

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

Klausur: IN0010 / Endterm

Datum: Mittwoch, 1. August 2018

Prüfer: Prof. Dr. Uwe Baumgarten

Uhrzeit: 08:00 – 09:30

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

Bearbeitungshinweise

- Diese Klausur umfasst
 - **16 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.
- Rechenergebnisse sind auf zwei signifikante Nachkommastellen arithmetisch gerundet anzugeben, 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 90 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.

Hörsaal verlassen von _____ bis _____ / Vorzeitige Abgabe um _____

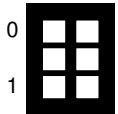
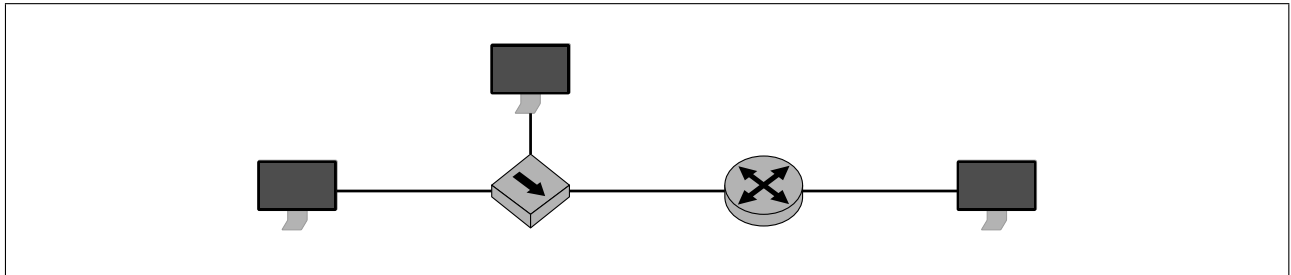
Aufgabe 1 Kurzaufgaben (13 Punkte)

Die nachfolgenden Teilaufgaben sind jeweils unabhängig voneinander zu beantworten.



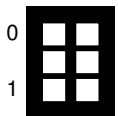
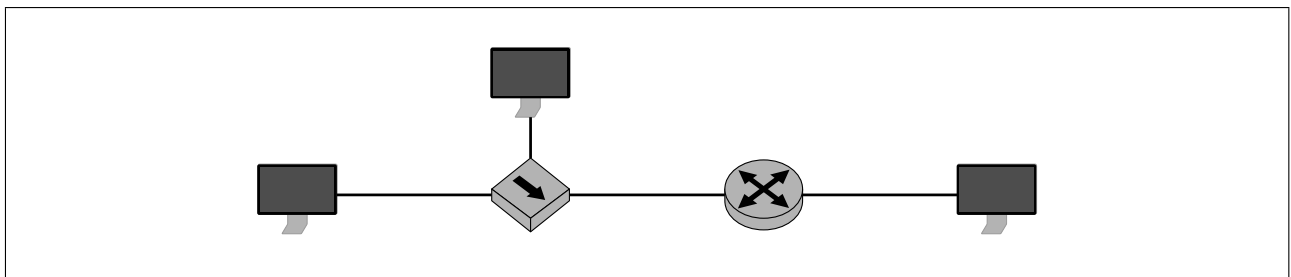
a)* Markieren Sie im untenstehenden Netzwerk alle *Kollisionsdomänen*.

Wichtig: Achten Sie darauf, bei der Markierung nur die Interfaces in die Markierung einzuschließen, die sich auch in der jeweiligen Kollisionsdomäne befinden!

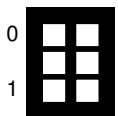


b)* Markieren Sie im untenstehenden Netzwerk alle *Broadcastdomänen*.

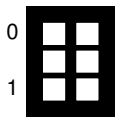
Wichtig: Achten Sie darauf, bei der Markierung nur die Interfaces in die Markierung einzuschließen, die sich auch in der jeweiligen Broadcastdomäne befinden!



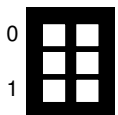
c)* Wie viele unterschiedliche IPv6-Adressen sind theoretisch möglich? (Angabe als Potenz ausreichend)



d)* Nennen Sie 2 routebare Protokolle. (ohne Begründung)

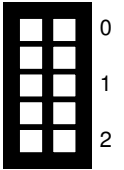


e)* Was versteht man unter einem Socket?

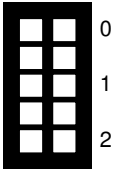


f)* Beschreiben Sie kurz den Unterschied zwischen *Interior* und *Exterior Gateway Protocols* (IGPs und EGPs).

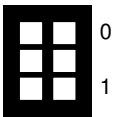
g)* Erläutern Sie den Unterschied zwischen den Syscalls `send()` und `sendto()`.



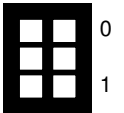
h)* Nennen Sie die notwendigen Syscalls **in der richtigen Reihenfolge**, um einen verbindungsorientierten Socket zu erstellen und für eingehende Verbindungen vorzubereiten.



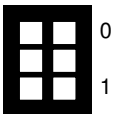
i)* Geben Sie das in Network-Byte-Order gegebene 32 bit Datum `0x01 23 45 67` in Big Endian an.



j)* Geben Sie 10 Gbit in der Einheit MiB an.



k)* Bestimmen Sie Netz- und Broadcast-Adresse des kleinst möglichen Subnetzes, welches mindestens die Adressen `203.0.113.17` und `203.0.113.46` umfasst.



Aufgabe 2 Ethernet Physical Layer (17 Punkte)

In dieser Aufgabe untersuchen wir zwei unterschiedliche Implementierungen des Ethernet Physical Layers. Zunächst diskutieren wir (das etwas veraltete) 10BASE-2. Als Leitungscodierung wird die Manchesterkodierung eingesetzt. Eine zusätzliche Kanalkodierung findet nicht statt. Gegeben sei das in Abbildung 2.1 idealisiert dargestellte 10BASE-2-Signal.

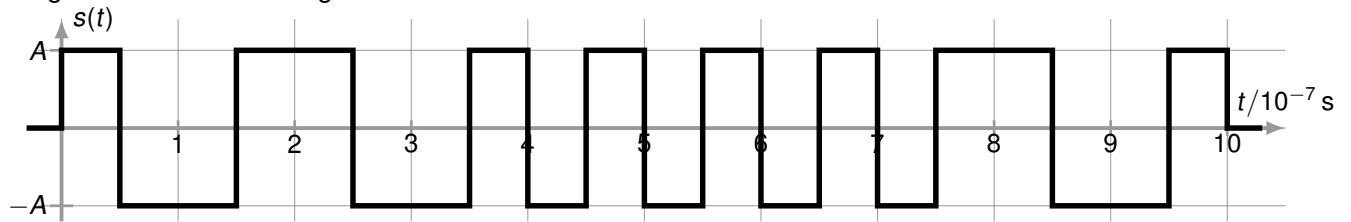


Abbildung 2.1: Idealisierter Verlauf eines 10BASE-2-Signals.

a)* Ist das Signal zeitkontinuierlich oder zeitdiskret (ohne Begründung)?

0 ☐

1 ☐

b)* Bestimmen Sie die im Zeitintervall $t \in [0 \mu\text{s}, 1 \mu\text{s})$ übertragene Bitfolge.
Hinweis: Es gibt zwei gültige Lösungen.

0 ☐

1 ☐

2 ☐

c)* Wie lange dauert es, ein einzelnes Bit zu serialisieren?

0 ☐

1 ☐

d)* Bestimmen Sie die mit 10BASE-2 erzielbare Datenrate (Rechnung oder Begründung).

0 ☐

1 ☐

e) Bestimmen Sie die nach Hartley minimal notwendige spektrale Bandbreite, um mit einem binären Leitungscodierung die in Teilaufgabe d) bestimmte Datenrate erreichen zu können.

0 ☐

1 ☐

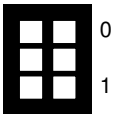
2 ☐

f) Begründen Sie, weswegen 10BASE-2 mindestens eine Bandbreite von $B' = 10 \text{ MHz}$ belegt.

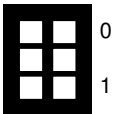
0 ☐

1 ☐

g) Begründen Sie, mit welchem **anderen binären** Leitungscode eine höhere Datenrate bei derselben spektralen Bandbreite erzielt werden kann.



h) Welchen wesentlichen Vorteil bietet der Manchestercode?



Im Folgenden betrachten wir den neueren 100BASE-TX Standard. Dieser verwendet als Leitungscode MLT-3 mit 4B5B-Kodierung. Die effektive Datenrate beträgt 100 Mbit/s. In Abbildung 2.2 ist ein idealisierter Signalverlauf einer einzelnen Ader dargestellt, welcher die ersten vier Bit der Nachricht 0101 111101 darstellt.

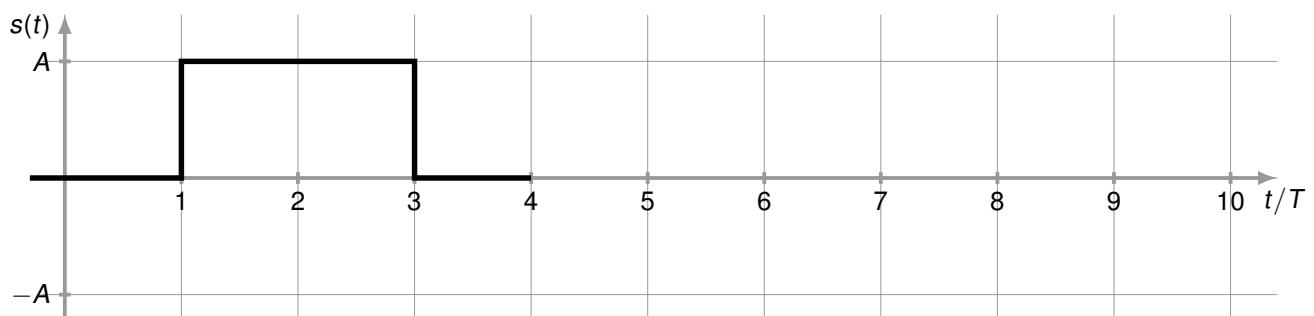
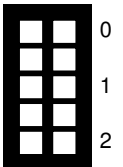
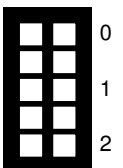


Abbildung 2.2: Idealisierter Verlauf eines MLT-3 Signals.

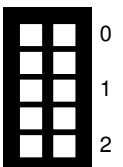
i)* Vervollständigen Sie den Signalverlauf in Abbildung 2.2 um die verbleibenden sechs Bit.



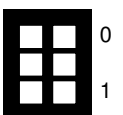
j) Begründen Sie, ob bei Verwendung der MLT-3 Kodierung Probleme beim Erkennen langer Null- oder Einsfolgen auftreten.



k)* Nennen Sie zwei Vorteile, die die 4B5B-Kodierung mit sich bringt.



l)* Wie hoch muss bei 100BASE-TX die Bitrate tatsächlich sein, um eine effektive Übertragungsgeschwindigkeit von 100 Mbit/s zu erreichen?



Hinweis: Es geht hier lediglich um die Bitrate aus Sicht der physikalischen Schicht. Sie brauchen also nicht den durch Protokollheader entstehenden Overhead zu berücksichtigen!

Aufgabe 3 TCP Fluss- und Staukontrolle (22 Punkte)

Das im Internet am weitesten verbreitete Transportprotokoll ist TCP. Dieses implementiert Mechanismen zur Fluss- und Staukontrolle. Konkret nehmen wir in dieser Aufgabe TCP „Reno“ wie in der Vorlesung eingeführt an.

Die folgenden 6 Teilaufgaben sind Multiple Choice Single Answer, d. h. Sie müssen sich pro Teilaufgabe für genau eine Lösung entscheiden.

Kreuzen Sie richtige Antworten an

Kreuze können durch vollständiges Ausfüllen gestrichen werden

Gestrichene Antworten können durch nebenstehende Markierung erneut angekreuzt werden



Ordnen Sie die folgenden Konzepte und Begriffe jeweils der Stau- bzw. Flusskontrolle zu:

a)* Überlast beim Empfänger

☐ beides

☐ nicht zutreffend

☐ Staukontrolle

☐ Flusskontrolle

b)* Überlast beim Sender

☐ beides

☐ nicht zutreffend

☐ Staukontrolle

☐ Flusskontrolle

c) Verbindungsaufbau

☐ beides

☐ nicht zutreffend

☐ Staukontrolle

☐ Flusskontrolle

d) Sendefenster

☐ beides

☐ nicht zutreffend

☐ Staukontrolle

☐ Flusskontrolle

e) Empfangsfenster

☐ beides

☐ nicht zutreffend

☐ Staukontrolle

☐ Flusskontrolle

f) Paketverlust im Netzwerk

☐ beides

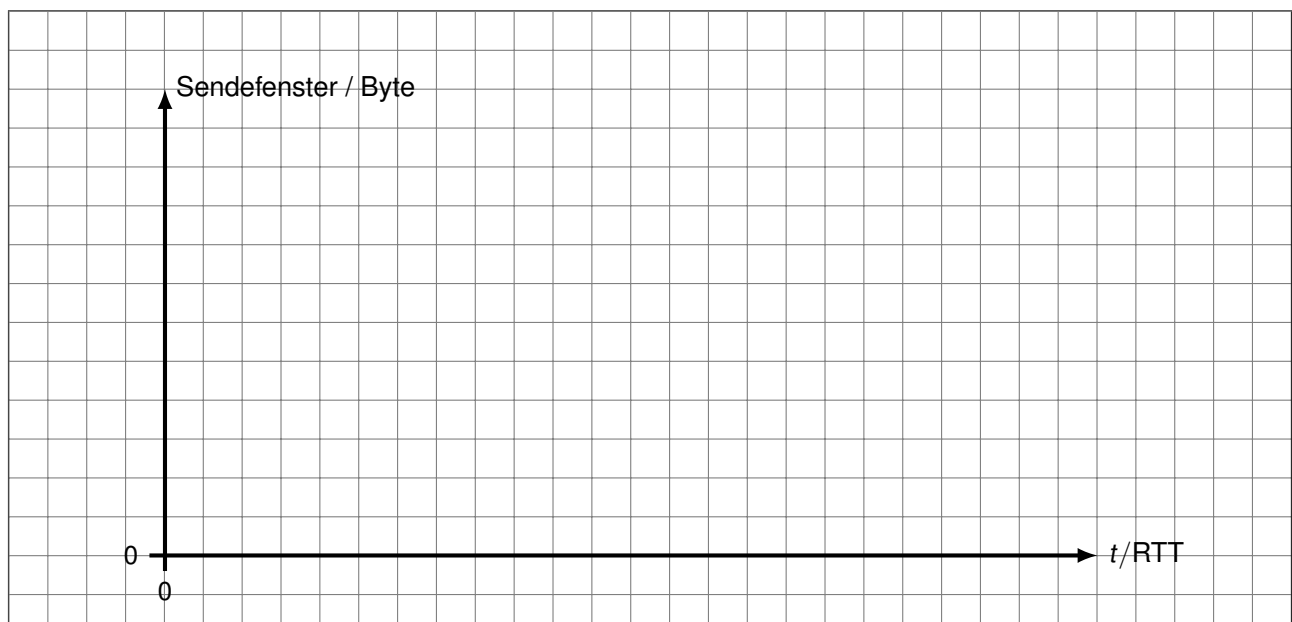
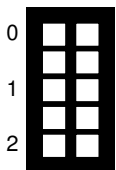
☐ nicht zutreffend

☐ Staukontrolle

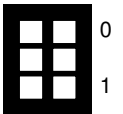
☐ Flusskontrolle

Wir gehen nachfolgend davon aus, dass die Empfangsfenster stets größer sind als die Sendefenster.

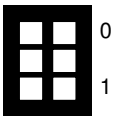
g)* Skizzieren Sie frei Hand im Lösungsfeld einen für TCP typischen Verlauf der Sendefenstergröße. Gehen Sie davon aus, dass die TCP-Verbindung zum Zeitpunkt $t = 0$ gerade aufgebaut wurde.



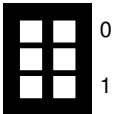
h) Markieren und benennen Sie in der Lösung von Teilaufgabe g) die beiden Phasen der Staukontrolle.



i) Was löst den Übergang zwischen den beiden Staukontrollphasen aus? (ohne Begründung)

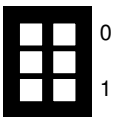


j)* Unter welchen Umständen beginnt der Staukontrollmechanismus von vorne? (ohne Begründung)

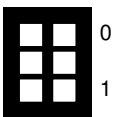


Zur Analyse der TCP-Datenrate betrachten wir den Verlauf einer zusammenhängenden Datenübertragung, bei der die erste Phase der Staukontrolle bereits abgeschlossen ist. Da das Empfangsfenster als stets ausreichend groß angenommen wird, entspricht die Größe w_s des Sendefensters stets der des Staukontrollfensters. Es treten keinerlei Verluste auf, solange das Sendefenster kleiner als ein Maximalwert x ist, also $w_s < x$. Hat das Sendefenster den Wert x erreicht, so geht genau eines der versendeten TCP-Segmente verloren.

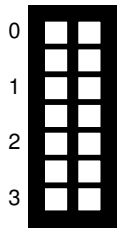
k)* Wie erkennt der Empfänger den Verlust eines Segments? (ohne Begründung)



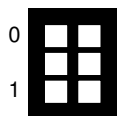
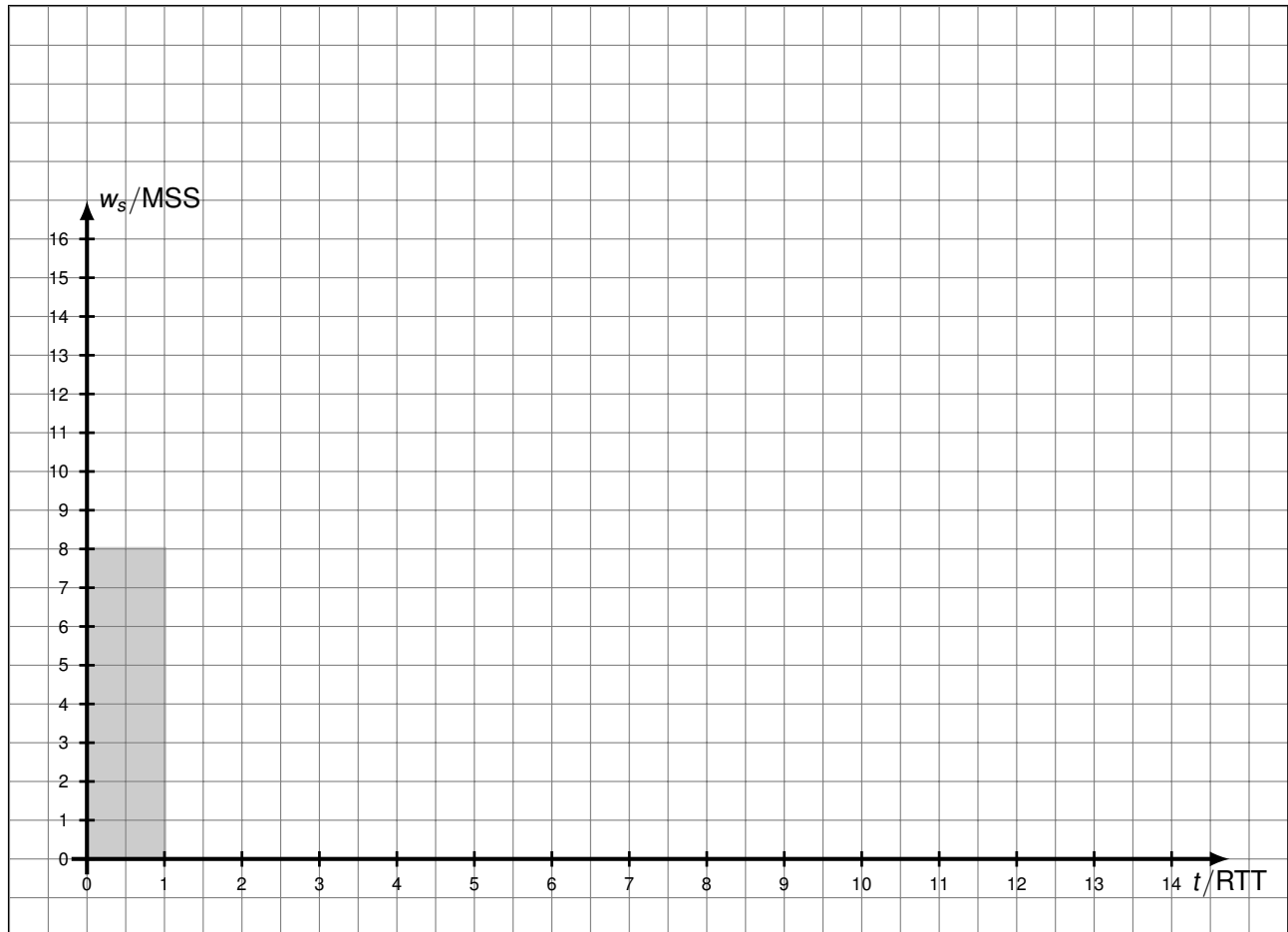
l)* Wie beeinflusst ein einzelnes verlorengesegantes Segment das Sende- bzw. Staukontrollfenster?



Als konkrete Zahlenwerte nehmen wir an, dass die maximale TCP-Segmentgröße (MSS) 1460 B und die RTT 200 ms beträgt. Die Serialisierungszeit von Segmenten sei gegenüber der Ausbreitungsverzögerung vernachlässigbar klein. Segmentverlust trete ab einer Sendefenstergröße von $w_s \geq x = 16$ MSS auf.

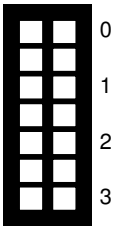


m)* Erstellen Sie ein Schaubild, in dem die aktuelle Größe des Sendefensters w_s gemessen in Vielfachen der MSS über der Zeitachse t gemessen in Vielfachen der RTT aufgetragen ist. In Ihrem Diagramm soll zum Zeitpunkt $t_0 = 0$ s gerade $w_s = x/2$ gelten. Zeichnen Sie das Diagramm im Zeitintervall $t = \{0, \dots, 14\}$.

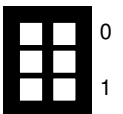


n) Bestimmen Sie die Periodendauer T zwischen der Reduktion des Sendefensters und dem nächsten Segmentverlust allgemein in Abhängigkeit von x .

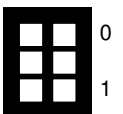
o) Leiten Sie allgemein die Anzahl N pro Periodendauer übertragener Segmente (inkl. des verlorenen Segments am Ende) allgemein in Abhängigkeit von x her. Vereinfachen Sie das Ergebnis soweit wie möglich.



p) Bestimmen Sie die Verlustrate θ allgemein und als Zahlenwert.



q) Bestimmen Sie mit Hilfe des Ergebnisses aus den Teilaufgaben n) – p) die in der betrachteten TCP-Übertragungsphase durchschnittlich erzielbare Übertragungsrate in kbit/s.



Aufgabe 4 Wireshark (18 Punkte)

Gegeben sei das Netzwerk aus Abbildung 4.1a. *PC1* hat zuvor ein Paket an *Srv* versendet. Der abgebildete Rahmen ist eine Fehlernachricht, welche daraufhin von *R* versendet wurde.

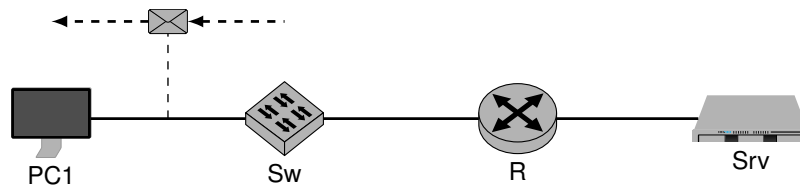


Abbildung 4.1a: Netztopologie

```

0x0000  90 e2 ba 2a 8d 97 90 e2    ba 86 dd 60 08 00 45 c0
0x0010  00 53 20 dc 00 00 40 01    d2 5b c0 a8 02 fe c0 a8
0x0020  02 64 03 00 82 42 00 00    00 00 45 00 00 37 59 84
0x0030  00 00 40 11 9c 24 c0 a8    02 64 c0 00 02 01 cc 1a
0x0040  00 35 00 23 b2 4b 86 b2    01 20 00 01 00 00 00 00
0x0050  00 00 05 67 72 6e 76 73    03 6e 65 74 00 00 1c 00
0x0060  01
  
```

Abbildung 4.1b: Ethernet-Rahmen zwischen Sw und PC1

Der Offset ist der Index in das Byte-Array und muss 0-basiert (so wie in C oder Java) angegeben werden. Geben Sie interpretierte Daten wie Adressen oder Ports jeweils in ihrer üblichen und gekürzten Schreibweise an.

Hinweis: Verwenden Sie zur Lösung die am Cheatsheet abgedruckten Header und Informationen.

Beispiel: Bestimmen Sie die Layer 2 Adresse des Empfängers.

Offset: **0** Länge **6**
 Adresse: **90:e2:ba:2a:8d:97**
 gehört zu Knoten: **PC1**

0				
1				
2				
3				

a)* Bestimmen Sie die Layer 2 Adresse des Absenders.

Offset: Länge:
 lesbares Format:
 gehört zu Knoten:

0				
1				
2				

b) Bestimmen Sie die Layer 3 Adresse des Empfängers.

Offset: Länge:
 Adresse:

0				
1				
2				

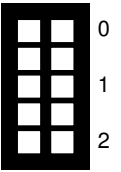
c) Bestimmen Sie die Layer 3 Adresse des Absenders.

Offset: Länge:
 Adresse:

Wiedereinstieg: Die ICMP Fehlernachricht beginnt an Index 34

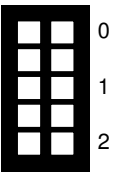
d) Bestimmen Sie Type und Code der Fehlernachricht.

Offset: Länge:
Bedeutung Type/Code:



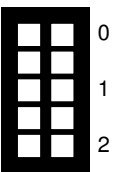
e) Bestimmen Sie die Layer 3 Adresse von *Srv* aus dem in der Fehlernachricht enthaltenen Paket.

Offset: Länge:
Adresse:



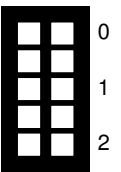
f) Bestimmen Sie das in der ursprünglichen Nachricht verwendete Layer 4 Protokoll.

Offset: Länge:
Protokoll:

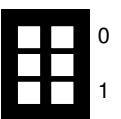


g) Bestimmen Sie den in der ursprünglichen Nachricht verwendete Zielport.

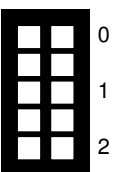
Offset: Länge:
Port:



h) Welches Anwendungsprotokoll wurde somit wahrscheinlich verwendet?



i) Argumentieren Sie, durch was für einen Fehler die Fehlernachricht ausgelöst wurde.



Aufgabe 5 IP-Fragmentierung und Path-MTU-Discovery (20 Punkte)

In dieser Aufgabe betrachten wir zunächst Fragmentierung bei IPv4. Hierzu ist die Netzwerktopologie in Abbildung 5.1 gegeben.

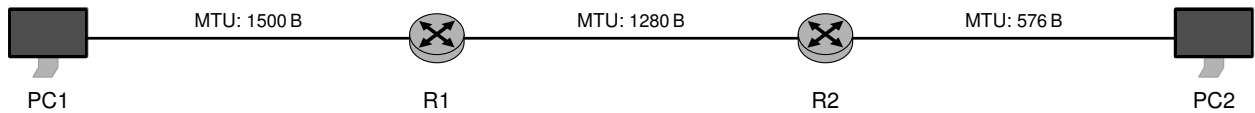


Abbildung 5.1: Netztopologie

Die Router *R1* und *R2* sind so konfiguriert, dass die beiden Hosts PC1 und PC2 miteinander kommunizieren können. Die drei Netzsegmente sind voneinander unabhängig und verwenden verschiedene Übertragungstechnologien, sodass sich die in der Abbildung ersichtlichen MTUs ergeben.

0



a)* Wie sollte die MSS in Abhängigkeit von der MTU gewählt werden (Formel mit Variablen)?

1

0



b) Geben Sie für die Formel aus Teilaufgabe a) soweit möglich typische Zahlenwerte an.

1

0



c)* Begründen Sie, ob Fragmente nochmals fragmentiert werden können.

1

0

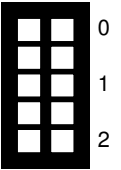


d)* An welcher Stelle im Netzwerk werden Fragmente reassembliert (Begründung)?

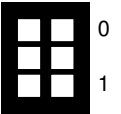
1

e)* Woran erkennt man, dass es sich um ein Fragment handelt? Geben Sie die Lösung als Pseudocode an, wobei die IP-Headerfelder jeweils als gleichnamige Variable vorliegen.

fragmented =



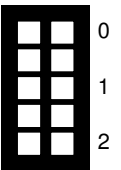
f)* Was muss bei dem *Fragment Offset* Feld im IPv4 Header berücksichtigt werden?



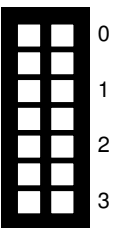
Gehen Sie nun davon aus, dass PC1 eine TCP-Verbindung zu PC2 aufgebaut hat. PC1 möchte nun 1460 B Nutzdaten über diese TCP-Verbindung an PC2 senden.

PC1 versendet diese Daten unter Berücksichtigung der benötigten minimalen IP- und TCP-Header. Der Router R1 kann das resultierende Paket nicht direkt weiterleiten und muss es zunächst fragmentieren.

g) Geben Sie die jeweilige Größe aller von R1 an R2 gesendeten IP-Pakete an.



h) Router R2 muss diese Pakete jetzt auf geeignete Weise verarbeiten. Geben Sie die jeweilige Größe aller von R2 an PC2 gesendeten IP-Pakete an.

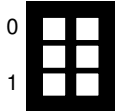


Als Alternative zur IP-Fragmentierung betrachten wir nun die Path-MTU-Discovery. Hierzu nutzen wir weiterhin die Netzwerktopologie aus Abbildung 5.1. PC1 möchte weiterhin über eine schon bestehende TCP-Verbindung Nutzdaten mit einer Länge von 1460 B an PC2 versenden.

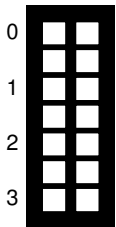
Path-MTU-Discovery wird verwendet, um Fragmentierung im Netzwerk zu verhindern. Damit auch der Sender keine IP-Fragmentierung durchführen muss, kann dieser die TCP MSS entsprechend anpassen.

Path-MTU-Discovery funktioniert wie folgt:

- Der Sender versendet zunächst Pakete der Größe der lokalen MTU.
- Diese Pakete dürfen im Netzwerk nicht fragmentiert werden.
- Wenn ein Router ein solches Paket erhält, es aber wegen der MTU im nachfolgenden Netzsegment nicht direkt weiterleiten kann, so schickt er eine ICMP Destination Unreachable, Fragmentation Needed (Type 3, Code 4) Nachricht an den Sender.
- Diese Nachricht enthält die MTU des nachfolgenden Netzsegments und der Router verwirft das ursprüngliche Paket.
- Der Sender muss die Daten erneut unter Einhaltung dieser MTU versenden. Bei TCP ist dies durch die Anpassung der MSS möglich.
- Der Sender speichert sich die MTU für nachfolgende Pakete mit demselben Ziel.



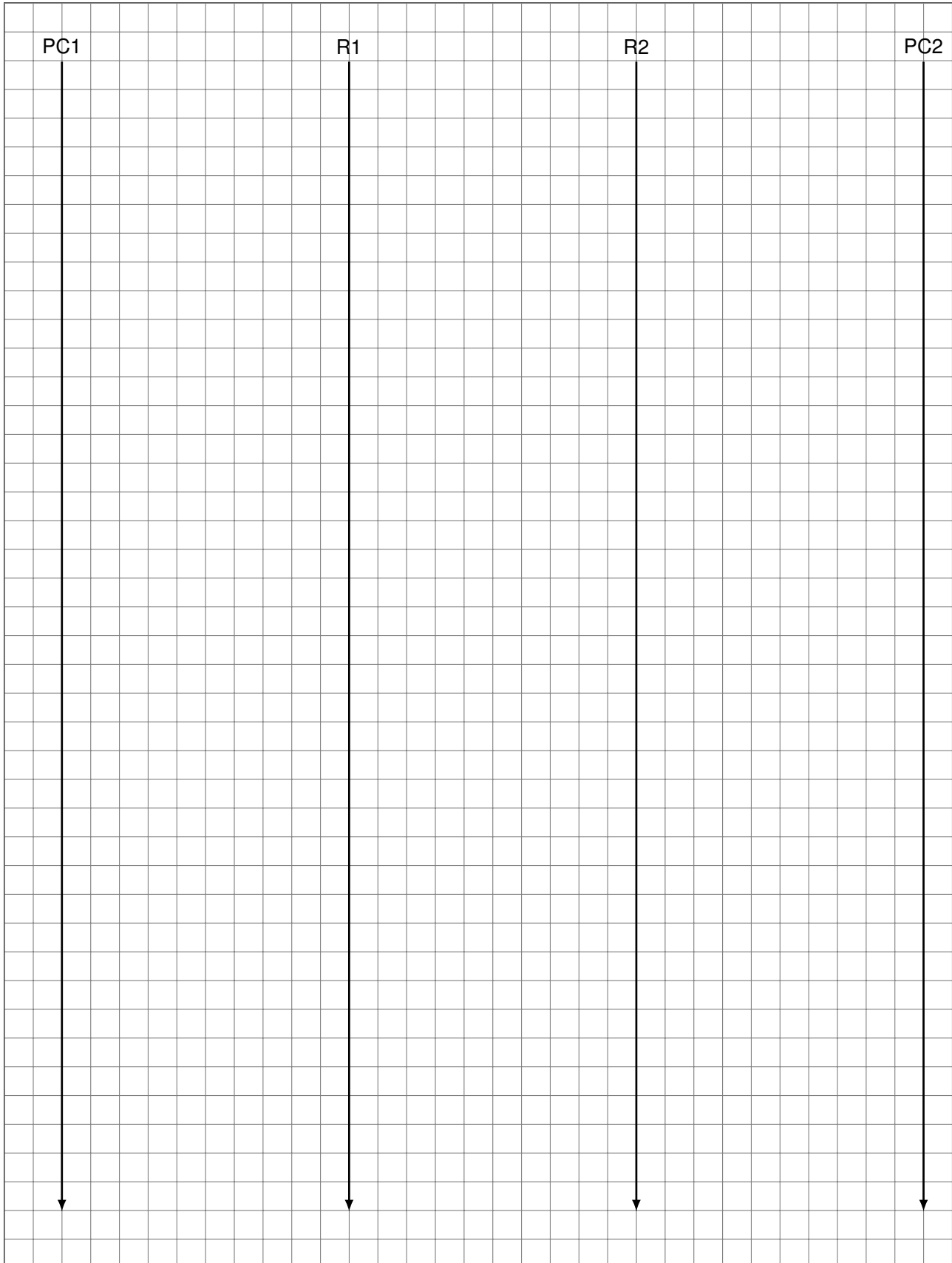
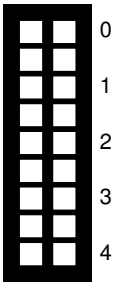
i)* Wie stellt der Sender sicher, dass seine Pakete im Netzwerk nicht fragmentiert werden dürfen?



j) Berechnen Sie die jeweilige Größe aller benötigten IP-Pakete, um TCP-Nutzdaten mit einer Länge von 1460 B von PC1 zu PC2 ohne jedwede Fragmentierung zu übertragen. Berücksichtigen Sie hierbei alle notwendigen Header in ihrer minimalen Größe.

k) Zeichnen Sie nun ein vereinfachtes Weg-Zeit-Diagramm (Serialisierungszeit und Ausbreitungsverzögerung können vernachlässigt werden) für die Path-MTU-Discovery und das Versenden der Nachricht (1460 B TCP-Nutzdaten). Geben sie bei Datenpaketen die Gesamtgröße des IP-Pakets an („IP-Paket, 128 B“). ICMP Fragmentation Needed Pakete sind als solche zu markieren und die zurückgegebene MTU ist anzugeben („ICMP Frag. needed, 256 B“).

Hinweis: Das initiale Congestion Window für TCP beträgt 4 MSS. Vernachlässigen Sie TCP-Acknowledgements und eventuelle Layer 2 Nachrichten.



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

