

Rechnernetze

Kapitel 3: Link Layer

Prof. Dr. Wolfgang Mühlbauer

Fakultät für Informatik

`wolfgang.muehlbauer@th-rosenheim.de`

Wintersemester 2021/22

Slides are based on:

J. Kurose, K. Ross: Computer Networks – A Top-Down Approach

A. Tanenbaum, D. Wetherall: Computer Networks

- ❑ **Einführung**
- ❑ Rahmenbildung, Fehlererkennung
- ❑ Ethernet 802.3
- ❑ Mehrpunktverbindungen, Vielfachzugriff
 - CSMA/CD beim „Legacy Ethernet“
 - CSMA/CA bei WLAN
- ❑ Punkt-zu-Punkt Verbindungen, „geswitchte“ Netze

Terminologie

□ Ende-zu-Ende Pfad:

- Besteht aus vielen, **heterogenen** Links.
- Beispiel: Von HTTP Client zu Webserver über WLAN, Ethernet und Mobilfunknetz.

□ Hosts und Nodes

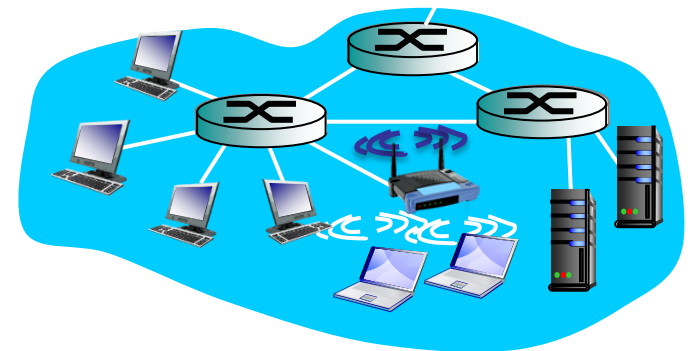
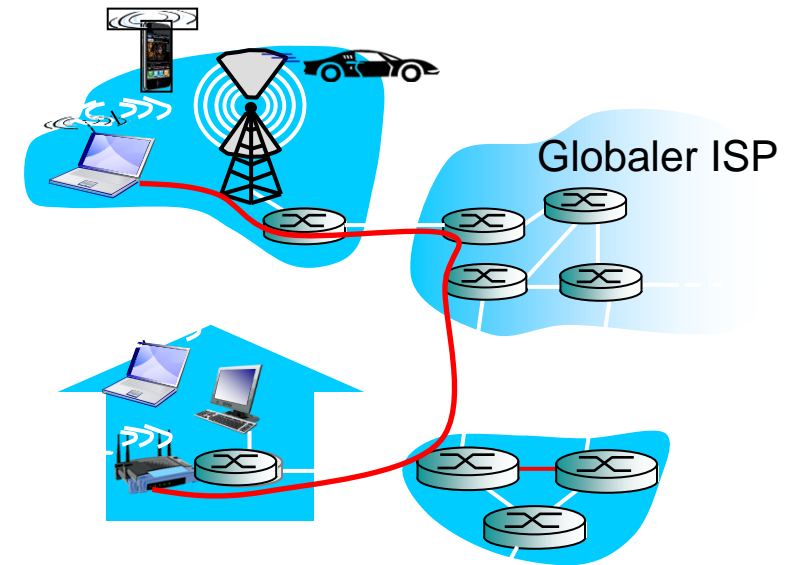
- *Host* == Endpunkt eines Ende-zu-Ende Pfades
- *Node* == Jedes Gerät, das am Netzwerk teilnimmt also Host, Router, Switch, Access Point, usw.

□ Link

- Verbindet benachbarte "Nodes"
- WLAN, Ethernet, Mobilfunk, (Bluetooth), usw.

□ Frame

- Nachricht auf Schicht 2
- Frame ist "Briefumschlag" für Schicht 3 Paket.



Link (Schicht 2)

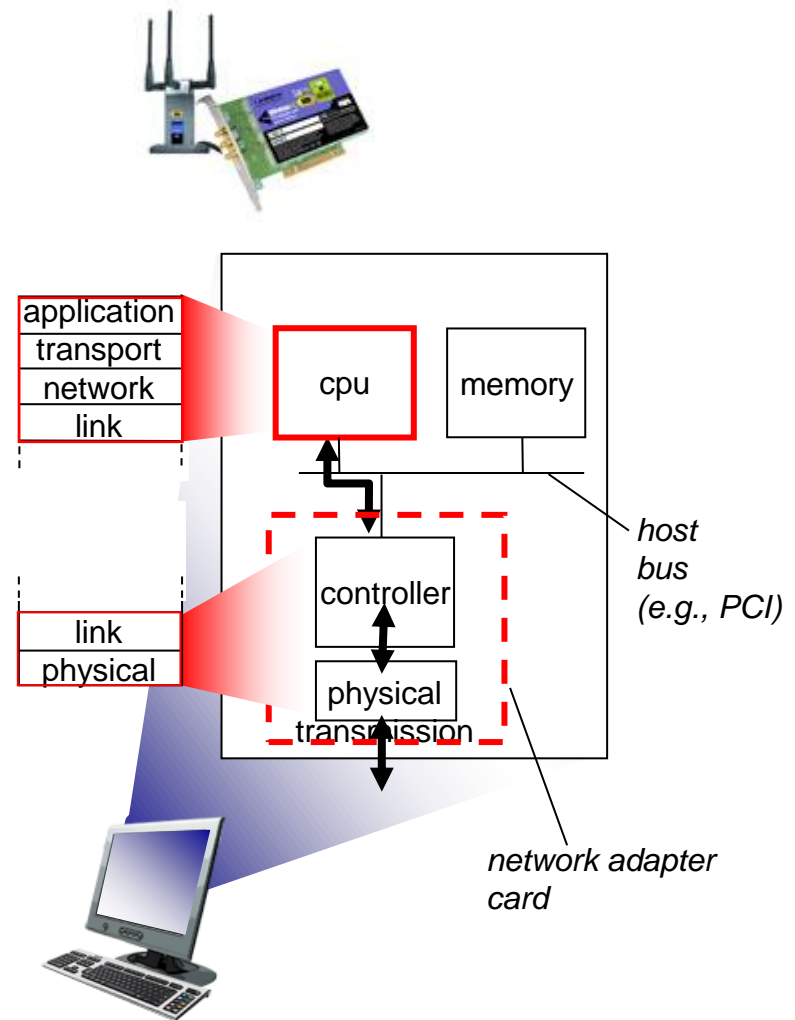
— Ende-zu-Ende Pfad (Schicht 3)

Dienste der Link Layer (dt. "Sicherungsschicht")

- ❑ **Übertragung von Frames zwischen benachbarten Nodes**
- ❑ **Rahmenbildung (engl. Framing)**
 - Positionsrichtige Erkennung von Zeichen, Erkennung von Blockbegrenzungen.
 - *Frame == Header + Payload*
 - Hier Payload == IP Paket
- ❑ **Vielfachzugriff: Wer darf wann das Medium nutzen?**
 - Nötig, falls „Multipunktmedium“
 - Beispiele: WLAN, Satellitennetze, Zugangsnetz bei Kabelanschluss, klassisches Ethernet
- ❑ **Fehlererkennung und -korrektur**
 - Umgang mit Bitfehlern auf der Physical Layer.
 - Hinzufügen von Redundanz, um Fehler zu erkennen bzw. zu korrigieren.
- ❑ **Zuverlässige Datenübertragung ("Reliable Data Delivery")**
 - Korrektur von Paketverlusten, korrekten Reihenfolge, Vermeidung von Duplikaten
 - Bei WLAN teilweise, bei Ethernet gar nicht!

Wo implementiert man die Link Layer?

- ❑ In **allen** Nodes
 - Auch Router und Switches!
 - Nicht in Hubs!
- ❑ Implementierung der meisten Funktionalität in Hardware
 - Fehlererkennung
 - Rahmenbildung
 - ...
- ❑ **Network Interface Card (NIC)** oder Netzwerkkarte
 - Implementiert große Teile der Link Layer und der Physical Layer (Leitungscode, etc.).
 - Über Bus mit CPU verbunden.



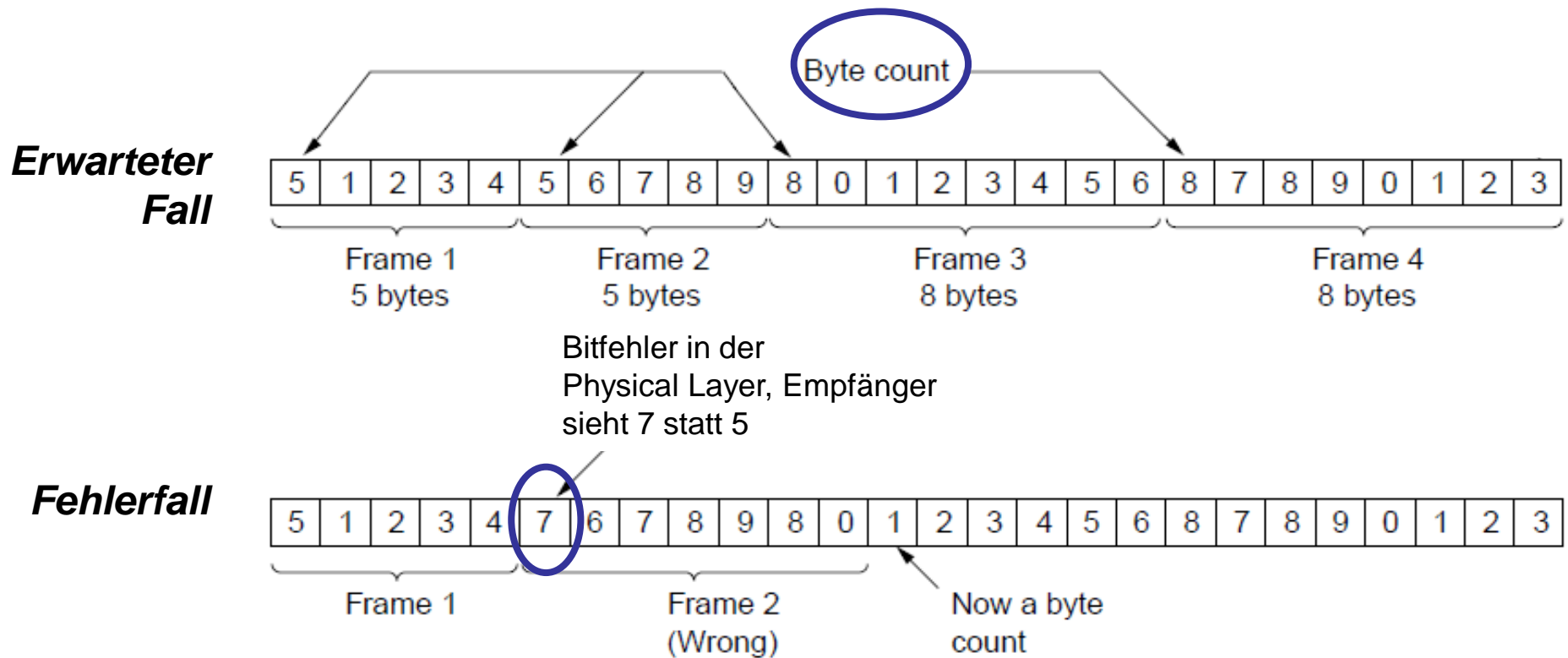
- ❑ Einführung
- ❑ **Rahmenbildung, Fehlererkennung**
- ❑ Ethernet 802.3
- ❑ Mehrpunktverbindungen, Vielfachzugriff
 - CSMA/CD beim „Legacy Ethernet“
 - CSMA/CA bei WLAN
- ❑ Punkt-zu-Punkt Verbindungen, „geswitchte“ Netze

Rahmenbildung (engl. Framing)

- ❑ Physical Layer empfängt und sendet **Bitstrom**.
- ❑ Fehlerbehandlung durch Link Layer nur möglich
 - falls Bits in endliche Sequenzen (= **Frame**) zerlegt werden.
 - Frame Redundanz hat (z.B. Checksum), siehe nächster Abschnitt.
- ❑ **Probleme**
 - Wie erkennt Empfänger Frameanfang und –ende in Bitstrom?
 - Wie überträgt man beliebige Bit- und Zeichenkombinationen?
- ❑ **Lösungsansätze**
 - *Byte Count*
 - *Byte Stuffing*
 - *Bit Stuffing*
 - *Coderegolverletzungen*: Blockbegrenzung durch Verwendung ungültiger Codes in Physical Layer

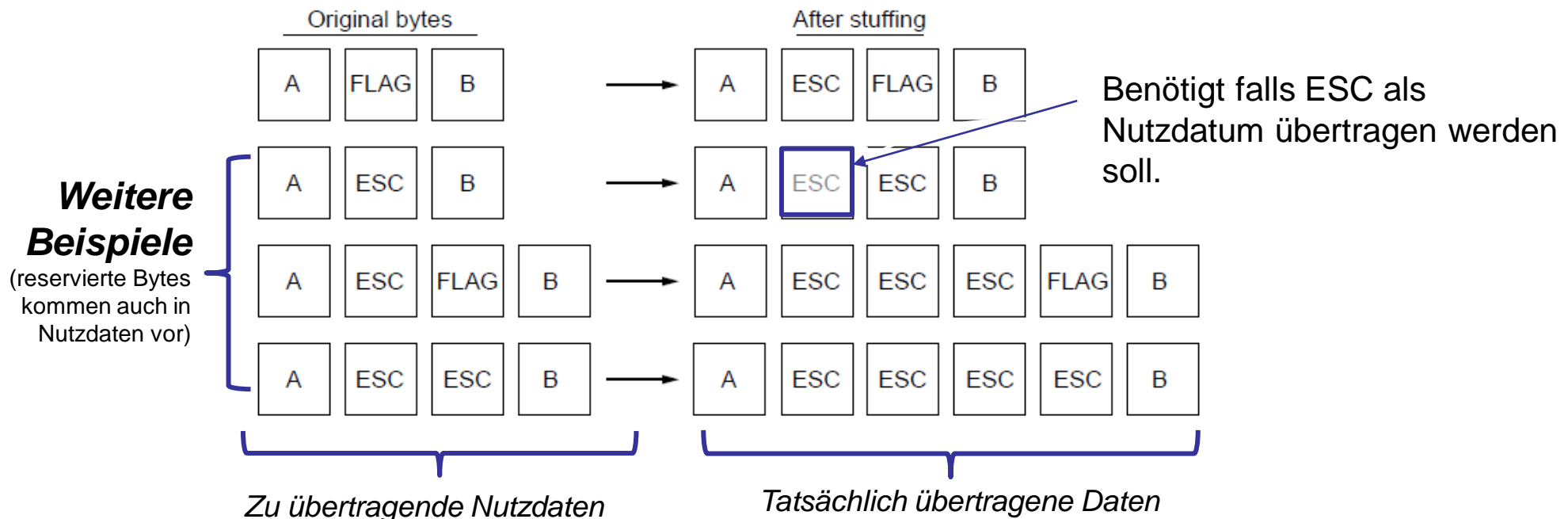
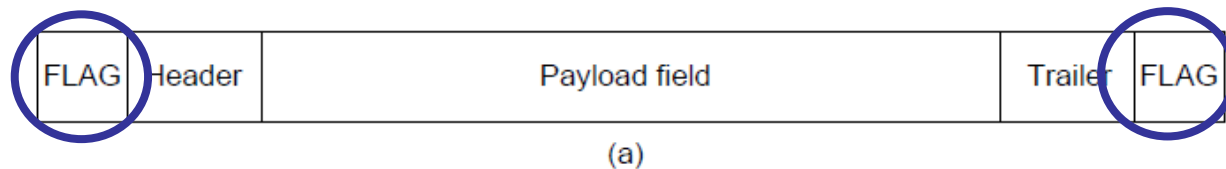
Byte Count: Längenangabe der Nutzdaten

- ❑ Jeder Frame beginnt mit Feld, das Anzahl der enthaltenen Bytes angibt.
- ❑ Nachteil: Erneute Synchronisation nach Fehler schwierig bzw. unmöglich!



Byte Stuffing: Steuerzeichen und Zeichenstopfen

- ❑ Reserviertes Byte **FLAG** markiert **Frameanfang und -ende**
- ❑ Mögliches Problem: FLAG kommt in Nutzdaten vor
 - Ausweg: Verwenden eines weiteren reservierten Bytes **ESC** (=Escape)
- ❑ Einfache Synchronisation nach Fehler, aber Overhead!



Bit Stuffing: Begrenzungsfeld und Bitstopfen

- ❑ Vorteil: Framelänge muss **kein Vielfaches** von 8 Bit sein!
- ❑ Jeder Frame beginnt mit **speziellem, reservierten** Bitmuster:
 - Hier im Beispiel: **01111110**
- ❑ **Regeln**
 - Beim **Senden** : Nach 5 1er-Bits wird **immer** ein 0-Bit eingefügt.
 - Beim **Empfang**: Nach 5 1er Bits wird **immer** ein 0-Bit gelöscht.

Datenbits 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 0

Übertragene Bits
(nach Stuffing) 0 1 1 0 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 0 1 0 0 1 0

Stuffed bits

The diagram illustrates the bit stuffing process. The original data bits are shown as a sequence of 21 bits: 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 0. These are then stuffed with three zeros at specific positions (indicated by arrows from the label 'Stuffed bits') to ensure the total length is a multiple of 5. The resulting transmitted bits are: 0 1 1 0 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 0 1 0 0 1 0.

Publikums-Joker: Bit Stuffing (Single Choice)

Zwei Frames sind gleich groß, aber unterscheiden sich im Inhalt. Es wird Bit Stuffing verwendet.

Welche Aussage ist **falsch**?

- A. Bit Stuffing wird meist in Hardware implementiert.
- B. Die dazugehörigen Frames haben verschiedene Checksums.
- C. Das Propagation Delay beim Senden ist für beide Pakete gleich groß.
- D. Beim Senden der IP Pakete sind die dazugehörigen Frames gleich groß.



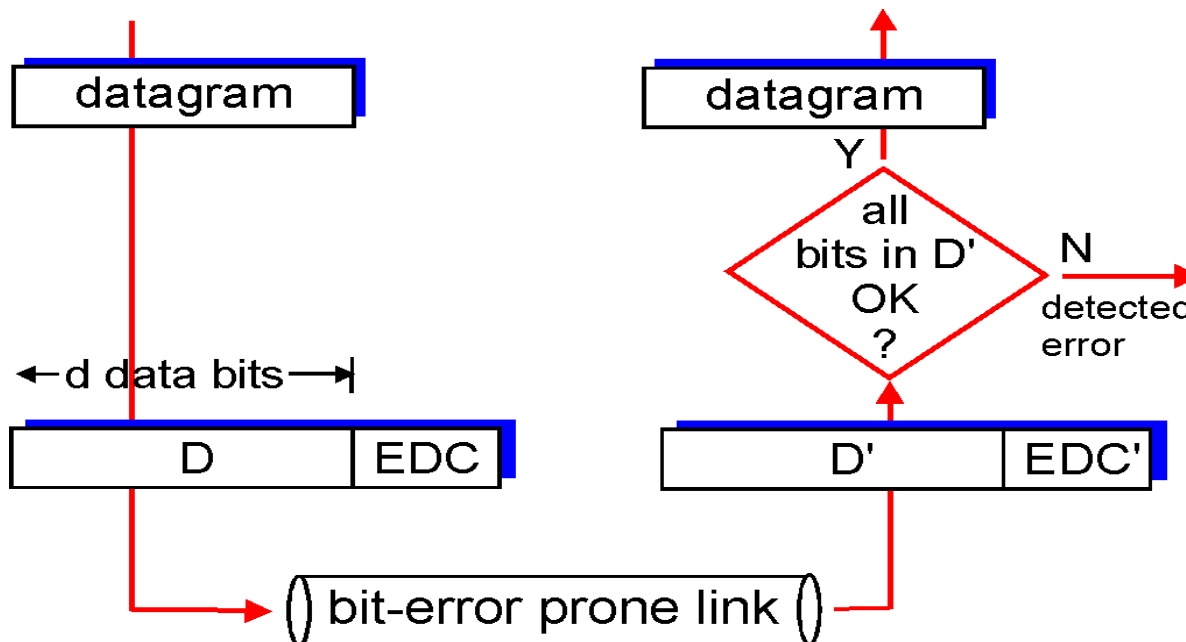
Umgang mit Bitfehlern

- ❑ **Ursachen für Bitfehler:** Rauschen, Dämpfung, Verzerrung, usw.
- ❑ **Grundidee**
 - Rahmenbildung (engl. „Framing“)
 - Redundanz (z.B. Prüfsumme über Frames)
- ❑ **Fehlerkorrektur** durch Redundanz
 - Benötigt viel Redundanz
 - Üblich bei nicht "wiederholbaren" Medien (CD, RAM, etc.), nicht bei TCP/IP
- ❑ **Fehlererkennung** durch Redundanz
 - Fehler wird nur erkannt, aber nicht behoben.
 - Maßnahmen:
 - **Ethernet 802.3:** Keine Retransmission. Wiederanforderung des fehlerhaften Blocks durch höhere Schichten, z.B. TCP nach Ablauf eines Timeouts.
 - **WLAN 802.11:** Aktive Wiederanforderung des fehlerhaften Blocks durch Link Layer (=Active Repeat Request)

Allgemeiner Ansatz

□ Bezeichner

- **EDC**: Error Detection und Correction Bits
- **D**: Nutzdaten, die durch Fehlerbehandlung abgesichert werden



Ansätze

(teilweise in anderen Lehrveranstaltungen besprochen)

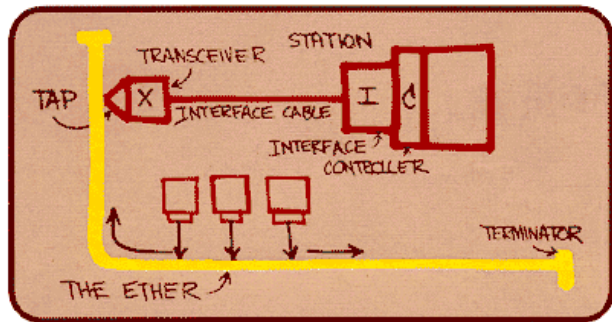
- Paritätsbits
- Checksumme
 - IP, TCP
- Cyclic Redundancy Check (CRC)
 - Ethernet, WLAN

EDC == EDC' ?

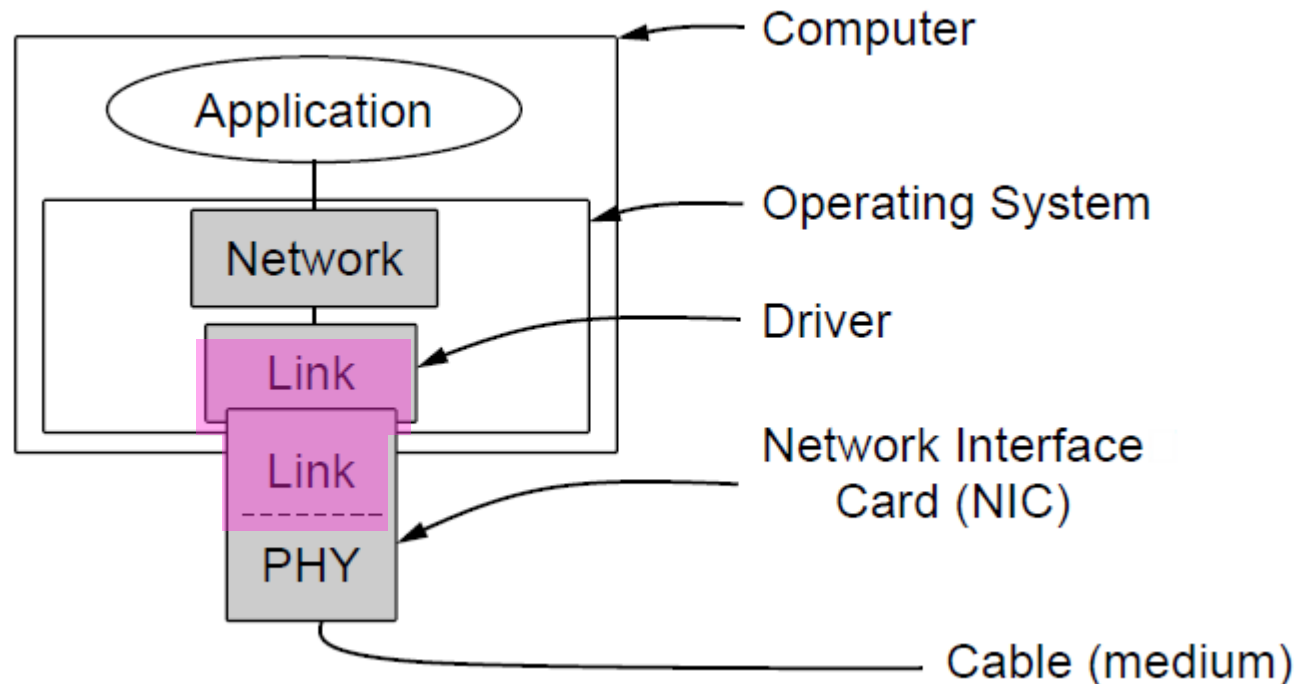
- ❑ Einführung
- ❑ Rahmenbildung, Fehlererkennung
- ❑ **Ethernet 802.3**
- ❑ Mehrpunktverbindungen, Vielfachzugriff
 - CSMA/CD bei klassischem Ethernet
 - CSMA/CA bei WLAN
- ❑ Punkt-zu-Punkt Verbindungen, „geswitchte“ Netze

Ethernet


- ❑ Dominierende LAN Technologie
- ❑ Netzwerkkarten sind preiswert (< 3 Euro)
- ❑ Geschwindigkeiten nahmen ständig zu: 10 Mbps – 10 Gbps



Ethernet Schema von Metcalf

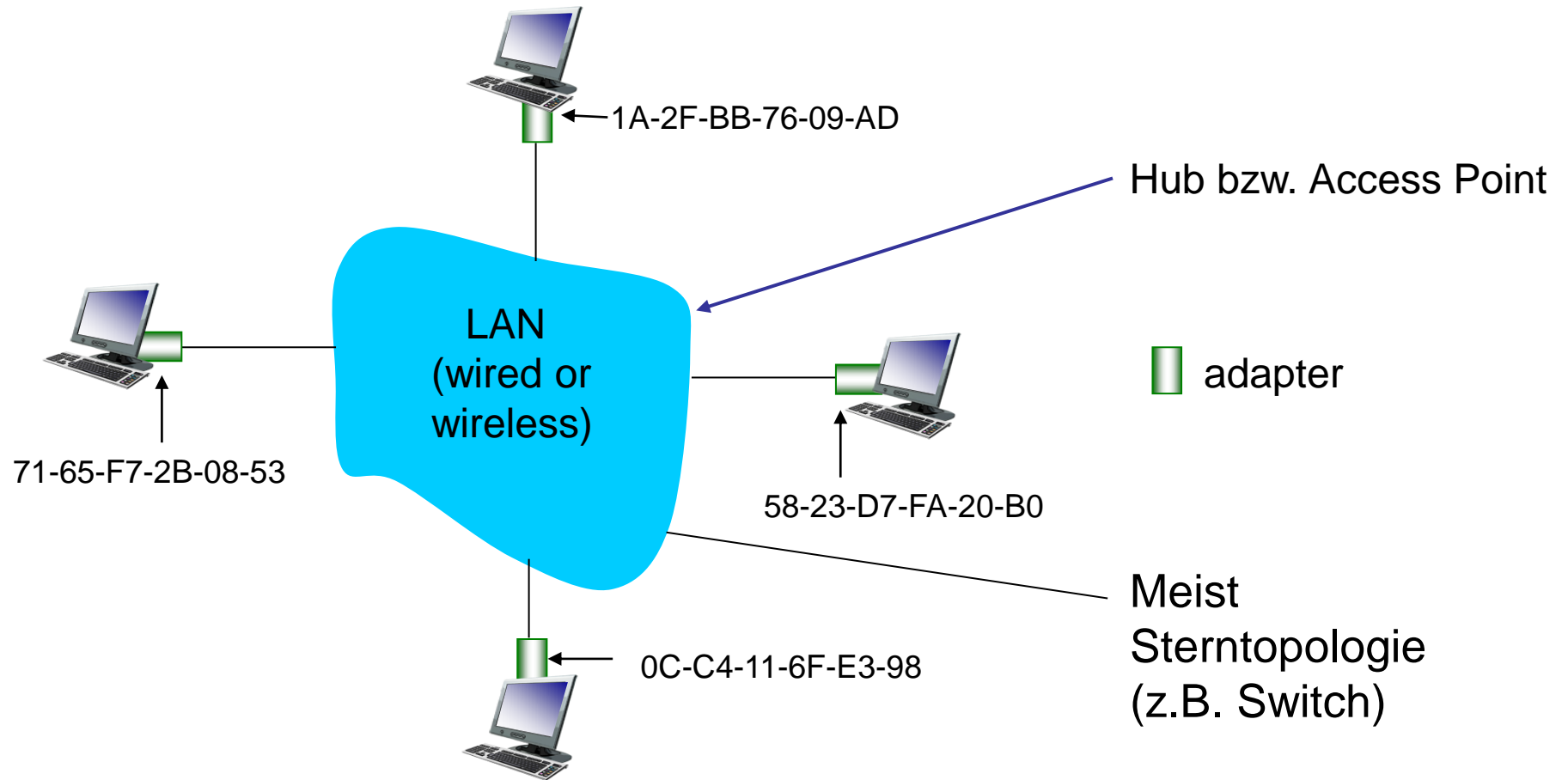


MAC Adresse

- ❑ **Adresse** der Link Layer
 - Identifiziert Nachbarn, wichtig vor allem bei Mehrpunktverbindungen.
 - Nur lokal gültig (LAN, WLAN).
- ❑ Jedes Interface eines Hosts / Routers hat eigene MAC Adresse
 - Ein Gerät kann also mehrere MAC Adressen haben.
- ❑ Ethernet und WLAN: 48 Bit
 - Teils fest mit Netzwerkkarte verknüpft
 - Manchmal per SW änderbar
 - Beispiel: 1A-2F-BB-76-09-AD 
 - Broadcast-Adresse: FF-FF-FF-FF-FF-FF
- ❑ Adressen werden durch IEEE zugewiesen
 - Hersteller kaufen Adressräume

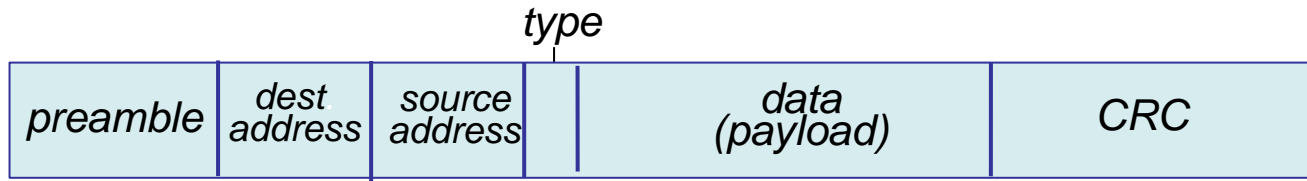
Hexadezimal
(jede Nummer
repräsentiert 4 Bits)

LAN Adressen



Jede Netzwerkkarte muss **eindeutige** MAC Adresse haben!

Ethernet 802.3 Frames



❑ **Präambel**

- Zu Beginn: 7mal 10101010, dann 1mal 10101011
- Synchronisation von Sender- und Empfänger, Start des Frames.

❑ **Adressen**

- Jeweils 6 Byte Sende- und Empfänger MAC Adresse
- Netzwerkkarte leitet empfangenen Frame nur an Betriebssystem weiter, falls Dest. MAC der eigenen MAC entspricht. Ausnahmen:
 - Dest. MAC ist FF:FF:FF:FF:FF:FF
 - *Promiscuous Mode*

❑ **Type**

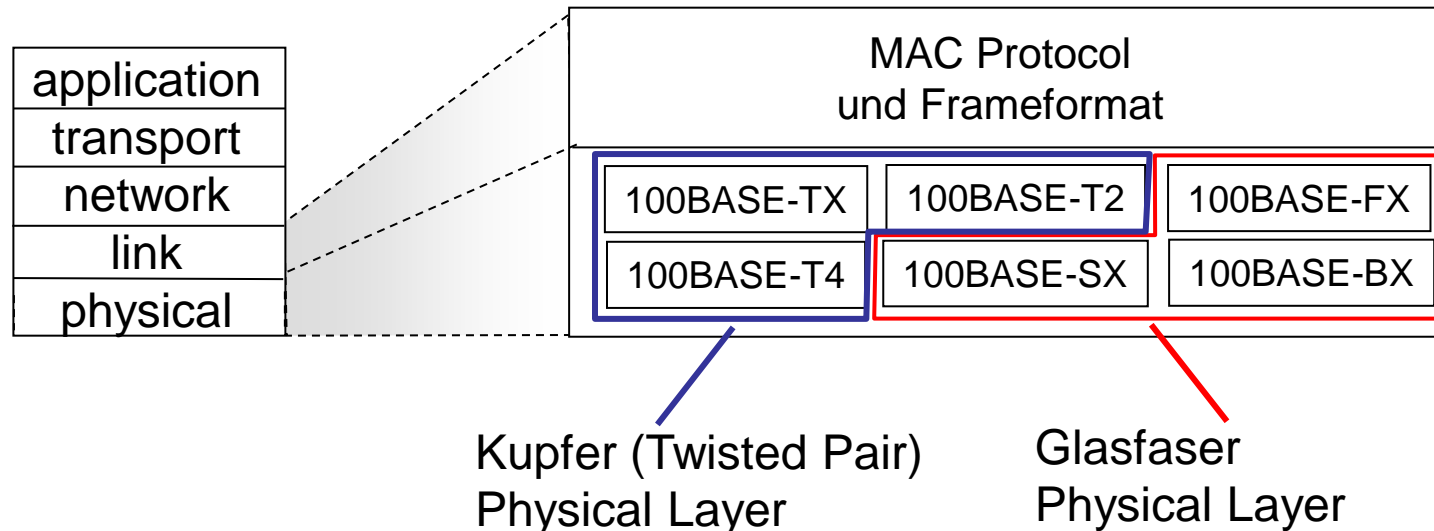
- 2 Byte → spezifiziert Art des Netzwerkprotokolls
- IPv4=0x0800, IPv6=0x86DD, ...

❑ **CRC**

- 4 Byte

Ethernet 802.3: Eigenschaften

- ❑ **Verbindungslos**
 - Kein Verbindungsaufbau vor Datenaustausch
- ❑ **Keine zuverlässige (reliable) Verbindung**
 - Verlust von Frames möglich
 - Absicherung muss durch höhere Schichten erfolgen.
- ❑ **Vielfachzugriff**
 - Nur bei „altem“ Ethernet: Unslotted CSMA/CD mit Binary Backoff.
- ❑ **Unterstützt verschiedene Übertragungsmedien**
 - Beispiel: 100BASE-SX, 100BaseTX



Publikums-Joker: MAC Adressen (Single Choice)

Welche der folgenden Aussagen ist **falsch**?

(Annahme: alle Geräte sind ans Internet angebunden und haben ausschließlich Ethernet Interfaces)

- A. Ein klassischer Router hat mehrere MAC Adressen.
- B. Ein klassischer Switch hat mehrere MAC Adressen.
- C. Ein Host kann mehr als 1 MAC Adresse haben.
- D. Die MAC Adresse lässt sich teils leicht per SW ändern.



- ❑ Einführung
- ❑ Rahmenbildung, Fehlererkennung
- ❑ Ethernet 802.3
- ❑ **Mehrpunktverbindungen, Vielfachzugriff**
 - CSMA/CD beim „Legacy Ethernet“
 - CSMA/CA bei WLAN
- ❑ Punkt-zu-Punkt Verbindungen, „geswitchte“ Netze

Zwei Arten von „Links“

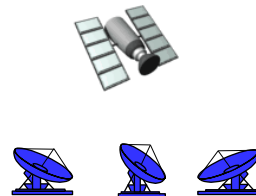
- ❑ **Link:** Kommunikation zwischen **benachbarten** Hosts, Routern und Switches
- ❑ **Punkt-zu-Punkt**
 - 2 kommunizierenden Nodes haben einen **eigenen dedizierten** Link für jede Richtung.
 - Beispiel: Ethernet LAN, das nur Switches verwendet und PPP für SONET und DSL
- ❑ **Mehrpunktverbindungen**
 - > 2 kommunizierende Nodes **teilen** sich einen Link.
 - WLAN 802.11, Bluetooth 802.15, „Legacy Ethernet“, Powerline



Geteilte Leitung
„Legacy Ethernet“



Geteiltes HF
802.11



Geteiltes HF
(Satellit)



Menschen auf einer Party
(geteilte Musik)

Vielfachzugriff (engl. Media Access)

❑ Wer darf wann senden?

❑ **Annahmen**

- Geteilter Broadcastkanal
- Interferenz == **Kollision** falls mehrere Stationen gleichzeitig senden.

❑ **Multiple Access Control**

- *Verteilter* Algorithmus, der entscheidet, wann Host senden darf
- Entscheidung muss "**inband**" getroffen werden (kein extra Kanal)

❑ **Anforderungen** (Annahme: Link hat Kapazität R)

- Nur 1 Host möchte senden → Host sendet mit Rate R !
- M Hosts senden → Jeder Host bekommt Rate R/M („*Fairness*“)
- Dezentral ohne koordinierende Station
- Einfach zu implementieren

Random Access Verfahren

- ❑ Kollision wird zugelassen
 - Falls >2 Stationen senden, tritt **Kollision** auf

- ❑ **Zu lösen ist:**
 - Wie **erkennt** man Kollision?
 - Wie **reagiert** man auf eine Kollision?
 - Erneutes Übertragen solange bis Erfolg.
 - Ggfs. zufällige Wartezeit, um weitere Kollisionen zu verhindern.

- ❑ **Beispiele** von Random Access Verfahren
 - Slotted ALOHA, Unslotted ALOHA
 - **CSMA/CD** („Legacy Ethernet“, PowerLine),
 - **CSMA/CA** (WLAN)

Carrier Sense Multiple Access (CSMA)

- ❑ *Teilweise*: Unterteilung der Zeit in Timeslots
 - Sender muss nach Ablauf eines Timeslots erst mal aufhören.

- ❑ **Carrier Sensing** == Mitlauschen auf dem Kanal.
 - *Falls Kanal frei*: Übertragung beginnen.
 - *Falls Kanal belegt*: Verschiebe Übertragung.

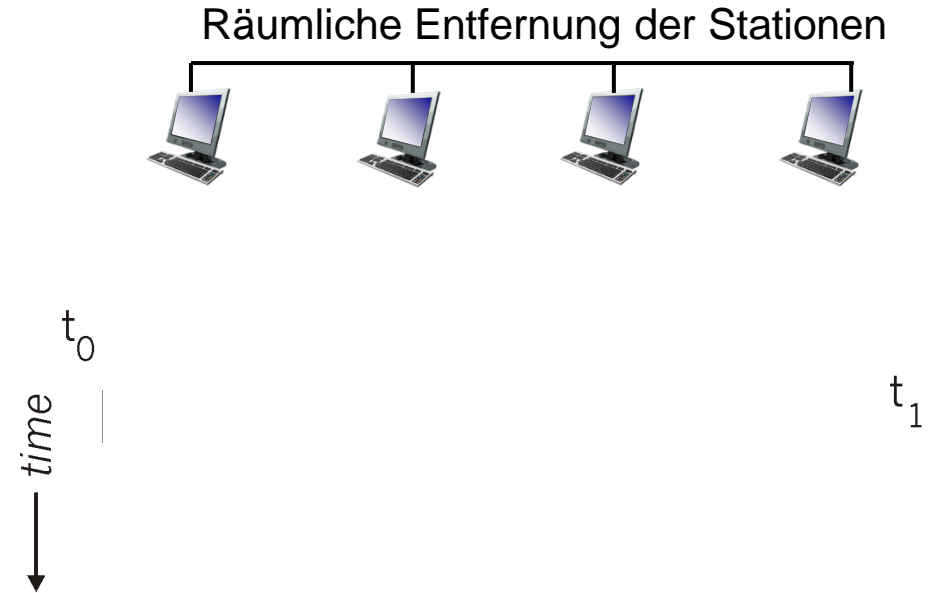
- ❑ **Varianten** falls Kanal belegt:
 - *1-persistent*: Sende, sobald Kanal wieder frei wird.
 - *p-persistent*: Sende im nächsten Slot mit der Wahrscheinlichkeit p falls Kanal frei ist (bei "Slotted" Verfahren).
 - *Non-persistent*: Warte eine zufällig Zeit und prüfe erneut ob Kanal frei ist
→ Legacy Ethernet!

CSMA: Kollisionen

- ❑ **Kollisionen** können trotz CSMA wegen des ***Propagation Delays*** d_{prop} auftreten.
 - Es dauert bis Sender von gleichzeitiger Übertragung eines anderen Senders erfährt.
 - *Folge*: d_{prop} hat Einfluss auf Kollisionswahrscheinlichkeit

- ❑ **Problem:**

- Bei spät erkannter Kollision ist das bereits losgesendete Teilpaket wertlos.



Carrier Sense Multiple Access (CSMA)

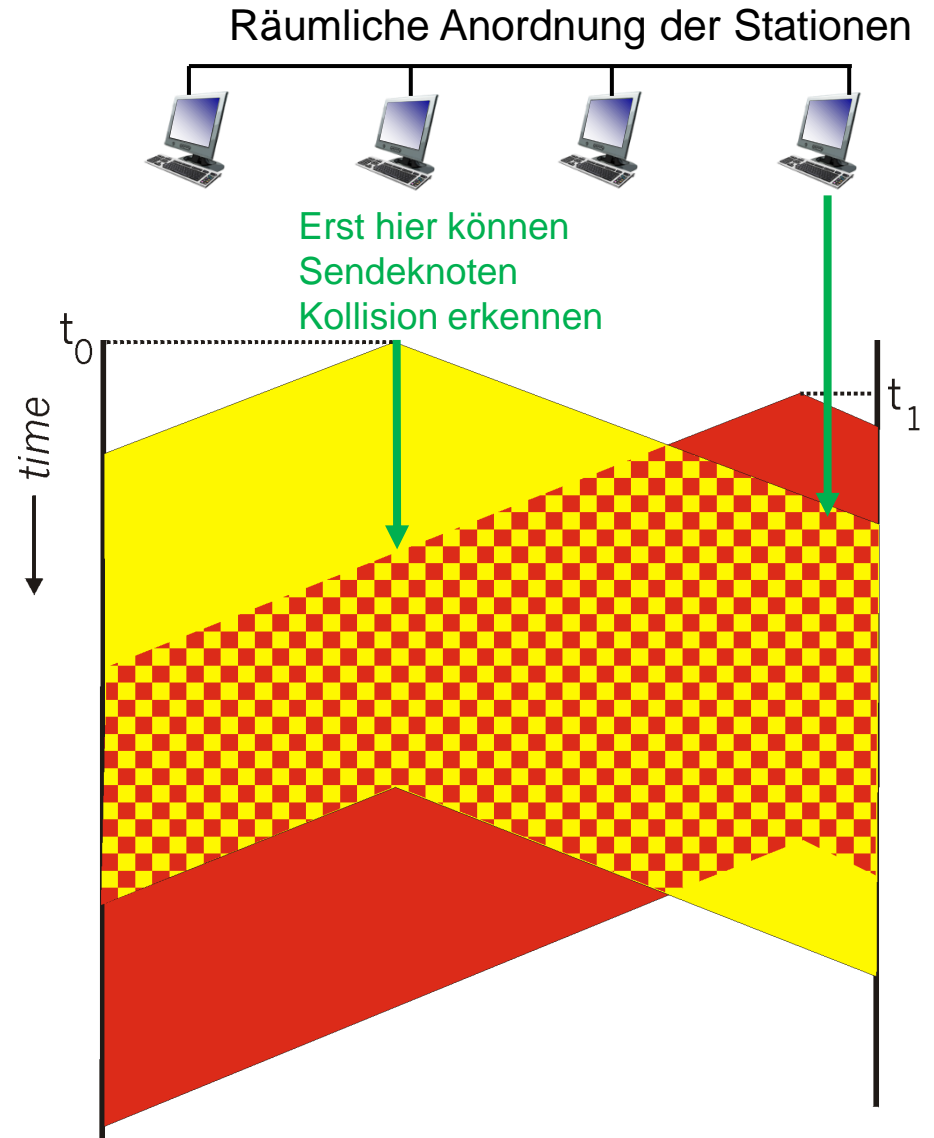
❑ **Kollisionen** können wegen **Propagation Delay d_{prop}** auftreten.

- Es dauert bis Sender von gleichzeitiger Übertragung eines anderen Senders erfährt.
- Folge: d_{prop} hat Einfluss auf Kollisionswahrscheinlichkeit

❑ **Problem:**

- Bei spät erkannter Kollision ist das bereits losgesendete Teilpaket wertlos.

Animation: http://pi4.informatik.uni-mannheim.de/pi4.data/content/animations/ethernet/ethernet_applet.zip



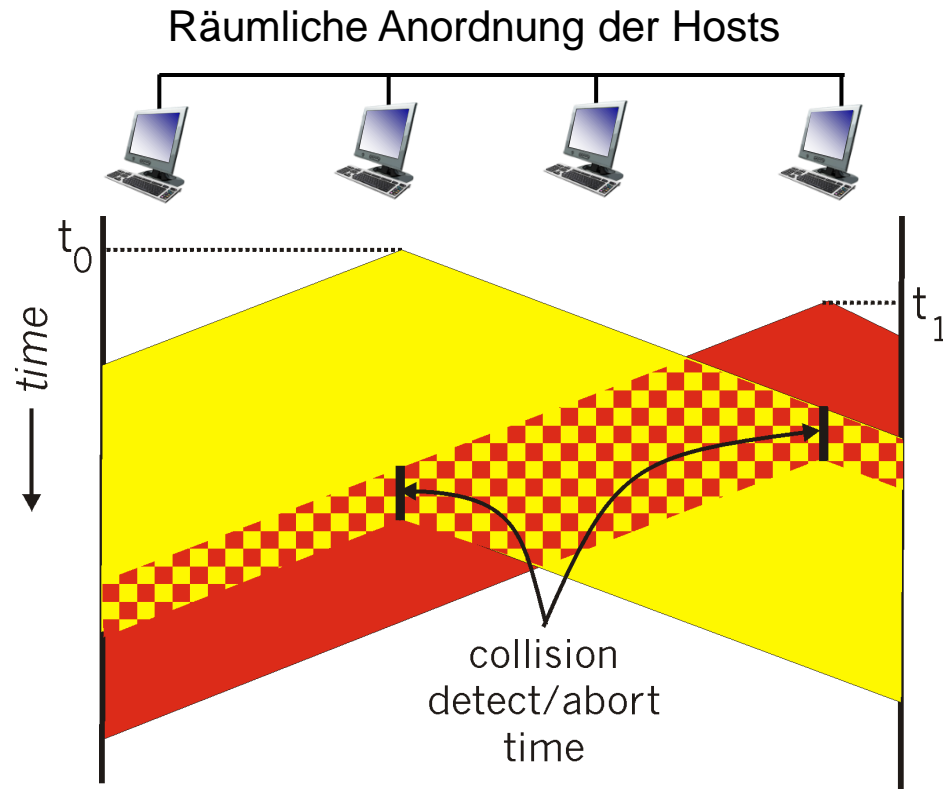
CSMA mit Kollisionserkennung: CSMA/CD

□ CSMA/CD: Carrier Sensing + Collision Detection

□ Collision Detection:

- „*Listen while talk*“: Sender hört während des Sendens Medium weiter zu, ob Kollisionen auftreten.
- Falls ja:
 - *Sofortiger* Abbruch der Übertragung
 - *Jam Signal* zur Benachrichtigung aller beteiligten Sender → diese sollen ihre Sendetätigkeiten einstellen.
- Erneuter Sendeversuch nach *zufälliger* Wartezeit.
- *Binary Exponential Backoff*: Verdoppelung der mittleren Wartezeit nach jeder erneuten Kollision.

Kollisionserkennung bei CSMA/CD



CSMA/CD beim „Legacy Ethernet“: Algorithmus

1. Netzwerkkarte (NIC) hört Kanal ab

- **Frei:** Frame wird gesendet
- **Belegt:** NIC wartet bis Kanal frei

2. NIC beobachtet während Senden ob Kollision auftritt.

- Falls keine Kollision → Übertragung erfolgreich.

3. Falls Kollision während Senden

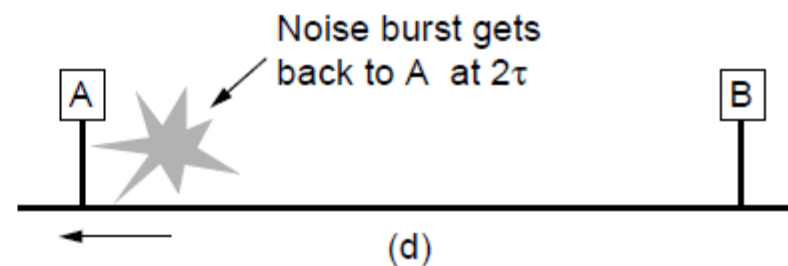
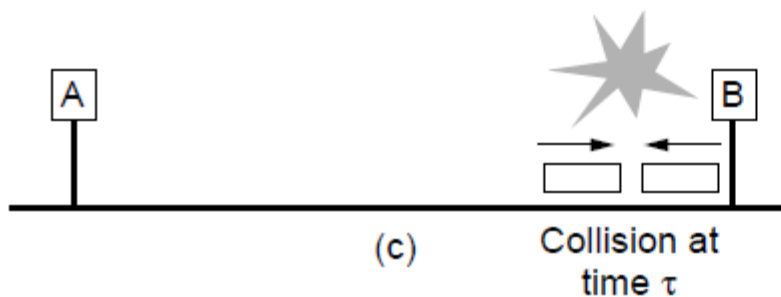
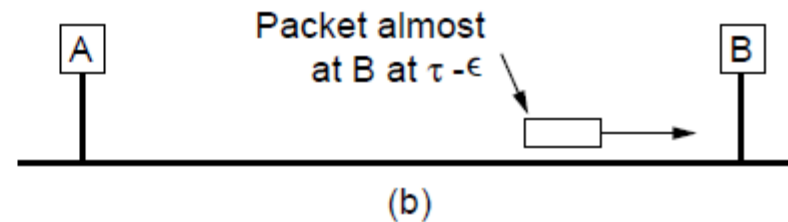
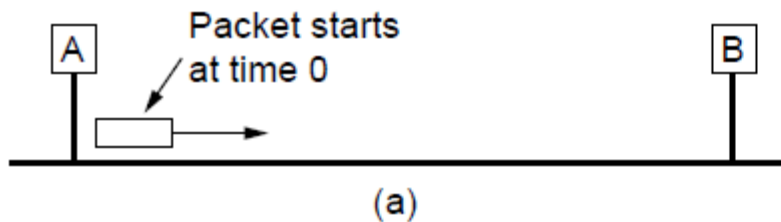
- Übertragungsabbruch
- Senden des JAM Signals

4. Nach Abbruch: **Exponential Backoff**

- Nach der m . Kollision: Wähle K **zufällig** aus $\{0, 1, 2, \dots, 2^m - 1\}$.
- NIC wartet $K * T$, wobei T bei 100 Mbps Ethernet die Zeit ist um 512 Bits zu übertragen.
- Anschließend gehe zu Schritt 2
- Backoff-Intervall wird also größer je mehr Kollisionen bereits aufgetreten sind.

Mindestlänge von Ethernet Frames

- ❑ Worst Case: Es kann 2τ Zeiteinheiten dauern, bis Kollision erkannt wird.
 - $\tau = d_{\text{prop}}$: Propagation Delay
- ❑ **Ziel** im klassischen Ethernet für Broadcast-Netze:
 - Sender soll Kollision noch erkennen können, **bevor er alle Bits seines Frames losgesendet hat** ("Transmission Delay").
- ❑ Deshalb benötigt: **Minimale Paketgröße** für verlässliche Kollisionserkennung.
 - $d_{\text{trans}} > 2d_{\text{prop}}$



Publikums-Joker: Mindestlänge bei Ethernet (Single Choice)

Welche der folgenden Aussagen bzgl. CSMA/CD ist **falsch**?

- A. Bei Gigabit Ethernet muss die Mindestpaketlänge kleiner sein als bei FastEthernet (100 Mbit/s).
- B. Hohe Bandbreite / Datenrate bedeutet, dass alle Ethernet Stationen räumlich **nahe** beieinander liegen müssen.
- C. CSMA/CD wird direkt in der Netzwerkkarte implementiert.
- D. CSMA/CD wird heutzutage im Ethernet nur noch selten verwendet.



CSMA/CA bei WLAN 802.11: Konzept

❑ **Carrier Sensing**

- Höre das Medium vor dem Senden ab. (wie bisher)

❑ **Congestion Avoidance (CA)**

- Versuche Kollisionen soweit als möglich zu vermeiden.
- Dennoch: Kollisionsbehandlung notwendig (Begriff „CA“ deshalb irreführend)

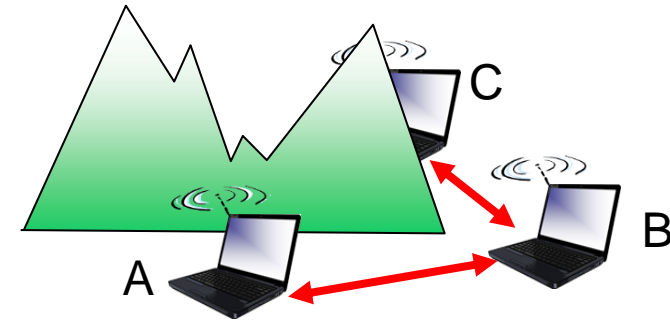
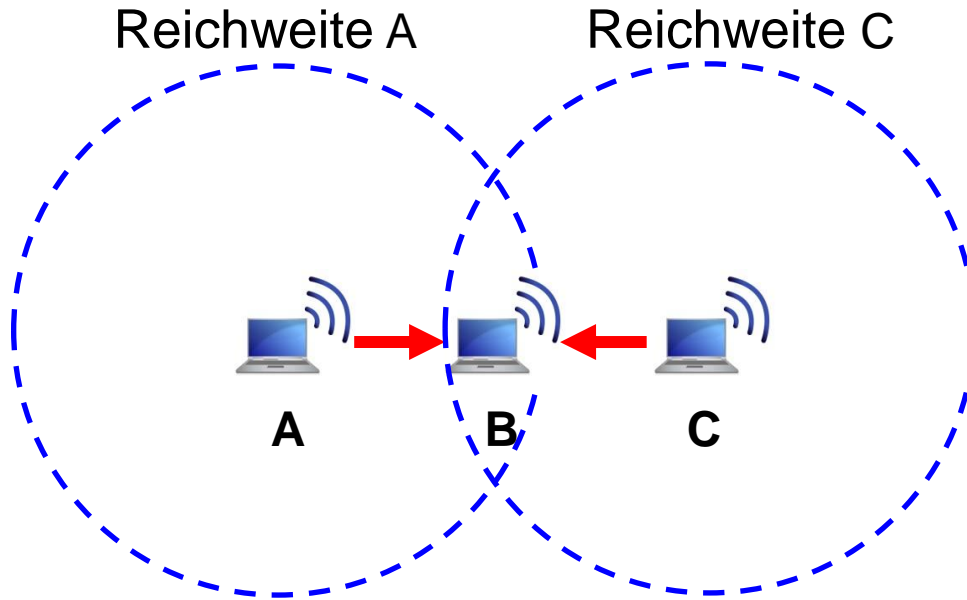
❑ **Binary (exponential) Backoff** (wie bisher)

- Nach der ***m***. Kollision, wähle zufällig ein ***K*** aus $\{0, 1, 2, \dots, 2^m - 1\}$.
- Warte dann ***K*** Zeitslots, bevor erneut ein Sendeversuch gestartet wird.
- Zufall hilft, eine erneute gleichzeitige Übertragung zu vermeiden.
- Längere Wartezeiten wahrscheinlich, falls hintereinander wiederholt Kollisionen auftreten.

❑ **Problem:** Kollisionserkennung bei WLAN schwierig.

- WLAN Stationen können sich *nicht alle gegenseitig hören*.
 - **Hidden Station Problem** (siehe nächste Folie)
- WLAN ist **halbduplex**
 - Meist kein Mithören während des Sendens implementiert, da empfangenes Signal sehr schwach im Vergleich zu gesendetem Signal.

Hidden Station Problem



- ❑ Versteckte Station: A kann Mitbewerber C nicht hören
 - A und C senden gleichzeitig → Kollision bei B
- ❑ Mögliche Ursachen:
 - C zu weit von A entfernt oder Hindernis zwischen A und C.
- ❑ Eigentlich müssen **Kollisionen beim Empfänger erkannt werden.**

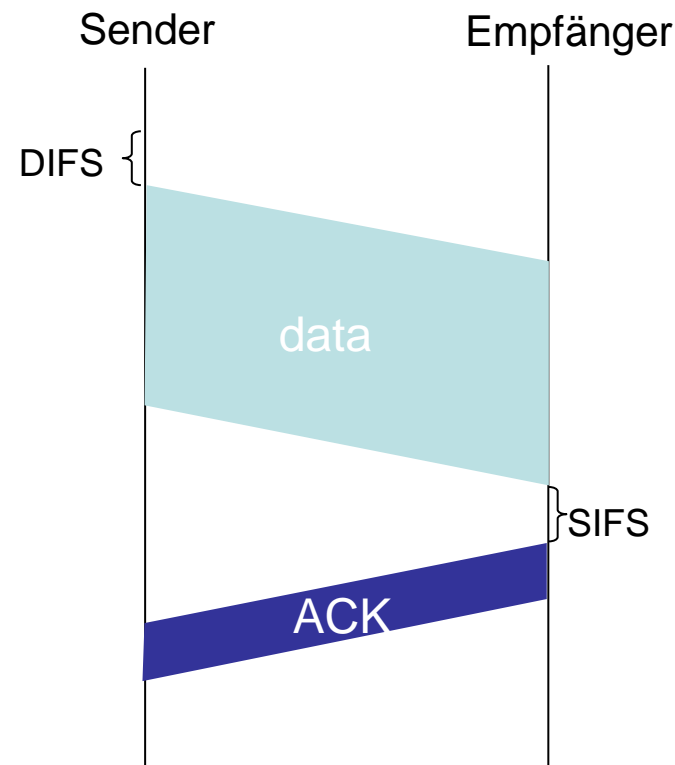
CSMA/CA Algorithmus

□ Sender (bei Sendewunsch)

- **Frei:** Kanal mind. für Zeitspanne DIFS frei
 - Sende *kompletten* Frame (ohne Carrier Sense)
- **Belegt:** Kanal gerade belegt.
 - Bereits hier Exponential Backoff
 - Unterschied zu CSMA/CD!
 - Höre Kanal ständig ab, **dekrementiere Timer** nur während Zeiten, in denen Kanal frei.
 - Erneute Übertragung wenn Timer ausläuft
- Falls **kein ACK** eintrifft
 - Gehe in "Belegt"-Fall
 - Vergrößere ggfs. Backoff Intervall.

□ Empfänger

- Bestätigt Datenempfang durch ACK nach Zeitspanne SIFS (=Kollisionserkennung beim Empfänger)



SIFS kürzer als DIFS: Priorisierung von ACKs!

CSMA/CA: Beispiel

- ❑ Zufällige Wartezeiten auch ohne Kollision
 - Wenn Carrier Sense bei Sendewunsch ergibt, dass Kanal gerade belegt.
- ❑ Backoff Timer zählt nur runter, wenn Kanal auch wirklich frei.
- ❑ Bleiben ACKs aus → Retransmissions (hier nicht gezeichnet)

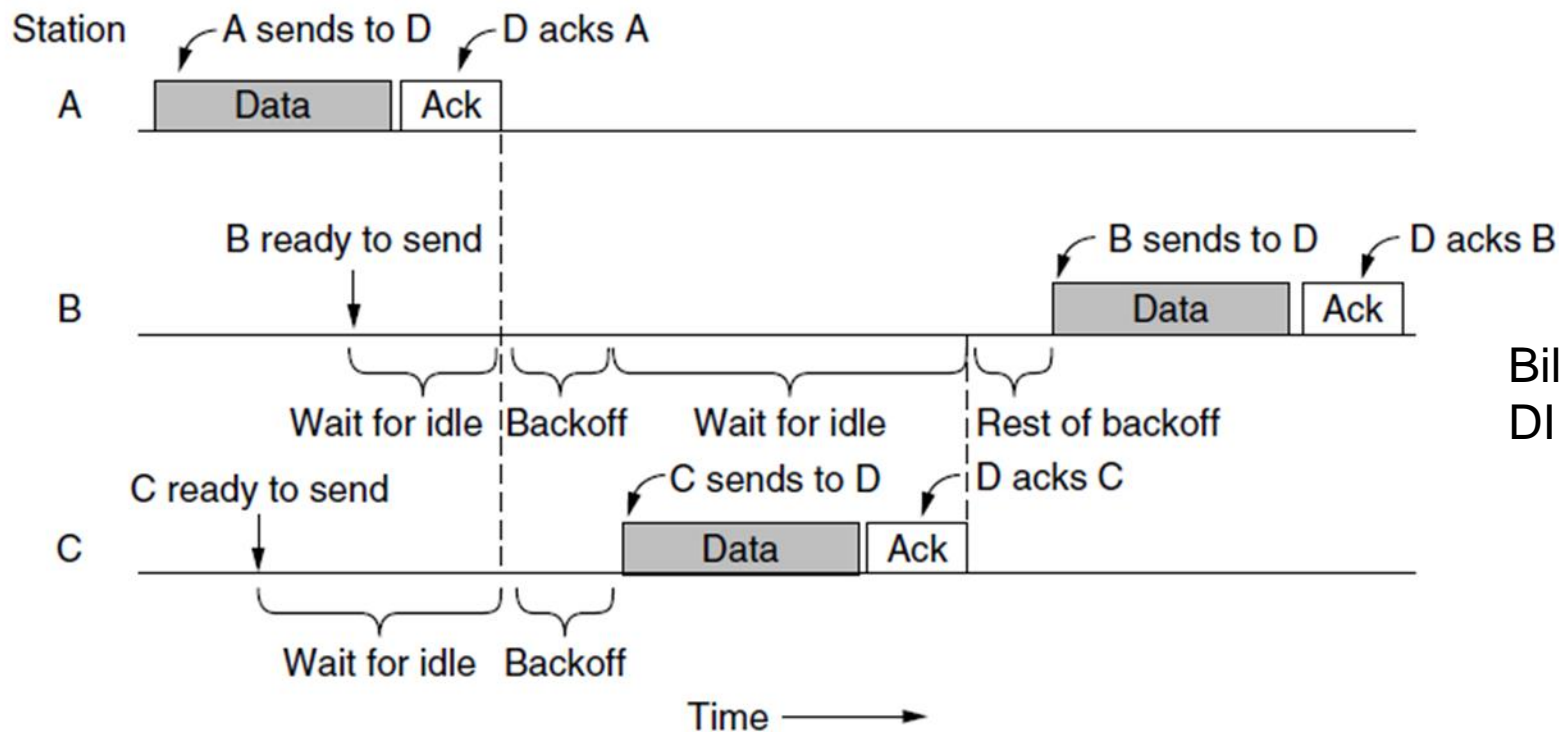


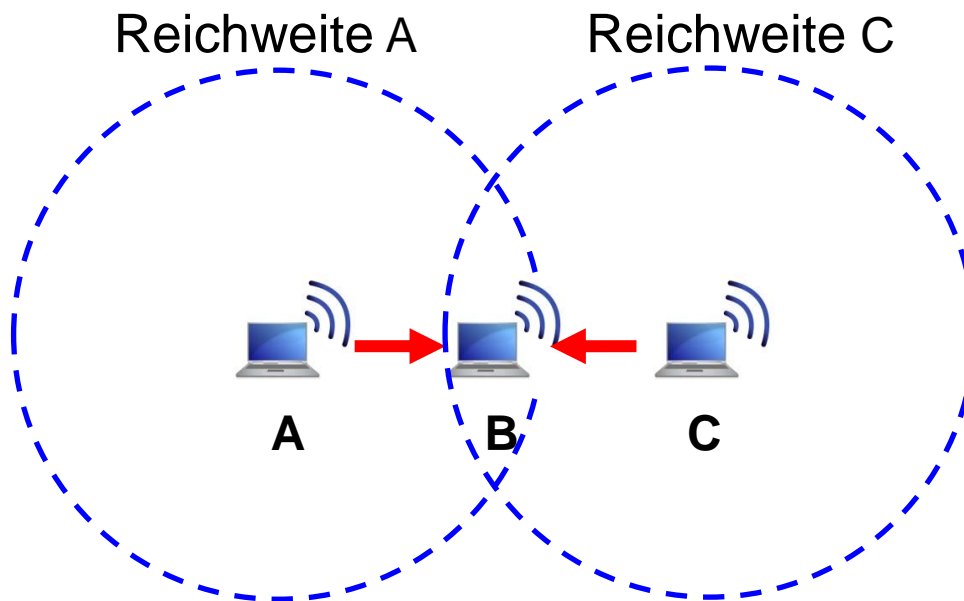
Bild ohne SIFS und
DIFS Wartezeiten!

Quelle: Tanenbaum

Publikumsjoker: Hidden Station (Single Choice)

- ❑ Vermeidet CSMA/CA Kollisionen im Hidden Station Problem *immer*?

- A = JA
- B = NEIN



- ❑ Einführung
- ❑ Rahmenbildung, Fehlererkennung
- ❑ Ethernet 802.3
- ❑ **Mehrpunktverbindungen, Vielfachzugriff**
 - CSMA/CD beim „Legacy Ethernet“
 - CSMA/CA bei WLAN
- ❑ **Punkt-zu-Punkt Verbindungen, „geswitchte“ Netze**

Zwei Arten von „Links“

❑ **Punkt-zu-Punkt**

- 2 kommunizierenden Nodes haben einen **eigenen dedizierten** Link für jede Richtung.
- Beispiel: Ethernet LAN, das nur Switches verwendet und PPP für SONET und DSL

❑ **Mehrpunktverbindungen**

- > 2 kommunizierende Nodes **teilen** sich einen Link.
- WLAN 802.11, Bluetooth 802.15, Klassisches Ethernet (Hub), PowerLine

❑ **Autonegotiation:** Ethernet Host erkennt, ob andere Hosts am Medium sind

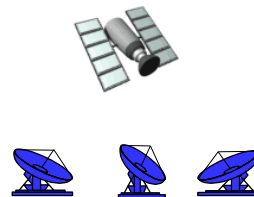
- Weitere Infos (hier nicht behandelt): <https://de.wikipedia.org/wiki/Autonegotiation>



Geteilte Leitung
“Legacy Ethernet”



Geteiltes HF
802.11



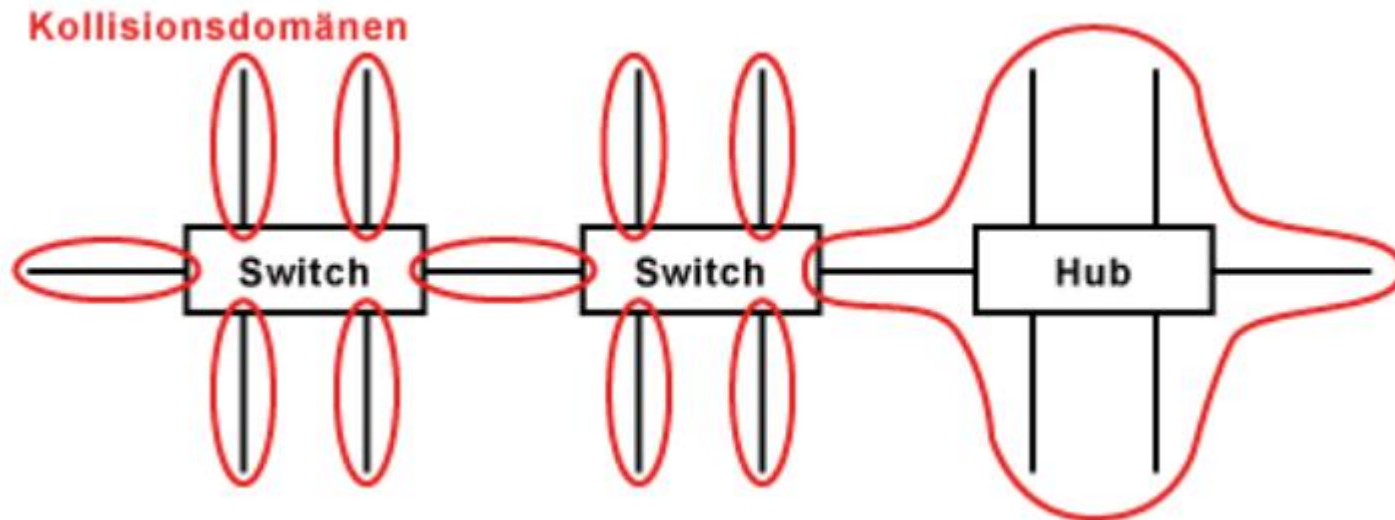
Geteiltes HF
(Satellit)



Menschen auf einer Party
(geteilte Musik)

Switched Ethernet

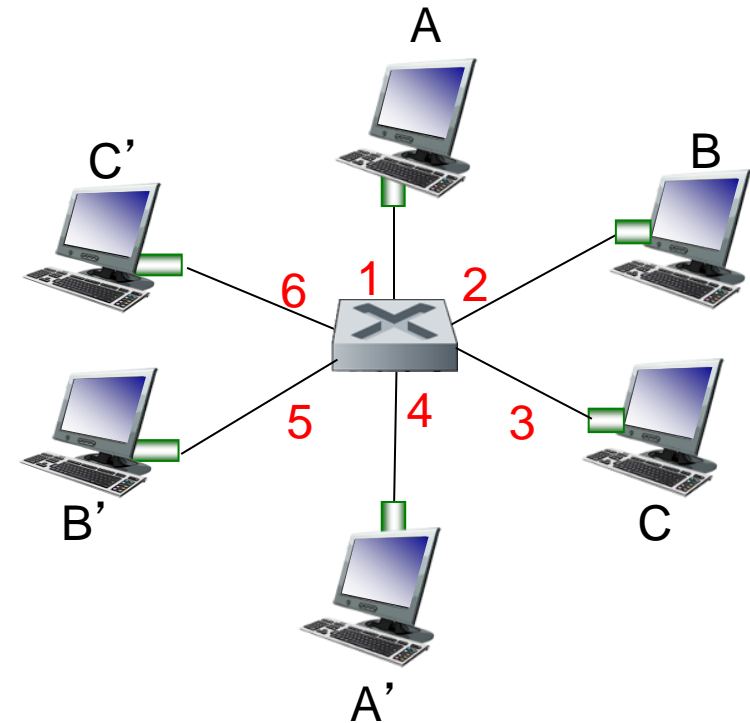
- ❑ **Hub:** Alle Leitungen sind quasi miteinander verbunden
 - Eine einzige **Kollisionsdomäne** (== Bereich, in dem nur 1 Host gleichzeitig sprechen darf)
 - Es muss CSMA/CD verwendet werden.
- ❑ **Switch:** Isoliert jeden Port in eine eigene Kollisionsdomäne
 - Bei Vollduplex-Kabeln: Kein CSMA/CD nötig!
- ❑ **Hinweis:** Nicht verwechseln mit **Broadcastdomäne**
 - == Reichweite eines Ethernet Broadcast-Frames (FF:FF:FF:FF:FF:FF)



Quelle:
<https://www.elektronik-kompodium.de/sites/net/1406181.htm>

Switched Ethernet: Das „moderne“ Ethernet

- ❑ Jeder Host direkt mit Switch-Port verbunden.
 - Jedes Kabel ist ein Punkt-zu-Punkt Netz
 - Keine Kollisionen möglich, falls Vollduplex.
 - Kein CSMA/CD nötig
- ❑ Aufgaben von Switches
 - Zwischenspeichern von Frames
 - Weiterleiten von Frames
- ❑ Hinweis. Gleichzeitige Übertragung von A zu A' und B zu B' möglich



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

Ethernet Switch

❑ **Arbeitet auf Link Layer**

- Empfang, Zwischenspeicherung und Weiterleitung von Ethernet Frames
- Untersucht MAC Adresse der ankommenden Frames und leitet Frame selektiv nur an "richtigen" Port weiter.
- Klassischer Link-Layer Switch hat keine IP Adresse!

❑ **Transparenz**

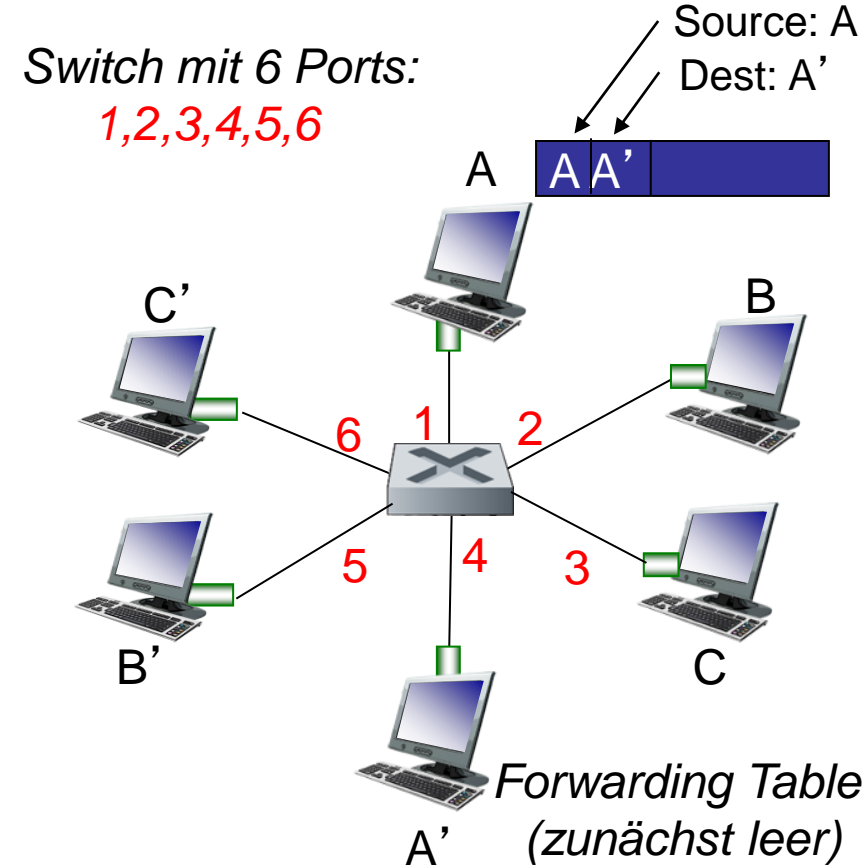
- Ethernet Hosts merken nichts von Anwesenheit eines Switches

❑ **Plug-and-Play**

- Selbstlernend
- Switches müssen nicht konfiguriert werden

Switch: Forwarding

- Zu welchen Ports muss Frame weitergeleitet werden?
 - Nachschlagen in Forwardingtabelle
- Einträge der **Forwardingtabelle**:
 - **MAC** des Zielhosts
 - **Port** des Zielhosts
 - **Time-to-Live (TTL)**: nach bestimmter Zeit wird Eintrag gelöscht
- Switches sind **selbstlernend**
 - Jeder empfangene Frame wird untersucht und für den Aufbau der Forwardingtabelle verwendet.
 - **Ankommender** Frame: Eintragen von Port und MAC des **Senders**



MAC Adr	Port	TTL
A	1	60

Switch: Lernalgorithmus und Forwarding

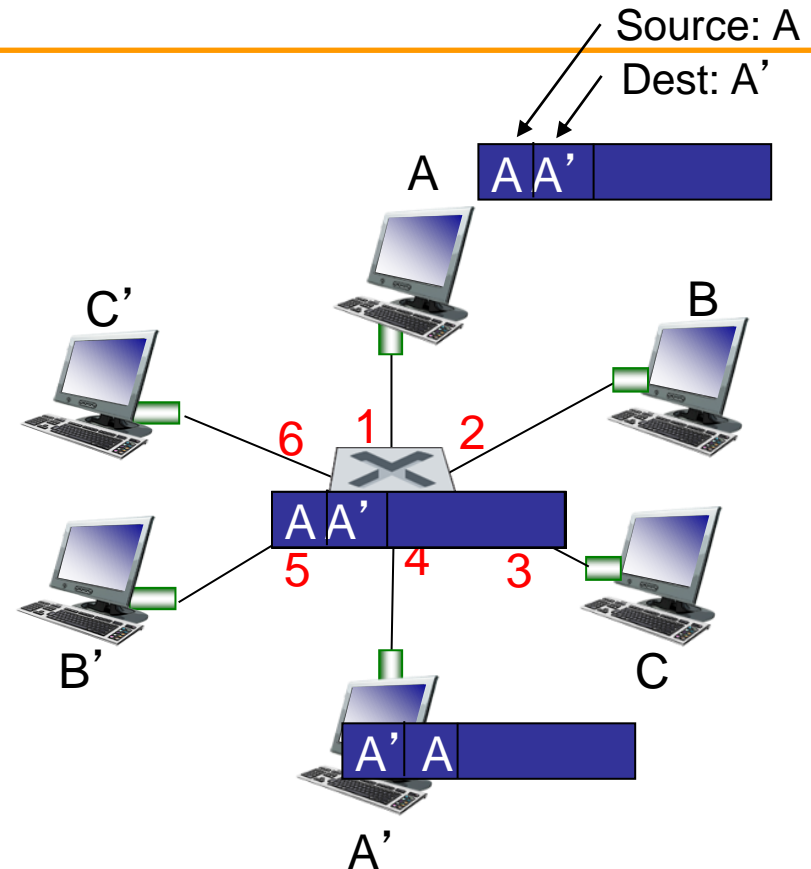
Bei Empfang eines Frames

- ❑ Switch merkt sich Eingangsport und MAC Adresse des **Senders**
 - Eintrag in Switch Forwarding Tabelle
- ❑ Nachschlagen ob Eintrag für **MAC Zieladresse** bereits in Forwarding Tabelle:
 - **Falls** Eintrag vorhanden: Ermitteln des Zielports
 - **Falls** Zielport == Quellport: Frame verwerfen
 - **Sonst:** Leite Frame an entsprechenden Zielport weiter
- ❑ **Sonst:** Fluten
 - Weiterleiten an alle Hosts mit Ausnahme des Senders.

Switch: Forwarding Beispiel

Fluten

- ❑ Zielpport A' unbekannt
 - Fluten
- ❑ Zielpport A bekannt:
 - Leite Frame nur an entsprechenden Port weiter



MAC addr	interface	TTL
A	1	60
A'	4	60

*Forwarding Tabelle
(zunächst leer)*

Publikums-Joker: Switches (Single Choice)

Ein Host sendet an einen Switch mit $n > 1$ Ports einen korrekten, sinnvollen Ethernet Frame. Welche der folgenden Fälle ist **nicht möglich**?
(Annahme: Keinerlei Firewalls!)

- A. Er leitet Frame an **keinen** Port weiter.
- B. Er leitet Frame an **1** Port weiter.
- C. Er leitet Frame an **$n-1$** Ports weiter.
- D. Er leitet Frame an **alle Ports** weiter.



- ❑ Einführung
 - Network Interface Cards (NIC)
- ❑ Rahmenbildung
 - Byte Count, Byte Stuffing, Bit Stuffing
- ❑ Fehlererkennung und Fehlerkorrektur
 - Parität, Checksumme, CRC
- ❑ Ethernet 802.3
 - Frameformat, MAC Adressen
- ❑ Broadcast Networks: Problem des Vielfachzugriffs
 - ALOHA, CSMA, CSMA/CD, CSMA/CA, TokenRing
- ❑ Switched Networks (dt. vermittelte Netze)
 - Hub vs. Switch, Forwarding, Lernalgorithmus