



**Note:**

- During the attendance check a sticker containing a unique QR code will be put on this exam.
- This QR code contains a unique number that associates this exam with your matriculation number.
- This number is printed both next to the QR code and to the signature field in the attendance check list.

---

## Advanced Computer Networks

**Module:** IN2097

**Date:** 8.4.2016

**Examiner:** Prof. Dr.-Ing. Georg Carle

**Exam:** Final exam

---

	P 1	P 2	P 3	P 4	P 5
First correction					
Second correction					

Left room from \_\_\_\_\_ to \_\_\_\_\_

from \_\_\_\_\_ to \_\_\_\_\_

Early submission at \_\_\_\_\_

Notes \_\_\_\_\_



# Final exam

## Advanced Computer Networks

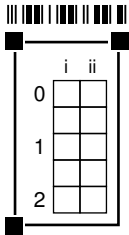
Prof. Dr.-Ing. Georg Carle  
Chair of Network Architectures and Services  
Department of Informatics  
Technical University of Munich

**Friday, 8.4.2016**  
**8:30 – 9:30**

- This exam consists of
  - **16 pages** with a total of **5 problems** and
  - a two-sided printed **cheat sheet**.Please make sure now that you received a complete copy of the exam.
- Subproblems marked by \* can be solved without results of previous subproblems.
- **Answers are only accepted if the solution approach is documented.** Give a reason for each answer unless explicitly stated otherwise in the respective subproblem.
- Do not write with red or green colors nor use pencils.
- The total amount of achievable credits in this exam is 60.
- Allowed resources:
  - a **non-annotated, printed vocabulary** English ↔ native language.
- Physically turn off all electronic devices, put them into your bag and close the bag.

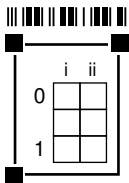
## Problem 1 Quiz (11 credits)

The following questions cover multiple topics and can be solved independently of each other.

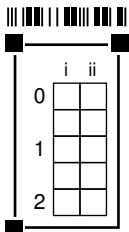
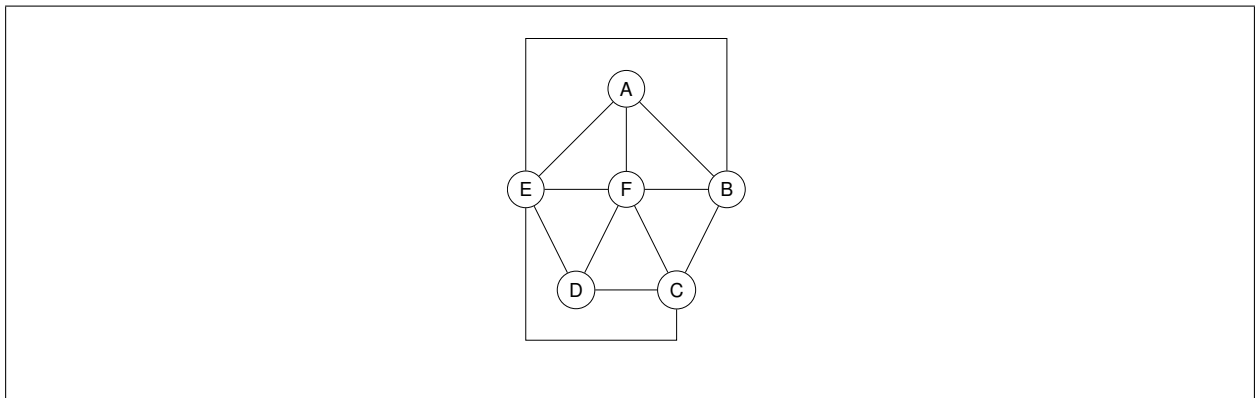


a)\* Connect the protocols to their corresponding ISO/OSI layer as well as to their corresponding term for their PDU. **Hint:** There may be terms where no/several pairings are possible.

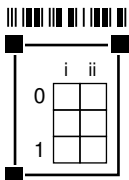
1	IP	Frame
2	Ethernet	Segment
3	TCP	Package



b)\* Add a single edge to the network in Figure 1b) to increase the maximum value of  $k$  when performing the  $k$ -core algorithm.



c)\* Given the IP address 61.149.21.16 and subnet mask 255.255.255.192, determine the corresponding network and broadcast addresses.



d)\* Longest prefix matching is an algorithm commonly used in computer networks. Where is it used specifically and how does it work?

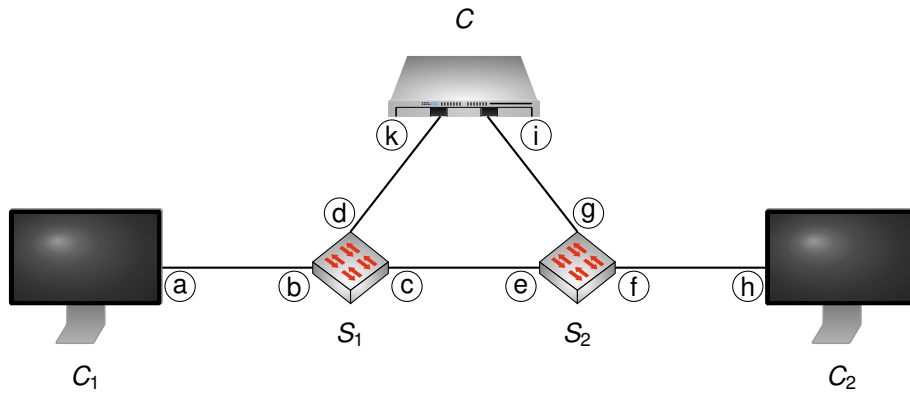


Figure 1.1: Software defined network topology

Consider the software defined network in Figure 1.1. The network consists of two clients  $C_1$  and  $C_2$  and an OpenFlow controller  $C$ . In between the clients two switches  $S_1$  and  $S_2$  are installed. Each of the switches is connected to a single client, to the respective other switch and to the controller. There are two flow tables already installed on the switches:

Match fields	Action
$d1\_type = 0x0800, nw\_proto = 0x1, nw\_src = C_1\_src\_ip$	[port_c]

Table 1.1: Flow table of  $S_1$

Match fields	Action
$d1\_type = 0x0800, nw\_proto = 0x6, nw\_src = C_1\_src\_ip$	[port_e]

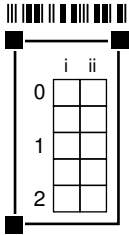
Table 1.2: Flow table of  $S_2$

The standard action for a packet if no rule matches is to transmit it to the controller. The default action of the controller is to send incoming packets back to the switch and to instruct it to forward the packet to all interfaces except the original source interface and the management port. Explanation for match fields:

- $d1\_type$ : used type of payload for the data link layer protocol (**Hint:** see cheat sheet for values, section Ethernet)
- $nw\_proto$ : used type of payload for the network layer protocol (**Hint:** see cheat sheet for values, section IPv4)
- $nw\_src$ : source address of network protocol

e)\*  $C_1$  pings  $C_2$ . List all interfaces an ICMP echo request packet travels through the topology in the correct order (only the request packet, not the answer packet).

	i	ii
0		
1		
2		
3		



---

f)\* Consider a switch based on off-the-shelf hardware such as Open vSwitch and a dedicated SDN switch based on specialized hardware. List two advantages for each of the switches.

	i	ii
0		
1		
2		



## Problem 2 Receive Side Scaling (9 credits)

Receive Side Scaling (RSS) is a feature to distribute the network traffic to different hardware queues assigned to different CPU cores. Figure 2.1 presents the sequence performed during packet reception on an RSS system.

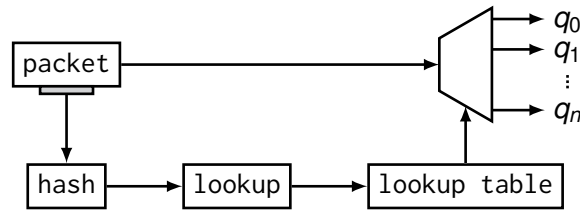


Figure 2.1: Receive Side Scaling

a)\* Why is the support for multiple hardware queues beneficial for distributing CPU load to different cores instead of using a single queue.

The following calculations use the *XOR* (exclusive or) and *MOD* (modulo) operators. The hash function is applied on a given IPv4 address A.B.C.D:

$$\begin{aligned} \text{hash}(\text{IPv4}_{\text{src\_addr}}) &= \\ \text{hash}(A.B.C.D) &= A \text{ XOR } B \text{ XOR } C \text{ XOR } D \end{aligned}$$

After calculating the hash a lookup in a hardware lookup table is performed. This determines the hardware queue where packets are enqueued. CPU cores assigned to a queue process the packets afterwards. The lookup is calculated as follows:

$$\text{lookup}(\text{hash}(\text{IPv4}_{\text{src\_addr}})) = \text{hash}(\text{IPv4}_{\text{src\_addr}}) \bmod 2^n$$

For this problem:  $n = 3$ .

b)\* Perform a hash and a lookup operation on the IP address 192.0.2.1.

c) Determine the IP addresses of the 192.0.2.0 / 29 subnet for the following mappings.

Lookup table	
0	queue <sub>0</sub>
1	queue <sub>1</sub>
2	queue <sub>2</sub>
3	queue <sub>3</sub>
4	queue <sub>4</sub>
5	queue <sub>5</sub>
6	queue <sub>6</sub>
7	queue <sub>7</sub>

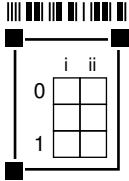
  

→

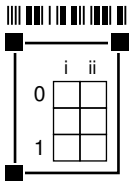
→

The lookup table has a fixed size of 8. All cells of the lookup table must be filled at all times. As the cores should have a similar utilization, the incoming packets should be distributed in a uniform manner among the cores. You can assume that the source IP addresses are uniformly distributed which leads to a uniform distribution for hash and lookup function alike.

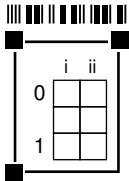


d)\* Give the content of a lookup table if three queues/cores are in use. Try to approximate the uniform distribution as closely as possible.

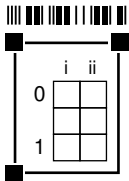
MOD result	0	1	2	3	4	5	6	7
Queue number								



e) Calculate the relative probability of packet distribution for each core in percent.



f) What would be the ideal value and how big is the maximum deviation.



g)\* Depending on n, how many CPU cores/queues should be used if the distribution should be met as accurately as possible?



### Problem 3 BGP (7 credits)

Figure 3.1 shows a small AS topology including border routers R1, R2 and R3. AS20 and AS77 are costumers of AS10. AS20 and AS77 have a peering agreement, i. e., they exchange traffic for free. AS10 and AS77 own prefixes that are announced to their customers/peering partners. AS77 owns the prefix 191.168.0.0/20, AS 10 owns the prefix 151.17.0.0/16.

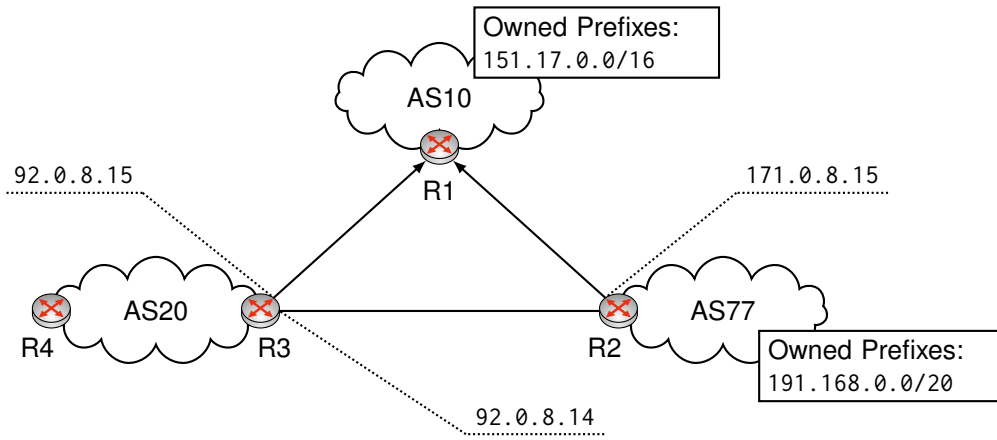


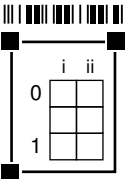
Figure 3.1: AS topology

a)\* There exist two “flavors” of BGP. Which ones are used between the listed routers?

Between R4 and R3:

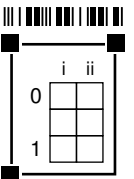
  

Between R1 and R3:



b)\* Create the routing table entry/entries for the border router of A10 (R1) for the owned prefixes of AS77.

Network destination	Subnetmask	Next hop IP address



An attacker controlling AS20 wants to sniff the traffic that is exchanged between AS10 and AS77 for their respective prefixes 151.17.0.0/16 and 191.168.0.0/20. Therefore, this traffic shall be routed through AS20, i.e., AS20 performs a man-in-the-middle attack.

c)\* Describe what the attacker has to do to perform the man-in-the-middle attack.

d) List the new routing table entry/entries for the border router of AS10 (R1) after a successful man-in-the-middle attack of AS20 with as little entries as possible.

Network destination	Subnetmask	Next hop IP address

## Problem 4 Wireshark (14 credits)

We consider the IP packet depicted in Figure 4.1 as hexdump in network byte order.

```

IHL  TOS  Total Length  Identification  Flags/FragOffset  TTL
|45| c0| 00 5c| d4 f6| 00 00| 3f| 01 6e 6c C0 00 02 00
0A 00 00 02 03 03 34 ae 00 00 00 00
  
```

Figure 4.1: Partial hexdump of an IP packet in network byte order

The topology of the network for this problem is given in Figure 4.2. A client *C* wants to connect from its local network to a server *S* on the Internet via the NAT router *R*.

**Note:** To solve this problem use the cheat sheet that is handed out separately.

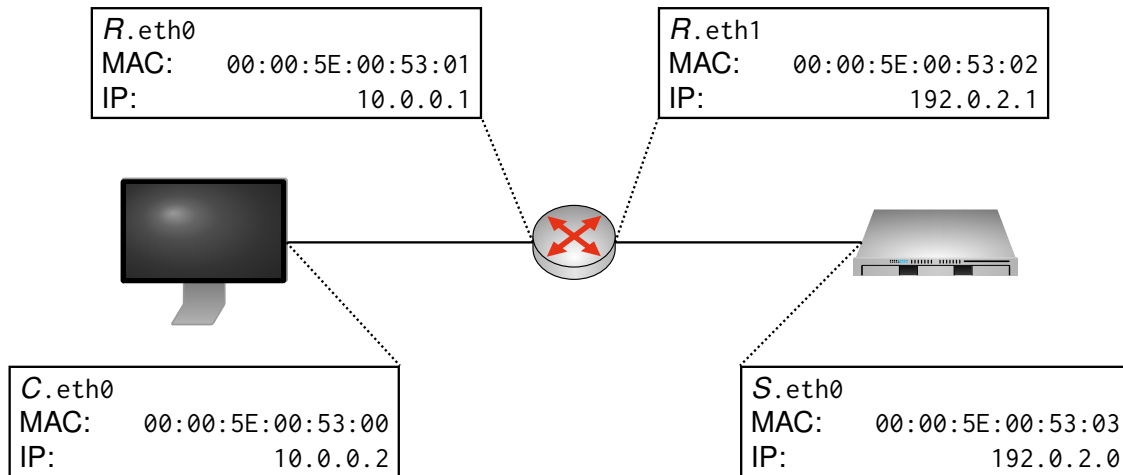
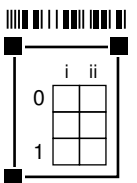
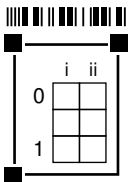
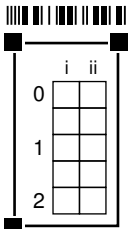


Figure 4.2: The network topology in which the packet was recorded

a)\* Mark and name the header fields in Figure 4.1.

b)\* Argue from which interface to which interface the packet (see Figure 4.1) was transferred.

c)\* Argue what protocol is contained in the payload of the IP packet in Figure 4.1.



The IP packet in Figure 4.1 contains an incomplete ICMP message starting at offset 0x0014.

d)\* Argue which kind of ICMP message it is and why this message could be generated.

0	i	ii
1		
2		

This ICMP message was received after client *C* tried to open an **SSH connection** to *S*. With this information it is possible to recreate the missing payload of the ICMP message.

e) Name the protocol(s) contained in the payload of the ICMP message.

0	i	ii
1		

f)\* Create a **hexdump** of the missing payload for the ICMP message of Figure 4.1.

**Hint:** The calculation of correct checksums is not necessary, fill in 0xFF blocks if needed. Not all listed headers might be used for SSH.

0	i	ii
1		
2		
3		
4		
5		

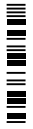
IPv4-Header																																			
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31			
0B					0x5					0x00																	0x00					0x40			
4B																0	1	0												0x00					0x00
8B																																			
12B																																			
16B																																			

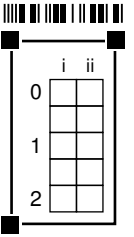
UDP-Header																																
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0B																																
4B																																

TCP-Header																																				
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31				
0B																																				
4B																																				
8B																																				
12B																																				
16B																																				

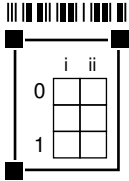


g)\* Generate a hexdump of the Ethernet frame for the IP packet given in Figure 4.1. The entire payload of the Ethernet packet may be abbreviated with '...'.  
**Hint:** The calculation of correct checksums is not necessary, fill in 0xFF blocks if needed.

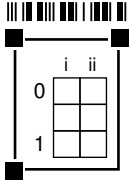


## Problem 5 TCP (19 credits)

TCP has the ability to detect packet loss via timeouts. This timeout value, called RTO, must be adopted to the properties of a TCP connection to work properly.



a)\* What happens to the detection of packet loss when the RTT of a TCP connection is estimated too high?



b)\* What happens to the detection of packet loss when the RTT of a TCP connection is estimated too low?

For a network experiment the connections  $S_1$  to  $C$  and  $S_2$  to  $C$  over router  $R$  as shown in Figure 5.1 are tested. During the experiment different buffer configurations for  $R$  shall be tested. The bandwidth of each depicted link is the same.

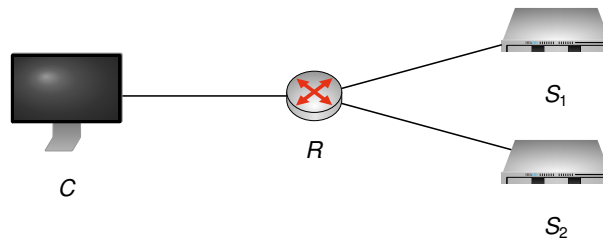


Figure 5.1: Network topology

Several active TCP connections try to use the full bandwidth available between  $S_1$  and  $C$  and between  $S_2$  and  $C$ . Additionally a ping from  $C$  to  $S_1$  is executed. The average RTT measured by ping for a large buffer configuration and a small buffer configuration is plotted in Figure 5.2.

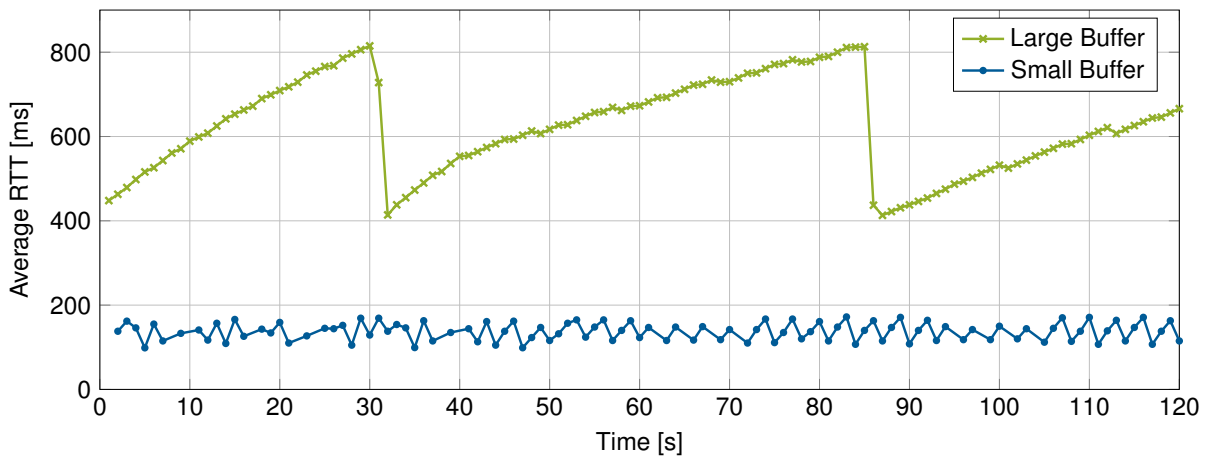
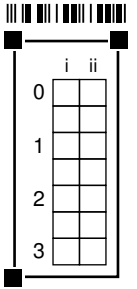
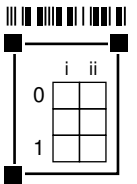


Figure 5.2: Latency with different buffer sizes

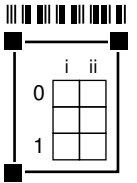
c) What is the influence of different buffer sizes on the accuracy of the RTO estimation. Argue with the results for the two buffer sizes from Figure 5.2.

d)\* What happens to latency critical applications such as VoIP in the presence of large buffers?

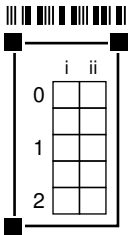



e) How can the situation for latency critical applications be improved, without changing the buffer size?

To improve the behavior of TCP with large buffers an active queue management is introduced.

f)\* Name and explain two different discard policies.

The algorithm applied is weighted fair queuing. For that three classes of traffic shall be considered A, B and C with their corresponding weights  $w_A = 2$ ,  $w_B = 4$ ,  $w_C = 3$ . The weight of the traffic classes gives the maximum burst size which is allowed for a certain traffic class. The algorithm starts to send a burst of A traffic, after the burst size is exhausted or the queue for this traffic class is empty, the same is done for the traffic class B and afterwards for traffic class C. At each timeslot an arbitrary number of packets can be received but only a single packet can be sent.

g)\* Calculate the overall possible bandwidth capacity for each traffic class in percent. Assume that enough packets for all three classes are available to always fill the respective bursts.

i		ii	
0			
1			
2			

For the next problems consider the following incoming packets:

Timeslot	1	2	3	4	5	6	7	8	9	10	11	12
A	1	0	0	1	1	0	0	0	0	0	0	0
B	1	1	0	1	1	0	0	0	0	0	0	0
C	1	2	2	0	0	0	0	0	0	0	0	0

h) Compute which packet is sent out at each time slot. Additionally, give the content of the queues at this time. **Hint:** Timeslot 1 is already computed, proceed accordingly.

Timeslot	1	2	3	4	5	6	7	8	9	10	11	12
Queue A	0											
Queue B	1											
Queue C	1											
Sent	A											

In the following a token bucket approach for QoS shall be applied. The initial tokens are  $t_A = 0$ ,  $t_B = 2$ , and  $t_C = 2$ . An additional token is generated for the queue A at every fourth, queue B at every second, and queue C at every third timeslot. Token handling/checking is done before making the decision for sending.

i)\* Compute which packet is sent out at each time slot. Additionally, give the content of the queues and the available tokens for the respective traffic class at this time. **Hint:** Timeslot 1 is already computed, proceed accordingly.

Timeslot	1	2	3	4	5	6	7	8	9	10	11	12
new Token		B	C	AB		BC		AB	C	B		ABC
Queue A	1											
Tokens A	0											
Queue B	0											
Tokens B	1											
Queue C	1											
Tokens C	2											
Sent	B											



---

**Additional space for solutions—clearly mark the (sub)problem your answers are related to and strike out invalid solutions.**

