

Folie 1

Die **physikalische Schicht** ist für die "Drecksarbeit" zuständig: Sie bewegt rohe Bits als physikalische Signale von A nach B und kämpft dabei mit den Gesetzen der Physik (Rauschen, Dämpfung).

Teil 1: Die Physikalische Schicht – Signale und Kodierung (PDF 2)

Dieses Dokument erklärt, wie digitale Daten (Bits) in physikalische Signale umgewandelt und über ein Medium übertragen werden.

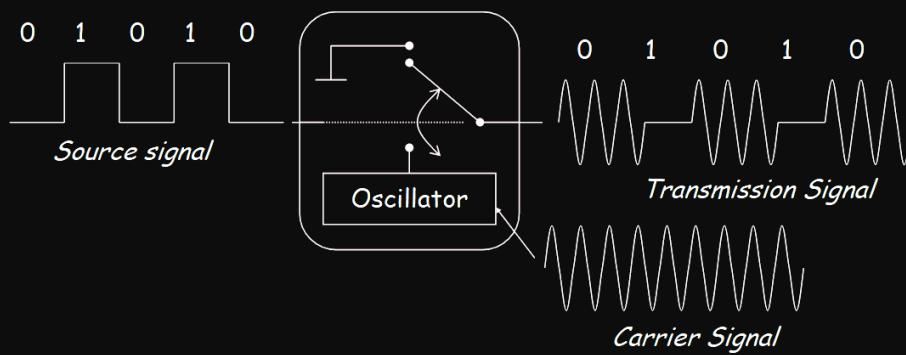
Hauptaufgaben der Physikalischen Schicht:

1. **Kodierung und Dekodierung:** Umwandlung von digitalen Daten (0en und 1en) in physikalische Signale (z. B. elektrische Spannung, Licht oder Funkwellen) und umgekehrt.
2. **Signalübertragung:** Das tatsächliche Senden der Signale über ein Übertragungsmedium wie Kupferkabel, Glasfaser oder die Luft.

Kernkonzepte:

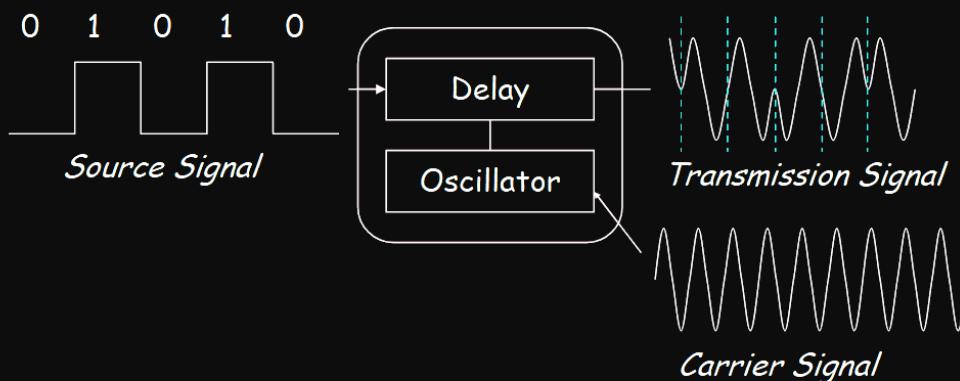
- **Modulation (Digital-zu-Analog-Wandlung):** Dies ist der Prozess, bei dem digitale Informationen auf ein analoges Trägersignal aufmoduliert werden. Dies ist für Funkübertragungen (wie WLAN oder Mobilfunk) unerlässlich. Die drei grundlegenden Techniken sind:
 - **Amplitudenmodulation (ASK - Amplitude Shift Keying):** Die Stärke (Amplitude) des Signals wird verändert, um 0en und 1en darzustellen. Diese Methode ist einfach, aber anfällig für Störungen.
 - **Frequenzmodulation (FSK - Frequency Shift Keying):** Die Frequenz des Signals wird geändert. Eine Frequenz steht für eine 0, eine andere für eine 1. Diese Methode ist robuster, benötigt aber mehr Bandbreite.
 - **Phasenmodulation (PSK - Phase Shift Keying):** Die Phase des Signals wird verschoben, um Daten zu kodieren. Sie ist sehr robust gegen Störungen.
- **Digitalisierung von analogen Signalen (Analog-zu-Digital-Wandlung):** Um analoge Signale (wie Sprache) digital zu übertragen, werden sie mittels **Puls-Code-Modulation (PCM)** umgewandelt:
 - **Abtastung (Sampling):** Das Signal wird in regelmäßigen Abständen gemessen. Das **Shannon-Theorem** besagt, dass die Abtastfrequenz mindestens doppelt so hoch sein muss wie die höchste Frequenz des Signals, um Informationsverluste zu vermeiden.
 - **Quantisierung (Quantification):** Der gemessene Wert wird einem vordefinierten, diskreten Wert zugeordnet. Dies führt zu einem kleinen Genauigkeitsverlust (Quantisierungsrauschen).
- **Herausforderungen bei der Übertragung:** Physikalische Signale sind anfällig für Störungen, die zu Übertragungsfehlern führen:
 - **Rauschen:** Zufällige Störsignale überlagern das Nutzsignal.
 - **Dämpfung:** Das Signal wird über die Distanz schwächer.
 - **Interferenz:** Störungen durch andere Funksignale (z. B. WLAN und Bluetooth).
 - **Mehrwegeausbreitung:** Das Signal erreicht den Empfänger über mehrere Pfade, was zu Echos führt.

Amplitude Shift Keying



- ❖ Modification of carrier signal's amplitude
 - Simple Method: On-Off-Keying
- ❖ Application
 - DCF77 Signal for radio clocks
 - Synchronisation: periodically reduce amplitude to 25%

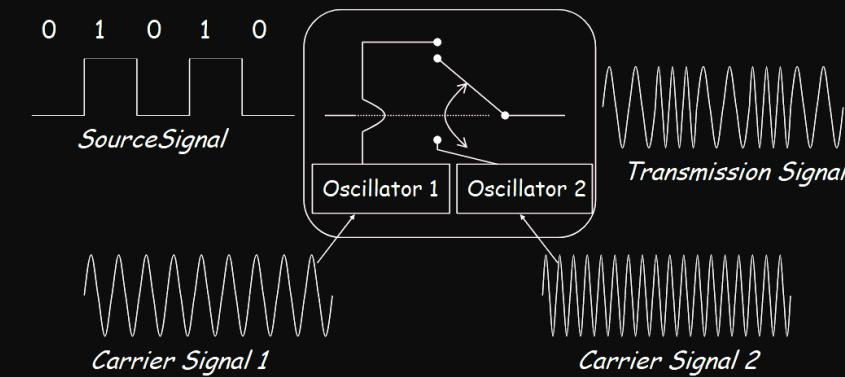
Phase Shift Keying



- ❖ Coding via Phase Shift

- Transmission of „0“ leads to $\lambda/2$ phase shift

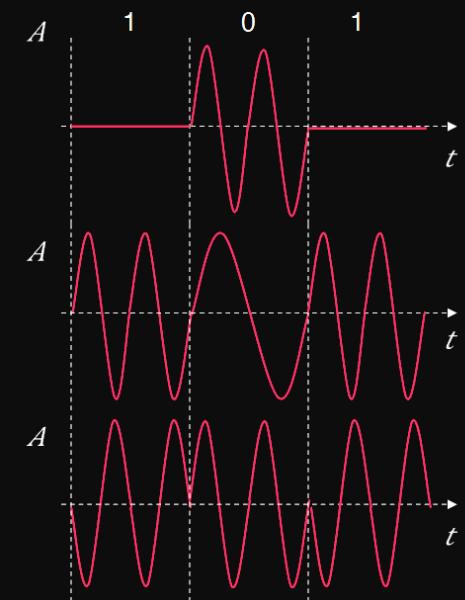
Frequency Shift Keying



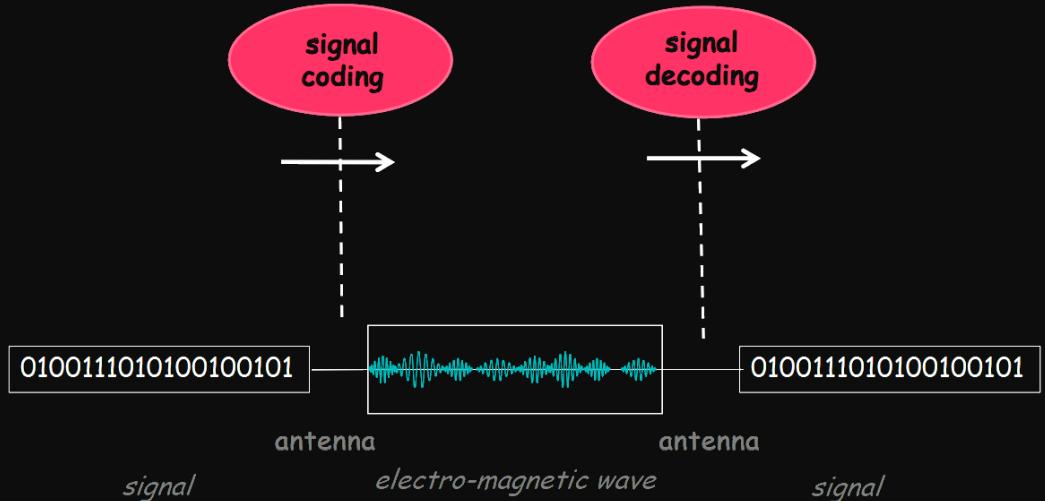
- ❖ Switching between different carrier signals
 - Binary Frequency Shift Keying (BFSK) - two carrier signals with different frequencies
- ❖ Application
 - Wireless telegraphy (oldest app) and general telecommunication

Summary

- ❖ **Amplitude Shift Keying**
 - technically simple
 - requires small bandwidth
 - susceptible for interference
- ❖ **Frequency Shift Keying**
 - higher bandwidth
- ❖ **Phase Shift Keying**
 - complex demodulation
 - robust against interference



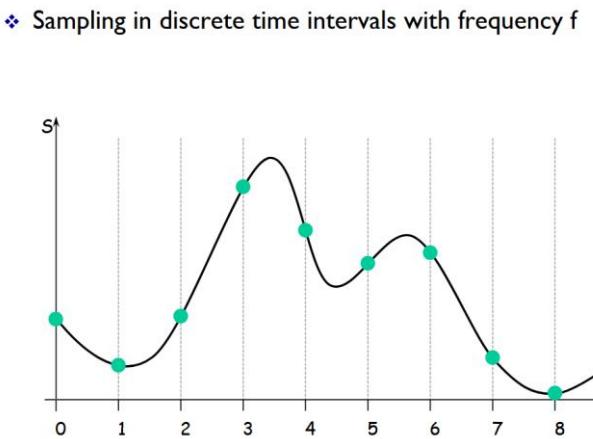
Transmission of digital data over a physical medium



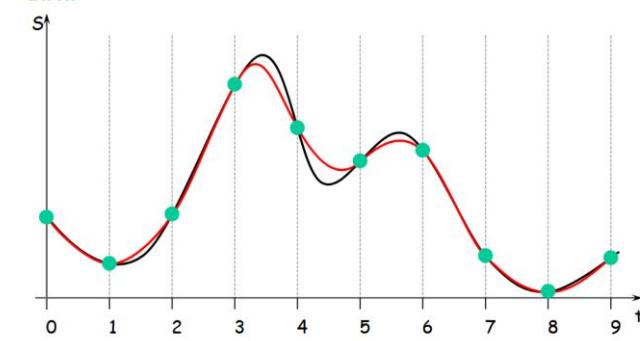
From signals to digital data

- ❖ Pulse Code Modulation (PCM)
 - Sampling → time discrete signal
 - Quantification → time and value discrete signal

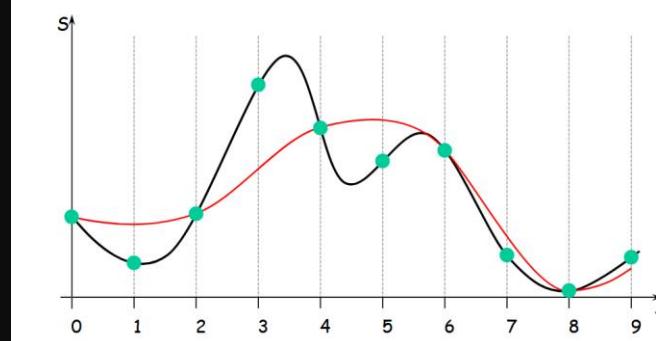
- ❖ Theorem of Shannon and Raabe
 - Sampling frequency f must be at least twice the signal's frequency
 - In ISDN: voice signals are understandable if sampled between 0 and 4500Hz → with some safety margin
ISDN samples with 9600 Hz



- ❖ Different signals might result in the same digital data

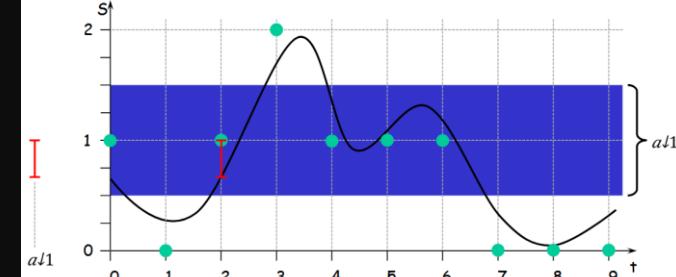


- ❖ If f does not adhere to Shannon's theorem we lose information



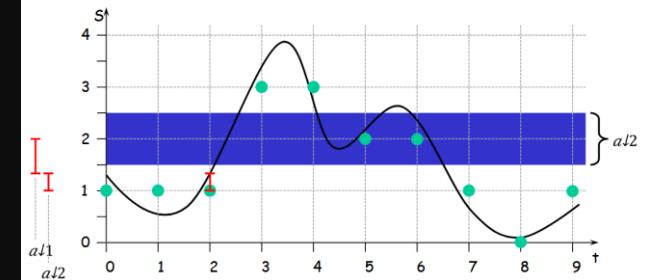
Quantification

- ❖ Mapping the continuous value space of the signal to a finite set of predefined values
 - All values of an interval are mapped to value representing this interval
 - Error margin is half the interval size



Quantification

- ❖ Trade-off: Error margin vs. size of target number set



Summary

- ❖ Achievement: We mapped an analogous signal which is time and value continuous to a time-discrete signal of pre-defined values

- ➔ A digital representation of the analogous signal
 - Loss of signal quality to meet channel limits

Eine einfache Analogie: Körpergröße messen



Stellen Sie sich vor, Sie müssen die Körpergröße von verschiedenen Personen messen.

1. Die "analoge" Realität:

In der echten Welt ist Körpergröße ein **kontinuierlicher** Wert. Jemand könnte 180,12345... cm groß sein. Es gibt unendlich viele mögliche Zwischenwerte. Das ist wie ein **analoges Signal**.

2. Das "digitale" Problem:

Sie haben aber nur ein einfaches Maßband, das nur **ganze Zentimeter** anzeigt (178, 179, 180, 181 cm usw.). Sie können die exakten Millimeter oder noch kleinere Einheiten nicht ablesen oder notieren. Ihr Maßband hat nur eine begrenzte Anzahl von vordefinierten Werten.

3. Der Schritt der Quantisierung:

Wenn Sie nun die Person messen, die 180,123... cm groß ist, müssen Sie eine Entscheidung treffen. Sie schauen auf Ihr Maßband und sagen: "Okay, das ist am nächsten an **180 cm** dran."

- **Was Sie gerade getan haben, ist Quantisierung!**
- Sie haben den exakten, kontinuierlichen Wert (180,123... cm) genommen und ihn dem nächstgelegenen, vordefinierten Wert (180 cm) aus Ihrem begrenzten Satz an Möglichkeiten zugeordnet.

Jede Person, die zwischen 179,5 cm und 180,49... cm groß ist, wird von Ihnen in die "Schublade" **180 cm** einsortiert.

Übertragung auf die Technik und die Folien

Genau das passiert bei der Umwandlung eines analogen Signals in ein digitales Signal:

- **Das analoge Signal:** Auf den Folien (z.B. Seite 19 im zweiten PDF) ist das die ursprüngliche, geschwungene schwarze Linie. Sie kann unendlich viele verschiedene Höhen (Spannungswerte) annehmen.
- **Die Quantisierung:** Man legt vorher eine feste Anzahl von "Stufen" oder "Werte-Schubladen" fest (z.B. -2V, -1V, +1V, +2V). Jeder gemessene Wert des analogen Signals wird nun auf die nächstgelegene Stufe **gerundet**.

Auf der Folie sehen Sie das sehr gut:

Der ganze Bereich, der von der **blauen Fläche** abgedeckt wird, wird auf einen einzigen digitalen Wert abgebildet (in diesem Fall der Wert, der a/1 entspricht). Egal, ob das Signal am oberen oder unteren Rand der blauen Fläche ist, es wird immer als derselbe Wert interpretiert.

1. Der Nachteil: Der "Quantisierungsfehler"

Wenn Sie die Körpergröße von 180,123... cm auf 180 cm runden, geht ein kleines bisschen Information verloren. Diese winzige Differenz zwischen dem echten Wert und dem gerundeten Wert nennt man **Quantisierungsfehler** (auf der Folie "Error margin"). Er ist unvermeidbar.

2. Der Vorteil und der Kompromiss

Warum tun wir das, wenn es ungenau ist? Weil wir Computer haben!

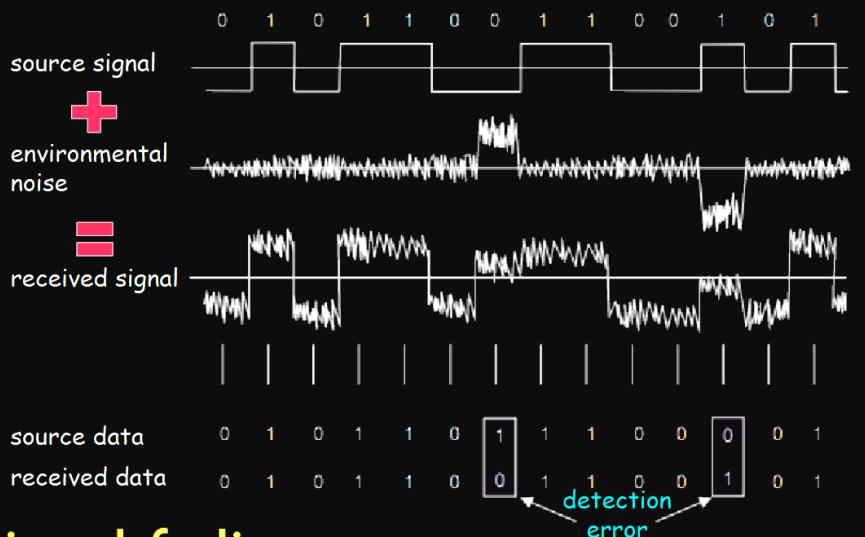
Computer können nicht mit unendlich vielen Werten umgehen. Sie arbeiten mit Bits (0 und 1).

- Wenn wir nur **2** Stufen (z.B. "hoch" und "tief") hätten, bräuchten wir nur **1 Bit**, um den Wert zu speichern (0 oder 1).
- Wenn wir **4** Stufen hätten, bräuchten wir **2 Bits** (00, 01, 10, 11).
- Je mehr Stufen wir haben, desto **genauer** wird die Abbildung (der Fehler wird kleiner), aber desto **mehr Bits** brauchen wir, um den Wert zu speichern. Die Datenmenge wächst.

Quantisierung ist also immer ein Kompromiss zwischen Genauigkeit und der Datenmenge, die man erzeugen möchte. Für eine CD-Aufnahme wählt man viel mehr Stufen (hohe Genauigkeit) als für ein einfaches Telefongespräch.

Challenges

Environmental noise

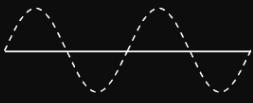


Signal fading

- ❖ Signal amplitude fades quadratic with distance for undirected senders



Original Signal



Signal at receiver 1



Signal at receiver 2

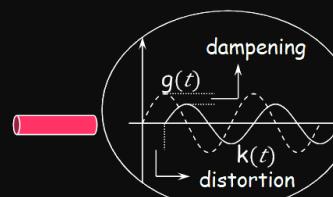
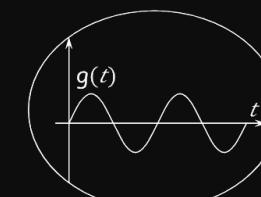
- Distance from sender to receiver 2 is twice the distance from sender to receiver 1
- The signal strength received at receiver 2 is $\frac{1}{4}$ of that received at receiver 1!

Environmental challenges on signal transmission

- Signal Dampening



- Signal distortion



More challenge sources

- Interference from other signals
 - Bluetooth on WiFi
 - WiFi on cellular
 - Current on Ethernet
- Echo, multi path fading
- Clock drift
- Hum signals (low frequency signals)
- Peak impulses (short but high amplitude)
- ...

Folie 2

Teil 2: Die Link-Schicht und LANs (PDF 1)

Dieses Dokument beschreibt, wie die Link-Schicht auf der unzuverlässigen physikalischen Übertragung aufbaut, um eine strukturierte und zuverlässigere Kommunikation zwischen direkt verbundenen Knoten in einem lokalen Netzwerk (LAN) zu schaffen.

Die **Link-Schicht** baut darauf auf und schafft Ordnung:

Hauptaufgaben der Link-Schicht:

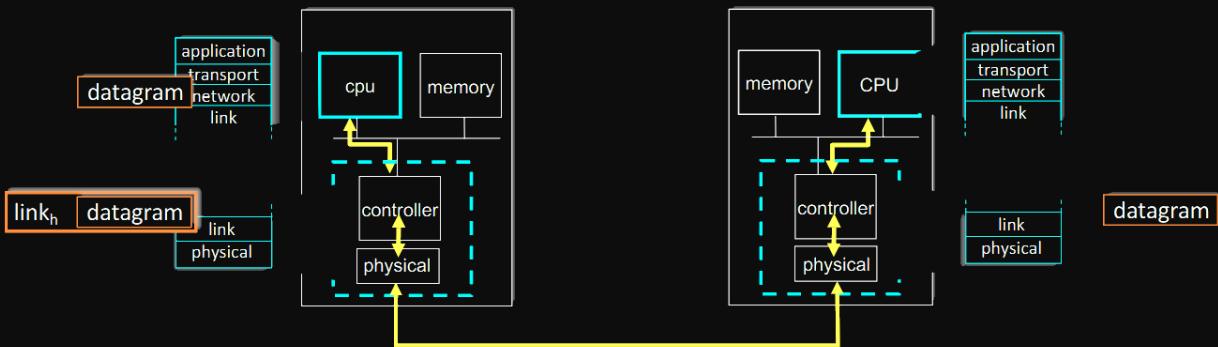
1. **Framing:** Gruppierung der von der physikalischen Schicht kommenden Bits in Datenpakete, die als **Frames** bezeichnet werden.
2. **Adressierung:** Sicherstellen, dass Frames an den richtigen Empfänger im selben lokalen Netzwerk gesendet werden.
3. **Fehlererkennung:** Überprüfen, ob die Daten während der Übertragung beschädigt wurden.
4. **Zugriffskontrolle:** Regeln, wer das Übertragungsmedium wann nutzen darf, um Kollisionen zu vermeiden oder zu behandeln.

- Sie strukturiert den Bitstrom in **Frames**.
- Sie verwendet **MAC-Adressen**, um Frames an den richtigen lokalen Empfänger zu leiten.
- Sie nutzt **Fehlererkennung (CRC)**, um Übertragungsfehler aufzudecken.
- Sie organisiert mit **MAC-Protokollen (CSMA/CD)** den Zugriff auf das Medium.

Kernkonzepte:

- **Adressierung mit MAC-Adressen:**
 - Jede Netzwerkkarte (NIC) hat eine weltweit eindeutige, 48-Bit lange **MAC-Adresse** (Media Access Control-Adresse), die fest in die Hardware eingebrannt ist.
 - Diese Adresse wird verwendet, um Frames innerhalb eines lokalen Netzwerks (z. B. vom PC zum Router) zu versenden.
- **Fehlererkennung:**
 - Um Fehler zu erkennen, die auf der physikalischen Schicht entstanden sind, werden dem Frame zusätzliche Prüfbits hinzugefügt.
 - **Paritätsprüfung:** Eine einfache Methode, die einzelne Bitfehler erkennen kann.
 - **Cyclic Redundancy Check (CRC):** Ein sehr leistungsfähiges und weit verbreitetes Verfahren (z. B. in Ethernet und WLAN), das auch Fehlerbündel (Burst Errors) zuverlässig erkennt.
- **Mediumzugriffskontrolle (Multiple Access Control - MAC):**
 - Wenn sich mehrere Geräte ein Medium teilen (z. B. bei WLAN), regeln MAC-Protokolle den Zugriff.
 - **Kanalaufteilung (z. B. TDMA, FDMA):** Jedem Gerät wird ein fester Zeit- oder Frequenzschlitz zugewiesen. Effizient bei hoher Last, aber verschwenderisch bei niedriger Last.
 - **Abwechselnde Nutzung (z. B. Polling, Token Passing):** Geräte wechseln sich kontrolliert ab.
 - **Zufallszugriff (Random Access):** Geräte senden, wenn sie Daten haben. Dies kann zu Kollisionen führen.
 - **CSMA/CD (Carrier Sense Multiple Access with Collision Detection):** Wird in kabelgebundenen Ethernet-Netzwerken verwendet. Ein Gerät hört erst, ob der Kanal frei ist. Wenn es während des Sendens eine Kollision feststellt, stoppt es, wartet eine zufällige Zeit und versucht es erneut.
 - **CSMA/CA (Collision Avoidance):** Wird in drahtlosen Netzwerken (WLAN) verwendet, um Kollisionen von vornherein zu vermeiden, da deren Erkennung schwierig ist.

Interfaces communicating



sending side:

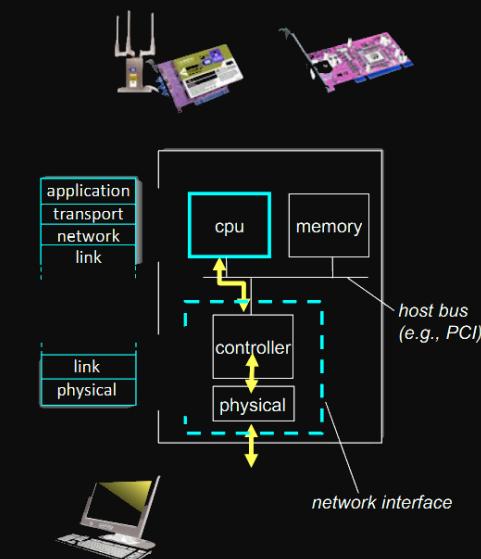
- encapsulates datagram in frame
- adds error checking bits, reliable data transfer, flow control, etc.

receiving side:

- looks for errors, reliable data transfer, flow control, etc.
- extracts datagram, passes to upper layer at receiving side

Where is the link layer implemented?

- in each-and-every host
- link layer implemented in *network interface card* (NIC) or on a chip
 - Ethernet, WiFi card or chip
 - implements link and physical layer
- attaches into host's system buses
- combination of hardware, software, firmware



Link layer: roadmap

- introduction
- addressing
- error detection, correction
- multiple access protocols

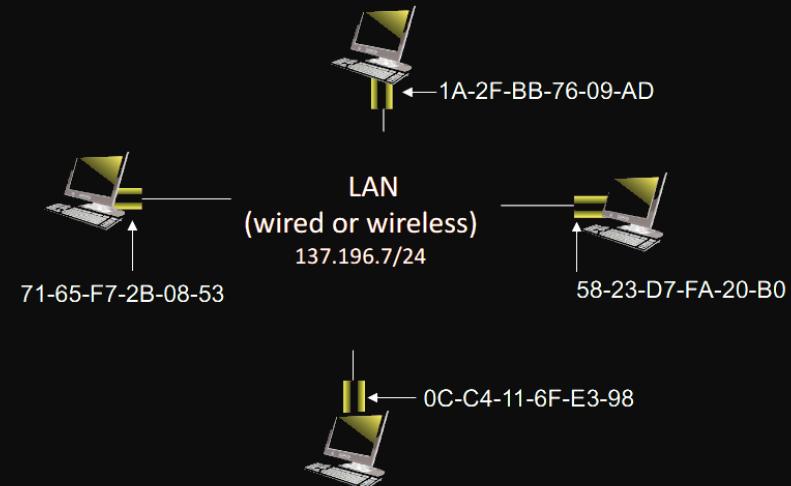
MAC addresses

- MAC (or LAN or physical or Ethernet) address:
 - function: used “locally” to get frame from one interface to another physically-connected interface
 - 48-bit MAC address (for most LANs) burned in NIC ROM, also sometimes software settable
 - e.g.: 1A-2F-BB-76-09-AD
 - hexadecimal (base 16) notation
(each “numeral” represents 4 bits)*
- MAC address allocation administered by IEEE
 - manufacturer buys portion of MAC address space (to assure uniqueness)
- MAC flat address: portability
 - can move interface from one LAN to another

MAC addresses

each interface on LAN

- has unique 48-bit MAC address



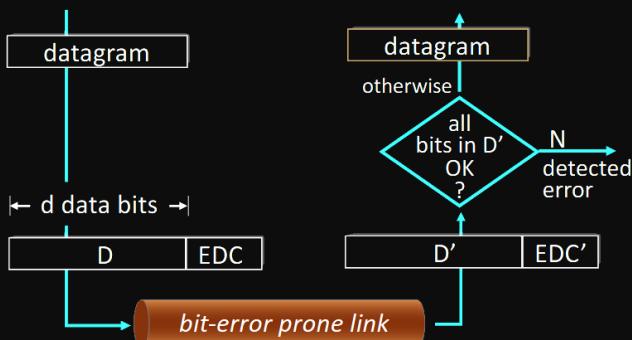
Link layer: roadmap

- introduction
- addressing
- **error detection, correction**
- multiple access protocols

Error detection

EDC: error detection and correction bits (e.g., redundancy)

D: data protected by error checking, may include header fields



Error detection not 100% reliable!

- protocol may miss some errors, but rarely
- larger EDC field yields better detection and correction

Parity checking

single bit parity:

- detect single bit errors

0111000110101011	1
← d data bits →	↑ parity bit

two-dimensional bit parity:

- detect *and correct* single bit errors

d _{1,1}	...	d _{1,j}	row parity
d _{2,1}	...	d _{2,j}	d _{2,j+1}
...
d _{i,1}	...	d _{i,j}	d _{i,j+1}
d _{i+1,1}	...	d _{i+1,j}	d _{i+1,j+1}

no errors:	1 0 1 0 1 1
	1 1 1 1 0 0
	0 1 1 1 0 1
	0 0 1 0 1 0

detected and correctable single-bit error:	1 0 1 0 1 1
	1 0 1 1 0 0
	0 1 1 1 0 1
	0 0 1 0 1 0

Check out the online interactive exercises for more practice!

Internet checksum

Goal: detect errors (i.e., flipped bits) in transmitted segment

sender:

- treat contents as sequence of 16-bit integers
- **checksum:** addition (one's complement sum) of pdu content
- checksum value put into checksum field

Cyclic Redundancy Check (CRC)

- more powerful error-detection coding
- **D:** data bits (given, think of these as a binary number)
- **G:** bit pattern (generator), of $r+1$ bits (given)



goal: choose r CRC bits, **R**, such that $\langle D, R \rangle$ exactly divisible by G (mod 2)

- receiver knows G, divides $\langle D, R \rangle$ by G. If non-zero remainder: error detected!
- can detect all burst errors less than $r+1$ bits
- widely used in practice (Ethernet, 802.11 WiFi)

receiver:

- compute checksum of received pdu
- check if computed checksum equals checksum field value:
 - not equal - error detected
 - equal - no error detected.
 - *But maybe errors nonetheless?*

Cyclic Redundancy Check (CRC): example

We want:

$$D \cdot 2^r \text{ XOR } R = nG$$

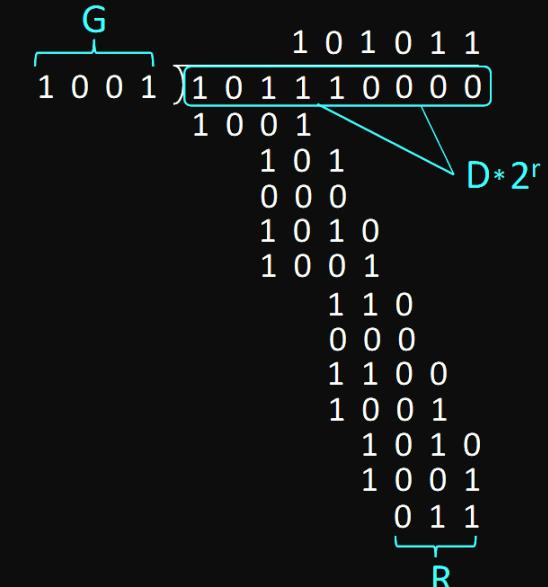
or equivalently:

$$D \cdot 2^r = nG \text{ XOR } R$$

or equivalently:

if we divide $D \cdot 2^r$ by G, want remainder R to satisfy:

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



Link layer: roadmap

- introduction
- addressing
- error detection, correction
- **multiple access protocols**

Multiple access protocols

- single shared broadcast channel
- two or more simultaneous transmissions by nodes: interference
 - *collision* if node receives two or more signals at the same time

multiple access protocol

- distributed algorithm that determines how nodes share channel, i.e., determine when node can transmit
- communication about channel sharing must use channel itself!
 - no out-of-band channel for coordination

Multiple access links, protocols

two types of “links”:

- point-to-point
 - point-to-point link between Ethernet switch, host
 - PPP for dial-up access
- **broadcast (shared wire or medium)**
 - old-fashioned Ethernet
 - upstream HFC in cable-based access network
 - 802.11 wireless LAN, 4G/4G, satellite



An ideal multiple access protocol

given: multiple access channel (MAC) of rate R bps

desiderata:

1. when one node wants to transmit, it can send at rate R .
2. when M nodes want to transmit, each can send at average rate R/M
3. fully decentralized:
 - no special node to coordinate transmissions
 - no synchronization of clocks, slots
4. simple

MAC protocols: taxonomy

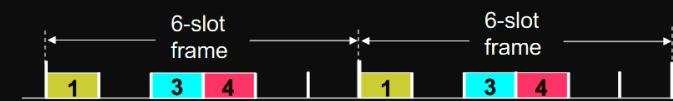
three broad classes:

- **channel partitioning**
 - divide channel into smaller “pieces” (time slots, frequency, code)
 - allocate piece to node for exclusive use
- **“taking turns”**
 - nodes take turns, but nodes with more to send can take longer turns
- **random access**
 - channel not divided, allow collisions
 - “recover” from collisions

Channel partitioning MAC protocols: TDMA

TDMA: time division multiple access

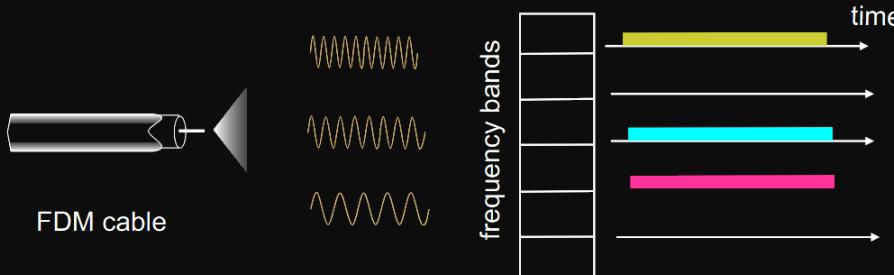
- access to channel in “rounds”
- each station gets fixed length slot (length = packet transmission time) in each round
- unused slots go idle
- example: 6-station LAN, 1,3,4 have packets to send, slots 2,5,6 idle



Channel partitioning MAC protocols: FDMA

FDMA: frequency division multiple access

- channel spectrum divided into frequency bands
- each station assigned fixed frequency band
- unused transmission time in frequency bands go idle
- example: 6-station LAN, 1,3,4 have packet to send, frequency bands 2,5,6 idle



“Taking turns” MAC protocols

channel partitioning MAC protocols:

- share channel *efficiently* and *fairly* at high load
- inefficient at low load: delay in channel access, $1/N$ bandwidth allocated even if only 1 active node!

“taking turns” protocols

- look for best of both worlds!

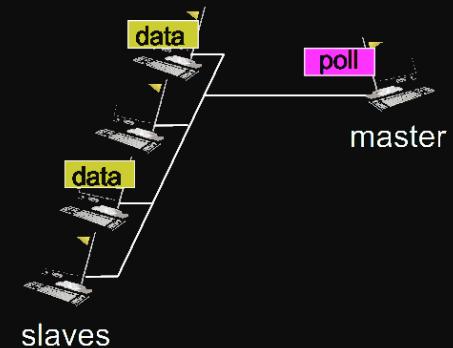
random access MAC protocols

- efficient at low load: single node can fully utilize channel
- high load: collision overhead

“Taking turns” MAC protocols

polling:

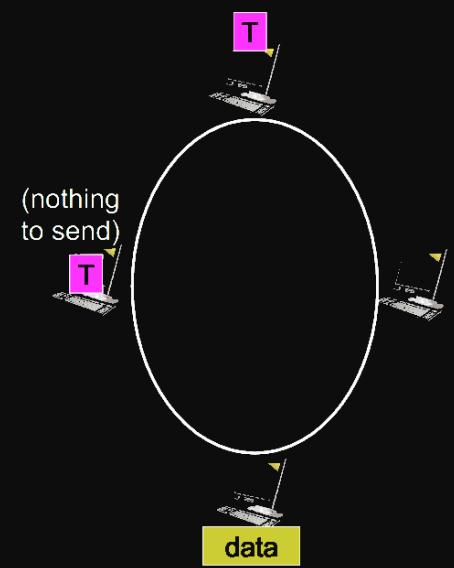
- master node “invites” other nodes to transmit in turn
- typically used with “dumb” devices
- concerns:
 - polling overhead
 - latency
 - single point of failure (master)



“Taking turns” MAC protocols

token passing:

- control *token* passed from one node to next sequentially.
- token message
- concerns:
 - token overhead
 - latency
 - single point of failure (token)



“Taking turns” MAC protocols

channel partitioning MAC protocols:

- share channel *efficiently* and *fairly* at high load
- inefficient at low load: delay in channel access, $1/N$ bandwidth allocated even if only 1 active node!

“taking turns” protocols

- look for best of both worlds!

random access MAC protocols

- efficient at low load: single node can fully utilize channel
- high load: collision overhead

Random access protocols

- when node has packet to send
 - transmit at full channel data rate R.
 - no *a priori* coordination among nodes
- two or more transmitting nodes: “collision”
- random access MAC protocol specifies:
 - how to detect collisions
 - how to recover from collisions (e.g., via delayed retransmissions)
- examples of random access MAC protocols:
 - ALOHA, slotted ALOHA
 - CSMA, CSMA/CD, CSMA/CA

Slotted ALOHA

assumptions:

- all frames same size
- time divided into equal size slots (time to transmit 1 frame)
- nodes start to transmit only slot beginning
- nodes are synchronized
- if 2 or more nodes transmit in slot, all nodes detect collision

Slotted ALOHA: efficiency

efficiency: long-run fraction of successful slots (many nodes, all with many frames to send)

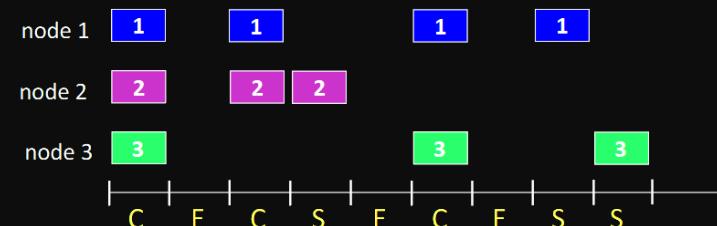
- suppose: N nodes with many frames to send, each transmits in slot with probability p
 - prob that given node has success in a slot = $p(1-p)^{N-1}$
 - prob that *any* node has a success = $Np(1-p)^{N-1}$
 - max efficiency: find p^* that maximizes $Np(1-p)^{N-1}$
 - for many nodes, take limit of $Np^*(1-p^*)^{N-1}$ as N goes to infinity, gives:
 $\text{max efficiency} = 1/e = .37$
- at best: channel used for useful transmissions 37% of time!

operation:

- when node obtains fresh frame, transmits in next slot
 - *if no collision*: node can send new frame in next slot
 - *if collision*: node retransmits frame in each subsequent slot with probability p until success

randomization – why?

Slotted ALOHA



C: collision
S: success
E: empty

Pros:

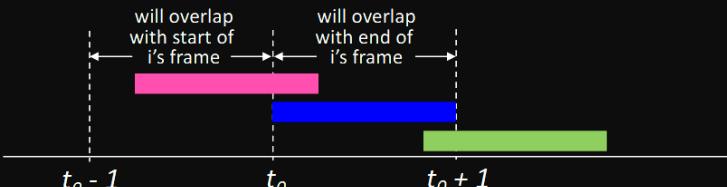
- single active node can continuously transmit at full rate of channel
- highly decentralized: only slots in nodes need to be in sync
- simple

Cons:

- collisions, wasting slots
- idle slots
- nodes may be able to detect collision in less than time to transmit packet
- clock synchronization

Pure ALOHA

- unslotted Aloha: simpler, no synchronization
 - when frame first arrives: transmit immediately
- collision probability increases with no synchronization:
 - frame sent at t_0 collides with other frames sent in $[t_0-1, t_0+1]$



▪ pure Aloha efficiency: 18% !

CSMA (carrier sense multiple access) CSMA: collisions

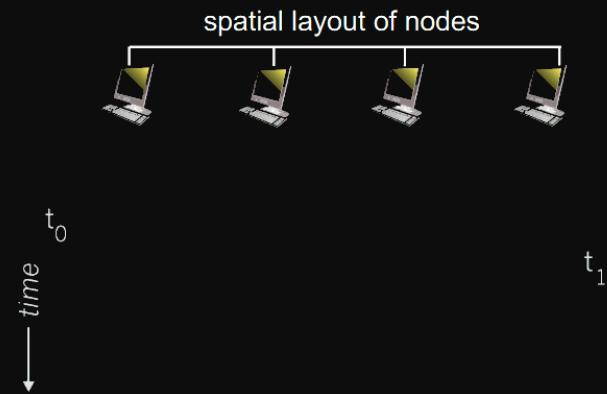
simple CSMA: listen before transmit:

- if channel sensed idle: transmit entire frame
 - if channel sensed busy: defer transmission
- human analogy: don't interrupt others!

CSMA/CD: CSMA with *collision detection*

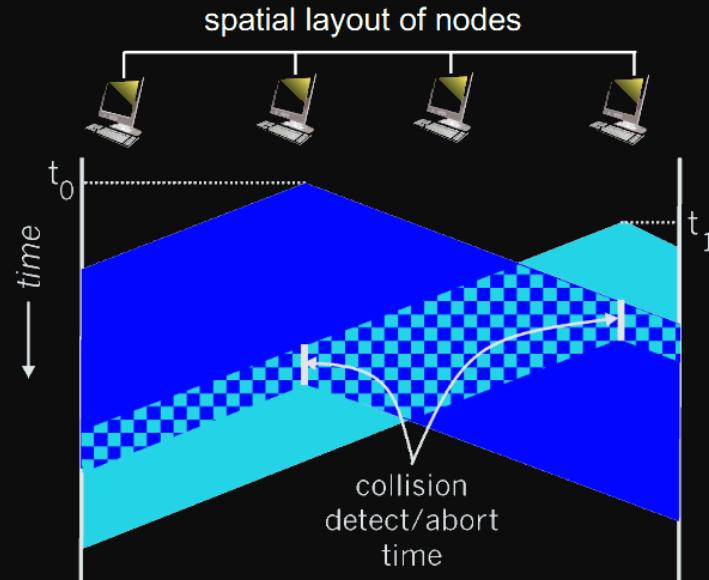
- collisions *detected* within short time
 - colliding transmissions aborted, reducing channel wastage
 - collision detection easy in wired, difficult with wireless
- human analogy: the polite conversationalist

- collisions *can* still occur with carrier sensing:
 - propagation delay means two nodes may not hear each other's just-started transmission
- collision: entire packet transmission time wasted
- distance & propagation delay play role in determining collision probability



CSMA/CD:

- CSMA/CD reduces the amount of time wasted in collisions
 - transmission aborted on collision detection



Ethernet CSMA/CD algorithm

1. NIC receives datagram from network layer, creates frame
2. If NIC senses channel:
 - if **idle**: start frame transmission.
 - if **busy**: wait until channel idle, then transmit
3. If NIC transmits entire frame without collision, NIC is done with frame !
4. If NIC detects another transmission while sending: abort, send jam signal
5. After aborting, NIC enters ***binary (exponential) backoff***:
 - after m th collision, NIC chooses K at random from $\{0,1,2, \dots, 2^m-1\}$. NIC waits $K \cdot 512$ bit times, returns to Step 2
 - more collisions: longer backoff interval

CSMA (carrier sense multiple access)

- simple **CSMA**: listen before transmit:
 - if channel sensed **idle**: transmit entire frame
 - if channel sensed **busy**: defer transmission
- human analogy: don't interrupt others!

CSMA/CD: CSMA with *collision detection*

- collisions *detected* within short time
- colliding transmissions aborted, reducing channel wastage
- collision detection easy in wired, difficult with wireless
- human analogy: the polite conversationalist

CSMA/CD efficiency

- T_{prop} = max prop delay between 2 nodes in LAN
- t_{trans} = time to transmit max-size frame

$$efficiency = \frac{1}{1 + 5t_{prop}/t_{trans}}$$

- efficiency goes to 1
 - as t_{prop} goes to 0
 - as t_{trans} goes to infinity
- better performance than ALOHA: and simple, cheap, decentralized!

1. Das Grundproblem: Warum benötigen wir Zugriffsprotokolle?

Wenn sich mehrere Geräte ein einziges Übertragungsmedium teilen (wie ein gemeinsames Kabel beim alten Ethernet oder die Luft bei WLAN), spricht man von einem **Broadcast-Medium**. Das grundlegende Problem ist: Wie koordiniert man, wer wann senden darf?

- **Kollision:** Wenn zwei oder mehr Geräte gleichzeitig senden, überlagern sich ihre Signale auf dem Medium. Der Empfänger erhält nur ein verstümmeltes, unbrauchbares Signal. Dies wird als **Kollision** bezeichnet.
- **Die Herausforderung:** Ein **Multiple Access Protocol** (oder MAC-Protokoll) ist ein verteilter Algorithmus, der bestimmt, wie die Geräte das gemeinsame Medium nutzen, um Kollisionen zu vermeiden oder zu beheben. Die Schwierigkeit besteht darin, dass die Kommunikation über die Kanalnutzung über denselben Kanal stattfinden muss, den man gerade zu koordinieren versucht.

2. Eigenschaften eines idealen MAC-Protokolls

Ein perfektes MAC-Protokoll würde folgende Eigenschaften aufweisen:

1. **Effizient bei einem Sender:** Wenn nur ein Gerät senden möchte, sollte es die volle Bandbreite des Kanals nutzen können.
2. **Fair bei vielen Sendern:** Wenn M Geräte gleichzeitig senden möchten, sollte jedes im Durchschnitt $1/M$ der Bandbreite erhalten.
3. **Vollständig dezentralisiert:** Es sollte keinen zentralen Koordinator oder "Master" geben, der ausfallen könnte.
4. **Einfach und kostengünstig** in der Implementierung.

In der Praxis gibt es kein Protokoll, das alle diese Punkte perfekt erfüllt. Es handelt sich immer um einen Kompromiss.

3. Die drei Hauptkategorien von MAC-Protokollen

Die Folien unterteilen die MAC-Protokolle in drei große Klassen:

A) Kanalaufteilung (Channel Partitioning)

Die Idee hier ist, den Kanal im Voraus in feste, nicht überlappende "Stücke" aufzuteilen und jedem Gerät ein Stück exklusiv zuzuweisen.

- **TDMA (Time Division Multiple Access - Zeitmultiplexverfahren):**

- Die Zeit wird in Runden aufgeteilt, und jede Runde besteht aus einer festen Anzahl von Zeitschlitzten.
- Jedes Gerät erhält pro Runde einen festen Zeitschlitz, in dem es exklusiv senden darf.
- **Nachteil:** Wenn ein Gerät in seinem Zeitschlitz nichts zu senden hat, bleibt der Kanal ungenutzt. Dies ist ineffizient bei "bursty" Verkehr (d. h. wenn Daten unregelmäßig anfallen).

- **FDMA (Frequency Division Multiple Access - Frequenzmultiplexverfahren):**

- Das Frequenzspektrum des Kanals wird in kleinere Frequenzbänder aufgeteilt.
- Jedes Gerät erhält ein festes Frequenzband, auf dem es exklusiv senden kann.
- **Nachteil:** Ähnlich wie bei TDMA; wenn ein Gerät nichts sendet, bleibt sein Frequenzband ungenutzt.

Fazit zur Kanalaufteilung: Diese Methoden sind fair und effizient, wenn alle Geräte kontinuierlich Daten senden (hohe Last). Sie sind jedoch sehr ineffizient bei niedriger Last, da die zugewiesenen Ressourcen ungenutzt verschwendet werden.

B) Abwechselnde Nutzung ("Taking Turns")

Diese Protokolle sind flexibler. Die Geräte wechseln sich aktiv ab, aber wer gerade an der Reihe ist, kann den vollen Kanal nutzen.

- **Polling:**

- Es gibt einen zentralen **Master-Knoten**, der die anderen **Slave-Knoten** nacheinander "fragt" (poll), ob sie Daten zu senden haben. Nur der gefragte Slave darf senden.
- **Nachteile:**
 - **Polling-Overhead:** Die Zeit, die für das Abfragen verbraucht wird.
 - **Latenz:** Ein Gerät muss warten, bis es an der Reihe ist.
 - **Single Point of Failure:** Fällt der Master aus, bricht das gesamte Netzwerk zusammen.

- **Token Passing:**

- Ein spezielles, kleines Datenpaket, der **Token**, wird in einer vordefinierten Reihenfolge (oft in einem logischen Ring) von Gerät zu Gerät weitergereicht.
- Nur das Gerät, das den Token besitzt, darf senden. Wenn es fertig ist, gibt es den Token an das nächste Gerät weiter.
- **Nachteile:**
 - **Token-Overhead:** Die Zeit für die Weitergabe des Tokens.
 - **Latenz:** Wie beim Polling.
 - **Single Point of Failure:** Geht der Token verloren oder fällt ein Gerät aus, kann das Netzwerk blockieren.

C) Zufallszugriff (Random Access)

Hier gibt es keine feste Reihenfolge oder Aufteilung. Ein Gerät sendet, wann immer es Daten hat. Diese Protokolle müssen einen Weg finden, um mit den unvermeidlichen Kollisionen umzugehen.

• ALOHA-Protokolle (Der Vorläufer):

- **Pure ALOHA:** Sende sofort, wenn Daten ankommen. Sehr einfach, aber hohe Kollisionswahrscheinlichkeit. Die maximale Effizienz liegt bei nur 18 %.
- **Slotted ALOHA:** Die Zeit ist in Zeitschlüsse unterteilt. Ein Gerät darf nur am Anfang eines Zeitschlusses senden. Dies halbiert die Kollisionswahrscheinlichkeit. Bei einer Kollision sendet das Gerät in einem der folgenden Zeitschlüsse mit einer bestimmten Wahrscheinlichkeit p erneut. Die maximale Effizienz ist doppelt so hoch wie bei Pure ALOHA, nämlich ca. 37 %.

• CSMA (Carrier Sense Multiple Access - "Hören, bevor man spricht"):

- **Grundprinzip:** Bevor ein Gerät sendet, "hört" es auf den Kanal (Carrier Sense).
 - Wenn der Kanal **frei** ist, wird gesendet.
 - Wenn der Kanal **belegt** ist, wird gewartet.
- **Problem:** Aufgrund der Signallaufzeit (Propagation Delay) können zwei weit voneinander entfernte Geräte den Kanal gleichzeitig als frei "hören" und trotzdem eine Kollision verursachen.

• CSMA/CD (CSMA with Collision Detection - "Hören, während man spricht"):

- Dies ist die entscheidende Verbesserung für **kabelgebundene Netzwerke (Ethernet)**.
- Ein Gerät hört nicht nur vor, sondern auch **während** es sendet.
- Wenn es eine Kollision entdeckt (weil das, was es auf dem Kabel "hört", nicht mit dem übereinstimmt, was es sendet), bricht es die Übertragung sofort ab.
- **Ethernet CSMA/CD Algorithmus:**
 1. Frame vorbereiten.
 2. Kanal abhören. Wenn belegt, warten. Wenn frei, senden.
 3. Während des Sendens weiterhören.
 4. Wenn **keine Kollision** auftritt, ist die Übertragung erfolgreich.
 5. Wenn eine **Kollision entdeckt** wird:
 - Senden abbrechen.
 - Ein **Jam-Signal** senden, um sicherzustellen, dass alle anderen die Kollision bemerken.
 - Eine **zufällige Zeit warten** und dann bei Schritt 2 neu beginnen.
- **Exponentieller Backoff:** Der Mechanismus für die zufällige Wartezeit. Nach der m -ten Kollision wird eine Zufallszahl K aus dem Intervall $[0, 1, \dots, 2^m - 1]$ gewählt. Die Wartezeit beträgt dann K mal eine feste Zeitspanne. Das bedeutet: Je mehr Kollisionen, desto größer wird das Intervall für die zufällige Wartezeit, was die Wahrscheinlichkeit weiterer Kollisionen reduziert.

• Effizienz von CSMA/CD:

- Die Effizienz ist deutlich höher als bei ALOHA. Sie hängt vom Verhältnis zwischen der **Signallaufzeit (T_prop)** und der **Übertragungszeit eines Frames (T_trans)** ab.
- Für eine hohe Effizienz muss die Übertragungszeit viel länger sein als die Signallaufzeit.

4. Zusammenfassung aller MAC-Protokolle (Folie 38)

- **Kanalaufteilung:**

- Nach Zeit (TDMA), Frequenz (FDMA).
 - Gut für konstante Datenströme, schlecht für bursty Traffic.

- **Abwechselnde Nutzung:**

- Polling, Token Passing (verwendet in Bluetooth, FDDI, Token Ring).
 - Guter Kompromiss, aber mit Overhead und potenziellen Single Points of Failure.

- **Zufallszugriff:**

- ALOHA, CSMA, CSMA/CD, CSMA/CA.
 - Sehr effizient bei niedriger Last und bursty Traffic.
 - **CSMA/CD** ist die Grundlage des klassischen **Ethernet**.
 - **CSMA/CA** (Collision Avoidance) wird in drahtlosen Netzwerken wie **WLAN (802.11)** verwendet, da die Kollisionserkennung in der Luft schwierig ist.

