Chair of Network Architectures and Services
TUM School of Computation, Information, and Technology
Technical University of Munich

Esolution
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:**	Friday 17th February, 2023
Examiner:	Prof. Dr.-Ing. Georg Carle	**Time:**	08:00 – 09:15

Working instructions

- This exam consists of **16 pages** with a total of **6 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
- 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.
- 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 (9.5 credits)

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

a)* What is the correct short form of the following IPv6 address `2000:0000:0000:8080:0000:0000:0000:0110`?

- [ ] `2000::8080::110`
- [ ] `2000::8080:0:0:0110`
- [x] `2000:0:0:8080::110`
- [ ] `2000:0:0:8080::11`

b)* Which of the following is a correct destination MAC address for IPv6 neighbor solicitation?

- [ ] `33:ff:bb:b9:fd:7f`
- [x] `33:33:ff:b9:fd:7f`
- [ ] `ff:ff:ff:b9:fd:7f`
- [ ] `33:33:bb:b9:fd:7f`

c)* What is the originally specified unit of the TTL value in the IPv4 header?

- [ ] milliseconds
- [ ] hops
- [ ] router
- [x] seconds

d)* Features of which layers of the ISO-OSI model are integrated in the QUIC protocol?

- [ ] Layers 2,3, and 4
- [ ] Layers 1,2, and 3
- [x] Layer 4,5, and 7
- [ ] Layer 3,4, and 5

e)* Name two reasons from the lecture why Top Lists should be treated carefully (no explanation required).

Two out of: frequent changes over time, weekend effect, clustering effect, not always one million

f)* Shortly explain how ZMap verifies that incoming IP packets belong to the scan and the source IP address is correct even though it is stateless?

It encodes information into the request that is also visible in the response. The target IP address is encoded as TCP sequence number and the response used for validation.

g)* Briefly explain one reason a lame delegation might appear and what issues it might cause.

- NS record points to name server without providing DNS or without authoritative information on the zone
- As a consequence the zone might be unreachable

h)* Briefly explain the core concept behind HTTP-based load balancing.

Load balancer (or front end) redirects client to a content server (or backend) using HTTP status codes

Problem 2 DNS (8 credits)

The following questions will focus on principles in DNS.

1	example1.tum.de.	300	IN	NS	dns1.lrz.de.
2	example2.tum.de.	7200	IN	NS	dns2.lrz.bayern.
3	example3.tum.de.	86400	IN	NS	dns3.lrz.eu.
4	dns3.lrz.eu.	86400	IN	AAAA	2001:718:1:1f:50:56ff:feee:180
5	dns3.lrz.eu.	86400	IN	A	78.128.211.180
6	.	33554432	-	OPT	-

Table 2.1: Relevant DNS Records

a)* Given a resolver receives the records from Table 2.1. How long does the resolver cache each entry? Give the time in seconds/minutes/hours/days in the shortest form possible for each record. Use the record numbers in front of each record as the reference.

- record 1: 5 minutes
- record 2: 2 hours
- record 3: 1 day
- record 4: 1 day
- record 5: 1 day
- record 6: OPT records do not get cached

b)* Which NS record's TTL is the most reasonable according to the lecture's information?

NS records rarely change and therefore the largest TTL decreases the number of queries on the authoritative name server. Record 3 is the one with the largest TTL

c)* Assume the resolver receives a response with record *3* in the answer section and records *4* and *5* in the authority section. What purpose do records *4* and *5* serve and how are these records named?

Records 4 and 5 are glue records. They are needed for in-bailiwick delegations to resolve the circular dependency.

d)* Extract the client identifier in the record's data of record *4* and explain what type of identifier it is.

- Client identifier: `50:56ff:feee:180`
- `0xfffe` in the middle is an EUI-64 address

e)* Why does record *6* exist? Name two features this record can provide.
Hint: The record type of record *6* should only appear in the additional section.

- It is an OPT record which is the most important part of the extension mechanism for DNS (EDNS)
- Contains: maximum UDP payload size, extended RCODE, extension information, . . .

Problem 3 BGP and Traceroute (13 credits)

This problem investigates the autonomous system (AS) relationships in a given network and their impact on routing and traffic. Furthermore, traceroute is analyzed and evaluated.
Circles are individual routers or network devices. Large boxes are ASes. The relation between ASes depends on the link between border routers.

- $\rightarrow$ represents a customer $\rightarrow$ provider relationship
- $\leftrightarrow$ represents a peering relationship

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.

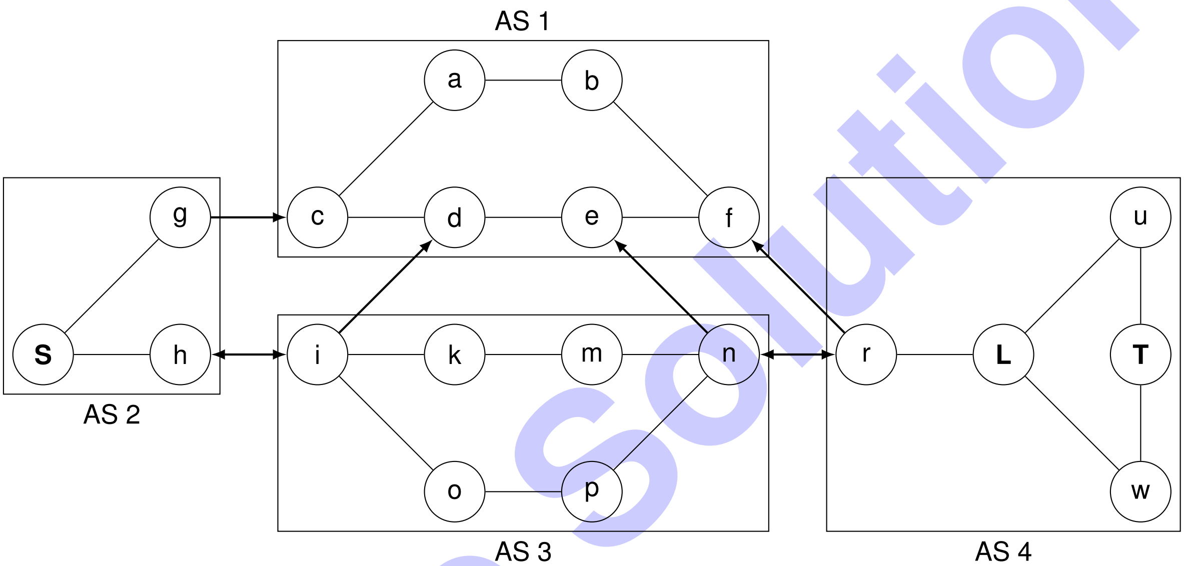

Figure 3.1: AS Network

0
1

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

- Tier-1: Default-Free-Zone, only peerings, no providers
- AS1

AS4 owns the prefix 172.0.0.0/23 and announces the prefix to the network.

0
1

b)* Which AS path takes traffic originated in AS2 towards the announced prefix?

- AS3 does not announce the route towards AS2. It would provide transit for AS2 without any benefit for AS 3.
- AS2 -> AS1 -> AS4

0
1

c) How could the previously uninvolved AS eavesdrop traffic from AS2 to AS4 by announcing address space itself? Describe one possible solution.

One out of:

- AS3 can announce the /23 to AS2, the link is cost free and thus preferable or
- AS3 can announce the two more specific /24 prefixes, the prefixes are more specific and thus preferable

And: Forward all traffic to AS4

d)* Which AS has the lowest k-core degree?

AS 2 with a k-core degree of 1, all others have a k-core degree of 2

e)* Shortly explain the concept of hot potato routing.

Always hand over traffic to another AS as fast as possible.

f)* Shortly explain how an IPv6 traceroute works:

- The sender sends IPv6 packets with increasing Hop Limit values (**not TTL**)
- The hop limit is decreased on the path
- Once the value is zero, the packet is discarded and an ICMPv6 error message is sent

Assume the source S in AS2 wants to trace the route towards the target T in AS4 for the following subproblems. Refer to specific nodes with the AS and hop name, e.g., AS2 S. The previous subproblem c) does **not** take effect here.

g)* How many steps are required for traceroute to reach the target?

With a TTL/Hop Limit of 9 the target is reached.
AS2 $S \rightarrow$ AS2 $g \rightarrow$ AS1 $c \rightarrow$ AS1 $a \rightarrow$ AS1 $b \rightarrow$ AS1 $f \rightarrow$ AS4 $r \rightarrow$ AS 4 $L \rightarrow$ AS4 $u \rightarrow$ AS4 T

h)* How many different routes can be reported by traceroute? Briefly explain your answer.

- There are 8 paths.
- There is a split at AS1 c and AS4 L with 2 potential paths each.
- Each probe is independent of each other, thus AS1 b and e can be reached with each hop beforehand (a, d).

i)* Name a path that could be reported by traceroute but does not actually exist.

The path should either contain AS1 $a \rightarrow$ AS1 e or AS1 $d \rightarrow$ AS1 b.
If someone routes via AS3, it should contain either AS3 $k \rightarrow$ AS3 p or AS3 $o \rightarrow$ AS3 m.

j)* Assume L in AS4 is a Layer 4 load balancer that always forwards the same connection via the same route. How can a sender make sure to identify specific routes with its traceroutes?

- Layer 4 means either TCP or UDP, thus connections are identified by its 5-Tuple
- The sender needs to use those protocols and send with fixed 5-Tuples

Problem 4 Network Calculus (11.5 credits)

This problem investigates performance bounds in networks using Network Calculus.

a)* Consider the following topologies. The servers use strict priority scheduling. Flow f_1 has a low priority, Flows f_2 and f_3 have a high priority.

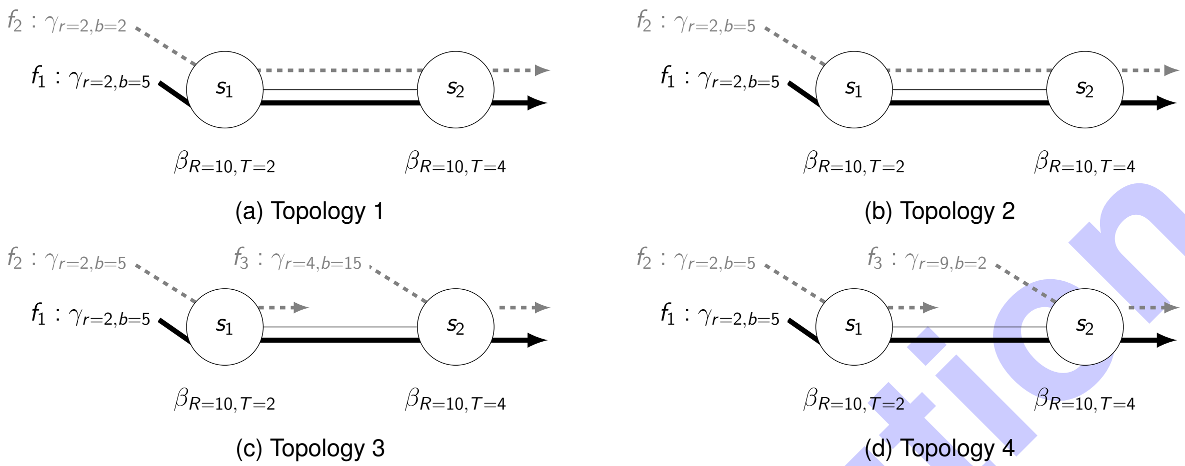

Figure 4.1: Four topologies

The following equations are part of a Separate Flow Analysis calculated from the point-of-view of **Flow f_1**.

- $\beta_{s1}^{l.o.<f_1>} = \left[\beta_{10,2} - \gamma_{2,5}\right]^+$
- $\beta_{s2}^{l.o.<f_1>} = \left[\beta_{10,4} - \gamma_{2,9}\right]^+$

The formulas are correct for **exactly one** topology. Argue **for each topology** why they are correct or incorrect.

- Topology 1: The output envelope of f_2 after s_1 is $\gamma^*_{r=2,b=6}$
- Topology 2: The output envelope of f_2 after s_1 is $\gamma^*_{r=2,b=9}$
- Topology 3: f_3 at s_2 has shape $\gamma_{r=4,b=15}$
- Topology 4: f_3 at s_2 has shape $\gamma_{r=9,b=2}$

$\Rightarrow$ Only Topology 2 matches both formulas

b)* Consider the topology in Figure 4.1a. The priorities stay the same (Flow f_1 has low priority, Flow f_2 has high priority). Calculate the delay bound of **Flow f_2** using the Separate Flow Analysis.

- Left-over service curves:

$\beta_{s_1}^{l.o.} = \beta_{s_1} = \beta_{10,2}$
$\beta_{s_2}^{l.o.} = \beta_{s_2} = \beta_{10,4}$

- Concatenation theorem:

$\beta_{e2e}^{l.o.} = \beta_{s_1}^{l.o.} \otimes \beta_{s_2}^{l.o.} = \beta_{min(10,10),2+4} = \beta_{10,6}$

- Delay bound:

$d = T + \frac{b}{R} = 6 + \frac{2}{10} = 6.2$

c)* Consider the following statement: 0 1

- A token-bucket arrival curve is defined as $\gamma_{r,b}(s) = r \cdot [s - b]^+$, where $[x]^+$ is $max(0, x)$, with rate parameter r and burst parameter b.

Argue whether or not the statement is correct.

Any one of:

- No, the definition of a token-bucket arrival curve is $\gamma_{r,b}(s) = r \cdot s + b$
- No, this is the definition of a rate-latency service curve
- No, token-bucket arrival curve has $\gamma(0) = b$

d)* Consider a token-bucket constrained flow, traversing a single rate-latency server. The corresponding arrival- and service curves are shown in Figure 4.2. Determine the **delay bound** and the **backlog bound including units**. 0 1 2

Visual determination:

- Delay bound: 80 s
- Backlog bound: 80 bit

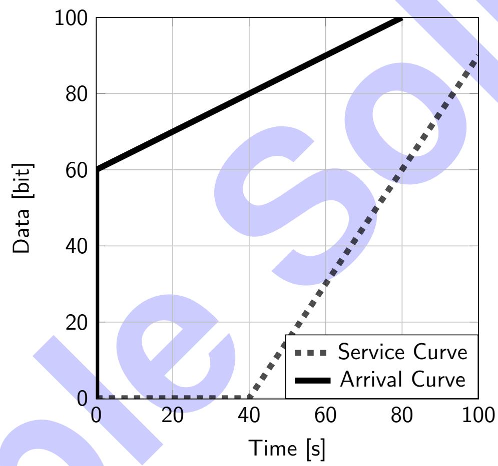

Figure 4.2: Arrival- and service curve

e)* Name one approach that can be used to obtain best-effort guarantees (according to the lecture) **and cannot** be used to obtain hard real-time gurantees. 0 1

Any one of: Simulation, Queuing Theory

f)* Consider two token-bucket constrained flows traversing a single rate-latency constrained server. Both flows have a rate of 10 Mbit/s and a burst of 50 kbit. The server has a rate of 15 Mbit/s and a processing latency of 200 µs. Assume arbitrary scheduling. What is the delay bound of the two flows respectively and why? 0 1 2

For both flows we calculate the left-over service curve, which has a rate of 5 Mbit/s. Both flows have a rate larger than 5 Mbit/s, therefore, $r > R$ and the delay bounds of both flows are infinity.

Problem 5 Hexdump (17.5 credits)

This problem investigates a captured Ethernet frame. The given hexdump starts with the Ethernet header in Figure 5.1.

```
                                     Source MAC              Ethertype
0x0000   3c ec ef 98 01 35 [3c ec    ef 98 06 b9] [08 00]  45 00

           Total Length              Protocol          Src IP Addr
0x0010   [05 1c] e1 64 40 00 40 [11]    40 6a [0a 00 00 01] 0a 00

                          Dst Port            Flags   Version      Dst CID Len
0x0020   00 02 8f 19 [01 bb] 05 08    19 1c [c1] [00 00 00 01] [08]

                       Dst CID
0x0030   [44 32 b8 bc 30 14 07 34]    . . .
```

Figure 5.1: Hexdump of an Ethernet frame starting with the Ethernet header. Only the first 56 B are displayed, the rest is omitted.

In this problem you **always** have to substantiate your answers using the bytes of the hexdump in Figure 5.1. Always make clear which bytes are relevant for each answer. 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] = 0x3cecef`. Note, counting starts at 0 and the noted numbers are included.
Hexadecimal notation is also allowed: e. g., `[0x10, 0x12] = 0x051ce1`.

0 1

a)* Mark the **Source MAC address** and note it in its common notation.

Source MAC: `[6, 11]= 0x3cecef9806b9`
`3c:ec:ef:98:06:b9`

0 1

b)* Identify the used **Layer 3** protocol. (Do not forget to mark and name relevant fields.)

Ethertype: `[12, 13]= 0x0800` ⇒ IPv4

0 1 2

c) Mark the **Source IP address** and note it in its common notation.

Source IP address: `[26, 29]= 0x0a000001`
`10.0.0.1`

d) Figure 5.1 only shows the first 56 B of the Ethernet frame. Determine the size of **the whole frame**. 0 1 2 3

Total length: `[16, 17]= 0x051c` $\Rightarrow$ 1308
Frame length = Total length + Ethernet header+ FCS= $1308 + 14 + 4 = 1326$

Alternative:
UDP length: `[38, 39] = 0x0508` $\Rightarrow$ 1288
Frame length = UDP length + IP header + Ethernet header + FCS = $1288 + 5 \cdot 4 + 14 + 4 = 1326$

e) Identify the used **Layer 4** protocol. 0 1

Protocol: `[23]= 0x11` $\Rightarrow$ UDP

f) Mark the **Destination Port** and note it in decimal notation. 0 1 2

Destination Port: `[36, 37]= 0x01bb`
`443`

g) Network connections usually have one client side and one server side. The client initiates the connection while the server waits for incoming connections. Argue, if the captured frame is sent from client to server or vice versa. 0 1

UDP destination port is in well known range (< 1024)
Frame is sent by the client.

h) Which protocol do you expect **on top of Layer 4**? 0 1

UDP port 443 $\Rightarrow$ QUIC

i) Mark **all** header fields of the protocol identified in Subproblem h), which are in the hexdump, and name them. 0 1 2 3 4

Flags: `[42]= 0xc1`
Version: `[43, 46]= 0x00000001`
Dst CID Len: `[47]= 0x08`
Dst CID: `[48, 55]= 0x4432b8bc30140734`

j) What is the purpose of the header field at position `[48, 55]` (dotted in Figure 5.1)? 0 1

It is the Destination Connection ID (CID).
QUIC uses CIDs to uniquely identify different connections.

Problem 6 Software-defined Networks (15.5 credits)

In this problem, we investigate techniques used in the area of Software-defined Networks (SDNs). We want to design a new network layer protocol called Network Universal Packet Service (NetUPS) using a novel, specific headers.

In the NetUPS protocol all packets, the so-called parcels, have a specific age. Every time a parcel is processed by a NetUPS-capable device the age is incremented by 1. The maximum age of parcels is 4095. In our implementation of NetUPS, all parcels with an age of 2048 or higher are forwarded to port `2` of the switch, the original EtherType is restored, and the NetUPS header is invalidated. All other parcels should be transmitted via port `3`. The source code of a NetUPS-enabled P4 switch is given in Listing 1.

```
header eth_t   { /* see Subproblem a) */ }

header net_ups { /* see Subproblem c) */ }

struct metadata { /* unused */ }

struct headers  { eth_t    eth;
                  net_ups  ups; }

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) { 0x88B5: parse_ups;
                                                             default: accept; } }
    state parse_ups { packet.extract(hdr.net_ups)
                      transition accept; }
    state start     { transition parse_eth; }

}

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

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

    action set_egress()           { standard_metadata.egress_spec = 2;
                                    hdr.eth_t.etherType = hdr.net_ups.orgEtherType;
                                    hdr.net_ups.setInvalid(); }
    action set_default_egress()   { hdr.net_ups.age = hdr.net_ups.age + 1;
                                    standard_metadata.egress_spec = 3; }

    table forward { actions = { set_egress; my_drop; set_default_egress; }
                    key =     { hdr.net_ups.age: ternary; }
                    default_action = set_default_egress(); }

    apply { forward.apply(); }

}

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

Listing 1: Simple P4 program

a)* The header fields for the Ethernet header (`header eth_t`) are missing from the program in Listing 1. Complete the code of `header eth_t`.

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

b)* What is the Ethertype used for NetUPS?

```
0x88B5
```

c)* The header fields for the NetUPS header (`header net_ups`) are missing from the program in Listing 1. Complete the code of `header net_ups`. **Hint:** Have a look at the `Pipeline` control block to extract the header information.

```
header net_ups  { bit<12> age;
                  bit<16> orgEtherType; }
```

The `forward` table in Listing 1 uses a ternary match. A ternary match is specified with a mask and a value. The mask defines the relevant bits for the ternary match (the relevant bits should be set to 1). The value defines if the relevant bits must be set to 1 or 0.

d)* Write down the mask and the value for the ternary match as **hexadecimal** numbers so the P4 program adheres to the specified behavior for processing NetUPS parcels . **Hint:** Have a look at the textual description of NetUPS and the `Pipeline` control block.

Mask: 0x800

Value: 0x800

e) Could table `forward` also use a different match type to realize the functionality? Discuss if this kind of match type would be more efficient than the ternary match.

- Yes, the functionality could also be realized using exact matches
- It would be less memory efficient because it would need a single entry for every parcel that is considered too old

f)* Briefly argue if OpenFlow-enabled switches can process NetUPS packets.

- OpenFlow switches do not know NetUPS's header structure, therefore, NetUPS is not supported
- The OpenFlow standard must be updated to allow NetUPS support

g)* Briefly argue if P4-enabled switches can process NetUPS packets.

- NetUPS' header could be defined as a P4 programm, therefore, NetUPS can be supported on P4 switches

0
1

h) P4 programs contain a programmable parser. What is the task of such a parser?

- The parser is used dissect the individual header fields

0
1
2

i) Argue why P4-enabled devices need such a programmable parser whereas OpenFlow-enabled devices do not.

- P4 programs need programmable parsers to define headers of novel protocols.
- OpenFlow programms only operate on known protocols, therefore, a parser only needs to operate on known protocols

0
1

j) What needs to be done to include support for NetUPS in OpenFlow switches?

- NetUPS needs to be included in the OpenFlow Standard and the switches need to be updated

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