Technische Informatik II

Alle Angaben ohne Gewähr.

Andreas Selig

1. Oktober 2000

Inhaltsverzeichnis

1 Prozesse und Organisationskonzepte

1.1 Zuteilungsalgorithmen bei Mutiprocessing unabhäniger Tasks

Mittlere Aufenthaltszeit eines Auftrags im System

$$\boxed{t_F = \frac{1}{n}\sum_{i=1}^{n} t_{Fi}}$$

Minimale Länge des Scheduels

$$\boxed{t_{OPT} \geq \frac{1}{m}\sum_{i=1}^{n} \tau_i}$$

1.1.1 LPT-Scheduel (Lagrest Processing Time first)

<u>Definition:</u> Beim Freiwerden eines Processes wird stets der längste Auftrag zugeteilt.

Länge des Scheduels bei LPT

$$\boxed{\frac{t_{LPT}}{t_{OPT}} \leq \frac{3}{4} - \frac{1}{3m}}$$

1.1.2 SPT-Scheduel (Shortest Processing Time first)

<u>Definition:</u> Beim Freiwerden eins Processes wird stets der kürzeste der verbleibendes Aufträge zugeteilt.

<u>Satz:</u> Der SPT-Scheduel liefert die minimale Durchlaufzeit und minimiert die mittlere Verspätung.

1.1.3 PR-Scheduel (Preemptive Resume)

<u>Definition:</u> Tasks können während ihrer Bearbeitung an beliebiger Stelle unterbrochen (Preemtion) und später weiter fortgesetzt werden. Im idealisierten Fall erfordert die Unterbrechung keine Prozessorzeit(Overhead).

Die minimale Länge eines PR-Scheduels

$$\boxed{t_{MIN} = \max\{\max_{i=1,\ldots,n}(\tau_i), \frac{1}{m}\sum_{i=1}^{n}\tau_i\}}$$

1.1.4 Echtzeitscheduel

<u>a) Ankunftsreihenfolge</u>

<u>Definition:</u> Tasks werden nach Ankunftreihenfolge zugeteilt.
FIFO (First-In, First-Out)
LIFO (Last-in, First-Out)
Random

<u>b) Priotritätenklassen</u>

<u>Definition:</u> Tasks gehören zu Klassen verschiedener Dringlichkeit. Die Zuteilung orientiert sich am Prioritätsindex einer Task. Es wird zwischen unterbrechenden Tasks (UP) und nichtunterbrechenden Tasks (NUP) unterschieden.

<u>c) Fertigstellungstermin-Scheduel (Due Date, Deadline Scheduel)</u>

<u>Definitionen:</u> Sei d_i der Satz von Fertigstellungsterminen, welcher für die Tasks $T_1 ... T_n$ möglichst nicht überschritten werden soll (Due Dates). Sei t_{Fi} die Fertigstellungszeit der Tasks T_i, $(t_{Fi} - d_i)$ die Verspätung (Lateness) von T_i , $t_{Fi} \geq d_i$, $\max\{t_{Fi} - d_i, 0\}$ die Verspätung des Scheduels. Sei $d(i)$ der Fertigstellungstermin an der i-ten Stelle Task.

<u>Satz:</u> Die Reihung der Tasks nach nichtfallenden Fertigstellungsterminen minimiert die maximale Taskverspätung auf einem Einzelprozessorsystem.

<u>d) Time Sharing (Round Robin)</u>

<u>Definition:</u> Jeder Task erhält maximal Δt pro Zuteilung.

Mathematische Abstraktion: $\Delta t \to 0$ 'Processor Sharing' (Jeder Task erhält $\frac{1}{n}$ Prozessorkapazität, wenn Tasks gleichzeitig im System sind.)
Theoreme zum optimalen Multitasking

Minimale Fertigungszeit:

$$\boxed{t_{min} \geq \max_{0 \leq j \leq L} \{j + \lceil \frac{|Q(j+1)|}{m} \rceil\}}$$

$\lceil ... \rceil$	... Aufrunden auf nächste ganze Zahl
L	... Anzahl der Schichten
m	... Anzahl der Prozessoren
$\|Q\|$	... Mächtigkeit der Menge Q
$Q(j)$	... Menge der Tasks

Minimale Anzahl von Prozessoren, um ein gegebenes Tasksystem innerhalb von $t (t \geq L)$ abzuarbeiten

$$\boxed{m_{min} \geq \max_{0 \leq j \leq L} \lceil \frac{|Q(j+1)|}{t-j} \rceil\}}$$

1.2 Zuteilungsalgorithmen bei Multitasking

Mutitasking entspricht einem Mutiprozessorsystem, wobei die einzelen Tasks nicht unabhänig voneinander sind, sondern über eine Vorgänger-Nachfolger-Relation (Präzedenzgraoh) verknüpft sind.

Optimales Mutitasking
Voraussetzungen:
- Alle Tasks sind gleich lang (normiert 1)
- Präzedenzgraph ist reiner Baum

Optimale Zuteilung:
Bei Freiwerden eines Prozesses ist stets diejenige Task auszuwählen, deren Vorgänger alle schon fertig bearbeitet sind und welche sich in der höchsten Schicht aller unbearbeiteten Tasks befindet.
Bei Gleichheit mehrerer Tasks kann eine bliebige gewählt werden.

underlineVerallgemeinerung auf nichtgleiche Bearbeitungszeiten:
Aufteilen der langen Tasks in mehrere 1-lange.

1.3 Feldverarbeitung (Array Processing)

1.3.1 Formen des Parallelismus

a) Impliziter Parallelismus

Gekennzeichnet durch:
Partielle Ordnung
Erkennung durch einen Compiler
Anwendbarkeit auf Operationsebene, Befehlsebene, Taskebene

b) Expliziter Parallelismus

Gekennzeichnet durch:
Verknüpfung strukturierter Datenmengen
z.B.: Vektor- und Matrizenoperationen

1.4 Zuverlässigkeit und Leistungsfähigkeit

1.4.1 Systemausfälle

Hardwarebedingte Systemausfälle erfolgen nach einer zeitabhänigen Funktion $\lambda(t)$, (Ausfallrate nach Lebendsdauer t, der sogenannte Badenwannenfunktion). Während des Normalbetribs ist die Ausffallrate näherungsweise konstant.

T_F	... Zeitspanne zwischen zwei Ausfällen
$E[T_F]$	... Mittlere Zeit Zwischen zwei Ausfällen (MTBF)

Verfügbarkeit
$$\boxed{a = \frac{\text{MTBF}}{\text{MTBF} + \text{MTTR}}}$$

Ausfallwahrscheinlichkeit
$$\boxed{p = 1 - a = \frac{\text{MTTR}}{\text{MTBF} + \text{MTTR}}}$$

1.4.2 Verfügbarkeit von Gesamtsystemen

a) Serienschaltung

Die Gesamtfunktion wird durch n Teilfunktionen erbracht, die alle erforderlich sind.
$$\boxed{a = \prod_{i=1}^{n} a_i}$$

b) Parallelschaltung

Die Gesamtfunktion kann von allen Teilsystemen erbracht werden.
$$\boxed{a = 1 - \prod_{i=1}^{n} (1 - a_i)}$$

c) Serien-Paralleschaltungen

Stand-by
Das Gesamtsystem entsteht durch die Parralellschaltung zweier Systeme I und II (Redundanzprinzip) Jedes System ist identisch aus n Teilsystemen in Reihenschaltung aufgebaut.
$$\boxed{a_S = 1 - (1 - \prod_{i=1}^{n} a_i)^2}$$

Vermaschung
Das Redundanzprinzip wurde verfeinert, indem jedes der Teilsysteme parallelgeschaltet wurde.
$$\boxed{a_V = \prod_{i=1}^{n} (1 - (1 - a_i)^2)}$$

d) (m aus n)-Schaltung
Das System bestehe aus n Teisystemen, von denen mindestens m für die Erbringung der Gesamtfunktion verfügbar sein müssen.
$$\boxed{a = \sum_{i=m}^{n} \binom{n}{i} \cdot a_0^i \cdot (1 - a_0)^{n-1} = \sum_{i=0}^{n-m} \binom{n}{i} \cdot a_0^{n-i} \cdot (1 - a_0)^i}$$

1.5 Leistungsfähigkeit

1.5.1 Ankunftsprozeß (Arrival Process)

Definition: Stochastischer Punktprozeß zur Beschreibung des Eintreffens von Aufträgen.

$A(t) = P(T_A \leq t)$	... Ankunftsabstands-Verteilungsfunktion
$\frac{\partial A}{\partial t}$	... Verteilungsdichtefunktion (VDF) von T_A
$E[T_A] = \int\limits_0^\infty t \cdot \frac{\partial A}{\partial t} dt$	... Mittlerer Ankunftsabstand

1.5.2 Bedienprozeß (Service Process)

Definition: Zufallsabhänige Dauer T_H der Belegung eines bedienenden Betriebsmittels wie CPU oder Kanal.

$H(t) = P(T_H \leq t)$	... Bediendauer-Verteilungsfunktion
$\frac{\partial H}{\partial t}$	... Verteilungsdichtefunktion (VDF) von T_H
$E[T_H] = \int\limits_0^\infty t \cdot \frac{\partial H}{\partial t} dt$	... Mittlere Bediendauer h

1.6 Das Wartesystem M/M/n

Charakteristische Größen

Wartewahrscheinlichkeit eines Auftrags

$$\boxed{W = \sum_{x=n}^{\infty} P_x = P_0 \cdot \frac{A^n}{n!} \cdot \frac{n}{n-A}}$$

Mittlere Waretenschlangenlänge

$$\boxed{\Omega = \sum_{x=n}^{\infty} (n-x) P_x = P_0 \cdot \frac{A^n}{n!} \frac{\frac{A}{n}}{(1-\frac{A}{n})2}}$$

Mittlere Wartezeit

$$\boxed{w = \frac{\Omega}{\lambda}}$$

Mittlere Durchlaufzeit

$$\boxed{t_F = w + h}$$

Verkehrswert (mittlere Anzahl belegter BE)

$$\boxed{Y = A}$$

Wartezeit-Verteilungsfunktion für FIFO

$$\boxed{W(t) = P\{T_W \leq t\} = 1 - W \cdot e^{-\frac{t(n-a)}{h}}}$$

2 Konzepte der Rechnerarchitektur

2.1 Parallelverarbeitung

2.1.1 Ebenen der Paralleverarbeitung

Ebene der Paralleverarbeitung	Betriebsart	Architekturtyp	Merkmale
Programm (Job)	Multiprogramming Multiprocessing	MIMD	Unabhänige Teil- programme zu einer Aufgabe
Task	Mutitasking	MIMD	Tasks eines Programmes stehen über eine Präzedenz- relation in Beziehung zueinander
Befehl	Feldverarbeitung Array Processing Marco-Piplining	SIMD MIMD (MISD)	Parallelismus bekannt: expilzit oder implizit Pipline-Prinzip angewand auf Befelhsebene
Befelsphase	Pipelining Vektor Processing	(MISD)	Parallele Ausführung einzelner Befehlsphasen auf speziellen Werken
Mikrobefehl	Mikrogrogrammierung		Parallele Ausführung mehrerer Mikrooperationen durch Hardwaresteuerung

2.1.2 Paralleverarbeitungsstrukturen

a) Eng gekoppelte Systeme (Shared Memory)

Merkmale:

* eine oder mehrere CPU's
* ein gemeinsamer Speicher
* CPU's arbeiten parallel an unterschiedlichen Programmen oder Tasks
* ein übergeordnetes Betriebssystem
* Speicherzugriffskonflikte

Abhilfe: Speichermodularisierung, Adreßverschränkung

b) Lose gekoppelte Systeme (Distributed Memory)

Merkmale:

∗ mehrere Rechner aus lokaler CPU mit localem Speicher
∗ Prozessoren arbeiten parallel an i.a. verschiedenen Programmen/Tasks
∗ lokale Betriebssysteme
∗ Kommunikation zwischen Auftragsverwaltung und Einzelrechnern über Botschaften
∗ bei Parallearbeit eines Programms ist ein übergeordnetes Betriebsmittel erforderlich

c) Feldrechner (Array Processing)

Merkmale:

∗ Feld von Verarbeitungselementen
∗ Shared oder Distributete Memory
∗ alle Prozessoren arbeiten synchron
∗ geeignet zur Verarbeitung von großen Matrizen

2.2 Speicherhierarchie und Speichermanagment

2.2.1 Das virtuelle Speicherkonzept

Gründe:

∗ Ablauf großer Programme durch automatisches Segmentieren
∗ Automatische Speicherverwaltung
∗ Verhinderung von Speicherzerstückelung
∗ Automatischer Transport zwischen den Speicherebenen ASP und PSP
∗ Durchsatzerhöhung durch Multiprogrammbetrieb

2.2.2 Adreßumsetzungsverfahren beim Paging-Verfahren

a) Direktes Umsetzen der über Seitentabelle

Merkmale:

∗ reine Softwarelösung
∗ zeitaufwendig, da Durchführung auf Befehlsebene

b) Assoziative Umsetzung

Merkmale:

∗ paralleles Suchen im Assoziativspeicher
∗ schnelle hardwareunterstützte Lösung
∗ assoziativspeicher i.a. klein, da sehr aufwendig

c) Kombiniertes Verfahren

Erweiterung:
Dreistufige Umsetzung entsprechender virtueller Adresse aus:
∗ Blocknummer
∗ Seitennummer
∗ Distanz

Bei Fehlseitenzugriff (Page Fault):
Ergänzungstransport der referierten Seite, dabei gegebenenfalls Verdrängung einer anderen Seite aus dem ASP, Eintrag in Seitentabelle

2.2.3 Lokalität von Programmen

Das Programmevrhalten kann neutral, das heißt ohne Zugrundelegen einer bestimmten Strategie, durch eine Menge von unterschiedlichen Seiten gekennzeichnet werden, auf welche in einem zurückliegenden Zugriffsfenster zugegriffen wurde.Eigenheit der Programmlokalität ist, daß sich diese Seitenmenge über der Zeit i.a. nur langsam verändert und daher immer auch eine gute Vorhersage für das Zugriffsverhalten in der unmittelbaren Zukunft angesehen werden kann.

Seitenreferenzkette
$\boxed{\omega = r_1 r_2 ... r_t \quad , r_t \in N}$ ... Menge der Seiten eines Programmes

Working Set
$\boxed{W(t,T)}$ ... Menge,bestehend aus allen unterschiedlichen Seite, auf welche im zurückliegenden Zeitfenster $[t-T+1,t]$ zugegriffen wurde.

Working Set Größe
$\boxed{w(t,T) = |W(t,T)|}$ ... Mächtigkeit der Menge $W(t,T)$ (Anzahl der Seiten)

$\boxed{s(T) = \lim_{t \to \infty} E[w(t,T)]}$ ... mittlere Working Set Größe

Fehlerzugriffswahrscheinlichkeit
$\boxed{m(T) = \lim_{t \to \infty} P\{r_{t+1} \notin W(t,T)\}}$

2.2.4 Seitenersetzungsstrategienen

a) RANDOM

Unter den im ASP gespeicherten Seiten eines Programmes soll eine beliebige durch eine neue Seite ersetzt werden.

b) FIFO (First-In, First-Out)

Es wird die älteste Seite ausgewählt und ersetzt. Das Alter einer Seite ist die Zeit (in Zugriffszeiten) von ihrem Eintransfer bis zur Gegenwart; die Seite muß dabei ununterbrochen im Speicher sein.

c) LRU (Last Recently Used)

Es wird diejenige Seite verdrängt, deren letzter Zugriff am weitesten zurückliegt. Hierzu werden die Seiten des ASP in einem Stack angeordnet.

d) PFF (Page Fault Frequency)

Es wird diejenige Seite ersetzt, die die häufigsten Fehlseiten-Zugriffe verursacht.

2.3 Schnellspeicherkonzepte (Cache Memory)

2.3.1 Cache-Organisation

a) Voll-assoziativ Cache

Merkmale:
∗ Aufteilung des Cache-Speichers in N Blöcke à n Worte
∗ Unterteilung des Hauptspeichers in Blöcke à n Worte
∗ Kennzeichnung der physikalischen Adresse durch Adreßkennung
    $\boxed{\text{Adress Tag = phsy. Block-Adresse + Gültigkeits-/Status-Kennung}}$
∗ Kennungsspwicher (Tag Memory)
    - N Plätze, je 1 Platz für einen Cache-Block
    - Abbildungsinformationen zwischen phys. Block-Adressen des Hauptspeichers und den Adressen im Cache-RAM
    - Gültigleits-/Status-Kennung

∗ Kennungsspeicher enthält Kennungen aller momentan im Cach-RAM gepufferten Cache-Blöcke
∗ Transport grungsätzlich nur auf Block-Ebene, d.h. Realisierung des Look-Ahead-Prinzips

b) Cache mit direkter Abbildung (Direct-Mapped Cache)

Merkmale:
∗ Cache-Ram, Tag Memory und HSP in N Zeilen unterteilt
∗ Cache-Memory faßt 1 Cache-Block à n Worten pro Cache Line
∗ HSP faßt K Cache-Blöcke pro Zeile entsprechend der K Spalten
∗ durch Low Order Interleaving werden aufeinanderfolgende HSP Adressen auf unterschiedliche Speicherbänke verteilt
∗ Abbildungsinformationen zwischen Chache Line und HSP Spalte abgelegt in Tag Memory
∗ Cache-RAM kann stets nur eine Spalte eineer HSP-Zeile aufnehmen

c) Mengen-assozitiv Cache (Set Associative Cache)

Merkmale:
∗ Unterteilung des Cache-RAM und des Tag-Memory in mehrere Spalten
∗ Jede Spalte des Cache-RAM enthält eine sogenannte Cache Refill Line aus n Worten
∗ direkte Abbildung von HSP-Spalten in Cache-RAM-Spalten über Tag Memory
∗ Aufnahme einer Menge von HSP-Spalten in einer Zeile des Cache-RAM bzw. des Tag Memory
∗ Assoziatives Suchen beschränkt sic hjeweils auf die Einträge einer Zeile des Tag Memory

d) Cache mit Sektorweiser Abbildung (Sector Mapped Cache)

Merkmale:
∗ Unterteilung von Cache-RAM und HSP in N Zeilen (Sektoren) von je K Blöcken (Cache Refill Lines)
∗ Unterteilung des Tag Memory in 2 Spalten
  - linke Spalte enthält Abbildung zwischen Cache Line und HSP-Zeile (Sektor-Adresse)
  - rechte Spalte entält das Abbildungsmuster, nach dem bis zu K Blöcke einer HSP-Adresse direkt in das Cache-RAM abgebildet sind (Validity Bit)

∗ Assoziative Abbildung beschränkt sich auf das Suchen der Zeilen des Cache-RAMs, in die ein Sektor des HSP abgebildet ist

2.3.2 Dateninkonsistenz

a) Ursachen

Mutiprozessor-Systeme mit m CPU's, jeweils einem Cache pro CPU:
Ein HSP-Objekt kann bis zu m Copien auf den Cache-Speichern besitzen. Durch Schreibeoperationen auf demselben Datenobjekt durch simultan arbeitende CPU's können unterschiedliche Versionen des Objekts entstehen, selbst wenn jeder Cache nach den Write-Trough-Verfahren arbeitet.

Single- und Multiprozessorsysteme mit Cache-Speichern, welche nach dem Write-Back-Verfahren können Dateninkonsistenzen erzeuge, wenn ein I/O-Befehl zur Ausführung kommt, bevor das Rückschreibenbegonnen hat oder abgeschlossen wurde.

b) Maßnahmen gegen Dateninkonsistenz

∗ Gleichzeitiges Schreiben auf alle Cache-Speicher mit Kopien ist sicherzustellen.
∗ Löschen und Kennzeichnen von inkonsiesten gewordenen Cache-ojebten durch Betriebssystem
∗ Markierung von HSP-zeilen als non-cacheable
∗ Überwachung durch einen Bus Watcher (snoopy cache)
∗ Rundsenden von Informationen bei Schreiboperationen

2.4 Kopplungsarchitekturen

2.4.1 Kopplung von Systemen

a) Kopplungsgarde

	Enge Kopplung (Speicherkopplung)	Lose Kopplung(Nachrichtenkopplung)
System	Multiprozessorsystem	Multirechnersystem
Betriebssystem	gemeinsam	getrennt
Speicher	gemeinsam gemeinsame Daten gemeinsame Programme	autonome Rechner mit eigenem ASP verteilte Datenbanken verteilete Anwendungsprogramme
Verbindung durch	gemeinsamen Bus Muliport Memory Koppelnetz	LAN WAN Comunication Controler(CC)
Austausch auf	Wortebene	Blockebene

b) Kopplungsarten

∗ Direktverbindungen (Raummultiplex)
∗ Bussysteme (Zeitmultiplex)
∗ Kollpungsnetze (Raum- und Zeitmultiplex)

3 Architektur von Rechnernetzen

3.1 Das Schichtenmodell

Nummer	Schicht	Aufgabe	Dienste
1	Physical	ungesicherte Systemverbingungen	- Aktivierung/Deaktivierung eines Übertragungsabschnittes - Bitserielle übertragung von Schicht-1 Blöcken - Festlegung aller elektrischen,mechanischen, funktionellen und prozeduralen Eigenschften
2	Data Link	gesicherte Systemverbingungen	- Auf-/Abbau von DL-Verbindungen - Übertragung von Schicht-2-Blöcken - Reihenfolge- und Datenflußsteuerung - Fehlererkennung und -behebung
3	Network	Endsystemverbingungen	- Auf-/Abbau von N-Verbindungen - Verkehsklenkung und Vermittlung von Schicht-3-Blöcken/Kanälen - Festlegung eines einheitlichen Adreß- bzw. Nummerierungs-schemas
4	Transport	Teilnehmerverbindungen	- Auf-/Abbau von T-Verbindungen über das Netz hinweg - Multiplexen von T-verbindungen auf N-Verbindungen - Austausch von Schicht-4-Nachrichtenblöcken - Adreßtransformation - Ausgleich unterschiedlicher Netzeigenschaften durch Wahl eines Transportprotokolls - Reihenfolgesicherung, Datenflußsteuerung, Kettung, Segment-ierung, Fehlerbehandlung von Nachrichtenblöcken
5	Session	Kommunikations-verbindungen	- Auf-/abbau von S-Verbindungen - Steuerung des Informationsaustausches - Verstlegung des übertragungsmodus - Sychronisation des Dialogs
6	Presentation	gemeinsames Sprach-mittel	- Auf/-Abbau von P-Verbindungen - Transformation von Daten hinsichtlich Code, Fdormat und Syntax - Verschlüsselung von Daten
7	Application	Informationsaustausch	- Auf-/Abbau von A-Verbindungen - Anwenderprozesse der Benutzer, des System- und des Anwendungsmanagments

3.1.1 Dienstprimitive

Zur Beschreibung der Schnittstelle zwischen den benachbarten Schichten (N) und (N+1) werden sogenannte (N)-Dienstprimitive eingeführt:
* REQUEST
* RESPONSE
* INDICATION
* CONFIRM

3.1.2 Verbindungskonzepte

a) Verbindungsorientierter Nachrichtenaustausch (LLC 1)

Der verbindungsorientierte Nachichtenaustausch (Connection Orientated 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 Folgenummer, Reihenfolgesicherung, Fehlererkennung, automatische wiederholung sowie datenflußsteuerung sehr zuverlässig gesteuert werden. Die Parameter hierzu sind i.a. Gegenstand eines gegenseitigen Abstimmungsprozesses beim Verbindungsaufbau. Sie sind während einer Verbindung nicht änderbar. Die Verbindung wird nach Abschluß des Datenaustausches wieder abgebaut (Connection Relase).

b) Verbindungsloser Nachrichtenaustausch (LLC 2)

Diese Variante wird insbesondere für LAN's interessant, wo die zuverlässige Datenübertragung nicht in allen Schichten dieselben Funktionen wie Reihenfolgesicherung, Fehlerbehandlung und Datenflußsteuerung erfordert.

c) Quittierter Datagramm-Verbindungstyp (LLC 3)

Hierbei handelt es sich um einen quittierten, verbindungslosen Dienst.

3.2 Netzstrukturen und Adressierung

3.2.1 Elementare Strukturen

a) Maschennetz

* reines Maschennetz aus N Knoten
* $\frac{N \cdot (N-1)}{2}$ Verbindungswege
* kürzestmögliche Verbindungen
* unwirtschaftlich für große N
* Anwendung: Höhere Netzebenen

b) Sternnetz

* zentraler Netzknoten
* N Verbindungsleitungen
* 2 Leitungen pro Verbindung erforderlich
* wirtschaftlich für große N
* Anwendung: Zellulare Mobilfunknetze

c) Linien-/Busnetze

* Zentraler Bus als Breitband-Übertragungssystemmit passiver Ankopplung
* Betrieb mit Vielfachzugriffsverfahren
* Anwendung: Lokale Netze mit Paketvermittlung

d) Ringnetz

* Zentraler Ring als Breitband-übertragungssystemmit aktiven Kopplungselementen
* Betrieb mit zentralem Takt und sychronem TDM oder dezentral im Paketmode (Token Ring)
* Anwendung: Lokale Netze mit Durchschalte- oder Paketvermittlung

e) Dualbus

* Paar unidirektionaler Busse
* Zeitschlitzstruktur durch Kopfstationen
* Reservierungsanforderungen
* Anwendung: DQDB (Distributed Queue, Dual Bus)

f) Faltbus

* Ein Bus mit ausgehendem (outbound) und ankommendem (inbound) Segment
* Zentrales Reservierungsverfahren durch Kopfstation

3.2.2 Adressierung

Unter Adressiereung wird die Kennzeichnung von Instanzen innerhalb eines Nachrichtennetzes mittels einer Adresse verstanden. Die Adresse ist i.d.R. aus alphanumerischen Zeichen zusammengesetzt und folgt einem allgemeingültigen Adressierungsschema innerhalb eines Adressierungsbereichs.

3.3 Kommunikationsprotokolle

3.3.1 Protokollfunktionen im LAN

Nummer	Name	Funktion
1	Ph	- Aktivierung, Deaktivierung - Codierung, Decodierung von Leitungssignalen - Präambel-Erzeugung - Bitsynchronisation, Rahmensynchronisation - übertragung von unstrukturierten Bitströmen
2a	MAC	- Empfang von MA.PDU's (MAC-Rahmen) - Medienzugriff für MA-PDU's einschließlich Konfliktlösung - Erzeugung der FCS (Frame Checking Sequence) - Auswertung der FCS - Pufferung der MAC-Rahmen
2b	LLC	- LLC1, LLC2, LLC3

3.3.2 Das Alternierende-Bit-Protokol

a) Dienstrimitive

Dienst	Dienstprimitive
Verbindungsaufbau	CON_REQ, CON_IND, CON_RESP, CON_CONF
Verbindungsabbau	DISC_REQ, DISC_IND, DISC_CONF
Datentransfer	DATA_REQ, DATA_IND

b) Protokolldateneinheiten

Protokoll	Protokolldateneinheiten	Rahmenart
Verbindungsauf-/abbau	CR, CC, DR	Unnumerierte Steuer-PDU
Datentransfer	DT(X,N) AK(N)	Nummerierte Daten-PDU Nummerierte Steuer-PDU

3.3.3 Das Handshake-Verfahren

Das Handshake-Verfahren ist die einfachste Methode, bei der Sender und Empfänger sich nach jedem Übertragungsvorgang neu sychronisieren durch unmittelbare Quittiersignale.

Vorteil: Einhalten der Reihenfolge
Einfache Steuerung
Nachteil: Geringe Auslastung des Kanals bei großen Ausbreitungsverzögerungen

Kanalauslastung:

$$\boxed{\rho = \frac{t_0}{t_0+2\tau+t_{ACK}}}$$

Effektive Kanalauslastung:
$\boxed{v_{max} = \rho \cdot v}$

Das Handshake-Verfahren ist ungeeignet bei großen $\frac{2\tau}{t_{ü}}$.

3.3.4 Fenster-Mechanismen (Sliding Window Protocols)

Sender		Empfänger	
S	Sendefolgenummer	R	Empfangsfolgenummer
S_U	niedrigste Folgenummer der gesendeten, nichtquittierter Blöcke	R_U	niedrigste Folgenummer, die beim Empfänger akzeptiert wird
S_O	Folgenummer des nächsten zu sendenen Blocks	R_O	erste Folgenummer, die beim Empfänger nicht mehr akzeptiert wird

Maximale Fenstergröße beim Sender
$\boxed{1 \leq W_S \leq M - 1}$

Fenstergröße Empfänger
$\boxed{W_R = R_0 - R_U,\ 1 \leq W_R \leq M - 1}$

4 Protokolle für Rechnernetze

4.1 Zentrale Zugriffsverfahren

Vorteile:	-Vollständige Übersicht
	-Anpassungsfähigkeit (fairness, lastabhänige Zuteilung, ...)
Nachteile:	-Verfügbarkeit bei Ausfall
	-Kommunikationsaufwand (Signalisier-Overhead)

4.1.1 Sendeaufrufverfahren

a) Polling-Verfahren

b) CRMA (Cyclic Reservation Multiple Access)

4.1.2 Reservierungsverfahren

a) FIFO

b) Daisy Chaining

4.2 Dezentrale Zugriffsverfahren

Vorteile:	- Hohe Verfügbarkeit
Nachteile:	- Probleme der Übersicht
	- Konfliktlösungsstrategie erforderlich

4.2.1 Sendeaufrufverfahren

a) Token-Ring-Verfahren

Single-Token-Verfahren
Es existiert stets nur ein Token (free oder busy)

Single-Frame-Verfahren
Es kann stets nur ein Rahmen übertragen werden. Nach Abschluß der Übertragung wird ein neues Token generiert und weitergesandt.

Multiple-Token-Verfahren
Im Anschluß an einen gesandten Rahmen wird sofort ein neues (Free-)Token ausgesandt.

Timed-Token-Verfahren
Nach erhalt des token erhält Station das Senderecht für bestimmte Zeit.

Token-Ring-Prioritäten
8 Prioritätsklassen.

b) Token-Bus-Verfahren

* Bildung eines logischen Rings auf einer physikalischen Busstruktur
* Adreßverkettung der angeschlossenen Stationen, Festlegung durch Konfigurierung
* Token-Weitergabe mit Zieladresse
* Senderecht nach erhalt des Tokens; Sendedauer nach Zeitvergabeverfahren

c) Empty Slot Indication

d) Register Insertion

4.2.2 Reservierungs-Verfahren

DQDB (Distributed Queue Dual Bus)
* Dualbusstruktur
* Zeitschlitzstrukturauf den Bussen durch die Kopfstationen

4.2.3 Wettbewerbsverfahren

a) Aloha- und Slotted Aloha-Verfahren

b) CSMA

c) CSMA/CD

d) Dominanz-Verfahren

4.3 Gesicherte Datenübertragung mit HDLC

4.3.1 Rahmenaufbau

a) I-Rahmen

FLAG	ADDRESS	CONTROL	INFORMATION	FCS	FLAG
01111110	8 bit	8 bit	X bit	16 bit	01111110

CONTROL-Feld:

1	2 3 4	5	6 7 8
0	N(S)	P/F	N(R)

b) S-Rahmen

FLAG	ADDRESS	CONTROL	FCS	FLAG
01111110	8 bit	8 bit	16 bit	01111110

CONTROL-Feld:

1	2	3 4	5	6 7 8
1	0	S	P/F	N(R)

c) U-Rahmen

FLAG	ADDRESS	CONTROL	FCS	FLAG
01111110	8 bit	8 bit	16 bit	01111110

CONTROL-Feld:

1	2	3 4	5	6 7 8
1	1	M	P/F	M

4.3.2 Rahmenaufgaben

a) S-Rahmen

S-Rahmen sind Steuerrahmen mit Folgenummer, die nach der Initialisierung ausgetaucht werden.

RR Reaceive Ready
Zeigt empfangsbereitschaft für I-Rahmen an

RNR Receive Not Ready
Es können vorübergehend keine I-Rahmen angenommen werden

REJ Reject
Wiederholungsanforderung für alle I-Rahmen ab $N(S) = N(R)$

SREJ Selective Reject
Wiederholungsanforderung nur für I-Rahmen mit $N(S) = N(R)$

b) U-Rahmen

U-Rahmen sind Steuerrahmen ohne Folgenummer für:
* Initialisierung des Abschnittes
* Rücksetzen der internen Zähler der Stationen
* Beenden einer Data Link Verbindung

SNRM Set Normal Response Mode
SARM Set Asynchronous Response Mode
SABM Set Asychronous Balanced Mode
Initialisieung der Data Link Verbindung

DISC Disconnected
Stationen gehen in den Wartezustand

UA Unnumbered Acknowledge
Bestätigt Empfang eines U-Rahmen

FRMR Frame Reject
Sendet der Gegenstation das Steuerfeld eines Rahmens zurück

SNRME Set NRM Extended
SARME Set ARM Extended
SABME Set ABM Extended
Vereinbart ein weiteres Steuerfeld