Chair of Network Architectures and Services
Department of Informatics
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:**	Thursday 4th March, 2021
Examiner:	Prof. Dr.-Ing. Georg Carle	**Time:**	11:00 – 12:15

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.
- Removing pages from the exam is prohibited.
- Allowed resources:
  - one **analog dictionary** English ↔ native language
  - the provided cheat sheet without any annotations
- 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.

Left room from _____ to _____ / Early submission at _____

Problem 1 P4 (16.5 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; }

header ip6_t { bit<4>   version;
               bit<8>   trafficclass;
               bit<20>  flowlabel;
               bit<16>  payloadlength;
               bit<8>   nextheader;
               bit<8>   hoplimit;
               bit<128> srcAddr;
               bit<128> dstAddr; }

header udp_t { bit<16> srcPort;
               bit<16> dstPort;
               bit<16> length;
               bit<16> checksum; }

struct metadata { /* unused */ }

struct headers  { eth_t  eth;
                  ip6_t  ipv6;
                  udp_t  udp; }

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

    state parse_udp { packet.extract(hdr.udp);
                      transition accept; }

    state parse_ip6 { packet.extract(hdr.ipv6);
                      transition select(hdr.ipv6.nextheader) { 0x11:    parse_udp;
                                                               default: accept; }}
    state parse_eth { packet.extract(hdr.eth);
                      transition select(hdr.eth.etherType) { 0x86dd:  parse_ip6;
                                                             default: accept; }}
    state start     { transition parse_eth; }

}

control DeparserImpl(packet_out packet, in headers hdr) {
    apply { packet.emit(hdr.eth);
            packet.emit(hdr.ipv6);
            packet.emit(hdr.udp); }
}

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; }

    table filter { actions = { set_egress; my_drop; set_default_egress; }
                   key =     { hdr.udp.dstPort: exact; }
                   default_action = set_default_egress(); }

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

    apply { if (hdr.udp.isValid()) {
                filter.apply();
            } else if (hdr.eth.isValid()) {
                forward.apply();
            }
    }

}

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

Listing 1: Simple P4 program

For the following problems use the network topology given in Figure 1.1. Switch S is a P4 switch running the P4 program of Listing 1.

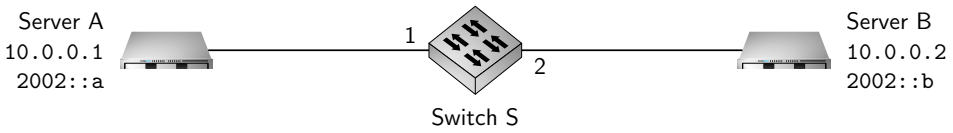


Figure 1.1: Network topology

Match field(s)	Key	Action	Action data
`standard_metadata.ingress_port`	2	`set_egress`	1

Table 1.1: Rule entered into the table `forward`

For the following subproblems you can assume that Servers A and B know the MAC address of each other, i.e., you do not need to describe or consider address resolution in your answers.

The administrator of the network in Figure 1.1 wants test the connectivity in his network using `ping6`. Therefore, he executes the command `ping6 2002::b` on Server A.

a)* What are the protocols used for the packet created by mentioned `ping6` command?

0
1

Ethernet, IPv6, ICMPv6

b) Explain the path of the previously parsed packet through the `ParserImpl` of the P4 program in Listing 1, mentioning all passed states and the decisions taken in these states.

0
1
2

- Path begins at `start`
- Path continues in `parse_eth`, where the Ethertype of IPv6 is cecked and `parse_ip6` is selected
- In `pars_ip6`, the default path is taken, as the ICMPv6 protocol number is contained in the nextheader field

c) Explain the path of the packet through the `Pipeline` of the P4 program in Listing 1, mentioning all passed decisions, tables, table entries, actions, and what happens to the packet.

0
1
2
3

- Path begins at `apply()`
- Table `forward` is applied, because udp header is not valid but the Ethernet header is
- `ingress_port` is 1, no more specific table entry applies and the default action is applied
- `set_default_egress` is executed and packet is sent out on port 2

0 1 2

d) Explain the differences of the packet's path through the P4 program if the administrator executes `ping6 2002::a` on Server B.

- Path through parser is the same, also path towards the `forward` table
- `ingress_port` is 2 this time, therefore the more specific entry in Table 1.1 is chosen
- `set_egress` is executed and packet is sent out on port 1

The administrator wants to test the connectivity of a webserver running on Server A from Server B. Therefore, he uses the QUIC protocol (on IPv6).

0 1 2 3 4

e)* Create an entry in Table `filter` so the QUIC packets sent by Server B can reach Server A. **Hint:** The table below may contain more rows than the number of rules actually required to perform the described task.

Match field(s)	Key	Action	Action data
`hdr.udp.dstPort`	`443`	`set_egress`	`1`

0 1 2

f) Explain what happens to the reply message of Server A to the QUIC request message of Server B on Switch S?

- Answer of A is sent to the UDP source Port chosen by Server B
- By default no rule given, therefore, the default rule is executed
- Sent back to Server B (default rule)

0 1 2

g)* The administrator of the network in Figure 1.1 wants to block ssh. Reason why he can or cannot specifically block ssh (and no other protocols) using only table entries on Switch S?

- the P4 program can filter only on UDP ports
- sshuses TCP, therefore the given P4 program cannot be used to block ssh

Problem 2 Quiz (17 credits)

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

a)* Modern network interface cards (NICs) use standardized switchable transceivers, such as the SFP module. What is a benefit of such a system compared to a NIC with a fixed transceiver?

0 1

- Support for multiple systems using different physical media on the same NIC, i.e. multi mode fiber, single mode fiber, coax
- Replacing of defect lasers/LEDs for fiber based systems

b)* Given the IP address `10.0.21.16` and subnet mask `255.255.255.192`, determine the corresponding network and broadcast addresses.

0 1 2

Netmask `255.255.255.192` corresponds to a prefix length of 26, leaving 6 bit for the host part, i. e., the subnet has a total of 64 addresses. `10.0.21.16` thus belongs to the first subnet starting at `10.0.21.0` (network address) and ending at `10.0.21.63` (broadcast address).

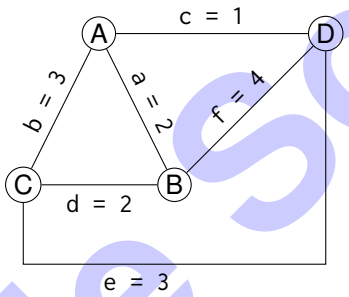


Figure 2.1: Network topology

c)* Figure 2.1 shows a network topology with redundant connections and their corresponding costs. List the edges that are path of a minimum spanning tree on this topology.

0 1

c, a, d

d)* List the edges that would be part of the resulting graph if the spanning tree protocol (STP) is performed on the topology in Figure 2.1 (with A as its root bridge).

0 1

b, a, c

e)* How does the memory consumption change for a growing number of entries in a routing table using the DIR-24-8 algorithm?

0 1

Memory is preallocated in DIR-24-8 (`table_24`), memory consumption constant for a growing number of entries. (The second table (`table_8`) may be preallocated or can increase linearly with new table entries.)

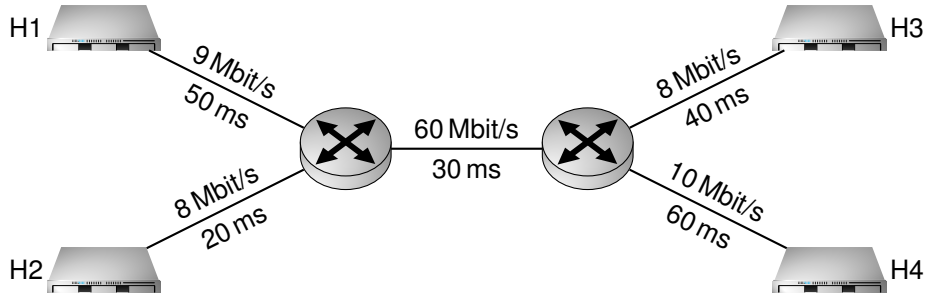


Figure 2.2: The latency and capacity is given for each link in the network.

0 1 2 3

f)* Consider the network topology given in Figure 2.2.

- Flow F_1 from Host H1 to Host H4
- Flow F_2 from Host H2 to Host H3.

Compute the BDP for F_1 and F_2 in kbit.

$F_1 = (50\,\text{ms} + 30\,\text{ms} + 60\,\text{ms}) \cdot \min(9\,\text{Mbit/s}, 60\,\text{Mbit/s}, 10\,\text{Mbit/s}) = 140\,\text{ms} \cdot 9\,\text{Mbit/s} = 1260\,\text{kbit}$
$F_2 = (20\,\text{ms} + 30\,\text{ms} + 40\,\text{ms}) \cdot \min(8\,\text{Mbit/s}, 60\,\text{Mbit/s}, 8\,\text{Mbit/s}) = 90\,\text{ms} \cdot 8\,\text{Mbit/s} = 720\,\text{kbit}$

0 1 2

g)* Briefly explain how Head of Line (HoL) blocking can cause trouble when running HTTP via a single TCP connection **and** how QUIC tries to solve this problem.

- all elements are multiplexed into one connection
- if one segment is lost, all fully arrived elementshave to wait until the segment is retransmitted
- QUIC uses multiple parallel streams, which do not block each other

0 1

h)* Explain when glue records are required for a zone's successful resolution.

- glue records are required if the zone's nameserver is in bailiwick. I.e. the nameserver name is a subdomain of the zones apex.

0 1

i)* DNS is based on a resilient architecture. Shortly name one resilience requirement for a zone presented in the lecture.

- two nameserver (names and/or IP addresses)
- IP addresses to be topological divers
- IP addresses to be geographical divers

j)* Consider the following command outputs: One is executed from a host in Munich while the other is executed from a host in Frankfurt.

0 1 2

```
Munich ~$ dig +short example.com
52.208.128.101
52.18.15.9
52.19.40.147
```

Listing 2: Command execution from Munich

```
Frankfurt ~$ dig +short example.com
54.76.83.27
54.77.81.254
54.77.186.213
```

Listing 3: Command execution from Frankfurt

Describe the DNS-based load balancing mechanisms deployed by `example.com` that can be directly derived from the commands and their outputs.

- Geo based load balancing. Depends on your geographical location
- Multiple DNS entries. Client should select randomly

k)* Describe two effects (besides the simple traffic increase) of the COVID-19 pandemic on the Internet, which were presented during the lecture.

0 1 2

- Weekend-Weekday traffic pattern. Traffic behaved on weekdays like on weekends.
- Hypergiants do not see as much traffic increase as other ASes.
- QUIC and NAT traversal traffic increased while ESP and GRE decreased
- The number of assumed gaming households doubled
- Mobility of Londoners. typical weekend increase disappeared. A lot of mobility in rural districts where Londoners fled during lockdown
- ...

Problem 3 Wireshark (15.5 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.

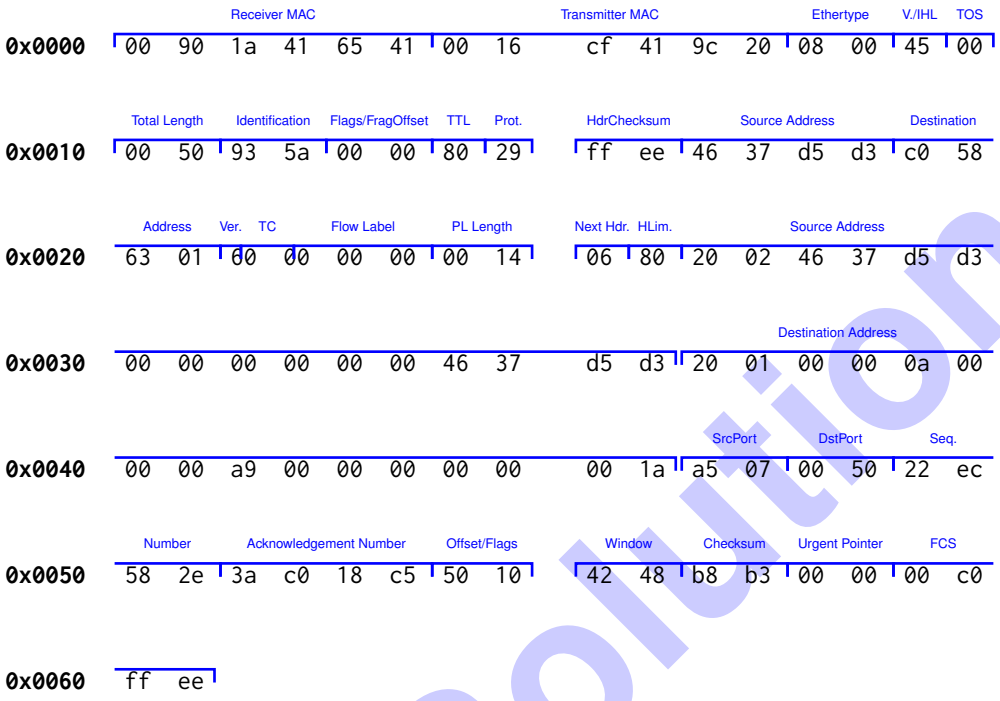


Figure 3.1: Hexdump of a complete Ethernet frame including FCS

For this problem you must indicate the location of the bytes in the hexdump in Figure 3.1. You can **either** mark the corresponding bytes directly in the figure **or** list the locations of the corresponding bytes using **[ ]**. Example: the bytes from position 0 to 2 can be written as `[0, 2] = 0x00901a`.

0 1 2

a)* Mark and name all parts of the protocol specific information for layer 2 for the hexdump in Figure 3.1.

`[0, 5]` Destination Address, `[6, 11]` Source Address, `[12, 13]` Ethertype, `[94, 97]` FCS

In the next six subproblems you are asked to identify which protocols were used for each layer. For each question do the following:

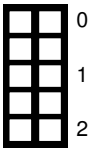
- **either** mark the corresponding bytes in the hexdump **or** list the bytes using the `[ ]` notation,
- **and** write the corresponding bytes in the solutionbox.

0 1

b)* Name the L3 protocol.

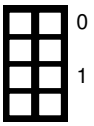
`[12, 13]` Ethertype: `0x0800`
IPv4

c) What is the length (in bytes) of the header of the protocol identified in Subproblem b).



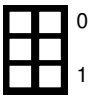
[14] Protocol: 0x5
5 (32-byte words) = 20 B

d) Name the protocol of the payload of the protocol identified in Subproblem b).



[23] Protocol: 0x29
IPv6

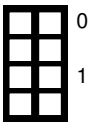
e) Write down the destination address of the payload of the protocol identified in Subproblem d).



[58, 73] Address: 2001:0000:0a00:0000:a900:0000:0000:001a

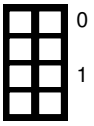
If you could not solve the previous subproblems, you can start with the evaluation of the TCP header at the marked position in Figure 3.1.

f)* Identify the application layer protocol.



[76] Destination Port 0x50
HTTP

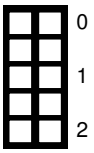
g)* Was the hexdump taken during the connection setup or teardown of the connection?



[87] Flags 0x10
No FIN/SYN flag set (only ACK), this means the packet was not part of connection setup or teardown

For the following questions you do **NOT** need to mark/list the corresponding bytes in the hexdump.

h) Note the address of Subproblem e) in the **shortest** way possible.

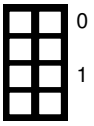


Address: 2001:0:a00:0:a900::1a

i) Explain a situation where a combination of protocols identified in Subproblems b) and d) may be necessary.

Provide IPv6 connectivity over a network where only IPv4 is available.

j) Name all the checksums contained in the hexdump in Figure 3.1.



Ethernet FCS, IPv4 header checksum, TCP checksum

Problem 4 AS Relations and BGP (14.5 credits)

This problem investigates the autonomous system (AS) relationships in a given network and their impact on routing and traffic. All ASes apply standard routing behavior. Furthermore, the following policies are applied:

- If routes with the same prefix exist, an AS selects the cost-efficient route.
- If routes with the same prefix exist, and the cost to route traffic is equal for all routes, the shorter route is selected.

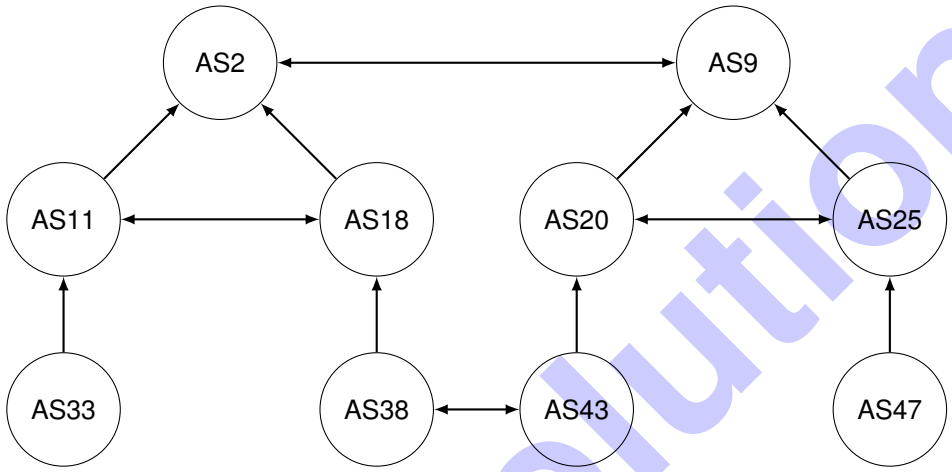


Figure 4.1: AS Network

0 1 2 a)* Calculate the k-core of the given graph. What is the degree of the k-core? List all removed nodes and briefly explain your steps.

- Find the maximal connected subgraph with degree at least k
  - Start with k = 1
  - Step 1: Remove all nodes of degree k recursively
  - Step 2: If there are nodes left, increase k by 1 and continue, otherwise the core is k - 1
- The core is 1, Removed nodes: AS33 and AS47

0 1 b)* 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
- AS2 and AS9

0 1 c)* Briefly explain the term **stub network** based on the lecture and name all stub networks in the given network.

- Stub ASes have exactly one BGP relationship (with their provider)
- AS33 and AS47

AS47 owns the prefix 10.0.0.0/22 and announces the prefix to the network.

d)* How is traffic from AS20 routed towards the announced prefix? Briefly explain your answers.

0 1

- AS25 announces routes towards its customer to its peer AS20
- AS20 -> AS25 -> AS47

e)* How is traffic from AS38 routed towards the announced prefix? Briefly explain your answers.

0 1

- AS38 has no route via AS43, which needs to pay its provider. AS38 only learns a route from its provider.
- AS38 -> AS18 -> AS2 -> AS9 -> AS25 -> AS47

f)* How is traffic from AS33 routed towards the announced prefix? Briefly explain your answers.

0 1

- AS33 only connects to AS11. AS11 only knows a route via its provider. AS18 does not announce the prefix towards AS11.
- AS33 -> AS11 -> AS2 -> AS9 -> AS25 -> AS47

g)* AS18 wants to *eavesdrop* traffic from AS33 towards prefix 10.0.0.0/22 owned by AS47 without **hijacking** the prefix. Is it possible for AS18 to *eavesdrop* the traffic. If yes, explain how.

0 1 2

- Yes AS18 can eavesdrop the traffic
- AS18 can announce its route to AS47 (AS18 -> AS2 -> AS9 -> AS25 -> AS47) to AS11
- AS11 prefers this route over AS1 because it is a peering session and thus cheaper.
- AS18 needs to handle more traffic and pay more but can eavesdrop the traffic

h)* AS18 wants to hijack the prefix from AS47 and announces 10.0.0.0/23. Which other ASes are impacted by this announcement? Is this a successful hijack of 10.0.0.0/22? Briefly explain your answer.

0 1 2

- All other ASes are impacted, the announcement is more specific and will thus be accepted
- But the announced prefix does not cover the complete /22 and only hijacks part of it

i) Announcements from AS18 propagate through the network and also reach AS47. AS47 realizes this is a hijack and wants to protect itself. Which BGP announcements can AS47 send to reclaim the hijacked prefix throughout the complete network. Briefly explain your answer.

0 1 2

- AS47 can announce more specific prefixes as well
- It has to announce 10.0.1.0/24 and 10.0.0.0/24 to cover the complete /23 hijacked by 18

Problem 5 Network Calculus (11.5 credits)

This problem investigates Network Calculus and its applications to determine delay bounds in networks. Always document your approach and simplify terms as much as possible, unless specified otherwise.

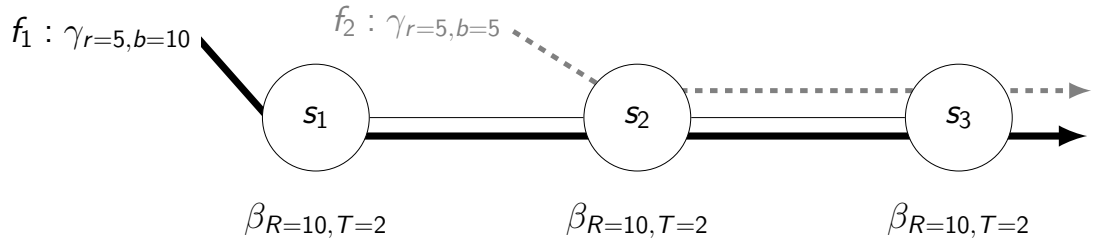


Figure 5.1: Network topology and flow description

Let the output envelope of a token-bucket constrained flow $\gamma_{r,b}$ traversing a rate-latency server $\beta_{R,T}$ be defined as $(\gamma_{r,b} \oslash \beta_{R,T})(t) = \gamma_{r,b+r\cdot T}(t)$
Let the left-over service curve for a token-bucket constrained flow and a rate-latency server be defined as $\beta^{l.o.} = \left[\beta_{R,T} - \gamma_{r,b}\right]^+ = \beta_{R-r,\frac{b+R\cdot T}{R-r}}$
Consider the network shown in Figure 5.1. Assume preemptive Strict Priority Queuing at each server. Flow f_1 has a low priority and Flow f_2 has a high priority. We are interested in calculating the delay bound for Flow f_1 using the Separate Flow Analysis.

Note: You are not required to use special symbols (e.g. $\beta_{R,T}$ and $\otimes$). But make sure your notation is consistent and understandable (e.g., *beta_R,T* and *convolution*).

0 1 2 3 4

a)* Perform the first step of the Separate Flow Analysis.

- $\beta^{l.o}_{s_1} = \beta_{10,2}$
- $\beta^{l.o}_{s_2} = \left[\beta_{10,2} - \gamma_{5,5}\right]^+ = \beta_{10-5,\frac{5+10\cdot 2}{10-5}} = \beta_{5,5}$
- $\beta^{l.o}_{s_3} = \left[\beta_{10,2} - \gamma^{\star}_{5,5}\right]^+ = \left[\beta_{10,2} - \gamma_{5,5+5\cdot 2}\right]^+ = \left[\beta_{10,2} - \gamma_{5,15}\right]^+ = \beta_{10-5,\frac{15+10\cdot 2}{10-5}} = \beta_{5,7}$

0 1

b) Perform the second step of the Separate Flow Analysis.

$\beta_{e2e} = \beta^{l.o.}_{s_1} \otimes \beta^{l.o.}_{s_2} \otimes \beta^{l.o.}_{s_3} = \beta_{min(R^{l.o.}_1, R^{l.o.}_2, R^{l.o.}_3), T^{l.o.}_1 + T^{l.o.}_2 + T^{l.o.}_3} = \beta_{5,14}$

0 1

c) Perform the third step of the Separate Flow Analysis.

$d = T_{e2e} + \frac{b_{f_1}}{R_{e2e}} = 14 + \frac{10}{5} = 16$

d) Assume two separate scenarios:

0
1

1. The service curve of only s_2 is changed to $\beta_{R=10,T=4}$
2. The service curve of only s_3 is changed to $\beta_{R=10,T=4}$

Briefly argue which scenario produces the larger delay bound for Flow f_1 under the Separate Flow Analysis.

Scenario 1 produces a larger delay bound since the change in service curve affects the calculation of the output envelope of f_2 after s_2 and, therefore, the left-over service curve at s_3 in addition to the service curve at s_2.

e)* Why is it important to distinguish between preemptive and non-preemptive scheduling when performing the Separate Flow Analysis?

0
1

The left-over service curveneeds to consider (maximum) frame/packet sizes under non-preemptive scheduling.

f)* What is given by the maximum vertical distance between service curve and arrival curve?

0
1

Backlog bound or buffer bound

g)* Assume a token-bucket constrained flow $\gamma_{r=10,b=50}$ traversing a single rate-latency server $\beta_{R=12,T=8}$. Assume arbitrary multiplexing. Argue whether or not a finite delay bound for the flow can be computed.

0
1

The service curve rate is larger than the arrival curve rate.Therefore, the maximum horizontal distance is a finite value or the two curves intersect at $t \neq 0$.

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