

# D. Rechnernetze

---



Autor: Markus Möller



- Bezug: Vorlesung bei Herrn Martini und die dort angegebene Referenzliteratur.
- Dieser Extrakt entstand als Vorbereitung auf meine Diplomprüfung (Teil: Praktische Informatik). Er faßt einige Themen einfach zusammen und mag etwas unorthodox erscheinen.
- Erstellt auf Apple Macintosh.

## 1. Einleitung

Beim klassischen Rechner sind Leistungssteigerungen durch bessere Komponenten oder Parallelisierung möglich. Man kann innerhalb eines Rechners die Parallelisierung vornehmen, was diesen sehr teuer macht, oder man koppelt getrennte Systeme sehr an per Rechnerbus, was die mögliche Entfernung stark einschränkt. In diesen Fällen wird der Datenaustausch durch die jeweilige CPU vorgenommen.

Bei Rechnernetzen ist jeweils ein Netzcontroller für die Regelung des Datenaustauschs verantwortlich. Das Ziel eines Rechnernetzes ist, daß das Netz der Rechner und die Knoten die Komponenten sind. Der auch hier anfallende hohe Verwaltungsaufwand betrifft aber den Controller und nicht die CPU. Bei zu geringer Geschwindigkeit kann man den Controller ersetzen oder neue Übertragungsmedien verwenden. Durch die hohe Komplexität eines Netzes muß man möglichen Fehlern geschickt begegnen. Hier wird die Gesamtproblematik strukturiert und mit formalen Methoden angegangen.

Ein **Multicomputer-System** besteht aus mehreren Rechnern, die private Speicher besitzen und untereinander verbunden sind. Hier werden nur verteilte Systeme betrachtet, die

- keinen gemeinsamen Speicher haben,
- sich aus mehreren (verschiedenen) Rechnern zusammensetzen und
- bei denen Betriebssystemprozesse durch physikalische Rechnerkoppelung miteinander kommunizieren können.



Die Rechnerkopplung kann direkt (LAN) oder indirekt über mehrere Teilnetze erfolgen.

Im Unterschied zu klassischen Systemen findet die Kommunikation zwischen Prozessen durch Nachrichten statt und nicht durch Zugriff auf gemeinsamen Speicher.

## 1.1 Klassifikation und Trends

Multiprozessorsysteme haben eine Distanz von 10 cm und eine Datenrate von bis zu  $10^9$  Bit pro Sekunde. Ähnlich schnell sind am oberen Ende High Speed LANs und MANs und am unteren Ende Local Area Networks (LAN). Beide haben eine Distanz von 10 bis 1000 Metern. Wide Area Networks (WAN) zwischen 1 und 10000 km groß.

### Wichtige Trends:

1. Computer nicht nur zum Rechnen, sondern für Kommunikation.
2. Standardisierung von Protokollen (FDDI, FDDI-II, FFOL, DQDB, ATM,...)
3. Daten, Grafik und Video als wichtige Anwendungen. Der Sprachanteil in öffentlichen Netzen sinkt stetig.
4. Die Technologie eilt den Anwendungen weit voraus. Siehe ISDN.

Das Produkt aus Datenrate und Distanz ist ein entscheidender Faktor für die Leistungsfähigkeit eines Netzes. Zur Zeit werden die Netz schneller und überbrücken immer größere Entfernungen.

## 1.2 Zur Leistungsfähigkeit von Netzen

Wichtiger Parameter für die Leistungsfähigkeit: Produkt aus Bandbreite und Distanz. Die Bandbreite ist der nutzbare Frequenzbereich. Je größer er ist, umso mehr Daten können gleichzeitig übertragen werden.

$$\begin{aligned} \frac{\text{Bandbreite} \times \text{Distanz}}{\text{Signalgeschwindigkeit}} &= \text{Bandbreite} \times \frac{\text{Distanz}}{\text{Lichtgeschwindigkeit}} \\ &= \text{Bandbreite} \times \text{Signallaufzeit} \\ &= \text{Länge des Übertragungsmediums in bit} \end{aligned}$$



Vergleich mit der Länge (L) einer typischen Nachricht (a heißt Lokalitätsfaktor):

$$a = \frac{\text{Länge des Datenwegs (in bit)}}{\text{Länge einer mittleren Nachricht}}$$

$$= \frac{\text{Bandbreite} \times \text{Distanz}}{\text{Signalgeschwindigkeit} \times \text{Durchschnittslänge } L}$$

### Was ist der „Lokalitätsfaktor“ a?

Die Frequenzbreite zeigt, wieviele Nachrichten parallel verschickt werden können. Die Signallaufzeit gibt an, wie lange eine Übertragung dauert (1 Bit sozusagen). Wenn die Signallaufzeit sehr groß ist, dann können mehrere Nachrichten hintereinander auf der Leitung liegen. D.h. also, die Frequenzbreite gibt die Breite (Spuren) der Autobahn an und die Signallaufzeit die Länge der Autobahn. Das Produkt der beiden ist folglich ein Maß dafür, wieviele Nachrichten gleichzeitig (parallel und sequentiell) auf dem Übertragungsmedium „liegen“ können.

Dividiert man dieses Produkt durch die Länge der Durchschnittsnachricht, so weiß man, wieviele Nachrichten im Schnitt auf diesem Medium liegen können. Dieser Wert a heißt Lokalitätsfaktor.

### Wieviel ist 1 Gbit/s?

Ein Netz Erde-Mars ist mit 64 kbit/s genauso leistungsfähig wie ein Netz Boston-Frisco mit 1Gbit/s. Denn um 1Gbit 49 Millionen km zu transportieren braucht das Marsnetz circa 15 s (wegen Langsamkeit) und das Frisconetz (wegen Entfernung) ebenfalls.

## 2. Protokolle

### 2.1 Aufgaben eines Protokolls

Festlegung von Regeln

- für Datenaustausch im Normalfall
- für den Fehlerfall
- für Form und Interpretation der elektrischen Signale
- zum Verbindungsaufbau



- für Adressierung
- für Wegwahl zum Empfänger
- zur Synchronisation der Partner
- zur Sicherung gegen
  - (a) Übertragungsstörungen
  - (b) unerlaubtes Mithören
  - (c) Senden durch Fremde
- für einheitliche Datendarstellung
- für die Semantik des Datenaustausches
- für Quittungen

## 2.2 Das Hierarchieprinzip

Wegen der Vielschichtigkeit der Anforderungen an Protokolle werden diese in Schichten aufgeteilt. Jede kommuniziert nur mit ihren direkten Nachbarn. Die unterste Schicht sendet eine PDU (Protocol-Data-Unit) über die physikalische Leitung. Eine PDU enthält die eigentlichen Daten, eingepackt in Zusatzinfos jeder Schicht, die sie durchläuft.

Jede Schicht kennt nur den Service Access Point (SAP) der unteren. Keine Interna.

Der Vorteil des Hierarchieprinzips ist seine Überschaubarkeit. Nachteilig sind, daß jede Ebene zusätzliche Kontrollinformationen hinzufügt zu einer Nachricht und eventuell die PDU segmentiert werden muß. Das hätte zur Folge, daß der Overhead tieferer Ebenen pro Segment hinzugefügt wird anstatt pro Paket. Ferner wird Numerierung erforderlich.

Bei der Anzahl der Schichten macht man einen Kompromiß zwischen unübersichtlichen wenigen Schichten, die schwierig zu implementieren sind, aber wenig Overhead produzieren, und vielen Schichten, die übersichtlicher zu handhaben sind, jedoch viel Overhead provozieren und einfach umzusetzen sind.

Funktionell betrachtet gibt es drei komplexe Kategorien von Schichten:

1. Anwendungsorientierte, die oben liegen:
  - (a) Anwenderebene,
  - (b) Darstellungsebene,
  - (c) Sitzungs- bzw. Steuerungsebene.
2. Transportorientierte in der Mitte zum zuverlässigen Datentransport bei beliebigem Netzwerk:



- (a) Transportebene und
  - (b) Netzebene.
3. Technologieorientierte unten:
- (a) Leitungs- bzw. Sicherungsebene und
  - (b) Bitebene.

Das OSI (Open Systems Interconnection) Modell standardisiert keine Protokolle, sondern legt lediglich einen Rahmen für die Standardisierung fest. Es ist unabhängig von Implementierungen. Auch Protokolle im Rahmen dieses Modells sind implementierungsunabhängig.

**TABELLE 7: ISO - Open System Interconnection Referenzmodell**

Schicht	Layer-Name	Aufgabe	Ausgetauschte Einheiten	Orientierung
7	<b>Application/</b> Anwendung	Bereitstellung von <u>Netzwerkdiensten</u> für spezielle Anwendungen.	APDU	Anwendung
6	<b>Präsentation/</b> Darstellung	Nachrichtenübersetzung von lokaler Syntax in <u>Transfersprache</u> und umgekehrt, evtl. auch De- bzw. Encodierung.* Datenkompression, Formatierung für Ausgabegerät.	PPDU	Anwendung
5	<b>Session/</b> Steuerung	<u>Organisieren</u> und <u>Synchronisieren</u> des Dialogs sowie Managen des Datenaustauschs: Verbindung für gesamte Transaktion herstellen und abrechnen.	SPDU	Anwendung
4	<b>Transport</b>	Bereitstellung einer <u>zuverlässigen</u> , definierten Menge von <u>Ende-zu-Ende-Nachrichtentransfermöglichkeiten</u> unabhängig vom darunterliegenden Netzwerktyp.	TPDU	Transport
3	<b>Network</b>	Adressierung ( <u>Routing</u> , Wegwahl).	Paket	Transport
2	<b>Data Link/</b> Sicherung	Bereitstellung einer <u>zuverlässigen Informationstransfermöglichkeit zwischen direkten Nachbarn</u> : Fehlererkennung, Retransmission.	Rahmen (frame)	Technologie
1	<b>Physical</b>	Bereitstellung von Mitteln zur <u>Übertragung eines Bitstroms</u> zwischen zwei Netzknoten.	Bit	Technologie

\*. Es können also drei Syntaxen existieren: Sender-, Transfer- und Empfängersyntax.

Für jede dieser Schichten müssen Protokolle standardisiert werden. Wichtige Gremien für sowas: **I**nternational **S**tandards **O**rganisation, **C**omité **C**onsultatif **I**nternational de **T**élégraphique et **T**éléphonique, **I**nstitute of **E**lectrical and **E**lectronics



Engineers, European Computer Manufactures Association, American National Standards Institute.

## 3. Physical Layer (Bitebene, „1“)

Die Bitübertragungsschicht befaßt sich mit den elektrischen (optischen) Verbindungen zwischen Rechensystemen.

Protokolle der Bitebene bestimmen:

- die physikalische Darstellung von „0“ und „1“,
- die zeitliche Dauer eines Bits (d.h. die Datenrate),
- die Übertragungsrichtung für den Kanal
  - (a) simplex: Nur eine Richtung nutzbar.
  - (b) (voll-)duplex: Zeitgleiche Übertragung in beide Richtungen.
  - (c) halb-duplex: Beiden Richtungen, aber nicht gleichzeitig.
- ob die Übertragung bitseriell oder bitparallel ist,
- ob eine Übertragung synchron (in festem Zeittakt), oder asynchron abläuft,
- ob die Leitung exklusiv oder im Multiplex-Betrieb genutzt wird.

### 3.1 Medien

#### 3.1.1 (verdrilltes) Kupferkabel

Verdrillt, weil durch parallele Leitungen eine einfache Antenne entstehen würde, so aber nicht. Damit läßt sich die Einstreuung in benachbarte Leitungspaare reduzieren.

- geringe Kosten
- geringe Übertragungsrate (max. wenige Mbit/s)
- hohe Fehlerrate

#### 3.1.2 Koaxialkabel

- hohe Datenrate
- geringe Fehlerrate
- problemloser Passiv<sup>1</sup>-Anschluß

---

1. „Passiv“ bedeutet „anzapfen“, wogegen „aktiv“ empfangen, evtl. verstärken und erneut senden (also kein „Abzapfen“) bedeutet.



### 3.1.3 Glasfaser

- hohe Kosten für Sender/Empfänger
- extrem hohe Datenrate (praktisch unbegrenzt)
- problematischer Passiv-Anschluß<sup>2</sup>
- geringe Fehlerrate

## 3.2 Multiplexing

Multiplexen dient der wirtschaftlich besseren Ausnutzung eines Kanals. Die Multiplexverfahren können grundsätzlich in zwei Gruppen aufgeteilt werden: FDM (Frequency Division Multiplexing, Frequenzmultiplexverfahren) und TDM (Time Division Multiplexing, Zeitmultiplexverfahren). Beim FDM wird das Frequenzspektrum unter den einzelnen logischen Kanälen aufgeteilt; somit erhält jeder Benutzer allein ein einziges Frequenzband. Beim TDM wechseln sich die Benutzer der Reihe nach ab und jeder erhält die ganze Bandbreite für eine kurzen Augenblick.

## 3.3 Gleichlaufverfahren<sup>3</sup>

Zeichen werden eindeutig mit Codewörtern codiert. Z.B. ASCII oder EBCDIC. Damit das empfangende Gerät das übertragene Bitmuster korrekt decodieren und interpretieren kann, muß es folgendes wissen:

1. die benutzte Bitrate (d.h. die Dauer jeder Bitzelle),
2. Anfang und Ende jedes Elements (Buchstabe oder Byte), und
3. Anfang und Ende jedes kompletten Nachrichtenblocks oder Frame.

Diese drei Faktoren sind als Bit- oder Clocksynchronisation, Byte- oder Buchstabsynchronisation und Block- oder Framesynchronisation bekannt.

Wenn die Daten aus Buchstaben mit beliebig großen Pausen zwischen zwei Buchstaben bestehen, dann wird jeder Buchstabe unabhängig übermittelt und der Empfänger resynchronisiert sich beim Start jedes neuen empfangenen Buchstabens. Hierfür wird asynchrone Übertragung benutzt. Wenn die Daten jedoch aus kompletten Blöcken bestehen, die jeder mehrere Bytes oder Buchstaben enthalten, dann müssen Sender- und Empfänger-clock längere Zeit synchron laufen, und daher wird synchrone Übertragung verwendet.

---

2. Weil das Aufspalten von Glasfasern schwierig und das Verschmelzen noch schwieriger durchzuführen ist, womit Anschlußpunkte gesetzt werden.

3. Gleichlauf bedeutet im gleichen Zeitmaß arbeiten.



Beim Asynchronverfahren wird die Bitsynchronität folgendermaßen gewährleistet: Die Flanke des Startbit wird zusammen mit der Clock des Receivers benutzt, um die Mitte jedes Bits abzuschätzen. Die Clock ist einige Male höher getaktet als die übermittelte Bitrate. Dies funktioniert, weil die maximale Bitrate beim asynchronen Verfahren relativ gering ist; unter 19 kbps und immer eine synchronisierende Flanke vor jedem Buchstaben gesendet wird.

Beim synchronen Verfahren werden keine Start- und Stopbits benutzt. Daher muß eine andere Bit(Clock-)synchronisationsmethode verwendet werden. Zwei Verfahren sind möglich: Entweder eine Leitung für Daten und eine zweite für die Clock (benötigt aber zwei Leitungspaare, die normalerweise nicht gegeben sind) oder die Clock ist in den Bitstrom eingebettet und wird vom Empfänger extrahiert oder (Möglichkeit 2b) die Information wird so codiert, daß eine ausreichende Anzahl von Flanken im Bitstrom auftaucht, die eine separate Clock beim Empfänger synchronisieren kann.

### 3.3.1 Asynchronübertragung (Start/Stop-Übertragung)

Hierbei wird jedes Zeichen allein in einen Rahmen verpackt. Diese Methode wird benutzt, wenn die zu übertragenden Daten nach zufälligen Intervallen anfallen. Daher ist es für den Empfänger notwendig, sich beim Start zu resynchronisieren. Dafür wird jedes Zeichen eingekapselt bzw. „framed“ mit einem Start- und einem Stopbit. Das Startbit ist low und das Stopbit ist high und der untätige Kanal ebenso. D.h. der erste 1 0 Übergang signalisiert dem Empfänger den Start jedes neuen Buchstabens. Der Status der folgenden Bits wird jeweils durch Abtasten in der Mitte, mit Hilfe einer Clock, gecheckt. Durch die Start- und Stopbits ist auf jeden Fall ein 1 0 1 Übergang zwischen aufeinanderfolgenden Buchstaben.

### 3.3.2 Synchronverfahren

Wenn große Blöcke von Daten verschickt werden, dann ist es nicht sinnvoll, jeden Buchstaben / jedes Byte mit zusätzlichen Rahmenbits zu versehen. Der Bitstrom ist passend codiert (s.u.), daß der Empfänger bitsynchron ist. Alle Frames haben vorher einige reservierte Bytes, um sicherzustellen, daß der Empfänger die korrekten Buchstabenlängen benutzt. Jeder Frame wird ferner direkt von einem Paar reservierter Bytes eingekapselt, die seinen Anfang und sein Ende signalisieren. Während der unbeschäftigten Zeit werden kontinuierlich Bytes gesendet, um Bit-



und Bytesynchronisation beim Empfänger aufrechtzuerhalten. Der Empfänger ist also immer auf den Sender synchronisiert.

## 3.4 Signaldarstellung

Die Darstellung binärer Signale kann prinzipiell nach drei unterschiedlichen Methoden erfolgen:

### 3.4.1 NRZ (No-Return-To-Zero)<sup>4</sup>

Hier wird „0“ und „1“ ein fester Spannungswert zugewiesen. Vorteil: Bei jedem Takt wird ein Bit übertragen. Also gute Ausnutzung. Nachteilig ist, daß es keine Flankenwechsel bei „0“- bzw. „1“-Blöcken gibt. Eine Selbsttaktung ist so folglich nicht machbar.

### 3.4.2 Differentieller Code

Bei „0“ kein Pegelwechsel, bei „1“ grundsätzlich ein Pegelwechsel, also auch bei „1“-Blöcken. Auch hier eine gute Ausnutzung, weil bei jedem Takt ein Bit übertragen wird. Nachteil: kein Flankenwechsel bei „0“-Blöcken. Da aber auch „1“-Blöcke vorkommen, hat der Empfänger die Möglichkeit, seine Clock zu synchronisieren.

### 3.4.3 Manchester Code

Stets Pegelwechsel in der Bitmitte. „0“ ist high-low und „1“ ist low-high. Der Bitstrom muß also in der zweiten Hälfte jeder Bitzelle abgetastet werden. Die in jedem Bittakt auftauchende Flanke ermöglicht ein ständige Synchronisieren der Empfängerclock. Selbsttaktung. Es gibt dadurch auch keine Problem bei Blöcken mit „1“en oder „0“en. Das Datenende ist leicht am Nicht-Pegelwechsel zu erkennen. Der Nachteil ist, daß zwei charakteristische Einheiten pro Bit übertragen werden müssen. D.h. doppelte Taktrate bzw. halbe Datenrate.

### 3.4.4 (4B,5B)-Code

Codiert 4 Bit in 5 Bit dermaßen, daß maximal zwei „0“ aufeinander folgen. Anschließend wird das Ergebnis noch mit differentiell codiert, d.h. eine „1“

---

4. Der Name kommt daher, daß im Gegensatz zum bipolaren Codieren nicht nach jedem Bit das Signal auf 0 zurückgeht. Beim bipolaren Verfahren gibt es drei Zustände, also Ausschläge nach oben und unten. Nach jedem geht das Signal wieder auf 0 zurück. Return To Zero (RTZ-Verfahren).

erzeugt immer einen Pegelwechsel, eine „0“ keinen Wechsel. Auf diese Weise wird ein Pegelwechsel nach spätestens zwei Bits garantiert.

Es handelt sich hier um einen Kompromiß zwischen Effektivität und Selbsttaktung. Es sind genug Flanken vorhanden, um Selbsttaktung zu ermöglichen und bei jedem Takt wird ein Bit übertragen. Allerdings darf nicht vergessen werden, daß 20% der Bits der Codierung gelten. Im Vergleich zum Manchester Code wird aber 50% geringerer Takt benötigt. Also eine sinnvolle Kombination.

## 3.5 Wichtige Protokolle der Ebene 1:

- V.24. asynchron, zeichenorientiert, 21 Leitungen
- X.21: synchron, bitorientiert, 8 Leitungen
- FDDI (Fiber Distributed Data Interface): Verwendung des (4B,5B)-Codes
- Busprotokolle, z.B. IEC-Bus (parallele Schnittstellen)

Die obersten beiden benutzen nur eine Leitung pro Richtung für den Datentransport.

### 3.5.1 Das V.24-Protokoll

Es wurde für analoge Telefonnetze mit Datenraten von maximal 20 kbit/s entwickelt. Es beschreibt die Schnittstelle zwischen Computer (DTE, Data Terminal Equipment) und Modem (DCE, Data Communication Equipment).

Der V.24-Standard beschreibt physikalische Größen (Steckerabmessung, Pindurchmesser, Pinanordnung, Widerstand, etc.) sowie Zahl, Art und Funktion der Leitungen. Man unterscheidet 5 Klassen von Leitungen:

- Erdleitungen (2)
- Datenleitungen (2)
- Steuer- und Meldeleitungen (9)
- Taktleitungen (3)
- Hilfskanäle (5)

Jede Leitung hat genau eine eigene Bedeutung.

Dieses Protokoll der Bitebene muß die mechanischen, elektrischen, funktionellen und verfahrenstechnischen Einzelheiten der Schnittstelle genau festlegen.



Die mechanische Spezifikation enthält die Abmaße des 25-poligen Steckers und die Numerierung der Pins.

Die elektrische Spezifikation legt die Spannungsbereiche für 0 und 1 fest, die maximale Übertragungsrate und die maximale Kabellänge (15m).

Die funktionale Spezifikation bestimmt, welche Schaltungen mit jedem der 25 Anschlüsse verbunden sind und was sie bedeuten. Wird z.B. der Computer/Termi-nal eingeschaltet, bestätigt es dies über Data Terminal Ready (Pin 20), indem der Anschluß auf logisch 1 gesetzt wird; Daten werden auf der Transmitleitung (Pin 2) gesendet und auf der Receive-Leitung (Pin 3) empfangen.

Die verfahrenstechnischen Spezifikationen stellen das Protokoll dar, das festlegt, welche Abfolge von Ereignissen überhaupt erlaubt ist. Das Protokoll ist durch Aktion-Reaktionspaare definiert. Stellt z.B. das Terminal die Anfrage Request to Send (4), dann meldet das Modem Clear to Send(5), falls es in der Lage ist, Daten zu empfangen. Für die anderen Anschlüsse existieren ähnliche Aktion-Reaktionspaare.

Oft muß man zwei Computer mit dieser Schnittstelle verbinden. Da keiner von beiden ein Modem ist, ergibt sich ein Schnittstellenproblem. Dieses wird gelöst, indem die Computer über ein Gerät mit dem Namen Nullmodem miteinander verbunden werden. Das Nullmodem verbindet den Transmit-Anschluß des einen Computers mit dem Receive-Anschluß des anderen, auch einige andere Anschlüsse werden in gleicher Weise gekreuzt.

### 3.5.2 Parallelschnittstellen

Über parallele Leitungen werden mehrere Bits gleichzeitig übertragen. Die Hauptanwendung ist der Rechnerbus, da bei größeren Entfernungen Synchronisationsprobleme auftauchen.

Bestandteile der Parallelschnittstelle sind

- Datenleitungen
- Erden<sup>5</sup>
- Steuerleitungen

---

5. Wie der Name sagt: Verbindung zur Erde, so daß alle Geräte dasselbe 0-Potential haben.



**Beispiel:**

- 8 Datenleitungen D1,...,D8
- Steuerleitungen:
  - DR (Data Request<sup>6</sup>), d.h. Daten sind bereit.
  - DA (Data Acknowledgement<sup>7</sup>), d.h. Daten sind vom Empfänger abgenommen worden.

Fünf Phasen:

**TABELLE 8: Phasen der Parallelübertragung**

Phase	Steuerleitungen		Datenleitungen D1,...,D8
	DR (kommt vom Sender)	DA (kommt vom Empfänger)	
1	aus	aus	ungesetzt
2	aus	aus	gesetzt
3	ein	aus	gesetzt
4	ein	ein	gesetzt
5	aus	aus	ungesetzt

## 4. Data Link Layer (Sicherungsebene, „2“)

### Wichtige Aufgaben der Schicht 2:

- Synchronisation zwischen Sender und Empfänger (logisch, nicht physikalisch!)
- Erkennung/Behebung von Übertragungsfehlern (ggf. Weitermelden unkorrigierbarer Fehler)
- Flußkontrolle

Protokolle der Ebene 2 befassen sich nur mit dem Informationsaustausch zwischen

- unmittelbar benachbarten Knoten/Stationen, die durch eine
- direkte Leitung verbunden sind.

Die von der Sicherungsebene angebotenen Dienste sind der Transportschicht sehr ähnlich. Der wesentliche Unterschied besteht darin, daß die Sicherungsebene die Kommunikation zwischen zwei benachbarten Knoten regelt, während die Trans-

6. Daten (Sende-)Anfrage/Wunsch

7. Daten Empfangsbestätigung



portebene eine Ende-zu-Ende-Verbindung zwischen Sender und Empfänger steuert.

## 4.1 Fehlererkennung/-behebung

Es gibt zwei Möglichkeiten der Fehlerbehebung:

- Fehlerbehebung durch Wiederholung (z.B. in HDLC, LLC, LAPB): Der Empfänger erkennt Fehler, fordert Wiederholung.
- Forward Error Correction (FEC): Nur eine Teilmenge der möglichen Signalfolgen ist zulässig. Andere werden auf die wahrscheinlichste Codefolge korrigiert.

Die sogenannte **Hammingdistanz** zweier Codeworte<sup>8</sup> ist die Anzahl der Bitstellen, an der sie sich unterscheiden. Die Hammingdistanz eines ganzen Codes ist die das Minimum über alle (paarweise verschiedenen) Codewörter, also die geringste Anzahl zu ändernder Bits, um irgendein Codewort in ein anderes zu verwandeln.

Bei einfachem Parity-Check wird jedes Codewort um ein weiteres Bit ergänzt, so daß die Anzahl der Einsen gerade ist. Um hier von einem Wort zum anderen zu kommen muß man also zwei Bits kippen.

Bei gegebener Hammingdistanz  $d$  eines Codes werden alle Fehler, die weniger als  $d$  Bits umfassen erkannt. Bei größeren Fehlern kann ein anderes –scheinbar– korrektes Wort entstehen. Alle Fehler, die weniger als  $d/2$  Bits betreffen, können durch Decodierung zum „nächsten“ Codewort korrigiert werden. Größere Fehler werden evtl. falsch korrigiert.

Probleme beim Forward Error Control:

- FEC reduziert Nutzdatenrate auch bei Fehlerfreiheit, da die Kontrollbits immer gesendet werden.
- Die Korrektur kann trotzdem, s.o., falsch sein bei großen Fehlern.

Daher wird FEC nur in speziellen Fällen eingesetzt, wo eine erneute Sendung nicht möglich (Simplex-Kanal) ist oder zu lange dauern würde (Realzeitanforderungen, Weltraumsatelliten).

---

8. Ein Codewort besteht aus Nachrichtenbits, den eigentlichen Daten, und Prüfbits als Kontrollezusatz



## 4.1.1 Fehlererkennende Codes

Einfachstes Beispiel ist der einfache Parity-Check. Allerdings wird eine zufällig empfangene Bitfolge mit Wahrscheinlichkeit 0,5 als korrekt akzeptiert, was natürlich indiskutabel ist.

### 4.1.1.1 Blockweise Sicherung

Es stellt eine deutliche Verbesserung dar, wenn jeder gesendete Block als eine rechteckige Matrix,  $n$  Bits breit und  $k$  Bits hoch, betrachtet wird. Für jede Spalte wird ein eigenes Paritätsbit errechnet und in der letzten Reihe der Matrix angehängt. Die Matrix wird dann Zeile für Zeile übertragen. Sobald ein Block ankommt, prüft der Empfänger alle Paritätsbits. Ist ein Bit falsch, bittet er um Sendewiederholung.

Mit dieser Methode können einzelne Fehlerbündel der Länge  $n$  gefunden werden, da nur ein Bit pro Spalte verändert wird. Ein Fehlerbündel der Länge  $n+1$  kommt allerdings unentdeckt davon, wenn das erste und das letzte Bit invertiert werden und alle anderen unverändert bleiben (ein Fehlerbündel bedeutet nicht automatisch, daß alle Bits falsch sind, sondern nur, daß zumindest das erste und das letzte Bit fehlerhaft sind). Wird ein Block durch eine lange oder viele kurze Fehlerbündel verstümmelt, so ist die Wahrscheinlichkeit, daß eine der  $n$  Spalten zufällig die richtige Parität hat, 0,5. Folglich ist die Wahrscheinlichkeit, daß ein fehlerhafter Block dennoch angenommen wird,  $2^{-n}$ . Obwohl dieses Verfahren durchaus praktikabel ist, wird in der Praxis häufig ein anderes Modell benutzt:

### 4.1.1.2 Polynomcode, CRC (Cyclic Redundancy Check)

Viele Fehler-Erkennungs-codes für Datenübertragungen benutzen eine Logik, die auf dem CRC basiert. Er wird so genannt, weil die Bits in einer Nachricht zyklisch, um jeweils eine Stelle durch ein Register geschiftet werden. Die CRC-Technik benötigt keine Speicher und kann in kombinatorischer Logik implementiert werden. Zyklischer Code besitzt algebraische Eigenschaften, die ihm relativ einfache Fehlererkennungs-Implementationen verleihen.

Mit dieser Technik generiert der Sender ein Bitmuster (Frame Check Sequence), basierend auf dem Inhalt des Frame.<sup>9</sup> Der konkatenierte Inhalt des Frame und des

---

9. Hänge Nullen an Frame soviel wie Bits in FCS. Dividiere das durch das Generatorpolynom, was ein Bit länger ist als FCS. Der Rest der Division wird das FCS-Feld.



FCS ist exakt durch das Generatorpolynom teilbar. Falls der Inhalt fehlerhaft wird durch die Übertragung, liefert die Division des Empfängers durch das Generatorpolynom einen Rest ungleich 0.

Nochmals: Der zu sendende Frame wird vom Sender um eine Prüfsumme verlängert dergestalt, daß das gesamte Ding durch das Generatorpolynom teilbar ist. Der Empfänger checkt dann die Teilbarkeit nochmals.

Ein Polynomcode mit  $r$  Prüfbits erkennt alle Fehlerbündel mit maximaler Länge von  $r$ . Eine 16-Bit-Prüfsumme wie CRC-16 oder CRC-CCITT entdeckt alle Einzel- und Doppelfehler, alle Fehler mit ungerader Bitzahl, alle Fehlerbündel<sup>10</sup> mit 16 oder weniger Bits, 99,997% aller 17-Bit-Fehlerbündel und 99,998% aller Fehlerbündel mit 18 oder mehr Bits.

## 4.2 Fehlerbehebung durch Wiederholung

Fast alle Protokolle der Ebene 2 korrigieren Fehler durch Wiederholung. Dieser Ansatz basiert auf

- Quittungen (ACK=positiv, NAK=negativ) und
- Timer-Mechanismen

Beachte: Protokolle der Schicht 2 arbeiten nur zwischen direkten Nachbarn und nicht End-End!

Es gibt keine (vollständige) sichere Übertragung über unsichere Kanäle. Quittung der Quittung der Quittung...

## 4.3 Flußkontrolle

Anpassung von Sende- und Empfangsgeschwindigkeit (bei langsamem Empfänger, Verfälschung von Nachrichten, Verlust ganzer Nachrichten).

Einfachstes Verfahren: **Stop-and-Wait**. D.h. warte nach jeder Nachricht auf die Quittung.

---

10. Fehlerbursts



Beachte: Quittungsmechanismen (Senden von expliziten Quittungen oder „Piggybacking“<sup>11</sup>, d.h. Quittung mit Daten in Gegenrichtung) setzen die Existenz eines „Rückkanals“ voraus!

Besser: **Schiebe-Fenster-technik** (Sliding-Window). Dabei vereinbaren die Stationen eine Fenstergröße  $W$ , die mindestens 1 ist und kleiner als  $m$ , mit  $m (= 2^n)$  als Modulus. Nachrichten werden modulo  $m$  nummeriert in einem  $n$ -bit Feld. Sonst kommen identische Rahmennummern für unterschiedliche Rahmen vor.

Sendefenster:

- gibt an, welche (bisher unbestätigten) Frames gesendet werden dürfen.
- wird jeweils nach Senden eines Frames verkleinert.
- wird jeweils nach Empfang einer Quittung vergrößert.

Empfangsfenster:

- gibt an, welche Frames der Empfänger akzeptiert.
- wird jeweils nach Senden einer Quittung verschoben.

Bei Piggybacking kann das kurze Warten auf einen zu schickenden Rahmen den Durchsatz vermindern. Noch längeres Warten veranlaßt die Gegenstelle zu erneutem Senden desselben Rahmens, da sie keine Quittung erhält.

Bei Empfangsfenstern, die mehr als einen Rahmen Kapazität haben, werden auch Rahmen angenommen, die nicht der richtigen Reihenfolge entsprechen.

Wenn die Fenstergröße über 1 liegt, dann kann der Sender einige Rahmen schicken, ohne eine Quittung erhalten zu haben. Dieses Pipelining hat den Vorteil, daß der Kanal besser ausgelastet wird, da neue Rahmen geschickt werden, eh die erste Quittung eintrifft. Was passiert aber, wenn ein Rahmen inmitten eines langen Datenflusses verloren geht? Sehr viele nachfolgende Rahmen kommen beim Empfänger an, bevor der Sender herausfindet, daß etwas schief gegangen ist. Sobald ein zerstörter Rahmen beim Empfänger ankommt, sollte er logischerweise verworfen werden; aber was soll der Empfänger mit all den darauffolgenden Rahmen tun? Die Pakete müssen nämlich in einer ganz bestimmten Reihenfolge an die nächsthöhere Schicht weitergereicht werden.

---

11. Huckepack





**Lösungsansätze:**

1. Gehe n zurück:  
Alle Rahmen ab dem gestörten nochmals senden. Der Empfänger verwirft alle nachfolgenden Rahmen und schickt keine Rückmeldungen. Entspricht Empfänger mit Fenster der Größe 1. Die Empfängersicherungsschicht nimmt nur den Rahmen an, den sie als nächsten an die Vermittlungsschicht weitergeben muß. Bei hoher Fehlerquote wird durch die unnütz gesendeten Rahmen viel Kanalkapazität verschwendet. Allerdings einfache Implementierung.
2. Selektive Wiederholung:  
Hier speichert die empfangende Sicherungsschicht alle korrekten Rahmen, die auf den fehlerhaften folgen. Ab dem Fehler wird nicht mehr bestätigt. Wenn der Sender endlich bemerkt, daß etwas schiefgegangen ist, schickt er die nicht bestätigten Rahmen erneut. Der Empfänger quittiert alle Rahmen durch die höchste der empfangenen Rahmen-Nummern, sobald der fehlerhafte ankommt. So werden etwas weniger Rahmen wiederholt.
3. Selektive Rückweisung:  
Der Empfänger bestätigt alle korrekt empfangenen Rahmen. Der Sender schickt nur den unbestätigten erneut.

Abwägung zwischen Bandbreitenausnutzung und Pufferspeicher der Sicherungsschicht.

Bei Protokollen mit Fenster größer als eins ist es besser, die Fenstergröße auf die Hälfte der Anzahl der möglichen Folgenummern zu beschränken. Denn bei Fehlern wird der Sender einige Rahmen wiederholen und die neue Reihe der gültigen Folgenummern darf sich nicht mit der schon dagewesenen überschneiden, nachdem der Empfänger sein Fenster weitergeschoben hat. Denn wenn die Rückmeldungen verloren gegangen sind, wüßte der Empfänger nicht, ob da nur Duplikate kommen, oder neue Rahmen.

**Leistungsfähigkeit von Flußkontrollmechanismen:**

Interessant ist die optimale Länge des Datenfeldes innerhalb eines Rahmens bei möglichen Übertragungsfehlern und Stop & Wait:

$$= \sqrt{\frac{\text{Rahmenkopflänge} + \text{Kanalkapazität} \cdot \text{Timeout-Intervall}}{\text{Wahrscheinlichkeit für ein Bit mit Fehler}}}$$

Bei zunehmender Kabellänge fällt die Leistung des Stop & Wait-Protokolls schnell ab, weil die meiste Zeit damit verbracht wird, auf Rückmeldungen zu warten. Bei



Fenstergrößen über 1 fällt die Kanalauslastung auch mit der Kabellänge, allerdings nicht so steil.

Die Kanalauslastung verbessert sich, wenn die Fenstergröße erhöht wird. Bei jeder beliebigen Fenstergröße ist die Leistung bei langen Kabeln schlechter als bei kurzen.

Ist das Fenster mindestens um eins größer als die Anzahl der Rahmen, die in das Kabel passen, kann die Übertragung ununterbrochen vor sich gehen. Der Sender muß dann nicht aufgrund eines vollen Fensters blockieren, wenn das Fenster dann noch nicht voll ist, wenn der erste Rahmen vollständig beim Empfänger angekommen ist, ganz aufgenommen wurde und die Rückmeldung vollständig bis zum Sender zurückgekommen ist. Das Kabel kann dann in beiden Richtungen gefüllt sein und der Sender hat noch eine Fensterposition mindestens frei.

## 4.4 Protokolle der Sicherungsebene

Die Synchronisation auf Sicherungsebene betrifft die Unterscheidung von Nutzdaten und Kontrollzeichen.

### 4.4.1 Zeichengesteuerte Protokolle

In den 60er Jahren wurden zeichengesteuerte Protokolle der Ebene 2 entwickelt, die auch heute noch in einigen Bereichen eingesetzt werden.

Charakteristika zeichengesteuerter Protokolle:

- Abhängigkeit von einem speziellen Code
  - Steuerzeichen sind Bestandteil des Codes, z.B. enthält ASCII die Zeichen NAK, ACK, STX, ETX
- Einpacken von zufällig auftretenden (scheinbaren) Steuerzeichen innerhalb der Nutzdaten

### 4.4.2 Längengesteuerte Protokolle

In den 70er Jahren wurden Protokolle entwickelt, die vor der Übertragung die Länge der nachfolgenden Nutzdaten angeben.

Folge: Zufällig auftretende (scheinbare) Steuerzeichen innerhalb der Nutzdaten werden irrelevant.



### 4.4.3 Bit-gesteuerte Protokolle

Diese ebenfalls in den 70er Jahren entwickelten Protokolle bilden die Basis fast aller modernen Protokolle der Ebene 2.

Charakteristika:

- Nachrichtenübertragung in Rahmen variabler Länge.
- Markierung von Anfang und Ende eines Frames durch spezielle Bitfolgen „Flags“.
- Übertragung der Protokollinformation (PCI) durch einzelne Bits an spezielle Positionen

## 5. Lokale Netze

Charakteristika von lokalen Netzen:

- begrenzte geographische Ausdehnung, ca. 100 m bis 10 km, typisch ca. 1 km
- kurze, aber nicht vernachlässigbare Signallaufzeit
- hohe Datenrate, einige Mbit/s
- geringe Fehlerrate, ( $<10^{-9}$ )
- serielle Übertragung
- privater Besitz, keine Einschränkung durch die Post

### 5.1 Zugangsverfahren

Problem: Mehrere Stationen nutzen ein Medium, daher Regelung des Zugangs erforderlich. Man verwendet ein Medienzugangsprotokoll MAC („Media Access Control“). Dafür unterteilt man Schicht 2 in:

- Ebene 2b: LLC (Logical Link Control)
- Ebene 2a: MAC

Diese Aufteilung ermöglicht eine gemeinsame, vom Medienzugangsverfahren des LANs unabhängige Schnittstelle zur Netzwerkebene.



Beispiele:

- Festes TDMA (Time Division Multiple Access):  
Vergabe fester Zeitscheiben an die Benutzer; exklusive Nutzung für begrenzte Zeit.
- Festes FDMA (Frequency Division Multiple Access):  
Vergabe fester Frequenzbereiche an die Benutzer.
- Zentrales Polling:  
Zentrale fragt ab, ob Sendewunsch besteht.
- Dezentrales Polling (Tokenprinzip):  
Monitorstation statt Zentrale. Diese generiert oder regeneriert Sendeerlaubnis (Token), das die Stationen reihum weitergeben.
- Reservierungsverfahren:  
Stationen reservieren zunächst Zeitscheiben.
- Ungesteuerte Verfahren (Random Access):
  - ALOHA (Völlig ungesteuert):  
Nachrichten können (auch teilweise) durch Überlagerung zerstört werden. Korrektur durch Wiederholung der Sendung.
  - Abhören des Kanals zur Konfliktvermeidung:
    - (a) CSMA (Carrier Sense Multiple Access):  
Zuerst hören, dann senden.
    - (b) CSMA/CD (.../Collision Detection)

Es gibt drei große Klassen von MAC-Protokollen:

1. Auswahlverfahren  
Erteilung der Sendeberechtigung unmittelbar vor Sendebeginn (spezielles Signal)
2. Random Access  
Jederzeit Zugriff zum Übertragungsmedium (einige Einschränkungen)
3. Reservierungstechniken  
Zuteilung von Senderechten für bestimmte Zeitintervalle. Nach Abschluß einer Reservierungsphase kann jede Station selbst berechnen, wann sie senden darf.



## 5.1.1 Der Token-Ring

### Punkt-zu-Punkt-Verbindungen bilden Ring:

- „aktiver“ Anschluß, (d.h. Signale werden regeneriert).
- gerichtete Übertragung von Nachrichten, also uni-direktional.
- i.a. kein „store-and-forward“-Betrieb<sup>12</sup>, bei dem jedes Paket komplett empfangen und getestet wird, bevor es weitergeleitet wird. Dabei muß entschieden werden, an welchen Nachbarn. Der Token Ring leitet bitweise weiter und jeder Knoten hat nur einen Nachfolger, außerdem genügt eine End-to-End-Fehlerkontrolle, da lokale Kanäle recht fehlerfrei sind.

### Vorteile von Ring-Netzen:

- kein Routing, Multicast und Broadcast unproblematisch.
- (fast) keine Einschränkung der geographischen Distanz.

### Nachteile von Ring-Netzen:

- Paketverzögerung durch Stationslatenz.
- Ein defekter Controller bewirkt Netzzusammenbruch.
- Abschalten des Rings erforderlich bei Hinzufügen oder Entfernen von Stationen, allerdings nicht beim Ein- und Ausschalten von Stationen.

### Token-Passing-Medienzugangsprotokoll:

In einem **Token-Ring (IEEE-Norm 802.5)** kreist ein spezielles Bitmuster, genannt das Token, um den Ring, wenn alle Stationen untätig sind. Falls eine Station einen Rahmen übertragen will, muß sie zuerst das Token erlangen und aus dem Ring entfernen (Single Token), bevor sie senden darf. Da es nur ein Token gibt, kann immer nur eine Station übertragen, wodurch das Problem des Kanalzugriffs auf die gleiche Weise wie beim Token-Bus gelöst wird.

Das Token enthält vier Felder: Ein Bit T zeigt besetzt oder nicht an, das Monitorbit dient zur Identifizierung von herrenlosen Rahmen, in die Reservationsbits kann eine sendewillige Station die Priorität des nächsten von ihr zu sendenden Rahmens eintragen, die Prioritybits erlauben eine Priorisierung von Rahmen. Diese Accesscontrolbits enthält auch ein Informationspaket, außer T.

---

12. Wird hauptsächlich in WANs benutzt, die unstrukturierte Topologie haben



Der 3-Byte-Token-Rahmen enthält ein Feld im mittleren Byte, das die **Priorität** des Tokens angibt. Will eine Station einen Rahmen der Priorität  $n$  übermitteln, so muß sie warten, bis sie ein Token mit einer Priorität kleiner oder gleich  $n$  erlangt. Weiterhin kann eine Station versuchen, sich das nächste Token zu reservieren, indem sie in die **Reservierungsbits** eines passierenden Datenrahmens die Priorität des abzuschickenden Rahmens einträgt. Wenn allerdings dort bereits eine höhere Priorität vorgemerkt ist, kann die Station keine Reservierung durchführen. Nach Beendigung des aktuellen Token wird der nächste Token mit der vorgemerkten Priorität erstellt.

Mit etwas Überlegung kommt man dahinter, daß dieser Vorgang wie eine Ratsche funktioniert, weil er die Reservierungspriorität immer höher schraubt. Das Protokoll enthält mehrere komplexe Regeln, um dieses Problem zu eliminieren. Der Grundgedanke ist, daß eine Station, welche die Reservierungspriorität erhöht, dafür verantwortlich ist, die Priorität nach Erledigung auch wieder herabzusetzen.

Beachte dabei, daß sich dieses Prioritätsschema grundsätzlich vom Token-Bus-Schema unterscheidet, wo jede Station immer einen gerechten Anteil der Bandbreite bekommt, unabhängig davon, was bei anderen Stationen im Moment geschieht. Im Token-Ring kann eine Station verhungern, wenn sie nur Rahmen mit niedriger Priorität abzusenden hat und warten muß, bis ein Token mit niedriger Priorität auftaucht. Anscheinend hatten die beiden Komitees verschiedenen Geschmack, als es darum ging, guten Service für Datenverkehr hoher Priorität gegenüber Gleichbehandlung aller Stationen abzuwägen.

### **Aktiver Anschluß beim Token Ring:**

Grundzustand:

- Daten vom Ring werden seriell empfangen.
- An die Station selbst gerichtete Pakete werden kopiert.
- Daten werden seriell gesendet.

Innerhalb jeder Station entsteht dabei eine gewisse Latenz.

Sendezustand:

- Ring wird aufgetrennt.
- Eigene Daten werden seriell gesendet.
- Vom Ring kommende Daten werden nicht weitergeleitet, sondern für Fehlerkorrektur (Vergleich mit gesendeten Daten z.B.) ausgewertet.



Auch hier tritt eine gewisse Latenz auf.

Beim Token Ring wird das Recht zum Übergang in den Sendezustand durch ein „Token“ (spezielles Bit bzw. Bitfolge) geregelt.

Nach Empfang des freien Tokens gibt es für sendewillige Stationen zwei Möglichkeiten:

1. Exhaustive Service  
Die Station sendet alle sendebereiten Daten.
2. Nonexhaustive Service  
Nach Tokenempfang darf nur ein bestimmter Teil der sendebereiten Daten übertragen werden.

Sendet keine Station Daten, dann ist die Rotationszeit des freien Tokens gleich der Ringlatenz, d.h. Summe aus Signallaufzeit und Stationslatenzen.

Wenn eine Station Daten übertragen will, dann muß sie

- auf den Empfang des freien Tokens warten
- das freie in ein belegtes Token modifizieren
- Daten übertragen
- die gesendeten Daten vom Ring entfernen
- ein neues freies Token generieren

Je nach dem Zeitpunkt der Generierung des neuen Tokens unterscheidet man:

- Multiple Token Ring
- Single Token Ring
- Single Frame Ring

#### **5.1.1.1 Multiple Token Ring (Verwendung bei FDDI und 802.4)**

- Generierung eines neuen Tokens unmittelbar nach Übertragung des letzten Bits der gesendeten Nachricht.
- Es können gleichzeitig mehrere Token auf dem Ring sein, aber maximal ein freies Token.



### 5.1.1.2 Single Token Ring

Generierung eines neuen Tokens frühestens nach Empfang des eigenen belegten Tokens

- Paket länger als Ring (in bit)  
Verhalten wie Multiple Token Ring
- Paket kürzer als Ring  
zusätzliche Wartezeit bis Empfang des busy Token

### 5.1.1.3 Single Frame Ring (Verwendung bei 802.5)

Generierung des neuen Tokens erst nach Empfang des gesamten eigenen Pakets (belegtes Token und gesendete Daten). Es können folglich nicht mehrere Pakete auf dem Ring sein.

Bei kurzen Paketen und großer Ringlatenz sind die Unterschiede der Leistungsfähigkeit erheblich! Daher ist für Hochgeschwindigkeitsnetze auf Token-Basis nur der Multiple Token Ring sinnvoll.

### Fehler und Fehlerkorrektur beim Token Ring

- Störung, Ausfall einer Leitung oder Station:  
Hardwareüberbrückung durch Verkabelungszentrum mit Umgehungsrelais.
- Verlorenes Token:  
Monitorstation kontrolliert Zeit zwischen vorbeilaufenden Token; säubert ggf. Ring und generiert ein freies Token.
- Endlos kreisendes Token:  
Monitor-Bit im Token-Feld; der Monitor säubert ggf. den Ring und generiert ein freies Token.
- Mehrfaches Token:  
Sender überprüft Quelladresse bei eintreffenden Paketen; falls nicht die eigene Adresse, dann Abbruch der Übertragung, verlorenes Token.
- Monitorausfall:  
Stand-by-Monitor. Stationen erwarten freies Token spätestens nach maximaler Sendezeit \* Anzahl der Stationen-1 + Ringlatenz. Wird mit Timer kontrolliert. Danach bestimmen die Stationen einen neuen Monitor.

## 5.1.2 Token Bus (IEEE-Norm 802.4)

Logischer Ring auf physikalischem Bus. Das freie Token kreist zwischen den Stationen in einem logischen Ring.





**Probleme und Lösungen:**

- Freiwilliges Abschalten einer Station:  
Sage Vorgänger die Adresse des neuen Nachfolgers.
- Unfreiwilliges Abschalten einer Station:  
Kontrolliere nach Weitergabe des Tokens, ob Nachfolger wirklich sendet.  
Falls nicht sende „who follows“ und entferne den bisherigen Nachfolger.
- Zuschalten neuer Stationen:  
Öffne in periodischen Abständen ein „Fenster“ Response Window, das Zuschaltung neuer Stationen ermöglicht.  
Bei Konflikten wird die logische Anordnung ausgehend von der MAC-Adresse festgelegt (unterschiedliche Wartezeit bis neuer Versuch).
- Verlust des Tokens:  
Sende „Claim Token Frame“  
Konfliktbereinigung wie bei Zuschaltung.

Bei der Initialisierung des Rings werden die Stationen in der Reihenfolge ihrer Stationsadressen eingefügt, von der höchsten zu niedrigsten. Die Übergabe des Tokens erfolgt ebenfalls von oben nach unten. Jedesmal, wenn eine Station das Token erhält, darf sie für eine bestimmte Zeitspanne Rahmen übertragen; dann muß sie das Token weitergeben. Falls die Rahmen klein genug sind, können mehrere hintereinander gesendet werden. Hat die Station noch keine Daten, wird das Token sofort nach Erhalt weitergegeben. Der Token Bus führt die Prioritätsstufen 0, 2, 4 und 6 für den Datenverkehr ein, wobei 0 die niedrigste und 6 die höchste Stufe ist. Am einfachsten kann man sich vorstellen, daß jede Station intern in vier Unterstationen entsprechend den vier Prioritätsstufen aufgeteilt wird. Wenn die MAC-Teilschicht Daten schickt, werden sie auf ihre Priorität überprüft und zu einer der vier Unterstationen weitergeleitet. Dadurch enthält jede Unterstation ihre eigene Warteschlange von zu übertragenden Rahmen.

Wenn eine Station über das Kabel das Token erhält, wird es intern an die Unterstation der Priorität 6 weitergeleitet, die nun Rahmen übertragen kann, wenn welche vorhanden sind. Wenn sie fertig ist (oder der Timer abläuft), wird das Token intern an die Unterstation der Priorität 4 weitergegeben, die nun ihrerseits Rahmen übertragen kann, bis der Timer abläuft, worauf das Token intern der Unterstation 2 übergeben wird. Dieser Vorgang wird wiederholt, bis entweder die Unterstation der Priorität 0 alle eigenen Rahmen übertragen hat oder der Timer abgelaufen ist. Jetzt wird das Token in jedem Fall an die nächste Station im Ring weitergeleitet.

Ohne genau wissen zu müssen, wie die verschiedenen Timer arbeiten, sollte klar sein, daß durch genaues Setzen der Timer ein Mindestanteil der gesamten Token-



zeit für Datenverkehr der Priorität 6 garantiert werden kann. Die niedrigeren Prioritäten müssen von dem leben, was übrigbleibt. Wenn die Unterstationen mit den höheren Prioritäten ihren Anteil der verfügbaren Zeit nicht brauchen, wird er nicht verschwendet, sondern kann durch die Stationen der niedrigeren Priorität verwendet werden.

Dieses Prioritätsschema, das dem Datenverkehr der Priorität 6 einen festen Anteil der Netzwerkbandbreite garantiert, kann für die Übertragung von Sprache und anderen Echtzeit-Datenverkehr benutzt werden.

FDDI nutzt auch diesen Prioritätsmechanismus. *FDDI und Token Bus verwenden beide das Multiple-Token-Prinzip*, da das neue Token direkt nach der eigenen Sendung generiert wird.

### 5.1.3 CSMA/CD

#### Carrier Sense

Eine sendewillige Station prüft zunächst, ob der Kanal frei oder belegt ist. Ist er frei, werden die Daten sofort übertragen.

#### Multiple Access

Alle Stationen sind über einen Broadcast-Kanal verbunden. Mehrere Stationen können (quasi)-gleichzeitig Daten auf diesen Kanal geben.

Wenn der Kanal frei ist, wird sofort mit der Übertragung begonnen. Ist der Kanal zunächst belegt, so wird sofort nach dem Freiwerden gesendet. Dieses Verfahren wird als 1-persistent<sup>13</sup> bezeichnet. Alternativen hierzu sind p-persistent (es wird mit bestimmten Wahrscheinlichkeiten in verschiedenen Zeitintervallen nach dem Freiwerden gesendet) und non-persistent (es wird nach einer zufälligen Zeit gesendet).

#### Collision Detection

Die sendende Station hört den Kanal während der Übertragung ab. Entdeckt sie eine Kollision, stoppt sie die Übertragung sofort.

---

13. ausdauernd, hartnäckig



Wird eine Kollision erkannt, so wartet jede Station eine zufällige Zeit, bis eine Übertragungswiederholung versucht wird. Zur Berechnung der zufälligen Wartezeit wird die Zeitachse in Slots der Länge 512 Bitzeiten eingeteilt, gemäß Standard entspricht dies der maximal möglichen Übertragungsverzögerung auf dem Kanal<sup>14</sup>. Nach der ersten Kollision wartet jede Station 0 oder 1 Slotzeit, nach der zweiten 0, 1, 2 oder 3 Slotzeiten, bis  $n=9$  liegt die Zufallszahl nach der  $n$ -ten Kollision zwischen 0 und  $2^n - 1$ , ab  $n=10$  wird die Wartezeit nicht mehr vergrößert. Nach 16 Kollisionen wird das Verfahren abgebrochen und ein Fehler gemeldet. Dieses Verfahren wird als truncated binary exponential backoff bezeichnet

Jeder übertragene Rahmen enthält ein sog. Pad-Feld, welches bei extrem kurzen Nachrichten benötigt wird, um auf die vom Standard verlangte Mindestrahmenlänge von 64 Bytes zu kommen. Diese Mindestlänge soll die Unterscheidung zwischen gültigen Rahmen und Rahmenresten, die bei entdeckten Kollisionen entstanden sind, vereinfachen. Darüberhinaus ist eine solche Mindestgröße erforderlich, um zu verhindern, daß ein kurzer Rahmen vollständig gesendet werden kann und an einem weit entfernten Ende des Kanals mit einem anderen Rahmen kollidiert.

## 5.2 Leistungsfähigkeit von LAN

Die Beurteilung der Leistungsfähigkeit von LANs erfordert die Definition einer Metrik.

### **Betreibersicht:**

Ein LAN ist genau dann leistungsfähig, wenn es sich ohne Einwirkung des Betreibers optimal den aktuellen Anforderungen anpaßt.

### **Anwendersicht:**

Ein LAN ist genau dann leistungsfähig, wenn es den Anforderungen des Benutzers gerecht wird.

Benutzer kann der Mensch vor dem Rechner sein, aber auch ein bestimmter Prozeß auf dem Rechner. Offenbar kann die Leistungsfähigkeit von LANs nur bei vorgegebener Anwendung beurteilt werden.

---

14. maximale doppelte End-End-Laufzeit



Andererseits können einzelne Anwendungsgebiete zu verschiedenen Zeiten unterschiedlichste Anforderungen haben (z.B. Banken). Es gibt keine klaren Trennungen zwischen Anwendungsgebieten, sondern erhebliche Überlappungen. Anforderungen konkreter Anwendungen sind häufig schwer quantifizierbar und einer Evolution unterworfen.

Der heutige Einsatz von Rechnernetzen ist im wesentlichen durch drei Paradigmen charakterisiert:

- Großrechner
- Terminal-Timesharing-Systeme
- Transaktionsbasierte Systeme

Stark an Bedeutung gewinnen z.Z. die

- Client-Server-Systeme

## **5.2.1 Anforderungen an Rechnernetze**

### **5.2.1.1 Großrechner**

Großrechner verfügen i.d.R. über direkt angeschlossene Peripherie. Ein Netzzugriff erfolgt daher nur

- für Transfer großer Datenmengen (Files) von kbyte an aufwärts
- über ggf. große Entfernungen
- mit schwachen Forderungen bzgl. Antwortzeit, Sekunden bei kleinen Files bis mehrere Minuten bei großen Files.

Aufgrund der ohnehin langen Transferzeiten sind Verbindungsaufbau und -abbau unproblematisch.

Während des Bestehens einer Verbindung erfolgt der Informationstransfer i.w. gleichmäßig.

### **5.2.1.2 Terminal-Timesharing-Systeme**

Zeichenorientierte Terminals (analog: „remote login“) greifen auf einen zentralen (Groß?-) Rechner zu.



Typische Datenraten:

- 2000 bit/s im Burst, gemittelt über eine Sekunde
- 200 bit/s gemittelt über eine Minute
- 10 bit/s gemittelt über eine Stunde

Folglich genügt ein 10 oder 20 kbit/s Leitung.

I.d.R. ist der Verkehr asymmetrisch, d.h. der Rechner liefert erheblich mehr (z.B. Faktor 10) als das Terminal.

Die Verbindung zwischen Rechner und Terminal besteht dauerhaft.

Der Datentransfer unterliegt starken Schwankungen (Bursts mit ca. 100-facher Last im Vergleich zum Mittelwert)

### **5.2.1.3 Transaktionsbasierte Systeme**

Dieses ebenfalls klassische Gebiet des Einsatzes von Rechnernetzen ermöglicht die Kommunikation zwischen

- (Groß?-) Rechnern, welche die Berechnungen durchführen und
- Endgeräten, welche dem Benutzer intelligente Schnittstellen zur Verfügung stellen und selbst verwalten.

Der Netzzugriff erfolgt hier für

- regelmäßigen Austausch von aktuellen Informationen, Update
- benutzergesteuerte Aktualisierung

Der Verkehr ist i.w. symmetrisch. Die Anforderungen sind ähnlich den Time-Sharing-Systemen (ca. 10 bis 20 kbit/s Leitungen reichen aus.

### **Entwicklungen bei PCs und Workstations:**

Exponentielles Wachstum verzeichnen:

- Hauptspeicherkapazität
- Prozessorgeschwindigkeit
- Plattenkapazität

Logarithmische Beschleunigung bei

- Zugriff auf Platten



### 5.2.1.4 Client-Server Systeme

Leistungsfähige Workstations und leistungsfähige LANs erlauben neuartige Client-Server Systeme.

#### **Idee:**

Führe so viele Berechnungen wie möglich am Arbeitsplatz durch.

Greife für spezielle Aufgaben auf zentrale Server zu: File-, Print-, Compute-Server und zentrale Datenbank.

Die Kommunikation zwischen Client und Server erfolgt i.a. durch „Remote Procedure Calls“ (RPCs):

- innerhalb der Maschine „normale“ Prozeßkommunikation mit einem Stellvertreter (Stub Procedure)
- Kommunikation zwischen den Maschinen durch Request/Response-Mechanismen.

#### **Folge:**

Nur die Stubs müssen netzspezifische Charakteristika berücksichtigen.

Bei Einsatz des Client-Server Modells muß ein Endsystem (End User System, EUS) schnell mit anderen EUS kommunizieren können.

#### **Beachte:**

- Beziehungen zwischen EUS's unterliegen einem ständigen Wandel.
- Ein EUS kann mit mehreren EUS's gleichzeitig kommunizieren.
- Multicasting kann große Bedeutung haben.

#### **Client-Server System fordern:**

- sehr kurze Antwortzeit (Zugriff über das Netz=lokaler Zugriff)
  - Transfer von Hauptspeicherinhalt: 10  $\mu$ s
  - Aufruf von Prozessen (RPC): 100  $\mu$ s
  - Demand Paging: 10 ms
- schnellen Transfer bei Lastspitzen
  - Raten bis ca. 100 Mbit/s
  - Bursts mit 1000-facher Last im Vergleich zum Mittelwert
- einfache Protokolle



- klassische verbindungsorientierte Protokolle mit Flußkontrolle und Fehlererkennung i.d.R. zu langsam
- häufig: „roll your own“-Protokolle statt Standardprotokolle
- sehr geringe Fehlerraten
  - Fehlerrate wie bei direkt angeschlossener Peripherie
- verbindungslose Netze
  - kein Schalten von Verbindungen im Netz (vgl. Ebene 3)
- Multicast-Kommunikation
  - einer an viele
  - bei „freundlichen“ Benutzern durch „Broadcast“ ersetzbar
- geringe Sensitivität bzgl. der Paketlänge
  - keine Probleme bei Nachrichten unterschiedlichster Länge
- Mechanismen zum Schutz sensibler Daten
  - Übertragung von Informationen, die „normalerweise“ den Rechner nicht verläßt
- schnelle Erreichbarkeit aller Stationen
  - Das Netz muß auch die entfernteste Station erreichen und dabei Kabelkanälen folgen
- große Anzahl anschließbarer Stationen
  - Das Netz soll mit dem Unternehmen wachsen

Konventionelle LANs erfüllen i.w. alle Forderungen von Client-Server Systemen.

### **Wichtige Ausnahme:**

Die Datenrate (klassisch 10 Mbit/s) wird zu gering. Dieser Effekt verschärft sich bei Einsatz von Graphik.

## **5.2.2 Analysetechniken**

1. Exakte Analyse
  - + exakte Resultate für das analysierte Modell
  - Einschränkungen für Lastmodell
  - Einschränkungen bei der Modellierung technischer Systeme
2. Approximation
  - + geringer Bedarf an Speicherplatz und Rechenzeit
  - Güte der Ergebnisse oftmals schwer abschätzbar
3. Simulation
  - + prinzipiell universell einsetzbar
  - ggf. hoher Bedarf an Rechenzeit und/oder Speicherplatz
  - Problematik der Überprüfung von Simulationsergebnissen



### 5.2.3 Der Lokalitatsfaktor

LANs zeichnen sich durch kurze aber nicht vernachlassigbare End-End-Laufzeiten aus. Fur die Beurteilung der Leistungsfahigkeit von MAC-Protokollen ist jedoch nicht die absolute Laufzeit, sondern die Laufzeit relativ zur Paketsendezeit von Bedeutung.

Daher wurde  $a = \frac{\text{End-End-Laufzeit}}{\text{mittlere Paketsendezeit}} = \frac{d}{N} \frac{R}{V_s}$  als der Lokalitatsfaktor definiert mit  $d$  maximale Distanz in m,  $V_s$  Signalgeschwindigkeit in m/s,  $R$  Nominelle Datenrate in bit/s,  $N$  Mittlere Nachrichtenlange in bit.

Anschaulich:  $a$  gibt an, wieviele mittlere Pakete hintereinander auf die Leitung passen. Bei Bussystemen gibt  $a$  die relative Laufzeit zwischen den am weitesten entfernten Stationen an. Buslange in Anzahl der Pakete. Bei Ring-Systemen beschreibt  $a$  die relative Ringlatenz. Ringumfang in Anzahl der Pakete.

### 5.2.4 Random Access Verfahren

Random Access Netze vom Typ CSMA/CD mit nominellen Datenraten von 10 Mbit/s sind heute die kommerziell erfolgreichsten LANs.

Bei Messungen wurde festgestellt, da die Zeit zwischen Ankunft zweier Pakete eine Verteilung aufweist, die mit der Poisson-Funktion angenahert werden kann.<sup>15</sup> die meisten Abstande betragen zwischen 0 und 20 Millisekunden. Danach fallt die Kurve stark ab. Grundsatzlich unterliegt die Last auf dem Netz uber den Tag hinweg starken Schwankungen.

Die deutlichen Lastspitzen konnen schnell zu Engpassen fuhren. Groe Hersteller empfehlen inzwischen, nicht mehr als funf Workstations mittels Ethernet zu einem Cluster zusammenzufassen.

#### Durchsatz

Durchsatz in **bit/s** kann definiert werden durch die mittlere Anzahl der Bits, die einen gegebenen Punkt im Netz pro Zeiteinheit passieren. Da Pakete demoliert sein konnen, werden fur gewohnlich bei der Messung *nur Bits von fehlerfreien Paketen* gezahlt.

---

15. mit negativ exponentielle Form





Häufig wird der Durchsatz *normalisiert*, indem er *durch die Kanalübertragungsrate*<sup>16</sup> (in bit/s) *dividiert* wird, um eine Zahl zwischen 0 und 1 zu erhalten. Normalisierter Durchsatz (bzw. einfach „Durchsatz“) wird mit  $S = \frac{\text{\# erfolgreich empfangene Bits pro Zeiteinheit}}{\text{nominelle Datenrate}}$  bezeichnet. S ist dann dimensionslos. Der maximale Durchsatz wird Kanalkapazität genannt. Er wird durch die Größe des Overhead für jedes Paket nach oben begrenzt. Paket plus Overhead nennt man Frame. Siehe dazu Tabelle 7, „ISO - Open System Interconnection Referenzmodell,“ auf Seite 5. Wenn der Overhead 10% des Frame ausmacht, dann bekommt man maximal 90% Durchsatz. Dabei ist fehlerfreie Übertragung und kontinuierliches Senden vorausgesetzt.

Verwandt mit dem Durchsatz ist die *Kanalausnutzung* oder *Effizienz*. Dabei werden drei Zeiten betrachtet:

- fehlerfreie Übertragungszeit
- Busyzeit
- Freezeit

Kanalausnutzung ist Busyzeit durch Gesamtzeit. Ohne Fehler und ohne Overhead entspricht das dem Durchsatz. Effizienz ist die Zeit der Datenübertragung (LLC-Daten) dividiert durch die Zeit der Daten (LLC-Daten)- und Overheadübertragung.

Manchmal ist von Last bzw. Anzahl der Versuche bzw. Anzahl der Stationen die Rede. Dabei ist es so, daß in den Testfällen für jede Station eine Last festgelegt wird z.B. 10% Last. Dann müssen 10 Stationen arbeiten, um einer volle Auslastung  $G=1$  bzw.  $G=100\%$  bzw.  $G=\text{full Load}$  zu entsprechen. Im Idealfall erhält man bei 100% Last auch 100% Durchsatz (Kanalausnutzung). Tatsächlich mindert sich der Durchsatz aber in Abhängigkeit von zu hoher Last (zu viele Kollisionen, d.h. die fehlerfreie Übertragungszeit nimmt ab).

### Angebotene Last

Die angebotene Last wird als  $G = \frac{\text{\# gesendete Bits pro Zeiteinheit}}{\text{nominelle Datenrate}}$  definiert.

Logischerweise steigt der Durchsatz erstmal, wenn man die Last erhöht. Wenn man mehr sendet, kann auch mehr empfangen werden. Das geht aber nicht beliebig

---

16. oder „nominelle Datenrate“

so weiter, denn mit steigender Nutzung des Kanals nimmt auch die Möglichkeit einer Kollision zu. Bei einem Lokalfaktor von 0 wird jede Kollision sofort erkannt und keine Zeit mit Wiederholen schon gesendeter Daten verbracht. Bzw. die Laufzeit ist 0. In diesem Fall nimmt der Durchsatz (erfolgreicher Anteil der Sendungen) bis auf 1 zu bei steigender Last. Bei Lokalfaktoren, die größer sind, steigt auch die Möglichkeit für eine Kollision, da die Pakete länger unterwegs sind. Der maximale Durchsatz ist abhängig von  $a$  und von  $G$ . Bei kleinem  $a$  (nahe 0) ist der erreichbare Durchsatz 1. Er fällt mit steigendem  $a$ . Wenn  $a$  1 ist, wird ein maximaler Durchsatz (bei optimaler Last) von 0,2 erreicht. Ein Wert, der von ALOHA bei jedem  $a$  eingefahren wird. Interessant ist, daß bei  $a > 0$  der Durchsatz ab einer bestimmten Belastung wieder zurückgeht. Je größer  $a$  ist, desto eher wird der maximale Durchsatz erreicht.

Carrier Sense liefert bei kleinem  $a$  bessere Information über den tatsächlichen Zustand des Kanals. Bei großem  $a$  kann ein Paket lange unterwegs sein, ohne bemerkt zu werden.

Im Skript: Die Grafik ist aus dem Buch übernommen, wo  $G$  die durchschnittliche Anzahl versuchter Übertragungen (erfolgreiche und andere) für  $P$  Sekunden<sup>17</sup> angibt.  $S$  ist definiert als die Anzahl erfolgreicher Übertragungen in der selben Zeit.

### **Mittlere Paketverzögerung (Transfer Delay)**

Die Zeit die vergeht zwischen dem Zeitpunkt, wo das letzte Bit eines Pakets in einer Station des Netzwerks ankommt, bis das letzte Bit des Pakets über das Netz an die Empfängerstation abgeschickt wurde.

Entspricht der Pakettransferzeit plus die Wartezeit im Interfacebuffer, bevor gesendet werden kann.

CSMA/CD benötigt ein  $a < 0,5$  damit es auch im schlimmsten Fall Kollisionen entdeckt. Ohne CD nicht nötig. Dort wird die volle Sendung durchgezogen und nur vorher auf Kollisionen geachtet. Nachher wartet der Sender auf die Quittung. Mit CD wird nicht quittiert, unnötig denn nach der doppelten Endlaufzeit hat der Sender die Leitung sicher, falls bis dahin keine Kollision vorkommt.

---

17. Paketübermittlungszeit  $P$ : Maximale Signallaufzeit über das Kabel plus die Zeit, alle Bits eines Pakets abzuschicken.



## 5.2.5 Token Bus

Hier entstehen die Verzögerungen nicht durch Kollisionen und Wiederholungen, sondern durch Weitergabe des Token. Im Mittel dauert die Weitergabe des Token ein Drittel der End-End-Laufzeit.

Falls kein Overhead zur LLC-Information hinzugefügt würde käme man bei unendlicher Tokenholdingzeit auf einen maximalen Durchsatz von 1 (falls also die Station ständig sendet).

Im Gegensatz zum CSMA/CD ist der Tokenbus auch bei großem Lokitätsfaktor leistungsfähig und erreicht seinen maximalen Durchsatz auch bei großer Last, wohingegen CSMA/CD bei größeren Lasten zusammenbricht.

Allerdings gibt es ein Beispiel, wo CSMA/CD bei fast jedem Durchsatz (also entsprechend hoher Last) eine geringere mittlere Paketverzögerung aufweist. Hängt mit dem Warten auf ein Token zusammen. Bei Ethernet muß im günstigsten Fall nicht gewartet werden.

## 5.2.6 Token Ring

Unterschiede zum Token Bus:

- Overhead durch Weitergabe des Tokens: Das Token enthält die Quell- und die Zieladresse. Beim Ring ist das nur beim Datenrahmen der Fall, denn der Tokenrahmen hat dort keine Adreßangaben.  
Das weitergegebene Token legt im Schnitt beim Token Bus einen weitem Weg zurück als beim Ring.
- Aktiver Anschluß statt passiver Anschluß: Signal wird regeneriert an jeder Station (Stationslatenz). Diese Verzögerung gibt es beim Token Bus nicht. Der Token Ring ist bei Ausfall einer Station dafür im Eimer.
- Unterschiedliche Verfahren:
  - Multiple Token
  - Single Token
  - Single Frame (Paket)

Auch hier kommt man im Idealfall auf einen Durchsatz von 1: Falls „exhaustive service“ (ohne Token-Holding-Zeit) und kein MAC-Overhead und ununterbrochenes Sendebedürfnis der sendenden Station.



Zu Meßzwecken wird das „average transfer delay“ normalisiert in Einheiten von „mean transmission times“. „Normalized average transfer delay“ genannt dann. Es bedeutet:

$$\frac{\text{Verzögerung bis zum Eintreffen am Ziel}}{\text{mittlere Paketlänge/nominelle Datenrate}} = \frac{\text{Übermittlungszeit}}{\text{reine Sendezeit}}$$

Man dividiert also die mittlere Zeit, die vergeht vom Erhalten des Pakets vom Benutzer der Station bis zum Empfang an der Zielstation durch die reine Sendedauer eines Pakets.

Der beste Wert ist hierfür die 1. Sie bedeutet, daß keine Wartezeit im Puffer der sendewilligen Station auftritt. Das „avd“ ist Wartezeit im Puffer plus Übertragungszeit.

Bei der Analyse der Leistungsfähigkeit der drei verschiedenen Ringe betrachtet man den Durchsatz als Maß für die angebotene *Last* (keine Last, dann Durchsatz 0; viel Last, dann Durchsatz 1) *im Bezug auf* die oben diskutierte normalisierte Paketverzögerung. Ein Wert von (3/0,5) bedeutet z.B., daß bei einer Kanalnutzung von 50% das Senden eines Paketes nur ein 1/3 der gesamten Übertragungszeit ausmacht. Die Warterei auf ein passendes Token also 2/3.

Bei der Analyse wird „exhaustive service“ vorausgesetzt u.a.

Der Multiple Token Ring erweist sich als sehr guter Kandidat für Hochgeschwindigkeitsnetze, da er besonders deutlich bei großem Lokalitätsfaktor für denselben Durchsatz geringere Gesamt-Transferzeiten benötigt.

Da die normalisierte Paketverzögerung die Gesamtzeit/Paketzeit ist, fragt man sich: Woraus besteht die Gesamtzeit „average transfer delay“ beim Ring?

- **Walktime:** Die Zeit, die vergeht, bis die sendewillige Station die Kontrolle über den Kanal erlangt. In Abhängigkeit von der Anzahl der Stationen im Ring und vor allem dem Durchsatz. Im Falle eines Durchsatzes nahe 0, d.h. kaum was los im Ring, entspricht die Walktime im Schnitt der halben Ringlatenz (Kabellatenz und Stationslatenz), die sich aus Kabel-

propagationstime und Stationslatenzen summiert. Die Walktime gibt an, wie lange eine Station auf das Freitoken warten muß.

- Signallaufzeit: Wenn man davon ausgeht, daß im Schnitt die Zielstation auf halber Entfernung liegt, dann ist auch diese Zeit im Mittel die halbe Ringlatenz.
- Paketzeit: Wieviel Zeit es braucht, alle Paketbits in das Netz zu pumpen. Die sog. mittlere Paketsendezeit, die beim Lokalitätsfaktor im Nenner steht.

Damit wäre die „normalisierte Paketverzögerung“ (mit  $S=0$ ):

$$\begin{aligned} & \frac{(\text{Walktime} + \text{Signallaufzeit}) + \text{Paketzeit}}{\text{Paketzeit}} \\ &= \frac{\text{Ringlatenz}}{\text{Paketzeit}} + \frac{\text{Paketzeit}}{\text{Paketzeit}} \\ &= a + 1 \end{aligned}$$

Die Bedingung  $S=0$  ist für die Annahme  $\text{Walktime}=\text{Ringlatenz}/2$  nötig. Da auch die  $\text{Signallaufzeit}=\text{Ringlatenz}/2$  ist kann man die beiden zusammenfassen zu „Ringlatenz“.  $\text{Ringlatenz}/\text{Paketzeit}$  ist die Definition für den Lokalitätsfaktor  $a$ . So kommt es, daß alle Kurven „throughput“ vs. „average normalized delay“ die y-Achse bei  $a+1$  schneiden.

Wenn man die Gleichungen für das nicht normalisierte Transferdelay betrachtet, dann bekommt man für den größten Wert von  $S$ , der kein unendliches  $T$  erzeugt, den maximalen Durchsatz des Ringtyps: Multiple Token1, Single Token  $1/a$  (mit  $a>1$  und 1 sonst), Single Frame  $1/(a+1)$ . Alles bei fester Paketlänge.

Was passiert, wenn man die Datenrate erhöht? Die Ringlatenz nimmt ab, da die Stationslatenzen geringer werden. Die Paketzeit nimmt noch stärker ab, da ein Paket mehr Bits enthält als es Stationen mit deren Bitverzögerungen gibt. Insgesamt wächst also der Lokalitätsfaktor in der Gleichung. Dadurch verschiebt sich das Transferdelay entsprechend in der Grafik nach oben.

Bei festen Paketlängen sind die Werte ähnlich denen bei exponentiell verteilten Längen. Nur bei exponentieller Verteilung weicht die Single Token Kurve vom



Multiple schon bei  $a=1$  deutlich ab. Die Kurven von Multiple Token und Single Frame variieren kaum beim Wechsel auf exponentielle Verteilung.

Für  $S=0$  haben alle Kurven bei beliebiger Verteilung den selben y-Abschnitt, falls sie gleiches  $a$  haben.

## 5.3 Hochgeschwindigkeitsnetze

### Wichtige Einsatzgebiete:

1. Rechenzentrum  
Kopplung von Großrechnern und schneller Peripherie (Alternative zu parallelen Bussystemen).
2. Büro-/ Gebäudevernetzung  
Kopplung von Workstations mit hochauflösenden Bildschirmen und Mini-computern sowie Peripherie.
3. Backbone-Netze  
Kopplung von Teilnetzen in einer hierarchischen Struktur.
  - innerhalb eines privaten Geländes
  - innerhalb einer Stadt (MAN)
  - innerhalb eines Landes
  - international

### 5.3.1 Fibre distributed data interface (FDDI)

100 Mbit/s. Maximal 200 km Ausdehnung insgesamt. Max 1000 Anschlüsse. Multiple Token Ring mit Timersteuerung.

Die **Priorityoperation** ist bei FDDI wie beim Token-Bus: die Übermittlung von Rahmen höchster Priorität wird kontrolliert, um sicherzustellen, daß die verfügbare Ringkapazität (Bandbreite) zwischen allen Stationen aufgeteilt wird. Jede Station darf bei Erhalt des Tokens high-priority Rahmen bis zur **high-priority token-hold time** senden. Für jede der drei unteren Prioritätsebenen (unter high-priority) zählt ein separater Timer die Zeit, die seit dem letzten Erhalt des Tokens auf dieser Ebene in dieser Station vergangen ist. Während die drüberliegenden Ebenen senden läuft der Timer der unteren Ebene weiter bis das Token innerhalb der Station weitergereicht wird. Dies ist der **token rotation timer (TRT)**. Falls die TRT unter dem Schwellwert „**target token rotation time (TTRT)**“ liegt, dürfen entsprechend dieser Zeitdifferenz Rahmenversendet werden. Bei nicht-positiver Differenz muß die (Teil-) Station das Token weitergeben.



Der TRT-Wert steigt offensichtlich mit zunehmendem Verkehr auf dem Ring, da die Weitergabe des Tokens durch versendete Pakete verzögert wird. Somit ist der TRT ein Indikator für die Last bzw. Kanalausnutzung (Throughput).

### **In der Vorlesung:**

Dort sind die Stationen nicht unterteilt. Ein „Token Holding Timer (THT)“ entspricht dort Differenz aus Rotationszeit und Schwellwert. Dieser kann wie oben stationsabhängig verschieden sein. Damit haben die Stationen verschiedene Prioritäten möglicherweise. Soll ohne Prioritäten gearbeitet werden verwenden alle Stationen denselben Schwellwert „Operational Target Token Rotation Time“ T<sub>OPR</sub>. Man kann natürlich jetzt Stationen mit den Unterstationen oben gleichsetzen. Die High-Priority-Unterstationen arbeiten oben mit derselben (high-priority) token hold-time. Die Token Holding Time der tieferen richtet sich nach der Rotationszeit. Ferner wurde beim Token-Bus (in der Vorlesung) gesagt, daß hier eine Token-Holdingzeit für jede Station die Sendezeit begrenzt. In diesem Fall wäre das ein Unterschied zu FDDI, da dort nur für die höchst Prioritätsebene konstante Zeiten vergeben werden. Laut Vorlesung wäre das Prioritätsschema bei beiden nur gleich, wenn beim FDDI ohne Prioritäten gearbeitet wird.

In der Vorlesung haben Stationen Prioritäten nicht die Rahmen!

Die **high-priority token-hold time** wird auch **Synchronzeit** genannt. Die von TRT abhängigen low-level-Frames werden **asynchrone Frames** genannt.

Der Einsatz optischer Übertragungstechnologien favorisiert eine Netztopologie, die aus unidirektionalen Punkt-zu-Punkt-Verbindungen zwischen einzelnen Stationen aufgebaut ist. Ein Ring mit aktiv an das Übertragungsmedium angeschlossenen Stationen erfüllt diese Voraussetzung, wobei die Gefahr eines totalen Systemausfalls infolge eines Leitungs- oder Stationsfehlers sehr hoch ist. Wegen der besonderen Sicherheitsanforderungen an ein Hochgeschwindigkeitsnetz wurde für FDDI eine gegenläufige Doppelringkonfiguration gewählt. Der Primär-Ring wird zur eigentlichen Datenübertragung genutzt, während der Sekundärring als Sicherheitsring dient, auf den im Fehlerfall umgeschaltet werden kann. Man kann dann die beiden Ringe an der Ausfallstelle verbinden und bekommt einen großen.

Die Rate, mit der eine Station **synchrone Daten** übertragen darf, wird ihr von einer Monitorstation zugeteilt. Sie ist abhängig von der aktuellen Verkehrslast. Direkt nach dem Empfang eines freien Token kann mit der Übertragung synchroner Daten



begonnen werden. Die darüberhinausgehende Verweildauer des Token und das damit verbundene Übertragungsrecht für **asynchrone Daten** an einer Station wird dezentral durch lokale Timer gesteuert.

Die synchrone Datenübertragung hat höchste Priorität. Zusätzlich können noch bis zu acht Prioritätsklassen realisiert werden. Weiterhin besteht mit einem sogenannten **Restricted Token** die Möglichkeit, die gesamte asynchrone Bandbreite für eine dialogorientierte Verbindung zwischen ausgewählten Stationen zu reservieren. Während dieses Dialogs können andere Stationen synchrone Rahmen senden, aber keine asynchronen.

Das Token besteht aus Präambel, Startbegrenzer, FrameControl und Endbegrenzer. Ein Frame hat Präambel (Synchronisation), Startbegrenzer (Identifizierung), FrameControl (FrametypID), Zieladressen, Absendeadresse, Info, Frame-CheckingSequence und Endbegrenzer und FrameStatus (Frame kopiert? etc.).

### 5.3.2 Dienstintegration

Klassische Formen der Datenkommunikation wie

- File Transfer
- Remote Login und
- Paging

sind i.d.R. die einzigen Verkehrsarten in konventionellen LANs. Mit wachsenden Datenraten verstärkt sich der Wunsch nach Integration.

Im Resümee ist über das Gesamtvolumen betrachtet der klassische Telefonverkehr kaum zu übertreffen.

Dennoch zeichnen sich Anwendungen wie File-Transfers durch kurzzeitige Lastspitzen „Bursts“ aus. Diese extremen Lastschwankungen schaffen Probleme, die heute nur für bestimmte Spezialfälle gelöst sind.

### 5.3.3 FDDI-II

FDDI-II unterstützt zeitgleiche „isochrone“ Dienste durch die Schaltung fester Kanäle. Die 100Mbit/s Datenrate wird nach einem Time-Division-Multiple-Access-Verfahren in 16 sog. „Wide-Band-Channels“ unterteilt, die jeweils eine Datenrate von 6,144 Mbit/s haben.





Einige der Channels werden für paketvermittelnde Dienste, andere für leitungsvermittelnde Dienste verwendet. Alle paketvermittelnden Channels bilden einen einzigen Kanal mit MAC gemäß FDDI-I.

STM-Technik.

Mit der Entwicklung von FDDI-II sollen über die paketvermittelnden Dienste hinaus die Anforderungen leitungsvermittelnder Dienste erfüllt werden, die typischerweise für Sprach- oder Videoübertragungen erforderlich sind. *Unter dem Token-Protokoll wird eine Zeitmultiplex-Struktur (in STM!) auf dem Medium realisiert, indem eine ausgezeichnete Station im Ring alle 125  $\mu$ s einen Slotrahmen generiert, mit dem alle Stationen zentral getaktet werden. Die Gesamtbreite des FDDI Ringes von 100 Mbit/s wird in 16 isochrome Unterkanäle und eine Minimalbandbreite für das Token-Protokoll von 768 kbit/s aufgeteilt.*

### 5.3.4 Metropolitan Area Networks: DQDB

Einsatzgebiete für MANs:

- Kopplung von LANs
- Übertragung von hochauflösender Graphik
- Bildübertragung in der Medizin
- Videoübertragung
- Zugriff auf zentrale Server
- Kopplung von Nebenstellenanlagen

Es ergibt sich wieder die Forderung nach

- paket-vermittelnden Diensten (vgl. LAN) und nach
- isochronen<sup>18</sup> Diensten (vgl. Telefon, Video)

Beim **Distributed Queue Dual Bus** wird die Sendezeit durch eine dezentral verteilte Warteschlange verwaltet. Topologie eines doppelten Bussystems.

ATM-Technik.

Während der Einsatzbereich von FDDI die direkte Rechnerkopplung und die Kopplung von lokalen Netzen ist, so liegen die Hauptaufgaben für DQDB in der *Vernetzung von städtischen Gebieten*. DQDB ist ein Vertreter der Metropolitan Area Networks (MANs). Ein MAN soll wesentlicher Bestandteil der Kommunika-

---

18. leitungsvermittelnden



tionsinfrastruktur einer Stadt oder eines Stadtteils sein. Daraus folgt, daß insbesondere geeignete Mechanismen zur Sprach- und gegebenenfalls zur Videoübertragung (isochrone<sup>19</sup> Übertragung) zur Verfügung gestellt werden müssen.

### Netzaufbau

DQDB basiert auf einer Architektur bestehend aus zwei unidirektionalen, gegenläufigen Bussen, passiven oder aktiven Stationen (der Standard sieht beides vor) und einer Kopfstation für jeden Bus.

Beide Busse arbeiten unabhängig voneinander, die Gesamtkapazität des Netzes ist somit doppelt so hoch wie die eines einzelnen Busses. Da die Busse unidirektional arbeiten, muß jede Station an beide Busse angeschlossen sein, um alle anderen Stationen erreichen zu können.

Der Zugriff ist so geschaltet, daß eine Station Daten nur kopieren, aber nicht vom Netz entfernen kann. Es ist problemlos möglich, eine Station vom Netz zu entfernen bzw. eine neue Station einzufügen. Dies gilt nicht für die Kopfstationen, nach deren Ausfall der Bus nicht mehr betriebsbereit ist.

### Medienzugangskontrolle

DQDB ermöglicht sowohl asynchrone als auch isochrone Übertragung. Entsprechend müssen unterschiedliche Medienzugangsverfahren unterstützt werden. Hierbei ist darauf zu achten, daß für die Übertragung von isochronen Daten ausreichend Bandbreite garantiert werden kann. Der verwendete Algorithmus ist als Queued-Packet Distributed Switch (QPSX) bekannt.

Die Kopfstationen der beiden Busse generieren Slots fester Länge. Daten werden übertragen, indem eine Station ihre Daten in einen als „frei“ markierten Slot schreibt und diese Daten vom empfangenden Knoten über den Read-Port kopiert werden.

Die DQDB-Ebene bietet zwei Modi des Medienzugangs an:

- **Queued Arbitrated (QA)** für nicht-isochrone Daten,
- **Pre-Arbitrated (PA)** für isochrone Übertragung.

---

19. für eine isochrone Übertragung ist eine garantierte Bandbreite erforderlich



### Queued Arbitrated

Für diesen Modus stellt DQDB spezielle QA-Slots zur Verfügung. Ein solcher Slot besteht aus einem Header und einem Datenteil fester Länge, dem QA-Segment. Die prinzipielle Arbeitsweise des Protokolls in diesem Modus kann wie folgt beschrieben werden:

Das Verfahren nutzt zwei Bits, die in jedem Slot enthalten sind: Das Busy Bit (B-Bit) und das Request Bit (R-Bit). Station *i* möchte an Station *j* eine Nachricht senden. Die auszuführenden Schritte sind nun:

- **Auswahl des geeigneten Busses**  
Falls Bus A zu Station *j* führt, wählt Station *i* diesen für die Übertragung.
- **Das R-bit des nächsten Slots auf Bus B wird gesetzt**  
Hierdurch erfahren alle Knoten oberhalb (upstream) von *i*, daß ein Reservierungswunsch vorliegt. Sie inkrementieren einen Zähler (den Request (RQ) Counter). Der Wert dieses Zählers beim Setzen des R-Bits sagt aus, wieviele leere Slots eine Station abwarten muß, bis sie senden darf. Der Zähler wird mit jedem leeren Slot dekrementiert. Erreicht er den Wert „0“, so darf die Station mit der Übertragung beginnen. Es existiert somit eine verteilte **Warteschlange** für das gesamte Netz. **Prioritäten** lassen sich einfach realisieren, indem für jede Priorität eine solche Warteschlange gehalten wird. Das Request-Field jedes Slots enthält vier Bits, über die sich vier Prioritäten realisieren lassen.

### Pre-Arbitrated

Der PA-Modus dient der Übertragung von isochronen Daten. Auch hierfür stehen wieder Slots fester Länge zur Verfügung. Während im QA-Modus ein solcher Slot zu einem Zeitpunkt zu genau einer Station gehört, können sich mehrere Stationen einen PA-Slot teilen.

Die Kopfstation eines Busses muß so viele PA-Slots erzeugen, daß für alle Benutzer des isochronen Dienstes die erforderliche Bandbreite zur Verfügung steht. Ein PA-Slot enthält einen Virtual Channel Identifier (VCI). Jede Station, die isochrone Daten übertragen möchte, hält für jeden VC in einem PA-Slot, über den sie Daten empfängt bzw. sendet, eine Tabelle, welche Oktette für sie bestimmt ist. Alle anderen Oktetten werden ignoriert. Ein Slot mit einem VCI, den diese Station nicht benutzt, wird ebenfalls ignoriert.

## 6. Network Layer (Netzebene, „3“)

Aufgabe: Aufbau, Betrieb, Abbau von netzweiten Verbindungen.

Protokolle zwischen Endsystemen, ggf. über mehrere Zwischensysteme, im Gegensatz zu den ersten Ebenen, die nur zwischen direkten Nachbarn protokollieren.

### 6.1 Kopplung von lokalen Netzen

In der Praxis erfolgt die Netzkopplung nicht immer auf Ebene 3. Man unterscheidet:

- Repeater: kopiert alles bitweise ins andere Netz, d.h. keine Filterung. Physikalische Ebene 1.
- Bridge: empfängt ganze Rahmen und leitet ggf. in anderes Netz weiter. Sicherungsebene 2.
- Router: wie Bridge, aber Verwendung von Adressen der Ebene 3; modifiziert ggf. auch Adressen. Netzwerkebene 3.
- Gateway: übersetzt Protokolle mindestens ab Ebene 3; manchmal sogar bis Ebene 7 (bei nicht-OSI-Netzen).

## 7. Transport Layer (Transportebene, „4“)

Die Transportschicht soll den höheren Ebenen die gewünschte Transportdienstqualität, unabhängig vom Netz, zu möglichst geringen Kosten bereitstellen.

Schicht 4 muß die Unterschiede zwischen Forderungen der höheren Ebenen und Möglichkeiten des Netzes durch geeignete Maßnahmen anzugleichen versuchen.

Aufgabe: Bereitstellung einer netzunabhängigen, zuverlässigen End-End-Verbindung für uninterpretierten, d.h. anwendungsunabhängigen, Datenfluß.



## **8. Session Layer (Sitzungsebene, „5“)**

Aufgabe: Strukturierung und Synchronisation der Kommunikation über von Ebene 4 gestellte Transportkanäle. Bereitstellung von Diensten, die je nach Anwendung sinnvoll sind (oder nicht).

## **9. Presentation Layer (Darstellungsebene, „6“)**

Aufgaben: Anpassung unterschiedlicher lokaler Formate an eine einheitliche, globale Syntax unter Beibehaltung der Semantik.

## **10. Application Layer (Anwendungsebene, „7“)**

Unterschied zu anderen Ebenen: Benutzer ist nicht „Einheit“ der nächsthöheren Ebene, sondern Prozeß z.B. Anwendungsprogramm.

Ein Anwendungsprozeß wird

- lokal von Benutzer/Programm oder
- auf Wunsch des Kommunikationspartners

aufgerufen.

Er besteht aus

- Application Agent und
- Application Entity

Application Agent ermöglicht es dem „Benutzer“ bzw. dem Betriebssystem eines Endsystems, lokal auf system-spezifische Ressourcen zuzugreifen. Liegt außerhalb des OSI-Modells.

Application Entity führt systemunabhängige (Standard-) Aktivitäten durch, die als Anwendungsdienste durch den Application Agent verfügbar gemacht werden.



Statt 2 Standards (Service und Protokoll) umfaßt Ebene 7 alle Standards der Anwendungsebene, die für bestimmte Anwendungen sinnvoll sind – für andere nicht.

