ETH zürich

Institut für Technische Informatik und Kommunikationsnetze

Department ITET
August 2023

Prof. Dr. Laurent Vanbever
C. Busse-Grawitz, T. Bühler, Y. Chen, T. Schneider

Exam: Communication Networks

15 August 2023, 13:30–16:00, Room HIL F 75 and HIL G 75

Sample Solution

General remarks:

- ▷ Write your **name** and your **ETH student number** below on this front page and **sign it**.
- ▷ Put your **legitimation card** on the top right corner of your desk. Make sure that the side containing your name and **student number** is visible.
- ▷ Check that you have received **all task sheets** (Pages **1 – 36**).
- ▷ Do **not separate** the **task sheets** as we collect the exams **only after you left** the room.

- ▷ Write your answers directly on the task sheets.
- ▷ **All answers fit within the allocated space and often in much less.**
- ▷ If you need more space, use the three extra sheets at the **end of the exam**. Indicate the **task** in the corresponding field.

- ▷ **Read each task completely before you start solving it**.
- ▷ **For the best mark, it is not required to score all points.**

- ▷ Please answer either in **English or German**.
- ▷ **Write clearly** in blue or black ink (not red) using a **pen**, not a pencil.
- ▷ **Cancel** invalid parts of your solutions **clearly**.
- ▷ At the end of the exam, **place the exam face up on the top left corner** of your desk. Then collect all your belongings and **exit the room** according to the given instructions.

Special aids:

- ▷ All written materials (vocabulary books, lecture and lab scripts, exercises, etc.) are allowed.
- ▷ Using a calculator is allowed, but the use of electronic communication tools (mobile phone, computer, etc.) is strictly forbidden.

Family name: Student legi nr.:

First name: Signature:

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

Task	Points
Ethernet & IP	/33
Intra-domain routing	/24
Inter-domain routing	/35
Reliable transport	/36
Applications	/22
Total	/150

leave blank

Task 1: Ethernet & IP — 33 Points

leave blank

a) Warm-Up — (5 Points)

You are participating in a large LAN party with some friends from CommNet. The LAN consists of one Layer-2 network without VLANs. It runs on several Layer-2 switches and has a connected DHCP server, no routers are involved. Also, all participants' interfaces are connected to the LAN.

Note: You know the MAC addresses of all your friends.

(i) You are already set up, but one friend is late. You start recording packets on your interface while it is *not* in promiscuous mode. Can you see when your friend arrives (and connects his computer)? Explain why, or why not. (1 Point)

Solution: Yes, the friend's DHCP request is broadcasted. There are no VLANs, so he will be in the same broadcast domain.

(ii) You suddenly spot your delayed friend in the crowd but lose sight of him. You want to use `traceroute` to see what switch he's connected to, and so locate him. Explain why this idea does not work. (1 Point)

Solution: Examples for accepted reasons:

- You don't know his IP address, since the DHCP reply is unicasted.
- Switches don't decrement the TTL, so traceroute won't report the switches on the path/switches are layer-2 devices, so they don't appear in traceroute output.

(iii) Your CommNet friends randomly change the IP addresses of their interfaces. How can you check who of them accidentally uses your IP address, only by sending custom packets and looking at the incoming packets? (2 Points)

Solution: Send an ARP request: "Who has IP < your IP>? Tell < your IP >". You receive ARP replies from all devices using your MAC address. (not expected as part of the answer: Since you know the mapping of friend to MAC address(es), you can also identify the concrete friends that use your IP address.)

(iv) Inspired by your friends, you decide to change the **MAC** address of your interface. You inform them via SMS of your new MAC address. Can you already receive packets (destined specifically to your MAC address) even though none of the switches has learned your new address yet? Explain what the switches would do in this case. (1 Point)

Solution: Yes, you can already receive packets. If a switch does not have a forwarding entry for a MAC address, it broadcasts the packet to all (active) interfaces (except the one on which it received the packet). Therefore, the packets destined to you will reach you already now.

b) Per-VLAN spanning trees **(4 Points)**

leave blank

Consider the L2 network in Figure 1. It consists of seven switches (A - G) and links with different bandwidths (high and low). Additionally, one router (R), two hosts (H) and two servers (S) are connected to different switches. All hosts belong to VLAN 1 while all servers belong to VLAN 2. (4 Points)

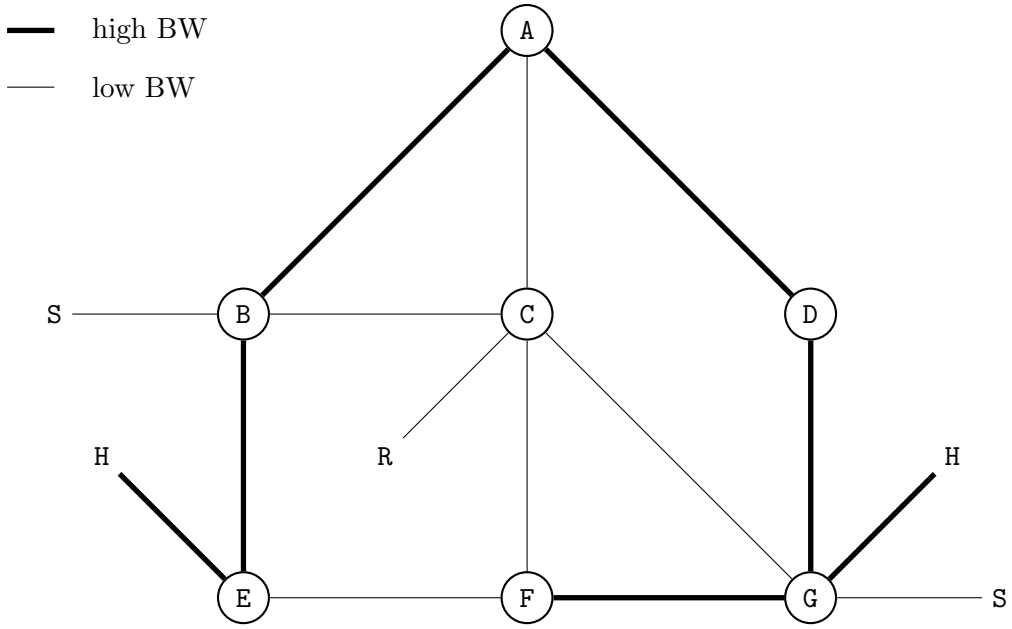


Figure 1: An L2 network with different VLANs.

Find two per-VLAN spanning trees (one for VLAN 1 and one for VLAN 2) such that the following conditions hold:

- Hosts and servers can reach each other;
- The number of hops (i.e., switches) between hosts is minimized;
- The traffic exchanged between servers uses high-bandwidth links.

Clearly indicate in the figure if a link belongs to VLAN 1, VLAN 2, both or neither (in which case, do not write anything next to the link). Note that the links to hosts/servers should also belong to VLAN(s).

Solution: Multiple solutions are possible. For example:
VLAN 1 contains: H-E, E-F, F-G, G-H, and C-F
VLAN 2 contains: S-B, B-A, A-D, D-G, G-S, and A-C
R-C belongs to both VLANs

leave blank

c) Spanning trees with multiple roots **(6 Points)**

For the following two questions we explore what happens to the spanning tree protocol if multiple switches have the same ID. We make the following assumptions: (i) There are no VLANs; and (ii) if a switch knows two paths towards the root of the spanning tree with equal hop count (i.e., same amount of switches on the path), it picks the path over the neighbor with the lowest ID.

(i) Consider the L2 network in Figure **??** which consists of eleven switches. The shown numbers indicate the switch IDs. Note that there are *two* switches with the lowest ID (ID 1). Draw the resulting spanning tree directly onto the figure. (3 Points)

Solution:

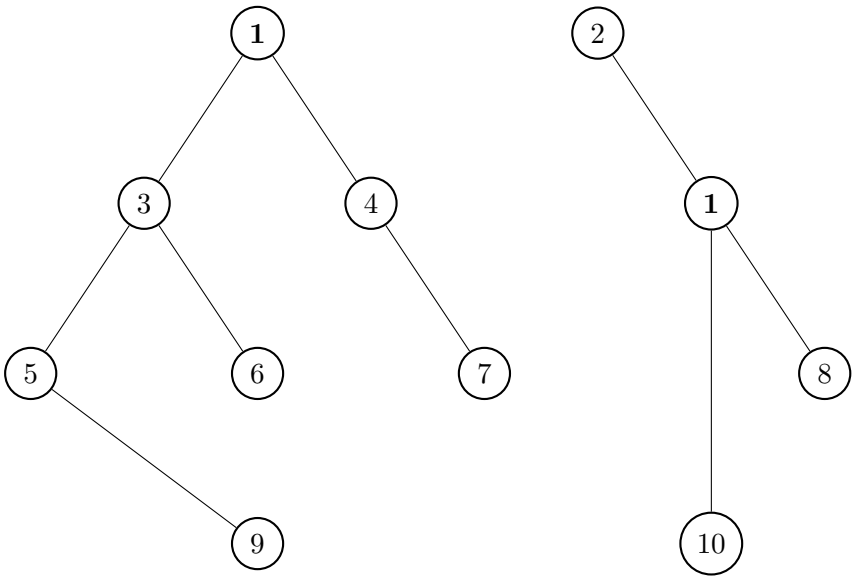


Figure 2: Solution

(ii) Consider now that you have an arbitrary L2 network which contains *two* switches with the lowest ID (ID 1). All other switches have unique IDs. In such a scenario, is it possible that the computed spanning tree leads to a connected network, that means all nodes are part of the *same* spanning tree? If you think that is possible, draw a corresponding network below. Otherwise argue why it is not possible. (3 Points)

Solution: It is not possible. Given the clear tie-breaking mechanism, there will never be a switch which is connected to both roots (over the spanning tree).

leave blank

d) A network with tunnels **(9 Points)**

In this question, we consider the network in Figure 3 which consists of six routers. A packet with IP address `2001::A47C:5` enters the network via router `R1` and should eventually reach the corresponding host connected to router `R6`. In addition, the figure shows the forwarding table of each router. An entry in a forwarding table consists of an ID, a prefix and an action. The action is either a specific next hop (`NH`) to which the packet is forwarded or a tunnel operation. For tunnels, the action either adds (`add hdr`) or removes (`del hdr`) an IP header with a new destination IP. ***After tunnel operations, the packet re-enters the same forwarding table again.***

router 2

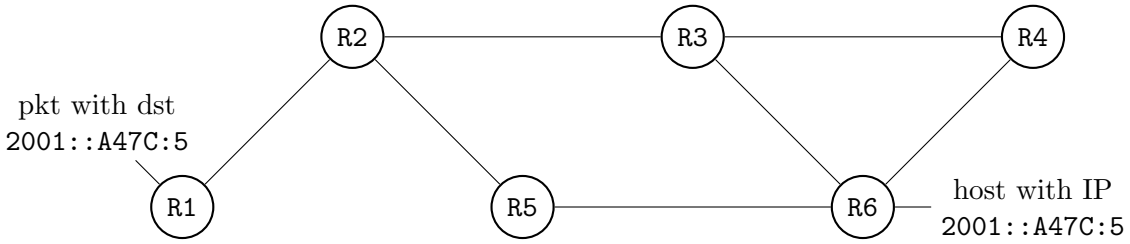
ID	prefix	action
2.1	2001::A478:0/110	NH: R5
2.2	2001::A47C:0/111	add hdr 3.64.106.9
2.3	3.64.104.0/21	NH: R3
2.4	3.64.100.0/22	NH: R5

router 3

ID	prefix	action
3.1	2001::A47C:0/112	NH: R6
3.2	81.9.0.0/16	NH: R4
3.3	3.64.64.0/18	add hdr 81.9.0.15
3.4	3.64.104.0/23	NH: R6

router 4

ID	prefix	action
4.1	0::0/0	NH: R6
4.2	0.0.0.0/0	NH: R6
4.3	3.64.0.0/16	del hdr 3.64.106.9
4.4	81.9.0.12/30	del hdr 81.9.0.15



router 1

ID	prefix	action
1.1	2001::A400:0/105	NH: R2
1.2	2001::A440:0/106	NH: R2
1.3	2001::A450:0/108	NH: R2
1.4	2001::A47A:0/111	NH: R2

router 5

ID	prefix	action
5.1	2001::A000:0/100	NH: R6
5.2	3.64.106.0/24	del hdr 3.64.106.9
5.3	3.64.106.9/32	add hdr 81.9.0.15
5.4	81.9.0.0/18	NH: R6

router 6

ID	prefix	action
6.1	2001::0/16	NH: host
6.2	3.64.106.9/32	del hdr 3.64.106.9
6.3	81.9.0.15/32	del hdr 81.9.0.15
6.4	2001::A47C:5/128	NH: host

Figure 3: A L3 network with tunnels.

(i) Assume the packet with destination IP `2001::A47C:5` just reached router `R1`. According to which forwarding table entry in `R1` is the packet forwarded towards `R2`? Write down the ID and explain your choice. (3 Points)

Solution: We need to find the longest-prefix match which is 1.2

(ii) We now continue to track the path of the packet with destination IP `2001::A47C:5` through the network. Assume that the packet just reached router `R2`. Fill out the table below and always indicate the ID of the forwarding table entry that matched (e.g., 3.2) and the destination IP that you used (at the moment `2001::A47C:5`). Additionally, indicate the number of IP headers that the packet contains *after* you apply the action of the forwarding table entry. ***Currently, the packet contains a single IP header with destination IP `2001::A47C:5`.*** You can stop once the packet reaches the host connected to `R6`.

Hint: You might not need all the rows in the table. (6 Points)

Solution: (1.2 - 2001::A47C:5 - 1 hdr)
2.2 - 2001::A47C:5 - 2 hdr
2.3 - 3.64.106.9 - 2 hdr
3.3 - 3.64.106.9 - 3 hdr
3.2 - 81.9.0.15 - 3 hdr
4.4 - 81.9.0.15 - 2 hdr
4.3 - 3.64.106.9 - 1 hdr
4.1 - 2001::A47C:5 - 1 hdr
6.4 - 2001::A47C:5 - 1 hdr

e) CSMA/CD (9 Points)

leave blank

Here, you simulate two CSMA/CD exchanges on an old setup with a shared copper cable.

Setup: Figure 4 shows the setup. To simplify the delay model, we omit units. The copper cable has a propagation speed of 1, and a transmission rate of 1. For instance, if A sends a frame of length 1, B will start hearing it after 1 time step (since the distance is 1), and stop hearing it at after 2 time steps; C will start hearing it after 2 time steps, and stop hearing it after 3 time steps (the table in (ii) directly illustrates this).

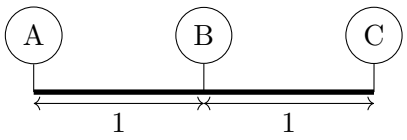


Figure 4: Shared copper cable setup.
The distances A – B and B – C are 1.

(i) Given the copper cable setup in Figure 4, what should be the minimal frame length? What problem can occur when the frames are too short? (Note that the following subtasks may or may not violate this requirement.) (2 Points)

Solution:
If the frame length is too small, collisions may go undetected. Therefore, the minimal frame length should double the maximal latency; here, this is $2 * 2 = 4$.

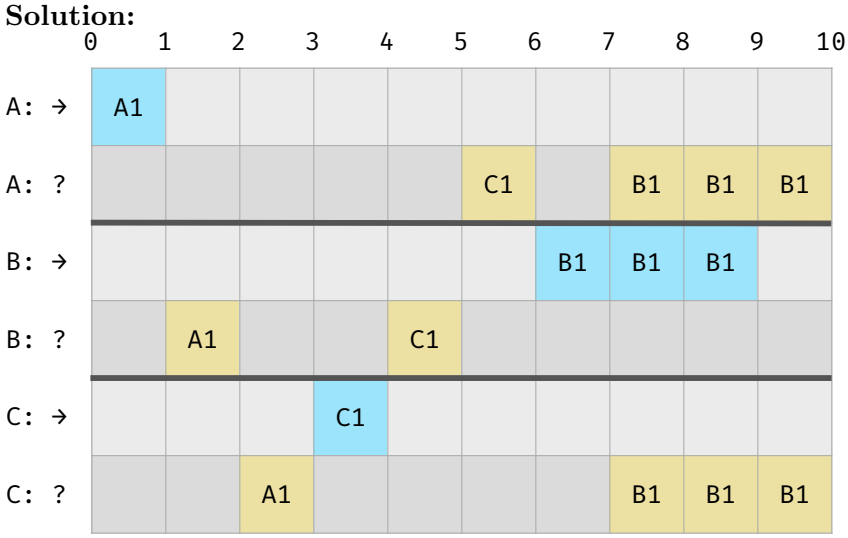
Next, you will simulate two message exchanges by filling in tables.

Instructions on how to fill in the tables: The tables have two rows per sender: Fill in the top row with what frames a sender *sends*, the bottom row with what frames it *hears*.

(ii) Fill in the **top table on the next page** for the following *demands*; we filled in the table for the first demand as an example. (3 Points)

- time 0: A wants to send frame A1 with length 1
- time 3: C wants to send frame C1 with length 1
- time 6: B wants to send frame B1 with length 3

Solution:



(iii) Fill in the table below for the following *demands*, including the row `C: hear`. Read the notes below carefully. (4 Points)

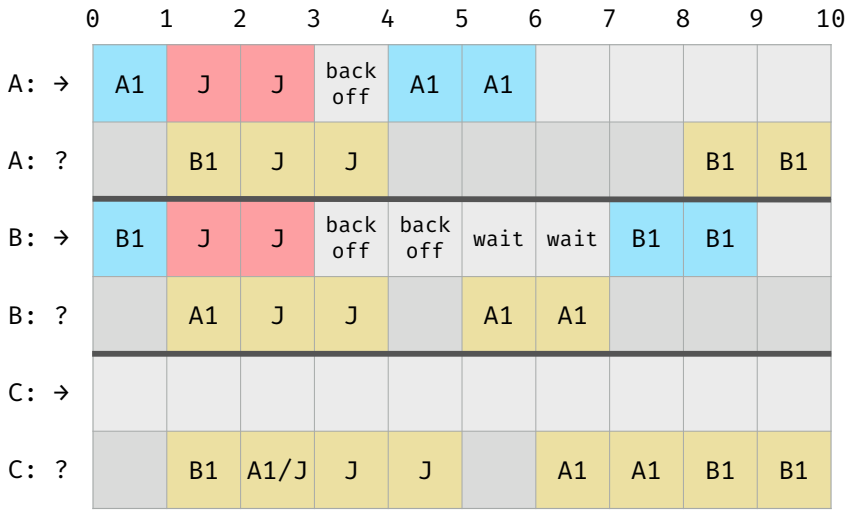
- time 0: A wants to send frame A1 with length 2
- time 0: B wants to send frame B1 with length 2

Notes on the protocol behavior:

- A sender detects a collision if and only if it is currently sending a frame *and* hears someone else sending.
- If a sender detects a collision, it immediately stops sending its data and starts sending a jam signal of length 2.
- If a sender *starts* to hear something just at time x, and it also wants to send a frame at time x, it *waits* with sending, and therefore also does not create a collision.
- For simplicity, the senders have a constant back-off time of 1 for A, 2 for B, and 3 for C.
- If jam signals overlap, this does not count as a collision or corruption.
- If jam signals interfere with data frames, this does not count as a collision.
- Since the copper cable is shared, data frames get corrupted if they interfere, but continue travelling across the copper wire.

Note on the table: Denote jam signals with `J`, and corrupted frames with a slash `/`.

Solution:


0 1 2 3 4 5 6 7 8 9 10
A: →
A: ?
B: →
B: ?
C: →
C: ?
A1
J
J
back off
A1
A1
B1
J
J
J
B1
B1
B1
J
J
back off
back off
wait
wait
B1
B1
A1
J
J
A1
A1
B1
A1/J
J
J
A1
A1
B1
B1

leave blank

Task 2: Intra-domain routing **24 Points**

leave blank

a) Warm-Up (6 Points)

(i) Besides the obvious scalability problems, describe one other reason why we cannot use a link-state protocol (e.g., OSPF) for Internet-wide routing (i.e., across different ASes). (1 Point)

Solution: Multiple answers are possible. It would reveal details about your topology to all other ASes, you would need a unified concept of weights between all ASes, ...

(ii) We use a link-state protocol in a small network. The flooding process of link-state advertisements as well as the Dijkstra computation are very fast. In such a setup, which are the main factors that influence how long it takes until new shortest paths are computed after a link failure? (1 Point)

Solution: The frequency of hello messages as well as the failure condition (i.e., amount of missing hello messages to trigger a failure).
Other solutions might also be possible.

(iii) In a distance-vector protocol, does a router know the precise path a packet takes towards its destination? Explain your answer. (1 Point)

Solution: No, a single router only receives distance vectors which do not reveal the entire topology. Only precise knowledge of directly connected routers.

For the remaining three questions, we consider the network in Figure 5. All the routers on the left side run OSPF and use IPs inside `10.0.0.0/8`, while the routers on the right side run IS-IS and use IPs inside `20.0.0.0/8`. The link weights used by the two protocols are indicated in the figure. Router `X` and `Y` in the middle take part in both protocols. To ensure connectivity between both sides, router `X` and `Y` advertise the corresponding `/8` prefixes to the other side. More precisely, `X` and `Y` advertise the `10.0.0.0/8` prefix into IS-IS and the `20.0.0.0/8` prefix into OSPF.

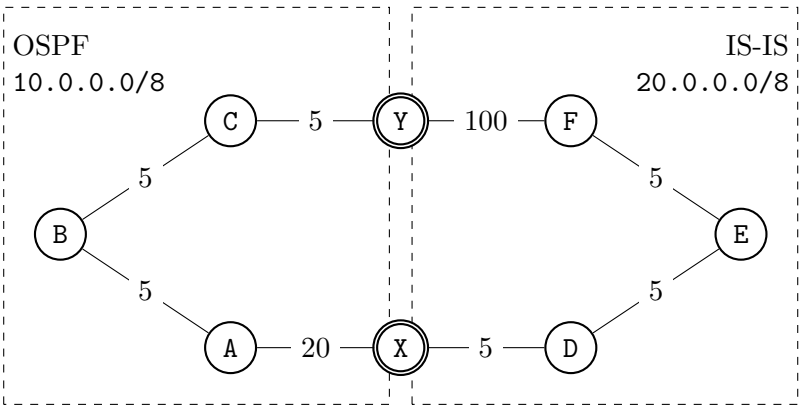


Figure 5: A network which runs OSPF and IS-IS at the same time.

(iv) Give one reason why a network operator might deploy two different intra-domain routing protocols in the same network. (1 Point)

Solution: Multiple answers are possible. It could be that certain devices only support a specific protocol. It could also be that the specific network parts (e.g., topology) are better suited for one specific protocol.

(v) Briefly explain why the current way of advertising the `20.0.0.0/8` prefix into OSPF leads to suboptimal paths between the OSPF part and the IS-IS part. (1 Point)

Solution: No. For example, if router B wants to send traffic towards E, it will use the shortest path towards Y. But Y then needs to use a path with a very high cost to reach E.

(vi) Find better prefixes to advertise into the OSPF part such that the problems described in **(v)** are minimized. Make sure that reachability between the OSPF and IS-IS parts remains, even if one of the middle routers (`X` or `Y`) fails. (1 Point)

Solution: In addition to 20.0.0.0/8, X should also advertise 20.0.0.0/9 and 20.128.0.0/9 into OSPF.

leave blank

b) Traffic engineering with a link-state protocol (9 Points)

We focus on the network in Figure 6 which consists of five routers running a link-state protocol. The link costs are bidirectional (they apply in both directions). The links have different bandwidths, between 1 and 100 Gbps.

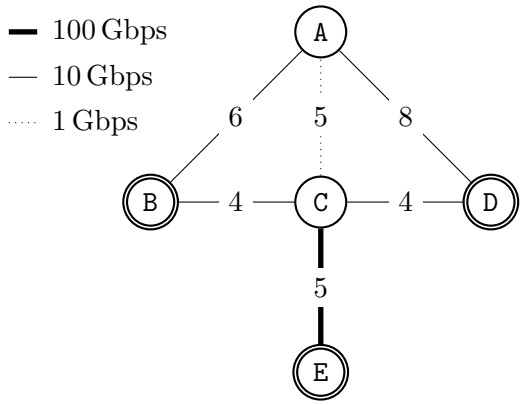


Figure 6: The link between router `A` and `C` has the lowest bandwidth.

As a reminder, in a link-state protocol, each router first floods their local view using Link-State Advertisements (LSA). Afterwards, each router individually constructs the entire network topology based on the received LSAs. Finally, each router computes shortest path(s) towards the different destinations.

We make the following assumption: If a router knows multiple shortest paths towards a destination, it will load-balance its traffic equally over all of them.

(i) Router `A` wants to send 50 Gbps of traffic towards router `E`. Given the network in Figure 6, which path(s) will the traffic take? Do you see a problem? (1 Point)

Solution: The path is A, C, E. That is not optimal as it uses the link with the lowest bandwidth.

(ii) You modify the cost of the link between router `A` and `C` (5 in Figure 6). Assume you start with a cost of 3 and increase it up to 15. Indicate in the table below the total throughput achieved between router `A` and `E` for the different link costs. Group link costs that achieved the same throughput together by writing ranges, for instance: $3 \leq cost < 5$. Make sure your answer covers the entire range from 3 to 15.

Hint: You might not need all rows. (3 Points)

Solution: For $3 \leq cost < 10$ total throughput is 1 Gbps.
For $cost = 10$ total throughput is 11 Gbps. For $cost > 10$ total throughput is 10 Gbps.

(iii) To achieve an even higher throughput, you would like to load balance the traffic from router `A` towards router `E` over `A`'s three neighbors: `B`, `C` and `D`.

Unfortunately, you can only access router `B`, `D`, and `E` (indicated in Figure 6 with double circles), *not* `C`. Assume that router `B`, `D` and `E` can create *crafted* LSAs for their own, local network view which are then flooded in the entire network. Fill in the tables below with the crafted LSAs such that eventually, router `A` will start to load-balance its traffic towards router `E`. The link cost between router `A` and `C` (5) is unchanged.

Important: Keep in mind that router `C` also takes part in the link-state protocol but is outside of your control. Therefore, make sure that the LSAs do not contradict each other. As an example, if an LSA from router `C` claims to have a link towards `B` with cost 4. It would be confusing if router `B` claims to have a link towards router `C` with cost 10.

Hint: You might not need all the rows in the tables. Also, Figure **??** is identical to Figure 6 and shown for convenience.

(5 Points)

Solution: To achieve load balancing, B and E can claim to have a direct link, the same holds for D and E. It is important that all the shortest paths have a total cost of 10, which is the same total cost as the direct path from A over C to E. Possible solution:

LSA of B

link to	**cost**
A	6
C	4
E	4

LSA of E

link to	**cost**
B	4
C	5
D	2

LSA of D

link to	**cost**
A	8
C	4
E	2

leave blank

c) Distance-vector protocol with a cheating node **(9 Points)**

Throughout the entire question we consider the network in Figure 7 which consists of four routers. The link costs are bidirectional (they apply in both directions). All routers in the network run a distance-vector protocol. No special features (e.g., poisoned reverse) are used. The link between router `A` and `C` has a higher bandwidth than the other links.

As a reminder, once a router receives a distance vector over a given link, it adds the corresponding link cost to all entries in the distance vector and then updates its internal distances if the vector contains lower costs for certain destinations.

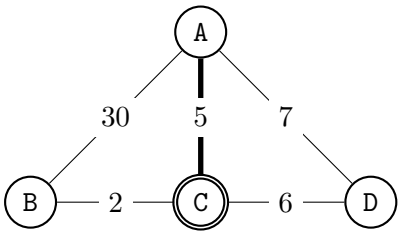


Figure 7: The link between router `A` and `C` has a high bandwidth.

(i) For routers `A`, `B` and `D`, fill in the tables below with the cost and next-hop for each destination computed by the protocol once converged. The first entry in the table of router `A` is already given. (3 Points)

Solution:

Router A

dst	cost	next hop
A	0	A
B	7	C
C	5	C
D	7	D

Router B

dst	cost	next hop
A	7	C
B	0	B
C	2	C
D	8	C

Router D

dst	cost	next hop
A	7	A
B	8	C
C	6	C
D	0	D

(ii) We now assume that router `C` is cheating to influence the traffic forwarding behavior. Concretely, router `C` is able to send crafted (modified) distance vectors to its neighbors. All the other routers follow the normal distance vector protocol and do not cheat.

Assume that router `C` knows the entire network topology (all routers and links including their costs) and can send exactly *one* crafted distance vector to each of its neighbors (`A`, `B`, `D`). Use the three tables below to indicate one possible set of distance vectors such that the following three conditions hold (once the network converges). All distances have to be integers ≥ 0:

- Router `C` observes *all* traffic in the network, i.e., all traffic crosses router `C`.
- Router `C` receives as much traffic as possible over the high-bandwidth link between `A` and `C`.
- Your solution works no matter the tie-breaking mechanism used by the distance-vector protocol.

(6 Points)

Solution:

DV to A

dst	**cost**
B	0
C	0
D	0

DV to B

dst	**cost**
A	< 28
C	> 33
D	> 33

DV to D

dst	**cost**
A	0
B	> 6
C	> 6

leave blank

Task 3: Inter-domain routing **35 Points**

leave blank

a) Warm-Up (5 Points)

You are the network operator of AS 17. Figure 8 shows your connections with AS 23.

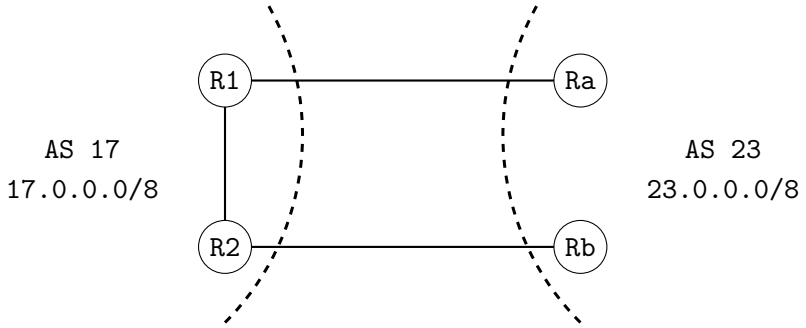


Figure 8: Connections between AS 17 and AS 23

The dashed curves represent the boundary of each AS. Each circle represents a router. Each solid line represents a BGP session. Each AS owns and announces its `/8` IP prefix.

(i) There are 3 sessions in the topology (`R1-R2`, `R1-Ra` and `R2-Rb`). Which sessions are iBGP sessions and which are eBGP sessions? (1 Point)

Solution: iBGP session: `R1-R2`. eBGP sessions: `R1-Ra` and `R2-Rb`.

All BGP sessions are working properly and AS 17 is receiving BGP routes for `23.0.0.0/8` from both `Ra` and `Rb`. Both `R1` and `R2` have **not** been configured with any routing policy. Consider whether each of the following scenarios can happen when the network converges. If your answer is yes, fill in the route attributes AS 17 could receive such that the scenario can happen (the right-most AS in an AS path is the route origin). If your answer is no, leave the table empty and explain why it is impossible.

(ii) Both `R1` and `R2` prefer `Ra`'s route. (1 Point)

Solution: The MED value for `Ra` → `R1` is smaller. Or the two MED values are the same, and `Ra` → `R1` has a shorter AS path that only contains AS number 17.

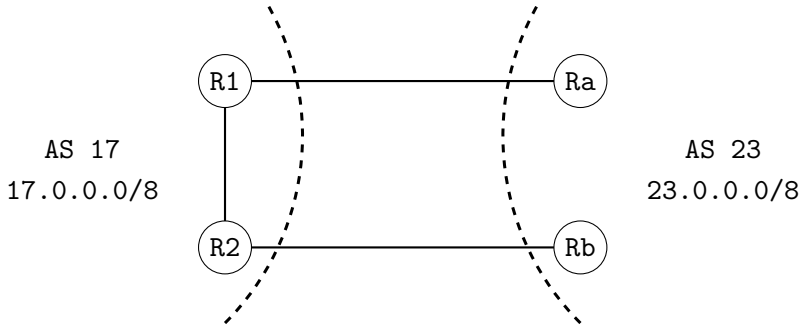


Figure 9: Copy of Figure 8

(iii) `R1` prefers `Ra`'s route and `R2` prefers `Rb`'s route. (1 Point)

Solution: Two routes have the same attribute values for both MED and AS path.

(iv) `R1` prefers `Rb`'s route and `R2` prefers `Ra`'s route. (2 Points)

Solution 1: This scenario is impossible. If `R1` prefers `Rb`'s route which it receives from `R2`, `R1` will withdraw `Ra`'s route because BGP only exports the best route. Then `R2` cannot select `Ra`'s route anymore.

Solution 2: Impossible since when MED or AS path are different in 2 external routes, both `R1` and `R2` prefers the same route. When both BGP attributes and IGP cost are the same, each router prefers route from eBGP over iBGP.

leave blank

b) Path analysis **(10 Points)**

Figure 10 shows the business relationships between ASes around AS 17. Lines with double arrows represent that two connected ASes are *peers* (AS 17 and AS 23 are peers). Lines with a single arrow point from a *provider* to a *customer* (AS 17 is a provider of AS 29). All ASes have correctly configured their routing policies to follow the above business relationships. No AS has configured any other routing policy. Each AS announces its own `/8` prefix to **all** BGP neighbors (AS X announces `X.0.0.0/8`).

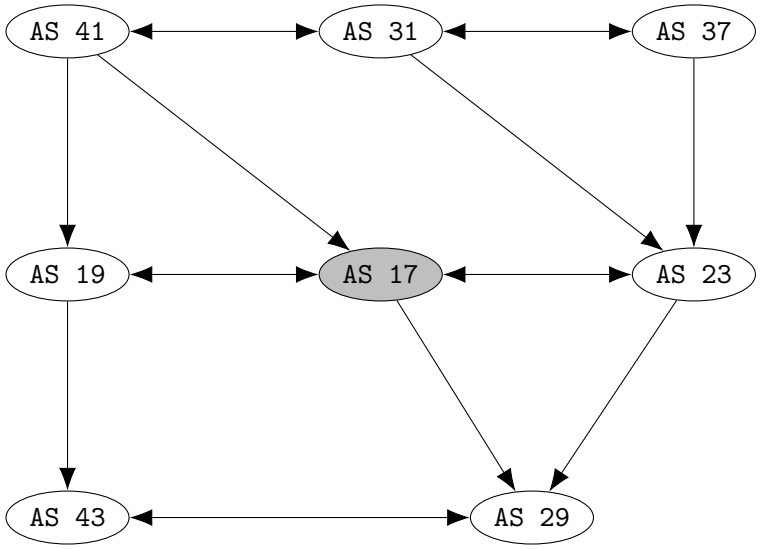


Figure 10: AS topology

(i) What is the AS path of the route that AS 17 selects to reach `41.0.0.0/8`? Note that the right-most AS in an AS path is the route origin. (1 Point)

Solution: AS 17 only learns `41.0.0.0/8` from AS 41, its AS paths is [41].

(ii) Not all ASes can learn `37.0.0.0/8`. Is it possible to add **one peer** session between any two ASes such that after adding this new session, all ASes can learn `37.0.0.0/8`? If your answer is yes, directly write down the peer session. Use AS X–AS Y to represent the peer session between AS X and AS Y. You do **not** need to justify your answer. If your answer is no, explain why such a session does not exist. (3 Points)

Solution: AS 41-AS 37 or just AS 41.

Figure 11 shows the full topology of AS 17. The bold solid lines inside AS 17 represent IGP links. Each IGP link has a cost of 1. There is a BGP session between **every** pair of routers. All AS-level relationships remain the same as in Figure 10 and work properly (not all are drawn in Figure 11). All routers in AS 17 and AS 23 have been configured with business relationships but no other routing policies.

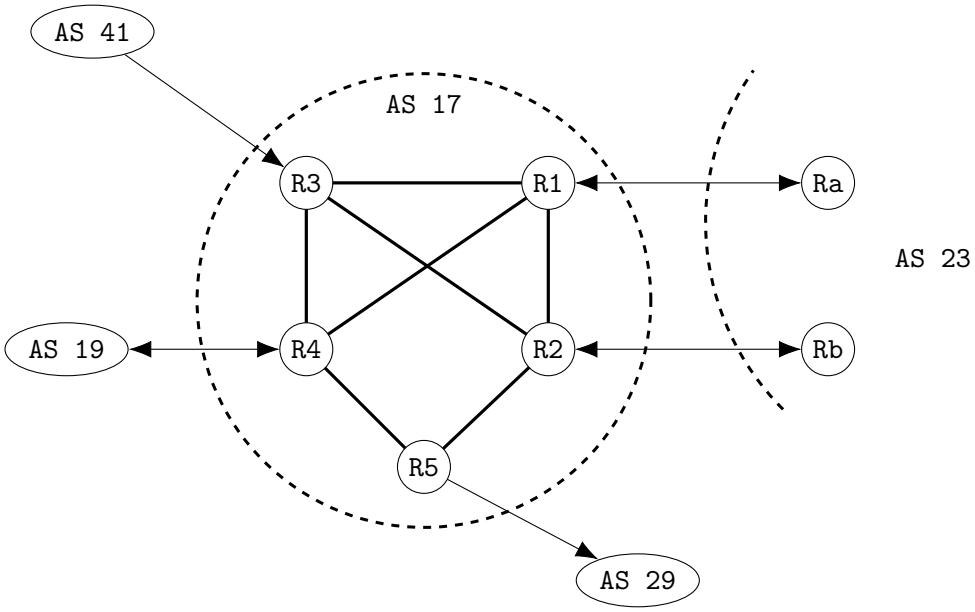


Figure 11: Full topology of AS 17 with partial AS-level relationships
(See Figure 10 for the full AS-level relationships)

(iii) What is the **minimum number** of IGP links that must fail in AS 17 to prevent `R2` from learning `43.0.0.0/8`? Write down all failed links. Use `Ri-Rj` to represent the IGP link between `Ri` and `Rj`.

Hint: A router can only send a BGP route to another router inside an AS if an IGP path exists between the two routers where all IGP links are up. (3 Points)

Solution 1: `R2` can learn the route from AS 41 or AS 19. `R1-R2`, `R2-R3` and `R2-R5`.
Solution 2: `R1-R2`, `R2-R3` and `R4-R5`.

(iv) What is the **maximum number** of IGP links that can fail in AS 17 while still allowing `R2` to learn `43.0.0.0/8`? Write down all failed links. Use `Ri-Rj` to represent the IGP link between `Ri` and `Rj`. (3 Points)

Solution: All links except `R2-R3` can fail, so it is 6 at most.

leave blank

c) BGP hijack **(9 Points)**

AS 17 would like to eavesdrop on sensitive data sent to AS 19. The business relationships shown in Figure 12 remain the same as in Figure 10. Lines with double arrows represent that two connected ASes are *peers* (AS 17 and AS 23 are peers). Lines with a single arrow point from a *provider* to a *customer* (AS 17 is a provider of AS 29). Each AS announces its own `/8` prefix to **all** BGP neighbors (AS X announces `X.0.0.0/8`). All ASes have correctly configured their business relationships but no other routing policy (e.g., each AS prefers its local route to any other route for `X.0.0.0/8`).

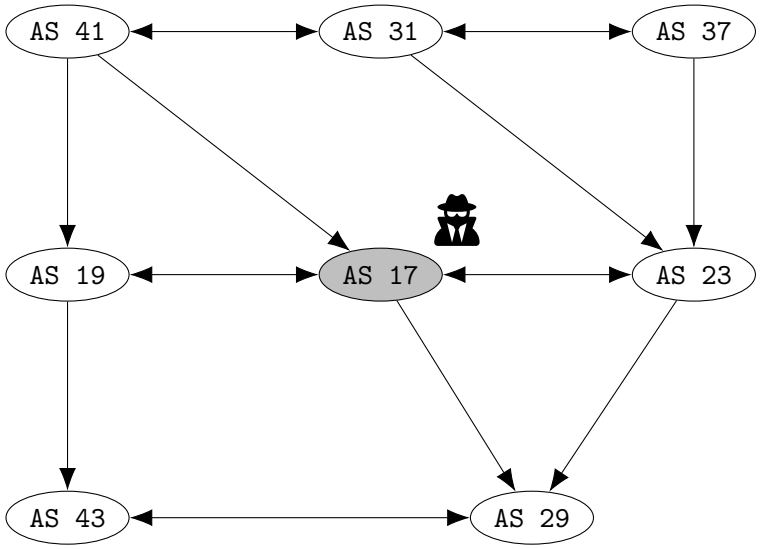


Figure 12: AS topology with the attacker

AS 17 plans to hijack `19.0.0.0/8`. However, there exists a protection mechanism called RPKI. If an AS has configured RPKI, it registers its own prefix in the RPKI database and validates all received routes. Whenever the AS receives an *invalid* route whose origin does not match the record in the RPKI database, it immediately drops the route. The route is accepted if it is *valid* or the prefix is *not found* in the RPKI.

(i) If AS 19 has configured RPKI, can AS 17 hijack `19.0.0.0/8` and eavesdrop on the data AS 41 sends to AS 19? Justify your answer. (2 Points)

Solution: AS 17 can hijack the route even if AS 41 has configured RPKI. To do so, AS 17 append AS 19 as the origin of the route with AS path poisoning, and advertise more specific prefixes to AS 41, e.g., `19.0.0.0/9` and `19.128.0.0/9`.

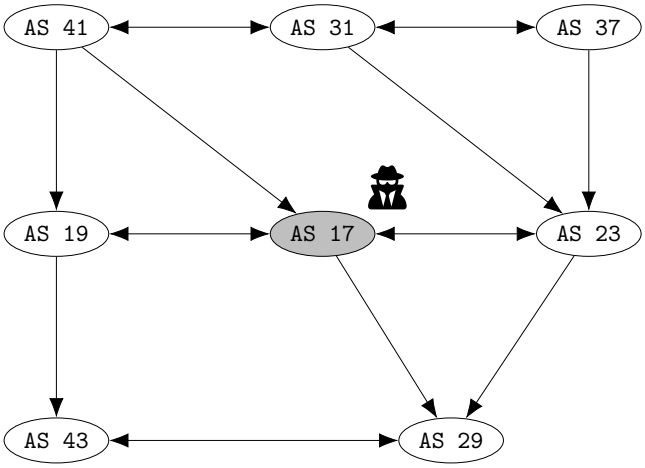


Figure 13: Copy of Figure 12

(ii) AS 17 would like to check whether AS 23 has configured RPKI. Assume AS 17 can access the looking glass of AS 29, which records all routes that AS 29 is selecting. AS 17 also knows AS 29 has **not** configured RPKI. Explain how AS 17 could check whether AS 23 has configured RPKI by hijacking `19.0.0.0/8` and sending it to AS 23. (3 Points)

Solution 1: If AS 23 has configured RPKI, it will immediately notice this route is faked and drop it. Otherwise, AS 23 will accept and prefer AS 17's route to the route it receives from AS 31. AS 23 will then advertise this route to AS 29. Therefore, the route with AS 17 included in the AS path will be present in AS 29's looking glass.

Solution 2: AS 17 waits and see whether AS 23 sends packets to reach AS 19 via AS 17.

(iii) AS 17 now also controls the entire AS 29. Assume **only** AS 19 has configured RPKI. Is it possible to hijack `19.0.0.0/8` from AS 17 or AS 29 (or both), such that all other ASes select the hijacked route, but AS 19 does not observe it? If your answer is yes, explain where (AS 17 or AS 29) you would announce `19.0.0.0/8` to which AS(es). If your answer is no, justify your answer. (3 Points)

Solution: Impossible. If AS 41 selects AS 17's route, AS 41 will always send it to AS 19 so that AS 19 notices. Otherwise, AS 41 does not select the bogus route.

(iv) Write down one countermeasure that AS 19 could take to attract back the traffic without cooperating with other ASes. (1 Point)

Solution: Announce more specific prefixes, e.g., `19.0.0.0/9` and `19.128.0.0/9`.

leave blank

d) IXP connection **(11 Points)**

Figure 14 shows a topology consisting of ASes and an IXP. The business relationships between existing ASes remain the same as in Figure 12. Lines with double arrows represent that two connected ASes are *peers* (AS 41 and AS 31 are peers). Lines with a single arrow point from a *provider* to a *customer* (AS 31 is a provider of AS 23). All ASes have correctly configured their routing policies to follow business relationships. No AS has configured any other routing policy. Each AS announces its own `/8` prefix to **all** BGP neighbors (AS X announces `X.0.0.0/8`). There is no BGP session between any AS and the IXP yet.

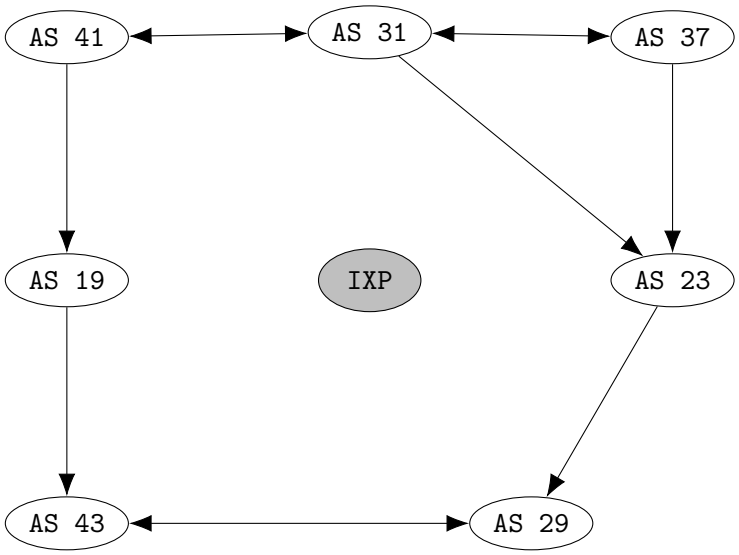


Figure 14: AS topology with the IXP

(i) Which AS(es) can reach all other ASes in Figure 14? (2 Points)
Solution: AS 31, AS 23 and AS 29.

To restore full AS-level connectivity, some ASes add a BGP session with the IXP. ASes treat the BGP session with an IXP as a *peer* session. When the IXP receives a route from some AS, it directly relays the route onto **all** other peering ASes without any modification (e.g., the IXP does **not** prepend its AS to the AS path).

(ii) What is the **minimum number** of IXP sessions that must be added such that every AS can learn the prefix of every other AS? Write down all of them. (2 Points)

Solution: 2. One with AS 41 and the other with AS 37.

After adding all IXP sessions, each AS can learn the prefix of **every** other AS. Table 1 lists **all** messages some AS X receives after adding all IXP sessions. Each message can be received from **any** BGP neighbor. A message with a lower index is received earlier.

Index	Message type	Prefix
1	`UPDATE`	`41.0.0.0/8`
2	`UPDATE`	`19.0.0.0/8`
3	`UPDATE`	`43.0.0.0/8`

Table 1: All BGP messages AS X receives after all IXP sessions are added

(iii) Could AS 31 be AS X? Justify your answer. (3 Points)

Solution 1: If students add IXP sessions with AS 31 and AS 41 (and potentially with AS 19 and AS 43, if assume the IXP combines multiple same `UPDATE`) but no other sessions in task (ii), then it is possible to receive the routes from the IXP.

Solution 2: If student answer task (ii) correctly, then it is impossible: AS 41 has advertised these routes before, and AS 37 des not update routes to AS 31.

(iv) Is it possible for **any** AS to receive an **explicit** `WITHDRAW` message after adding all IXP sessions? If your answer is yes, write down one possible case (which AS could receive a route withdrawal for which prefix from where). If your answer is no, justify your answer. Note that an `UPDATE` message indicating the replacement of an existing route with a better route is an implicit withdrawal, **not** an explicit one. (4 Points)

Solution: It is impossible. We prove it by contradiction.

Assume some AS Y is the first AS that sends a withdrawal message to AS Z for some prefix p after the IXP is turned on. This implies: *(i)* Y learns a route r for p before the IXP is on, and r can be advertised to Z; *(ii)* Y learns a new route r' for p after the IXP is on, and r' is not allowed to advertise to Z.

The two implications above further imply: *(iii)* Z is either a peer or a provider of Y; *(iv)* r was learnt from Y's customer, and r' is learnt from either a peer or provider.

The new implications above further imply that Y prefers r' (received from a peer or a provider) to r (received from a customer), which violates the business relationships.

Therefore, the assumption is contradicted.

leave blank

Task 4: Reliable transport **36 Points**

leave blank

a) Warm-Up **(5 Points)**

(i) Describe 3 scenarios in which the sender receives duplicate ACKs when using a transport protocol based on cumulative ACKs? (1 Point)

Solution: Packet loss (corruption), reordering and duplication

(ii) Explain how individual and cumulative ACKs can lead to unnecessary retransmissions. (2 Points)

Solution:

- *Individual ACKs*: Loss of an ACK,
- *Cumulative ACKs*: Reordering of ACKs

(iii) Explain one concrete problem that could happen if TCP would start with an initial sequence number of 0. Ignore any security-related concerns. (2 Points)

Solution: IP addresses and port numbers uniquely identify a connection. Port numbers eventually get reused. There is a small chance that old packets are still in flight. If the sequence number of the old packet fits into the receiver window of the new connection, the receiver may receive invalid data.

leave blank

b) Efficiency and Fairness of AIMD (12 Points)

In today's Internet, Congestion Control (CC) algorithms try to achieve a *fair* and *efficient* bandwidth utilization of links shared by multiple flows. The lecture introduced one example in the form of Additive-Increase Multiplicative-Decrease (AIMD) CC algorithms. As a reminder, AIMD is defined in the following way:

$$\texttt{cwnd}_{i+1} = \begin{cases} \texttt{cwnd}_i + \alpha & \text{if no congestion detected} \\ \texttt{cwnd}_i \ / \ \beta & \text{if congestion detected} \end{cases}$$

$$\text{with } \alpha > 0 \text{ and } \beta > 1$$

In the following tasks, you will analyze how the parameters α and β impact efficiency (throughput and packet loss) and fairness.

(i) How does increasing or decreasing the additive factor α affect the throughput and packet loss of AIMD if we keep β constant? (4 Points)

Solution:

- Increasing α will lead to a **steeper** ramp-up, and thus to a **higher oscillation frequency**. This higher frequency will **increase packet loss**.
- Decreasing α will lead to a less steep ramp-up, hence lower oscillation frequency and reduced packet loss.
- However, reducing α will lead to a slower ramp-up, which yields to **lower throughput for short-lived flows**.

(ii) How does increasing or decreasing the multiplicative factor β affect the throughput and packet loss of AIMD if we keep α constant? (4 Points)

Solution:

- Increasing β will **decrease throughput** as the sending rate is reduced significantly after congestion. The resulting ramp-up takes longer to reach a high throughput again. Packet loss also decreases, as congestion is reached less often due to slow wramp-up.
- Hence, decreasing β will **increase throughput** and packet loss.
- However, decreasing β significantly will cause massive packet drops, as soon as the decrease after detecting congestion is **not large enough to get out of congestion**.

(iii) Two independent flows with AIMD CC algorithms share the same link. Will they always converge to a fair bandwidth allocation if they use *different* AIMD parameters (i.e., α and β)? Justify your answer.

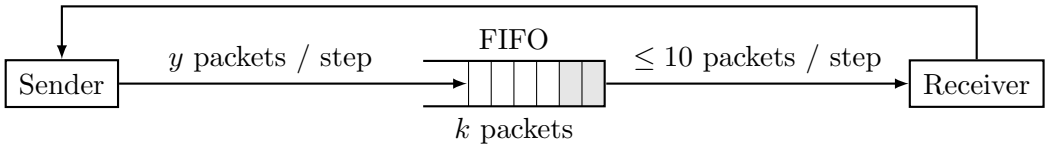
Hint: Argue from an initial state where both use the same bandwidth. (4 Points)

Solution: No! If one flow uses aggressive parameters (large α and small β), it will increase faster. Assume both start using half the available bandwidth. Then, due to congestion, the aggressive flow will decrease less than the other one. Further, the aggressive flow will increase faster. Next time we see congestion, the aggressive flow will have a much higher throughput. At some point, we will reach a fix-point, where the aggressive flow will get higher throughput on average.

leave blank

c) Network Queues **(19 Points)**

In this question, you will explore the impact of queues on congestion control algorithms. We will model the network as a discrete event system, consisting of a sender, a First-In First-Out (FIFO) queue, and a receiver:



The queue can transmit **at most 10 packets per step** to the receiver, and hold **at most k packets** (k will vary throughout this question). We assume that the queue's enqueue, dequeue, and drop operations happen in a given, fixed order. At each step t, the sender first generates $y[t]$ packets that are enqueued (at the tail). The queue then delivers at most 10 packets to the receiver (starting from the head). The queue finally drops packets from its tail until k packets (or fewer) remain. The sender uses an Additive-Increase Multiplicative-Decrease (AIMD) congestion control algorithm that generates $y[t]$ packets per step t:

$$y'[t+1] = \begin{cases} y[t]/2 & \textit{Loss:} \text{ if the queue dropped packets in step } t \\ y[t]/2 & \textit{Timeout:} \text{ if packets sent in step } t-1 \text{ are still enqueued after step } t \\ y[t]+1 & \text{Otherwise} \end{cases}$$

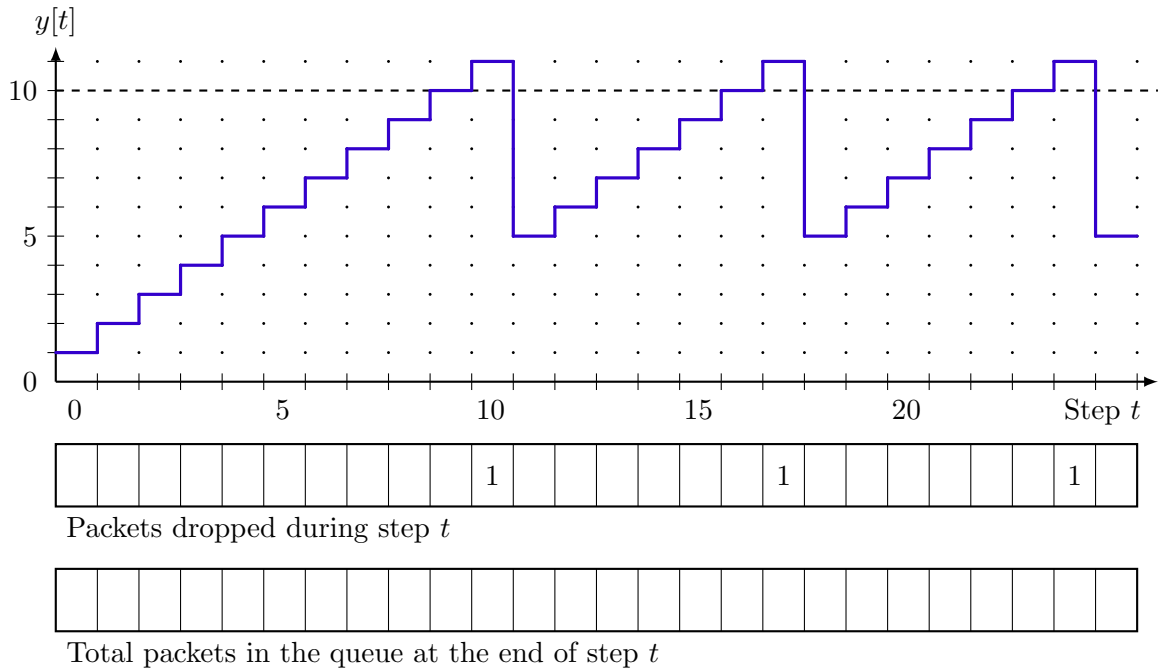
$$y[t+1] = \left\lfloor \max\left(y'[t+1], 1\right) \right\rfloor$$

The first equation describes the AIMD behavior with losses and timeouts, while the second equation describes that the sending rate y is always **rounded down** and **at least 1**.

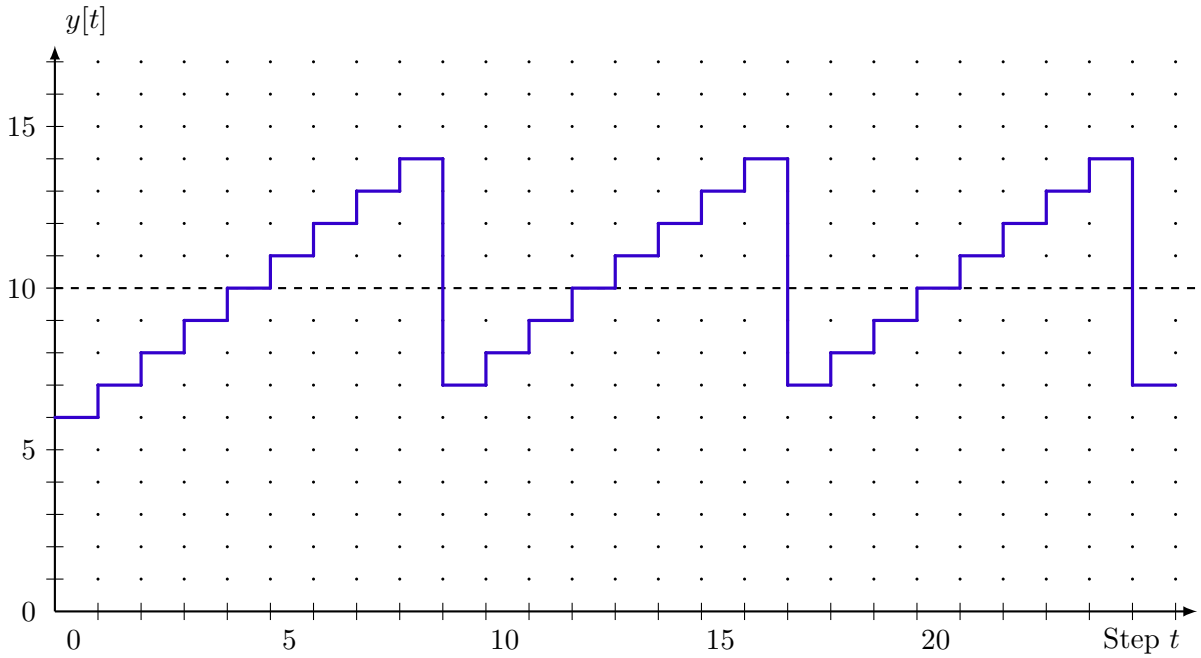
In this question, you will draw multiple throughput diagrams that consist of three parts:

1. The number of packets $y[t]$ sent by the sender at the beginning of step t.
2. The number of packets dropped during step t (empty means zero).
3. The total number of packets in the queue at the end of step t (empty means zero).

As an example, here is the corresponding throughput diagram for $k = 0$:



(i) Draw the throughput diagram for $k = 7$ and include the number of packets dropped and enqueued at each time step. *Hint:* Consider the area between $y[t]$ and 10. (9 Points)



								3							4							4	

Packets dropped during step t

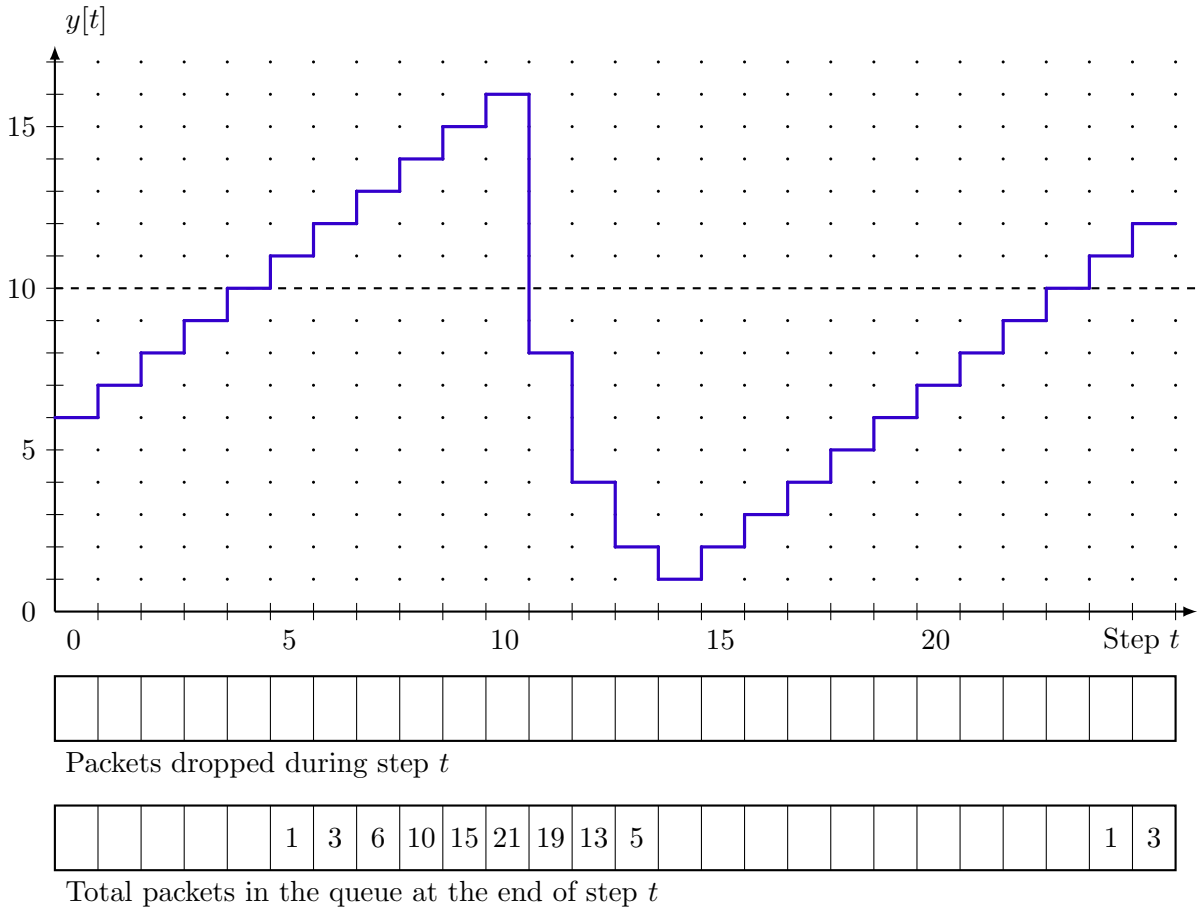
					1	3	6	7	4	2	1	1	2	4	7	7	4	2	1	1	2	4	7	7	4

Total packets in the queue at the end of step t

(ii) Assume the sender keeps sending forever. What is the average number of packets delivered per step for our network with $k = 7$? Justify. (2 Points)

Solution: The average throughput is **10 packets per step**, because the queue is never empty and will therefore always deliver 10 packets per step.

(iii) Draw the throughput diagram for $k \to \infty$ (i.e., the queue can hold an infinite number of packets) and include the number of packets dropped and enqueued at each time step.
(8 Points)



Packets dropped during step t

					1	3	6	10	15	21	19	13	5										1	3	

Total packets in the queue at the end of step t

leave blank

Task 5: Applications 22 Points

a) Warm-Up (10 Points)

leave blank

(i) Briefly explain how it is possible for the same domain name (e.g. `www.google.ch`) to be available on both IPv4 and IPv6. (2 Points)

Solution: It is possible as IPv4 and IPv6 rely on different DNS records: A for IPv4 and AAAA for IPv6. Depending on the DNS query, either the IPv4 or the IPv6 is returned.

(ii) Briefly describe one advantage and one disadvantage of using HTTP to stream videos. (2 Points)

Solution: Example advantage: easier deployment since all Internet-connected devices support HTTP. Example disadvantage: inefficient encoding of videos as HTTP is text-based, hence it forces video to be translated to text.

(iii) Many application-layer protocols rely on some form of caching to improve user performance. Briefly explain two distinct examples of caching. For each example, make sure to indicate: the protocol being sped up, why caching is used, where are the caches located and what do they store, and how they work. (6 Points)

Solution:

Example 1: HTTP

- Why: To allow clients to render webpages faster, to reduce the load on the servers/network.
- Where/What: The caches are located at/close to the client and/or close to the destination. The caches store the content itself (e.g., static images/files).
- How: The client asks a proxy web server (instead of the origin web server) for the content. If the proxy has the content cached and the content hasn't changed since it obtained it, then the proxy serves it to the client.

Example 2: DNS

- Why: To reduce resolution times of DNS mappings, to reduce the load on (authoritative) DNS servers and the network.
- Where/What: The caches are located at DNS servers/clients and store DNS records.
- How: They store answers to previous DNS requests for TTL time and serve those when asked again.

b) fun.comm-net! (6 Points)

leave blank

You just have been granted the right to create a new top-level domain name: `.comm-net`. Congratulations! Now comes the time to set it up though.

(i) To administer it, you decide to set up two name servers:

- `a.comm-net 192.0.2.1`
- `b.comm-net 192.0.2.2`

Indicate the resource records you would need to add to any DNS root server. For each record, indicate its corresponding name, type, and value. *Note that you might not need all 5 records.* (2 Points)

Solution:

```
comm-net. NS a.comm-net.
comm-net. NS b.comm-net.
a.comm-net. A 192.0.2.1
b.comm-net. A 192.0.2.2
```

(ii) You want to create `www.comm-net` as an alternative domain name for the course website which is currently hosted on `comm-net.ethz.ch`. Explain the resource record(s) you would add on `a.comm-net` to make `www.comm-net` an alias of `comm-net.ethz.ch`. For each record, indicate its corresponding name, type, and value. *Note that you might not need all 3 records.* (2 Points)

Solution: `www.comm-net. CNAME comm-net.ethz.ch`

(iii) Eager to be the very first one in the Internet to test your domain name, you make a typo writing the name in your browser and issue a request for `w.comm-net`. Facing the DNS error, you quickly realize your mistake and issue a second (correct) request for `www.comm-net`.

Briefly describe the content requests seen **by the DNS root server** (if any) after each of your request. (2 Points)

Solution: Root server sees *one* request for `.comm-net` when `w.comm-net` is issued, and none upon the second request because the answer is cached.

leave blank

c) HTTP retrieving times **(6 Points)**

Consider the round trip time between a web browser and an HTTP server to be 10 ms. The HTTP server stores a web page composed of a base HTML file and 9 small pictures. Compute the time (in ms) the browser takes to retrieve the entire web page considering different flavors of HTTP. You can assume that the web browser already knows the IP address of the server (that is, no DNS lookup is necessary) and that the transmission time of each object is negligible.

(i) Compute the time needed considering HTTP 1.0 (non-persistent) is used with no parallel connection. Briefly explain. (2 Points)

Solution: 2 * RTT (to get the base page) + 2 * 9 * RTT (to get each picture in sequence) = 20 * RTT = 200 ms

(ii) Compute the time needed considering HTTP 1.0 (non-persistent) is used with 10 parallel connections. Briefly explain. (2 Points)

Solution: 2 * RTT (to get the base page) + 2 * RTT (to get all the pictures in parallel) = 4 * RTT = 40 ms

(iii) Compute the time needed considering HTTP 1.1 (persistent) is used with support for pipelining. Briefly explain. (2 Points)

Solution: 2 * RTT (to get the base page) + 1 * RTT (to get all the pictures) = 3 * RTT = 30 ms

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: _____

Extra Sheet 3

Task: _____

Task: _____