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

Vorlesung
Rechnernetze
(AIN 5 – WS 2017/18)

Klausur (K90)

Datum: 15. Februar 2018, 9:00
Bearbeitungszeit: 90 Minuten
Gesamtpunkte: 90

Name:
Matrikelnummer:
Semester:

Hinweise**:**

- Zur Klausur sind alle Unterlagen auf Papier zugelassen. Elektronische Hilfsmittel (Handy, Laptop, Tablet, etc.) außer einfachen Taschenrechnern sind nicht erlaubt.
- 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 JETZT vor dem Lesen dieser Aufgabenstellung melden.
- Auf jedem Blatt Name und Matrikelnummer vermerken. Am Ende alle Blätter in den Bearbeitungsbogen legen.
- Sie können einzelne Seiten zur besseren Bearbeitung heraustrennen
- Geben Sie bei komplexeren Aufgaben kurze Begründungen oder Rechenschritte an. Für eine falsche Antwort gibt es ohne Zwischenrechnung keine Teilpunkte.
- Auf der Teilnehmerliste unterschreiben.

Name: Matrikelnummer:

Aufgabe 1 [Paketübertragung] (20 Punkte)

Gegeben sei die in Abbildung 1 dargestellte Übertragungsstrecke von einer Quelle Q zu einem Ziel Z, die über drei Router R_1, R_2 und R_3 verläuft. Die Link-Kapazitäten sowie die Ausbreitungsverzögerungen der vier Links sind in der Abbildung angegeben. Ebenso können Sie der Grafik die Größe der Output-Buffer für alle Links entnehmen. Jedes Paket enthält 1125 Bytes.

Hinweis: Geben Sie alle zeitlichen Ergebnisse in Millisekunden an.

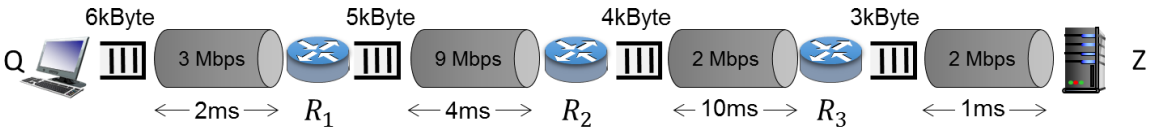

Abbildung 1: Übertragungsstrecke

1) Bestimmen Sie die Ende-zu-Ende Übertragungsdauer für ein Paket. (2P)
2) Die Quelle versendet Pakete mit einer Rate von 500 Paketen pro Sekunde. Bestimmen Sie für jeden Link, welche Pakete verloren gehen. Betrachten Sie dazu die ersten 20 gesendeten Pakete. (9P)
3) Um die Anzahl der Paketverluste zu reduzieren, wird auf der Transportschicht ein Go-Back-N Protokoll mit einem Sendefenster von N=15 Paketen eingesetzt. Der Timeout beträgt 100ms, die Übertragung eines ACKs von Z nach Q benötigt 15ms. Bestimmen Sie die Ankunftszeit des letzten Pakets, wenn die Quelle die Übertragung nach 20 Paketen beendet. Gehen Sie weiterhin davon aus, dass die Quelle Pakete mit einer Rate von 500 Paketen pro Sekunde überträgt. (9P)

Aufgabe 2 [Sockets](8 Punkte)

In dieser Aufgabe geht es darum, den Zusammenhang von Socket-Befehlen und übertragenen Pakete zu verstehen. Dazu ist in Abbildung 1Abbildung 2 ein Python-Code dargestellt. Dieser Python Code wird auf einem Rechner zweimal mit unterschiedlichen Konfigurationen nacheinander ausgeführt. Bei der ersten Ausführung (A) ist My_IP=127.0.0.2 und Remote_IP=127.0.0.1. Bei der zweiten Ausführung (B) ist My_IP=127.0.0.1 und Remote_IP=127.0.0.2.

Ein Mitschnitt der ausgetauschten Pakete ist in Tabelle 1 dargestellt.

1. Tragen Sie in die erste Spalte von Tabelle 1 ein, welche Code-Zeile bei welcher Ausführung des Codes (A oder B) diese Paketübertragung bewirkt hat, sofern diese explizite Zuordnung möglich ist.
2. Tragen Sie in die zweite Spalte von Tabelle 1 ein, welche „blockierende“ Befehlszeile bei welcher Ausführung des Codes (A oder B) erfolgreich (ohne Fehler) „beendet“ wird, wenn das Paket empfangen wird. Auch hier soll natürlich nur dann ein Eintrag erfolgen, wenn der Empfang dieses Pakets die „Fertigstellung“ eines Socket-Befehls bewirkt.

Hinweis: Die Eintragungen in die Tabelle sollen also beispielweise die Form A10 haben, wenn Programmzeile 10 der ersten Ausführung (A) des Codes die Übertragung des Pakets bewirkt hat, oder B10 wenn dementsprechend die zweite Ausführung (B) des Codes die Übertragung des Pakets bewirkt hat.

Name: Matrikelnummer:

Abbildung 2 Code Listing

```
1   import socket
2   socket.setdefaulttimeout(30)
3
4   My_IP = '127.0.0.1'
5   My_PORT = 50000
6   Remote_IP='127.0.0.2'
7   Remote_PORT=50000
8
9   def start_task(sock,message):
10      sock.send(message.encode('utf-8'))
11      msg=sock.recv(1024)
12      sock.close()
13
14  def start_server():
15      sock=socket.socket(socket.AF_INET,socket.SOCK_STREAM)
16      sock.bind((My_IP, My_PORT))
17      sock.listen(1)
18      try:
19          conn, addr = sock.accept()
20          start_task(conn,"Thx for connecting!!!")
21      except socket.timeout:
22          pass
23
24  sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
25  try:
26      sock.connect((Remote_IP, Remote_PORT))
27      start_task(sock,"Thx for accepting!!!");
28  except socket.error:
29      start_server()
```

Name: Matrikelnummer:

Tabelle 1: TCPdump

Übertragung ausgelöst von Programmzeile	Empfang bewirkt "Fertigstellung" von Programmzeile	No.	Time	Source	Destination	Protocol	Length	Src Port	Dst Port	Info
		1	1,340	127.0.0.2	127.0.0.1	TCP	52	56835	50000	56835 > 50000 [SYN] Seq=0
		2	1,340	127.0.0.1	127.0.0.2	TCP	40	50000	56835	50000 > 56835 [RST, ACK] Seq=1 Ack=1
		3	1,842	127.0.0.2	127.0.0.1	TCP	52	56835	50000	[TCP Spurious Retransmission] 56835 > 50000 [SYN] Seq=0
		4	1,842	127.0.0.1	127.0.0.2	TCP	40	50000	56835	50000 > 56835 [RST, ACK] Seq=1 Ack=1
		5	2,342	127.0.0.2	127.0.0.1	TCP	48	56835	50000	[TCP Spurious Retransmission] 56835 > 50000 [SYN] Seq=0
		6	2,342	127.0.0.1	127.0.0.2	TCP	40	50000	56835	50000 > 56835 [RST, ACK] Seq=1 Ack=1
		7	3,646	127.0.0.1	127.0.0.2	TCP	52	56837	50000	56837 > 50000 [SYN] Seq=0
		8	3,646	127.0.0.2	127.0.0.1	TCP	52	50000	56837	50000 > 56837 [SYN, ACK] Seq=0 Ack=1
		9	3,646	127.0.0.1	127.0.0.2	TCP	40	56837	50000	56837 > 50000 [ACK] Seq=1 Ack=1
		10	3,646	127.0.0.1	127.0.0.2	TCP	60	56837	50000	56837 > 50000 [PSH, ACK] Seq=1 Ack=1 Len=20
		11	3,646	127.0.0.2	127.0.0.1	TCP	40	50000	56837	50000 > 56837 [ACK] Seq=1 Ack=21 Len=0
		12	3,646	127.0.0.2	127.0.0.1	TCP	61	50000	56837	50000 > 56837 [PSH, ACK] Seq=1 Ack=21 Len=21
		13	3,646	127.0.0.1	127.0.0.2	TCP	40	56837	50000	56837 > 50000 [ACK] Seq=21 Ack=22 Len=0
		14	3,646	127.0.0.1	127.0.0.2	TCP	40	56837	50000	56837 > 50000 [FIN, ACK] Seq=21 Ack=22 Len=0
		15	3,646	127.0.0.2	127.0.0.1	TCP	40	50000	56837	50000 > 56837 [FIN, ACK] Seq=22 Ack=22 Len=0
		16	3,646	127.0.0.1	127.0.0.2	TCP	40	56837	50000	56837 > 50000 [ACK] Seq=22 Ack=23 Len=0

Name: Matrikelnummer:

Aufgabe 3 [HTTP] (14 Punkte)

In dieser Aufgabe wird eine Web-Seite betrachtet, die aus einem 4500 Byte großen HTML-Code und fünf 9000 Byte großen Bildern besteht. Die Requests für den HTML-Code sowie für die Bilder haben eine Größe von 300 Bytes. Der Download der Web-Seite erfolgt über eine Übertragungsstrecke, die durch eine Bottleneck-Link-Kapazität von 2,4 Mbps sowie eine Ende-zu-Ende-Ausbreitungsverzögerung von 50 ms charakterisiert ist. Die Übertragungsverzögerungen für Header sind zu vernachlässigen.
Die Übertragung der Web-Seite erfolgt mit Persistent http über zwei parallele TCP Verbindungen zum Web-Server, die beide bereits vor dem Senden des ersten GetRequests aufgebaut werden.
Für beide TCP Verbindungen beträgt die Maximum Segment Size (MSS) 1500 Bytes und das Initial Window 3 MSS. Der Retransmission Timout wird jeweils auf das Vierfache der Round-Trip-Time gesetzt, die vom Versenden des SYN-Pakets bis zum Eintreffen der Bestätigung gemessen wird.

1. Skizzieren Sie den Verlauf der Seitenübertragung. Sie können in der Skizze entweder eine Linie pro Segment oder eine Linie für mehrere Segmente nutzen. Kennzeichnen Sie die beiden TCP-Verbindungen z.B. durch Verwendung unterschiedlicher Farben oder zumindest durch die Beschriftung der in einer Linie übertragenen Segmente. Beispiel: B1/S1-4/T1 für Segmente 1-4 von Bild 1 in TCP Verbindung 1. (8P)
2. Bestimmen Sie die Page-Load-Time. (6P)

Aufgabe 4 [TCP] (18 Punkte)

Ein Sender überträgt über eine TCP Verbindung, über die bereits einige Daten übertragen wurden, eine Anzahl von N=75 Segmenten. Die TCP Verbindung und die Übertragungsstrecke sind durch die folgenden Parameter spezifiziert:

- Initial Window=3 Segmente, IW=3*MSS
- Retransmission Timeout: 7*RTT
- ssthresh wird mit 65535 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 das 20te und 30te von den 75 übertragenen Segmenten verloren geht. Nutzen Sie dazu Tabelle 2, in der in der ersten Zeile der aktuelle Status der TCP Verbindung eingetragen ist.

Tabelle 2 Ablauf einer TCP Verbindung

Sender									Empfänger	
RTT	ACKs An	FlightSize 1	DupACK	ssthresh	cwnd	Segmente Ab	FlightSize 2	CA Segment	Segmente AN	ACKs Ab
0	x	0	0	16	25					

Aufgabe 5 [IP-Adressen] (15 Punkte)

Betrachten Sie das in Abbildung 3 dargestellte Netz, das aus vier Routern R_0 bis R_3, fünf einfachen Switches S_1 bis S_5, den beiden VLAN Switches V_1und V_2 sowie diversen Hosts und Servern besteht. Auf den VLAN-Switches sind zwei VLANs konfiguriert. Die Zugehörigkeit der Ports zu den beiden VLANs ist über Dreieck-, Kreis- und Rechteck-Symbole gekennzeichnet, die in der Legende erklärt werden. Das Netz ist über den Router R_0 mit dem Internet verbunden.

1) Kennzeichnen bzw. notieren Sie alle Netzwerksegmente und bestimmen Sie, wie viele Internet-Adressen in dem jeweiligen Netzwerksegment vergeben sind. (5P)

2) Sie haben für dieses Netz die Adressbereiche 200.200.100.64/27 und 200.200.103.64/26 zur Verfügung. Weisen Sie den Netzwerksegmenten IP-Adressen und Subnetzmasken zu. (3P)

3) Stellen Sie für den Router R_3 eine Routing-Tabelle auf. Im Falle mehrerer gültiger Möglichkeiten für eine Route dürfen Sie selbst entscheiden, für welche Sie sich entscheiden. (4P)

4) Ein Router R habe die in der linken Hälfte von Tabelle 3 dargestellte Routing-Tabelle. Bestimmen Sie für die IP-Adressen auf der rechten Seite der Tabelle, auf welcher Route der Router R ein Datagramm mit dieser Ziel-IP-Adresse weiterleiten würde. (3P)
Hinweis: Die Spalte „IP Adressen“ muss nicht ausgefüllt werden.

Tabelle 3: Routing-Tabelle

Route	Adress-Bereich	Subnetzmaske	IP Adressen	Liste der Ziel IP-Adressen	Route
1	0.0.0.0	0.0.0.0		121.17.63.34	
2	121.0.0.0	255.240.0.0		121.223.32.1	
3	121.16.0.0	255.252.0.0		121.17.64.175	
4	121.17.0.0	255.255.192.0			
5	121.17.0.0	255.255.128.0			
6	121.17.64.0	255.255.255.128			
7	121.17.64.160	255.255.255.240			
8	121.192.0.0	255.224.0.0			
9	121.223.0.0	255.255.224.0			
10	121.223.16.0	255.255.248.0			

Dezimal	Binär	Dezimal	Binär		
128	1000 0000	1	0000 0001	160	1010 0000
192	1100 0000	16	0001 0000	175	1010 1111
224	1110 0000	17	0001 0001	223	1101 1111
240	1111 0000	32	0010 0000		
248	1111 1000	34	0010 0010		
252	1111 1100	63	0011 1111		
254	1111 1110	64	0100 0000		
255	1111 1111	121	0111 1001		

Name: Matrikelnummer:

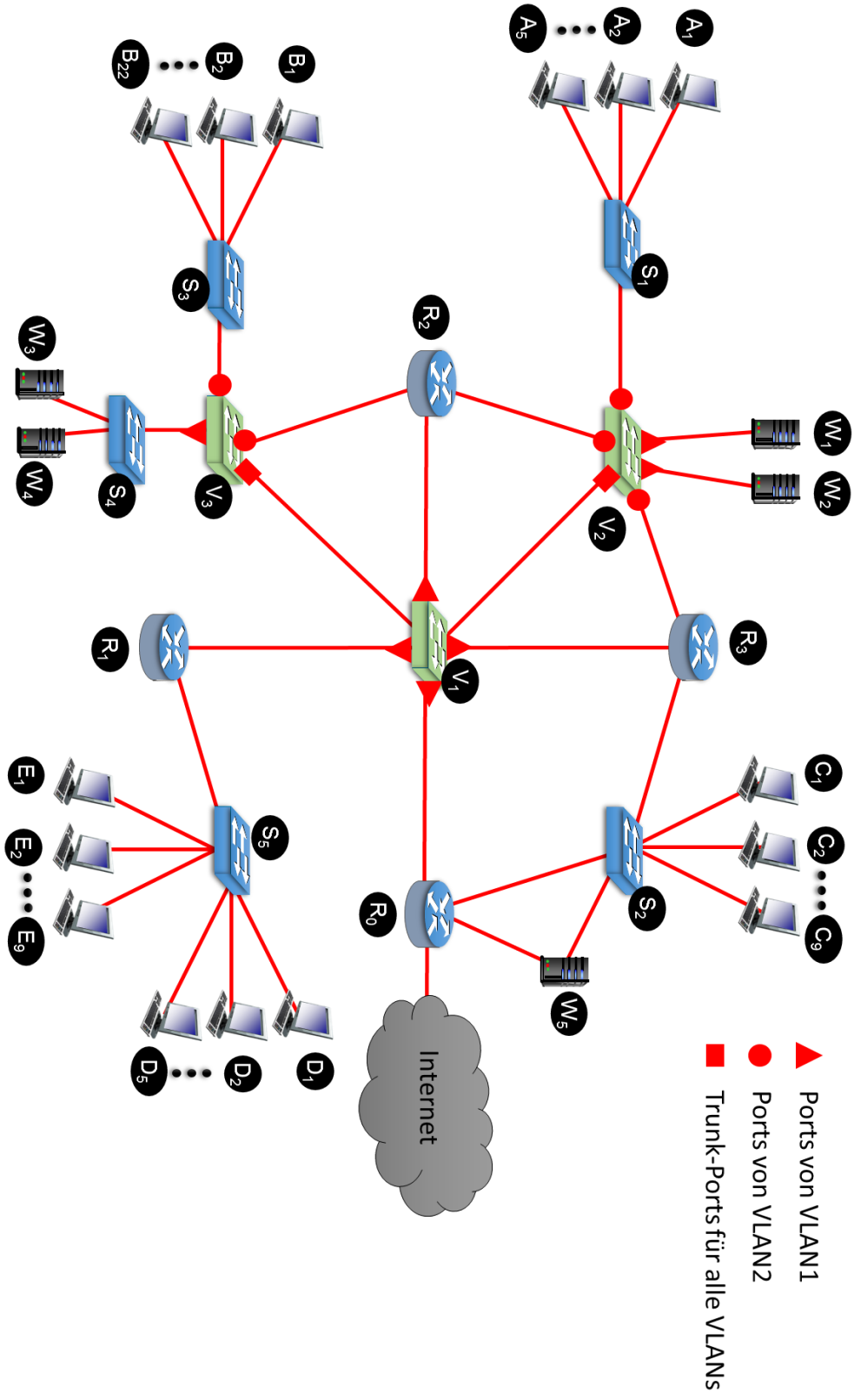

Abbildung 3: Netz mit mehreren Netzwerksegmenten

Name: Matrikelnummer:

Aufgabe 8 [Verständnisfragen] (15 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. Beim Versenden mehrerer Nachrichten nacheinander hat UDP hat gegenüber TCP den Vorteil, dass eine explizite Kennzeichnung des Beginns und des Endes einer einzelnen Nachricht nicht notwendig ist.
2. In einem VLAN-Switch dienen sogenannte Trunk-Ports dazu, Daten zwischen zwei unterschiedlichen VLANs auszutauschen.
3. Die in http/2.0 eingeführte Header-Compression führt vor allem im Zusammenspiel mit den ebenfalls in http/2.0 eingeführten Streams zu einer deutlichen Verringerung der Page-Load-Times. Würde die Header-Compression in http/1.1 (Persistent http ohne Pipelining) angewandt, so würde sich kaum ein Effekt zeigen.
4. Durch die VLAN-Technologie werden generell Loops vermieden, so dass hier kein Spanning-Tree-Verfahren mehr eingesetzt werden muss.
5. Ein Cache wird von einem ISP im Normalfall zusammen mit einem transparenten Forward-Proxy betrieben.
6. Der Betrieb eines Caches lohnt sich für einen ISP vor allem, weil dadurch der teure Transit Traffic in andere Netze reduziert wird.
7. Durch das regelmäßige Versenden von Hello-Paketen teilt ein OSPF-Router seinen Nachbarn Veränderungen in seiner lokalen Umgebung mit.
8. Ein Netzwerksegment ist ein Subnetz, das nicht in kleinere Subnetze aufgeteilt werden kann. Ein Subnetz kann mehrere Netzwerksegmente umfassen, die nicht direkt über einem Router verbunden sein müssen.
9. Der Domain Name Service (DNS) wird gerne zusammen mit Reverse-Proxys eingesetzt, um einen Lastausgleich zwischen den und innerhalb der verschiedenen Data-Centers eines Content-Delivery-Network-Betreibers zu erreichen.
10. Das Address Resolution Protocol (ARP) dient generell dazu, Adressen in lokalen Netzen aufzulösen. So kann ein Host mittels ARP die MAC-Adresse zu einer bekannten IP Adresse oder aber die IP-Adresse zu einer bekannten MAC-Adresse bestimmen.
11. Ein transparenter Proxy schaltet sich in die Kommunikation zwischen Client und Server ein, ohne dass Client oder Server dies bemerken. Technisch wird das umgesetzt, indem der transparente Proxy die Segmente einer TCP Verbindung abfängt und gegebenenfalls verändert an den ursprünglichen Empfänger weiterleitet.
12. Eine erfolgreiche Fast Recovery Phase in TCP dauert genau eine Round Trip Time.
13. Das Border Gateway Protocol (BGP) bestimmt ausschließlich Routen von den Border Gateway Routern des eigenen Netzes zu den jeweiligen Ziel-Netzen.
14. Local Preference ist das wichtigste BGP Attribut, um in einem AS zu bestimmen, über welchen Border Gateway Router und über welche Route Verkehr zu einem Ziel AS gesendet wird.
15. Ein Netzwerkdienst wie beispielsweise der Domain Name Service (DNS) kann gleichzeitig mit TCP und mit UDP als Transportprotokoll auf demselben Port angeboten werden.