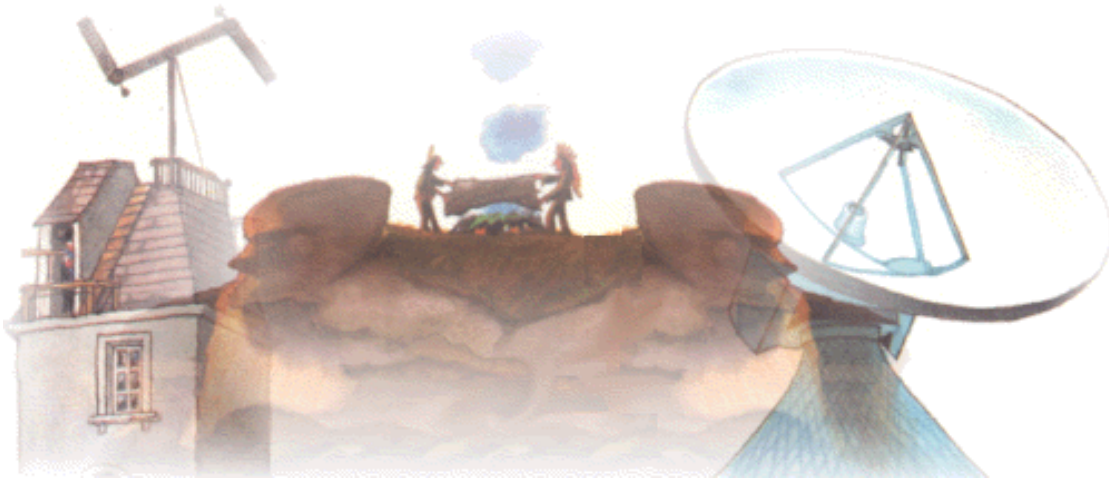


Kommunikationstechnik



Prof. Dr. Werner Erhard

WS 2004/05

Jörg Sommer

Vorwort

Dieses Dokument wurde als Skript für die auf der Titelseite genannte Vorlesung erstellt und wird jetzt im Rahmen des Projekts „[Vorlesungsskripte der Fakultät für Mathematik und Informatik](#)“ weiter betreut. Das Dokument wurde nach bestem Wissen und Gewissen angefertigt. Dennoch garantiert weder der auf der Titelseite genannte Dozent, die Personen, die an dem Dokument mitgewirkt haben, noch die Mitglieder des Projekts für dessen Fehlerfreiheit. Für etwaige Fehler und dessen Folgen wird von keiner der genannten Personen eine Haftung übernommen. Es steht jeder Person frei, dieses Dokument zu lesen, zu verändern oder auf anderen Medien verfügbar zu machen, solange ein Verweis auf die Internetadresse des Projekts <http://uni-skripte.lug-jena.de/> enthalten ist.

Diese Ausgabe trägt die Versionsnummer 2934 und ist vom 27. März 2010. Eine neue Ausgabe könnte auf der Webseite des Projekts verfügbar sein.

Jeder ist dazu aufgerufen, Verbesserungen, Erweiterungen und Fehlerkorrekturen für das Skript einzureichen bzw. zu melden oder diese selbst einzupflegen – einfach eine E-Mail an die [Mailingliste <uni-skripte@lug-jena.de>](mailto:uni-skripte@lug-jena.de) senden. Weitere Informationen sind unter der oben genannten Internetadresse verfügbar.

Hiermit möchten wir allen Personen, die an diesem Skript mitgewirkt haben, vielmals danken:

- [Jörg Sommer <joerg@alea.gnuu.de>](mailto:joerg@alea.gnuu.de) (2004/5)

Inhaltsverzeichnis

1	Begriffsbildung und Übersicht	6
1.1	Kommunikation, Kommunikationssystem	6
1.2	Nachricht, Information	7
1.3	Wie nehmen wir Information wahr?	7
1.4	Kommunikationsmedien	8
2	Aufgabenstellung und Klassifikation	9
2.1	Technische Grundlagen	10
2.1.1	Übertragungswege	10
2.1.2	Übertragungsprinzipen	11
2.2	Abstimmung von Sender und Empfänger – Synchronisation	11
2.2.1	Asynchronbetrieb (Start-Stop-Betrieb)	11
2.2.2	Synchron-Betrieb	12
2.2.3	Synchronisation über einen Datenblock	12
3	Wirtschaftliche Ausnutzung von Übertragungskapazitäten	13
3.0.4	Frequenzmultiplex (Breitbandtechnik)	13
3.0.5	Zeitmultiplex	13
4	Schichtenmodell versus Modularisierung	16
4.1	Modularisierung (Dekomposition)	16
4.1.1	Schichtenmodell/ Hierarchiebildung	16
4.2	CSP – Communicating sequential processes	17
5	Grundlagen der Kommunikationsprotokolle	24
5.1	Entwicklungspasen für Kommunikationsprotokolle	24
5.2	A closer look to ISO/OSI	24
5.2.1	Bestätigter Dienst	26
5.2.2	Welche Probleme treten bei der Kommunikation unabhängig von irgendwelchen Protokollen immer wieder auf?	26
5.2.3	Fehlererkennung und -korrektur	29
5.2.4	Identifikation mit Sequenznummern	33
5.2.5	Flußkontrolle	33
6	Verkehrlenkung (Routing)	35
6.1	Nichtadaptive Verfahren	35

6.2	Adaptive Verfahren	36
6.2.1	Lastbegrenzung	37
6.2.2	Lastbegrenzungsmethoden	39
6.3	Netzübergang	39
7	Leistungsbewertung	40
7.1	Graphentheoretische Verfahren	40
7.2	Verkehrstheoretische Verfahren	40
7.2.1	Wiegroß ist P_0 ?	40
7.2.2	Wieviel Pufferplätze sollen zur Verfügung gestellt werden?	41
7.2.3	Berechnung der mittleren Warteschlangenlänge \bar{Q}	41
7.2.4	Berechnung der effektiven Übertragungszeit	41
8	Beispiele von Standardprotokollen	42

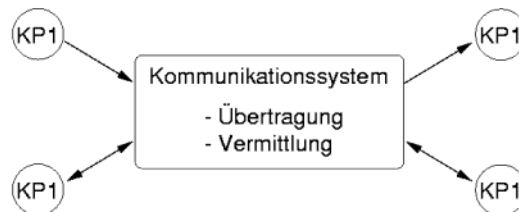
1 Begriffsbildung und Übersicht

1.1 Kommunikation, Kommunikationssystem

Unter *Kommunikation* versteht man einen Informationsfluß, der letztlich zum Austausch von Nachrichten und Informationen (siehe dazu [Abschnitt 1.2](#)) führt. Kommunikation gibt es:

- zwischen verschiedenen Organismen bzw. Maschinen untereinander,
- innerhalb von Organismen oder Maschinen,
- zwischen Organismen und Maschinen.

Ein *Kommunikationssystem* ist ein System zum Austausch von Nachrichten zwischen verschiedenen Kommunikationspartnern:



Je nach ihren Fähigkeiten können die Kommunikationspartner in vier verschiedene (Komplexitäts-)Klassen eingeteilt werden, nämlich in solche die

1. nur empfangen
2. senden und empfangen
3. senden, empfangen und zwischenspeichern
4. senden, empfangen, zwischenspeichern und verarbeiten

1.2 Nachricht, Information

Hierunter versteht man jede Art von *Nachricht*. Die kleinste Einheit der *Information* ist das *Bit*, die der Kommunikation ist die Anzahl der übertragenen Bits pro Zeiteinheit (bit/s oder *bps* für bits per second). Sehr gebräuchlich und bei den heutigen Möglichkeiten sinnvoll sind auch:

- KB/s oder *Kbps* z. B. ISDN mit 64 Kbps
- MB/s oder *Mbps* z. B. Ethernet mit bis zu 1 Mbps
- GB/s oder *Gbps*

Um eine Information interpretieren zu können, muß bekannt sein, wie diese aufgebaut ist. Der *Syntax* ist die formale Beschreibung dieses Aufbaus. Die *Semantik* steht für die inhaltliche Bedeutung der Information, die (bei schlechter Syntax) nicht eindeutig sein muß.

Beispielsweise steht die phonetische Lautschrift i:gl im deutschen für (das Tier) Igel und im englischen für (das Tier) eagle, wobei es sich aber um einen Adler handelt. Die *Pragmatik* der Information, beschreibt deren Auswirkung auf das Verhalten des Empfängers. Repräsentiert werden kann Information physikalisch durch:

- Elektrizität (elektrisch/elektronisch)
- Licht (optisch)
- Schall (akustisch)

1.3 Wie nehmen wir Information wahr?

Die Informationsaufnahme beim Menschen läßt sich in folgende Schritte einteilen:

1. Informationsaufnahme (unbewußt)
2. Reduktion (unbewußt)
3. Erkennen/Wahrnehmen (bewußt)
4. Kurzspeichern (bewußt)
5. Verarbeiten (bewußt)
6. Verstehen (bewußt)
7. Reagieren (bewußt)

Beim letzten Schritt (Reagieren) werden Informationen in Form von Sprache, Gestik, Mimik, Motorik etc. abgegeben. Dabei entsteht ein Informationsfluß von ca. 107 bps. Entsprechende Größenordnungen für die Aufnahme von Informationen zeigt die folgenden

Tabelle:

Auge (Licht)				
Ohr (Ton)	3*10 ⁴	10 ⁴	Nase (Geruch) 10 ⁷	10 ²
			Zunge (Geschmack) 10 ⁷	10
				Haut (Wärm

1.4 Kommunikationsmedien

Kommunikation ist möglich über räumliche und zeitliche Distanz. Man unterscheidet dabei Individual- und Massenkommunikation:

Komm.-Mittel Komm.-Type	Sprache	Bilder	Schrift
Individuelle Komm.	Telefon (Welt: 1 Mrd.)	Bildtelefon, Telefax, Foto, Video	Brief, Telefax, BTX
Massenkomm.	Radio, Tonträger	Fernsehen, Kino	Buch, Zeitung

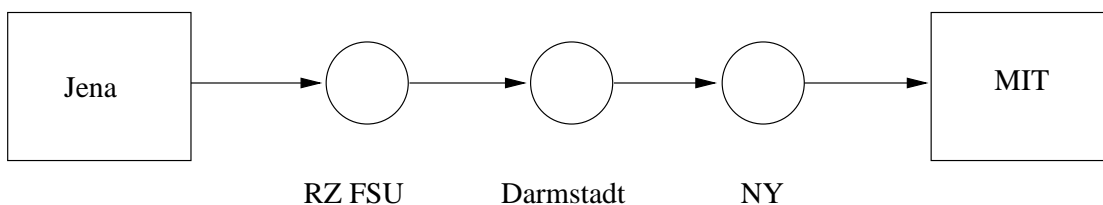
Das Internet paßt dabei in jedes Feld! Typische Anforderungen bei der Individualkommunikation zeigt die folgende Tabelle:

1. mail: ≤ 64 kbit
2. video on demand: 10 Mbps – 1 Gbps
3. Menge von Daten:
 - a) Kurzdialog: wenige 100 Bytes; eMails, Kontenautomat, Meßwerke
 - b) Langdialog: 60 kB – 1 MB; Datenbankrecherche
 - c) Stapelverkehr: >1 MB, Downloads

2 Aufgabenstellung und Klassifikation

Die Aufgabe eines Datenkommunikationssystems ist die Übermittlung (Übertragung + Vermittlung) und teilweise auch die Verarbeitung von Datenströmen zwischen einzelnen Teilnehmern des Systems.

Beispiel für die Verarbeitung: Aufbau einer Verbindung von Jena zum MIT



Die Qualität der Datenübermittlung hängt ab von:

1. Kosten: Investitionskosten, laufende Kosten, Wartungskosten, Ersatzkosten
2. Funktionalität: Anzahl/Art der Kommunikationsdienste (Schlagwort: Multimedia)
3. Verkehrsgüte: Reibungsloser Ablauf des Informationsaustausch
 - a) Durchsatz (Zahl der Aufträge, die pro Zeiteinheit im Mittel/Maximum ausgeführt werden)
 - b) Wartezeit (Zeit, die eine Anforderung im System bzw. pro Station warten muss)
 - c) Verweil- oder Durchlaufzeit (Summe aller Warte-, Bearbeitungs- und Transportzeiten im System für einen bestimmten Auftrag)
4. Übertragungsgüte
 - a) Bit-/Blockfehlerwahrscheinlichkeit (WK, dass ein Bit bzw. eine Menge von Bits (Block) in der Übertragung gestört ist)
 - b) Effektive Übertragungszeit (Zeit, die notwendig ist, die Nachricht korrekt zu übertragen)
5. Zuverlässigkeit (möglichst wenig Ausfälle des Gesamtsystems oder von Komponenten)

a) Ausfallwahrscheinlichkeit (WK, dass das System bzw. die Systemkomponente ausfällt)

b) Verfügbarkeit = $\frac{MTBF}{MTBF+MTTR}$

MTBF: Mean timebetween failure, MTTR: Mean time to repair

6. Robustheit

a) Überlastungsfähigkeit

b) Übertragungssicherheit (Empfindlichkeit gegen Fehler)

c) Ausfallsicherheit

7. Erweiterbarkeit/Flexibilität (Möglichkeit der dynamischen Anpassung des Netzes bei erhöhten Verkehrsbedürfnis oder verändertem Verkehrscharakter)

8. Benutzerfreundlichkeit

2.1 Technische Grundlagen

2.1.1 Übertragungswege

Kabelgebunden • Lichtwellenleiter (LWL): 10-100 μm

Vorteil: extrem Störsicher, Abhöricher, leicht zu verlegen, Datenraten bis GBit/Tbit

Nachteil: höhere Kabelkosten (als Kupferkabel), teure Anschlusskosten

• 2-Drahtleitung (twisted pair): Kupferkabel, Dicke im Millimeterbereich

Vorteil: niedrige Kosten, oftmals bereits vorhanden, gut verlegbar

Nachteil: hohe Störemfindlichkeit, Datenraten \leq 100 Mbit, nur einige 100 m, mit Verstärkern 30 km Entfernung möglich

• Koaxialkabel:

Vorteil: Störsicherheit, Datenrate bis Gbit-Bereich, bei 100 Mbit bis 20 km Entfernung (ohne Verstärker)

Nachteil: höhere Kosten (10-fache eines Kupferkabels), Verlegung kompliziert, teure Anschlusstechnik

Kabellos • Satellitenübertragung

Vorteil: Flexibilität, hohe Benutzerraten, Datenrate im GBit/TBit-Bereich

Nachteil: extrem hohe Kosten, Laufzeit nicht vernachlässigbar

- Laser
- Infrarot, Bluetooth (genehmigungsfrei)

Vorteil: geringe Kosten (da Massenprodukt), flexibel

Nachteil: geringe Datenraten (z. B. Infrarot bis 5 MBit), störanfällig, extrem unsicher

2.1.2 Übertragungsprinzen

Basisbandübertragung

- Frequenzband ist begrenzt $0 - f_{max} < 5 \text{ kHz}$
- gleichmäßige Übertragungsqualität von niedrigen bis hohen Frequenzen erforderlich

Trägerbandübertragung

- gleichmäßige Übertragungsqualität im Bereich $f_{min} - f_{max}$ erforderlich, Frequenzmodulation mit $f_2 = c f_1$, z. B. 3,75 MHz, 7,5 MHz

notwendig: Modem (Modulator/Demodulator)

2.2 Abstimmung von Sender und Empfänger – Synchronisation

2.2.1 Asynchronbetrieb (Start-Stop-Betrieb)

- zeitlich willkürliche Übertragung der Signale
- Synchronisation zwischen Sender und Empfänger nur für die Dauer eines Zeitraums

Startbit, 5-8 Bits, Paritätsbits, 1, $1\frac{1}{2}$ oder 2 Stopbits

Sendetakt — — — Empfangstakt um eine halbe Periode verschoben

Vorteil geringer techni. Aufwand → billig

Nachteil schlechte Leitungsnutzung

2.2.2 Synchron-Betrieb

- Zentraler Taktgeber notwendig → räumliche Begrenzung
- Synchronisation für die gesamte Dauer der Kommunikation

Vorteil: hohe Nutzinformativdichte

Nachteil: teuer

2.2.3 Synchronisation über einen Datenblock

Vorraussetzung: Taktgeber bei Sender und Empfänger laufen für einen Datenblock gemeinsam

SYN — STX — Daten — ETX — CRC (—SYN)

SYN: Synchronisationszeichen STX: Start of text – Nachrichtenbeginn ETX: End of text – Nachrichtenende CRC: Cyclic redundancy check (Fehlerinformation)

3 Wirtschaftliche Ausnutzung von Übertragungskapazitäten

3.0.4 Frequenzmultiplex (Breitbandtechnik)

Die Übertragungsfrequenz $0 - f_{max}$ wird aufgeteilt in $f_{min_1} - f_{max_1}, f_{min_2} - f_{max_2}, f_{min_3} - f_{max_3}, \dots$ (Die einzelnen Bereiche müssen disjunkt sein)

3.0.5 Zeitmultiplex

Die Eingänge werden zyklisch abgetastet und

Post: PCM 30 (puls code modulation; 30 = 30 Anschlüsse)

zyklische Zuordnung STM - Synchronous Transfer Mode

statistische Zuordnung ATM Paketform: SYN Adresse, Daten, CRC, SYN

Betriebsarten auf einer Verbindung:

Simplex-Betrieb: $A \rightarrow B$ oder $B \rightarrow A$ (Sender und Empfänger sind für die gesamte Verbindung fest)

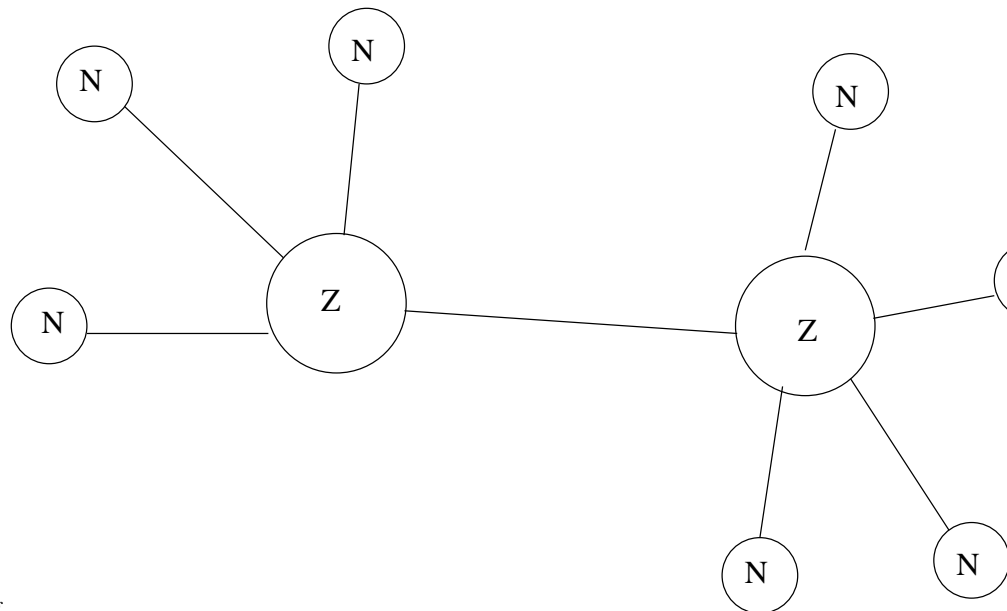
Halbduplex-Betrieb: $A \rightarrow B$ oder $B \rightarrow A$, wechselseitig, A und B tauschen die Rolle von Sender und Empfänger während der Verbindung

Duplex-Betrieb: $A \rightarrow B$ und $B \rightarrow A$, gleichzeitig, A und B sind Sender und Empfänger zugleich

Klassifikation von Vermittlungssystemen

Entfernung zwischen den Stationen

Entfernung	Wo?	übliche Bezeichnung	Fachbezeichnung
$-10^{-2} m$	Chip	integrierter Bus	SAN (Small area network)
$-10^{-1} m$	Platine	Leiterplattenbus	SAN
$-10^0 m$	Multiprozessoren	Front-/Endbus	SAN
$-10^1 m$	Büro/Labor	Labor-/Clusterbus	Fieldbus
$-10^2 m$	Campus/Fabrik	Feldbus	Fieldbus
$-10^3 m$	Universität	lokales Netz	LAN (PBX – privat branch exchange)
$-10^4 m$	Stadtbereich	lokales Netz	LAN
$-10^5 m$	Land	DFÜ-/Weitverkehrsnetz	WAN (wide), MAN (metropolitan)
$> 10^5 m$	Kontinent	Verbund von DFÜ-Netzen	TAN (terrestrial), SAN (satellite)



Ort der Vermittlung

dezentral: Alle Stationen verwalten gemeinsam den Datenverkehr Grafik Bus

Art der Vermittlung Leitungsvermittlung (Kanal-/Durchschaltvermittlung, Line-/Circuit switching)

1. Vollständiges Durchschalten des gesamten Weges (End-zu-End-Verbindung)
2. Übertragung der Daten
3. Abbau der Verbindung

Speichervermittlung (Teilstreckenvermittlung, Store and forward notwendig)

1. Aufbau einer *virtuellen* Verbindung
2. Datenübertragung von Knoten zu Knoten
3. Zwischenspeichern im Knoten und Quittung

Unterklassifikation:

- Nachrichtenvermittlung (message switching) Übertragung der Nachricht in einem Block: H — Nachricht — T
- Paketvermittlung (packet switching)
Nachricht Paket1, Paket2, Paket3 H1 — Paket 1 — T1, H2 —Paket2 —T2,
H3— Paket3—T3

Anwendungszweck

4 Schichtenmodell versus Modularisierung

4.1 Modularisierung (Dekomposition)

Zerlegung eines Systems in Teile, die diejenigen Komponenten enthalten, die in einem logischen Zusammenhang stehen. Die Verbindungstelle zwischen zwei Modulen bezeichnet man als *Schnittstelle*, die Regeln für den Informationsaustausch heißen *Protokolle*.

Kriterien für Modularisierung:

Abgeschlossenheit Jedes Modul übernimmt nur eine Teilaufgabe und diese Teilaufgabe wird nur von diesem Modul gemacht.

Handhabbarkeit Die Größe des Moduls muss überschaubar sein

Prüfbarkeit das korrekte Verhalten eines Moduls kann unter Berücksichtigung der Schnittstellen- und Protokollspezifikation validiert bzw. verifiziert werden.

Geheimnisprinzip/Virtualität Wichtig ist, was geleistet wird, nicht wie es geleistet wird.

Gegenseitige Nichtbeeinflussung Änderungen in einem Modul dürfen sich nicht auf andere Module auswirken.

4.1.1 Schichtenmodell/ Hierarchiebildung

Festlegung einer Rangordnung für einzelne Aufgaben oder Dienste und Zusammenfassung und in einem Schichtenmodell. Die einzelnen Hierarchieebenen sind funktionell abgeschlossen. Dienste werden erbracht

1. durch eigene Leistungen innerhalb der Schicht
2. durch Inanspruchnahme von Diensten niedrigerer Schichten

Beispiel 4.1.1

ISO-OSI-Referenzmodell

Schicht		
7	Anwendungsschicht/ application layer	Dienste, Anwendungsentstützung (ftp, rpc,...)
6	Darstellungsschicht/ presentation layer	einheitliche Informationsdarstellung
5	Kommunikations-Steuerschicht/ session layer	Organisation und Steuerung des Dialogs (Zugangskontrolle)
4	transport layer	End-zu-End-Kontrolle zuverlässiger und effizienter Datenübertragung
3	Vermittlungsschicht/ network layer	Wegewahl und Flusskontrolle
2	Sicherungsschicht/ data link layer	Auf- bzw. Abbau einzelner Übertragungsabschnitte, Fehlererkennung
1	Bitübertragungsschicht/ physical layer	mechanische, elektrische und funktionelle Steuerung der Übertragung

5–7 nennt man die anwendungsorientierten Schichten, 1–4 nennt man die transportorientierten Schichten

1. Jede Schicht realisiert einen oder mehrere Dienste mithilfe von Prozessen (Instanzen o. ä.)
2. Zur Erbringung eines Dienstes können Instanzen *einer* Schicht miteinander kooperieren oder kommunizieren.
3. Die Kooperation erfolgt mithilfe von Schichtprotokollen.
4. Instanzen greifen auf Dienste der *direkt* darunterliegenden Schicht zurück.

Über die Schnittstelle zwischen n -ter und $(n - 1)$ -ter Schicht werden folgende Informationen ausgetauscht:

- Benutzerdaten der $(n + 1)$ -ten Schicht
- Kontrollinformationen zwischen den n Instanzen
- Steuer- und Kontrollinformationen für die $(n - 1)$ -te Schicht zur Erbringung bestimmter Dienste.

4.2 CSP – Communicating sequential processes

Ist eine Beschreibungsmöglichkeit, basierend auf einer mathematischen Theorie (Ansammlung von logischen Aussagen und Gesetzen), zur Spezifikation, zum Entwurf und zur Implementierung von Kommunikationsprotokollen und interagierenden Systemen.

Die Grundidee dabei ist eine Zerlegung des System in parallel arbeitende Teilsysteme, die untereinander und mit der Umgebung kommunizieren.

1. Prozeß A
2. Prozeß B
3. Kanal $A \rightarrow B$
4. Kanal $B \rightarrow A$

Durch die Anwendung der mathematischen Gesetze läßt sich die Korrektheit des Systems nachweisen oder die Fehler erkennen.

Grundbegriffe:

Objekt ist ein Gegenstand, der agiert oder mit seiner Umgebung interagiert. Dies kann z. B. ein Rechner oder ein Programm sein.

Prozeß ist ein Modell zur Verhaltensbeschreibung eines Objekts. In einem Prozess läuft eine Folge von Ereignissen ab.

Ereignis ist die zeitlose Veränderung des Verhaltens eines oder mehrerer Prozesse.

Alphabet ist die Menge von Ereignissen, die einen Prozess beschreiben. Beschreibung:
 αP

Präfixnotation Sei P ein Prozess und αP ein Alphabet. Dann beschreibt $R = (x \rightarrow P)$ mit $x \in \alpha P$ ein Objekt, das am Ereignis x teilnimmt und sich anschließend wie der Prozess P verhält.

Ereignis STOP/HALT Dieses Ereignis ist in jedem Alphabet enthalten. Zur Unterscheidung schreibt man $STOP_{\alpha P}$ oder $HALT_{\alpha P}$

Operator \rightarrow ist rechtsassoziativ

$$(x \rightarrow y \rightarrow P) = (x \rightarrow (y \rightarrow P)) \quad (4.1)$$

Beispiel 4.2.1

$$Uhr := (tick \rightarrow tack \rightarrow STOP_{\alpha P}) \quad (4.2)$$

Problem: Die Uhr ist nach dem *tack* kaputt. Hier fehlt eine Rekursion.

Regeln:

Auswahlregel Ein Prozess nimmt entweder am Ereignis x oder am Ereignis y teil.

$$R = (x \rightarrow P | y \rightarrow Q) \tag{4.3}$$

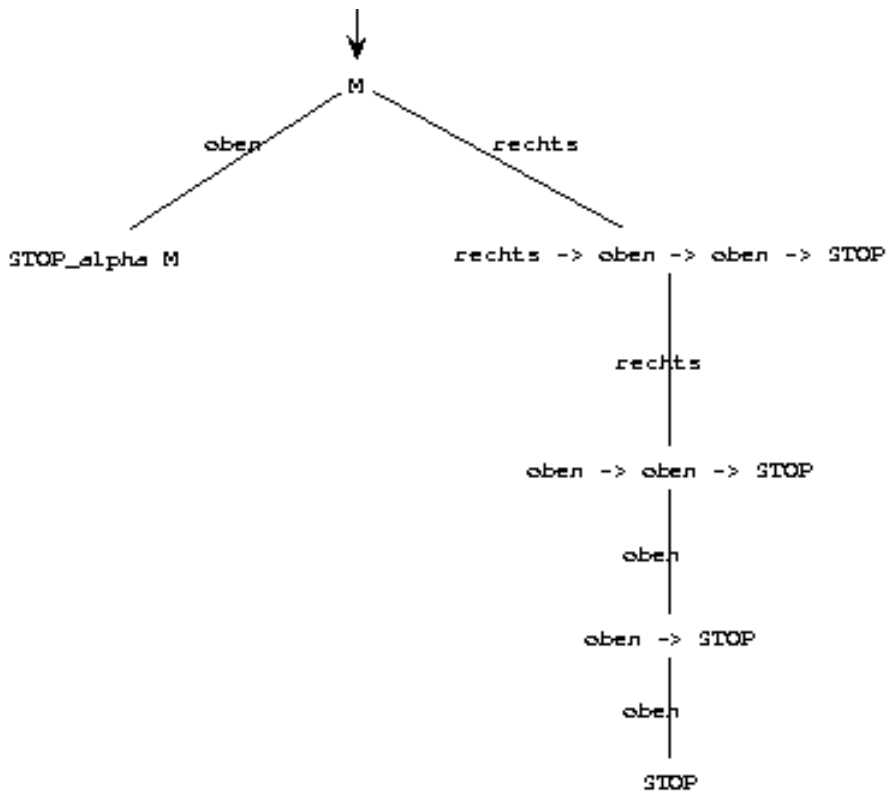
beschreibt eine Objekt R , das entweder am Ereignis x teilnimmt und sich anschließend wie der Prozess P verhält oder am Ereignis y teilnimmt und sich anschließend wie der Prozess Q verhält.

Beispiel 4.2.2 (Maus im Labyrinth)

$$\alpha M = \{oben, unten\} \tag{4.4}$$

$$M = \{oben \rightarrow STOP_{\alpha M} | rechts \rightarrow rechts \rightarrow oben \rightarrow oben \rightarrow STOP_{\alpha M}\} \tag{4.5}$$

Prozessbaum:



Rekursionsregel Bei einer rekursionsgleichung kommt die Variable auf beiden Seiten der Gleichung vor

$$P = (x \rightarrow P) \tag{4.6}$$

Beispiel 4.2.3

$$Uhr = (tick \rightarrow tack \rightarrow Uhr) \quad (4.7)$$

Beispiel 4.2.4 (Bitspeicher)

Formulierung eines Bitpuffers (Flipflops).

$$\alpha BP = (in0, in1, out0, out1) \quad (4.8)$$

$$BP = (in0 \rightarrow out0 \rightarrow BP | in1 \rightarrow out1 \rightarrow BP) \quad (4.9)$$

Gegenseitige Rekursion Verallgemeinerung der Rekursionsregel auf ein System von Prozeßgleichungen. Jeder Prozeß tritt genau einmal auf der linken Seite der Gleichung auf, alle Prozesse haben das selbe Alphabet

Beispiel 4.2.5

$$P = (a \rightarrow P | b \rightarrow Q) \quad (4.10)$$

$$Q = (b \rightarrow c \rightarrow Q | b \rightarrow d \rightarrow R | d \rightarrow P) \quad (4.11)$$

$$R = (a \rightarrow b \rightarrow Q) \quad (4.12)$$

$$\alpha P = \alpha Q = \alpha R = \{a, b, c, d\} \quad (4.13)$$

parallele Kompositionen Bisher haben wir nur einzelne Prozesse betrachtet. Jetzt wollen wir Prozesse betrachten, die nebenläufig ablaufen und sich gegenseitig beeinflussen.

Unter der Komposition zweier Prozesse P und Q verstehen wir einen Prozess $P \parallel Q$, der sich so verhält wie ein System zusammengesetzt aus den beiden Prozessen P und Q .

Ein wichtiger Begriff ist das synchronisierende Ereignis x mit $x \in \alpha P \wedge x \in \alpha Q$

Es gibt zwei Möglichkeiten der Komposition:

Interaktion $\alpha P = \alpha Q \longrightarrow$ alle Ereignisse sind synchronisiert

Beispiel 4.2.6

$$\alpha P = \alpha Q = \{a, b, c\} \quad (4.14)$$

$$P = (a \rightarrow b \rightarrow P | b \rightarrow P) \quad (4.15)$$

$$Q = (a \rightarrow (b \rightarrow Q | c \rightarrow Q)) \quad (4.16)$$

-Die beiden Bäume-

baum für $P \parallel Q$

$$P \parallel Q = (a \rightarrow b \rightarrow P \parallel Q) \quad (4.17)$$

Beispiel 4.2.7

$$\alpha P = \alpha Q = \{a, b\} \quad (4.18)$$

$$P = (a \rightarrow b \rightarrow P) \quad (4.19)$$

$$Q = (b \rightarrow a \rightarrow Q) \quad (4.20)$$

$$P \parallel Q = STOP \quad (4.21)$$

Regeln

1. $P \parallel Q = Q \parallel P$

2. $P \parallel (Q \parallel R) = (P \parallel Q) \parallel R$

3. $P \parallel STOP = STOP$

4. $(c \rightarrow P) \parallel (c \rightarrow Q) = c \rightarrow P \parallel Q$

5. $(c \rightarrow P) \parallel (d \rightarrow Q) = STOP \quad (c \neq d)$

Concurency

Regeln: $a \in (\alpha P \setminus \alpha Q), b \in (\alpha Q \setminus \alpha P), c \in (\alpha P \cap \alpha Q)$

7. $(a \rightarrow P) \parallel (c \rightarrow Q) = a \rightarrow (P \parallel (c \rightarrow Q))$

8. $(c \rightarrow P) \parallel (b \rightarrow Q) = b \rightarrow ((c \rightarrow P) \parallel Q)$

9. $(a \rightarrow P) \parallel (b \rightarrow Q) = (a \rightarrow P \parallel (b \rightarrow Q)) \parallel (b \rightarrow (a \rightarrow P) \parallel Q)$

Beispiel 4.2.8

$$\alpha P = \{a, c\} \quad \alpha Q = \{b, c\}$$

$$P = (a \rightarrow c \rightarrow P)$$

$$Q = (c \rightarrow b \rightarrow Q)$$

$$\begin{aligned} P \parallel Q &= (a \rightarrow c \rightarrow P) \parallel (c \rightarrow b \rightarrow Q) \\ &= a \rightarrow ((c \rightarrow P) \parallel (c \rightarrow b \rightarrow Q)) \\ &= a \rightarrow c \rightarrow (P \parallel (b \rightarrow Q)) \end{aligned}$$

$$\begin{aligned} P \parallel (b \rightarrow Q) \parallel &= (a \rightarrow c \rightarrow P) \parallel (b \rightarrow Q) \\ &= a \rightarrow ((c \rightarrow P) \parallel (b \rightarrow Q))|b \rightarrow (a \rightarrow c \rightarrow P) \parallel Q \\ &= a \rightarrow b \rightarrow ((c \rightarrow P) \parallel Q)|b \rightarrow ((a \rightarrow c \rightarrow P) \parallel (c \rightarrow b \rightarrow Q)) \\ &= a \rightarrow b \rightarrow ((c \rightarrow P) \parallel (c \rightarrow b \rightarrow Q)) \parallel \\ &\quad b \rightarrow ((a \rightarrow c \rightarrow P) \parallel (c \rightarrow b \rightarrow Q)) \\ &= a \rightarrow b \rightarrow c \rightarrow (P \parallel (b \rightarrow Q))|b \rightarrow \\ &\quad a \rightarrow ((c \rightarrow P) \parallel (c \rightarrow b \rightarrow Q)) \\ &= a \rightarrow b \rightarrow c \rightarrow (P \parallel (b \rightarrow Q))|b \rightarrow a \rightarrow c \rightarrow (P \parallel (b \rightarrow Q)) \end{aligned}$$

Beispiel 4.2.9

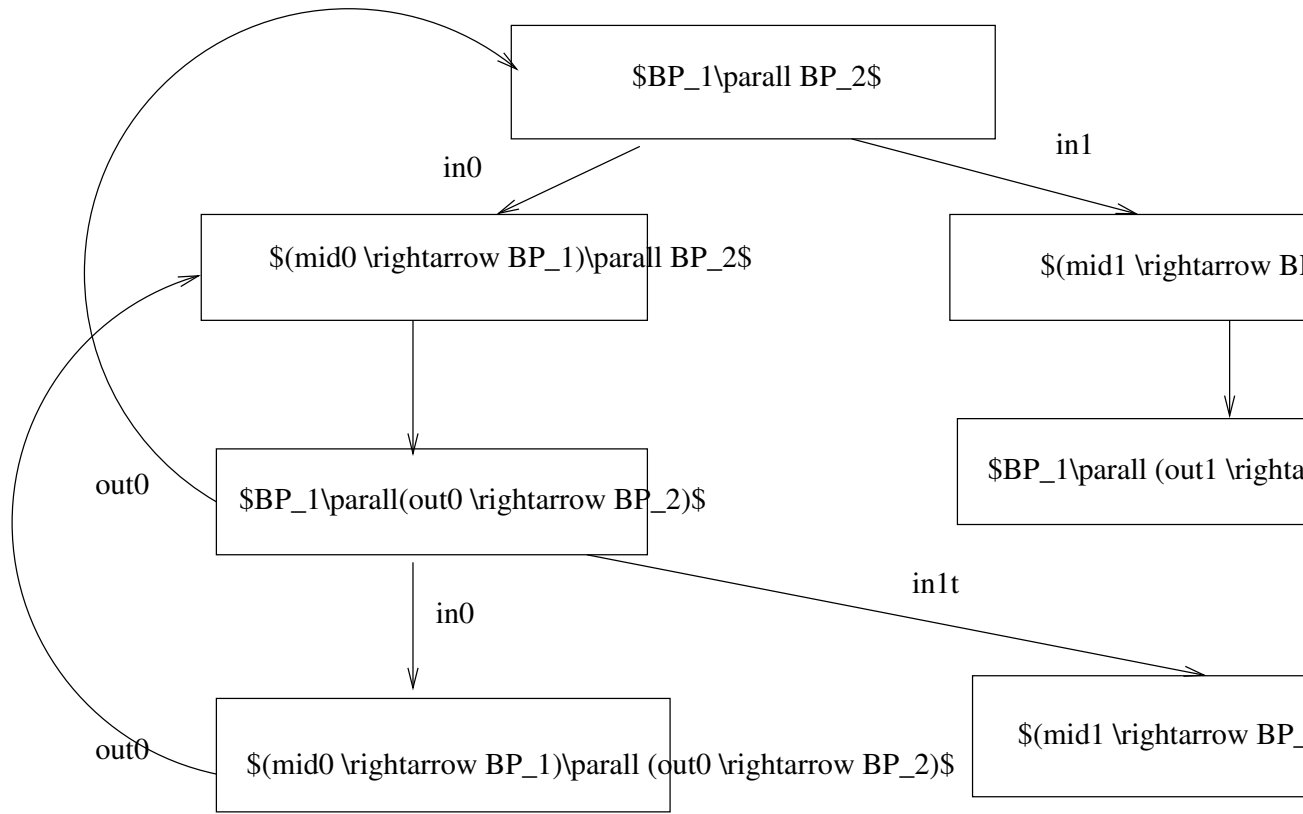
$$\alpha BP_1 = \{in0, in1, mid0, mid1\}$$

$$\alpha BP_2 = \{mid0, mid1, out0, out1\}$$

$$BP_1 = (in0 \rightarrow mid0 \rightarrow BP_1|in1 \rightarrow mid1 \rightarrow BP_1)$$

$$BP_2 = (mid0 \rightarrow out0 \rightarrow BP_2|mid1 \rightarrow out1 \rightarrow BP_2)$$

$$BP_1 \parallel BP_2 = (\dots) \parallel (\dots)$$



5 Grundlagen der Kommunikationsprotokolle

5.1 Entwicklungspasen für Kommunikationsprotokolle

Ausgehend von einem *Problem* wird eine *informelle Anforderungsdefinition* erstellt. Am Beispiel von USB wäre dies z. B., dass das Protokoll eine serielle Übertragung nutzt, mit *Quitierung* arbeitet oder die angestrebte Datenrate. Danach erstellt man einen *Diensteentwurf* (Auswahl aus Normen) und erstellt eine *Dienstspezifikation* (Pflichtenheft).

Danach folgt der *Protokollentwurf* der dann die *Protokollspezifikation*. An dieser Stelle sollte unbedingt eine *Verifikation* stehen, ob das Protokoll das leistet, was die Dienstspezifikation fordert. Gegebenfalls sollte die Dienstspezifikation noch einmal überarbeitet werden.

Der *Programmmentwurf* liefert die *Programmspezifikation*. Diese sollte einer Prüfung gegen die Protokollspezifikation (*Kommunikationsprüfung*) unterzogen werden. Danach beginnt man mit der *Implementierung*.

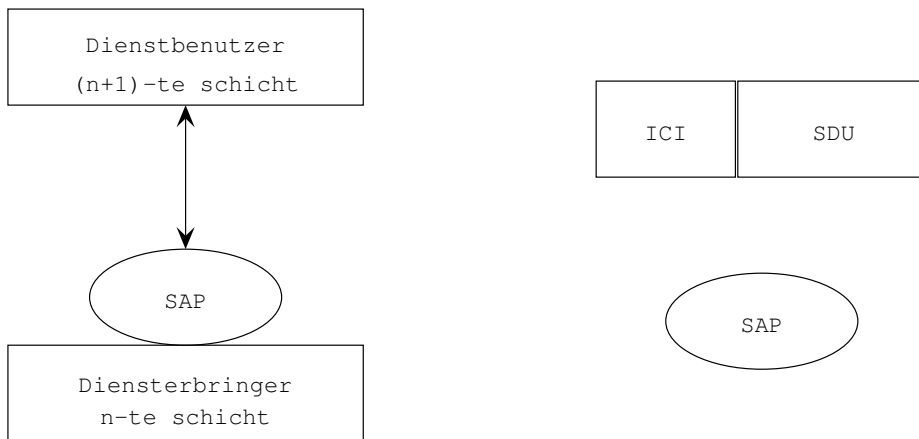
Die Prüfung der einzelnen Stufen wird oft aus Kostengründen unterlassen, was im schlimmsten Falle dazu führt, dass an der Implementation Fehler findet, die einem von ganz vorn beginnen lassen.

5.2 A closer look to ISO/OSI

Jede Schicht im ISO/OSI-Modell realisiert einen oder mehrere Dienste durch

- eigene Leistung innerhalb der Schicht und/oder
- Inanspruchnahme von Diensten niedrigerer Schichten

und stellt diese Dienste der nächst höheren Schicht am sogenannten *Dienstzugangspunkt* (engl. Service Access Point, SAP) zur Verfügung.



ICI: Interface control information

SDU: Service data unit

Die Dienste werden mit symbolischen Namen bezeichnet, z. B.

T-connect: Aufbau einer Verbindung innerhalb der Transportschicht

P-data: Datentransport in der Darstellungsschicht (presentation layer); Formatumwandlung, Datenanpassung

Ein Dienstelement ist ein logisch unteilbares Ereignis. Eine Folge von Dienstelementen realisiert den Dienst. Es gibt vier Typen von Dienstelementen:

- Anforderung (request)
- Ankündigung (indication)
- Antwort (response)
- Bestätigung (confirmation)

Beispielaufruf eines Dienstelementes: `T-connect.request(called address, calling address, user data)`

Für den Austausch von Dienstelementen werden Regeln benötigt, die sogenannten Dienstprotokolle. Es gibt zwei Typen von Dienstprotokollen

- bestätigter Dienst
- unbestätigte Dienste (datagramm)

5.2.1 Bestätigter Dienst

todo: Hier fehlt eine Grafik

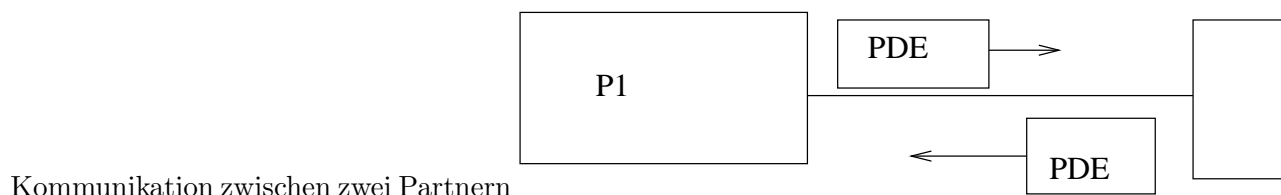
Unbestätigter Dienst: request, indication

Weitere Begriffe Jede Schicht realisiert ihre Teilaufgaben durch HW/SW-Modelle, d. h. Schicht- n -Protokollinstanzen genannt werden (n -PI, n -entity) Instanzen der selben Schicht, die mit anderen kommunizieren, heißen Partnerinstanzen oder peer entity

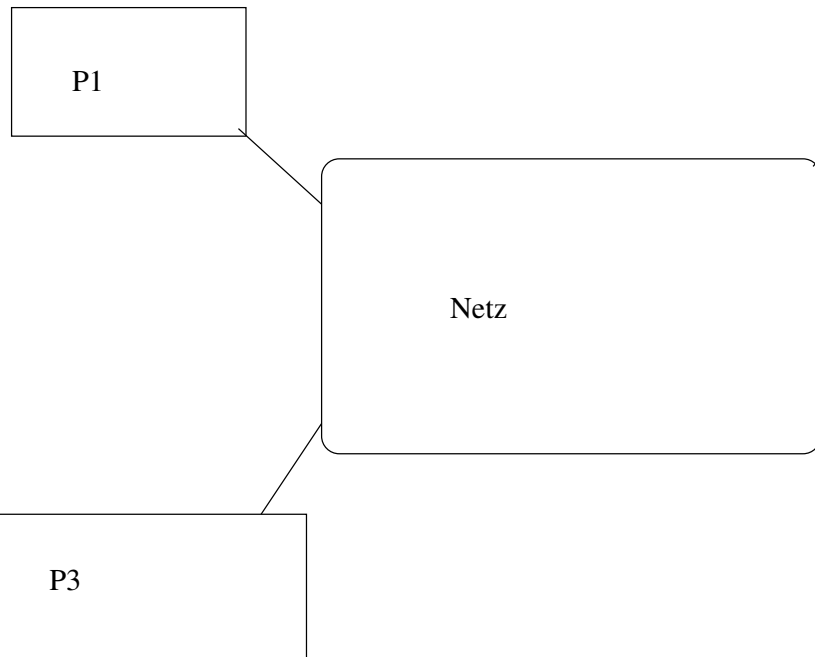
Die Kooperation von n -PI's würde durch ein Schicht- n -Protokoll geregelt. Der Austausch von Informationen erfolgt über Schicht- n -Protokolldaten (n -PDE, n -PDU)

Jedes n -PDE besteht aus Nutzinformationen und Protokollkontrollinformationen (n -PKI).

5.2.2 Welche Probleme treten bei der Kommunikation unabhängig von irgendwelchen Protokollen immer wieder auf?



- Festlegung der Datenstruktur
- Codetransparenz
- Segmentierung und Füllen
- Identifikation mit Sequenznummern
- Fehlererkennung und -korrektur
- Flußkontrolle



Kommunikation mit mehreren Partnern im Netz

- Namensgebung und Adressierung
- Zugriffskontrolle
- Verkehrslenkung (routing)

1. Datenstrukturen

PDE enthalten • Daten

- Befehle (bzgl. Rechte, Flußkontrolle)
- Attribute (z. B. Sequenznummern)

Probleme dabei sind:

- wo beginnt oder endet die Nachricht
- von welchem Typ ist die Nachricht (implizite Angabe oder explizite Angabe mit Header)

Beispiel für Datenstrukturen anhand ISO/OSI

Schicht 1 (physical layer)

Zeitmultiplexen

Schicht 2 Zeichensynchrone Übertragung (von IBM eingeführt – SDLL Synchronus Data Link Control)

Bitsynchrone Übertragung (HDLC High Level Data Link control)

P	A	K	Nutzin
---	---	---	--------

F: Flag 011 11 110 A: Adresse K: Kontrollinformation PS: Prüfsequenz

2. Codetransparenz

Drei Möglichkeiten:

- Verwendung eines Codes, in dem bestimmte Bitfolgen für Steuerzeichen reserviert sind (z. B. ASCII, EBCDIC, ...)
- Stopf-Bit (staffing bit) bei bitsynchroner Übertragung. Ein sogenannter Stopf-bitgenerator fügt nach jedem Auftreten von fünf Einsen in Folge eine Null als Stopfbit ein. Damit ist die Folge sechs Einsen unmöglich. Der Empfänger entfernt nach fünf Einsen immer die folgende Null.

Beispiel: 011 11 110 → 001 111 110 → ... → 011 11 110

- Ausweichzeichen (DLI – data link escape) bei bytesynchroner Übertragung

Möglichkeit 1 Jedes Steuerzeichen in der Nutzinformation wird mit einem Ausweichzeichen DLE gekennzeichnet.

Beispiel: Nutzinformation: $\text{;D}_i\text{;ETX}_i\text{;D}_i\text{;DLE}_i\text{;STX}_i$ Übertragung: $\text{;D}_i\text{;DLE}_i\text{;ETX}_i\text{;D}_i\text{;DLE}_i\text{;STX}_i$

Möglichkeit 2 Kennzeichnung der Steuerzeichen in der Kontrollinformation mit ;DLE_i . Tritt ;DLE_i in der Nutzinformation auf, dann wird es verdoppelt.

3. Segmentieren und Füllen

Festlegung einer oberen Grenze für die Information. Dabei sollten gewisse Punkte berücksichtigt werden:

- Übertragungsfehler (Nachricht lang: hohe Fehlerwahrscheinlichkeit, Nachricht kurz: hoher Overhead)
- Flußkontrolle (eventuell großen Pufferbereich, ab Flaschenhälsen)
- Behandlung in Vermittlungsknoten (Annahme, Zwischenspeicherung, Bearbeitung, Ausgabe von Nachrichten)

Die zu übertragende Nachricht stimmt nicht mit der Größe des Nutzdatenblocks überein.

- Nachricht zu lang → segmentieren
Die Nutznachricht wird über mehrere Pakete verteilt.
Beispiel: Schicht 3
- Nachricht ist zu kurz → Füllen
- kombinieren: Segmentieren und Füllen
zu kleine Pakete werden mit Daten von Folgepaketen gefüllt

5.2.3 Fehlererkennung und -korrektur

- Störung der Kommunikation durch
 - Übertragungsfehler
 - Fehler bei der Zwischenspeicherung
 - Fehler des Interfaces
 - Protokollfehler
- Maßnahmen auf der Gegenseite:
 - Segmentieren
 - Rechenelemente Kontrollinformation:
 - * Paritätsbits
 - * zyklische Blocksicherung (CRC – cyclic redundancy check)
 - * Sequenznummern
- Maßnahmen auf der Empfängerseite:
 - Überprüfung der Sicherungsbits
 - Formatprüfung bei PDE (Protokolldateneinheit)
- Korrektur des Fehlers:
 - Vorwärtskorrektur (Hemming-Code, CRC)
 - Aktive Fehlerkontrolle (Quittierungen) (praxistypisch)
 - passive Fehlerkontrolle (Zeitüberwachung)

Beispiel für (Vorwärts-)Korrektur mit CRC

Sender Nachricht (Daten + Steuerinformationen) wird durch ein Prüfpolynom dividiert, der Rest der Division wird übertragen = 2 Bits

Empfänger Nachricht (Daten + Steuerinformationen + CRC) wird ebenfalls durch das Prüfpolynom dividiert

$$\text{Ergebnis} \begin{cases} = 0 & : \text{korrekte Übertragung} \\ \neq 0 & : \text{fehlerhafte Übertragung} \end{cases}$$

HDLC (Ebene 2)

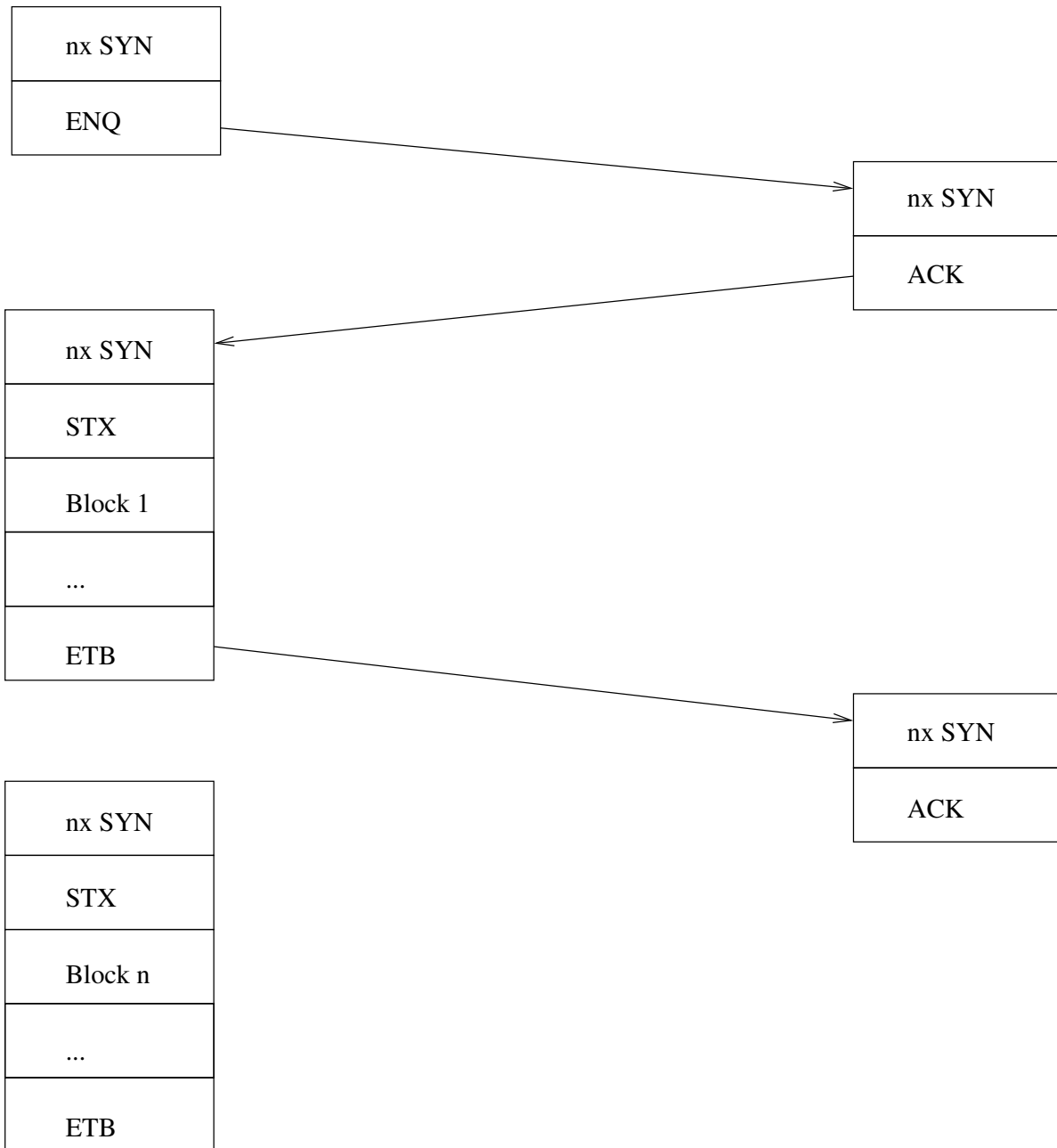
Prüfpolynom $x^{16} + x^{12} + x^5 + 1$

Aktive Fehlerkontrolle

- keine Übertragungsfehler: Quittieren mit ACK
- Übertragungsfehler: Quittieren mit NAK oder REJ
- Beispiel: MSV 2 (medium speed version 2)

Sender

Empfänger



....

Was passiert im Fehlerfall

Vorteil einfaches Protokoll, kleiner Pufferbereich

Nachteil Große Lücken in der Kommunikation

→ Verbesserung: Sammelquittung (Rollback #1, #2, #3)

Problem:

Um das Problem allgemein zu erkennen, braucht man einen systematischen Ansatz: CSP

Was passiert im Fehlerfall?

Vorteil: einfaches Protokoll, kleiner Pufferbereich. Nachteil: große Lücken in der Kommunikation

Eine Verbesserung: Sammelquittung (Pullblock #1, #2, #3)

Ein Problem ist, wenn eine Quittung gestört ist. Dann sendet der Sender wieder eine Fehlermeldung.

um das Problem allgemein zu erkennen, braucht man einen systematischen Ansatz: CSP

Prozessbeschreibung: Sendeprozess

$$SP = (NachrichtSP \rightarrow SPwait)$$

$$SPwait = (ACKKP2 \rightarrow SP/NAKKP2 \rightarrow SP | \\ NoiseKP2 \rightarrow NAKSP \rightarrow SPnoise)$$

$$SPnoise = (ACKKP2 \rightarrow SP | NAKKP2 \rightarrow NAKSP \rightarrow SPnoise | NoiseKP2 \rightarrow NAKSP \rightarrow SPnoise)$$

$$EP = (NachrichtKP1 \rightarrow ACKEP \rightarrow EPwait | NoiseKP1 \rightarrow NACKEP \rightarrow EPnoise)$$

$$EPwait = (NachrichtKP1 \rightarrow ACKEP \rightarrow EPwait | NAKKP1 \rightarrow ACKEP \rightarrow EPwait | NoiseEP)$$

Passive Fehlerkontrolle

kein Übertragungsfehler: ACK Übertragungsfehler/Empfänger reagiert nicht: Timerablauf beim Sender

Der Sender startet nach dem Versenden der Nachricht einen Timer τ . Wird die Nachricht korrekt übertragen, bestätigt dies der Empfänger mit einem ACK. Trifft beim Sender vor Timerablauf nicht ein ACK ein, sendet dieser die Nachricht noch einmal.

Problem: Bei Schwankungen der Übertragungszeit oder bei Verzögerungen beim Empfänger muß τ auf ein Maximum eingestellt werden. Kommt das ACK nicht rechtzeitig vor Timerablauf, wird die Information nochmals gesendet und somit verdoppelt, obwohl die Übertragung in Ordnung war.

5.2.4 Identifikation mit Sequenznummern

Die Sequenznummern müssen nur in einem kleinem Zeitbereich eindeutig sein. In der Praxis hat sich gezeigt, dass für Sequenznummern 3 Bit ausreichend sind.

Realisierung auf Ebene 2 in HDLC

5.2.5 Flußkontrolle

Definition: Anpassung von Sende- und Empfangsleistung

Methoden:

Abweismechanismus (discarding) Der Empfänger verwirft einfach die Pakete, der er (aus zeitlichen Problemen oder Speicherproblemen) nicht annehmen kann.

Unterbrechungsmethode (stop and go) Der Empfänger steuert die Informationsverarbeitung. Er schickt dem Sender *stop* oder *go* Befehle.

Wenn er z.B. feststellt, dass der Pufferbereich überlaufen wird, sendet er dem Sender ein *stop*.

Problem: schwankende Übertragungszeiten

- worst case Abschätzung \implies schlechte Kommunikationsauslastung
- großer Pufferbereich beim Empfänger

stop and wait Der Sender wartet auf die Bestätigung seiner Nachricht, bevor er die nächste Nachricht versendet.

Reservierungsmethode Beim Verbindungsaufbau muss der Sender dem Empfänger seinen Wunsch für die Puffergröße mitteilen. Der Empfänger teilt ihm mit, wieviel er ihm zugesteht.

Der Sender kann dann so viele Nachrichten auf den Weg schicken, wie auf der Empfängerseite Puffer zur Verfügung steht.

unflexibel auf Kommunikationsänderungen reagieren die Puffergröße ist für die Verbindung fest.

Beispiel: ARPA-Net

1. Der Sender S_1 schickt dem Empfänger S_2 in der Initialisierungsphase (ICP) einen STX (send transmission request) mit der gewünschten Puffergröße. S_2 bestätigt mit RTS (request transmission start) und sendet parallel die bestätigte Puffergröße (\leq gewünschte Größe) an S_1 .

Danach werden die Daten übertragen (?CP).

Am Ende wird die Verbindung abgebaut (CCP)

Credit-Verfahren Der Empfänger räumt dem Sender Kredit (Menge an Puffer) ein. Sobald der Kredit verbraucht ist, muss das Senden eingestellt werden. Während der Kommunikation kann der Kredit durch positive Quittungen wieder aufgefüllt oder vergrößert bzw. verkleinert werden.

Problem: Treten beim Empfänger Änderungen der Bandbreite auf, so sendet er eine Kreditänderung (Erhöhung oder Verringerung). Wird die Bestätigung des neuen Kredits gestört, sendet der Empfänger den Kredit neu. Kreditverdoppelung
Lösung: Sequenznummern für die Kredite

Fensterverfahren (window method)

Vorteil: Reihenfolge garantiert, Flußkontrolle, geschützte Übermittlung

Nachteil: Aufwand

Beispiele: ARPA-Netz, X.25 (virtual call)

Aufbau einer virtuellen Verbindung mit Hilfe des Konzepts „logische Kanäle“

- Ebene 2 stellt zwischen der Datenendeinrichtung und dem Vermittlungsrechner im Netz einen gerichteten Datenübertragungskanal bereit.

Beispielsweise durch Zeitmultiplexen können mehrere Anwendungsprozesse über diesen Kanal kommunizieren.

- Man unterscheidet bis zu 2^{12} logische Kanäle.
- Die logischen Kanäle werden den Anwendungsprozessen permanent oder temporär zugeordnet.

Aufbau einer virtuellen Verbindung:

1. Sender an Vermittlungsrechner ein `call request`-Paket mit Zieladresse und Kanalnummer
2. Vermittlungsrechner schickt das Paket durchs Netz
3. Auswahl von logischem Kanal beim Empfänger beim letzten Vermittlungsrechner
4. Der Empfänger schickt ein `call accepted`-Paket zurück zum Sender.
⇒ virtuelle Verbindung ist aufgebaut
5. Datenübertragung beginnt

Abbau der virtuellen Verbindung mit `clear request`-Paket und `clear confirm`-Paket

Leitungsvermittlung: Netz aus Teilstrecken, die zu durchgehenden Verbindungen zusammengeschaltet werden.

6 Verkehrslenkung (Routing)

Normalerweise ist die Adresse des Empfängers bekannt und ortsgebunden.

Aufgabe der Verkehrslenkung ist, es nach vorgegebenen Kriterien den optimalen Weg zwischen Sender und Empfänger auszuwählen.

Dabei gibt es verschiedene Kriterien:

- Minimierung der mittleren Verzögerungszeit; schnellster Weg
- Maximierung des Gesamtdurchsatzes
- Minimierung der Teilstrecken im Netz

Für unterschiedliche Routingalgorithmen sollen möglichst alle folgenden Eigenschaften gelten:

- funktionell Korrekt
- einfach
- robust (hinsichtlich Hardware-/Software-Fehler, Topologie)
- fair
- optimal

Die Eigenschaften können sich widersprechen.

Wir unterscheiden zwei Klassen von Verfahren: adaptive und nicht adaptive Verfahren.

6.1 Nichtadaptive Verfahren

Fluten (flooding) Die Nachricht wird an jedem Vermittlungsrechner an seine Nachbarn (mit Ausnahme des Rechners, von dem die Nachricht stammt) weitergegeben und der Hopzähler dekrementiert. Wenn der Hopzähler 0 ist, wird das Paket vernichtet. Zu Beginn wird der Hopzähler auf den maximalen Durchmesser des Netzes gesetzt.

Nachteil: Exponentielle Vermehrung der Pakete, Vorteil: sehr robust

Feste Leitungslenkung $A \rightarrow B = 8$ $A \rightarrow D = 4$ $A \rightarrow H = 3$ $B \rightarrow C = 3$ $B \rightarrow D = 5$ $B \rightarrow E = 2$ $C \rightarrow E = 5$ $c \rightarrow F = 2$ $d \rightarrow g = 2$ $e \rightarrow f = 2$ $e \rightarrow g = 2$ $e \rightarrow h = 2$ $f \rightarrow i = 2$ $g \rightarrow h = 1$ $h \rightarrow i = 4$

Strecken gewichtet mit der Verzögerungszeit

A	Nachbarknoten	Nachbarknoten 2	Nachbarknoten 3
B	H	B	D
C	H	B	D
D	D	H	B
E	H	D	B
F	H	D	B
G	H	D	B
H	H	D	B
I	H	D	B

Nachteil: zu starr

Alternative Leitweglenkung Zum besten Weg wird in der Tabelle eine Spalte für den zweitbesten, drittbesten, ... Weg angelegt. Zeitweilige Störungen können umgangen werden

Alle Methoden haben aber das Problem, dass sie nicht die Dynamik des Netzes berücksichtigen können.

6.2 Adaptive Verfahren

Periodische Anpassung der Leitwegtabellen unter Berücksichtigung des Verkehrs und der Topologien.

Frage: Zentrale oder dezentrale Aufbereitung der Tabellen?

zentral Vorteil: Konsistenz – alle Rechner arbeiten mit denselben Tabellen, Nachteil: hoher Aufwand die Tabellen zu verteilen, Ausfall des Netzmanagers = Ausfall des Netzes

dezentral Periodischer Austausch von Lastinformationen mit benachbarten Rechnern. Nachteil/Vorteil genau umgekehrt zu zentral

Vorteil: nur noch lokale Last durch wenige Tabellen

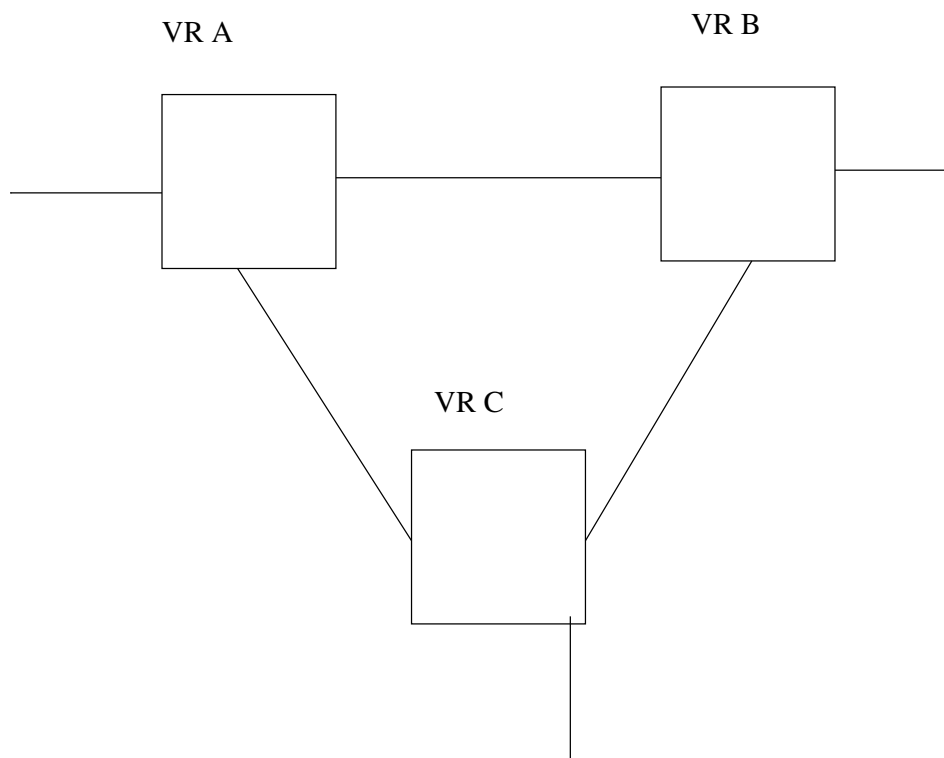
(Nachteil:) Lastsituationen aus der Ferne wird nicht berücksichtigt; jedoch nicht so problematisch, da dieser Verbindungen uninteressanter

6.2.1 Lastbegrenzung

Diagramm: Gesamtverzögerungszeit(Verkehrsdichte) e^{x-10} mit Senkrechter bei $x = 10$
links davon akzeptabel, rechts nicht mehr akzeptabel

Probleme: „Store and forward lockup“ und „Reassembly lockup“

Store and forward lockup



inXY: Daten, die an Rechner X eingegeben werden und an Y gehen sollen. Diese werden im Transferspeicher abgelegt

transXY: Daten, die von Rechner X kommen und zu Rechner Y. $Y \neq$ eigener Rechnernache \rightarrow Paket in Transferspeicher (beschränkte Größe), andernfalls in den lokalen Speicher (unendlich groß)

Spezifikation in CSP:

$$\begin{aligned}
PvA &= (inAC \rightarrow PvAAC | inAB \rightarrow PvAAB | transCA \rightarrow PvA | transCB \rightarrow PvAAB) \\
PvAAC &= (transAC \rightarrow PvA | transCA \rightarrow PvAAC) \\
PvAAB &= (transAB \rightarrow PvA | transCA \rightarrow PvAAC) \\
\\
PvB &= (inBC \rightarrow PvBBA | inBC \rightarrow PvBBC | transAB \rightarrow PvB | transAC \rightarrow PvBBC) \\
PvBBA &= (transBA \rightarrow PvB | transAB \rightarrow PvBBA) \\
PvBBC &= (transBC \rightarrow PvB | transAB \rightarrow PvBBA) \\
\\
PvC &= (inCB \rightarrow PvCCB | inCA \rightarrow PvCCA | transBC \rightarrow PvC | transBA \rightarrow PvCCA) \\
PvCCB &= (transCB \rightarrow PvC | transBC \rightarrow PvCCB) \\
PvCCA &= (transCA \rightarrow PvC | transBC \rightarrow PvCCB)
\end{aligned}$$

um jetzt einen Deadlock zu finden, schalten wir die drei Prozesse parallel

$$\begin{aligned}
PvA || PvB || PvC &= (&& inAC \rightarrow PvAAC || PvB || PvC | \\
&&& inAB \rightarrow PvAAB || PvB || PvC | \\
&&& inBA \rightarrow PvA || PvBBA || PvC | \\
&&& inBC \rightarrow PvA || PvBBC || PvC | \\
&&& inCB \rightarrow PvA || PvB || PvCCB | \\
&&& inCA \rightarrow PvA || PvB || PvCCA) \\
&= inAC \rightarrow inBA || PvBBA || \dots \\
&= inAC \rightarrow inBA \rightarrow inCB \dots
\end{aligned}$$

Damit hat man einen Lockup erreicht

Reassembly lockup

Durch ungünstiges Eintreffen der Pakete, können alle Speicherrahmen reserviert, jedoch noch nicht voll, sind. Wenn die daran angeschlossenen Vermittlungsrechner wiederum mit Paketen für andere Verbindungen zu diesem Rechner gefüllt sind, können die restlichen Pakete für die geöffneten Rahmen nicht durchkommen, der Vermittlungsrechner kann aber auch keine neuen Pakete annehmen, da bei ihm alle Rahmen für das reassembling belegt sind.

6.2.2 Lastbegrenzungsmethoden

Linkmethode: Für jeder Nachricht muss eine fiktive Leitung (Link) bis zum Ziel belegt werden. Die Anzahl der Links zwischen Sender und Empfänger ist begrenzt.

Für diese maximale Anzahl von Links muss Reassamblingspuffer zur Verfügung gestellt werden. Pro Link darf nur eine Nachricht verschickt werden. (ARPA-net, altes Verfahren)

Reservierungsmethode: Vor der Übertragung wird im Zielknoten Speicher reserviert für die komplette Nachricht \Rightarrow Anzahl der Nachrichten im Netz begrenzt (ARPA-net, neueres Verfahren)

Leerpaket-Methode: Im Netz wird zu jedem Zeitpunkt die selbe Anzahl von Nachrichten bewegt.

Aufforderungsmethode: Nachrichten können nur als Antwort auf eine Anfrage verschickt werden.

6.3 Netzübergang

Sollen zwei Netzwerke verbunden werden ist dafür eine Hardware notwendig – ein Gateway. Ein Gateway ist ein Protokollumsetzer auf der Ebenen 4–7 des ISO/OSI-Protokolls. Auf der Ebene 1 bezeichnet man diesen Netzübergang als Repeater (einen Verstärker). Auf der Ebene 2 nennt man es eine Bridge (Beispiel: Ethernet-token-ring). Auf Ebene 3 ist es ein Router – ein Router arbeitet selektiv, der nur Pakete weitergibt, wenn ein Übergang notwendig ist.

7 Leistungsbewertung

Methoden zur Leistungsbewertung:

- graphentheoretische Verfahren
- Verkehrstheoretische Verfahren (Warteschlangenmethode)

7.1 Graphentheoretische Verfahren

Ziel der graphentheoretischen Verfahren ist die Bestimmung des maximalen Flusses zwischen Sender und Empfänger.

7.2 Verkehrstheoretische Verfahren

Warteschlangenmodell:

Warteschlange mit n ($1 \leq n \leq \infty$) Plätzen und m ($1 \leq m < \infty$) Bedienstationen.

Ankunftsrate $\lambda = \frac{1}{a_m}$ (a_m : mittlere Ankunftsrate) Bedienrate $\mu = \frac{1}{b_m}$ (b_m : mittlere Bedienrate) Verkehrsgüte $\sigma = \frac{\lambda}{\mu} \leq 1$ Für > 1 würde die Warteschlange volllaufen und wir kämen in eine Deadlock-Situation.

$P_i = \sigma^i P_0$ Wahrscheinlichkeit, dass i Aufträge in dem System sind.

7.2.1 Wie groß ist P_0 ?

$$1 = \sum_{i=0}^{\infty} P_i = \sum_{i=0}^{\infty} \sigma^i P_0 = P_0 \sum_{i=0}^{\infty} \sigma^i = \frac{P_0}{1 - \sigma}$$

$\Rightarrow P_0 = 1 - \sigma \Rightarrow \sigma$ sollte sehr nahe bei 1 liegen.

7.2.2 Wieviel Pufferplätze sollen zur Verfügung gestellt werden?

P_k ($> k$ Warteplätze sind erforderlich)

$$P_r = \sum_{i=k+1}^{\infty} P_i = \sum_{i=k+1}^{\infty} \sigma^i P_0 = \sum_{i=k+1}^{\infty} \sigma^i (1 - \sigma) = (1 - \sigma) \left(\sum_{i=0}^{\infty} \sigma^i - \sum_{i=0}^k \sigma^i \right) = \sigma^{k+1} (1 - \sigma) \sum_{i=0}^{\infty} \sigma^i = \sigma^{k+1}$$

7.2.3 Berechnung der mittleren Warteschlangenlänge \bar{Q}

$$\bar{Q} = \sum_{i=0}^{\infty} i P_i = \sum_{i=0}^{\infty} i \sigma^i (1 - \sigma) = (1 - \sigma) \sum_{i=1}^{\infty} i \sigma^i = \sigma (1 - \sigma) \sum_{i=1}^{\infty} i \sigma^{i-1} = \sigma (1 - \sigma) \sum_{i=1}^{\infty} \frac{\partial \sigma^i}{\partial \sigma} = \sigma (1 - \sigma) \frac{\partial \sum_{i=1}^{\infty} \sigma^i}{\partial \sigma} = \sigma$$

7.2.4 Berechnung der effektiven Übertragungszeit

Wielange dauert es, eine Nachricht von n Bits zu übertragen. Da zu einer Nachricht immer ein Quittung gehört und bei einer gestörten Quittung die Nachricht nochmal übertragen werden muss, betrachten wir diese als zu der nachricht gehörend.

n : Länge der Nachricht + Länge der Quittung in Bits v : Übertragungsgeschwindigkeit in Bits pro Sekunde P_B : Bitfehlerwahrscheinlichkeit $T_{eff}(n)$: Effektive Übertragungszeit der Nachricht mit der Länge n

$P_0 = 1 - P_B$: Wahrscheinlichkeit, dass kein Bitfehler auftritt

$$P_0(n) = (1 - P_B)^n$$

$$P_1(n) = 1 - (1 - P_B)^n = 1 - \left(1 - nP_B + \frac{n(n-1)}{2!} P_B^2 - \dots \right)$$

$$\approx nP_B T_{eff}(n) = \frac{n}{v} (1 + P_1(n) + (P_1(n))^2 + (P_1(n))^3 + \dots)$$

$$\approx \frac{n}{v} (1 + nP_B)$$

8 Beispiele von Standardprotokollen

Die Welt der heutigen Protokolle basiert auf dem ISO/OSI-Schichten-Modell (application layer, presentation layer, session layer, transport layer, network layer, data link layer, physical layer). Das heute in der Praxis übliche TCP/IP-Protokoll kennt jedoch nicht die Schichten presentation layer und session layer.

TCP/IP gibt es in drei Versionen:

- Basisband – mit CSMA/CD
- Breitband – Token Access
- Glasfaser – Reservierten Zugriff

ISO/OSI-Referenzmodell:

Bitübertragungsschicht Dienste: ungesicherte Systemverbindung, Sequenzialisierung, Störungsbenachrichtigung an die Sicherungsschicht

Praxisrelevante Normen:

V24: Öffentliches Fernsprechnetz mit analoger Übertragung

X.21: Öffentliches Fernsprechnetz mit digitaler Übertragung

Sicherungsschicht (Bei TCP/IP besteht diese Schicht aus zwei Teilen: LLC – logical link control, MAC – medium access control)

Aufgaben: Flußkontrolle, Fehlerkontrolle, Reihenfolgekontrolle, Fehlerbenachrichtigung in Schicht 3

Dienste:

- unbestätigte, verbindungsunabhängige Dienst, d. h. senden von unabhängigen Paketen ohne Empfangsquittung (Datagrammdienst)
- bestätigter, verbindungsunabhängiger Dienst
- verbindungsorientierter Dienst, d. h. aufbau einer logischen Verbindung zwischen Sender und Empfänger: Nummerieren der Pakete, Quittung

Vermittlungsschicht Aufgaben: Routing, Flußkontrolle, Lastbegrenzung, Festlegung von Struktur und Eigenschaften von Laufwegen (Datagrammdienst oder viertuelle Verbindung in Schicht 2)

	Datagramm (verbindungsunabhängig)	virtuelle Verbindung (verbindungsabhängig)
Initialaufbau	—	erforderlich
Zieladresse	in jedem Paket erforderlich	nur zum Verbindungsaufbau erforderlich
Reihenfolge	nicht garantiert	garantiert durch das Netz
Fehlerüberwachung	—	durch die Vermittlungsschicht
Flußkontrolle	—	im Netz vorhanden

Transportschicht Aufgaben: zuverlässige End-zu-End-Kontrolle (Transportverbindung vom Sender zum Empfänger im Netz), Fragmentieren, Multiplexen (aufwärts oder abwärts), Flußkontrolle (DEE-zu-DEE)

Abwärtsmultiplexen: zusammenfassen von mehreren Kanälen zu einem, Aufwärtsmultiplexen: verteilen von einem Kanal auf mehrere Kanäle

Es gibt drei unterschiedliche Verbindungsklassen:

Type A Netzverbindung mit geringer Restfehlerrate der unteren Schichten und einer geringeren Rate von Fehlern, die der Transportschicht gemeldet werden (sehr hypothetisch, Spezialnetze)

Type B Netzverbindung mit geringer Restfehlerrate der unteren Schichten, aber einer nicht vertretbaren Rate von gemeldeten Fehlern (z. B. digitale Nachrichtenübertragung in öffentlichen Netzen – X.25 Protokoll)

Type C Die Restfehlerrate ist nicht vernachlässigbar, die Fehlerkorrektur muss komplett durch die Transportschicht erfolgen (z. B. Netze mit analoger Übertragung)

dazu gehören fünf Transportprotokollklassen:

Klasse 0 (simple class) keine fehlerbehebung, es stehen nur die Dienste Verbindungsaufbau, Datenübertragung und Verbindungsabbau zur Verfügung – da jedoch auch Fehler in der Transportschicht auftreten können ist diese Klasse zu einfach

Klasse 1 (basic error recovery class) Klasse 0 plus Fehlerbehandlung

Klasse 2 (multiplexing class) Klasse 0 plus Multiplexen

Klasse 3 (error recovery and multiplexing class) Klasse 1 plus Klasse 2

Klasse 4 (error detection and recovery class) Klasse 3 plus umfangreiche Fehlerbehandlung

Sitzungsschicht Aufgaben: Identifikation von externen Benutzern, Synchronisation, Wiederherstellungspunkte, Transparente Absicherung gegen Fehler der Transportschicht