



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

# **Vorlesung Rechnernetze (AIN 5 – SS 2019)**

## **Klausur (K90)**

**Datum: 16. Juli 2019, 11:00**

**Bearbeitungszeit: 90 Minuten**

**Gesamtpunkte: 120**

**Name:**

**Matrikelnummer:**

**Semester:**

### Hinweise:

- Falls Sie sich gesundheitlich nicht in der Lage fühlen, die Klausur mitzuschreiben, bitte melden BEVOR Sie anfangen, die Klausur zu lesen.
- Auf der Teilnehmerliste unterschreiben.
- Zur Klausur sind alle Unterlagen auf Papier zugelassen. Elektronische Hilfsmittel (Handy, Laptop, Tablet, etc.) außer nicht-kommunizierenden Taschenrechnern sind nicht erlaubt.
- Falls Sie während der Klausur einen dringenden Anruf erwarten, bitte vorher anmelden. Ansonsten Handy ausschalten!
- Nutzen Sie die ausgegebenen Papierbögen oder die Klausur selbst zur Bearbeitung. Alle anderen Abgaben auf selbst mitgebrachtem Papier werden nicht gewertet.
- Schreiben Sie ihren Namen auf alle Bögen. Legen Sie am Ende die Klausur in den oder die Bögen und schreiben Sie ihren Namen auf den äußeren Bogen.
- Sie können gerne einzelne Seiten aus der gehefteten Klausur entnehmen, wenn das die Bearbeitung erleichtert.
- Wenn Sie sich beim Ausfüllen einer Tabelle oder Zeichnung verschrieben haben, können Sie Ersatzkopien erhalten, solange der Vorrat reicht.

## Aufgabe 1 [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 oder **nicht entschiedene Aussagen** erhalten Sie einen **halben Minuspunkt**. Das Gesamtergebnis dieser Aufgabe kann also auch eine negative Punktzahl sein.

1. Das http-Protokoll ist zustandslos.
2. Ein TLS-Zertifikat verhindert eine Man-in-the-Middle-Attacke bei der Übertragung des Public Keys vom Web-Server zum Browser.
3. Das Three-Way-Handshaking beim Aufbau einer TCP Verbindung wird benötigt, um verschiedene Parameter wie beispielsweise die TCP Version zu vereinbaren.
4. Sobald ein Datagram Socket an Port 5000 gebunden wird, kann über diesen Socker nur noch mit dem Remote Port 5000 kommuniziert werden.
5. Ein Border Router ermöglicht den Austausch von IP-Paketen zwischen zwei Netzwerksegmenten.
6. Wenn ein Rechner in ein neues Netz wechselt, erhält er über DHCP unter anderem seine MAC Adresse, seine IP Adresse, seine Subnetzmaske und sein Defaultgateway.
7. Zwei Rechner, die sich in zwei verschiedenen VLANs befinden, können direkt keine IP-Pakete austauschen. Stattdessen ist eine Verbindung auf der Transportschicht notwendig.
8. Eine Netzwerkmaske dient ganz allgemein dazu, einen Adressbereich zu einer IP-Adresse zu bestimmen.
9. Beim Packet Switching enthält jedes Paket Informationen, die ein Netzknoten zur Weiterleitung des Pakets zum Zielknoten benötigt.
10. Einträge in einer Forwarding-Tabelle sind jeweils nur für eine Ziel-Adresse gültig, während ein Eintrag in einer Routing-Tabelle auch für mehrere Adressen gültig sein kann.

## Aufgabe 2 [Verständnisfragen] (12 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. Ein ISP betreibt Caching in erster Linie, um die Kosten für Transit-Traffic niedrig zu halten.
2. Ein „transparent“ Proxy wird im Browser oder im Betriebssystem konfiguriert.
3. Die MTU eines Pakets ist die maximale Größe eines IP Paket.
4. Zwei Backbone-Router können über einen Interior-Gateway-Router verbunden sein.
5. Ein Rechner bestimmt die IP-Adresse zu einem Hostnamen, indem er zunächst den DNS-Root-Server kontaktiert und sich dann durch die DNS-Hierarchie hangelt.
6. Ein Host bestimmt aufgrund der eigenen IP-Adresse und der eigenen Subnetzmaske, zu welcher IP-Adresse mit Hilfe von ARP die MAC Adresse ermittelt wird.
7. Das Spanning-Tree Verfahrens verhindert, dass IP-Pakete unendlich lange in einem Netz zirkulieren.
8. Ein ISP exportiert alle ihm bekannten Routen an seine Kunden.
9. In Link-State-Routing Protokollen teilt jeder Router allen seinen direkten Nachbarn alle seine lokalen Verbindungen mit.
10. Interior-Gateway Routing Protokolle dienen dazu, Routen zu Zielen innerhalb eines Netzwerksegments zu bestimmen.
11. Die Broadcastadresse des Subnetzes 141.168.37.27/22 ist die 141.168.37.255.
12. Über den Multi-Exit Discriminator teilt ein ISP mit AS A einem Peering-ISP mit AS B mit, über welches Gateway der Verkehr von AS A ins AS B bevorzugt fließen soll.

### Aufgabe 3 [Paketübertragung] (20 Punkte)

Gegeben sei die in Abbildung 1 dargestellte Übertragungsstrecke mit den 3 Netzknoten  $R_1$ ,  $R_2$  und  $R_3$ . Die 3 Quellen Q1, Q2, und Q3 senden Pakete zu dem Zielknoten Z. Die Kapazitäten, Ausbreitungsverzögerungen und Puffergrößen der Links, die die Netzknoten verbinden, sind in der Abbildung gegeben.

In den folgenden Teilaufgaben beträgt die Paketgröße 1500 Bytes. ACKs benötigen 15ms vom Zielknoten Z zu einem Quellknoten Qx.

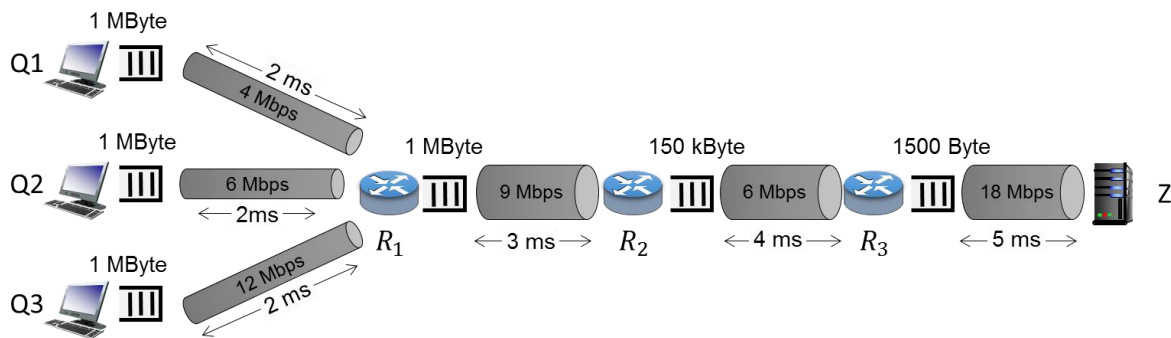


Abbildung 1: Übertragungsstrecke

- 1) Bestimmen Sie für jede der 3 Quellen die Ende-zu-Ende-Übertragungsdauer eines Pakets. (2P)
- 2) Quelle Q3 versendet Pakete mit einer Datenrate von 12,5 Mbps. Bestimmen Sie die ersten 4 Pakete, die verloren gehen. Geben Sie auch an, vor welchem Link die Pakete verloren gehen. (4P)
- 3) Betrachten Sie nun die Situation, dass zusätzlich zu Quelle Q3 auch die Quellen Q1 mit 5 Mbps und Q2 mit 8 Mbps übertragen. Quelle Q1 fängt zuerst an Pakete zu senden, Quelle Q2 startet 1ns später und Quelle Q3 noch 1ns später. Bestimmen Sie das erste Paket, das verloren geht. Geben Sie das Paket in der Form  $P_{q,k}$  an, wobei  $P_{q,k}$  das k-te von Quelle q gesendete Paket ist. (4P)
- 4) Betrachten Sie nun die Situation, dass zwischen Quelle und Senke auf der Transportschicht ein Go-Back-N-Protokoll mit einem Sendefenster von  $N=35$  Paketen läuft. Gehen Sie von einer saturierten Quelle aus, d.h. an der Quelle sind immer genügend Daten vorhanden, um das Sendefenster voll auszunutzen. Betrachten Sie nicht die Situation am Anfang der Kommunikation sondern nachdem bereits zahlreiche Pakete ausgetauscht wurden. Bestimmen Sie
  - a) die Paketverlustrate für die Quellen Q1, Q2 und Q3 (2P)
  - b) die Ende-zu-Ende Übertragungsdauer für ein Paket von Q3 inklusive Wartezeit aber ohne Verarbeitungszeit. (8P)

## Aufgabe 4 [HTTP] (22 Punkte)

In dieser Aufgabe wird eine Web-Seite betrachtet, die aus einem 6250 Byte großen HTML-Code, einem 1250 Byte großen Bild sowie 11 weiteren Bildern mit je 19000 Bytes besteht. Das 2000 Byte große Bild wird vor den größeren Bildern angefordert. Die Requests für den HTML-Code sowie für die Bilder haben eine Größe von 2500 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 0.8 Mbps sowie eine Ende-zu-Ende-Ausbreitungsverzögerung von 50 ms charakterisiert ist. Die Übertragungsverzögerungen für Header sind zu vernachlässigen.

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

- MSS: 1000 Bytes
- Initial Window: 2 MSS
- Retransmission Timeout: 500ms

- 1) Skizzieren Sie in Abbildung 2 den Verlauf der Seitenübertragung beginnend mit dem ersten Request bis zur vollständigen Übertragung des dritten 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. (6P)

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:  $MO:Ax-y$ ,  $IO_n:Ax-y$ ,  $MO:AREq$ ,  $IO_n:AREq$

- 2) Bestimmen Sie die Page-Load-Time für die gesamte Seite. (6P)

Hinweis: Die Übertragungsverzögerung für ein 1000 Byte großes Segment beträgt 10ms.

- 3) Skizzieren Sie in Abbildung 3 die Übertragung des zehnten und elften Bildes, wenn das 4te und 8te Segment des 10ten Bildes verloren gehen. Wie groß ist das Sendefenster des Web-Servers nach Empfang des letzten ACKs für ein Segment von Bild 11. (10P)

Name:

Matrikelnummer:

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:  $MO:Ax-y$ ,  $IO_n:Ax-y$ ,  $MO:AREq$ ,  $IO_n:AREq$

Übersicht der Zahlenwerte:

HTML Code	Bild1	Bild 2-12	Request	MSS	IW
6250 Bytes	2000 Bytes	19000 Bytes	1250	1000 Bytes	2 MSS

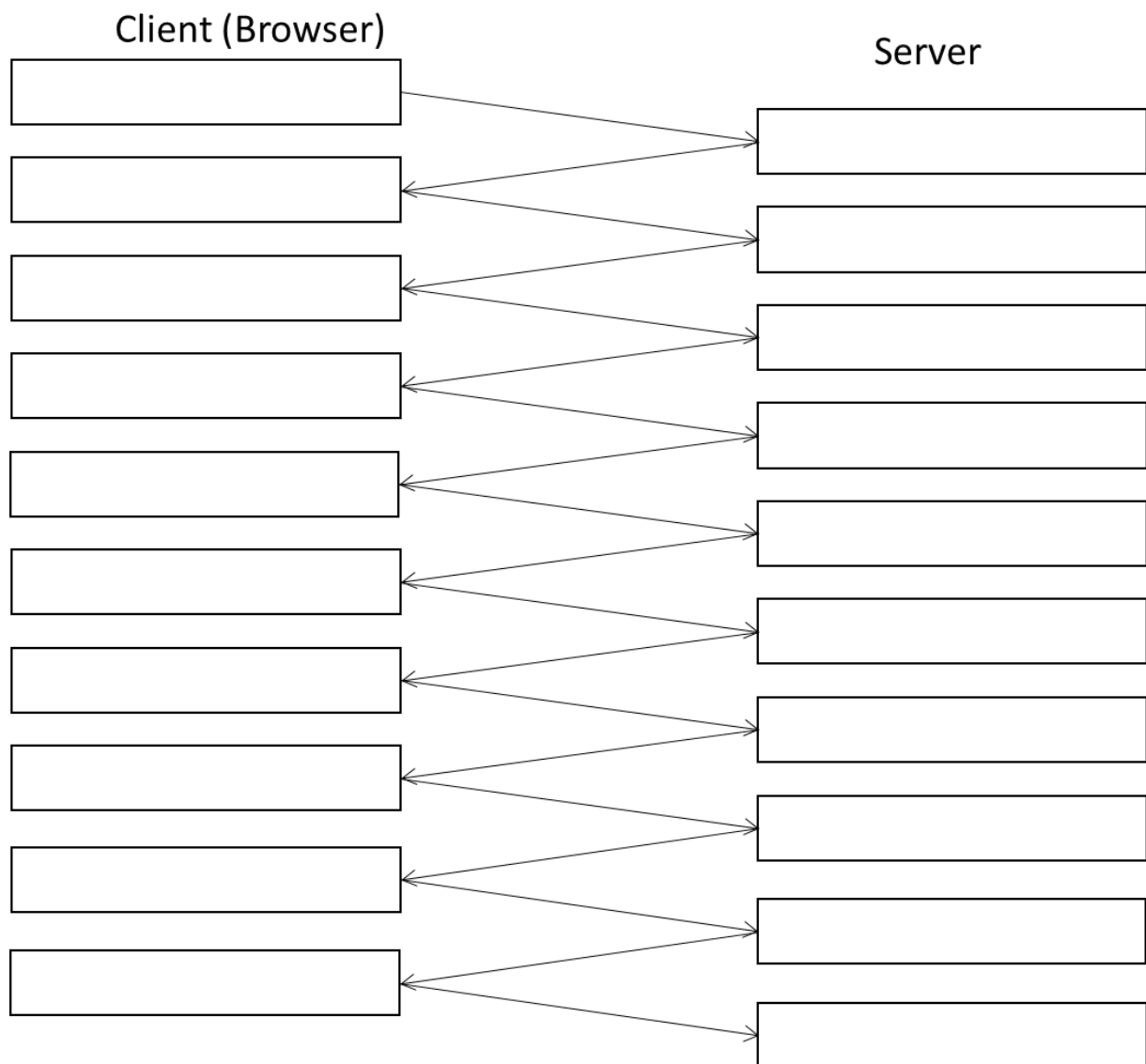


Abbildung 2: Vorlage für Skizze in Aufgabe 2.1)

Name:

Matrikelnummer:

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:  $MO:Ax-y$ ,  $IO_n:Ax-y$ ,  $MO:AREq$ ,  $IO_n:AREq$

Kurzfassung der Aufgabe:

Übertragung von Bild 10 und 11, wenn Segmente 4 und 8 von Bild 10 verloren gehen.

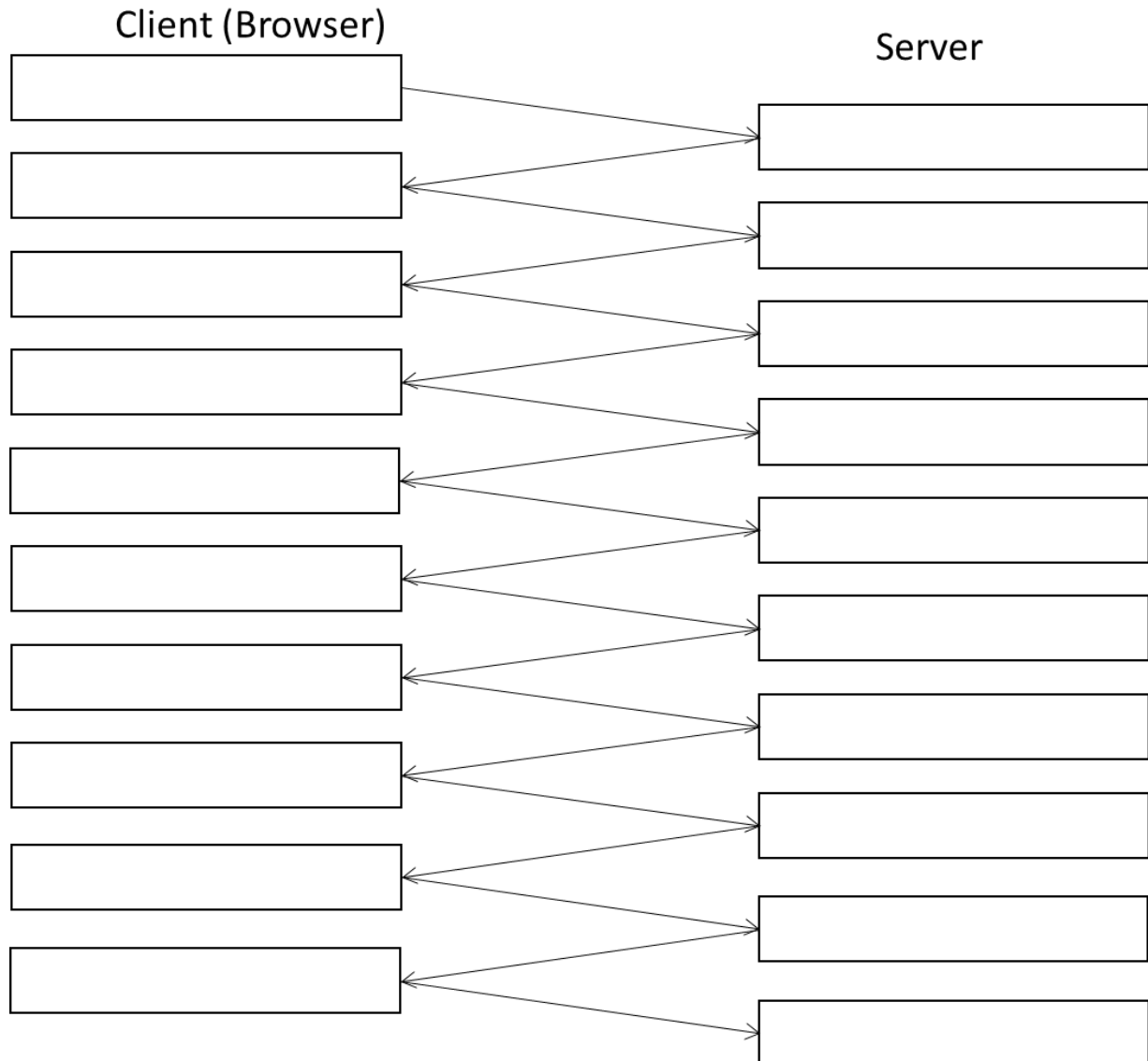


Abbildung 3: Vorlage für Skizze in Aufgabe 2.3)

## Aufgabe 5 [Python: Nachrichten und Segmente] (18 Punkte)

In Abbildung 5 und Abbildung 6 sind die Python-Codes „client.py“ für einen Client und „server.py“ für einen Server gegeben. Zunächst wird der Server und eine Sekunde später wird der Client gestartet.

- 1) Erklären Sie kurz, was die Ausführung der beiden Skripte bewirkt. Geben Sie dazu die Sequenz der Nachrichten an, die auf der Anwendungsschicht zwischen Client und Server übertragen werden. (10P)
- 2) Skizzieren Sie in Abbildung 4 die Sequenz der ausgetauschten Segmente auf der Transportschicht. Eine exakte Angabe der Zeiten ist nicht notwendig, es kommt lediglich darauf an, welche (Größe und Typ z.B. SYN, ACK, FIN, Data) und wie viele Pakete jeweils gesendet werden. (8P)

Gehen Sie von folgenden Parametern aus:

- Ausbreitungsverzögerung zwischen Client und Server: 20ms
- Bottleneck-Datenrate zwischen Client und Server: 1Gbps
- Maximum Segment Size: 1500 Bytes
- Initial Window: 2 Segmente

### Tipps zur Python-Syntax:

- 1) Der Befehl `c*n` schreibt den String/das Byte-Array `c` `n`-mal hintereinander.

Beispiel: `str=b'X'*5` → `str=b'XXXXX'`

- 2) Die Befehle `pack` und `unpack` aus dem Modul `struct` dienen dazu, einen Zahlenwert in eine Binärzahl mit bestimmtem Format zu konvertieren und umgekehrt. Das Format `'I'` steht für Unsigned Integer in Little Endian Notation.

Beispiel: `struct.pack('I',5)` → `b'\x05\x00\x00\x00'`  
`struct.unpack('I',b'A\x00\x00\x00')` → `(65,)`

- 3) Der Befehl `for i in range(n)`: entspricht einer FOR-Schleife, in der die Laufvariable `i` die Werte `0` bis `n-1` annimmt.
- 4) Der Befehl `break` unterbricht die Ausführung einer Schleife. Das Programm wird mit dem ersten Befehl nach der Schleife fortgesetzt.



Name:

Matrikelnummer:

Client

Server



Abbildung 4: Vorlage für Skizze der Paketsequenz  
(Die „schräge“ Linie kennzeichnet die erste Übertragung von Client zu Server)

**Abbildung 5: Python Code „client.py“ für Client**

```
1 import socket
2 import time
3 import struct
4
5 socket.setdefaulttimeout(10)
6
7 def start_task(sock, todo):
8     while True:
9         cmd=sock.recv(1000)
10        if cmd==b'Name?':
11            sock.send(todo[0])
12        elif cmd==b'Data':
13            for i in todo[1]:
14                sock.send(struct.pack('I',i[0])+i[1]) # Tipp 2
15                for k in range(i[0]): # Tipp 3
16                    str=sock.recv(1)
17        elif cmd==b'Fini':
18            sock.send(b'Bye!!!')
19            sock.close()
20            break # Tipp 4
21        elif cmd==b'':
22            break # Tipp 4
23
24 sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
25
26 tries=0;
27 while tries<5:
28     try:
29         sock.connect(('127.0.0.1',50000))
30         start_task(sock,(b'Client1',[ (5000,b'A'), (20000,b'B') ]))
31         break # siehe Python Hinweis 4
32     except socket.timeout:
33         tries+=1;
```

**Abbildung 6: Python Code „server.py“**

```
1 import socket
2 import struct
3 import time
4
5 def start_task(sock,cmd):
6     sock.settimeout(0.1)
7     for i in cmd:
8         sock.send(i)
9         if i==b'Name?':
10             name=sock.recv(1000).decode()
11         elif i==b'Data':
12             while True:
13                 try:
14                     data=sock.recv(4)
15                     n=struct.unpack('I',data)[0] # Tipp 2
16                     c=sock.recv(1)
17                     sock.send(c*n) # Tipp 1
18                 except socket.timeout:
19                     sock.send(b'Fini')
20                     bye=sock.recv(1000)
21                     if bye==b'Bye!!!':
22                         sock.close()
23                         break # Tipp 4
24
25 def start_server(addr):
26     sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
27     sock.bind(addr)
28     sock.listen(1)
29     while True:
30         try:
31             conn, addr = sock.accept()
32             start_task(conn,(b'Name?',b'Data'))
33         except socket.timeout:
34             break # Tipp 4
35
36 socket.setdefaulttimeout(5)
37
38 sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
39 start_server(('127.0.0.1',50000))
```

## Aufgabe 6 [TCP] (16 Punkte)

In den folgenden beiden Teilaufgaben ist jeweils der Zustand einer bereits geöffneten TCP Verbindung gegeben. Weitere Segmente werden übertragen und es können Paketverluste auftreten. Vervollständigen Sie die angegebenen Tabellen, bis alle Segmente und ACKs angekommen sind. Verwenden Sie die folgenden Parameter:

- Initial Window=3 Segmente,  $IW=3 \cdot MSS$
  - Retransmission Timeout:  $10 \cdot RTT$
  - ssthresh wird mit 500 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.
- 1) Vervollständigen Sie in Tabelle 1 den Ablauf der Übertragung von 60 Segmenten. Nach den 60 Segmenten werden keine weiteren Segmente versendet. Es gehen keine Segmente verloren. (6P)
  - 2) Vervollständigen Sie in Tabelle 2 den Ablauf der Übertragung von 40 Segmenten. Nach den 40 Segmenten werden keine weiteren Segmente versendet. Die Segmente 21 bis 24 gehen verloren. (10P)

Name:

Matrikelnummer:

### Tabelle 1: TCP Aufgabe 1)

[illegible]

Matrikelnummer:

### Tabelle 2: TCP Aufgabe 2)

[illegible]

## Aufgabe 7 [LAN] (7 Punkte)

In Abbildung 7 ist ein LAN dargestellt. Das LAN besteht aus den 6 VLAN Switches VS A bis VS F und den 3 „normalen“ Switches  $S_1$  bis  $S_3$ . An den VLAN Switches sind jeweils 3 Hosts angeschlossen. Ein Kreis mit der Zahl  $k$  bedeutet, dass der Port für VLAN  $k$  konfiguriert ist. Ein Rechteck ohne Zahl bedeutet, dass der Port als Trunk-Port für alle VLANs konfiguriert ist.

Bestimmen Sie, welche Hosts jeweils zu dem gleichen VLAN gehören.

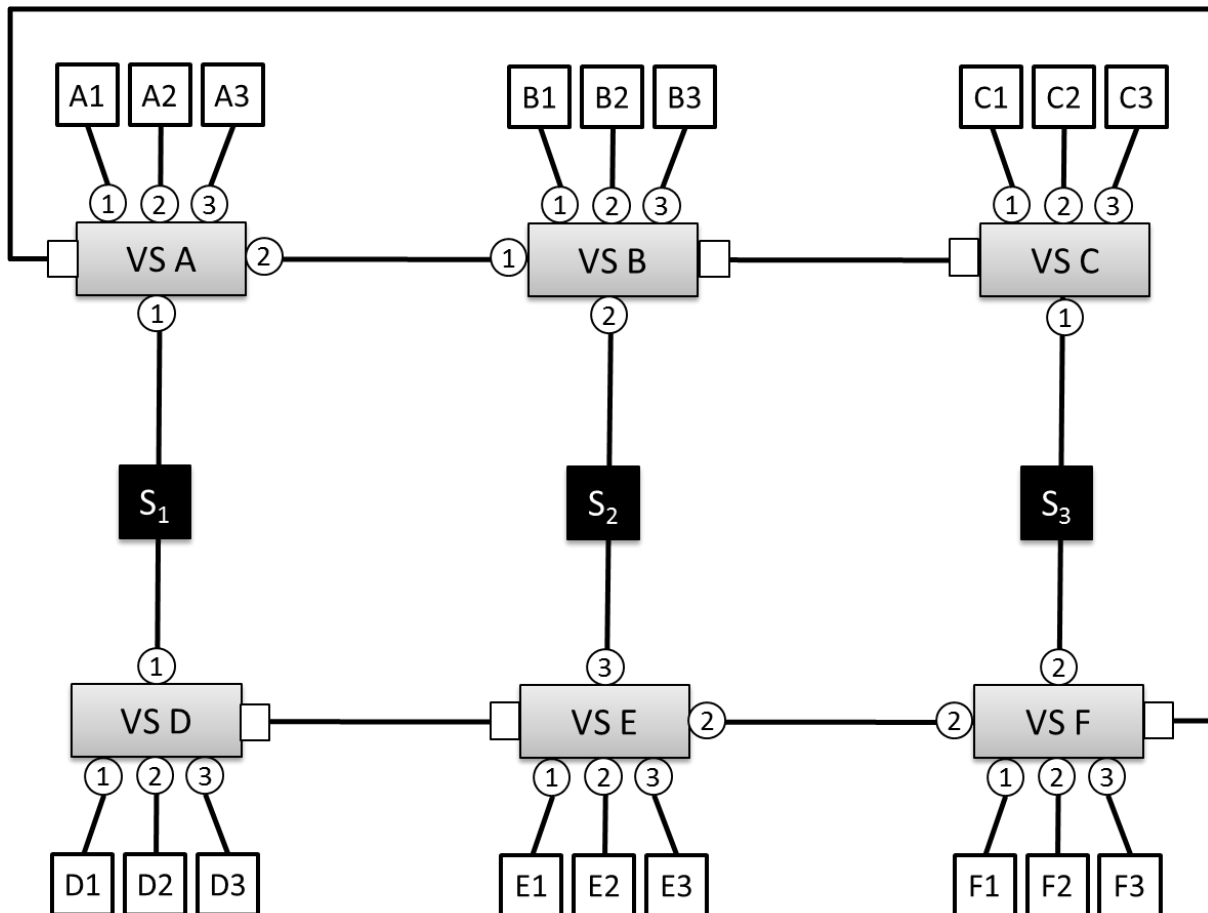


Abbildung 7: LAN mit eingebetteten VLANs

**Aufgabe 8 [IP-Adressen] (20 Punkte)**

- 1) In einem Netz befinden sich 4 Netzwerksegmente, in denen 3, 5, 14 und 15 IP-Adressen für Hosts und Router-Interfaces benötigt werden. Ihnen steht das Subnetz 100.100.100.128/26 zur Verfügung. Teilen Sie den Netzwerksegmenten jeweils einen Adressbereich zu und geben Sie die Subnetzmaske an. (3P)
- 2) In zwei Netzwerksegmenten werden jeweils 6 IP-Adressen für Hosts und Router-Interfaces benötigt. Bestimmen Sie jeweils das kleinstmögliche Netzwerkssegment, wenn die folgenden IP-Adressen bereits vergeben sind: (3P)
  - a) 100.100.100.65, 100.100.100.67, 100.100.100.69
  - b) 100.100.100.32, 100.100.100.39, 100.100.100.63
- 3) Bestimmen Sie für das Netz in Abbildung 8 alle Netzwerksegmente und geben Sie die Anzahl der IP-Adressen (ohne Netzwerk- und Broadcastadresse) an, die in diesen Netzwerksegmenten benötigt werden. (7P)
- 4) Stellen Sie für den Router  $R_2$  in Abbildung 8 eine Routing-Tabelle auf. Führen Sie für die Netzwerksegmente aus Teilaufgabe 3) Bezeichnungen ein und verwenden Sie diese anstelle der Adressbereiche in der Routing-Tabelle. (7P)

Verwenden Sie die Notation  $R/I$  für das Interface, das von einem Router  $R$  zu einem Netzknoten  $I$  führt. Falls über dieses physikalische Interface mehrere virtuelle Interfaces (VLANs) laufen, spezifizieren Sie das virtuelle Interface als  $R/I [V]$ , wobei  $V$  die VLAN ID ist.



Name:

Matrikelnummer:

- ⊗ als Port für VLAN x konfiguriert
- als Trunk Port für alle VLANs konfiguriert
- ⊠ als Trunk Port für VLANs x und y konfiguriert

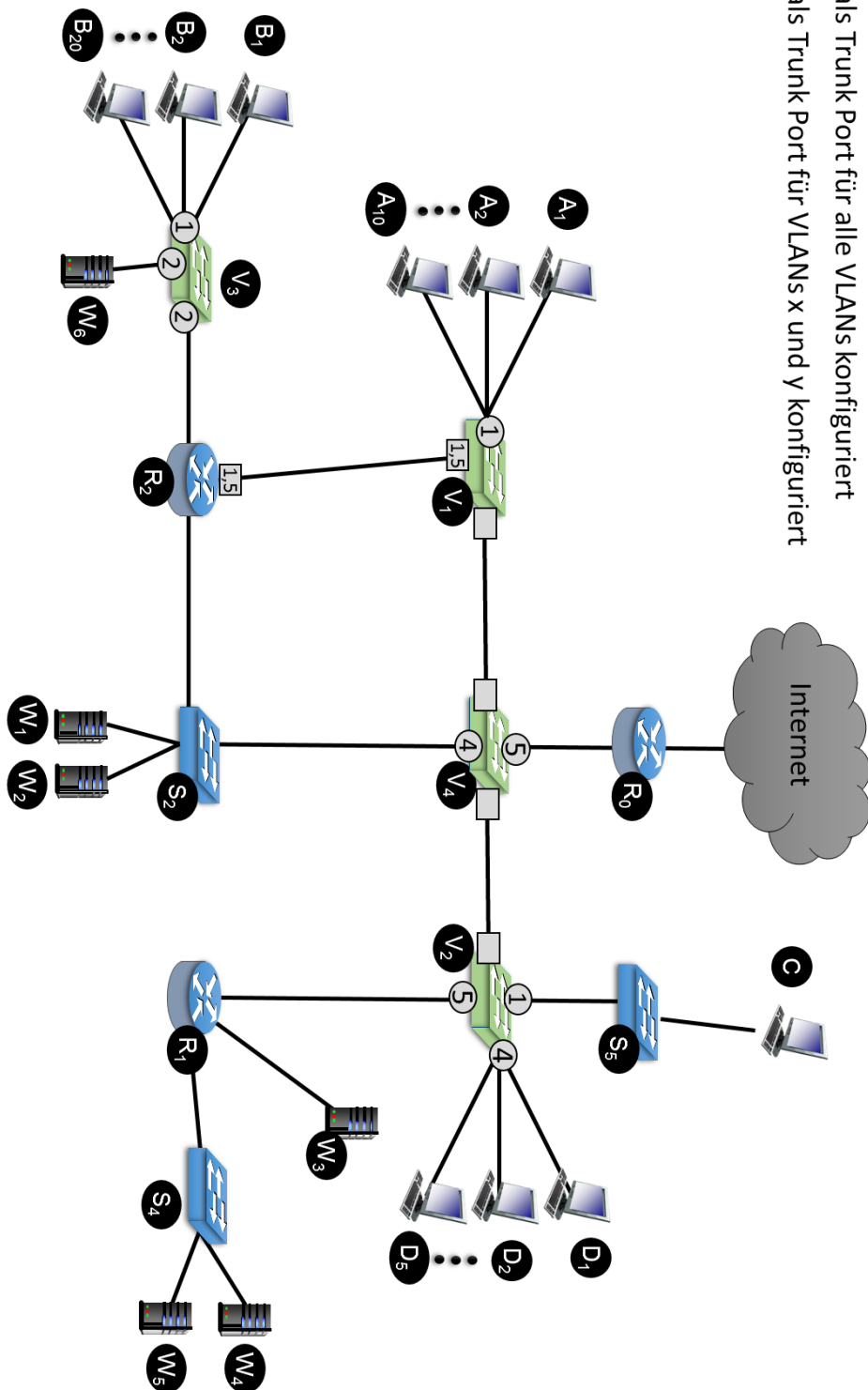


Abbildung 8: Netz mit mehreren Netzwerksegmenten

Name:

Matrikelnummer: