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

Eexam
Place student sticker here

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	**Date:**	Monday 28th February, 2022
Examiner:	Prof. Dr.-Ing. Georg Carle	**Time:**	11:30 – 12:45

Working instructions

- This exam consists of **14 pages** with a total of **5 problems**.
  Please make sure now that you received a complete copy of the exam.
- The total amount of achievable credits in this exam is 75 credits.
- Detaching pages from the exam is prohibited.
- Allowed resources:
  - one **analog dictionary** English ↔ native language without annotations
  - the **provided cheatsheet** without annotations (print or digital copy)
- 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.
- Code of conduct:
  - I participate without the help of others and only use the allowed resources.
  - I do not share, discuss, or exchange any information related to the exam with anybody.
  - I feel in good health and I am able to participate in the exam.
  - I understood the examination policy, agree to the video supervision, and adhere to this process.

Problem 1 Quiz (17.5 credits)

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

0 1

a)* How does ZMap close a connection on the server even though it is stateless?

0 1

b)* What is the major difference between a ZMap UDP and TCP scan?

0 1

c)* A server has 723 (virtual) network interface cards (NICs). You want to assign a unique ID to each NIC. What is the minimum number of bits needed to encode the IDs?

0 1

d)* Name two disadvantages if network interface cards (NICs) or switches have fixed transceivers.

0 1 2

e)* How does TCP BBR use windowed filters to estimate the bandwidth-delay product (BDP)?

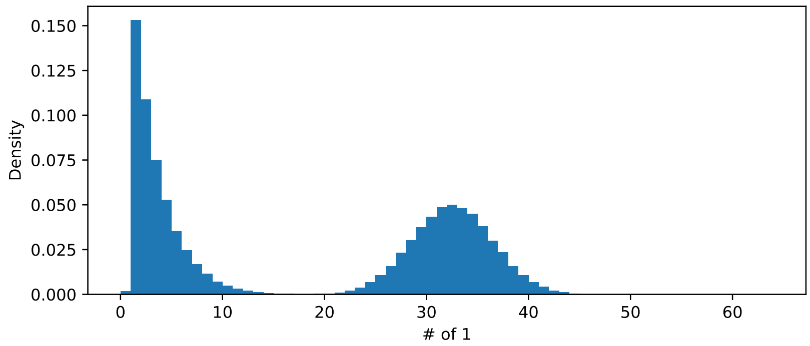

Figure 1.1: Distribution of bits set to one in the host part of a set of IPv6 addresses.

f)* Given the distribution of bits set to one in the host part of a set of IPv6 addresses shown in Figure 1.1, which device types do you expect to be represented by the addresses and why?

0
1

g)* A researcher wants to classify data packets based on their time of origin and builds a machine learning (ML) model to perform this task. Name one metric which the researcher can use to check the validity of the ML model. How can the researcher determine when the ML model is optimal?

0
1

h)* Shortly explain the three different types of DNS resolvers and the typical place you can find them as discussed in the lecture.

0
1
2
3

0
1

i)* What does DNS ECS stand for and what problem does it solve?

0
1

j)* Name two of the main components in DNS ECS (except the IP address family).

0
1

k)* Explain the basic principle of HTTP-based load balancing.

0
1

l)* Explain the basic principle of anycast-based load balancing.

0
1

m)* Assume you operate a Content Delivery Network where the number of available cache servers changes frequently. You assign users to cache servers based on the following hash function: $u \bmod S$, where u is the user and S is the number of servers. Argue whether this is a good approach or not.

Problem 2 **Wireshark (17 credits)**

According to the OSI model network protocols are distributed to seven different layers each containing several protocols. In this problem, a frame is analyzed, referring to the involved protocols. You are given a hexdump of an Ethernet frame, starting with the Ethernet header. For simplicity, we only show the first 80 B of the frame, the omitted parts are not relevant for this problem.

```
0x0000   a0 36 9f 54 c1 22 90 e2   ba 1c 55 11 86 dd 60 0e
0x0010   3c 97 04 d8 2c 40 20 01   0d b8 00 00 f1 01 00 00
0x0020   00 00 00 00 00 01 20 01   0d b8 00 00 f1 01 00 00
0x0030   00 00 00 00 00 02 11 00   00 01 57 9d 08 0e e1 82
0x0040   01 bb 05 08 ab b7 c5 00   00 00 01 08 c8 a4 5d 51
0x0050     . . .
```

Figure 2.1: Hexdump of an Ethernet frame cut off after 80 B

For this problem you must indicate the location of the bytes in the hexdump in Figure 2.1. You can **either** mark the corresponding bytes directly in the figure **or** list the locations of the corresponding bytes using **[...]**. Example: the **three** bytes from position 0 to 2 can be written as `[0, 2] = 0xa0369f`.

a)* Mark the **destination MAC** address in the hexdump in Figure 2.1 **and** note the address in the common format in the solutionbox.

b)* Identify the used layer 3 protocol. Mark the corresponding bytes in the hexdump and list the protocol name.

c) Mark the bytes, which are part of the layer 3 **destination address**. Note the address in the shortest, common notation.

0 1 2

d) Identify the used layer 4 protocol. Mark the corresponding bytes in the hexdump and name the protocol name.

0 1 2 3

e) Make an educated guess for the used protocols above of TCP / UDP in the hexdump. Mark corresponding bytes in the hexdump, list the protocol names, and give a brief reason for your answer.

The following problems can be answered without the results of the previous subproblems.

0 1 2 3

f)* Name **and** briefly explain two important goals (other than fixing head-of-line blocking) of QUIC, which target to fix problems with TCP / TLS.

0 1

g)* What is the purpose of the QUIC spin bit?

0 1 2 3

h)* QUIC claims that it fixes the head-of-line (HoL) blocking problem of TCP. For this multiple streams are multiplexed in the same connection. In the following, three different scheduling approaches to multiplex k streams are explained:

1. Send first all data of one stream, then continue with the next stream.
2. In each QUIC packet, $\frac{1}{k}$ of the payload are from each stream.
3. Send data from one stream in a single QUIC packet, then fill the next packet with the next stream's data.

For each scheduling approach, assess if it fixes HoL blocking and briefly explain your answer.

Problem 3 AS Relations and BGP (14 credits)

This problem investigates the autonomous system (AS) relationships in a given network and their impact on routing and traffic. $\rightarrow$ represents a customer $\rightarrow$ provider relationship, while $\leftrightarrow$ represents a peering relationship. Dashed lines are unknown policies, which need to be evaluated in Subproblem h). All ASes apply standard routing behavior. Furthermore, the following policies are applied:

- For routes with the same prefix, the AS selects the most cost-efficient route.
- For routes with the same prefix and with an equal traffic cost, the shorter route is selected.

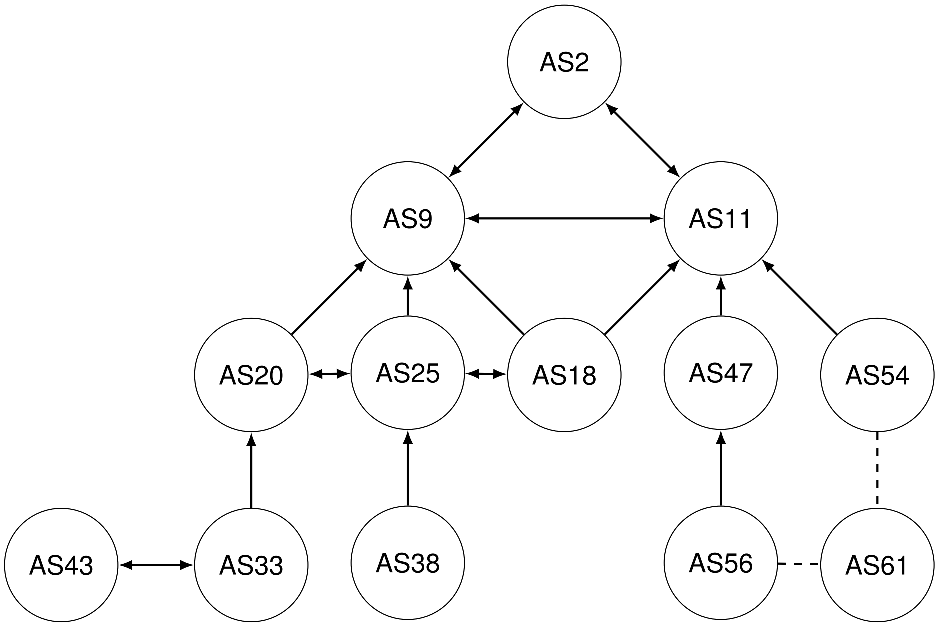

Figure 3.1: AS Network

a)* If one removes AS38 and AS43, is the first step of the k-core algorithm finished?

b)* Of which degree is the core of the network?

c)* In routing, what does FIB and RIB stand for and what is the difference?

0
1

d)* Briefly explain the term **Tier-1 provider** based on the lecture and name all Tier-1 providers in the given network.

0
1

e)* Briefly explain the term **stub network** based on the lecture and name all stub networks in the given network.

AS18 owns the prefix 10.0.0.0/22 and announces the prefix to the network.

0
1

f)* How is traffic from AS33 routed towards the announced prefix?

0
1

g)* Can AS43 reach the prefix?

0
1
2
3

h)* Given that AS61 is not a provider and has to send traffic towards the prefix 10.0.0.0/22 via AS56, which type of relationships does AS61 have with AS56 and AS54? Explain your answer.

0
1
2

i)* AS54 wants to hijack the prefix from AS18 and announces 10.0.0.0/21. Which other ASes are impacted by this announcement? Is this a successful hijack?

Problem 4 P4 (13 credits)

This problem investigates a Software-Defined Network (SDN) powered by P4. The source code of a P4 switch program is given in Listing 1.

```
header eth_t { bit<48> dstAddr;
               bit<48> srcAddr;
               bit<16> etherType; }

struct metadata { /* unused */ }

struct headers  { eth_t  eth; }

parser ParserImpl(packet_in packet, out headers hdr, inout metadata meta, inout standard_metadata_t
    standard_metadata) {

    state parse_eth { packet.extract(hdr.eth);
                      transition select(hdr.eth.etherType) { default: accept; }}
    state start     { transition parse_eth; }

}

control DeparserImpl(packet_out packet, in headers hdr) {
    apply { packet.emit(hdr.eth); }
}

control Pipeline(inout headers hdr, inout metadata meta, inout standard_metadata_t standard_metadata) {

    action my_drop()                { mark_to_drop(standard_metadata); }
    action set_egress(bit<9> port) { standard_metadata.egress_spec = port; }
    action set_default_egress()     { standard_metadata.egress_spec = 2; }
    action set_brdcast()            { \* __INSERT_P4_CODE_HERE__ *\ }

    table brdcast { actions = { set_brdcast; NoAction }
                    key =     { standard_metadata.ingress_port: exact; }
                    default_action = set_brdcast(); }

    table forward { actions = { set_egress; my_drop; set_default_egress; }
                    key =     { standard_metadata.ingress_port: exact; }
                    default_action = set_default_egress(); }

    apply { brdcast.apply();
            forward.apply(); }

}

V1Switch(ParserImpl(), Pipeline(), DeparserImpl()) main;
```

Listing 1: Simple P4 program

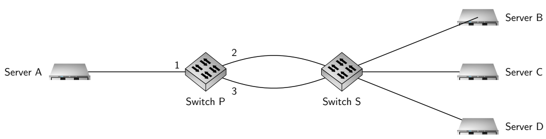

Figure 4.1: Network topology

For the following problems, the network topology is given in Figure 4.1. Switch P is a P4 switch running the P4 program of Listing 1 with Ports 1-3. Switch S is a regular, non-programmable switch. Switches P and S are connected via two links.

The network administrator of the network in Figure 4.1 wants to distribute all frames from Server A to Servers B, C, and D. Servers B, C, and D are configured to record each frame they receive. To distribute the frames, the two Switches P and S are used. Switch P uses the P4 program in Listing 1 that prepares frames for broadcasting, Switch S then performs the actual broadcast.

a)* Complete the `set_brdcast()` action in Listing 1.

The network administrator notices that both switches become overloaded within seconds after connecting all devices. A closer investigation uncovers that the servers do not create enough traffic to overload the network.

0 1 2

b)* Explain the reason for the overload in the network.

0 1 2 3

c) List possible solutions how this kind of Problem (cf. Subproblem b)) can be solved in general (not using an SDN-capable hardware but regular switches and routers). Name and briefly explain one solution for Layer 1, one solution for Layer 2, and one solution for Layer 3.

0 1 2 3

d)* The administrator does not rely on the general solutions mentioned in Subproblem c), but wants the servers to reach each other using the correct table entries for the given P4 program. **Hint:** The tables below may contain more rows than the number of rules actually required to perform the described task.

Brdcast Table

Match field(s)	Key	Action	Action data

Forward Table

Match field(s)	Key	Action	Action data

e)* Name the four P4 targets presented in the lecture.

0 1 2

f) A researcher wants to modify the physical layer of a P4 device. Which P4 target(s) allow(s) this kind of modification?

0 1

Problem 5 **Network Calculus (13.5 credits)**

This problem investigates performance bounds in networks using Network Calculus.
Consider the following network topology with flow and server descriptions. Flow f_1 traverses Servers s_1, s_2, and s_3. Flow f_2 traverses Servers s_1 and s_2. Assume each server handles flows according to strict priority scheduling with preemption. Assume Flow f_1 has a low priority and Flow f_2 has a high priority. In the following, we want to apply the Separate Flow Analysis to compute an end-to-end delay bound for Flow f_1.
Hint: Use the following formula to calculate a left-over service curve: $\beta^{l.o.} = \left[\beta_{R,T} - \gamma_{r,b}\right]^+ = \beta_{R-r,\frac{b+R\cdot T}{R-r}}$. An output arrival curve is given by $\alpha^\star = \alpha_{r,b+r\cdot T}$.

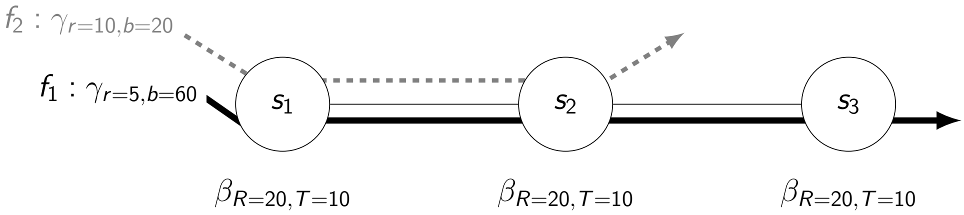

Figure 5.1: Network topology with flow and server specifications

a)* Perform the first step (left-over service curves) of the Separate Flow Analysis.

b) Perform the second step (end-to-end left-over service curve) of the Separate Flow Analysis.

c)* Perform the third step (calculate the delay bound) of the Separate Flow Analysis. If you have no solution to b), assume an end-to-end left-over service curve of $\beta_{R=6,T=7}$.

d)* Calculate the end-to-end delay bound of Flow f_2 using the Separate Flow Analysis.

e) Consider the following scenario for the network in Figure 5.1:

- The rate of all servers (s_1, s_2, and s_3) is set to $R = 12$.

Argue what the influence on the end-to-end delay bound of Flow f_1 is.

f)* Consider a token-bucket constrained flow, traversing a single rate-latency server. The corresponding arrival- and service curves are shown in Figure 5.2. Determine the backlog bound.

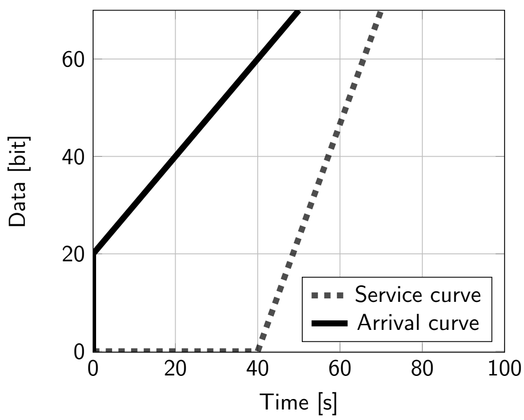

Figure 5.2: Arrival- and service curve

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