

Exam Cheatsheet

ISO/OSI Modell:

```
[7] Anwendung
[6] Darstellung
[5] Sitzung
[4] Transport
[3] Vermittlung
[2] Sicherung
[1] Bitübertragung
```

TCP/IP Modell:

```
[4] Anwendung/Application-Layer
[3] Transport Layer
[2] Internet Layer
[1] Host-to-Network Layer
```

ISO/OSI <-> TCP/IP:

[7] Anwendung	<->	[4] Anwendung/Application-Layer
[6] Darstellung	<->	[4] Anwendung/Application-Layer
[5] Sitzung	<->	[4] Anwendung/Application-Layer
[4] Transport	<->	[3] Transport Layer
[3] Vermittlung	<->	[2] Internet Layer
[2] Sicherung	<->	[1] Host-to-Network Layer
[1] Bitübertragung	<->	[1] Host-to-Network Layer

Ethernet-Rahmen:

```
[Ethernet-Header] [IP-Header] [TCP-Header] [HTTP-Header] [Daten] [Ethernet-Tail]
```

Dienstprimitive:

```
CONNECT
DISCONNECT
DATA
ABORT
```

Dienstfunktionen:

Request

Indication

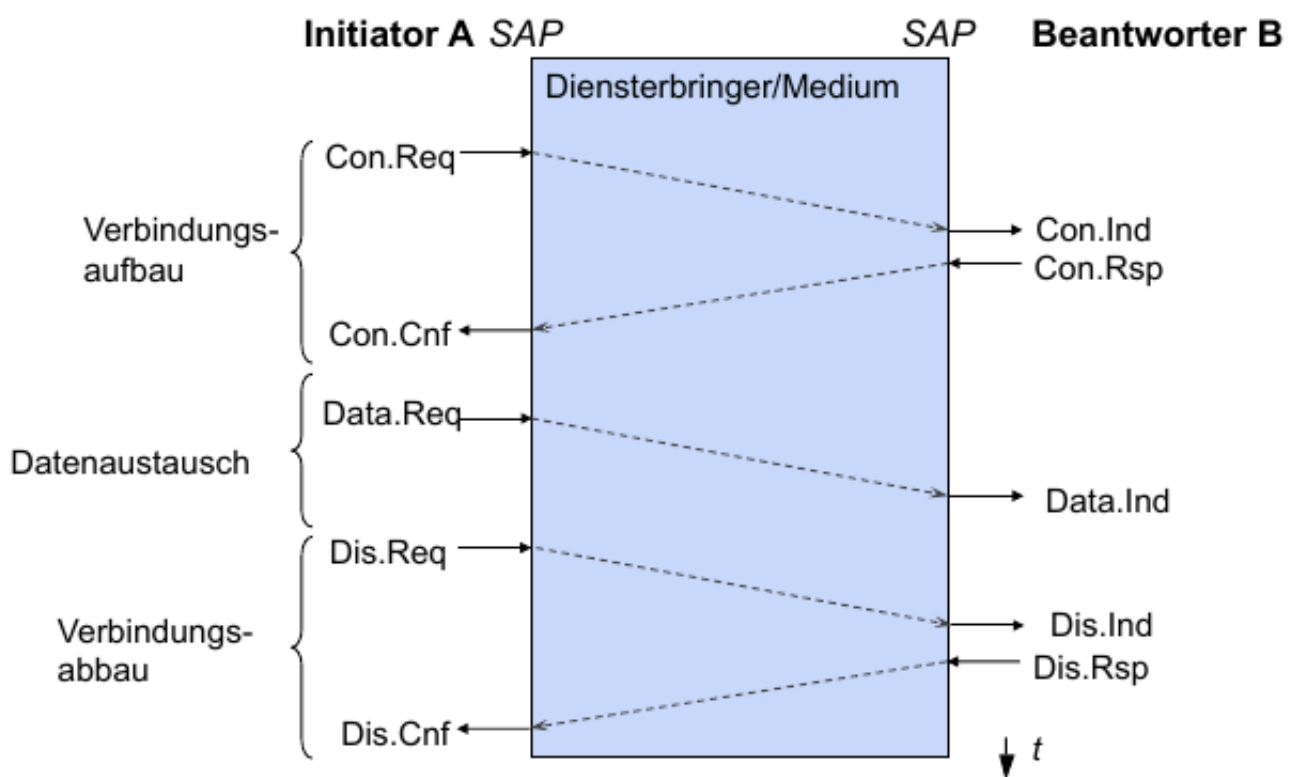
Response

Confirmation

Weg/Zeit-Diagramm:

Drei Phasen:

1. Verbindungsaufbau
2. Datenübertragung
3. Verbindungsabbau Bsp: Verbindungsorientiert Verbindungsaufbau und Verbindungsabbau sind bestätigt Datenaustausch ist unbestätigt



Zustandsübergangsdiagramm:

- 4 Zustände:
- Ruhezustand,
 - Verbindung im Aufbau,
 - Verbindung aufgebaut (Datenaustausch),
 - Verbindung im Abbau.

Shannon und Nyquist:

Datenrate Berechnung

B: Bandbreite
n: Signalstufen
S/N: Signal-Rausch-Abstand
SNR: Signal to Noise Ratio

Nyquist:

max.Datenrate = $2 * B * \lg(n)$ [Bit/s]

Shannon:

max.Datenrate = $B * \lg(1 + S/N)$ [Bit/s]

Bsp1:

Frequenzbereich von 4000Hz bis 9000Hz
mit 50dB Signal-Rauschabstand

50dB -> $S/N = 10^5$
 $B = 9000 \text{ Hz} - 4000 \text{ Hz} = 5000 \text{ Hz}$
 $\text{max.Datenrate} = 5000 \text{ Hz} * \lg(1 + 100000) \text{ Bit/s}$
 $= 83048,274507 \text{ Bit/s}$

Bsp2:

Frequenzbereich von 20kHz bis 30kHz
mit 30dB Signal-Rauschabstand

30dB -> $S/N = 10^3$
 $B = 30\text{kHz} - 20\text{kHz} = 10\text{kHz} = 10000 \text{ Hz}$
 $\text{max.Datenrate} = 10000 \text{ Hz} * \lg(1 + 1000) \text{ Bit/s}$
 $= 99672,262588 \text{ Bit/s}$

Frage:

Es soll nun 256-QAM zur Codierung der Daten verwendet werden. Ist dies für beide Kanäle problemlos möglich?

256-QAM -> $n=256$
für Berechnung mit Nyquist

Berechne dann **die** maximale Datenrate mit Nyquist
und dann vergleiche ob dies größer als Shannon ist.

Wenn **die** Datenrate beim Nyquist größer als Shannon ist,
dann ist 256-QAM für **die** Kanal nicht geeignet.

Selbsttaktung:

Selbsttaktung bezeichnet die Fähigkeit eines Leitungscodes, die Synchronisation einer Bitfolge zwischen Sender und Empfänger ohne weitere Hilfsmittel sicherzustellen.

Extra: Dazu wird in Kombination mit den zu übertragenden Daten eine Taktinformation mit in die erzeugte Signalfolge ein- gebettet, anhand derer der Empfänger den Takt auslesen und die Daten synchronisiert entnehmen kann.

Selbsttaktend:

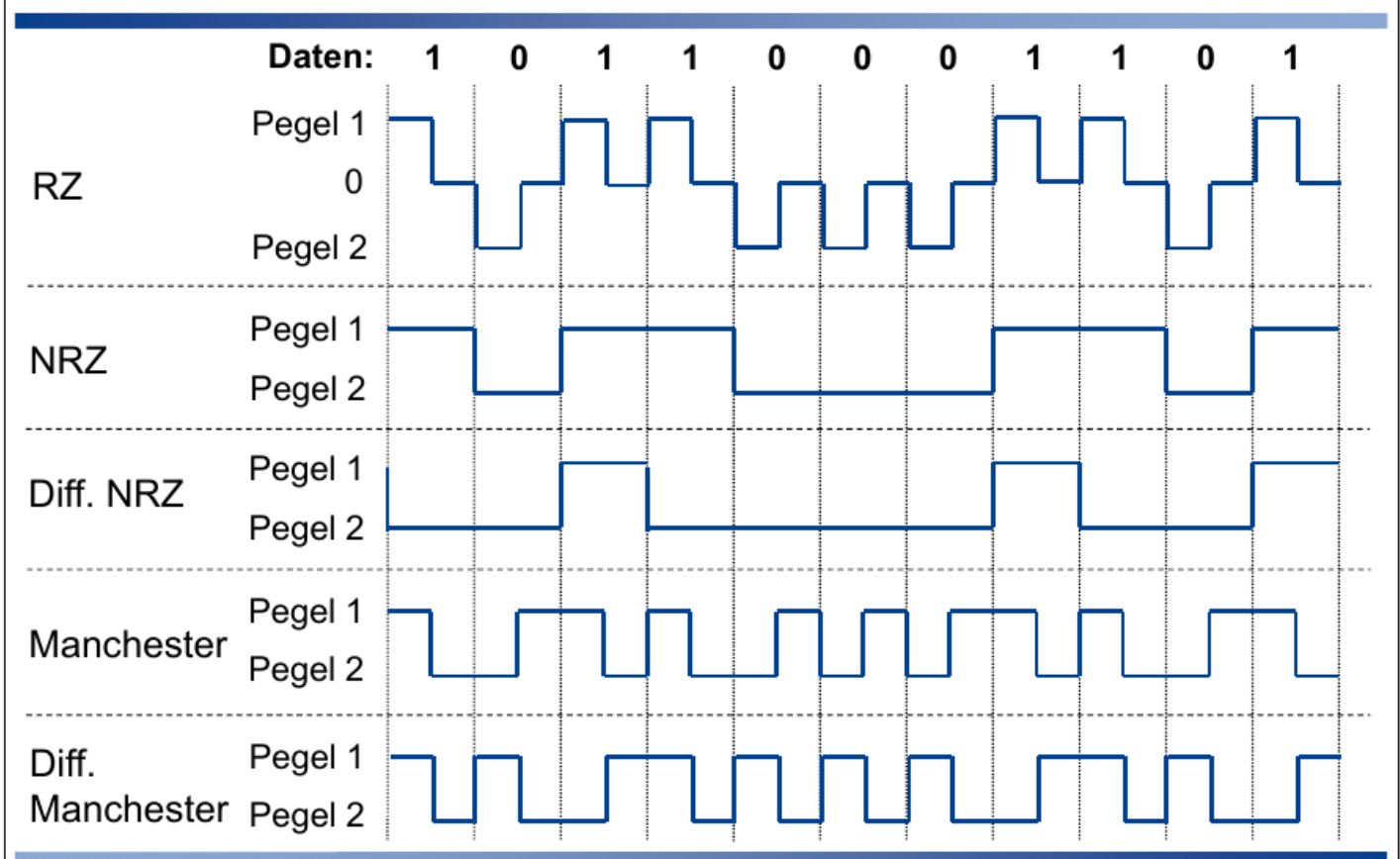
Manchester (Biphase-L),
Differential Manchester,
Biphase-M,
Biphase-S,
4B/5B (oder ähnliche Codierverfahren)

NICHT selbsttaktend:

NRZ*,
Uni-/Bipolar RZ
QAM o.ä. wäre hier auch nicht richtig,
da wir ja über Basisbandübertragung sprechen.

Codes:

Leitungscodes - Beispiele



(1) RZ:

Level: [Pegel1, 0, Pegel2]
Bit 1: Falling staircase **from** Pegel1 to 0
Bit 0: Rising staircase **from** Pegel2 to 0
AFTER: Connect signals **with** a direct line between each bit

(2) NRZ:

Level: [Pegel1, Pegel2]
Bit 1: Constant horizontal line on Pegel1
Bit 0: Constant horizontal line on Pegel2
AFTER: Connect signals with a direct line between each bit

(3) Diff. NRZ:

Level: [Pegel1, Pegel2]
Bit 1: Constant horizontal line reverse to the previous one
Bit 0: Constant horizontal line same **as** the previous one
AFTER: Connect signals **with** a direct line between each bit

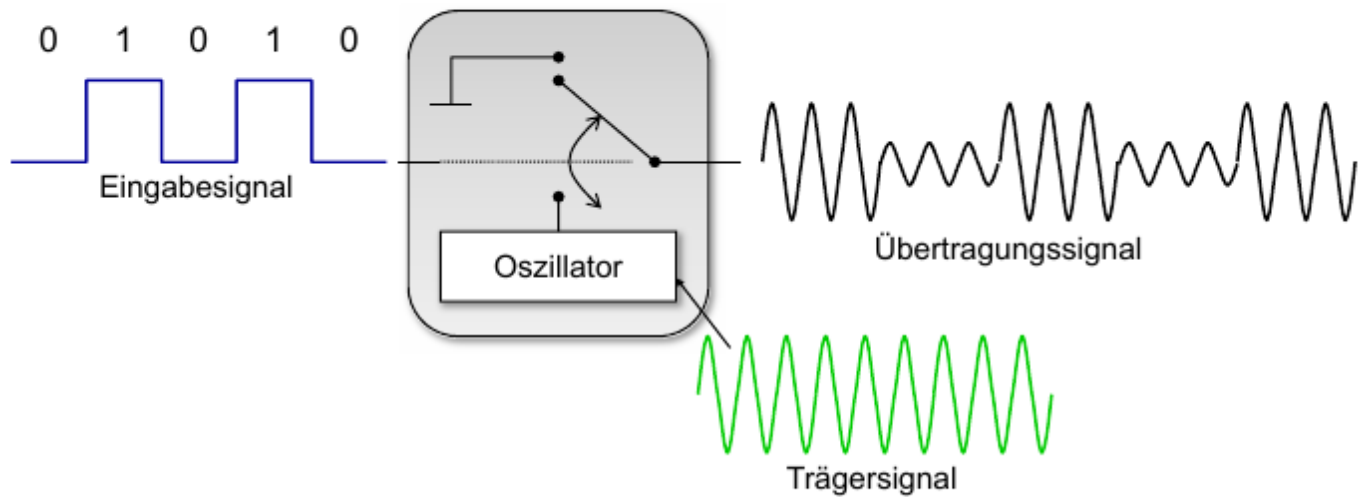
(4) Manchester:

Level: [Pegel1, Pegel2]
Bit 1: Falling staircase **from** Pegel1 to Pegel2
Bit 0: Rising staircase **from** Pegel2 to Pegel1
AFTER: Connect signals **with** a direct line between each bit

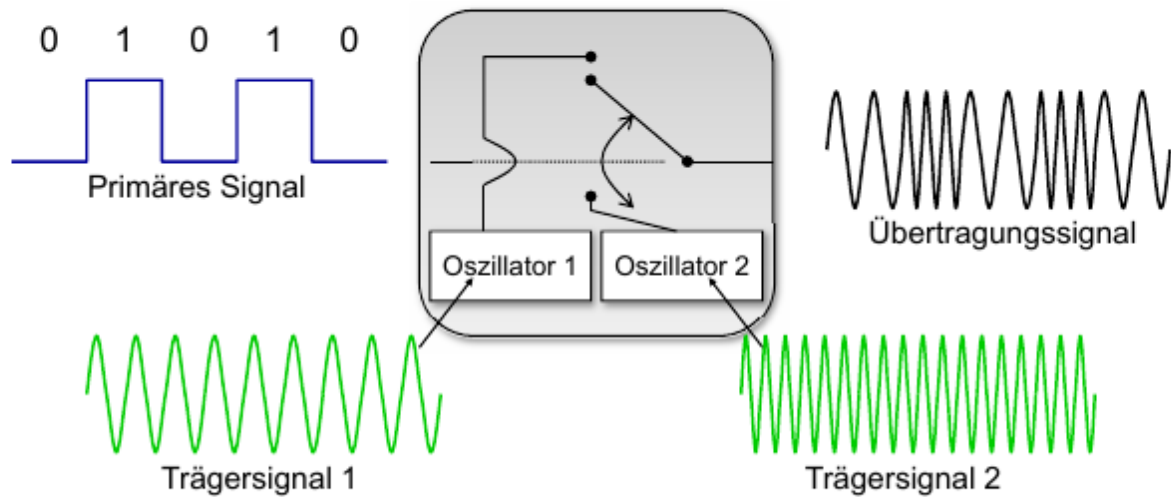
(5) Diff. Manchester:

Level: [Pegel1, Pegel2]
Bit 1: Reverse staircase **from** the previous one
Bit 0: Same staircase **as** the previous one
AFTER: Connect signals **with** a direct line between each bit

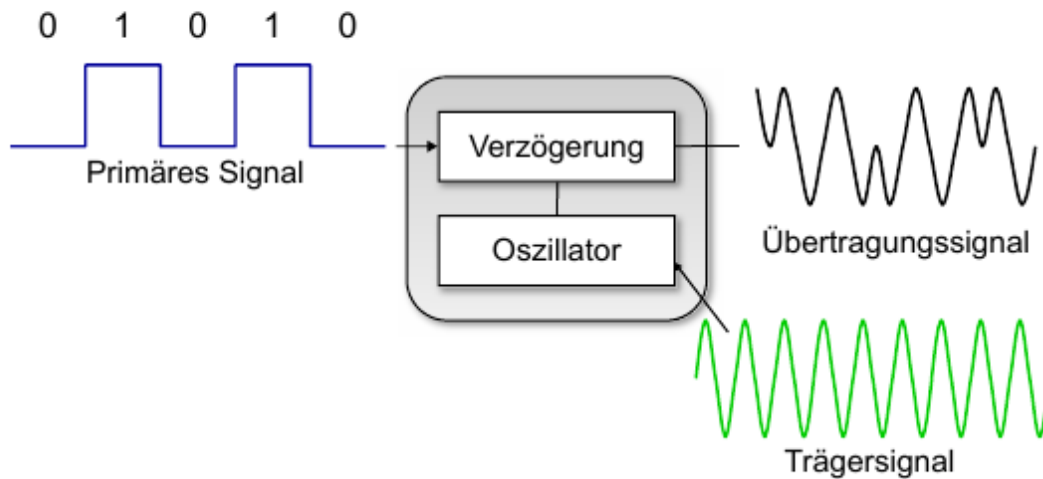
Modulationen:



- Primäres Signal wird durch *Amplitudenveränderung* auf Trägersignal *moduliert*.
- Amplitudenmodulation ist sehr störanfällig



- Primäres Signal wird durch gezielte *Änderung der Frequenz des Trägersignals* moduliert



- Primäres Signal wird mittels gezielter *Phasensprünge* des Trägersignals moduliert.
- Für Datenkommunikation bestes Verfahren

Abtasttheorem:

Die Abtastfrequenz f_A muss mindestens doppelt so hoch sein wie die höchste im abzutastenden Signal vorkommende Frequenz f_{Grenz} : $f_A \geq 2 * f_{\text{Grenz}}$

Bsp:

Frage:

Angenommen, ein analoges Stereosignal wurde mit PCM digitalisiert. Welche Datenrate ist für **die** Über- tragung der digitalen Daten mindestens erforderlich, wenn das Stereosignal ein Spektrum von 20Hz bis 20kHz umfasst und mit 16Bit quantisiert wird?

Die Grenzfrequenz ist 20kHz, also muss mit 40kHz abgetastet werden.
Pro Abtastung werden 16Bit benötigt.
Also $40.000\text{Hz} * 16\text{Bit} = 640.000 \text{ Bit/s}$.

Da wir ein Stereosignal haben, benötigen wir **die** doppelte Datenrate.
 $2 * 40.000\text{Hz} * 16\text{Bit} = 2 * 640.000\text{Bit/s} = 1.28 \text{ MBit/s}$

Cyclic Redundancy Check (CRC):

Bildung der CRC-Prüfsumme: 1. Zu prüfende Bitfolge wird als **Polynom** aufgefasst 2.

Erweiterung um **0-Folge** (Anzahl 0en = Grad des Prüfpolynoms), Teilung durch vereinbartes Prüfpolynom (**Generatorpolynom**) 3. **FCS** ist **Rest der Division**, wird an die Bitfolge angehängt 4. Beim Empfänger wird neu dividiert (einschließlich Rest), bei fehlerfreier Übertragung muss das Ergebnis 0 sein

Note: Hier nutzt man Modulo-2-Arithmetik

Dadurch entsteht eine Art Codiervorschrift: eine übertragene Bitfolge ist nur dann **gültig**, wenn sie (als Polynom interpretiert) **ein Vielfaches des Generatorpolynoms** ist. Der Empfänger teilt die gesamte empfangene Bitfolge durch das gleiche Generatorpolynom. **Bei korrektem Empfang darf bei der Division kein Rest bleiben.**

Bsp:

Zu sendender Block: 110011
Generatorpolynom: $x^4 + x^3 + 1$

Verfahren:

Addition/Subtraktion Modulo-2 entspricht bitweiser XOR-Verknüpfung

Dividend ist teilbar durch Generatorpolynom,
falls der Dividend mindestens so viele Stellen besitzt
wie das **Generatorpolynom** (führendes Bit muss 1 sein)

Länge der FCS = Grad des Generatorpolynoms = 4

Beim Senden: `` Gegeben: Zu sendender Block: 110011 Generatorpolynom: 11001

Berechnung der FCS: 110011 0000 : 11001 = 10000 11001 ----- 000001 0000 1 1001 ----- 0
1001 = Rest

Zu übertragender, erweiterter Block: 110011 1001 ``

Beim Empfangen (fehlerfrei): `` Gegeben: Zu sendender Block: 110011 Generatorpolynom: 11001

Empfangen eines korrekten Blocks: 110011 1001 : 11001 = 100001 11001 ----- 000001 1001
1 1001 ----- 0 0000 = Rest

Kein Rest: Somit (mit sehr hoher Wahrscheinlichkeit) Daten fehlerfrei ``

Beim Empfangen (fehlerhaft): `` Gegeben: Zu sendender Block: 110011 Generatorpolynom: 11001

Empfangen eines verfälschten Blocks: 111111 1000 : 11001 = 10100 11001 ----- 001101 1
1100 1 ----- 0001 0000 1 1001 ----- 0 1001 = Rest != 0

Rest ungleich 0: Somit Fehler in Übertragung ``

Sliding-Window-Verfahren zusammen mit Go-Back-N:

Bsp:

Modulus $M = 16$
Fenstergröße $W = 9$

Zum aktuellen Zeitpunkt seien **die** Rahmen mit den Sequenznummern **11, 12, 13, 14, 15** vom Sender gesendet worden, ohne dass eine Quittung eingegangen ist.

i) Welche Rahmen dürfen **in** dieser Situation ohne jede Quittung gesendet werden?

Sliding-Window erlaubt es, mehrere Rahmen zu schicken, ohne für die vorherigen eine Quittung empfangen zu haben.

Die Fenstergröße gibt vor, wie viele Rahmen versendet werden können, bevor man auf Quittungen warten muss.

Im Beispiel sind dies neun Stück, also können **in** der gegebenen Situation noch vier weitere Rahmen – 0, 1, 2 und 3 – gesendet werden.

ii) Wie ändert sich **die** Situation, wenn ein Rahmen mit einer Quittungsnummer (ACK) **14** empfangen wird?

Bei einer Quittung gilt alles bis zum vorherigen Rahmen als bestätigt, der Empfänger wartet auf Zustellung aller Rahmen ab dem vierzehnten.

In der gegebenen Situation wird das Fenster also bis zur **14** vorgeschoben, es sind drei weitere Plätze (also **7** insgesamt) frei und wir können insgesamt **0** bis **6** übertragen.

iii) Was passiert, wenn anstelle des ACK **14** ein NAK für **die** Sequenznummer **12** em

Mit einem NAK signalisiert der Empfänger, dass der bezeichnete Rahmen nicht empfangen worden ist.

Laut Go-Back-N müssen alle Rahmen ab dem bezeichneten wieder neu übertragen werden.

Also **12, 13, 14** und **15** – und da mit dem NAK alles vorherige quittiert ist, kann auch das Fenster um eine Position vorgerückt werden und noch **0** bis **4** übertragen werden.

Durchschnittliche Datenrate von Go-Back-N und Selective Repeat:

Bsp:

i) Der Kanal habe eine Datenrate von **100** MBit/s.
Die Paketgröße (inklusive aller Header) betrage **900** Byte.
Wie groß ist **die** Sendedauer t_s für ein Paket?

Teile Paketgröße durch Datenrate:

$$(900 \text{ Byte} * 8 \text{ Bit/Byte}) / 100 \text{ MBit/s} = 72 \mu\text{s}$$

ii) **Die** Bitfehlerrate liege bei $1 * 10^{-5}$.

Geben Sie einen Ausdruck an, um **die** Paketfehlerrate auszudrücken, und vereinfachen Sie diesen soweit Ihnen das ohne Taschenrechner möglich ist.

BER in PER:

$$1 - (1 - 10^{-5})^{900*8} = 1 - 0,99999^{7200}$$

iii) Nehmen Sie zur Vereinfachung der Rechnungen eine Paketfehlerrate von 10 % an.

Wie groß ist **die** mittlere Anzahl N korrekt übertragener Pakete zwischen zwei defekten Paketen?

Wenn 1 von 10 Paketen defekt ist, liegen im Mittel 9 heile zwischen 2 defekten.

iv) Der Kanal sei unbelegt. Der Sender kann nun beginnen, eine Folge von Paketen abzusenden. Einige dieser Pakete werden erfolgreich empfangen, andere nicht. Welche Zeit t_1 wird im Mittel benötigt, bis das erste Paket versendet wurde, das später erfolgreich empfangen wird? Hinweis: Wenn es keine Paketfehler gäbe, entspräche t_1 genau der Sendedauer.

Für 10 Pakete (korrekte und defekte gemischt) benötigen wir 720 μs . Davon sind im Mittel 9 korrekt. Also schaffen wir **in** 720 μs 9 korrekte Pakete. Das sind $t_1 = 80 \mu\text{s}$ für ein korrektes Paket.

v) Wie lange dauert es folglich im Mittel (t_3), drei Pakete zu versenden, die korrekt empfangen werden?

$$t_3 = 240 \mu\text{s}.$$

vi) Ihr Transportprotokoll soll Paketfehler durch Triple-Duplicate-Acknowledgements erkennen. Die Übertragungsverzögerung des Netzwerkes betrage 1020 μs . Wie lange dauert **es** (t_e) ab dem Zeitpunkt, an dem das Absenden eines Rahmens R abgeschlossen ist, bis der Sender den Verlust von R erkennt?

Nehmen Sie an, dass Bestätigungen ohne Verzögerungen abgesendet werden können, keine Sendedauer benötigen und nicht verloren gehen können.

Wir müssen nach dem defekten Paket drei weitere Pakete korrekt verschicken, um drei Dup-ACKs zu triggern. Dafür benötigen **wir** (s.o.) 240 μs . Dann muss das letzte Paket aber noch zum Empfänger und das DUP-ACK zurück.

$$\text{Also: } t_e = 240 \mu\text{s} + 2 * 1020 \mu\text{s} = 2280 \mu\text{s}$$

