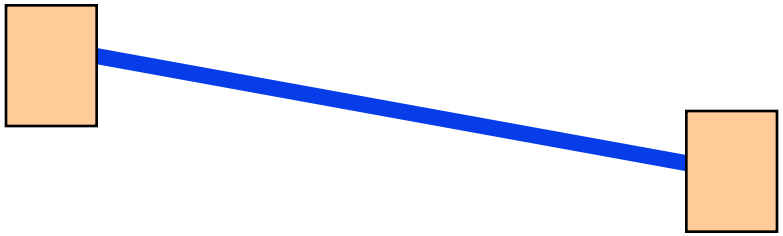
Rechnernetze und verteilte Systeme (BSRvS II)

Prof. Dr. Heiko Krumm

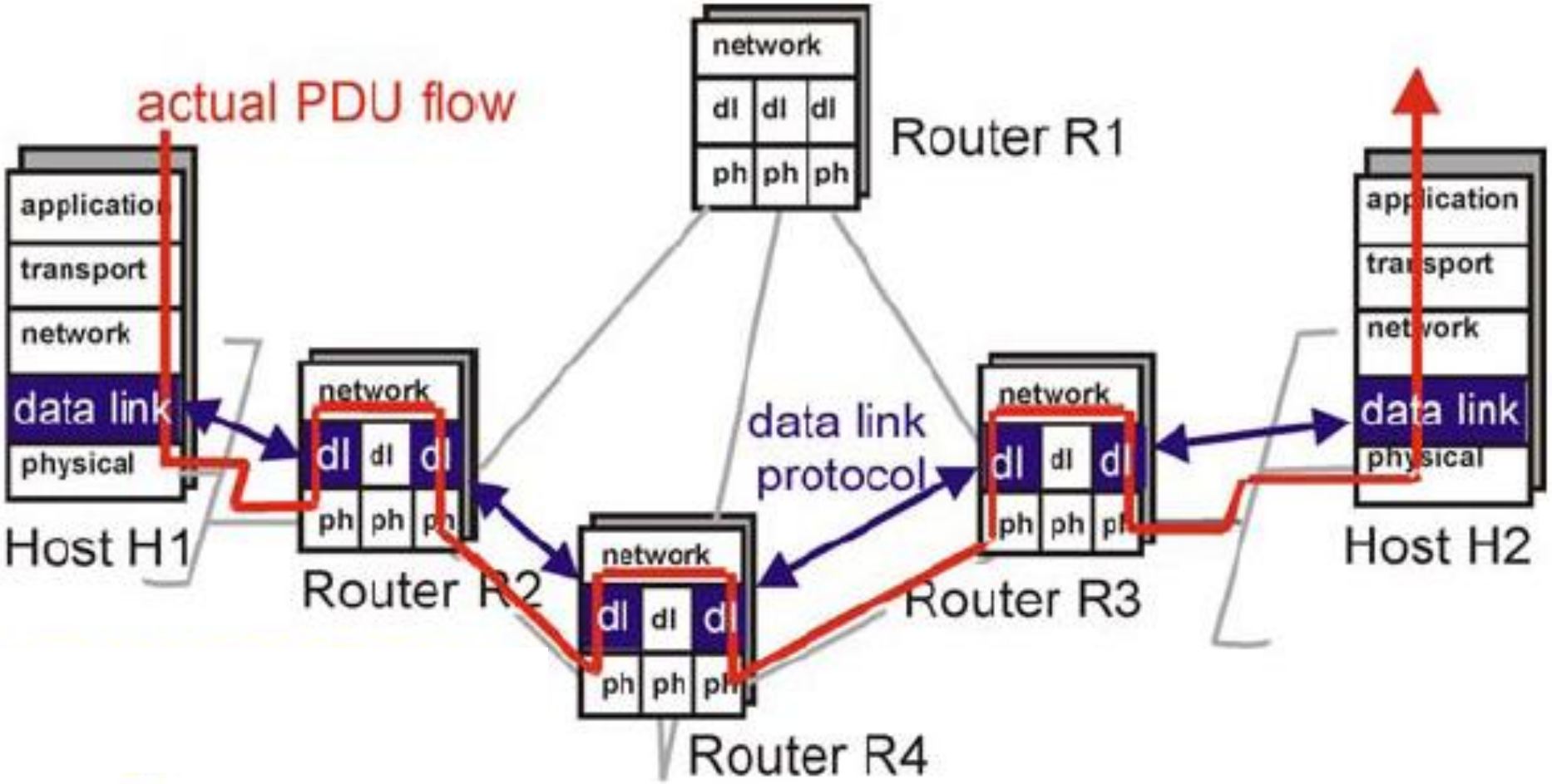
FB Informatik, LS IV, AG RvS
Universität Dortmund



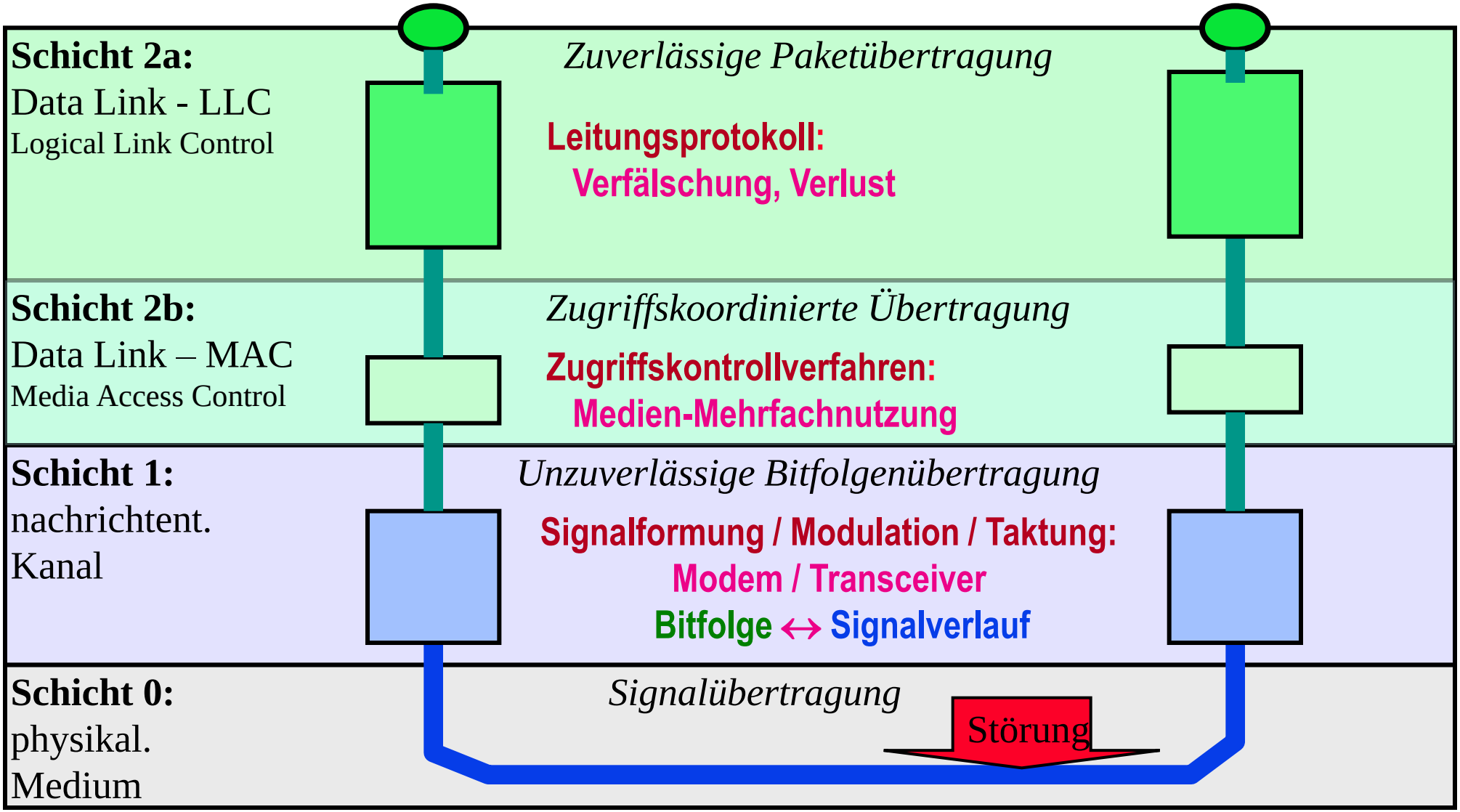
- Aufgaben
- Fehlererkennung und -korrektur
- Medienzugriff
- Adressierung
- Ethernet
- Switching
- Funk-LAN
- Point-to-Point Protokoll

- **Computernetze und das Internet**
- **Anwendung**
- **Transport**
- **Vermittlung**
- **Verbindung (Link)**
- **Multimedia**
- **Sicherheit**
- **Netzmanagement**
- **Middleware**
- **Verteilte Algorithmen**

Übersicht


actual PDU flow
application
transport
network
data link
physical
Host H1
network
dl dl dl
ph ph ph
Router R1
network
dl dl dl
ph ph ph
Router R2
network
dl dl dl
ph ph ph
Router R4
network
dl dl dl
ph ph ph
Router R3
data link
protocol
application
transport
network
data link
physical
Host H2

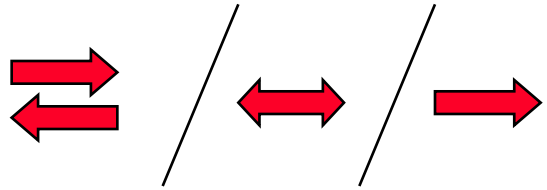
Schichten im Kontext eines physikal. Mediums



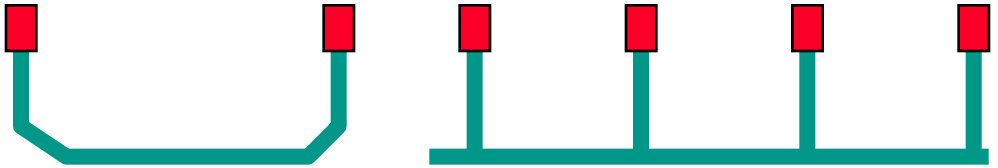
Aufgaben der Schicht 2

- Frames und Verbindungsverwaltung
- Zugriffskontrolle (vor allem bei Bus-Medien)
- Zuverlässige Übertragung
  - Verluste, Verfälschungen, (Phantome, Duplikate, Vertauschungen)
  - Fehlererkennung, Fehlerkorrektur
- Flusskontrolle

Betriebsarten
Halbduplex / Vollduplex / Simplex

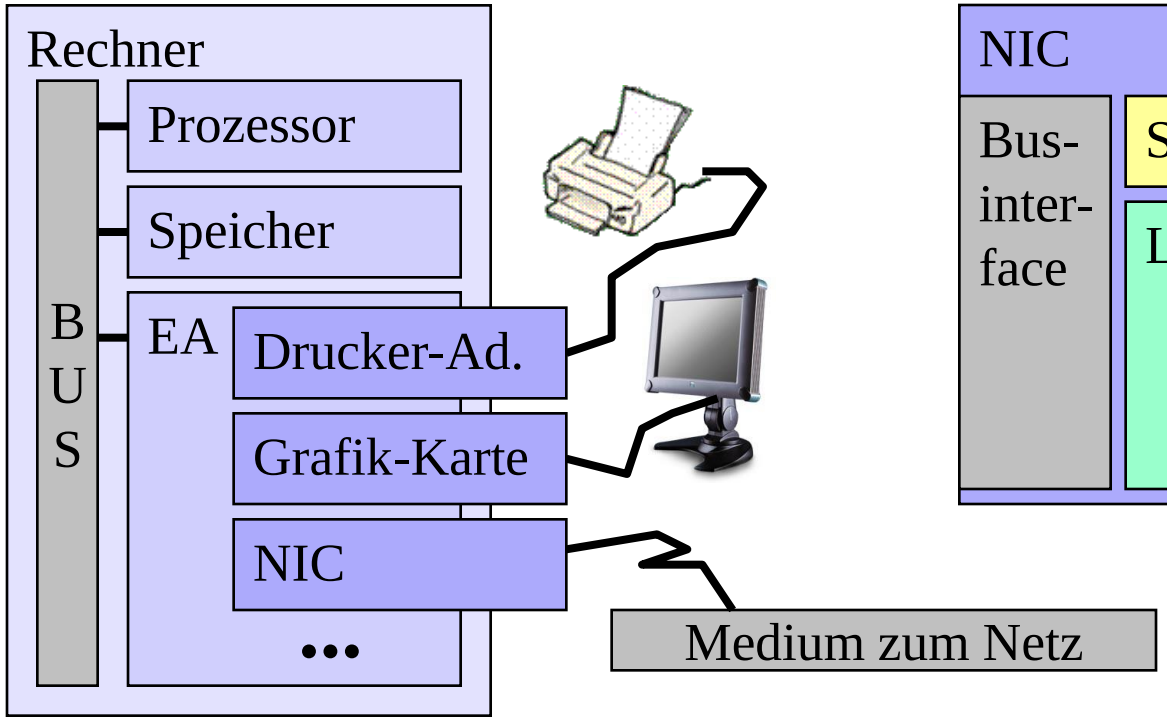


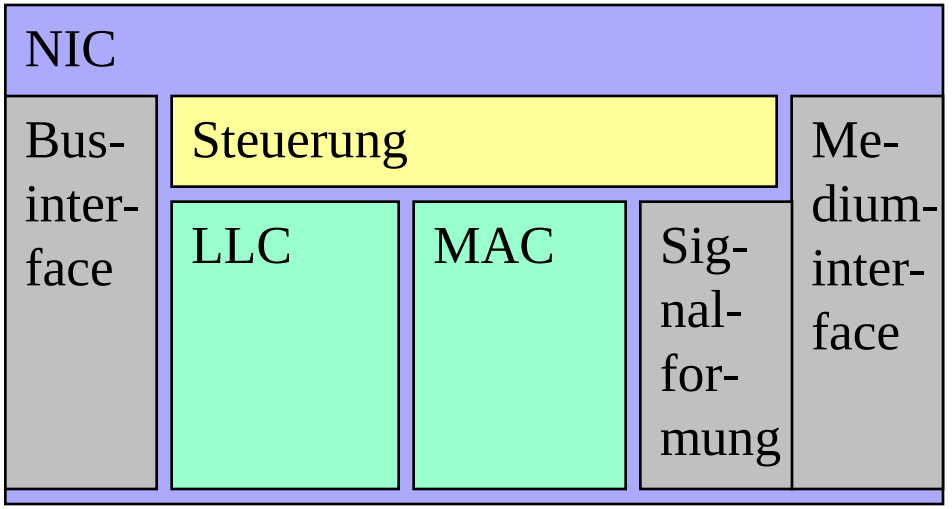
Leitungsarten
Zweipunkt- / Sammelleitung



NIC: Netzinterface-Controller (Netzadapter)

- vergleichbar mit einem Gerätekontroller
  - kontrolliert den Endpunkt eines Mediums
  - ist für das lokale Betriebssystem das „Gerät“
- enthält üblicherweise die Schichten 1 und 2

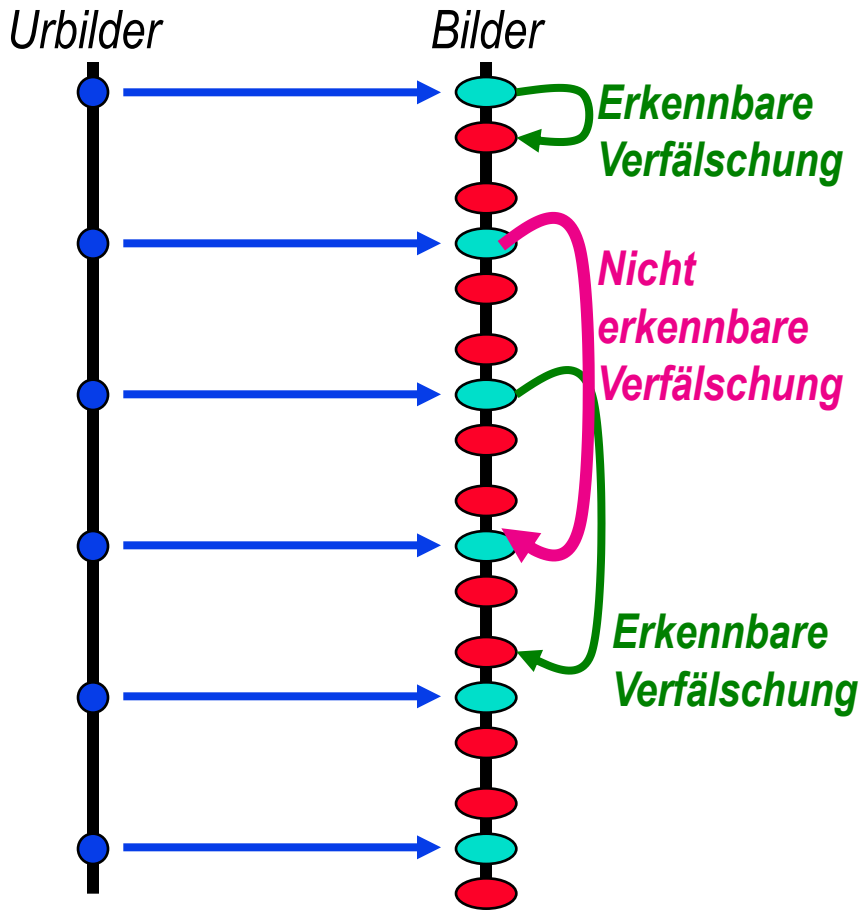




Verfälschungerkennung

Prinzip

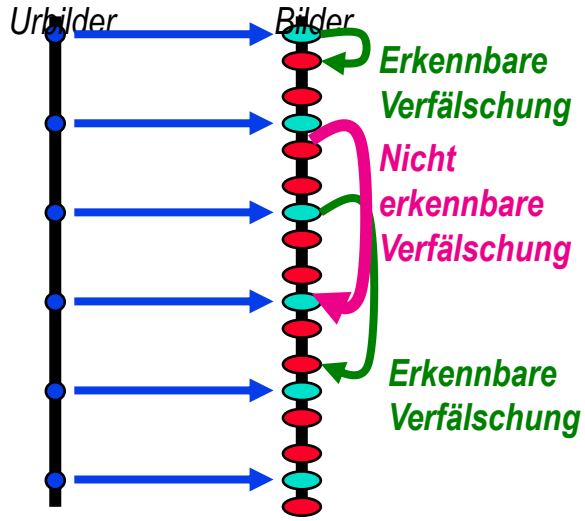
- Bilde Daten auf Erweiterte-Daten ab, so dass Redundanz entsteht
- Nutze Redundanz, um mit ausreichender Zuverlässigkeit aufgetretene Verfälschungen zu erkennen



Verfahren

- Internet-Checksum (vgl. Kap. 3)
- Parity
- CRC-Prüfsumme
- Hamming-Codes

Verfälschungerkennung



Datenpaket

Berechne Prüfsequenz

Datenpaket | Prüfsequenz

Prüfe Prüfsequenz

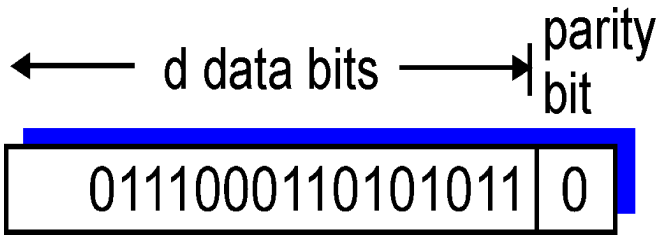
Datenpaket* | Prüfsequenz*

Übertragung, u.U. Verfälschung

Parity-Prüfung / Bitparitätsprüfung

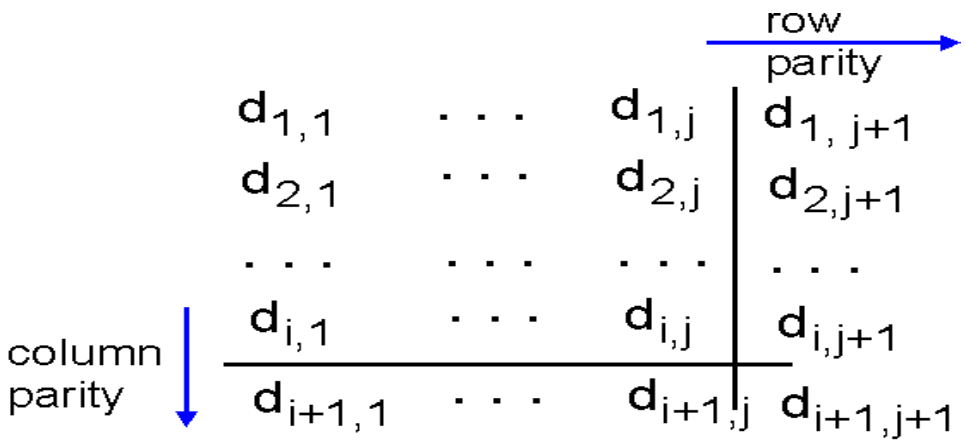
Single Bit Parity:

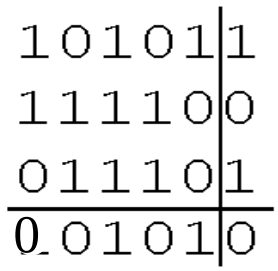
Erkennung von 1-Bit-Fehlern



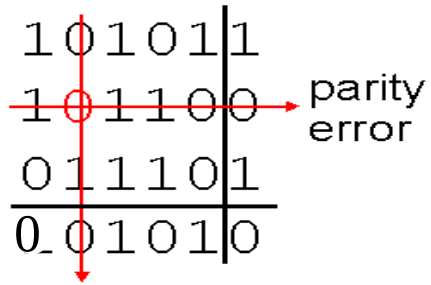
Two Dimensional Bit Parity:

Erkennen und Korrigieren von 1-Bit-Fehlern





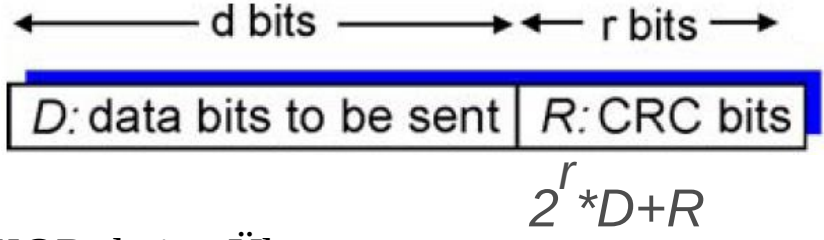
no errors



correctable single bit error

Cyclic Redundancy Check (CRC)

- Datenpaket **D** wird als Binärzahl gesehen
- Eine (r+1) Bit lange Binärzahl liegt als Generatorzahl **G** fest
- Ziel: Berechne r Bit lange Prüfsequenz **R**, so dass
  - **E** = <**D**, **R**> ist ganzzahlig teilbar durch **G**; **R = (2^r*D) / G**
- Der Empfänger empfängt **E‘**, er kennt **G** und prüft, ob
  - **E‘** ganzzahlig durch **G** teilbar ist, falls ja: Mit hoher Wahrscheinlichkeit unverfälscht
  - Falls nein: Verfälschung **E ≠ E‘**
- Verfahren erkennt alle Bündelfehler mit weniger als r+1 Bit Länge
  - *p* (Erkenne Bündelfehler mit Länge q > r) = $1 - (½)^r$
- Weitverbreitetes Verfahren: ATM, HDLC



- Berechnungen in **Modulo 2 – Arithmetik**:
  - Addition und Subtraktion entsprechen bitweisem XOR, keine Überträge
  - Kann auch als Rechnen mit binären Polynomen betrachtet werden, z.B. 100101 / 101 entspricht $(1x^5+0x^4+0x^3+1x^2+0x+1) / (1x^2+0x+1)$

Beispiel: Sendeseite

D = 1101011011 G = 10011

2^4*D = 11010110110000

11010110110000 / 10011 = 1100001010

10011

010011

10011

0000010110

10011

0010100

10011

001110 Rest = 1110

Gesendete Nachricht: 11010110111110

Beispiel: Empfangsseite nach unverfälschter Übertrag.

```
D    = 1101011011          G = 10011
Empfangene Nachricht = 1101011011 1110

1101011011 1110 / 10011 = 1100001010
10011
010011
 10011
 0000010111
      10011
      0010011
        10011
        000000
```

Rest = 0000 → **p (unverfälscht) nahe 1**

Beispiel: Empfangsseite nach verfälschter Übertragung

```
D    = 1101011011          G = 10011
Empfangene Nachricht = 11100110111010

11100110111010 / 10011 = 1111011101
10011
-----
011111
 10011
 -----
 011001
  10011
  -----
  010100
   10011
   -----
   001111
     10011
     -----
     011001
      10011
      -----
      010100
       10011
       -----
       001110
         10011
         -----
         01101
```

Rest = 1101 → ***p*** **(unverfälscht) = 0**
p **(verfälscht) = 1**

CRC: Weitverbreitetes Verfahren

- ATM, HDLC, Ethernet-LLC, …

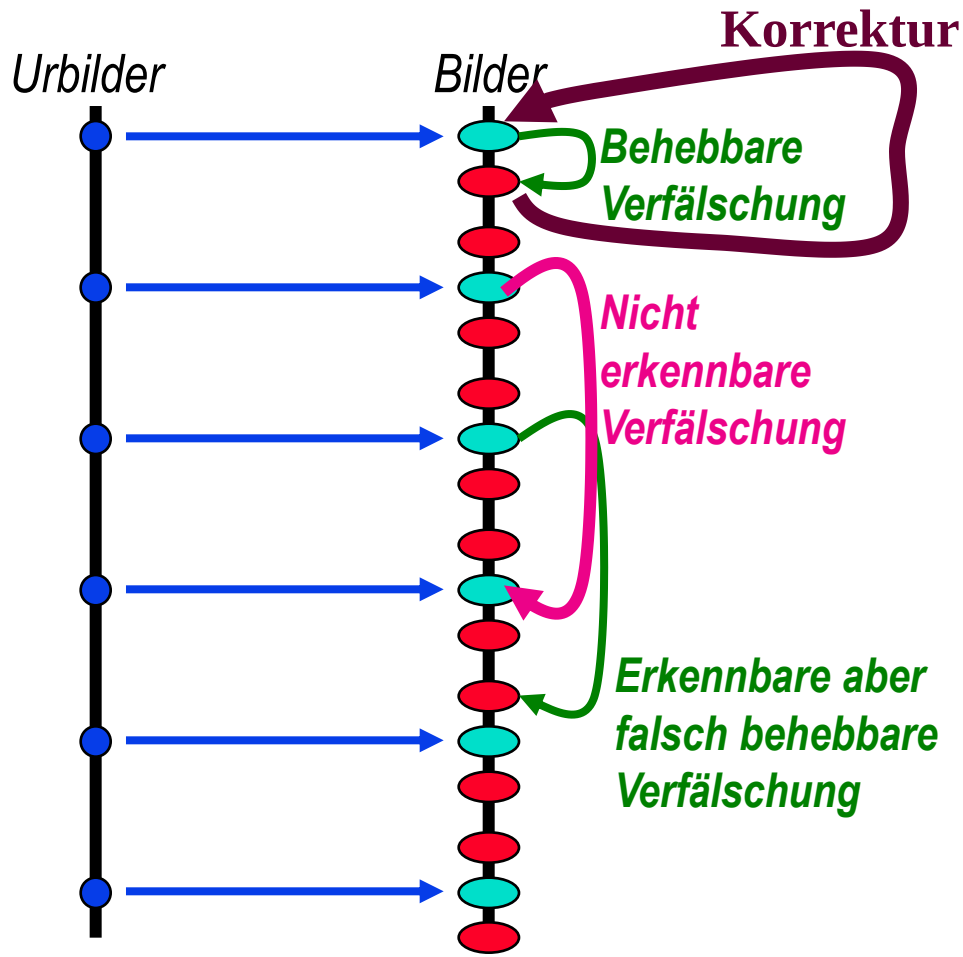
- Standard-Generatorpolynome:
  - **ISO CRC-12**: $x^{12} + x^{11} + x^3 + x^2 + x + 1$ **1 1000 0000 1111**
  - **ISO CRC-16**: $x^{16} + x^{15} + x^2 + 1$ **1 1000 0000 0000 0011**
  - **CRC-CCITT**: $x^{16} + x^{12} + x^5 + 1$ **1 0001 0000 0010 0001**
- alle enthalten (x + 1) als Faktor:
  - ungerade Anzahl von Fehlern wird erkannt
- alle Einzel- oder Doppelfehler werden erkannt
- für CRC-16 und CRC-CCITT werden Bursts bis Länge 16 erkannt,
- für Länge 17 und mehr liegt die Wahrscheinlichkeit über 99, 997 %

FEC: Forward Error Correction - Fehlerkorrektur

- **Hamming-Code**

  Grundidee In zwei Stufen:

  1. Für einen Bitstring D von fest vorgegebener Länge *m* sind *r* Redundanzbits mitzuschicken, so dass Einzelfehler entdeckt und korrigiert werden können.
  2. Das Verfahren unter 1. wird so erweitert, dass auch Burst-Fehler der Länge *l* behandelt werden.



FEC: Forward Error Correction - Fehlerkorrektur

- **Def: Hamming-Distanz *d* eines Codes**
  - Sei *Code* eine Menge von Bitstrings der Länge *k*
  - *d* = kleinste Zahl von Bits in welchen sich zwei Elemente aus *Code* unterscheiden

Beispiel:
Bei „Even-Parity“-Zeichen unterscheiden sich 2 Bytes in mindestens 2 Bits:
Hamming-Distanz=2

000 0000 0	
000 0001 1	Δ an 2 Stellen
000 0010 1	Δ an 2 Stellen
110 0011 0	Δ an 4 Stellen

FEC: Forward Error Correction - Fehlerkorrektur

- **Stufe 1**
  - Quellzeichen der Länge *m* Bit: → 2^m verschiedene Quellzeichen D_1, D_2, ..
  - *r* Zusatzbits, Codezeichen der Länge $n = m+r$: → 2^{m+r} verschiedene Codezeichen
  - Annahme höchstens 1 Bit wird verfälscht:
    Für jedes Quellzeichen gibt es 1 korrektes und *n* illegale Codezeichen
    →
    Codezeichen-Menge: $(n+1)*2^m$ Zeichen und $(n+1)*2^m \leq 2^n$ bzw. $(m+r+1) \leq 2^r$
  - Wie kann man r minimal wählen, so dass Einfachfehler entdeckt und korrigiert werden können?
  - Beispiele:
    - m = 4 → $\min\{r: 5 + r \leq 2^r\} = 3$
    - m = 7 → $\min\{r: 8 + r \leq 2^r\} = 4$
    - m = 64 → $\min\{r: 65 + r \leq 2^r\} = 7$

FEC: Hamming-Verfahren

1. Betrachte das Quellzeichenformat D und berechne r.
2. Ordne die Check-Bits in D so ein, dass die Positionsnummer der Check-Bits genau eine 1 in der Binärdarstellung haben, also

   00 . . . 01, 00 . . . 010, 00 . . . 0100, . . . , 100 . . . 0,
   d. h. in Position 1, 2, 4, 8, ...

   Der ergänzte String sei der zu D gehörige Code-String.
3. Berechne die Check-Bits sukzessive in folgender Weise:
   Für das Check-Bit an der Position 2^i betrachte alle Positionen (in Binärdarstellung!) des Code-Strings, in denen die i-te Komponente der Positionsnummer (von rechts) 1 ist.
4. Zähle die unter diesen Positionen im Code-String auftretenden Einsen.
5. Ergänze zur geraden Parität.
   Das Ergänzungbit (1 oder 0) ist das Bit für die Positionsnummer 2^i.

FEC: Hamming-Verfahren – Beispiel mit m=4, r=3

Check-Bits

Position	1	2	3	4	5	6	7
Position binär	001	010	011	100	101	110	111
Code-String	0	1	1	0	0	1	1

Eine 1 ganz rechts erscheint in Positionen 3, 5, 7. Ergänzung 0.

Eine 1 an zweiter Stelle erscheint in Positionen 3, 6, 7. Ergänzung 1.

Eine 1 an dritter Stelle erscheint in Positionen 5, 6, 7. Ergänzung 0.

FEC: Hamming-Verfahren – Prüfung

6. Der Code-String wird gesendet. Auf der Empfängerseite werden die Check-Bits daraufhin geprüft, ob die Parität stimmt.
7. Wenn ja, wird das Code-Wort akzeptiert.
8. Wenn nein, werden die ”falschen“ Check-Bits weiter betrachtet. Laut Annahme kann es nur Einzelfehler geben:
   a) Es ist genau ein Check-Bit falsch
      Dann ist das Check-Bit selbst verfälscht. Denn wenn ein Datenbit verfälscht wäre, würden, da jedes Datenbit in 2 Check-Bits geprüft wird, 2 Check-Bits nicht stimmen.
   b) Mehr als ein Check-Bit ist falsch
      Dann ist ein Datenbit verfälscht. Die Summe der Positionsnummern der falschen Check-Bits ist die Position des verfälschten Datenbits.

FEC: Hamming-Verfahren – Prüfung - Beispiel

Position	1	2	3	4	5	6	7
Position binär	001	010	011	100	101	110	111
Code-String	0	1	1	0	0	1	1
falscher String C_1'	0	1	1	0	**1**	1	1
falscher String C_2'	0	1	1	**1**	0	1	1

Für C_1': 1. Eins von rechts: in 3, 5, 7: Test Check-Bit ist falsch.
2. Eins von rechts: in 3, 6, 7: Test Check-Bit ist ok.
3. Eins von rechts: in 5, 6, 7: Test Check-Bit ist falsch.
$\Rightarrow$ Position des falschen Bits: $1 + 4 = 5$ bzw. $001_2 + 100_2 = 101_2$

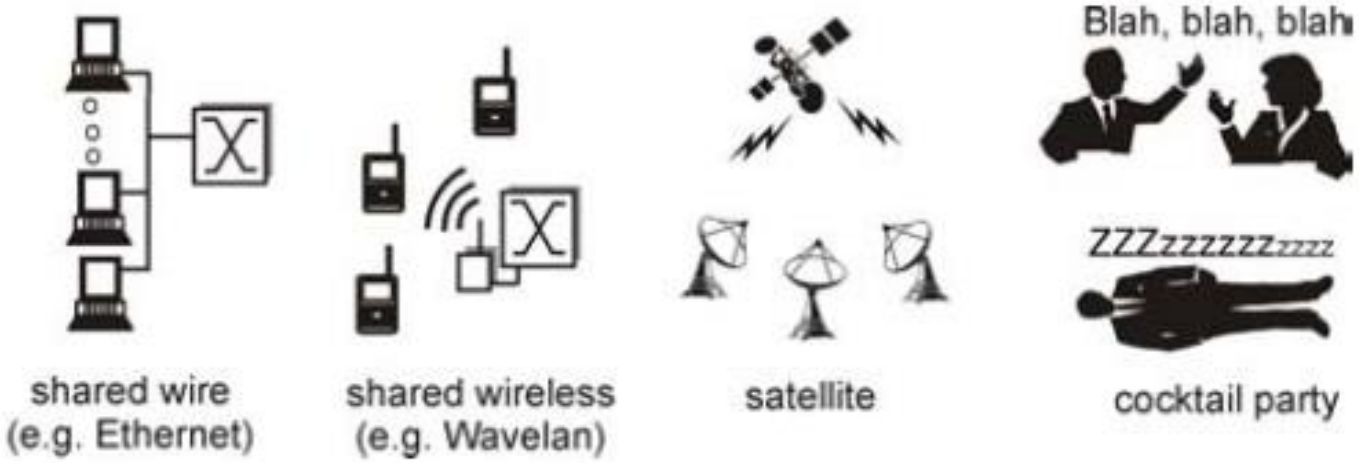
Für C_2': 1. Eins von rechts: in 3, 5, 7: Test Check-Bit ist ok.
2. Eins von rechts: in 3, 6, 7: Test Check-Bit ist ok.
3. Eins von rechts: in 5, 6, 7: Test Check-Bit ist falsch.
$\Rightarrow$ Position des falschen Bits: 4 bzw. 100_2

FEC: Hamming-Verfahren

- **Stufe 2 - Bündelfehler der Länge *k: Erweitertes Verfahren***
  - Ein String D wird in ***k*** Teilstrings der Länge *l* unterteilt.
  - Die Teilstrings werden untereinander geschrieben.
  - Die Teilstrings werden zu Code-Strings erweitert. Von der entstandenen Matrix wird zuerst die erste Spalte gesendet, nachdem entsprechend die Spalten in Code-Spalten erweitert wurden, dann sukzessive die nächsten Spalten, aber insgesamt eine Sendung.
  - Die empfangenen Daten werden wieder als Matrix arrangiert.
  - Ein Burst der Länge ***k*** für die ganze Nachricht bedeutet aber, dass in jeder Zeile der Matrix nur höchstens ein Einzelfehler vorkommt. Er kann korrigiert werden. Das ursprünglicheWort kann dann richtig rekonstruiert werden.

MAC: Media Access Control

- Es gibt drei Arten von Leitungen / physikalischen Verbindungen:
  - 2-Punkt-Leitungen (Point-to-Point-Links)
  - geschaltete Leitungen (Switched Links)
  - **Sammelleitungen / Busse / Mehrfachzugriffsmedien**



Problem: *Durcheinander-Reden*

Im selben Zeitintervall soll höchstens eine Station senden

→ Zugriffskontrolle

MAC: Ideale Lösung

Ein Sammelkanal könne ***R*** Bit pro Sekunde übertragen

Ideale Lösung

- Wenn nur 1 Knoten senden möchte, soll er dies mit voller Rate ***R*** tun können.
- Wenn *m* Knoten senden möchten, sollen sie sich die Übetragungskapazität teilen und jeder soll mit einer Durchschnittsrate von ***R***/*m* senden (Fairness und Kanal-Ausnutzung).
- Lösung soll völlig dezentral sein
  Kein spezieller Verwalter, keine besonderen Synchronisationsverfahren (z.B. Verteilertakt)
- Lösung soll einfach und leicht zu implementieren sein

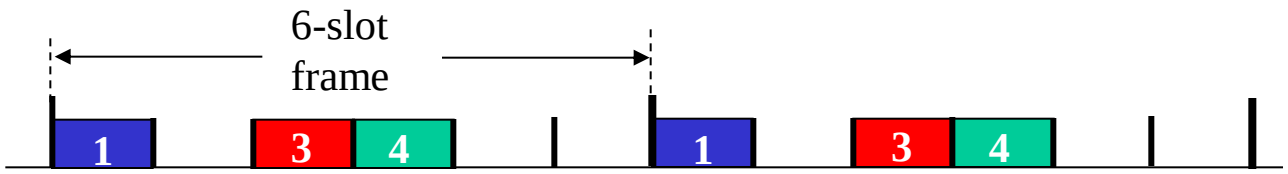
MAC: Verfahrensgrundtypen

- Statische Kanalaufteilung
  - Spalte Kanal in kleinere Portionen auf
    - » Zeitschlitze (Time Division Multiple Access TDMA)
    - » Frequenzbänder (Frequency Division Multiple Access FDMA)
  - Ordne Portionen fest und exklusiv einzelnen Sendern zu
- Bedarfsgesteuerte dynamische Kanalzuteilung
  - Zentrale Zuteilung
  - Dezentrale Zuteilung
    - » Dezentrale zufällige Konkurrenz
      - Jeder, der senden möchte probiert es: Es gibt Kollisionen.
      - Kollisionsauflösung
    - » Dezentrale Rechtevergabe
      - kursierendes Token

MAC: Statische Zuteilung per TDMA

TDMA: time division multiple access

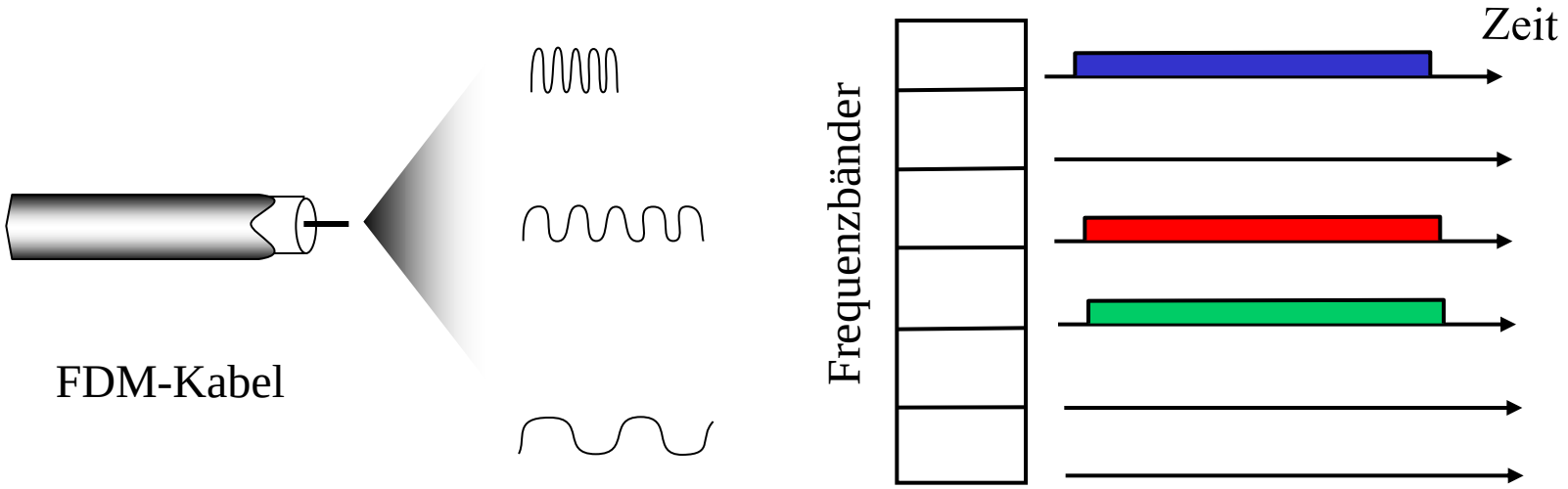
- Kanalzugriff in “Runden”
- Jede Station bekommt Slot fester Länge (Länge in Paket pro Zeit) in jeder Runde
- Unbenutzte Slots bleiben leer
- Beispiel: LAN mit 6 Stationen, 1,3,4 wollen senden, Slots 2,5,6 bleiben leer



MAC: Statische Zuteilung per FDMA

FDMA: frequency division multiple access

- Spektrum des Kanals wird in Frequenzbänder geteilt
- Jeder Station wird ein festes Frequenzband zugeordnet
- Kapazität in Frequenzbändern, die zu nicht sendewilligen Stationen gehören bleibt ungenutzt
- Beispiel: LAN mit 6 Stationen, 1, 3, 4 wollen senden, Frequenzbänder 2, 5, 6 bleiben ungenutzt



MAC: Neues Verfahren CDMA **Code Division Multiple Access**

- Jeder Station wird ein eigener Code zugewiesen: Coderaum-Aufteilung
- Alle Stationen nutzen dasselbe Frequenzspektrum, aber jeder Station ist eine eigene „Chipping Sequenz“ zugewiesen, mit der deren Daten codiert werden:
  - **Code Signal = <Original Data> x <Chipping Sequence>**
- Orthogonale Chipping Sequenzen erlauben es, dass mehrere Stationen gleichzeitig senden können und die Daten trotzdem decodierbar sind.

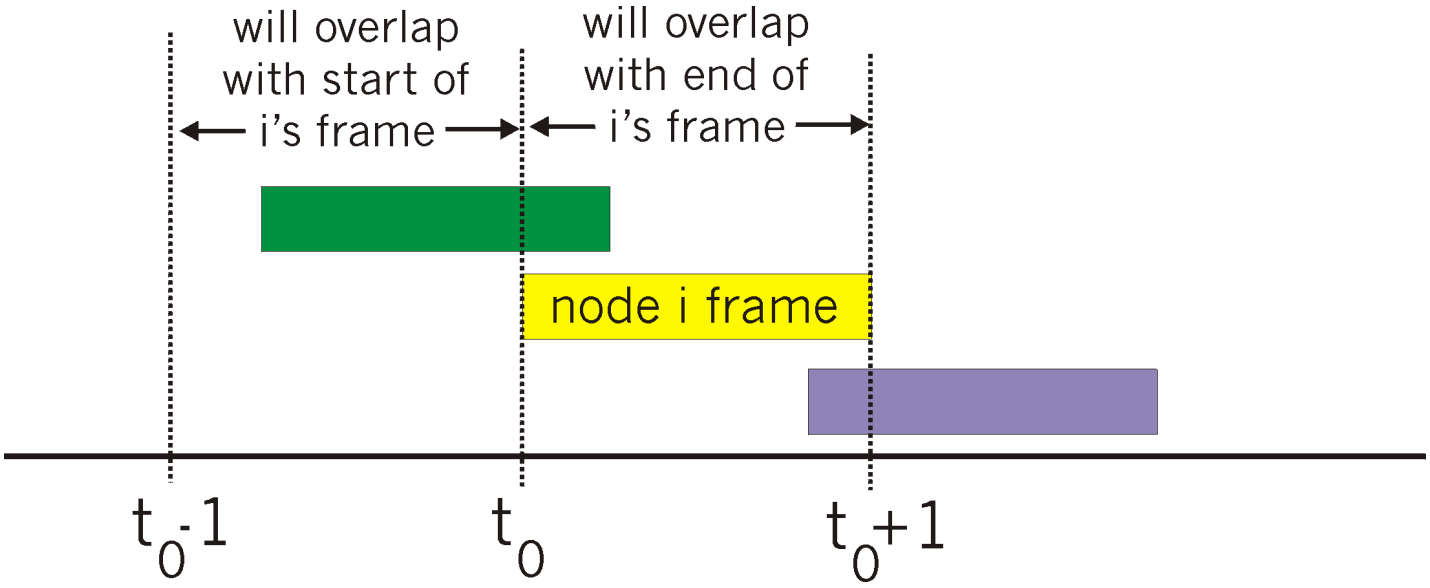
Einsatz bei UMTS / 3G-Mobilfunk

MAC: Dezentrale konkurrierende Zuteilung

- Wenn ein Knoten ein Paket senden will, dann
  - sendet er mit der größtmöglichen Datenrate R.
  - ohne vorherige Koordination mit anderen Knoten
- Zwei oder mehr Knoten wollen senden ➜ “Kollision”,
- Random Access MAC Protokolle spezifizieren:
  - Wie Kollisionen erkannt werden
  - Wie mit erkannten Kollisionen umgegangen wird (z.B., durch verzögerte Neuübertragung)
- Beispiele für Random Access MAC Protokolle:
  - Slotted ALOHA
  - Pure ALOHA
  - CSMA, CSMA/CD, CSMA/CA

MAC: Konkurrierende Zuteilung – **Pure Aloha**

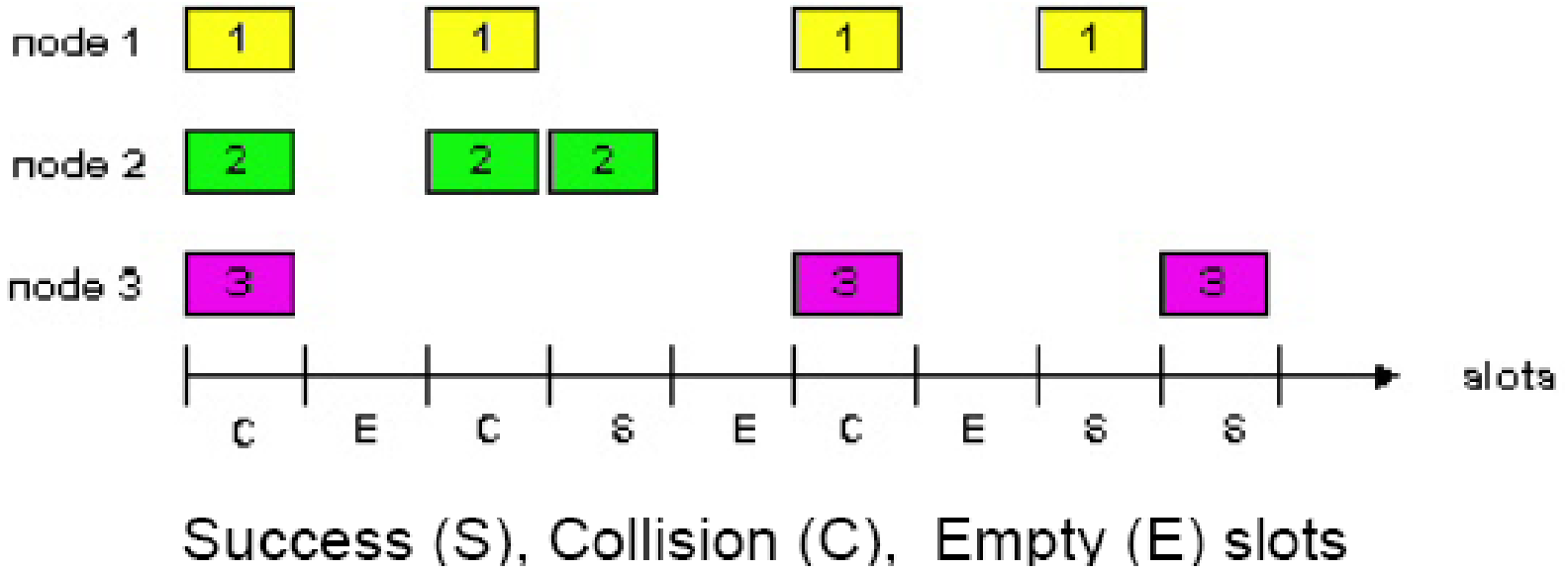
- Sende bei Sendwunsch ohne vorherige Abstimmung
- Sende ganzes Paket: Es kommt an oder nicht, falls Kollision auftritt



- Problem: p(Paket durch Kollision zerstört) deutlich > 0
- Paket ist während der ganzen Sendezeit verletzlich:
  - Idee: Slotted Aloha

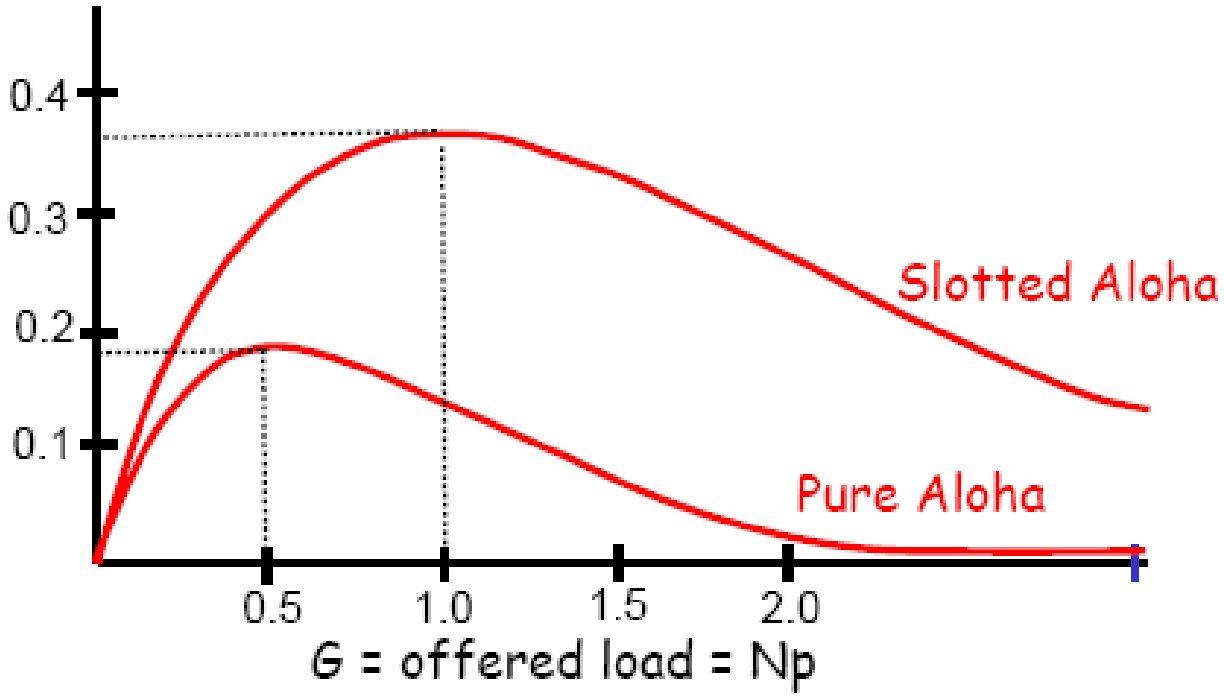
MAC: Konkurrierende Zuteilung – **Slotted Aloha**

- Es gibt ein Zeitraster, das Zeit in Intervalle / Slots / Schlitze einteilt
- Sende bei Sendwunsch bei Beginn des nächsten Schlitzes
- Sende ganzes Paket: Es kommt an oder nicht, falls Kollision auftritt



- Paket ist nur noch zu Beginn eines Slots verletzlich

MAC: **Aloha - Leistungsfähigkeit**



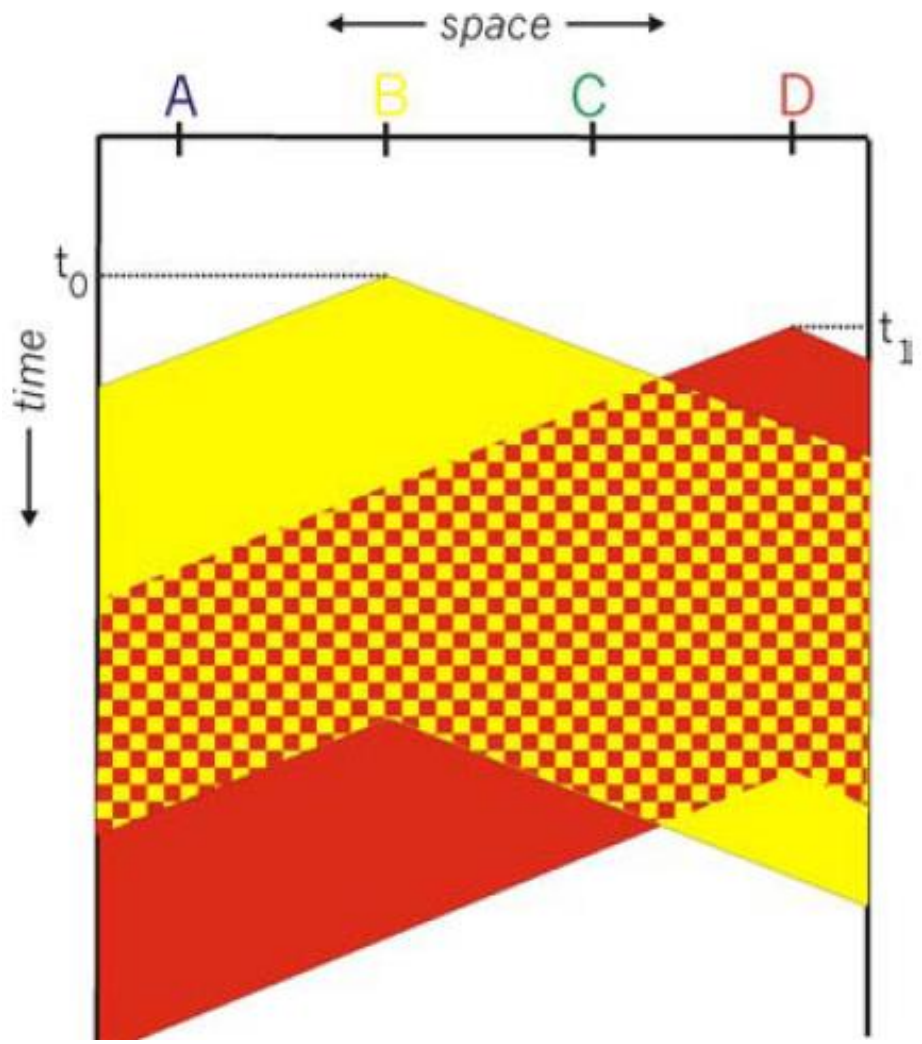
- Bei beiden Varianten gibt es Stauverhalten
- Einfaches Aloha kann Kanal nur zu maximal 18% ausnutzen
- Slotted Aloha verdoppelt die maximale Ausnutzung auf 36%

Aloha: Weitere Verbesserungen

- Nicht, wenn Sender sich gegenseitig nicht hören oder wenn lange Signallaufzeiten vorhanden sind
- Aber sonst:
  - Hören vor dem Senden, ob Kanal frei
  - Hören, während des Sendens, ob Paketsenden wegen aufgetretener Kollision abgebrochen werden soll
  - Variable Wartezeit vor dem nächsten Sendeversuch nach einem erfolglosen Versuch
  - Staukontrolle:
    Messen der Belastung des Kanals
    Drosseln durch exponentiell längere Wartezeiten vor dem nächsten Versuch
- Führt zu Ethernet
  - über 90% der Kanalleistungsfähigkeit sind nutzbar
  - kein Stauverhalten
  - aber doch deutlich wachsende Latenzzeiten

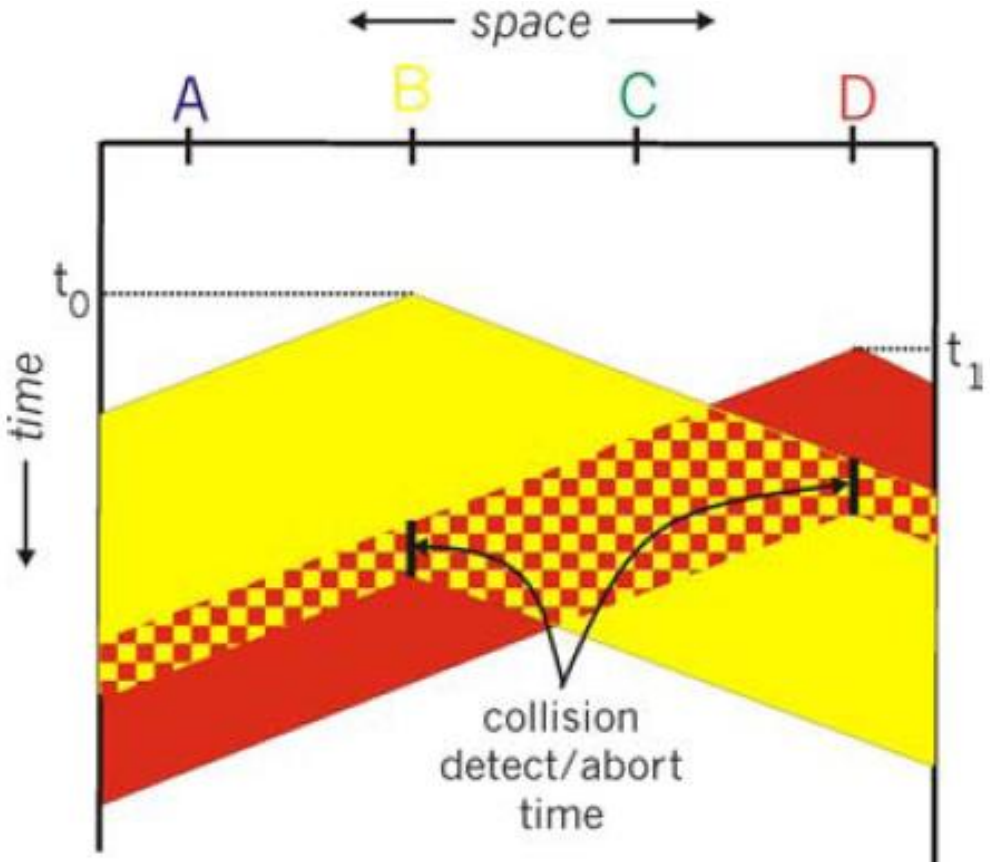
Carrier Sense Multiple Access (CSMA)

- Hören, ob anderer schon sendet, bevor man sendet
  - Wenn Kanal frei: Senden eines Pakets
  - Wenn Kanal belegt: Für die Dauer eines zufällig langen Zeitintervalls warten
- Analogie: Nicht unterbrechen, wenn jemand spricht.
- Kollisionen sind dennoch möglich
  - Signallaufzeiten bedingen, dass jemand schon senden kann, obwohl ihn ein anderer noch nicht hört
  - Die Wahrscheinlichkeit für Kollisionen hängt von der maximalen Signallaufzeit (also der Kabellänge) ab
- Bei einer Kollision ist das ganze Paket zerstört



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

- Auch während des Sendens weiter Mithören, ob anderer „*dazwischen funkt*“
  - Wenn Kollision erkannt, sofort mit dem Senden aufhören, um Kanal schnell wieder frei zu machen, sowie „Jam“-Signal senden
- Bei Kabel-Medien leicht realisierbar
- Bei Funk-Medien schlecht, weil dort Empfänger während des Sendens abgeschaltet wird
- Varianten zur Wiederholung des abgebrochenen Versuchs
  - persistent (stur wieder probieren)
  - non-persistent (nach Wartezeit)
- Wenn im Vergleich zur Signallaufzeit lange Nachrichten gesendet werden, ist eine hohe Kanalausnutzung möglich



Zentrale Rechtevergabe

(der Vollständigkeit halber und zum Vergleich)

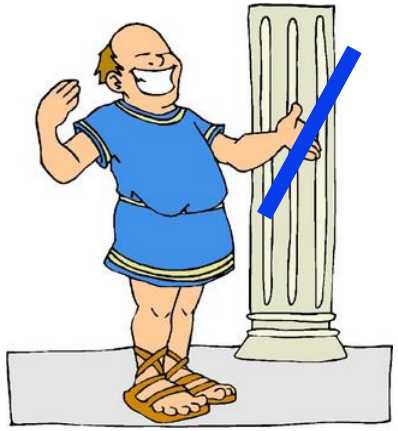
- Verfahren: Polling
  - „Busmaster“ frägt nacheinander die anderen Knoten am Bus ab, ob sie senden wollen
  - Angefragter Knoten antwortet, z.B. entweder mit Fehlanzeige oder mit Nachricht
    - » Anfragenachrichten (Poll-Messages)
    - » Antwortnachrichten (Response-Messages)

- Probleme
  - Polling Overhead
  - Wartezeit bis man an die Reihe kommt
  - wenn der zentrale Verwalter ausfällt, fällt Gesamtsystem aus

Dezentrale Rechtevergabe: Kursierendes Token

- *Der Rednerstab bei der Volksversammlung im antiken Athen*

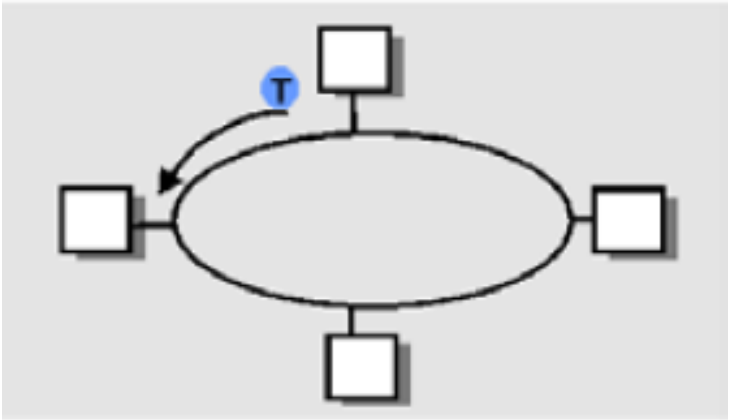
Wer reden will, muss warten, bis ihm der Rednerstab gebracht wird.



- Statische Kanalaufteilung:
  - bei hoher Last: sehr effizient
  - bei geringer Last: sehr ineffizient, lange Wartezeiten, selbst wenn nur 1 Sender aktiv ist, bekommt er nur 1/N der Übertragungskapazität
- Konkurrierende Zuteilung:
  - bei geringer Last: schneller Zugang, 1 Sender kann ganze Kapazität bekommen
  - bei hoher Last: Instabilität und Overhead durch häufige Kollisionen
- Dezentrale Rechtevergabe:
  - Die Vorteile beider Ansätze vereinen?

Dezentrale Rechtevergabe: Kursierendes Token

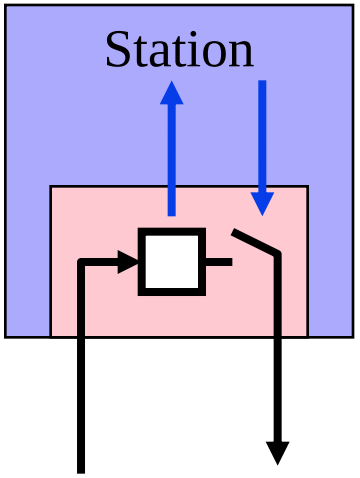
- Ein Kontroll-Token wird reihum von einem zum anderen Knoten weitergereicht
  - Token – Nachricht kreist
  - Wer das Token hat und senden will, sendet. Danach gibt er das Token weiter.
  - Wer das Token hat und nicht senden will, gibt das Token gleich weiter.
  - Fairness:
    Zyklische Weitergabe
    Maximale Sendedauer je Runde und Station
  - Realzeit:
    Maximale Wartezeit pro Station

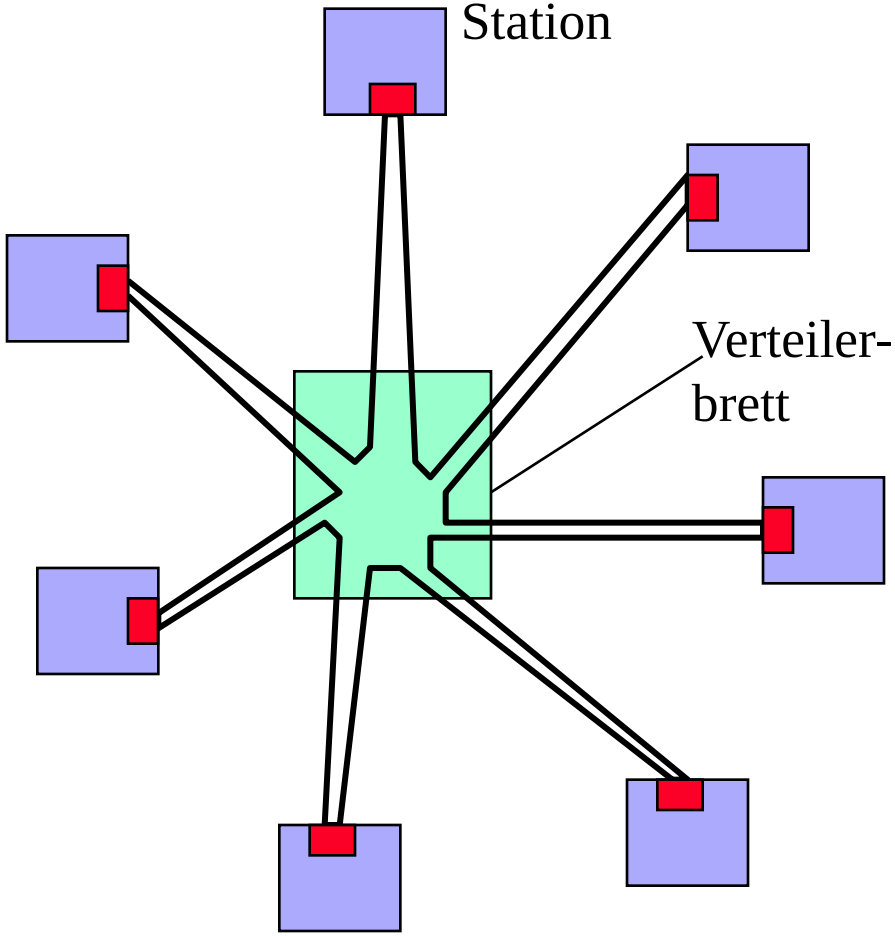


- Probleme:
  - Overhead durch kreisendes Token
  - Wartezeit, bis Sendewilliger an der Reihe (ist aber beschränkt)
  - wenn das Token verloren geht, steht das System
    (ist durch Erkennung/Behebung lösbar)

Kursierendes Token: Beispiel IEEE 802.5 „Token Ring“

- Ring in Stern-Gestalt
  - Aus 2-Punkt-Simplex-Leitungen,
  - die aber wegen der besonderen Kopplung ein gemeinsames Medium bilden
- Kopplung: 1-Bit-Verzögerung

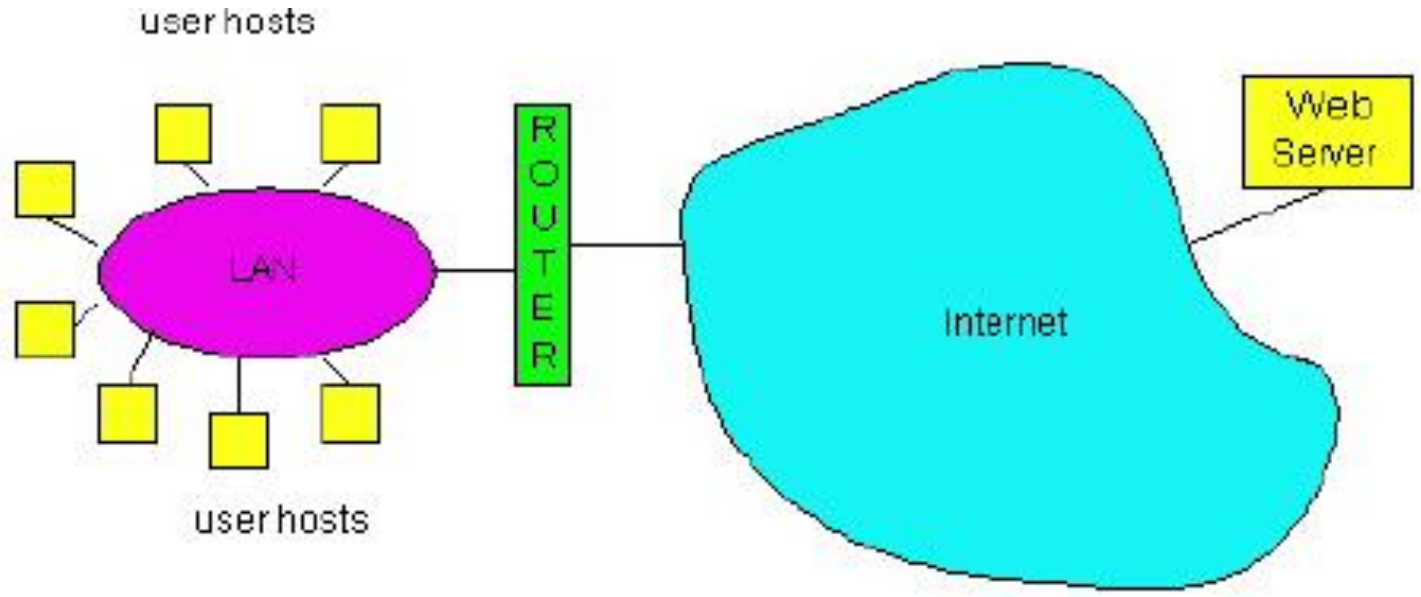




LAN-Technologien

In LANs genutzte MAC Protokolle:

- **Token Ring** IEEE 802.5 (IBM token ring), bis zu 16Mbps; Frame, läuft einmal durch den Ring und wird vom Sender entfernt
- **FDDI** (Fiber Distributed Data Interface), für mittlere Größen (campus, metropolitan), bis zu 200 Station und 100Mbps Übertragungsrate
- **Ethernet** nutzt CSMA/CD; 10Mbps (IEEE 802.3), Fast Ethernet (100Mbps), Giga Ethernet (1,000 Mbps); am weitesten verbreitet

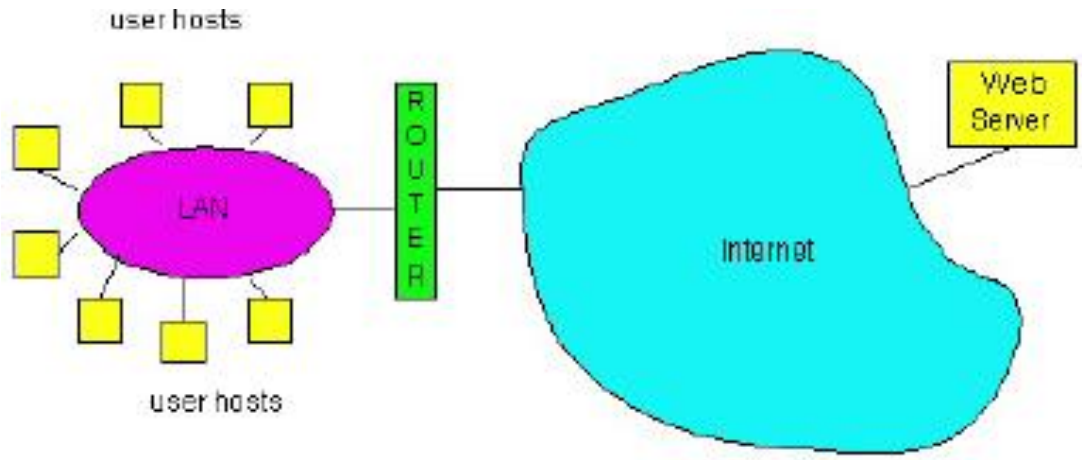


LAN Technologien

- In diesem Kapitel bisher: Link Layer
  - Fehlererkennung und Korrektur
  - Zugriffskontrolle

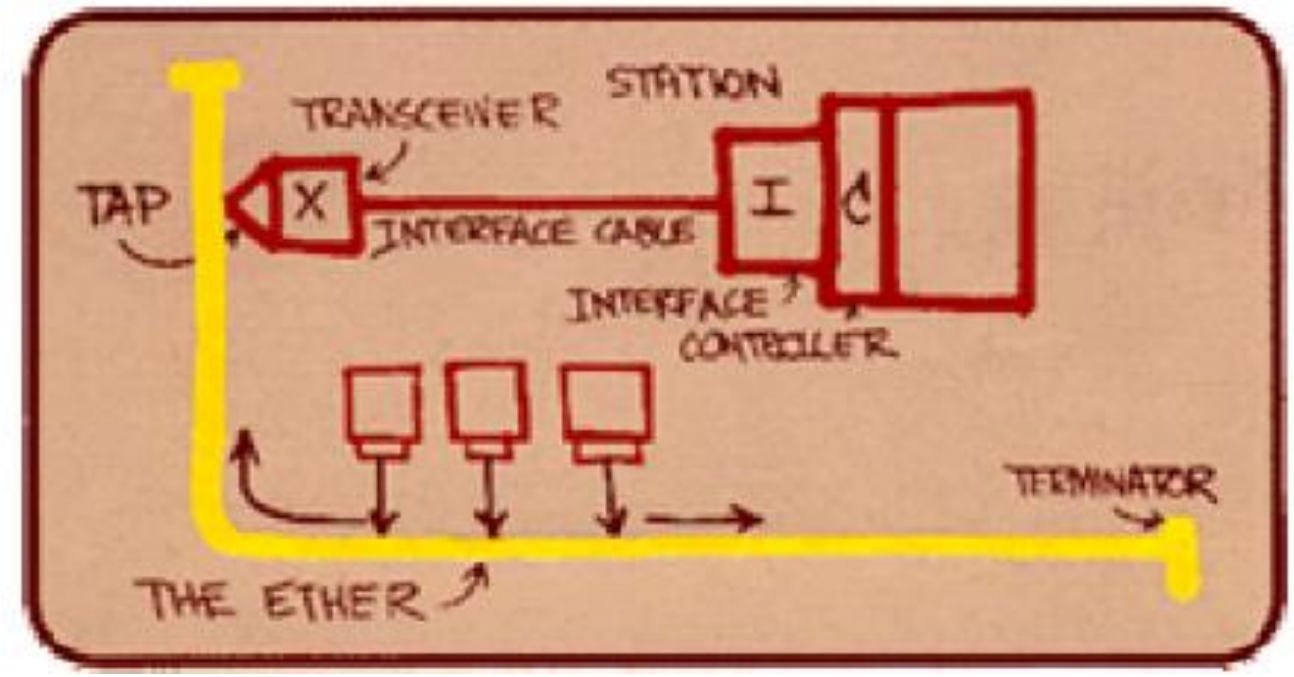
- Weiterhin: Lokale Netze (nach ISO 8802 / IEEE 802 – Standards)
  - Adressierung: MAC-Adressen
  - Ethernet
  - Hubs, Brücken / Bridges, Switches
  - WLAN: 802.11

- Point-to-Point-Protocol: PPP



Ethernet: Dominierende LAN-Technologie

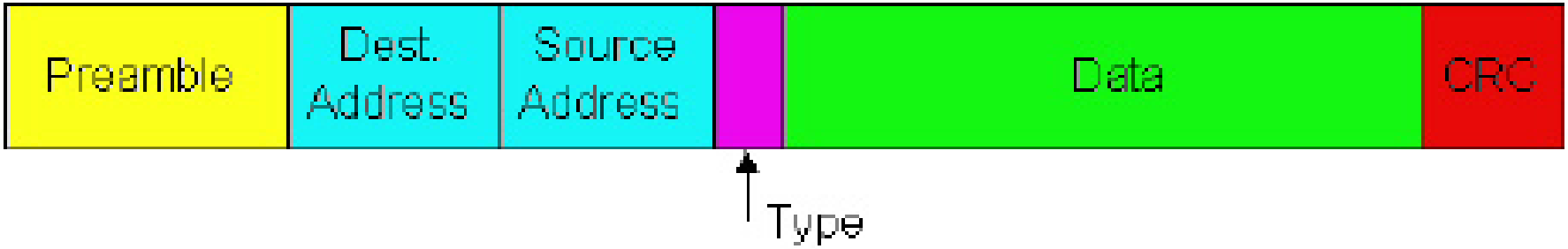
- Hohe Stückzahlen, günstige Preise
- Wesentlich günstiger als ATM und Tokenring
- Hält mit der Entwicklung Schritt: 10, 100, 1000 MBit/sec



Original Ethernet sketch

Ethernet-Frames

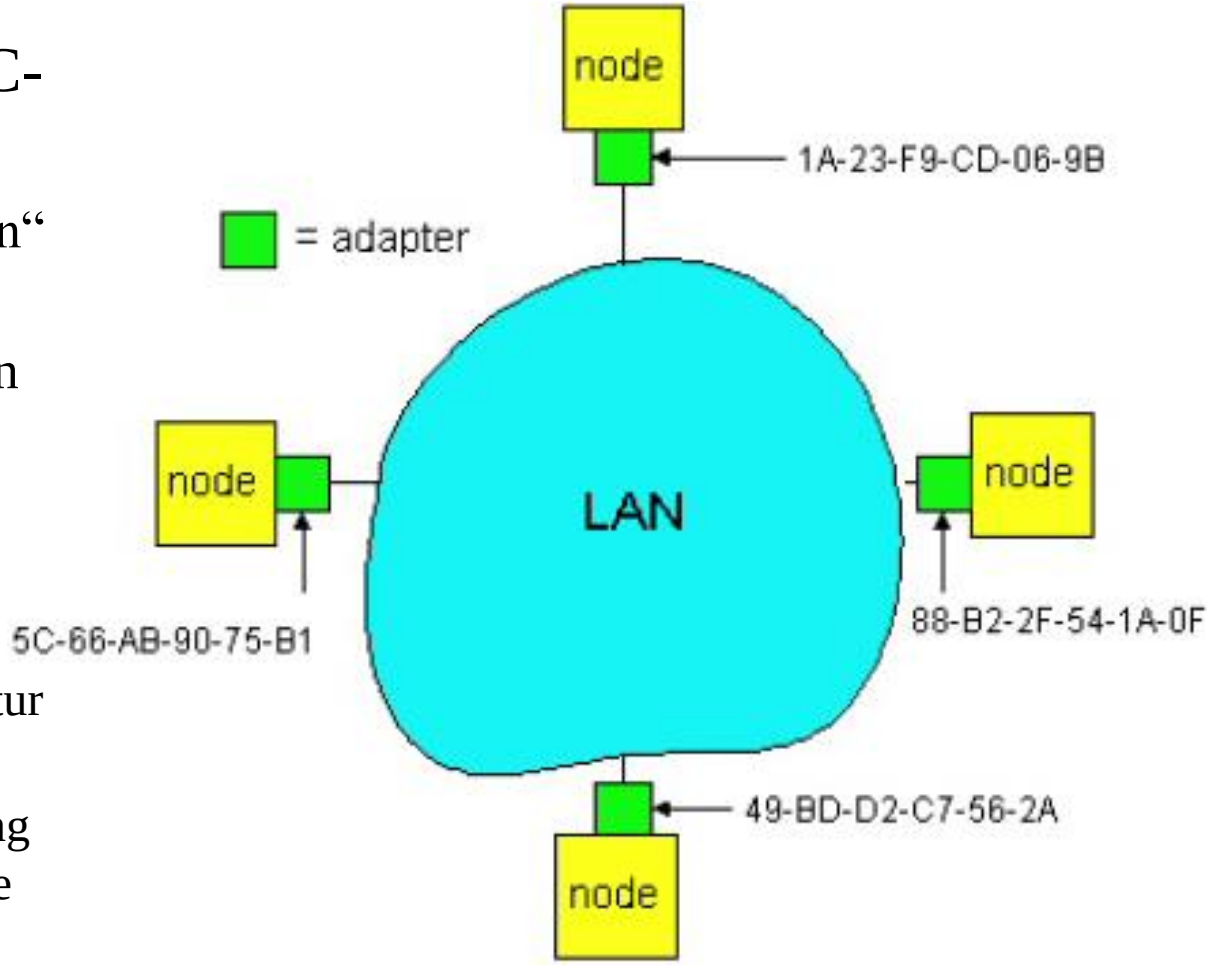
- Es werden Schicht-2-Pakete, so genannte Ethernet-Frames ausgetauscht



- Präambel:
  7 Bytes mit 10101010-Wert gefolgt von einem Byte 10101011
  Hilft den Empfänger einzusynchronisieren
- 6 Byte lange Adressen: MAC-Adressen
- **IEEE 802.2 Typ 1 LLC**:
  - Verbindungslose unzuverlässige Datagramm-Übertragung, nur Verfälschungserkennung, Behebung durch Verlust
- **IEEE 802.2 Typ 2 LLC**:
  - HDLC-artige verbindungsorientierte zuverlässige Übertragung, Sequenznummern, positive und negative Quittungen

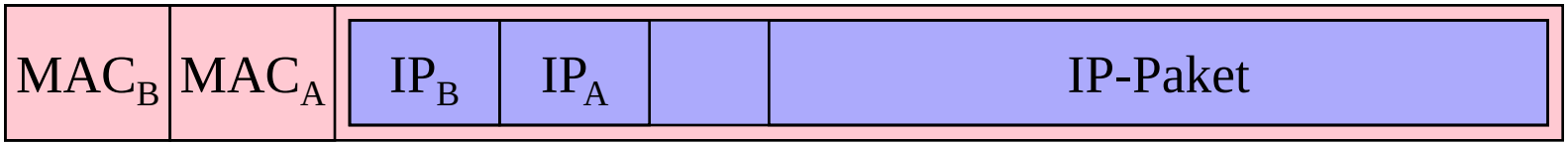
Adressierung im LAN-Segment

- Segment: Medium, MAC-Ebene muss adressieren
  - IP-Adresse hat „hier unten“ nichts zu suchen
  - eigene Schicht 2 Adressen für Unterscheidung der Stationen am selben Segment:
    MAC – Adresse:
    - » 48 Bit breit, ohne Struktur
    - » früher bekam jedes NIC schon bei der Herstellung eine feste MAC-Adresse mit (heute teilweise per Software einstellbar), Adressen werden durch IEEE vergeben



Adressierung im LAN

- Station A will an Station B mit IP-Adresse IP_B senden
- Link-Instanz der Station A muss zu IP_B die passende MAC-Adresse finden



Ethernet-Frame

A) B ist im selben LAN-Segment:
Ziel-MAC-Adresse = MAC_B

B) B ist in einem anderen LAN-Segment:
Ziel-MAC-Adresse = $MAC_{ForwardingRouter}$

- In beiden Fällen wird eine Adressauflösung benötigt: $IP_X \rightarrow MAC_X$
  - Aufgabe von Address Resolution Protocol (ARP) und
  - ARP-Cache

Adressierung im LAN: ARP-Protokoll

- A soll ein IP-Paket an B senden, kennt aber Bs MAC-Adresse nicht
- A sendet einen LAN-Segment-Broadcast Frame:
  ARP-Query für IP_B
- Alle Stationen an diesem LAN-Segment empfangen dies, also auch B
- B sendet ARP-Reply an A: IP_B hat MAC_B
- A kann nun das Paket in einen MAC-Frame packen und an MAC_B senden
- A merkt sich die Zuordnung in einem Zwischenspeicher (ARP-Cache)
  - Cache-Einträge haben Lebenszeit (z.B. 20 min), danach werden sie gelöscht

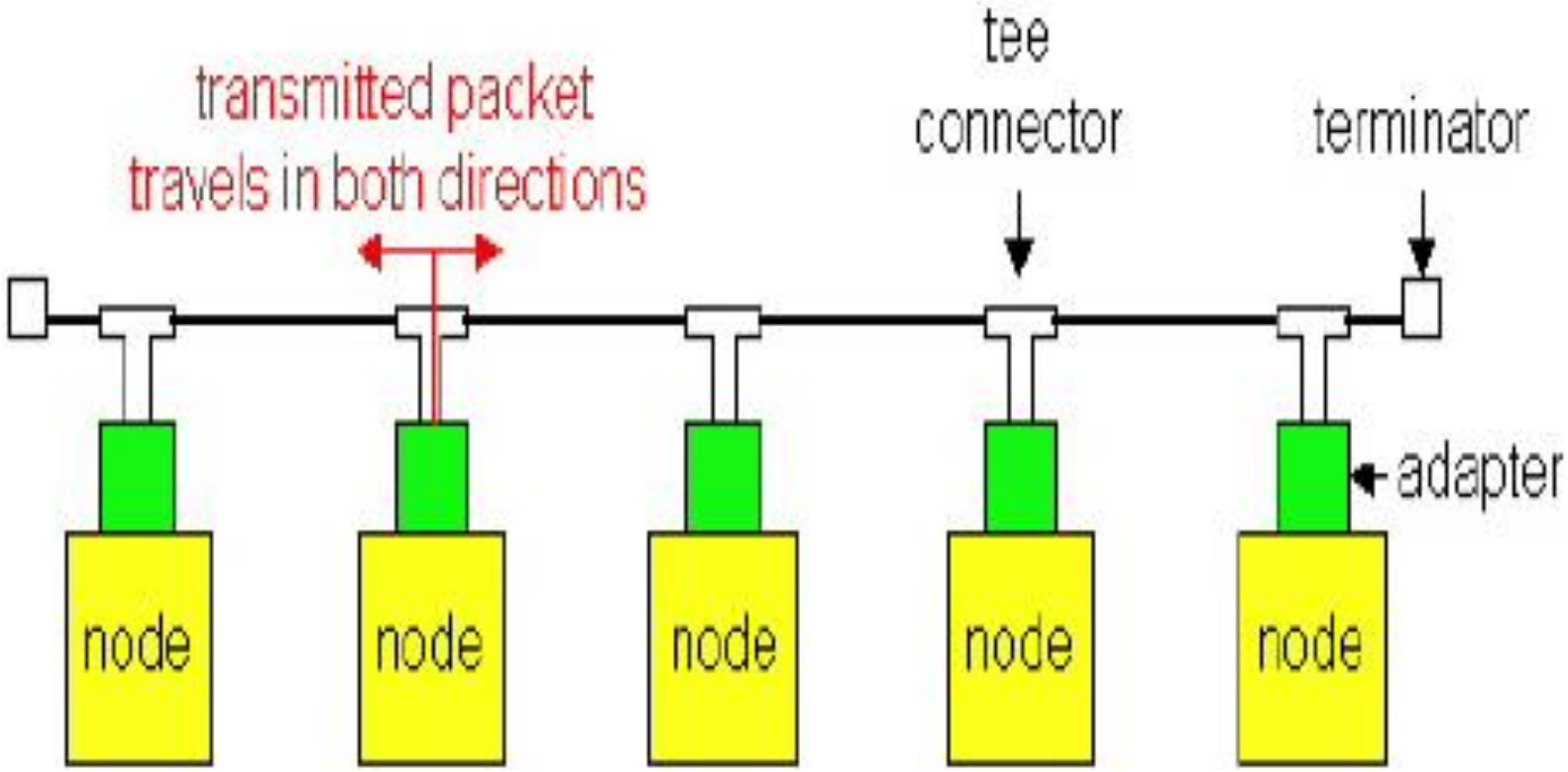
Ethernet: Zugangskontrolle per CSMA/CD

- **A:** sense channel, **if** idle **then** { // **CSMA/CD**
  transmit and monitor the channel;
  If detect another transmission **then** {
  abort and send jam signal;
  update # collisions;
  delay as required by exponential backoff algorithm;
  goto A }
  else {done with the frame; reset # collisions to zero} }
  else {wait until ongoing transmission is over and goto A}
- Jam-Signal: 48-Bit Spezialmuster „Alle Sendungen sofort abbrechen!“
- Exponential Backoff: Staukontrolle im LAN-Segment
  - first collision: choose k randomly from {0,1}; delay is k x 512 bit transmission times
  - after second collision: choose k from {0,1,2,3}
  - after m collisions, choose k from {0,1,2,…,2^m-1}
  - after ten or more collisions, choose k from {0,1,2,3,4,…,1023}

Im Mittel wächst k exponentiell mit der Segmentbelastung

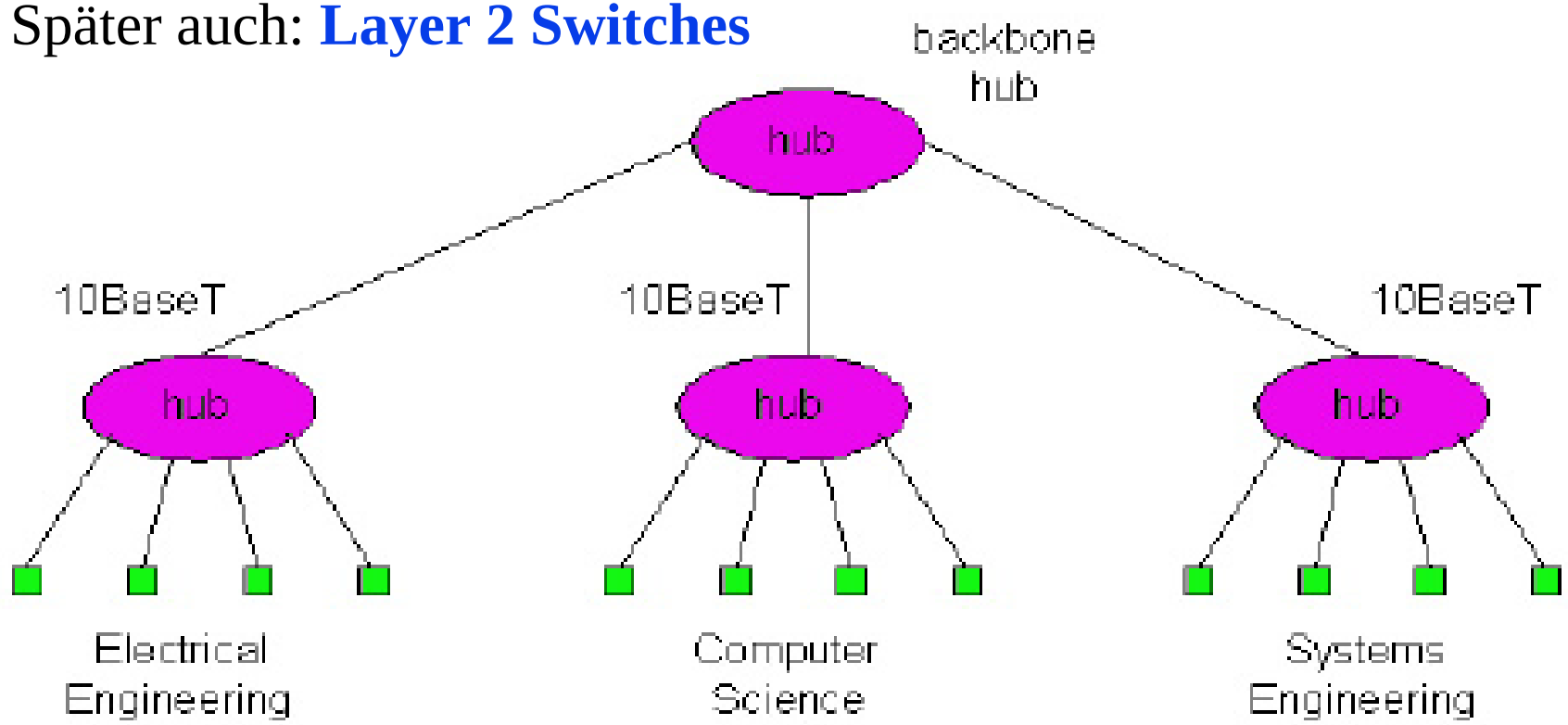
Ethernet: 10Base2

- **10**: 10 MBit/sec; **2**: < 200 m Kabellänge
- Medium: Koaxialkabel als Bus
- Verlängerung durch Repeater (Repeater koppeln auf Signalebene)

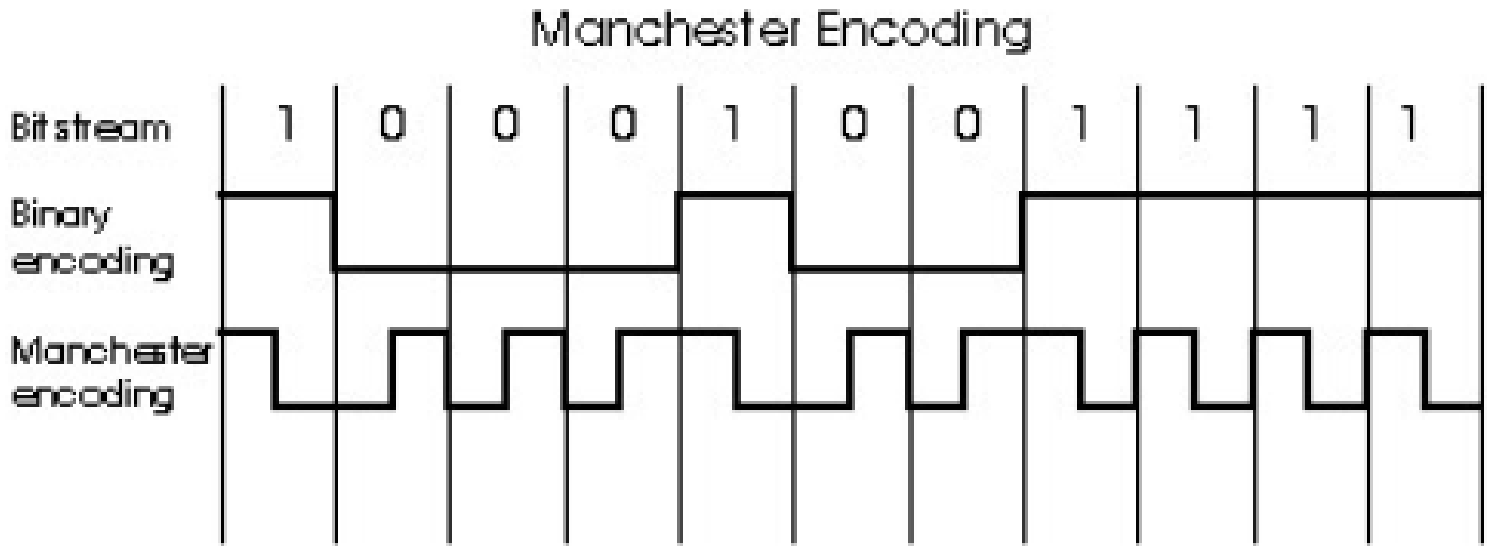


Ethernet: 10BaseT und 100BaseT

- **10 / 100**: Bitrate; **T** : Twisted Pair (Verdrillte Zweidrahtleitung)
- **Hub**: Stern-Repeater, Radius < 100 m, Medium bleibt Bus
- 100BaseT verwendet eine verbesserte Codierung
- Später auch: **Layer 2 Switches**



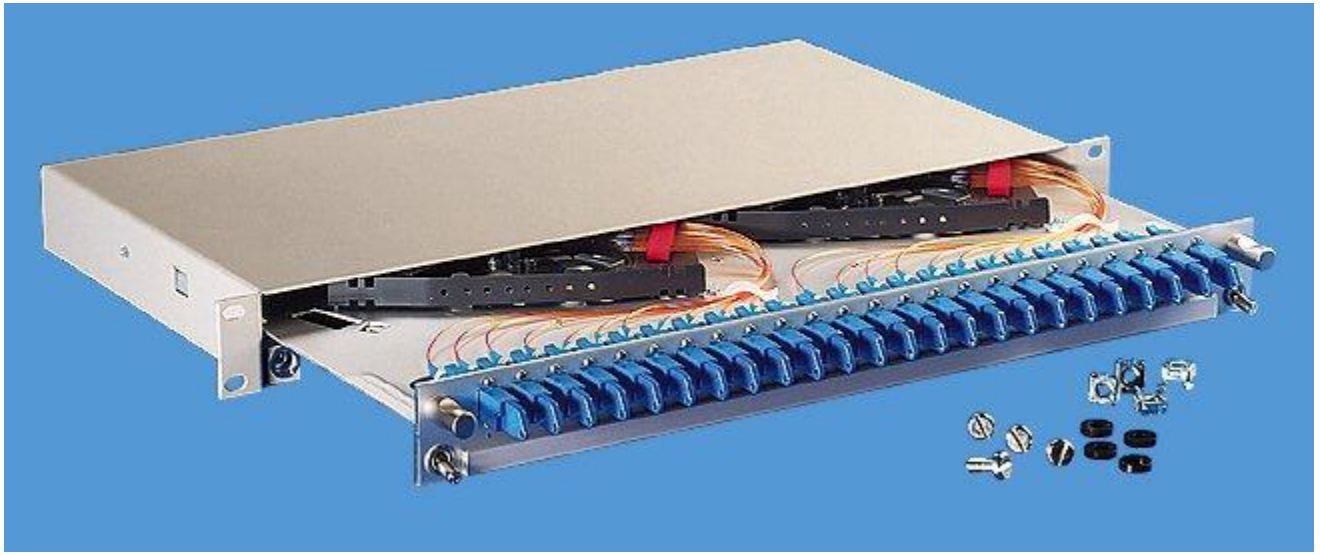
10BAse2, 10BaseT: Manchester Codierung



- Vorteil: Selbstsynchronisierender Code
  jedes Bit wird durch eine Flanke dargestellt,
  dazwischen u.U. Wechselflanke.
- Vorteil: Carrier Sense benötigt nur 1 Bit

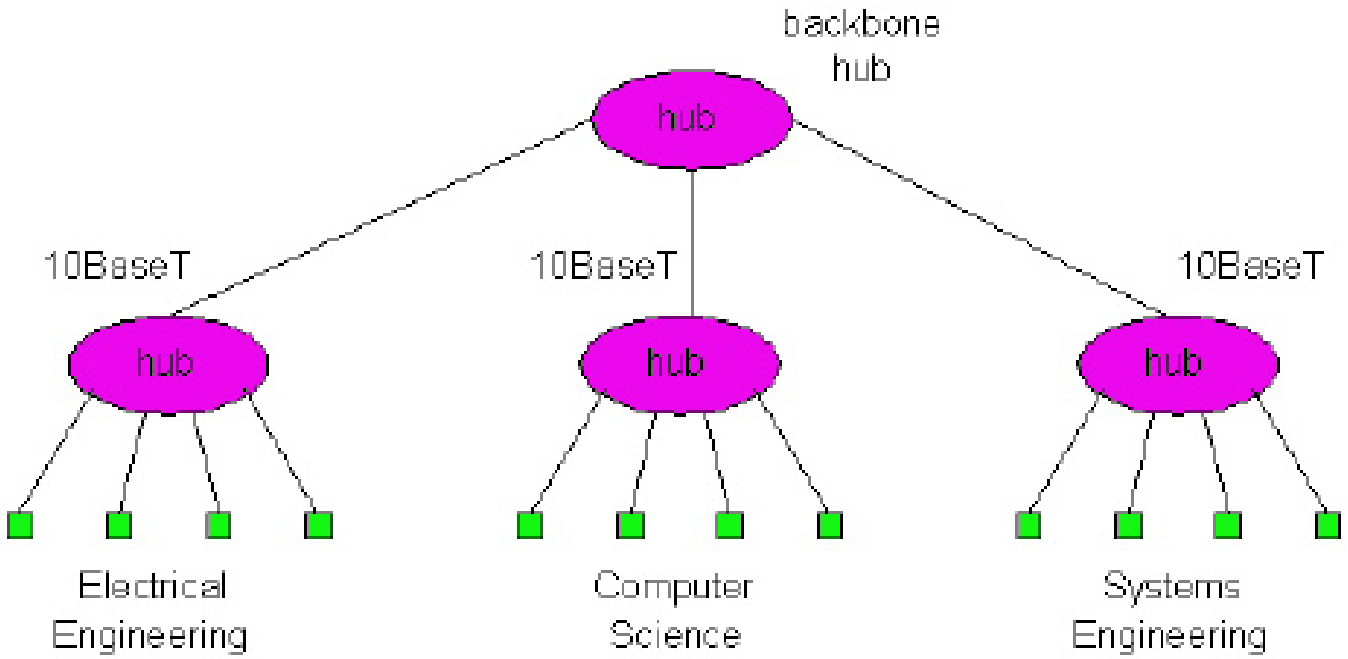
Ethernet: Weitere

- Gbit Ethernet (1000MBit/sec)
  Standard Ethernet-Frames
  - sowohl CSMA/CD Busse (müssen sehr kurz sein)
  - als auch Zweipunkt-Leitungen
- 10 GBit/sec schon verfügbar



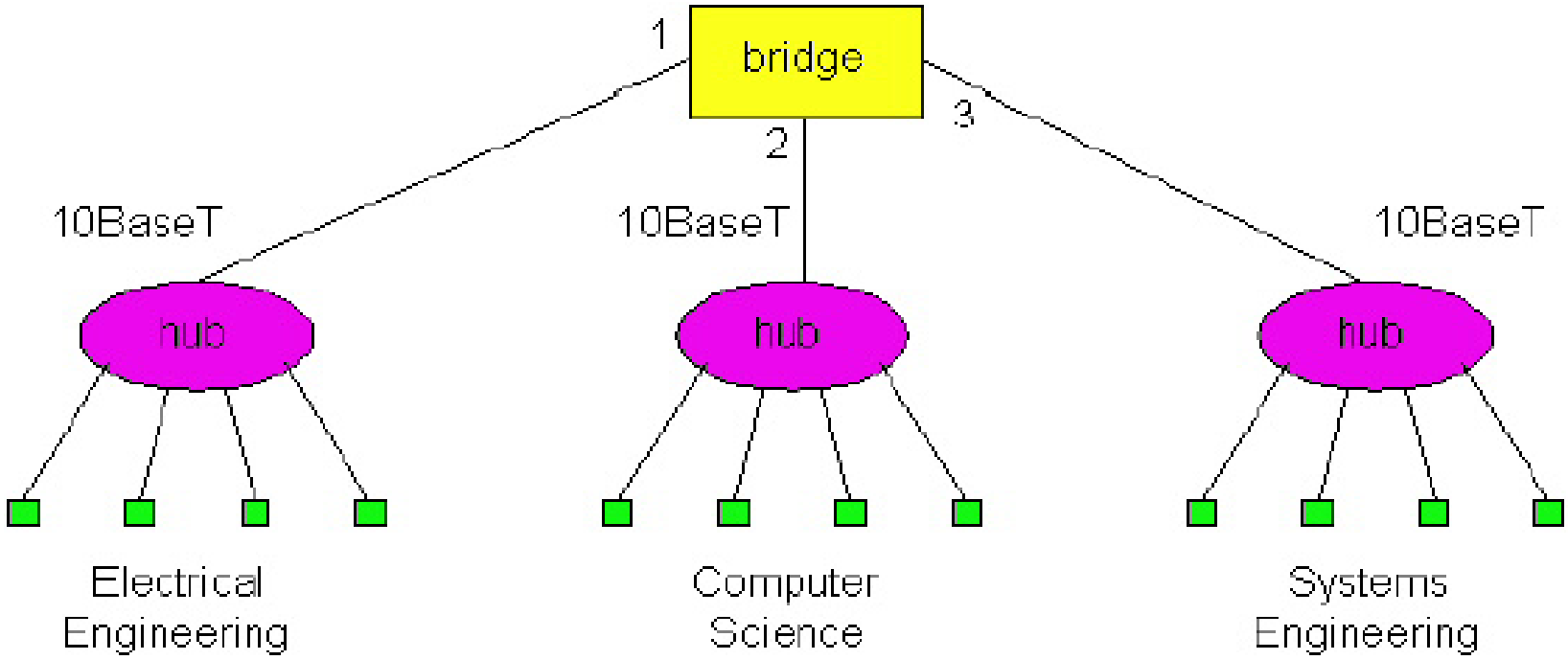
Ethernet: Segment-Kopplung

- Repeater verbinden 2 Mediensegmente auf Signalebene
- Hubs verbinden in Sternform
- Vorteile: Preis, Größere räumliche Ausdehnung als mit 1 Segment
- Nachteil: Alle Stationen teilen sich Bandbreite, Großer Kollisionsbereich
- Nachteil: Keine Verbindung unterschiedlicher Verfahren



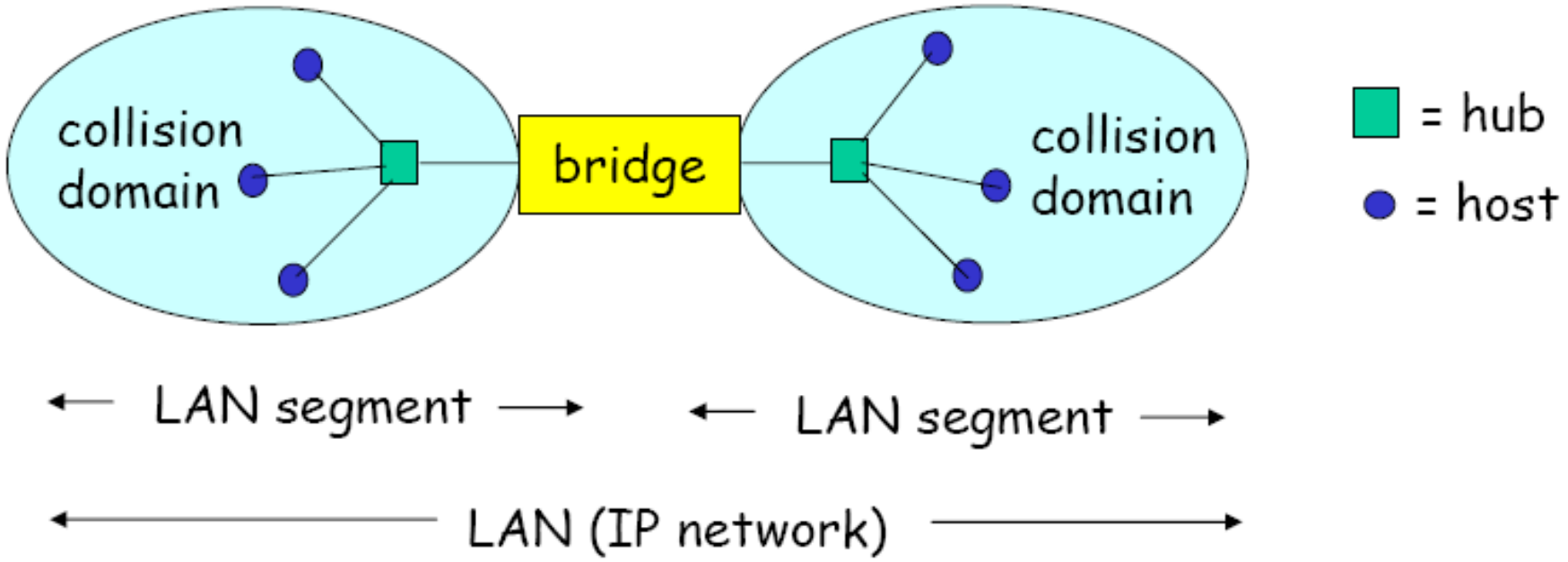
Ethernet: Segment-Kopplung

- Brücken (Bridges) koppeln auf Schicht 2 Ebene: Separate Kollisionsbereiche, selektive Weiterleitung, verschiedene Segmentverfahren möglich
- Selbstlernende Brücken, transparent für Hosts, billige Administration



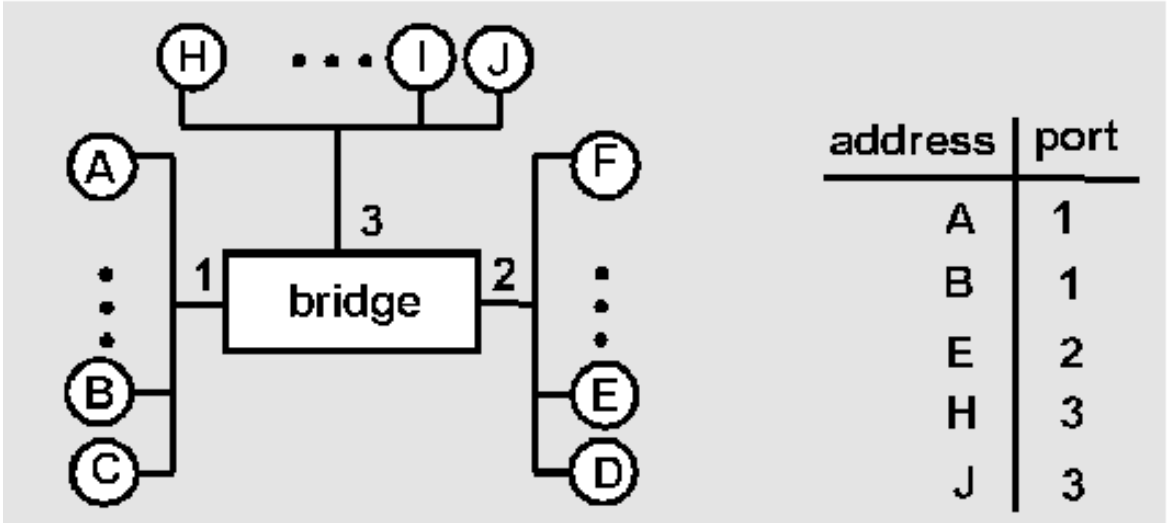
Brücken: Verkehrsisolation

- Brücke verbindet LAN-Segmente, die auf Signalebene isoliert sind
  - Separate Kollisionsbereiche
  - Größerer Summendurchsatz
  - Nur die Anzahl Stationen pro Segment ist beschränkt
  - Segmente können unterschiedlich sein (z.B. Token Ring – Ethernet)



Brücke: Weiterleitung

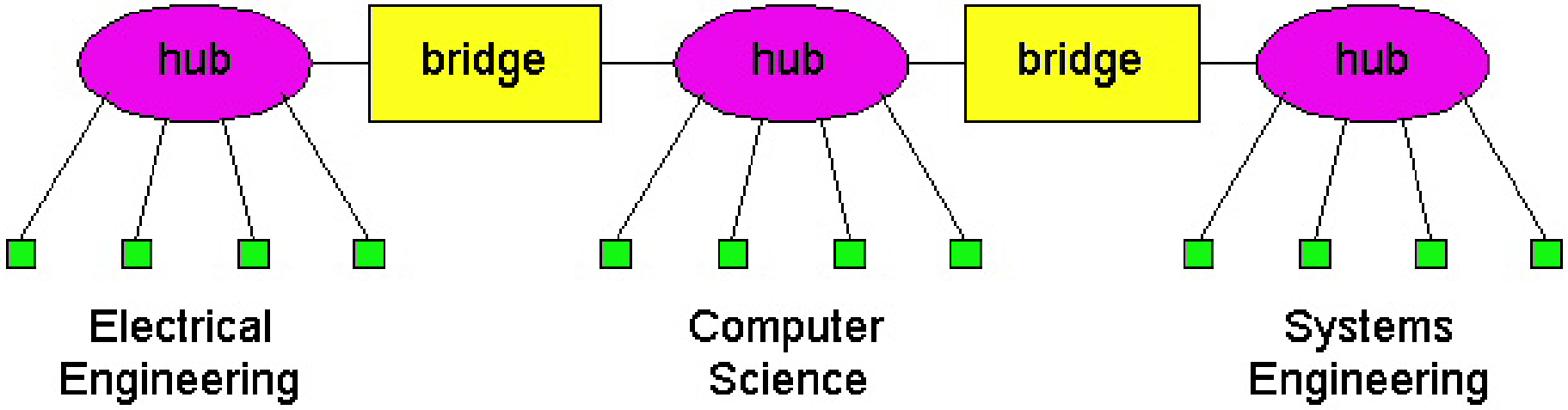
- Selektive Weiterleitung: wie Schicht 3, nur auf Schicht 2
  - Brücken-Tabelle aus Einträgen pro Zielknoten
    - » <LAN Adresse, Brückeninterface-Nummer, Zeitstempel>
    - » Ältere Einträge werden gelöscht
  - Selbstlern-Verfahren für neue Einträge:
    - » Wenn ein Ethernet-Frame empfangen wird, lernt man den Absender
    - » Wenn man ein Paket an eine Zieladresse weiterleiten muss, die man nicht kennt, sendet man eine Kopie dieses Ethernet-Frames an alle Interfaces (Fluten)



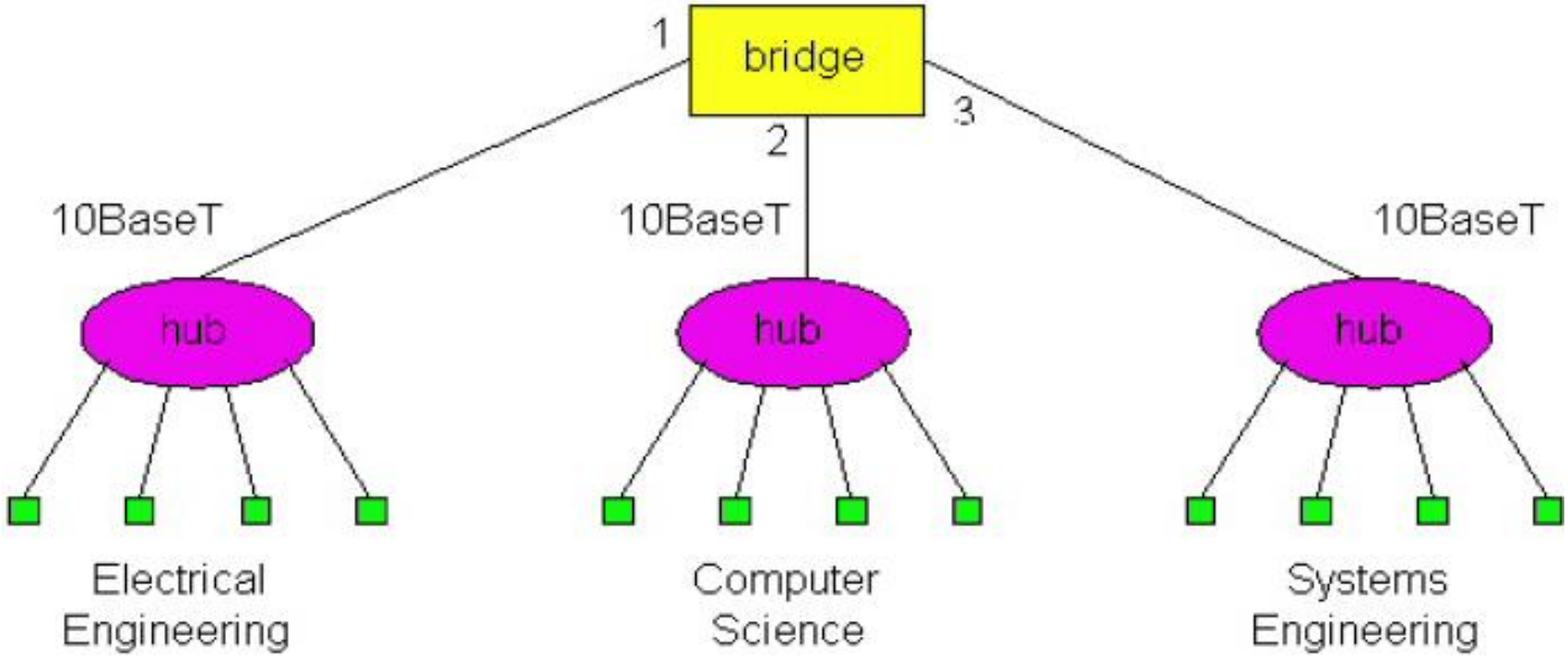
address	port
A	1
B	1
E	2
H	3
J	3

LAN-Netz-Bildung ohne Backbone-Netz

- Nicht zu empfehlen
  - Fehlertoleranz: Zentrale Durchleitungssegmente → Single Point of Failure
  - Summenverkehr, Lokalität des Verkehrs wird nicht ausgenutzt

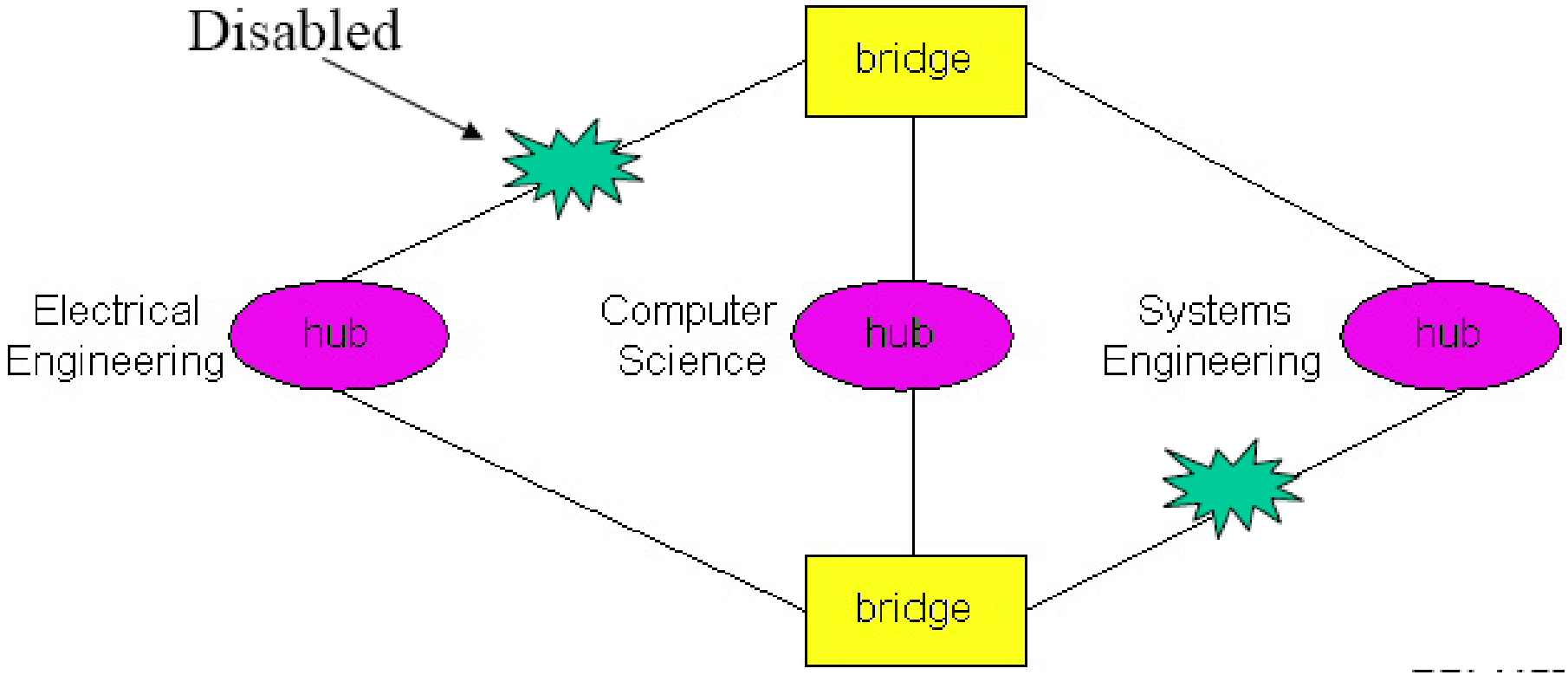


LAN-Netz-Bildung mit Backbone-Netz



- Fehlertoleranz: Subnetze sind selbständig
- Lokalität des Verkehrs wird ausgenutzt

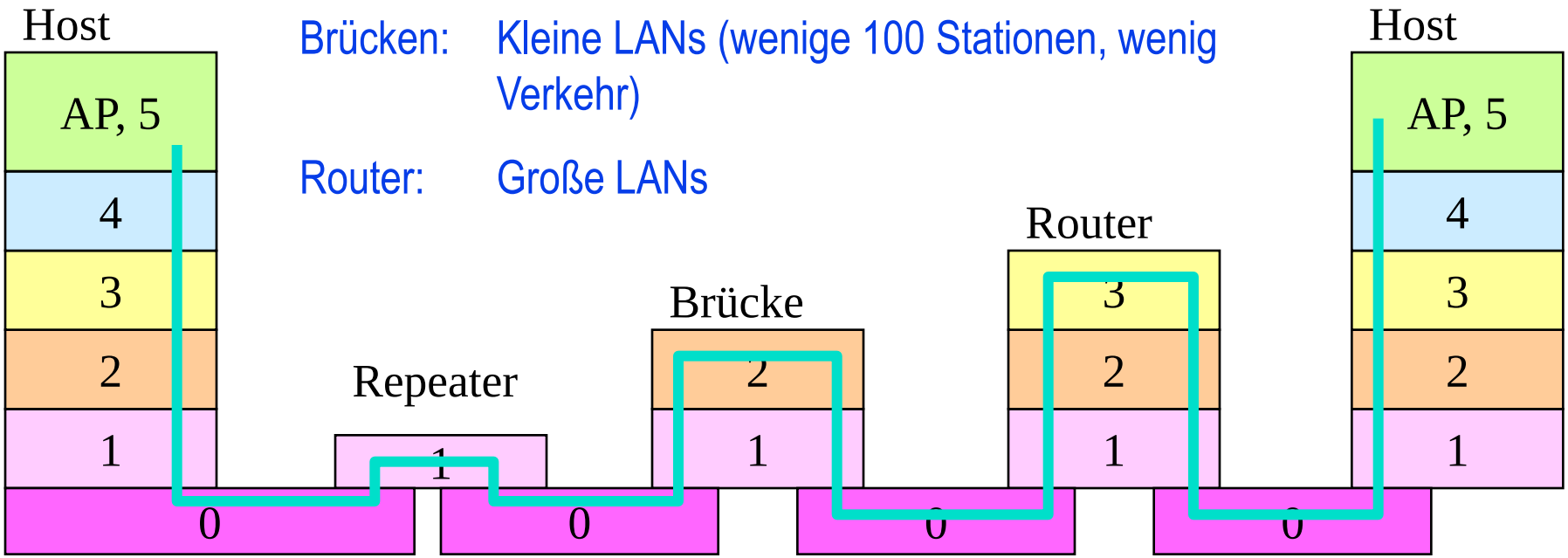
LAN-Netz-Bildung mit redundantem Backbone-Netz



- Fehlertoleranz: Alternative Wege in Backbone
- Brückentabellen müssen alternative Wege verwalten

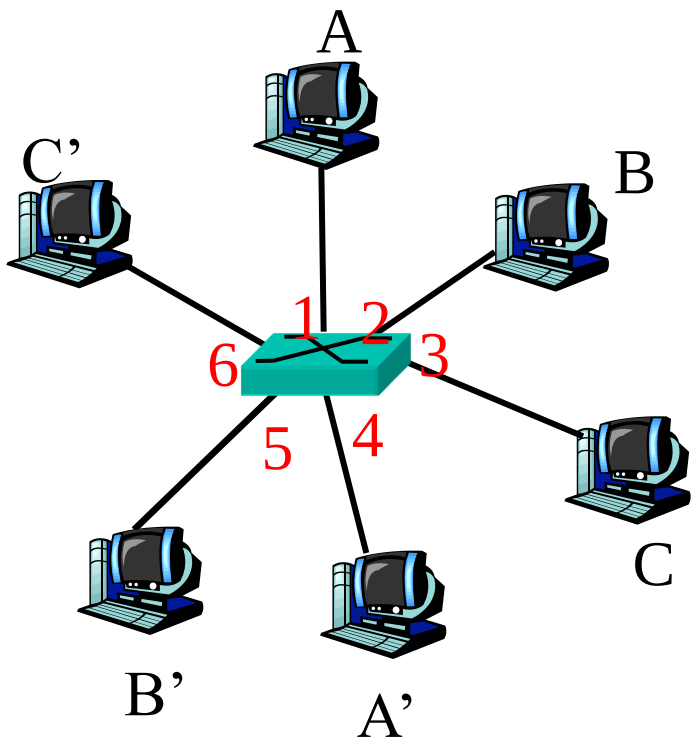
Brücken versus Router

- Beides sind „Store-and-Forward“-Vermittler
  - Router: Schicht 3 – Forwarding nach IP-Adresse
  - Brücke: Schicht 2 – Forwarding nach MAC-Adresse
- Router verwalten und pflegen Routing Tabellen
- Brücken haben einfache Brücken-Tabellen-Verwaltung (Selbstlernend)



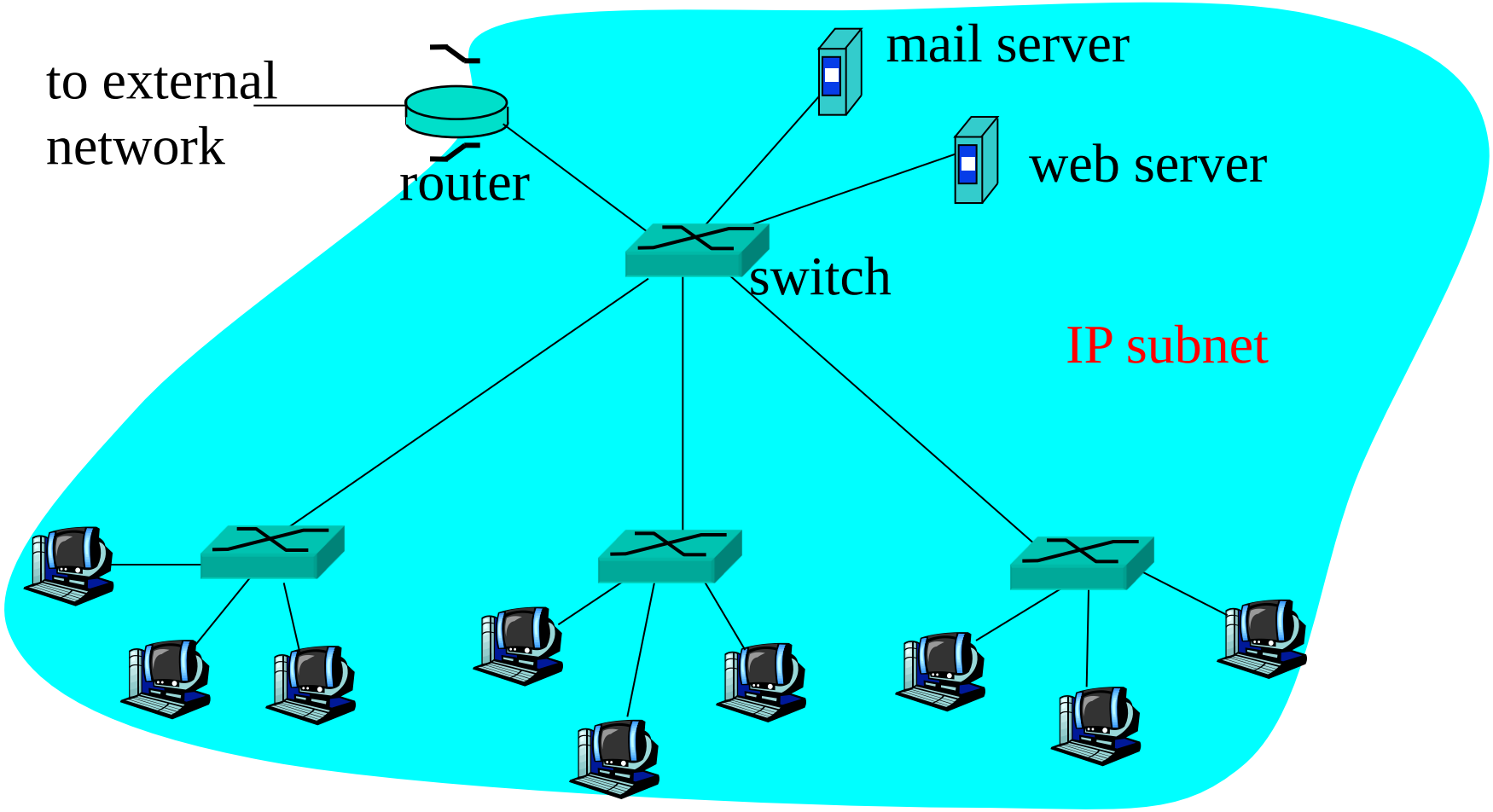
Ethernet - Switches

- Brücke mit sehr vielen Interfaces
  - Ziel: Nur 1 Host pro Segment
- Durchgriff-Switching (Cut-Through)
  - Frame wird ohne Zwischenpuffern weitergeleitet
- Switches mit unterschiedlichen Interfaces
  - 10 / 100 / 1000 MBit/sec
  - Token Ring
  - ...



Switch mit 6 Interfaces (1,2,3,4,5,6)

Ethernet – LAN Beispiel



Netzsegment-Kopplung: Vergleich

	hubs	bridges	routers	switches
traffic isolation	no	yes	yes	yes
plug & play	yes	yes	no	yes
optimal routing	no	no	yes	no
cut through	yes	no	no	yes

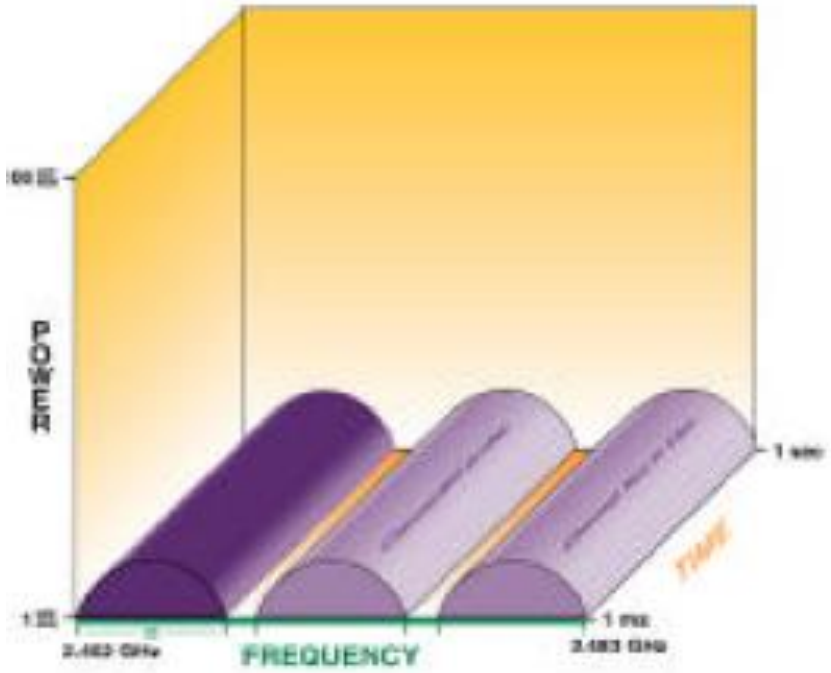
IEEE 802.11: Wirelesss LAN

- **CSMA/CA** (Collision Avoidance) über Nahbereich-Funk
- 2 Varianten:
  - Ad-hoc-Netze / Peer-to-Peer
  - Access Point / Base Station
- 802.11 b
  - 2.4 GHz unlicensed radio spectrum (2.4-2.462 GHz, 3 channels at 25 MHz apart), **up to 11 Mbps per channel**, Aufspreizung des Signals über gesamtes frequenzband (DSS) zur Störanfälligkeitsverminderung
- 802.11 a
  - 5-6 GHz range, **up to 54 Mbps**
- 802.11 g
  - 2.4 GHz range, **up to 54 Mbps**

Direct Sequence Modulation (DSS)

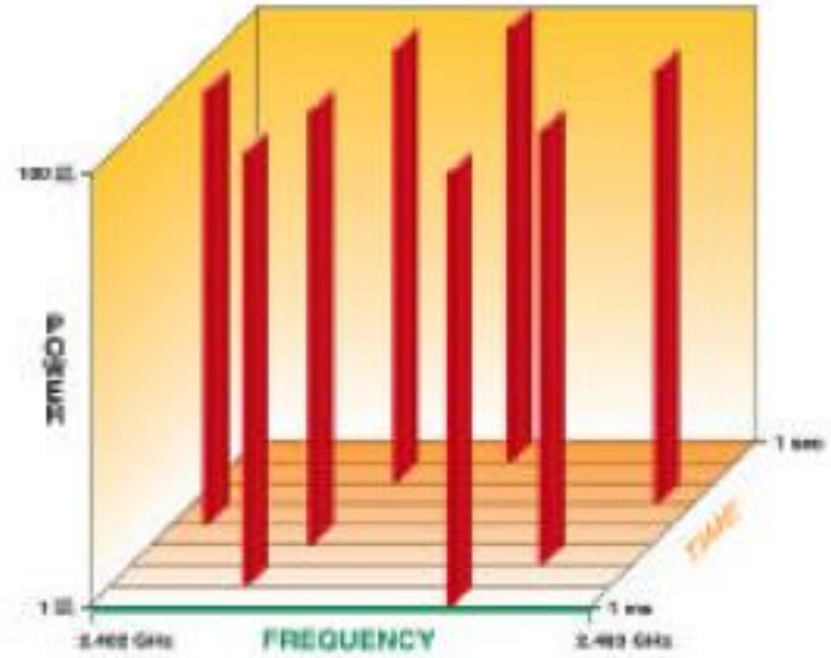
Direct Sequence

Bei WLAN



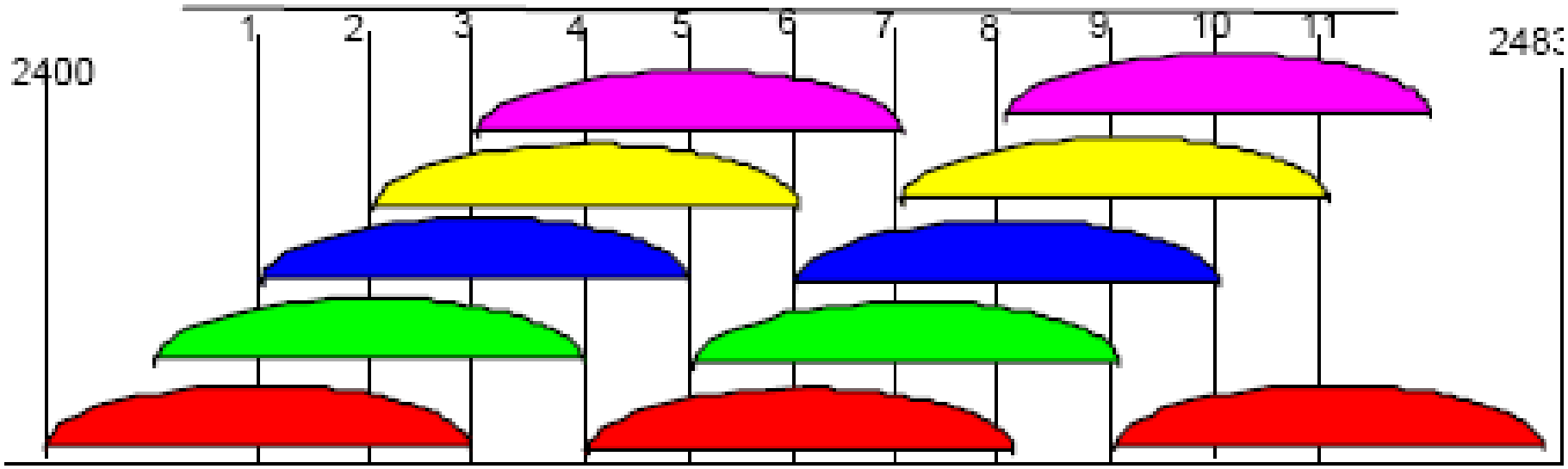
Frequency Hopping

Bei Bluetooth und bei „Internet über Stromnetz“



- Frequenzbereich wie Mikrowellen-Ofen: Störungen

Direct Sequence Modulation (DSS)



- **11mal 22 MHz breite Kanäle**
- **x “chips per bit”, jedes Bit redundant übertragen**
- **11 Mbps Datenrate**
- **3 nichtüberlappende Kanäle = 3 Access Points im gleichen Bereich möglich**

Direct Sequence Modulation (DSS)

- **Jedes Bit bekommt eine Zeichenkette “Chips” (Chipping Sequence), die parallel übertragen wird**
- **Minimale Chip Menge bei 1 und 2MB (BPSK/QPSK) sind 10 Chips und 8 Chips bei 11MB (CCK) Datenrate**

Beispiel-Datenpaket: 1001

Chipping code ist : 1=00110011011 und 0=11001100100

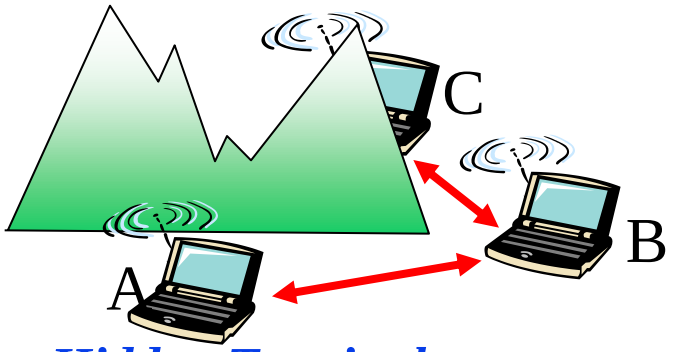
Die Übertragung sieht wie folgt aus:

00110011011	11001100100	11001100100	00110011011
1	0	0	1

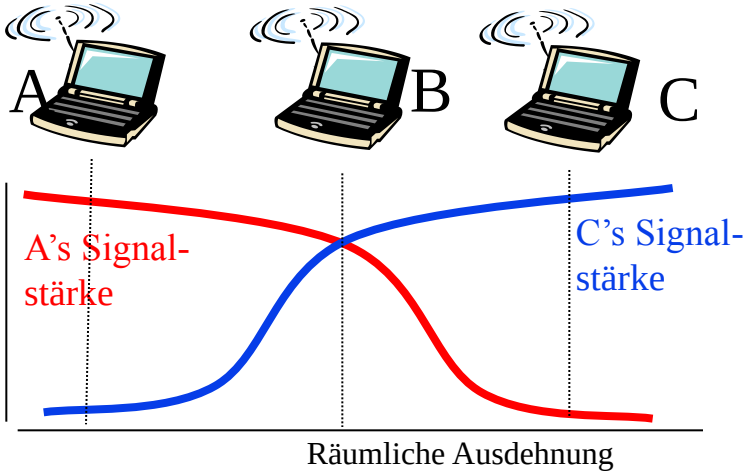
Resultat: mehr als 5 Bits müssten falsch übertragen werden um den Wert zu ändern !!!!!!

IEEE 802.11: Zugriffskontrolle

- Kollisionsproblem wie bei Draht-gebundenem Ethernet, Frequenzband ist Bus
- CSMA ist sinnvoll: Nur senden, wenn Kanal frei
- Collision Detection (CD) ist schlecht möglich
  - Hidden Terminal Problem
  - Senden und Mithören wäre bei Funk technisch überaus aufwendig



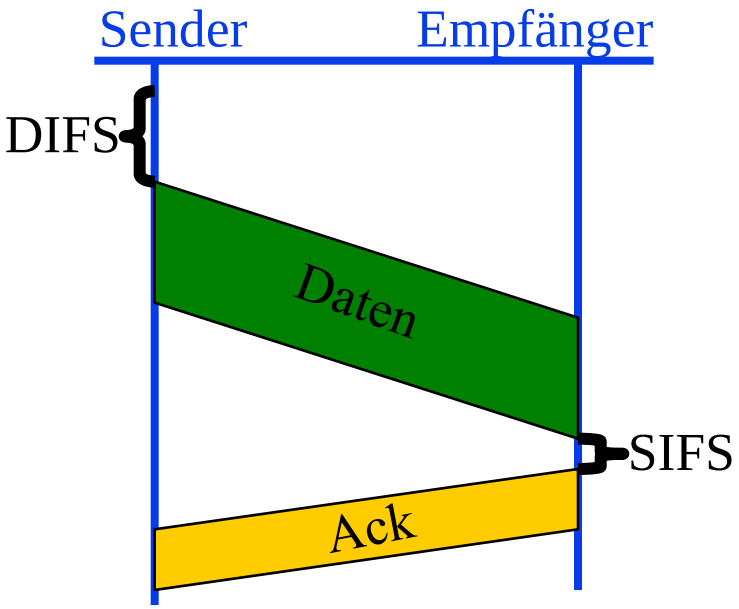
Hidden Terminal



Dämpfungsverlauf

IEEE 802.11 MAC Protocol: CSMA

- Zeitkonstanten:
  - DISF: Distributed Interframe Space
  - SISF: Short Interframe Space
- 802.11 CSMA: **sender**
  - if sense channel idle for **DISF** sec. then transmit entire frame
  - if sense channel busy then binary backoff
- 802.11 CSMA: **receiver**
  - if received OK return ACK after **SIFS** (ACK is needed due to hidden terminal problem)

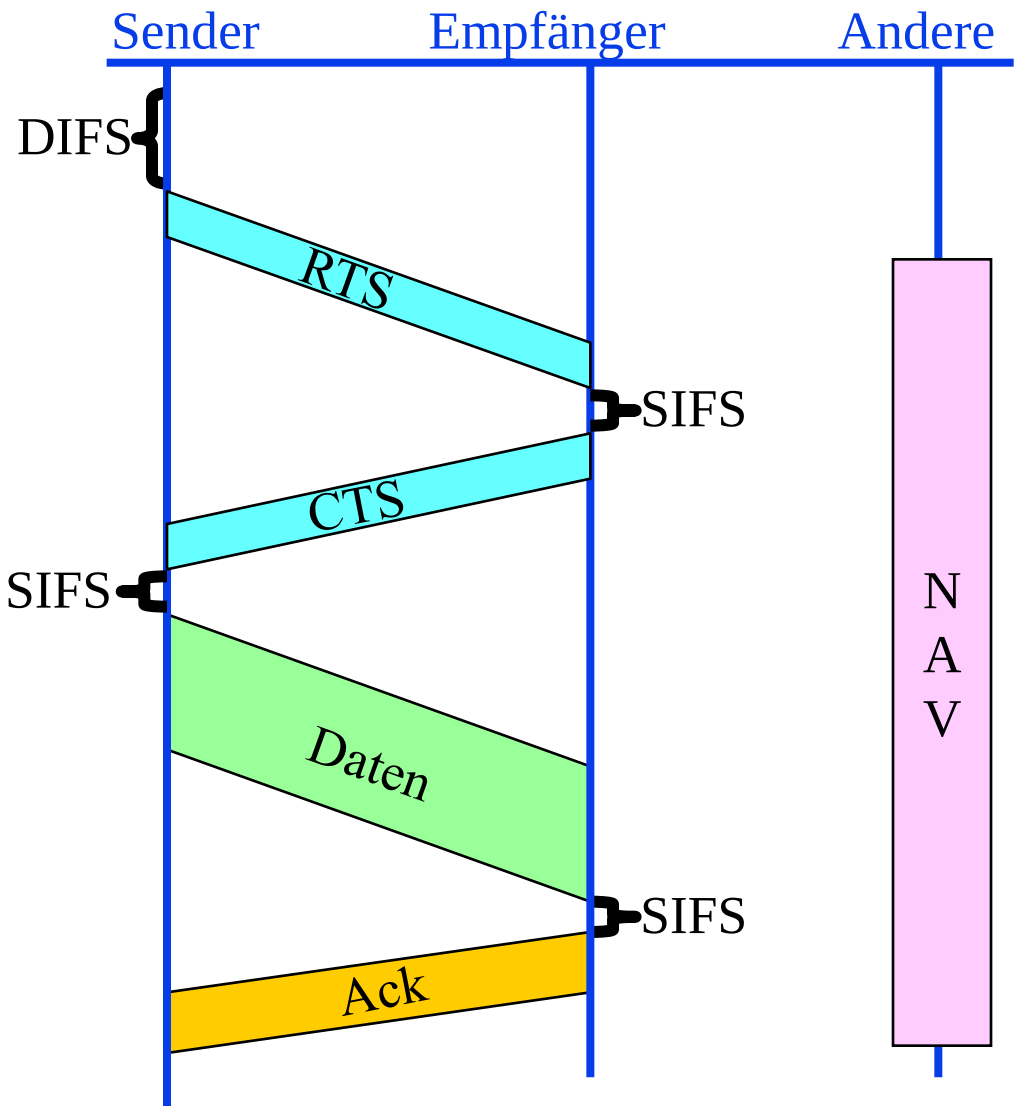


IEEE 802.11 MAC Protocol: CSMA/CA

- Problem
  - 2 Stationen, die sich gegenseitig nicht hören können, senden komplette (lange) Frames an Access Point
  - Für die gesamte Zeit ist Funkkanal nutzlos blockiert
- Lösung
  - Verwendung kurzer Reservierungspakete
    - » Request to Send (**RTS**), Clear to Send (**CTS**)
  - Stationen verfolgen die Medienreservierung mit „Network Allocation Vector (**NAV**)“
    - » Keine Station sendet RTS, wenn sie von fremden Reservierungen weiß

CSMA/CA: RTS – CTS Austausch

- RTS und CTS sind kurz:
  - geringere Kollisionswahr-scheinlichkeit wegen kurzer Dauer
  - Leistung ähnlich CD
- IEEE 802.11 unterstützt
  - CSMA
  - CSMA/CA
  - reservation polling from AP



Bluetooth

- Low-power, small radius, wireless networking technology
  - 10-100 meters
  - omnidirectional, not line-of-sight infrared
- Interconnects gadgets
- 2.4-2.5 GHz unlicensed radio band
  - up to 721 kbps
  - Interference from wireless LANs, digital cordless phones, microwave ovens:
    - » frequency hopping helps
- MAC protocol supports:
  - error correction
  - ARQ
- Each node has a 12-bit address

Point-to-Point-Protocol: PPP

- Verbindung zum Internet über Wählleitungen
  transportiert IP-Pakete als Link-Layer-Frame-Nutzdaten
- Verbindungsorientierter Dienst
  - Verbindungsaufbauphase
  - Datentransferphase
  - Verbindungsabbauphase
- HDLC-artige Übertragung
- Beim Verbindungsaufbau können die Partner authentifiziert werden
  - z.B. Password Authentication Protocol (PAP)
    Challenge Response Authentication Protocol (CHAP)
- Varianten:
  - PPPoE: PPP over Ethernet (Einsatz bei DSL)
  - PPTP: Point-to-Point Tunneling Protocol

Point-to-Point-Protocol: PPP

- PPP-Dateneinheit

Protocol (8/16 Bit)	Information/Daten	Padding

  - Protokollkennung: IP / Apple Talk / Novell / ..

- HDLC-Frame

Flag	Address	Control	PPP Dateneinheit	FCS	Flag
01111110	11111111	00000011			01111110

  - Ist die PDU des Link-Protokolls
  - packt PPP-Dateneinheit als Nutzdatum ein
  - Flags dienen zur Synchronisation und Paketstart/ende-erkennung
  - Adresse spielt bei 2-Punkt-Leitung keine Rolle
  - Keine Sequenzzahlen, deshalb konstante Kontrollbits