



Rechnernetze I

SoSe 2020

Roland Wismüller
Universität Siegen
roland.wismueller@uni-siegen.de
Tel.: 0271/740-4050, Büro: H-B 8404

Stand: 11. Mai 2020



Rechnernetze I

SoSe 2020

3 Direktverbindungsnetze

Inhalt

- ➔ Hardware-Bausteine: Knoten und Verbindungsleitungen
- ➔ Modulation
- ➔ Codierung
- ➔ Framing
- ➔ Fehlererkennung und Fehlerkorrektur
- ➔ Medienzugriffssteuerung (MAC)
 - ➔ Ethernet (CSMA-CD)
 - ➔ Token-Ring

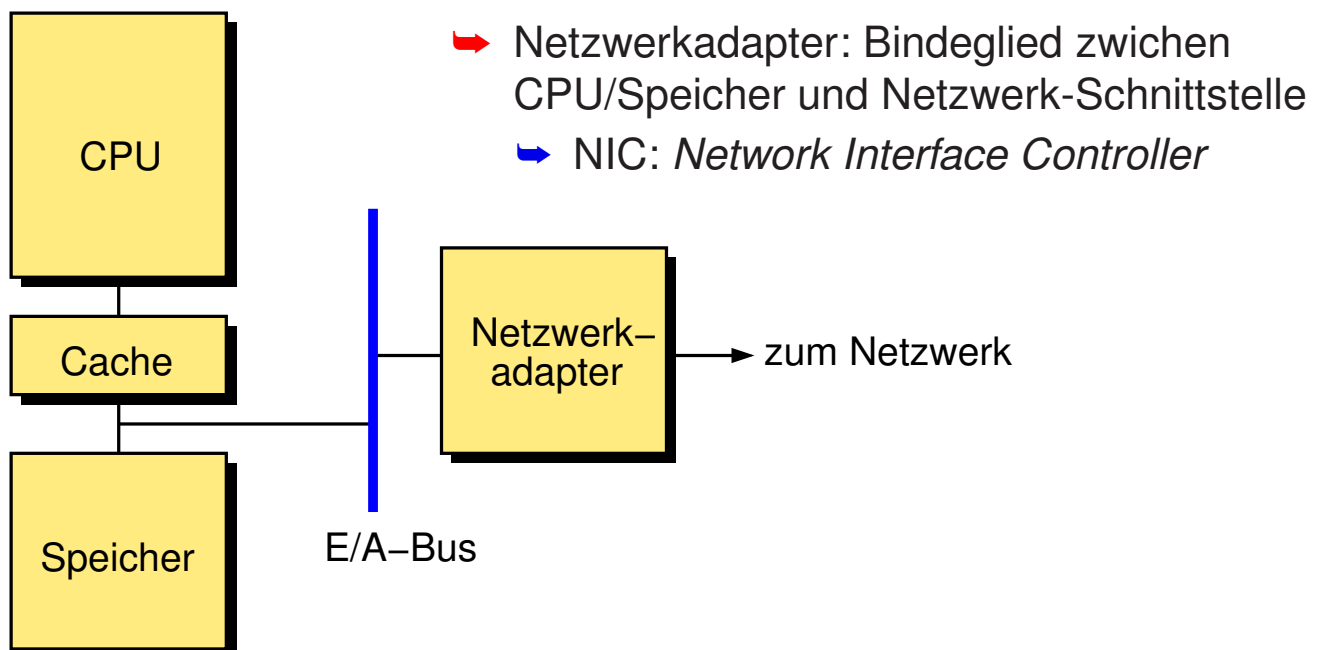
- ➔ Peterson, Kap. 2.1 – 2.6, 2.7.2
- ➔ CCNA, Kap. 4, 5.1

3.1 Hardwarebausteine

OSI: 1



Aufbau eines Knotens



Verbindungs„leitungen“

- ➔ Übertragen Signale als elektromagnetische Wellen
- ➔ Typische Attribute:
 - ➔ Frequenz- bzw. Wellenlängenbereich (Bandbreite)
 - ➔ Dämpfung (max. Kabellänge)
 - ➔ Richtung des Datenflusses
 - ➔ **Simplex**: nur in eine Richtung
 - ➔ **Vollduplex**: in beide Richtungen, gleichzeitig
 - ➔ **Halbduplex**: in beide Richtungen, abwechselnd
- ➔ Grundlegende Arten:
 - ➔ Kupferkabel
 - ➔ Glasfaserkabel (Lichtwellenleiter)
 - ➔ Drahtlose Verbindung (Funk, IR) (☞ **RN_II**)

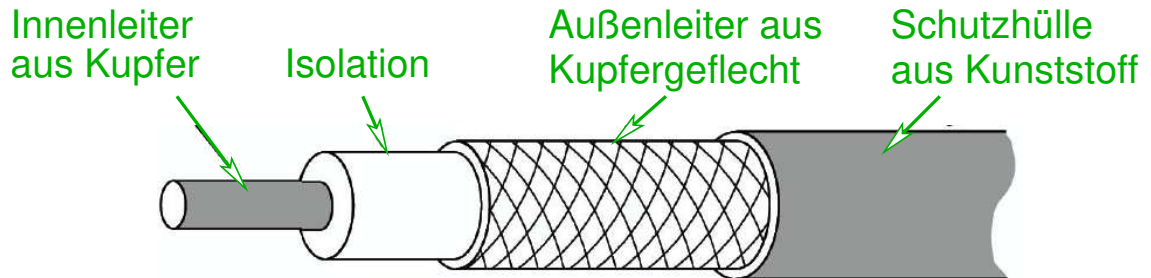
Anmerkungen zu Folie 70:

Die OSI-Schicht 1 hat eine gewisse Sonderrolle im Schichtenmodell, da es die einzige Schicht ist, die keine darunterliegende Schicht hat. Das bedeutet, dass die tatsächliche Kommunikation (im Gegensatz zu allen andern Schichten) auf derselben Schicht stattfindet. Daher müssen auf der Bitübertragungsschicht neben dem Dienst, den diese Schicht nach oben anbietet, auch die Eigenschaften der tatsächlichen Verbindung definiert werden.

Auf dieser Folie sind dabei nicht nur die reinen Eigenschaften der Verbindungsleitung genannt, sondern teilweise auch die Art ihrer Verwendung (z.B. bei „Frequenz- bzw. Wellenlängenbereich“ und bei „Richtung des Datenflusses“).

Kupferkabel: Koaxialkabel

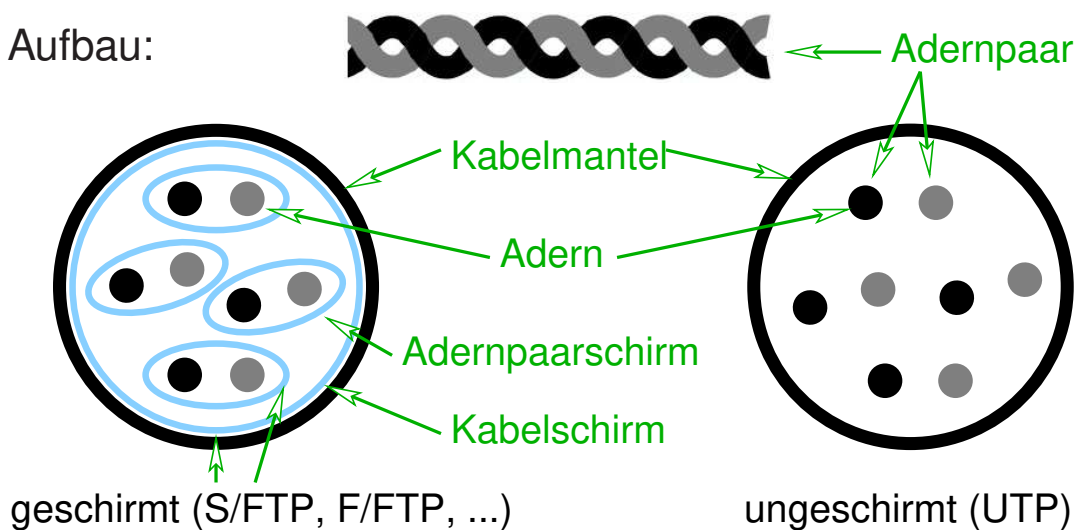
➔ Aufbau:



- ➔ Hohe Bandbreite, geringe Dämpfung, teuer
- ➔ Basisband-Kabel (direkte Übertragung, 1 Kanal, <500m)
 - Beispiele: Ethernet 10BASE-5, 10BASE-2
- ➔ Breitband-Kabel (Modulation auf Träger, mehrere Kanäle, mehrere km)
 - Beispiel: Fernsehkabel

Kupferkabel: Twisted-Pair (verdrilltes) Kabel

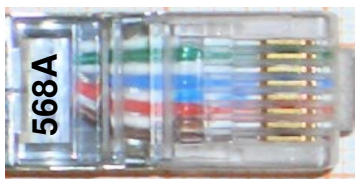
➔ Aufbau:



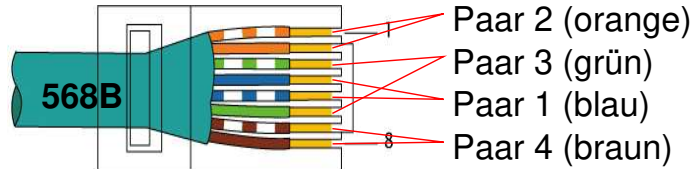
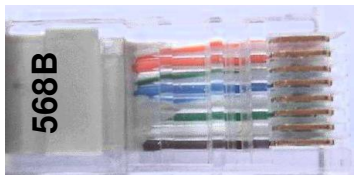
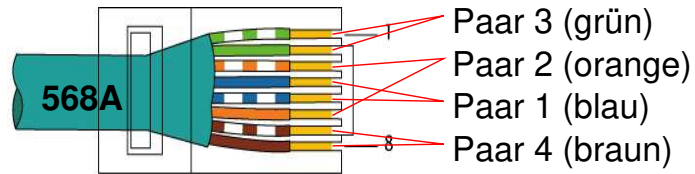
- ➔ Geringe Kosten, relativ gute Bandbreite
- ➔ Beispiel: Ethernet 100BASE-TX

Anmerkungen zu Folie 72:

- ➔ Für die Belegung der RJ-45-Stecker von Ethernet-Kabeln gibt es zwei unterschiedliche Normen: EIA/TIA-568A und EIA/TIA-568B:



(c) Uweschoebel at the German language Wikipedia



- ➔ Bei einem normalen Ethernet-Patch-Kabel (*Straight-Through-Kabel*) sind beide Enden nach demselben Standard belegt, also entweder beide nach 568A oder beide nach 568B. Diese Kabel werden zur Verbindung von Rechnern mit Switches eingesetzt.
- ➔ Zur Verbindung zweier Rechner (oder zweier Switches) gibt es sog. *Crossover-Kabel*, bei denen (für 10 bzw. 100 Mb/s) die Adernpaare 2 und 3 getauscht werden. Ein Ende ist also nach 568A, das andere nach 568B belegt.

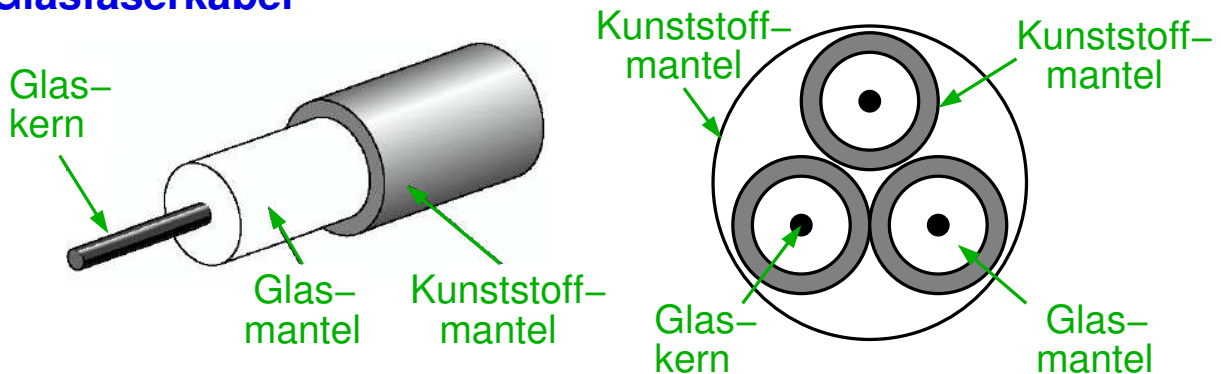
72-1

3.1 Hardwarebausteine ...



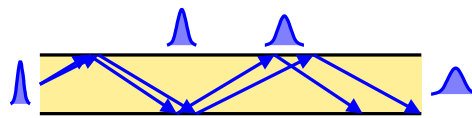
(Animierte Folie)

Glasfaserkabel



- ➔ Führung von Lichtwellen durch Totalreflexion
- ➔ Bandbreite im Bereich Gb/s, Länge im Bereich km
- ➔ Varianten:

- ➔ Multimode-Faser



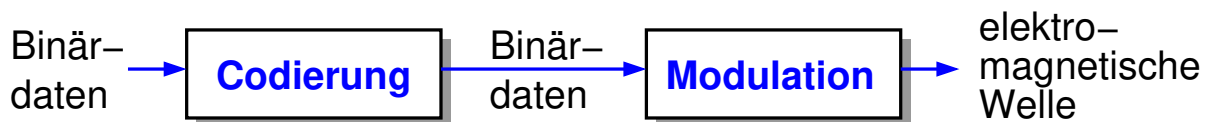
- ➔ Monomode-Faser



- ➔ hohe Bandbreite, teuer (Laserdioden)



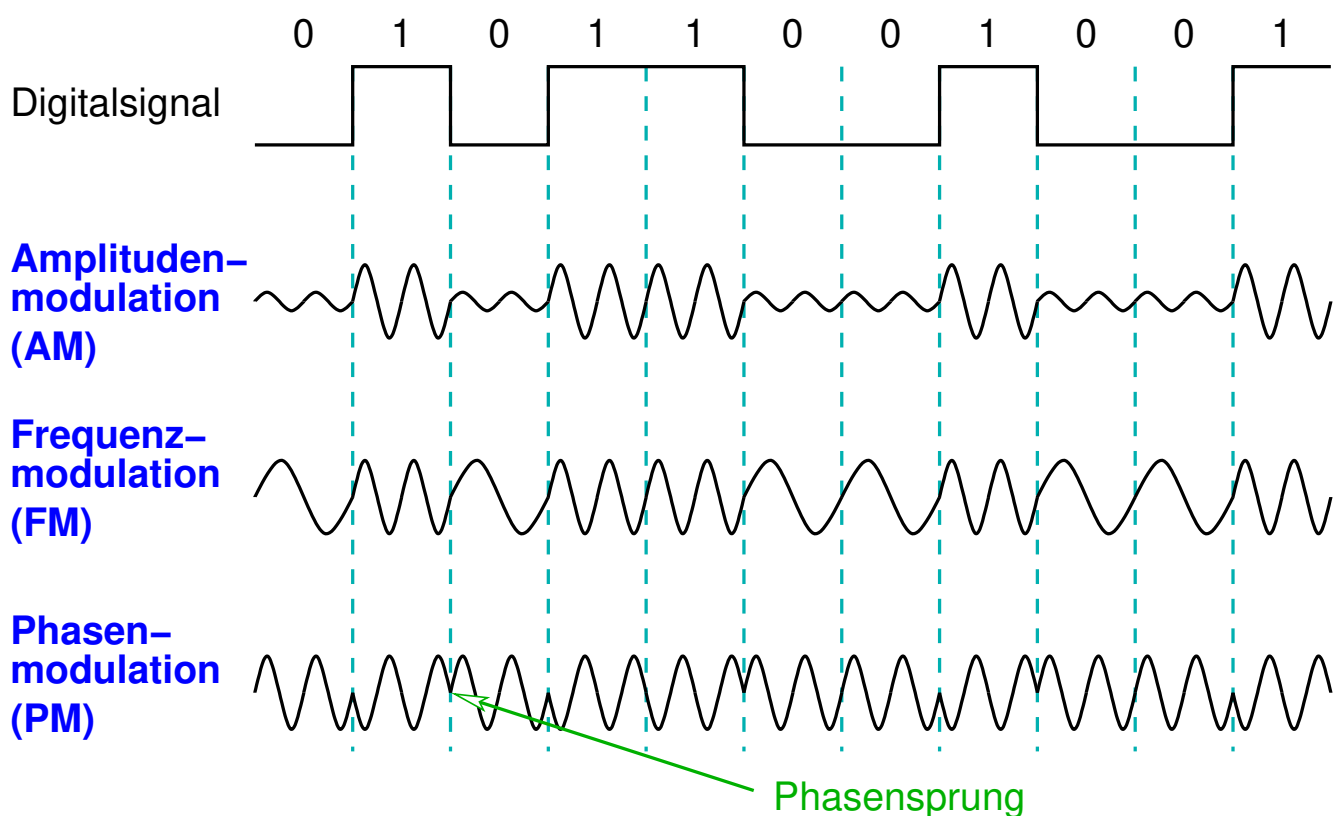
- ➔ Zur Übertragung müssen Binärdaten (digitale Signale) in analoge elektrische Signale (elektromagnetische Wellen) umgesetzt werden
- ➔ Umsetzung in zwei Schritten:



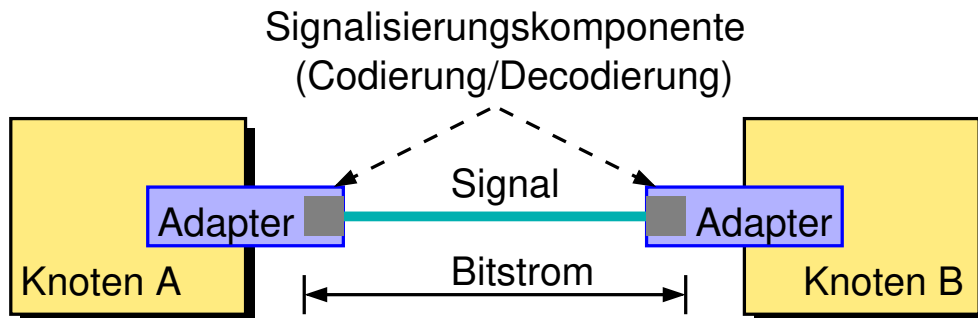
➔ Modulation:

- ➔ Variation von Frequenz, Amplitude und/oder Phase einer Welle
- ➔ zur Überlagerung der (Träger-)Welle mit dem Nutzsignal
 - ➔ z.B. bei Funk, Modem, Breitbandkabel, ...
- ➔ (entfällt bei Basisband-Übertragung)

3.2 Modulation ...



- ➔ Übertragung eines Bitstroms zwischen zwei Knoten:



- ➔ Einfachste Codierung:
 - ➔ **Non-Return to Zero (NRZ):** $1 \hat{=} high$, $0 \hat{=} low$
- ➔ Probleme:
 - ➔ Festlegung der Spannungspegel für *high* und *low*
 - ➔ **Taktwiederherstellung (Synchronisation)**
 - ➔ wo ist die „Grenze“ zwischen zwei Bits?

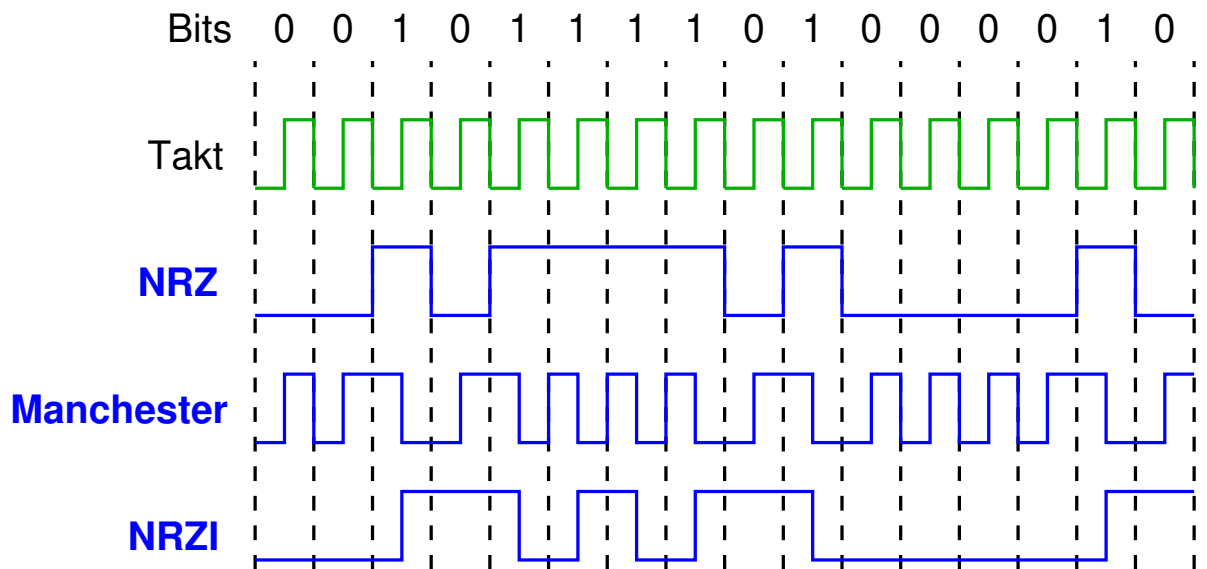
Anmerkungen zu Folie 76:

Man unterscheidet zwischen **Leitungscodierung** und **Kanalcodierung**. In diesem Abschnitt wird ausschließlich die Leitungscodierung betrachtet!

Die Kanalcodierung dient dazu, einen Bitstrom bei der Übertragung über gestörte Kanäle durch Hinzufügen von Redundanz gegen Übertragungsfehler zu schützen.



➔ Abhilfe: Codierungen mit Taktwiederherstellung



NRZI: *Non-Return to Zero Inverted*



Manchester-Codierung

- ➔ Bitstrom wird mit Taktsignal EXOR-verknüpft
- ➔ Anwendung z.B. bei 10 Mb/s Ethernet
- ➔ Problem:
 - ➔ **Baudrate** (Rate, mit der das Signal abgetastet werden muß) ist doppelt so hoch wie die Bitrate
 - ➔ verschwendet Bandbreite

NRZI

- ➔ Signal wird bei jedem 1-Bit invertiert
- ➔ Problem: keine Taktwiederherstellung bei aufeinanderfolgenden Nullen möglich

4B/5B-Codierung

- ➔ 4 Datenbits werden auf 5-Bit Codeworte so abgebildet, daß nie mehr als 3 aufeinanderfolgende Nullen übertragen werden müssen
 - ➔ jedes der 5-Bit Codeworte hat
 - ➔ höchstens eine Null am Anfang
 - ➔ höchstens zwei Nullen am Ende
 - ➔ Übertragung der Codeworte z.B. mit NRZI
 - ➔ Overhead nur noch 25%

- ➔ Bei schnellen Netzen (z.B. Fast Ethernet, GBit-Ethernet) oder auch schnellen Modems werden noch effizientere Verfahren zur Taktrückgewinnung eingesetzt

Anmerkungen zu Folie 79:

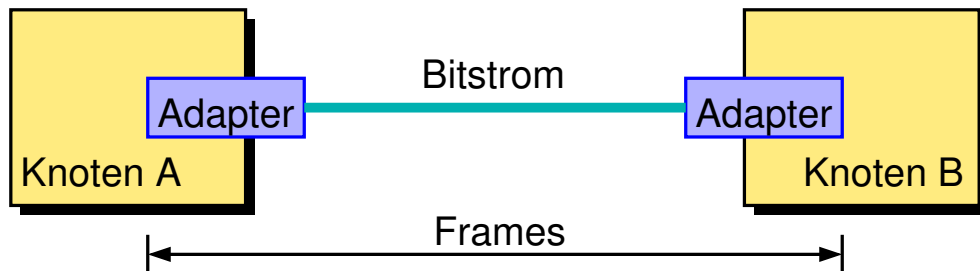
Die genaue Codetabelle der 4B/5B-Codierung ist:

Daten (4 Bit)	codierte Daten (5 Bit)	Daten (4 Bit)	codierte Daten (5 Bit)
0000	11110	1000	10010
0001	01001	1001	10011
0010	10100	1010	10110
0011	10101	1011	10111
0100	01010	1100	11010
0101	01011	1101	11011
0110	01110	1110	11100
0111	01111	1111	11101

Man erkennt, dass die Codierung auch die Länge von 1-Folgen begrenzt (maximal 8 aufeinanderfolgende 1-Bits).



- ➔ Wir betrachten nun die Übertragung von Datenblöcken (**Frames**) zwischen Rechnern:



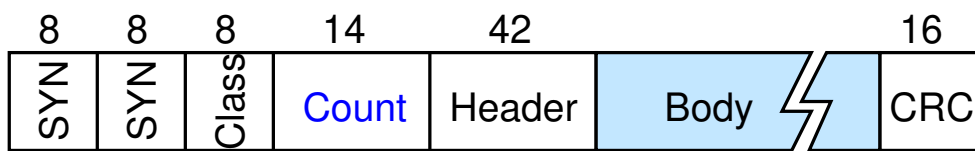
- ➔ Gründe für die Aufteilung von Daten in Frames:
 - einfaches Multiplexing verschiedener Kommunikationen
 - bei Fehler muss nur betroffener Frame neu übertragen werden
- ➔ Zentrale Aufgabe des Framings:
 - Erkennung, wo Frame im Bitstrom anfängt und wo er aufhört
 - dazu: Framegrenzen müssen im Bitstrom erkennbar sein

3.4 Framing ...



Byte-Count Methode

- ➔ Frame-Header enthält Länge des Datenteils
- ➔ Beispiel: Frame im DDCMP-Protokoll, DECNET:



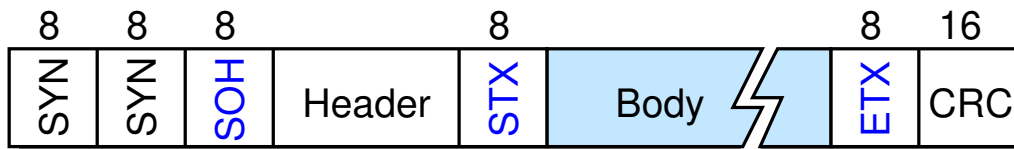
- ➔ Problem: was passiert, wenn die Länge fehlerhaft übertragen wird?
 - Frame-Ende wird nicht korrekt erkannt
 - SYN-Zeichen am Beginn jedes Frames, um (wahrscheinlichen!) Anfang des Folgeframes zu finden
- ➔ Verwendet u.a. beim ursprünglichen Ethernet

3.4 Framing ...



Sentinel-Methode

- ➔ Frame-Ende wird durch spezielles Zeichen markiert
- ➔ Beispiel: Frame im BISYNC-Protokoll (IBM):



- ➔ Problem: Das Endezeichen kann auch im Datenteil (Body) vorkommen
- ➔ Lösung: **Byte-Stuffing**
 - ➔ ersetze ETX im Datenteil durch DLE ETX
 - ➔ ersetze DLE im Datenteil durch DLE DLE
 - ➔ verwendet u.a. bei PPP

3.4 Framing ...



Sentinel-Methode ...

- ➔ Lösung: **Bit-Stuffing**
 - ➔ Eindeutigkeit durch Einfügen von Bits in den Bitstrom erreicht
 - ➔ Beispiel (HDLC-Protokoll):
 - ➔ Anfangs- und Endemarkierung ist 01111110_2
 - ➔ nach 5 aufeinanderfolgenden 1-Bits wird vom Sender ein 0-Bit in den Bitstrom eingeschoben
 - ➔ wenn Empfänger 5 aufeinanderfolgende 1-Bits gelesen hat:
 - ➔ nächstes Bit = 0: ignorieren, da eingeschoben
 - ➔ nächstes Bit = 1: sollte Endemarkierung sein (prüfe, ob die 0 folgt; falls nicht: Fehler)
- ➔ Lösung: Nutzung „ungültiger“ Codeworte
 - ➔ Anfang und Ende durch Codeworte markiert, die sonst nicht vorkommen (z.B. bei 4B/5B-Codierung)



- ➔ Ziel: Übertragungsfehler in Frames erkennen (und behandeln)
- ➔ Möglichkeiten zur Fehlerbehandlung:
 - Korrektur des Fehlers beim Empfänger
 - Verwerfen der fehlerhaften Frames, Neuübertragung durch das Sicherungsprotokoll (☞ 7.4)
- ➔ Vorgehensweise: Hinzufügen von **Redundanzbits** (Prüfbits) zu jedem Frame
- ➔ Theoretischer Hintergrund: **Hamming-Distanz**
 - Hamming-Distanz d = Minimale Anzahl von Bits, in denen sich zwei Worte eines Codes unterscheiden
 - $d \geq f + 1 \Rightarrow f$ Einzelbitfehler erkennbar
 - $d \geq 2 \cdot f + 1 \Rightarrow f$ Einzelbitfehler korrigierbar
- ➔ Beispiel: Paritätsbit führt zu $d = 2$

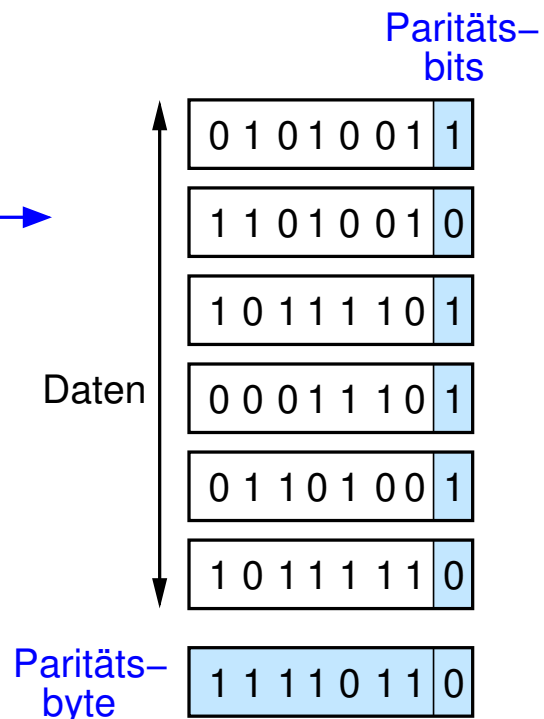
3.5 Fehlererkennung ...



(Animierte Folie)

Zweidimensionale Parität

- ➔ Erweiterung der einfachen Parität
- ➔ Beispiel: 6 Worte á 7 Bit
- ➔ Erkennt alle 1, 2, 3 sowie die meisten 4-Bit-Fehler
- ➔ Erlaubt auch die Korrektur von 1-Bit-Fehlern



Anmerkungen zu Folie 85:

Die zweidimensionale Parität hat keine praktische Bedeutung in Rechnernetzen, sondern dient hier nur zur Veranschaulichung, wie eine Fehlerkorrektur erreicht werden kann.

85-1

3.5 Fehlererkennung ...



(Animierte Folie)

CRC (Cyclic Redundancy Check)

- ➔ Ziel: hohe Wahrscheinlichkeit der Fehlererkennung mit möglichst wenig Prüfbits
- ➔ Basis des CRC-Verfahrens: Polynomdivision mit Modulo-2-Arithmetik (d.h. Add./Subtr. entspricht EXOR)
- ➔ Idee:
 - ➔ jede Nachricht M kann als Polynom $M(x)$ aufgefaßt werden, z.B.
7 6 5 4 3 2 1 0
➔ $M = 1\ 0\ 0\ 1\ 1\ 0\ 1\ 0$ (Bits 7, 4, 3, 1 sind 1)
➔ $M(x) = x^7 + x^4 + x^3 + x^1$
 - ➔ wähle Generatorpolynom $C(x)$ vom Grad k
 - ➔ erweitere M um k Prüfbits zu Nachricht P , so daß $P(x)$ ohne Rest durch $C(x)$ teilbar ist

3.5 Fehlererkennung ...



(Animierte Folie)

CRC (Cyclic Redundancy Check) ...

➔ Beispiel zur Polynomdivision

➔ Nachricht M : 10011010

➔ 3 Prüfbits ($k = 3$)

➔ Generator C : 1101

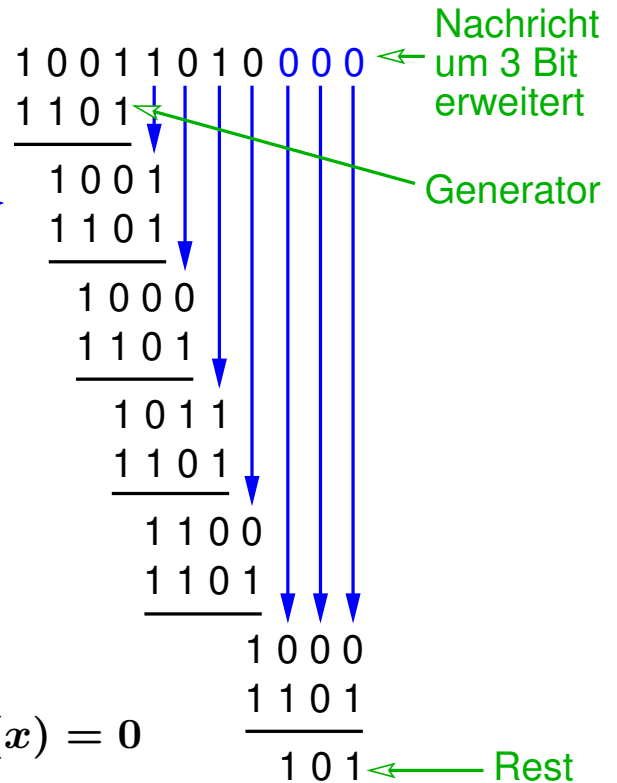
➔ Divisionsrest R wird an die Nachricht M angefügt

➔ Versendete Nachricht P :
10011010101

➔ Diese Nachricht ist durch den Generator ohne Rest teilbar:

➔ $R(x) = M(x) \bmod C(x)$

$\Rightarrow (M(x) - R(x)) \bmod C(x) = 0$



3.5 Fehlererkennung ...



CRC (Cyclic Redundancy Check) ...

➔ Wahl des Generatorpolynoms?

➔ So, daß möglichst viele Fehler erkannt werden!

➔ Beispiel für ein übliches CRC-Polynom:

➔ CRC-16: $x^{16} + x^{15} + x^2 + 1$

➔ CRC-16 erkennt:

➔ alle Ein-und Zweibitfehler

➔ alle Fehler mit ungerader Bitanzahl

➔ alle Fehlerbündel mit Länge ≤ 16 Bit

➔ Gründe für den Einsatz von CRC:

➔ Gute Fehlererkennung

➔ Sehr effizient in Hardware realisierbar





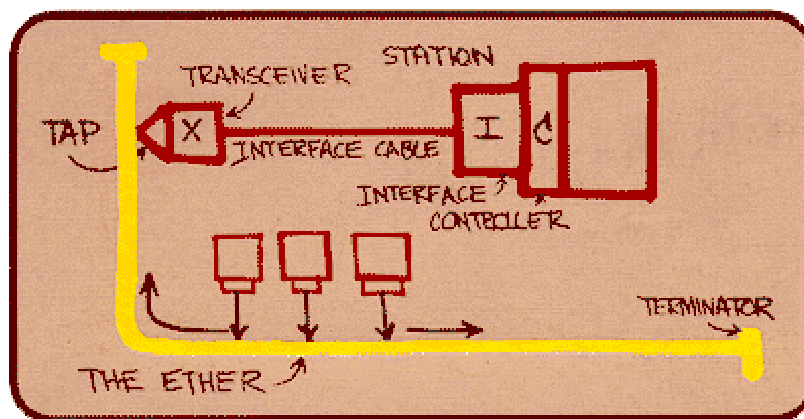
- ➔ In vielen LANs:
 - ➔ Knoten greifen auf ein gemeinsames Medium zu
 - ➔ Zugriff muß geregelt werden, um **Kollisionen** zu vermeiden:
 - ➔ zu jeder Zeit darf nur jeweils ein Knoten senden

- ➔ Typische Verfahren:
 - ➔ CSMA/CD (Ethernet)
 - ➔ Tokenbasierte Verfahren (Token-Ring)
 - ➔ CSMA/CA (WLAN, CAN-Bus, ↗ **RN-II**)

3.6.1 Ethernet



- ➔ Erfolgreichste LAN-Technologie der letzten Jahre
- ➔ Im Folgenden: Grundlagen, 10 Mb/s und 100 Mb/s Ethernet
- ➔ Ursprünglich Mehrfachzugriffsnetz: alle Rechner nutzen eine gemeinsame Verbindungsleitung



Bob Metcalfe
Xerox, 1976

- ➔ Heute: Punkt-zu-Punkt Verbindungen

Anmerkungen zu Folie 90:

Heute gibt es viele verschiedene Ethernet-Standards, die durch *Autonegotiation*-Mechanismen häufig kompatibel zueinander sind. Die wichtigsten Standards sind:

- ➔ 10BASE-2, 10BASE-5: 10 Mb/s, Busverkabelung, Koaxialkabel, veraltet
- ➔ 10BASE-T: 10 Mb/s, Sternverkabelung, UTP, veraltet
- ➔ 100BASE-TX: 100 Mb/s, UTP-Kupferkabel, heute Standard
- ➔ 100BASE-FX: 100 Mb/s, Multimode-Glasfaserkabel
- ➔ 1000BASE-T: 1 Gb/s, UTP-Kupferkabel, heute Standard
- ➔ 1000BASE-SX, 1000BASE-LX: 1 Gb/s, Multi- bzw. Monomode-Glasfaser
- ➔ 10GBASE-T: 10 Gb/s, UTP-Kupferkabel
- ➔ 10GBASE-SR, 10GBASE-LR: 10 Gb/s, Multi- bzw. Monomode-Glasfaser, im Core-Bereich
- ➔ 40 und 100 Gb/s Standards existieren seit Juni 2010
- ➔ 200 und 400 Gb/s Standards existieren seit 2017

90-1

3.6.1 Ethernet ...

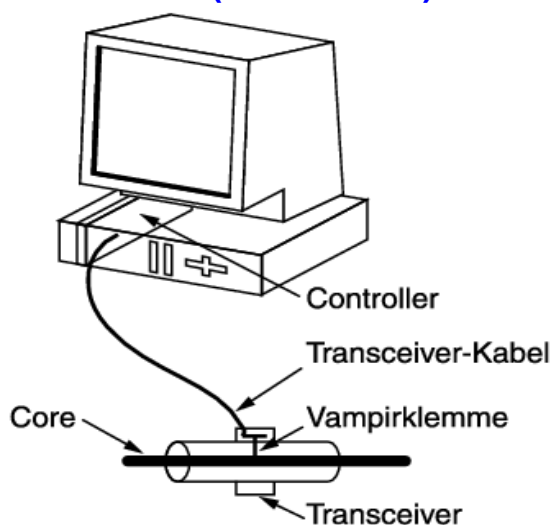
OSI: 1



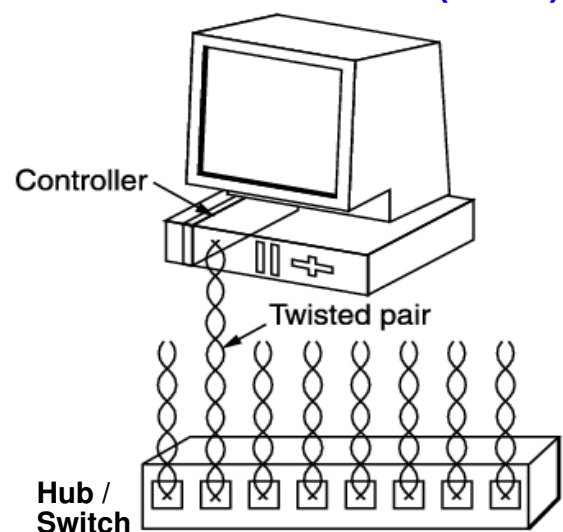
(Animierte Folie)

Verkabelung

10Base5 (max. 500m)



10BaseT / 100BaseTx (100m)



➔ **Hub**: „Verstärker“ und Verteiler (OSI-Schicht 1)

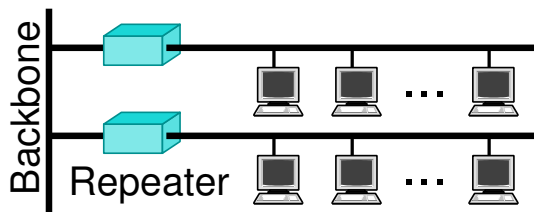
➔ **Switch**: Vermittlungsknoten (OSI-Schicht 2)



Physische Eigenschaften

10BASE-5 (10 Mb/s)

- ➔ Segmente aus Koaxialkabel, je max. 500m
- ➔ Segmente über **Repeater** (Hub mit 2 Ports) verbindbar

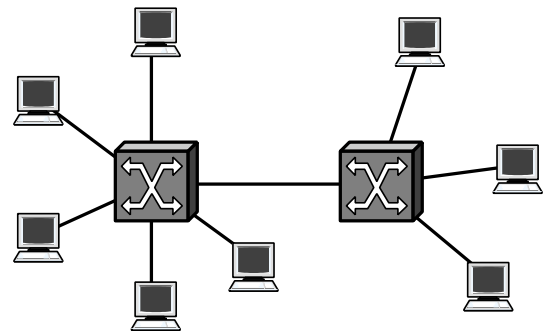


- ➔ max. 4 Repeater zw. zwei Knoten erlaubt

- ➔ Manchester-Codierung

100BASE-TX (100 Mb/s)

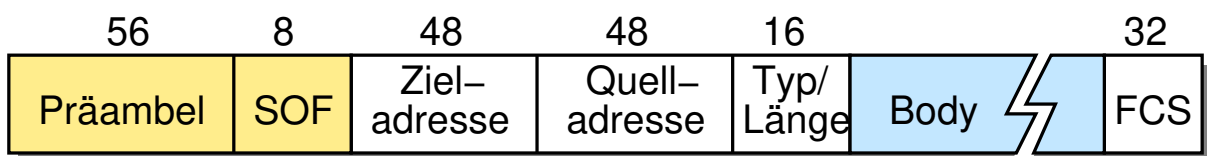
- ➔ Twisted-Pair Kabel, je max. 100m
- ➔ Sternförmige Verkabelung mit Hubs / Switches



- ➔ 4B5B-Codierung



Frame-Format



- ➔ **Präambel/SOF:** Bitfolge 10101010 ... 10101011
 - ➔ zur Takt- und Frame-Synchronisation des Empfängers
 - ➔ letztes Byte (SOF, *Start of Frame*) markiert Frame-Anfang
- ➔ **Typ/Länge:**
 - ➔ Wert < 1536 (0600_{16}): Framelänge
 - ➔ Wert ≥ 1536 : *EtherType*, spezifiziert Protokoll im *Body*
- ➔ **FCS** (*Frame Check Sequence*): 32-Bit CRC Wert

Anmerkungen zu Folie 93:

Im ursprünglichen Ethernet-Standard von Xerox war nur die Länge des Datenteils im Header angegeben. Bei Ethernet II (auch DIX Ethernet genannt) gibt das Typ/Länge-Feld nur den *EtherType* an, d.h. an welches Schicht-3 Protokoll der Empfänger die Nutzdaten übergeben soll. Bei der späteren Standardisierung durch die IEEE (Norm IEEE 802.3) wurde das Feld dann zunächst wieder nur als Längengeld definiert, erst später wurde die beschriebene Doppelverwendung standardisiert.

Wenn das Typ/Länge-Feld einen *EtherType* enthält, muß das Frameende auf OSI-Schicht 1 erkannt (Fehlen eines (Takt-)Signals bei 10 Mb/s Ethernet, „ungültiger“ 4B5B-Code bei 100 Mb/s Ethernet) und an Schicht 2 weitergemeldet werden.

93-1

3.6.1 Ethernet ...



Ethernet-Adressen (MAC-Adressen)

- ➔ Identifizieren die Netzwerkkarte
- ➔ 6 Byte (48 Bit) lang, weltweit eindeutig
- ➔ Schreibweise: byteweise hexadezimal, mit ':' oder '-' als Trennzeichen, z.B. 01:4A:3E:02:4C:FE
- ➔ jeder Hersteller erhält ein eindeutiges 24 Bit Präfix und vergibt eindeutige Suffixe
- ➔ Niedrigstwertiges Bit = 1: Multicast-Adresse
- ➔ Adresse ff:ff:ff:ff:ff:ff als Broadcast-Adresse
- ➔ Die Netzwerkkarte bestimmt, welche Frames sie empfängt

Anmerkungen zu Folie 94:

Bei Ethernet MAC-Adressen (nach IEEE 802.3) wird dabei das **niedrigstwertige** Byte zuerst notiert.

Ein Beispiel für eine Multicast-Adresse wäre daher: 01:00:0c:cc:cc:cc

Das „Multicast“-Bit ist damit Teil des Hersteller-Präfixes, ebenso wie ein weiteres spezielles Bit, das die Adresse als lokal bzw. weltweit eindeutig kennzeichnet.

94-1

3.6.2 CSMA/CD

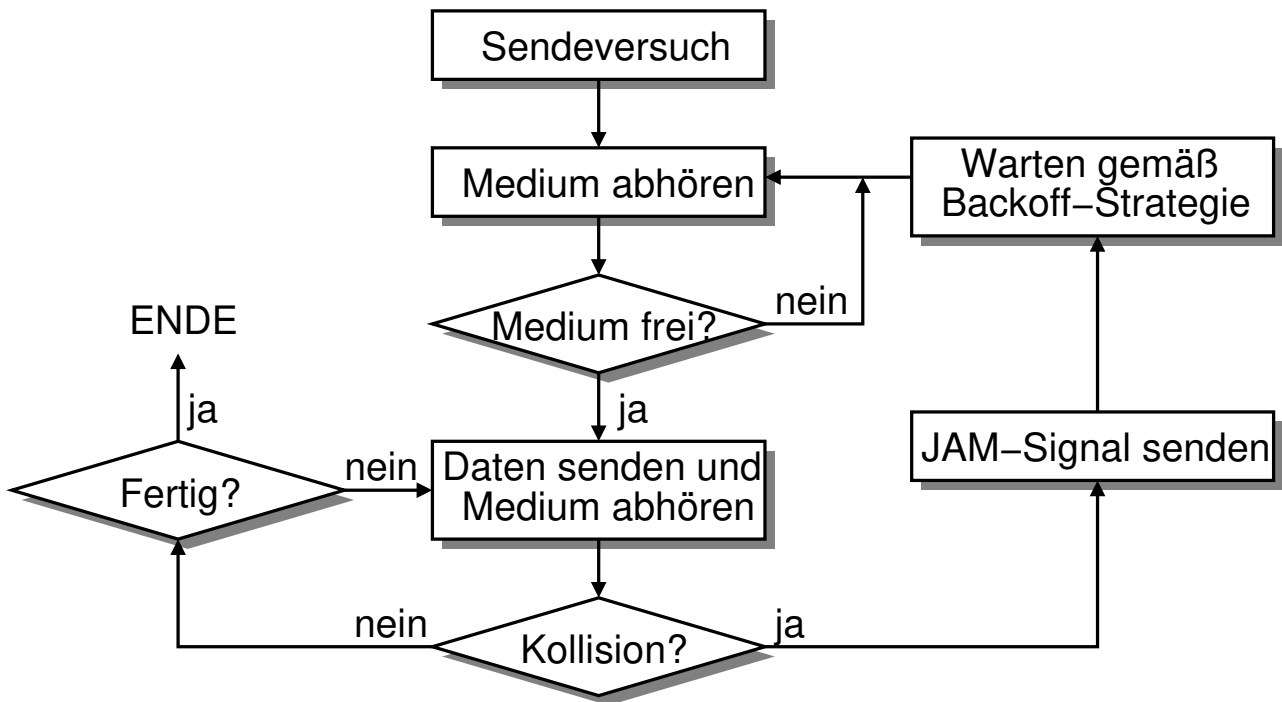
OSI: 2



Begriffsdefinition

- ➔ **Zugangsprotokoll** zum gemeinsamen Übertragungsmedium beim Ethernet
 - ➔ **Carrier Sense Multiple Access**
 - ➔ jede Netzwerkkarte prüft zunächst, ob die Leitung frei ist, bevor sie einen Frame sendet
 - ➔ wenn die Leitung frei ist, sendet die Netzwerkkarte ihren Frame
 - ➔ **Collision Detection**
 - ➔ beim Senden erkennt der Sender Kollisionen mit Frames, die andere Netzwerkkarten evtl. gleichzeitig senden
 - ➔ bei Kollision: Abbruch des Sendens, nach einiger Zeit neuer Sendeversuch

CSMA/CD – Algorithmus



Anmerkungen zu Folie 96:

- ➔ Kollisionen werden durch Messung des Spannungspegels auf der Leitung erkannt. Falls mehrere Stationen gleichzeitig senden, überlagern (addieren) sich deren elektromagnetische Wellen auf der Leitung, wodurch sich der Spannungspegel (verglichen mit der Situation, daß nur eine Station sendet) erhöht.
- ➔ Das JAM-Signal dient dazu, die Kollision zu verlängern, damit auch alle anderen Stationen die Chance haben, die Kollision zu erkennen.

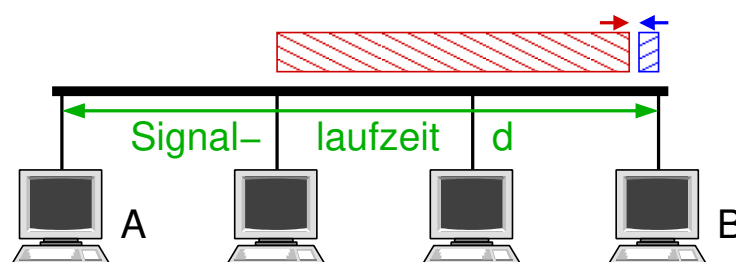
Auch ein Hub, der eine Kollision erkennt (weil an mindestens zwei Eingängen ein Signal empfangen wird), gibt ein JAM-Signal an alle Ausgänge weiter.

Exponential Backoff-Strategie

- ➔ Aufgabe: Bestimmung der Wartezeit zwischen Kollision und erneutem Sendeversuch
- ➔ Ziel: erneute Kollision möglichst vermeiden
- ➔ Vorgehensweise:
 - c = Anzahl der bisherigen Kollisionen für aktuellen Frame
 - warte $s \cdot 51,2 \mu s$ (bei 10 Mb/s), wobei s wie folgt bestimmt wird:
 - falls $c \leq 10$: wähle $s \in \{ 0, 1, \dots, 2^c - 1 \}$ zufällig
 - falls $c \in [11, 16]$: wähle $s \in \{ 0, 1, \dots, 1023 \}$ zufällig
 - falls $c = 17$: Abbruch mit Fehler
 - damit: neue Sendeversuche zeitlich entzerrt, Wartezeit an Netzlast angepaßt

(Animierte Folie)

Kollisionserkennung: Worst-Case Szenario



- ➔ $t = 0$: A beginnt, einen Frame zu senden
- ➔ $t = d - \epsilon$: B beginnt ebenfalls, einen Frame zu senden
- ➔ $t = d$: A's Frame kommt bei B an \Rightarrow Kollision!
- ➔ $t = 2 \cdot d$: B's (Kollisions-)Frame kommt bei A an
 - wenn A zu diesem Zeitpunkt nicht mehr sendet, erkennt A keine Kollision



Sichere Kollisionserkennung

- ➔ Um Kollisionen immer erkennen zu können, definiert Ethernet:
 - maximale RTT: 512 Bit-Zeiten
 - 51,2 μ s bei 10 Mb/s, 5,12 μ s bei 100 Mb/s
 - legt maximale Ausdehnung des Netzes fest, z.B. 200m bei 100BASE-TX mit Hubs
 - minimale Framelänge: 512 Bit (64 Byte), zzgl. Präambel
 - kleinere Frames werden vor dem Senden aufgefüllt
- ➔ Das stellt im Worst-Case Szenario sicher, daß Station A immer noch sendet, wenn B's Frame bei ihr ankommt
 - damit erkennt auch Station A die Kollision und kann ihren Frame wiederholen

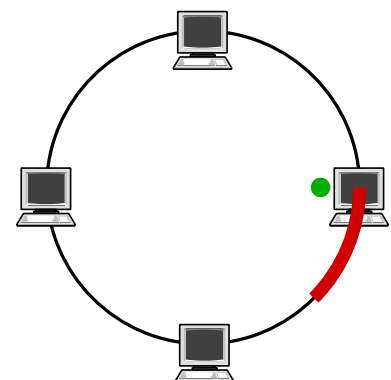
3.6.3 MAC im Token-Ring

OSI: 2



(Animierte Folie)

- ➔ Netze mit Ringtopologie, z.B. IBM Token-Ring, FDDI
- ➔ Das Token (spezielle Bitfolge) umkreist den Ring
- ➔ Ein Knoten, der senden will, kann das Token „ergreifen“ und dann senden
 - Frame umkreist den Ring und wird vom Sender wieder entfernt
 - jeder Knoten reicht den Frame weiter
 - der Empfänger macht sich eine Kopie
- ➔ Das Token kann nur für bestimmte Zeit behalten werden
 - danach muß der Knoten das Token wieder freigeben





Diskussion

- ➔ Mit gegebener Token-Haltezeit THT kann jeder Knoten garantiert nach Ablauf der Zeit
$$TRT \leq \text{Ringlatenz} + (\text{AnzahlKnoten} - 1) \cdot THT$$
seinen Frame senden
- ➔ Daher: Eignung für Realzeitanwendungen
- ➔ Mit CSMA/CD sind keine derartigen Garantien möglich
- ➔ Andererseits: bei unbelastetem Netz kann ein Knoten bei CSMA/CD immer sofort senden

3.7 Zusammenfassung



- ➔ Hardware: Knoten, Leitungen (Kupfer, Glasfaser, Funk)
- ➔ Codierung und Modulation
 - ➔ Umsetzen des Bitstroms in ein elektrisches Signal
 - ➔ wichtig: Taktwiederherstellung
- ➔ Framing: Erkennung des Anfangs / Endes eines Datenblocks
- ➔ Fehlererkennung (und -korrektur)
- ➔ Medienzugriffssteuerung (MAC)
 - ➔ Ethernet (CSMA-CD)
 - ➔ Token-Ring: garantierte maximale Sendeverzögerung

Nächste Lektion:

- ➔ Paketvermittlung, LAN-Switching