



HOCHSCHULE KONSTANZ TECHNIK, WIRTSCHAFT UND GESTALTUNG (HTWG)
Fakultät Informatik
Rechner- und Kommunikationsnetze
Prof. Dr. Dirk Staehle

Vorlesung Rechnernetze (AIN 5 – WS 2018/19)

Klausur (K90)

Datum: 14. Februar 2019, 9:00

Bearbeitungszeit: 90 Minuten

Gesamtpunkte: 120

Name:

Matrikelnummer:

Semester:

Hinweise:

- Falls Sie während der Klausur einen dringenden Anruf erwarten, bitte vorher anmelden. Ansonsten Handy ausschalten!
- Falls Sie sich gesundheitlich nicht in der Lage fühlen, die Klausur mitzuschreiben, bitte VOR dem Lesen der Klausuraufgaben melden.
- Auf der Teilnehmerliste unterschreiben.
- Auf jedem Blatt Name und Matrikelnummer vermerken. Am Ende alle Blätter mit einer Heftklammer zusammenfügen bzw. Angaben in den Bearbeitungsbogen legen.
- Sie können einzelne Blätter rausreißen, wenn das die Bearbeitung vereinfacht.
- Ersatzblätter sind vorhanden, falls Sie sich verschrieben haben.
- Zur Klausur sind alle Unterlagen auf Papier zugelassen. Elektronische Hilfsmittel (Handy, Laptop, Tablet, etc.) außer nicht-kommunikationsfähigen Taschenrechnern sind nicht erlaubt.
- Teilpunkte können nur für dokumentierte Zwischenergebnisse vergeben werden.

Aufgabe 1 [Paketübertragung] (30 Punkte)

Gegeben sei die in Abbildung 1 dargestellte Übertragungsstrecke, die zwei Quellen Q_A und Q_B mit zwei Senken Z_A und Z_B verbindet. Die beiden Quellen befinden sich auf einem Host und schreiben ihre Pakete in den gemeinsamen Puffer einer Netzwerkkarte, die über einen Link mit dem Netzknoten R_1 verbunden ist. Die beiden Senken Z_A und Z_B befinden sich auf zwei verschiedenen Hosts. Der Netzknoten R_3 sendet die Pakete von Q_A an Z_A und die Pakete von Q_B an Z_B . Die Kapazitäten sowie die Ausbreitungsverzögerungen der Links sind in der Abbildung angegeben. Ebenso können Sie der Grafik die Größe der Output-Buffer für die ersten drei Links entnehmen. Die Paketgröße beträgt 1250 Bytes. ACKs haben eine Größe von 50 Byte und durchlaufen die Übertragungsstrecke in umgekehrter Richtung.

Hinweis: Geben Sie alle zeitlichen Ergebnisse in Millisekunden an.

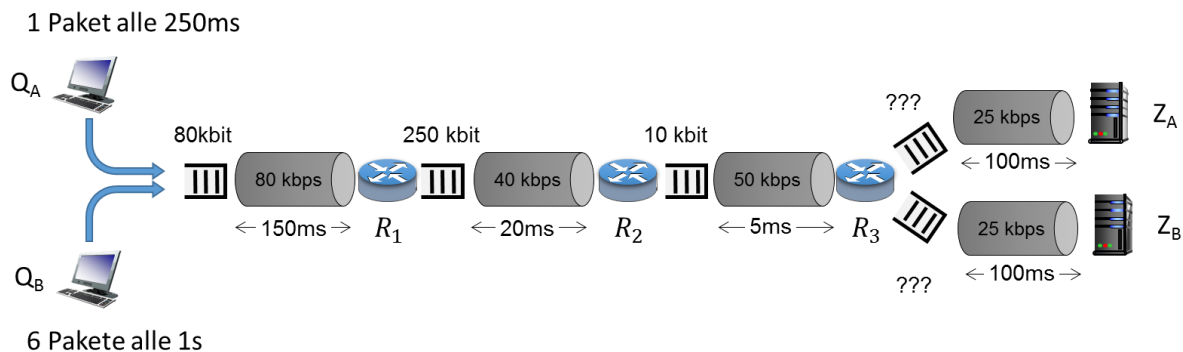


Abbildung 1: Übertragungsstrecke

- Bestimmen Sie die Ende-zu-Ende Übertragungsdauer für ein Paket. (2P)
- Über eine Zeitspanne von 5s sendet die Quelle Q_A alle 250ms ein Paket und die Quelle Q_B jede Sekunde 6 Pakete auf einmal. Die Quelle Q_B beginnt 1ms nach der Quelle Q_A mit der Übertragung.

Hinweis: Sollten die Ankunft eines Pakets und das Übertragungsende eines anderen Pakets auf den gleichen Zeitpunkt fallen, so behandeln Sie zuerst das Übertragungsende und danach die Ankunft des neuen Pakets.

- Bestimmen Sie für Link 1 (80 kbps), welche Pakete verloren gehen. (6P)

Hinweis: Bevor Sie eine Tabelle mit allen Paketen erstellen, sollten Sie darüber nachdenken, ob sie eine Tabelle benötigen und wenn ja, was genau die Tabelle enthalten soll.

- Begründen Sie, warum auf den Links 2 (40 kbps) und 3 (50 kbps) keine Pakete verloren gehen. (3P)
- Bestimmen Sie für die beiden Links 4_A (von R_3 nach Z_A) und 4_B (von R_3 nach Z_B) jeweils die Puffergröße, die notwendig ist, um Paketverluste zu vermeiden. (4P)

Name:

Matrikelnummer:

d) Auf der Transportschicht wird ein Go-Back-N-Protokoll eingesetzt. Das Sendefenster hat eine Größe von $N=5$ und der Timeout beträgt 2s.

i) Bestimmen Sie für die ersten 12 Pakete, die am Netzknoten R_3 ankommen,

(1) zu welchem Zeitpunkt Sie jeweils bei R_3 eintreffen, (2P)

(2) um welche Pakete es sich handelt.

Hinweis: Bezeichnen Sie das k -te Paket von Q_A als A_k und das j -te Paket von Q_B als B_j (3P)

(3) zu welchem Zeitpunkt diese Pakete bei Z_A bzw. Z_B eintreffen. (4P)

ii) Bestimmen Sie, welche Pakete verloren gehen und wiederholt übertragen werden.

Begründen Sie ihre Antwort. (4P)

Hinweis: Zur Beantwortung der Fragen zu Teilaufgabe d) ist keine **sehr aufwändige** Berechnung oder Tabelle notwendig, obwohl eine kleine Tabelle helfen kann. Wenn Sie keinen unaufwändigen Lösungsweg sehen, machen Sie besser mit einer anderen Aufgabe weiter.

Aufgabe 2 [HTTP] (21 Punkte)

In dieser Aufgabe wird eine Web-Seite betrachtet, die aus einem 3000 Byte großen HTML-Code und 15 Bildern mit einer Größe von jeweils 20000 Bytes besteht. Die Requests für den HTML-Code sowie für die Bilder haben eine Größe von 1500 Bytes. Im Browser und auf dem Web-Server läuft die http-Version Persistent http ohne Pipelining.

Der Download der Web-Seite erfolgt über eine Übertragungsstrecke, die durch eine Bottleneck-Link-Kapazität von 80 Mbps sowie eine Ende-zu-Ende-Ausbreitungsverzögerung von 5 ms charakterisiert ist. Die Übertragungsverzögerungen für Header sind zu vernachlässigen.

Nehmen Sie für die Aufgabe die folgenden TCP-Parameter an:

- **MSS: 500 Bytes**
- **Initial Window: 4 MSS**
- **Retransmission Timeout: 100ms**

Name:

Matrikelnummer:

- 1) Skizzieren Sie in Abbildung 2 den Verlauf der Seitenübertragung beginnend mit dem ersten Request bis zur vollständigen Übertragung des zweiten Bildes. Tragen Sie pro RTT die Anzahl und den Inhalt der vom Server bzw. Client übertragenen Segmente ein. Gehen Sie davon aus, dass alle Pakete, die gleichzeitig gesendet werden auch gleichzeitig und in der richtigen Reihenfolge ankommen. (5P)

Verwenden Sie folgende Notation

- Requests: $MO:Req$, $IO_n:Req$ bzw. $IO_{n-m}:Req$
- Segmente: $MO:Sx-y$, $IO_n:Sx-y[k \text{ Bytes}]$ bzw. $IO_n[k \text{ Bytes}]$
(Anzahl k Bytes nur, wenn Teile von Segmenten übertragen werden)
- ACKs: $MO:Ax-y$, $IO_n:Ax-y$, $MO:AREq$, $IO_n:AREq$

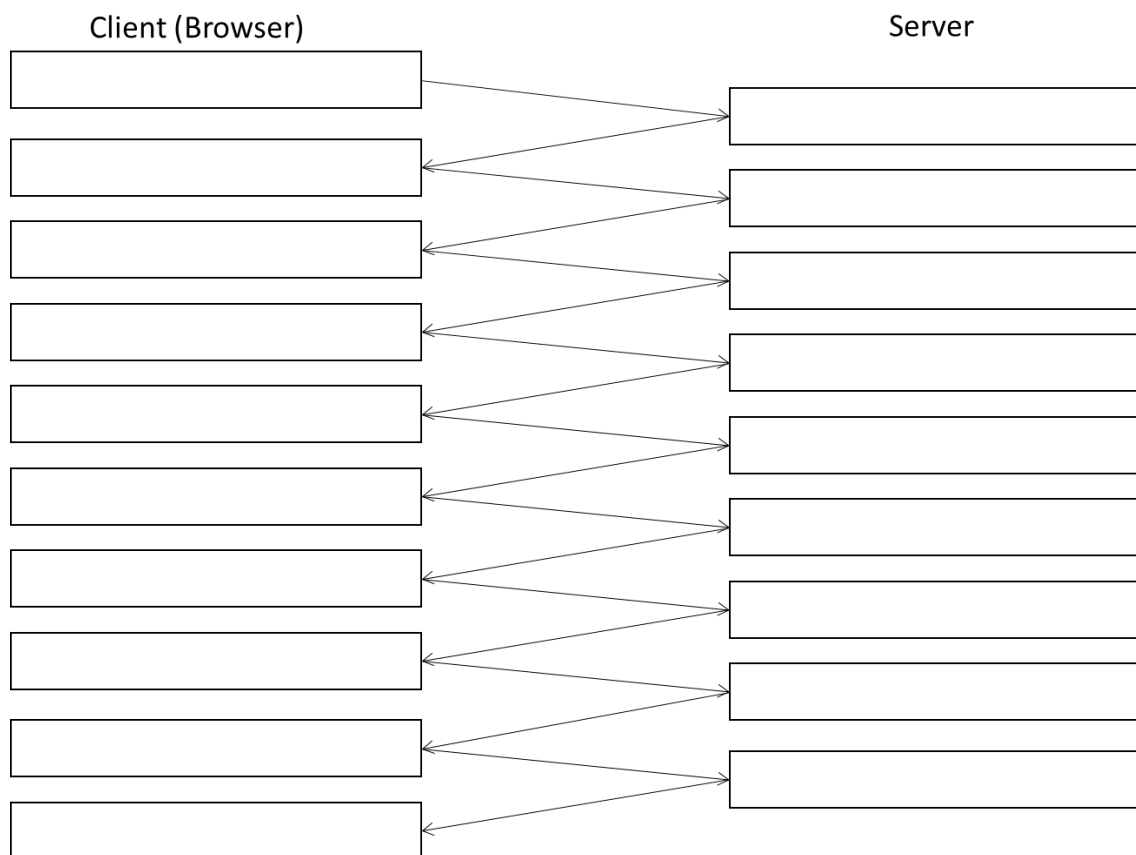


Abbildung 2: Vorlage für Skizze in Aufgabe 2.1)

Name:

Matrikelnummer:

2) Bestimmen Sie die Page-Load-Time für die gesamte Seite.

(4P)

Sie können die folgende Tabelle zur Berechnung der Übertragungsdauer verwenden:

Datenvolumen	500	1000	1500	2500	5000	8000	20000
Übertragungsverzögerung	50µs	100µs	150µs	250µs	500µs	800µs	2000µs

3) Skizzieren in Abbildung 3: Skizze für Aufgabe 2.3) Sie die Übertragung des zehnten und elften Bildes, wenn das 5te Segment des 10ten Bildes verloren geht. (6P)

4) Bestimmen Sie die Zeit, die zur Übertragung der 15 Bilder benötigt wird, wenn sechs TCP Verbindungen parallel genutzt werden. (6P)

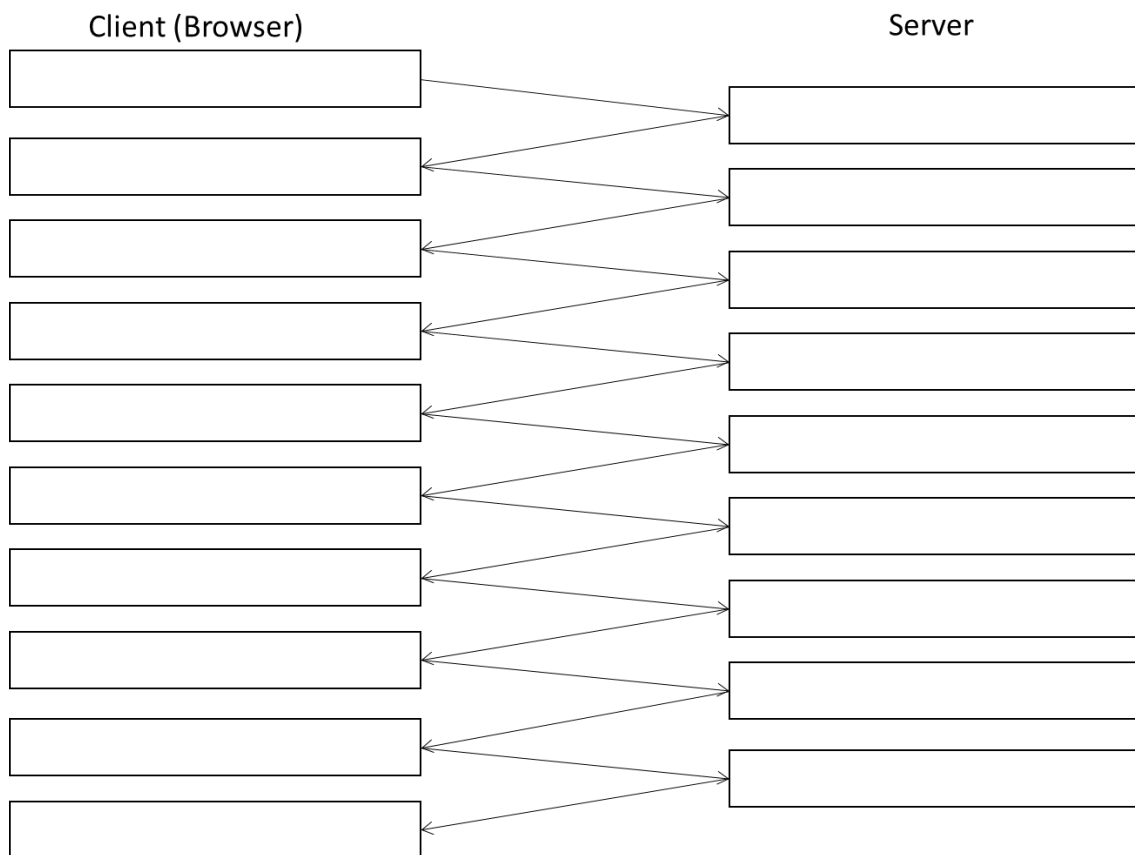


Abbildung 3: Skizze für Aufgabe 2.3)

Aufgabe 3 [Sockets](18 Punkte)

In Abbildung 4 und Abbildung 5 sind die Python-Codes „Code1.py“ und „Code2.py“ gegeben. Zunächst wird der Befehl `python code1.py 10000` und eine Sekunde später der Befehl `python code2.py 10000 20000` ausgeführt. Ein Mitschnitt der ausgetauschten Pakete ist in Tabelle 1 dargestellt.

1. Ergänzen Sie die Skripte „Code1.py“ und „Code2.py“ an den markierten Stellen. (4P)
2. Tragen Sie in die markieren Felder der ersten Spalte von Tabelle 1 ein, welche Code-Zeile welches Skripts diese Paketübertragung unmittelbar bewirkt hat. Verwenden Sie das Format $x:n$, wobei $x \in \{1,2\}$ die Nummer des Skripts ist und n die Programmzeile ist. (4P)
3. Bestimmen Sie anhand der Skripte die beiden mit X und Y markierten Ports in den Spalten D und F. (2P)
4. Tragen Sie in die markieren Felder der Spalte „Payload als ASCII-Text“ von Tabelle 1 jeweils die Payload der versendeten Pakete ein. (4P)
5. Tragen Sie in der Spalte „TCP Pakettyp“ von Tabelle 1 ein, welche Flags (ACK, SYN, FIN) gesetzt sind, wenn es sich bei dem übertragenen Paket um ein TCP Paket ohne Payload handelt. (4P)

Name:

Matrikelnummer:

Tabelle 1: Packet Trace

Auslösende Programmzeile	Source	Src Port	Destination	Dst Port	Length	Payload als ASCII-Text	TCP Paketttyp (SYN/ACK/FIN)
	127.0.0.1	54511	127.0.0.1	X	33		
	127.0.0.1	X	127.0.0.1	54511	30		
	127.0.0.1	58718	127.0.0.1	Y	52		
	127.0.0.1	Y	127.0.0.1	58718	52		
	127.0.0.1	58718	127.0.0.1	Y	40		
	127.0.0.1	Y	127.0.0.1	58718	49		
	127.0.0.1	58718	127.0.0.1	Y	40		
	127.0.0.1	58718	127.0.0.1	Y	70		
	127.0.0.1	Y	127.0.0.1	58718	40		
	127.0.0.1	Y	127.0.0.1	58718	41		
	127.0.0.1	58718	127.0.0.1	Y	40		
	127.0.0.1	58718	127.0.0.1	Y	40		
	127.0.0.1	Y	127.0.0.1	58718	40		
	127.0.0.1	Y	127.0.0.1	58718	40		
	127.0.0.1	58718	127.0.0.1	Y	40		

Name:

Matrikelnummer:

```
1 import socket
2 import sys
3 from threading import Thread
4
5 socket.setdefaulttimeout(5)
6
7 def main(lp):
8     s=socket.socket(socket.AF_INET, )
9     s.bind(('127.0.0.1',int(lp)))
10    while True:
11        try:
12            msg,addr=s.recvfrom(1500)
13            s.sendto(b'ok',addr)
14        except:
15            return
16        t=Thread(target=g,args=((addr[0],int(msg)),))
17        t.start()
18
19 def g(addr):
20     s=socket.socket(socket.AF_INET, )
21     while True:
22         try:
23             s.connect(addr)
24             break
25         except:
26             s.close()
27             return
28     while True:
29         try:
30             command=s.recv(1)
31             num=int(s.recv(2))
32             msg=s.recv(1500)
33         except:
34             return
35         if command==b'u':
36             s.send((msg*num).upper())
37         elif command==b'l':
38             s.send((msg*num).lower())
39         elif command==b'c':
40             s.close()
41             break
42
43 if __name__=='__main__':
44     main(sys.argv[1])
```

Abbildung 4: Python Code „Code1.py“

Name:

Matrikelnummer:

```
45 import socket
46 import sys
47 from threading import Thread
48
49 socket.setdefaulttimeout(5)
50
51 def main(rp,lp):
52     s=socket.socket(socket.AF_INET, )
53     s.sendto(lp.encode('utf-8'),('127.0.0.1',int(rp)))
54     while True:
55         try:
56             msg,addr=s.recvfrom(1500)
57         except:
58             s.close()
59             return
60         if msg==b'ok':
61             t=Thread(target=g,args=(int(lp),))
62             t.start()
63         elif msg==b'quit':
64             s.close()
65             return
66
67 def g(lp):
68     s=socket.socket(socket.AF_INET, )
69     s.bind(('127.0.0.1',lp))
70     s.listen(1)
71     try:
72         conn,addr=s.accept()
73         s.close()
74     except:
75         s.close()
76         return
77     conn.send(b'l05tcpudp')
78     try:
79         msg=conn.recv(1500)
80     except:
81         return
82     conn.close()
83     conn.send(b'c')
84     conn.close()
85
86 if __name__=='__main__':
87     main(sys.argv[1],sys.argv[2])
```

Abbildung 5: Python Code „Code2.py“

Name:

Matrikelnummer:

Aufgabe 4 [TCP] (18 Punkte)

Ein Sender überträgt über eine bereits geöffnete TCP Verbindung, über die auch schon einige Daten ausgetauscht wurden, eine Anzahl von $N=20$ Segmenten. Die TCP Verbindung und die Übertragungsstrecke sind durch die folgenden Parameter spezifiziert:

- Initial Window=4 Segmente, $IW=4 \cdot MSS$
- Retransmission Timeout: $10 \cdot RTT$
- ssthresh wird mit 1000 Segmenten initialisiert
- Die RTT wird von der Ausbreitungsverzögerung und nicht der Bottleneck-Bandbreite dominiert. Es kann davon ausgegangen werden, dass alle Segmente eines Sendefensters gleichzeitig aber nacheinander am Empfänger ankommen und auch die daraufhin gesendeten Acknowledgements wieder gleichzeitig aber nacheinander am Sender ankommen.

Protokollieren Sie aus Sicht des Senders den Ablauf der Übertragung, wenn die Segmente 4, 14 und 16 der übertragenen 20 Segmente verloren gehen. In der ersten Zeile von Tabelle 2 ist der aktuelle Status der TCP Verbindung eingetragen. Füllen Sie die Tabelle bis zur Bestätigung der erfolgreichen Übertragung von Segment 20 aus. Sie können auch eine eigene Tabelle anfertigen.

Betrachten Sie das in Abbildung 6 dargestellte Netz, das aus den Routern R_0 und R_1 , den Switches S_1 bis S_4 , den VLAN-Switches V_1 bis V_4 sowie diversen Hosts und Servern besteht. Das Netz ist über den Router R_0 mit dem Internet verbunden.

- Verwenden Sie folgende Nomenklatur:

- Tabelle 3: Tabelle mit Netzsegmenten und Routing-Tabelle für R₁**

[illegible]

Name:

Matrikelnummer:

Tabelle 4: Routing Tabelle und Liste mit Ziel-IP-Adressen

Route	Adress-Bereich	Subnetzmaske	IP Adressen	Liste der Ziel IP-Adressen	Route
1	0.0.0.0	0.0.0.0		23.64.40.4	
2	23.48.0.0	255.248.0.0		23.64.17.137	
3	23.64.0.0	255.255.224.0		23.53.17.137	
4	23.64.0.0	255.255.240.0		23.64.41.56	
5	23.64.16.0	255.255.248.0			
6	23.64.17.128	255.255.255.192			
7	23.64.17.137	255.255.255.255			
8	23.64.17.140	255.255.255.248			
9	23.64.25.0	255.255.255.0			
10	23.64.28.0	255.255.252.0			
11	23.64.32.0	255.255.224.0			
12	23.64.40.0	255.255.240.0			
13	23.64.40.0	255.255.248.0			
14	23.64.40.0	255.255.252.0			

Dezimal	Binär	Dezimal	Binär	Dezimal	Binär
128	1000 0000	4	0000 0100	41	0010 1001
192	1100 0000	16	0001 0000	48	0011 0000
224	1110 0000	17	0001 0001	53	0011 0101
240	1111 0000	23	0001 0111	56	0011 1000
248	1111 1000	25	0001 1001	64	0100 0000
252	1111 1100	28	0001 1100	137	1000 1001
254	1111 1110	32	0010 0000	140	1000 1100
255	1111 1111	40	0010 1000	41	0010 1001

Name:

Matrikelnummer:

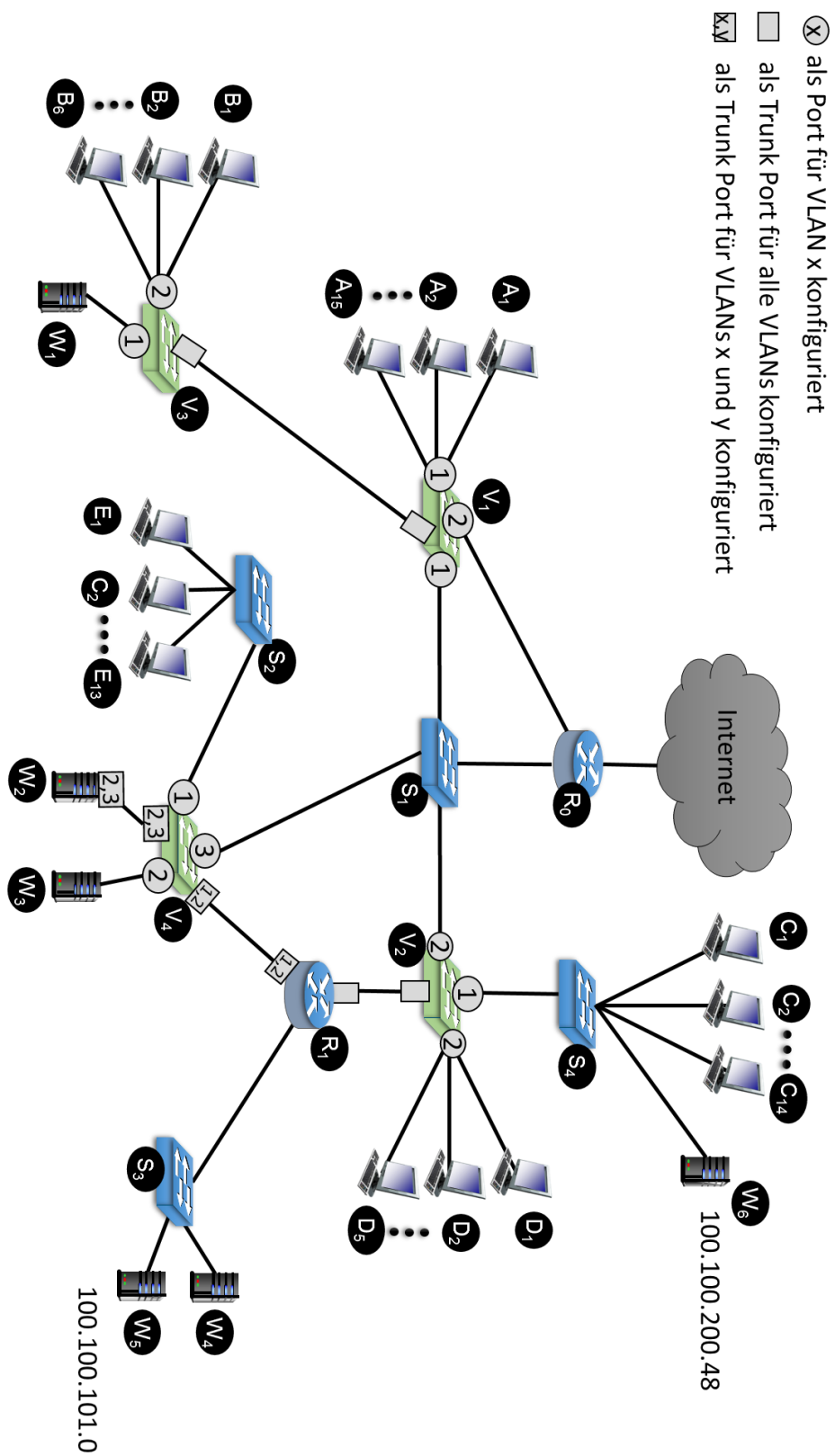


Abbildung 6: Netz mit mehreren Netzwerksegmenten

Aufgabe 6 [Wissensfragen](5 Punkte)

Entscheiden Sie für die folgenden Aussagen, ob sie richtig oder falsch sind, indem Sie sie mit einem R (richtig) oder einem F (Falsch) kennzeichnen. Für korrekte Antworten erhalten Sie einen **halben Punkt**. Für falsche Antworten erhalten Sie einen **halben Minuspunkt**. Das Gesamtergebnis dieser Aufgabe kann also auch eine negative Punktzahl sein.

1. In Distance-Vector Routing Protokollen wird die Route basierend auf der vollständigen Topologie bestimmt.
2. Symmetric NAT ist das restriktivste NAT Verfahren.
3. Ein Proxy dient dazu, häufig benötigte Daten möglichst nahe beim Client bereitzustellen.
4. Die MAC-Adresse zu einer IP-Adresse wird über das Address Retrieval Protocol (ARP) bestimmt.
5. Über das DHCP Protokoll erhält ein Client vom DHCP Server unter anderem seine IP Adresse, seinen lokalen DNS Server, seine MAC Adresse und das Default-Gateway.
6. Open Shortest Path First ist ein Interior-Gateway Routing Protokoll.
7. Die IP Adresse 224.56.78.13 ist eine Multicast-Adresse.
8. Die Broadcastadresse des Subnetzes 141.168.37.16/29 ist die 141.168.37.24.
9. Ein DHCP-Request enthält als Source-Adresse die 0.0.0.0 und als Ziel-Adresse die Adresse des DHCP-Servers.
10. Session-Cookies dienen dazu, einen Nutzer während einer begrenzten Zeit nach der ursprünglichen Authentisierung zu identifizieren.

Aufgabe 7 [Verständnisfragen] (10 Punkte)

Entscheiden Sie für die folgenden Aussagen, ob sie richtig oder falsch sind, indem Sie sie mit einem R (richtig) oder einem F (Falsch) kennzeichnen. Für korrekte Antworten erhalten Sie **einen Punkt**. Für falsche Antworten erhalten Sie **einen Minuspunkt**. Das Gesamtergebnis dieser Aufgabe kann also auch eine negative Punktzahl sein.

1. NAT Hole Punching bedeutet, dass auf einem NAT für einen Port eine feste Adresse konfiguriert wird.
2. Ein Netzwerksegment ist ein Subnetz, das nicht in kleinere Subnetze aufgeteilt werden kann. Ein Subnetz kann mehrere Netzwerksegmente umfassen, die nicht direkt über einen Router verbunden sein müssen.
3. Der Multi-Exit Discriminator dient dazu, das bevorzugte Gateway eines Autonomen Systems für ausgehenden Verkehr festzulegen.
4. Der lokale DNS Server befindet sich auf der untersten Ebene der DNS Hierarchie, er ist quasi ein Blatt der Hierarchie.
5. In TCP Cubic wird das Sendefenster nicht durch das Eintreffen von ACKs erhöht sondern ist eine Funktion der Zeit seit dem letzten Paketverlust.
6. Die verwendete TCP Version wird zwischen Client und Server beim Verbindungsaufbau vereinbart.
7. Ein ISP exportiert alle ihm bekannten Routen an seine Kunden und Peers.
8. Zwei Backbone-Router können über einen Interior-Gateway-Router verbunden sein.
9. Die http-Methode POST wird eingesetzt, um Formulardaten auf einen Web-Server zu laden, wie beispielsweise bei der Authentisierung durch Benutzername und Passwort.
10. Eine erfolgreiche Fast Recovery Phase in TCP dauert genau eine Round Trip Time.