



Hinweise zur Personalisierung:

- Ihre Prüfung wird bei der Anwesenheitskontrolle durch Aufkleben eines Codes personalisiert.
- Dieser enthält lediglich eine fortlaufende Nummer, welche auch auf der Anwesenheitsliste neben dem Unterschriftenfeld vermerkt ist.
- Diese wird als Pseudonym verwendet, um eine eindeutige Zuordnung Ihrer Prüfung zu ermöglichen.

Grundlagen Rechnernetze und Verteilte Systeme (GRNVS)

Modul: IN0010 **Prüfer:** Prof. Dr.-Ing. Georg Carle
Klausur: Endterm **Datum:** Dienstag, 26. Juli 2016, 10:30 – 12:00

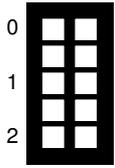
	A 1	A 2	A 3	A 4	A 5
I					
II					

Bearbeitungshinweise

- Diese Klausur umfasst
 - **20 Seiten** mit insgesamt **5 Aufgaben** sowie
 - eine beidseitig bedruckte **Formelsammlung**.
- Bitte kontrollieren Sie jetzt, dass Sie eine vollständige Angabe erhalten haben.
- Das Heraustrennen von Seiten aus der Prüfung ist untersagt.
- Mit * gekennzeichnete Teilaufgaben sind ohne Kenntnis der Ergebnisse vorheriger Teilaufgaben lösbar.
- **Es werden nur solche Ergebnisse gewertet, bei denen der Lösungsweg erkennbar ist.** Auch Textaufgaben sind **grundsätzlich zu begründen**, sofern es in der jeweiligen Teilaufgabe nicht ausdrücklich anders vermerkt ist.
- Schreiben Sie weder mit roter / grüner Farbe noch mit Bleistift.
- Die Gesamtpunktzahl in dieser Prüfung beträgt 85 Punkte.
- Als Hilfsmittel sind zugelassen:
 - ein **nicht-programmierbarer Taschenrechner**
 - ein **analoges Wörterbuch** Deutsch ↔ Muttersprache **ohne Anmerkungen**
- Schalten Sie alle mitgeführten elektronischen Geräte vollständig aus, verstauen Sie diese in Ihrer Tasche und verschließen Sie diese.

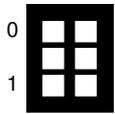
Aufgabe 1 Kurzaufgaben (21.5 Punkte)

Die nachfolgenden Teilaufgaben sind jeweils unabhängig voneinander lösbar.



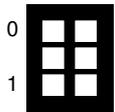
a)* Nennen Sie die Schichten (englisch oder deutsch) des ISO/OSI-Modells in aufsteigender Folge.

Physical Layer, Data-Link Layer, Network Layer, Transport Layer, Session Layer, Presentation Layer, Application Layer

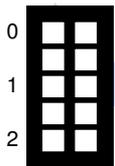
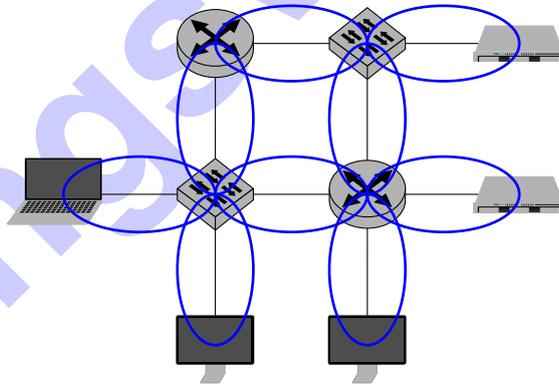


b)* Gegeben sei das 32 bit lange Datum $0x81ff0010$ in Network Byte Order. Wie lautet die Darstellung in Little Endian?

$0x1000ff81$



c)* Gegeben sei das folgende Netzwerk. Zeichnen Sie alle Kollisionsdomänen ein.

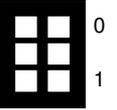


d)* Erklären Sie kurz zwei Probleme, die mit RIP auftreten können.

Nur Hop Count als Metrik \Rightarrow es wird immer die kürzeste Route gewählt, auch wenn eine längere Route höhere Bandbreite oder geringeren Delay bieten würde.
Bis Änderungen vollständig im Netzwerk bekannt sind, können mehrere Minuten vergehen, da Updates nur alle 30 s verschickt werden.
Maximale Distanz beträgt 15 Hops, da größere Werte als „nicht erreichbar“ interpretiert werden \Rightarrow maximale Größe des Netzwerks ist auf 15 Hops begrenzt.
Count-to-Infinity: Bei Topologieänderungen können sich abhängig von der Update-Reihenfolge falsche Routinginformationen verbreiten, bis der max. Hop Count erreicht ist.

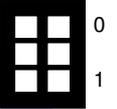
e)* Was ist der Unterschied zwischen RTT und TTL?

RTT: Zeit vom Senden einer Anfrage bis zum Erhalt einer Antwort vom Ziel
TTL: Feld im IPv4 Header, „zählt“ Hops ausgehend von einem Startwert herunter



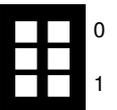
f)* Was ist die Funktion eines Resolvers?

Erlaubt rekursive Namensauflösung im DNS (meist FQDN in IP-Adresse).



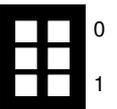
g)* Begründen Sie, ob sich ein Resolver in derselben Broadcastdomäne wie der anfragende Client befinden muss.

Nein, da Resolver auf Schicht 3/4 adressiert werden (IP + UDP/TCP 53).



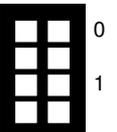
h)* Was versteht man unter einem „präfixfreien Code“?

Ein gültiges Codewort ist niemals Präfix eines anderen Codeworts desselben Codes.



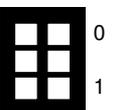
i)* Bestimmen Sie den Reverse-FQDN zu 128.66.50.60.

60.50.66.128.in-addr.arpa.



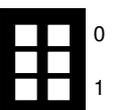
j)* Welche Systemaufrufe sind notwendig, um einen UDP-Socket zur bidirektionalen Kommunikation zu erzeugen?

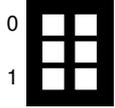
socket() und bind()



k)* Wie lautet die Kurzschreibweise der IPv6-Adresse 2001:0db8:0000:0000:0a00:0000:0000:000c?

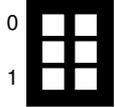
2001:db8::a00:0:0:c





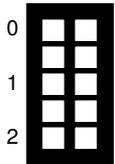
l)* Erklären Sie kurz, wofür SLAAC verwendet wird.

Automatische und zustandslose Erzeugung / Zuweisung von IPv6-Adressen basierend auf der MAC-Adresse.



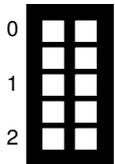
m)* Nennen Sie zwei Routingprotokolle.

RIP, OSPF, BGP, IGRP, EIGRP, IS-IS



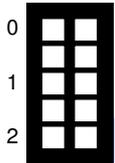
n)* Begründen Sie, warum bei IEEE 802.3 eine minimale Rahmengröße gefordert ist.

Ethernet verwendet keine Bestätigungen, sondern erkennt Kollisionen. Wenn eine Kollision auftritt muss der betroffene Rahmen erneut versendet werden. Damit eine Kollision einem Sendevorgang zugeordnet werden kann, muss diese noch während der Serialisierung des betreffenden Rahmens vom Sender detektiert werden.



o)* Berechnen oder begründen Sie die notwendige Signalleistung P_S , so dass bei einer Rauschleistung von $P_N = 1.0 \text{ mW}$ ein Signal-zu-Rauschabstand von 6 dB erreicht wird.

$$\text{SNR} = 10 \cdot \log\left(\frac{P_S}{P_N}\right) \Leftrightarrow P_S = 10^{\text{SNR}/10} \cdot P_N = 10^{6/10} \cdot 1 \text{ mW} \approx 3.98 \text{ mW}$$



p)* Gegeben sei ein Alphabet mit insgesamt 64 unterschiedlichen Zeichen deren Auftretswahrscheinlichkeit gleichverteilt ist. Begründen Sie, ob die durchschnittliche Codewortlänge bei Nutzung des Huffman-Codes größer, gleich oder kleiner 6 bit ist.

Da die Auftretswahrscheinlichkeit der Zeichen gleichverteilt ist, haben alle Codewörter dieselbe Länge. Es entsteht ein vollständiger Binärbaum der Höhe $\log_2(64) = 6$, womit auch die durchschnittliche Codewortlänge gleich 6 bit ist.

Aufgabe 2 Abtastung und Quantisierung (11 Punkte)

Gegeben sei das in Abbildung 2.1 dargestellte Zeitsignal $s(t)$. Dieses entstand durch Modulation von Rechteckimpulsen auf einen Kosinusträger der Frequenz $f = \frac{1}{2\pi}$. Dieses Signal soll im Folgenden abgetastet und quantisiert sowie dessen Information wiedergewonnen werden.

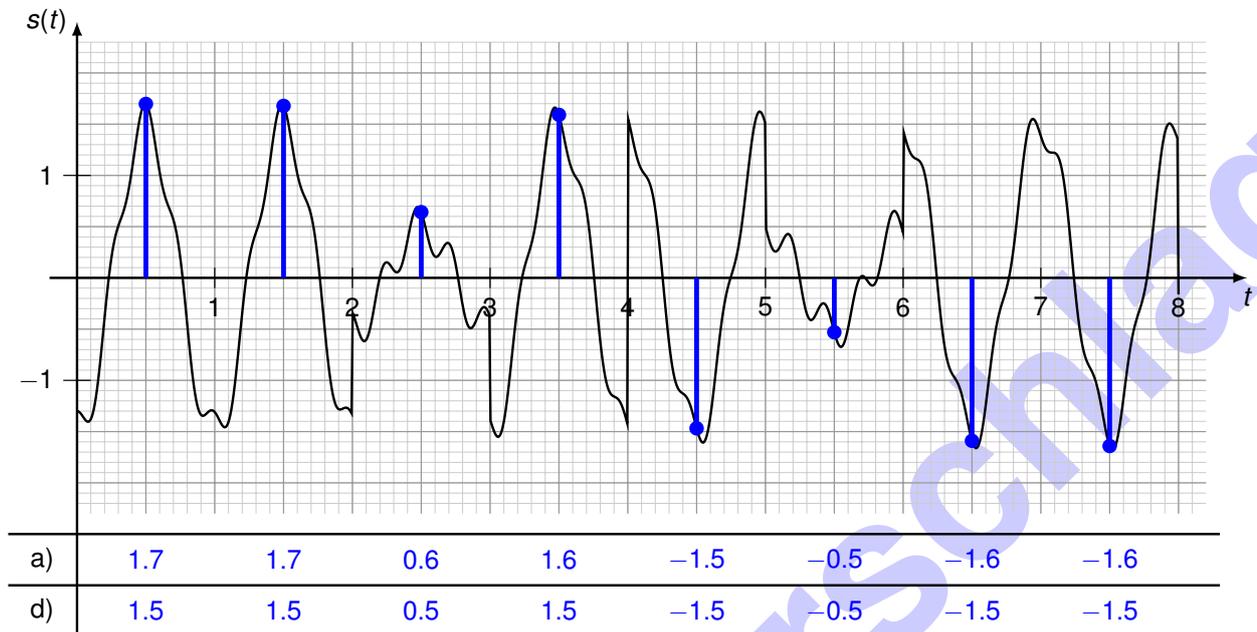


Abbildung 2.1: Moduliertes Signal $s(t)$ und Lösungsvordruck für Teilaufgaben a) und d)

a)* Tasten Sie das Signal $s(t)$ zu den Zeitpunkten $t[n] = n + 0.5$ für $n = 0, 1, \dots$ ab. Zeichnen Sie die Abtastwerte in Abbildung 2.1 ein **und** geben sie die Abtastwerte $s[n]$ auf eine Dezimalstelle genau in der dafür vorgesehenen Zeile an.

0
1
2
3

Wir gehen davon aus, dass $s(t)$ im Intervall $I = [-2; 2]$ gleichverteilt ist. Die Abtastwerte sollen nun auf vier diskrete Werte quantisiert werden, so dass der Quantisierungsfehler in I minimiert wird.

b)* Geben Sie die Quantisierungsstufen an.

0
1

In I äquidistant gewählte Stufen minimieren den Fehler. Die Stufen sind daher:

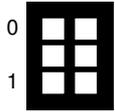
$$S = \{-1.5, -0.5, 0.5, 1.5\}$$

c)* Bestimmen Sie den maximalen Quantisierungsfehler innerhalb von I .

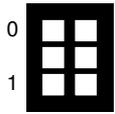
0
1

Bei $N = 4$ Quantisierungsstufen erhalten wir für den max. Fehler innerhalb von I (Quantisierungsfehler 1. Art)

$$q_{\max} = \frac{\Delta}{2} = \frac{b-a}{2N} = \frac{4}{8} = \frac{1}{2}$$

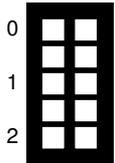


d) Tragen Sie die quantisierten Abtastwerte $\hat{s}[n]$ in Abbildung 2.1 in der dafür vorgesehenen Zeile ein.



e)* Wie viele Bit werden zur Darstellung der Quantisierungsstufen benötigt.

$$n = \log_2(N) = 2 \text{ bit}$$

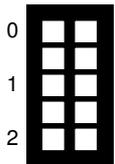


f) Begründen Sie kurz, welches Modulationsverfahren bei der Erzeugung von $s(t)$ mit hoher Wahrscheinlichkeit benutzt wurde.

QAM kann ausgeschlossen werden, da nach Aufgabenstellung lediglich ein Kosinusträger verwendet wurde.

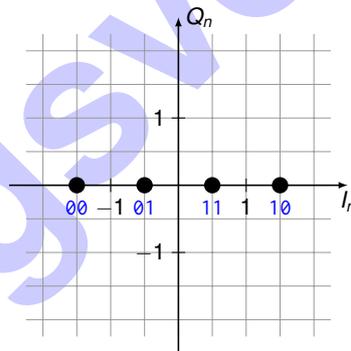
Bei 4-PSK könnten Phasensprünge um 90° und 270° auftreten, es gibt aber nur Phasensprünge um 0° und 180° .

⇒ Amplitude Shift Keying mit vier Stufen (4-ASK), da sich die einzelnen Abtastwerte nur in ihren Amplituden unterscheiden und nahe bei den vier erwarteten Quantisierungsstufen liegen.



g) Geben Sie eine gültige Signalraumzuordnung an.

Hinweis: Es gibt mehrere mögliche Lösungen. Die Angabe einer Lösung ist ausreichend.



Aufgabe 3 Datamatrix in Kornfeldern (15 Punkte)

Die Moepis vom Planeten Gliese 587c haben große technologische Fortschritte gemacht und einen neuen Planeten entdeckt – die Erde. Dieser, so scheint es, ist mit einer seltsamen Spezies bevölkert. Um mit diesen Wesen Kontakt aufzunehmen, benutzen die Moepis einen Laserstrahl, mit dessen Hilfe sie Datamatrix-Codes in Kornfelder brennen.

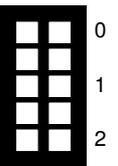
Zur Übertragung des i -ten Buchstabens wird dieser zunächst als 7 bit langes ASCII-Zeichen a_i dargestellt, welches im Anschluss als 8 bit langes Datum $d_i = a_i + 1$ kodiert wird. Eine Sequenz von k Buchstaben wird als Polynom

$$d(x) = \sum_{i=1}^k d_i x^{k-i} \text{ mit } d_i \in \{0, 1, \dots, 255\} \quad (1)$$

kodiert.

a)* Geben Sie $d(x)$ für die Nachricht „Hello“ an (ohne Anführungszeichen).

$$d(x) = 73x^4 + 102x^3 + 109x^2 + 109x + 112$$



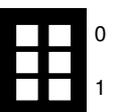
Da beispielsweise bei defekten Stellen im Kornfeld Übertragungsfehler auftreten können, wird die Nachricht mittels *Reed-Solomon-Code* gesichert. Dabei handelt es sich um einen Blockcode, der einen Datenblock $d(x)$ der Länge k Byte auf einen Codeblock $c(x)$ der Länge n Byte abbildet (s. Abbildung 3.1). Innerhalb eines Codeblocks können dafür eine gewisse Anzahl fehlerhafter Bytes korrigiert werden.



Abbildung 3.1: Reed-Solomon-Encoder

b)* Begründen Sie, ob es sich bei dem hier beschriebenen Verfahren um Kanal- oder Quellenkodierung handelt.

Kanalkodierung, da der zu übertragenden Nachricht strukturierte Redundanz zur Fehlerkorrektur hinzugefügt wird.



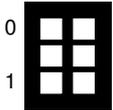
Hinweis: Sofern Sie Teilaufgabe a) nicht lösen konnten, verwenden Sie für die nachfolgenden Teilaufgaben das Polynom $d(x) = 78x^4 + 112x^3 + 102x^2 + 113x + 106$.

Der Codeblock $c(x)$ zu einem Datenblock $d(x)$ wird gemäß

$$c(x) = d(x) \cdot x^{n-k} + e(x) \tag{2}$$

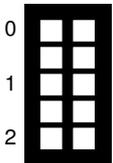
berechnet, wobei $e(x) = d(x) \cdot x^{n-k} \bmod r(x)$ die redundante Information darstellt. Im vorliegenden Fall werden $k = 5$ B auf $n = 12$ B abgebildet. Das zugehörige Reduktionspolynom $r(x)$ lautet

$$r(x) = 254x^6 + 92x^5 + 240x^4 + 134x^3 + 144x^2 + 68x + 23. \tag{3}$$



c)* Inwiefern unterscheidet sich $e(x)$ von einer Checksumme, wie sie z. B. bei CRC verwendet wird?

Kanalkodierung: Fehlerkorrektur
CRC: Fehlererkennung

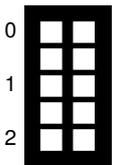


d)* Begründen Sie, wie sich die ersten 5 B des Codeblocks zusammensetzen.

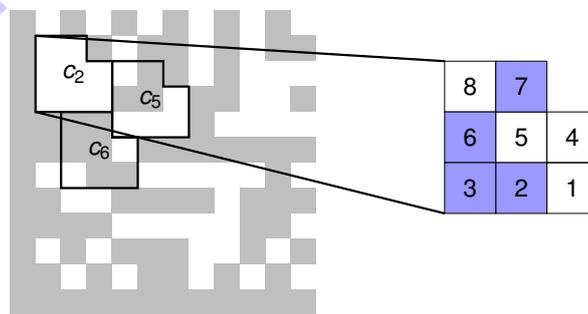
0x49 0x66 0x6d 0x6d 0x70

Die ersten 5 B entsprechen der Nachricht, da $d(x)$ mit x^{n-k} multipliziert wird, was einer Verschiebung der entsprechenden Positionen um 7 B entspricht. Zusätzlich ist der Grad von $e(x)$ kleiner als $n - k$ und somit hat $e(x)$ keinen Einfluss auf die ersten 5 B.

Wie im Lösungsfeld zu Teilaufgabe e) dargestellt, werden in der Datamatrix einzelne Symbole zu je 8 bit in Form von Blöcken zu je 3×3 Pixel dargestellt, wobei die rechte obere Ecke jedes Blocks frei bleibt. Die Pixel sind entsprechend ihrer Wertigkeit nummeriert („1“ markiert das niederwertigste bit).



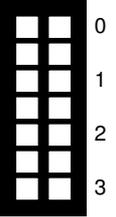
e) Markieren Sie für das **zweite** Symbol die gesetzten (logisch 1) Bitpositionen.



Das Moepi, welches den Laser bedient, versagt mit Wahrscheinlichkeit $0 < \epsilon < 1$, was dann zu einer fehlerhaften Markierung führt. Ein Symbol (wie im Lösungsfeld zu Teilaufgabe e) dargestellt ist defekt, sobald mindestens eine Position fehlerhaft ist.

Der Reed-Solomon-Code ist in der Lage, bis zu drei defekte Bytes innerhalb einer gesicherten Nachricht von $n = 12$ B zu korrigieren – unabhängig davon, wie viele Bitfehler in einem betroffenen Byte enthalten sind.

f)* Bestimmen Sie die Wahrscheinlichkeit in Abhängigkeit von ϵ , dass die Menschen die Datamatrix korrekt entziffern können. Vereinfachen Sie das Ergebnis soweit möglich.



Sei X eine ZV, welche die Anzahl an Bitfehlern je Byte zählt. Dann erhalten wir

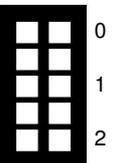
$$\Pr[X = 0] = (1 - \epsilon)^8 = \rho$$

Sei Y eine ZV, welche die Anzahl fehlerhafter Bytes in einer kodierten Nachricht zählt. Dann erhalten wir

$$\begin{aligned} \Pr[Y \leq 3] &= \sum_{i=0}^3 \binom{12}{i} (1 - \rho)^i \rho^{12-i} \\ &= (\rho^3 + 12 \cdot (1 - \rho)\rho^2 + 66 \cdot (1 - \rho)^2\rho + 220 \cdot (1 - \rho)^3) \rho^9 \end{aligned}$$

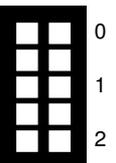
Leider ist den Moepis bei der Berechnung der Reed-Solomon-Codes jedoch ein Fehler unterlaufen, weswegen $e(x)$ falsch bestimmt wurde.

g) Erklären Sie, wie die Menschen auf der Erde die Nachricht dennoch dekodieren können.



Wir ignorieren die „Antiredundanz“ und betrachten nur die ersten 5 B je Datamatrix, welche unabhängig von $e(x)$ ist. Unter der Annahme, dass innerhalb dieser ersten 5 B der Nachricht kein fehlerhaftes Byte enthalten ist, kann die Nachricht dekodiert werden.

h) Bestimmen Sie die Wahrscheinlichkeit in Abhängigkeit von ϵ , dass eine Datamatrix mit dem in Teilaufgabe g) beschriebenen Verfahren korrekt dekodiert werden kann. Vereinfachen Sie das Ergebnis soweit möglich.



$$\Pr[Y = 0] = \rho^k = (1 - \epsilon)^{40}$$

Aufgabe 4 NAT und statisches Routing (15 Punkte)

Wir betrachten das Netzwerk aus Abbildung 4.1. PC1 und PC2 sind über den Switch S miteinander und mit dem Router R1 verbunden. Im lokalen Netzwerk werde das Subnetz 10.12.121.32/28 verwendet. Der Router R1 ist über ein Transportnetz der Präfixlänge 30 mit R2 verbunden, welcher das Gateway zum Internet auf der Seite eines Service Providers darstellt.

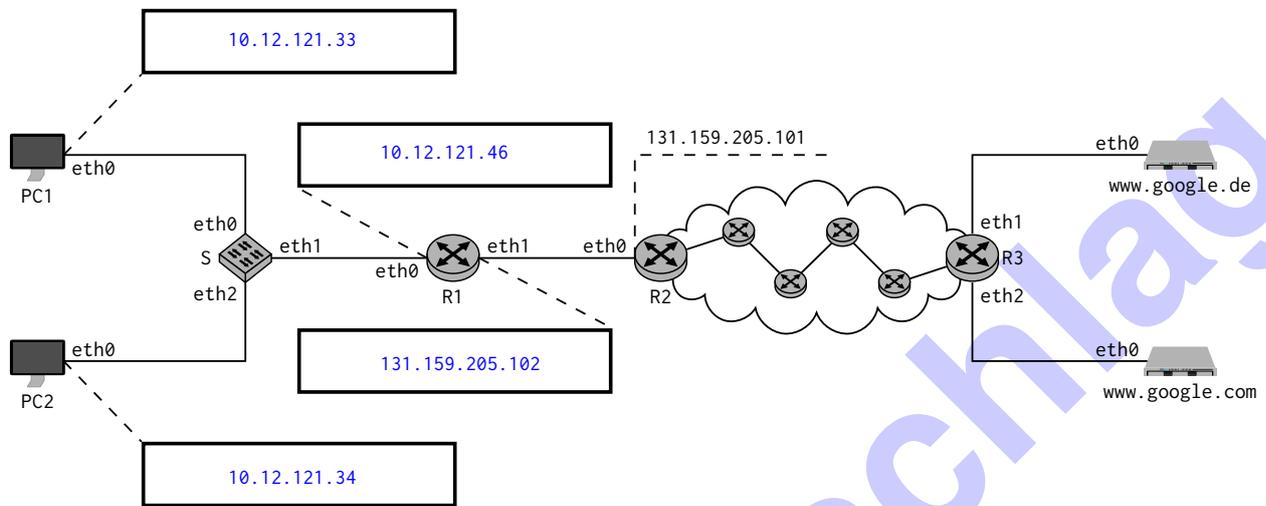


Abbildung 4.1: Netztopologie

a)* Bestimmen Sie die Broadcast-Adresse des Subnetzes 10.12.121.32/28.

Präfixlänge 28 \Rightarrow 4 bit Hostanteil $\Rightarrow 2^4 = 16$ Adressen im Subnetz
 \Rightarrow 10.12.121.47 ist Broadcast-Adresse.

b)* Vergeben Sie an PC1, PC2 und R1 IP-Adressen aus dem Subnetz 10.12.121.32/28. Tragen Sie diese direkt in Abbildung 4.1 ein.

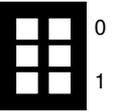
c)* Bestimmen Sie die Netzadresse des Transportnetzes zwischen R1 und R2.

Präfixlänge 30 \Rightarrow 2 bit Hostanteil \Rightarrow 4 Adressen im Subnetz. Da R2 131.159.205.101 hat, muss 131.159.205.103 die Broadcastadresse und 131.159.205.100 damit die Netzadresse sein.

d) Weisen Sie R1 eine Adresse aus dem Transportnetz zu. Tragen Sie diese direkt in Abbildung 4.1 ein.

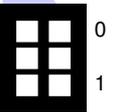
e)* Wie viele /30 Subnetze gibt es im Netz 131.159.0.0/16?

$2^{30-16} = 2^{14} = 16384$



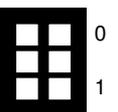
f)* Begründen Sie, warum R1 NAT unterstützen muss, um PC1 und PC2 Zugang zum Internet zu ermöglichen.

Das Subnetz 10.12.121.32/28 ist Bestandteil des privaten Adressbereichs 10.0.0.0/8, weswegen die IP-Adressen nicht global eindeutig sind.



g)* Welches Transportprotokoll und welcher Zielport wird verwendet, wenn PC1 mittels Browser auf die Webseite www.google.de zugreift?

TCP 443 (da Google mittlerweile standardmäßig HTTPS verwendet).
(TCP 80 da HTTP auch richtig)



Im Folgenden kürzen wir IP- und MAC-Adressen nach dem Schema <Gerät>.<Interface> ab, z. B. R1.eth0 für die entsprechende Adresse an Interface eth0 von Router R1. Beachten Sie für die nachfolgenden Teilaufgaben außerdem, dass sich zwischen R2 und R3 vier weitere Router befinden. PC1 greift nun auf die Webseite www.google.de zu.

h) Ergänzen Sie für die Anfrage von PC1 an www.google.de die Headerfelder in den drei leeren Kästen in Abbildung 4.2. Falls ein Feld nicht eindeutig bestimmt ist, treffen Sie eine sinnvolle Wahl.

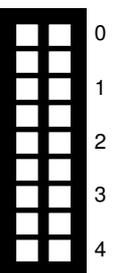
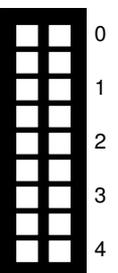
Hinweis:

- Falls Sie Teilaufgabe g) nicht lösen konnten, nehmen Sie Zielport 443 an.
- Der Hostname des Servers, auf dem www.google.de gehostet wird, kann durch „G“ abgekürzt werden.

i) Ergänzen Sie für die Antwort von www.google.de an PC1 die Headerfelder in den drei leeren Kästen in Abbildung 4.3. Falls ein Feld nicht eindeutig bestimmt ist, treffen Sie eine sinnvolle Wahl.

Hinweis:

- Der Hostname des Servers, auf dem www.google.de gehostet wird, kann durch „G“ abgekürzt werden.



Src MAC	R3.eth1
Dst MAC	google.eth0
Src IP	R1.eth1
Dst IP	google.eth0
TTL	56
Src Port	1235
Dst Port	80

Src MAC	R1.eth1
Dst MAC	R2.eth0
Src IP	R1.eth1
Dst IP	google.eth0
TTL	62
Src Port	1235
Dst Port	80

Src MAC	PC1.eth0
Dst MAC	R1.eth0
Src IP	PC1.eth0
Dst IP	google.eth0
TTL	63
Src Port	1234
Dst Port	80

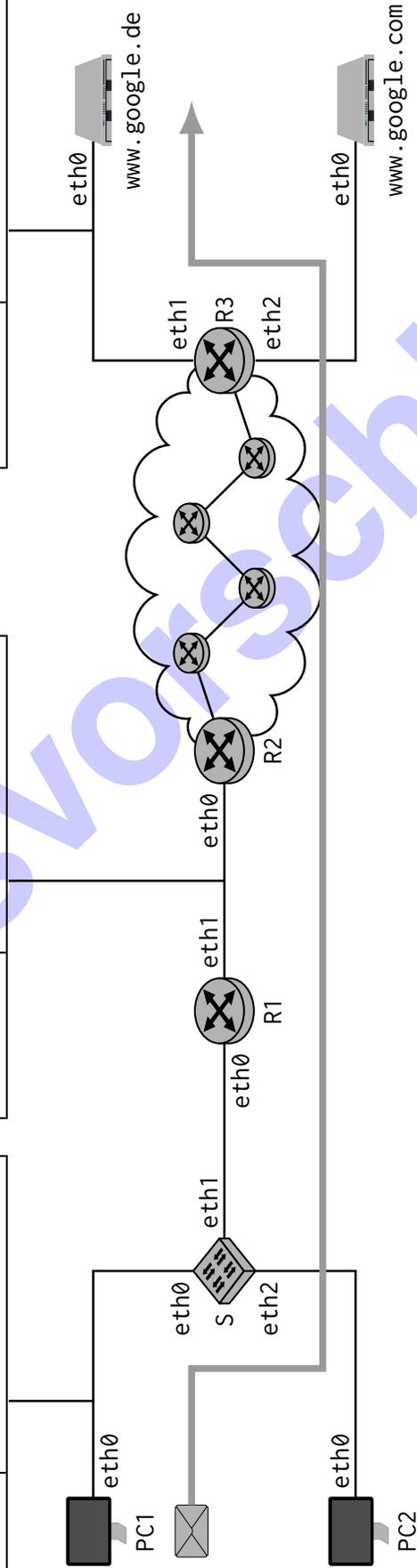


Abbildung 4.2: Lösungsvordruck für Teilaufgabe h)

Src MAC	google.eth0
Dst MAC	R3.eth1
Src IP	google.eth0
Dst IP	R1.eth1
TTL	63
Src Port	80
Dst Port	1235

Src MAC	R2.eth0
Dst MAC	R1.eth1
Src IP	google.eth0
Dst IP	R1.eth1
TTL	57
Src Port	80
Dst Port	1235

Src MAC	R1.eth0
Dst MAC	PC1.eth0
Src IP	google.eth0
Dst IP	PC1.eth0
TTL	56
Src Port	80
Dst Port	1234

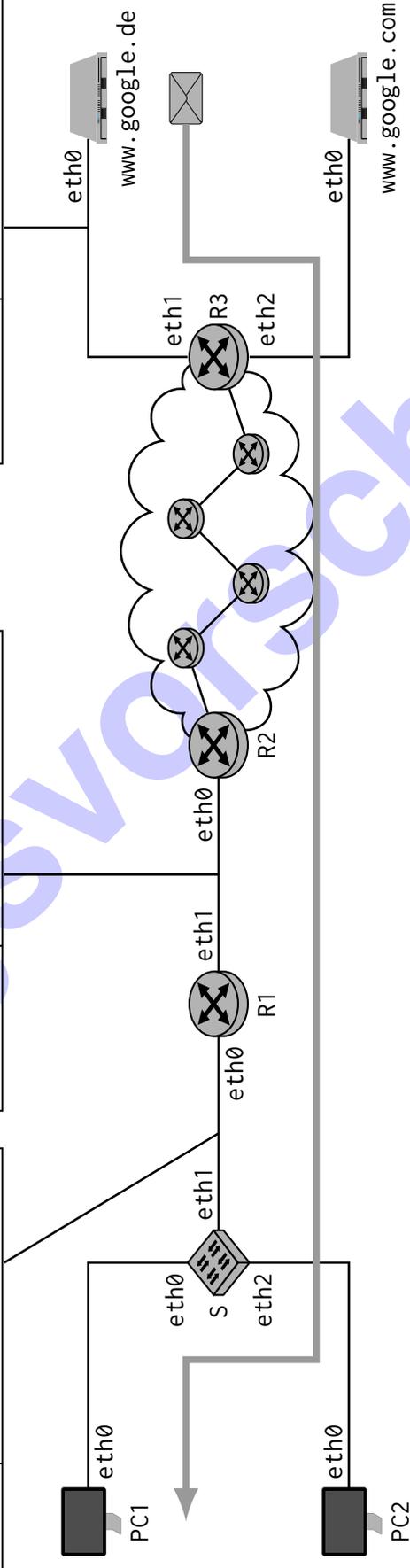


Abbildung 4.3: Lösungsvordruck für Teilaufgabe i)

Aufgabe 5 Transportprotokolle (22.5 Punkte)

Wir betrachten eine Verbindung zwischen einem Client C und einem Server S über das Internet. Vorgänge unterhalb der Transportschicht können vernachlässigt werden.

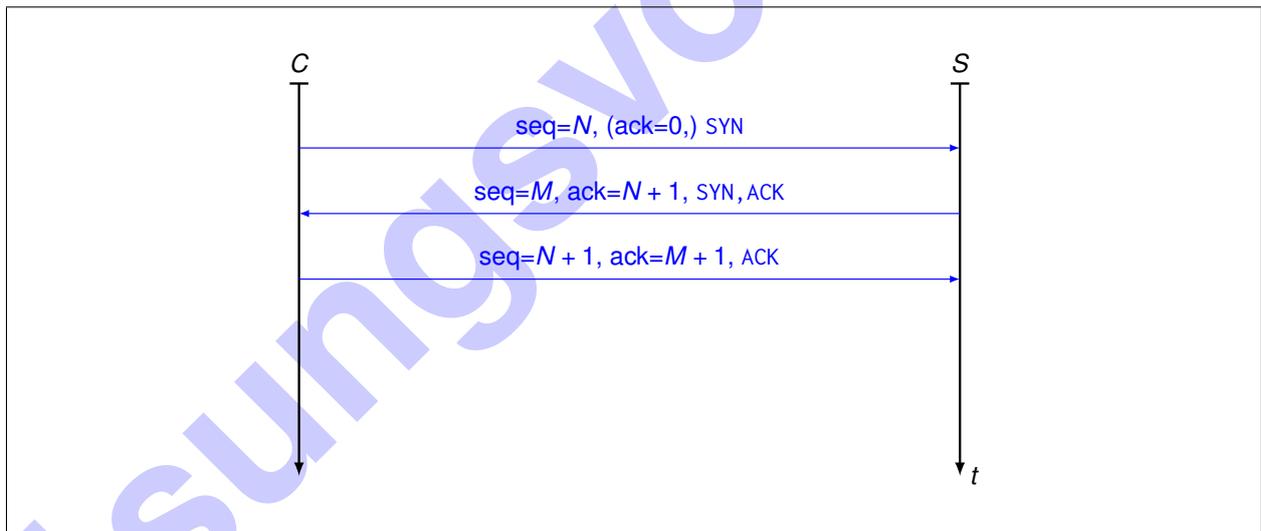
a)* Erläutern Sie zwei wesentliche Unterschiede zwischen TCP und UDP.

TCP ist verbindungsorientiert während UDP verbindungslos arbeitet, d. h. Verbindungsaufbau und Synchronisation entfallen bei UDP.
TCP bietet eine gesicherte Übertragung, d. h. kommen beim Empfänger vollständig und in der richtigen Reihenfolge oder gar nicht an. UDP arbeitet nach dem Best-Effort-Prinzip, d. h. es kümmert sich nicht um Übertragungsfehler oder Reihenfolge.

Wir betrachten zunächst nur TCP. Wir nehmen an, dass Client C eine Verbindung zum Server S auf TCP 80 aufbaut. Die initialen Sequenznummern seien N für den Client C und M für den Server S.

b)* Skizzieren Sie den Verbindungsaufbau als vereinfachtes Weg-Zeit-Diagramm. Geben Sie dabei für jede ausgetauschte Nachricht Sequenznummer, Bestätigungsnummer sowie die gesetzten Flags an.

Hinweis: Beachten Sie die oben angegebenen initialen Sequenznummern.



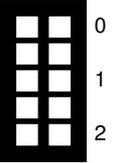
Nach dem Verbindungsaufbau fragt C den Download eines Films an, welcher daraufhin vom Server versendet wird. Die MSS betrage 1460 B.

c)* Was versteht man im Allgemeinen unter einer MSS?

Die maximale Länge der Nutzdaten pro TCP-Segment, d. h. die maximale Größe einer L4-SDU.

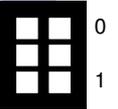
d)* Wie errechnet sich der häufig anzutreffende Wert $1 \text{ MSS} = 1460 \text{ B}$?

1500 B MTU abzüglich IPv4-Header und TCP-Header ohne Optionen zu je 20 B.



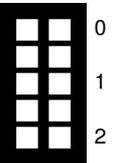
e)* Nennen Sie die beiden Phasen, die der Staukontrollmechanismus von TCP während der Übertragungsphase durchläuft (ohne Begründung).

Slow Start und Congestion Avoidance .



f) Beschreiben Sie kurz die Entwicklung der Größe w des Sendefensters während beider Staukontrollphasen.

Verdopplung für jedes vollständig bestätigte Sendefenster während des Slow-Starts.
Zuwachs von 1 MSS für jedes vollständig bestätigte Sendefenster während der Congestion-Avoidance Phase.

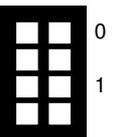


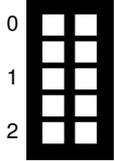
Wir wollen nun die benötigte Zeit zur Übertragung des Films bestimmen, welcher einer Größe von $L = 600 \text{ MiB}$ habe. Serialisierungs- und Verarbeitungszeiten können vernachlässigt werden. Der initiale Request zum Download sei bereits versendet, d. h. s beginnt direkt mit dem Versenden des Films.

Wir betrachten die Größe $w[k]$ des Sendefensters an s zu den Zeitpunkten $k \cdot \text{RTT}$ für $k = 0, 1, \dots$, d. h. zu Beginn des k -ten Zeitschritts sind $k \cdot \text{RTT}$ Sekunden vergangen. Die RTT betrage 10 ms und für $k = 0$ gelte $w[k] = 1 \text{ MSS}$. Vereinfachend sei angenommen, dass genau ein Segment verloren geht, sobald $w[k] \geq x = 128 \text{ MSS}$. Das Empfangsfenster an c sei so gewählt, dass es keinen Einfluss auf die Entwicklung des Sendefensters hat.

g)* Bestimmen Sie die Anzahl n der zu übertragenden Segmente.

$$n = \left\lceil \frac{L}{\text{MSS}} \right\rceil = \left\lceil \frac{600 \cdot 2^{20} \text{ B}}{1460 \text{ B}} \right\rceil = 430922$$



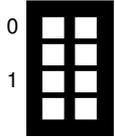


h)* Bestimmen Sie die Zeit t_1 bis zum Auftritt des ersten Segmentverlusts in Sekunden.

Zu Beginn wächst das Staukontrollfenster pro bestätigter MSS um 1 MSS beginnend bei $w[0] = 1$ MSS. Segmentverlust tritt ein, sobald $w[n_1] = x$, d. h.

$$\begin{aligned}w[k_1] &\stackrel{!}{=} x \\k_1 &= \log_2(128) + 1 = 8 \\t_1 &= k_1 \cdot \text{RTT} = 0.08 \text{ s}\end{aligned}$$

Achtung: Von $w[0]$ bis $w[7]$ finden nicht 7 sondern 8 Übertragungsschritte statt!



i) Bestimmen Sie die Anzahl n_1 der **erfolgreich** übertragenen Segmente bis zum Auftritt des ersten Segmentverlusts.

Innerhalb dieser ersten 8 RTT werden

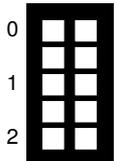
$$n_1 = \left(\sum_{i=0}^{k_1} 2^i \right) - 1 = 2^{(k_1+1)} - 2 = 254$$

Segmente übertragen.

Nach dem ersten Segmentverlust verbleiben $n_2 = n - n_1$ Segmente, die noch zu übertragen sind.

Hinweise:

1. Falls Sie Teilaufgabe i) nicht lösen konnten, gehen Sie von 430668 noch verbleibenden Segmenten aus.
2. Beachten Sie die Angaben im Abschnitt „Transportschicht“ auf der Formelsammlung.



j)* Bestimmen Sie die Dauer zwischen dem Auftreten weiterer Segmentverluste in Sekunden.

$$\begin{aligned}T &= \left(\frac{x}{2} + 1 \right) \cdot \text{RTT} \\&= \left(\frac{128}{2} + 1 \right) \cdot 10 \text{ ms} \\&= 0.65 \text{ s}\end{aligned}$$

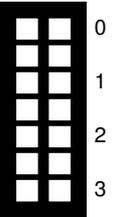
k)* Bestimmen Sie die verbleibende Zeit t_2 , bis die Übertragung abgeschlossen ist.

Nach Formelsammlung werden pro Congestion-Avoidance-Phase

$$n' = \frac{3}{8} \cdot x^2 + \frac{3}{4} \cdot x = \Big|_{x=128} 6240$$

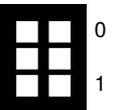
Segmente übertragen. Folglich werden $n_2/n' \approx 70$ Phasen durchlaufen, wobei je Phase genau ein Segment verloren geht. Damit ergibt sich für die verbleibende Zeit

$$t_2 = \frac{n_2}{n'} \cdot 0.65 \text{ s} \approx 44.86 \text{ s}$$



l) Bestimmen Sie die Gesamtzeit t für die Datenübertragung in Sekunden.

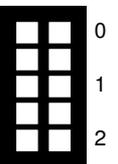
$$\begin{aligned} t &= t_1 + t_2 \\ &= 44.93 \text{ s} \end{aligned}$$



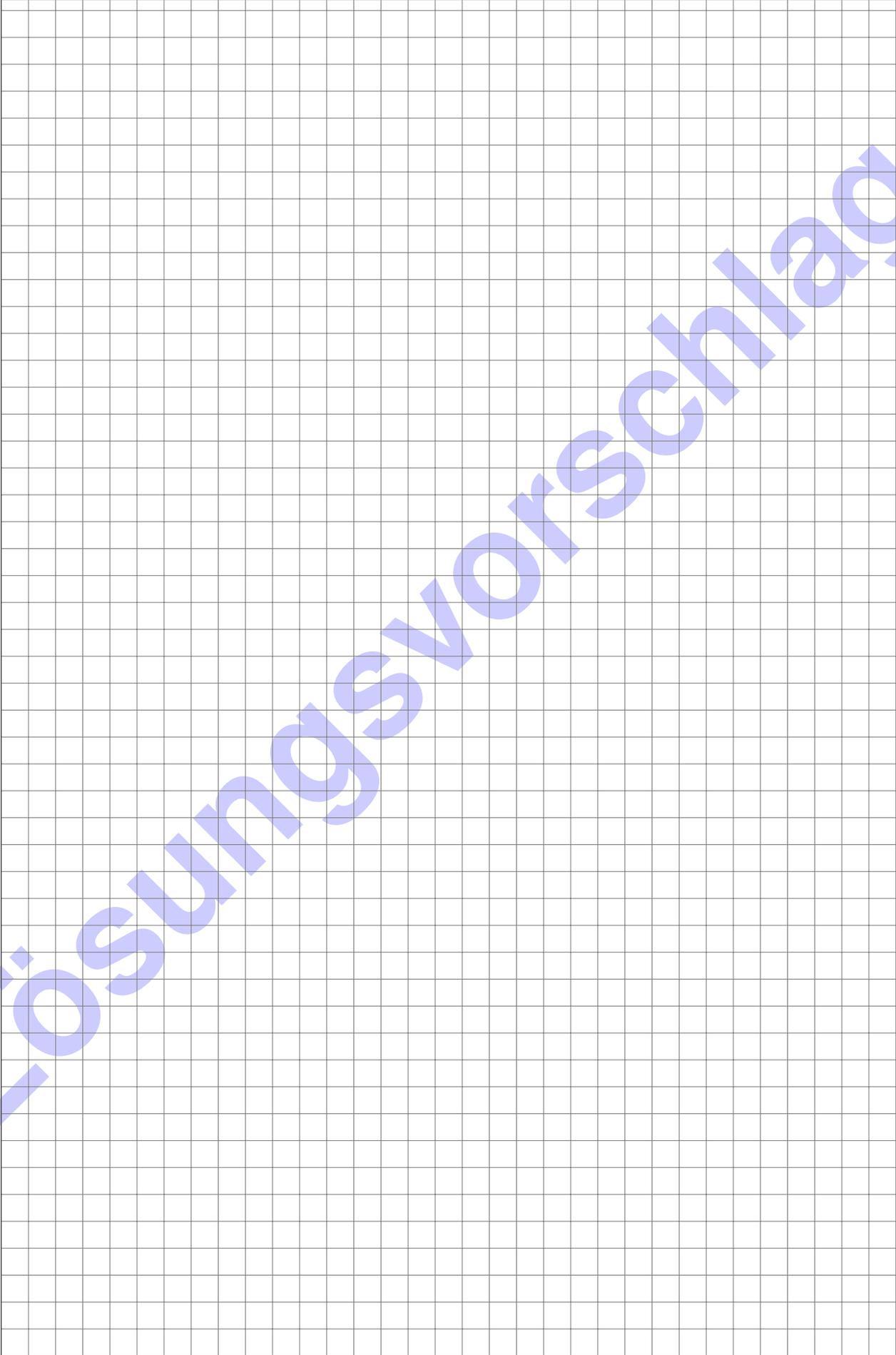
m)* Begründen Sie, ob für den hier betrachteten Download UDP als sinnvolle Alternative in Betracht käme.

Nein, da UDP keinerlei Staukontrollmechanismen beherrscht und dadurch bedingte Segmentverluste nicht ausgeglichen werden. Der heruntergeladene Film wäre mit hoher Wahrscheinlichkeit kaputt.

(Auch ok: Argumentation mit ja, sofern Staukontrolle anderweitig implementiert wird, da einzelne Fehler im Video vertretbar sein könnten)



Zusätzlicher Platz für Lösungen. Markieren Sie deutlich die Zuordnung zur jeweiligen Teilaufgabe. Vergessen Sie nicht, ungültige Lösungen zu streichen.



Lösungsvorschlag

Lösungsvorschlag