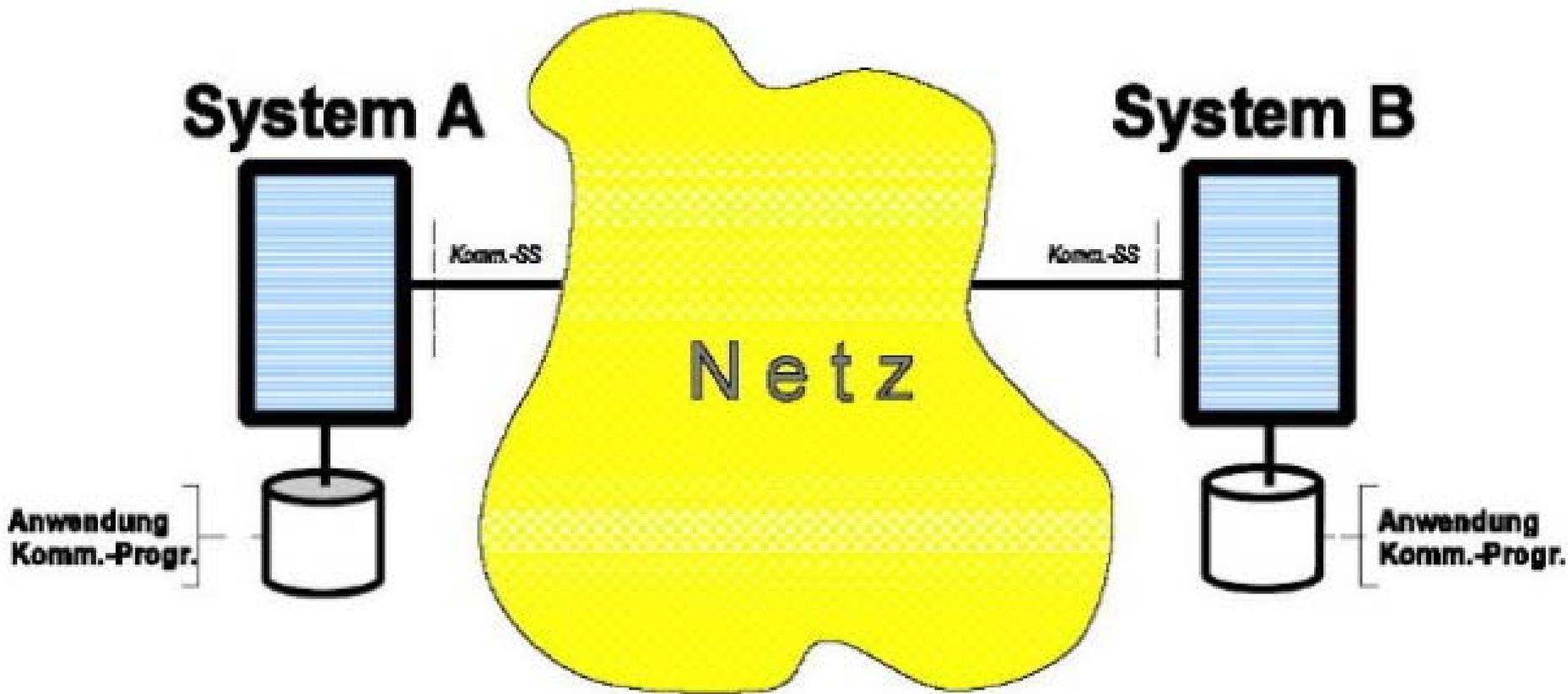
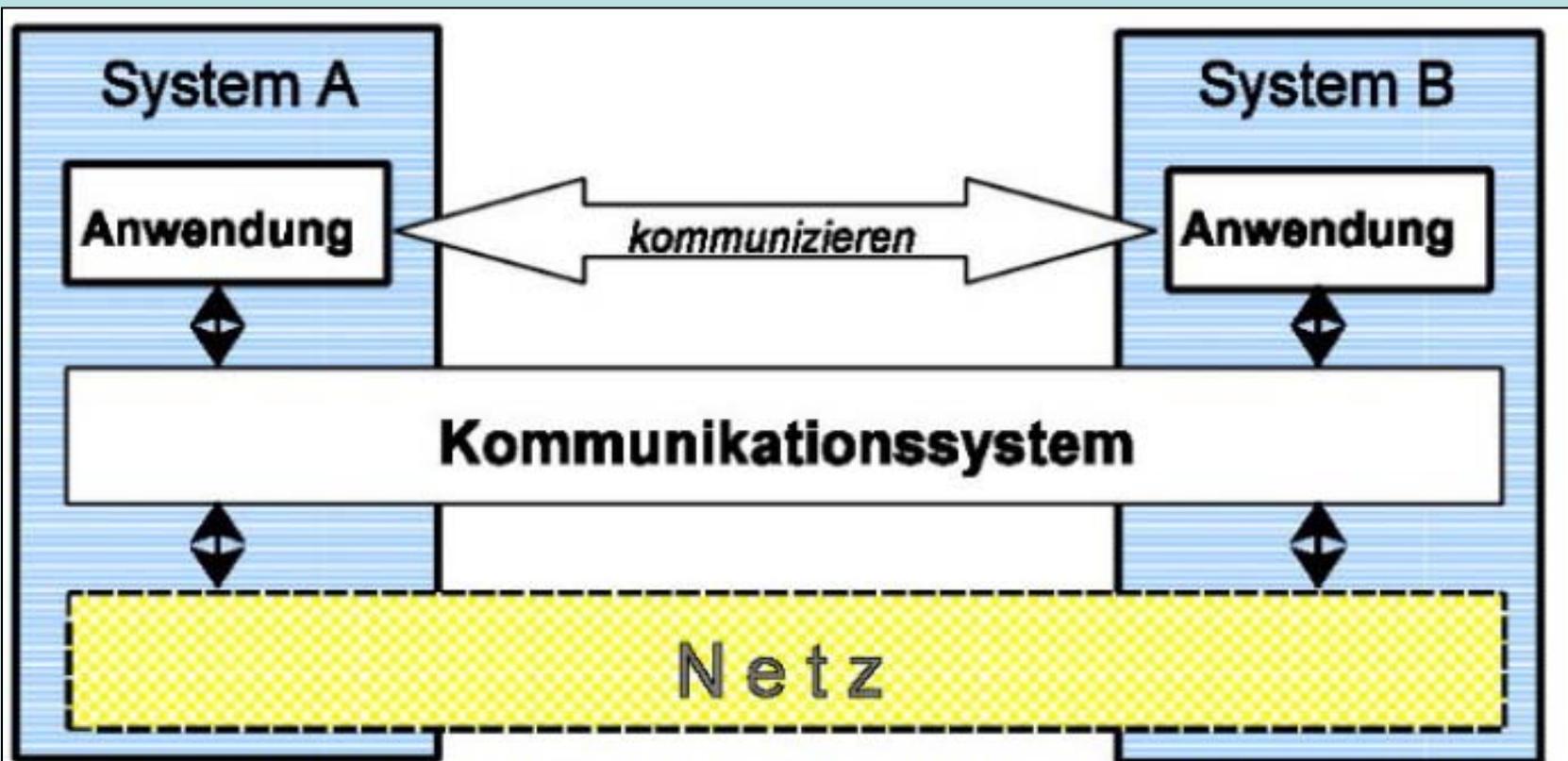


Kommunikation

**! Jetzt vollständiger !
aber noch im Fluß**



Kommunikation zweier Rechner über ein Netz: physikalische Sicht



Kommunikation zweier Rechner über ein Netz: abstrakte Sicht

Einführung

- Zwei Anwendungen (Rechner) A und B kommunizieren (Austausch von Daten)
- Reale physikalische Verbindung zwischen A und B über das Netz mit der Möglichkeit zahlreicher Knoten
- Die **Kommunikationsschnittstelle** verknüpft **Anwendung und Netz**.

Sie bereitet Daten auf, die an das Netz weitergegeben werden, bzw. arbeitet vom Netz empfangene Daten für die Anwendung.

Hierzu werden Protokolle und Dienste auf beiden Seiten A und B aufeinander abgestimmt.

Die beiden K.-Schnittstellen von A und B können daher als eine Einheit aufgefasst werden ,

dem **Kommunikationssystem**

Das Kommunikationssystem besteht aus Schaltungen und Programmen, die einen Ende-zu-Ende Datenübertragungsdienst erbringen

- **Monolithische K.-Systeme** (aus einem Guss) sind meist historisch gewachsen (propitär). Sie bestehen aus diversen K.-Protokollen, die jeweils einen bestimmten Dienst zur Verfügung stehen.

Nachteile :

Unübersichtlich, zerfleddert durch Vielzahl der Protokolle, umfangreiche Software, schwer modifizierbar

- **Modulare K.-Systeme** (Baukastensystem)
Diverse Funktionen können zu Diensten konfiguriert und kombiniert werden.

Anwendung von Objekt orientierten Technologien

Vorteile :

Transparenz, Kombinierbarkeit, Skalierbarkeit, Beherrschbarkeit von komplexeren Systemen.

ISO / OSI Referenzmodell

seit 1979 entwickelt

Auftraggeber zur Standardisierung :

US DOD (Departement of Defence)

ISO : **I**nternational **O**rganisation for **S**tandardization

OSI : **O**pen **S**ystem **I**nterconnection

Offenes System:

Abstraktes Rechnersystem (Rechner, Peripherie, Software), das für eine offene Kommunikation mit anderen offenen Systemen geeignet ist.

D.h., diese Systeme sind herstellungsunabhängig (nicht proprietär) und damit austauschbar.

Offene Kommunikation :

Kommunikation frei von Implementierungsaspekten nach genormten Verfahren (Protokolle), die allen Systemherstellern zugänglich sind.

Protokoll:

Satz von Regeln, auf den sich Kommunikationspartner geeinigt haben, damit Datenaustausch stattfinden kann.

Das OSI-Modell (ISO IS 7498) beendete das Zeitalter proprietärer Lösungen von Kommunikationsproblemen.

Prinzip des OSI Referenzmodells :

Das Model ist ein Schichtenmodell, das aus 7 Schichten besteht.

- Jede Schicht erbringt wohl definierte **Dienstleistungen**. Diese bezeichnet man auch als **Instanzen**
- Instanzen der N-ten Schicht sind **einerseits Dienstbenutzer**, indem sie Dienste der darunterliegenden (N-1)-ten Schicht aufrufen, und **andererseits selbst Diensterbringer**, da sie Dienstleistungen an die darüberliegende (N+1)-te Schicht zurückgeben. **Instanzen kommunizieren miteinander.**
- Jede Schicht kann als schwarzer Kasten angesehen werden, die Ihre Aufgaben unabhängig von den anderen Schichten löst.

Vergleich :

**Regierungschef delegiert an Minister (Diener),
Minister an Staatssekretär ,
Staatssekretär an Räte, etc.**

Die Schichten des OSI-Modells

auf deutsch

Schicht 7	Anwendungsschicht
Schicht 6	Darstellungsschicht
Schicht 5	Sitzungsschicht
Schicht 4	Transportschicht
Schicht 3	Vermittlungsschicht
Schicht 2	Sicherungsschicht
Schicht 1	Bitübertragungsschicht

auf englisch

Layer 7	Application - Layer
Layer 6	Presentation - Layer
Layer 5	Session - Layer
Layer 4	Transport - Layer
Layer 3	Network - Layer
Layer 2	Link - Layer
Layer 1	Physical - Layer

Der Weg der Kommunikation:



Aufgaben der einzelnen Schichten

- **Schicht 7 : Anwendungsschicht:**

Bereitstellung zahlreicher maßgeschneiderter Funktionalitäten ,

- **ASO : application service object**
- **CF : control function**
- **ASE : application specific elements**
- **ACSE : association control specific elements**
- **Anwendungsspezifische Protokolle**
e-mail , FTP (file transfer protokol) ,
RPC (remote procedure call)
- **API : application program interface**
Schnittstelle zwischen Anwenderschicht und
Anwendungsprogramm (ALU : application
layer user)

- **Schicht 6 : Darstellungsschicht**

Systemabhange Darstellung der Daten (z.B. ASCII , EBCDIC) werden einheitlich aufeinander abgebildet.

gemeinsame Kodierungsregeln

BER (basic encoding rules)

gemeinsame Transfersyntax

auch Datenkompression und -verschlusselung

Bei einigen Netzen ist diese schicht leer (z.B. Internet)

- **Schicht 5 : Sitzungsschicht**

Dienste fur einen organisierten und synchronisierten Datenaustausch (Steuerung der Ende-zu-Ende Kommunikation)

Einfuhrung von Wiederaufsetzpunkten, an denen bei Zusammenbruchen der Transportverbindung wieder synchronisiert werden kann.

Bei einigen Netzen wiederum leer (z.B. ARPANET, Internet)

● Schicht 4 : Transportschicht

Ende-zu-Ende Kontrolle : unterste Schicht, die eine Ende-zu-Ende Kommunikation zur Verfügung stellt.

Sie bietet den anwendungsorientierten Schichten 5-7 einen einheitlichen Zugriff, so dass diese die Eigenschaften des Kommunikationsnetzes nicht zu berücksichtigen brauchen.

- **Segmentierung von Datenpaketen**
- **Stauvermeidung**
- **5 Dienstklassen unterschiedlicher Güte (0 – 4) sind definiert (Dienstgüte , QoS : Quality of Service), z.B. Multiplexmechanismen, Fehlersicherungs- und Fehlerbehebungsverfahren**
- **Protokollbeispiele : TCP , UDP , NNTP**

Die Transportschicht kann auf beliebigem Netz , auch nicht-OSI , aufsetzen

- **Schicht 3 : Vermittlungsschicht**

Reihenfolgerichtige Datenübertragung über eine sichere transparente DEE-zu-DEE Verbindung

DEE : DatenEndEinrichtung

- **Datenübertragung, Flußmengensteuerung**
- **Wegsuche (routing) , Vermittlung ? (switching)**
Deshalb auch oberste Schicht an Netzknoten (Router)

Beispiele für Protokolle :

- **PLP : packet layer proticol**
- **IP : internet protokol**
- **SNAcP : subnet access protocol**
- **SNDcP : subnet dependent convergence protocol**
- **SNICP : subnet independent convergence protocol**

- **Schicht 2 : Sicherungsschicht**

Gewährleistung einer weitgehend fehlerfreien Übertragung (gesicherte Punkt-zu-Punkt Übertragung)

- **Bildung von Datenübertragungsblöcken , Hinzufügen von Zusatzinformationen, z.B. Kontrollbits (Rahmen)**
- **Flußsteuerung , Fehlerbehandlung (Quittierung, Wiederholungsaufforderung)**

Beispiele für Protokolle :

- **BSC : binary synchronous communication**
- **HDLC : high level data link control**
- **LAPB : link access procedure for balanced mode**

- **Schicht 1 : physikalische Schicht**

ungesicherte Punkt-zu-Punkt Übertragung

- **Leitungskodierung nach Mediumswahl**
- **Geräte- und Übertragungstechnik (Netzwerkkarten, Stecker, Antenne, Hub)**

Beispiele für Protokolle :

- **V24 : ITU-T Schnittstellenempfehlung für Fernsprechnetze**
ITU-T : International Telecommunications Union – Telecommunication
- **X24 : ITU-T Schnittstellenempfehlung für Datennetze**
- **RS-232 : recommended standard der EIA**
EIA : Electronic Industries Association

Das OSI-Modell im Überblick (siehe im Vergleich dazu das [TCP/IP-Referenzmodell](#)):

OSI-Schicht	Englisch	Einordnung	Protokollbeispiel	TCP/IP-Schicht	Einordnung	Protokollbeispiel		
7	Anwendung	Application	Anwendungs-orientiert	FTAM	Anwendung	Ende zu Ende (Multihop)	HTTP	
6	Darstellung	Presentation					FTP	
5	Kommunikation	Session		ISO 8326			HTTPS	
4	Transport	Transport	Transport-orientiert	ISO 8072	Host to Host	TCP UDP ICMP IGMP		
3	Vermittlung	Network		CLNP			Internet	IP IPX
2	Sicherung	Data Link		HDLC			Netzwerk	Punkt zu Punkt
1	Bitübertragung	Physical	X.21					

Es muss zwischen horizontaler Kommunikation zwischen Anwendungen A und B und vertikaler Kommunikation zwischen den Instanzen unterschieden werden.

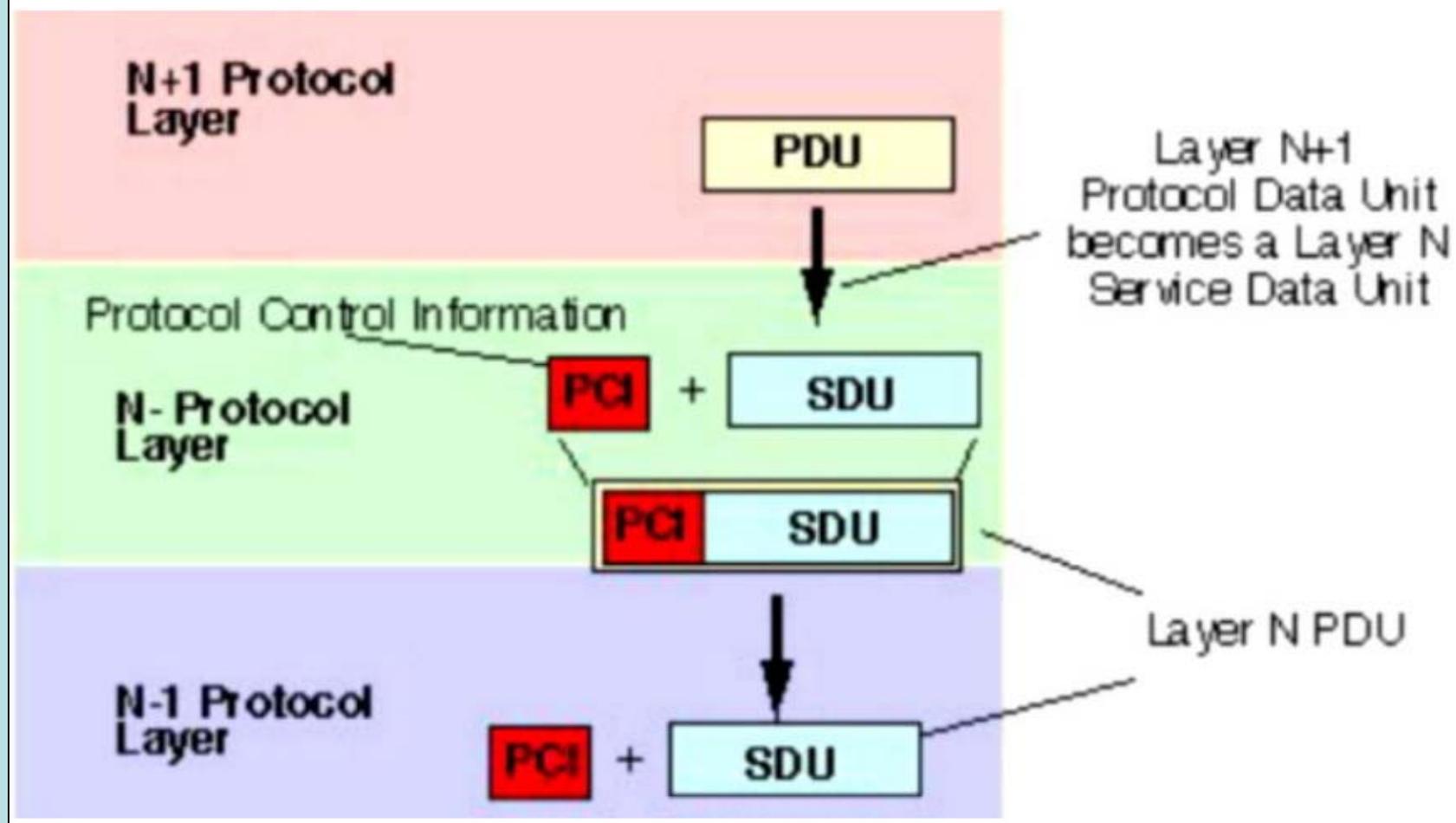
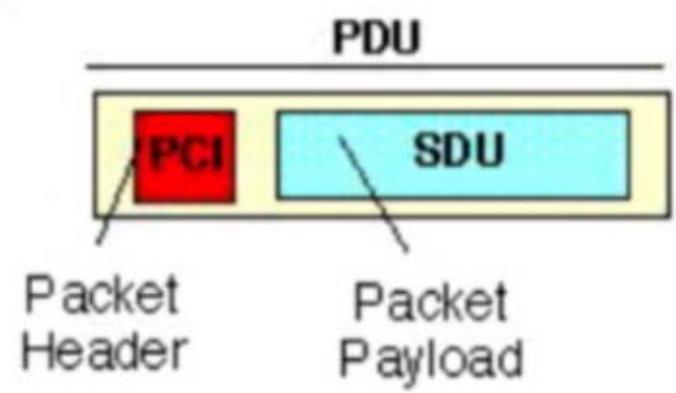
- horizontal :

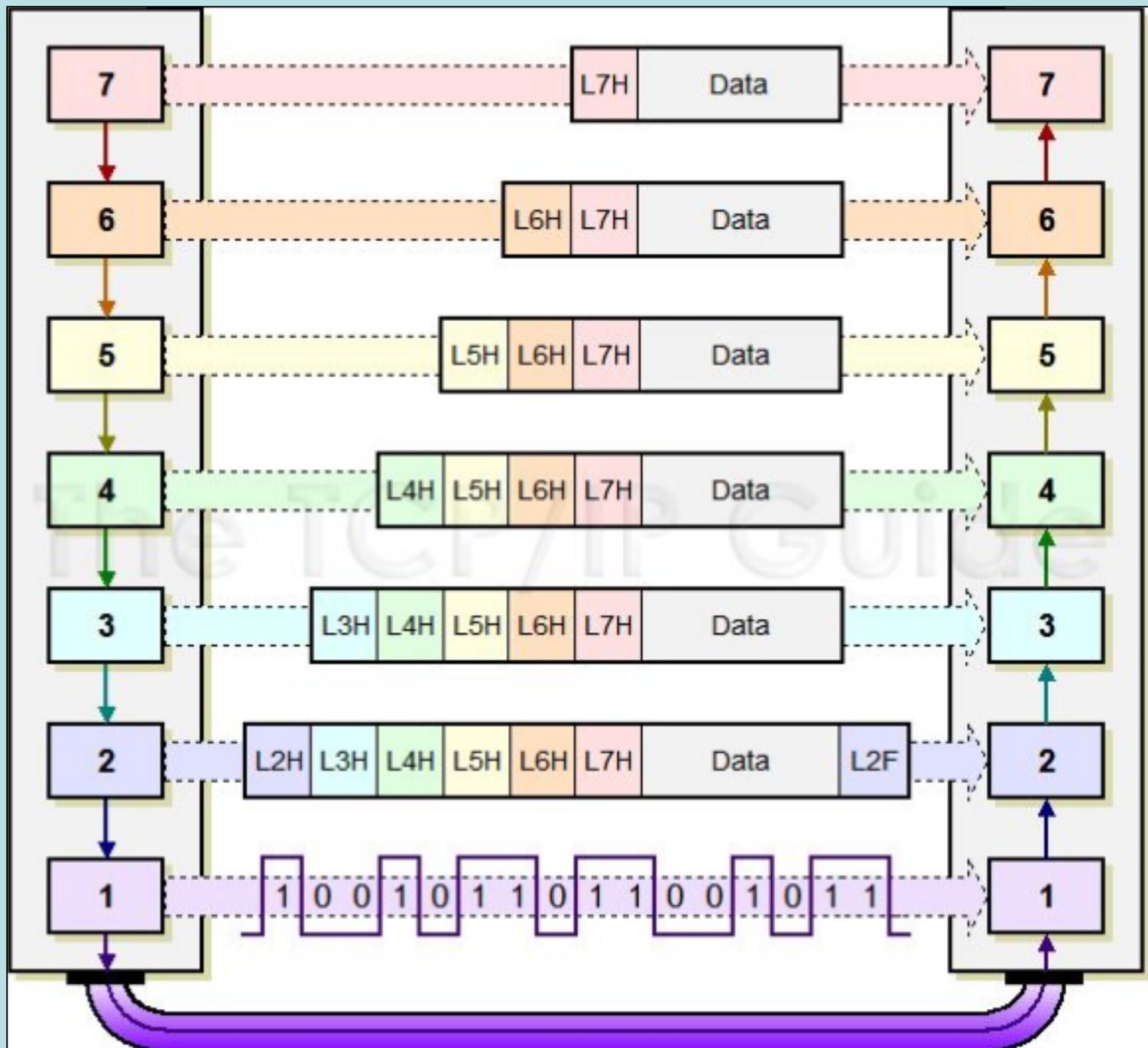
Diese Kommunikation auf einer Ebene ist nur eine **virtuelle (logische) Übertragung**. Diese Kommunikation zwischen den Partnerinstanzen von A und B wird durch die Protokolle geregelt.

Nur auf der **physikalischen Schicht findet eine reale Kommunikation** zwischen A und B statt. Es besteht immer eine direkte Verbindung, die nur aktiviert werden muss (Aktivierungsdienst).

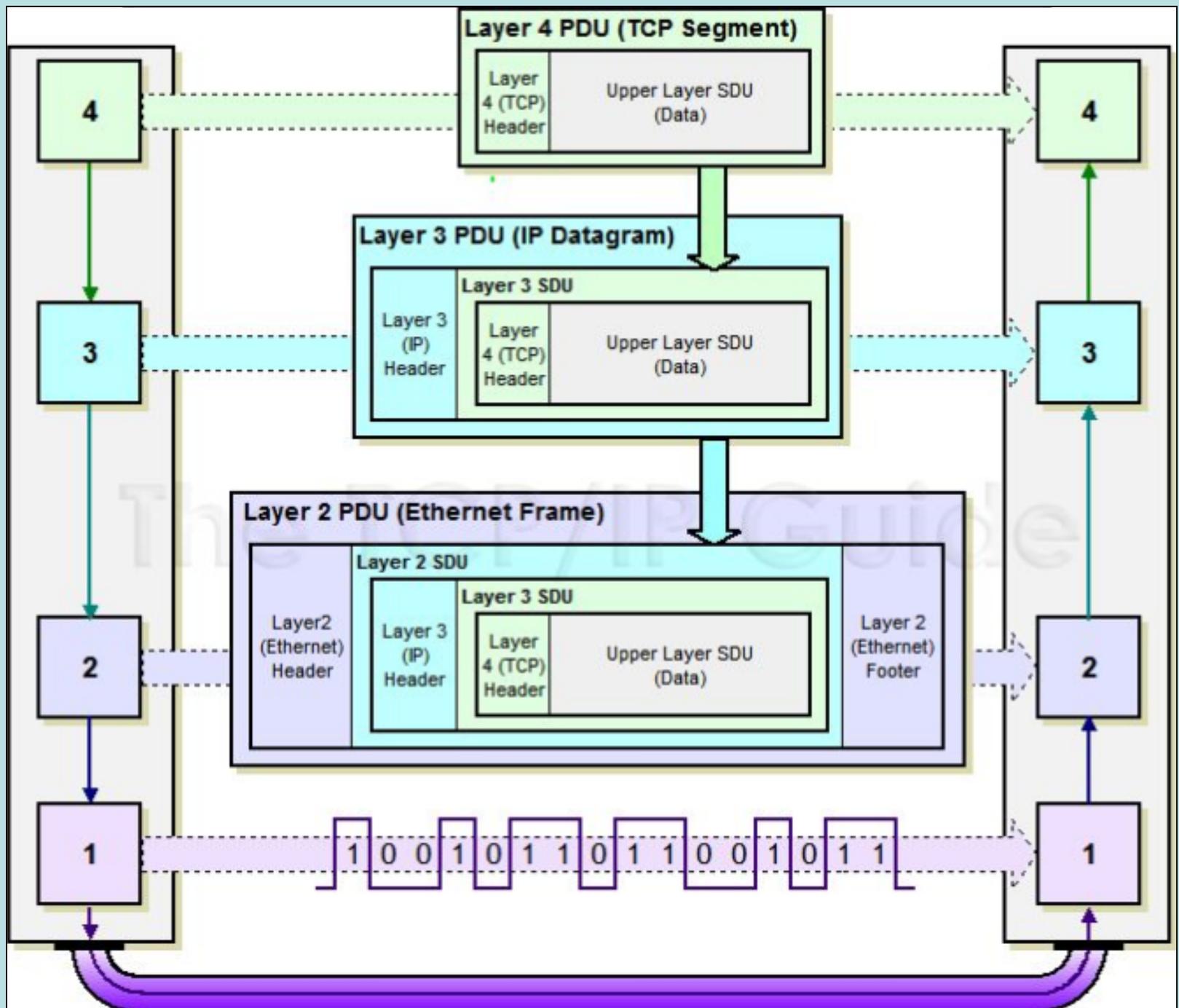
- vertikal :

Diese reale Kommunikation wird durch Dienste geregelt. Nachrichten laufen auf dem Weg von der oberen Schicht zur phys. Schicht durch die einzelnen Instanzen.

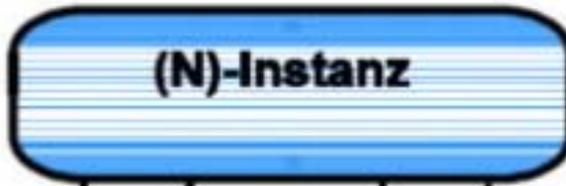




- **Kleinste Einheit der phys. Ebene ist das Bit (SDU)**
- **PDU der DL verfügt auch über einen Schwanz**
- **PhL : Antwort „ich will nicht“**
- **RTS : ready to send Steuer.....**
CTS : clear to send Melde.....
 - **auf „0“ nicht bereit**
 - **in der Regel statische Signale, aber auch als Flanken (IRQ) realisierbar.**



Schicht (N)



(N-1)_Dienst.conf

(N-1)_Dienst.ind

①

④

③

②



(N-1)_Dienst.req

(N-1)_Dienst.resp

Schicht (N-1)



Dienstprimitive:

- | | | |
|---|-------------|------------------------|
| ① | Anfrage | <i>(N)_Dienst.req</i> |
| ② | Anzeige | <i>(N)_Dienst.ind</i> |
| ③ | Antwort | <i>(N)_Dienst.resp</i> |
| ④ | Bestätigung | <i>(N)_Dienst.conf</i> |

Dienstdirektiven :

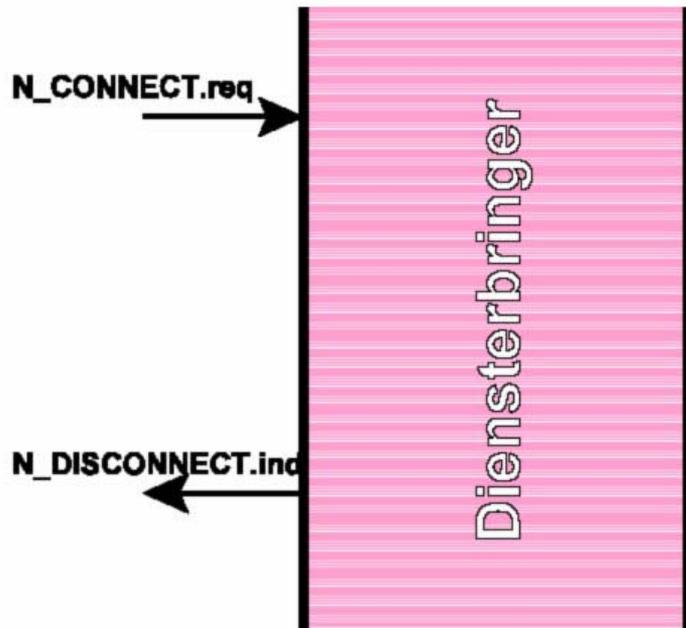
Nicht nur auf OSI beschränkt, d.h. Prinzipien überall ähnlich, nur Namen oft unterschiedlich in den einzelnen Modellen.

Internet ist unsicher, da es nicht auf OSI-Model beruht.

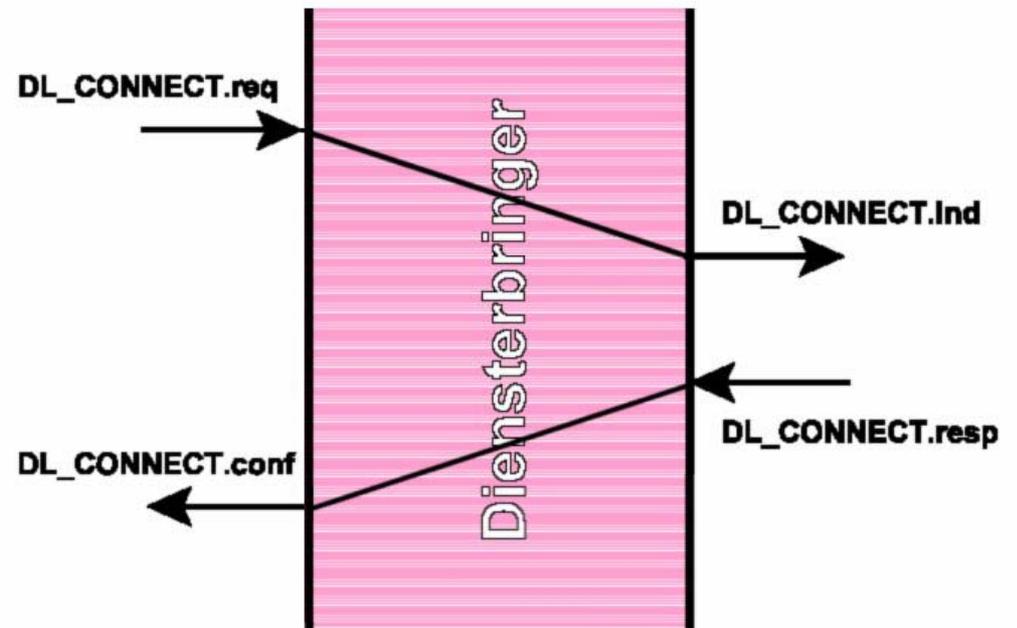
OSI : warten auf Response

- **kostet Zeit**
- **muss im gewissen Zeittakt abgefragt werden (mailbox)**
- **Schichten müssen schnell durchlaufen werden :
Assembler ist schneller als jeder denkbare Compiler
(durch Studie bewiesen).**

2 Beispiele für Zeitverlaufdiagramme



*Dienstleister lehnt
Verbindungswunsch ab*

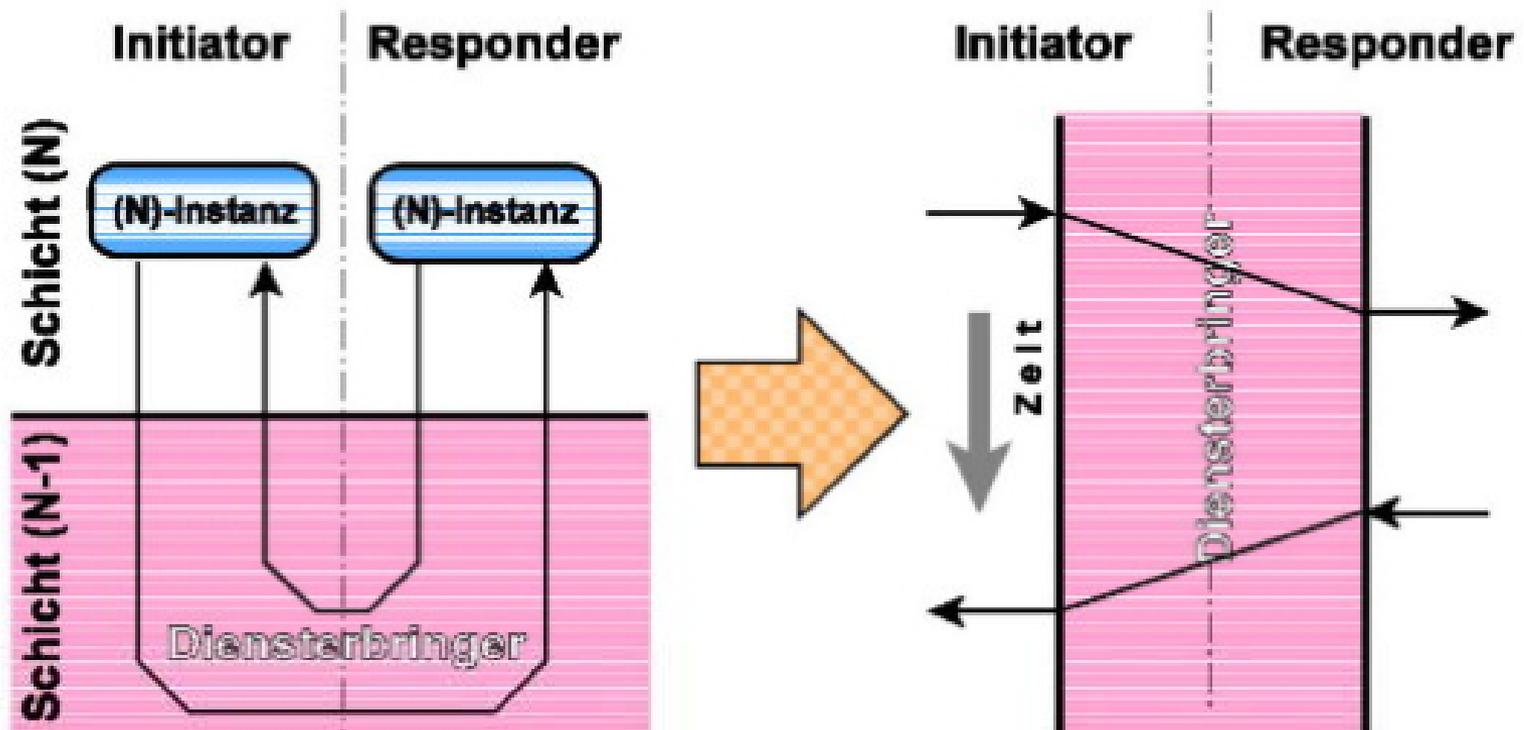


*Normaler (bestätigter)
Verbindungsaufbau*

Zeitverlaufdiagramme

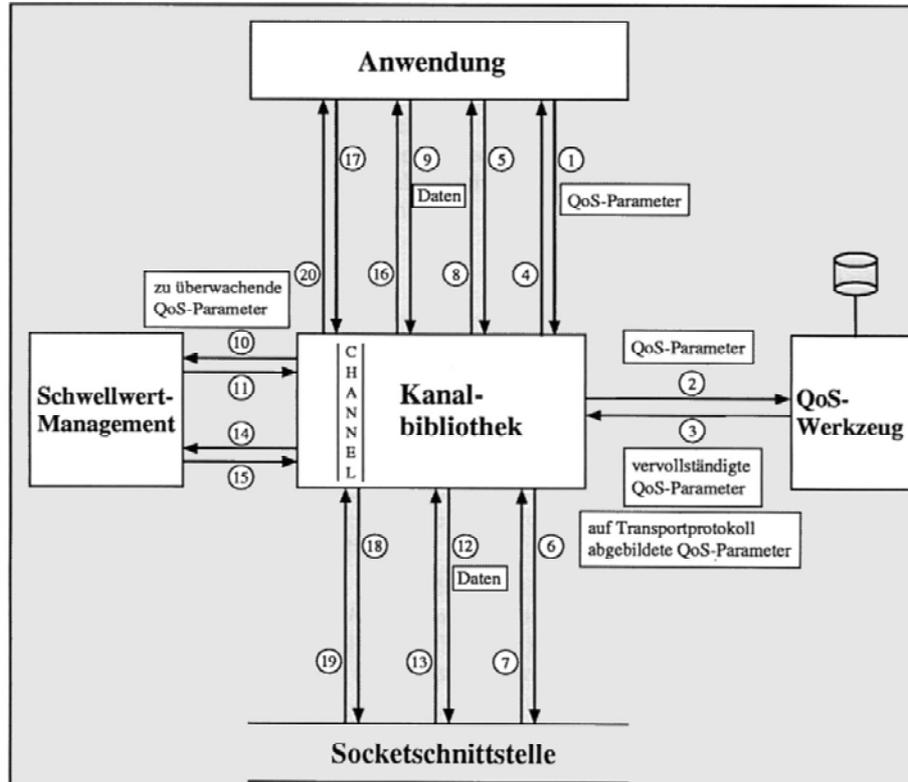
MSC: Message Sequence Charts

Zur Darstellung des Zeitverlaufs der Dienstaktionen zwischen Dienstbenutzer(n) und Dienstleister



Systemmodell zur qualitätsorientierten Datenübertragung

TU - Dresden



Legende

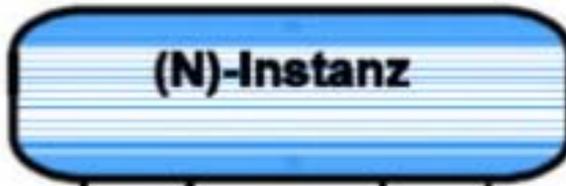
- 1-4 Erzeugung eines Kanals mit bestimmten QoS-Parametern
- 2-3 Vervollständigung der QoS-Parameter und Abbildung dieser auf das Transportprotokoll
- 5-8 Aufbau der Verbindung
- 9-16 Senden von Daten
- 10-11 Beauftragung des Schwellwertmanagements mit der Überwachung von Leistungsparametern
- 12-13 Übergabe der zu sendenden Daten an das Transportprotokoll
- 14-15 Beendigung der Schwellwertüberwachung
- 17-20 Abbau der Verbindung

PAUSE

Zu den Aufgaben

- **Simplex :**
nur eine Richtung Sender und Empfänger festgelegt.
Ausnahme : Empfangsbestätigung / Empfangsbereitschaft
- **Port B ist geeigneter für Sender, Port A dagegen für Empfänger** (beachte Besonderheit für Aufgabe 1)
- Ihre Aufgabe : nur request und indication, response und confirmation entfällt (siehe folgende Folie).
Ausnahmen können **CTS** und **Pause** am Empfänger sein.

Schicht (N)



(N-1)_Dienst.conf

(N-1)_Dienst.ind

①

④

③

②



(N-1)_Dienst.req

(N-1)_Dienst.resp

Schicht (N-1)



Dienstprimitive:

- | | | |
|---|-------------|------------------------|
| ① | Anfrage | <i>(N)_Dienst.req</i> |
| ② | Anzeige | <i>(N)_Dienst.ind</i> |
| ③ | Antwort | <i>(N)_Dienst.resp</i> |
| ④ | Bestätigung | <i>(N)_Dienst.conf</i> |

- **h-feld mit Daten** ,
die übertragen werden sollen.
- Bytes des h-feldes in zwei Tetraden teilen (**Wo ?**),
die als zu übertragene Datenpakete betrachtet
werden.
Jede der beiden Tetraden wird **im** dazugehörigen
Header numeriert.
- Ferner wird ein Paritätsbit verwendet (**Wo
hinzugefügt ?**). Führt dessen Analyse zu einer
Fehlermeldung, wird die Kommunikation gestoppt.
- Diese Übertragungspakete (Datentetrade +
Zusatzinformationen) sollen seriell übertragen
werden.

- Schichtensteuerung über den Timer (PIAT)
Überlegen Sie sich einen sinnvollen Zeitablauf !
Es sind durchaus verschiedene Realisierungen denkbar.
- **Empfehlung :**
Verwendung einer IRQT-Auftragstabelle.
Diese enthält als wesentliche Elemente Adressen von Unterprogrammen.
Diese werden wiederum indirekt adressiert (siehe folgende Folie.)
- Für die Hauptprozesse, d.h. bei den Anwendern, sind zeitrelevante Prozesse mit Warteschleifen zu realisieren.

Wie kann ein Unterprogramm indirekt aufgerufen werden ?

- **Hierzu wird die absolut indirekte Adressierung des jmp-Befehls ausgenutzt.**

Beispiel :

```

LDA #<Subr
STA Pointer           ; Programm gesteuerte Festlegung
LDA #>Subr           ; des Zeigers
STA Pointer + 1

JSR Inter_Subr       ; direkter Unterprogrammaufruf
ρ :                  ; auf Stapel gelegte Rücksprungadresse

Inter_Subr : JMP(pointer) ; indirekter Sprung zum eigentlichen UP
                RTS        ;                Rücksprung zu ρ
Subr :          ; ausgewähltes
                ; Unterprogramm

JMP Inter_Subr + 3 ; Sprung zu RTS von „Inter_Subr“ ,
                  ; da JMP(Pointer) 3 Bytes belegt.
```

Welche Vereinbarungen müssen getroffen werden ?

zwischen Sender und Empfänger (horizontal)
zur Kommunikation unter Schichten (vertikal)

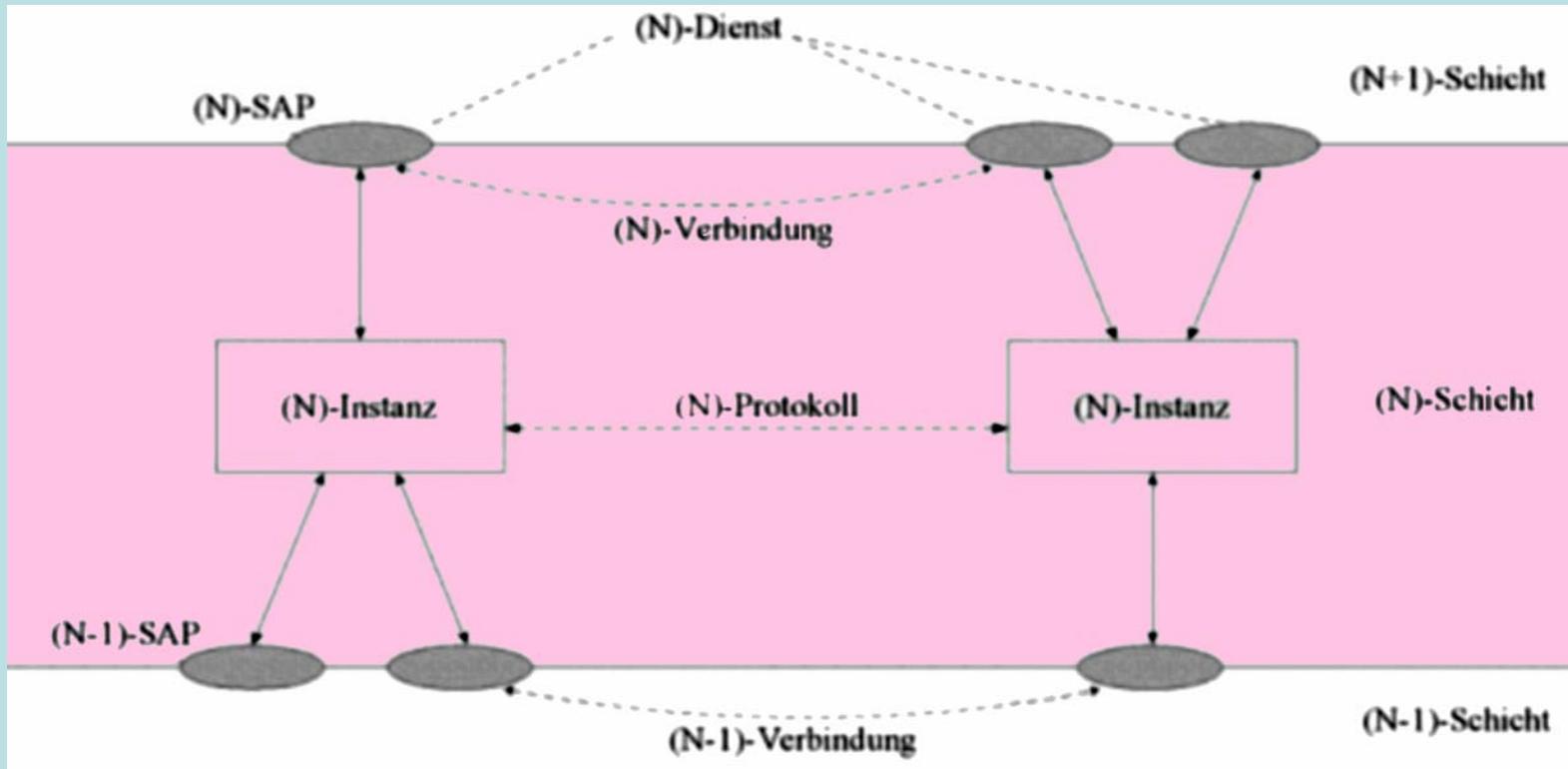
- Schnittstellen definieren (z.B. auch API (*Application Program Interface*))
- Daten in Rahmen bringen (Kopf und/oder Fuss) , d.h. Protokolle werden definiert.
- Protokollsteuerinformation vereinbaren. ICO : Info über **Dienstzugangspunkt**.
- Modell komplett vereinbaren.
- PhL : Wie auf Kabel bekommen (Aktivierungsdienst) ?

Das erfordert Teamarbeit !

Modellierung über SDL möglich

Was ist ein Dienstzugangspunkt ?

Service Access Point (SAP)



Eine Schnittstelle, über die Daten und Informationen zwischen 2 benachbarten Schichten ausgetauscht werden.

Speicherzelle(n): z.B. hfeld : Info vorne

Diensterbringer (Provider) :

Wie Dienst aufrufen ?

- **einfachste Methode : Unterprogramme streng genommen muss das anders geschehen, da umgekehrt in höhere Schicht zurückzugeben regelwidrig ist.**
- **Daten mit Zusatzinformation in Postkasten hinterlegen.
Schichten müssen im Postkasten nachschauen (pollen).**

- **Empfehlung :**

Verwenden Sie zwei Bytes als Dienstzugangspunkt

- **Byte für das Übertragungspaket** ,

das unter anderem
die Datentetradе enthält.



Datentetradе

- **Kommunikationsbyte (mailbox)** , das die
Information zur Steuerung der Schichten bzw.
Dienste enthält.

- Das Kommunikationsbyte kann auch die API
verwendet werden.

Mailbox



HP

Application Program Interface (API)

Kommunikationsbyte



Sendepuffer



AL

Kommunikationsbyte

Übertragungspaket

SAP



DL

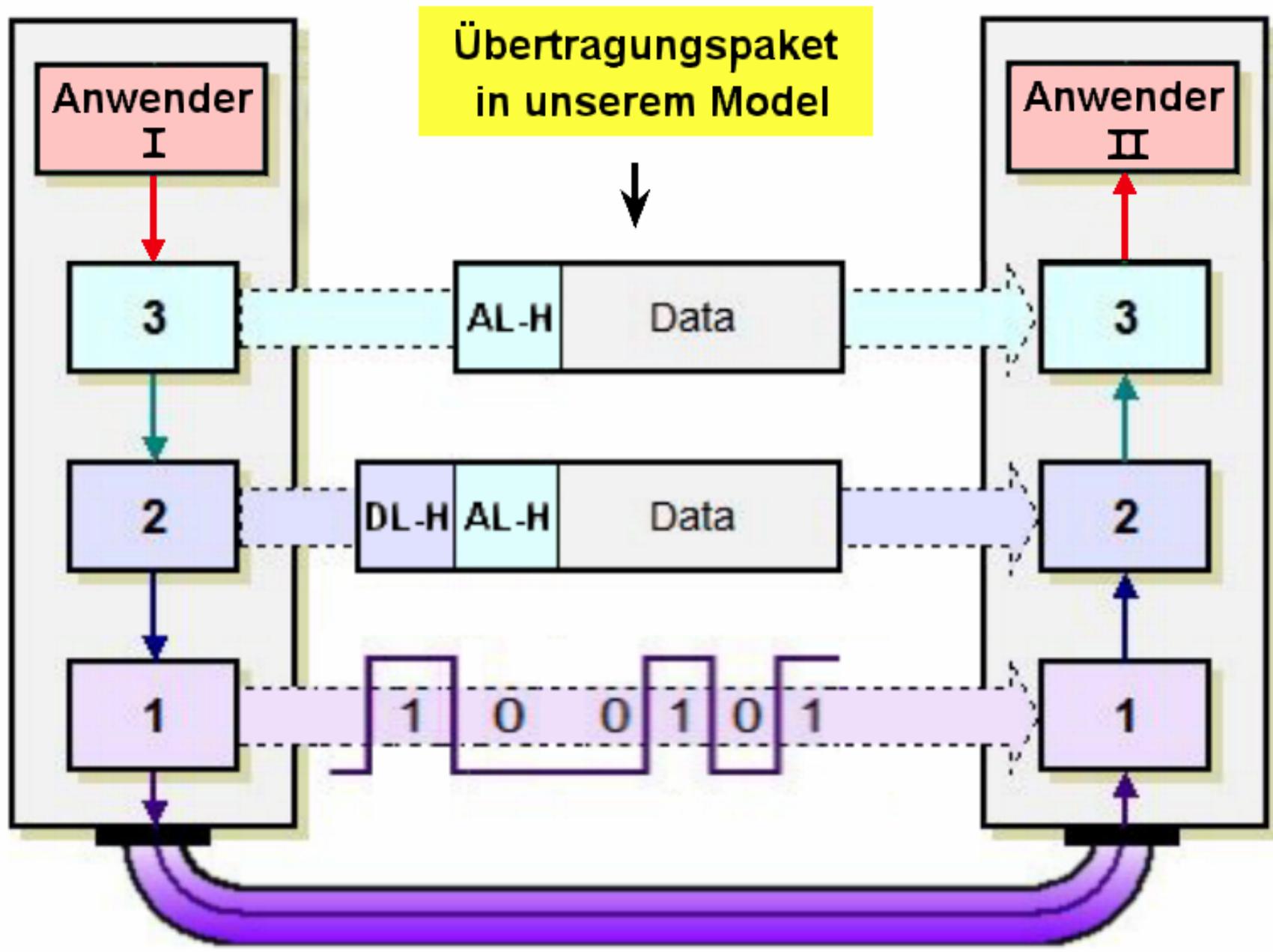
Kommunikationsbyte

Übertragungspaket

SAP



PhL



Wie realisieren ?

- **asynchron** :
auf Interrupt warten, schnell, aber problematisch wird; oft nicht unterstützt , da unsicher
- **synchron** :
Takt vorgeben : anpassen Ticken (interrupt timer)

1. Kommunikationsproblem :

- **parallele Bit** Übertragung pro Übertragungspaket
- **asynchron** mit automatischen Quittierungsbetrieb
- Da C1/C2 für den automatischen Quittierungsbetrieb verwendet werden müssen, müssen RTS und CTS gepollt werden.
- Da neben CTS und RTS auch eine Pause sowohl für Sender als auch Empfänger vorgesehen werden soll, empfiehlt es sich hier sowohl Post a als auch b zu verwenden.

2. Kommunikationsproblem :

- Übertragung in einem **UART** – ähnlichem Format (**U**niversal **A**synchronus **R**eceiver **T**ransmitter)
- **serielle Bit** Übertragung pro Übertragungspakete
- **asynchron** (mit automatischen bit-weisen Quittierungsbetrieb)

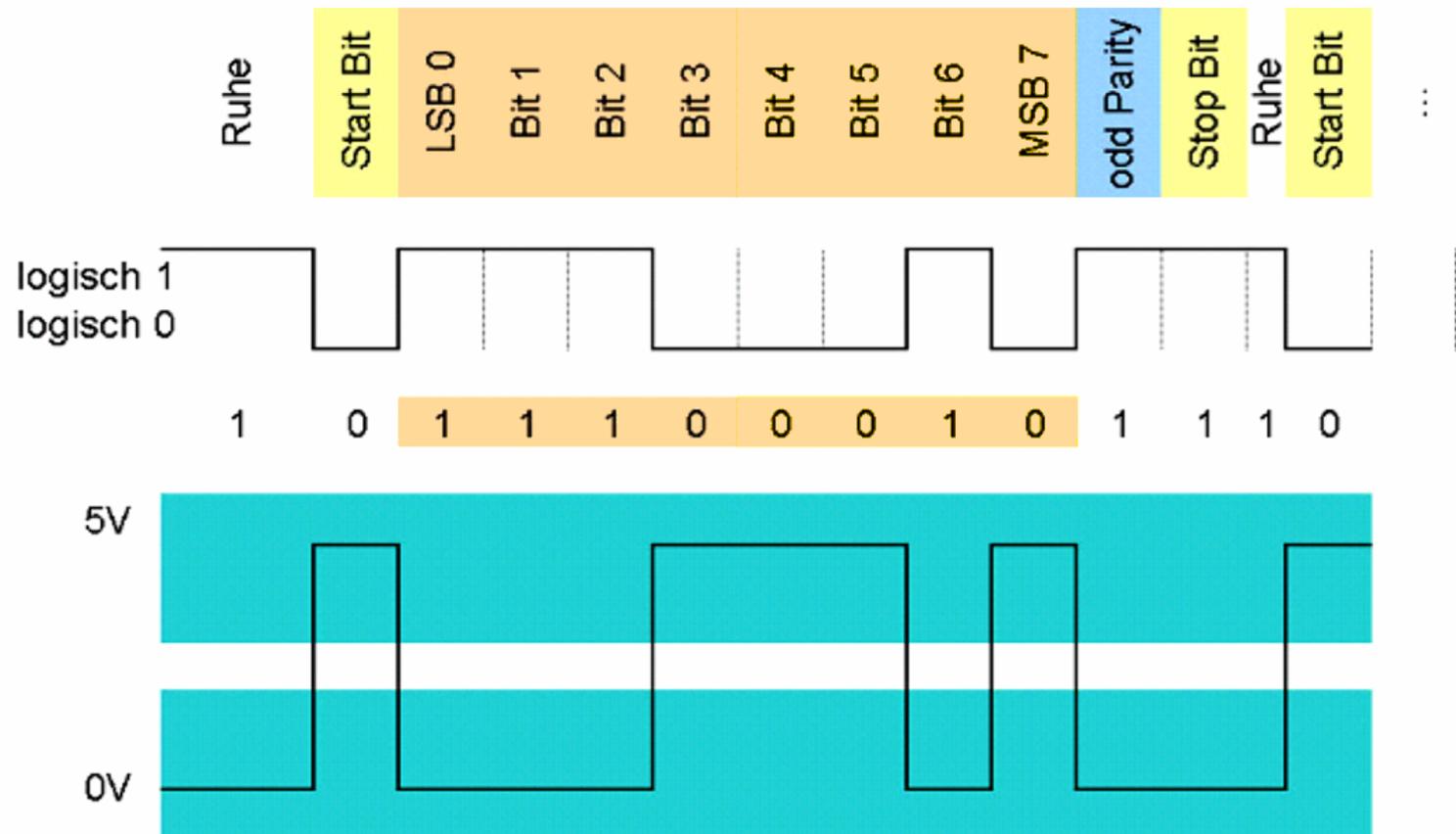
Wenn C1/C2 für den automatischen Quittierungsbetrieb verwendet werden, müssen RTS und CTS gepollt werden.

Zeitdiagramm und Spannungspegelverlauf

Synchronisation
Daten low & high
Check

1 Start Bit
8 Datenbits
odd Parity
1 Stopbit

ASCII "G" = \$47 = 0100 0111



UART Baustein :

Herkömmliche Schnittstellenbaustein :

8250 mit maximal 19.200 Bit/s Übertragungsrate

Nachfolger :

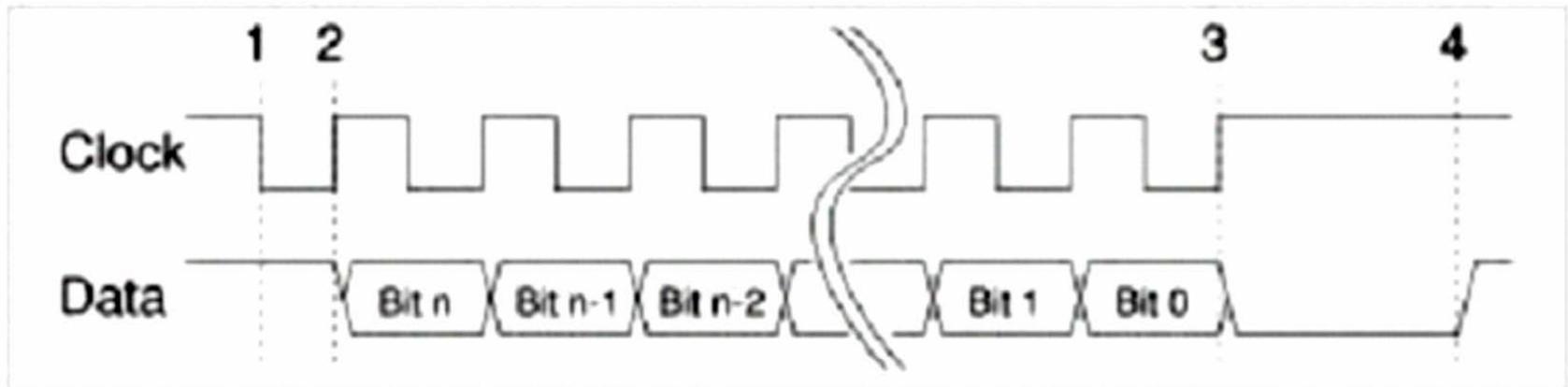
16450 mit maximal 115.200 Bit/s

UART 16550 : wie 16450, aber mit zusätzlichem 2 x16 Byte Pufferspeicher ausgestattet, die nach dem FIFO-Prinzip (First-In-First-Out) arbeiten, z.B.

- Silospeicher
- Pipe

3. Kommunikationsproblem :

- **serielle Bit** Übertragung pro Übertragungspakete
- **synchron** mit Signal zur Tetraden Rahmenbildung , Takt wird übertragen



Übertragung über die Synchron-Serielle Schnittstelle: 1. Messwert wird gespeichert. 2. Ausgabe des ersten Datenbits. 3. Alle Datenbits sind übertragen, Monoflop beginnt abzulaufen. 4. Monoflop fällt in seinen Grundzustand, eine neue Übertragung kann gestartet werden.

