

# Exam: Advanced Topics in Communication Networks

12 February 2024, 15:30-18:00, Room HIL G75

## Sample Solution

- ▷ Write **legibly** your ETH student number (legi number) below on this front page.
- ▷ **Do not write your name** or use a stamp with your name on it.
- ▷ **TRIPLE-check that your legi number is correct!**  
You will not be graded if you make a mistake when writing your number.
- ▷ Put your legitimization card (legi) 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 - 39).
- ▷ **Do not separate** the task sheets. We will collect the exams after you 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, and add a “**see Extra Sheet X**” note in the original task space.
- ▷ Read each task completely before you start solving it.
  
- ▷ 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 or calculator are allowed.
- ▷ It is not required to score all points to get the best mark.

Student legi nr.:

|  |
|--|
|  |
|--|

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

| Task                     | Points |
|--------------------------|--------|
| Advanced routing         | /33    |
| Programmable data planes | /21    |
| Network verification     | /32    |
| Network measurements     | /21    |
| Network security         | /11    |
| Transport                | /21    |
| Sustainable networking   | /11    |
| Total                    | /150   |



**Task 1: Advanced Routing****33 Points****a) Hierarchical routing****(9 Points)**

Consider the route reflection topology composed of five routers depicted in Fig. 1. RR1 and RR2 act as route reflectors, while RA, RB, and RC are route reflector clients. The dashed lines indicate iBGP sessions, with single-headed arrows indicating client sessions (they point towards the route reflector) and double-headed arrows indicating normal iBGP sessions (e.g., RA is the client of RR1, while RB is the client of both RR1 and RR2). The network relies upon an IGP whose weights are depicted next to each link. Three routers—RA, RB, and RR2—receive an *equivalent* eBGP route for the same prefix  $p$  ( $r1$ ,  $r2$ , and  $r3$  in Fig. 1) meaning that all of their BGP attributes are equal to each other.

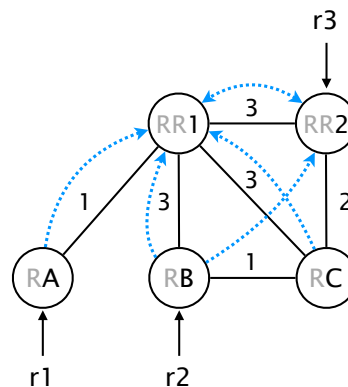


Figure 1: A simple route reflection topology.

- (i) Indicate which egress router **RC** chooses alongside with the path the traffic it sends to  $p$  takes. If something is sub-optimal and/or incorrect about this path, briefly explain it.

**(2 Points)**

**Solution:** RC selects RA as egress. Traffic flows alongside the path: RC, RR1, RA. It is sub-optimal: one can see that RB is a closer egress for RC, but RC doesn't learn the route coming from it. (It would in a full-mesh though.)

**Grading:** 0.5pt for the egress. 0.5pt for the path. 1pt for the justification.

- (ii) Consider now that the link between RC and RR1 fails. Indicate which egress router RC chooses after the failure alongside with the path the traffic it sends to  $p$  takes. If something is sub-optimal and/or incorrect about this path, briefly explain it.

**(2 Points)**

**Solution:** RC still selects RA as egress. Traffic flows alongside the path: RC, RB, then  $r2$ . There is a deflection, so the path is incorrect.

**Grading:** 0.5pt for the egress. 0.5pt for the path. 1pt for the justification.

- (iii) In addition of the failure of the link between RC and RR1, consider now what would happen if RB also loses its external route ( $r2$  stops being advertised). Indicate which egress router RC chooses alongside with the path the traffic it sends to  $p$  takes. If something is sub-optimal and/or incorrect about this path, briefly explain it.

**(2 Points)**

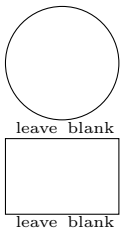
**Solution:** RC still selects RA as egress. Traffic flows alongside the path: RC, RB, RC, RB, ... There is a forwarding loop, so the path is clearly incorrect.

**Grading:** 0.5pt for the egress. 0.5pt for the path. 1pt for the justification.

- (iv) Briefly explain whether adding one extra client session between RC and RR2 would have been enough to prevent sub-optimal routing/forwarding and/or incorrect routing/forwarding in the situation (iii) depicted just above.

**(3 Points)**

**Solution:** Adding a session would prevent the loop. It would also prevent the sub-optimal routing case whenever  $r2$  is missing. If  $r2$  is there though, sub-optimal routing remains.



---

**Grading:** 1.5 pt for the loop. 1.5 pt for the deflection.

**b) Fast(er) convergence****(9 Points)**

Consider the weighted network topology composed of seven routers depicted in Fig. 2.

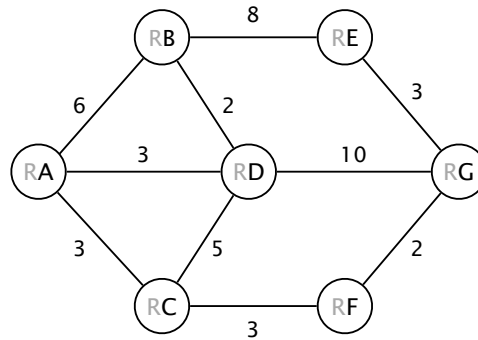
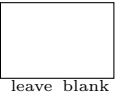


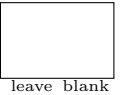
Figure 2: A simple internal topology.

- (i) Briefly explain: (i) what a Loop Free Alternate (LFA) is; and (ii) what a link-protecting LFA is. (2 Points)
- (ii) Is RA a link-protecting LFA for RD considering the failure of the link (RD, RC)? Briefly explain. (1 Point)
- RA is indeed a link-protecting LFA since it doesn't use (RD, RC) to reach any of the destinations RD reaches via (RD, RC): RC, RF, RG. 0.5 pt for acknowledging the fact. 0.5 pt for the explanation.
- (iii) Is RB a link-protecting LFA for RD considering the failure of the link (RD, RC)? Briefly explain. (1 Point)
- RB is not a link-protecting LFA since it does use (RD, RC) to reach some of the destinations RD reaches via (RD, RC): namely, RC, RF (it reaches RG via RE). 0.5 pt for acknowledging the fact. 0.5 pt for the explanation.
- (iv) Is RG a link-protecting LFA for RD considering the failure of the link (RD, RC)? Briefly explain. (1 Point)
- RG is a link-protecting LFA since it does not use (RD, RC) to reach any of the destinations RD reaches via (RD, RC). 0.5 pt for acknowledging the fact. 0.5 pt for the explanation.
- (v) Is RB a **per-prefix** LFA for RD for the prefixes originated by RG and when considering the failure of the link (RD, RC)? Briefly explain. (2 Points)
- RB is a per-prefix LFA for the prefixes originated by RG as RB doesn't use (RD, RC) to reach RG. 1 pt for acknowledging. 1 pt for the explanation.
- (vi) Is RE a **remote** LFA for RD considering the failure of the link (RD, RC)? Briefly explain. (2 Points)
- RE is a remote LFA for the failure of (RD, RC) because RB reaches RE directly, without crossing RD. And RE doesn't use (RD, RC) to reach any of the destination RD reaches via RC. 1 pt for acknowledging. 1 pt for the explanation.



## c) Local prefix filtering and aggregation

(11 Points)



- (i) Briefly explain the concepts of strong and weak forwarding consistency and how they differ from each other. (3 Points)

**Solution:** Both require destinations routable by the original FIB to keep the same next-hop after aggregation. Strong consistency also keeps non-routable destinations non-routable (dropped packets stay dropped); weak consistency may make them routable.

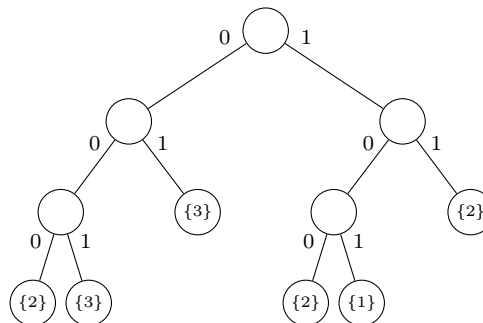
- (ii) Briefly explain two factors that will influence how “aggregatable” a forwarding table is. (2 Points)

**Solution:** (1) How prefixes are distributed over the next-hops: the fewer neighbors, the better (maximal if all prefixes share one next-hop). (2) How “aggregatable” the prefixes are, i.e. how many are adjacent or nested and can be merged.

In the rest of this question, we ask you to aggregate the forwarding table given below using the Optimal Routing Table Constructor (ORTC) algorithm.

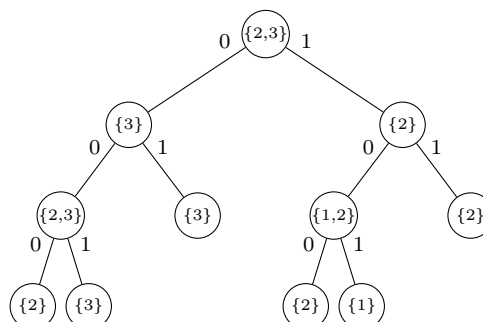
|                     |
|---------------------|
| $* \Rightarrow 1$   |
| $0* \Rightarrow 2$  |
| $1* \Rightarrow 2$  |
| $01* \Rightarrow 3$ |
| $001 \Rightarrow 3$ |
| $101 \Rightarrow 1$ |

- (iii) Fill in the empty binary tree below with the tree state obtained at the end of pass 1 (normalization) when running ORTC on the table above. (2 Points)



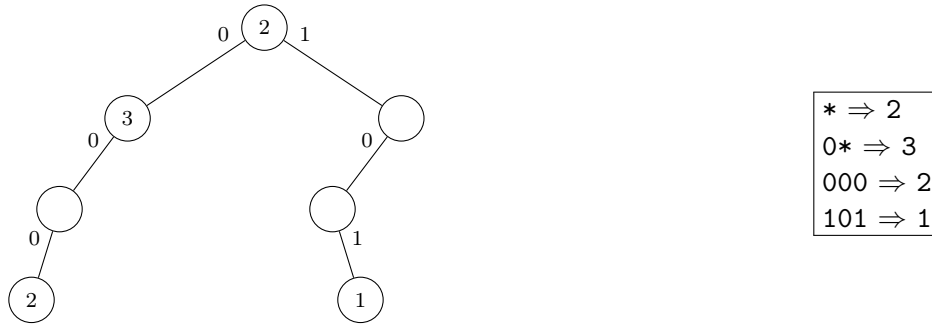
**Solution:** the tree is completed so that every node has either zero or two children (000, 100 and 11 are added). Next-hops are then pushed down to the leaves: each leaf without a next-hop inherits the one of its closest ancestor (000 inherits 2 from 0\*; 100 and 11 inherit 2 from 1\*), and internal nodes no longer carry a next-hop.

- (iv) Fill in the empty binary tree below with the tree state obtained at the end of pass 2 (next-hop ranking) when running ORTC on the table above. (2 Points)



**Solution:** the tree is traversed bottom-up. Each internal node gets the intersection  $A \cap B$  of the next-hop sets of its two children if it is non-empty, and their union  $A \cup B$  otherwise: 00 gets  $\{2\} \cup \{3\} = \{2, 3\}$ ; 0 gets  $\{2, 3\} \cap \{3\} = \{3\}$ ; 10 gets  $\{2\} \cup \{1\} = \{1, 2\}$ ; 1 gets  $\{1, 2\} \cap \{2\} = \{2\}$ ; the root gets  $\{3\} \cup \{2\} = \{2, 3\}$ .

- (v) Fill in the empty binary tree below with the tree state obtained at the end of pass 3 (prefix filtering) when running ORTC on the table above **alongside** with the final forwarding rules obtained (write the rules in the table below). (2 Points)

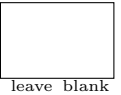


**Solution:** the tree is traversed top-down. The root picks any next-hop from its set (here 2). Every other node gets no next-hop if the one inherited from its closest ancestor is in its set; otherwise it picks any next-hop from its set, which becomes a forwarding rule: 0 picks 3 ( $2 \notin \{3\}$ ), 000 picks 2 ( $3 \notin \{2\}$ ) and 101 picks 1 ( $2 \notin \{1\}$ ). Nodes without a next-hop and without any descendant carrying one (001, 01, 100, 11) are removed. The table goes from 6 rules down to 4.

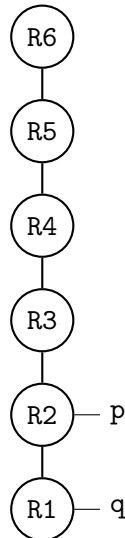
Picking 3 at the root is equally valid and also gives 4 rules:  $*$   $\Rightarrow$  3,  $1*$   $\Rightarrow$  2,  $000 \Rightarrow$  2,  $101 \Rightarrow$  1.

## d) Distributed prefix filtering and aggregation

(4 Points)

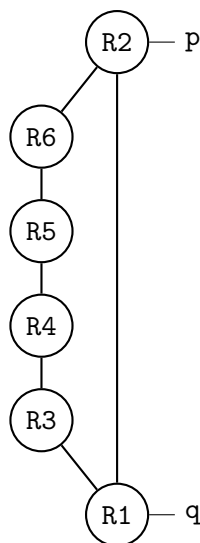


- (i) Draw an inter-domain network topology composed of 6 routers in which one router advertises a parent prefix  $p$ ; another router advertises a child prefix  $q$  (contained in  $p$ ); and **all** the other 4 routers can filter the prefix  $q$  following the DRAGON filtering rules. Draw provider-customer links using straight lines, with the providers above customers, and peer-to-peer links using dashed lines (as we have done in the exercises). (2 Points)

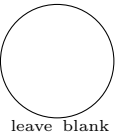
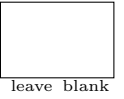


**Solution:** in the above interdomain topology,  $p$  and  $q$  are reached via a customer route by every other router.

- (ii) Same question as above except that this time your topology should be such that **none** of other 4 routers should be able to filter out the prefix  $q$  following the DRAGON filtering rules. Use the same drawing convention (see above). (2 Points)



**Solution:** in the above interdomain topology, routers R3, R4, R5 and R6 reach  $q$  via a customer route and  $p$  via a provider route (due to Gao-Rexford valley-freeness router R1 will not propagate  $p$  to R3). Since the attribute of the route to  $q$  is preferred over that of the route towards  $p$ , the DRAGON condition is not satisfied.

**Task 2: Programmable data planes****21 Points****a) General Questions****(4 Points)**

For each of the following statements, indicate whether they are *true* or *false*. There is always one correct answer. Grading is as such: 4 points for four correct answers, 2 points for three correct answers, and 0 points otherwise.

true    false  
☒    ☐

During packet recirculation in a P4 program, a packet can retain and transport computation state information by appending it as an additional header.

**Solution:** True. In P4, during recirculation, a packet can carry state information in the form of additional headers.

true    false  
☒    ☐

Detecting heavy-hitter flows in P4 requires the use of stateful data structures.

**Solution:** True. Identifying heavy-hitter flows in P4 requires stateful objects to track and analyze the flow characteristics over time.

true    false  
☐    ☒

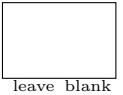
In a P4 switch, packet drops will not occur unless explicitly specified in the P4 code.

**Solution:** False. Packet drops can occur in P4 pipelines for various reasons, such as table misses or resource limitations, even if not explicitly specified in the P4 code.

true    false  
☐    ☒

The P4 programming language supports the use of a division operator.

**Solution:** False. P4 does not have a division operator as part of its language syntax. P4 focuses on packet processing constructs rather than general arithmetic operations.

**b) Probabilistic data structures****(17 Points)**

The objective of this task is to implement a Bloom filter designed to block network packets targeting specific ports. This functionality is essential in firewall configurations to prevent unauthorized packet entry into a network.

The Bloom filter you will work on comprises two distinct arrays, referred to as registers. Each register is associated with its unique hash function. For a given input  $x$ , each hash function  $h_i(x)$  corresponds to a specific register  $R_i$ .

The hash functions are defined as:

$$\begin{aligned} h_1(x) &= \text{sum}(x) \mod 10 \\ h_2(x) &= \text{mult}(x) \mod 10 \end{aligned}$$

In this context,  $\text{sum}(x)$  calculates the sum, and  $\text{mult}(x)$  the product, of  $x$ 's digits.

For instance, for  $x = 72$ ,  $h_1(x)$  yields  $\text{sum}(72) = 9$ , and  $h_2(x)$  gives  $\text{mult}(72) = 14$ . Post modulo 10, hash values are 9 and 4, setting bits at these positions in registers  $\mathbf{R}_1$  and  $\mathbf{R}_2$ .

**(i) Bloom filter configuration****(3 Points)**

Insert port numbers 731, 42, and 1410 sequentially into the Bloom filter as shown below. For each port, set the corresponding bits in the registers. Begin with the filter in its initial state (empty arrays) and proceed through each insertion step (first insertion being 731).

|   | $\mathbf{R}_1$ | $\mathbf{R}_2$ |   | $\mathbf{R}_1$ | $\mathbf{R}_2$ |   | $\mathbf{R}_1$ | $\mathbf{R}_2$ |   | $\mathbf{R}_1$ | $\mathbf{R}_2$ |
|---|----------------|----------------|---|----------------|----------------|---|----------------|----------------|---|----------------|----------------|
| 0 | 0              | 0              |   | 0              | 0              |   | 0              | 0              |   | 0              | 1              |
| 1 | 0              | 0              |   | 1              | 1              |   | 1              | 1              |   | 1              | 1              |
| 2 | 0              | 0              |   | 0              | 0              |   | 0              | 0              |   | 0              | 0              |
| 3 | 0              | 0              |   | 0              | 0              |   | 0              | 0              |   | 0              | 0              |
| 4 | 0              | 0              | → | 0              | 0              | → | 0              | 0              | → | 0              | 0              |
| 5 | 0              | 0              |   | 0              | 0              |   | 0              | 0              |   | 0              | 0              |
| 6 | 0              | 0              |   | 0              | 0              |   | 1              | 0              |   | 1              | 0              |
| 7 | 0              | 0              |   | 0              | 0              |   | 0              | 0              |   | 0              | 0              |
| 8 | 0              | 0              |   | 0              | 0              |   | 0              | 1              |   | 0              | 1              |
| 9 | 0              | 0              |   | 0              | 0              |   | 0              | 0              |   | 0              | 0              |

(1p each stage)

*Note: The table above demonstrates the configuration of the Bloom filter for blocking ports 731, 42, and 1410. Each '1' in the table represents a bit set by the hash functions for these ports, indicating that packets destined for these port numbers will be blocked.*

---

(ii) **Analysis of blocked web traffic** (2 Points)

Web traffic, typically using port 443, is unexpectedly blocked. Diagnose this issue by identifying the cause and provide a rationale for your conclusion.

**Solution:** These errors are termed *false positives*, where legitimate traffic is erroneously blocked (0.5pt). They occur due to hash collisions: port 443 shares hash values with blocklisted ports, causing the filter to mistakenly block it. Specifically, hash collisions occur between port 443 and the blocklisted ports 731 and 42 (1.5pt).

(iii) **Expansion of the Bloom Filter** (2 Points)

Consider adding an extra array to the Bloom filter. Discuss the criteria for choosing a hash function for this new array, especially in the context of the ongoing web traffic issue, assuming ports 731, 42, and 1410 remain in the filter.

**Solution:** A hash function that avoids collision with ports 731 and 42 should be chosen (1pt). This can be achieved by using a hash function that is different from the existing ones, such as  $h_3(x) = \text{sum}(x) \bmod 11$  (1pt).

(iv) **Evaluation of element removal from the Bloom filter** (2 Points)

Attempt to remove port 42 from the Bloom filter by resetting its associated bits to 0. Observe any side effects that occur and discuss how this reflects a fundamental limitation of Bloom filters.

**Solution:** The observed side effect is a false negative for port 1410. This occurs because removing port 42 also inadvertently clears bits associated with port 1410, leading to its unintended omission (1pt). This highlights a fundamental limitation of Bloom filters: their inability to selectively remove elements without affecting others. Removal of one element can inadvertently clear bits shared with other elements, leading to their unintentional omission (1pt).

**(v) Implementation of a Counting Bloom filter****(8 Points)**

This section focuses on implementing a Counting Bloom filter to address the limitations of a standard Bloom filter. In this design:

- Each cell in the standard Bloom filter's register is replaced by a counter.
- Inserting an element increments the counter for its associated bits, while removing an element decrements these counters.
- A query is positive if all relevant counters are greater than zero.
- This approach allows for the removal of elements, contrasting with the binary cells of the standard Bloom filter.

Listing 1: P4 code template for a standard Bloom filter.

```

1  #define N_CELLS 10
2
3  struct metadata {
4      bit<32> hash_one;
5      bit<32> hash_two;
6      bit<16> query_result;
7  }
8
9  metadata meta;
10 register<bit<16>>(N_CELLS) bloom_array_one;
11 register<bit<16>>(N_CELLS) bloom_array_two;
12
13 action compute_hashes() {
14     // Compute hashes, abbreviated for simplicity.
15     hash(meta.hash_one, ...);
16     hash(meta.hash_two, ...);
17 }
18
19 action insert() {
20     // Update Bloom filter arrays.
21     bloom_array_one.write(meta.hash_one, 1);
22     bloom_array_two.write(meta.hash_two, 1);
23 }
24
25 action query() {
26     // Read from Bloom filter arrays.
27     bit<16> result_one;
28     bit<16> result_two;
29     bloom_array_one.read(result_one, meta.hash_one);
30     bloom_array_two.read(result_two, meta.hash_two);
31
32     // Set query result based on Bloom filter arrays.
33     meta.query_result = result_one & result_two;
34 }
35
36 control bloom_filter_control {
37     compute_hashes();
38     insert();
39     query();
40
41     // Logic to handle query_result...
42 }
43

```

The P4 code template provided above demonstrates the standard Bloom filter implementation. Your task is to modify this code to develop the following actions for the Counting Bloom filter:

`increment()`: inserts `x` into the Counting Bloom filter

`decrement()`: deletes `x` from the Counting Bloom filter

`query()`: checks if `x` is present in the Counting Bloom filter

Based on the standard Bloom filter code template, write P4 actions for `increment()`, `decrement()`, and `query()` tailored to the Counting Bloom filter.

**Solution:**

```
action increment() { (3p)
    // Increment counts in Bloom filter arrays.
    bit<16> count_one;
    bit<16> count_two; (1p, initialization)
    bloom_array_one.read(count_one, meta.hash_one); (1p, read)
    bloom_array_two.read(count_two, meta.hash_two);

    if (meta.count_one < MAX_COUNT) {
        bloom_array_one.write(meta.hash_one, count_one + 1);
    }
    if (meta.count_two < MAX_COUNT) {
        bloom_array_two.write(meta.hash_two, count_two + 1); (1p, write)
    }
}

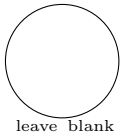
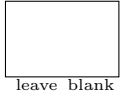
action decrement() { (3p)
    // Decrement counts in Bloom filter arrays.
    bit<16> count_one;
    bit<16> count_two; (1p, initialization)
    bloom_array_one.read(count_one, meta.hash_one); (1p, read)
    bloom_array_two.read(count_two, meta.hash_two);

    if (meta.count_one > 0) {
        bloom_array_one.write(meta.hash_one, count_one - 1);
    }
    if (meta.count_two > 0) {
        bloom_array_two.write(meta.hash_two, count_two - 1); (1p, write)
    }
}

action query() { (2p)
    // Read counts from Bloom filter arrays.
    bit<16> count_one;
    bit<16> count_two;
    bloom_array_one.read(count_one, meta.hash_one); (0.5p, read)
    bloom_array_two.read(count_two, meta.hash_two);

    // Determine if the item is present.
    meta.query_result = (meta.count_one > 0) && (meta.count_two > 0);
    (1.5p, condition checking)
}
```



**Task 3: Verification and Synthesis****32 Points****a) Warm-up****(4 Points)**

- (i) What guarantees can network verification provide to a network operator? (2 Points)

**Solution:** A verified configuration is guaranteed to satisfy the specification under all possible routing inputs.

Give 1 point for each of the following properties mentioned:

- Verifies a specification / will behave in a certain way / the specification works / behaves as specified
- Check in multiple environments / for different routing inputs / specific assumptions.
- Always finds a counter-example if the spec does not hold.

**Grading:**

- 0 points if they just say “correctness” or “reachability” or any concrete property.
- 0.5pt for answering “for a/one/the given environment”.
- 1pt for mentioning enough elements but incorrect connecting them.

- (ii) Describe two drawbacks of using simulation to verify network configurations. (2 Points)

**Solution:** Give 1 point for each of the following drawbacks mentioned:

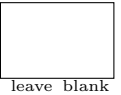
- Simulation may not correctly match the real hardware behavior / cannot simulate all aspects of a network.
- When the space of possible routing inputs / states / configurations is large or infinite, simulation does not scale.
- When the network changes / link fails, the simulation must be re-run / may not cover edge cases.
- Simulation may not cover edge cases / cannot give enough guarantees / may not be complete.
- Simulation lacks formal proof.

**Grading:**

- 0.5pt for only answering “scalability” or “accuracy” or “resource requirements” without explanation.
- 0.5pt for only answering “need to make assumptions” or “can be faulty” without explanation.
- 0pt for answering “cannot give suggestions”.

## b) Min-Hop Routing

(10 Points)



Min-Hop Routing is a protocol in which routers select the route with the least number of hops. Like BGP, Min-Hop Routing is a distance-vector protocol. All neighboring routers exchange their number of hops to the destination. Each router selects **any** one of its neighbors with the smallest number of hops. We encode the routing state of a router  $r$ :

- $r$ .Available: Does  $r$  know a route towards the destination?
- $r$ .SelectsFrom: From which neighbor  $n$  does  $r$  know its selected route?
- $r$ .Hops: How many hops does  $r$  take to reach the destination?

You want to verify a network running Min-Hop Routing. In the following, you are given three variations of route-selection equations. For each variation, determine whether it correctly encodes the route selection of Min-Hop Routing as described above. You should assume the route propagation has been correctly encoded.

*Hint:* We want to verify properties of an unknown implementation of Min-Hop Routing, i.e., the equation should model any correct implementation of Min-Hop Routing.

```
A = If(r.SelectsFrom == n,
      And(n.Available, r.Hops == n.Hops + 1),
      Implies(n.Available,
              Or(r.Hops < n.Hops + 1,
                 And(r.Hops == n.Hops + 1, r.SelectsFrom < n))))
```

```
B = If(r.SelectsFrom == n,
      And(n.Available, r.Hops == n.Hops + 1),
      Implies(n.Available, r.Hops <= n.Hops + 1))
```

```
C = If(r.SelectsFrom == n,
      And(n.Available, r.Hops == n.Hops + 1),
      Implies(n.Available, r.Hops < n.Hops + 1))
```

- (ii) Is A a correct encoding? Justify your answer. (2 Points)

**Solution:** A is incorrect. Give 2 points for the following answer:

- It assumes a particular tie-breaking rule: the router selects the neighbor with the lowest ID which may not be the real implementation.
- $r$ .SelectsFrom <  $n$  is restrictive

Other answers may be accepted with partial points if they are reasonable, e.g.,  $r$ .Available is never set.

**Grading:**

- 0.5pt for answering “ $r$ .SelectsFrom <  $n$  does not make sense” or “is not correct”
- 0pt if the student thinks A is correct because it has a clear tie-breaking rule.
- 0pt if the argument is wrong or not given.
- 1pt for answering  $r$ .SelectsFrom <  $n$  is not in the specification but has other incorrect arguments.

- (iii) Is B a correct encoding? Justify your answer. (2 Points)

**Solution:** B is correct. Give 2 points for the following answer:

- B properly handles networks where non-deterministic tie-breaking occurs: the router can select any of the best routes as both would make the equation true.
- B makes sure that  $r$  is at least as good as the route  $n$  selects.

**Grading:**

- 1pt for saying B is correct because “everything is implemented”.
- 0pt if the argument is wrong or not given.

(iv) Is C a correct encoding? Justify your answer. (2 Points)

**Solution:** C is not correct. Give 2 points for the following answer:

- C does not consider the possibility that a node can choose from multiple neighbors if they all have the same number of hops.
- Not selecting  $n$  does not imply  $r.Hops < n.Hops + 1$ , as it could be  $r.Hops == n.Hops + 1$ .
- The case where the hop count is optimal but another route is chosen is missing.

**Grading:**

- 1pt for “should be  $r.Hops \leq n.Hops + 1$ ” or “next hop counts can be equal”
- 0pt if the argument is wrong or not given.
- 1pt for mentioning two equal routes but wrong arguments.

(v) Rank the three equations from best to worst. Justify your answer. (2 Points)

**Solution:** If students answer the questions before correctly, then the accepted order is  $B > A > C$ . If students answer the previous questions incorrectly, then the accepted order depends on their previous answers. Full points are still possible even if all previous answers are wrong.

**Grading:**

- 2pt for a correct order based on the previous answers and a correct reason.
- 0pt if the argument is not given even if the order is correct.
- +1pt for the correct reason for one pair of equations, e.g., “A is less efficient than B or C, although they are both correct”.
- alternatively, +1pt for the correct reason to pick the best/worst equation, e.g., “B is the best because it is the only correct one”. If it is obvious to be the best, then the reason is not necessary.

(vi) Justify why two routers cannot select each other in any equation above. (2 Points)

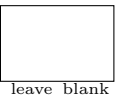
**Solution:** Assume  $r$  and  $n$  select each other. Then  $r.Hops = n.Hops + 1$  and  $n.Hops = r.Hops + 1$ , which is impossible.

**Grading:**

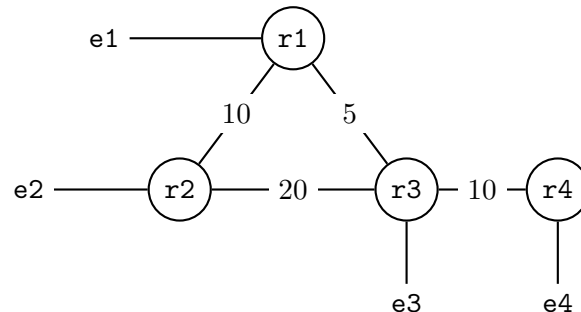
- 0pt if the argument is wrong, e.g., pointing to the wrong equation lines.
- 1pt for incomplete argument, e.g., “ $r.Hops$  cannot be equal to  $n.Hops + 1$ ”.

## c) Specifications

(10 Points)



You are given the BGP network shown above. It shows physical links and their IGP link weights:



The external networks **e1**, **e2**, **e3**, and **e4** *may* advertise a BGP route towards internal routers **r1**, **r2**, **r3**, and **r4**, respectively. All internal routers are connected in an iBGP full-mesh. The configured BGP Route Maps, however, are unknown.

You are given the following specification: (We use the same variable as in the lecture. A reference is provided on Page 19)

```

Spec = Or(And(e3.Available,
              r1.SelectsFrom == r3, r2.SelectsFrom == r3,
              r3.SelectsFrom == e3, r4.SelectsFrom == r3),
          And(Not(e3.Available),
              r4.SelectsFrom != e4,
              Implies(And(e1.Available, e2.Available,
                          e1.Route.AsPathLen == e2.Route.AsPathLen),
                      And(r1.SelectsFrom == e1, r2.SelectsFrom == e2,
                          r3.SelectsFrom == r1, r4.SelectsFrom == r1))))
  
```

- (i) According to **Spec**, how should the network handle routes from **e3**? (2 Points)

**Solution:** Routes from **e3** must be preferred over all others.

**Grading:**

- 2pt for the correct answer, 0pt for saying that **e3** must be selected if it is available *and* some other condition applies.
- 1pt if they say that **e3** must always be selected, omitting the condition that a route from **e3** should be available.
- 0pt for saying that **r3** “announces” a route to others, but nothing about **r1**, **r2**, and **r4** “selecting” a route.

- (ii) According to **Spec**, how should the network handle routes from **e4**? (2 Points)

**Solution:** Routes from **e4** must always be ignored, even if no other routes are present.

**Grading:** 2pt for saying that the route from **e4** is always ignored. Give full points if they say that **r4** will never select **e4**, without mentioning the remaining of the network.

- (iii) According to **Spec**, how should the network decide between routes from **e1** and **e2**? (2 Points)

**Solution:** If both **e1** and **e2** advertise a route with the same **AsPathLen** (while **e3** does not advertise a route), then all routers pick the egress with the lower **IgpCost** (i.e., **r2** picks **e2**, while **r1** picks **e1**, and both **r3** and **r4** pick **r1**).

**Grading:** 2pt for giving a correct answer. This can be either relying on the IGP cost. It is also fine to list the selected route of each router. A correct answer is also to first prefer egresses connected directly, or else prefer routes from `r1`.

Opt if they say anything about the routes in case their `AsPathLen` is not identical, or if they fail to mention anything about `AsPathLen`

(iv) The following table lists a set of routing states.

- For each external router, we list either `eX.AsPathLen`, or  $\infty$  if `eX.available` is `False`.
- For each internal router, we list `rX.SelectsFrom`.

For each of the routing states, determine whether it can be reached by a valid configuration, i.e., one that satisfies the specification. Briefly justify your answers. (4 Points)

| routing state |          |          |          |    |    |    |    | possible                            |                                     | reason                                                    |
|---------------|----------|----------|----------|----|----|----|----|-------------------------------------|-------------------------------------|-----------------------------------------------------------|
| e1            | e2       | e3       | e4       | r1 | r2 | r3 | r4 | Yes                                 | No                                  |                                                           |
| 1             | 1        | 1        | $\infty$ | e1 | e2 | e3 | r3 | <input type="checkbox"/>            | <input checked="" type="checkbox"/> | Would violate Property 1.                                 |
| 1             | 1        | $\infty$ | $\infty$ | r2 | r1 | r1 | r1 | <input type="checkbox"/>            | <input checked="" type="checkbox"/> | Contains a propagation loop.<br>Would violate Property 3. |
| 3             | 5        | $\infty$ | $\infty$ | e1 | r1 | r1 | r1 | <input checked="" type="checkbox"/> | <input type="checkbox"/>            | e1 has a lower <code>AsPathLen</code> .                   |
| 1             | $\infty$ | $\infty$ | $\infty$ | e1 | r1 | e3 | r3 | <input type="checkbox"/>            | <input checked="" type="checkbox"/> | r3 cannot “invent” a route from e3                        |

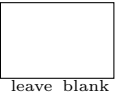
**Grading:** 1pt for each correct answer only if a correct reason is given! Not providing any (correct) reason or checking the wrong option (or not checking anything) will give 0 points. For question 3, be very generous with the provided reason.

**Variable reference** (same as in the lecture)

- An internal router (`r1`, `r2`, `r3`, and `r4`) has the following variables:
  - `rX.Available`: Whether the router selects a route.
  - `rX.SelectsFrom`: The neighbor from which `rX` selects its route.
  - `rX.Route`: The selected route attributes
- An external network (`e1`, `e2`, `e3`, and `e4`) has the following variables:
  - `eX.Available`: Whether the network advertises a route
  - `eX.Route`: The advertised route attributes
- A BGP route has the following attributes:
  - `Route.LocalPref`
  - `Route.AsPathLen`
  - `Route.Community`
  - `Route.IgpCost`

## d) Synthesis

(8 Points)

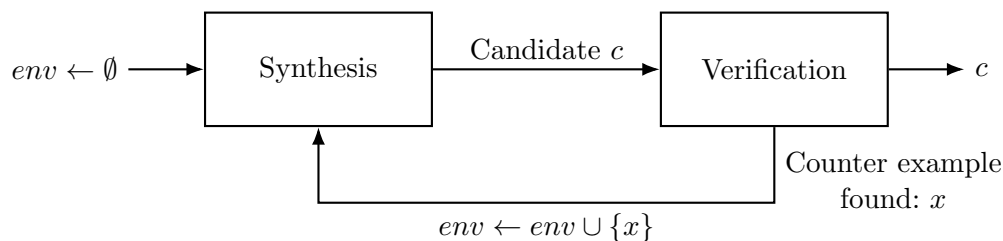


- (i) Why do we need to parametrize the configuration for network synthesis, and how are configuration parameters represented in SMT? (2 Points)

**Solution:**

- 1pt for mentioning that parameters are variables in SMT.
- 1pt for mentioning one of the following (or some variation):
  - The synthesizer needs degrees of freedom to synthesize a configuration.
  - The synthesizer must find an assignment for parameters (variables) such that the network satisfies the specification.
  - If there are no parameters, then the synthesizer cannot synthesize anything, and it will act as a verifier.

The following figure shows the inner loop of Counter Example Guided Inductive Synthesis (CEGIS), as presented in the lecture:



- (ii) The **Synthesis** block of CEGIS solves an SMT problem. The SMT solver may return either **sat** or **unsat**. How does CEGIS use that result? (3 Points)

**Solution:**

- **sat**: A configuration exists for which the specification is satisfied for all given counter-examples. The SMT model contains the configuration: our next candidate  $c$  to verify.
- **unsat**: No configuration exists in the given configuration space for which the specification is satisfied for all given counter-examples.

**Grading:**

- 2pt for correctly assigning the cases between **sat** and **unsat**.
- 1pt for vaguely describing what to do with the **model** (even if they don't name the model).
- -1pt if they don't say what to do now with the config.

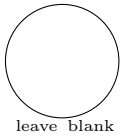
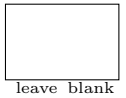
- (iii) The **Verification** block of CEGIS solves an SMT problem. The SMT solver may return either **sat** or **unsat**. How does CEGIS use that result? (3 Points)

**Solution:**

- **sat**: The verifier found a counter-example (the SMT model) for which the candidate configuration violates the specification. We add this counter-example to the list of counter-examples and go back to the Synthesis phase.
- **unsat**: There does not exist a counter-example for which the candidate configuration violates the specification. The candidate configuration is a valid solution to the Synthesis problem.

**Grading:**

- 2pt for correctly assigning the cases between `sat` and `unsat`.
- 1pt for vaguely describing what to do with the `model` (even if they don't name the model).
- -1pt if they don't mention that the CEX is added to env.

**Task 4: Network measurements****21 Points****a) Interdomain routing inference****(10 Points)**

We ask you to infer some of the business relationships in the inter-domain topology composed of 8 ASes depicted in Figure 3. To do so, you will use the **complete** BGP table observations from AS 7 and AS 8 depicted in Figure 4.

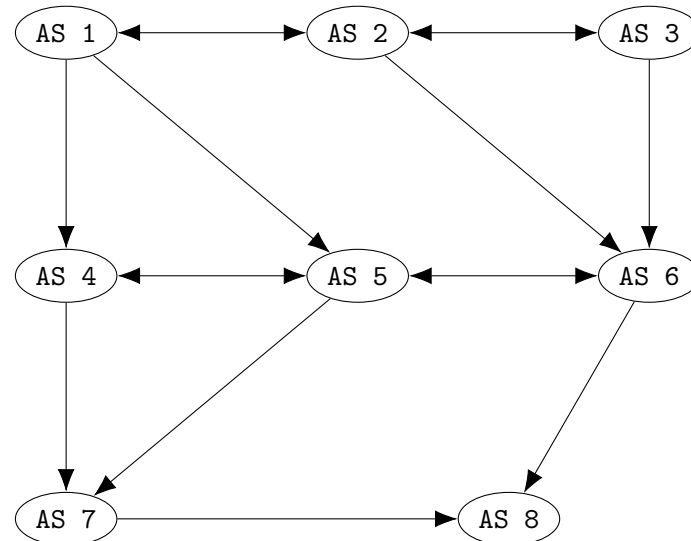


Figure 3: AS topology (solution)

| AS 7 Routing Table |                          | AS 8 Routing Table |                          |
|--------------------|--------------------------|--------------------|--------------------------|
| Origin AS          | Path                     | Origin AS          | Path                     |
| AS 1               | [AS 4, AS 1]             | AS 1               | [AS 7, AS 4, AS 1]       |
|                    | [AS 5, AS 1]             |                    | [AS 6, AS 2, AS 1]       |
| AS 2               | [AS 4, AS 1, AS 2]       | AS 2               | [AS 7, AS 4, AS 1, AS 2] |
|                    | [AS 5, AS 1, AS 2]       |                    | [AS 6, AS 2]             |
| AS 3               | —                        | AS 3               | [AS 6, AS 3]             |
| AS 4               | [AS 4]                   | AS 4               | [AS 7, AS 4]             |
|                    | [AS 5, AS 4]             |                    | [AS 6, AS 2, AS 1, AS 4] |
| AS 5               | [AS 5]                   | AS 5               | [AS 6, AS 5]             |
|                    | [AS 4, AS 5]             |                    | [AS 7, AS 5]             |
| AS 6               | [AS 5, AS 6]             | AS 6               | [AS 6]                   |
|                    | [AS 4, AS 1, AS 2, AS 6] |                    | [AS 7, AS 5, AS 6]       |
| AS 8               | [AS 8]                   | AS 7               | [AS 7]                   |
|                    | [AS 5, AS 6, AS 8]       |                    | [AS 6, AS 5, AS 7]       |

Figure 4: Complete routing tables observed at AS 7 and AS 8

You can make the following assumptions:

1. **Import policies:** ASes follow the classical BGP import policies, except for the tie-breaking criteria, i.e.:
  - (a) Prefer routes received from customers over those from peers, and routes from peers over those from providers.
  - (b) Amongst equally-preferred routes, prefer the routes with shorter AS-path lengths.
  - (c) Amongst equally-short routes, prefer the route received from the AS with the smaller AS number (tie-break).
2. **Export policies:** ASes follow the classical BGP export policies, i.e.:
  - (a) Routes received from customers are advertised to all: customers, peers, and providers.
  - (b) Routes received from peers are advertised only to customers.
  - (c) Routes received from providers are advertised only to customers.
3. **There is *no* sibling relationships.** The network topology does not include any sibling relationships.

- (i) Consider the path [AS 8, AS 7, AS 4, AS 1, AS 2]. Given that AS 1 and AS 2 have a peering relationship, infer the relationship between AS 7 and AS 8. Provide an explanation and justify why other relationship types are not possible. (1 Point)

**Solution:** According to the Valley-free property, the sequences AS 1 - AS 4, AS 4 - AS 7, and AS 7 - AS 8 must form a directed provider-customer chain. AS 7 cannot be a customer of AS 8, as this would imply the observation of the path AS 8, AS 6 by AS 7. Consequently, the only feasible arrangement is AS 8 being a customer of AS 7, AS 7 being a customer of AS 4, and AS 4 being a customer of AS 1.

**Grading:** 1 pt for identifying the relationship and for the explanation.

- (ii) Given that AS 1 is the provider of AS 5, determine the relationship between AS 5 and AS 7. Briefly explain and justify why other relationships are not possible. (1 Point)

**Solution:** The observed path AS 5, AS 1 by AS 7 indicates that AS 5 must be the provider of AS 7. This is because neither a peering relationship nor a scenario where AS 5 is a customer of AS 7 would result in the observed route exports.

**Grading:** 1 pt for identifying the relationship and for the explanation.

- (iii) Given that AS 3 is the provider of AS 6, infer the relationship between AS 5 and AS 6. Briefly explain and justify why other relationships are not possible. (2 Points)

**Solution:** The absence of the path AS 5, AS 6, AS 3 in AS 7's observations suggests that AS 5 is either a peer or provider of AS 6. However, if AS 5 were the provider of AS 6, AS 8 would have observed the path AS 6, AS 5, AS 4. Therefore, the relationship between AS 5 and AS 6 must be peering.

**Grading:** 1 pt for identifying the relationship type, 1 pt for the logical deduction.

- (iv) Gao's relationship inference algorithm starts by calculating the degree of each AS. Explain how the selection of vantage points can introduce a bias in this step. Discuss how such bias might degrade the algorithm's accuracy. (2 Points)

**Solution:** Gao's inference algorithm uses node degrees as a proxy for the power of a node, thus routes passing through a node affect its perceived strength. This introduces a bias based on where observation points are placed, as they influence the strength attributed to observed nodes. This may result in incorrect inferences about the relationships between ASes.

**Grading:** 1 pt for identifying the biased part, 1 pt for the explanation.

- (v) Cloud and content providers establish *many* peer-peer relationships without offering any transit services. How can this affect Gao's inference algorithm? What are the likely outcomes of the algorithm in this scenario? (2 Points)

**Solution:** In the context of Cloud and Content Delivery Networks, which peer with numerous eyeball networks globally, their node degrees would be exceptionally high. Gao's algorithm might incorrectly categorize these networks as transit providers, failing to represent the actual network dynamics.

**Grading:** 2 pts for the explanation.

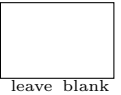
- (vi) Suppose a vantage point artificially increasing the degree of an AS by creating fake AS paths crossing it. Describe an approach to identify such inflated ASes in the dataset. (2 Points)

**Solution:** To systematically identify an inflated Autonomous System (AS) in Gao's inference algorithm when a vantage point maliciously distorts BGP paths:

1. **Exclude the Suspected Vantage Point:** Run the algorithm without the data from the suspected vantage point to establish a baseline of AS relationships and degrees.
2. **Include the Malicious Data:** Re-run the algorithm with the suspected malicious data and note any significant changes in AS neighbor degrees.
3. **Compare and Analyze:** Identify discrepancies by comparing the results of both runs. The AS with a notably higher degree in the second run is likely the inflated one.

This method isolates the malicious vantage point's impact and pinpoints the artificially inflated AS, assuming multiple vantage points are available and only one is malicious.

**Grading:** 2 pts for the explanation.

**b) Network tomography****(11 Points)**

This question is about inferring the performance of one or more links of the network  $N$  depicted below (Fig. 5). All links ( $l_1, \dots, l_7$ ) are indicated by solid lines, and all paths ( $p_1, \dots, p_5$ ) are indicated by dashed lines.

Recall that, during a time interval, a link, set of links, path, or pathset is “non-lossy” if it introduces negligible packet loss during that interval; otherwise, it is “lossy.”

Recall that, for a neutral network, performance refers to the logarithm of the probability of (a link, set of links, path, or pathset) being non-lossy. Whenever you indicate performance, **use the following notation:**

- $X_i$  to indicate the performance of link  $l_i$ ;
- $X_{ij}$  to indicate the performance of the set of links  $\{l_i, l_j\}$ ;
- $Y_i$  to indicate the performance of path  $p_i$ ;
- $Y_{ij}$  to indicate the performance of path pair  $p_{ij}$ .

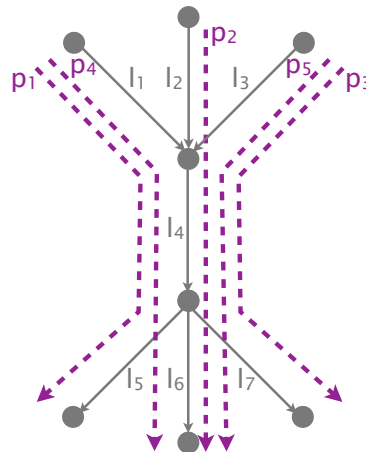


Figure 5: Network  $N$ .

**(i) Neutral network**

Consider that all the links of network  $N$  are neutral. Also, independence, stationarity, separability, and stability hold.

- ▷ Assume that we monitor all pathsets of  $N$  and we build a system of equations by writing an equation for each pathset/path. State a sufficient condition that allows uniquely inferring the performance of each link using the aforementioned system of equations. Briefly explain why the condition holds for  $N$ . (2 Points)

**Solution:**

We must verify that every two links are traversed by different sets of paths. Indeed, this condition holds for  $N$ :

- $l_1$ :  $\{p_1, p_4\}$ ;
- $l_2$ :  $\{p_2\}$ ;
- $l_3$ :  $\{p_3, p_5\}$ ;
- $l_4$ :  $\{p_1, p_2, p_3, p_4, p_5\}$ ;
- $l_5$ :  $\{p_1\}$ ;
- $l_6$ :  $\{p_2, p_4, p_5\}$ ;
- $l_7$ :  $\{p_3\}$ .

**Grading:** +1p for each of correct i) condition and ii) explanation of why the condition holds

- ▷ For each path/path pair  $p_1, p_2, p_{12}$ , write the equation that connects the path/path pair performance to the performance of its links. (2 Points)

**Solution:**

$$\begin{aligned} Y_1 &= X_1 && + X_4 &+ X_5 \\ Y_2 &= &X_2 &+ X_4 &+ X_6 \\ Y_{12} &= X_1 + X_2 &+ X_4 &+ X_5 &+ X_6 \end{aligned}$$

**Grading:** +0.5p for correct equation for each of  $p_1$  and  $p_2$   
+1p for correct equation for  $p_{12}$

## (ii) Non-neutral network

Consider now that all of  $N$ 's links are neutral *except for*  $l_4$ , meaning that the network  $N$  is non-neutral. For the rest of this task, **ignore paths  $p_4$  and  $p_5$** .

- ▷ To identify  $l_4$  as non-neutral, tomography builds a system of equations, say  $S$ . For each of the path/path pair  $p_1, p_2, p_3, p_{12}, p_{13}, p_{23}$ , write the equation of  $S$  that corresponds to it. (3 Points)

**Solution:**

$$\begin{aligned} Y_1 &= X_4 + X_{15} \\ Y_2 &= X_4 && + X_{26} \\ Y_3 &= X_4 && + X_{37} \\ Y_{12} &= X_4 + X_{15} + X_{26} \\ Y_{13} &= X_4 + X_{15} && + X_{37} \\ Y_{23} &= X_4 && + X_{26} + X_{37} \end{aligned}$$

**Grading:** max 3p if  $l_4$  is mapped to a single variable and for each path, all other links are mapped to a new variable; max 1.5p otherwise (i.e., each link is mapped to a different variable)

-0.5p for each missing/incorrect equation (an equation is incorrect if it includes wrong links or misses links)

- ▷ Assume that  $l_4$  treats differently  $p_1$  from  $p_2/p_3$  traffic:  $l_4$  is always non-lossy for  $p_1$  traffic, but lossy for  $p_2/p_3$  traffic with probability 0.5. All the other links are always non-lossy. We monitor  $p_1, p_2, p_3, p_{12}, p_{13}, p_{23}$  and observe that  $p_1$  is always non-lossy, while each of the rest is lossy with probability 0.5. Show that tomography identifies  $l_4$  as non-neutral. (2 Points)

**Solution:**

It is sufficient to show that the above system of equations is unsolvable.

Substituting the given observations, we obtain:

$$\begin{aligned} 0 &= X_4 + X_{15} \\ \log(0.5) &= X_4 && + X_{26} \\ \log(0.5) &= X_4 && + X_{37} \\ \log(0.5) &= X_4 + X_{15} + X_{26} \\ \log(0.5) &= X_4 + X_{15} && + X_{37} \\ \log(0.5) &= X_4 && + X_{26} + X_{37} \end{aligned}$$

This system is unsolvable: From the first equation,  $X_4 = X_{15} = 0$ . Substituting those into the second and third equations, we get  $X_{26} = X_{37} = \log(0.5)$ . Substituting those values in the last equation, we run into an inconsistency:

$$\log(0.5) = 0 + \log(0.5) + \log(0.5)$$

**Grading:** +1p for saying that the system of equations must be unsolvable  
+0.5p (resp. +1p) for partial (resp. correct) explanation of why the system is unsolvable

- ▷ Assume now that  $l_4$  is lossy only for  $p_3$  traffic, and all other links are always non-lossy. Can tomography conclusively identify  $l_4$  as non-neutral? Briefly explain why. (2 Points)

**Solution:**

No, as now there is no correlation between the performance of  $p_2$  and  $p_3$ , which would indicate that  $l_4$  is lossy for those paths. So,  $l_4$  being always non-lossy for all traffic is plausible. The loss that  $p_3$  traffic experiences can be attributed to links  $l_3$  and/or  $l_7$ .

**Grading:** 1.5p for partial explanation (i.e., saying that the loss that  $p_3$  traffic experiences can be attributed to links  $l_3$  and/or  $l_7$ ); 2p for correct explanation

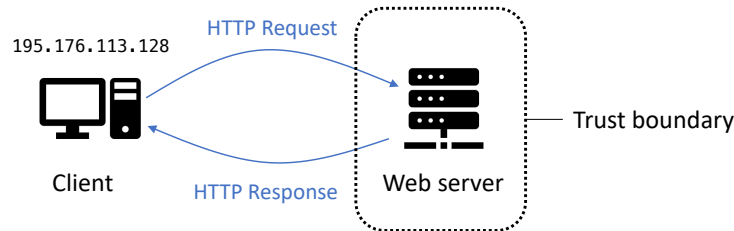
**Task 5: Network Security****11 Points****a) Applying STRIDE****(3 Points)**

Figure 6: A simple networked system.

The network in Figure 6 consists of a client (with IP 195.176.113.128) and a web server, connected over HTTP. Assume that you are the operator of the server (i.e., the server is within your trust boundary and the client is untrusted). Apply the STRIDE<sup>1</sup> methodology to identify **three** different threats to the server through its communication with the client. Each threat must belong to a different STRIDE category (e.g., only one threat can rely on spoofing). For each identified threat, specify its category and describe a possible mitigation.

*Example (exclude it from your answers):*

- *Threat:* The client may send an offensive HTTP request to the server and deny sending it during the post-attack analysis.
- *Category:* Repudiation.
- *Mitigation:* The server operator may require the client to sign the HTTP requests and send the signatures along with the requests.

**Solution:** Possible threats include:

- Spoofing.  
*Description:* An attacker can hijack the web server's IP address and impersonate the web server to send a fake website to clients.  
*Alternatives:* Spoofing the connection by replying faster to the server. *Mitigation:* Use certificates.
- Tampering.  
*Description:* A MitM can modify the message content and change, e.g., a Bitcoin address sent from the client to the server into the attacker's one.  
*Mitigation:* Sign the traffic (or simply use HTTPS).
- Repudiation.  
*Description:* The client may deny having obtained data from the server.  
*Mitigation:* Public signed logs.
- Information disclosure.  
*Description:* A MitM can read the content of all messages as they are not encrypted. Hence they can learn, e.g., submitted passwords.  
*Mitigation:* Encrypt the traffic (or simply use HTTPS).  
*Alternatives:* A client can access a webpage that they should not be able to. Use log-in.
- Denial of service.  
*Description:* An attacker can overwhelm the web server by sending excessive HTTP requests, such that it cannot respond to the client's requests.

<sup>1</sup>Spoofing, Tampering, Repudiation, Information disclosure, Denial of service, Elevation of privilege

Alternatives: TCP Flooding, Shrew attack, overloading the network by sending lot of traffic

*Mitigation:* To mitigate this threat, the webserver can block requests based on the attacker's IP address.

- Elevation of privilege.

*Description:* A MitM can read the exchanged cookie to then authenticate as the victim, .e.g. into an online shop.

*Mitigation:* Encrypt and sign the traffic (or simply use HTTPS).

Alternatives: Login as someone else; use MACs.

### Grading:

+0.5 pt for a correct description and matching category.

+0.5 pt for a successful mitigation against the proposed threat.

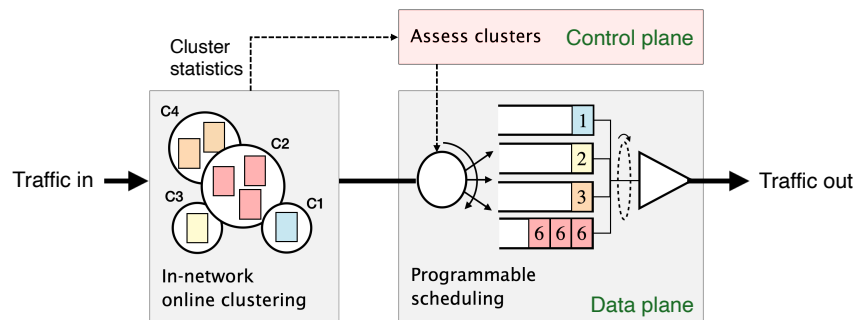
0.5 pt if the threat is only vague, but matches defense and category.

-1 pt when same category is written several times

no point for threat if it's too general ("the atatchker can launch a DoS-attack on the server").

### b) Revisiting *ACC-Turbo*

(4 Points)



leave blank

Figure 7: Architecture of *ACC-Turbo*.

*ACC-Turbo* defends against pulse-wave DDoS attacks by clustering incoming packets based on their similarity and deprioritizing malicious clusters (cf. Figure 7).

- (i) Name two reasons why existing DDoS defenses, which rely on a central controller to mitigate attacks, could be vulnerable to pulse-wave DDoS attacks. (1 Point)

### Solution:

- The pulses are short and the controller may not react in time.
- The attack vectors are unknown and unsupervision leads to misclassification.

### Grading:

+0.5 pt for mentioning short reaction time

+0.5 pt for mentioning unknown/different attack vectors *or* misclassification.

- (ii) What metric does *ACC-Turbo* use to assess packet similarity? (1 Point)

**Solution:** *ACC-Turbo* clusters packets based on the (Manhattan) distance of features in the header space (e.g., protocol, source and destination IP respectively port)

### Grading:

+0.5 pt for mentioning features of the header space, a concrete feature of the header space or the 5-tuple.

+0.5 pt for mentioning distance (students do not need to mention *manhattan* distance for this point) or "how close features are".

0 pt for mentioning "clustering". -0.5 pt for mentioning deep packet inspection.

- (iii) Explain one advantage of deprioritizing potentially-malicious traffic as opposed to just dropping it. (1 Point)

**Solution:** As ACC-Turbo is unsupervised, there are potentially false-positives (benign packets that are flagged as malicious). Hence we de-prioritize traffic to minimize the impact on these false-positives.

**Grading:**

+0.5 pt for mentioning false positives

+0.5 pt for mentioning something in the direction of minimizing the impact on them.

0 pt for “tricking the attacker” or not hurting “weirdly behaving clients”

0 pt for stating that delay leads to no false positives.

- (iv) Explain how an attacker could trick *ACC-Turbo* into allocating a high bandwidth share to malicious traffic. (1 Point)

**Solution:** The attacker must send very packets that have a very large distance in the header space and are thus not clustered together. Consequently, the malicious traffic does not get de-prioritized.

**Grading:**

+0.5 pt for mentioning large distance in the header space

+0.5 pt for mentioning that malicious packets must not be clustered together and thus are not de-prioritized.

0 pt for randomizing payload.

c) Defending DDoS with iptables

(4 Points)

|   |   |   |             |         |    |         |     |      |      |    |      |         |
|---|---|---|-------------|---------|----|---------|-----|------|------|----|------|---------|
| 1 | B | 1 | 2.889091491 | 1.0.0.2 | -> | 1.0.0.3 | TCP | 1429 | 6491 | -> | 6491 | Len=243 |
| 2 | B | 2 | 2.894215096 | 1.0.0.2 | -> | 1.0.0.3 | TCP | 1429 | 6491 | -> | 587  | Len=652 |
| 3 | M | 3 | 2.899809065 | 1.0.0.2 | -> | 1.0.0.3 | UDP | 1429 | 22   | -> | 2890 | Len=932 |
| 4 | M | 4 | 2.905407385 | 1.0.0.4 | -> | 1.0.0.3 | TCP | 1429 | 2191 | -> | 6490 | Len=193 |
| 5 | B | 5 | 2.910978684 | 1.0.0.3 | -> | 1.0.0.2 | TCP | 1429 | 6490 | -> | 6491 | Len=277 |
| 6 | M | 6 | 2.916530913 | 1.0.0.6 | -> | 1.0.0.3 | UDP | 1429 | 6490 | -> | 6491 | Len=487 |

leave blank

Listing 2: A Wireshark trace

You are managing a web server with IP address 1.0.0.3. Listing 2 describes the packets that you observe either incoming or outgoing the server. Note that packets are already classified as benign (B) or malicious (M).

For the following **three iptables** rule sets:

- Identify the packets that will get dropped if you deploy the rules.
- Describe one way the attacker could bypass the rules, and prevent their malicious packets from being dropped.

- (i) `iptables -A INPUT --dport !6491 -j DROP` (1 Point)

**Solution:** Dropped packets: 2, 3, 4 *or* all packets with destination port **not** being 6491  
Bypassing strategy: set the destination port to 6491.

**Grading:**

+0.5 pt for the correct list or description of dropped packets

+0.5 pt for the correct bypassing strategy

0 pt for “change destination port” -0.5 pt for not noting the packet numbers, but only giving a description.

- (ii) `iptables -A INPUT -p UDP -j DROP` (1 Point)

**Solution:** Dropped packets: 3, 6 *or* all packets sent over UDP  
Bypassing strategy: Do not send UDP packets, but e.g., TCP packets.

**Grading:**

+0.5 pt for the correct list or description of dropped packets

+0.5 pt for a correct bypassing strategy (not sending udp packets)

(iii) `iptables -A INPUT --src 1.0.0.2 -j DROP`

`iptables -A INPUT --src 1.0.0.3 -j DROP`

`iptables -A INPUT --src 1.0.0.6 -j DROP` (1 Point)

**Solution:** Dropped packets: 1, 2, 3, 6 (packet 5 does *not* get dropped as it is sent from the server to the client, hence the INPUT rule is not applied.)

Bypassing strategy: Spoof the IP address to a value different than 1.0.0.2, 1.0.0.3, and 1.0.0.6.

**Grading:**

+0.5 pt for the correct list of dropped packets

0 pt for “all packets sent from the listed IP addresses” as packet 5 does not get dropped, but is sent from a listed IP address. +0.5 pt for a correct bypassing strategy

(iv) Now, write a set of up to four `iptables` rules that drop *all and only* malicious packets. Explain your solution briefly. (1 Point)

**Solution:** Possible solutions include:

```
iptables -A INPUT -p UDP -j DROP
iptables -A INPUT --src 1.0.0.4 -j DROP
```

```
iptables -A INPUT --sport !6491 -j DROP
```

(packet 5 does *not* get dropped as it is sent from the server to the client, hence the INPUT rule is not applied.)

```
iptables -A INPUT --len 932 -j DROP
iptables -A INPUT --len 193 -j DROP
iptables -A INPUT --len 487 -j DROP
```

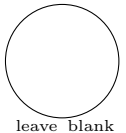
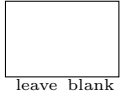
(even though this may not work in practice due to different lengths observed by iptables and wireshark.)

**Grading:** +1 pt for a correct set of `iptables` rules.

-0.5 pt if the explanation is missing or not correctly describing the written rules.

-0.5 pt if benign packets are dropped.

Note that it is not required to come up with a minimal set of rules. We also allow redundant rules.

**Task 6: Transport Protocols****21 Points****a) Modeling and Design****(4 Points)**

- (i) How do TCP RENO and CUBIC model the network, how do they update this model based on observations, and how do they control their sending rates? (2 Points)

**Solution:** They mainly model the network with three states: non-congested, congested, and severely congested. It observed this state via packet loss. Duplicate acknowledgments indicate isolated packet losses, which are interpreted as congestion. Timeouts indicate persistent packet loss, which is interpreted as severe congestion.

They control their sending rate with a congestion window. It first uses an exponential slow start to grow the congestion window until the first congestion is detected; afterward, it grows the congestion window following a cubic function. When congestion is detected, the window is reduced by a multiplicative factor. When severe congestion is detected, the window is reset to its minimum and the slow start is restarted.

**Grading:**

+0.5 pt for Model (congestion states).

+0.5 pt for Observations (both loss as well as duplicate acks and timeout are accepted).

+0.5 pt for Control with congestion window.

+0.5 pt if correct details are mentioned, e.g. slow start, or how exactly the congestion window is updated.

Point deductions for wrong statements.

- (ii) In the modern Internet, many flows are small and short: they transmit only a small amount of bytes and are only active for a short time. Explain the impact this has on congestion control with TCP RENO or CUBIC. (1 Point)

**Solution:**

The main impact is on fairness. Both RENO and CUBIC achieve fairness over several RTTs using an AIMD algorithm. If flows are too short for AIMD to converge, fairness is not achieved.

Alternatively, we also accept answers that explain the drawbacks of TCP slow start. A short flow may ultimately be transmitted in a single RTT, but still may need multiple RTTs as TCP needs to grow its congestion window first.

**Grading:**

+1 pt for correct and explained answer; in particular is should mention in some way that multiple RTTs are needed or that the bandwidth is not fully utilized. (0.5 for a partial answer).

- (iii) Explain why network address translation (NAT) is a challenge for Multipath TCP, and how this challenge is solved. (1 Point)

**Solution:**

Because of NAT, the receiver cannot easily identify whether two (sub) flow come from the same sender; each path may have different ip addresses and ports, which are not known to the sender either, as NAT happens in the network.

Multipath TCP negotiates a connection id token in the first flow and includes this token in all subsequent flows so that the receiver can identify the sender.

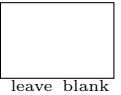
**Grading:**

+0.5pt for the problem, must include in some way that neither the sender nor receiver know how the addresses are translated.

+0.5pt for the solution.

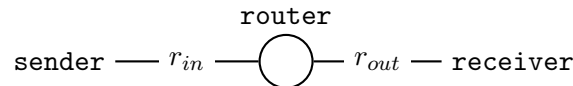
## b) Buffering

(11 Points)



In this task, you will analyze the impact of congestion control on network queues.

Consider the network shown below with a single sender and receiver connected through a router with link rates  $r_{in} = 100 \frac{\text{packets}}{\text{ms}}$  and  $r_{out} = 24 \frac{\text{packets}}{\text{ms}}$ . The rates are defined in terms of data packets, and you can assume that ACKs are infinitely small and transmitted instantly. The router has a buffer  $b$  for data packets with a capacity of 1200 *packets*.



The only traffic in this network is sent from the sender to the receiver. The sender is using a time-based AIMD congestion control algorithm with packet loss as its congestion signal.

- (i) Name an example of a real congestion control algorithm that updates its *cwnd* as a function of time since the last loss, and an example of an algorithm that updates its *cwnd* after every RTT. Describe a disadvantage of the RTT-based approach. (1 Point)

**Solution:**

Time-based: TCP CUBIC

RTT-based: TCP RENO

Disadvantage: Multiple options, e.g.:

- Problems with Bufferbloat: ramp-up becomes very slow with high RTTs.
- Unstable behavior if RTT varies a lot.
- Unfairness if multiple flows have different RTTs.

**Grading:**

+0.5 pt for the correct algorithms.

+0.5 pt for an appropriate disadvantage.

- (ii) Compute  $RTT_{max}$ , i.e., compute the maximum time between sending a (not lost) data packet and receiving its ACK in this network. Explain your steps. (1 Point)

**Solution:**

The maximum RTT is reached when the buffer is full and the packet is the  $b_{max}$ -th packet in the buffer (for any larger packet, the buffer would overflow and the packet would be dropped). As ACKs are transmitted instantly, the maximum RTT is the time it takes for the packet to arrive at the receiver.

$$RTT(b_{max}) = \frac{1}{r_{in}} + \frac{b}{r_{out}} = \frac{1 \text{ packet}}{100 \frac{\text{packets}}{\text{ms}}} + \frac{1200 \text{ packets}}{24 \frac{\text{packets}}{\text{ms}}} = 0.01 \text{ ms} + 50 \text{ ms} = 50.01 \text{ ms}$$

**Grading:**

+0.5 pt for the result.

+0.5 pt for explanation mentioning at least the full buffer.

The sender algorithm updates its congestion window ( $cwnd$ ) and sending rate  $R$  as follows:

- When loss is detected, the  $cwnd$  is halved and a new *congestion epoch* begins.
- The sender detects loss **instantly** at time  $t$  if the buffer is: (i) full ( $b(t) = 1200$ ) ; and (ii) still filling ( $\frac{db}{dt}(t) > 0$ ). You do not need to consider duplicate ACKs or timeouts.
- During a *congestion epoch*, use the following equation to determine the congestion window at the  $t$  since the beginning of the epoch, where  $cwnd_0$  is the  $cwnd$  at epoch start:

$$cwnd(t) = cwnd_0 + t \cdot 2 \frac{\text{packets}}{ms}$$

- During a *congestion epoch*, use the following linear approximation for the sending rate:

$$R(t) = \frac{cwnd(t)}{50 ms}$$

- Assume the algorithm to be in steady state, oscillating around its operating point. At the beginning of a *congestion epoch*, the  $cwnd$  is 800, and at the end, it is 1600.

- (iii) Analyze the *change* in buffer size over time. In the plot below, draw  $\frac{db}{dt}$  between  $t = 0ms$  and  $t = 500ms$ . Assume that at  $t = 0s$ , a new *congestion epoch* has just started and the buffer is full at  $b = 1200 \text{ packets}$  (otherwise we would not have detected loss).

*Hint: At which rates does the buffer fill and drain, and how long is a congestion epoch?*  
(6 Points)

**Solution:**



The buffer fills proportional to  $R(t) = \frac{800 \text{ packets} + t \cdot 2 \frac{\text{packets}}{ms}}{50 ms}$ ; this rate starts at  $16 \frac{\text{packets}}{ms}$  and increase by  $2 \frac{\text{packets}}{ms}$  per  $50 ms$ .

The buffer drains at a constant rate of  $r_{out} = 24 \frac{\text{packets}}{ms}$ .

A *congestion epoch* lasts until  $cwnd(t) = 1600$ , which is reached at  $t = 400ms$ .

Within an *congestion epoch*, we can thus write:

$$\frac{db}{dt} = 16 \frac{\text{packets}}{ms} + \frac{t}{25} \frac{\text{packets}}{ms} - 24 \frac{\text{packets}}{ms} = -8 \frac{\text{packets}}{ms} + \frac{t}{25} \frac{\text{packets}}{ms}$$

There are three critical points in the plot:

1. At  $t = 0ms$ , the buffer drains at a rate of  $-8 \frac{packets}{ms}$ .
2. At  $t = 200ms$ , we are in equilibrium.
3. At  $t = 400ms$ , we have filled the buffer just as much as we have drained it since  $t = 0ms$ . As we started with a full buffer, it will be full again at this point. Consequently, we again detect a drop and reduce the congestion window to  $800 packets$  again, resulting in a draining rate of  $-8 \frac{packets}{ms}$ .

**Grading:**

+2 pt for the correct starting point.

+2 pt for the correct rate.

+2 pt for the correct ending point, adjusted for starting point and rate if they are wrong.

- (iv) Is the buffer ever empty? If yes, give a time  $t$  at which this happens. If not, give the minimum buffer occupancy  $b_{min}$ . (2 Points)

**Solution:**

No, the buffer never completely drains. We can see that the minimum is reached at  $t = 200ms$ , as afterward, the buffer fills faster than it drains. Considering that we start with a full buffer, this gives us the minimum size:

$$\begin{aligned}
 b_{min} &= 1200 + \int_0^{200ms} \frac{db}{dt} dt \\
 &= 1200 + \left[ -8t + \frac{1}{50ms} t^2 \right]_0^{200ms} \frac{packets}{ms} \\
 &= 1200 - 8 \cdot 200ms + 4 \cdot 200ms = 400 packets
 \end{aligned}$$

**Grading:**

+1pt for correctly stating that it is never empty.

+1pt for the correct  $b_{min}$ .

- (v) Consider a second flow on the same sender that uses a *delay-based* congestion control algorithm instead of a loss-based one: a new *congestion epoch* starts if the observed RTT exceeds a threshold  $RTT_{thres}$ .

Is it possible to choose a  $RTT_{thres} < RTT_{max}$  such this both the loss-based and delay-based algorithm share the bottleneck bandwidth fairly?

Provide an example or explain why it is impossible.

*Hint: Which algorithm reacts to congestion first?* (1 Point)

**Solution:**

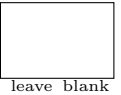
The algorithms will not share the bandwidth fairly. As we can see above, loss only happens when the buffer is already full. While the buffer is filling, the delay is already increasing, thus the delay-based algorithm will react first and reduce the congestion window, yielding bandwidth to the loss-based algorithm.

**Grading:**

+1pt for the correct answer (only if the explanation is correct).

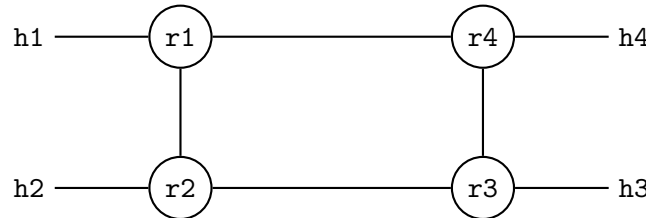
## c) BBR Control Space

(6 Points)



In this task, you will discuss why BBR uses *both* a congestion window and a pacing rate.

Consider the network shown below, where two flows transmit data using a simplified version of BBR. Router buffers have a size of **200 packets**.



**At time  $t = 0s$ ,** the two flows follow disjointed paths.

Flow one follows the path  $h1, r1, r4, h4$ ; flow two follows the path  $h2, r2, r3, h3$ .

**At time  $t = 5s$ ,** flow one is rerouted and now follows the path  $h1, r1, r2, r3, r4, h4$ ; both flows now share the bottleneck link  $r2, r3$ .

To simplify the state measurements, assume that the bottleneck bandwidth  $BtlBw$  and propagation delay  $RTTProp$  are known exactly, but changes are detected with  $100ms$  delay. The table below shows the state as well as the computed pacing rate and congestion window for both flows. The left side shows  $t < 5.1s$  and the right table shows  $t \geq 5.1s$ , reflecting the  $100ms$  delay in detecting the change after the top flow has been rerouted at  $t = 5s$ .

|          |              |                     | $t < 5.1s$ | $t \geq 5.1s$ |
|----------|--------------|---------------------|------------|---------------|
| Measured | $BtlBw$      | $\frac{packets}{s}$ | 1000       | 500           |
|          | $RTTProp$    | $ms$                | 120        | 120           |
| Computed | $PacingRate$ | $\frac{packets}{s}$ | 1000       | 500           |
|          | $Cwnd$       | $packets$           | 120        | 60            |

- (i) What if BBR would *only use the provided congestion window* without a pacing rate? Explain a downside and give a concrete numeric example in the scenario above in terms of reduced throughput and/or additional delays and/or packet losses. (3 Points)

**Solution:**

Without pacing, traffic becomes bursty which can lead to packet loss. In our examples, this leads to packet loss if the bursts of both flows coincide between  $t = 5s$  and  $t = 5.1s$ . If they have not updated their state to the new bottleneck bandwidth yet, they both send a burst of 120 packets. This exceeds the buffer size of 200 packets, resulting in lost packets.

Alternatively, one can also show the increased delay because of buffer buildup; e.g. at  $t = 0s$ , if all packets are sent at once, there will be 120 packets in the buffer, resulting in an increased delay.

**Grading:**

+1pts for the answer.

+1pts if an appropriate example is given.

+1pts if the numbers provided in the example are correct.

Point deductions if additional, wrong answers are given.

- (ii) What if BBR would *only use the provided pacing rate* without a congestion window? Explain a downside and give a concrete numeric example in the scenario above in terms of reduced throughput and/or additional delays and/or packet losses. (3 Points)

**Solution:**

Without a congestion window limiting the number of in-flight packets, BBR might end up with a standing queue at the bottleneck link.

Consider the time  $t = 5.1s$  in our example. Before, the flows send with a combined rate exceeding the link rate, resulting in a full queue and packet loss. Afterward, they send exactly at the link rate. No more packets are lost, but the queue remains full. With a congestion window, the flows would recognize that there are currently too many packets in-flight and pause sending.

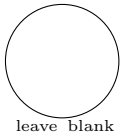
Concretely we see that the bottleneck link can support a rate of  $1000 \frac{\text{packets}}{\text{ms}}$  (from the provided rates). Between  $t = 0s$  and  $t = 5.1s$ , the flows send at a combined rate of  $2000 \frac{\text{packets}}{\text{ms}}$ , resulting 200 *packets* in the buffer which will never drain.

**Grading:**

+1pts for the answer.

+1pts if an appropriate example is given.

+1pts if the numbers provided in the example are correct.

**Task 7: Sustainable networking****11 Points**

- (i) The Greenhouse Gas (GHG) protocol classes emissions into three scopes. Explain briefly which type of emissions fall into each scope. (3 Points)

**Solution:** Scope 1: emissions from the fossil fuels you burn yourself

Scope 2: emissions from the fossil fuels burned to generate the electricity you purchase

Scope 3: emissions from your entire supply chain

**Grading:** +1 pt for each correct description

- (ii) Let us assume that the carbon footprint of streaming Netflix in Switzerland is estimated to be 55gCO<sub>2</sub>e per hour of streaming. How much would carbon emissions be reduced if you stream only one hour and go to bed instead of streaming two hours? Explain your answer. (2 Points)

**Solution:** Essentially zero. Carbon footprint numbers come from an attributional analysis, not consequential reasoning. 55gCO<sub>2</sub>e represents the average footprint for the entire Netflix service, but it does not capture the carbon cost of one hour of streaming!

**Grading:** 0.5 for the correct answer without explanation; 1 for the correct answer with partial explanation; 2 for a valid explanation.

- (iii) Let us assume a new-generation router consumes half the energy of its older generation. From a sustainability standpoint, why is it not always better to upgrade to the new generation? (2 Points)

**Solution:** One must account for the embodied footprint of the new router. If the device is lightly loaded, it will take a very long time until the savings in the operational footprint compensate the embodied carbon footprint.

**Grading:** 1pt for mentioning embodied footprint. 2pt for a compelling explanation.

- (iv) Name two energy-saving techniques that can theoretically save tens of % of energy in lightly-loaded networks. (2 Points)

**Solution:** Sleeping and rate-adaptation.

**Grading:** 1pt for each correct answer.

- (v) In practice, what limits the efficiency of these techniques today? (2 Points)

**Solution:** It takes too long for hardware to turn components on/off to sleep/adapt efficiently.

**Grading:** 1pt for a half-baked answer. 2pt for a compelling one.