1.7 Grundlagen: Schichtenmodelle und Protokolle

Version: Feb. 2003

- OSI Referenzmodell
- Erweiterte Referenzmodelle
- Protokollinstanzen und -dienste
- Dienstelemente
- Protokollmechanismen

Das ISO/OSI-Modell ist als **Referenzmodell** weltweit akzeptiert. Es hilft bei der Einordnung, der Diskussion und dem Verständnis von Kommunikationssystemen. Die Anzahl von sieben Schichten hat sich im Lauf der Entwicklung so ergeben. Manchmal wird kritisiert, dass die Zahl der Schichten zu groß und der Ablauf unnötig kompliziert sei. Es gibt deshalb Modelle, die mit weniger Schichten auskommen. Ferner wird bei der Implementierung der Protokolle und Dienste die starre Schichteneinteilung teilweise ignoriert, um den Implementierungsaufwand gering zu halten und eine brauchbare Leistung (gemessen am erreichten Durchsatz für Anwendungsdaten) zu gewährleisten. Das TCP/IP-Modell fasst zum Beispiel die beiden unteren Schichten 1 und 2 sowie die Schichten 5 - 7 zusammen. Damit werden insgesamt 4 Schichten unterschieden. Der TCP/IP-Protokollstapel ist jedoch schon älter als das OSI-Modell. Manchmal werden auch Unterschichten eingeführt, wie zum Beispiel bei lokalen Netzen (LANs), wo Schicht 2 auch den Schichten 2a, Medium Access Control (MAC) und Schicht 2b, Logical Link Control (LLC) besteht. Um mehrere Übertragungsmedien (d.h. verschiedene Arten von Kupfer- und Koaxialleitungen sowie verschiedenen Glasfasern) nutzen zu können, wird heute auch die Bitübertragungsschicht in zwei Subschichten aufgeteilt und zwar in einen gemeinsamen Teil und in einen mediumabhängigen Teil.

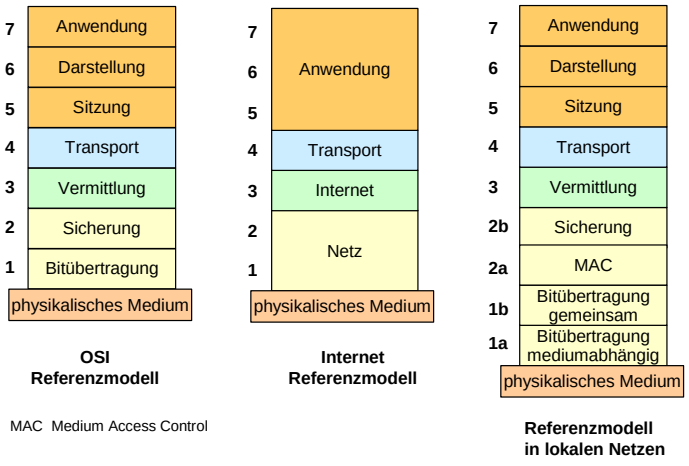


Bild: Protokoll-Referenzmodelle

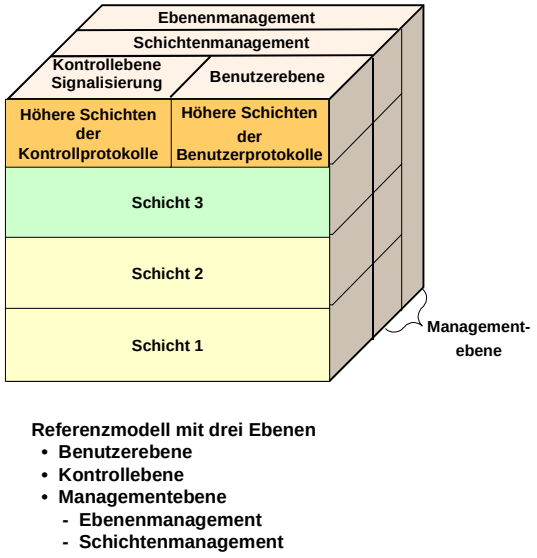


Bild: Protokoll-Stapel mit Schichten und Ebenen

Die im OSI-Modell definierten Schichten beziehen sich auf die eigentliche Informationsübertragung. Für bestimmte Aufgaben wird jedoch eine Signalisierung oder Kontrolle (control) benötigt, die an der eigentlichen Informationsübertragung nicht beteiligt ist. Damit ein Datennetz zuverlässig und mit guter Leistung funktioniert, muss es laufend überwacht werden. Aus der Analyse der Messdaten werden Maßnahmen zum Management des Netzes abgeleitet. Diese werden den betroffenen Netzknoten übermittelt und dort ausgeführt. Um diese Aufgaben einzuordnen, sind die Schichten des Schichtenmodells durch orthogonal dazu liegende Ebenen (planes) ergänzt. In der Regel werden drei Ebenen unterschieden, die ihrerseits in Schichten eingeteilt sind.

Die Aufgaben eines Protokolls innerhalb einer Schicht n hängen von der betreffenden Schicht ab. In vielen Fällen sind dies:

- Auf- und Abbau von Verbindungen der Schicht n,
- Festlegung (Aushandlung) von Verbindungsparametern,
- Reihenfolgesicherung der übertragenen Dateneinheiten,
- Quittierung von richtig empfangenen Dateneinheiten,
- Fehlererkennung (z. B. fehlende Dateneinheiten),
- Fehlerbehebung (z. B. durch wiederholte Übertragung),
- Zeitüberwachung zur Erkennung und Behebung von Verklemmungszuständen (deadlock),
- Datenflusssteuerung zur Anpassung der sendenden Instanz an die empfangende Instanz,
- Kommunikation mit den benachbarten Schichten (n-1) bzw. (n+1) über das Adjacent Layer Protokoll (Dienstprotokoll).

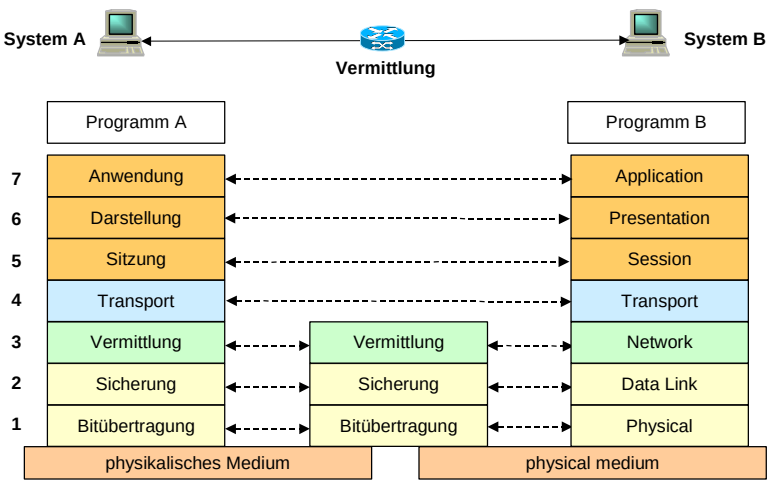


Bild: OSI-Referenzmodell über Zwischensystemen

Die Anzahl der Protokoll-Schichten

Die Protokoll-Ebenen beziehen sich teilweise auf unterschiedliche Abschnitte eines Übertragungsweges. Protokolle der Benutzererebene steuern die Abläufe zwischen Endknoten, während Protokolle der Steuerungsebene primär zwischen einem Endsystem und dem ersten Zwischensystem im Netz wirksam werden.
Managementprotokolle werden für die Kommunikation von Managementsystemen mit Zwischensystemen benötigt.

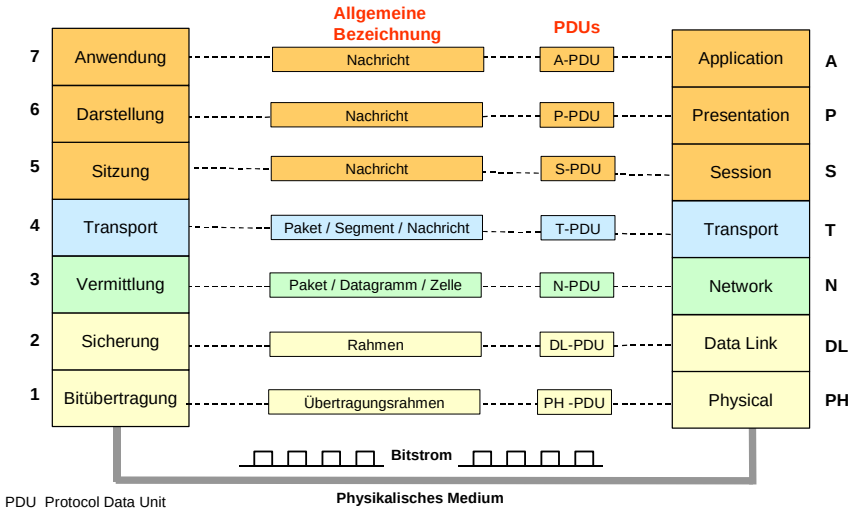
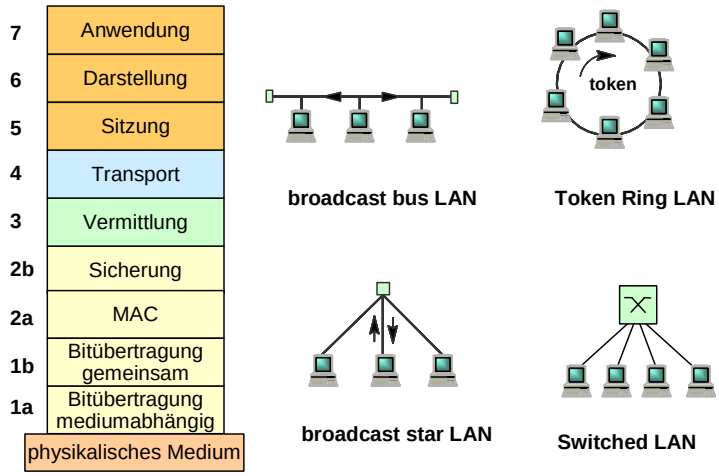


Bild: Übermittlungs- und Übertragungseinheiten



MAC Medium Access Control

Bild: Referenzmodell in lokalen Netzen

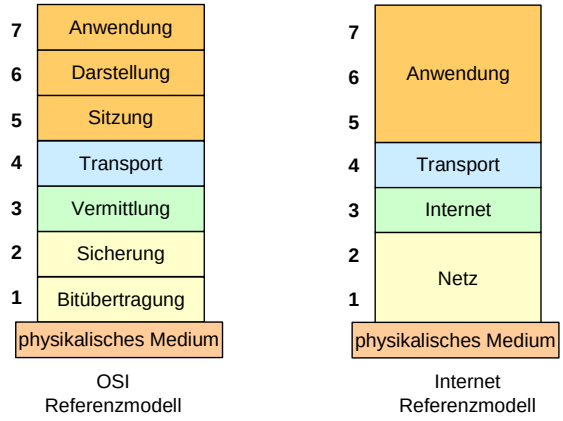


Bild: OSI- und Internet Referenzmodelle

Die technische Entwicklung hat eine Erweiterung des prinzipiellen 7-Schichten-Modells erforderlich gemacht durch Einführung von Leerschichten für Null-Funktionalitäten in bestimmten Anwendungsfällen

- Subschichten für eine feinere Unterteilung der Funktionalität wie z. B.

in Schicht 1 bei ATM-Netzen
in Schicht 2 bei LAN/MAN-Netzen
in Schicht 3 bei LAN/MAN-Netzen
in Schicht 7 für unterschiedliche Anwendungsklassen

Das Grundsätzliche das Schichtenprinzips bleibt jedoch erhalten

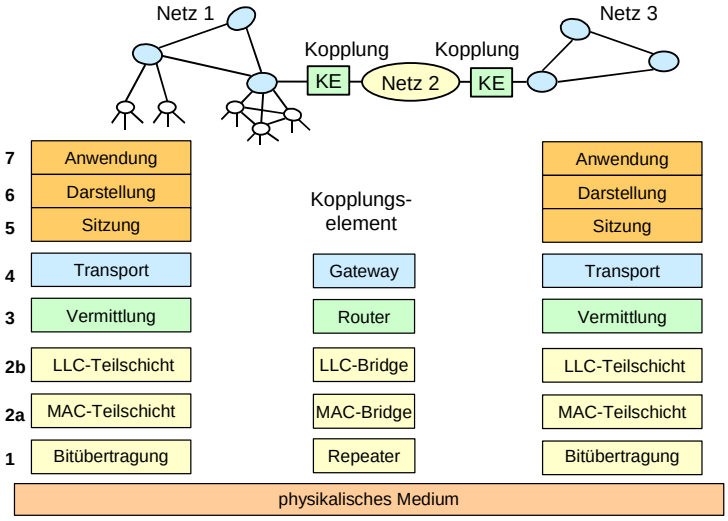


Bild: Gekoppelte Netze

Internetworking
Ein Internet ist ein Netz, das aus Subnetzen (Teilnetzen) besteht, die untereinander vernetzt sind. Die Verbindung zwischen Subnetzen wird durch Zwischensysteme (intermediate system) hergestellt. Die in den Subnetzen vorhandenen Systeme werden hingegen als Endsystem (end system) bezeichnet, sie bieten Dienste für die Netznutzer an.

Bemerkung: Der Begriff Internet ist ein generischer Begriff. Das Internet ist ebenfalls ein Internet, aber eine spezifische Implementierung, die als Kernprotokolle TCP/IP verwendet.

Zu Beginn der Rechnervernetzung wurden häufig zuerst lokale Netze aufgebaut. Diese können nach recht unterschiedlichen Konzepten funktionieren, weshalb sie sich in ihren Eigenschaften erheblich unterscheiden können. Später sollte das Internetworking existierende verschiedenartige Netzinseln miteinander vernetzen. Ziel war es, die Anzahl der erreichbaren Kommunikationspartner zu vergrößern und so den (betriebswirtschaftlichen) Nutzen aus den getätigten Investitionen zu steigern.

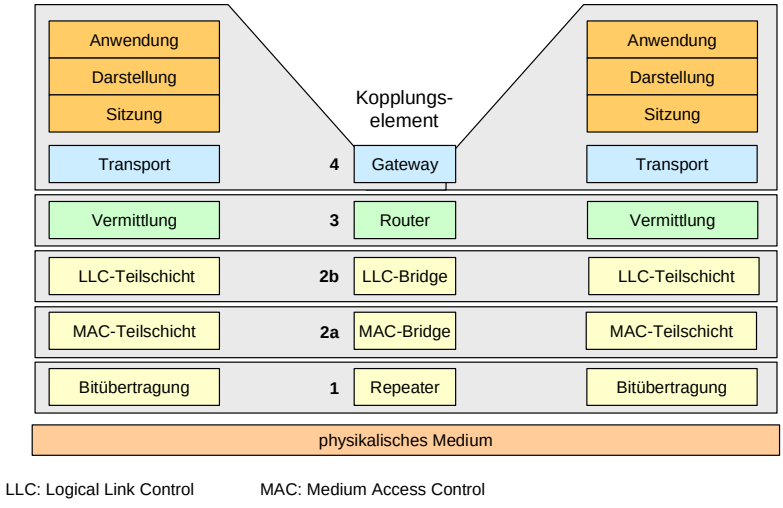


Bild: LAN-Kopplung: Schichtenmodell

Ein **Internet** kann als ein **virtuelles Netz** bezeichnet werden. Obwohl es aus vielen, möglicherweise unterschiedlichen Subnetzen aufgebaut ist, kann jedes Endsystem mit jedem anderen kommunizieren. Das Endsystem bekommt die Abstraktion eines großen, einheitlichen Netzes geboten, das einen einheitlichen Adressraum und einheitliche Protokolle (zumindest auf einigen OSI-Schichten) aufweist. Solche Netze werden mit **Routern** als Zwischensysteme realisiert.

Neben Routern werden auch Repeater, Brücken (bridge) und Gateways als Zwischensysteme betrachtet.

Repeater funktionieren auf der Schicht 1, Brücken auf den Schichten 1-2, Rou-

ter auf den Schichten 1-3 und Gateways auf den Schichten 1-7 des OSI-Modells.

Repeater
Ein Repeater verbindet Netzsegmente (z. B. Ethernet-Segmente) unmittelbar miteinander. Er funktioniert auf der OSI-Schicht 1, regeneriert und verstärkt elektrische Signale. Repeater werden zur Vergrößerung der Ausdehnung eines Netzes eingesetzt.

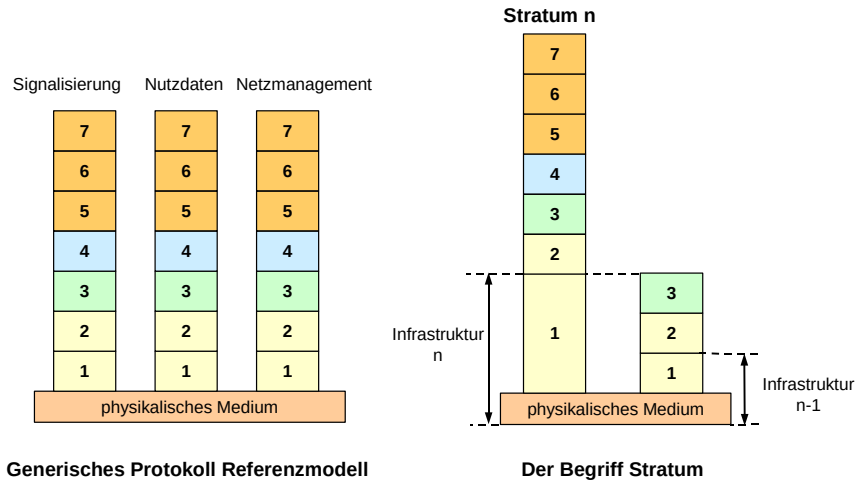
Brücken
Brücken verbinden Segmente zu einem Netz. Sie funktionieren auf der OS1-Schicht 2 und trennen Segmente logisch. Brücken **filtern Adressen** (es werden nur Rahmen weitergegeben, die tatsächlich die Brücke passieren müssen) und passen die Zugriffsmechanismen an, sofern diese auf den zu verbindenden Segmenten unterschiedlich sind. Brücken sind unabhängig von den höheren Protokollschichten. Sie werden primär zum **Verkehrsmanagement** eingesetzt, dessen Ziel es ist, lokalen Verkehr auch lokal zu begrenzen.

Remote Bridges können zur Kopplung weit entfernter Netzinseln eingesetzt werden. Dabei wird für die große Distanz entweder eine Glasfaserverbindung oder eine schmalbandige Telefon- oder Standleitung eingesetzt. Im ersten Fall reicht eine Brücke, im zweiten Fall sind Brücken an beiden Enden der schmalbandigen Verbindung erforderlich.

Router
Router verbinden Netze (die dadurch zu Subnetzen werden) zu einem Netz. Sie funktionieren auf der OSI-Schicht 3 und trennen die Subnetze logisch. Router sind vom eingesetzten Netzprotokoll abhängig, sie müssen die Netztopologie kennen. Router werden zum Aufbau von Internets, also zur Vernetzung einzelner Netze (Netzinseln) zu größeren Netzen, eingesetzt.

Brouter (zusammengesetzt aus den Wörtern bridge und router) sind eine Kombination aus Brücke und Router. Sie funktionieren demzufolge auf den OSI-Schichten 2 und 3 und bieten einerseits die Protokolltransparenz einer Brücke und andererseits das Routing bestimmter Protokolle. Brouter werden eingesetzt, wenn Netze mit unterschiedlichen Protokollen zu verbinden sind.

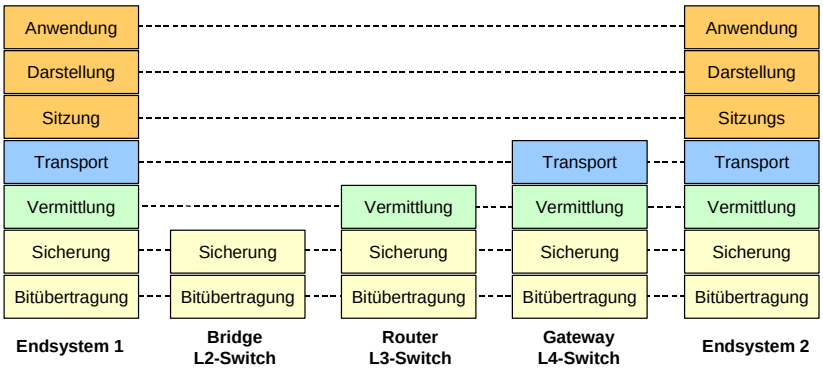
Gateways
Gateways verbinden Netze zu einem System, das heißt sie ermöglichen die Kommunikation zwischen Anwendungsprogrammen auf unterschiedlichen Endsystemen. Gateways funktionieren auf den OSI-Schichten 5 bis 7 und können so Anwendungsprotokolle ineinander übersetzen. Damit sind Gateways von den jeweiligen Anwendungsprogrammen abhängig.



Prinzip der Ebenen
Das aus der Paket-Kommunikation entstammende Strukturierungskonzept musste erweitert werden, um z. B. die in einem getrennten Steuernetz (Signalisiernetz des ISDN) ablaufende Verbindungssteuerung und Dienststeuerung einzubeziehen. Darüber hinaus gewann das System- und Netzmanagement an Bedeutung, so dass heute das Referenzmodell in drei Ebenen (Planes) aufgeteilt ist, welche nach Kontexten aufgeteilt sind und welche jeweils eine vertikale Schichtung aufweisen.

- Benutzerdaten (User Plane)
- Steuerdaten (Control Plane)
- Managementdaten (Management Plane)

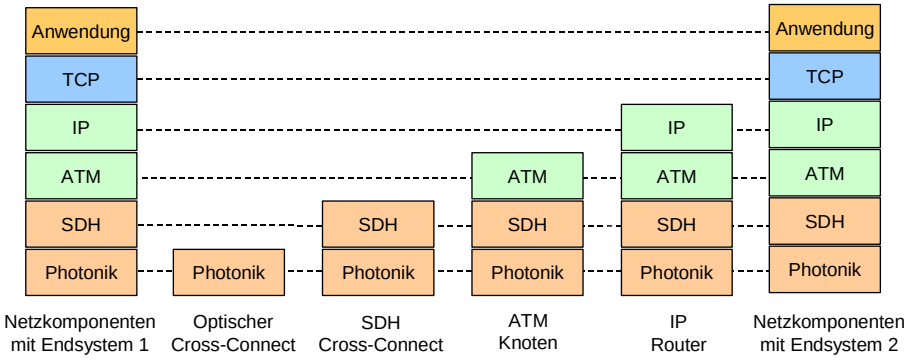
Bild: Referenzmodell- Erweiterungen



L2-Switch: Layer 2 Switch

Verschiedene Netzkomponenten

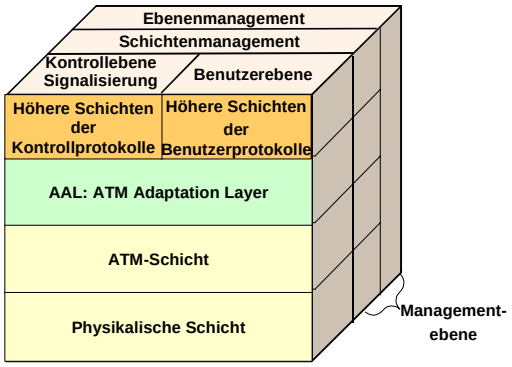
Bild: Protokollschichtung: Netzkomponente



Verschiedene Netztechnologien

Die Bitübertragungsschicht der Netztechnologie n wird durch die Protokollschichten der Netztechnologie n-1 ersetzt.
Beispiel: die Netzschicht von IP wird durch die ATM-Schichten ersetzt.

Bild: Protokollschichtung: Netztechnologien



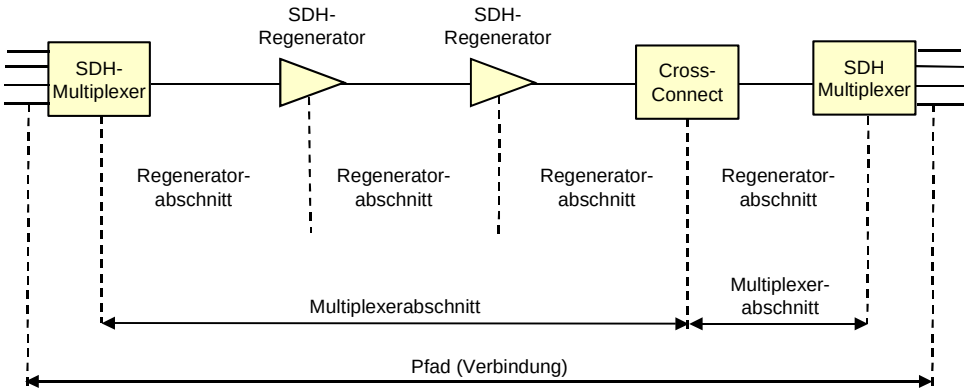
- **Signalisierungs- oder Kontrollebene**
- **Benutzerebene**
- **Managementebene**
  - Schichtenmanagement
  - Ebenenmanagement

ATM: Asynchronous Transfer Mode

Bild: ATM-Referenzmodell

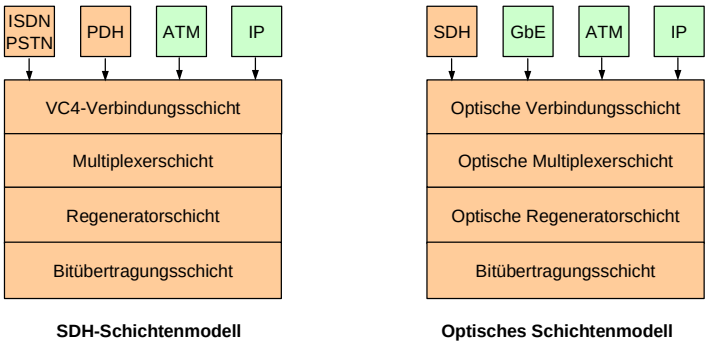
B-ISDN-Referenzmodell

Das Schichtenmodell für B-ISDN bzw. ATM zeigt eine konkrete Anwendung der Strukturierung in Schichten und Ebenen. Einzelne Schichten sind weiter unterteilt. Die Anwenderebene (userplane) repräsentiert die Funktionen für die Nutzdatenübertragung, die Steuerungsebene (control plane) beinhaltet die **Signalisierung,** die für die hier realisierte verbindungsorientierte Kommunikation erforderlich ist. Die **Managementebene** (management plane) ist aufgeteilt in das **Ebenenmanagement** (plane management) und das **Schichtenmanagement** (layer management). Das Ebenenmanagement ist für die Koordination aller Ebenen zuständig, das Schichtenmanagement führt schichtenspezifische Managementfunktionen aus.



SDH Synchronous Digital Hierarchy

Bild: SDH- Übertragungsstrecke



ISDN	Integrated Services Digital Network	VC4	Virtual Container number 4 (SDH-Technik)
PSTN	Public Switched Telephone Network	IP	Internet Pprotocol
PDH	Plesiochronous Digital Hierarchy	ATM	Asynchronous Transfer Mode
SDH	Synchronous Digital Hierarchy	GbE	Gigabit Ethernet

Bild: Übertragungsschichtenmodell

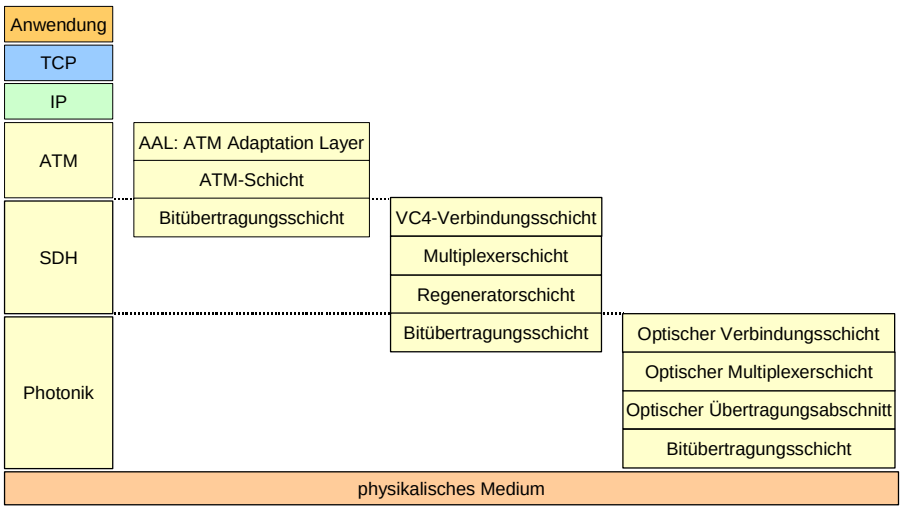


Bild: Stratum-Protokollschichtung

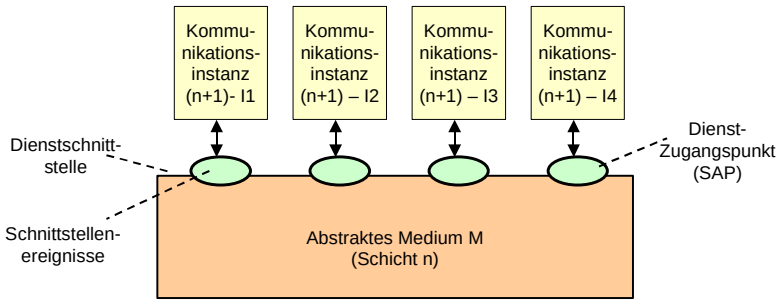


Bild: Dienstbegriff

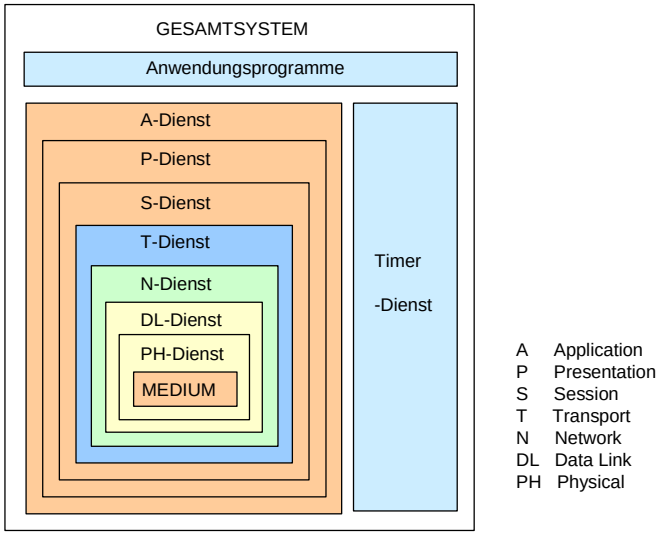
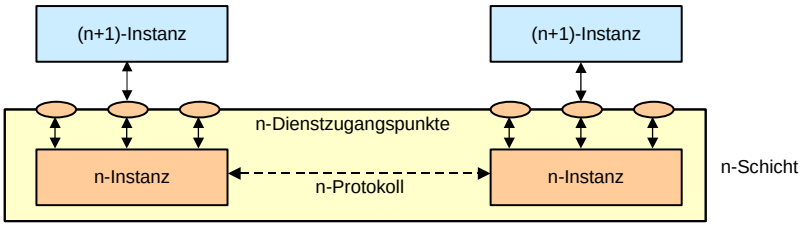
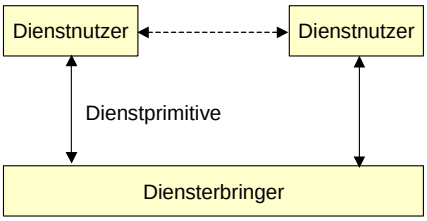


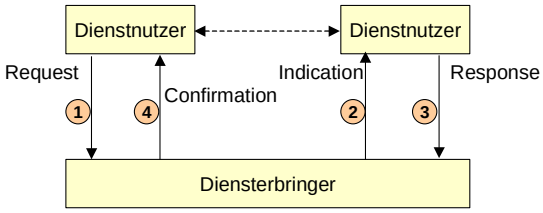
Bild: Hierarchische Dienststruktur



- **n-Instanz:**
  erbringt n-Dienst und stellt ihn (n+1)-Instanzen über n-Dienstzugangspunkte zur Verfügung
- **n-Schicht:**
  Abstraktionsebene mit definierten Aufgaben; besteht aus allen n-Instanzen
- **n-Protokoll:**
  Regeln zum Datenaustausch zwischen n-Instanzen

Bild: Instanz, Schicht und Protokoll



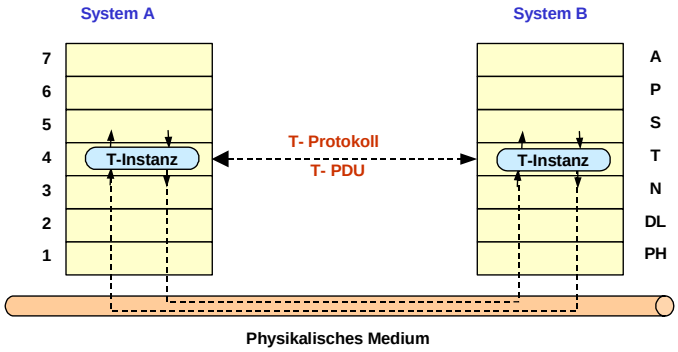


Diensterbringer:
z.B. gesamtes Kommunikationssystem, einzelne Schicht

Dienstbenutzer:
z.B. Benutzer, einzelne Schicht

4 Typen von Dienstprimitiven:
- Anforderung (Request)
- Anzeige (Indication)
- Antwort (Response)
- Bestätigung (Confirmation)

Bild: Dienstnutzer / Diensterbringer



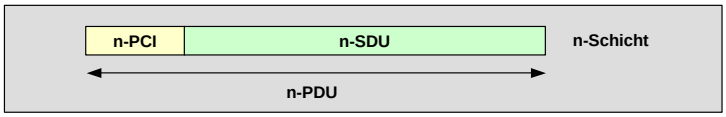
T: Transport
N: Network
DL: Data Link
PH: Phyical

A: Application
P: Presentation
S: Session

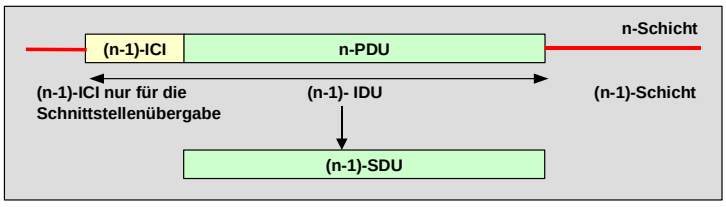
PDU: Protocol Data Unit

Bild: Kommunikation innerhalb einer Schicht

Horizontale Datenaustausch-Einheit: Protocol Data Unit



Vertikale Datenaustausch-Einheiten: Interface and Service Data Units



SDU : Service Data Unit
PDU : Protocol Data Unit
IDU : Interface Data Unit

PCI : Protocol Control Information
ICI : Interface Control Information

Bild: Datenaustausch-Schnittstellen

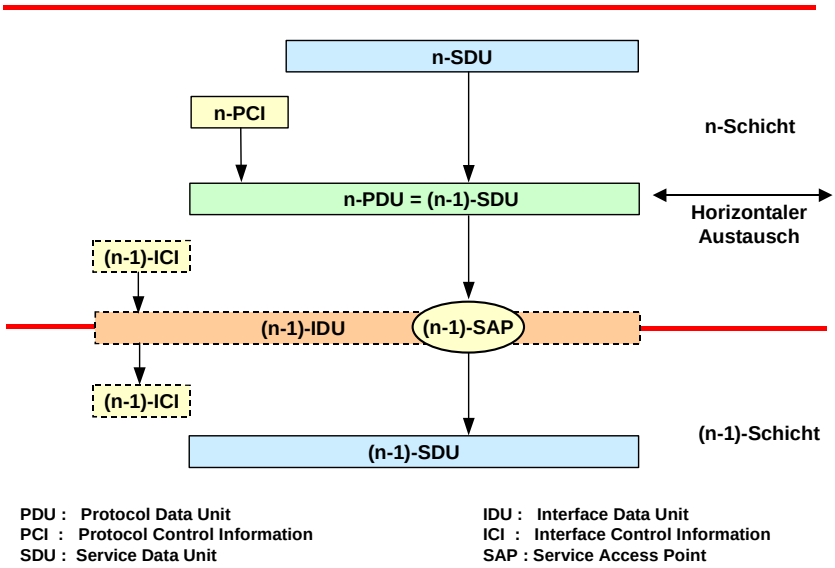


Bild: Schnittstellen zwischen Schichten: n nach (n-1)

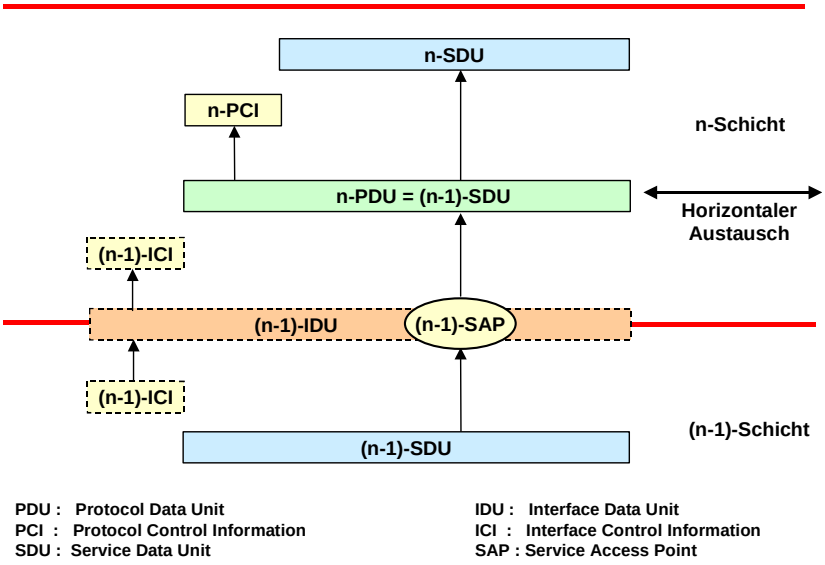


Bild: Schnittstellen zwischen Schichten: (n-1) nach n

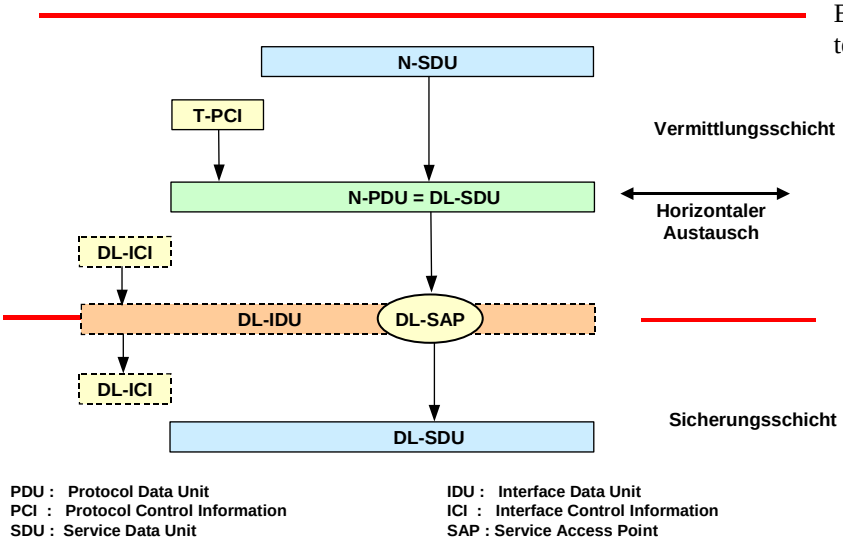
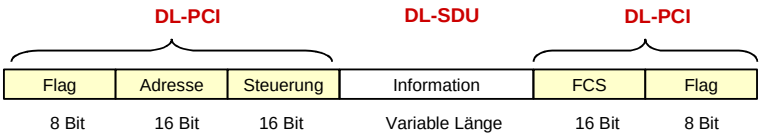


Bild: Beispiel: Schnittstellen zwischen Schichten

Schicht 2: Rahmen-Format



PCI : Protocol Control Information
SDU : Service Data Unit
FCS: Frame Check Sequence

Bild: Beispiel für eine Protocol Data Unit (PDU)

Services	SAP	Instanz	SDU	PCI	PDU
Application	-	A - Instanz	-	A - PCI	A - PDU
Presentation	P - SAP	P - Instanz	P - SDU	P - PCI	P - PDU
Session	S - SAP	S - Instanz	S - SDU	S - PCI	S - PDU
Transport	T - SAP	T - Instanz	T - SDU	T - PCI	T - PDU
Network	N - SAP	N - Instanz	N - SDU	N - PCI	N - PDU
DataLink	DL - SAP	DL - Instanz	DL - SDU	DL - PCI	DL - PDU
PHysical	PH - SAP	PH - Instanz	PH - SDU	PH - PCI	PH - PDU

SAP: Service Access Point
SDU: Service Data Unit
PDU: Protocol Data Unit
PCI: Protocol Control Information

Bild: Schichtenbezogene Namensgebung

Verbindungskonzepte
Zur Unterstützung des Datenaustausches zwischen zwei (N)-SAPs wird das Konzept einer Verbindung (Connection) eingeführt. Wie im Bild gezeigt, verbindet eine (N)-Verbindung logisch zwei (N)-SAPs miteinander, wobei die (N)-Verbindung als Funktion der Schicht (N) aufgefasst werden muss, welche ihrerseits auf Funktionen unterer Schichten zurückgreift. Innerhalb eines (N)-SAP können mehrere Endpunkte von (N)-Verbindungen liegen, welche jeweils durch Verbindungsendpunkt-Kennungen (Connection Endpoint Identifier, CEI) unterschieden werden.

Der **verbindungsorientierte** Nachrichtenaustausch (Connection Oriented Mode, CO) wird durch eine Reihe von Funktionen unterstützt. Bevor eine Kommunikation zwischen zwei (N+1)-Instanzen erfolgt, muss erst eine (N)-Verbindung hergestellt werden (Connection Establishment). Innerhalb einer Verbindung kann der Datenaustausch durch Merkmale wie Folgenumerierung, Reihenfolgesicherung, Fehlererkennung/automatische Wiederholung sowie Datenflusssteuerung sehr zuverlässig gestaltet werden. Die Parameter hierzu sind i. a. Gegenstand eines gegenseitigen Abstimmungsprozesses beim Verbindungsaufbau. Sie sind selbst während einer Verbindung noch änderbar. Die Verbindung wird nach Abschluss des Datenaustausches wieder abgebaut (Connection Release).

Neben dem verbindungsorientierten Nachrichtenaustausch existiert eine einfachere Form, der **verbindungslose** Nachrichtenaustausch (Connectionless Mode, CL). Diese Variante wird insbesondere für lokale Rechnernetze (LANs) interessant, wo die zuverlässige Kurzstreckenübertragung nicht in allen Kommunikationsschichten dieselben Funktionen wie Reihenfolgesicherung, Fehlerbehandlung und Datenflusssteuerung erfordert. Das Konzept ist auch in Weitverkehrsnetzen auf der Basis des Austausches voll adressierter, autonomer Pakete (Datagramm) realisiert worden.

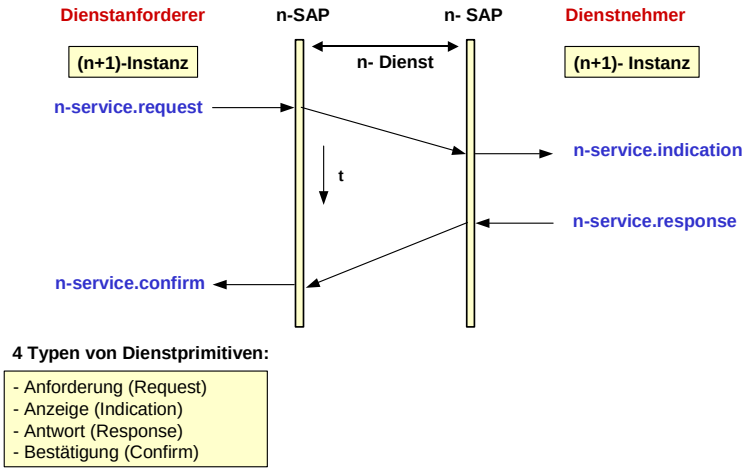


Bild: Dienstprimitive

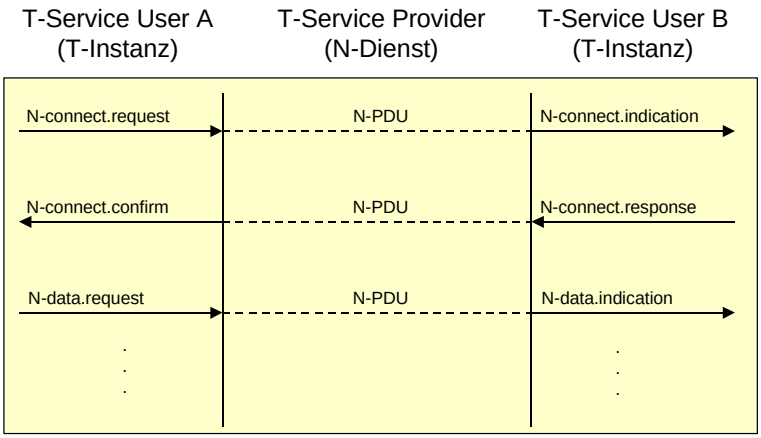


Bild: Zeitliche Folge von Primitiven

Die zeitliche Folge von Primitiven, welche zwischen benachbarten Schichten zur Dienstanforderung benutzt werden, unterliegt i. a. bestimmten Synchronisationsbedingungen. Zusammen mit dem Format der dazu benutzten Steuerdatenblöcke kann dieser Vorgang in Form eines Dienstprotokolls definiert werden. Die einzelnen Dienstprimitive werden mit einer Parameterliste versehen und in einem Steuerdatenblock übertragen. Das nachfolgende Bild zeigt das Beispiel eines Transportverbindungsaufbaus, der von der Schicht 4 (T) initiiert wird. Die übergebenen Parameter beziehen sich in diesem Falle auf die Adressen (Rufnummern) der rufenden und gerufenen Datenendeinrichtungen, die Option eines beschleunigten Transports, Dienstparameter sowie Benutzerdaten.

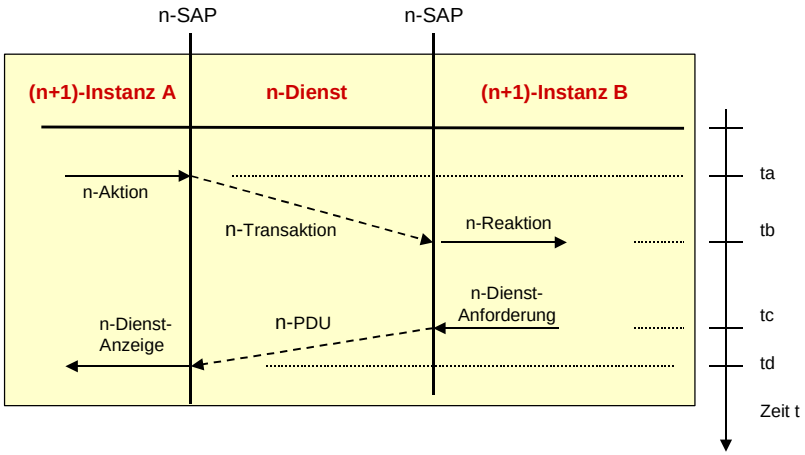


Bild: Zeitfolgediagramm

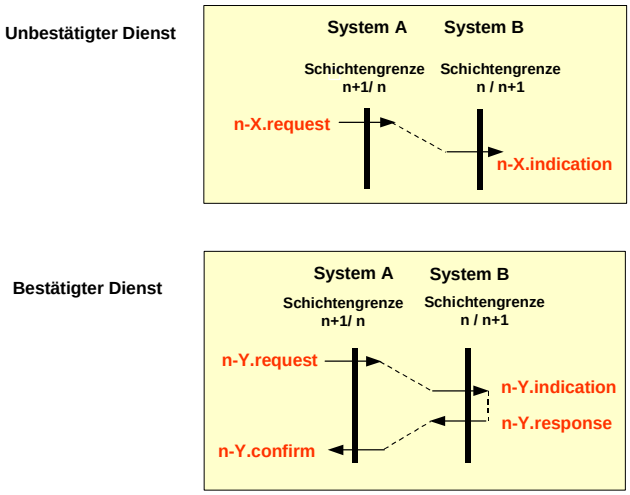


Bild: Kommunikation innerhalb Schicht n

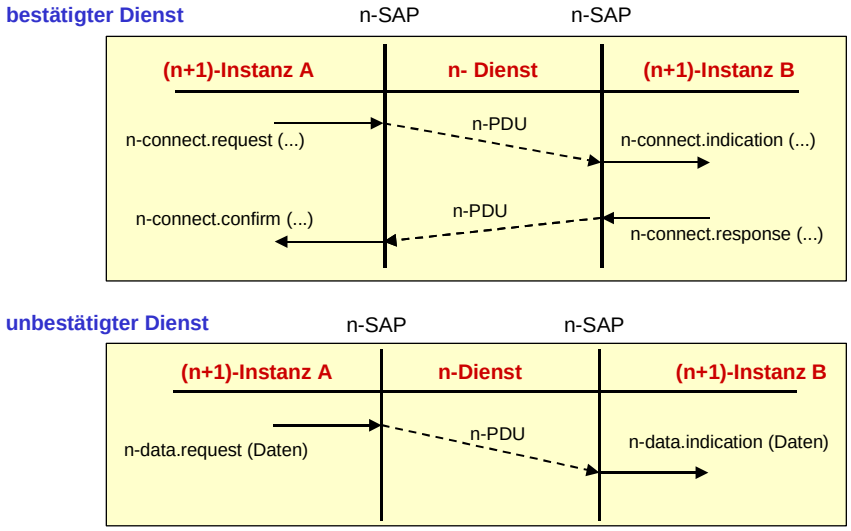


Bild: Kommunikationsabläufe

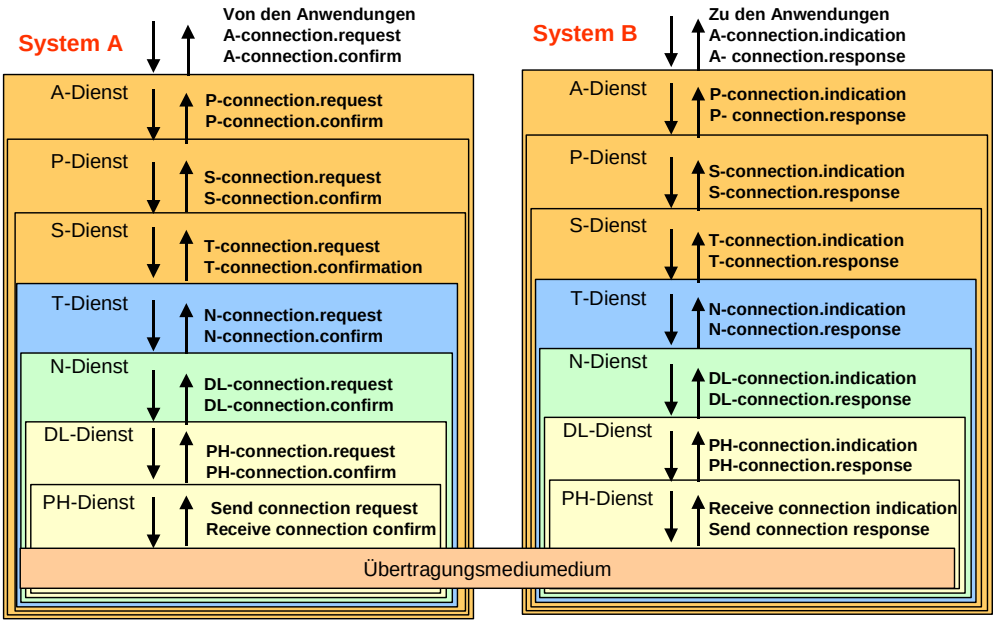
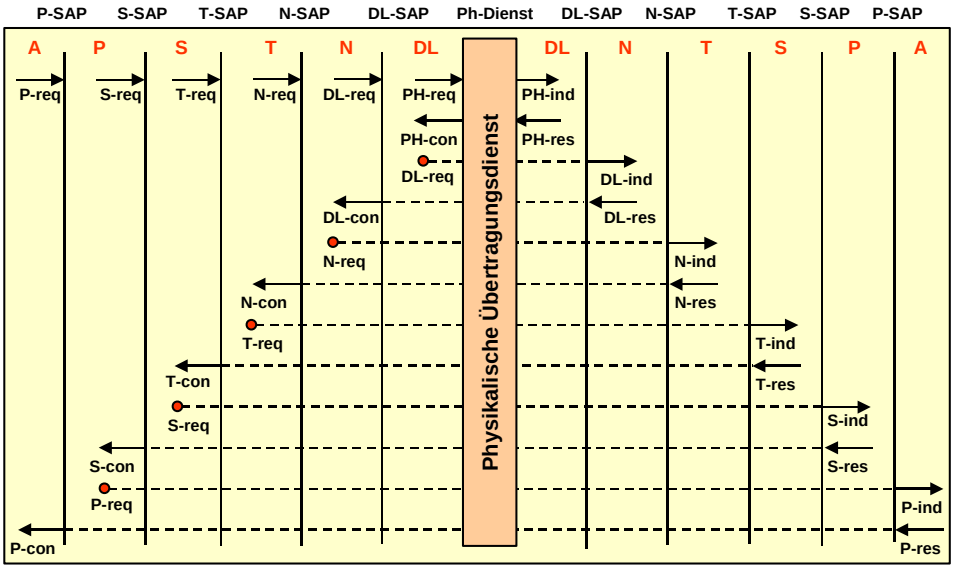


Bild: Verbindungsmanagement

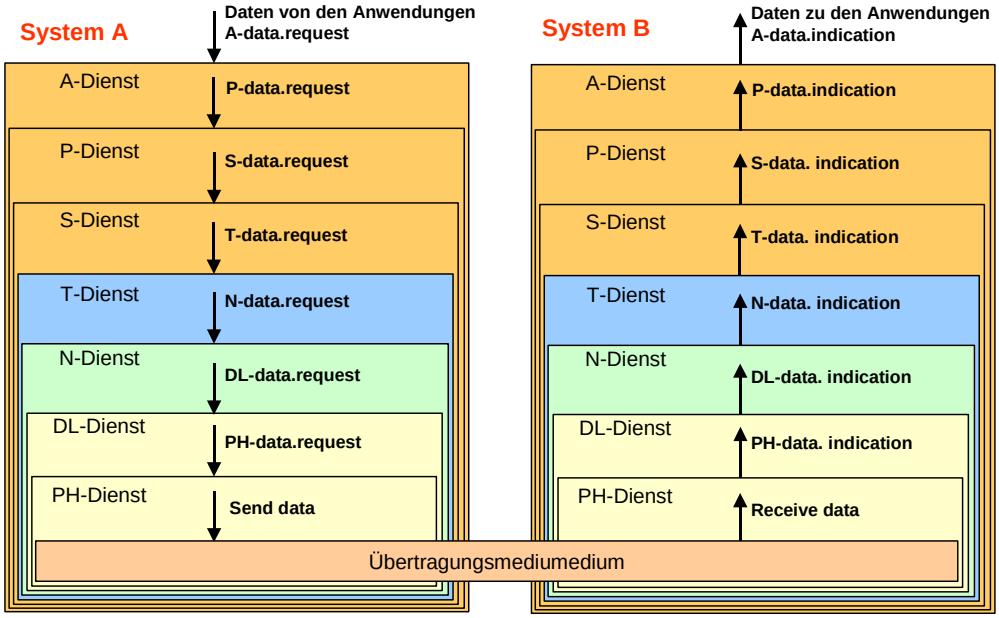
Das Bild zeigt die Dienstprimitive, die zwischen den Protokollschichten ausgetauscht werden, wenn eine logische Verbindung zwischen zwei A-Instanzen aufgebaut werden soll und nur der physikalische Dienst eingerichtet ist.



Das Bild zeigt den zeitlichen Ablauf beim Aufbau einer logischen Verbindung zwischen zwei A-Instanzen für den Fall, dass nur der physikalische Dienst eingerichtet ist

Requests können nur weiter geleitet werden, wenn der Dienst darunter vorhanden ist.

Bild: Zeitlicher Ablauf des Verbindungsaufbaus



Das Bild zeigt alle Dienstprimitive, die zwischen den Protokollschichten ausgetauscht, wenn eine A-Instanz Daten sendet und die Partner A-Instanz sie empfängt.

Daten zwischen Schichten werden immer unbestätigt ausgetauscht. Die Datenbestätigung geschieht immer zwischen Partnerinstanzen.

Beispiel: (n+1)-Daten, die durch n-Datenprimitive an einen n-Dienst übergeben werden, werden bei einer bestätigten Dienst durch die (n+1)-Instanzen gegenseitig bestätigt.

Bild: Datenaustauchprimitive

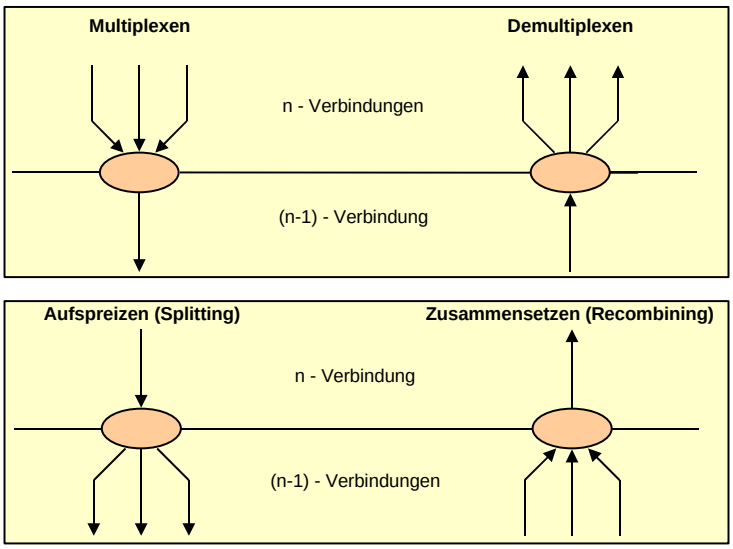
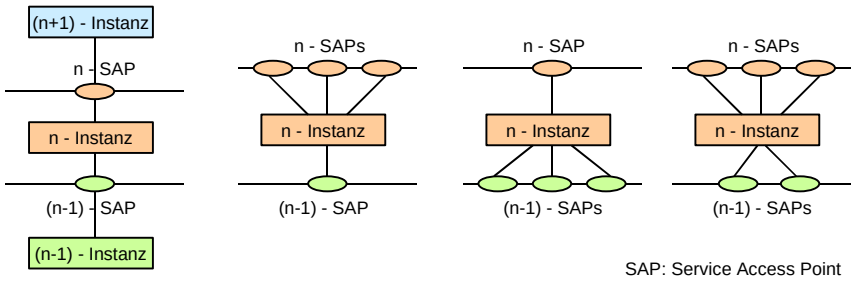


Bild: Verbindungsmanagement



- n-Instanz und (n+1)-Instanz, die über einen n-SAP verbunden sind, befinden sich im gleichen System
- (n+1)-Instanz kann mit mehreren n-SAPs verbunden sein
- n-SAPs können mit einer oder mehreren n-Instanzen verbunden sein
- n-Instanz kann mit mehreren (n+1)-Instanzen über mehrere n-SAPs verbunden sein

Bild: Beziehungen zwischen SAPs und Instanzen

- zu einem gegebenen Zeitpunkt ist ein n-SAP mit genau einer (n+1)-Instanz und genau einer n-Instanz verbunden
- n-SAP kann von einer n-Instanz und/oder einer (n+1)-Instanz getrennt und einer anderen n-Instanz und/oder (n+1)-Instanz zugeordnet werden
- n-SAP wird über seine n-Adresse lokalisiert
- Adresse wird von (n+1)-Instanzen bei der Anforderung einer n-Verbindung benötigt

Bild: Temporäre Beziehungen zwischen SAPs und Instanzen

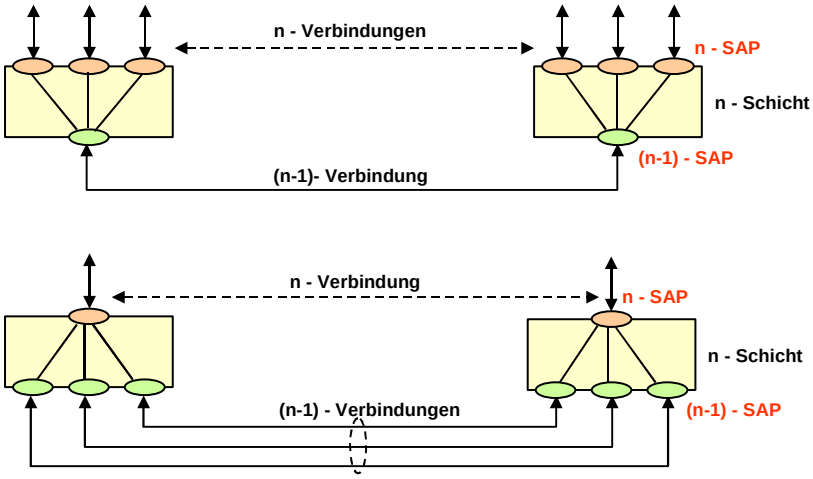


Bild: Multiplexen und Splitting

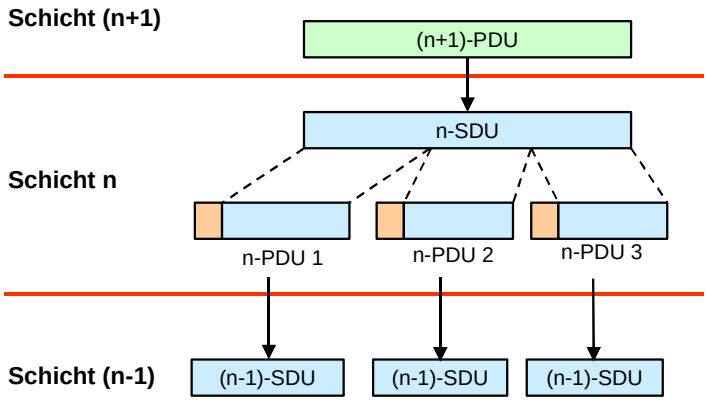


Bild: Aufteilen / Vereinigen

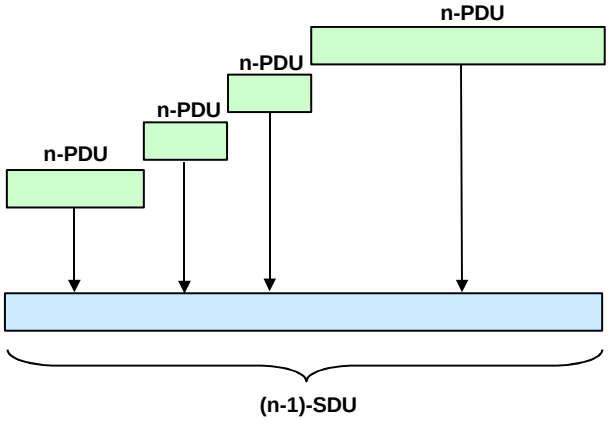
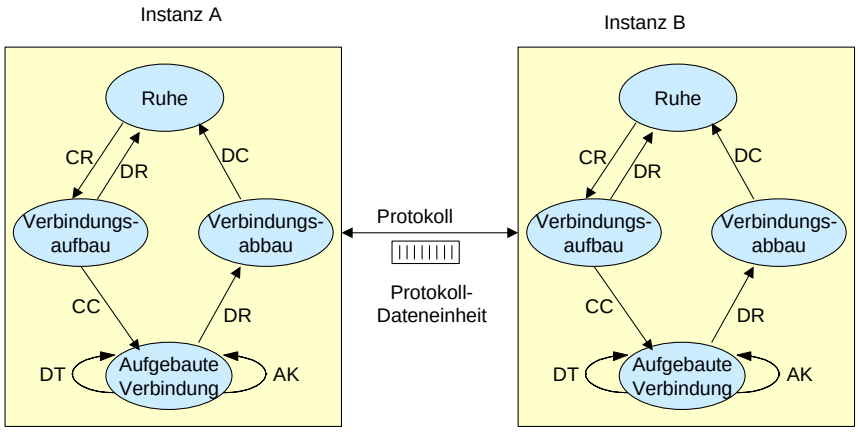


Bild: Verketten / Trennen



CR: Connect Request
CC: Connect Confirm
DT: Data Transfer
AK: Acknowledgement
DR: Disconnect Request
DC: Disconnect Confirm

Bild: Kommunikation von Instanz zu Instanz

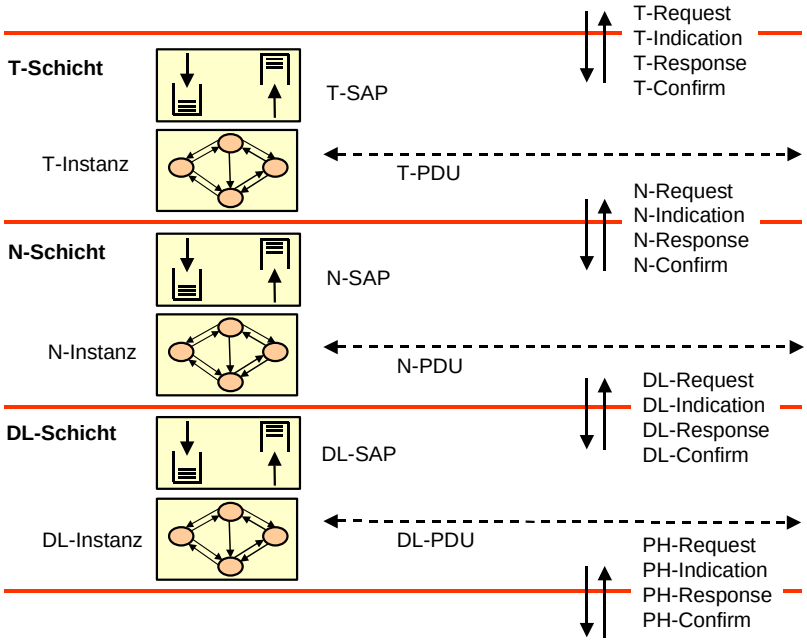


Bild: Peer-to-Peer und Schichten Kommunikation

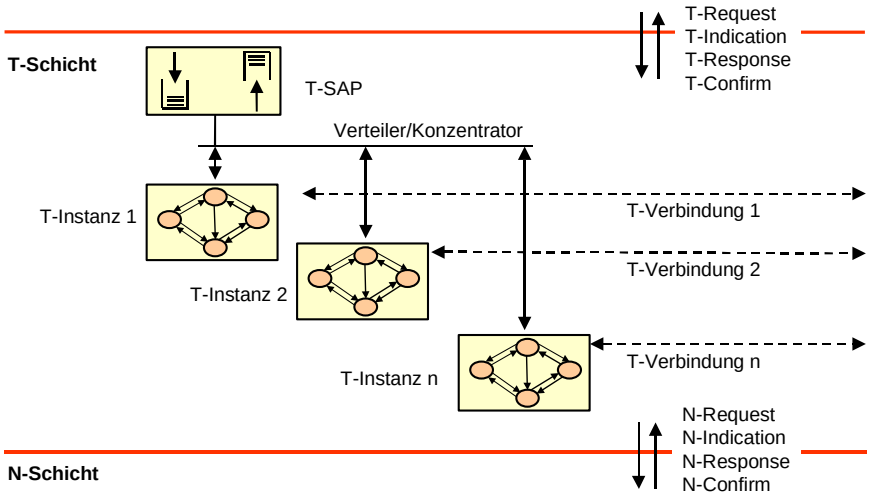


Bild: Peer-to-Peer Verbindungen

Protokollmechanismen

Protokolle der verschiedenen OSI-Schichten enthalten zum Teil vergleichbare Funktionen. Diese werden am besten gemeinsam betrachtet. Sie werden als Protokollmechanismen bzw. Protokollfunktionen bezeichnet. Den Gegensatz dazu bilden die schichtenspezifischen Protokolle, die getrennt betrachtet werden müssen..

Die Protokollmechanismen lassen sich nach Funktionsbereichen gliedern.

Basis-Protokollmechanismen

Datentransfer: Dieser Protokollmechanismus wird fast überall benötigt, meistens bezieht er sich auf einen gewöhnlichen Transfer von Datenblöcken. Abhängig von der Schicht betrachtet man Rahmen (frames), Pakete (packets), Zellen (cells) oder Nachrichten (messages). Zusätzlich kann ein Prioritäts-Datentransfer (expedited data transfer) sinnvoll sein, dessen Daten vor den normalen Daten ausgeliefert werden. Die prioritären Daten können dabei früher gesendete, normale Daten überholen.

Verbindungsverwaltung: Sorgt primär für den erfolgreichen Verbindungsaufbau. Eine Ablehnung der Verbindung durch den Empfänger (caller) oder den verwendeten Dienst wird dem Sender (Initiator, callee) mitgeteilt. Der Protokollmechanismus muss mit verlorenen, duplizierten oder verspäteten Datenblöcken zurechtkommen. Weiter realisiert er die bestätigte Dienstfunktion Verbindungsabbau (disconnect) und die unbestätigte Dienstfunktion Verbindungsabbruch (abort). Im ersten Fall können ausstehende Übertragungen noch ausgeführt werden, im zweiten Fall nicht.

Sequenznummern: Für die Fehlerbehandlung (auch Auslieferung in der richtigen Reihenfolge) und die Systemleistungsanpassung sind nummerierte Datenblöcke erforderlich.

Quittierung: Der korrekte Empfang von Datenblöcken wird vom Empfänger quittiert. Dazu kann eine eigene Quittungseinheit (ACK, Acknowledgment) verwendet werden oder die Quittung wird einem Datenblock, das in der Gegenrichtung übertragen wird, mitgegeben. Diese Variante wird als Huckepack-Quittung (piggy back acknowledgment) bezeichnet. Quittungen können sich auf mehrere, korrekt empfangene Datenblöcke beziehen.

Protokollmechanismen zur Fehlerbehandlung

Prüfsummen zur Fehlererkennung: Datenblöcke werden in der Regel durch redundante Information ergänzt, die dem Empfänger die Erkennung von Übertragungsfehlem ermöglicht.
Übertragungswiederholung: Datenblöcke, die nicht oder fehlerhaft empfangen wurden, werden vom Empfänger mittels einer **negativen Quittung** (NACK, Negative ACKnowledgement) nochmals angefordert. Dabei können ganze Folgen von Datenblöcke oder - aus Effizienzgründen besser - nur einzelne Datenblöcke wiederholt werden.
Fehlerkorrektur (forward error correction): Wenn einem Datenblock hinreichend redundante Information zugefügt wird, kann eine automatische Fehlerkorrektur beim Empfänger durchgeführt werden, ohne dass Datenblöcke wiederholt werden.
Zeitüberwachung (timeout): Falls eine vorgegebene Zeitspanne zwischen dem Absenden eines Datenblockes und dem Empfang einer Quittung überschritten wird, nimmt der Sender den Verlust des Datenblockes oder der Quittung an. Deshalb wird dieser Datenblock erneut gesendet. Die Wahl der Zeitspanne ist von großer Bedeutung. Eine zu frühe Reaktion überfüllt die Leitung oder das Netz mit Datenblockkopien. Eine zu späte Reaktion erhöht die Kommunikationsverzögerung.

Fehlerbehebung, ARQ-Verfahren

ARQ-Verfahren (Automatic Repeat Request) leisten die Behebung erkannter Fehler, indem verfälscht empfangene oder verlorene Pakete vom Sender nochmals gesendet werden.

Aufgaben und Konzepte
Die Erkennung von Übertragungsfehlem wird durch fehlererkennende bzw. fehlerkorrigierende Codes geleistet. Zusätzlich treten andere Probleme auf. Die Verzögerung eines Rahmens auf der Übertragungsstrecke kann schwanken und unter Umständen große Werte annehmen. Rahmen können ganz verloren gehen, wenn ein Rahmen so stark verfälscht ist, dass der Empfänger ihn nicht mehr als solchen erkennen kann. Deshalb ist über die Flusssteuerung hinaus ein Mechanismus erforderlich, der sich um die Beseitigung dieser Fehler kümmert. Da fehlende und fehlerhafte Rahmen nochmals übertragen werden (dies so oft wie nötig, allerdings wird bei Erreichen einer vorher festgelegten Anzahl Wiederholungen die Kommunikation mit einer Fehlermeldung abgebrochen), bezeichnet man diese Mechanismen als ARQ (Automatic Repeat Request). Korrekt empfangene Rahmen werden durch **positive Quittungen** (acknowledgment) quittiert, falsch empfangene Rahmen können durch **negative Quittungen** (reject) nochmals angefordert werden. Nicht empfangene Rahmen stellt der Sender dadurch fest, dass er innerhalb einer festgelegten Zeit (timeout) keine Quittung für einen gesendeten Rahmen erhält. Er führt daraufhin von sich aus eine Wiederholung der Übertragung durch.

Wichtig: Eine positive Quittung enthält die Sequenznummer des nächsten erwarteten Blocks.
Eine negative Quittung enthält die Sequenznummer eines fehlerhaften Blocks.

Die Bedeutung (Semantik) von Quittungen kann also positiv oder negativ sein und sich auf einzelne Rahmen oder eine Folge von Rahmen (Summenquittung) beziehen. Quittungen werden ausgelöst durch den Empfang einer Rahmens oder den Ablauf eines Zeitgebers (timeout).

Stop-and-Wait bei ARO
Das ARQ-Verfahren mit Stop-and-Wait sendet einen Block und wartet auf eine Quittung. Die Blöcke RO und RI werden mit ACKO und ACKI quittiert. Der Block R2 geht verloren, was den Sender nach Ablauf des Timeout veranlasst, R2 nochmals zu senden. Später geht ACK4, das ist die Quittung für Block R4, verloren. Nach Ablauf des Timeout wird R3 nochmals gesendet.

Go-Back-N
Go-Back-N verwendet ein Fenster der Größe N, d. h. der Sender darf maximal N Blöcke senden, für die eine Quittung noch aussteht. Der Empfänger quittiert jeden korrekt empfangenen Block mit der Sequenznummer des nächsten erwarteten Blocks. Im Beispiel Bild 2.37 werden die Blöcke RO und Rl korrekt übertragen, dann geht R2 verloren. Der Empfänger reagiert mit einer negativen Quittung (NAK2) für Block R2. Der Sender empfängt NAK2 zu einem Zeitpunkt, in dem R3, R4, R5 schon gesendet wurden. Deshalb werden im nächsten Sendevorgang R2 und alle folgenden Blöcke wiederholt. Beim Empfänger werden in der Zwischenzeit R3, R4, R5 verworfen. Anschließend wird R2 korrekt empfangen und mit ACK3 quittiert. Damit läuft die Übertragung fehlerfrei weiter.

Die **Effizienz eines ARQ-Verfahrens** ist definiert als der Anteil der verstrichenen Zeit, in der der Sender Pakete sendet.

Die Effizienz für eine fehlerfreie Übertragung ergibt sich zu:
Effizienz = min {N TRANS / (TRANS + ACK + 2PROP), 1}

Die Parameter N = Fensterlänge, TRANS = Sendedauer eines Paketes, ACK =.Sendedauer einer Quittung, PROP = Laufzeit des Signals sind für beide Übertragungsrichtungen als gleich angenommen. Die Formel besagt, dass die Effizienz von einem Minimalwert (bei N = 1) linear mit N ansteigt, bis der Maximalwert 1 erreicht ist. Bei einer Effizienz von 1 ist der Kanal voll ausgelastet, d. h., der Sender macht keine Pausen.

Selective Reject
Ein beispielhafter Ablauf geht davon aus, dass die Rahmen RO und R 1 mit ACK 1 und ACK2 quittiert wurden. Anschließend gehe R2 verloren, was durch NAK2 gemeldet wird. In der Zwischenzeit sendet der Sender R3 bis R5. Nach Eintreffen von NAK2 wird lediglich R2 wiederholt, anschließend wird mit R6, R7, ... weitergefahren. Das Sortieren der empfangenen Rahmen in die richtige Reihenfolge ist die Aufgabe des Empfängers. Der Vorteil liegt darin, dass ausschließlich der verlorene Rahmen wiederholt gesendet wird.

Protokollmechanismen zur Längenanpassung

Segmentierung/Reassemblierung: Die zulässige Paketlänge ist aus verschiedenen Gründen begrenzt. Falls die Nutzdaten länger sind als die auf einer Teilstrecke zulässige Länge, müssen sie segmentiert, also aufgespalten werden. Beim Empfänger muss die Segmentierung rückgängig gemacht werden (Reassemblierung), bevor die Nutzdaten dem Anwendungsprozess übergeben werden.

Protokollmechanismen zur Systemleistungsanpassung

Flusssteuerung (flow control): Schützt den Empfänger vor einer Überlastung durch den Sender. Dafür wird häufig ein Fenstermechanismus verwendet. Der Empfänger gibt dabei dem Sender vor, wie viele Pakete deren Quittungen noch ausstehen - dieser maximal senden darf (-> Abschnitt 2.7).

Überlaststeuerung (congestion control): Schützt das Netz vor Überlastung durch die von allen Sendern gesendeten Pakete. Hierzu kann ebenfalls ein Fenstermechanismus eingesetzt werden, wenn die Fenstergröße (ausgedrückt als die maximale Anzahl der unbestätigten Pakete) in Abhängigkeit der Netzbelastung verändert wird.

Ratensteuerung (rate control): Beim Verbindungsaufbau können Sender und Empfänger (evtl. unter Mitwirkung des Netzes) eine zulässige Rate (gesendete Datenmenge pro Zeiteinheit) aushandeln.

Protokollmechanismen zur Systemleistungsanpassung Flusskontrolle

Die Flusskontrolle stellt Strategien zur Vermeidung des Überlaufens von Empfangspuffern bereit. Sie steuert den ZuFluss von Paketen in Paketvermittlungssysteme, sowie zwischen den einzelnen Netzknoten. Ziel ist es, einen reibungslosen Fluss

von Nachrichten zu ermöglichen, Staus zu verhindern und wenn doch einer auftritt, wieder abzubauen. Man unterscheidet

Ende-zu-Ende-Flusskontrolle	Die End-zu-End-Flusskontrolle bewirkt eine Einspeisung eines Nachrichtenstoms entsprechend der Abnahmefähigkeit des Empfängers Dieser Teil der Flusskontrolle liegt damit im Bereich des Teilnehmers
Knoten-zu-Knoten- Fluss Kontrolle	Die Knoten-zu-Knoten-Flusskontrolle im Bereich der Vermittlungssysteme liegt und entsprechend der Auslastung der Knoten wirkt. Eine ungenügende Flusskontrolle kann das Netz in eine Überlastsituation geraten lassen, in der die Durchsatzleistung, drastisch abfällt.

Tritt eine Überlastsituation dennoch ein, muß eine Strategie zur Vermeidung der Überlastung der Aufnahmekapazität des Netzes hereitgestellt werden. Diese wird in der sog. Überlastkontrolle subsumiert. Ire der Paketvermittlungstechnik werden verschiedene Methoden der Flusskontrolle. Überlastkontrolle eingesetzt, z.B.:

Isarithmische Flusskontrolle	Die Gesamtzahl der Nachrichtenpakete im Netz bleibt stets nach oben begrenzt. In jedem Netzknoten wird eine Anzahl von "Paketbehälter" vorgegeben, die für ankommende Pakete zur Verfügung steht. Nur wenn noch ein "Paketbehälter" frei ist, darf ein Paket auch angenommen werden. "Paketbehältern" sind fiktive Einrichtungen und werden in den Netzknoten durch Zähler abgebildet.ı
Richtungsabhängige Flusskontrolle	Die Länge der Ausgabewarteschlange vor einer Leitung wird begrenzt. Ein Nachrichtenpaket wird akzeptiert, wenn die Ausgabewarteschlange vor der benötigten Leitung eine vorgegebene Länge nicht überschreitet. Damit werden nur die Nachrichtenpakete zurückgestaut, die eine überlastete Leitung benutzen wollen.

Grundlage der Flusskontrollsignalisierung ist das Fenster-Verfahren.

Flusssteuerung (flow control, auch Flusskontrolle): Schützt den Empfänger vor einer Überlastung durch den Sender. Dafür wird häufig ein Fenstermechanismus verwendet. Der Empfänger gibt dabei dem Sender vor, wie viele Pakete deren Quittungen noch ausstehen. **D**ie Aufgabe ist also, zu verhindern, dass ein schneller Sender einen langsamen Empfänger mit Daten überschwemmt. Der Empfänger besitzt einen Pufferspeicher, der eine bestimmte Anzahl empfangener Rahmen zwischenspeichern kann, bevor der Protokollstapel diese übernimmt. Große Pufferspeicher sind vorteilhaft, aber teuer. Folglich benötigt der Empfänger einen Mechanismus, mit dem er den Sender veranlassen kann, eine bestimmte Zeit zu warten. Hierzu sendet der empfangsbereite Empfänger dem Sender eine Quittung, die ihm erlaubt, bis zu n Blöcke zu senden.

Verfahren

Stop-and-Wait
Bei diesem Verfahren sendet der Sender genau einen Rahmen. Danach wartet er auf eine Quittung (Empfangsbestätigung) des Empfängers, nach deren Erhalt er einen weiteren Rahmen sendet. Abläufe dieser Art werden in Zeitdiagrammen dargestellt. In diesem sehr einfachen Fall ist angenommen, dass alle Rahmen in der Reihenfolge des Absendens unverfälscht beim Empfänger eintreffen. Vorteil des Stop-and-Wait-Verfahrens ist der geringe Aufwand für das Protokoll. Nachteil ist ein geringer Durchsatz, insbesondere auf Übertragungsstrecken mit langer Laufzeit, auf denen kurze Rahmen übertragen werden. Grund dafür ist, dass die Übertragungsstrecke nur während kurzer Zeit mit einer Übertragung beschäftigt ist und dann relativ lange auf eine Quittung (ACK: Acknowledgment) wartet.

Sliding Window (Schiebe-Fensterverfahren)
Eine höhere Auslastung der Übertragungsstrecke lässt sich erreichen, wenn mehrere Rahmen nacheinander gesendet werden, bevor eine Quittung erwartet wird. Dabei darf der Sender eine Anzahl n (Fenstergröße) Rahmen senden, ohne eine Quittung zu erhalten. Wenn diese Anzahl ausgeschöpft ist, muss der Sender warten, bis eine Quittung eintrifft. Die maximale Fenstergröße n_{max} ist ein vordefinierter Parameter. Die Rahmen haben zur eindeutigen Kennzeichnung Sequenznummern. Rahmen links des Sendefensters wurden bereits gesendet, Rahmen im Sendefenster dürfen noch ohne Quittungseingang gesendet werden. Beim Senden eines Rahmens verschiebt sich die linke Fenstergrenze um einen Block nach rechts, das Fenster schrumpft. Bei Erhalt einer Quittung verschiebt sich die rechte Fenstergrenze um einen Rahmen nach rechts, das Fenster dehnt sich (aber nur bis n_{max}). Bei n = 0 darf nichts weiter gesendet werden. Für die Nummerierung der Rahmen sollen k bit (k möglichst klein) verwendet werden, d. h., die Rahmennummern sind modulo k dargestellt. Damit lässt sich eine maximale Fenstergröße von 2k-1 darstellen. Nach Erreichen der größten Rahmennummer folgt die 0 als nächste Nummer (wrap around). Bei kleinen k kann dies zu Problemen führen, indem die Sequenznummer unter Umständen nicht mehr eindeutig einen Rahmen benennt.

Die für die Flusssteuerung verwendeten Verfahren werden - in erweiterter Form - auch für die Fehlerbehebung genutzt.

Überlaststeuerung (congestion control): Schützt das Netz vor Überlastung durch die von allen Sendern gesendeten Pakete. Hierzu kann ebenfalls ein Fenstermechanismus eingesetzt werden, wenn die Fenstergröße (ausgedrückt als die maximale Anzahl der unbestätigten Pakete) in Abhängigkeit der Netzbelastung verändert wird.

Ratensteuerung (rate control): Beim Verbindungsaufbau können Sender und Empfänger (evtl. unter Mitwirkung des Netzes) eine zulässige Rate (gesendete Datenmenge pro Zeiteinheit) aushandeln.

Protokollmechanismen zur Übertragungsleistungsanpassung

Multiplexen/Demultiplexen (multiplexing/demultiplexing): Hierbei werden mehrere n-Verbindungen auf eine (n-1)-Verbindung abgebildet. Auf diese Weise können über eine logische Verbindung die Daten mehrerer Anwendungsprozesse übertragen werden.

Teilung/Vereinigung (inverse multiplexing): Dieser Protokollmechanismus stellt eine Umkehrung des Multiplexens dar: Eine n-Verbindung wird auf mehrere (n -1)-Verbindungen verteilt. Dies ist sinnvoll, wenn die Endsysteme eine höhere Übertragungsleistung aufweisen als eine Verbindung des Transportsystems

Nutzerbezogene Mechanismen

Verbindungsklassen: Dienste können ihre Leistungen in verschiedenen Qualitätsstufen erbringen, die als Klassen bezeichnet werden. Beim Verbindungsaufbau kann eine geeignete Klasse ausgewählt werden.

Rechtevergabe: Bestimmte Rechte können zeitabhängig auf bestimmte Benutzer beschränkt werden.

Dienstgüte-Aushandlung: Ein Initiator kann beim Verbindungsaufbau eine bestimmte Dienstgüte (z.B. Durchsatz) verlangen, die vom Basisdienst und Responder voll oder teilweise akzeptiert wird. Der akzeptierte Wert (bzw. die Werte) wird dem Initiator beim Verbindungsaufbau mitgeteilt.