

HANSER

# Lehr- und Übungsbuch Telematik

Gerhard Krüger, Dietrich Reschke

Netze - Dienste - Protokolle

ISBN 3-446-22862-4

Leseprobe

Weitere Informationen oder Bestellungen unter  
<http://www.hanser.de/3-446-22862-4> sowie im Buchhandel

# 1 Einführung – der Begriff Telematik

**Dietrich Reschke**

Bollmann /1.1/ schreibt: „Flussers /1.2/ Gebrauch des Begriffs **Telematik** (zusammengesetzt aus **Tele**kommunikation und **Informatik**) geht auf dessen Einführung durch Simon Nora und Alain Minc zurück, die in ihrem 1978 publizierten Bericht an den französischen Präsidenten *L'informatisation de la Societe (Die Informatisierung der Gesellschaft*, dt. 1979 /1.3/ ) schreiben:

„Die wachsende Verflechtung von Rechnern und Telekommunikationsmitteln, die wir Telematik nennen, eröffnet einen völlig neuen Horizont. Schon immer haben Kommunikationsmittel die Gemeinwesen strukturiert... Die Telematik bewegt – im Gegensatz zur Elektrizität – nicht einen trägen Strom, sondern Information, das heißt Macht. (...). Die Telematik wird nicht nur ein weiteres Netz darstellen, sondern vielmehr ein Netz neuer Art, das Bild, Ton und Informationsinhalte in eine vielschichtige Wechselbeziehung treten läßt. Sie wird unser Kulturmodell verändern.“ In der Konsequenz des von Nora/Minc-Berichts kam es in Frankreich zum *Programme Télématique*, dessen sichtbare Auswirkung u.a. in der frühzeitigen erfolgreichen Durchsetzung von *Minitel*, dem französischen Pendant zum deutschen *BTX*, besteht.“ Eine spätere amerikanische Sicht auf die Telematik findet sich in /1.4/.

Wer heute im Internet unter dem Begriff Telematik sucht, der findet weit über 100.000 Einträge. Das europäische Forschungsprogramm umfasst 13 Telematik-Sektoren, und zwar in der Aus- und Weiterbildung, im Verkehr, in der Medizin, in der Administration, im ländlichen Raum, für Bibliotheken, Telematik für Behinderte bis hin zu Telematik in der Gesellschaft.

Die Vorstellungen über Lehr- und Forschungsinhalte der Wissenschaftsdisziplin Telematik gehen weit auseinander. Zunächst könnte man unterscheiden zwischen der **Telematik im engeren Sinne**, die „fortgeschrittene Dienste“ (Mehrwertdienste) wie Telefax, Teletext, Mailbox, Teletex (BTX) gegenüber den Standard-Sprachdiensten in Fernsprechnetzen beschreibt, und der **Telematik im weiteren Sinne**, die durch die Interpretation des oben genannten Berichts weiterreichende Inhalte adressiert. Während in der Informationstechnik schwerpunktmäßig Signale untersucht werden und der Begriff der Information aus den Signalen entsprechend der Shannon'schen Theorie abgeleitet wird, steht in der Telematik der Daten-Begriff im Mittelpunkt. Aus den Daten erhält man über das Hinzufügen von Semantik Nachrichten, die zeit- und ortsabhängig für eine bestimmte Person relevant sein können, also Informationen (relevante Nachrichten) enthalten können.

Betrachtet man die Forschungsinhalte der oben genannten EU-Sektoren (Telematik-Anwendungsfelder), so lassen sich eine Reihe von Grundlagen wie: Vernetzungsprinzipien, Dienste, Protokolle, verteilte Anwendungen, Mobilität, Sicherheit, Modellierung und

**Telematik**

**Daten  
Nachricht  
Information**

Simulation, Netzmanagement sowie informale und formale Beschreibungs- und Spezifikationsmethoden abstrahieren, die für die unterschiedlichen Anwendungsgebiete gleichermaßen benötigt werden. Diesen Telematik-Grundlagen, die bei der Bearbeitung von Projekten der Verkehrstelematik und z. B. auch der Medizin-Telematik benötigt werden, widmet sich das vorliegende Lehrbuch. Es beschäftigt sich also mit den Grundlagen der Telematik im weiteren Sinne. Der Untertitel: „Netze – Dienste – Protokolle“ soll zum Ausdruck bringen, welche Schwerpunkte in diesem Buch gesetzt werden.

Die **Telematik** untersucht als Wissenschaftsdisziplin Probleme der *praktischen Informatik* und der *angewandten Informatik*. Die Forschungsinhalte der Telematik betreffen die technische Infrastruktur verteilter Systeme (Netze), die Netz-Dienste und die darauf aufbauenden Anwendungen, die Regeln zum Austausch von Nachrichten (Protokolle) sowie auch die Werkzeuge zur Entwicklung verteilter Anwendungen. Die komplexen funktionalen Aufgaben, die in einem vernetzten Computer zu lösen sind, werden üblicherweise mit Hilfe von Schichtenmodellen separiert und so detaillierten Betrachtungen zugänglich gemacht. In dieser Sichtweise reicht das Betätigungsfeld der Telematik von der Bitübertragung auf dem physikalischen Medium bis hin zur komplexen Netzanwendung.

- /1.1/ Bollmann, S.: *Kursbuch neue Medien*. – Rowohlt Verlag, 1995
- /1.2/ Flusser, V.: *Ins Universum der technischen Bilder*. – Göttingen: 1985
- /1.3/ Nora, S.; Minc, A.: *L'informatisation de la Societe (1978)*. – übersetzt in: *Die Informatisierung der Gesellschaft*. – Hrsg. Uwe Kalbhen, Vorwort von Ulrich Lohmar. – Frankfurt/Main, New York: Campus Verlag, 1979
- /1.4/ Gillette, D.: *Combining Communications and Computing: Telematics Infrastructures in dem Buch: Cities and Their Vital Systems: Infrastructure Past, Present, and Future*. – U.S. National Academy of Engineering, 1988

# 2 Schichtenmodelle, Dienste und Protokolle

Jörg Deutschmann, Dietrich Reschke

## 2.1 Einleitung

### 2.1.1 Allgemeines

Betriebssysteme erlauben es Computeranwendern, in einfacher Weise die Hardware zu nutzen und Anwendungsprogramme ablaufen zu lassen. Ebenso einfach sollte es möglich sein, dass Anwender oder Anwendungen verschiedener Computer miteinander kommunizieren, um Informationen auszutauschen oder Probleme kooperativ zu lösen. Beide Szenarien erfordern die Bearbeitung ganzer Aufgabenkomplexe. Die Realisierung dieser Aufgaben mit Hilfe monolithischer Programme ist möglich, aber unflexibel und wartungsintensiv. *Aus diesem Grunde werden komplexe Aufgabenstellungen funktional zerlegt und in Schichtenarchitekturen (layered architectures) realisiert.* Die einzelnen Schichten übernehmen dabei spezifische Aufgabenstellungen. Zu ihrer Lösung benötigen sie im allgemeinen die Dienste anderer Schichten. Zur Inanspruchnahme und zum Liefern von Dienstleistungen besitzen Schichten definierte Schnittstellen (*interfaces*). Durch eine hierarchische Anordnung der Schichten steht in der obersten Schicht die Gesamtfunktionalität des Systems zur Verfügung. Aus diesem Konzept ergeben sich eine Reihe von Vorteilen, wie das unabhängige Entwickeln von Komponenten, das Austauschen von Protokollen, das Standardisieren einzelner Komponenten, die Verbesserung der Kompatibilität zwischen heterogenen Systemen, die bessere Wartbarkeit, die Erweiterbarkeit u. a.

**Schichten-  
architekturen**

#### **Welche Kriterien sind bei der funktionalen Schichteneinteilung eines Aufgabenkomplexes zu beachten?**

- Es sollten nicht zu viele Schichten erzeugt werden.
- Die Anzahl der Interaktionen über die Schnittstelle sollte minimiert sein.
- Ähnliche Funktionen sollten in der gleichen Schicht angeordnet werden.
- Schichten sollten so angelegt sein, dass neue Protokolle sowie ein völliges Redesign auf Soft- und Hardwarebasis, unter Beibehaltung der Schnittstellendienste, möglich sind.
- Bei der Schichteneinteilung sollten existierende Interfacestandards berücksichtigt werden.
- Unterschiedliche Abstraktionsgrade können zur Einführung einer neuen Schicht führen.
- Die genannten Kriterien gelten ebenso für die Einteilung einer Schicht in Subschichten.

<p><b>abstrakte Sicht</b></p> <p><b>funktionale Sicht</b></p> <p><b>Dienstesicht</b></p> <p><b>Protokollsicht</b></p>	<p><b>Ein Schichtenmodell kann unter vier Sichten betrachtet werden:</b></p> <ul style="list-style-type: none"> <li>• <b>Die abstrakte Sicht</b> beschreibt ein allgemeines, von dem speziellen Anwendungsfeld (Betriebssysteme, Netze usw.) unabhängiges Schichtenmodell mit einer variablen Schichtenanzahl.</li> <li>• <b>Die funktionale Sicht</b> beschränkt sich auf ein spezifisches Anwendungsfeld, z. B. verteilte Rechnersysteme, definiert in diesem Kontext die Aufgaben der einzelnen Schichten und legt auch ihre Anzahl fest.</li> <li>• <b>Die Dienstesicht</b> definiert die von jeder Schicht aufgrund ihrer speziellen Funktionalität an die übergeordnete Schicht zu erbringenden Dienste. Sie legt die Dienstenamen (service primitive) und die Ablauffolge fest, unter der sie in Anspruch genommen werden können.</li> <li>• <b>Die Protokollsicht</b> zeigt die zwischen Partnerinstanzen in einer Schicht spezifizierten Protokolle. Diese Protokolle legen Syntax und Semantik der auszutauschenden Datenelemente fest und realisieren damit die angeforderten Dienste.</li> </ul> <p>Im Folgenden werden Schichtenmodelle zur Realisierung von Rechnernetzfunktionalität näher betrachtet.</p>
<p><b>Anwendungsinstanzen</b></p> <p><b>Offene Systeme</b></p> <p><b>Transitsysteme</b></p>	<p><b>2.1.2 Die Grundelemente einer Kommunikationsarchitektur</b></p> <ul style="list-style-type: none"> <li>• <i>Anwendungsinstanzen</i> bilden die logischen Einheiten, zwischen denen die Kommunikation stattfindet. Sie stellen die Abstraktion eines Endnutzers (Programm oder Mensch) an einem Endsystem (Computer) dar.</li> <li>• <i>Offene Systeme</i> sind Endsysteme mit ihren Anwendungsinstanzen oder Transitsystemen, die diese Endsysteme verbinden, falls sie nicht direkt miteinander verbunden sind. Offene Systeme ermöglichen das Zusammenarbeiten, indem sie Regeln einhalten, die in einem Satz von Normen festgelegt sind.</li> <li>• <i>Übertragungsmedien</i> dienen zur Verbindung von Systemen.</li> </ul> <p>Aus Sicht der Anwendungsinstanzen bilden Endsysteme, Transitsysteme und Übertragungsmedien zusammengenommen das dienstbringende System.</p>
<p><b>Protokoll</b></p> <p><b>Funktionsschicht</b></p> <p><b>Kommunikationsdienstleistung</b></p> <p><b>entity</b></p>	<p><b>2.1.3 Das Schichtungsprinzip</b></p> <p>Es wird zwischen den inhaltlichen (anwendungsbezogenen) und den unterstützenden (formalen) Komponenten einer Kommunikation unterschieden.</p> <p><i>Die inhaltlichen Komponenten</i> umfassen Syntax und Semantik der auszutauschenden Anwendungsinformationen. Sie werden in einem anwendungsspezifischen <i>Protokoll</i> (<i>protocol</i>) vereinbart, das außerdem Festlegungen bezüglich Antwortzeiten, Ablauffolgen und Dateneinheiten enthalten kann.</p> <p>Die Kommunikation zwischen den Anwendungsinstanzen wird durch eine Hierarchie von <i>Kommunikationsdiensten</i> (<i>services</i>) unterstützt. Durch die Zerlegung der Gesamtdienstleistung (unterstützende Komponente) erhält man Funktionsschichten. Diese Funktionsschichten müssen über Systemgrenzen hinweg in allen beteiligten Systemen realisiert sein. Die Kommunikationsdienstleistung, die einer höheren Schicht angeboten wird, umfasst neben der eigenen auch alle in der Hierarchie niedrigeren Kommunikationsdienste. Die von <b>einer</b> Funktionsschicht zu erbringende Leistung ergibt sich als Differenz aus den in der Hierarchie benachbarten Kommunikationsdienstleistungen. Die Dienstleistung selbst wird durch Instanzen (<i>entities</i>) erbracht, die zwar unterschiedli-</p>

chen Systemen, aber derselben Schicht angehören. Dabei bleibt der höheren Schicht verborgen, wie diese Dienstleistung erbracht wird (*information hiding*). Eine Instanz benötigt zur Realisierung ihrer Dienste die Dienstleistung unterer Instanzen, d. h., sie ist *service user* und *service provider* zugleich. Die Instanzen kommunizieren auf der Basis standardisierter *Schichtenprotokolle*. Ein Subsystem umfasst alle Instanzen einer Schicht in einem System. Die Hierarchie der Kommunikationsdienste hat also eine Hierarchie von Subsystemen, Instanzen und Protokollen zur Folge.

### 2.1.4 Übungen

#### Aufgaben

- 2.1 Welche Nachteile hat die Lösung von Problemen der Rechnerkommunikation auf dem Niveau der Anwendungsprogrammierung?
- 2.2 Was sind die Voraussetzungen für eine transparente Austauschbarkeit von Hard- und Softwareinstanzen, die eine spezifische Aufgabe bei der Rechnerkommunikation erfüllen?
- 2.3 Was verstehen Sie unter einem offenen System im Zusammenhang mit der Realisierung von Rechnernetzfunktionalität?
- 2.4 Wie sind die Begriffe Dienst und Protokoll in Bezug auf vertikale und horizontale Kommunikation innerhalb eines Schichtenmodells einzuordnen?
- 2.5 Wovon hängt die Anzahl der für ein Modell der Rechnerkommunikation benötigten Schichten ab?

## 2.2 Das Basis-Referenzmodell, die funktionale Schichtung

Es gibt verschiedene Spezifikationen und Standards für offene Systeme. Jedes offene System gehört einer bestimmten Architektur an. Die Architektur wird durch ein *Referenzmodell* beschrieben. Das Referenzmodell spezifiziert das Organisationsprinzip der Architektur und liefert ein Modell zur Definition von Diensten und Protokollen sowie für die Beziehungen zwischen diesen.

Beispiele für Architekturen bzw. Referenzmodelle sind:

- OSI RM – **O**pen **S**ystems **I**nterconnection **R**eference **M**odel (1984)
- TCP/IP – **T**ransmission **C**ontrol **P**rotocol/**I**nternet **P**rotocol Architecture (1974),
- B-ISDN RM – **B**roadband **I**ntegrated **S**ervices **D**igital **N**etwork **R**eference **M**odel (1994).

#### Standardisierungsgremien:

- ITU International Telecommunication Union (UN-Organisation)
  - ITU-T (vormals CCITT, Fernsprech- und Datenkommunikation)
  - ITU-R (vormals CCIR, Funk)
  - ITU-D Entwicklung  
International Recommendation (Empfehlungen)
- ISO International Organization for Standardization (mehr als 90 nationale Standardisierungsbehörden)



- TC1 ... TCn Technical Committee, jedes TC besteht aus SCs Sub Committees und diese aus WGs Working Groups Internationale Standards
- IAB Internet Architecture Board
- IETF Internet Engineering Task Force
- IRTF Internet Research Task Force  
RFCs Request For Comments, Internetstandards

### 2.2.1 Das ISO/OSI-Referenzmodell

Das ISO/OSI RM (*ISO-International Organization for Standardization*) beschreibt allgemeine Prinzipien der Vernetzung von offenen Systemen und gibt Konstruktionsregeln für offene Systeme, die der OSI-Architektur folgen, an. Es stellt somit ein abstraktes Modell für offene Systeme zur Verfügung.

Das OSI RM beschreibt aber keine spezielle Implementierung in einem realen System. Es definiert transportorientierte und anwendungsorientierte Schichten. Dienste werden an *Dienstzugangspunkten* (*SAP – Service Access Point*) mit Hilfe von *Dienstelementen* (*service primitive*) der nächsthöheren Schicht zugänglich gemacht. Diese Dienstzugangspunkte befinden sich an der Schnittstelle zwischen den Schichten und werden nach der unterliegenden Schicht benannt. Es gibt verbindungsorientierte und verbindungslose Dienste. Verbindungsorientierte Dienste setzen sich aus den Phasen: Verbindungsaufbau, Datentransfer und Verbindungsabbau zusammen. Jede dieser Phasen besteht aus einer festgelegten Dienstelementefolge. Man unterscheidet in diesem Zusammenhang vier Dienstelementetypen: *Anforderung* (*request*) vom Dienstanutzer (*service user*), *Anzeige* (*indication*) einer Anforderung vom Dienstbereitsteller (*service provider*) des Partnersystems, *Antwort* (*response*) vom Dienstanutzer des Partnersystems und *Bestätigung* (*confirm*) der Verbindungsannahme vom Dienstbereitsteller in dem Initiatorsystem. Verbindungen (*connections*) zwischen Instanzen einer Schicht enden innerhalb von Dienstzugangspunkten an Verbindungsendpunkten (*CEP – Connection End Point*). Zur Adressierung von Instanzen und Verbindungsendpunkten werden Kennzeichner (*identifier*) vergeben. Für die Kommunikation zwischen benachbarten Schichten und zwischen Partnerinstanzen innerhalb einer Schicht werden Dateneinheiten (*data units*) definiert. Man unterscheidet in vertikaler und horizontaler Richtung des Schichtenmodells Nutzdaten (*user data*) und Steuerinformationen (*control information*). Die funktionale Schichtung im ISO/OSI Referenzmodell umfasst sieben Schichten, wobei insbesondere die Schichten zwei und drei in Subschichten unterteilt sind. Üblicherweise werden die Schichtkennzeichner in Klammern gesetzt, wenn keine spezielle funktionale Schicht gemeint ist.

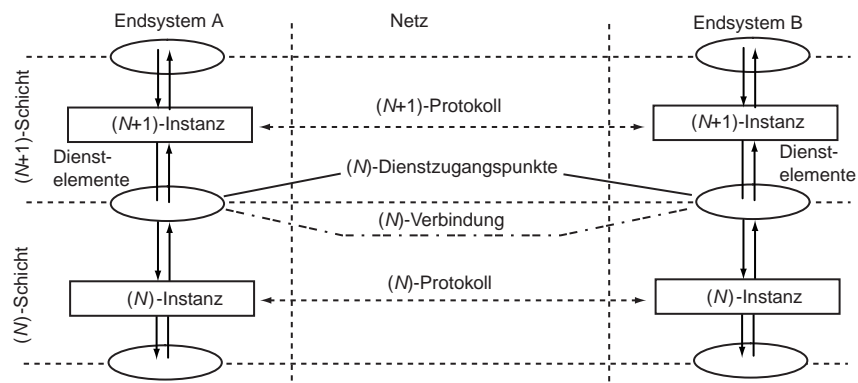


Bild 2.1  
Abstraktes  
Schichtenmodell

ISO/OSI

Dienstzu-  
gangspunkte

verbindungs-  
orientierte  
Dienste

## Realisierungsvariante

### Beispiel 2.1

Obwohl das Referenzmodell nichts über mögliche Realisierungen aussagt, soll eine denkbare Implementierungsvariante betrachtet werden. Die Instanzen seien Programme, in denen die Algorithmen zum Erbringen einer bestimmten Schichtfunktionalität (beispielsweise des Verbindungsmanagements) umgesetzt sind. Diese Programme stellen Protokollmaschinen dar, die als endliche Zustandsmaschinen realisiert sind. Eingangsgröße wird ein Ereignis (Dienstelement) sein, Ausgangsgröße eine Aktion (Dienstelement, Zustandsveränderung). Implementierungsvarianten sind Zustandstabellen oder Programmsteuerkonstrukte (CASE-, IF-THEN-ELSE Klauseln). Damit die Dienstleistung erbracht werden kann, müssen aus diesen Programmen (durch Übersetzen, Linken usw.) abarbeitfähige Einheiten (Prozesse, *Vorkommnisse*) werden. Zu jeder Instanz (als Prozessschablone) können eine Menge von Vorkommnissen (z.B. reale Verbindungen) gehören. Die Prozesse können quasiparallel (nebenläufig) oder parallel ablaufen. Zur Reduzierung der Prozessanzahl und zur Steigerung der Laufzeiteffizienz kann ein Prozess auch so entworfen sein (*multiple threads*), dass er mehrere Ereignisse gleichzeitig bearbeitet. Da ein Subsystem (Schicht in einem System) n-Funktionalitäten beinhalten kann, gehören in dieser Betrachtungsweise, n-Instanzen mit jeweils einer Anzahl von Vorkommnissen zu diesem Subsystem. Während die Instanzen über Service Access Points adressiert werden, verbinden die Connection End Points konkrete Vorkommnisse dieser Instanzen. Eine Transportprotokollinstanz (z. B. TCP-Instanz) kann z. B. mehrere Anwendungsverbindungsendpunkte (PortNummern, genauer: Sitzungsendpunkte) über einen Dienstzugangspunkt (Protokoll-Id für TCP) verwalten.

### Schicht 1: Physikalische Schicht (physical layer)

Sie beschreibt die mechanischen, elektrischen, funktionalen und prozeduralen Eigenschaften von physikalischen Verbindungen zur Bitübertragung zwischen Schicht-2-Instanzen. Die Instanzen der Schicht 1 sind über ein Übertragungsmedium (Lichtwellenleiter, Kupferleitung, Funk usw.) miteinander verbunden.

### Schicht 2: Sicherungsschicht (data link layer)

In ihr wird eine gesicherte Schicht 2 – Verbindung zwischen Schicht 3 – Instanzen aufgebaut. Dazu gehören das Verbindungsmanagement (Punkt-zu-Punkt-Verbindungen zwischen benachbarten Systemen), die Fehlererkennung und -korrektur, die Reaktion auf nicht selbst korrigierbare Fehler, sowie die Flusssteuerung mit dem Ziel einer bestmöglichen Ressourcenausnutzung (Übertragungsmedium, Pufferkapazitäten auf Sender- und Empfängerseite). In Shared-Medium-Netzen (lokale Netze, z. B. Ethernet) wird in der MAC- (Medium Access Control) Subschicht der Zugriff auf das gemeinsame Übertragungsmedium koordiniert.

### Schicht 3: Netzwerkschicht (network layer)

In der Netzwerkschicht werden Endsysteme über Transitsysteme und Netze hinweg adressiert bzw. bei verbindungsorientierten Diensten wird eine Netzwerkverbindung zwischen Transportinstanzen aufgebaut (virtuelle Ende-zu-Ende-Verbindung zwischen Endsystemen). Wesentliche Funktionen sind die Wegefindung (Routing) und die Absicherung von Verbindungsqualitäten (QoS – Quality of Service). Entspricht das softwaremäßig realisierte Schicht-3-Protokoll in einem Endsystem nicht dem im An-



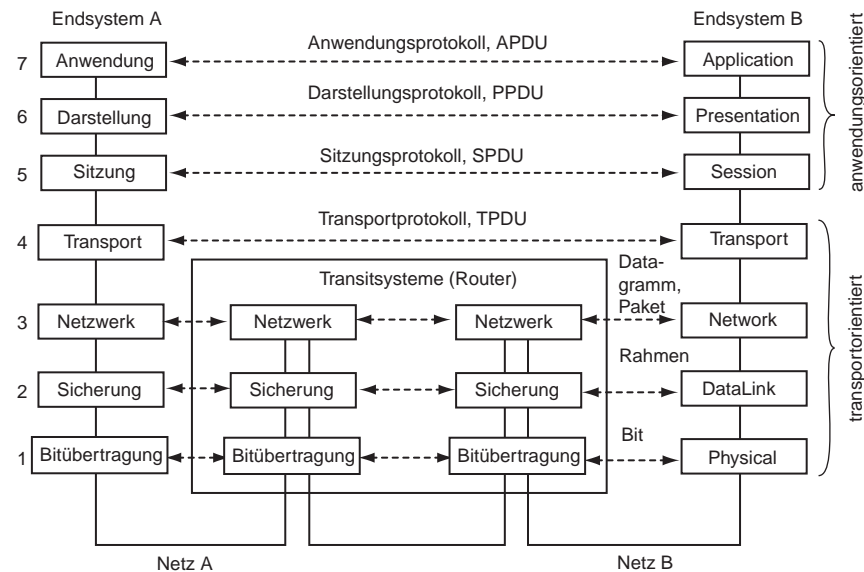
**Bitübertragung**

**Sicherungsschicht**

**Netzwerkschicht**



Bild 2.2  
ISO/OSI  
Referenzmodell



schlussnetz verwendet, so ist im Allgemeinen eine so genannte Konvergenzschicht zur Protokollangleichung notwendig.

#### Schicht 4: Transportschicht (transport layer)

Die Transportschicht hat die Differenz zwischen QoS-Parametern (z. B. Fehlererkennung und die Reaktion auf Fehler, Reihenfolgewiederherstellung) der nicht beeinflussbaren Transitsysteme und den QoS-Forderungen der Anwendung auszugleichen. Der Funktionsumfang ist also stark von der Netzwerkschicht in den Transitsystemen abhängig. Sie ist die letzte der transportorientierten Schichten und die erste Schicht, deren Protokolldateneinheiten für das Netz transparent sind, also in den Transitsystemen nicht ausgewertet werden.

#### Schicht 5: Sitzungsschicht (session layer):

Die Sitzungsschicht ist die erste der anwendungsorientierten Schichten. Eine Sitzungsverbindung ist nur in Verbindung mit einer bestimmten Anwendung sinnvoll. Ein wichtiger Dienst ist die Strukturierung von Anwendungen durch das Setzen von Synchronisationspunkten. Im Fehlerfall kann eine Anwendung ab definiertem Synchronisationspunkt neu gestartet werden. Durch die Vergabe von Token kann das Senderecht jeweils einer Anwendungsinstanz übertragen werden (Halbduplex-Betrieb).

#### Schicht 6: Darstellungsschicht (presentation layer)

Die Darstellungsschicht ist für die Passfähigkeit von Syntax und Semantik der zu übertragenden Anwendungsinformationen in den unterschiedlichen Systemen verantwortlich. Ziel ist es, dass sich die Anwendungen „verstehen“. Anwendungen benutzen unterschiedliche Sprachsyntaxen, Computer haben verschiedene interne Datendarstellungen, die Daten können mit verschiedenen Transfersyntaxen übertragen werden. Aus diesem Grunde handeln die beteiligten Darstellungsinstanzen für jede Anwendung einen geeigneten Darstellungskontext aus (Syntax der Anwendungsinformation, z. B. ASN.1- Abstract Syntax Notation One plus Transfersyntax, z. B. BER- Basic Encoding Rules).

**Transport-**  
**schicht**

**Sitzungs-**  
**schicht**

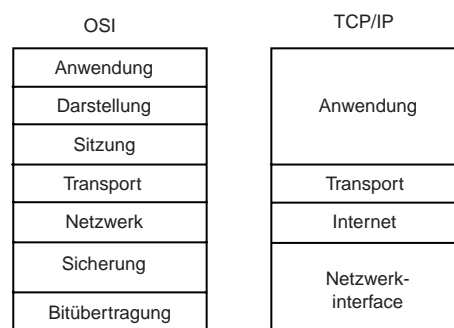
**Darstellungs-**  
**schicht**

### Schicht 7: Anwendungsschicht (application layer)

Die Anwendungsschicht besitzt keine Dienstzugangspunkte, von denen Verbindungen ausgehen könnten, weil sie die oberste Schicht in der Hierarchie bildet, und trotzdem liefert sie Kommunikationsdienste für Anwendungsinstanzen innerhalb der gleichen Schicht. Man spricht deshalb von Anwendungsassoziationen. Sie werden durch sogenannte Association Control Service Elements (ACSE) unterstützt. Die Funktionalität der Anwendungsschicht besteht in der Bereitstellung von anwendungsunterstützenden Elementen (ASE – Application Service Element). Hierbei unterscheidet man anwendungsspezifische Elemente (SASE – Specific Application Service Element), wie Filetransfer, Mailedienst, Verzeichnisdienst, Systemmanagement, künftig sicher auch office document management, banking, retrieval, electronic library application etc. und gemeinsam genutzte Dienste (CASE – Common Application Service Element) wie ACSE (Auf- und Abbau von Assoziationen, die die Verbindungen zwischen Anwendungsinstanzen darstellen), ROSE – Remote Operations Service Element (Austausch von Informationen zwischen Anwendungsinstanzen entspricht dem Remote Procedure Call), RTSE – Reliable Transfer Service Element (Synchronisation von Anwendungsinstanzen, dazu Setzen von Synchronisationspunkten, definierter Wiederanlauf nach Fehler), CCR – Commitment Concurrency and Recovery (Unterstützung von Transaktionen, beispielsweise in Banking-Anwendungen). Die Grenze zwischen SASE und CASE ist fließend. Die nutzerspezifische Anwendung kann auf diese vorhandenen Dienste zurückgreifen. Da die Pragmatiker auch im Falle des ISO/OSI- Modells schneller als die Standardisierungsbehörden waren, haben die aus diesem Modell abgeleiteten Protokolle bis heute nicht die erwünschte Bedeutung erlangt. Der Nutzen dieses Modells besteht in der relativ vollständigen und präzisen Definition der einzelnen Elemente eines verteilten Systems sowie ihres Zusammenwirkens. Damit existiert eine gemeinsame Sprache über die Grenzen bestimmter Netzwerkphilosophien hinweg.

### 2.2.2 Die TCP/IP Schichtenarchitektur

Das TCP/IP-Referenzmodell ist nach dem verbindungsorientierten Transportprotokoll TCP und dem verbindungslosen Netzwerkprotokoll IP benannt. Diese Protokolle sind unmittelbar mit der Entwicklung des Internets aus einem ursprünglich (1969) vom DoD (Department of Defense der USA) geförderten Projekt ARPA (Advanced Research Project Agency) etwa 1974 entwickelt worden. Die Standards werden über RFCs (Request For Comments) veröffentlicht, die die IETF (Internet Engineering Task Force) ausarbeitet.



**Anwendungs-  
schicht**

**TCP/IP-  
Referenzmodell**

*Bild 2.3  
Vergleich  
OSI –TCP/IP-  
Schichtung*

Das aus praktischen Erfahrungen entstandene TCP/IP-Referenzmodell weist nur vier Schichten auf. Die im ISO/OSI-Modell vorgesehene Funktionalität von Sitzungsschicht und Darstellungsschicht wird, wenn benötigt, in die Anwendungsschicht verlagert. Da die Schicht 4 im ISO/OSI-Modell die Transportschicht und im TCP/IP-Modell die Anwendungsschicht darstellt, ist es bei der Verwendung von Schichtnummern wichtig, das zugehörige Referenzmodell mit anzugeben.

### 2.2.3 Übungen

#### Aufgaben

- 2.6 Welche Bedeutung hat das Open System Interconnection Reference Model?  
 2.7 Das Bild 2.4 zeigt das Fragment eines abstrakten Schichtenmodells.

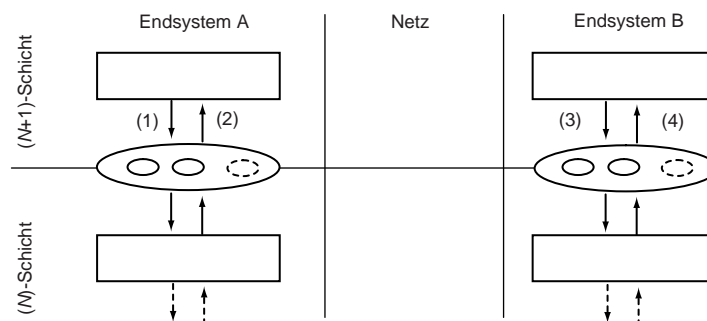


Bild 2.4  
Fragment  
eines abstrakten  
Schichtenmodells

- a) Machen Sie im Bild 1.4 die folgenden Begriffe kenntlich: Dienstanwender, Dienstbringer, Dienstzugangspunkt, Dienstelement, Verbindungsendpunkt, Protokoll!  
 b) Weisen Sie den mit (1), (2), (3), (4) gekennzeichneten Pfeilen die entsprechenden Dienstelementtypen für einen angenommenen erfolgreichen Verbindungsaufbau zu, wenn davon ausgegangen wird, dass Endsystem A als Initiator der Verbindung auftritt!
- 2.8 Warum hat man sich Ihrer Meinung nach beim TCP/IP-Referenzmodell für eine Anzahl von nur vier Schichten entschieden?

#### Referat

- 2.1 Charakterisieren Sie das OSI-Referenzmodell als ein Schichtenmodell zur Realisierung von Rechnernetzfunktionalität mit funktionaler Sichtweise! Gehen Sie dabei auf die Aufgabenverteilung zwischen den einzelnen Schichten des OSI-Referenzmodells ein!

## 2.3 Dienste und Protokolle

Dienste und Protokolle werden in Standards beschrieben. Jeder „Schicht-Standard“ enthält ein Paar aus Dienst-Spezifikation und Protokoll-Spezifikation. Aus einer *Dienst-Spezifikation* können mehrere Protokoll-Spezifikationen abgeleitet werden. Die Dienst-Spezifikation enthält das gesamte Dienstleistungsangebot an die nächsthöhere Schicht, die einzelnen Dienstelemente mit Hilfe derer auf die Dienstleistung zugegriffen werden kann, die Parameter der Dienstelemente, sowie die Abfolge der einzelnen Teildienste (z. B. Datentransfer nur nach Verbindungsaufbau). Diese Abfolgen können mit Zustandsübergangsdiagrammen oder Tabellen beschrieben werden. Die Dienste werden durch