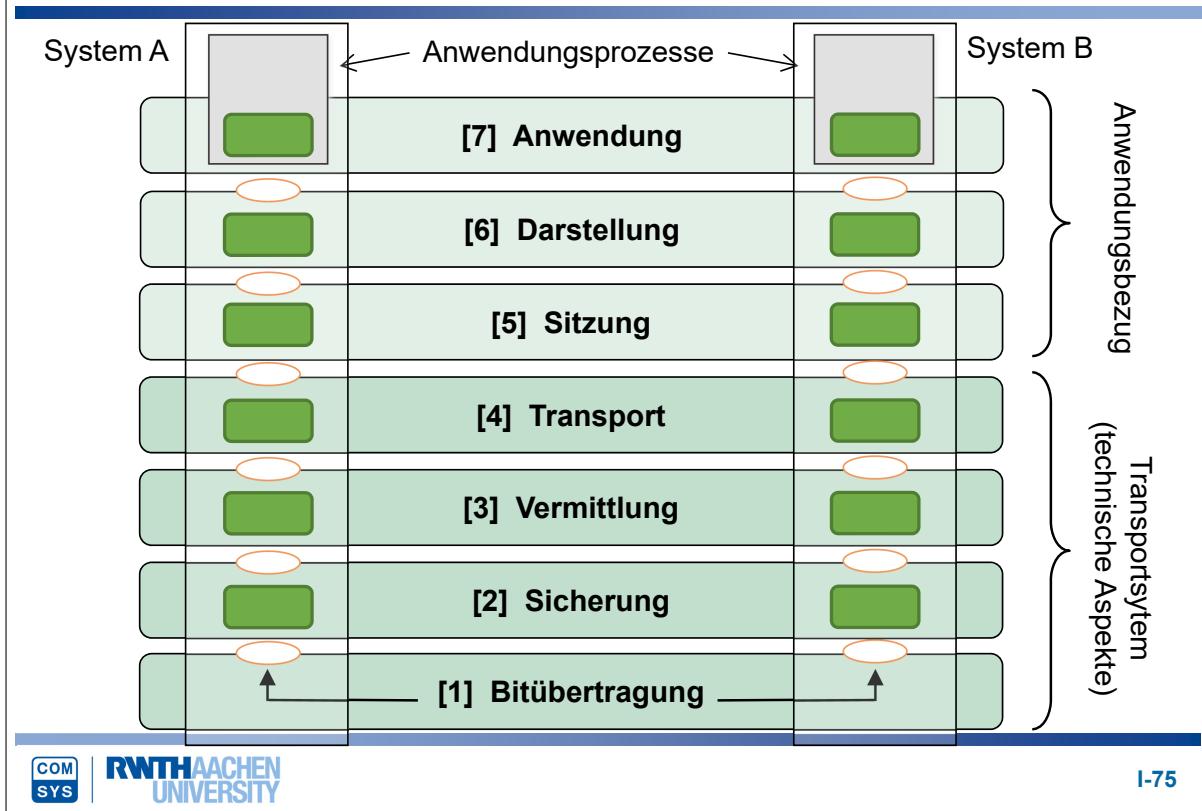
- Steuerung des Kommunikationsablaufs
- Kompensation von Fehlverhalten durch verteilte Transaktionen



Hierfür sind die anwendungsorientierten Schichten zuständig. Diese berücksichtigen Kooperationsaspekte, soweit sie in formaler Weise behandelbar sind. Teilschichten des Anwendungssystems kümmern sich außerdem um die Steuerung des Kommunikationsablaufs und um die Informationsdarstellung.

Der Zusammenhang mit dem Transportsystem findet dort im Anwendungssystem seinen Niederschlag, wo ein etwaiges Fehlverhalten, welches vom Transportsystem auf der reinen Übertragungsebene nicht erkannt werden kann, kompensiert werden muss.

ISO/OSI-Referenzmodell



Beim OSI-RM steht der *Modellierungsaspekt* im Mittelpunkt und weniger die durch OSI genormten Protokolle.

Diese Aussage gilt auch in Bezug auf die TCP/IP-Protokolle der Internet-Welt, wie auf einer der nachfolgenden Folien gezeigt wird.

Im OSI-RM bilden die unteren vier Schichten das sogenannte Transportsystem (transportorientierte Schichten), die drei oberen Schichten realisieren das sogenannte Anwendungssystem (anwendungsorientierte Schichten).

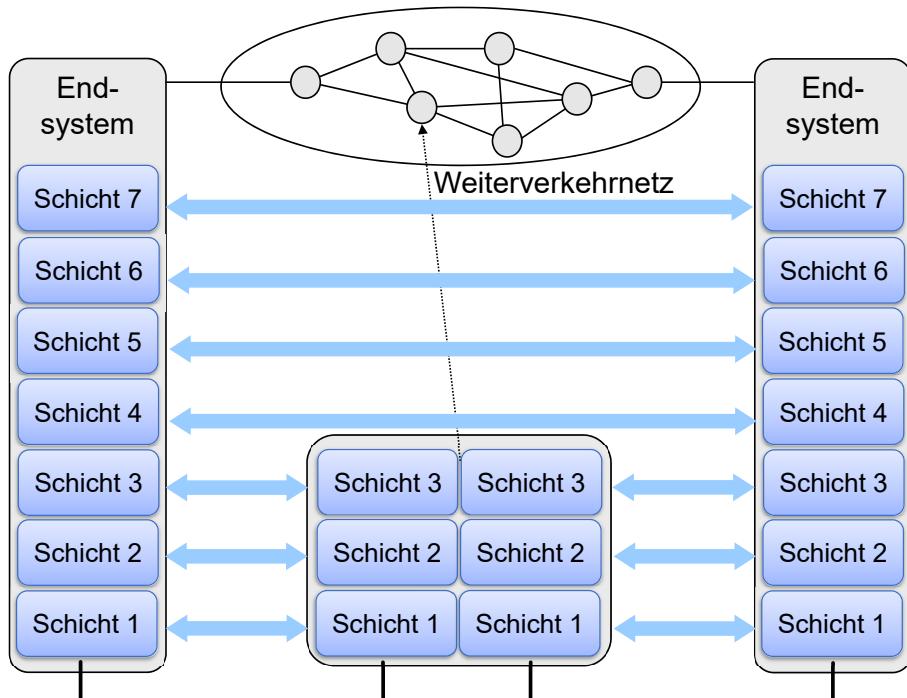
Die wesentlichen Aufgaben der Schichten sind:

- Bitübertragungsschicht: Darstellung von Bits auf dem physikalischen Medium, Normierung von Kabeltypen und Steckern
- Sicherungsschicht: Erkennung und Behebung von Übertragungsfehlern zwischen zwei direkt miteinander verbundenen Geräten, Flusskontrolle zur Vermeidung der Überlastung des Empfängers, Regelung des Medienzugriffs
- Vermittlungsschicht: Verbindung von Teilstrecken/Netzen zu einem globalen Netzwerk, weltweit eindeutige Adressierung, Bestimmung von günstigen Pfaden durchs Netz, Weiterleitung der PDUs, evtl. Auch Staukontrolle zur Vermeidung der Netzüberlastung und Regelung der Übertragungsqualität.
- Transportschicht: Adressierung von Anwendungsprozessen, Ende-zu-Ende-Verbindung, Behebung von Fehlern auf Schicht 3 (Erkennung von Datenverlust, Staukontrolle, ...)
- Sitzungsschicht: Strukturierung der Übertragung, Ablaufsteuerung/-koordinierung, Verwaltung von Dialogen

- Darstellungsschicht: Kommunikation wird trotz unterschiedlicher lokaler Datenformate der Teilnehmer ermöglicht, Konvertierung der Daten in Transportformat
- Anwendungsschicht: Stellt Dienste für bestimmte Anwendungsarten zur Verfügung, z.B. Dateitransfer, entfernter Prozeduraufruf, E-Mail. Z.B. Definition von Nachrichtentypen

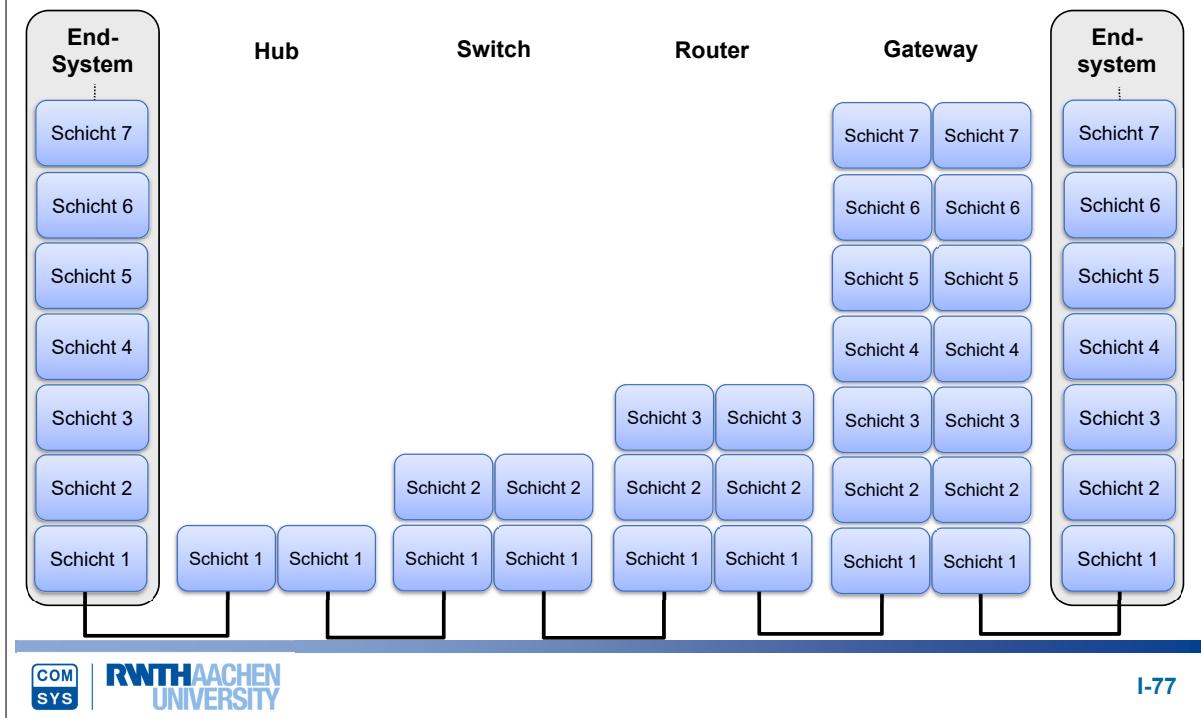
Das OSI-RM definiert verschiedene Protokolle für jede Schicht, die unterschiedliche Teile des Funktionsumfangs einer Schicht implementieren. Es hängt von einer konkreten Anwendung ab, welche Implementierungen und damit welche Funktionen genutzt werden.

OSI: Die 7 Schichten



Der Normalfall ist, dass Zwischenknoten im Netz, die Daten nur weiterleiten sollen (Router), nur die untersten drei Schichten implementieren müssen. Eine Kopplung ist allerdings auch mit mehr oder weniger Schichten möglich.

Kopplung von Kommunikationssystemen



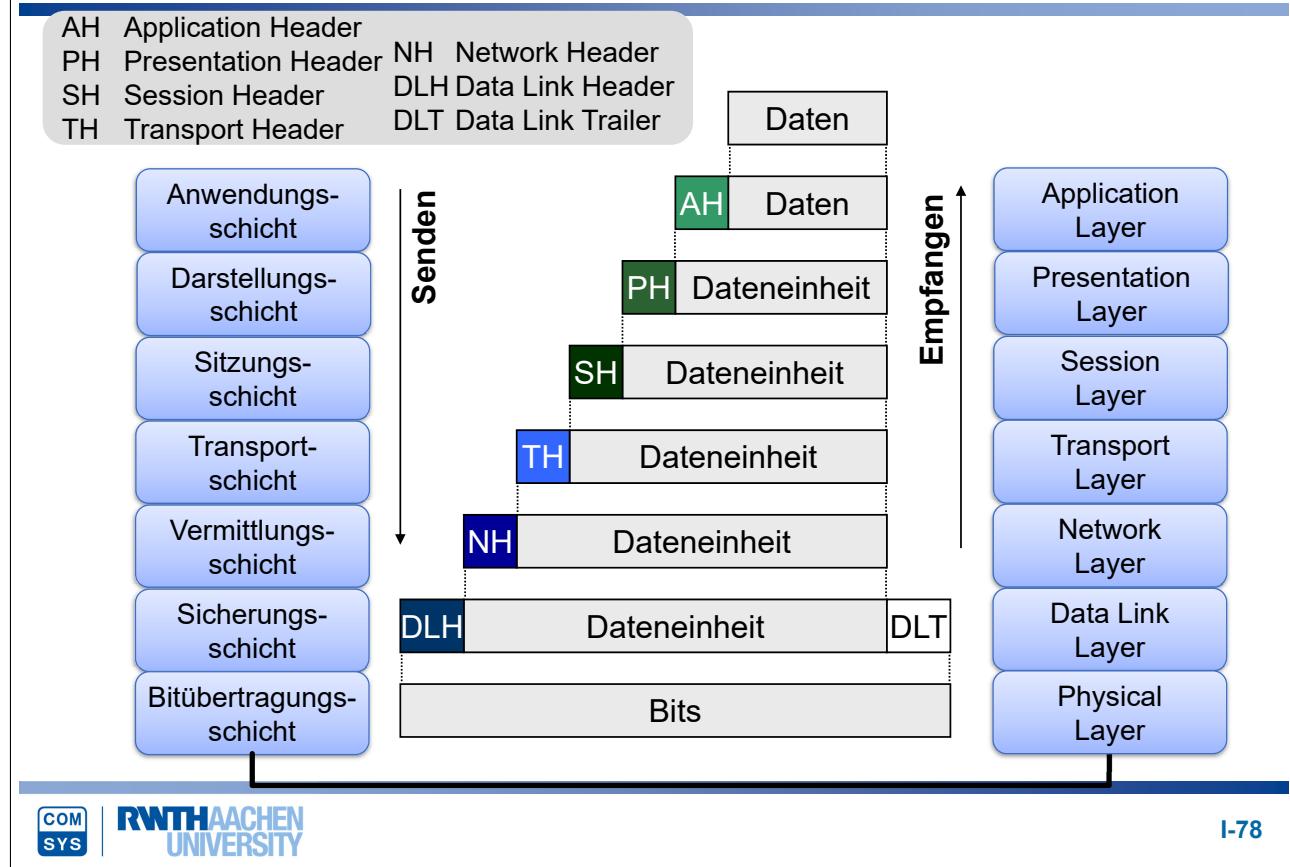
Kopplung auf Schicht 1: Repeater, nur Signalverstärkung; Verlängerung der Ausdehnung einer physikalischen Leistung

Kopplung auf Schicht 2: Switch, direkte Kopplung von Rechnern innerhalb eines lokalen Netzes; auch Bridge, Kopplung von lokalen Netzen, die unterschiedliche Schicht-2-Protokolle verwenden

Kopplung auf Schicht 3: Router, Kopplung unabhängiger Netze, auch über weite Strecken hinweg, Wegwahl innerhalb des Netzverbundes

Kopplung auf Schicht 7: Gateway, Kopplung von Systemen mit unterschiedlichen Anwendungsprotokollen. (Bitte Vorsicht: der Begriff „Gateway“ wird teilweise auch in anderer Bedeutung verwendet, z.B. in gewissem Kontext auch für Router, siehe Kapitel 4. Daher werden die hier dargestellten Gateways auch schon mal „Application Gateways“ genannt.)

Einkapselung von Daten



Wie wir gesehen haben, benutzen Instanzen die Dienste darunter liegender Instanzen. Entsprechend dem beschriebenen Zusammenspiel der Schichten werden die eigentlich zu übertragenden Daten in jeder Schicht um weitere Kontrolldaten (PCI) angereichert, die sich in einem Paketkopf (Header) oder Paketanhang (Trailer) befinden. Diese zusätzlichen Kontrolldaten müssen auf der empfangenden Seite in umgekehrter Reihenfolge wieder entfernt und ausgewertet werden, ehe die Daten an die nächsthöhere Schicht weitergereicht werden.

- **Situation**

- ▶ ISO/OSI-Basisreferenzmodell ist der „Klassiker“
 - Gut geeignet zur logischen Strukturierung
 - Oft weit weg von der Strukturierung einer Anwendung
 - Manchmal zu detailliert und komplex ... sagen zumindest Kritiker
- ▶ Tipp
 - Versuchen Sie in diesem Modell zu „denken“ – ordnen Sie immer wieder die Problemstellungen in Kommunikationssystemen sorgfältig in dieses Modell ein – das erleichtert die Beherrschung des umfangreichen Stoffs ungemein ☺

- **Modellierung vs. Implementierung**

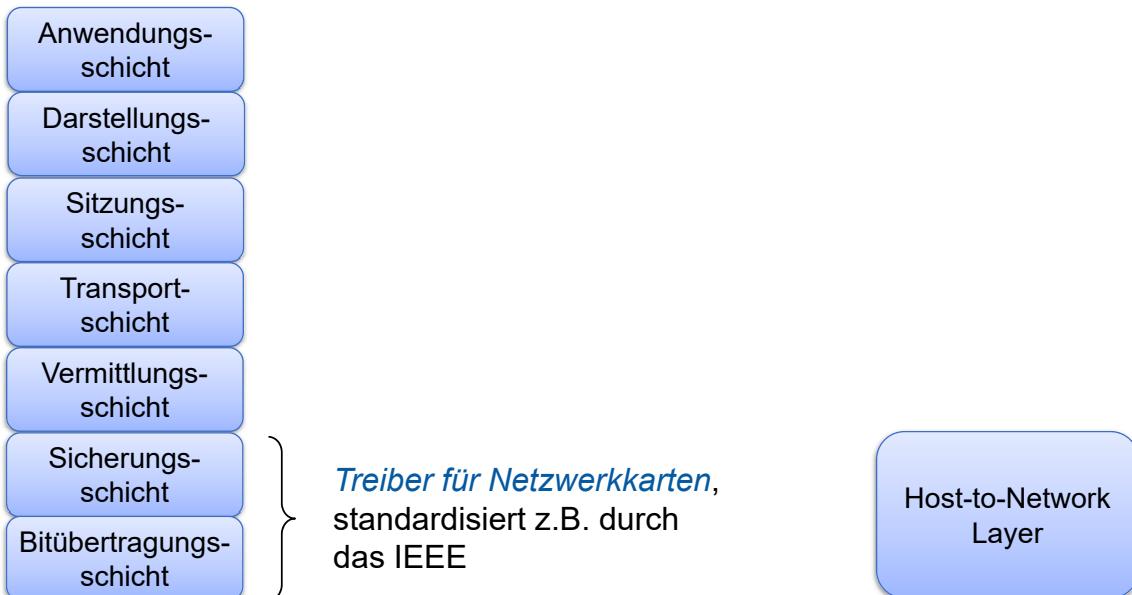
- ▶ Schichten werden in der Realität oft nicht sauber getrennt implementiert
- ▶ Daten werden zwischen Schichten möglichst nicht kopiert

Implementierungen des OSI-RM haben sich nicht durchgesetzt – aus verschiedensten Gründen: Komplexität, Implementierungsfehler, Politik, Dauer der Standardisierung. Die Terminologie der „7 Schichten“ hat sich allerdings bis heute gehalten, da die vorgenommene Strukturierung an sich durchaus sinnvoll ist. Auch bei heutigen Protokollimplementierungen hält man sich oft an diese Einteilungen.

In der Praxis findet dann aber nicht unbedingt eine saubere Implementierung statt; aus Effizienzgründen werden verschiedene Aspekte der Schichten gemischt oder aneinander angepasst.

Schichtenmodelle in der Praxis

- Orientierung am OSI-Referenzmodell, aber Reduktion des Overheads:



Ein großer Nachteil des ISO/OSI-RM ist der Overhead, der durch die Vielzahl an Schichten entsteht – zum einen wird eine große Menge an Kontrollinformationen zu den eigentlichen Nutzdaten hinzugefügt, zum anderen muss auf jeder Schicht eine Protokollinstanz eine PDU erstellen bzw. diese bei Empfang kontrollieren.

In der Praxis werden daher üblicherweise Schichten zusammengefasst, um den Umfang an Kontrollinformationen zu reduzieren und die Funktionalitäten effizienter implementieren zu können.

Die Sicherungs- und die Bitübertragungsschicht werden gemeinsam für bestimmte Netzwerke implementiert, als Treiber für Netzwerkkarten. Standardisiert werden Protokolle dieser Schichten z.B. durch das IEEE (Institute of Electrical and Electronics Engineers)

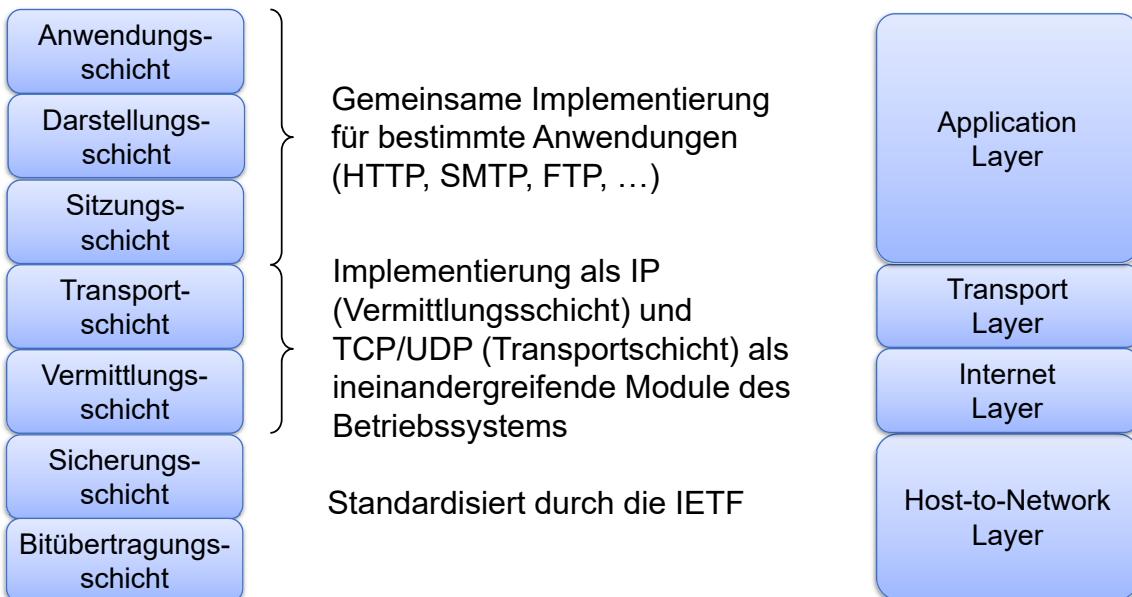
- Institute of Electrical and Electronic Engineers (IEEE)



- 802.1 Overview and Architecture of LANs
- 802.2 Logical Link Control (LLC)
- 802.3 CSMA/CD („Ethernet“)
- 802.4 Token Bus
- 802.5 Token Ring
- 802.6 DQDB (Distributed Queue Dual Bus)
- 802.7 Broadband Technical Advisory Group (TAG)
- 802.8 Fiber Optic TAG
- 802.9 Integrated Services LAN (ISLAN) Interface
- 802.10 Standard for Interoperable LAN Security (SILS)
- 802.11 Wireless LAN (WLAN)
- 802.12 Demand Priority (HP's AnyLAN)
- 802.14 Cable modems
- 802.15 Personal Area Networks (Bluetooth)
- 802.16 WirelessMAN (WiMAX)
- 802.17 Resilient Packet Ring
- 802.18 Radio Regulatory TAG
- 802.19 Coexistence TAG
- 802.20 Mobile Broadband Wireless Access (MBWA)
- 802.21 Media Independent Handover
- 802.22 Wireless Regional Area Networks (WRAN)
- 802.23 Emergency Services
- 802.24 Vertical Applications TAG

Schichtenmodell (TCP-IP-Modell) des Internets

- Orientierung am OSI-Referenzmodell, aber Reduktion des Overheads:



Während in der Praxis eine Vielzahl an Protokollen der Schicht 1/2 existiert, standardisiert die Internet Engineering Task Force die Protokolle ab Schicht 3, um eine Kopplung aller möglichen Netze mit beliebigen Schicht-1/2-Protokollen zu ermöglichen und um die Nutzung wichtiger Anwendungen netzweit einheitlich zu gestalten.

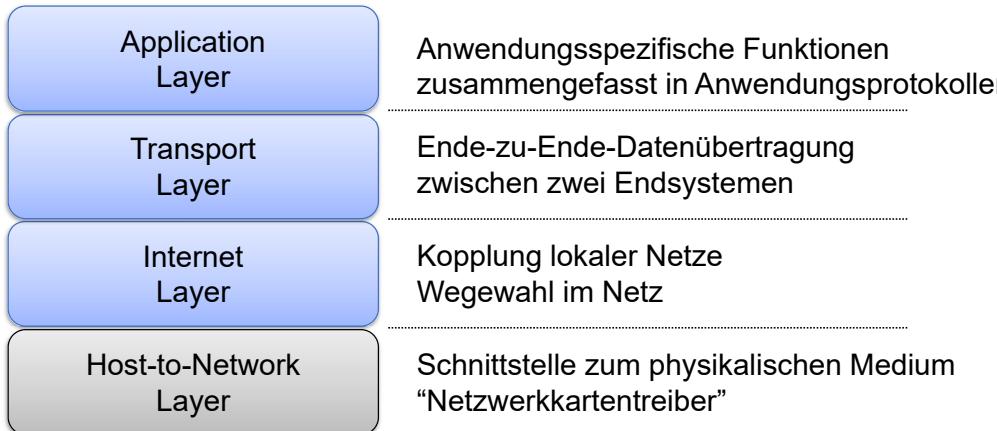
Standardisierungsgremien

• Internet Engineering Task Force (IETF)

- ▶ Forum für die technische Koordination der Arbeiten zum ARPANET, dem Vorläufer des Internets (seit 1986)
- ▶ Entwicklung zur großen, offenen und internationalen Gemeinschaft von Administratoren, Herstellern und Forschern
- ▶ Beschäftigt sich mit der Evolution der Internet-Architektur und der reibungslosen Operation des Internets
- ▶ Verschiedene Arbeitsgruppen zu einzelnen Aspekten des Internets
- ▶ COMSYS ist auch aktiv in der IETF:
RFCs 3662, 3754, 6069, 6253, ...



Die Internet-Protokollhierarchie: TCP/IP-RM



Durchgesetzt in der Praxis hat sich das TCP/IP-Referenzmodell (oder auch: Internet-Referenzmodell), das durch die IETF standardisiert wird. Es ist schlanker als das OSI-RM und damit effizienter einzusetzen... bietet allerdings auch weniger Wahlmöglichkeiten, welche Funktionen auf einer Schicht implementiert werden sollten.

Der Host-to-Network Layer wird von der IETF nicht betrachtet. Es wird lediglich verlangt, dass eine Schnittstelle zur Übertragung von IP-Paketen bereitgestellt wird. Die Spezifizierung der untersten Schicht wird anderen Gremien überlassen, z.B. der IEEE.

Die im Zusammenhang mit dem Internet entwickelten Protokolle werden als TCP/IP-Protokolle bezeichnet und umfassen neben dem Transportprotokoll TCP (Transmission Control Protocol) und dem Vermittlungsprotokoll IP (Internet Protocol) unter anderem auch Protokolle der Anwendungsschicht (z.B. File Transfer Protocol FTP oder HyperText Transfer Protocol HTTP).

Das TCP/IP-Modell ist als Modell weniger relevant. An dieser Stelle hat die ISO die wesentlichen Grundlagen erarbeitet, die z.T. auch Eingang in das TCP/IP-RM gefunden haben. Interessant im Zusammenhang mit TCP/IP sind vielmehr die konkreten Protokolle, die das Modell auffüllen. Hier besteht ein genau umgekehrtes Verhältnis zu der ISO, deren Protokolldefinitionen zu keinem Zeitpunkt die hohe Akzeptanz der TCP/IP-Protokolle erreicht hat. Ein kurzes Fazit lautet also:

Während die ISO die modelltechnischen Grundlagen für Kommunikationssysteme lieferte, resultierten aus den Internet-Aktivitäten allgemein akzeptierte Protokollstandards in Form der TCP/IP-Protokollfamilie.

Fazit

- **Datenkommunikation:**

- ▶ Sammlung aufeinander aufbauender Dienstinstanzen und Protokolle, die gemeinsam die gesamte Funktionalität eines Datenaustauschs definieren
 - ISO/OSI-Referenzmodell als Versuch, solche Protokolle zu standardisieren; die Terminologie des Modells wird weiter verwendet
 - ISO/OSI-Schichten 1 und 2 werden als *Netzwerkkarten (und deren Treiber)* implementiert. Innerhalb eines lokalen Netzes können Geräte mittels dieser Funktionalität kommunizieren.
 - TCP/IP-Referenzmodell baut auf den Netzwerkkarten(treibern) auf
 - IP und TCP/UDP sind als *Teil des Betriebssystems* implementiert
 - Über Anwendungen werden *Dienste* auf dem System realisiert (Web, Email, Zeitsynchronisation). Die Anwendungen koordinieren sich untereinander über Anwendungsprotokolle (HTTP, SMTP, NTP, ...), die wiederum die Dienste von TCP, UDP nutzen. Diese nutzen den Dienst von IP.

Lessons learned

- **Wichtige Begriffe/Konzepte des Kapitels**

- ▶ Kommunikationsdienst vs. –protokoll
 - Dienst beschreibt das „was“, Protokoll das „wie“
- ▶ Bestätigte und unbestätigte Dienste
 - Bitte deutlich unterscheiden von Bestätigungen im Rahmen von Protokollen!
- ▶ Verbindungslose und –orientierte Dienste
 - Verbindungsorientierte Dienste bauen einen Kontext auf, verbindungslose Dienste versenden alle PDUs unabhängig voneinander
- ▶ Schichten
 - Implementierung von Kommunikationssystemen als geschichtete Sammlung von Protokollen, die sich auf bestimmte Aufgaben spezialisieren
- ▶ Referenzmodelle
 - ISO/OSI-RM als theoretisches Modell, TCP/IP-RM in der Praxis