

# **SCHICHTENMODELL NACH ISO/OSI**

**Einführung in das ISO/OSI-Schichtenmodell  
für die Kommunikation  
in offenen Systemen**

# INHALTSVERZEICHNIS

<b>1 DAS OSI-REFERENZMODELL</b>	<b>5</b>
<b>1.1 Der Begriff „Offenes System“</b>	<b>5</b>
<b>1.2 Offene Systeme nach ISO/OSI</b>	<b>6</b>
<b>1.3 Schichtaufgaben</b>	<b>12</b>
1.3.1 Bitübertragungsschicht (Schicht 1), physical layer	13
1.3.2 Sicherungsschicht (Schicht 2), data link layer	13
1.3.3 Vermittlungsschicht (Schicht 3), network layer	15
1.3.4 Transportschicht (Schicht 4), transport layer	16
1.3.5 Kommunikationssteuerungsschicht (Schicht 5), session layer	17
1.3.6 Darstellungsschicht (Schicht 6), presentation layer	17
1.3.7 Anwendungsschicht (Schicht 7), application layer	18
<b>2 STRUKTUR DES OSI-MODELLES</b>	<b>20</b>
<b>2.1 Instanzen</b>	<b>20</b>
<b>2.2 Dienste</b>	<b>21</b>
2.2.1 Verbindungslose Dienste	22
2.2.2 Verbindungsorientierte Dienste	23
<b>2.3 Dienstelemente</b>	<b>25</b>
<b>2.4 Protokolle</b>	<b>27</b>
<b>2.5 Paketstrukturen</b>	<b>28</b>
<b>2.6 SAP-Adressierung</b>	<b>30</b>
<b>2.7 Diskussion des OSI-Modells</b>	<b>32</b>
<b>3 REPEATER, BRIDGES UND ROUTER IN LOKALEN NETZEN</b>	<b>33</b>

## **Begriffe und Abkürzungen**

### **Application Layer - Anwendungsschicht**

Siebte Schicht des ISO/OSI-Modells. Sie stellt die Schnittstelle des Kommunikationssystems zu Anwendungsprogrammen dar. Die Schicht 7 enthält allgemeine Hilfsdienste für die Kommunikation sowie spezielle Kommunikationsdienste wie File Transfer, Virtuelles Terminal etc.

### **Data Link Layer - Sicherungsschicht**

Zweite Schicht des ISO/OSI-Modells. Sie stellt eine gesicherte Punkt-zu-Punkt-Verbindung zwischen 2 Stationen zur Verfügung.

### **Daten-End-Einrichtung (DEE)**

siehe **End System**

### **Daten-Übertragungs-Einheit (DÜE) - Data Communication Equipment (DCE)**

Ein Rechner im Netz (Netzknoten), der zur Übermittlung dient.

### **DCE**

siehe Daten-Übertragungs-Einheit

### **DEE**

siehe End-System

### **DÜE**

siehe Daten-Übertragungs-Einheit

### **End System - Endsystem, Daten-End-Einrichtung (DEE)**

System am Netzwerkrand, das in der Lage ist, mit anderen Endsystemen über das Netzwerk zu kommunizieren.

### **Entity - Instanz**

Prozeß in einer Schicht, der die jeweiligen Schichtaufgaben erfüllt. In derselben Schicht eines Endsystems können mehrere Instanzen vorhanden sein.

### **Folgeinstanz**

Zwei vertikal benachbarte Instanzen nennt man obere und untere Folgeinstanz.

### **Interface Control Information - Kommandoteil**

Steuerinformation für die Folgeschicht zur Verarbeitung der übergebenen User-Daten.

### **ISO (International Standardization Organisation)**

erarbeitet und verabschiedet in Gremien Standards und Normen in Zusammenarbeit mit den nationalen Normungsgremien wie DIN (Deutsches Institut für Normung), etc. Eines der Gebiete, auf denen ISO tätig ist, ist die Telekommunikation.

**Network Layer - Vermittlungsschicht**

Dritte Schicht des ISO/OSI-Modells. Sie organisiert eine Ende-zu-Ende-Verbindung zwischen kommunizierenden Endsystemen.

**Netzknoten**

siehe Daten-Übertragungs-Einheit

**OSI (Open Systems Interconnection)**

Offene Kommunikation mit Hilfe elektronischer Medien

**OSIBRM - OSI Basic Reference Model**

Kommunikationsmodell der International Standard Organisation (ISO).

**Partnerinstanz**

Zwei horizontal in einer Schicht kommunizierende Instanzen heißen Partnerinstanzen (peer entities).

**Peer - Partner**

Instanzen in der gleichen Schicht in verschiedenen Endsystemen.

**Physical Layer - Bitübertragungsschicht**

Erste Schicht des ISO/OSI-Modells. Sie befaßt sich mit der Anbindung eines Endsystems an ein physikalisches Übertragungsmedium und der darauf ablaufenden Bitübertragung.

**Presentation Layer - Darstellungsschicht**

Sechste Schicht des ISO/OSI-Modells. Sie ist verantwortlich für die Strukturierung der Dateneinheiten, die ausgetauscht werden.

**Protocol - Protokoll**

Protokolle sind Regeln zur Kommunikation zwischen Partnerinstanzen.

**Protocol Control Information (PCI) - Protokollsteuerungsinformation**

Ein Dienstanbieter fügt zu den ihm übergebenen Nutzdaten einen kleinen Header hinzu, die sogenannte Protocol Control Information. Diese stellt eine Steueranweisung für den Protokollpartner derselben Ebene dar.

**Protocol Data Unit (PDU) - Protokolldateneinheit**

Eine Nachricht die zwischen Partnerinstanzen ausgetauscht wird. Sie besteht aus Nutzdaten und PCI.

**Rechner**

beinhaltet einen Rechner aus HW, Betriebssystem, Kommunikationssystem und Anwendungs- und anderen Systemprogrammen. Durch das Kommunikationssystem ist ein solcher Rechner fähig, über ein Netz mit anderen Rechnern zu kommunizieren. Der Begriff Rechner wird hier äquivalent zu Station und System verwendet. ISO/OSI spricht in abstrakter Weise nur von Systemen, nicht von Rechnern oder Stationen

**Service - Dienst**

Ein Kommunikationsdienst nach ISO/OSI ist eine Dienstleistung, welche auf einem Rechner-Knoten von einer ISO/OSI-Schicht am Dienstzugangspunkt (Service Access Point) dem Aufrufer bereit gestellt wird. Der Kommunikationsdienst wird dabei erbracht durch die Instanzen dieser Schicht und ihr Schichtenprotokoll unter Nutzung der Dienste der tieferen Schichten.

**Service Access Point (SAP) - Dienstzugangspunkt**

Die Schnittstelle zwischen Schichteninstanz und Dienstbenutzer.

**Service Data Unit (SDU) - Dienstdateneinheit**

Nutzdaten einer Schicht, die einer untergeordneten Folgeschicht im Dienstelement übergeben werden.

**Service Primitive - Dienstelement**

Kommandos zur Inanspruchnahme von Diensten und die Ergebnismeldungen heißen Dienstelemente (service primitives)

**Service Provider - Diensterbringer**

Instanz einer untergeordneten Schicht, die den Instanzen einer höheren Schicht am Service Access Point einen Dienst zur Verfügung stellt.

**Service User - Dienstbenutzer**

Instanz, die den Dienst einer Instanz der nächst tieferen Schicht nutzt.

**Session Layer - Kommunikationssteuerungsschicht**

Fünfte Schicht des ISO/OSI-Modells. Sie regelt die Synchronisation zwischen Prozessen in verschiedenen Systemen.

**Station**

siehe Rechner

**System**

siehe Rechner

**Transport Layer - Transportschicht**

Vierte Schicht des ISO/OSI-Modells. Sie stellt Transportdienste verschiedener Güte den Prozessen der Endsysteme zur Verfügung und zwar in transparenter Weise. Transparent bedeutet, daß die Prozesse nicht mehr die Physik des Netzes, sondern nur mehr die Güte des Transportdienstes (Durchsatz, Sicherheit, ...) sehen.

# 1 Das OSI-Referenzmodell

Das OSI-Referenzmodell, auch als ISO/OSI-Referenzmodell oder OSI Basic Reference Model (OSIBRM) bezeichnet, ist das Kommunikationsmodell der International Standard Organisation (ISO). Es modelliert das Kommunikationssystem eines Rechners und seine Zusammenarbeit mit dem Kommunikationssystem eines anderen Rechners. Das **Kommunikationssystem (communication system)** eines Rechners ist für den rechnerübergreifenden Austausch von Nachrichten verantwortlich.

## 1.1 Der Begriff „Offenes System“

Bei „Offenen Systemen“ ist stets zu unterscheiden, was gemeint ist:

- Offene Systeme im Sinne von ISO/OSI
- oder offene Systeme nach X/Open bzw. POSIX

Die Standardisierung von X/Open bzw. POSIX befaßt sich im wesentlichen mit der **Standardisierung des Betriebssystems UNIX**. Die Standardisierung von ISO/OSI befaßt sich im wesentlichen mit der **Standardisierung des Kommunikationssystems eines Rechners, welches zur Kommunikation zwischen Rechnern dient**.

Im Rahmen der Standardisierung **offener Systeme (d.h. herstellerunabhängiger Systeme) im Sinne des X/Open Portability Guides bzw. von POSIX** werden Schnittstellen von Programmen zu UNIX-Betriebssystemen (nicht der Aufbau der Betriebssysteme der verschiedenen Hersteller) standardisiert. Die bevorzugte Sprache, für die als erste standardisierte Schnittstellen festgelegt werden, ist hierbei die Programmiersprache C (siehe X/Open Portability Guide Version 4). Der X/Open Portability Guide befaßt sich über die Standardisierung der Schnittstellen von Programmen zu UNIX-Betriebssystemen auch mit der Standardisierung von Utilities, Netzwerkdiensten, Window-Management, Daten-Management etc.

## 1.2 Offene Systeme nach ISO/OSI

Bereits in den 60er Jahren waren teilweise weltweite Rechnernetzwerke vorhanden. Als Beispiel seien etwa das bekannte ARPA-Netz (Forschungsnetz), verschiedene Herstellernetze oder Rechnerverbunde von Fluglinien und Banken erwähnt. Alle Herstellernetze hatten eines gemeinsam: Sie erfüllten ihre Aufgabe nur für einen sehr beschränkten Teilnehmerkreis. Es war Teilnehmern verschiedener Netzwerke nicht möglich, miteinander zu kommunizieren. Jedes dieser Netze samt ihren Datenendgeräten bildete ein sogenanntes geschlossenes System.

Ein **geschlossenes System** hat folgende Merkmale: es baut auf der Technologie eines Herstellers auf und ist mit Konkurrenzprodukten nicht kompatibel. Die Ausdehnung eines derartigen Netzes beschränkt sich auf einen bestimmten Teilnehmerkreis und oft auf ein beschränktes räumliches Gebiet. Datenübertragung von einem Netz in ein anderes ist nur mittels verhältnismäßig großem Aufwand an Hard- und Software möglich.

Im Jahre 1977 erkannte die **ISO (International Standard Organization)** die Notwendigkeit, den Bereich der Rechnerkommunikation zu normen. Im "Subcommittee 16", **Open Systems Interconnection (OSI)**, wurde die Arbeit aufgenommen. Dabei war zunächst das Ziel, die Kommunikation von Endsystemen verschiedener Hersteller, die über ein Fernmeldenetz (Wide Area Network) miteinander kommunizieren, zu normieren., damit Rechner verschiedener Hersteller in der Lage sind, miteinander Informationen über das Netz auszutauschen.

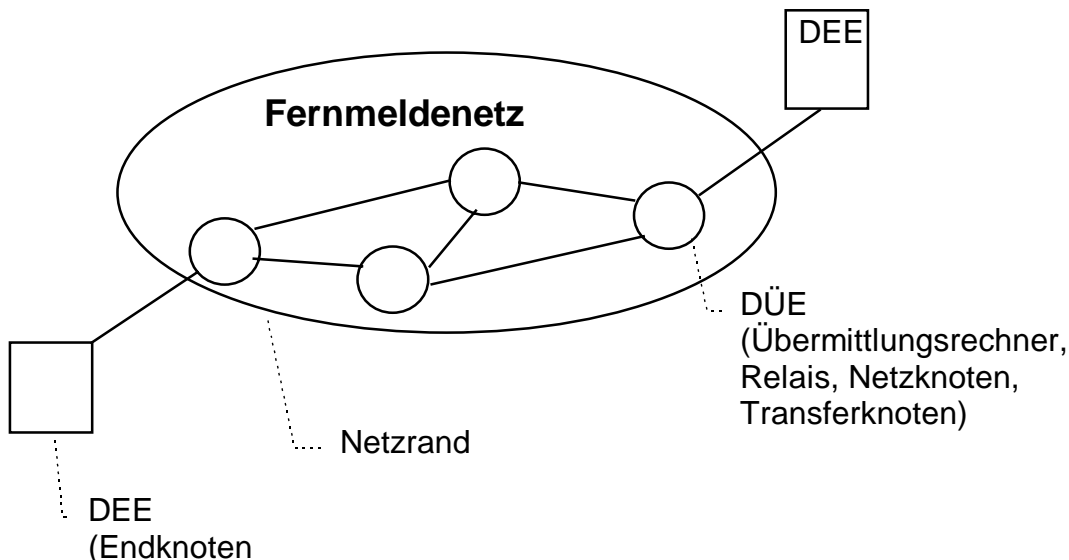


Bild 1.2 - 1 Rechner, die Datenendeinrichtungen (DEE) am Netzwerkrand des Fernmeldenetzes sind

Durch die Standardisierung sollte das Außenverhalten eines Systems, d.h. sein Kommunikationsverhalten, standardisiert werden. Mit anderen Worten, jeder Rechner soll sich nach außen gleichartig verhalten, ganz egal, von welchem Hersteller er kommt. Durch diese Standards soll erreicht werden, daß Systeme verschiedener Hersteller problemlos miteinander kommunizieren können.

Ein **offenes System** nach ISO/OSI ist also ein System, welches trotz herstellerabhängiger Implementierung mit anderen offenen Systemen nach festgelegten Normen kommuniziert.

Das Außenverhalten eines Rechners zeigt sich darin, daß er auf gewisse Reize (Telegramme) von außen, ein bestimmtes Verhalten zeigt, d.h. auf bestimmte Weise antwortet. Rein theoretisch hätte sich OSI darauf beschränken können, nur das **Außenverhalten** zu standardisieren (**Black box-Sicht**).

OSI ist etwas weiter gegangen. Wenn man nach außen passend reagieren soll, so muß man natürlich auch im Inneren des Rechners gewisse Arbeitseinheiten (Instanzen) haben, die für das Außenverhalten verantwortlich sind. OSI baute ein logisches Modell von **Arbeitseinheiten (Instanzen) eines Rechners** auf und sagte, welche Arbeitseinheit nach außen was tut. Jeder Rechner, mit dem man redet und der dem OSI-Modell genügt, soll dieselben Instanzen haben.

Dies bedeutet, daß OSI eine White-box-Sicht für die Logik, nicht für die Implementierung des Kommunikationssystems hat. Man sieht in das Kommunikationssystem hinein und sieht die Instanzen:

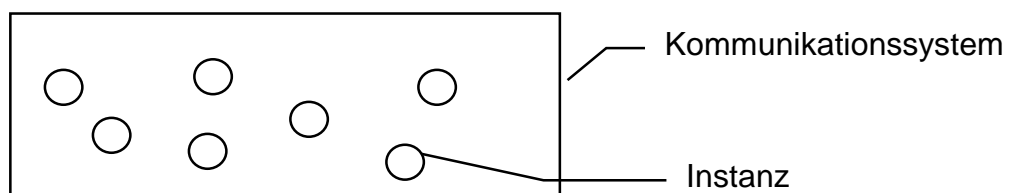
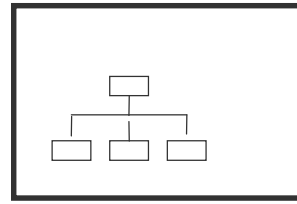


Bild 1.2 - 2 Instanzen als Arbeitseinheiten eines Kommunikationssystems

Da OSI sich für die white box-Sicht entschied, bedeutete dies mehr Arbeit als eine black box-Sicht. Die Logik im Inneren sollte ja klar strukturiert und auch überzeugend sein. Strukturierung bedeutet immer Zerlegung und Festlegung des Zusammenwirkens der Zerlegungsprodukte. Die Problemstellung der Modularisierung des Kommunikationssystems führte dann zur Zerlegung des Kommunikationssystems in 7 Schichten und damit auch zur Festlegung des Zusammenwirkens der 7 Schichten (Schichtenmodell nach ISO/OSI).

Erläuterung:

**BOX**



Bei einer Black-Box sieht man nur die Box selbst, nicht jedoch, was sich in ihrem Inneren verbirgt. Bei einer White-Box schaut man in das Innere der Box.

- ISO/OSI regelt das Außenverhalten des Kommunikationssystems.
- ISO/OSI zerlegt das Kommunikationssystem in Schichten.
- ISO/OSI legt das Zusammenwirken der Schichten fest.
- ISO/OSI muß für das Außenverhalten nicht nur die Logik, sondern auch den konkreten Aufbau einer Nachricht festlegen.
- ISO/OSI muß sich im Innern des Kommunikationssystems auf die Logik beschränken, da Kommunikationssystem und Betriebssystem oftmals herstellerspezifisch eng zusammenarbeiten.

Schon recht bald stellte sich heraus, daß die Modellierung einer verbindungsorientierten Kommunikation wie z.B. beim Telefon, nur ein Ausschnitt aus der Realität war. Auch eine verbindungslose Kommunikation, sei es in einem Funknetz oder in einem Lokalen Netz (LAN), mußte im ISO/OSI-Modell berücksichtigt werden. Ganz generell standardisiert das ISO/OSI-Modell also die Kommunikation in heterogenen Netzen. Heterogene Netze sind dabei Netze, die Stationen verschiedener Hersteller enthalten.

Die Modellierung jeglicher Kommunikation in heterogenen Netzen hatte zur Konsequenz, daß schon recht bald zum einen Subschichten (horizontale Strukturen), zum anderen aber alternative Möglichkeiten innerhalb einer Schicht (vertikale Strukturen) je nach Art der Kommunikation berücksichtigt werden mußten. Je nach Kommunikationsart kommen dann die spezifischen alternativen Möglichkeiten in einer Schicht zum Einsatz. Während also das berühmte 7-Schichtenmodell horizontale Schnitte durch das Kommunikationssystem beschreibt, beschreibt ein **Protokollstack**, welche Protokolle und welche Protokollmengen in welcher Schicht zum Zuge kommen und mit welcher zusätzlichen Vereinbarungen mögliche Varianten in einer Schicht ausgegrenzt werden. Dies ist für eine eindeutige Lösung erforderlich, da die Standards für die Protokolle in den einzelnen Schichten meist Optionen und Wahlmöglichkeiten beinhalten. Durch diese Festlegungen wird ein Protokollstack aufgebaut mit dem Ziel, eine bestimmte Kommunikationsfunktion zu erhalten. Solche Festlegungen werden auch als **funktionaler Standard** oder **Profil** bezeichnet. So gibt es beispielsweise in der Schicht 7 für die Dienste:

- File Transfer, Access and Management
- Message Handling
- Virtual Terminal
- Directory
- etc.

spezielle Profile.

Ein funktionaler Standard (ein Profil) beschreibt also einen Protokollstack, dabei werden in jeder Schicht des Protokollstacks bestimmte einzelne Profile festgelegt. Beispiele funktionaler Standards sind MAP (Manufacturing Automation Protocol) und TOP (Technical and Office Protocols). Sie gehen auf die Initiative großer Anwender zurück, nämlich auf General Motors bei MAP und Boeing bei TOP.

## Das ISO/OSI Basic Reference Model

Das **ISO/OSI Basic Reference Model**, auch bekannt unter dem kürzeren Namen **ISO/OSI Reference Model** darf die Implementierung des Kommunikationssystems nicht festlegen, da ja jeder Rechner seine eigene Architektur hat. Was es jedoch festlegen kann, ist ein allgemeines, abstraktes Modell, welches zum erwünschten Verhalten nach außen gemäß der gewünschten Verhaltensregeln führt. Dabei müssen die Verhaltensregeln vollständig angegeben sein, um ein definiertes Verhalten zu erreichen.

Die Frage der Modularisierung der Funktionalität eines Kommunikationssystems wurde nun nach einem „aristokratischen Prinzip“ durchgeführt. Es werden horizontale Schichten geschaffen. **Eine Instanz einer höheren Schicht hat als „Diener“ die Schicht darunter.** Die höhere Schicht arbeitet natürlich selbst auch etwas, aber es ist nicht vorgesehen, daß sie sich übernimmt. Für was hat sie schließlich ihren Diener. Dies bedeutet, daß sie Aufträge an ihren „Diener“ gibt und von ihm wieder Ergebnisse erhält.

Das Gesagte kann man auch etwas abstrakter formulieren:

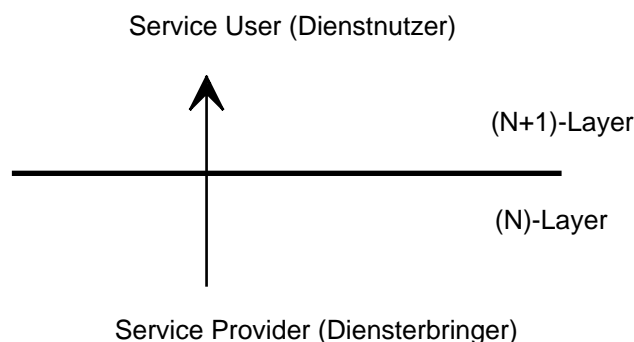


Bild 1.2 - 3 Services und Schichten

Es ist nicht begründbar, warum das ISO/OSI-Modell gerade 7 Schichten hat. Die Architekten der TCP/IP-Architektur erfanden 4 Schichten, die von DECnet erfunden 8 Schichten, SNA aus der IBM-Welt hat wiederum andere Schichten. Die Zahl der Schichten von ISO/OSI ist letztendlich auch nicht relevant, entscheidend ist aber, daß die verschiedenen Hersteller alle dieselben Schichten gemäß ISO/OSI implementieren, um damit interoperabel mit Rechnern anderer Hersteller zu werden.

Im folgenden seien diese Schichten einfach dargestellt:

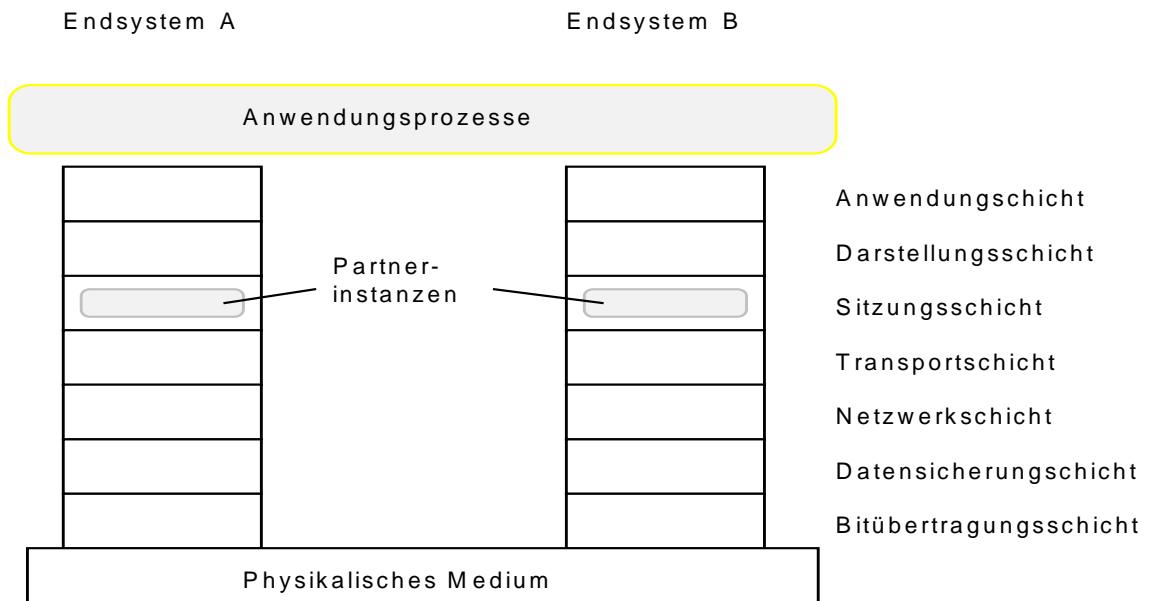


Bild 1.2 - 4 : Prinzipieller Aufbau des OSI-Schichtenmodells

Es sei an dieser Stelle aber bereits schon jetzt darauf hingewiesen, daß man im Schichtenmodell zum einen „aristokratische Prinzipien“ (siehe Diener oder Services) entdecken kann, daß man aber auch „streng demokratische Prinzipien“ entdecken kann. Während innerhalb eines Rechners gedient wird, verhalten sich ganze Rechner am Netzwerkrand gegeneinander als gleichberechtigte unabhängige Wesen. Eine Instanz in einer Schicht des einen Rechners redet mit der entsprechenden Instanz in derselben Schicht des anderen Rechners als gleichberechtigtem Partner (**peer-to-peer communication**).

Nicht jedes Schichtenmodell ist gleich auszulegen. Daher einige Worte zur Interpretation:

- In jeder Schicht gibt es Instanzen (Arbeitseinheiten, entities). Diese Instanzen erbringen die schichtspezifischen Leistungen.
- Die Schichtung ist streng hierarchisch. Ein Instanz der Schicht N (N-Instanz) kann nur eine Instanz der Schicht N-1 darunter als Diener haben. Mit anderen Worten, eine N-Instanz kann nur den Dienst einer (N-1)-Instanz in Anspruch nehmen.
- Genauso kann eine N-Instanz ihre Dienste nur einer Instanz der direkt darüber liegenden Schicht anbieten, also einer (N+1)-Instanz.

Bevor wir jedoch diskutieren, wie zwei verschiedene Rechner über das Netz miteinander reden, sollen im folgenden Kapitel erst einmal die Aufgaben der 7 Schichten erläutert werden.

### 1.3 Schichtaufgaben

[Quelle: Dieter Conrads, Datenkommunikation, Vieweg]

Gehen wir der Einfachheit halber von der historischen Situation aus (siehe nächstes Bild). Hierzu muß im Vorgriff gesagt werden, daß ein Rechner einer DEE alle 7 Schichten im Kommunikationssystem hat. Die Aufgabe einer DÜE ist zu übermitteln. Wie wir noch sehen werden, hat sie nur die unteren 3 Schichten des ISO/OSI-Modells, da Schicht 3 bereits die Vermittlung enthält.

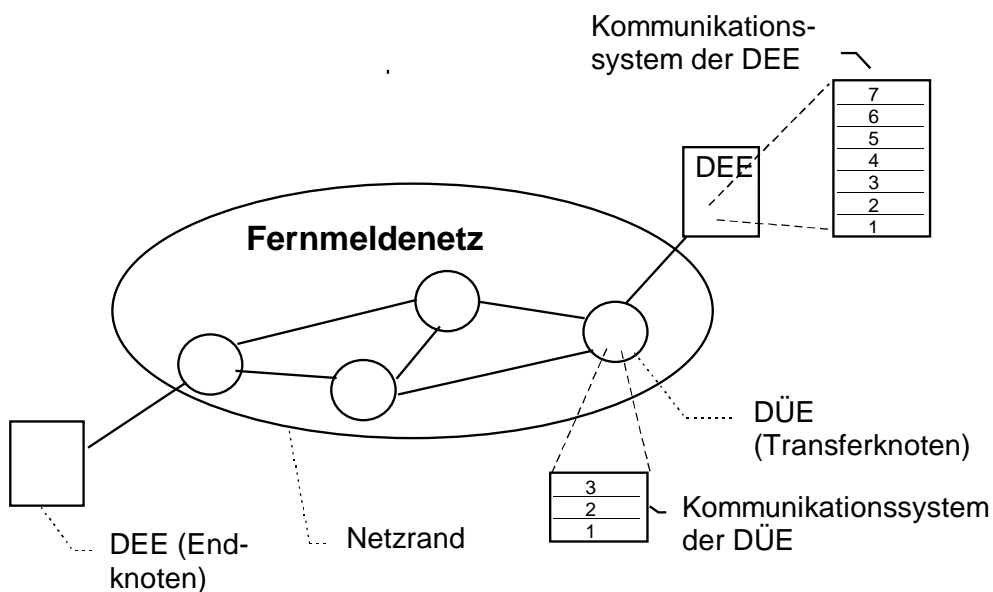


Bild 1.3 - 1 Schichten für DEE und DÜE. Eine DEE ist ein Endsystem am Netzwerkrand, das mit anderen Endsystemen über das Netz kommuniziert. Die DÜE wird auch als Transitsystem oder Relais bezeichnet.

Das folgende Bild 1.3 - 2 zeigt einen Schnitt durch ein Netz mit nur einem Relais. Dargestellt sind

- das Kommunikationssystem einer DEE (linke DEE),
- das Kommunikationssystem der mit dieser DEE kommunizierenden DEE (rechte DEE),
- das Kommunikationssystem des Transitsystems (zwischen beiden DEEs)
- das Übertragungsmedium zwischen der linken DEE und dem Transitsystem
- und das Übertragungsmedium zwischen der rechten DEE und dem Transitsystem.

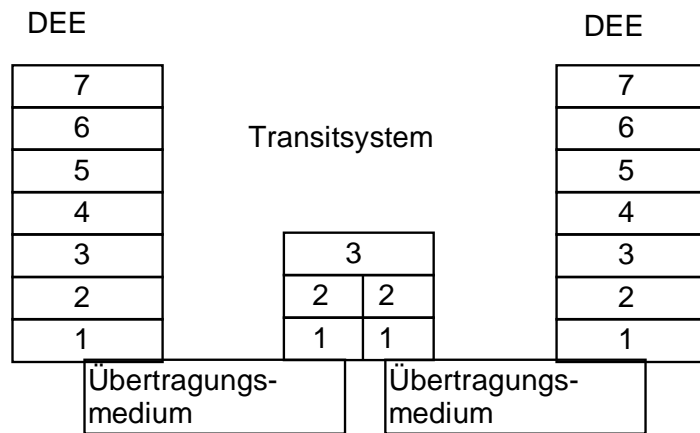


Bild 1.3 - 2 Ein Transitsystem (Relais) hat 3 OSI-Schichten, ein Endsystem hat 7 Schichten

Die Aufgaben der unteren 4 Schichten sind leicht verständlich. Mit den Aufgaben der oberen Schichten tut man sich etwas schwerer. Die Aufgaben der sieben Schichten sind folgende :

### 1.3.1 Bitübertragungsschicht (Schicht 1), physical layer

Die Schicht 1 beschreibt die physikalische Schnittstelle zum Übertragungsmedium. Zu der Spezifikation einer Schnittstelle zu einer Leitung gehört beispielsweise:

- die Spezifikation der Art der Leitung,
- die Spezifikation von Kabeln und Steckern,
- die Spezifikation der Übertragungstechnik (Modulation, elektrische Darstellung der Bits (Leitungscodes))
- etc.

Das Übertragungsmedium selbst gehört nicht zur Schicht 1.

Die Bitübertragungsschicht stellt ungesicherte Verbindungen zwischen Systemen für die Übertragung von Bits zur Verfügung. Als Fehlerfall kann nur der Ausfall der Verbindung gemeldet werden.

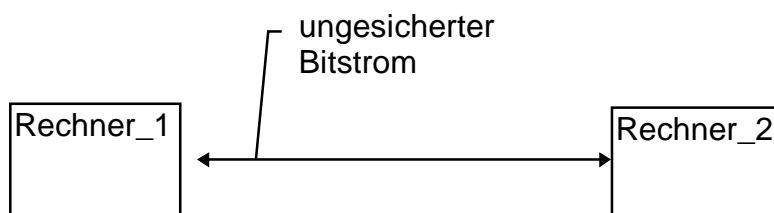


Bild 1.3.1 - 1 Ungesicherter Bitstrom zwischen benachbarten Rechnern. Rechner 1 sei etwa eine DEE, Rechner 2 eine DÜE

### 1.3.2 Sicherungsschicht (Schicht 2), data link layer

Aufgabe der Schicht 2 ist es, den Verkehr zwischen zwei benachbarten Stationen, also einer Punkt-zu-Punkt-Verbindung, zu regeln. Die Schicht 2 hat 2 Aufgaben:

## Flußkontrolle und Fehlerbehandlung.

### Flußkontrolle

Wenn zwei unterschiedlich leistungsfähige Einheiten miteinander kommunizieren, muß die leistungsfähigere die Sendegeschwindigkeit so weit herabsetzen, daß die leistungsschwächere Einheit in der Lage ist, die Daten aufzunehmen. Über das Aussenden von Bestätigungen kann eine empfangende Station den Datenfluß steuern und dafür sorgen, daß sie sich nicht verschluckt.

### Fehlerbehandlung

Die Informationen auf Schicht 2 sind kein kontinuierlicher Bitstrom, es sind Blöcke geeigneter Länge (frames oder Rahmen). Es ist nicht nötig, jeden Rahmen zu quittieren, sondern erst nach Empfang einer bestimmten Anzahl (Fenster-Technik: ein großes Fenster bedeutet, es wird erst nach einer großen Zahl von Messages quittiert, ein kleines Fenster bedeutet, es wird bereits nach einer kleinen Zahl von Messages quittiert.)

Ein Fehlercode in einem frame erlaubt die Fehlererkennung und -behebung (i.a. durch Wiederholung).

Bei lokalen Netzen erwies es sich als notwendig, die normalen Aufgaben der Schicht 2 in der Schicht 2b (Logical Link Control) unterzubringen. Schicht 2a wird benötigt für die Regelung des Zugriffs zum Medium (z.B. für das CSMA/CD-Verfahren im Falle von Ethernet). In lokalen Netzen hat man ja i.a. keine festen Punkt-zu-Punkt-Verbindungen zwischen direkt benachbarten Rechnern, da jede Station das Übertragungsmedium bekommen kann.



Bild 1.3.2 - 1 Gesicherter Bitstrom zwischen benachbarten Rechnern.  
Rechner 1 sei etwa eine DEE, Rechner 2 eine DÜE

Die Sicherungsschicht verbessert ungesicherte Verbindungen auf Teilstrecken zu gesicherten Verbindungen. Gesichert heißt dabei eine Verbindung, die Fehler in der Datenübertragung erkennt und korrigiert.

Mit anderen Worten, die Schicht 2 stellt eine gesicherte Punkt-zu-Punkt-Verbindung zwischen 2 Systemen zur Verfügung.

### 1.3.3 Vermittlungsschicht (Schicht 3), network layer

Die Vermittlungsschicht organisiert eine Ende-zu-Ende-Verbindung zwischen kommunizierenden Endsystemen.

Im Falle unseres Beispiels der Kommunikation zweier DEEs über ein Fernmeldenetz obliegt der Schicht 3 die Organisation der Ende-zu-Ende-Verbindung zwischen den beiden DEEs. Die Schicht 3 ist damit zuständig für die Wegewahl (das Routing), für das Multiplexen mehrerer Verbindungen über einzelne Teilstücke und für Aspekte der Fehlerbehandlung und Flußkontrolle zwischen den Endsystemen einer Verbindung. Mit anderen Worten, Schicht 3 kümmert sich um die Ende-zu-Ende-Verbindung zwischen kommunizierenden Rechnern und sorgt für den Transport von Paketen über die Teilstrecken des Netzes von Endsystem zu Endsystem.

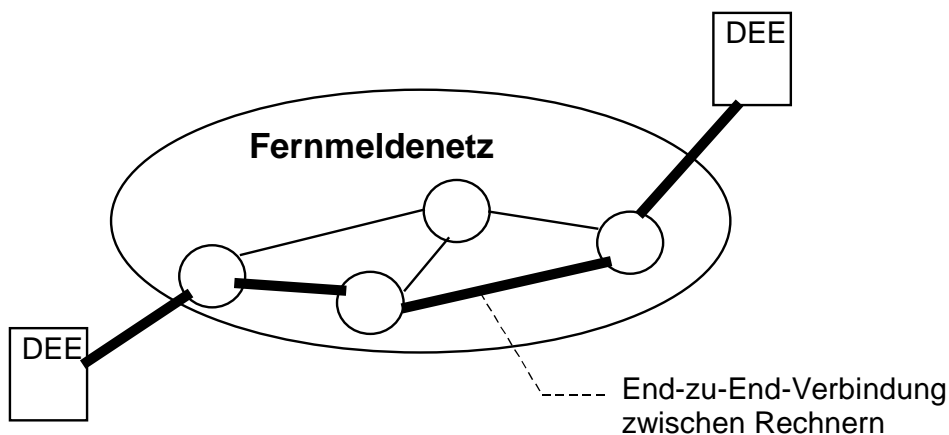


Bild 1.3.3 - 1 End-zu-End-Verbindungen zwischen Rechnern

Die Flußkontrolle auf der Schicht 3 macht die Flußkontrolle auf der Schicht 2 nicht überflüssig, da über eine Teilstrecke mehrere virtuelle Verbindungen, die zu verschiedenen End-zu-End-Verbindungen gehören, geführt werden können.

Die Fehlerbehandlung auf Schicht 3 bezieht sich nicht auf Übertragungsfehler - hierfür ist die Schicht 2 zuständig - sie bezieht sich auf Fehler, die beim Routing auftreten können. Hierzu gehört beispielsweise das Wiederherstellen der Sequenz, wenn Pakete in einer von der Sendefolge abweichenden Reihenfolge bei der Zielstation eintreffen.

Bei lokalen Netzen mit Datagramm-Dienst (siehe auch Kap. 2.2.1), bei denen also keine Verbindungen aufgebaut werden, ist die Schicht 3 praktisch funktionslos.

Es gibt weitere Ergänzungen zur Schicht 3, die das Internetworking zwischen verschiedenen Teilnetzen betreffen. Dies hat dazu geführt, daß die Schicht 3 in 3 horizontale Schichten neu eingeteilt wurde:

Schicht 3a (Subnetwork Access):

Wickelt die teilnetzspezifischen Protokolle (Routing etc.) ab.

Schicht 3c (Internet):

Wickelt die teilnetzunabhängigen Protokolle (Routing zu den Gateways zwischen Teilnetzen, falls Teilnetze verschiedene Protokolle fahren, globale Adressierung etc.) ab.

Schicht 3b (Subnet Enhancement):

ergänzt die Funktionen der Teilnetze so, daß die Anforderungen der Schicht 3c erfüllt werden.

Diese Einteilung kann an dieser Stelle nur erwähnt, nicht jedoch vertieft werden.

### 1.3.4 Transportschicht (Schicht 4), transport layer

Die Transportschicht befaßt sich mit End-zu-End-Verbindungen zwischen Prozessen in den End-Systemen.

Sie wissen ja, daß jedes Endsystem parallel ablauffähige Einheiten (Prozesse) enthalten kann. Ein System kann beispielsweise 4 Prozesse enthalten, von denen jeder eine Verbindung zu einem Partner-Prozeß in einem anderen Endsystem enthalten kann. Dabei ist es aber möglich, daß nicht alle 4 Verbindungen über dieselbe End-zu-End-Verbindung zwischen beiden Endsystemen gehen. Ein jeder Prozeß fordert nämlich einen Transportdienst einer gewissen Güte an. Der Transportdienst ist transparent. Er verbirgt, welches Netz zum Transport überhaupt zum Zuge kommt. Es kann durchaus möglich sein, daß es aus Performance-Gründen (um die Güte des angeforderten Transportdienstes zu erfüllen), notwendig sein kann, mehrere End-zu-End-Verbindungen von End-System zu End-System aufzubauen. Die Daten werden dann gesplittet, über mehrere Schicht-3-Verbindungen geleitet und beim Empfänger in der Schicht 4 wieder zusammengefügt.

Natürlich muß sich die Transportschicht auch mit der **Flußkontrolle zwischen den kommunizierenden Prozessen** befassen.

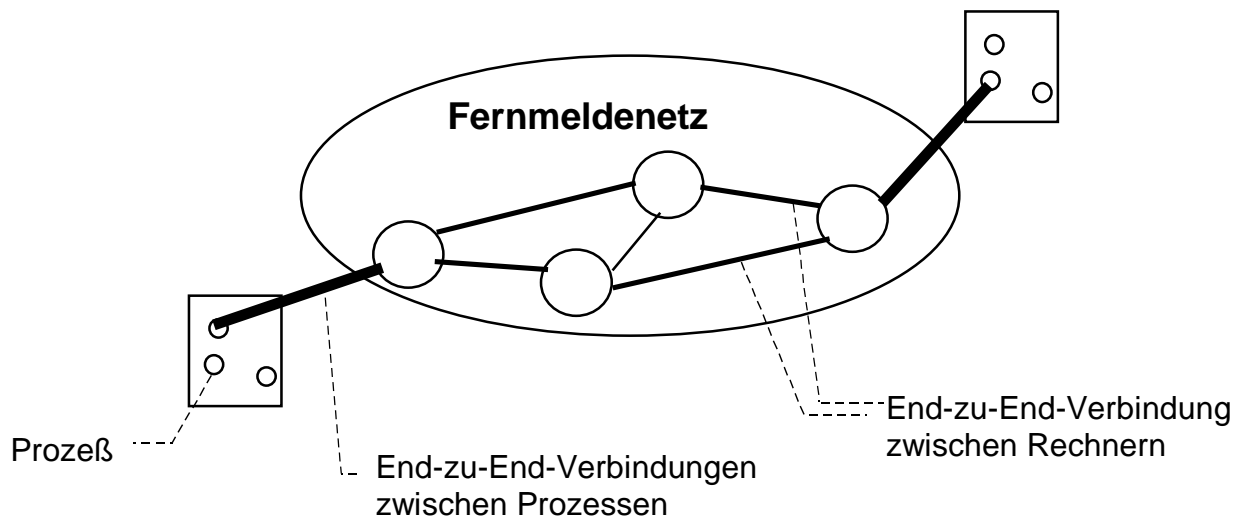


Bild 1.3.4 - 1 Die Transportverbindung **—————** zwischen zwei Prozessen kann auf mehrere Schicht 3-Verbindungen **—————** aufgesplittet werden.

### 1.3.5 Kommunikationssteuerungsschicht (Schicht 5), session layer

Die Kommunikationssteuerungsschicht regelt die Synchronisation zwischen Prozessen in verschiedenen Systemen.

Sie ermöglicht z.B. den Wechsel der Gesprächsrichtung, wenn die Anwendung dies erfordert (der Transportdienst darunter kann etwa halbduplex sein). Sie ist auch verantwortlich für eine Resynchronisation in Fehlerfällen. Hierzu braucht sie vereinbarte Punkte im Datenstrom (Synchronisationspunkte), auf die erneut aufgesetzt werden kann. An dieser Stelle ist es nicht möglich, hierauf genauer einzugehen.

### 1.3.6 Darstellungsschicht (Schicht 6), presentation layer

Verschiedene Rechner stellen Daten verschieden dar. So kann es sein, daß einer der an der Kommunikation beteiligten Rechner keinen Datentyp String hat, dafür aber ein Array von Zeichen. Erhält er nun ein String-Objekt, so kann er damit nichts anfangen. Es stellt für ihn einen Fehler dar. Damit brauchen zwei Rechner, die miteinander kommunizieren wollen, wenigstens dieselben Datentypen. Werden Werte übertragen, so muß im Datenpaket auch verschlüsselt sein, zu welchem Typ sie gehören, damit sie richtig interpretiert werden. Dies bedeutet, daß man für den Datentyp eine **Transfersyntax** braucht.

Für die Kommunikation zwischen den Rechnern gibt es nun ein „**Esperanto**“ für die Transfersyntax. **Teil 1 der Transfersyntax** betrifft die **Datentypen**. Die entsprechende „neutrale“ Syntax ist **ASN.1 (Abstract Syntax Notation 1)**. **Teil 2** des

Esperantos sind die **Basic Encoding Rules for ASN.1**, die einen genormten Satz von **Regeln zur Codierung der Datenwerte in die Transfersyntax** beinhalten, damit für einen übertragenen Wert auch erkenntlich wird, zu welchem Typ er gehört.

Mit Hilfe dieser **neutralen Transfersyntax ASN.1/BER** (BER für Basic Encoding Rules) können nun Rechner miteinander Daten austauschen. Der Sender wandelt vor der Datenübertragung seine lokale Syntax in die Transfersyntax. Der Empfänger wandelt wiederum die Transfersyntax in seine eigene lokale Syntax.

Kommunizierende Rechner können vereinbaren, daß sie als Transfersyntax nicht die neutrale Syntax nach ASN.1/BER nehmen, sondern eine Transfersyntax basierend auf der spezifischen lokalen Syntax, wenn beide diese spezifische lokale Syntax verstehen.

Die Aufgabe der Schicht 6 ist das **Aushandeln einer Transfersyntax**, die für beide Partner verständlich ist. Verstehen beide Rechner dieselbe lokale Syntax (z.B. Rechner desselben Herstellers) so können sie sich auf eine Transfersyntax auf Basis der lokalen Syntax einigen. Ist dies nicht der Fall, so müssen sie sich auf die neutrale Transfersyntax ASN.1/BER einigen. Die weiteren Aufgaben dieser Schicht sind die **Durchführung der Wandlung der lokalen Syntax in die Transfersyntax bzw. aus der Transfersyntax in die lokale Syntax**.

ASN.1 ist eine maschinenunabhängige Sprache für Datenstrukturen. ASN.1 ist eine formale Sprache mit einer eigenen Grammatik. ASN.1 ist die Netzwerk-Programmiersprache der neunziger Jahre. Die Basic Encoding Rules sind die Verpackungsregeln für die Verpackung der ASN.1-Werte in Nachrichten.

### 1.3.7 Anwendungsschicht (Schicht 7), application layer

Die Anwendungsschicht, die höchste Schicht des Referenzmodells, stellt die Schnittstelle der Anwendungsprozesse zum Kommunikationssystem dar. Es handelt sich hierbei um allgemeine Hilfsdienste für die Kommunikation oder aber spezielle Kommunikationsdienste wie File Transfer, Message Handling System (MHS), etc.

Der Anwendungsprozeß hat ausschließlich über die Schicht 7 Zugang zum OSI-Kommunikationssystem.

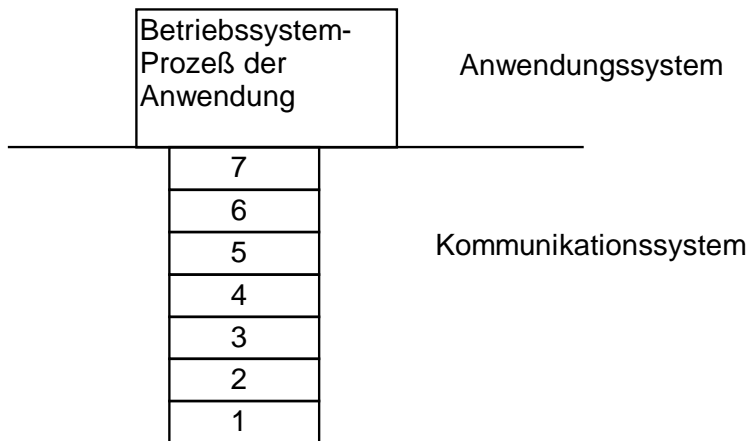


Bild 1.3.7 - 1 Ein Anwendungsprozeß (hier: ein Betriebssystem-Prozeß der Anwendung) kann auf das Kommunikationssystem nur über die Dienste der Schicht 7 des Kommunikationssystems zugreifen

## 2 Struktur des OSI-Modelles

[ Quelle : Helmut Kerner - Rechnernetze nach OSI ]

### 2.1 Instanzen

OSI zerlegt das Kommunikationssystem in 7 Schichten. In der OSI-Terminologie treten Stationen, Rechner, Prozesse oder Benutzer nicht auf, sondern man spricht von **entities (Instanzen)**.

In jeder Schicht eines jeden offenen Teilsystems existieren **Arbeitseinheiten (Instanzen, entities)**, die sowohl (vertikal) mit darüber- und darunterliegenden Instanzen als auch (horizontal) mit räumlich getrennten Instanzen derselben Schicht verkehrt. Diese **Instanzen (entities)** stellen logische parallele Prozesse in einer Schicht dar, welche die Aufgaben der jeweiligen Schicht durchführen können. Eine Schicht eines Endsystems kann mehrere Instanzen enthalten. Die Implementierung einer Instanz kann in Hard- oder Software erfolgen.

Zwei **vertikal benachbarte Instanzen nennt man obere und untere Folgeinstanz**. Korrespondierende Instanzen derselben Schicht in verschiedenen Systemen werden **Partnerinstanzen (peer entities)** genannt.

Aus dieser Struktur ergeben sich zwei Beziehungen zwischen Instanzen:

- eine **vertikale Beziehung, Dienst** genannt,
- und eine **horizontale Beziehung**, als **Protokoll** bezeichnet.

Jede Schicht N oder exakter jede Instanz in einer Schicht N bietet der übergeordneten Schicht N+1 **Services (Dienste)** an. Die übergeordnete Schicht N+1 ist der Nutzer der Dienste, die an der nächst tieferen Schicht N an den **Service Access Points (Dienstzugangspunkten)** zur Verfügung gestellt werden und durch alle Schichten, die unter der nutzenden Schicht liegen, erbracht werden. Die aufrufende Schicht - der Nutzer - heißt **Service User (Dienstnutzer)** und die genutzte Schicht heißt **Service Provider (Diensterbringer)**.

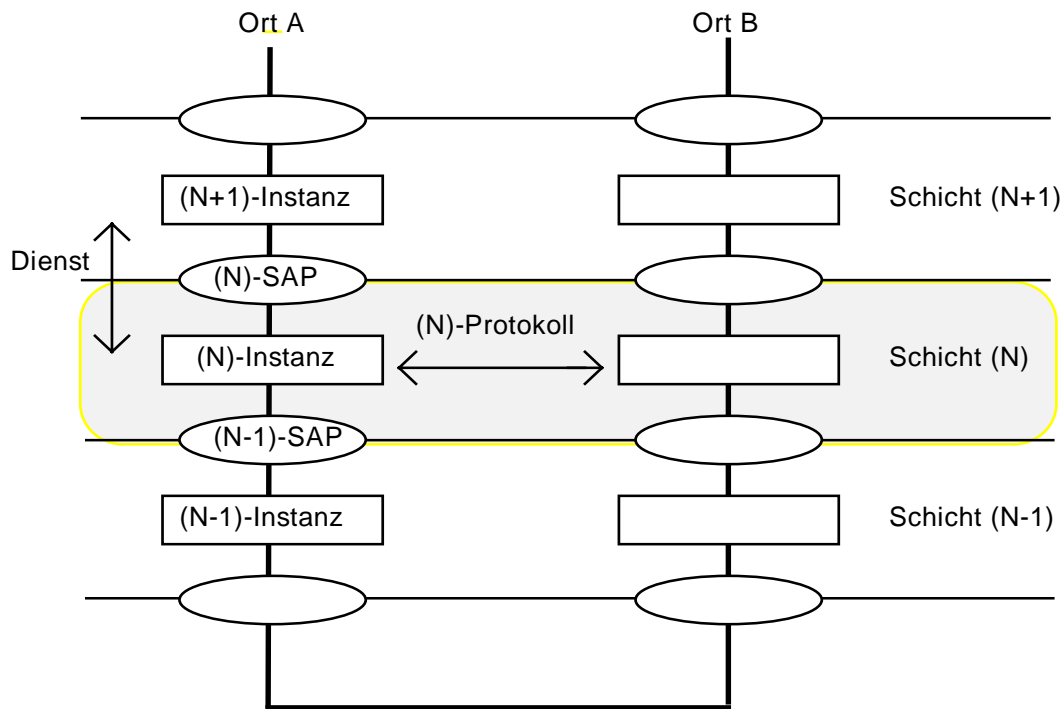


Bild 2. - 1 : Dienstzugangspunkte (Service Access Points), Instanzen und N-Schicht-Protokolle

## 2.2 Dienste

[Quelle : T. Rose - Verwaltung von TCP/IP-Netzen]

Eine Schicht N liefert der darüberliegenden Schicht N+1 eine eigene Leistung, sowie die Leistungen aller darunterliegenden Schichten. Diese Leistung nennt OSI den **Dienst (service)** der Schicht. Es besteht z.B. die Leistung der Netzwerkschicht in der Vermittlung zwischen mehreren Orten und (mittels der Leistung der darunterliegenden Schicht) in der Erkennung und Korrektur von Fehlern in der Datenübertragung. **Die Kommandos zur Inanspruchnahme von Diensten und die Ergebnismeldungen heißen Dienstelemente (service primitives)**. Ein Dienst stellt hierbei einen Satz von Funktionen dar, die einem Benutzer von einem Anbieter (Provider) zur Verfügung gestellt werden.

Die Eigenschaften eines Dienstes werden durch die Schnittstelle am **SAP (service access point)** definiert. Eine N-Schicht nimmt also die Dienste einer (N-1)-Schicht in Anspruch, ohne zu wissen, wie diese ihre Dienstleistungen erbringt (Prinzip des Information Hiding). Genauso stellt eine N-Schicht der (N+1)-Schicht ihre Dienste zur Verfügung.

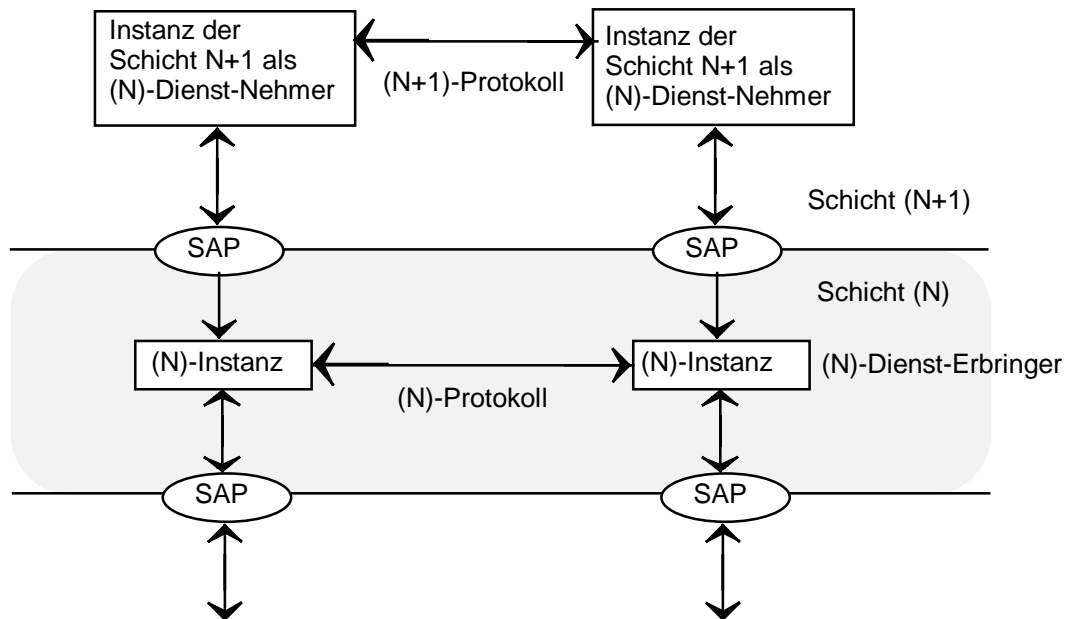


Bild 2.2 - 1 : Schichtung von Diensten

Obwohl der Zwei-Benutzer-Dienst die am meisten vorkommende Art von Dienst ist, gibt es auch andere mögliche Dienste, z.B. erhalten bei einem **Multicast (Gruppendienst)** mehrere Empfänger Daten gleichzeitig, die von einem Benutzer gesendet werden.

Sowohl für die Kommunikationsprotokolle der einzelnen Schichten des Referenzmodells als auch für die von den Schichten erbrachten Dienste gibt es internationale Normen, die z.T. noch in der Entwicklung sind. Dienstnormen beschreiben die elementaren Interaktionen zwischen einem Benutzer und dem Erbringer des Dienstes an **Dienstzugangspunkten (SAP)**. Sie werden beschrieben in Form von **Dienstelementen (Primitives)**, die logisch als unteilbare Ereignisse angesehen werden. Dabei werden von OSI nur die Dienstelemente beschrieben, die sich auf die Kommunikation von Dienstleister und Dienstbenutzer beziehen.

### 2.2.1 Verbindungslose Dienste

Der verbindungslose Dienst ist gekennzeichnet durch die Übertragung beziehungsloser Dateneinheiten, sogenannten Datagrammen, von einem Quell-SAP zu einem oder mehreren Ziel-SAP's.

Da keine Verbindung aufgebaut wird, ist der Dienstleister 'ohne Gedächtnis' über die Kommunikationsverbindung, es folgt daher keine Reihenfolgeprüfung und keine Fehlersicherung.

Verbindungslose Dienste gibt es in lokalen Netzen wie z.B. Ethernet, wo einfach Nachrichten abgefeuert werden, ohne daß eine Verbindung zwischen zwei Partnern aufgebaut wurde. Verbindungslose Dienste gibt es auch in Wide Area Networks. So gibt es beispielsweise bei DATEX-P einen Datagramm-Dienst. Hier werden auch

Nachrichten abgeschickt, ohne daß erst eine Verbindung aufgebaut wurde. Diese Nachrichten enthalten neben den eigentlichen Daten den Absender und den Adressaten. Sie werden durch das Netz geschleust und beim Empfänger abgeliefert. Hierbei können auch Datagramme verloren gehen, das Netz bietet keine Sicherheit.

### 2.2.2 Verbindungsorientierte Dienste

Eine verbindungsorientierte Datenübertragung ist durch folgende Merkmale gekennzeichnet:

- Eindeutig bestimmte Lebensdauer  
(Verbindungsaufbau, Datentransfer, Verbindungsabbau)
- Verbindungskennung  
(Eine N-Verbindungsendpunktennung wird auf beiden Seiten eines N-Dienstzugangspunkts (SAP) zur Identifikation der N-Verbindung benutzt.)
- Dreiseitige Übereinkunft. Sie umfaßt Verhandlung zwischen:
  - a) (N+1)-Instanz im System A und (N+1)-Instanz im System B
  - b) (N+1)-Instanz im System A und (N)-Instanz im System A
  - c) (N)-Instanz im System B und (N+1)-Instanz im System B
- Verhandlungen zwischen Instanzen. Verhandlungsgegenstände sind:
  - a) notwendige Betriebsmittel (Puffer)
  - b) Dienstgüteparameter (Durchsatz, Fehlerrate,...)
 etc.

Bewertung von verbindungslosen Protokollen mittels Datagrammen für die Kommunikation über Fernmeldenetze:

- + einfaches Verfahren, kein Verbindungsaufbau erforderlich
- höhere Belastung der Vermittlung
- höherer Aufwand in DEE für Resequencing

Es ist möglich, daß auf einer datagramm-orientierten Netzverbindung (häufig bei LANs) ein verbindungsorientierter Dienst bereit gestellt wird. In der Transportschicht gibt es fünf Dienstklassen 0 bis 4. Die Klasse 4 stellt einen solchen verbindungsorientierten Dienst bereit.

Ein anderes Beispiel sei der TCP/IP-Architektur entnommen: Über der Schnittstellenschicht und Internet-Schicht gibt es 2 Transportprotokolle, das verbindungslose User Datagram Protocol (UDP) und das verbindungsorientierte Transmission Control Protocol (TCP). Somit kann über Ethernet in der Schnittstellenschicht mittels TCP ein verbindungsorientierter Dienst aufgebaut werden.

TCP	UDP	Transportschicht
IP		Internetschicht
Ethernet		Schnittstellenschicht

Verbindungsorientierte Protokolle haben Mechanismen zur Erkennung fehlender Blöcke (Vollständigkeit nicht gewahrt), duplizierter Blöcke (Eindeutigkeit nicht gewahrt) und außerhalb der Sequenz eintreffenden Informationsblöcke (Sequenz nicht gewahrt) sowie zur Beseitigung der entsprechenden Fehlerzustände. Sie gewährleisten also Vollständigkeit, Eindeutigkeit und Sequenz.

## 2.3 Dienstelemente

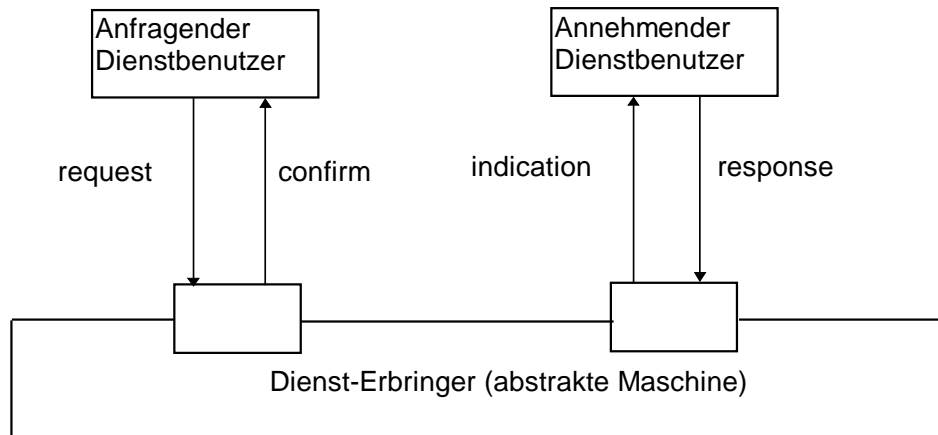


Bild 2.2.3 - 1 : Modell eines Dienstes

Es gibt vier verschiedene Typen von Dienstelementen, die gekennzeichnet werden durch die an den Namen angehängte Bezeichnung

.request  
.indication  
.response  
.confirm

Diese repräsentieren das **Anfordern (request)** eines Dienstes an einem Dienstzugangspunkt (SAP), die entsprechende **Anzeige (indication)** an dem Partner-Dienstzugangspunkt, die dortige **Antwort (response)** auf eine Anzeige und die **Bestätigung (confirm)** bei dem anfordernden Dienstzugangspunkt.

Die Dienstbezeichnung selbst besteht aus einem Kennbuchstaben für die Schicht:

P : presentation  
S : session  
T : transport  
N : network  
DL: data link

und einem Namen für die Dienstgruppe, z.B.

CONNECT	für Verbindungsaufbau
DISCONNECT	für Verbindungsabbau
DATA	für Datentransport

Beispielsweise beschreiben die vier Dienstelemente

T-CONNECT.request  
T-CONNECT.indication  
T-CONNECT.response  
T-CONNECT.confirm

den Aufbau einer Transportverbindung zwischen zwei Teilnehmern.

Diese Typen von Dienstelementen sind voneinander abhängig, d.h. sie treten in bestimmtem zeitlichen Zusammenhang auf. Beispiele hierzu geben die Zeitablaufdiagramme ( Bild 2.2.3 - 2 und Bild 2.2.3 - 3 ). In diesen wird die zeitliche Folge der Ereignisse an den beiden Dienstzugangspunkten modelliert, indem diese Dienstzugangspunkte als nach unten gerichtete Zeitachsen dargestellt werden.

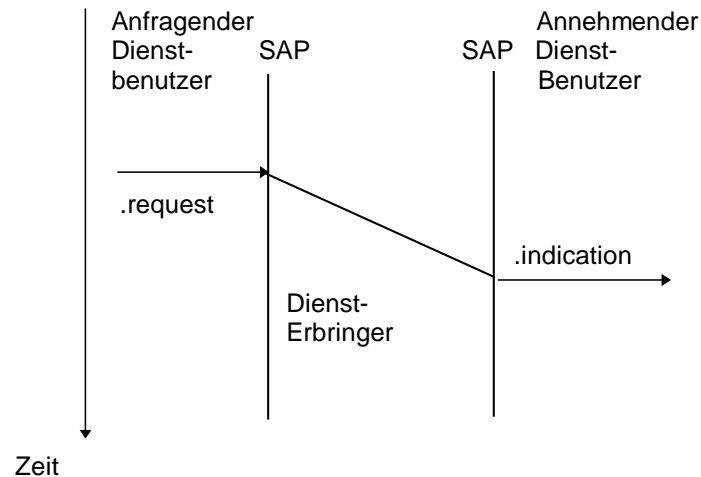


Bild 2.2.3 - 2 : Diensttyp 'unbestätigter Dienst'.

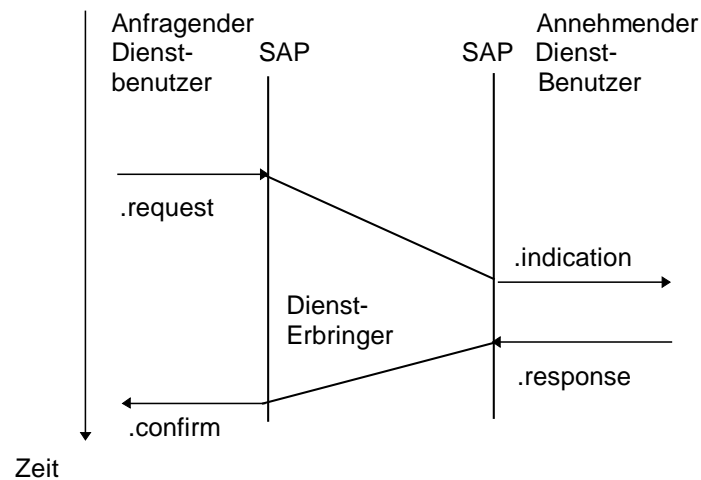


Bild 2.2.3 - 3 : Diensttyp 'bestätigter Dienst'

Diese Dienstelemente (Primitives) sind abstrakte Mittel der Dienstbereitstellung. Sie sind nicht identisch mit Schnittstelleneignissen einer konkreten Implementierung, bei der insbesondere die Einbettung in die vorliegende Betriebssystemumgebung vorzunehmen ist.

## 2.4 Protokolle

[ Quelle : Helmut Kerner - Rechnernetze nach OSI ]

Die zweite Art von Beziehungen sind die horizontalen Beziehungen zwischen Instanzen, **Protokolle (protocols)** genannt. Das Wort "Protokoll" erinnert an Verabredungen zweier diplomatischer Vertretungen für die Durchführung von Staatsakten. Protokolle erkennt man auch in der gegenseitigen Begrüßung, in der man kaum ein frohes "Guten Morgen" mit "Auf Wiedersehen" beantworten sollte, wenn man ein Gespräch mit dem Partner beginnen möchte.

Auch beim Telefonieren folgt man einem Protokoll: wird man gerufen, so darf man sich mit "Hallo" melden; antwortet der Anrufende ebenfalls mit "Hallo", so befindet sich das Protokoll in einer Endlosschleife und man kommt nie zum eigentlichen Gespräch.

Protokolle sind Regeln zur Kommunikation zwischen Partnerinstanzen. In realen Systemen gibt es nur eine einzige physisch vorhandene horizontale Verbindung, nämlich jene durch das Übertragungsmedium. Da aber mittelbar - über einige vertikale Übergänge abwärts und dann über das Medium und jenseits des Mediums wieder aufwärts - auch eine Verbindung zwischen zwei **Partnerinstanzen (peer entities)** derselben Ebene existiert, kann man für die Kommunikation zwischen Partnerinstanzen diese als verbunden ansehen. Die Partnerinstanzen verkehren über diese Verbindung miteinander gemäß einem Protokoll ihrer Schicht.

Aufgaben von Protokollen:

Adressierung  
Flußsteuerung  
Fehlererkennung  
Fehlerbehebung

Adreßangaben  
Quittungen, Empfangsfenster  
Prüfsumme, Sequenznummer,  
Quittungen, Zeitüberwachung  
Paketwiederholung, Korrekturverfahren

## 2.5 Paketstrukturen

[ Quelle : Helmut Kerner - Rechnernetze nach OSI ]

Es treten zwei Arten von Nachrichten auf, vertikal zwischen Folgeinstanzen laufende Nachrichten und horizontal zwischen Protokollpartnern auszutauschende Nachrichten.

Vertikal werden Dienstaufträge bzw. Zustimmungen zwischen der dienenden (z.B. (N)) und der fordernden Schicht (N+1) ausgetauscht. Sie bestehen aus einem **Kommandoteil, Interface Control Information, ICI** genannt, und aus **Nutzdaten (Userdaten)**. Die Nutzdaten, welche der Schicht (N) übergeben werden, heißen **(N)-Service Data Unit, (N)-SDU**. Das übergebene Gesamtpaket (Kommandoteil plus Nutzdaten) heißt **Dienstelement (service primitive)**.

Die Nutzdaten (user data), welche die (N+1)-Instanz ihrem Partner am anderen Ort senden will, muß sie der unteren Folgeinstanz im eigenen System zur Beförderung übergeben. Sie tut dies im (N)-SDU Feld des Übergabepaketes. Diese Information soll von der (N)-Instanz unverändert an ihren (N)-Partner und weiter zur (N+1)-Instanz befördert werden. Man bezeichnet die Übertragung daher als **transparent**, d.h. verborgen, da keine Information für die (N)-Instanzen darin enthalten ist.

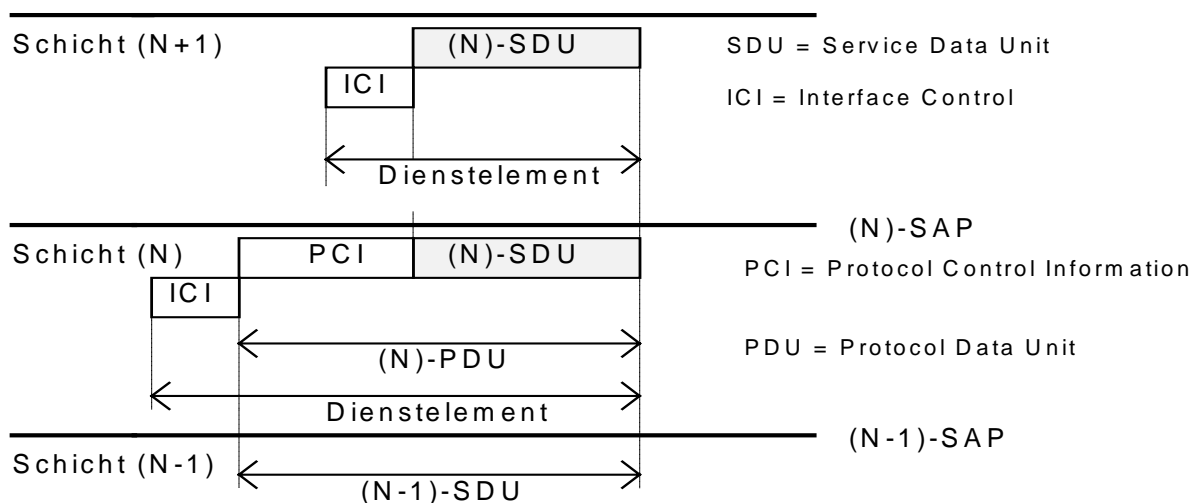


Bild 2.5 - 1 : Dateneinheiten aus Sicht der Schicht (N)

Die für die (N)-Instanz bestimmte Steuerinformation ist davon getrennt in der ICI (Interface Control Information) gelagert. Dort befindet sich das Kommando an die Folgeschicht sowie weitere Steuerinformationen, wie z.B. der Zielort.

Die Inhalte und das Format der ICIs sind nicht genormt, da sie nur Übergänge innerhalb eines Systems betreffen und daher dem Hersteller des Systems überlassen bleiben. Die (N)-Instanz stellt nun jenes Informationspaket zusammen, das sein Protokollpartner (horizontal) erhalten soll, man nennt es

**Protokoll dateneinheit, ((N)-Protocol Data Unit, (N)-PDU).** Es enthält in einem Feld unverändert die als (N)-SDU von oben übernommenen Daten (User data - Nutzdaten) und fügt noch Steueranweisungen an seinen Protokollpartner in einem eigenen Feld, dem **PCI-Feld (Protocol Control Information)** hinzu. Manche PDUs dienen nur reinen Steuerzwecken. Diese PDUs enthalten nur PCI-Informationen und in der Regel keine Userdaten (SDU).

Die (N)-PDU ist die Nutzinformation (Userdaten) der (N)-Schicht. Notgedrungen muß sie wieder an die nächst untere Folgeinstanz (N-1) zur Beförderung übergeben werden. Sie wird als (N-1)-SDU mit Steuerinformation (ICI) ergänzt und nach unten weiter gereicht.

Die eigentliche Nachricht wird in jeder Schicht mit einer Steuerinformation (PCI) für die Partnerinstanz versehen. In PDUs der untersten Schicht liegen somit 7 PCIs vor der ursprünglichen Nachricht. Nur die äußerste PCI ist für die unterste Schicht bestimmt. Den Vorgang könnte man damit vergleichen, daß ein Brief in ein Kuvert gesteckt wird, das wiederum in ein weiteres Kuvert verpackt wird, und das sieben mal. Jedes Kuvert enthält eine Anweisung für die Behandlung seines Inhalts.

Nachdem das Protokoll der Schicht 1 ausgeführt wurde, ist das äußerste Kuvert der Steuerinformation nicht mehr nötig. Bei jeder Weitergabe nach oben entfällt ein "PCI-Kuvert", bis zuletzt über der Schicht 7 nur noch die ursprüngliche Nachricht ankommt.

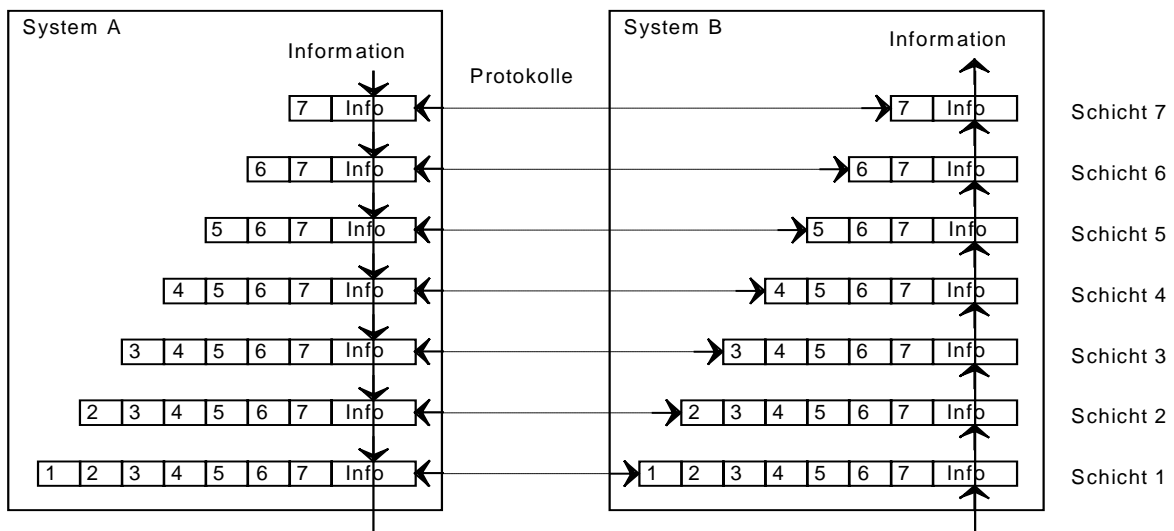


Bild 2.5 - 2 : Protokollweg durch den OSI-Turm

## 2.6 SAP-Adressierung

[ Quelle : Helmut Kerner - Rechnernetze nach OSI ]

Die Schnittstelle zwischen zwei Schichten, über welche Dienste angefordert und bereitgestellt werden, heißt **Dienstzugangspunkt (service access point, kurz SAP)**. Eine mögliche Implementierung eines SAP wäre ein Speicherbereich, auf den beide Folgeinstanzen zugreifen können und über den sie Dienstelemente austauschen können. Jeder SAP wird durch eine **Identifikation (SAPid)** gekennzeichnet.

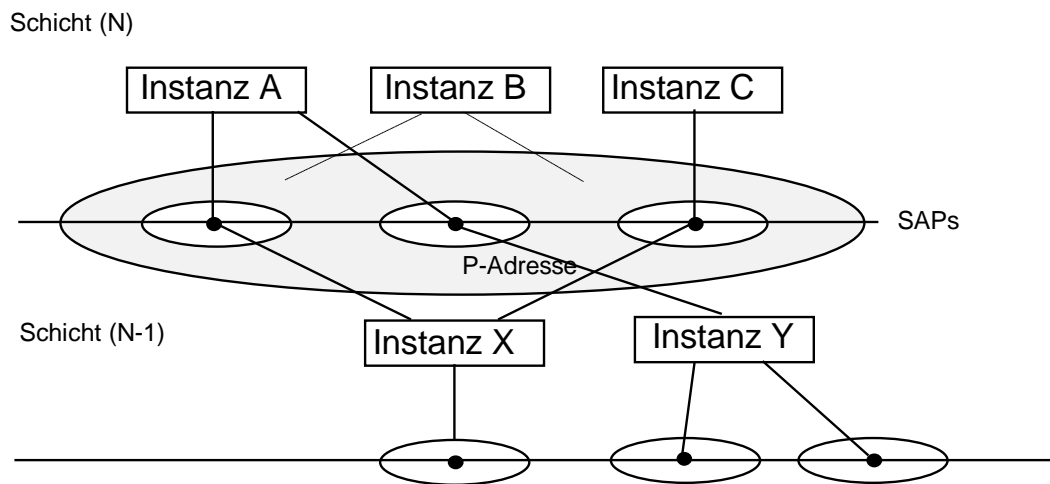


Bild 2.2.5 - 1 : Verzweigungen innerhalb eines Systems

Die Dienstzugangspunkte (SAPs) werden zur Kennzeichnung und Adressierung der Instanzen benötigt. Eine Adressierung von Instanzen ist offensichtlich notwendig, um den Ort und das offene System an diesem Ort zu kennzeichnen. Für die Vermittlungsschicht sind daher netzweit eindeutige Adressen der Instanzen dieser Schicht wichtig. Darüber hinaus müssen aber auch die Instanzen der anderen Schichten innerhalb eines jeden offenen Systems identifiziert werden, denn es gibt ja nicht nur eine einzige vertikale Säule von Folgeinstanzen innerhalb eines offenen Systems.

Eine Verzweigung nach unten ist möglich, wenn eine (N+1)-Instanz die Dienste verschiedener (N)-Instanzen beansprucht.

Man sieht, daß in einer Verzweigung ein SAP den Übergang zwischen genau zwei Folgeinstanzen darstellt, und daß der SAP diese Instanzen eindeutig bestimmt und sich somit zur Adressierung der Instanzen eignet. Da ein SAP zwischen zwei Instanzen steht, könnte er jeder der durch ihn verbundenen Schichten zugerechnet werden. Im OSI-Modell wird er zur darunterliegenden Schicht gezählt und entsprechend benannt. Eine Instanz kann mehreren SAPs zugeordnet werden. In diesem Fall stellt jeder der SAPs die Adresse für die (N)-Instanz dar.

Die eindeutige Zuordnung von SAP zu Instanz wird durch die **Regel zur Adressierung von Instanzen** gewährleistet :

"Es gibt keine Verzweigungen von einem SAP zu verschiedenen Instanzen"

Wohl aber gibt es Verzweigungen von einer Instanz zu verschiedenen SAPs. Zur Adressierung von Instanzen mittels SAPs ist ein netzweites, eindeutiges Schema zur Kennzeichnung von SAPs notwendig.

Man unterscheidet mnemonische Adressen mit einer Kurzbezeichnung der Instanz, die man **Namen (titles)** nennt, von den Adressen im engeren Sinne, d.h. SAP-Identifikationen. (Hinweis: mnemonisch bedeutet, daß eine symbolische Bezeichnung verwendet wird, die einen Hinweis auf die Bedeutung gibt). Namen werden vorwiegend für Anwendungsinstanzen verwendet. Sie werden durch **Namensverzeichnisse (directories)** in Adressen umgewandelt, wie man es vom Telefonbuch her kennt. Im allgemeinen muß ein Weg durch das Endsystem durch eine Liste von sieben Instanzadressen bezeichnet werden, wobei die Adresse der Instanz der Schicht 3 den Ort im Netz bezeichnet. Im Falle einer nur nach oben verzweigten Baumstruktur läßt sich die Adressierung aller Instanzen eines Wegs im Endsystem dadurch vereinfachen, daß die Adresse einer Instanz einer höheren Schicht durch Anfügen einer **Kennung (Präfix)** für die höhere Schicht an die Adresse der darunterliegenden Instanz gewonnen wird.

Eine Adresse sieht dann folgendermaßen aus:

Schicht 6 :	6666.55.444.33.22.1
Schicht 5 :	55.444.33.22.1

Hierbei muß nur die Adresse der Schicht 3 netzweit eindeutig sein, um jede Instanz netzweit identifizieren zu können.

## 2.7 Diskussion des OSI-Modells

[Quelle: H.W. Barz: Kommunikation und Computernetze]

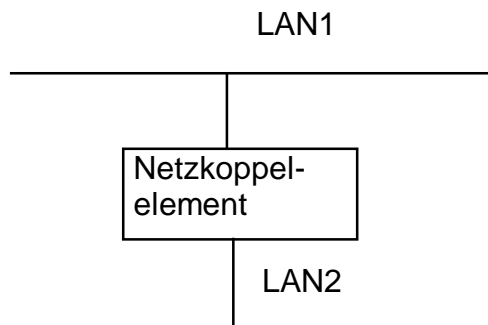
Das ISO-Modell spezifiziert nur die Protokolle zwischen zwei Entities verschiedener Systeme und die Funktionalität an einem Service Access Point (SAP).

Für die Schnittstellen zwischen den einzelnen Schichten (Layers) werden keine Implementierungsvorschriften gemacht. Das bedeutet, daß jeder Hersteller die Schnittstelle anders realisieren kann. Der Hersteller muß die Schnittstellen zwischen den Schichten nicht offen legen.

Dies bedeutet, daß man die Software für einen Rechner i.a. nicht schichtenweise bei verschiedenen Herstellern kaufen kann, sondern daß man in der Regel stets einen gesamten Protokollstack für alle 7 Schichten bei einem Hersteller kaufen muß.

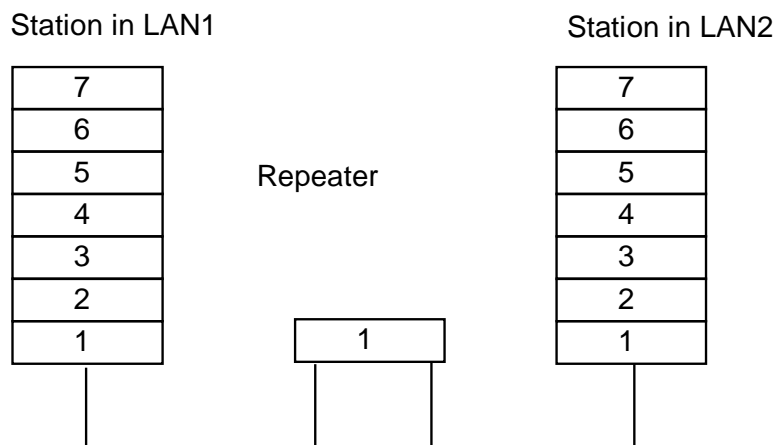
### 3 Repeater, Bridges und Router in Lokalen Netzen

Repeater, Bridges und Router dienen zur Kopplung lokaler Netze. Sie sind Stationen an zwei Netzen und stellen die Verbindung her. Sie stellen als Netzkoppelemente dar.



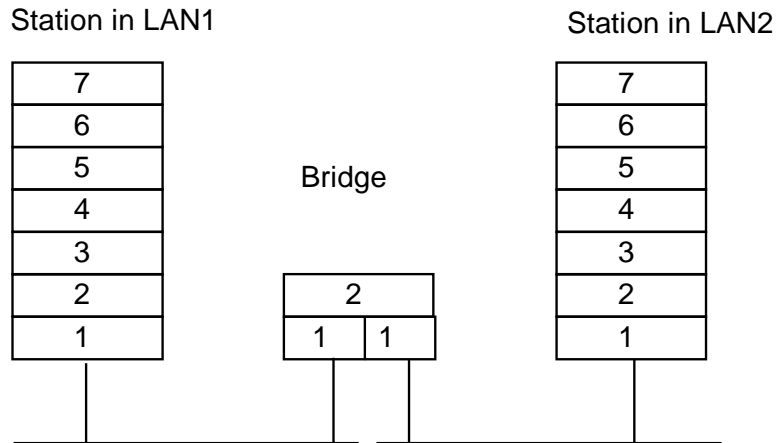
#### Repeater

Der Repeater ist ein Verstärker. Er hat nur mit Bitfolgen (Schicht 1) zu tun. Die LANs auf beiden Seiten müssen gleich sein: Die Schichten 1-7 müssen in den Stationen der beiden Netze identisch sein, z.B. links und rechts des Repeaters Ethernet Koaxialkabel.



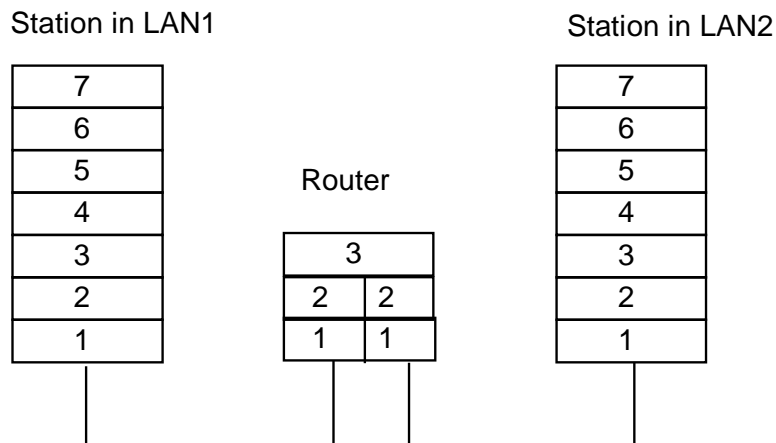
## Bridge

Die Schichten 2-7 müssen in den Stationen der beiden Netze identisch sein. z.B. links und rechts der Bridge Ethernet, aber links in Schicht 1 Koaxialkabel, rechts Twisted Pair. Eine Bridge kann ferner eingesetzt werden zur Adreßfilterung.



Repeater und Bridge sind transparent. Sie müssen nicht vom Absender direkt angesprochen werden, damit sie Pakete weiterleiten.

## Router



Man kann mit einem Router LANs mit unterschiedlichen Schichten 1 und 2 z.B. Token Ring und Ethernet verbinden. Die Schichten 3 bis 7 der Stationen an den beiden LANs müssen identisch sein. Dies bedeutet, sie haben denselben Adressierungsmodus in der Schicht 3.

Ein Router ist nicht transparent. Er muß vom Anwender direkt angesprochen werden, damit er Pakete weiterleitet.