Chair of Network Architectures and Services
Department of Informatics
Technical University of Munich

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

Advanced Computer Networking

Exam: IN2097 / Endterm
Examiner: Prof. Dr.-Ing. Georg Carle
Date: Thursday 1st March, 2018
Time: 08:30 – 09:30

	P 1	P 2	P 3	P 4	P 5
I					
II					

Working instructions

- 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.
- Detaching pages from the exam is prohibited.
- 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 credits.
- Allowed resources:
  - one **analog dictionary** English ↔ native language
- Physically turn off all electronic devices, put them into your bag and close the bag.

Left room from __________ to __________ / Early submission at __________

Problem 1 Quiz (7.5 credits)

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

a)* The spanning tree protocol (STP) creates an acyclic tree topology for a given network. Argue which kind of tree is created by the STP, a minimum spanning tree or a shortest path tree.

The STP creates an SPT, as the spanning tree is minimal from the perspective of a single node (the root bridge). A MSP would be minimal from a global perspective, but not necessarily from the perspective of a single node.

b)* A provider announces two prefixes, `188.95.7.0/24` and `188.95.8.0/24`. To simplify prefix announcement, he aggregated the neighboring prefixes and instead announces `188.95.7.0/23`. Is this simplification correct?

The network address of `188.95.7.0/24` and `188.95.8.0/24` do not match in the 23 most significant bits. Therefore, both address blocks cannot be aggregated into a common /23 network.

c)* Classify the routing protocols: RIP, OSPF, IS-IS, BGP

- Inter-domain routing protocol: (e)BGP
- Intra-domain routing protocol: (i)BGP, RIP, OSPF, ISIS, BGP

d)* Which header fields are usually used for 5-tuple hashing?

IP src/dst address, L4 src/dst port, L4 protocol

e)* QUIC relies on UDP as a transport layer protocol. Despite the unreliable nature of UDP, QUIC is used by protocols requiring reliable transport such as HTTP. Why does this work?

QUIC implements reliablity mechanisms on top of UDP. UDP is mainly used for compatiblity reasons (no new protocol was introduced from the perspecive of many middle boxes).

f)* The maximum packet rate of a 10 Gbit/s connection is 14.88 million packets per second. However, calculating the packet rate for 64-byte frames leads to 19.53 million packets per second. Explain the difference.

64 byte is the size of the packet on Layer 2 , on Layer 1 20 bytes are sent additionally including: IPG, preamble, SFD.

Problem 2 Software-Defined Network (11 credits)

This problem investigates the behavior of software defined networks.

a)* What is the forwarding plane and what does it do?

The forwarding plane is the entity in the SDN which handles the actual packet reception/transmission.

b)* What is the data plane and what does it do?

The forwarding plane is the entity in the SDN which handles the actual packet reception/transmission.

c)* Explain one similarity and one difference between OpenFlow and P4.

- Both define standardized interfaces for programming programmable switches
- OpenFlow supports a fixed set of protocols, with P4 entirely new protocols can be implemented

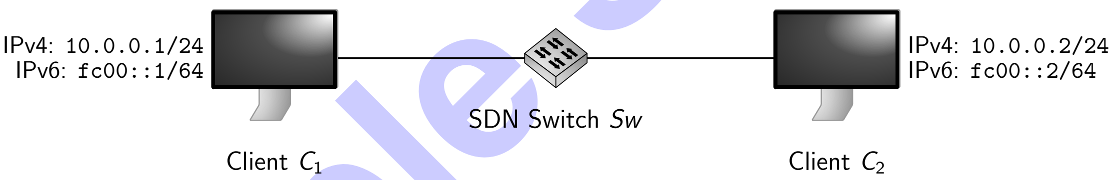


Figure 2.1: Software-defined network topology

Consider the Ethernet network topology given in Figure 2.1. Both clients, C_1 and C_2, are freshly booted, no connection has been established between them. The IP addresses (IPv4 and IPv6) are configured manually on both clients. The default rules on the SDN Switch *Sw* were deleted. After that the rules specified in Listing 1 were installed. If no rules are available for a packet, the *Sw* is configured to drop the packet.

```
ovs-ofctl add-flow Sw dl_type=0x86dd,actions=flood
ovs-ofctl add-flow Sw dl_type=0x0800,actions=flood
```

Listing 1: OpenFlow rules installed on *Sw*

d)* Explain the rules specified in Listing 1

- `add-flow S`: installs rule on switch *Sw*
- `dl_type=0x86dd`: rule is executed for packets with Ethertype 0x86dd (IPv6)
- `dl_type=0x0800`: rule is executed for packets with Ethertype 0x0800 (IPv4)
- `actions=flood`: packets are forwarded on every switch port except the receiving one

0
1
2

e) C_1 pings C_2 with the following command: `ping 10.0.0.2`. What is the expected result of ping? Explain the way of the packets in the network and the protocols involved.

- As a first step the IP addresses have to be resolved, i.e. the MAC addresses belonging to the IP addresses must be determined.
- For IPv4 networks ARP is used for this purpose
- Client C_1 generates an ARP request and sends it to Switch *Sw*
- *Sw* or its Controller have no rule matching the ARP Ethertype $\rightarrow$ message is dropped
- Ping does not work because address cannot be resolved

0
1
2
3

f) C_1 pings C_2 using IPv6 with the following command: `ping -6 fc00::2`. What is the expected result of ping? Explain the way of the packets in the network and the protocols involved.

- As a first step the IP addresses have to be resolved, i.e. the MAC addresses belonging to the IP addresses must be determined.
- For IPv6 networks use NDP for this purpose
- Client C_1 generates an Neighbor solicitation utlizing IPv6/ICMPv6 message and sends it to Switch *Sw*
- *Sw* finds a rule matching the Ethertype `0x86dd` and forwards the message to C_2
- C_2 answers with neighbor advertisement and sends it to S which forwards it C_1
- Now ping messages can be forwarded (using IPv6/ICMPv6)

Problem 3 Poor man's wireshark (16.5 credits)

This problem investigates a hexdump of the Ethernet frame given in Figure 3.1.

```
                 Receiver MAC                     Transmitter MAC        Ethertype  Ver  TC
0x0000   9e 2e f5 30 d9 06 c2 a3    05 ac 59 07 86 dd 60 00

         Flow Label  Payld Length Nxt Hdr Hop Ct          Source Address
0x0010   6c fd 00 40 3a 40 fc 00    00 00 07 40 00 00 00 00

                 Source Address                 Destination Address
0x0020   00 00 00 00 00 01 fc 00    00 00 07 40 00 00 00 00

              Destination Address
0x0030   00 00 00 00 00 02 80 00    e1 ed 04 9a 00 03 43 16

0x0040   8b 5a 00 00 00 00 06 b2    0d 00 00 00 00 00 10 11

0x0050   12 13 14 15 16 17 18 19    1a 1b 1c 1d 1e 1f 20 21

0x0060   22 23 24 25 26 27 28 29    2a 2b 2c 2d 2e 2f 30 31

                                FCS
0x0070   32 33 34 35 36 37 FF FF    FF FF
```

Figure 3.1: Hexdump of an Ethernet frame including FCS

a)* Mark and name the fields of the Ethernet frame in Figure 3.1.

b) Mark the address of the L3 sender and the L3 receiver. Report them in their typical address format.

Source: `fc00:0:740::1`, Destination: `fc00:0:740::2`

c) Which program might create messages of this kind? Support your statement using data from Figure 3.1.

The packet is a ICMPv6 (IPv6 next header: `3a`) echo request (ICMPv6 type `0x80` or `128` code: `0`). Packets of this kind are used by tools such as ping or traceroute.

The topology of the data center is shown in Figure 3.2. This data center separates the traffic of its customers using VLANs (IEEE 802.1q). Customer A has the VLAN ID 1020, Customer B has VLAN ID 846. DEI and PCP are both set to 0 for every VLAN. The packet in Figure 3.1 was observed on the link between Customer A and Switch S_1.

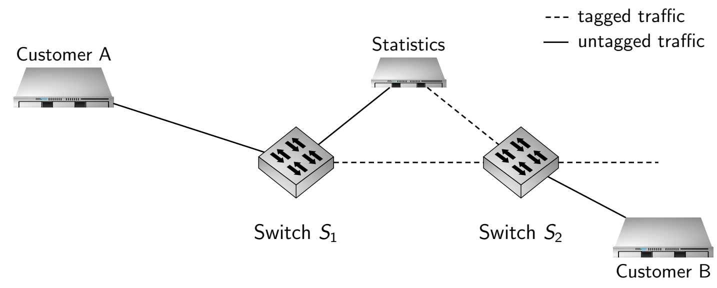


Figure 3.2: Network topology

d) Create a hexdump of the Ethernet frame for this packet how it can be observed on the link between Switch S_1 and Switch S_2. You may shorten the payload of the frame using [. . .]. Checksums do not need to be calculated, all bits should be set to 1.

0x9e2ef530d906 , 0xc2a305ac5907 , 0x8100 , 0x03FC , 0x86dd , [. . .], 0xFFFFFFFF

The data center wants to collect traffic statistics. Therefore, a server monitors the traffic on the switches in the tagged and untagged area of the network (see Figure 3.2). The statistics function counting the packets is given in Listing f). This function, called `update_stats()`, is called for every frame the statistics server receives. The parameter `hxdmp` contains a hexdump of the Ethernet frame received. Checksums are validated in hardware and do not need to be checked in software.

```
ipv4_pkts = 0
ipv6_pkts = 0
other_pkts = 0

def update_stats(hxdmp):
  if hxdmp[14] & 0xF0 == 0x40:
    ipv4_pkts += 1
  elif hxdmp[14] & 0xF0 == 0x60:
    ipv6_pkts += 1
  else:
    other_pkts += 1
```

Listing 2: Statistics function

e)* The data center provider realizes that the `update_stats()` does not count the packets correctly. Explain the reason**s** based on the code given in Listing f).

- VLAN tags are not considered by the code.
- Code assumes that every protocol in the payload of the Ethernet frame starts with a version field.

f) Create your own statistics function which collects correct statistics.

```python
ipv4_pkts = 0
ipv6_pkts = 0
other_pkts = 0

def update_stats(framedump):
  if framedump[12] == 0x81 and framedump[13] == 0x00:
    if framedump[16] == 0x08 and framedump[17] == 0x00:
      ipv4_pkts += 1
    elif framedump[16] == 0x86 and framedump[17] == 0xdd:
      ipv6_pkts += 1
    else:
      other_pkts += 1
  else:
    if framedump[12] == 0x08 and framedump[13] == 0x00:
      ipv4_pkts += 1
    elif framedump[12] == 0x86 and framedump[13] == 0xdd:
      ipv6_pkts += 1
    else:
      other_pkts += 1
```

Problem 4 TCP Congestion Control (18 credits)

This problem investigates differences in the congestion control algorithms of TCP.

a)* What is the goal of flow control?

Protect the destination from overloading

b)* What is the goal of congestion control?

Protect the network from overloading

c)* Name two sources of delay which cannot be influenced by TCP congestion control.

- Propagation / transmission delay
- Serialization delay
- Processing delay (at least the processing which is not part of TCP)

d)* Congestion algorithms, such as Cubic or BBR, can roughly be divided in two classes depending on the indicator they use for detecting congestion. Enter both sources.

Indicator of congestion	algorithm
Packet loss / duplicate ACKs	Cubic
RTT / RTT degression	BBR

For the following subproblems consider the following scenario: The Client C downloads a large file from the Server S. Figure 4.1 shows the network topology.

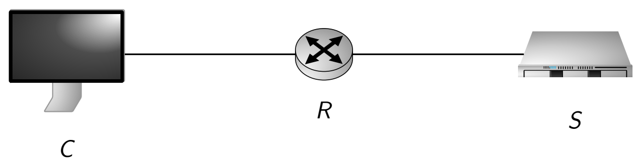
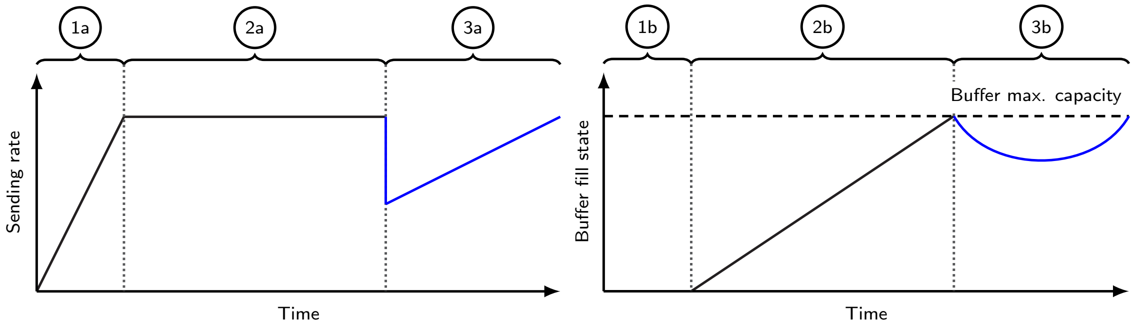


Figure 4.1: Network topology



(a) Sending rate (b) Buffer fill state

Figure 4.2: TCP measurements

Figure 4.2a presents the sending rate of *S* over time and Figure 4.2b depicts the fill state of the buffer on *R* over the same time span. Both figures do not show the entire download but a short period.

e) Argue which kind of TCP congestion control, BBR or Cubic, is displayed in Figure 4.2.

- Figure 4.2 shows a loss based congestion control.
- The buffer fills the buffer (2b) without the sending rate going down (2a)

Assume the download is not over after phase 2a/2b in Figure 4.2.

f) Explain what happens to the TCP connection between Server S and Client C in phase 3a and 3b of Figure 4.2.

- The buffer is full at the beginning of phase 3, any further packets will be dropped.
- Client C detects the dropped packets due to the missing sequence numbers of the received packets
- Client C signals missing packets to Server S and Server S throttles sending

g) Continue the graph in Figure 4.2a (additional preprints in Figure 4.6).

h) Continue the graph in Figure 4.2b (additional preprints in Figure 4.6).

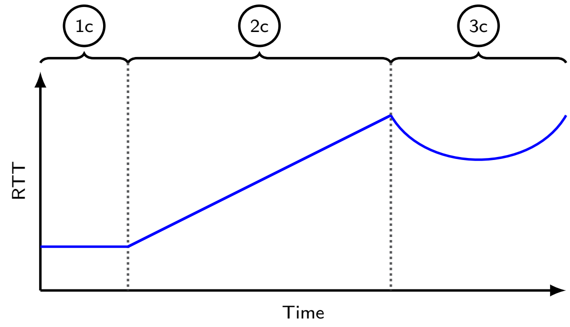


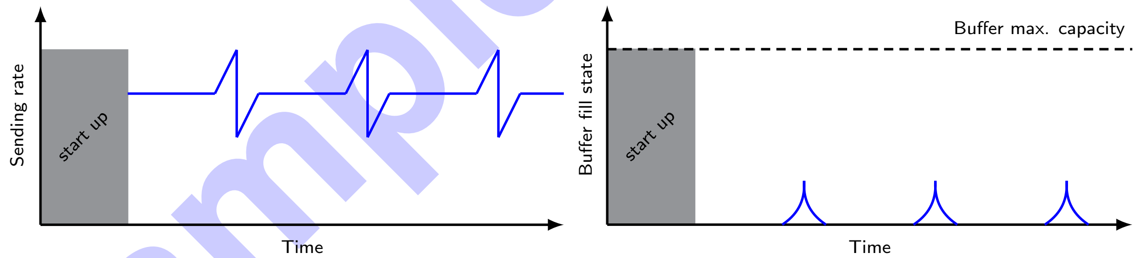
Figure 4.3: RTT

i) How does the RTT look like? Draw a graph for the phases 1c, 2c, and 3c in Figure 4.3 (additional preprints in Figure 4.6).

j) Explain why the graph looks that way for each of the phases 1c to 3c of Figure 4.3.

- 1c: Constant RTT (only serialization, propagation and processing delay)
- 2c: RTT increases because of additional queuing delay
- 3c: RTT increases/decreases regularly due to congestion control

You selected a specific congestion control algorithm, either Cubic or BBR, in Problem e). How would the graphs look like for the other congestion control algorithm? You do not need to consider the startup phase of the connection but only the later phases.



(a) Sending rate (b) Buffer fill state

Figure 4.4: TCP measurements

k) Create the sending rate graph for the other congestion control algorithm in Figure 4.4a (additional preprints in Figure 4.6).

l) Create the buffer fill state graph for the other congestion control algorithm in Figure 4.4b (additional preprints in Figure 4.6).

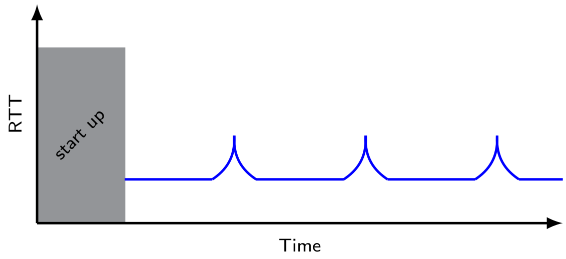


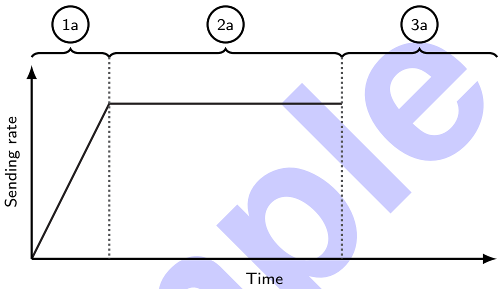
Figure 4.5: RTT

m) Create the RTT graph for the other congestion control algorithm in Figure 4.5 (additional preprints in Figure 4.6).

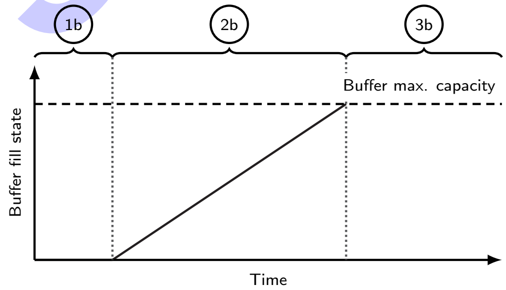
n) Explain for each graph why the graph looks that way for your chosen congestion control algorithm.

- BBR tries to keep the RTT low by regularily measuring the RTT.
- Sending rate is throttled sooner to keep the buffer fill rate low
- BBR tries to keep buffer empty

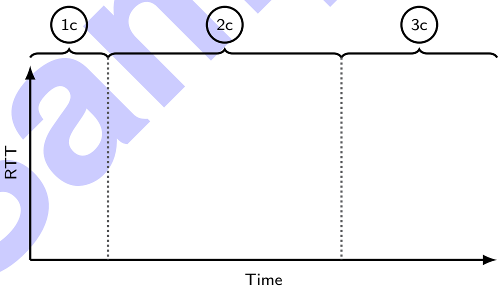
In case of correction use the plots below. Clearly mark the solution to be graded:



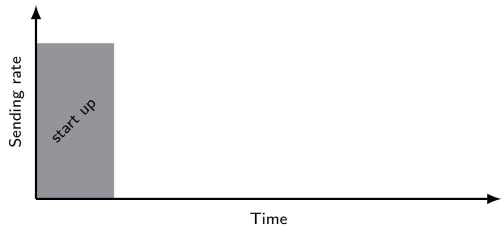
(a) Replaces plot of Subproblem g)



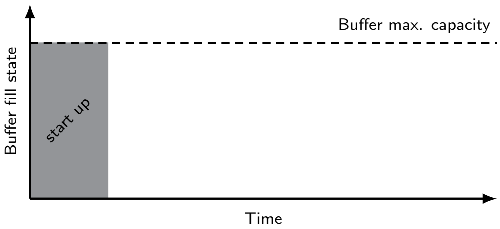
(b) Replaces plot of Subproblem h)



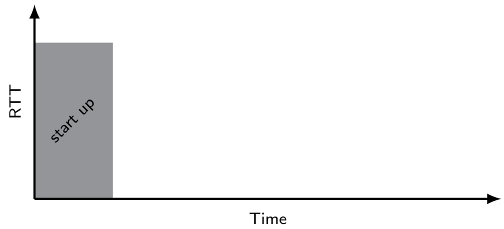
(c) Replaces plot of Subproblem i)



(d) Replaces plot of Subproblem k)



(e) Replaces plot of Subproblem l)



(f) Replaces plot of Subproblem m)

Figure 4.6: Additional preprints

Problem 5 Network Calculus (7 credits)

This problem analyzes a small network with deterministic network calculus.

a)* Which kind of guarantees can be given using deterministic network calculus?

Latency and buffer bounds

b)* In deterministic network calculus, flows are modeled according to their cumulative arrival function A. How is A defined?

$A(t)$ represents the amount of data sent by the flow in the time interval $[0, t)$

c)* What is the relationship between the cumulative arrival function A and the deterministic arrival curve α? *Hint: use only the mathematical definition.*

$A(t) - A(s) \leq \alpha(t - s), \forall 0 \leq s \leq t$

We are now interested in studying the network in Figure 5.1:

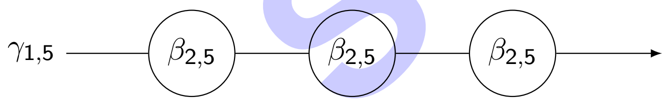


Figure 5.1: Network

d)* Draw the arrival curve $\gamma_{1,5}$ and the service curve $\beta_{2,5}$ into Figure 5.2.
Reminder:

- $\beta_{R,T}$ is a rate latency curve with rate R and latency T
- $\gamma_{r,b}$ is a token bucket curve with rate r and burst b

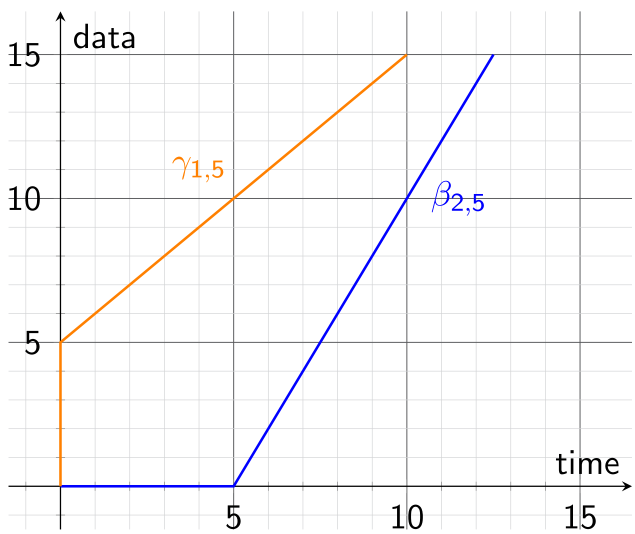


Figure 5.2

e)* What is the latency bound of the flow after having traversed the first server?

0 1

The latency bound is 7.5. The can be read graphically on the figure.

f)* Concatenate the three servers into one. What is the latency bound of the flow after having traversed this concatenated server?

0 1 2

The concatenation of the three servers is: $\beta_{2,15}$. The latency bound over the three servers is 17.5.

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

Sample Solution