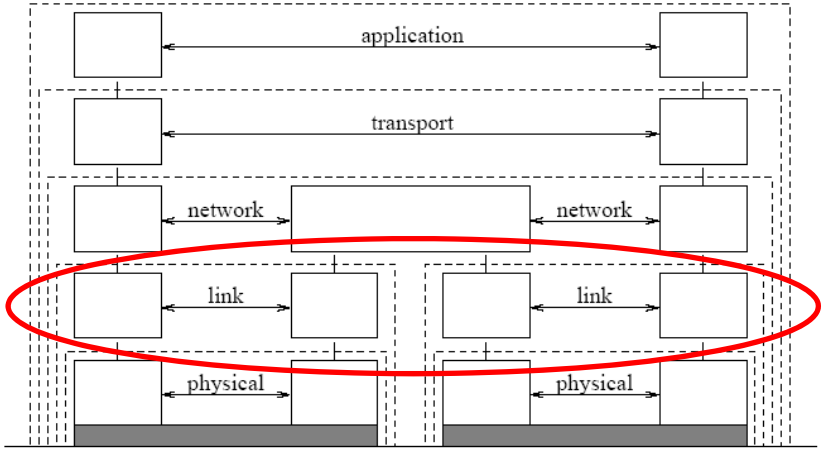
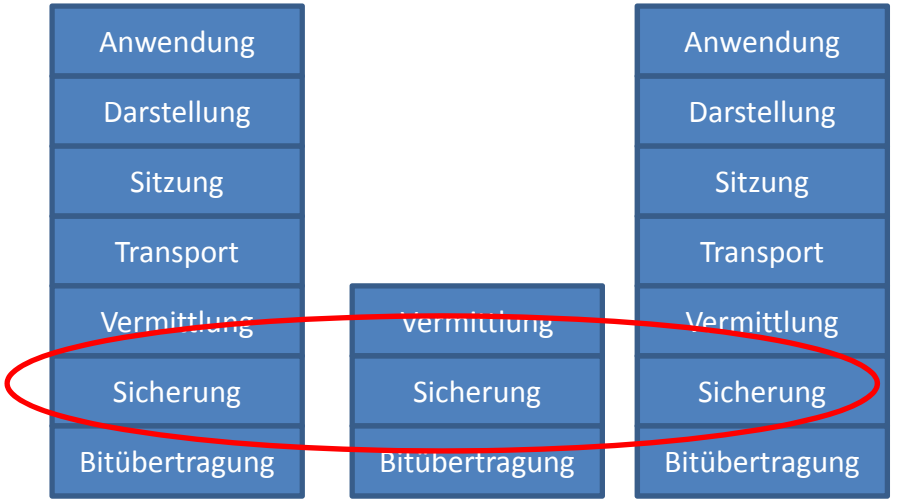
Die Sicherungsschicht

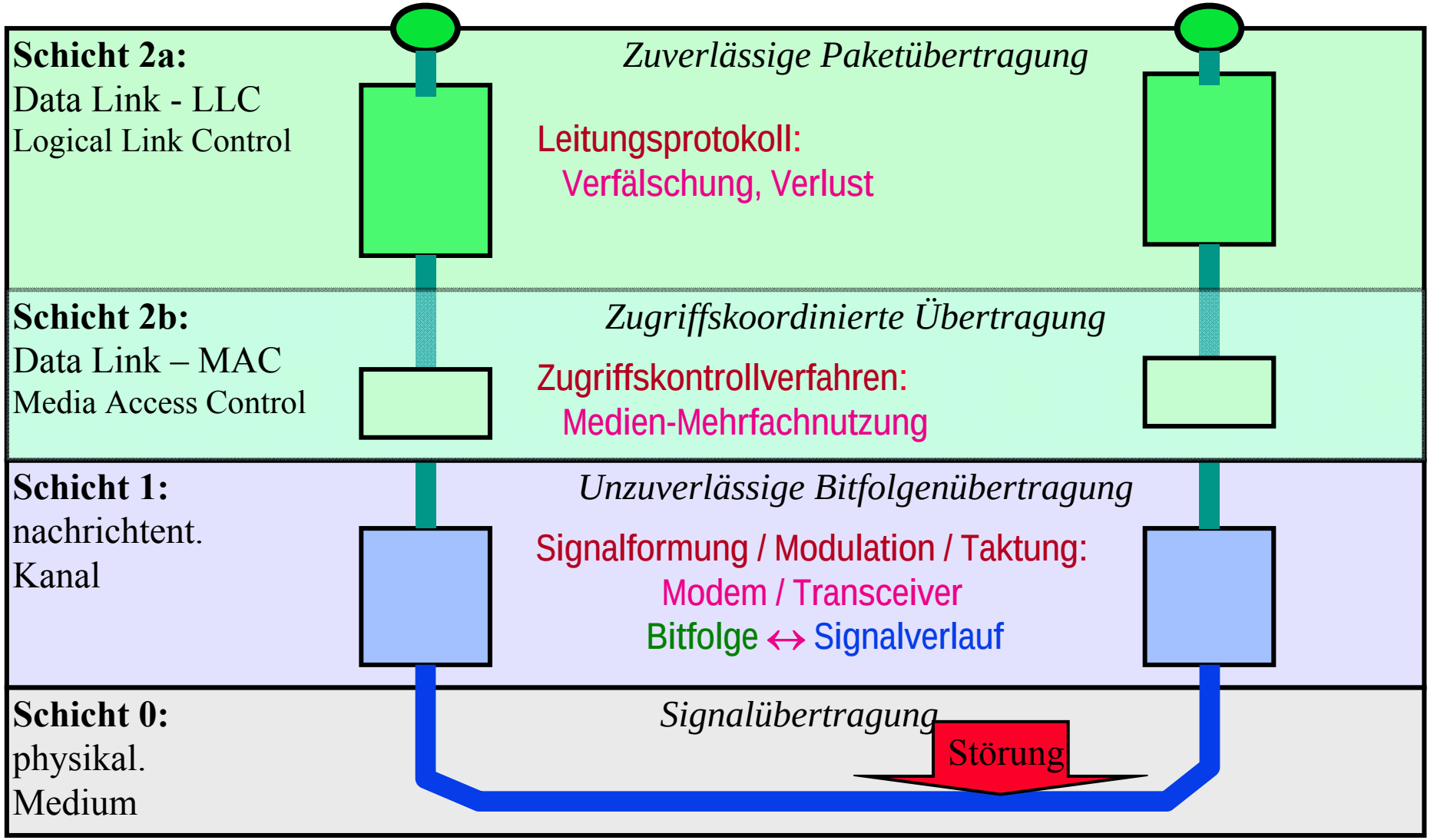
Gliederung

- Aufgaben
- Fehlererkennung und -korrektur
- Medienzugriff
- Adressierung
- Ethernet
- Switching
- Funk-LAN
- Point-to-Point Protokolle
- Datencenter
- Der Weg durch die Protokollschichten





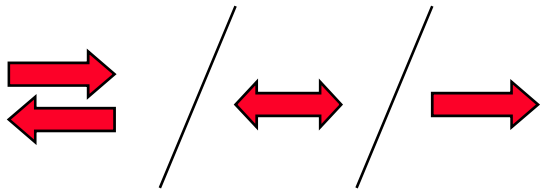
Schichten im Kontext eines physikal. Mediums



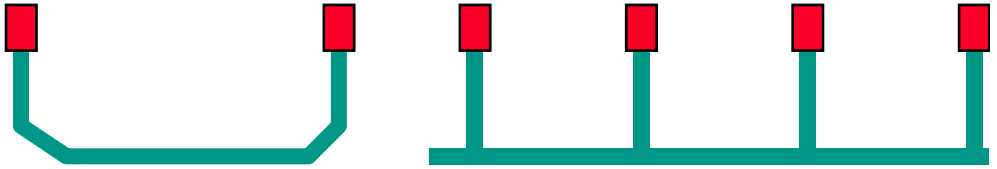
Aufgaben der Schicht 2

- 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

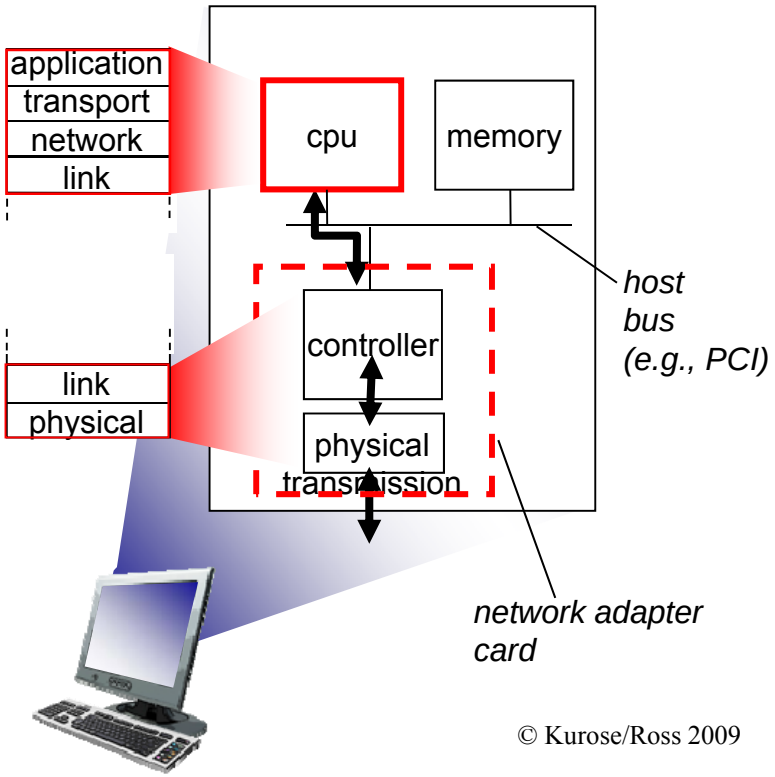


Aufgaben der Schicht 2

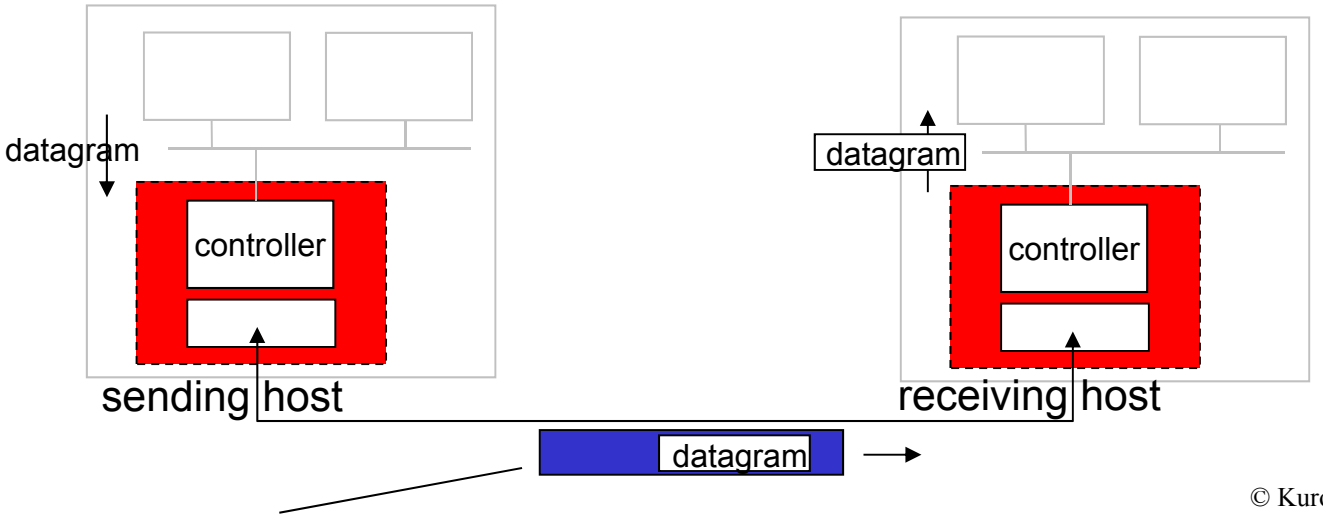
- *Bildung von Frames, Medienzugriff:*
  - Kapselung eines Datagramms in ein Frame unter Hinzufügen von Header und Trailer
  - Medienzugriff bei geteilten Medien
  - "MAC" –Adresse im Frame-Header zur Identifikation von Quelle und Ziel
    - Diese unterscheidet sich von der IP-Adresse!
- *Zuverlässige Übertragung zwischen benachbarten Knoten*
  - Methoden dazu haben wir bereits kennengelernt (Kap. 3)!
  - Wird nicht genutzt auf Medien mit geringer Fehlerwahrscheinlichkeit (z.B. Fiberglas, einige Twisted Pair-Varianten)
  - Notwendig für Medien mit hoher Fehlerrate (z.B. drahtlos)
  - Warum benötigt man zuverlässige Übertragung auf den Schichten 2 und 4?

NIC: Netzinterface-Controller (Netzadapter)

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



Kommunikation zwischen Adaptern



- Sender:
  - Einpacken des Datagramms in einen Frame
  - Hinzufügen von Bits zur Fehlererkennung/-korrektur
  - Medienzugriff, Flusskontrolle
- Empfänger
  - Fehlerprüfung
  - Auspacken des Datagramms und Übergabe an die höheren Schichten

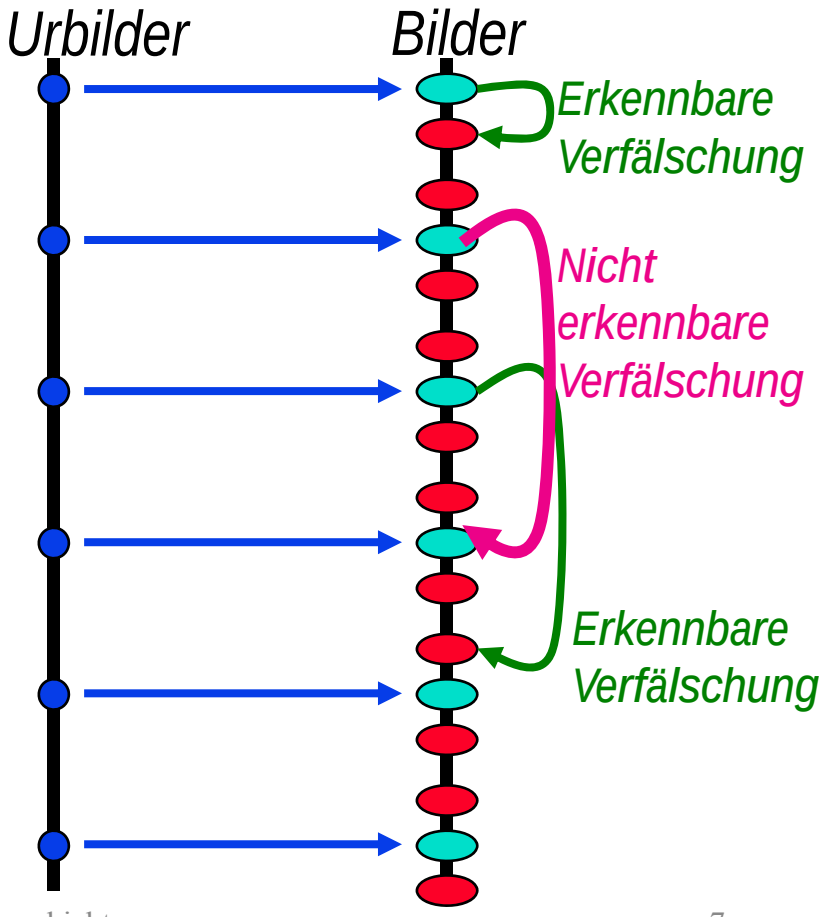
Verfälschungerkennung

Prinzip

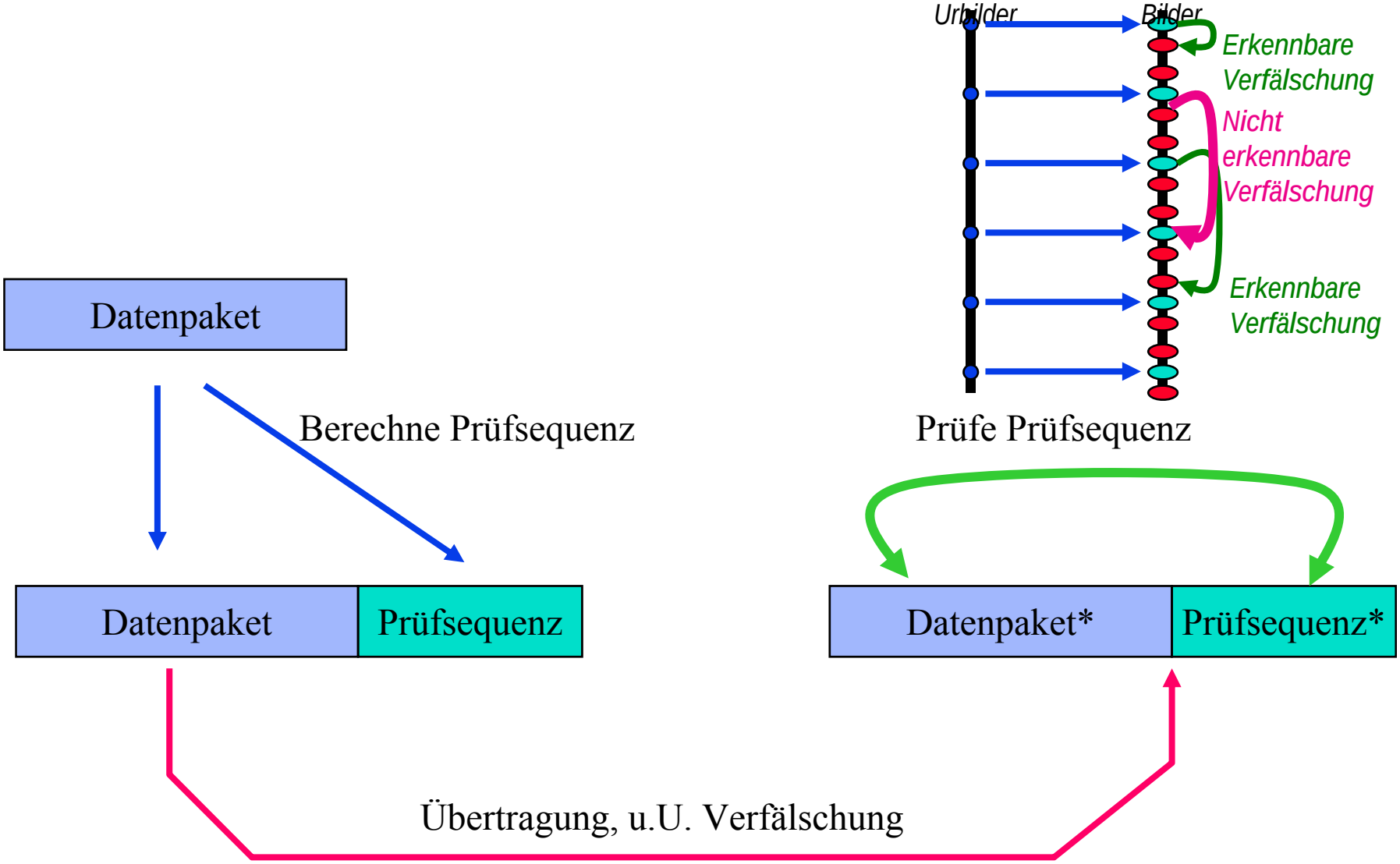
- Bilde Daten auf erweiterte Daten ab, so dass Redundanz entsteht
- Nutze Redundanz, um mit ausreichender Zuverlässigkeit aufgetretene Verfälschungen zu erkennen

Verfahren

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



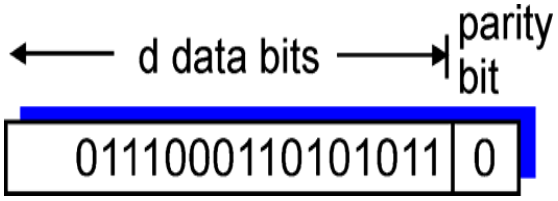
Verfälschungerkennung



Parity-Prüfung / Bitparitätsprüfung

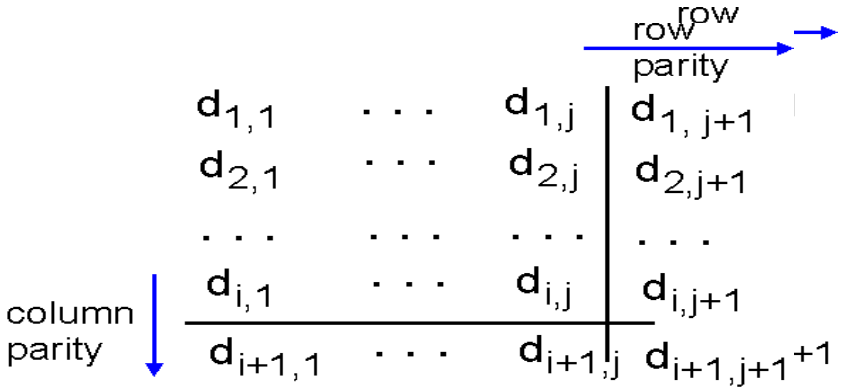
Single Bit Parity:

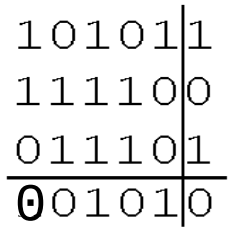
Erkennung von 1-Bit-Fehlern



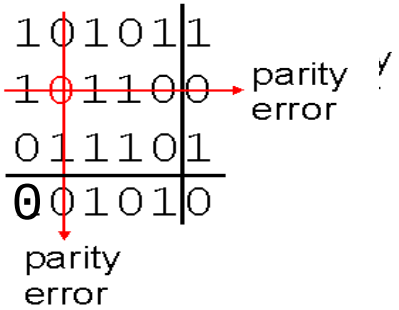
Two Dimensional Bit Parity:

Erkennen und Korrigieren von 1-Bit-Fehlern





no errors



correctable single bit error

Internet-Prüfsumme (Wiederholung)

Ziel: Fehlererkennung (z.B. gekippte Bits) im übertragenen Paket
(wird nur auf der Transportschicht genutzt!)

Sender:

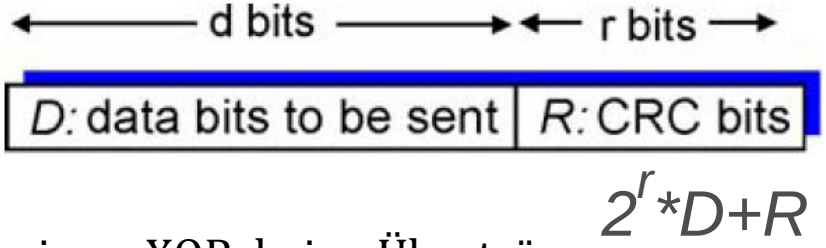
- Interpretiere jedes Segment als Sequenz von 16-bit Zahlen
- Prüfsumme: (1er-Komplement) Summe des Segmentinhalts
- Sender schreibt die Prüfsumme in das entsprechende Feld (UDP oder TCP)

Empfänger:

- Berechne die Prüfsumme des empfangenen Segments
- Vergleiche die berechneten Prüfsumme mit dem Wert im Prüfsummenfeld
  Identisch?
  - NEIN - Fehler
  - JA – kein Fehler erkannt
    Wurde das Segment damit korrekt übertragen?

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**> ganzzahlig durch **G** teilbar ist; **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, Ethernet, 802.11, ...)
- 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: Senderseite

D = 1101011011 G = 10011

2^4*D = 11010110110000

11010110110000 / 10011 = 1100001010

```
10011
010011
 10011
 0000010110
      10011
      0010100
        10011
        0011110
```

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

Rest = 1110

Gesendete Nachricht: 11010110111110

Beispiel: Empfängerseite 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: Empfängerseite nach verfälschter Übertragung

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

11100110111010 / 10011 = 1111011101
10011
011111
 10011
 011001
  10011
  010100
   10011
   0011111
     10011
     011001
      10011
      010100
       10011
       0011110
         10011
         01101
```

Rest = 1101 → verfälscht

CRC: Weitverbreitetes Verfahren

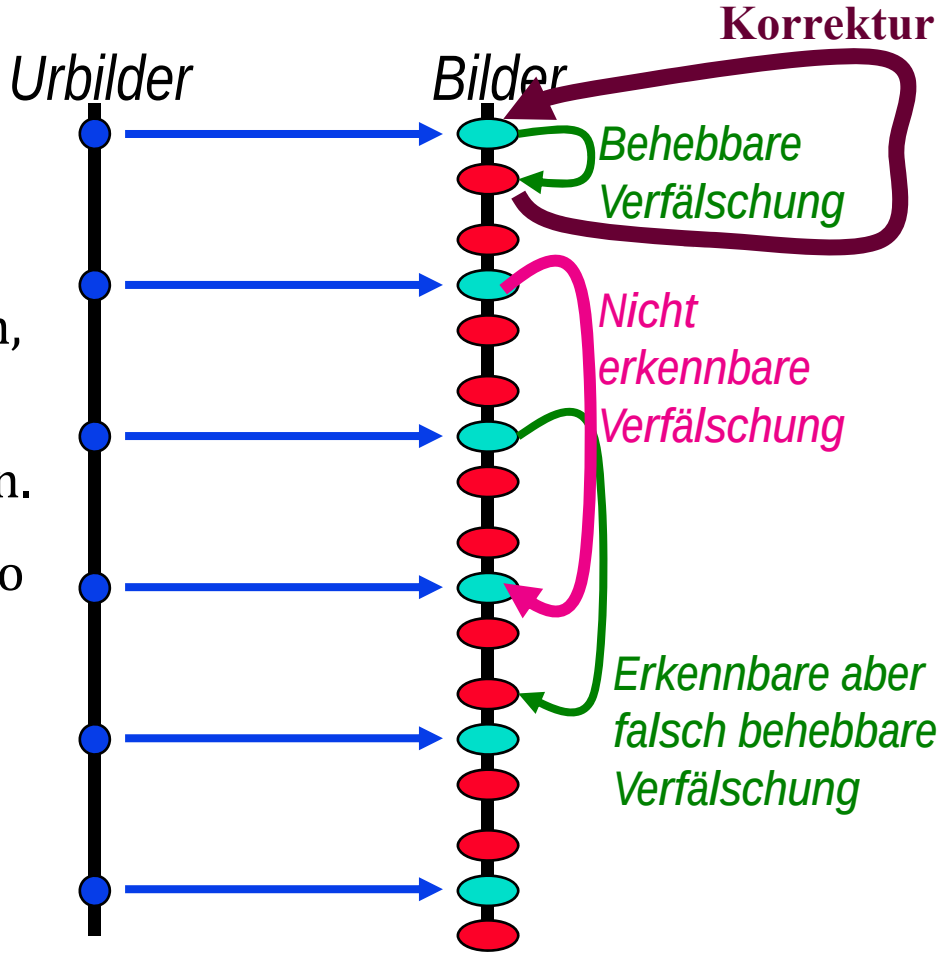
- ATM, HDLC, Ethernet-LLC, …
- Standard-Generatorpolynome:
  - **ISO CRC-12**: $x^{12} + x^{11} + x^3 + x^2 + x + 1$ **1 1000 0000 1111**
  - **ISO CRC-16**: $x^{16} + x^{15} + x^2 + 1$ **1 1000 0000 0000 0011**
  - **CRC-CCITT**: $x^{16} + x^{12} + x^5 + 1$ **1 0001 0000 0010 0001**
- alle enthalten (x + 1) als Faktor:
  - ungerade Anzahl von Bit-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
000 0010 1
110 0011 0

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 Ergänzungsbit (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-Bitits 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. $001_2 + 100_2 = 101_2$

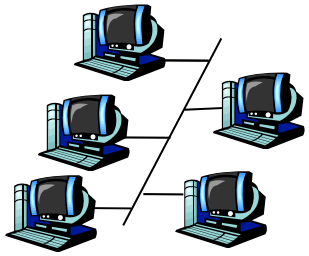
Für C_2': 1. Eins von rechts: in 3, 5, 7: Test Check-Bit ist ok.
2. Eins von rechts: in 3, 6, 7: Test Check-Bit ist ok.
3. Eins von rechts: in 5, 6, 7: Test Check-Bit ist falsch.
$\Rightarrow$ Position des falschen Bits: 4 bzw. 100_2

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 kodierte Spalte gesendet, dann sukzessive die nächsten Spalten.
  - 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 ursprüngliche Wort kann dann richtig rekonstruiert werden.

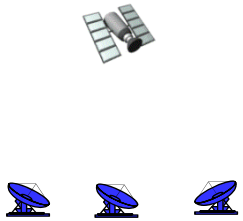
Medienzugriff (MAC: Media Access Control)

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



Geteiltes Medium
(z.B. Ethernet)

Geteilte Frequenz
(z.B., 802.11 WiFi)



Geteilte Frequenz
(satellite)

Menschen auf einer Party
(geteiltes Medium, Akustik)

Problem: *Durcheinander-Reden*

Im selben Zeitintervall soll höchstens eine Station senden
→ Zugriffskontrolle

MAC: Ideale Lösung

Ein Sammelkanal könne $\boldsymbol{R}$ Bit pro Sekunde übertragen

Ideale Lösung (angestrebt)

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

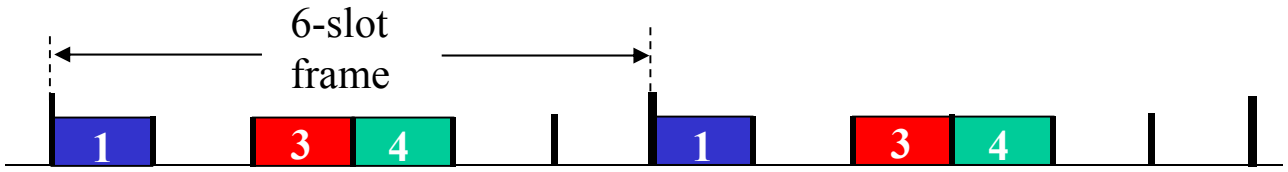
MAC: Verfahrensgrundtypen

- 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

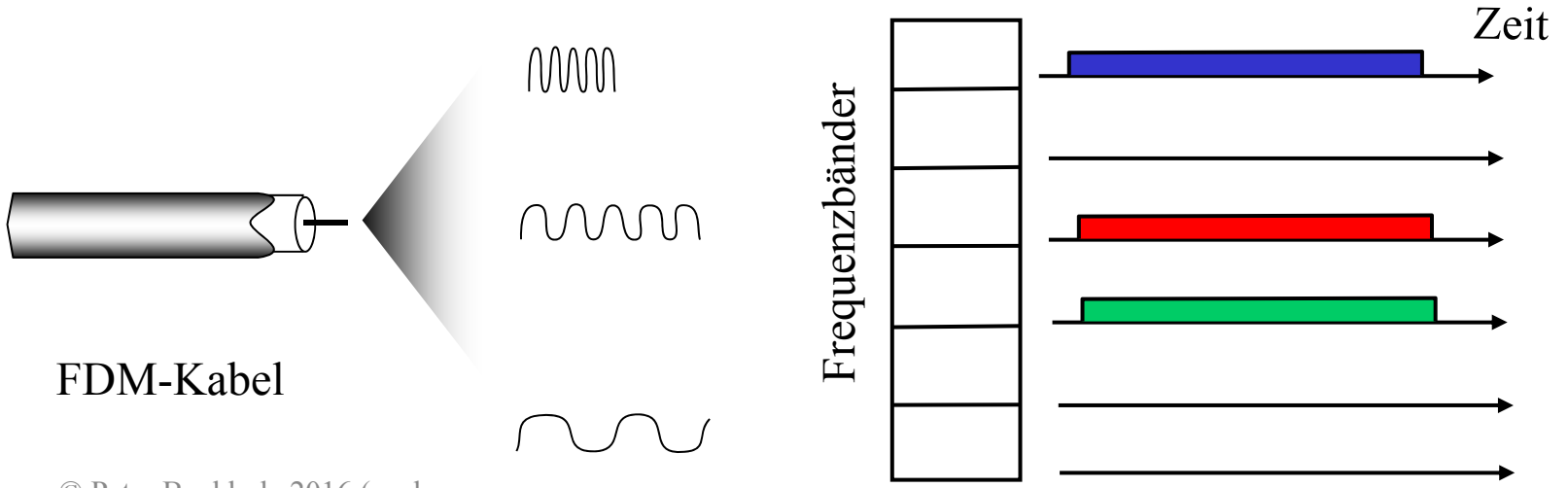
- Kanalzugriff in "Runden"
- Jede Station bekommt Slot fester Länge (Länge in Paketen pro Zeit) in jeder Runde
- Unbenutzte Slots bleiben leer
- Beispiel: LAN mit 6 Stationen, 1,3,4 wollen senden, Slots 2,5,6 bleiben leer



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 sende, Frequenzbänder 2,5,6 bleiben ungenutzt



MAC: Neues Verfahren CDMA

- 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.

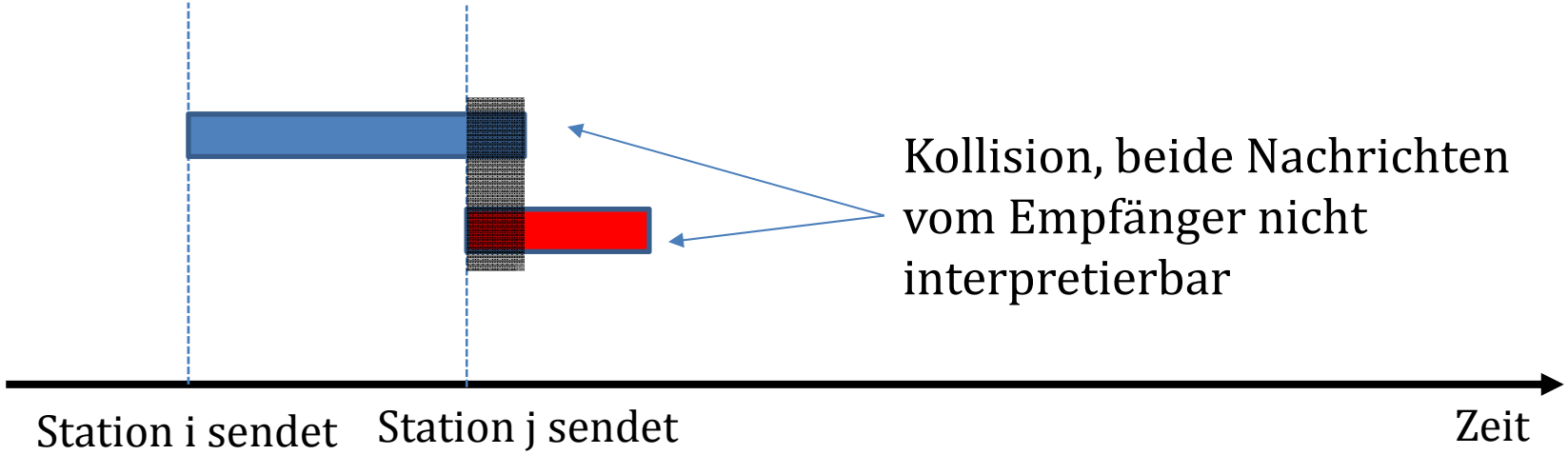
Einsatzgebiete: UMTS / 3G Mobilfunk

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
  - 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 warte eine Paketlänge, sende dann mit Wahrscheinlichkeit p, warte mit W. (1-p) eine weitere Paketlänge ….

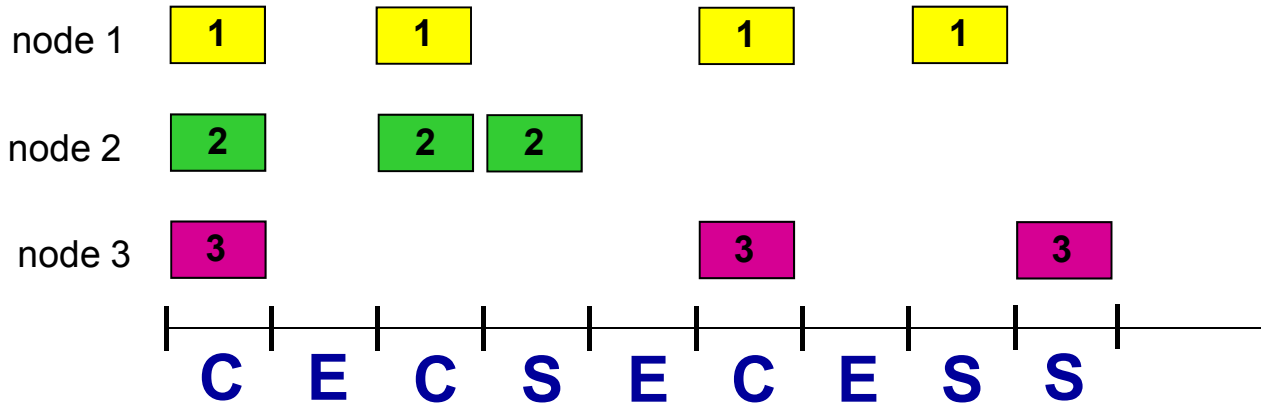


- Problem: W(Paket durch Kollision zerstört) deutlich > 0
- Paket ist während der ganzen Sendezeit verletzlich:

  -Idee: Slotted Aloha

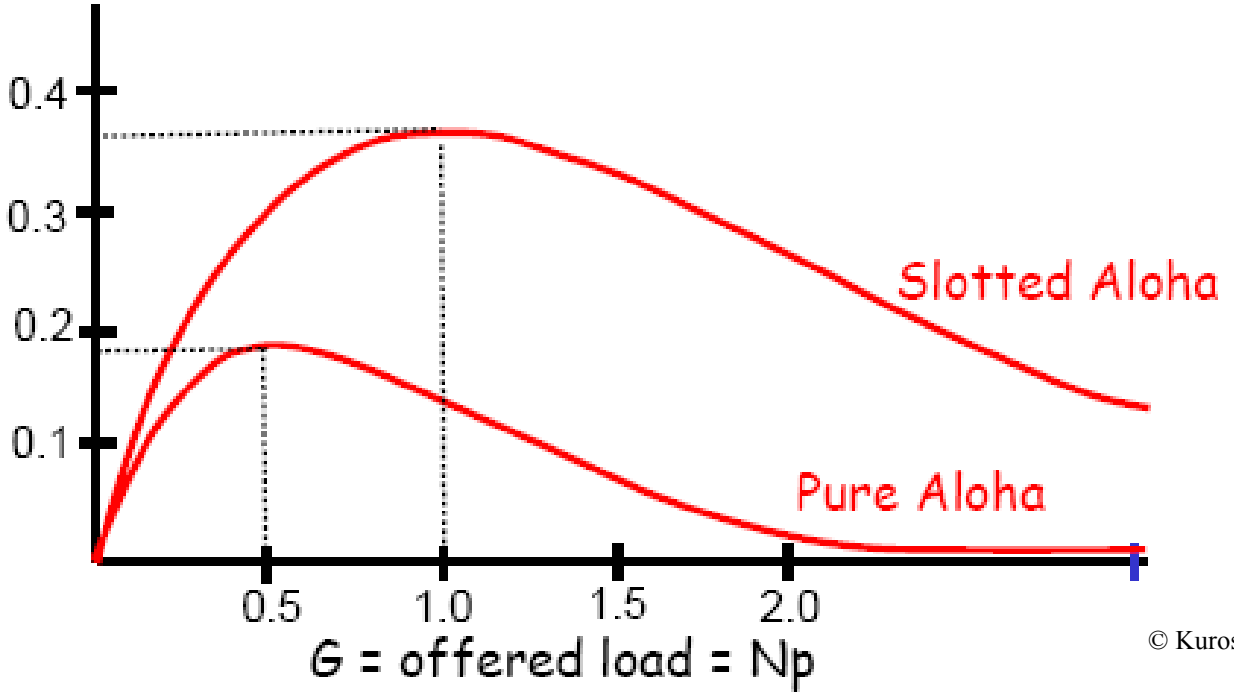
MAC: Konkurrierende Zuteilung – Slotted Aloha

- Es gibt ein Zeitraster, das Zeit in Intervalle / Slots / Schlitze einteilt
- Sende bei Sendwunsch bei Beginn des nächsten Schlitzes
- Sende ganzes Paket (alle Pakete haben die gleiche Größe): Es kommt an oder, falls Kollision auftritt, kommt es nicht an



- 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%

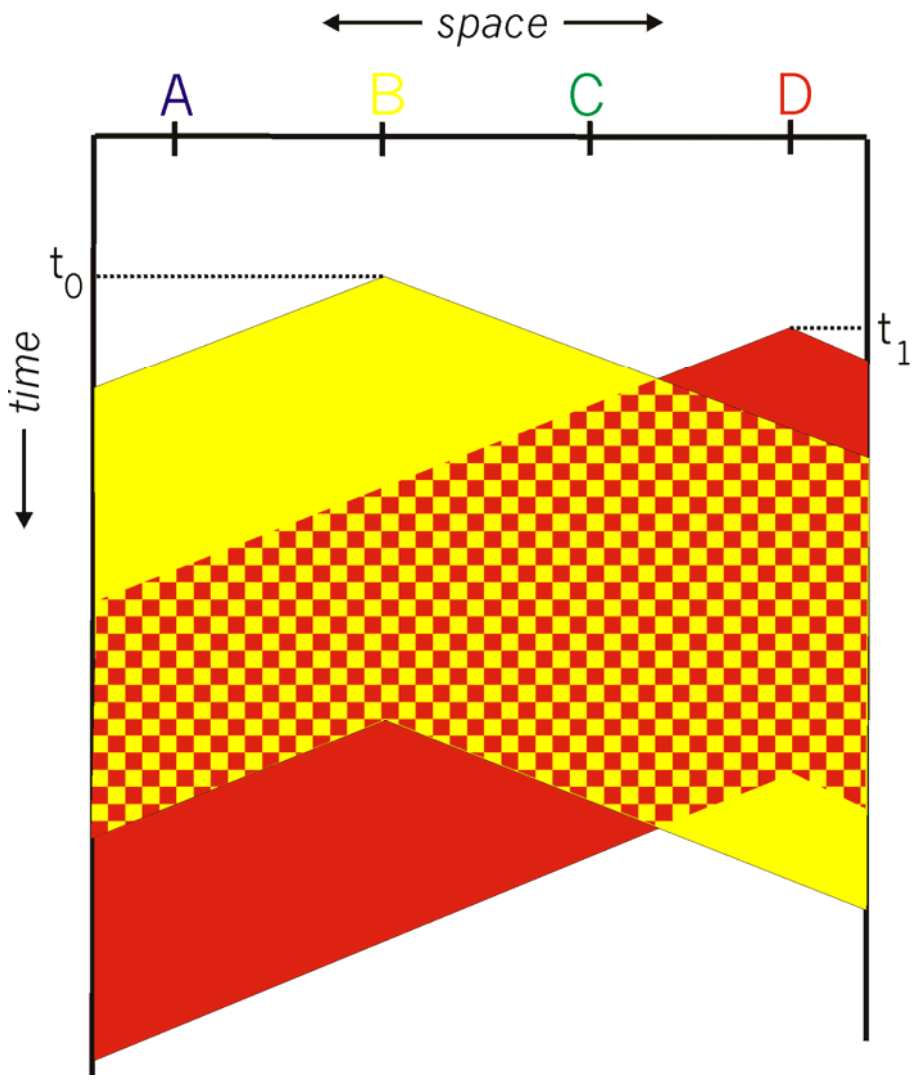
(jeweils unter Annahme zufälliger Last)

Aloha: Weitere Verbesserungen

- Keine Verbesserungsmöglichkeit, wenn Sender sich gegenseitig nicht hören oder wenn lange Signallaufzeiten vorhanden sind
- Aber sonst können folgende Schritte durchgeführt werden:
  - Hören vor dem Senden, ob Kanal frei
  - Hören, während des Sendens, ob Paketsendung 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)

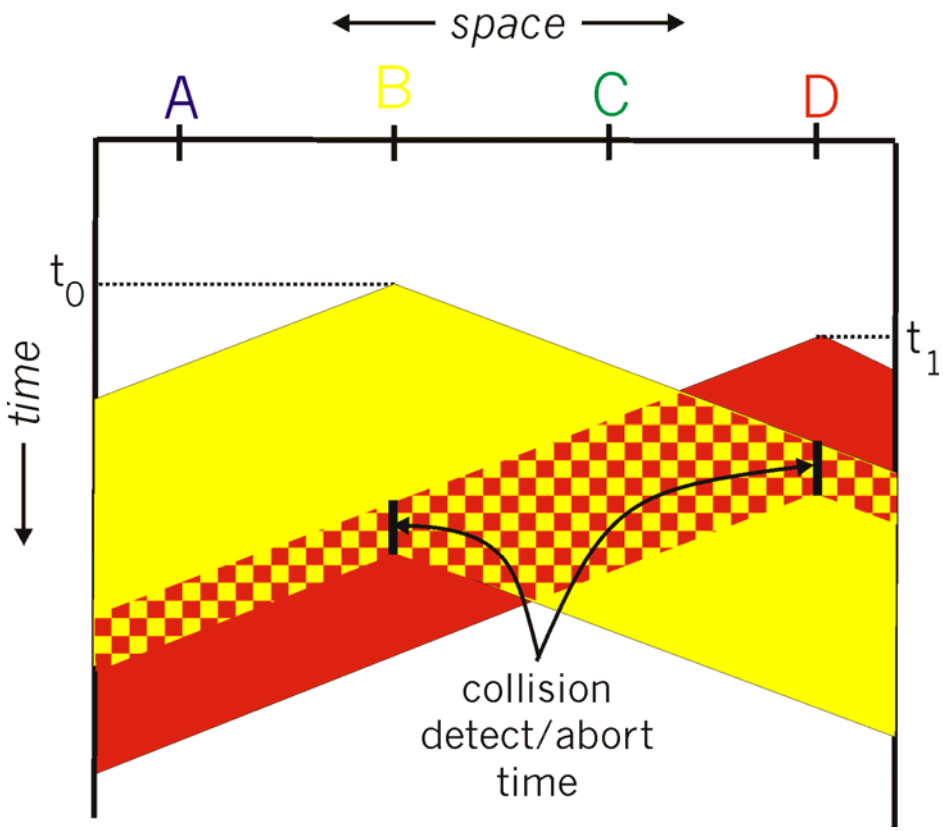
- Hören, ob anderer schon sendet, bevor man sendet
  - Wenn Kanal frei: Senden eines Pakets
  - Wenn Kanal belegt: Für die Dauer eines zufällig langen Zeitintervalls warten
- Analogie: Nicht unterbrechen, wenn jemand spricht.
- Kollisionen sind dennoch möglich
  - Signallaufzeiten bedingen, dass jemand 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



© Kurose/Ross 2009

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 oft 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



© Kurose/Ross 2009

Zentrale Rechtevergabe

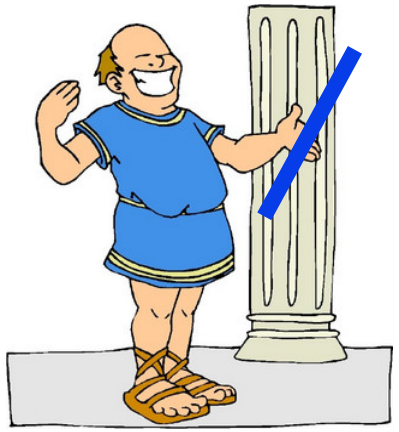
(der Vollständigkeit halber und zum Vergleich)

- Verfahren: Polling
  - „Busmaster“ fragt 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*

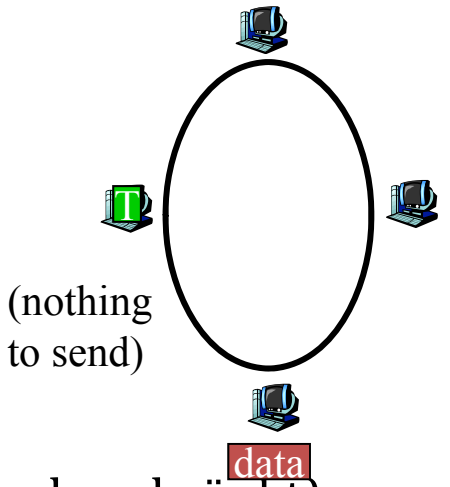
Wer reden will, muss warten, bis ihm der Rednerstab gebracht wird.



- 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?

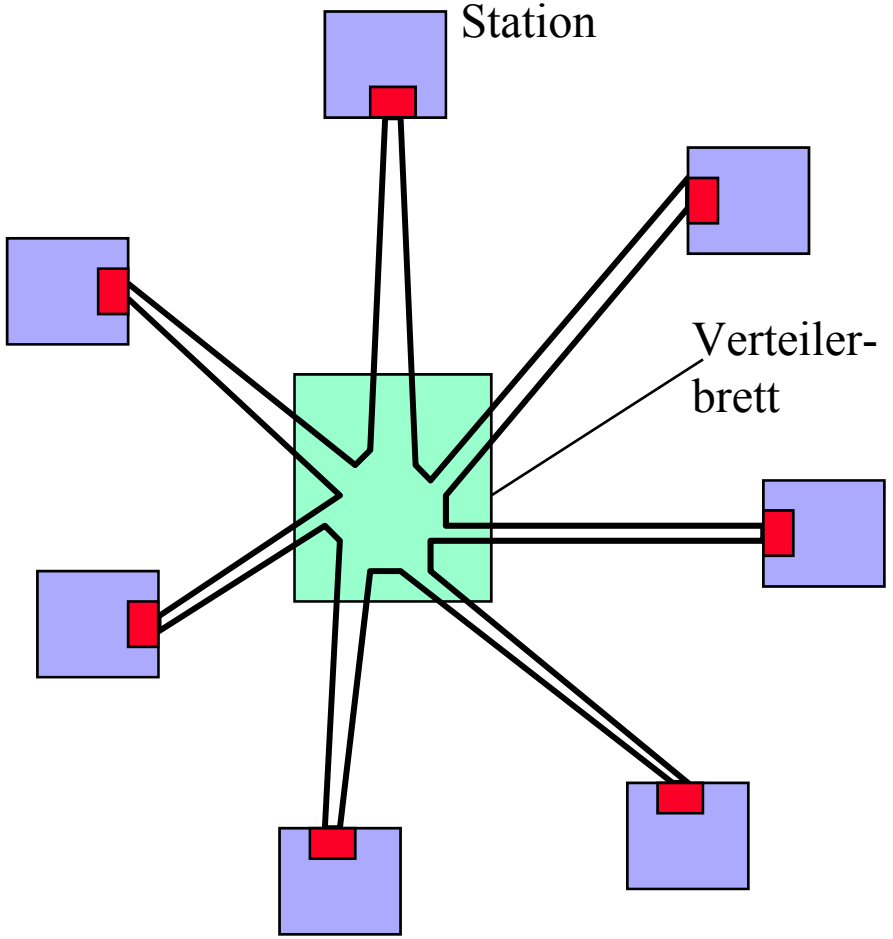
Dezentrale Rechtevergabe: Kursierendes Token

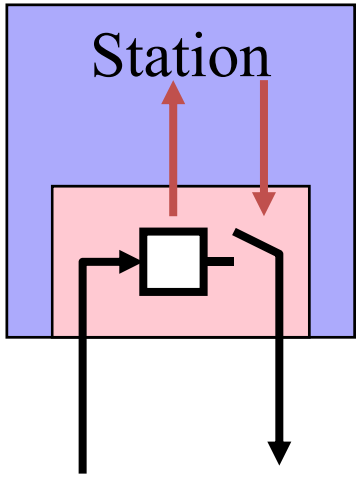
- Ein Kontroll-Token wird reihum von einem zum anderen Knoten weitergereicht
  - Token – Nachricht kreist
  - Wer das Token hat und senden will, sendet. Danach gibt er das Token weiter.
  - Wer das Token hat und nicht senden will, gibt das Token gleich weiter.
  - Fairness:
    Zyklische Weitergabe
    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“

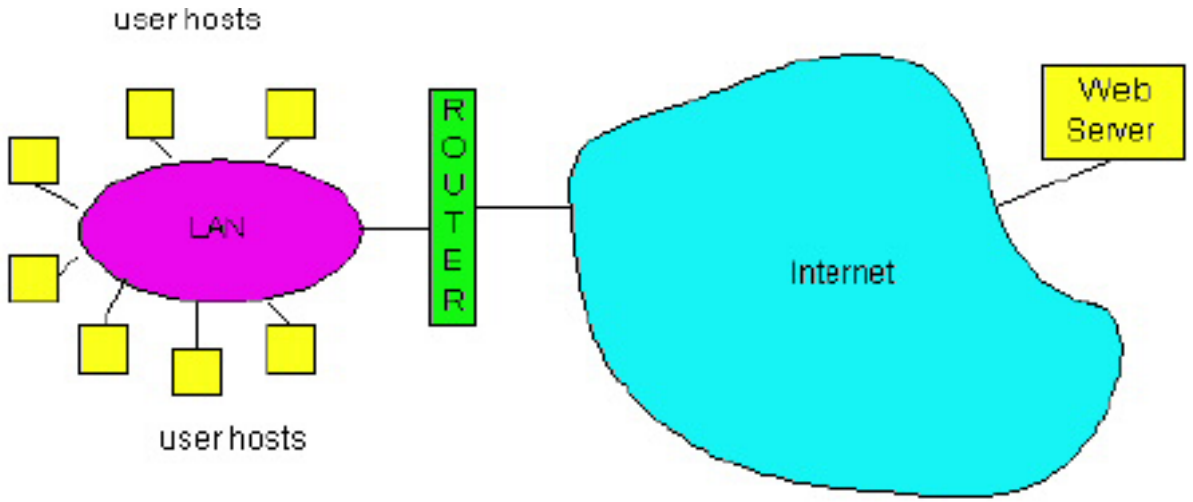
- Ring in Stern-Gestalt
  - Aus 2-Punkt-Simplex-Leitungen,
  - die aber wegen der besonderen Kopplung ein gemeinsames Medium bilden
- Kopplung: 1-Bit-Verzögerung





LAN-Technologien

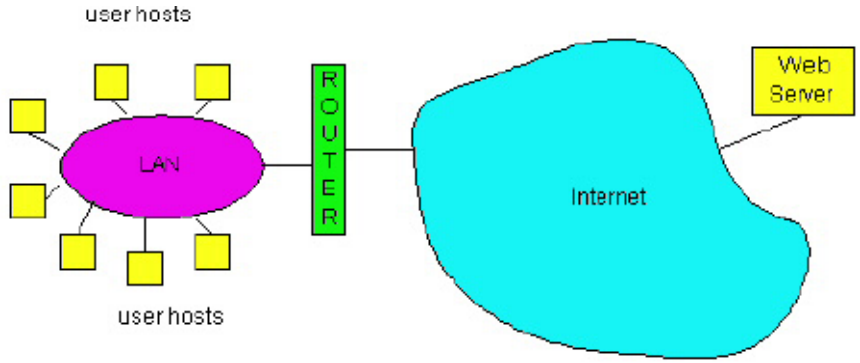
- In LANs genutzte MAC Protokolle:
  - **Token Rings** IEEE 802.5 (IBM token ring), bis zu 16Mbps; Frame, läuft einmal durch den Ring und wird vom Sender entfernt
  - **FDDI** (Fiber Distributed Data Interface), für mittlere Größen (campus, metropolitan), bis zu 200 Station und 100Mbps Übertragungsrate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



© Kurose/Ross 2009

LAN Technologien

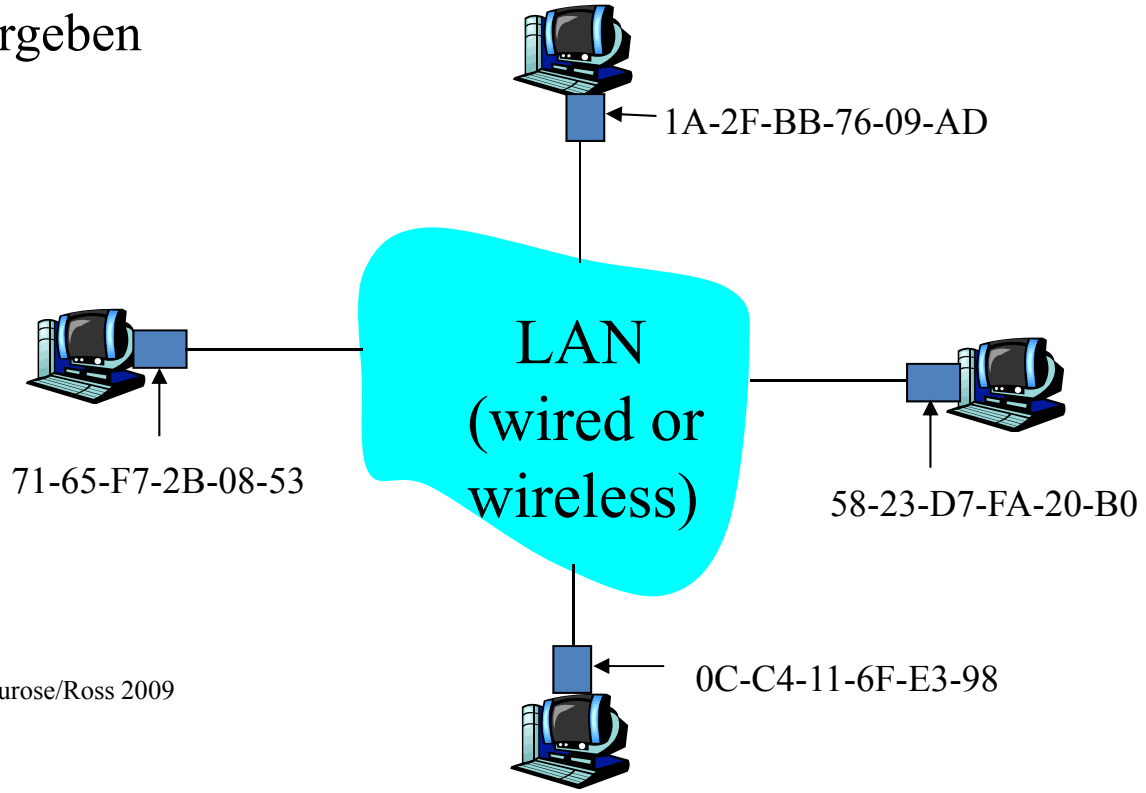
- In diesem Kapitel bisher: Link Layer
  - Fehlererkennung und Korrektur
  - Zugriffskontrolle
- Weiterhin: Lokale Netze
  (nach ISO 8802 / IEEE 802 – Standards)
  - Adressierung: MAC-Adressen
  - Ethernet
  - Hubs, Brücken / Bridges, Switches
  - WLAN: 802.11
- Point-to-Point-Protocol:
  PPP



Adressierung im LAN-Segment

Eigene Schicht 2 Adressen zur 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

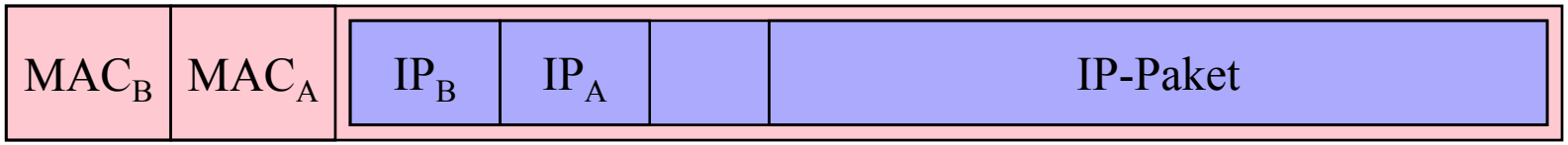


Broadcast address =
FF-FF-FF-FF-FF-FF

= adapter

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 = $MAC_{ForwardingRouter}$

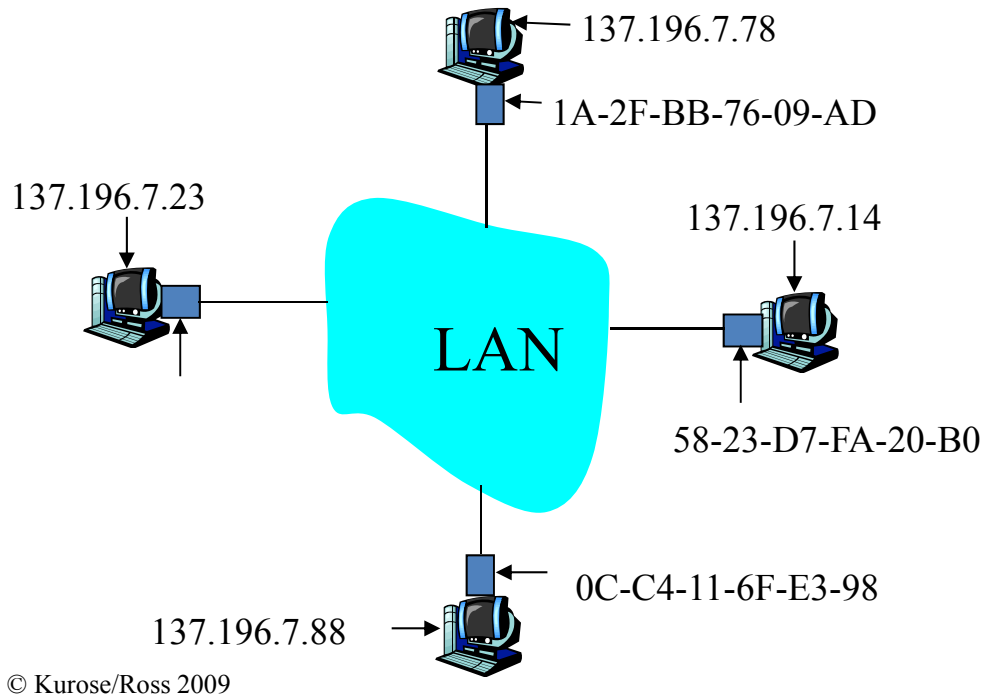
- In beiden Fällen wird eine Adressauflösung benötigt:

  $IP_X \rightarrow MAC_X$

  - Aufgabe übernimmt Address Resolution Protocol (ARP) und ARP-Cache

ARP: Address Resolution Protocol

Wie kann die MAC-Adresse eines Knotens aus der IP-Adresse abgeleitet werden?



- Jeder Knoten (Host, Router) in einem LAN hat ARP-Cache/Tabelle
- ARP-Cache: IP/MAC Adressabbildung für einige Knoten

< IP address; MAC address; TTL>

  - TTL (Time To Live): Zeit, der der Eintrag gelöscht wird (typischweise 20 min)

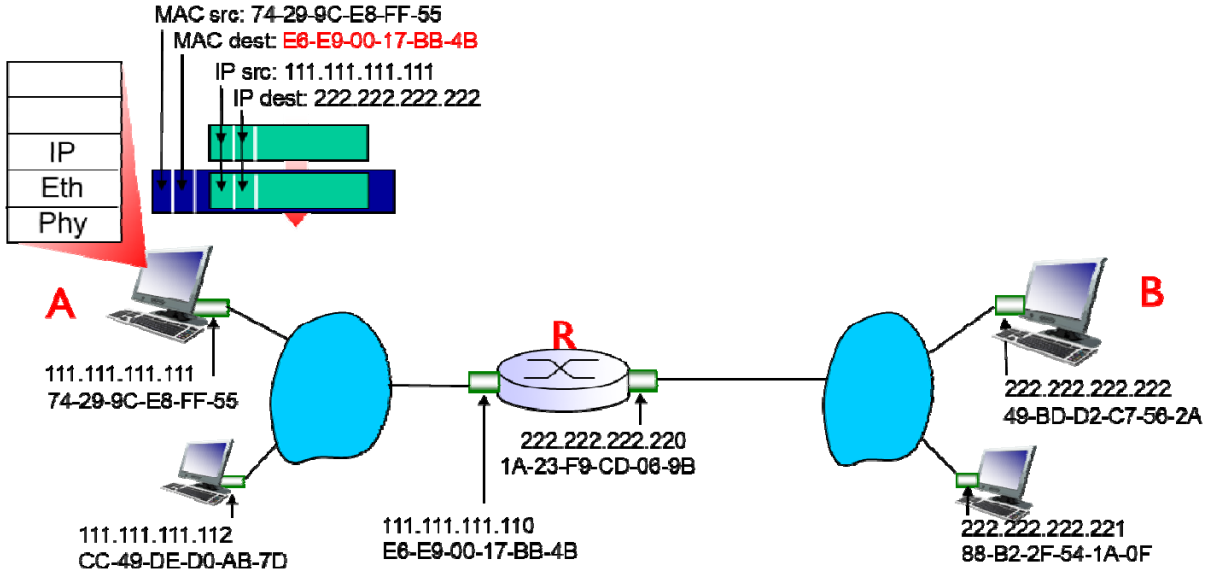
Adressierung im LAN: ARP-Protokoll

A soll ein IP-Paket an B senden,
kennt aber B‘s MAC-Adresse nicht

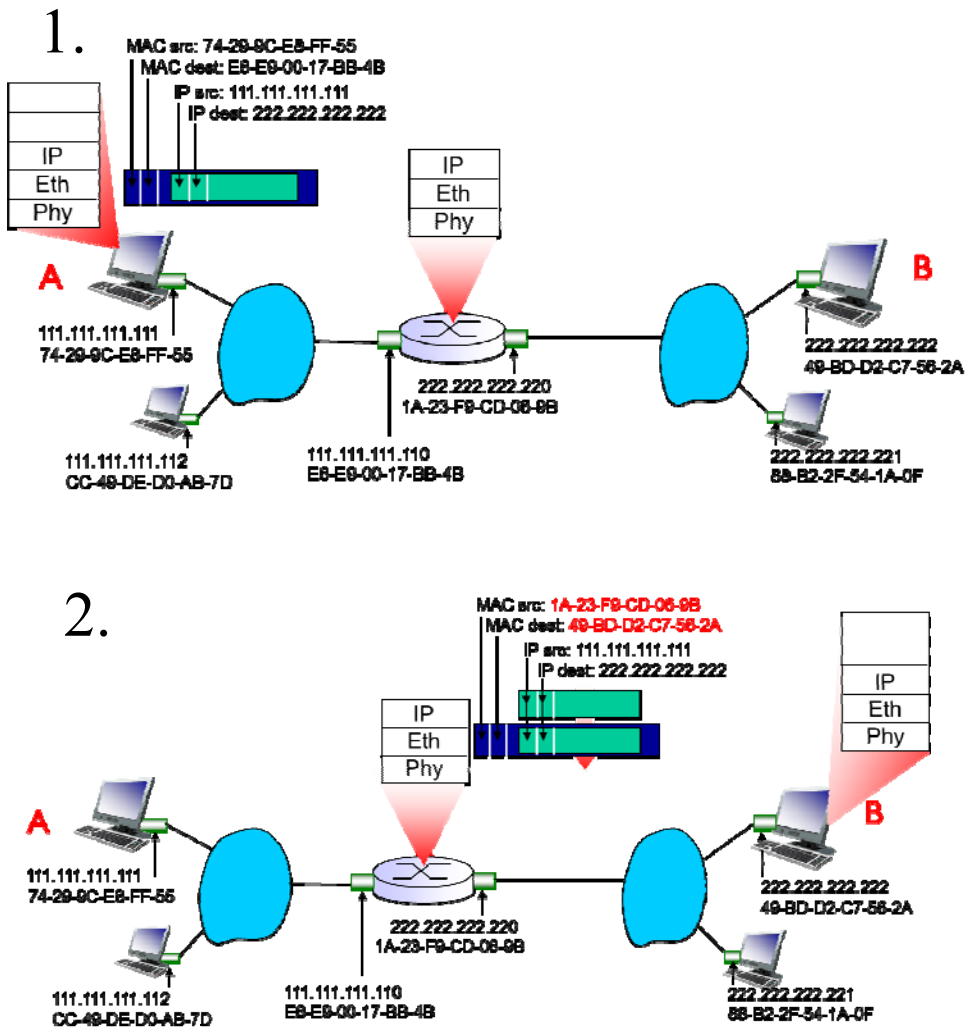
- A sendet einen LAN-Segment-Broadcast Frame:
  ARP-Query für IP_B
- Alle Stationen an diesem LAN-Segment empfangen dies, also auch B
- B sendet ARP-Reply an A: IP_B hat MAC_B
- A kann nun das Paket in einen MAC-Frame packen und an MAC_B senden
- A merkt sich die Zuordnung in einem Zwischenspeicher (ARP-Cache)
  - Automatische Konfiguration per Plug & Play
  - *Soft state* zur Vermeidung zu großer Caches

ARP-Beispiel

- A generiert ein IP-Datagram mit Quelle A, Ziel B
- A nutzt ARP, um R's MAC-Adresse zu erhalten 111.111.111.110
- A generiert ein Paket mit R's MAC-Adresse, dieses Paket enthält das IP-Datagramm
- A's Netzwerkkarte sendet das Paket
- R's Netzwerkkarte empfängt das Paket
- R entfernt das IP-Datagram aus dem Paket, interpretiert die Zieladresse
- R nutzt ARP um B's MAC-Adresse zu erhalten,
- R generiert ein Paket mit B's MAC-Adresse usw.

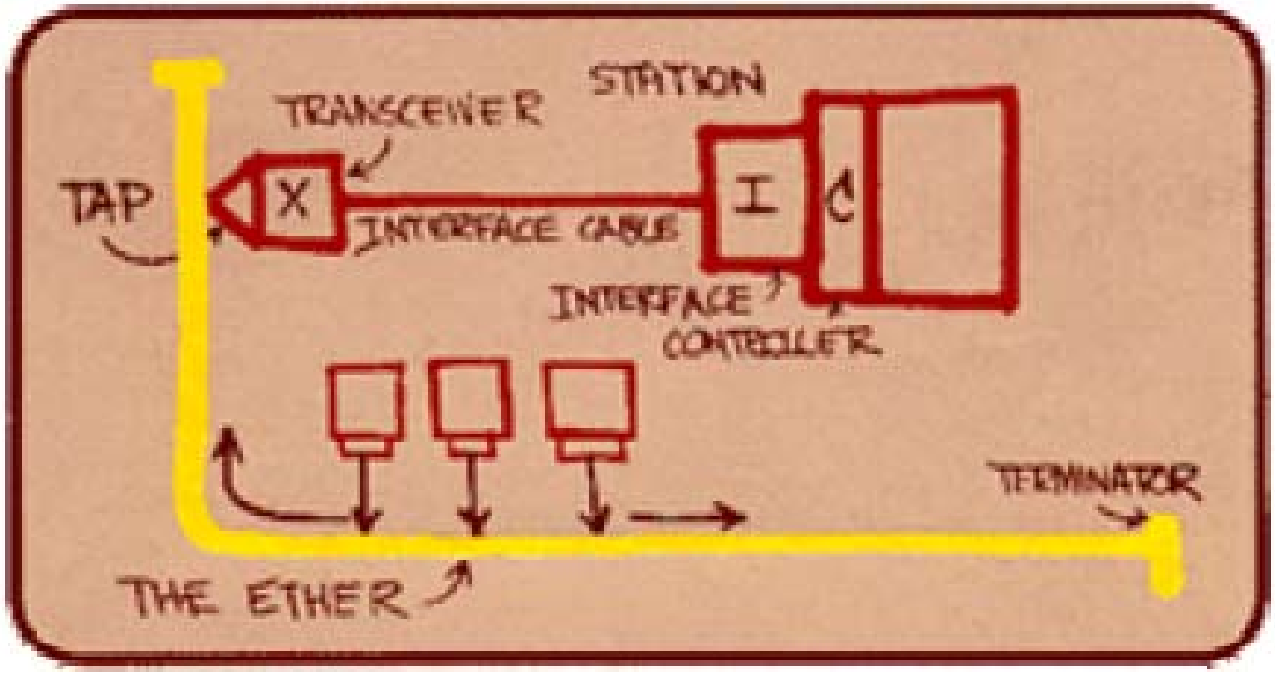


ARP-Beispiel



Ethernet: Dominierende LAN-Technologie

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

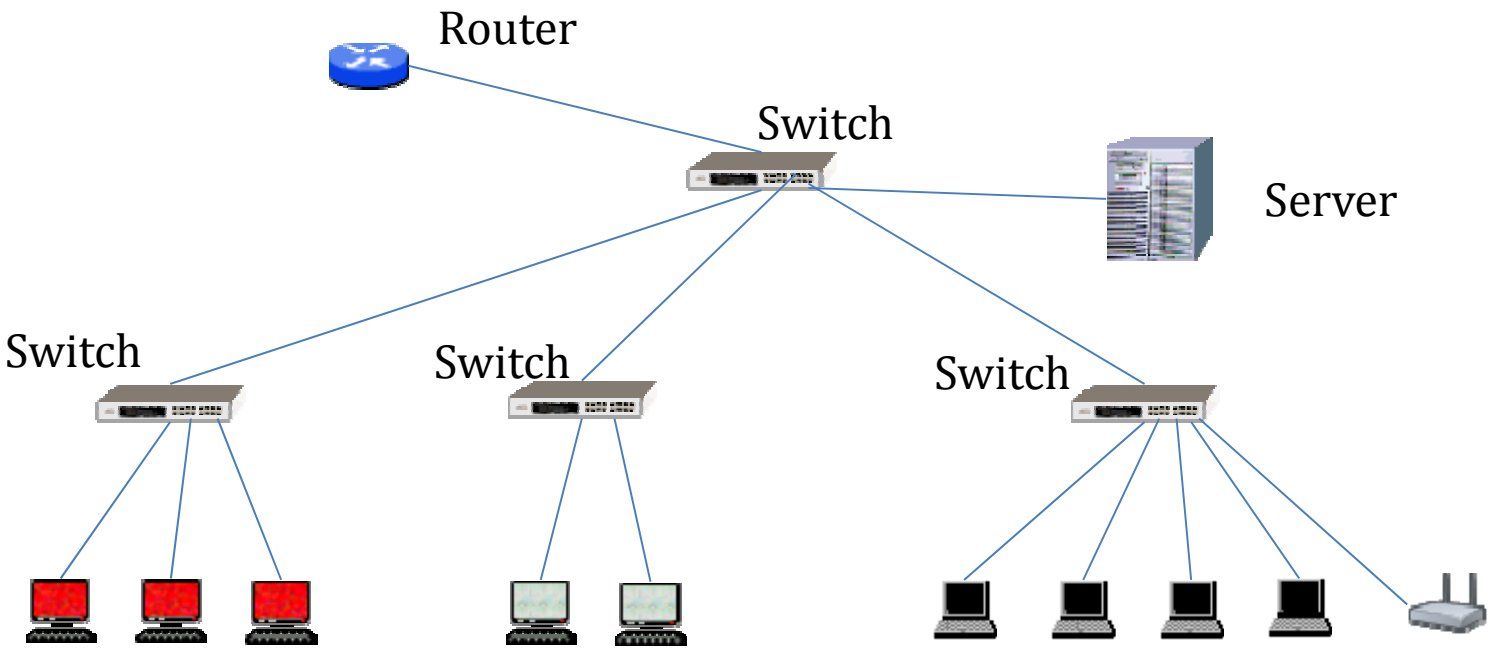


Original Ethernet sketch

Ethernet Technologie

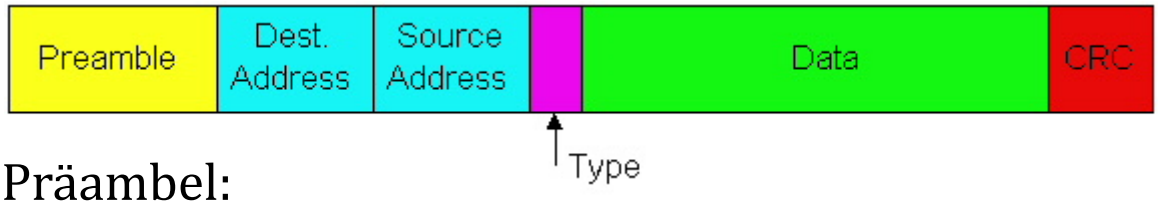
Heutige Ethernets geschaltet (switched)

- (fast) keine Konkurrenz und keine Kollisionen
- hoher Verkabelungsaufwand



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 Empfängern sich zu synchronisieren
- 6 Byte lange Adressen: MAC-Adressen
- Verbindungslose, unzuverlässige und unbestätigte Kommunikation zwischen Partnern
- Bei Fehlererkennung (durch CRC-Prüfung) wird der Rahmen gelöscht
- Mit Hilfe von HDLC (IEEE 802.2) auch verbindungsorientierte zuverlässige Übertragung auf Schicht 2 realisierbar

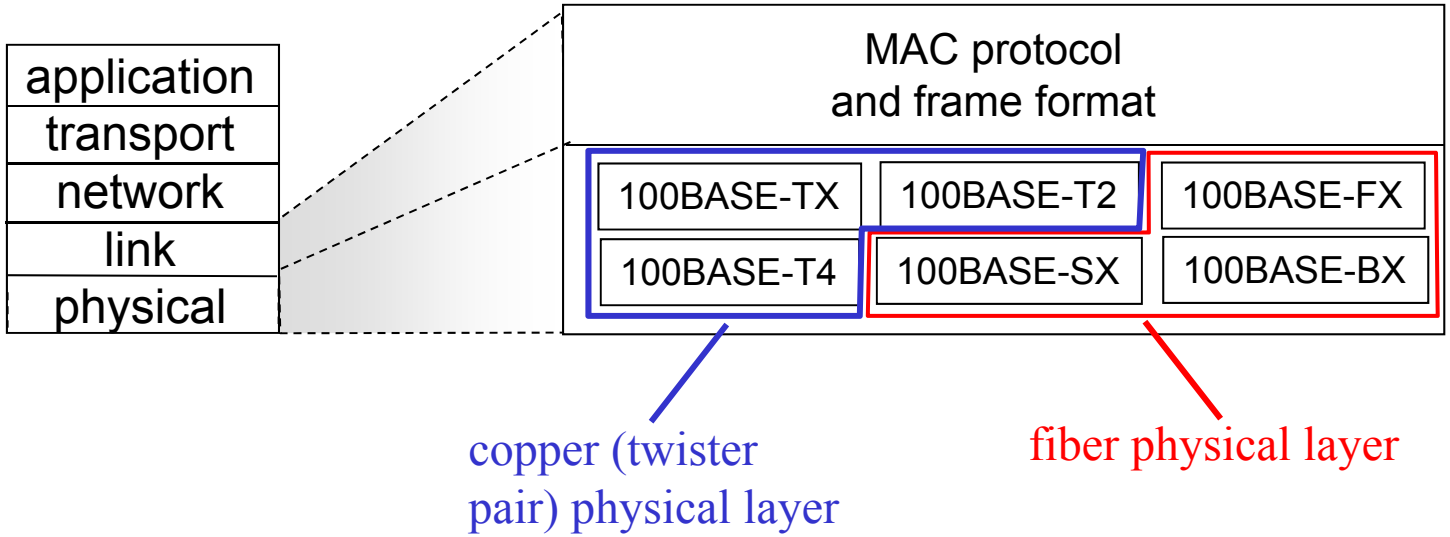
Ethernet: Zugangskontrolle per CSMA/CD

- **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
  - Erste Kollision: Wähle k zufällig aus {0,1}; Verzögerung entspricht k x 512 bit Übertragungszeit
  - Zweite Kollision: Wähle k aus {0,1,2,3}
  - Nach m Kollisionen, wähle k aus $\{0,1,2,\ldots,2^m-1\}$
  - Nach 10 oder mehr Kollisionen, wähle k aus {0,1,2,3,4,…,1023}

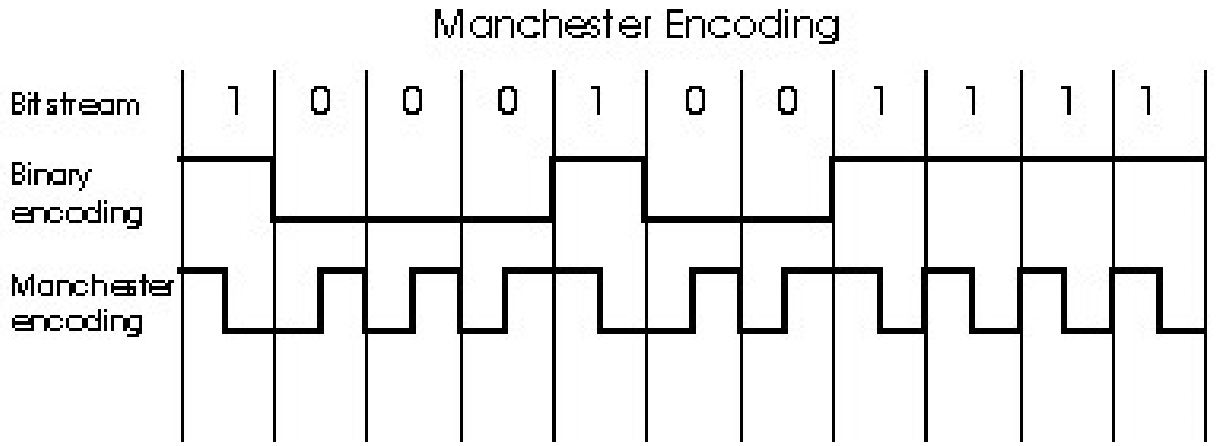
Im Mittel wächst k exponentiell mit der Segmentbelastung

802.3 Ethernet Standards: Link & Physical Layers

- *viele* unterschiedliche Ethernet-Standards
  - einheitliches MAC-Protokoll und Frame-Format
  - Unterschiedliche Geschwindigkeiten: 2 Mbps, 10 Mbps, 100 Mbps, 1Gbps, 10Gbps
  - Unterschiedliche physikalische Medien: Glasfaser, Koaxialkabel



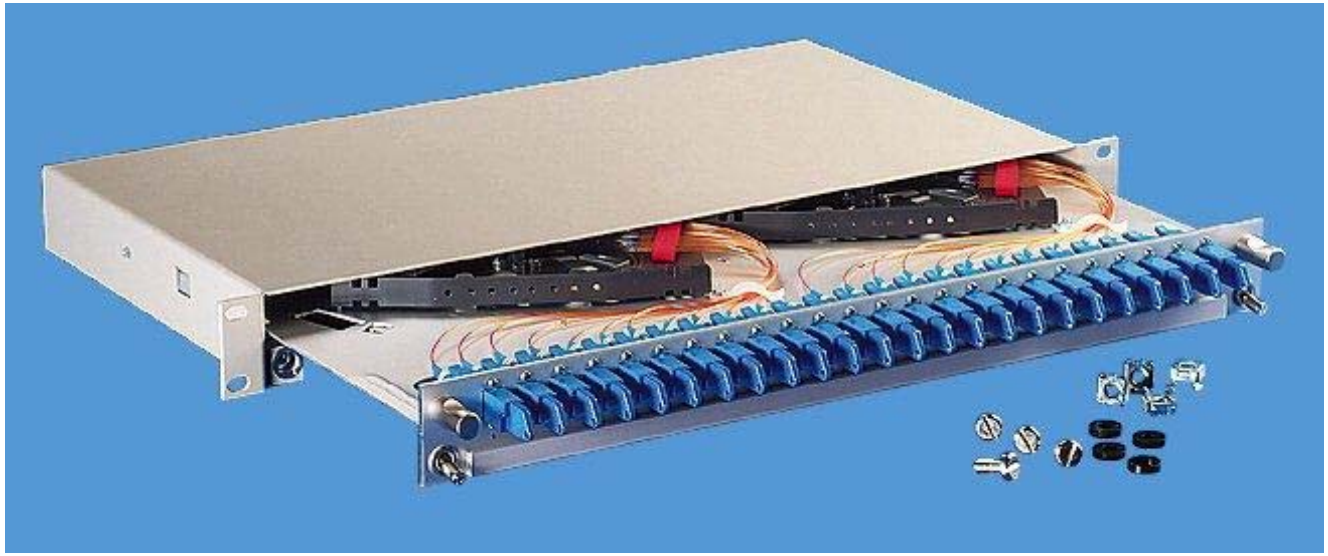
10BAse2, 10BaseT: Manchester Codierung



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

Ethernet: Weiteres

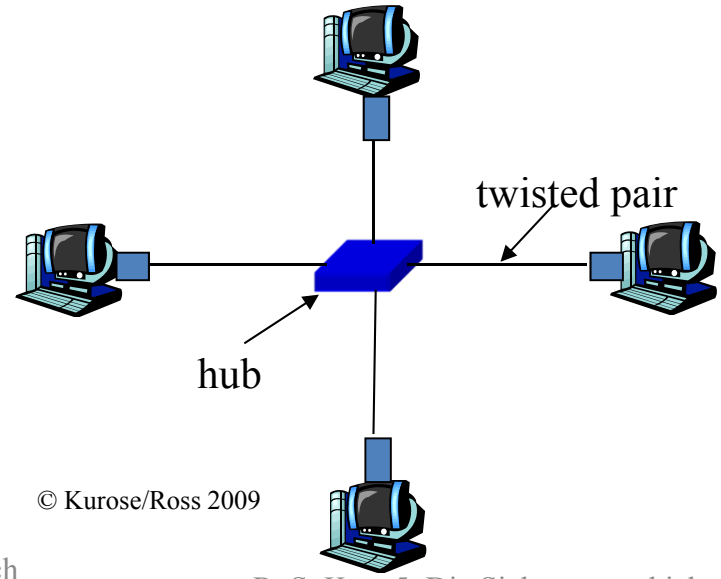
- Gbit Ethernet (1000MBit/sec)
  Standard Ethernet-Frames
  - sowohl CSMA/CD Busse (müssen sehr kurz sein)
  - als auch Zweipunkt-Leitungen
- 10 GBit/sec schon verfügbar



Hubs - Repeater

... physical-layer ("dumme") Repeater oder Hubs:

- Bits kommen auf einer Leitung an und werden an alle anderen Leitungen mit der selben Rate weitergeleitet
- Kollisionen können zwischen allen angeschlossenen Stationen auftreten
- Frames werden nicht gepuffert
- kein CSMA/CD am Hub: NICs der Hosts entdecken Kollisionen



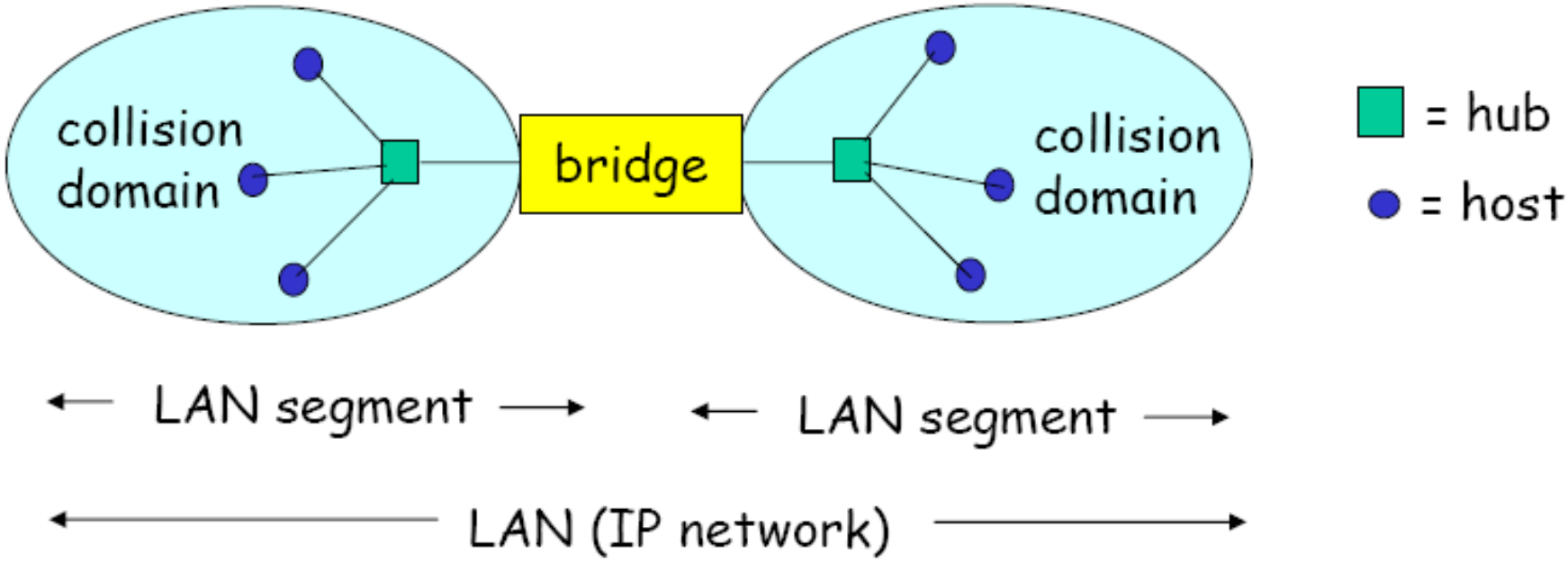
- Repeater verbinden 2 Segmente
- Hubs verbinden sternförmig

Brücken

- Geräte auf der Sicherungsschicht: Intelligenter als Repeater, übernehmen eine aktive Rolle
  - Speichern und weiterleiten von Ethernet Frames
  - Untersuchung der MAC-Adressen hereinkommender Frames und selektive Weiterleitung zu einem oder mehreren Segmenten, Nutzung von CSMA/CD zum Segmentzugriff
- *transparent*
  - Brücken für angeschlossene Hosts nicht sichtbar
- *plug-and-play, self-learning*
  - Heutige Brücken konfigurieren sich selbst

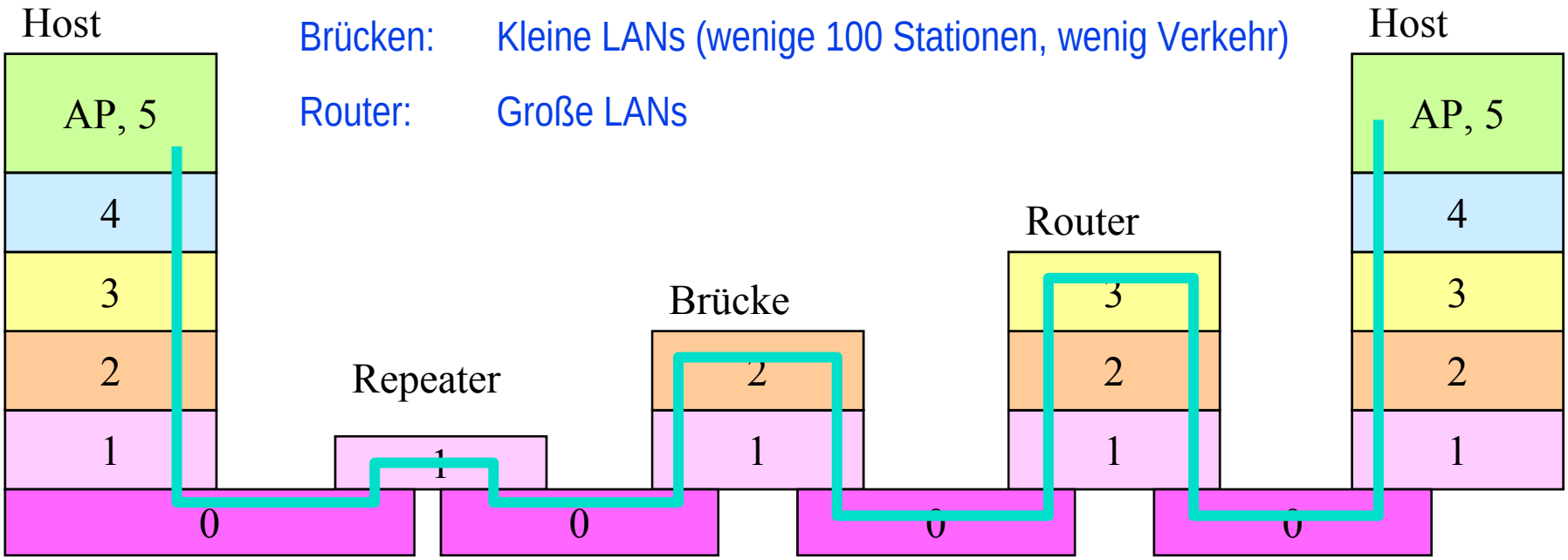
Brücken: Verkehrsisolation

- 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)



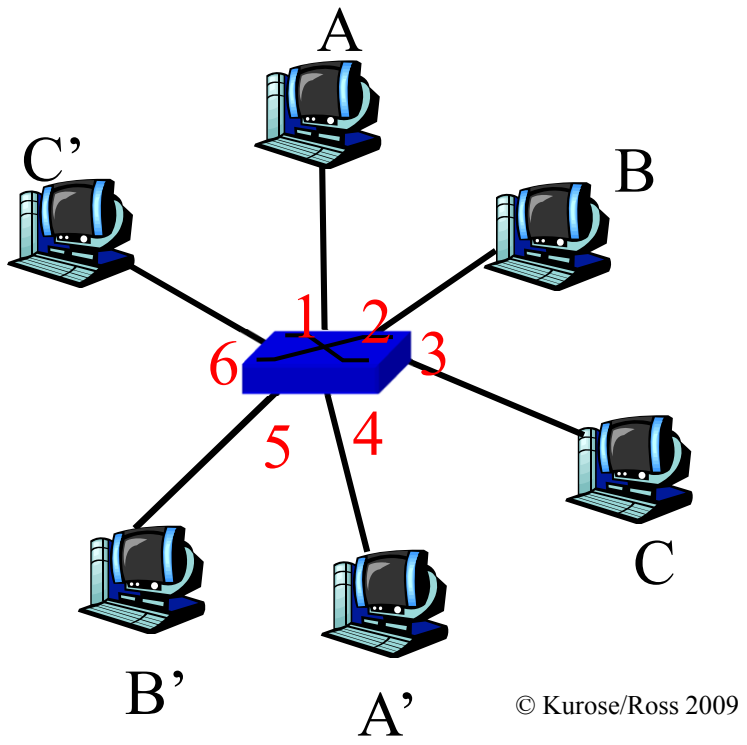
Brücken versus Router

- 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)



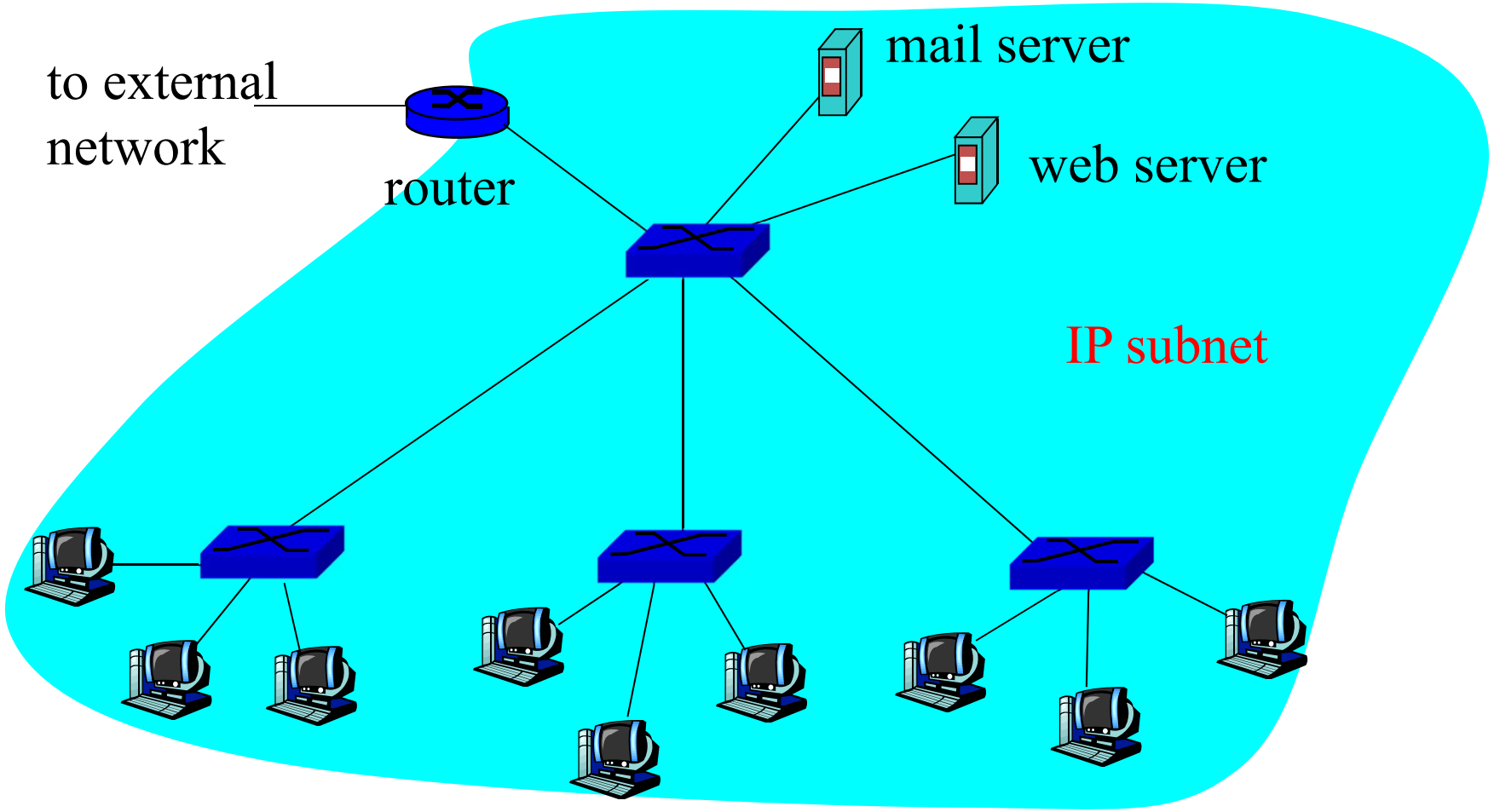
Ethernet - Switches

- 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
  - ...



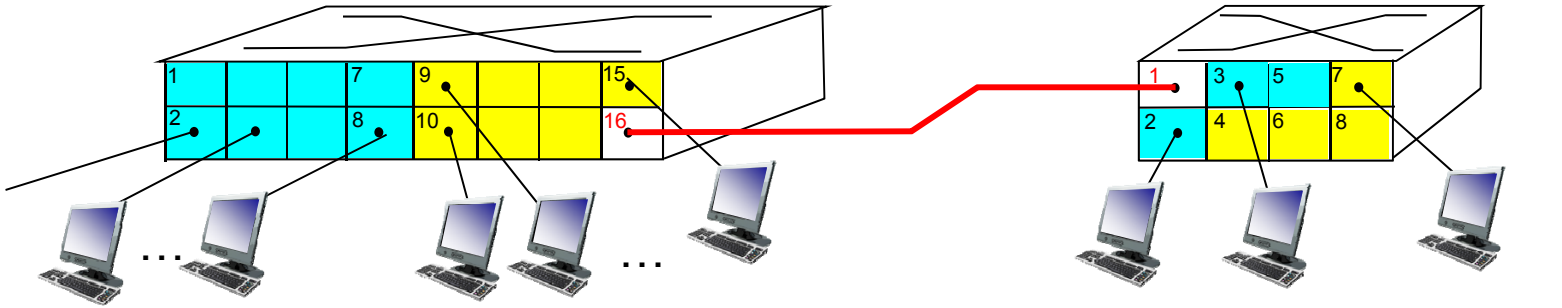
Switch mit 6 Anschlüssen (1,2,3,4,5,6)

Institutsnetz



VLANs Virtuelle LANs

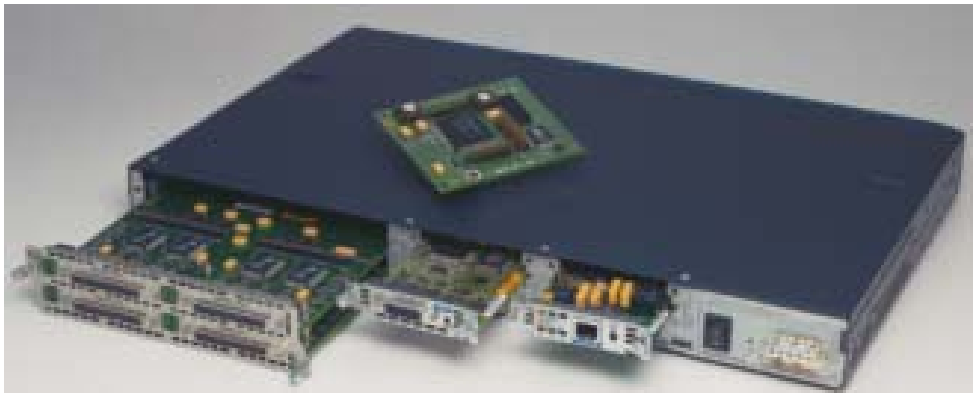
Aufbau mehrerer (virtueller) LANs in einem physikalischen LAN



- Zuordnung der Geräte zu einem LAN über die Zuordnung der Ports
- Kennung des VLANs im Frame-Header
- Zugehörige Protokolle 802.1q

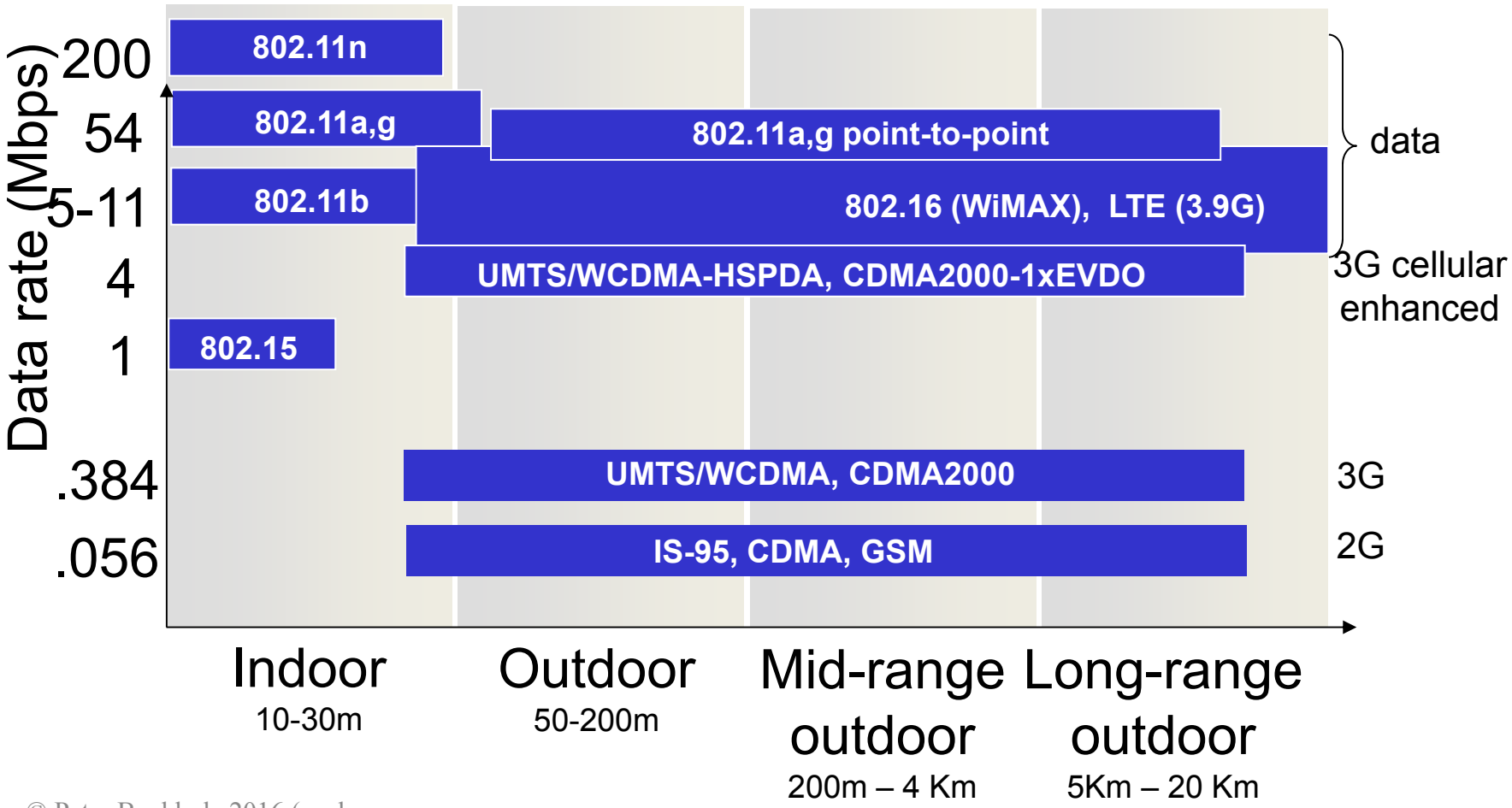
Netzsegment-Kopplung: Vergleich

	Hub/Repeater	Brücke	Router	Switch
Verkehrsisolation	Nein	ja	Ja	Ja
Plug & play	Ja	Ja	Nein	Ja
Optimales Routing	Nein	Nein	Ja	Nein
Durchschaltung (cut through)	Ja	Nein	Nein	Ja



Drahtlose Netze

Einige Standards

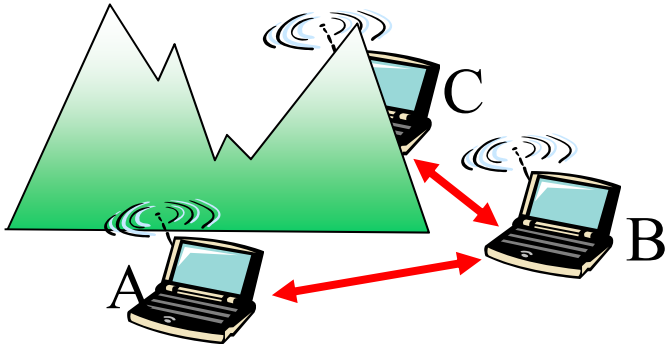


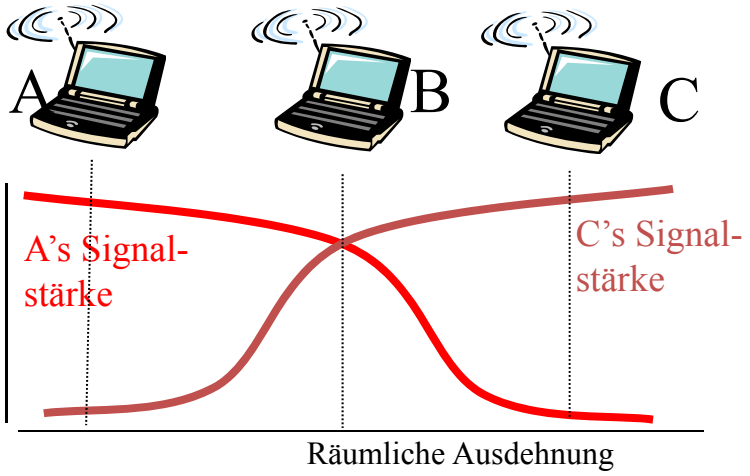
IEEE 802.11: Wirelesss LAN

Standard	Beschreibung
IEEE 802.11	WLAN bei Datenraten bis zu 2 Mbit/s im 2,4-GHz ISM (Industrial, Scientific and Medical) Band
IEEE 802.11a	WLAN bei Datenraten bis zu 54 Mbit/s im 5-GHz Unlicensed National Information Infrastructure (UNII) Band
IEEE 802.11b	Erweiterung von 802.11 bei Datenraten bis zu 11 Mbit/s im 2,4-GHz ISM (Industrial, Scientific and Medical) Band
IEEE 802.11e	MAC-Erweiterung zu 802.11a und b, um QoS und verbessertes Power Management zu ermöglichen
IEEE 802.11f	Roaming mit APs verschiedener Hersteller
IEEE 802.11g	Erweiterung für hohe Datenraten 10-20 Mbit/s im 2,4 GHz Band
IEEE 802.11h	Dynamische Frequenzselektion für 802.11a (wg. EU-Regulierung)
IEEE 802. 11i	MAC-Erweiterung, um verbesserte Sicherheits- und Authentifikationsmechanismen zu ermöglichen
IEEE 802. 11n	High throughput task group, 540 Mbit/s mit MiMo-Technik

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 aufwändig





Dämpfungsverlauf

IEEE 802.11 MAC Protocol: CSMA

802.11 Sender

1 Falls der Kanal unbelegt für **DIFS,** dann

übertrage die gesamte Nachricht (ohne CD)

2 Falls Kanal belegt, dann

Starte eine zufällige backoff-Zeit

Timer wird runter gezählt solange Kanal unbelegt

Übertrage bei Ablauf des Timers

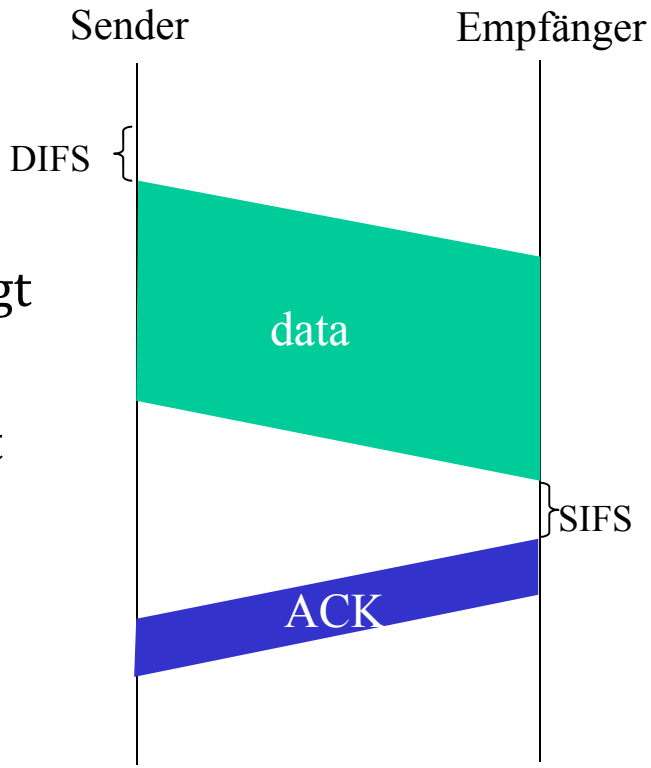
Vergrößere backoff-Intervall, falls ACK ausbleibt

802.11 Empfänger

- Falls Nachricht OK

schicke ACK nach **SIFS**

SIFS < DIFS



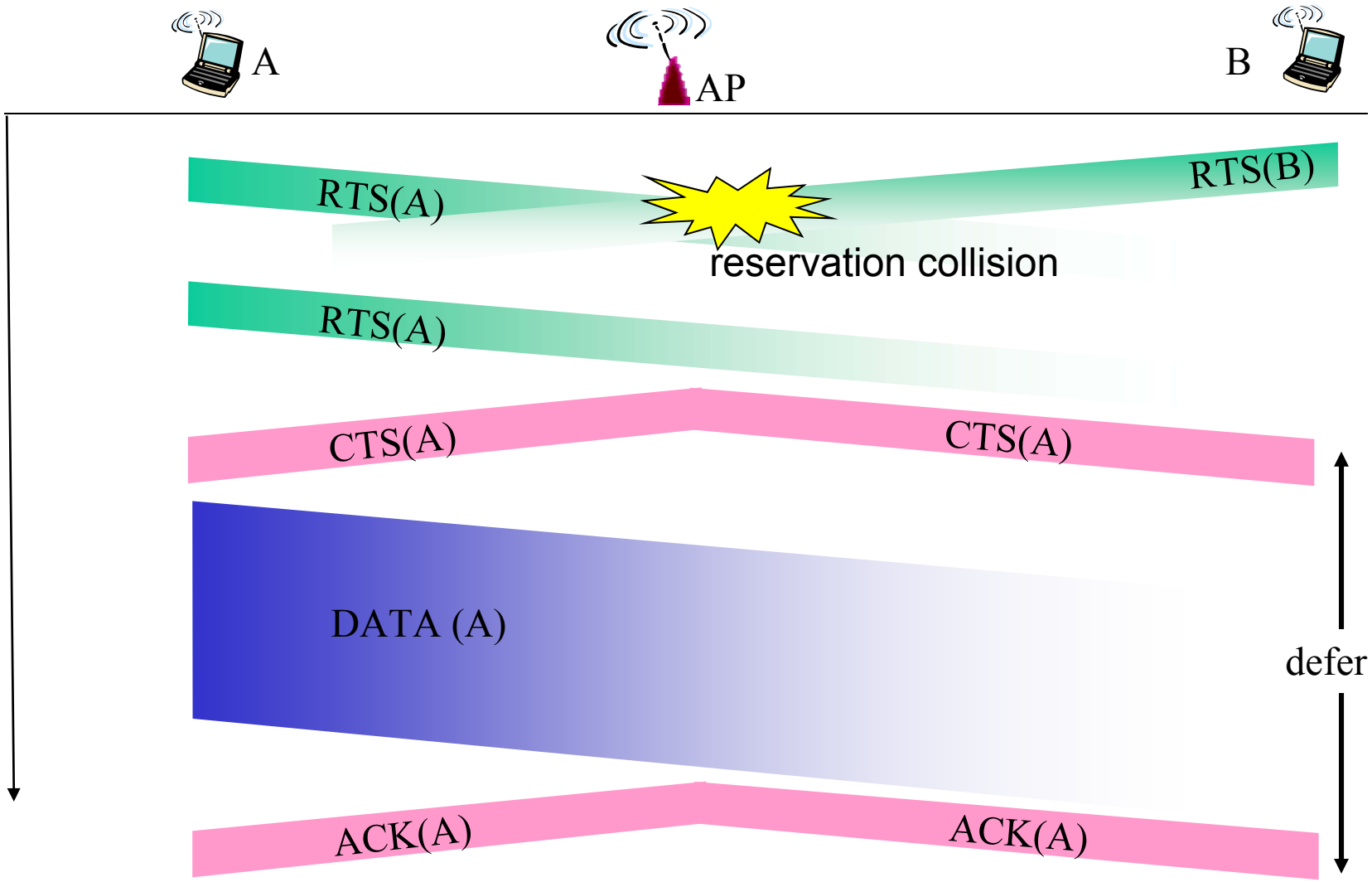
Zugriffsverfahren mit RTS/CTS

Idee: Sender kann den Kanal reservieren und vermeidet den zufälligen Zugriff bei der Übertragung großer Datenpakete

- Sender schickt zuerst ein kleines request-to-send (RTS) Paket unter Nutzung von CSMA
  - RTS Pakete können kollidieren, sind aber sehr kurz
- Empfänger sendet clear-to-send CTS als Antwort auf RTS
- CTS wird von allen Stationen im Sendebereich des Empfängers gehört
  - Sender überträgt sein Datenpaket
  - Andere Station verzögern ihre Übertragungen

Vollständige Vermeidung von Kollisionen der Datenpakete!
Kein hidden terminal-Problem mehr!

Zugriffsverfahren mit RTS/CTS

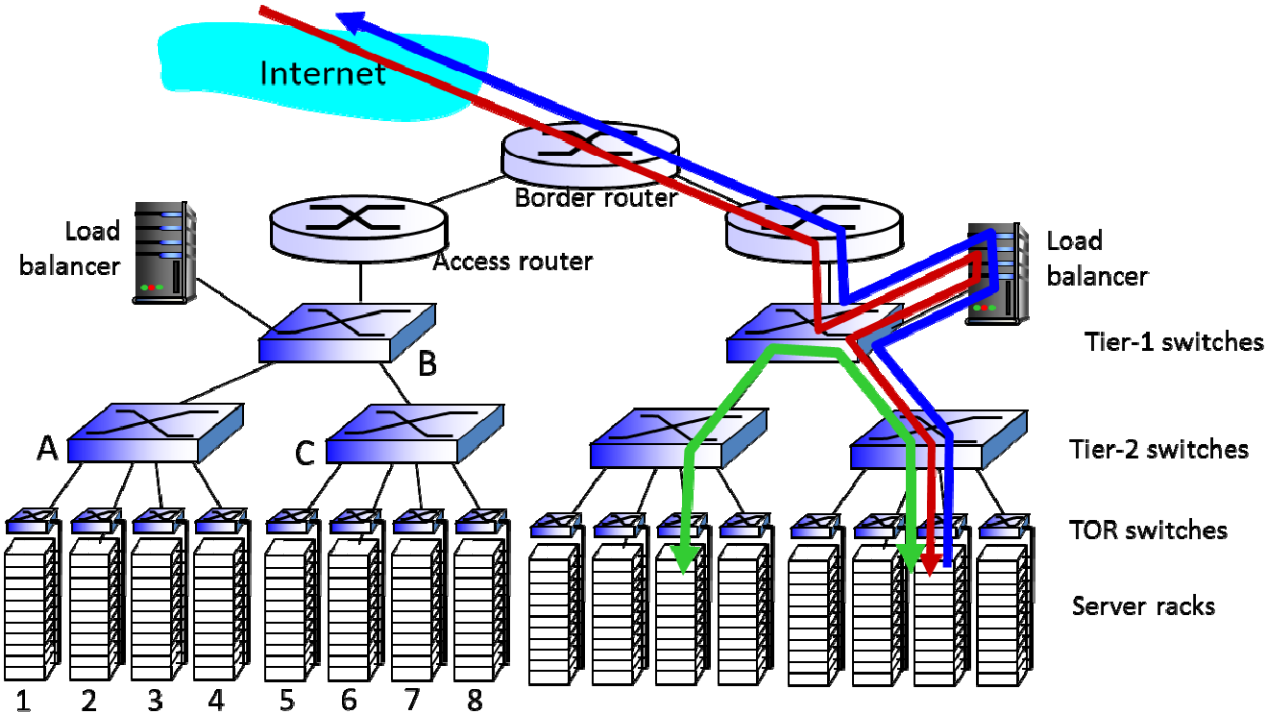


Point-to-Point-Protocol: PPP

- Verbindung zum Internet über Wählleitungen
  transportiert IP-Pakete als 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

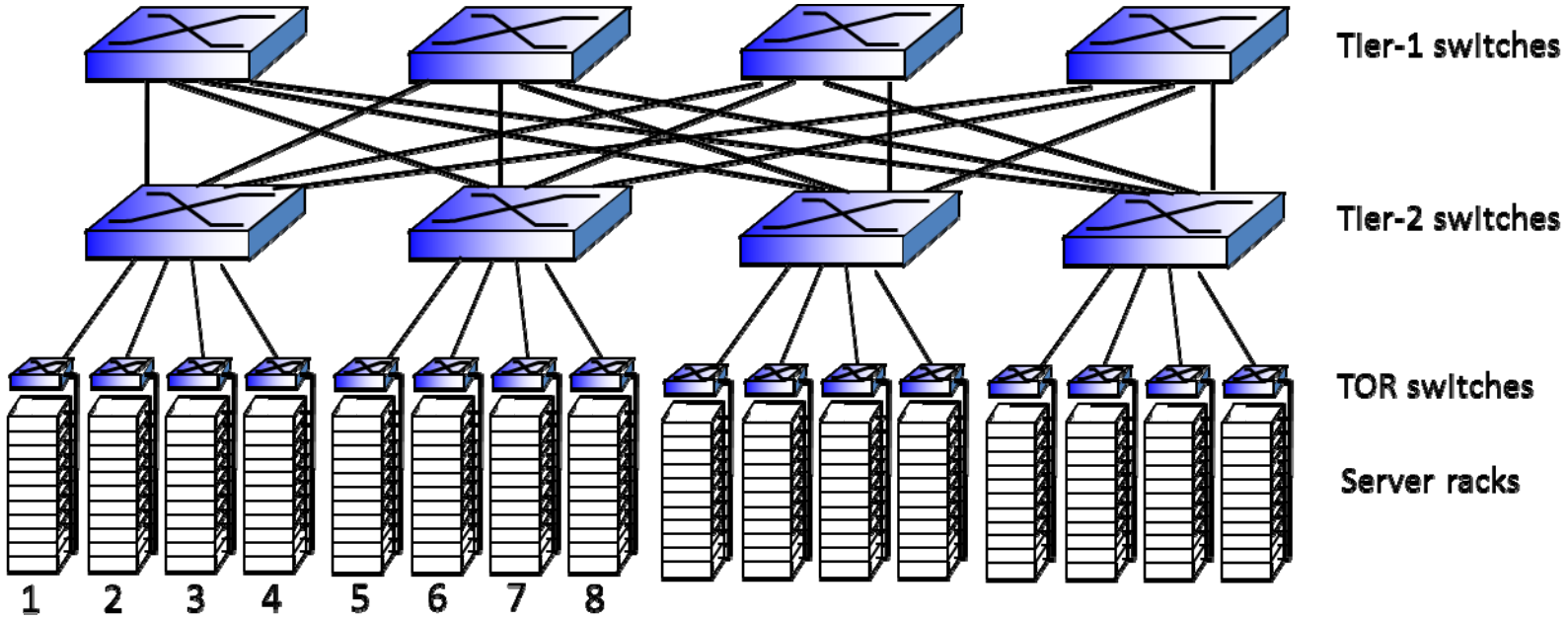
Datencenter

- 1000 bis 10000 Hosts, die eng verbunden arbeiten
  (e-commerce Amazon; content provider YouTube4, Microsoft, Apple; Suchmaschinen Google,…)
- Herausforderungen: Lastbalancierung, Engpässe in Kommunikation und Datenzugriff vermeiden, …



Datencenter

- Redundanz der Hardware und Software ermöglicht hohe Flexibilität durch Nutzung verschiedener Pfade durch das Netz
- Switches müssen entsprechende Ansätze unterstützen



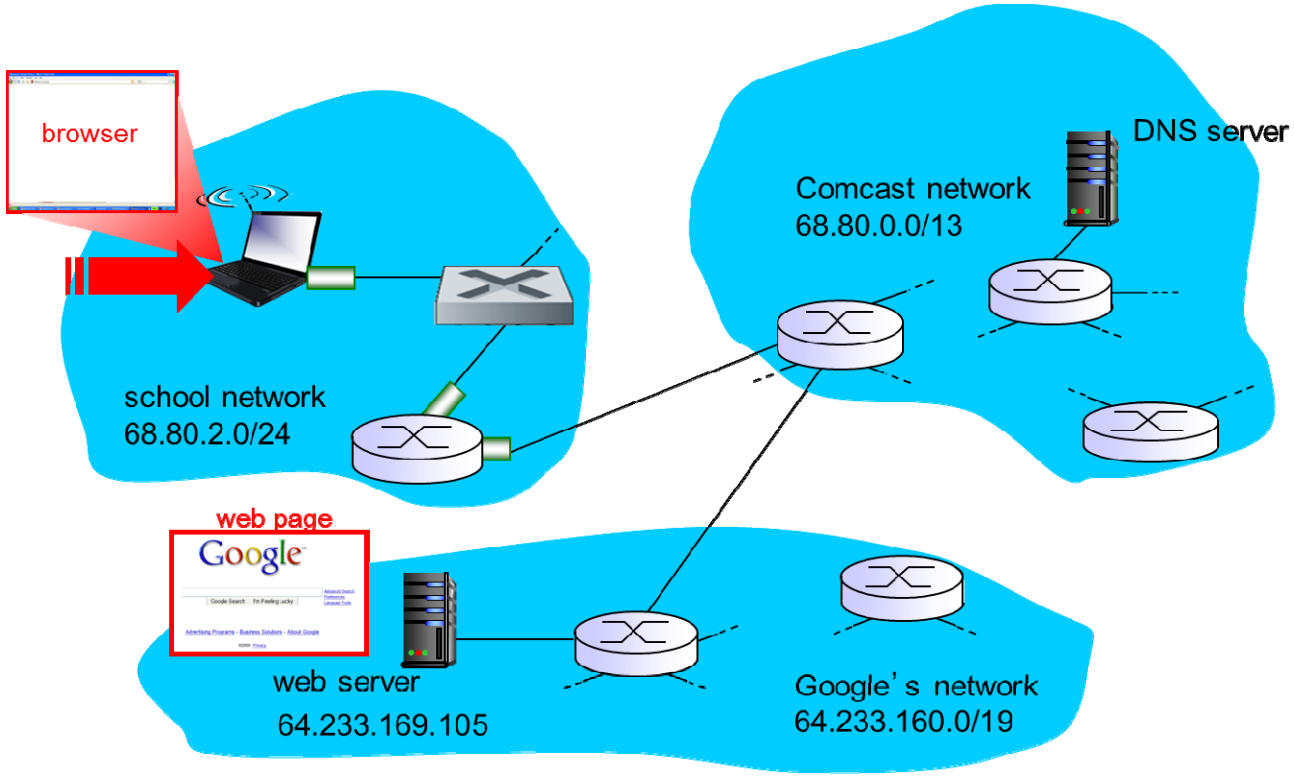
Der Weg durch die Protokollschichten

Alle Ebenen im Protokollstapel wurden vorgestellt:

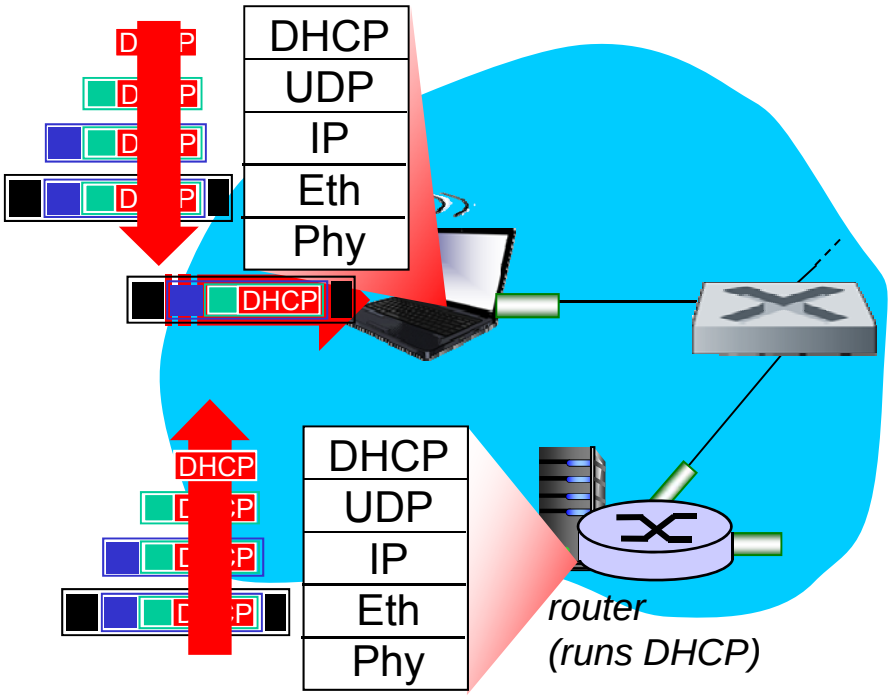
- Eine kurze Zusammenfassung anhand eines kleinen Beispiels

Szenario:

Zugriff von einem Laptop in einem Uni-Netz auf www.google.com

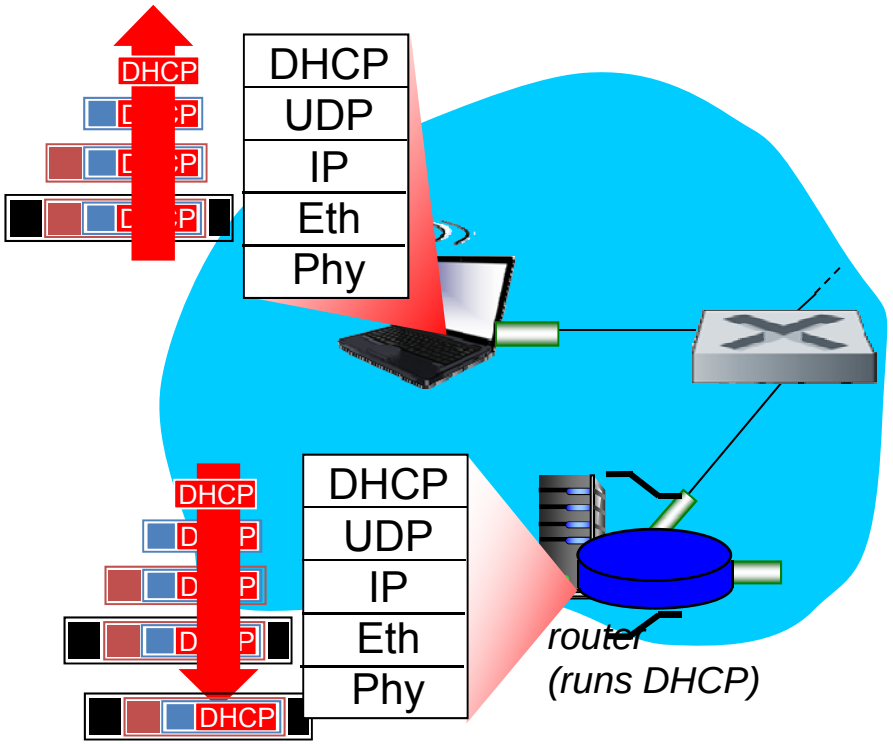


Verbindung zum Internet



- Laptop benötigt eine eigene IP-Adresse, die Adresse des ersten Routers und des DNS-Servers und nutzt dazu *DHCP*
- DHCP Anfrage wird in in *UDP* Paket gekapselt, dies in *IP* und dieses in *802.3* Ethernet gekapselt
- Ethernet sendet *broadcast* (Ziel: FFFFFFFFFFFF) auf dem LAN, dieser wird vom Router auf dem der *DHCP* Server läuft empfangen
- Ethernet, IP und UDP Daten werden ausgepackt und an DHCP weitergeleitet

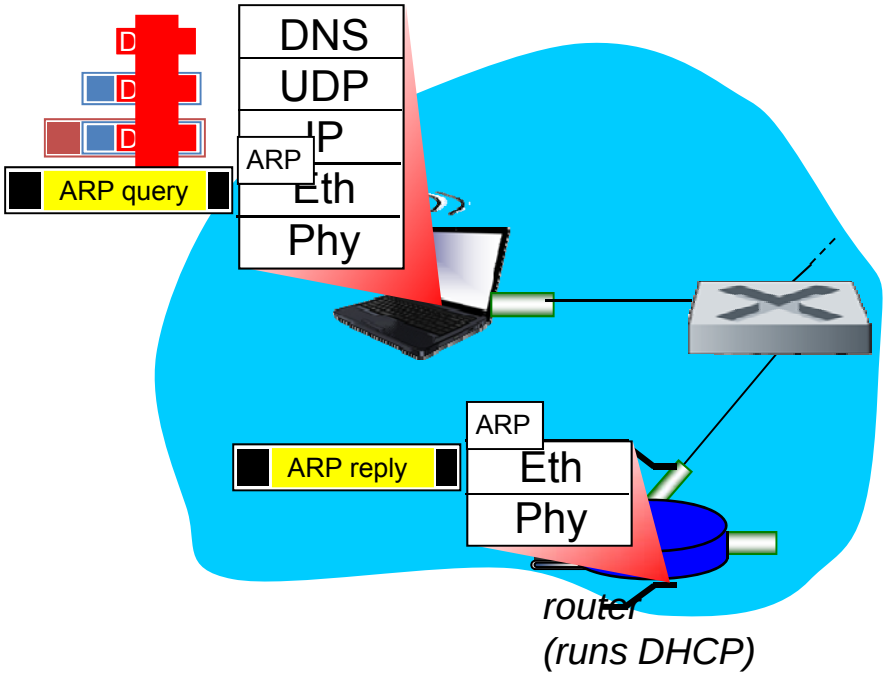
Verbindung zum Internet



- DHCP Server sendet *DHCP ACK* mit IP-Adresse des Clients, IP-Adresse des ersten Router, Name & IP-Adresse des DNS-Servers
- DHCP-Server verpackt Daten in Rahmen, Rahmen wird durch das LAN weitergeleitet (*lernender Switch*), demultiplexing beim Client
- DHCP empfängt DHCP ACK reply

Client hat eine IP Adresse, kennt Namen & Adresse des DNS-Servers, die IP-Adresse des ersten Routers

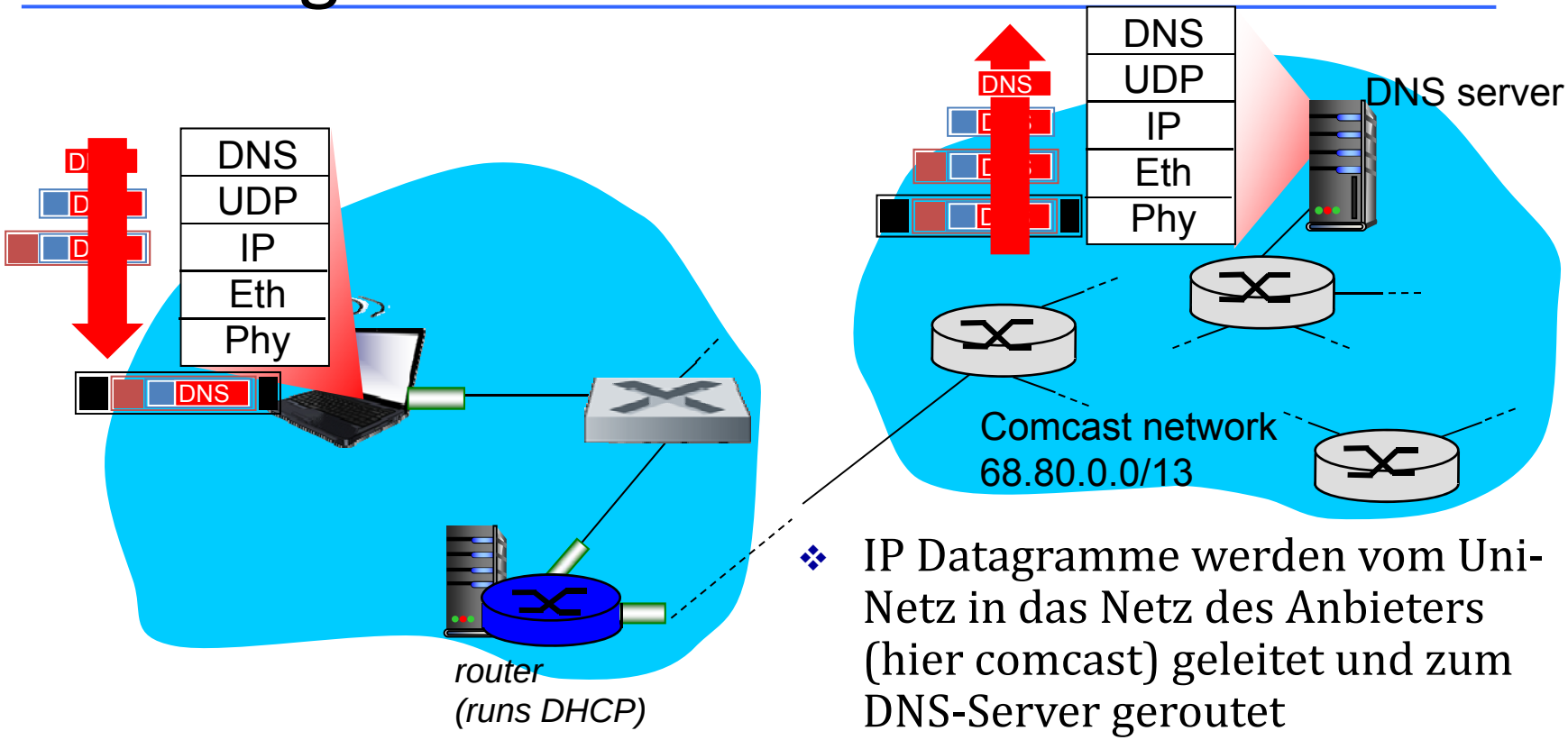
ARP (vor DNS, vor HTTP)



- Vor dem Senden der *HTTP*-Anfrage, wird die IP-Adresse von www.google.com benötigt: *DNS*
- DNS-Anfrage wir generiert und in ein UDP-Segment verpackt, dieses wird in ein IP-Datagramm verpackt, dieses in einen Ethernet-Rahmen. Zur Sendung wird die MAC Adresse des Router-Interfaces benötigt: *ARP*
- *ARP query* Broadcast, wird vom Router empfangen, der mit *ARP reply* inkl. MAC-Adresse antwortet

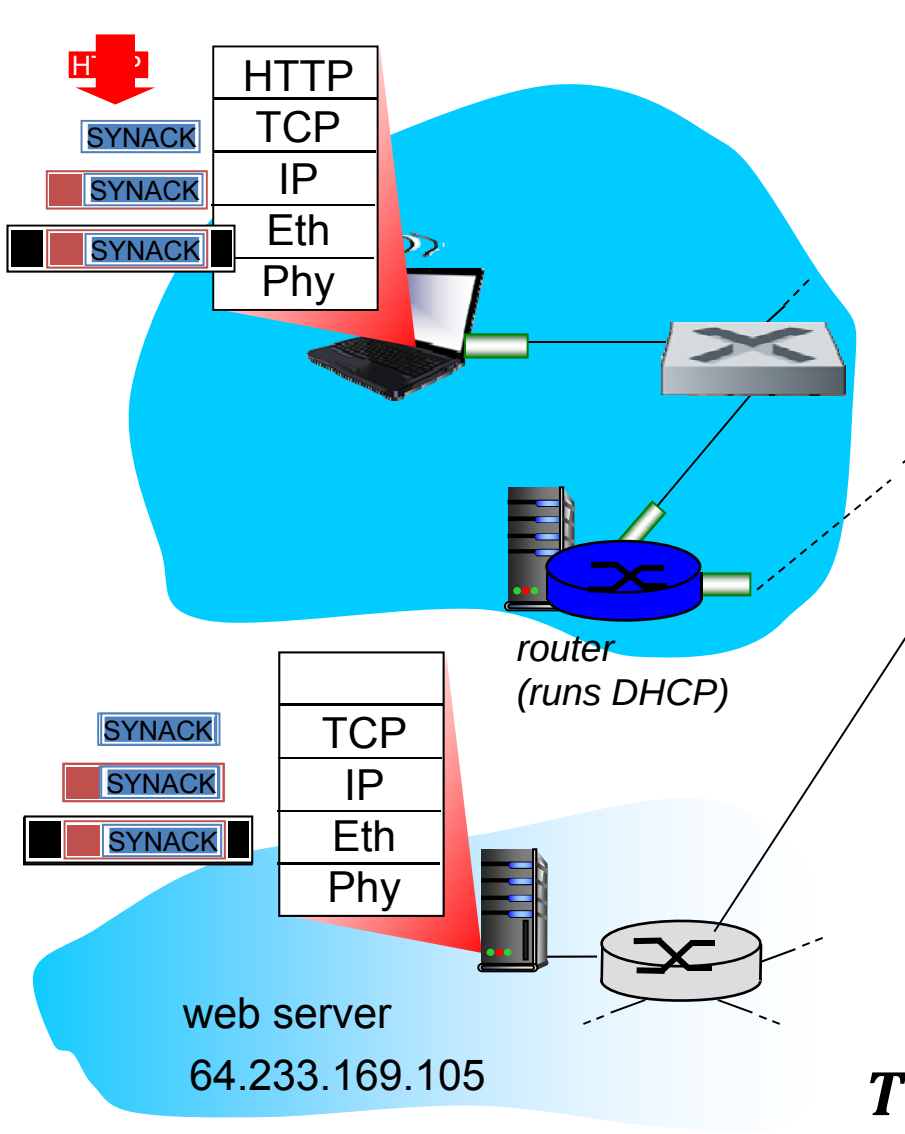
Client kennt nun die MAC-Adresse des ersten Routers und kann die DNS-Anfrage senden

Nutzung von DNS



- IP Datagramme werden vom Uni-Netz in das Netz des Anbieters (hier comcast) geleitet und zum DNS-Server geroutet (Routingtabellen wurden mit *RIP, OSPF, IS-IS* und/oder *BGP* generiert)
- DNS -Server antwortet dem Client mit der IP-Adresse von www.google.com

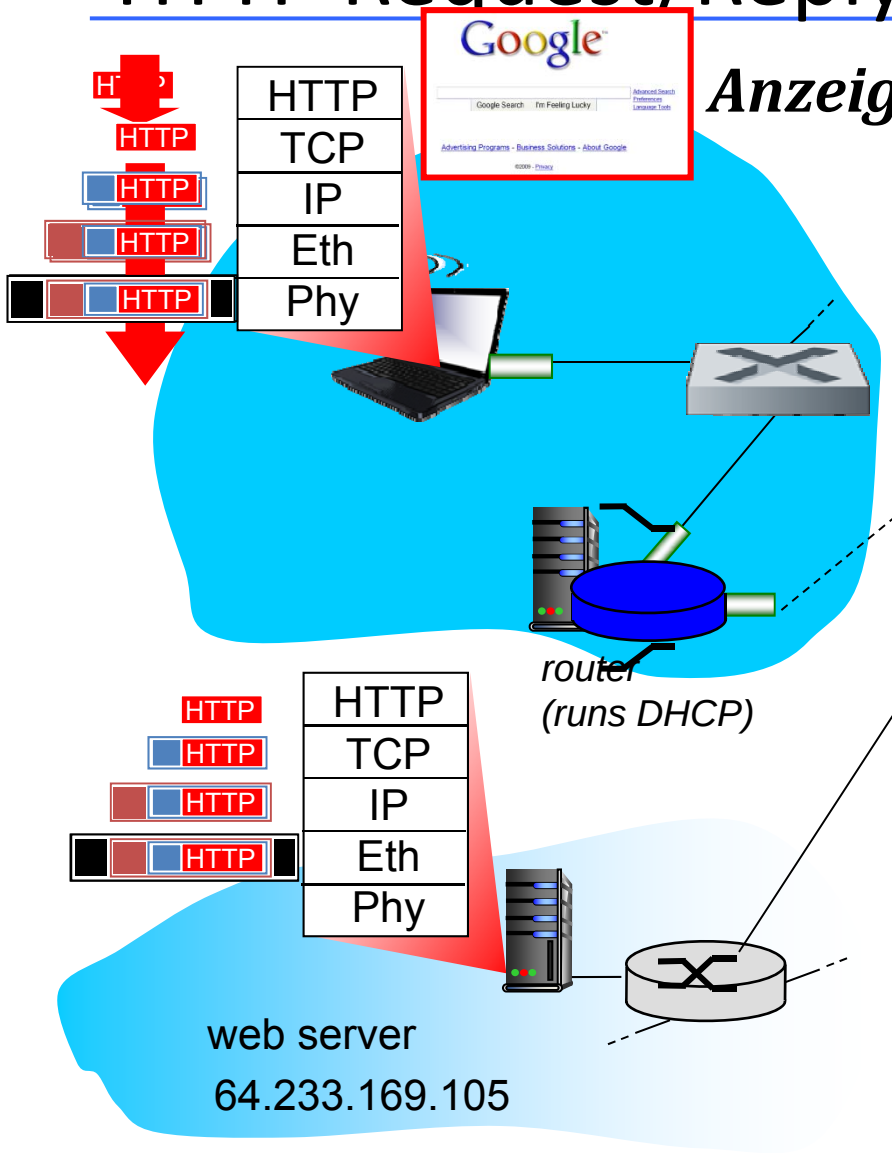
TCP-Verbindung für HTTP



- Zuerst einen *TCP-Socket* zum Web-Server eröffnen
- TCP *SYN Segment* (Schritt 1 im 3-Wege Handshake) *wird zum Web-Server geroutet* (über mehrere Domänen)
- Web-Server antwortet mit *TCP SYNACK* (Schritt 2 im 3-Wege Handshake)

TCP-Verbindung aufgebaut

HTTP Request/Reply



- *HTTP Request* wird über den TCP-Socket gesendet
- IP-Datagram, das denHTTP-Request beinhaltet, wird zu www.google.com *geroutet*
- Web-Server antwortet mit *HTTP-Reply* (das die Web-Seite beinhaltet)