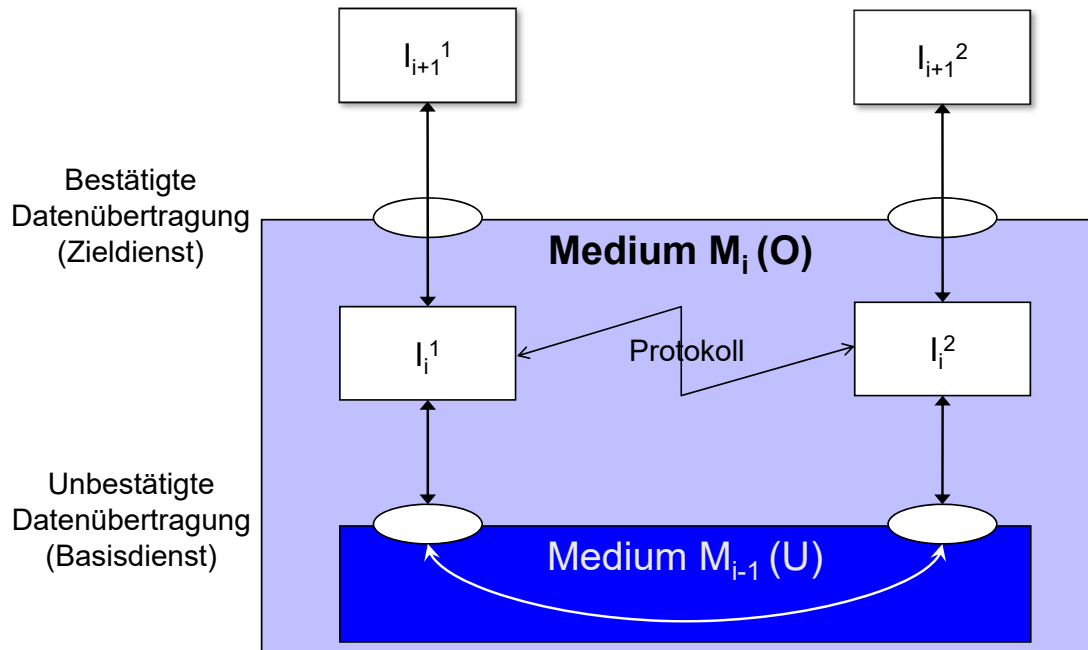


- **Beispiel: Alternating Bit Protocol**
 - ▶ Dienst vs. Protokoll
 - ▶ Interaktion zwischen Schichten

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 Dienstbringers 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.

- **Dienst zur bestätigten Datenübertragung**

- ▶ Auf Basis eines Dienstes zur unbestätigten Datenübertragung
- ▶ Nötig: *Quittungen*
 - Mit alternierenden Quittungsnummern 0 und 1
 - Nachdem eine quitierte 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: unbestätigte und unzuverlässige Datenübertragung
- i-Dienst: bestätigte 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.

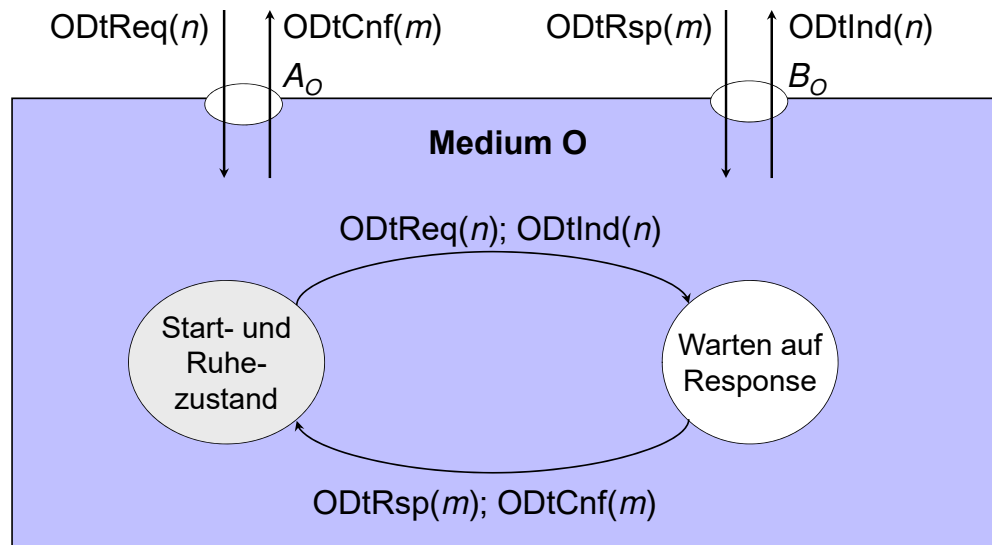
Protokollfunktionalität: Quittierung und Fehlerbehandlung

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 – erster Protokollautomat

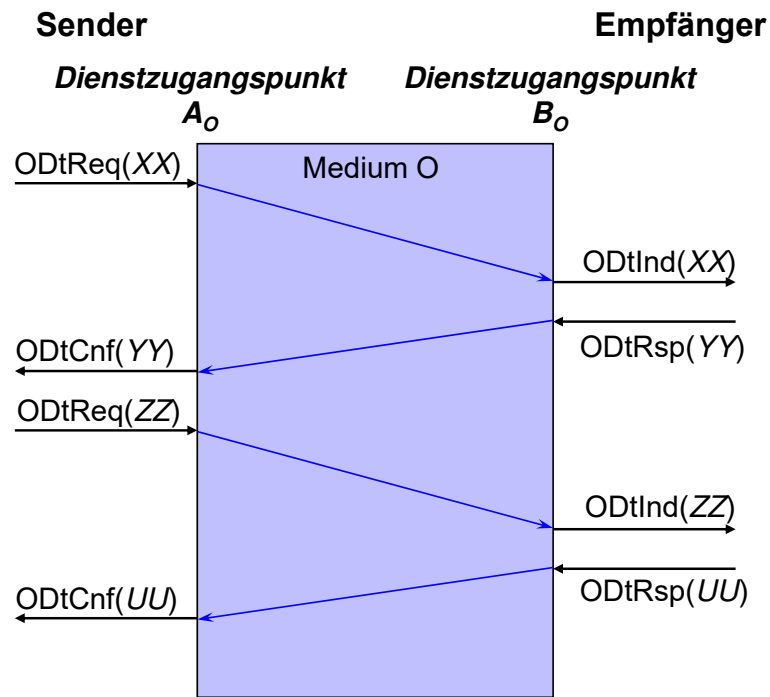


Wie das Diagramm (sehr grober Protokollautomat) für den zu realisierenden Dienst O aus dem Beispiel zeigt, handelt es sich hier um einen einfachen Dienst mit nur einer Dienstfunktion ODt , die einen bestätigten Datentransfer realisiert.

Aus der Sicht des Dienstinutzers 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.)

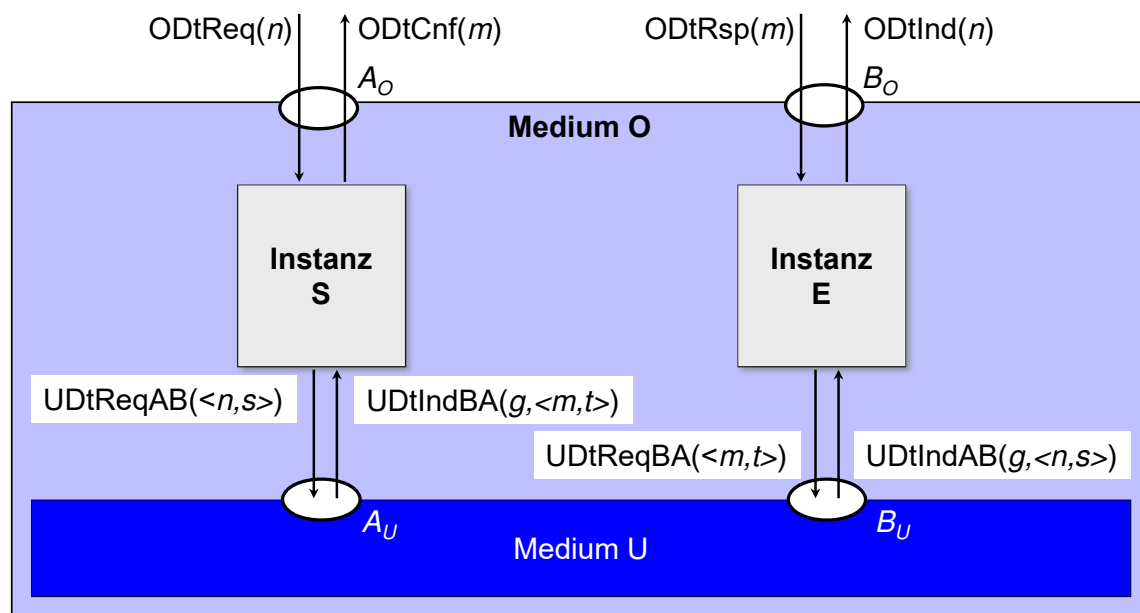
Die Initiierung der bestätigten Datenübertragung erfolgt über den Zugangspunkt A_O – dieser beschreibt die Rolle des sendenden Dienstinutzers. Zugangspunkt B_O beschreibt dementsprechend die Rolle des empfangenden Dienstinutzers. Über B_O werden zwar auch Dateneinheiten übertragen, die aber lediglich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



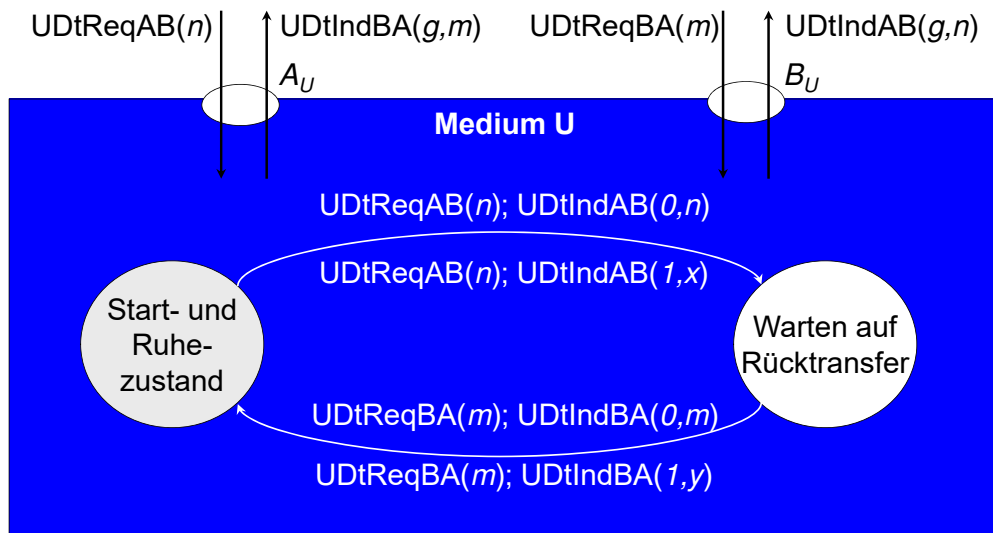
Die Nutzdaten n werden zur bestätigten Übertragung durch ODtReq am SAP A_O bereitgestellt. Medium O realisiert nun das Alternating Bit Protocol. Die Protokollinstanz S(ender) benutzt den unbestätigten Übertragungsdienst am unteren Zugangspunkt A_U und überträgt neben den Nutzdaten n außerdem eine Sequenznummer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



- **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(mpfänger), die das Alternating Bit Protocol realisieren), ist diese Dienstschnittstelle also symmetrisch.

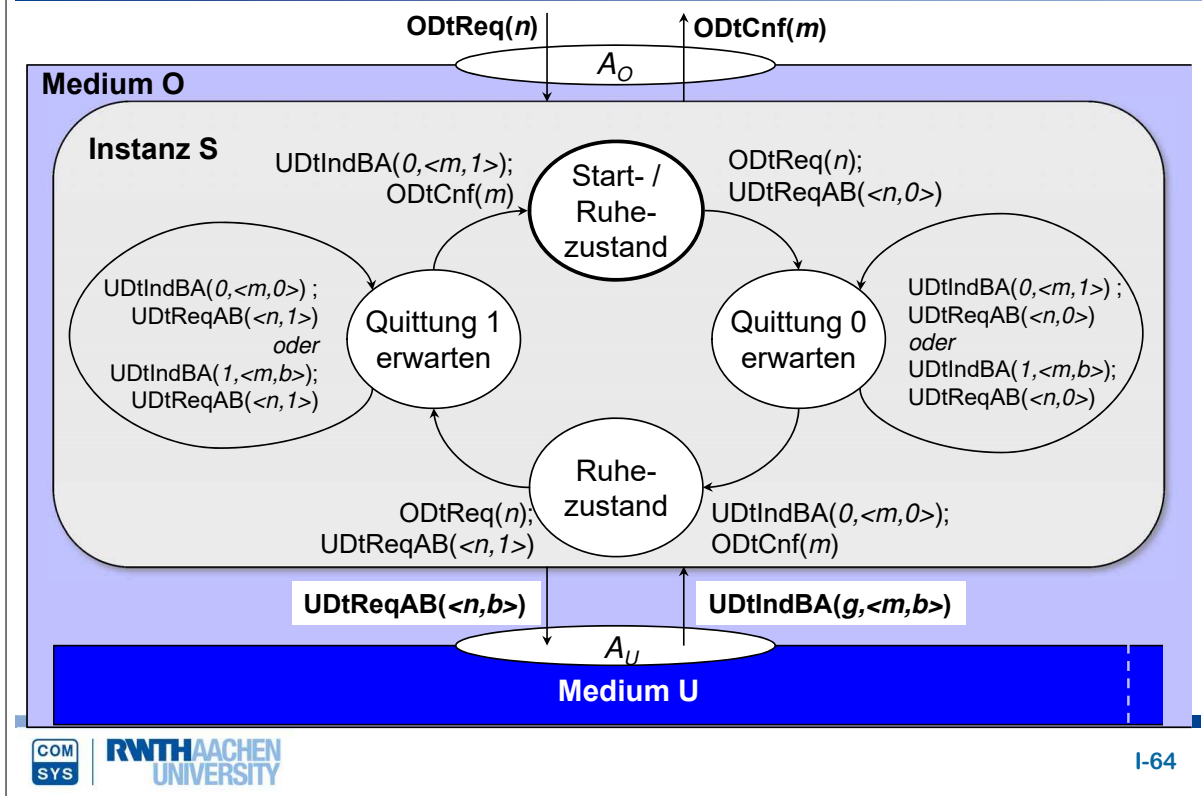
Betrachtet man jetzt aber das Zustandsübergangsdiagramm, erkennt man recht deutlich, dass zunächst eine Übertragung in der Richtung $A \rightarrow B$ erfolgen muss und daraufhin eine Übertragung in die entgegengesetzte Richtung erfolgt.

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 Dienstonutzer 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.

Protokollinstanz S als endlicher Automat



Schnittstellen des Protokollautomaten für S:

- $ODtReq(n)$: Anforderung der Dienstleistung „bestätigte Datenübertragung der Nutzdaten n“
- $UDtReqAB(\langle n, 0 \rangle)$: Anforderung der Dienstleistung „unbestätigte Datenübertragung von einem Dienstzugangspunkt A_U nach B_U “ mit Nutzdaten n und Sequenznummer 0
- $UDtIndBA(g, \langle m, b \rangle)$: 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)

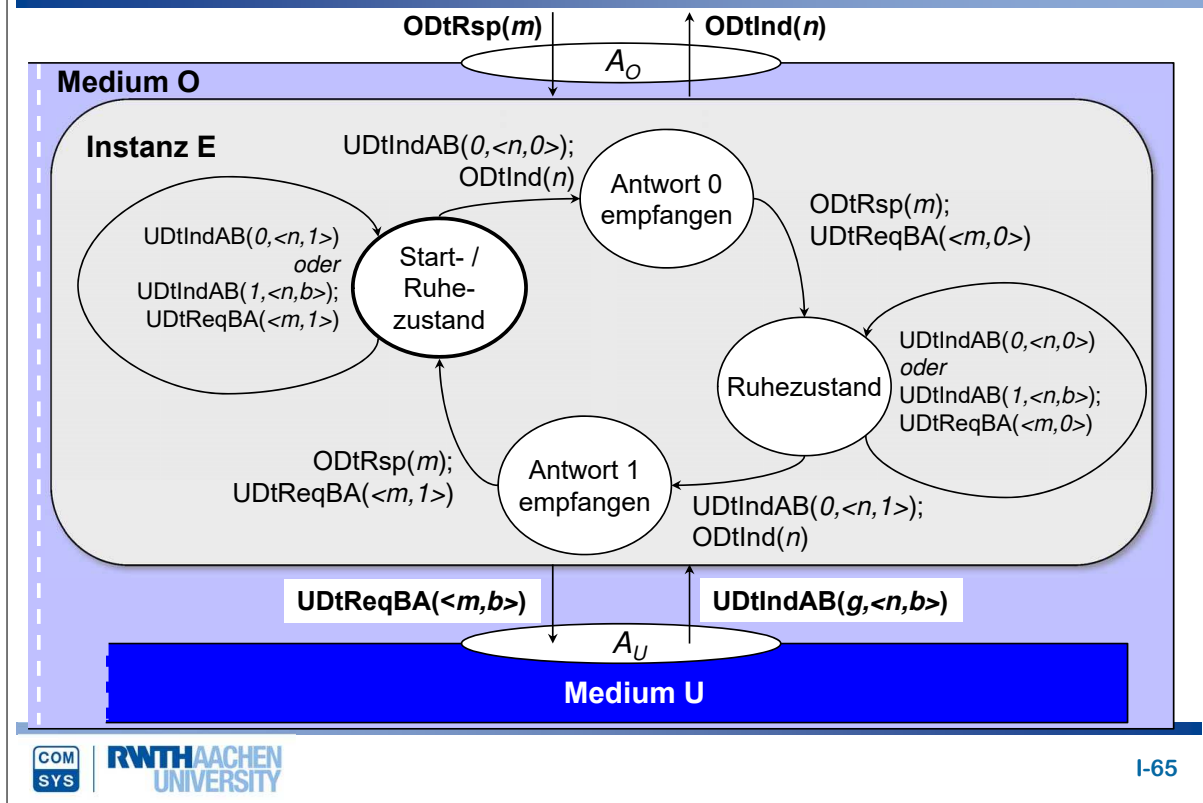
Aus diesen Beschreibungen lässt sich die Bedeutung der Dienstprimitive ableiten, die die Ausgaben des Automaten bilden: $ODtCnf(m)$ ist die Bestätigung der Übertragung von Nutzdaten m, $UDtReqAB(\langle n, b \rangle)$ ist die unbestätigte Übertragung der Nutzdaten n und der Sequenznummer b.

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 $ODtReq(n)$ (eine Anfrage vom oberen Dienstzugangspunkt, Nutzdaten n bestätigt zu übertragen). Hierauf erfolgt die Ausgabe $UDtReqAB(\langle n, 0 \rangle)$, 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 $UDtInd$ 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

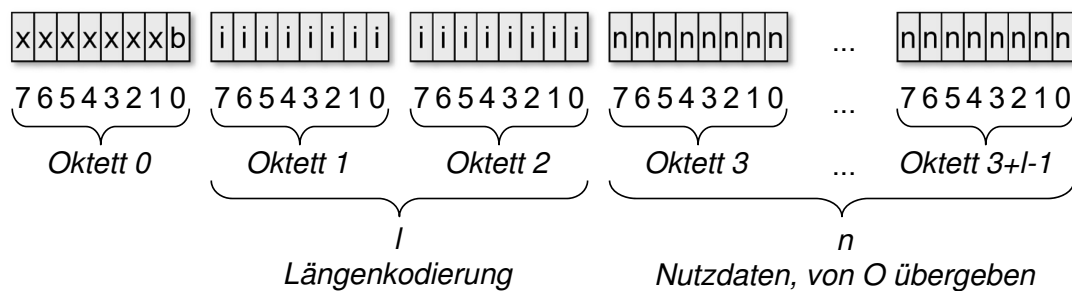


Die Auswertung des Verfälschungsparameters erfolgt je nachdem, wo man sich im alternierenden Ablauf gerade befindet, im “Start- und Ruhezustand” oder im “Ruhezustand” (selbiges für Instanz S in den anderen Zuständen). Die Reaktion der Instanz E auf eine Verfälschungsanzeige (also Wert = 1) besteht in 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



► Oktett 0:

- Kodierung der Sequenznummer b im niederwertigsten Bit

► Oktett 1+2:

- Kodierung der Länge l der Nutzdaten n

► Folgeoktette:

- Nutzdaten n

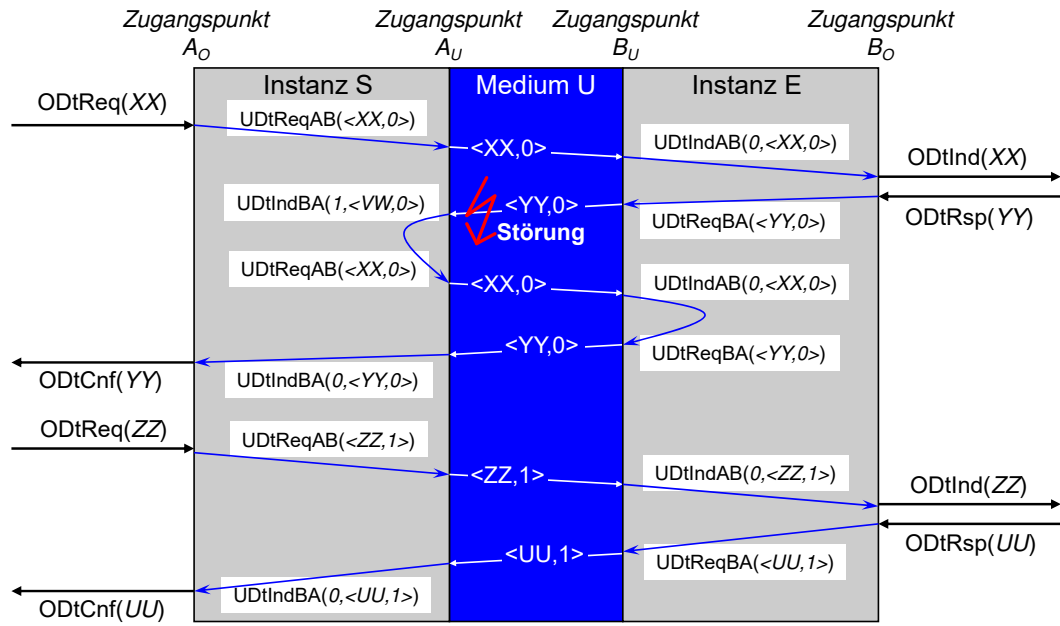
Die Protokolldateneinheiten unterscheiden sich von den Dienstprimitiven in einem ganz elementaren Punkt: während Dienstprimitive in einer Protokollnorm nur in ihrer Semantik beschrieben werden, muss zu jeder im Protokoll benutzten PDU auch der genaue syntaktische Aufbau festgeschrieben werden.

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 Oktetten die zu kodierenden Protokollelemente enthalten sind; Entscheidung hier: Sequenznummer b im ersten Oktett; Länge der Nutzdaten in den zwei folgenden Oktetten; Nutzdaten in den übrigen Oktetten.

Vollständiger Beispielablauf



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

- **Einführung und Begriffe**
 - ▶ Was ist Datenkommunikation?
 - ▶ Information, Daten, Signale
 - ▶ Netze
- **Allgemeine Grundlagen**
 - ▶ Dienste
 - ▶ Protokolle und Schichten
 - ▶ **Kommunikationsarchitekturen**

- **Architektur von Kommunikationssystemen**
 - ▶ Offen
 - ▶ Geschlossen (proprietär)
- **Relevante Architekturen**
 - ▶ ISO/OSI
 - ▶ IBM SNA
 - ▶ SPX/IPX (Novell Netware)
 - ▶ ISDN
 - ▶ ATM/B-ISDN
 - ▶ 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.

- 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!



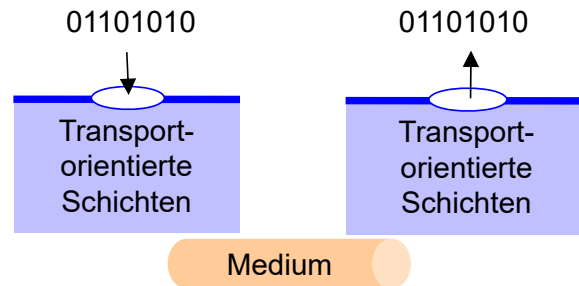
www.iso.ch

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.

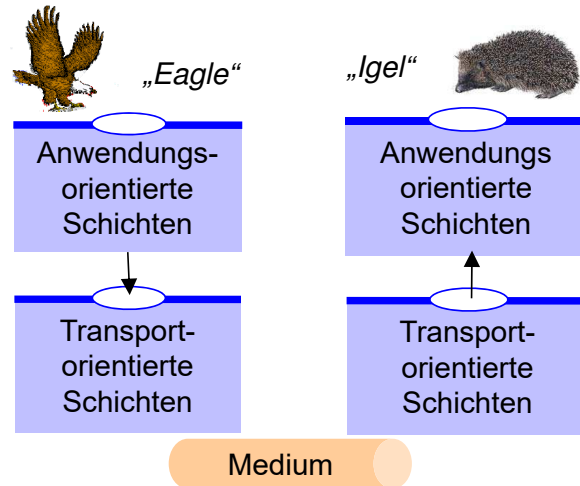
Die transportorientierten Schichten dienen, wie der Name schon sagt, ausschließlich zum Transport der Daten. Weitere Aspekte, die z.B.

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

- **Anwendungsorientierte Schichten**

► Anwendungsbezogene Aspekte der Datenübertragung

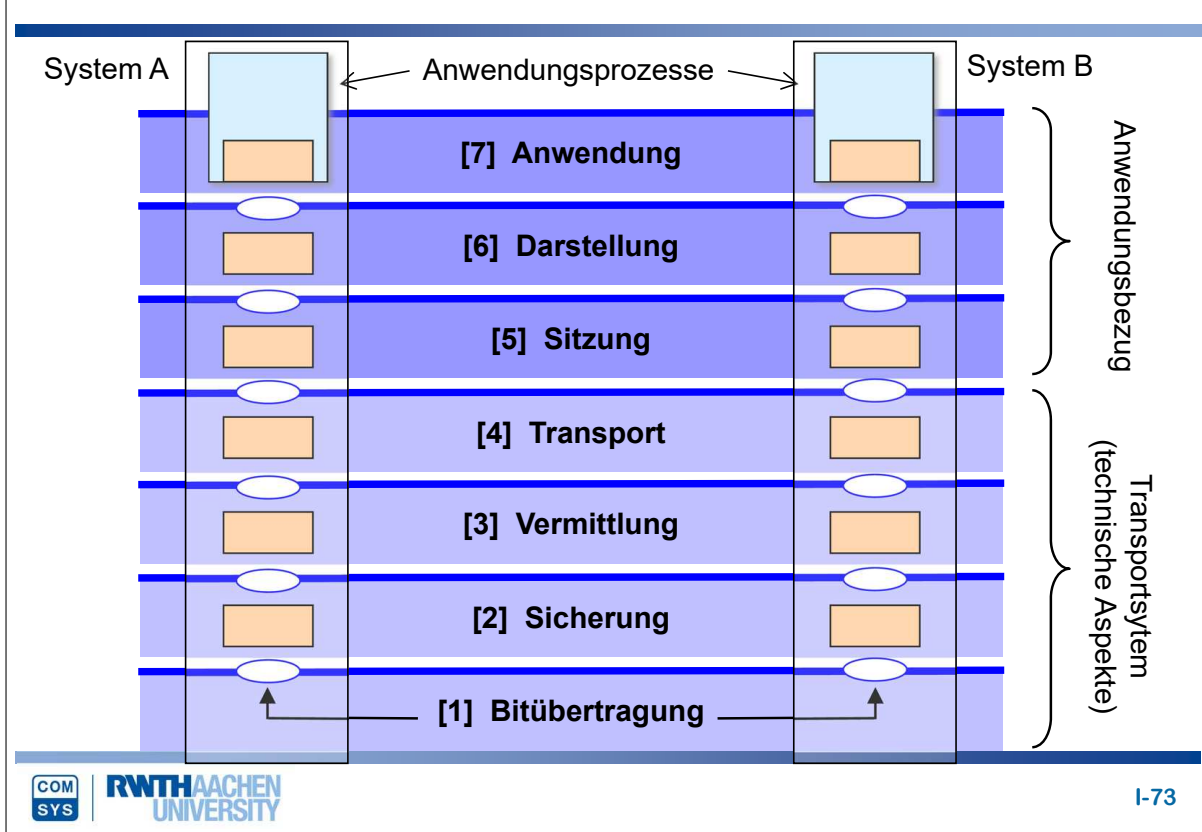
- Semantik der Daten ist wichtig
- Kooperation der Teilnehmer unter formalen Gesichtspunkten (Strukturierung, Präsentation), z.B.
 - Steuerung des Kommunikationsablaufs
 - Kompensation von Fehlverhalten durch verteilte Transaktionen



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



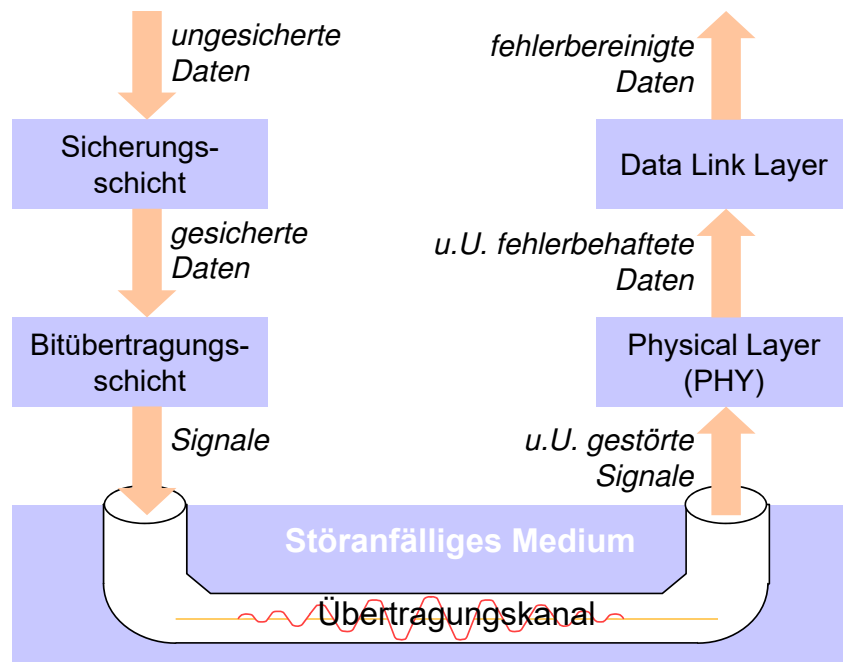
Beim OSI-RM steht der *Modellierungsaspekt* im Mittelpunkt und weniger die durch OSI genormten Protokolle.

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).

Bitte beachten: im Folgenden werden die Aufgaben der Schichten beschrieben – dies ist aber lediglich die Gesamtmenge der möglichen Funktionalitäten. Das OSI-RM definiert verschiedene Protokolle für jede Schicht, die unterschiedliche Teile des Funktionsumfangs implementieren. Es hängt von einer konkreten Anwendung ab, welche Implementierungen und damit welche Funktionen genutzt werden.

Bitübertragungsschicht und Sicherungsschicht



Die beiden untersten Schichten sorgen gemeinsam für eine fehlerfreie Datenübertragung zwischen benachbarten Rechnern. „Benachbart“ heißt, dass die kommunizierenden Rechner durch ein physikalisches Medium direkt miteinander verbunden sind.

Die Bitübertragungsschicht bietet als Medium nachrichtentechnische Kanäle, die einen ungesicherten, längenbeschränkten, ungepufferten Nachrichtenaustausch zwischen einer beschränkten Anzahl von angeschlossenen Einrichtungen unterstützen:

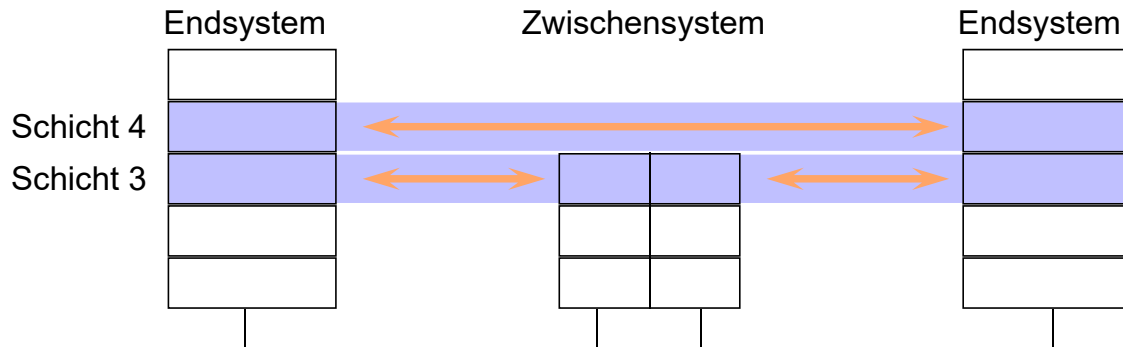
- ungesicherte Verbindung zwischen Systemen (d.h. es können Fehler bei der Übertragung auftreten)
- Übertragung unstrukturierter Bitfolgen über das physikalische Medium
- Normung der physikalischen Schnittstellen zwischen Rechner und Medium
- Umsetzung von Daten in Signale

Die Sicherungsschicht erweitert einen nachrichtentechnischen Kanal zum eigenständigen, abstrakten Medium „gesicherter Kanal“. „Gesichert“ heißt hierbei: fehlerfrei:

- Zerlegung des Bitstromes in Rahmen (Frames) als Basiseinheit der Übertragung
- Fehlererkennung und -behandlung (Quittungen und Übertragungswiederholungen, siehe Alternating Bit Protocol)
- Flusskontrolle (Vermeidung der Überlastung/Pufferüberlauf des Empfängers)
- Regelung des Zugriffs auf ein Medium, welches gemeinsam von mehreren Stationen verwendet wird.

Vermittlungsschicht und Transportschicht

- **Schicht 3: Vermittlungsschicht (Network Layer)**
- **Schicht 4: Transportschicht (Transport Layer)**



Die beiden nächsten Schichten verknüpfen kommunizierende Anwendungen auf unterschiedlichen Rechnern, auch über Netzgrenzen hinweg.

Die Vermittlungsschicht verbindet die gesicherten Kanäle der Sicherungsschicht zwischen beliebigen Knoten eines größeren Rechnernetzes – sie verbindet isolierte lokale Netze, die auch unterschiedliche Protokolle einsetzen können, zu größeren Netzen. Dazu ist es nötig, Daten über Zwischenstationen (Router) hin zum Zielnetz weiterzuleiten. Diese Zwischenstationen agieren als Vermittlungsstellen, wodurch der Name Vermittlungsschicht stammt. Die Vermittlungsschicht wird häufig auch als Netzwerkschicht bezeichnet. Aufgaben dieser Schicht sind:

- Verknüpfung von Teilstreckenverbindungen zu Endsystemverbindungen über Zwischenknoten hinweg
- Adressierung von Endsystemen (Rechnern) und Anwendungen auf den Endsystemen
- Wegwahl (Routing) zur Ermittlung des besten Pfades zur Weiterleitung der Daten; dabei auch Staukontrolle (Vermeidung der Überlastung einzelner Leitungen im Netz)
- Bereitstellung einer bestimmten Übertragungsqualität (z.B. zuverlässige Übertragung)

Die Transportschicht schließlich ermöglicht es, von unterschiedlichen Vermittlungsschichtdiensten zu abstrahieren, so dass sämtliche die eigentliche Datenübertragung betreffenden Aspekte verborgen werden. Es kann passieren, dass einige Fehler auf tieferen Schichten nicht erkannt oder behoben werden können – dies können schwerwiegende Probleme sein, aber üblicherweise entsteht diese Situation, wenn im konkreten Protokoll-Stack Protokolle ausgewählt wurden, die nicht alle Funktionalitäten anbieten. Es kann beispielsweise sein, dass auf der Vermittlungsschicht ein Datagramm-Dienst eingesetzt wird. Dieser behandelt keine Fehler, so dass nicht garantiert ist, dass eine PDU überhaupt auf Empfängerseite ankommt. Solche Fehler können dann auf der Transportschicht erkannt und behoben werden (Siehe Alternating Bit Protocol als vereinfachtes Beispiel). Generelle Aufgaben der Transportschicht sind:

- Adressierung von Transportdienstbenutzern
- Datentransfer zwischen Benutzern in Endsystemen
- Bietet Transparenz bzgl. Übertragungsweg und Ende-zu-Ende-Verbindung
- Kann verbindungslos oder verbindungsorientiert arbeiten

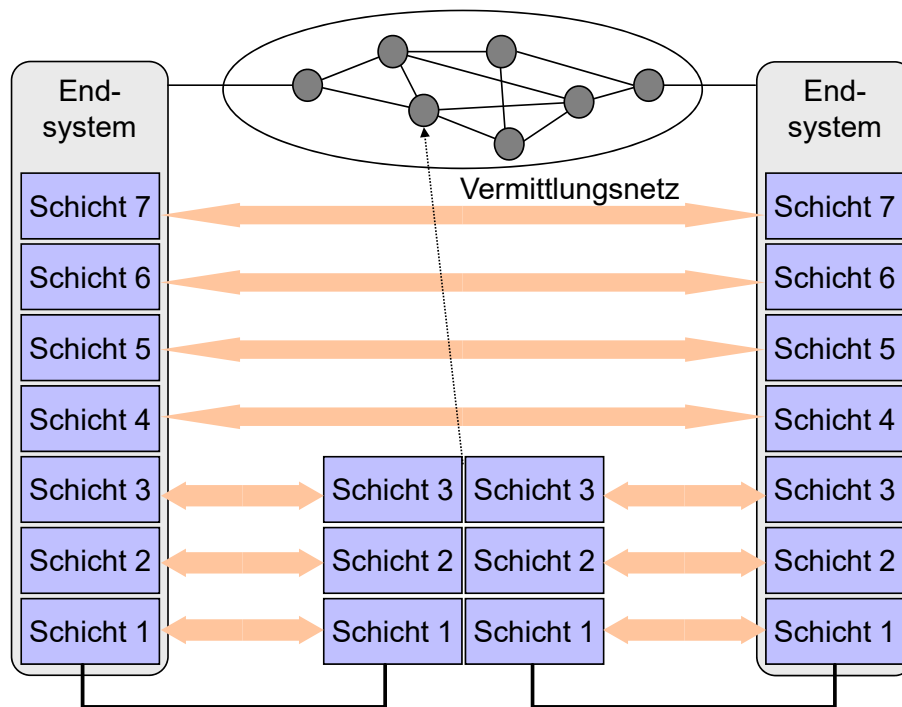
- **Sitzungsschicht (Session Layer, Schicht 5)**
 - ▶ Steuerung des Nachrichtenaustauschs zweier Kommunikationspartner:
 - Rechteübergabe
 - Strukturierung der Übertragung, Ablaufsteuerung/-koordinierung, Verwaltung von Dialogen
- **Darstellungsschicht (Presentation Layer, Schicht 6):**
 - ▶ Kommunikation wird trotz unterschiedlicher lokaler Datenformate der Teilnehmer ermöglicht
 - ▶ Nachrichten setzen sich aus typisierten Daten zusammen
- **Anwendungsschicht (Application Layer, Schicht 7):**
 - ▶ Austausch von Nachrichten anwendungsabhängigen Aufbaus und Zwecks
 - Stellt Dienste für bestimmte Anwendungsarten zur Verfügung, z.B. Dateitransfer, entfernter Prozeduraufruf, E-Mail

In den anwendungsorientierten Schichten hingegen wird eng Bezug zu den Anwendungen genommen. Die Sitzungsschicht (auch: Kommunikationssteuerungsschicht) ermöglicht die Nichtunterbrechbarkeit von Kommunikationsbeziehungen, die Gliederung des Nachrichtenaustausches nach Anwendungsgesichtspunkten und somit eine strengere Ablauforganisation des Nachrichtenaustausches.

Die Darstellungsschicht sorgt dafür, dass eine Kommunikation auch zwischen Einrichtungen möglich ist, die eine unterschiedliche lokale Darstellung der Daten aufweisen, wie beispielsweise die Speicherung eines Integer-Wertes in 16 oder 32 Bit.

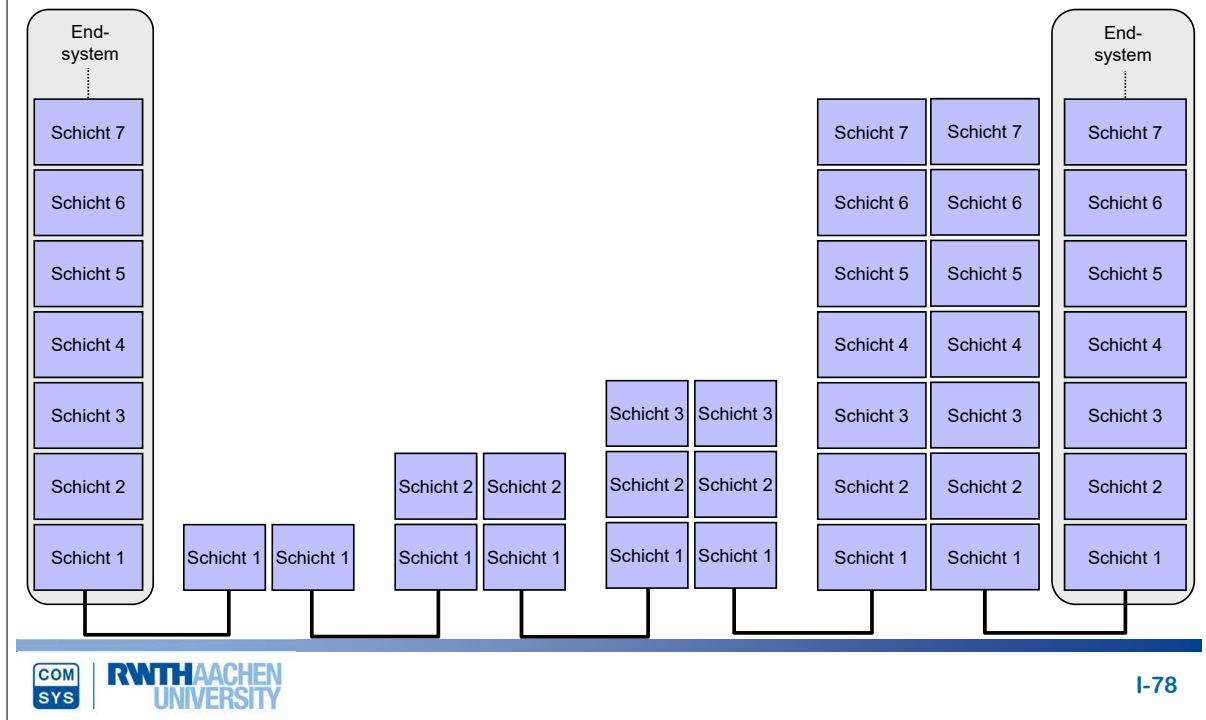
Die Anwendungsschicht ist dann die Basis für Anwendungsprozesse, so dass hier anwendungsspezifische Dienste realisiert werden. Sie stellt Dienste zur Verfügung, z.B. für Dateitransfer, Zustellung von E-Mail, Remote Procedure Call, etc.

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

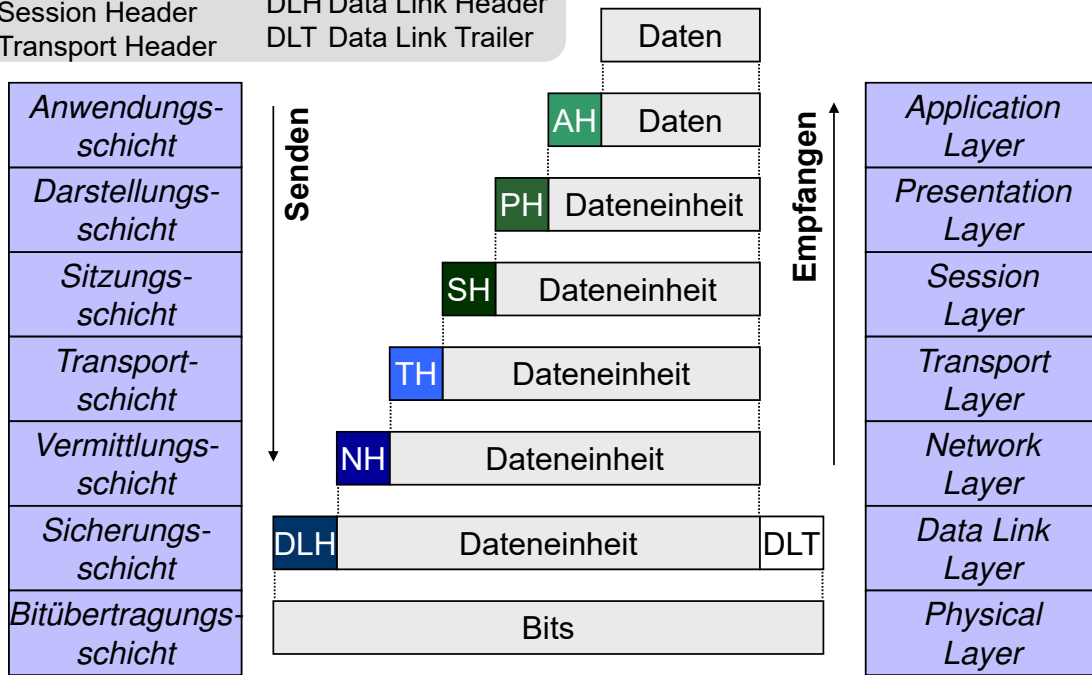
Kopplung auf Schicht 2: Switch, direkte Kopplung von Rechnern innerhalb eines lokalen Netzes

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

Einkapselung von Daten

AH	Application Header	NH	Network Header
PH	Presentation Header	DLH	Data Link Header
SH	Session Header	DLT	Data Link Trailer
TH	Transport Header		



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 😊

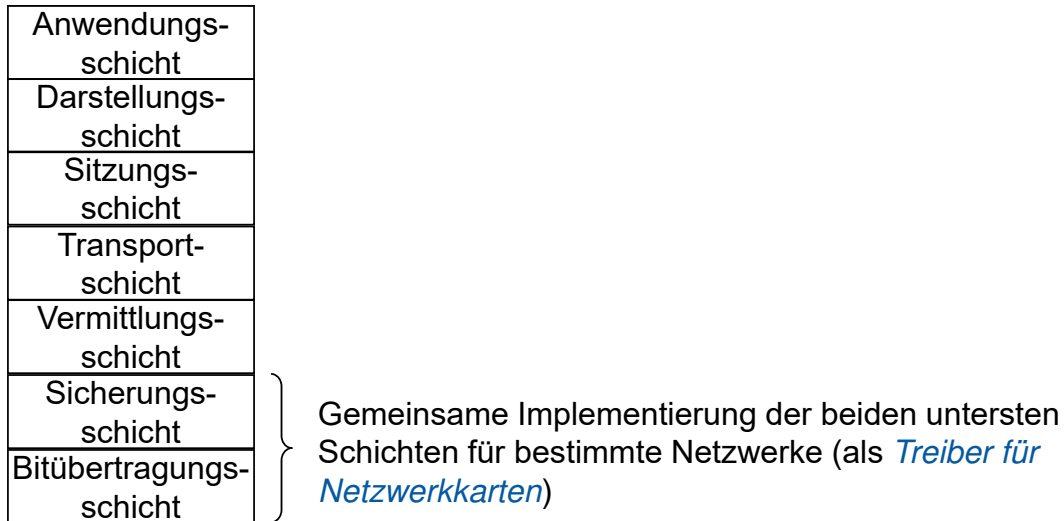
- **Modellierung vs. Implementierung**

- ▶ 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.

- 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.

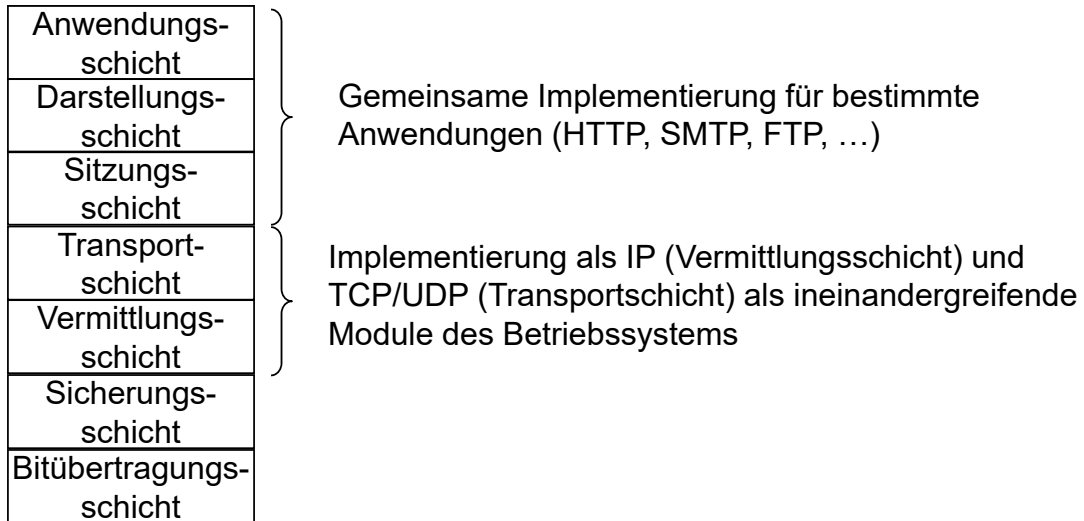
In der Praxis werden daher üblicherweise Schichten zusammengefasst, um den Umfang an Kontrollinformationen zu reduzieren und die Funktionalitäten effizienter implementieren zu können.

- **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)

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

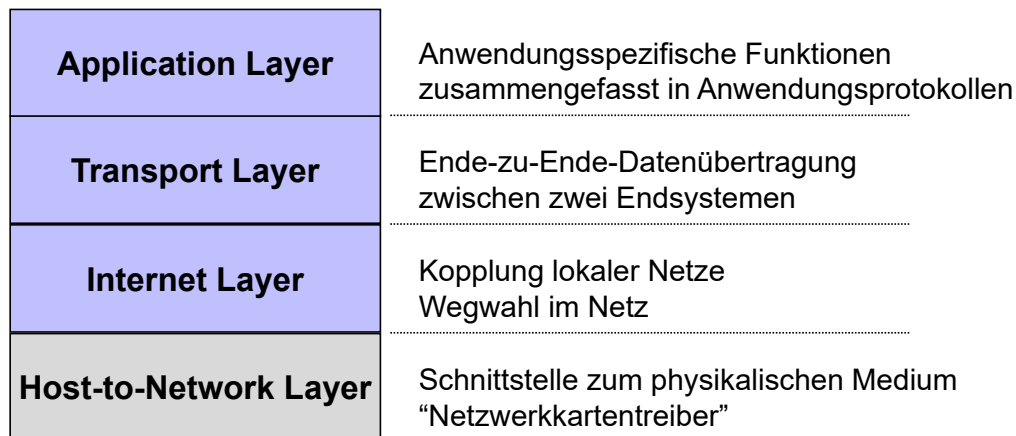


- **Internet Engineering Task Force (IETF)**

- ▶ 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
- ▶ Standardisierungsentwürfe können erst dann zum Standard werden, wenn eine funktionierende Implementierung an zwei unabhängigen Standorten mindestens 4 Monate lang getestet wurde
- ▶ Ergebnis dieses Standardisierungsprozesses: durchschlagender Erfolg der Internetprotokolle TCP/IP
- ▶ 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.

- **Datenkommunikation:**

- ▶ Sammlung aufeinander aufbauender Dienstanstzen 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 *Treiber für Netzwerkkarten* implementiert. Innerhalb eines lokalen Netzes können mittels dieser Funktionalität Rechner kommunizieren
 - TCP/IP-Referenzmodell baut auf den Netzwerkkartentreibern auf
 - IP und TCP/UDP sind als *Teil des Betriebssystems* implementiert
 - Anwendungsprotokolle können als *zusätzliche Dienste* auf dem System laufen