Exam: Advanced Topics in Communication Networks

3 February 2022, 09:00–11:00, Room HG E 7

▷ You must wear a face mask at all times during the exam, except for short eating and drinking breaks. Only medical (IIR) and FFP2 masks are allowed. Contact the assistants in case you need a spare mask.

▷ Write your name and your ETH student number below on this front page and sign it.
▷ Put your legitimation card on the most accessible corner of your desk. Make sure that the side containing your name and student number is visible.
▷ Verify that you have received all task sheets (Pages 1 - 32).
▷ Do not separate the task sheets. We will collect the exams only after you have left the room.

▷ Write your answers directly on the task sheets.
▷ All answers fit within the allocated space—often in much less.
▷ If you need more space, use the extra sheets at the end of the exam. Indicate the task in the corresponding field.
▷ Read each task completely before you start solving it.
▷ It is not required to score all points to get the best mark.

▷ Answer in English.
▷ Write clearly in blue or black ink (not red) using a pen, not a pencil.
▷ Cancel invalid parts of your solutions clearly (e.g., by crossing them out).
▷ At the end of the exam, place the exam face up on the most accessible corner of your desk. Then collect all your belongings and exit the room according to the given instructions.

▷ No written material nor calculator are allowed.

Family name: 
Student legi nr.: 
First name: 
Signature: 

Do not write in the table below (used by correctors only):

<table>
<thead>
<tr>
<th>Task</th>
<th>Points</th>
<th>Sig.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Programmable data planes</td>
<td>/25</td>
<td></td>
</tr>
<tr>
<td>Managing network traffic</td>
<td>/20</td>
<td></td>
</tr>
<tr>
<td>Optimizing network performance</td>
<td>/25</td>
<td></td>
</tr>
<tr>
<td>Design question</td>
<td>/50</td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>/120</td>
<td></td>
</tr>
</tbody>
</table>
Task 1: Programmable data planes 25 Points

a) General Questions (8 Points)

For each of the following statements, indicate whether they are true or false. There is always one correct answer. Each block of questions is awarded up to 4 points: 4 points for 4 correct answers, 2 points for 3 correct answers, and 0 points otherwise.

(i) P4 language, architecture, and programs (4 Points)

- No part of a P4 program can contain loops.
  - true □ false □

- P4 supports variable length headers and a variable number of headers.
  - true □ false □

- The `extern` construct provides an interface to functionalities that are not implemented in the P4 program itself.
  - true □ false □

- Any correct P4 program can be compiled for any P4 device.
  - true □ false □

(ii) Load balancing (4 Points)

Programmable data planes enable a variety of load balancing decisions. Consider the network below, where all links have a capacity of 10 Mbps, equal delay, and equal IGP weights (e.g. equal OSPF cost).

10 Mbps of TCP traffic are entering at A and B respectively, both destined for C. Assume the traffic to consist of many flows.

```
10 Mbps
A
10 Mbps
B
```

- In the network above, ECMP ensures optimal throughput.
  - true □ false □

- In the network above, randomly assigning flowlets to equal-cost paths leads to better throughput than ECMP on average.
  - true □ false □

- In general, traffic created by any transport protocol leads to flowlets whose size reacts to congestion.
  - true □ false □

- Per-packet load balancing prevents reordering packets in the network.
  - true □ false □
b) **Probabilistic data structures**

You are given a Bloom filter that uses 10 cells and 3 hash functions. The hash functions for this task are defined as follows:

\[
\begin{align*}
hash_1(x) &= x \mod 10 \\
hash_2(x) &= 3 \cdot (x + 1) \mod 10 \\
hash_3(x) &= 5 \cdot (x + 1) \mod 10
\end{align*}
\]

where \(x\) is the numerical input value. In this task, we will input words, and use the sum of the letters’ ASCII values as \(x\). For example, the word *AdvNet* has value 578. The numerical values you need are provided in Table 1.

<table>
<thead>
<tr>
<th></th>
<th>(x)</th>
<th>(3 \cdot (x + 1))</th>
<th>(5 \cdot (x + 1))</th>
</tr>
</thead>
<tbody>
<tr>
<td>AdvNet</td>
<td>578</td>
<td>1737</td>
<td>2895</td>
</tr>
<tr>
<td>exam</td>
<td>427</td>
<td>1284</td>
<td>2140</td>
</tr>
<tr>
<td>easy</td>
<td>434</td>
<td>1305</td>
<td>2175</td>
</tr>
<tr>
<td>hard</td>
<td>415</td>
<td>1248</td>
<td>2080</td>
</tr>
</tbody>
</table>

Table 1: Numerical word values.

(i) Start with an empty Bloom filter and update it by inserting the words *AdvNet, exam,* and *easy* one after another. Use the table below to indicate the state of the Bloom filter after each word. (3 Points)

<table>
<thead>
<tr>
<th>(empty)</th>
<th>AdvNet</th>
<th>exam</th>
<th>easy</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(ii) After inserting all words, query the Bloom filter for the word *hard*, and explain the result: what happened, and why? (3 Points)

Query result for *hard*: □ true □ false

Explain the result: ______________________________________

________________________________________________________________

________________________________________________________________

________________________________________________________________

________________________________________________________________

(iii) You decide to improve the overall performance of the Bloom filter by using a fourth hash function in addition to the current hash functions:

\[ \text{hash}_4(x) = 7 \cdot (x + 1) \mod 10 \]

Again, insert the words \textit{AdvNet}, \textit{exam}, and \textit{easy} into an empty Bloom filter. Use the table below to indicate the state of the Bloom filter after each word. Afterwards, query the Bloom filter for the word \textit{hard}. The numerical values you need are provided in table 2. (2 Points)

\[
\begin{array}{cccc}
\text{x} & 3 \cdot (x + 1) & 5 \cdot (x + 1) & 7 \cdot (x + 1) \\
\hline
\text{AdvNet} & 578 & 1737 & 2895 & 4053 \\
\text{exam} & 427 & 1284 & 2140 & 2996 \\
\text{easy} & 434 & 1305 & 2175 & 3045 \\
\text{hard} & 415 & 1248 & 2080 & 2912 \\
\end{array}
\]

Table 2: Numerical word values.

<table>
<thead>
<tr>
<th>(empty)</th>
<th>AdvNet</th>
<th>exam</th>
<th>easy</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Query result for \textit{hard}: \(\square\) \(\square\)

(iv) Do additional hash functions \textit{always} improve a Bloom filter? Explain your reasoning. (2 Points)
(v) Consider the following code taken from a P4 program that implements the Bloom filter above. Your final task is to modify this program to turn the Bloom filter into a CountMin sketch. Assume that you will need to count $2^{16}$ elements at most.

For each of the three sections of the program (metadata and register declaration, insert, and query), explain: (i) which parts of the code can remain the same, and (ii) which parts require changes, and how they need to be changed. Motivate your answers.

Only explain. Do not write P4 code.

(7 Points)
Metadata and register declaration: ________________________________


insert action: __________


query action: __________


a) Label switching theory

For each of the following statements, indicate whether they are true or false. There is always one correct answer. Each block of questions is awarded up to 4 points: 4 points for 4 correct answers, 2 points for 3 correct answers, and 0 points otherwise.

true □ false □ The Resource Reservation Protocol (RSVP) establishes Label Switched Paths (LSPs) bidirectionally.

true □ false □ A Label Switched Router (LSR) may assign multiple labels for a particular address prefix.

true □ false □ MPLS allows for both traffic aggregation and disaggregation, whereas classical, destination-based IP forwarding allows for aggregation only.

true □ false □ Only the Egress Label Edge Routers can remove MPLS labels.
b) **Label switching application**  
(16 Points)

This subtask is based on the topology shown in Figure 1.

![Network Topology Diagram]

Figure 1: The topology of the network. Link annotations indicate the link’s bidirectional capacity, and all links have the same delay.

(i) Consider the network topology shown in Figure 1. Link annotations indicate the link’s bidirectional capacity, and all links have the same delay. Find the shortest constrained path(s) for an aggregated traffic demand of 75 Gbps from Router A to Router I. Describe the steps of your reasoning.  

Path:  
Algorithm steps:
(ii) Consider the label forwarding tables shown in Table 3. In this forwarding state, there exists at least two paths from Router A to Router I which are non-overlapping; i.e., the LSP tunnels do not share any edge. For a packet entering in Router A, find two possible labels (one for each path) that will eventually deliver the packet to Router I. Indicate the path taken for each case with the associated label headers at each hop. (6 Points)

**Hint:** For example, a packet initially labeled with $L1$ entering in Router B terminates at Router H with $L5$. The path taken in this case is as follows:

$L1 \to (B) \to L3, L1 \to (C) \to L1 \to (E) \to L5 \to (H)$

Path 1: 

Path 2:
(iii) Explain one purpose of label stacking. Then, find two links with label stacking based on Table 3. How do the stacked labels on these links help with the indicated purpose?

(4 Points)

Purpose: 

__________________________________________________________________________

__________________________________________________________________________

__________________________________________________________________________

__________________________________________________________________________

Links: 

__________________________________________________________________________

__________________________________________________________________________

__________________________________________________________________________

__________________________________________________________________________

How do they help: 

__________________________________________________________________________

__________________________________________________________________________

__________________________________________________________________________

__________________________________________________________________________

(iv) Find one forwarding loop in the network. (3 Points)

*Hint: One loop includes Router F.*

__________________________________________________________________________

__________________________________________________________________________

__________________________________________________________________________

__________________________________________________________________________
February 2022

Exam: Advanced Topics in Communication Networks

Task 3: Optimizing network performance 25 Points

a) QoS-enabled router (10 Points)

Due to the recent remote-working restrictions, you find yourself at home sharing your Internet connection with your parents and your brother. Even though you are subscribed to the maximum capacity that your ISP can provide, you identify congestion in the upstream of your home router (see Figure 2). Being a student of the Adv-Net lecture, you decide to put in practice what you have learned about Quality of Service (QoS) to handle the congestion.

| Mother | Download (FTP) |
| Father | VoIP, Web browsing |
| Brother | Video game |
| You | Video streaming |

Table 4: Used applications

Figure 2: Router setup

(i) First, you verify that your home router is QoS-enabled. What does it mean for a router to be QoS-enabled? Which components do you expect a QoS-enabled router to have? Describe briefly the purpose of each component. (3 Points)

QoS-enabled meaning:


QoS components:
(ii) For the Download and VoIP applications, how important is it to be served with low delay and high bandwidth? Classify as “low/medium/high” and justify your answers. What QoS policy would you apply in order to optimize the QoS across these two applications? (3 Points)

Importance of delay for VoIP: ____________________________

Importance of bandwidth for VoIP: ____________________________

Justification: ____________________________________________

Importance of delay for Download: ____________________________

Importance of bandwidth for Download: ____________________________

QoS policy: ____________________________________________

(iii) The first step in implementing QoS at the router is to classify the ingress traffic of each application. What traffic features would you use to classify the traffic from Download and VoIP applications? Justify your answer. (2 Points)

_____________________________________________________

_____________________________________________________

_____________________________________________________

_____________________________________________________
(iv) You want to reduce the quality of your video-streaming service from its default 4K resolution, to Full-HD (FHD). You want to do that from the router. Explain what QoS technique would you use, and why. In Table 5, you will find the required bandwidth of each resolution for the codec used by your video-streaming service (H.265). (2 Points)

<table>
<thead>
<tr>
<th>Resolution</th>
<th>Bandwidth</th>
</tr>
</thead>
<tbody>
<tr>
<td>1280×720 (HD)</td>
<td>3Mbps</td>
</tr>
<tr>
<td>1920×1080 (FHD)</td>
<td>6Mbps</td>
</tr>
<tr>
<td>3840×2160 (UHD)</td>
<td>25Mbps</td>
</tr>
<tr>
<td>4096×2160 (4K)</td>
<td>32Mbps</td>
</tr>
</tbody>
</table>

Table 5: Bandwidth requirements

Technique: ___________________________________________________________

_____________________________________________________________________

_____________________________________________________________________

_____________________________________________________________________

Reason: ____________________________________________________________

_____________________________________________________________________

_____________________________________________________________________
b) Scheduling algorithms

(i) For the Fair-Queuing scheduling algorithm, explain its main objective, detail its design, and name one of its disadvantages. Is it a work-conserving or non-work conserving algorithm? Tick the correct box. (3 Points)

Objective: __________________________

Design: __________________________

Disadvantage: ________________________

□ Work-conserving □ Non work-conserving

(ii) Figure 3 depicts the drop probability as a function of the average queue depth of a queue-management algorithm. What is the name of that algorithm? Explain what the algorithm is trying to achieve and how. What is the purpose of $x_1$ and $x_2$? (2 Points)

![Figure 3: Drop probability of a queue-management algorithm](image)

Algorithm name: __________________________

Objective and design: __________________________

Purpose of $x_1$ and $x_2$: __________________________
c) Fast convergence (10 Points)

Consider the network topology in Figure 4. Each link is annotated with its IGP weight. In the following, we are interested in identifying possible Loop-Free Alternates (LFAs).

![Network topology diagram](image)

**Figure 4: Network topology.**

(i) What are LFAs, and why are they important in fast convergence? (3 Points)

(ii) Compute the shortest path from each of the different routers towards R6, as well as their cost. (2 Points)

Path between R1→R6: __________________________
Cost of R1→R6: __________________________

Path between R2→R6: __________________________
Cost of R2→R6: __________________________

Path between R3→R6: __________________________
Cost of R3→R6: __________________________

Path between R4→R6: __________________________
Cost of R4→R6: __________________________

Path between R5→R6: __________________________
Cost of R5→R6: __________________________
(iii) Which router(s), if any, can be considered a per-link per-prefix LFA of R1, to reach R6? Justify your answer.


In this task, you are hired to deploy a new link-layer protocol in the network shown in Figure 5. The protocol is an alternative to Ethernet called Advnet. Its packets have a similar structure as Ethernet; the Advnet header consists of the following fields:

- **dstAddr** (11 bits): the destination address
- **srcAddr** (11 bits): the source address
- **advType** (16 bits): indicates the protocol used in the payload (e.g., IPv4)
- **flags** (10 bits): can be used for custom applications. Packets coming from hosts always have all flags set to 0.

Advnet is designed specifically to work in the topology shown in Figure 5. Note the following properties of the topology and Advnet:

- The topology has a tree structure and hosts are only connected to the leaf switches
- The network is isolated (there is no gateway to other networks or the Internet)
- Each switch has only 3 active ports, except for $S_0$ (2 ports) and the leaf switches (31 ports)
- Each leaf switch is connected to 30 hosts. A host $H_{i,j}$ is connected to switch $S_i$ on port $j$

Further note that Advnet addresses are constructed in a particular way: The format is $A:B:C:D$ where $A$, $B$, $C$ and $D$ describe the path to the addressed host starting from $S_0$:

- $A$ (2 bits) identifies the port on which the switch at level 0 ($S_0$) needs to send the packet
- $B$ (2 bits) identifies the port on which the switch at level 1 ($S_1$ or $S_2$) needs to send the packet
- $C$ (2 bits) identifies the port on which the switch at level 2 ($S_3$ – $S_6$) needs to send the packet
- $D$ (5 bits) identifies the port on which the leaf switch at level 3 ($S_7$ – $S_{14}$) needs to send the packet

For example, host $H_7,2$ has the Advnet address 2:2:2:2.

During this task, you will develop a P4 program that runs on all switches in this network (the same code runs on all switches, only table entries are different for each switch). To start, Figure 6 contains a code skeleton that you will extend and modify during this task.
#include <core.p4>
#include <v1model.p4>

const bit<16> TYPE_IPV4 = 0x0800;

header advnet_t {
    bit<11> dstAddr;
    bit<11> srcAddr;
    bit<16> advType;
    bit<10> flags;
}

header ipv4_t {
    bit<4> version; bit<4> ihl; bit<8> diffserv;
    bit<16> totallen; bit<16> identification; bit<3> flags;
    bit<13> fragOffset; bit<8> ttl; bit<8> protocol;
    bit<16> hdrChecksum; bit<32> srcAddr; bit<32> dstAddr;
}

struct metadata {
}

struct headers {
    advnet_t advnet;
    ipv4_t ipv4;
}

parser MyParser(packet_in packet, out headers hdr, inout metadata meta, inout standard_metadata_t standard_metadata) {
    state start {
        transition parse_advnet;
    }
    // ToDo: parse Advnet and IPv4
}

control MyVerifyChecksum(...) { apply { } }

control MyIngress(inout headers hdr, inout metadata meta, inout standard_metadata_t standard_metadata) {
    action send_to_port(bit<9> port) {
        standard_metadata.egress_spec = port;
    }
    action send_to_port_and_set_flags(bit<9> port, bit<10> flags) {
        standard_metadata.egress_spec = port;
        hdr.advnet.flags = flags;
    }

table l2_forward {
    key = {
        hdr.advnet.dstAddr: ternary;
        hdr.advnet.flags: ternary;
    }
    actions = {
        send_to_port;
        send_to_port_and_set_flags;
        NoAction;
    }
    default_action = NoAction();
    size = 1024;
}
register<bit<32>>(2048) packets_sent;
register<bit<32>>(2048) packets_received;

action update_statistics() {
    bit<16> packets_sent_before
    packets_sent.read(packets_sent_before, (bit<32>) hdr.advnet.srcAddr);
    packets_sent.write(packets_sent_before + 1, (bit<32>) hdr.advnet.
    srcAddr);
    bit<16> packets_received_before;
    packets_received.read(packets_received_before, (bit<32>) hdr.advnet.
    dstAddr);
    packets_received.write((bit<32>) hdr.advnet.dstAddr,
    packets_sent_before + 1);
}

table monitoring {
    key = {
        hdr.srcAddr: exact;
    }
    actions = {
        NoAction;
    }
    default_action = NoAction();
}

#include <legacy_1.p4>

apply {
    l2_forward.apply();
    monitoring.apply();
    #include <legacy_2.p4>
}

control MyEgress(...) { apply { } }

control MyComputeChecksum(...) { apply { } }

control MyDeparser(packet_out packet, in headers hdr) {
    apply {
        // todo
    }
}

V1Switch(MyParser(), MyVerifyChecksum(), MyIngress(), MyEgress(),
    MyComputeChecksum(), MyDeparser()) main;

Figure 6: P4 code skeleton
a) Getting familiar with the network (5 Points)

Before you start implementing Advnet, the customer wants to make sure that you understand how the protocol and the topology works. So she asks you some short questions.

(i) What is the Advnet address of host $H_{10,11}$ (1 Point)

(ii) How many links does a packet from 3:2:3:5 to 3:3:3:4 traverse if it follows the shortest path? (Include the links between hosts and switches in your count). (1 Point)

(iii) One advantage of Advnet compared to Ethernet is that its header is shorter (the Ethernet header is 112 bits long). How long does it take to send 1'500'000 bytes of IP payload over an Advnet and an Ethernet link with the following properties:

- 100 Gbps line rate ($12.5 \times 10^9$ bytes per second)
- no packet loss or congestion
- maximum 1500 bytes IP payload per packet

Your final result can be a fractional number (there is no need to compute the division). (3 Points)

with Ethernet

with Advnet
b) Parsing packets (5 Points)

Now it is time to extend the program such that it parses and deparses \texttt{Advnet} packets correctly.

(i) The parser states for \texttt{Advnet} and \texttt{IPv4} are missing in the skeleton. Write their code below. Note that the skeleton already contains the header definitions and, if a packet is not \texttt{IPv4}, it is enough to parse the \texttt{Advnet} header. (4 Points)

```plaintext
state parse_advnet { 
  
}

state parse_ipv4 { 
  
}
```

(ii) Write the code for the deparser. (1 Point)

```plaintext
apply { 
  
}
```
c) Basic forwarding

The code skeleton already defines and applies a table called `l2_forward`. In this task, you need to fill the table with entries.

Note that the table uses ternary matches. In table entries, you can write ternary matches using “*” for bits/digits that should be ignored (for example, you can write “1:*:3:4” or “01:**:11:00100” to match on Advnet addresses). Since multiple entries can match on the same packet, list the rules with decreasing priority (first entry has the highest priority).

(i) The customer tells you that she wants to forward packets along the shortest paths. Which entries do you need to add to the `l2_forward` table on switch $S_1$? Use the smallest number of entries possible. Abbreviate the action names as follows: “SP” for `send_to_port`, “SPF” for `send_to_port_and_set_flags` and “NA” for `NoAction`. (5 Points)

<table>
<thead>
<tr>
<th>dstAddr</th>
<th>flags</th>
<th>name</th>
<th>parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(ii) Could you reduce the number of entries if you could also change the P4 code? Explain why and how, if it is possible; or why not, if it is not possible. (1 Point)
(iii) The customer changes her mind and wants that all traffic transit via switch $S_0$ such that it can be inspected there (as opposed to sending it along the shortest path). How can you achieve this through the $\texttt{12.forward}$ table? Only explain your high-level idea here. You will be asked to provide the table entries later. (3 Points)

<table>
<thead>
<tr>
<th>dstAddr</th>
<th>flags</th>
<th>name</th>
<th>parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(iv) Which entries do you need to add to the $\texttt{12.forward}$ table on switch $S_0$ to implement your idea from above? Use the smallest number of entries possible. Abbreviate the action names as follows: “SP” for $\texttt{send.to.port}$, “SPF” for $\texttt{send.to.port.and.set.flags}$ and “NA” for $\texttt{NoAction}$. (4 Points)
(v) Which entries do you need to add to the 12_forward table on switch S₁ to implement your idea from above? Use the smallest number of entries possible. Abbreviate the action names as follows: “SP” for send_to_port, “SPF” for send_to_port_and_set_flags and “NA” for NoAction. (4 Points)

<table>
<thead>
<tr>
<th>dstAddr</th>
<th>flags</th>
<th>name</th>
<th>parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(vi) Currently, the table reserves space for 1024 entries. But as you have found out above, the number of entries required is much smaller. How many entries need to fit in the table at most in this network? Consider the case where all traffic is sent via S₀ and no other changes to the code are allowed. Explain your answer. (2 Points)
d) Computing statistics

The customer wants to compute statistics about the traffic in her network. In particular, she wants a way to specify a list of hosts for which the switches count the packets they send.

The code skeleton already contains some code for a table in which the monitored hosts can be specified and some code to compute the statistics using registers. The relevant lines are shown again below (Figure 7).

```plaintext
register <bit<32>>(2048) packets_sent;
register <bit<32>>(2048) packets_received;

action update_statistics() {
  bit<16> packets_sent_before
  packets_sent.read(packets_sent_before, (bit<32>) hdr.advnet.srcAddr);
  packets_sent.write(packets_sent_before + 1, (bit<32>) hdr.advnet.srcAddr);

  bit<16> packets_received_before;
  packets_received.read(packets_received_before, (bit<32>) hdr.advnet.dstAddr);
  packets_received.write((bit<32>) hdr.advnet.dstAddr, packets_sent_before + 1);
}

table monitoring {
  key = {
    hdr.srcAddr: exact;
  }
  actions = {
    NoAction;
  }
  default_action = NoAction();
}
```

Figure 7: Section of the code skeleton with many errors

(i) Unfortunately, the programmer who wrote this code made some mistakes (logical and syntactical ones), which you must find. Use the table below to list the lines that contain errors and write down the fixed version of them (rewrite the entire line). (6 Points)

*Hint: The first error is in line 69 and there are 6 lines which contain at least one error.*

<table>
<thead>
<tr>
<th>Line</th>
<th>Correct code</th>
</tr>
</thead>
<tbody>
<tr>
<td>69</td>
<td></td>
</tr>
</tbody>
</table>
(ii) The code uses registers to count the packets sent, but not all P4 devices support registers. How would you need to change the code for it to run on a device that supports counters but not registers. Explain the required changes when using indirect and direct counters. You do not need to write the code. (4 Points)

Changes when using indirect counters: 

Changes when using direct counters: ____
e) Legacy code

Your customer also wants to have some legacy code included. This code is in the files legacy_1.p4 and legacy_2.p4 (Figures 8 and 9). The code compiles without errors but the customer does not remember what it does exactly. So she asks you to explain it to her. Unfortunately, the programmer did not document the code properly.

Figure 8: legacy_1.p4

```p4
1 register<bit<48>>(2048) X;
2 register<bit<32>>(2048) Y;
3
4 table A {
5   key = {
6     hdr.advnet.srcAddr: exact;
7   }
8   actions = {
9     NoAction;
10   }
11   default_action = NoAction();
12 }
```

Figure 9: legacy_2.p4

```p4
1 if (A.apply().hit) {
2   bit<48> M;
3   bit<32> N;
4   bit<48> T = standard_metadata.ingress_global_timestamp; // microseconds
5   X.read(N, (bit<32>) hdr.advnet.srcAddr);
6
7   if (T > M+1000000) {
8     X.write((bit<32>) hdr.advnet.srcAddr, T);
9     Y.write((bit<32>) hdr.advnet.srcAddr, 1);
10    }
11 else {
12     Y.read(N, (bit<32>) hdr.advnet.srcAddr);
13     Y.write((bit<32>) hdr.advnet.srcAddr, N + 1);
14     if (N > 3) {
15       mark_to_drop(standard_metadata);
16     }
17   }
18 }
```

(i) The code defines two register arrays (X and Y). What kind of information is stored in these registers? (2 Points)

Register X: __________________________________________

________________________________________

Register Y: __________________________________________

________________________________________
The code defines a table (A) but the only available action in this table is **NoAction**, which does not do anything to a packet. What is the purpose of this table?  

---

How many entries can be added to table A? Explain your answer.  

---

What does the legacy code do?  

*Hint: investigate what happens if a host’s advnet address is in table A.*
Extra Sheet 1

In case you need more space, use the following pages. Make sure to always indicate the task to which the answer belongs (e.g., 3 d) (ii)).

Task: ________________

__________________________________________________________________________________
__________________________________________________________________________________
__________________________________________________________________________________
__________________________________________________________________________________
__________________________________________________________________________________
__________________________________________________________________________________

Task: ________________

__________________________________________________________________________________
__________________________________________________________________________________
__________________________________________________________________________________
__________________________________________________________________________________
__________________________________________________________________________________
__________________________________________________________________________________
Extra Sheet 2

Task: ______________

________________________________________________________________________
________________________________________________________________________
________________________________________________________________________
________________________________________________________________________
________________________________________________________________________
________________________________________________________________________

Task: ______________

________________________________________________________________________
________________________________________________________________________
________________________________________________________________________
________________________________________________________________________
________________________________________________________________________
________________________________________________________________________