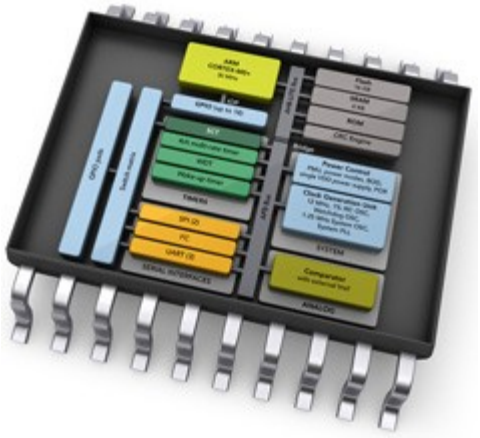
FH AACHEN
UNIVERSITY OF APPLIED SCIENCES

Kommunikationssysteme

(Modulcode 941306)

Prof. Dr. Andreas Terstegge

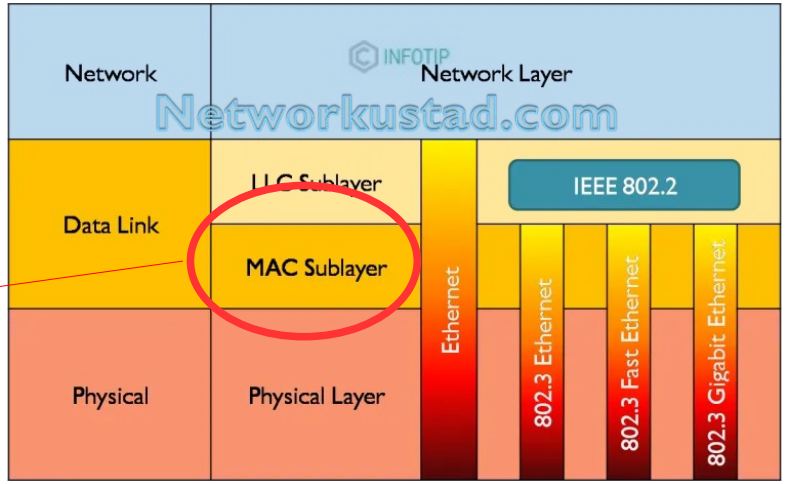


Schichten 1 und 2 im ISO/OSI Referenzmodell und im TCP/IP Referenzmodell

	OSI		TCP / IP	
Anwendungsschicht	7	APPLICATION	PROCESS / APPLICATION	- Anwendung - User Interface
Darstellungsschicht	6	PRESENTATION		
Kommunikations-steuerungsschicht	5	SESSION		
Transportschicht	4	TRANSPORT	HOST-TO-HOST	- Paketbildung - Kennzeichnung der Quell- und Zielapplikation
Vermittlungsschicht	3	NETWORK	INTERNET	- Adressierung und Routing
Sicherungsschicht	2	DATALINK	NETWORK ACCESS	- Umwandlung IP <-> MAC (ARP) - Physische Übertragung
Physische Verbindung (Bitübertragungsschicht)	1	PHYSICAL		

Medium **A**ccess **C**ontrol:

Welcher Kommunikationsteilnehmer kann wann und wie lange das Medium nutzen?



Mehrfachzugriffsprotokolle

Zwei Grundtypen von Verbindungsleitungen:

- Punkt-zu-Punkt
  - PPP für Einwahlverfahren mit Modem/ISDN
  - Verbindungsleitung zwischen einem Ethernet-Switch und Rechner

- **Broadcast (geteiltes Medium)**
  - Traditionelle Ethernet
  - 802.11 Wireless LAN

Mehrfachzugriffsprotokolle

Verteilter Algorithmus zur Regelung der gemeinsamen Nutzung eines geteilten Übertragungsmediums

- **Wann** kann **wer** etwas senden?
- Störungsfreier Kanalzugriff

Hierbei ist zu berücksichtigen, dass die zur Koordination notwendige Kommunikation auch über das geteilte Medium abgewickelt werden muss!

- **Kontrollnachrichten**
- Keine out-of-band-Signalisierung

Eine Ideales Mehrfachzugriffsprotokoll

Broadcast-Medium mit Maximalrate R bps

1. Ein einzelner Knoten kann mit der Rate R übertragen
2. M Knoten können mit einer mittleren Rate von R/M übertragen
3. Das Protokoll ist dezentral, d.h. es gibt keine Master-Knoten, die ausfallen und das ganze System zum Absturz bringen können
4. Das Protokoll ist einfach und kann somit kostengünstig implementiert werden

MAC-Protokolle: Eine Taxonomie

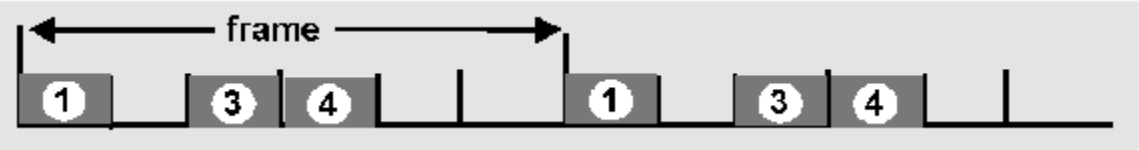
Drei grobe Kategorien:

- **Kanalaufteilungsprotokolle (Multiplexing)**
  - Aufteilung des Übertragungskanals in kleine Übertragungseinheiten (Zeitfenster, Frequenzen, Kodierung)
  - Exklusive Nutzung der Übertragungseinheiten für einzelne Knoten
- **Zufallszugriffsprotokolle**
  - Keine Unterteilung des Kanals, jeder überträgt mit der gesamte Leitungskapazität. Allerdings sind Kollisionen von Nachrichten möglich und müssen korrekt behandelt werden
- **Rotationsprotokolle**
  - Der Zeitpunkt, wann ein Knoten senden kann wird durch eine spezielle Koordinierung nach einem flexiblen Rotationsprinzip im Übertragungsmedium ausgetauscht.

Kanalaufteilungsprotokolle: Zeitmultiplexing

TDMA: Time Division Multiple Access

- Aufteilung des Kanals in kleine Zeiteinheiten, die sich in Runden wiederholen
- Jeder Knoten kann eine bestimmte Anzahl an Zeiteinheiten in jeder Runde Nutzen
- Ungenutzte Einheiten (slots) bleiben ungenutzt
- Beispiel: 6-Knoten LAN, 1,3,4 haben Daten, Slots 2,5,6 idle

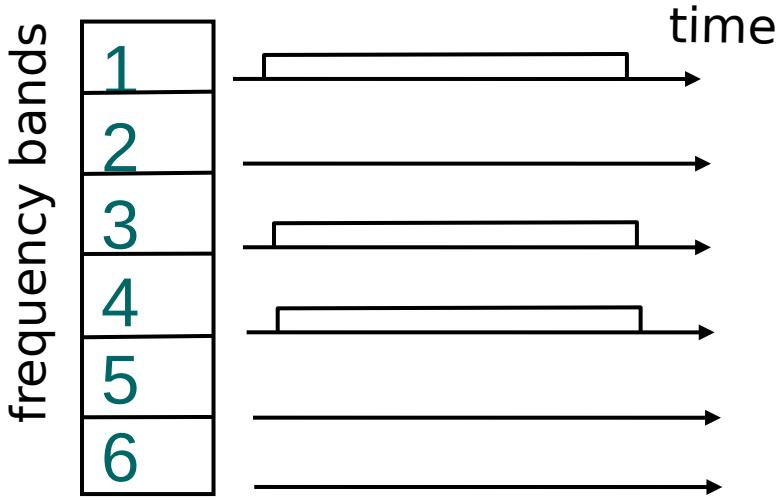


- Die Zeiteinheiten können meist auch gruppiert angefordert werden
- Anforderung entspricht Konfiguration der NIC-Karte

Kanalaufteilungsprotokolle: Frequenzmultiplexing

FDMA: Frequency Division Multiple Access

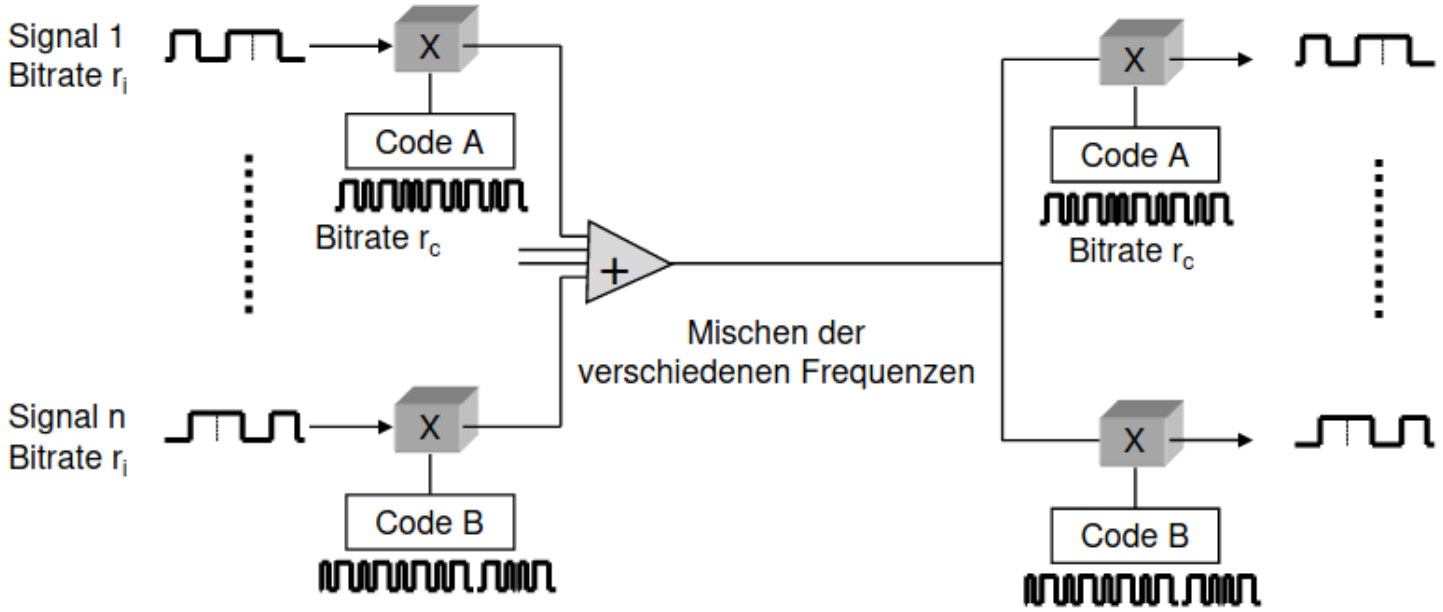
- Aufteilung des Kanals in Frequenzbänder
- Nach Nyquist kann ein rauschfreier Kanal mit einem Frequenzband der Breite x Hz binäre Signale nicht mit mehr als 2x Bps übertragen (Nyquist-Bandbreite)
- Jede Station nutzt ein Frequenzband
- Die Kapazität nicht genutzter Frequenzbänder geht verloren
- Beispiel: 6-Knoten LAN, 1,3,4 haben Daten, 2,5,6 idle



Kanalaufteilungsprotokolle: Frequenzmultiplexing

CDMA: Code Division Multiple Access

- Keine Unterteilung in Zeitslots oder Frequenzen
- Nutzung von ‚orthogonalen Codes' bei der (De-) Modulation, die sich im Medium überlagern/mischen, aber trotzdem auf Empfängerseite eindeutig wiederhergestellt werden können
- Vorteil: Es gibt keine ungenutzten Resourcen wie bei TDMA/FDMA



Zufallszugriffsprotokolle

Sendet ein Knoten, so
- nutzt er die vollständige Kanalrate *R*.
- Keine a priori Koordination

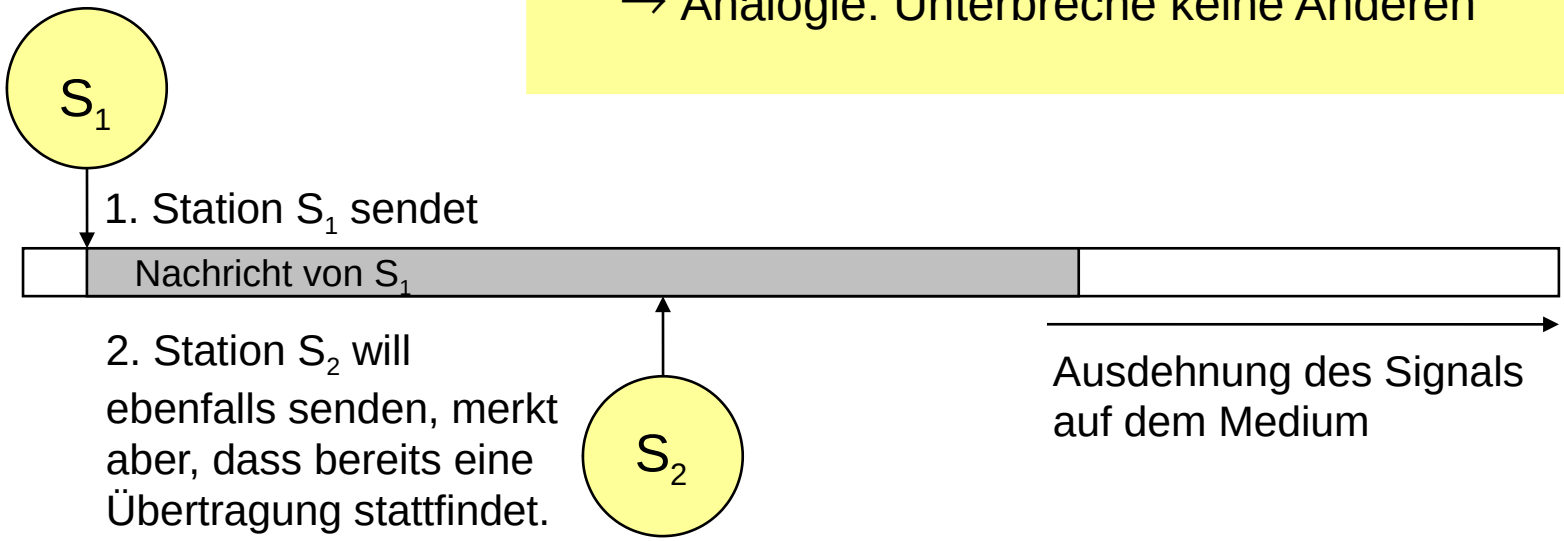
Kollisionen treten auf, wenn mehrere Knoten gleichzeitig senden:
- Wie werden diese erkannt?
- Berücksichtigung der Signal-Ausbreitungsgeschwindigkeit!
- Wie werden diese behandelt (z.B. erneutes Verschicken nach einem zufälligen Zeitintervall) via ‚delayed retransmissions'

Beispielprotokolle:
- slotted ALOHA
- ALOHA
- **CSMA**, CSMA/CD, CSMA/CA

CSMA

Carrier Sense Multiple Access (**CSMA**)

Prinzip:
→ höre vor der Übertragung das Medium ab
→ sende nur, falls das Medium frei ist
→ Analogie: Unterbreche keine Anderen

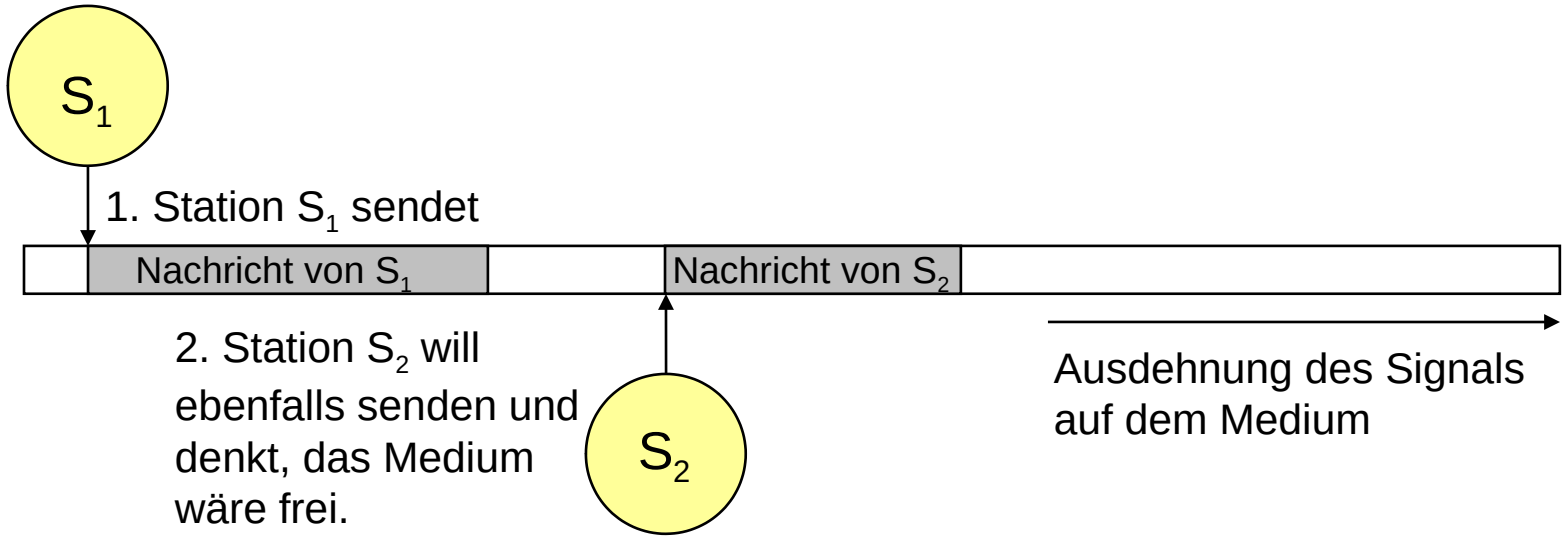


Vorteile: einfach, da die Stationen nicht koordiniert werden müssen; mit einigen Erweiterungen trotzdem gute Ausnutzung der Netzkapazität

Nachteil: kein garantierter Zugriff, es ist eine große Verzögerung möglich
→ keine Echtzeitfähigkeit!

Problem bei CSMA

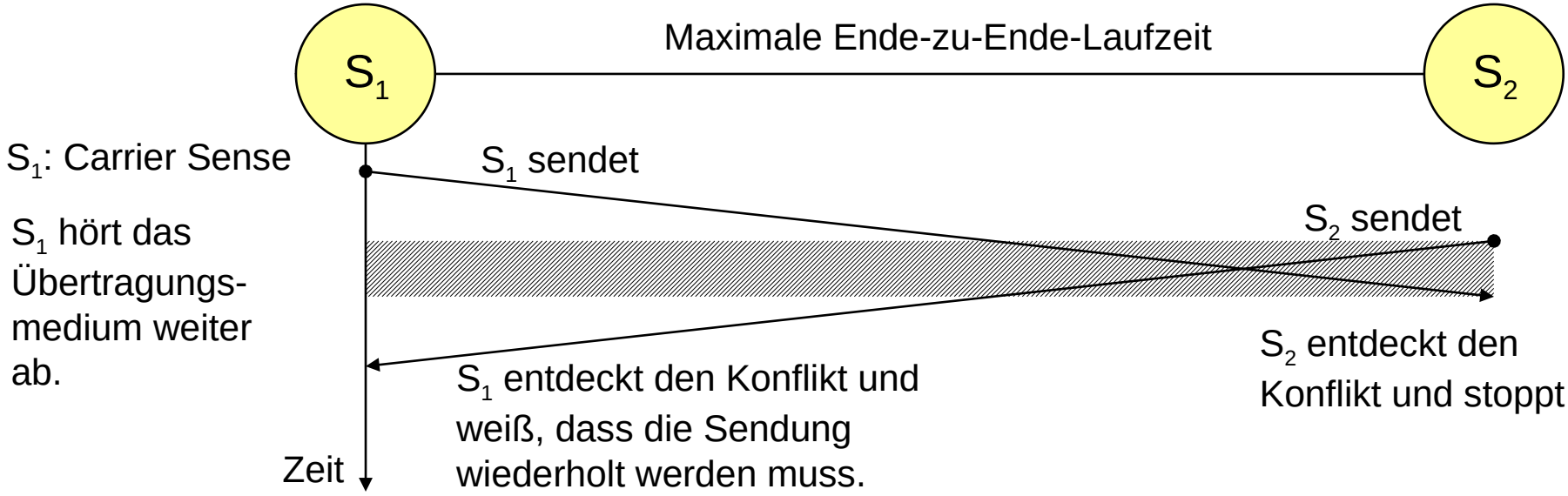
Problem: die Nachricht, die S_1 sendet, breitet sich mit endlicher Geschwindigkeit auf dem Medium aus. Daher kann es sein, dass S_2 denkt, das Medium wäre frei, obwohl S_1 schon mit der Sendung begonnen hat. Beide Nachrichten überlagern sich auf dem Medium und werden unbrauchbar.



Prüfe auf Kollision (CD)

Carrier Sense Multiple Access with *Collision Detection* (**CSMA/CD**)

- → wie CSMA, zusätzlich: höre *während der Sendung das Medium weiter ab* und breche die Übertragung ab, wenn eine Kollision auftritt
- → sende ein Jamming-Signal, damit jede Station weiß, dass eine Kollision aufgetreten ist und die Nachricht nutzlos geworden ist.
- → Wichtig: Man muss so lange senden, dass man ein Jamming -Signal während dessen auch mitbekommt



Anmerkung: mit zunehmender Ausdehnung des Netzes steigt die Gefahr eines Konflikts. Daher ist diese Technik nur für "kleine" Netze geeignet.

Ethernet

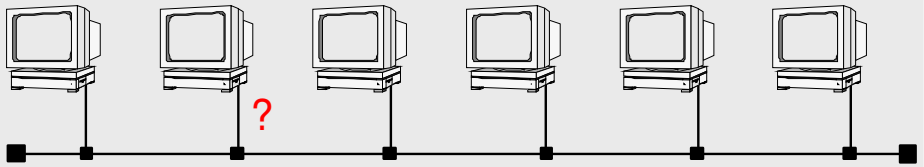
basiert auf dem Standard IEEE 802.3 “**CSMA/CD**”

(Carrier Sense Multiple Access/Collision Detection)

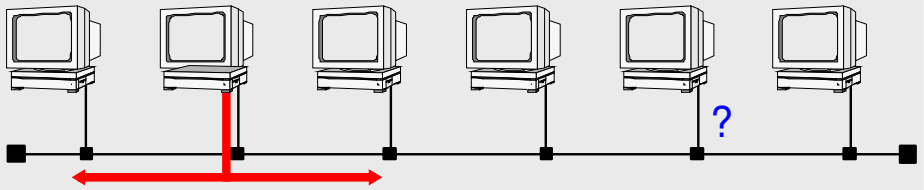
mehrere (passive) Rechner - ein gemeinsames Medium (Random Access)

ursprünglich Bustopologie:

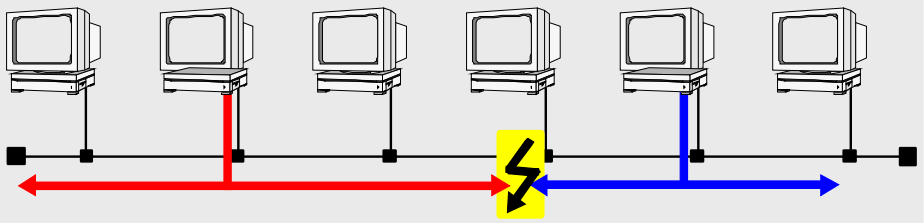
1. Ist das Medium frei?
   (Carrier Sense)



2. Übertragen



3. Prüfe auf Kollision (Collision Detection)
Falls ja: sende Jamming und breche ab.
Danach weiter mit Binary Exponential Backoff Algorithmus



Vorgehen bei Kollisionen

Problem:
Wenn zwei Nachrichten eine Kollision haben, versuchen die Stationen eine erneute Übertragung. Wann sollen die Stationen die Wiederholung starten?

Binary Exponential Backoff

Um nach einer Kollision die gleichzeitige Wiederholung der kollidierten Sendungen zu vermeiden (Folgekollision), wird eine zufällige Wartezeit aus einem vorgegebenen Intervall gezogen. Das Intervall wird *klein* gehalten, um große Wartezeiten bis zur Widerholung zu vermeiden. Dadurch ist allerdings das Risiko eines Folgekonflikts groß. Kommt es so zu einer weiteren Kollision, wird das Intervall vor dem nächsten Versuch vergrößert, um mehr Spielraum für alle sendenden Parteien zu schaffen.

Die Wartezeit wird dabei folgendermaßen ermittelt:

- Hatte eine Station bereits i Kollisionen, würfelt sie eine Zahl x aus dem Intervall $[0, 2^i-1]$ (bei Kollision 10 - 15 bleibt das Intervall fix bei $[0, 2^{10}-1]$, nach der 16. Kollision erfolgt ein Abbruch)
- Sobald das Medium frei ist, wartet der Sender x Zeitslots, wobei ein Zeitslot der minimalen Ethernet-Rahmenlänge von 512 Bit, also bei 10 Mbit/s 51.2 µs, entspricht.
- Nach dem x-ten Zeitslot beginnt die Station erneut mit Carrier Sense.

Datenübertragung bei Ethernet

- Verwendet CSMA/CD, bei Kollisionen wird das Binary Exponential Backoff-Verfahren angewendet
- Ziel: Realisieren großer Netzwerke, die aber trotzdem eine geringe Kollisionswahrscheinlichkeit haben...

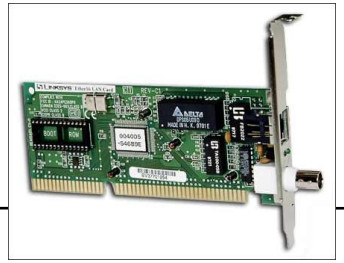
  ... *Resultat*: die maximale Ausdehnung wird auf **2500 Meter** festgelegt

  Die maximale Zeit zur Entdeckung einer Kollision ist knapp zweimal so lang wie die Signallaufzeit auf dem Medium.

  Bei einer Signalgeschwindigkeit von ungefähr 100000 km/s (10 µs/km) erhält man (unter Berücksichtigung der Zeit in 4 Repeatern) eine maximale Ende-zu-Ende-Signallaufzeit von 25 µs. Die maximale Konfliktdauer ist damit ca. **50 µs**. Um einen Konflikt mit Sicherheit zu erkennen, muss die Station mindestens 50 µs auf dem Medium horchen und senden → bei 10MBit/s mindestens 500Bit

  Darauf basierend wurde für eine Senderate von 10 MBit/s eine *minimale Rahmenlänge* (aufgerundet **64 Byte**) definiert, um eine Kollisions-erkennung auch im Wort-Case zu ermöglichen!

Entwicklung des Ethernet



- 70er Jahre: experimentelles Netzwerk auf Basis von Koaxialkabeln, Datenrate von 3 Mb/s. Entwickelt von der Xerox Corporation als ein Protokoll für LANs mit sporadischem aber burst-artigem Verkehrsverhalten.
- 1980: gemeinsame Weiterentwicklung durch die Digital Equipment Corporation, die Intel Corporation und Xerox zu einer 10 Mb/s-Variante.
- Original Ethernet-Struktur: Bus-Topologie mit einer maximalen Segmentlänge von 500 Metern, Anschlussmöglichkeit für maximal 100 passive Stationen. Repeater werden verwendet, um mehrere Segmente zusammenzuschließen.
- Gängigstes Medium: Kupferkabel. Aber auch Glasfaser-Kabel kommen zum Einsatz (erhöht die Segmentlänge).
- Frühe 90er Jahre: die Bus-Topologie wird mehr und mehr von einer Stern-Topologie verdrängt, bei der ein zentraler Switch Punkt-zu-Punkt-Verbindungen (auf Twisted-Pair oder Glasfaserkabel basierend) zu allen Stationen realisiert. Der Switch bietet hierbei den Vorteil, dass mehrere Verbindungen parallel ablaufen können.

Ethernet

Basiert auf IEEE 802.3 „CSMA/CD“
4 Klassen von Ethernet-Varianten:

- Standard Ethernet → 10 Mb/s
- Fast Ethernet → 100 Mb/s
- **Gigabit Ethernet** → 1000 Mb/s
- 10Gigabit-Ethernet → 10000 Mb/s

Kaum noch im Einsatz

Ehemals meistverbreitete Variante

Heute meist verbreitet

Standardisiert

Ethernet hat sich im LAN-Bereich durchgesetzt. Es wird in der überwiegenden Anzahl der LANs als Infrastruktur eingesetzt:

- Es ist einfach zu verstehen, umzusetzen und zu überwachen
- Das Netzwerk ist in der Anschaffung **billig**
- Bringt **Flexibilität** bzgl. der Topologie mit sich
- Es gibt **keine Kompatibilitäts-Probleme**, jeder Hersteller kennt und befolgt den Standard
- Das Netz ist prinzipiell nicht echtzeitfähig...

Daten zu Ethernet

Parameter	Ethernet	Fast Ethernet	Gigabit Ethernet
Maximale Ausdehnung	bis zu 2800 Meter	205 Meter	200 Meter
Kapazität	10 Mb/s	100 Mb/s	1000 Mb/s
Minimale Rahmenlänge	64 Byte	64 Byte	520 Byte
Maximale Rahmenlänge	1526 Byte	1526 Byte	1526 Byte
Signal-darstellung	Manchester-Code	4B/5B-Code, 8B/6T-Code, ...	8B/10B, ...
Maximale Anzahl Repeater	5	2	1

Zusätzlich:
Ein weiterer „Rahmen“ wird zum „Jamming“ benutzt:
Dies ist ein 4-Byte-Störsignal zur Kollisionserkennung

Nachrichtenformat bei Ethernet

Ein Ethernet-Frame im Kontext mit maximalen IPv4- / TCP-Daten

Schicht 4: TCP-Segment								TCP-Header	*Nutzlast (1460 bytes)*		
Schicht 3: IP-Paket							IP-Header	*Nutzlast (1480 bytes)*			
Schicht 2: Ethernet-Frame			**MAC-Empfänger**	**MAC-Absender**	**802.1Q-Tag (opt.)**	**EtherType**	***Nutzlast (1500 bytes)***			**Frame Check Sequence**	
Schicht 1: Ethernet-Paket+IPG	Präambel	Start of Frame	*Nutzlast (1518/1522 bytes)*								Interpacket Gap
Oktette	7	1	6	6	(4)	2	20	20	(6–)1460	4	12

1 2 3 4 5 6 7 8

1: *7 Byte Synchronisation*
Jedes Byte beinhaltet 10101010

2: *1 Byte Start Frame Delimiter*
Markierung des Rahmenbeginns durch das Byte 10101011

3: *6 Byte Destination Address*

4: *6 Byte Source Address*

5: *2 Byte Length/Type*
Angabe der Länge des Datenfelds bzw. Typ des Schicht-3-Protokolls

6: *(0-1500) Byte Daten*

7: n *Byte Padding*
Auffüllen des Rahmens auf mindestens 64 Byte (Kleinere Fragmente werden im Netz verworfen, abgesehen von dem Jamming-Signal)

8: *4 Byte Frame Check Sequence*
Verwendung eines zyklischen Codes

Der Ethernet-Rahmen

- *Präambel* : kennzeichnet eine folgende Übertragung und synchronisiert den Empfänger mit dem Sender.
- Der *Start-of-Frame-Delimiter* : (bzw. die beiden aufeinanderfolgenden Einsen) zeigen an, dass endlich Daten folgen.
- *Destination Address* : das erste Bit kennzeichnet den Empfänger: entweder eine einzelne Station (1. Bit = 0) oder eine Gruppenadresse (1. Bit = 1; Broadcast ist auch hier durch 11...1 gegeben).
- *Length/Type:* Bei einem Wert bis 1500 wird die Angabe als Länge des Datenteils aufgefasst (dies ist der Fall beim so genannten CSMA/CD), bei einem Wert ab 1536 wird hier angegeben, an welches Schicht-3-Protokoll die Daten weitergegeben werden sollen (verwendet bei Ethernet).
- *FCS* : Checksumme, 32-Bit CRC. Diese erstreckt sich über die Felder DA, SA, Length/Type, Data/Padding.

Ethernet-Varianten

Angabe der verwendeten Ethernet-Variante durch 3 Namenskomponenten:

1 – Kapazität in Mb/s (also 10, 100, 1000 oder 10G)

2 – Übertragungstechnik (z.B. **Base** für Basisband, Broad für Broadband)

3 – maximale Segmentlänge in Einheiten von 100 Metern, bzw. Art des verwendeten Mediums

Beispiele:

- 10Base-T: 10 Mb/s, Basisband, Twisted-Pair-Kabel
- 100Base-T2: 100 Mb/s, Basisband, 2 Twisted-Pair-Kabel
- 1000Base-X: 1000 Mb/s, Basisband, Glasfaser-Kabel

Von der Variante hängen noch Parameter wie z.B. die minimale Rahmenlänge ab (wegen unterschiedlicher Signallaufzeiten):
1000Base-X: minimale Rahmenlänge 416 Bytes
1000Base-T: minimale Rahmenlänge 520 Bytes

Fast Ethernet

Prinzip: behalte Ethernet bei, aber mache es schneller:

- Kompatibilität mit existierenden Ethernet-Netzen
- 100 MBit/s an Übertragungsrate, erreicht durch bessere Technik, effizientere Codes, Nutzung mehrerer Leitungspaare, Switches, ...
- Resultat: IEEE 802.3u, 1995

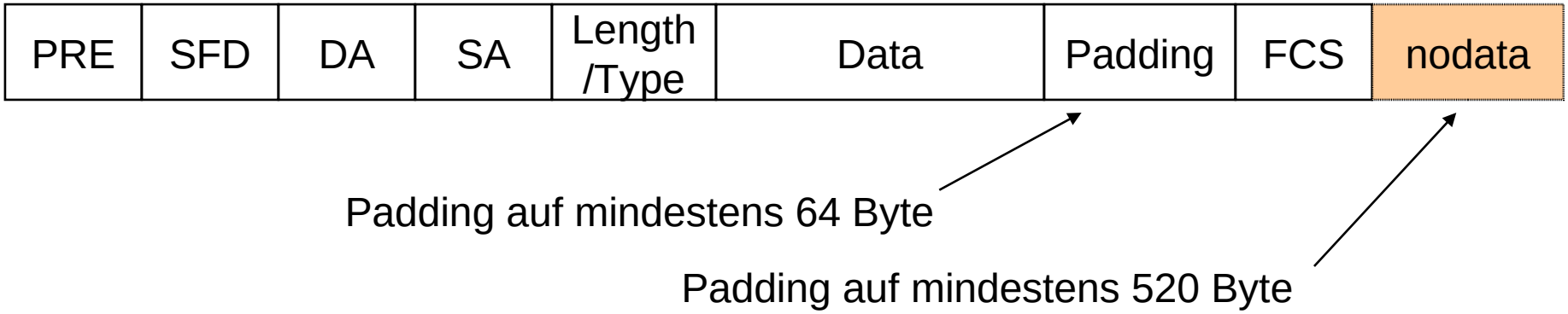
Problem:

- Die minimale Rahmenlänge zur Kollisionserkennung bei Ethernet beträgt 64 Byte. Bei 100 Mb/s wird der Rahmen aber ca. 10 Mal so schnell abgesendet, so dass eine Kollisionserkennung nicht mehr gewährleistet ist.
- *Resultat*: **für Fast Ethernet musste die Ausdehnung ca. um den Faktor 10 auf etwas mehr als 200 Meter reduziert werden**...

Wechsel zu Gigabit-Ethernet

Gigabit-Ethernet (IEEE 802.3z, 1998)

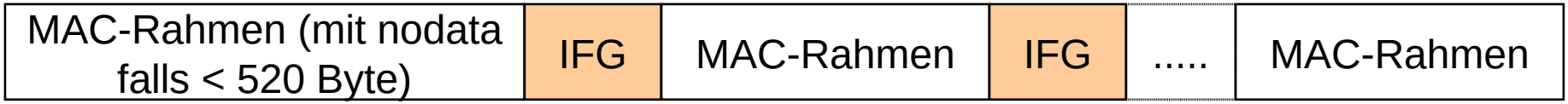
- Kompatibilität zu Fast Ethernet beibehalten!
- Problem: zur Kollisionserkennung wäre eine **Reduktion** der Ausdehnung **auf 25m notwendig**…
  > Daher: **Ausdehnung von Fast Ethernet beibehalten (200m)**
  > Neue minimale Rahmenlänge von 520 Byte
  > Trotzdem Beibehaltung des alten Rahmenformats: füge zweites Padding-Feld hinter dem Rahmen an (**Carrier Extension**)



Wechsel zu Gigabit-Ethernet

Auch maximale Rahmenlänge unzureichend

- Ermögliche Versenden mehrerer aufeinanderfolgender Rahmen (**Frame Bursting**) ohne wiederholt CSMA/CD anzuwenden
- Versendung von bis zu 5 Rahmen in Folge, getrennt durch „Inter-Frame Gaps“ (IFGs)
  > Spezielles Bitmuster, Medium bleibt für andere Stationen belegt



- Im Normalfall werden nur noch Switches eingesetzt
  > Es treten keine Kollisionen mehr auf
  > Maximale Kabellänge nur noch durch die Signalabschwächung bestimmt
  > Trotzdem ist CSMA/CD noch implementiert (half duplex erlaubt!)

1000Base-T/X (Gigabit Ethernet)

1000Base-T

- Basiert auf Fast Ethernet (mindestens UTP Kat. 5)
- Nutzung aller 4 Kabelpaare mit quaternärem Code, Kabellänge: 100 m

1000Base-SX

- Multimode-Glasfaser mit 550 m Segmentlänge
- Übertragung auf dem 850nm-Band

1000Base-LX

- Übertragung auf 1300nm
- Mono- oder Multimode mit Reichweite bis 5000 m

1000Base-LH

- Monomode Übertragung auf 1550nm
- Reichweite bis zu 70 km

Neuere Entwicklungen beim Ethernet

10-Gigabit Ethernet (IEEE 802.3ae)

- Erst nur für Glasfaser spezifiziert (LX oder SX), bis 40km
- Mittlerweile auch für Twisted Pair (Kat. 6 oder 7)
  > Wüste Trickserei auf Schicht 1, um trotz Dämpfung und Verzerrung 50 bzw. 100m Reichweite zu erreichen
- Verzicht auf CSMA/CD, da sowieso keine Kollisionen mehr auftreten können (Sterntopologie mit Switch)

Mittlerweile

- 40G-Ethernet, 100G-Ethernet (802.3ba)
  > Beibehaltung der Rahmenformate...
  > 7 Meter über Twisted Pair, bis zu 40 km über Glasfaser

Gibt es auch Alternativen zu Ethernet?

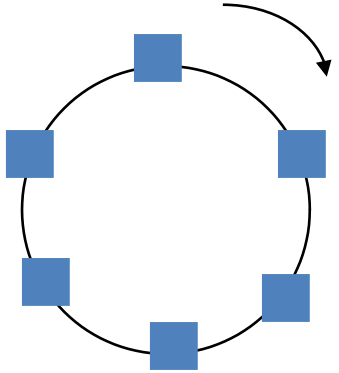
Token Ring (IEEE 802.5)

Ursprüngliche Konkurrenz zu Ethernet

- “Token“-Verfahren, nur wer ein bestimmtes ***Token*** (=Bitfolge) besitzt, darf senden
- Die Rechner teilen sich einen Ring aus Punkt-zu-Punkt-Verbindungen
- Das Token wird zyklisch weitergegeben

Eigenschaften:

- Garantierter Zugriff, keine Kollisionen
- Sehr gute Ausnutzung der Netzkapazität,
- hohe Effizienz
- Fair, garantierte Antwortzeiten
- Aber: aufwändig und teuer

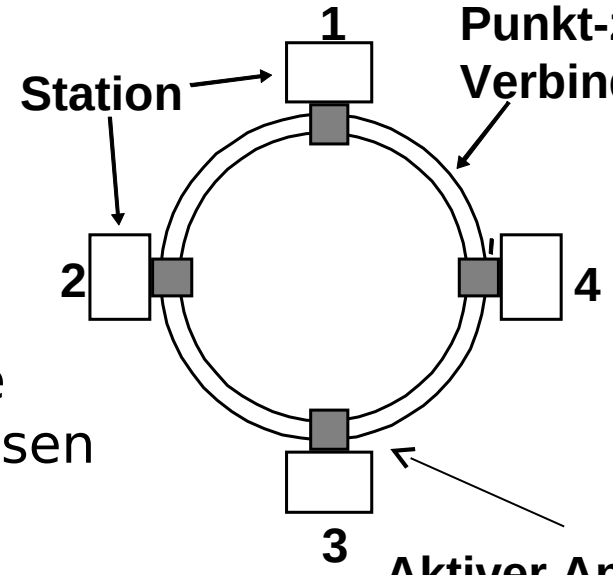


Weitergabe des Tokens

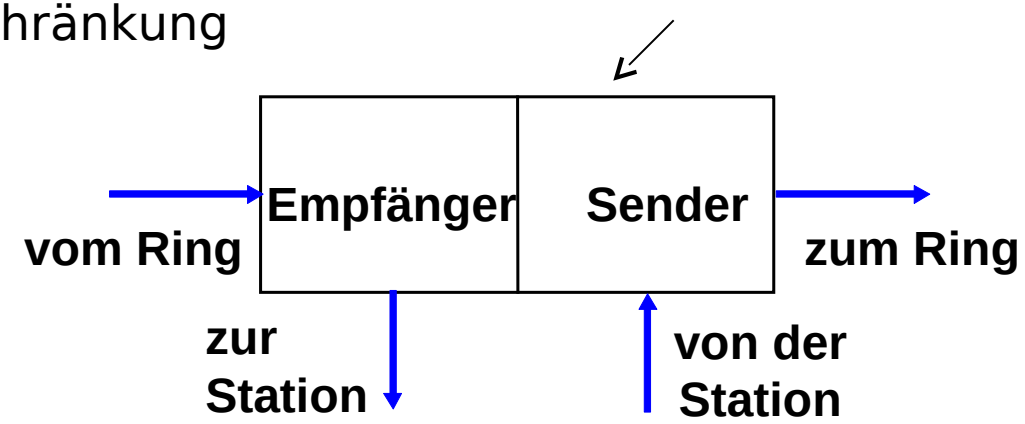
Token Ring

Parameter

- Twisted Pair, Koaxialkabel oder Glasfaser
- Datenrate von 4 bzw. 16 Mbit/s (100MBit/s mit Glasfaser)
- Differentieller Manchester-Code
- Stationen sind aktiv angeschlossen
- Empfangene Signale werden regeneriert (wie bei Repeatern, daher prinzipiell keine Beschränkung der Ausdehnung)



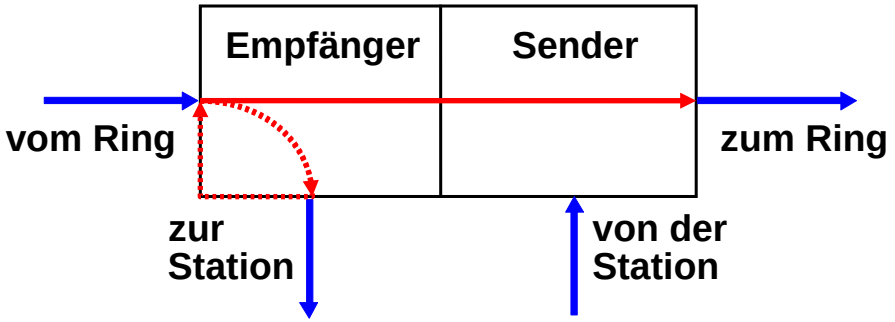
Media Access Unit (MAU)



Senden und Empfangen

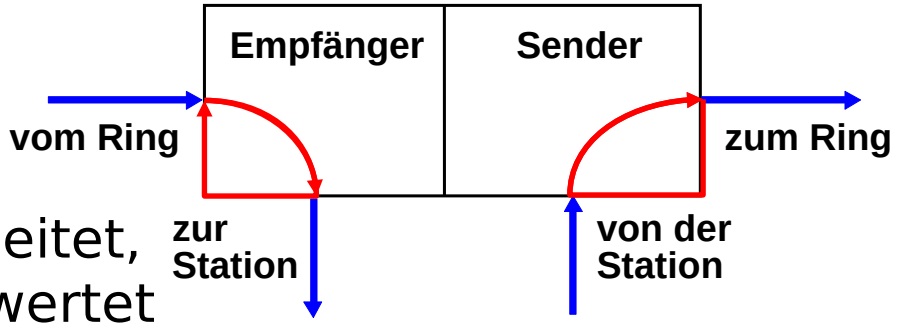
Grundzustand

- Daten werden vom Ring seriell empfangen
- An Station gerichtete Daten
- werden kopiert
- Daten werden seriell
- weitergeleitet



Sendezustand

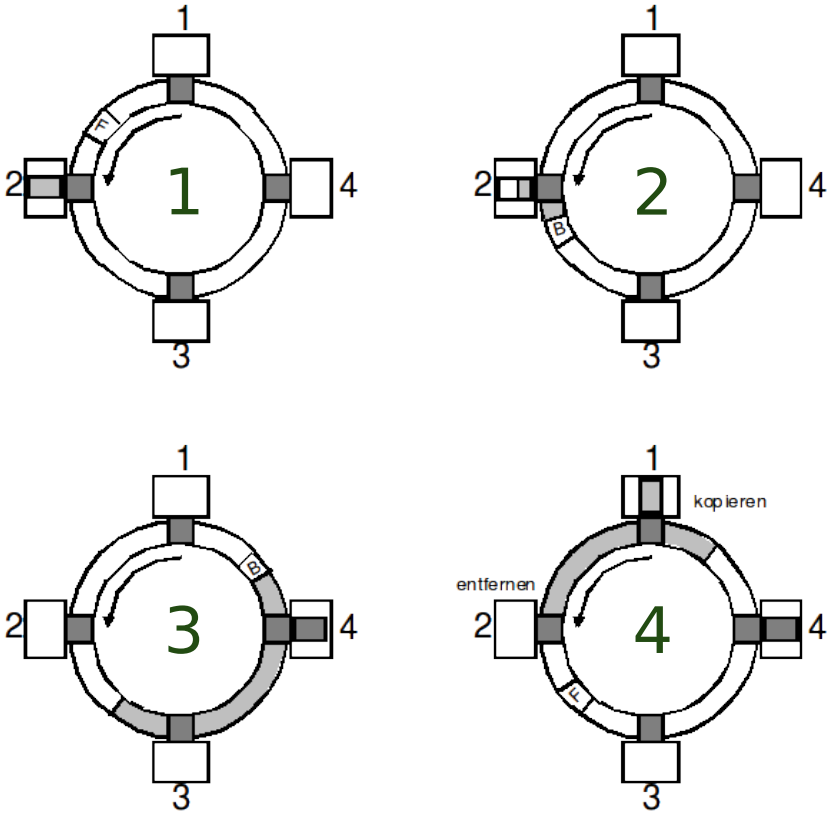
- Der Ring wird aufgetrennt
- Eigene Daten werden
- seriell gesendet
- Vom Ring kommende
- Daten werden nicht weitergeleitet,
- sondern in der Station ausgewertet



Zugriff beim Token Ring

Beispiel: Station 2 sendet an Station 1

1. Station 2 wartet auf freies Token (Sendeberechtigung, 3-Byte-Token)
2. Station 2 wandelt freies Token in belegtes um (= Rahmen-Header); danach sendet 2 den Rahmen (und ggfs. weitere Rahmen, für maximal 10 ms)
3. Station 2 beendet den Rahmen und wartet, bis dieser wieder bei ihr ankommt
4. Station 1 kopiert den Rahmen; Station 2 entfernt ihn vom Ring und erzeugt ein neues, freies Token



Token Ring als Alternative?

Token Ring WAR eine Konkurrenz zu Ethernet:

- Fair, garantierte Antwortzeiten, keine Kollisionen
- Aber: Verwaltungsaufwand! Was passiert, wenn durch einen Übertragungsfehler das Token verfälscht wird?

→ Ethernet ist **DER** Standard für lokale Netze

Fazit: Ethernet

Standardnetz für LANs

- Vielzahl von Varianten, flexible Topologie

  > Heutzutage vorwiegend Stern, teilweise erweitert zu Bäumen

- Einschränkungen durch Beibehaltung der Kompatibilität zu älteren Varianten

  > Rahmenformat

  > Kollisionserkennung heute kaum noch nötig

- Oft auch im MAN verwendet

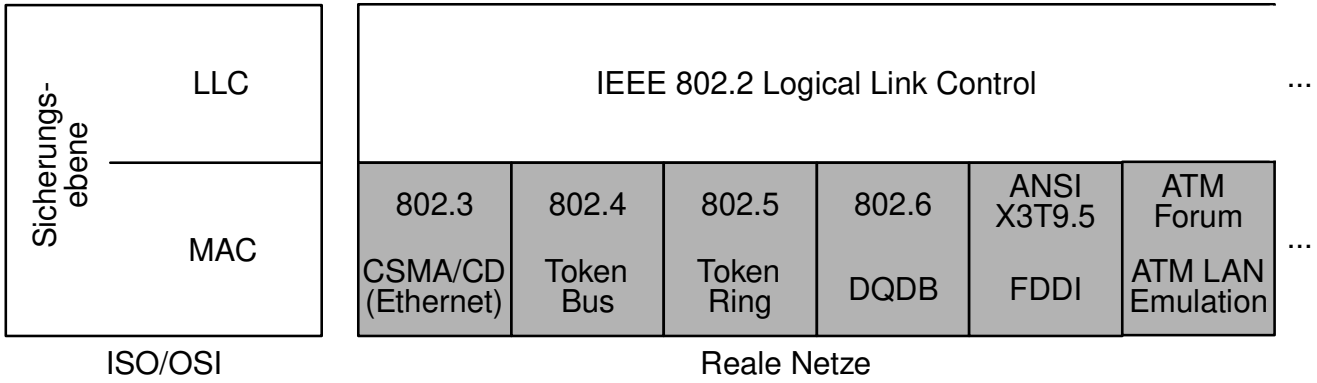
Wird durch weitere Technologien z.B. im Bereich Funkübertragung und Weitverkehrsnetze ergänzt:

- WLAN im drahtlosen Bereich
- DSL zum Anschluss von Heimnetzen an das Internet
- SDH im Weitverkehrsbereich

Schicht 2: Aufteilung in zwei Aufgabenbereiche

- **Logical Link Control (LLC)** (Schicht 2b)
  - → Einteilung der zu sendenden Daten in *Rahmen (Frames)*
  - → Multiplexing mehrerer Layer-3 Protokolle über das gleiche Medium
  - → Bereitstellung einer (möglichst) fehlerfreien Übertragung zwischen benachbarten Knoten durch:
    - → *Erkennung (und Behebung) von **Übertragungsfehlern***
    - → *Flusskontrolle* (Vermeidung der Überlastung des Empfängers)
    - → *Pufferspeicher*

- **Medium Access Control (MAC)** (Schicht 2a)
  - → Regelung des Zugriffs auf den Kommunikationskanal bei Broadcast-Netzen



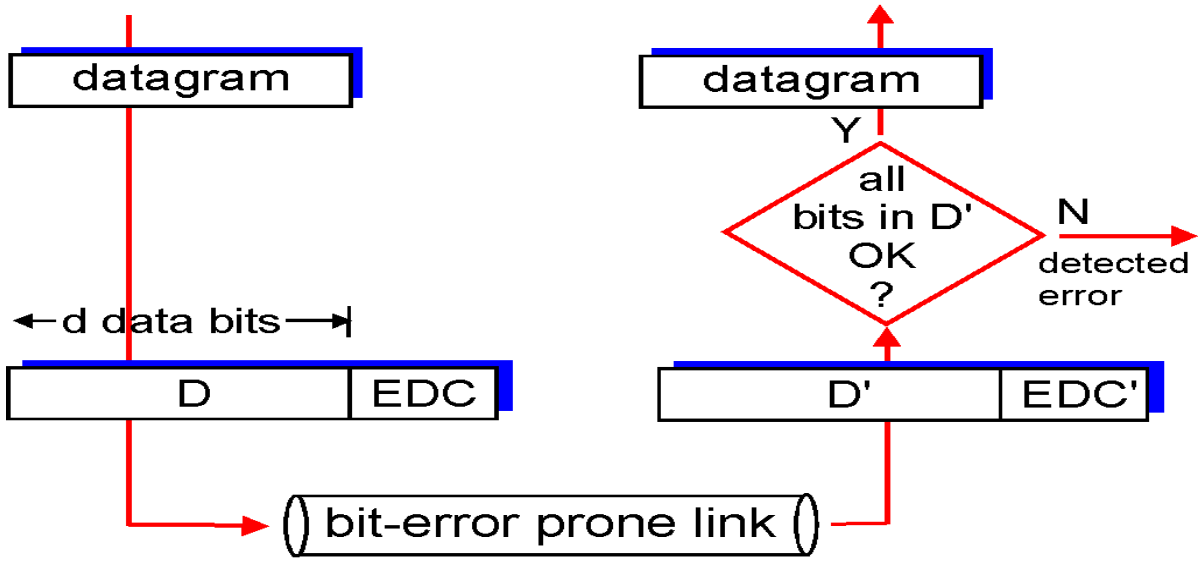
Sicherungsschicht

Wie können eigentlich Fehler erkannt und ggf. sogar behoben werden?

Fehlererkennung

EDC = Error Detection and Correction bits (Redundante Information)
D = Daten, die durch EDC geschützt werden

- Durch Fehlererkennungs- und Korrektur-Mechanismen kann der Empfänger manchmal – aber nicht immer – erkennen, dass Bitfehler aufgetreten sind
- Höhere Redundanz hilft



Fehlererkennung auf Bit-Ebene

- Idee für die Verwendung von Redundanz
  - Daten werden in Form eines Codes erwartet
  - Ausnutzung der „**Distanz**“ zwischen gültigen Codewörtern, d.h. **nicht alle möglichen Bitkombinationen sind gültige Codewörter**

- **Hamming-Abstand**
  - Anzahl unterschiedlicher Bits zweier Codewörter c_1 und c_2, d.h. Anzahl der 1-Bits von c_1 XOR c_2.
  - Beispiel: $\mathrm{d}(10001001, 10110001) = 3$
  - Hamming-Abstand D von vollständigem Code C :

$$D(C) := \min\{\, d(c_1,c_2) \mid c_1, c_2 \in C; c_1 \neq c_2 \,\}$$

Fehlererkennung- und Korrektur

- Hamming-Abstand eines Code bestimmt die Fähigkeit des Codes, Fehler zu erkennen und zu beheben
  - **erkennen** von *e*-Bit-Fehlern: Hamming-Abstand ***e+1*** notwendig
  - **beheben** von *e*-Bit-Fehlern: Hamming-Abstand ***2e+1*** notwendig

Beispiele

- Fehlererkennender Code:
  - Code mit einem einzigen Paritätsbit
  - Hamming-Abstand des Codes = 2
    (Änderung eines einzelnen Bits erfordert
    Änderung des Paritätsbits)
  - Erkennung eines 1-Bit-Fehler möglich
    (allg.: Fehler mit ungerader Anzahl von Bits)

7 bits of data	(count of 1-bits)	8 bits including parity	
		even	odd
0000000	0	00000000	00000001
1010001	3	10100011	10100010
1101001	4	11010010	11010011
1111111	7	11111111	11111110

- Fehlerbehebender Code (vereinfacht): `00000 00000, 00000 11111, 11111 00000, 11111 11111`

  Hamming-Abstand → des Code = 5
  - Korrektur von 2-Bit-Fehlern möglich
    Beispiel: `00000 00111` → `00000 11111`
  - **FEC** (**F**orward **E**rror **C**orrection)

Hamming–Code

- Wie kann z.B. ein Code entwickelt werden, der automatisch alle 1-Bit Fehler korrigieren kann?
- Bisherige Erkenntnis: Hamming-Distanz muss mindestens *2e+1* = 3 sein (mit e=1)
- Übertragung von .... *m* Datenbits
  r Prüfbits
  n = *m* + *r* Gesamtbits
- Wenn *n* Bits übertragen werden, gibt es auch *n* Fehlermöglichkeiten mit einem Einzelbitfehler. D.h. jede der 2^m verschiedenen Datenwörter braucht jeweils *(n+1)* Repräsentationen (*n* Fehler + eine fehlerfreie Version)
- Die Summe aller Repräsentationen muss in 2^n passen (Gesamtzahl aller darstellbarer Kombinationen).
- Abschätzung für *r* :

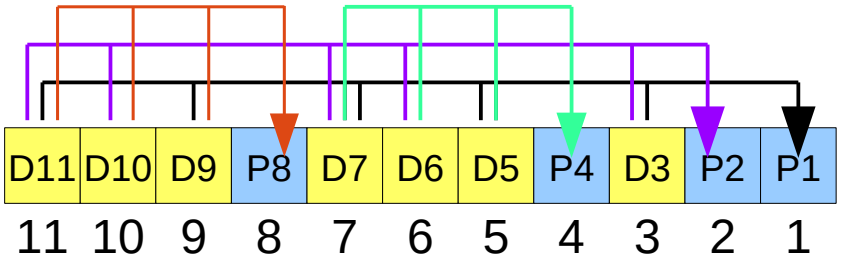
$$(n+1)\cdot 2^m \le 2^n$$
$$mit\ n = m+r$$
$$(m+r+1)\cdot 2^m \le 2^m \cdot 2^r$$
$$(m+r+1) \le 2^r$$

- Beispiel: *m*=7 → *r*=4

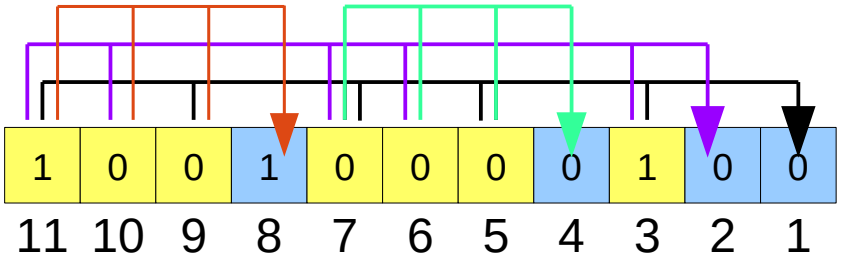
Hamming–Code (11/7)

- Mit dem Hamming Code kann das theoretische Minimum von *r* realisiert werden. (11/7) bedeutet: 11 Bit übertragen, 7 Datenbits
- Die Prüfbits befinden sich (ab 1 gezählt) an allen Positionen von 2-er Potenzen:

- Die Prüfbits geben das Parity-Bit (even) aller DatenBits an, die das entsprechende Prüfbit (als Dualzahl) gesetzt haben.

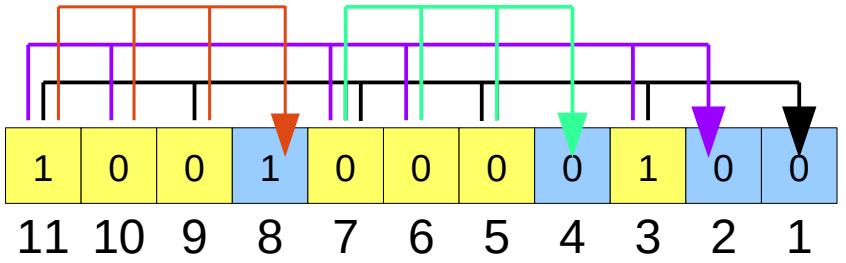


- <u>Beispiel:</u>
  Übertragung von
  ASCII ‚A‘: 0x41 = 1000001

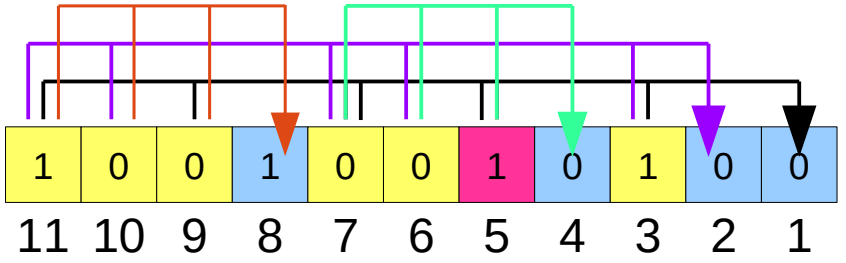


Hamming–Code (11/7)

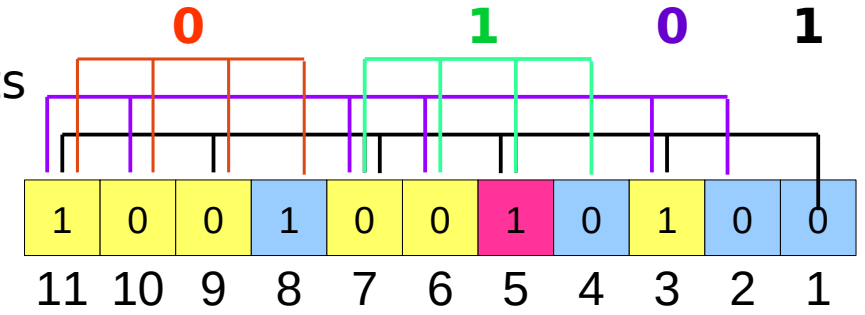
- Beispiel:
  Übertragung von
  ASCII ‚A‘: 0x41 = 1000001



- Fehler bei der Übetragung.



- Empfänger berechnet Paritätsbits nochmal **incl.** der übertragenen Paritästsbits
  → müsste immer 0 sein …
- Hier 101 = 5 →
  Fehler ist an Position 5

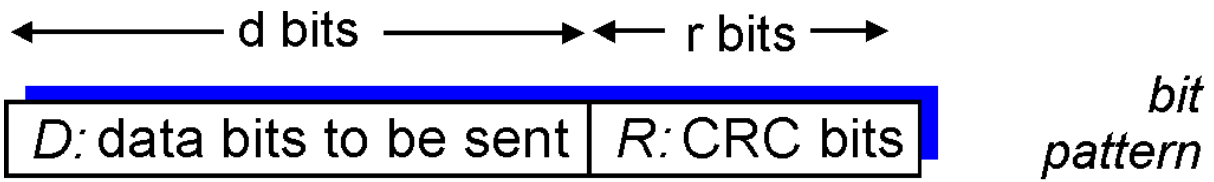


Eine Hardware-Lösung
Cyclic Redundancy Check (CRC)

- Polynom-Codes: Interpretiere Datenbits D als Bitkette eines Polynoms, dessen Koeffizienten die 0-1-Werte der Bitkette sind

  $u^3 + u + 1 \rightarrow 1\ 0\ 1\ 1$

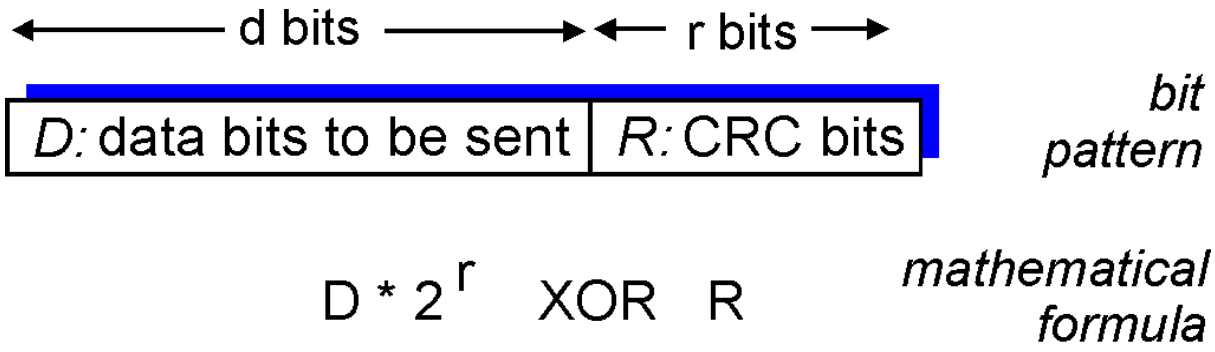
- Prüfung basiert auf Polynom-Arithmetik



$D * 2^r \quad XOR \quad R$ *mathematical formula*

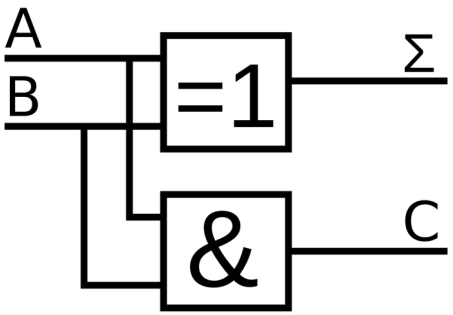
Prüfsummen: Cyclic Redundancy Check

- Sender und Empfänger einigen sich auf ein gemeinsam verwendetes Bitmuster der Länge r+1 Bit, das als **Generator G** bezeichnet wird. Das höchstwertige Bit hiervon ist 1
- Konzept: Berechne die r CRC-bits **R** so, dass die d + r Bits (als Binärzahl interpretiert) mit der **Modulo-2-Arithmetik** genau durch G teilbar sind
  - Empfänger kennt G, teilt <D,R> durch G. Falls der Rest ungleich 0, so liegt ein Fehler vor!
  - Kann Burst-Fehler von weniger als r+1 Bits und jede ungerade Fehlerzahl erkennen



Modulo-2 Arithmetik

- Die Rechenoperationen werden in der Modulo-2-Arithmetik einfacher, da hierbei keine Überträge zu berücksichtigen sind!
- Addition und Subtraktion führen so zu dem gleichen Ergebnis.
- Wir können einfach mit XOR arbeiten!
- Digitaltechnik, hier **Halbaddierer**: Eine Addition ist logisch ein XOR, das Carry ein UND



Wie kann R berechnet werden?

Es gibt eine Bitkombination n, so dass gilt:

$(D \cdot 2^r)$ XOR R = n G

D.h., R soll so gewählt werden, dass G in $D \cdot 2^r$ ohne Rest teilbar ist

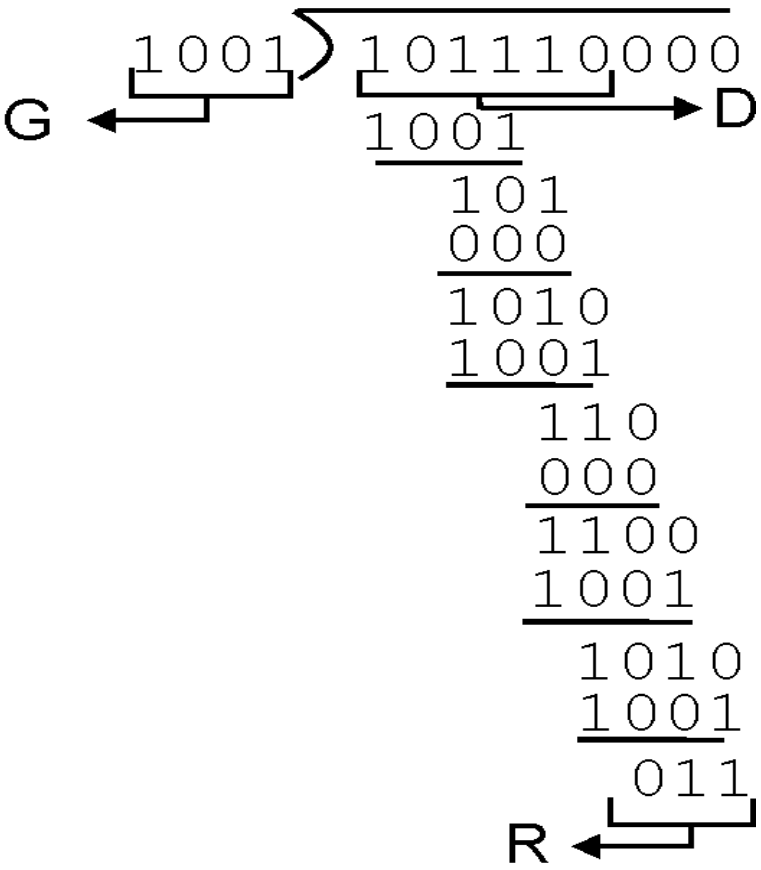
↔

$D \cdot 2^r = nG$ XOR R

↔

Damit kann man R berechnen, denn wenn man **$D \cdot 2^r$ durch G teilt, ist der Rest des Wertes genau R**

$$R = \text{Rest}\left[\frac{D \cdot 2^r}{G}\right]$$



Aufgabe

- Wir betrachten das CRC-Verfahren am Beispiel des Generatorpolynoms $x^4 + x^3 + 1$. Berechnen Sie die CRC-Prüfsumme zur Bitfolge 10110101110

Wir berechnen 10110101110**0000**:**11001**

$\mathbf{1}\cdot 2^4 + \mathbf{1}\cdot 2^3 + 0\cdot 2^2 + 0\cdot 2^1 + \mathbf{1}\cdot 2^0$

```
10110101110 0000
11001
01111101110 0000
 11001
 00110011100000
   11001
   00000 11000
        11001
        0000 0100
```

R ist somit 0100, gesendet wird 10110101110**0100**

Rahmenerstellung in der Sicherungsschicht

- Einteilung einer Nachricht in einheitliche Einheiten (einfachere Übertragung)
- wohldefinierte Schnittstelle nach oben (Schicht 3)
- Kennzeichnung der Einheiten:

Fehlerprüfung
(Prüfsumme zum Rahmen)

FCS = Frame Check Sequence

Kontrollinformationen (Adressen, Rahmennummern, ...)

Fehlererkennung: **ARQ** (**A**utomatic **R**epeat Re**q**uest)

- Verwendung eines fehlererkennenden Codes (CRC)
- Fehler werden erkannt, können aber nicht korrigiert werden! Daher müssen verfälschte Daten neu angefordert werden.
- Einführung einer *Flusskontrolle* (wie bei TCP: Sliding Window):
  - Nummerierung der zu sendenden Datenblöcke
  - Quittierung von Blöcken durch den Empfänger
  - Wiederholung fehlerhaft übertragener Blöcke

FH Aachen
Fachbereich 9 Medizintechnik und Technomathematik
Prof. Dr.-Ing. Andreas Terstegge
Straße Nr.
PLZ Ort
T +49. 241. 6009 53813
F +49. 241. 6009 53119
Terstegge@fh-aachen.de
www.fh-aachen.de