ETH zürich

Institut für Technische Informatik und Kommunikationsnetze

Department ITET
August 2024

Prof. Dr. Laurent Vanbever
L. Brandner, Y. Chen, A. Dietmüller, G. Fragkouli, L. Röllin, T. Schneider

Exam: Communication Networks

6 August 2024, 9:00–11:30, Room HIL F 41

Sample Solution

General remarks:

- ▷ 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 **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 – 35**).
- ▷ 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.

Student legi nr.:

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

Task	Points
Ethernet & IP	/30
Intra-domain routing	/28
Inter-domain routing	/38
Reliable transport	/35
Applications	/19
Total	/150

leave blank

leave blank

Task 1: Ethernet & IP — 30 Points

a) Warm-Up — (6 Points)

(i) Name the type of links where collisions can happen even when two directly-connected hosts communicate with each other. (1 Point)

Solution: Half-duplex links.

(ii) Name a mechanism to mitigate forwarding loops present at the **network layer**. (1 Point)

Solution: Decrementing TTL and discarding packets with TTL=0.

(iii) Why would it be interesting for a layer-2 switch to also have a MAC address? (1 Point)

Solution: It may be used as the switch ID in STP, or for accessing/managing the switch.

(iv) Briefly explain why switches do **exact** (MAC) lookups instead of the **longest-prefix-match** (IP) lookups that routers do. (1 Point)

Solution: In MACs, the prefix indicates the vendor (and not the location where the MAC resides).

(v) State one disadvantage of allowing access links to carry VLAN identifiers. (1 Point)

Solution: Security issue, as a host in the "student" VLAN could broadcast to the "professor" VLAN.

(vi) Can we always remove an entry from a routing table that refers to a sub-prefix of a prefix that exists in the table without changing the forwarding? Briefly explain. (1 Point)

Solution: No, as the sub-prefix may be forwarded to an interface different from the one of the prefix.

leave blank

b) MACs and IPs **(8 Points)**

Consider two networks (Network 1 and 2) connected through the Internet (Figure 1):

- S_1, S_2, and S_3 are layer-2 switches;
- R_1 and R_2 are layer-3 routers;
- S_2 is connected to a DHCP server;
- H_x are hosts where x is the host number: Network 1 has 2 hosts, H_1 and H_2; Network 2 has 1000 hosts, H_3, H_4, ..., H_{1002} (not all of them are shown);
- layer-2 switches do not have any MACs or IPs, while each of the other devices has one MAC and one IP address for each network they are part of.

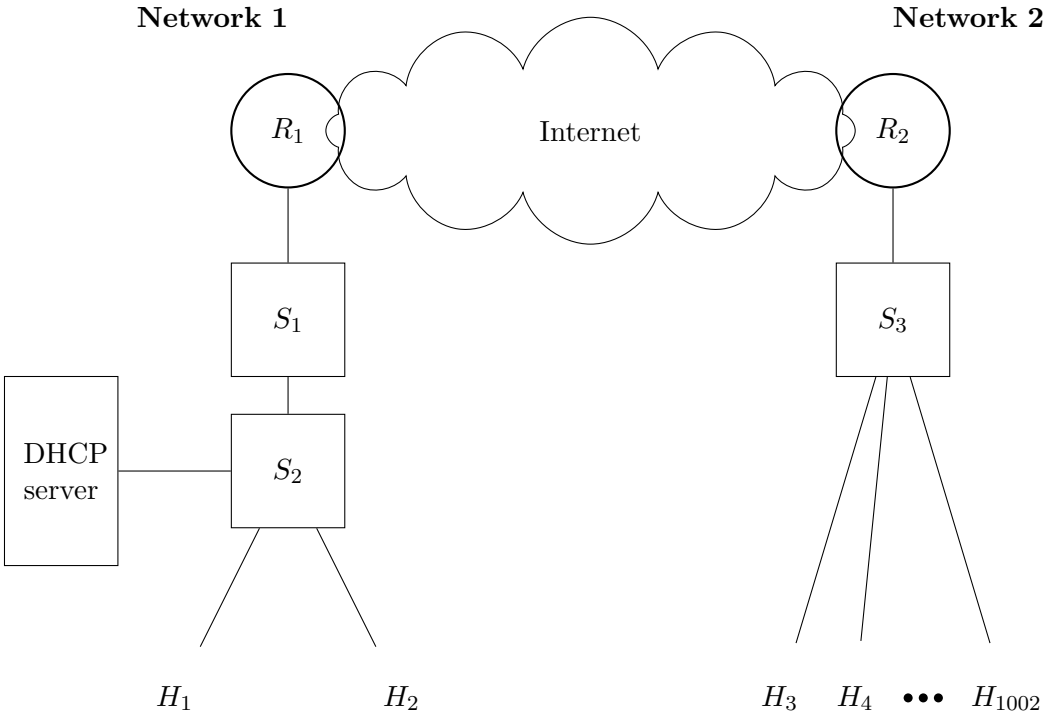


Figure 1: Two networks connected through the Internet.

(i) Assume that the devices in Network 1 communicate only with each other.
If you monitor *all* traffic in Network 1, you will observe at most _____ distinct MACs, and at most _____ distinct IPs. **Fill in the blanks.** (2 Points)

Solution: At most 5 distinct MACs, and at most 5 distinct IPs.

(ii) Assume that prefix p_1 is assigned to Network 1, and prefix p_2 is assigned to Network 2 such that:

- p_1 and p_2 belong to prefix $p =$ `125.0.0.0/8`;
- the first addresses from p are assigned to Network 1;
- each of p_1 and p_2 has the smallest possible size to accommodate all the hosts;
- the last address of p_1 is as consecutive as possible with the first address of p_2.

Then, $p_1 =$_____ and $p_2 =$_____
Fill in the blanks. (2 Points)

Solution: p1=125.0.0.0/29; p2=125.0.4.0/22

(iii) Assume that all switch forwarding tables and all ARP tables are empty. Only the DHCP server and H_2 already know/have learned their IP addresses (H_1 does not). Devices in Network 1 have the following MAC addresses:

- H_1 has MAC a;
- H_2 has MAC b;
- the DHCP server has MAC c;
- R_1 has MAC d.

H_1 knows H_2's IP and H_1 wants to send a packet to H_2.

State all the frames/packets that S_1 and S_2 observe in **ascending chronological order**, respectively, up until H_2 receives H_1's packet. Answer by filling in Table 1 for S_1 and Table 2 for S_2. As an example for the "Protocol & Purpose" column, if a table row indicates a DNS request for resolving `fun.nsg.ee.ethz.ch`, its "Protocol & Purpose" would be "DNS request for `fun.nsg.ee.ethz.ch`'s IP." (You may not need all the table entries.) (4 Points)

Solution:

#	MAC		Protocol & Purpose
	src	**dst**	
1	a	broadcast	DHCP discovery of an IP address
2	c	broadcast	DHCP offer of an IP address
3	a	broadcast	ARP request for H2's MAC
4			
5			
6			

Table 1: Frames/packets observed by **S1**.

#	MAC		Protocol & Purpose
	src	**dst**	
1	a	broadcast	DHCP discovery of an IP address
2	c	a OR broadcast	DHCP offer of an IP address
3	a	broadcast	ARP request for H2's MAC
4	b	a	ARP response associating H2's IP to H2's MAC
5	a	b	H1's packet to H2
6			

Table 2: Frames/packets observed by **S2**.

leave blank

c) Spanning Tree Protocol **(16 Points)**

Consider the topology in Figure 2. Squares represent layer-2 switches (S_X indicates a switch with ID X), and circles represent layer-3 routers (R_Y indicates a router with ID Y). Upon running the Spanning Tree Protocol (STP), if there exist multiple shortest paths to the root node, a node picks the next hop which has the lowest node ID.

Throughout this task, assume that links don't fail, there are *no* VLANs, and that you *cannot* change the topology, switch IDs, or STP, unless explicitly stated.

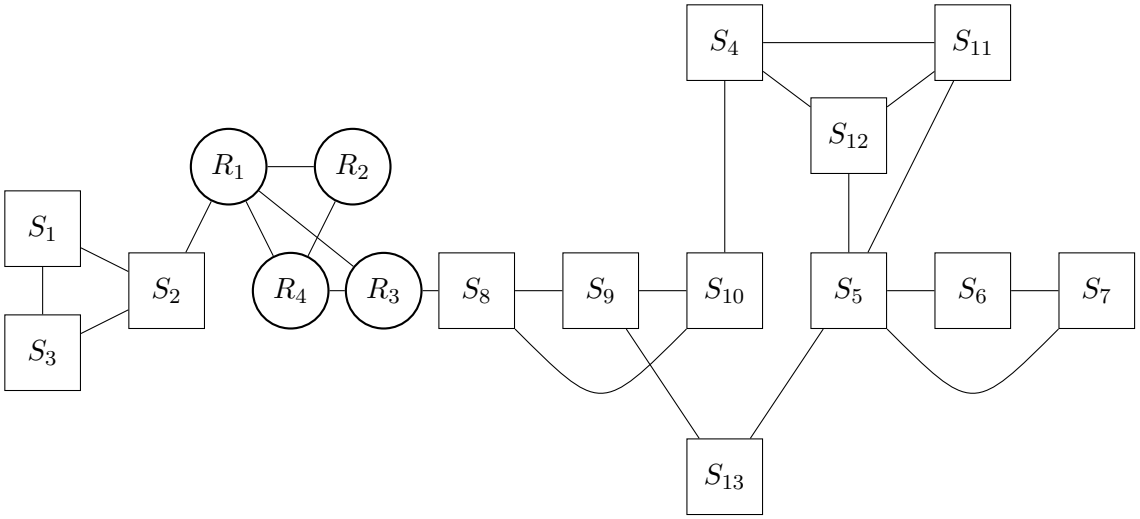


Figure 2: Topology of layer-2 switches (squares) and layer-3 routers (circles). (You can find a copy for taking notes on page 30.)

(i) Indicate below which links are **de-**activated after running the STP in Figure 2. For example, write "S_X – S_Y" to indicate that the link between S_X and S_Y is de-activated. Indicate each de-activated link on a different line. (You may not need all the lines.) (4 Points)

Solution: De-activated links: S2 – S3, S6 – S7, S5 – S12, S11 – S12, S8 – S9, S9 – S13.

(ii) Hosts H_7 and H_8 (not shown in the figure) are attached to S_7 and S_8, respectively, and frequently communicate with each other.

Is it possible to make the number of *switches* between H_7 and H_8 equal to 5 after **removing a single link** from Figure 2 and re-applying STP? If yes, state which link should be removed and the path that will be used between H_7 and H_8. If not, briefly explain why. (3 Points)

Solution: Yes, by removing link S_4 – S_{10}, hosts H_7 and H_8 communicate through five switches (S_8 – S_9 – S_{13} – S_5 – S_7).

(iii) Consider now only the right-hand side of the topology depicted in Figure 2 (switches S_8, S_9, …, S_{13}). Assume that an attacker can take over *one* switch and modify the fields of the BDPU messages sent by the switch. (The attacker can only modify the originated BPDU, it cannot drop any traffic.) The attacker's goal is to maximize the number of switches that cannot reach S4 after re-applying STP.

Which switch should the attacker take over, how should she attack, and why does that maximize the number of switches that cannot reach S4? (4 Points)

Solution: The attacker needs to create two disconnected spanning trees, one rooted at S_4 and the other at the switch the attacker takes over, such that the attacker's spanning tree has the maximum possible number of switches (those switches cannot communicate with S_4). One solution is for the attacker to take over S_{13}, then switches S_8, S_9, S_{13}, S_5, S_6, S_7 are part of the attacker's spanning tree.

(iv) Assume now that there can exist multiple VLANs (the spanning tree that you computed in subtask i) does *not* need to be one of the spanning trees of those VLANs). You can adapt the switch IDs for each VLAN.

State the **minimum** number of VLANs that we need for all links to remain active after applying STP in Figure 2 and briefly explain why. What is the root of the spanning tree in each of those VLANs? (5 Points)

Solution: We need at least four VLANs: two on the LAN on the left-hand side (e.g., one with root S_1, and one with root S_2), and two on the LAN on the right-hand side (one with root S_6 or S_7, and one with root S_{12}).

leave blank

leave blank

Task 2: Intra-domain routing **28 Points**

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

(i) You set the weights of your links to be proportional to their **propagation delay**. Is it guaranteed that the packets following the shortest path will arrive earlier than packets using any other path? Why or why not? (1 Point)

Solution: No it isn't guaranteed since there is also processing and queueing delay. A path with less hops but overall higher propagation delay can still be faster due to other delays e.g. queueing. Link failure on shortest path is also considered an acceptable solution.

(ii) How can limiting the maximum routing weight in distance vector protocols improve convergence time? (1 Point)

Solution: Distance vector suffers from the "count to infinity problem" in which networks converge slowly because they update the weight in small increments until they reach the weight that was set. By limiting the maximum weight they reach this new weight faster.

(iii) Why link state protocols do not work well in large networks such as large-scale data center networks? (1 Point)

Solution: The reason for this is that link state protocols need to send the link state to all other nodes. So if there are lots of links and nodes the size and number of messages get larger and larger. Second reason is that the shortest path calculations are taking longer for bigger topologies. Size of state that each node needs to keep of the network topology is also considered a valid reason.

(iv) Consider a network with non-negative link weights. Consider two scenarios: (i) Add 1 to each link weight; or (ii) multiply each weight by 2. Explain which scenario does *not* guarantee that the shortest paths remain unchanged. Provide an example. (2 Points)

Solution: By adding +1 to each link we increase the weight of paths with more hops more. In turn favouring paths with fewer hops thus creating different shortest paths.

leave blank

b) Load Balancing and Traceroute **(9 Points)**

(i) Congratulations, you just got hired as a network engineer! Your first task is to load balance traffic in the network shown below using Equal Cost Multipath (ECMP). The list of nodes and the corresponding paths you should load balance the traffic on can be found in Table 3. Write down the four missing link weights that accomplish the load balancing in Figure 3 (one weight per highlighted box). Recall that ECMP makes routers load balance traffic on all the shortest-paths towards a destination node. (6 Points)

Nodes:	Path 1	Path 2
(A, E)	A-B-C-D-E	A-B-F-G-D-E
(G, H)	G-H	G-F-H
(C, F)	C-B-F	C-D-G-F
(B, H)	B-H	B-F-H

Table 3: Paths to load balance

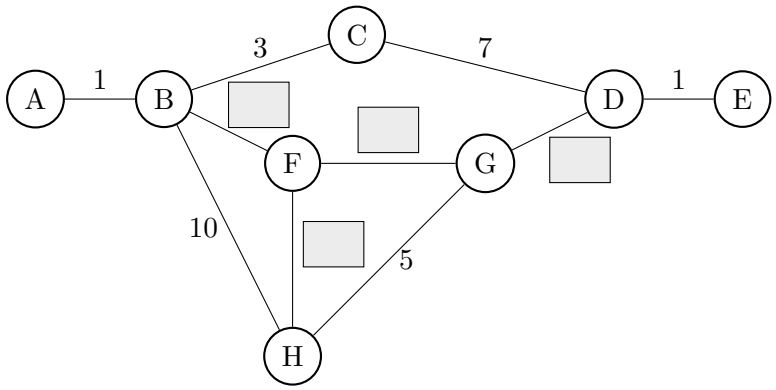


Figure 3: The network to load balance. (You can find a copy for taking notes on page 31.)

Solution: Solution: Figure 4

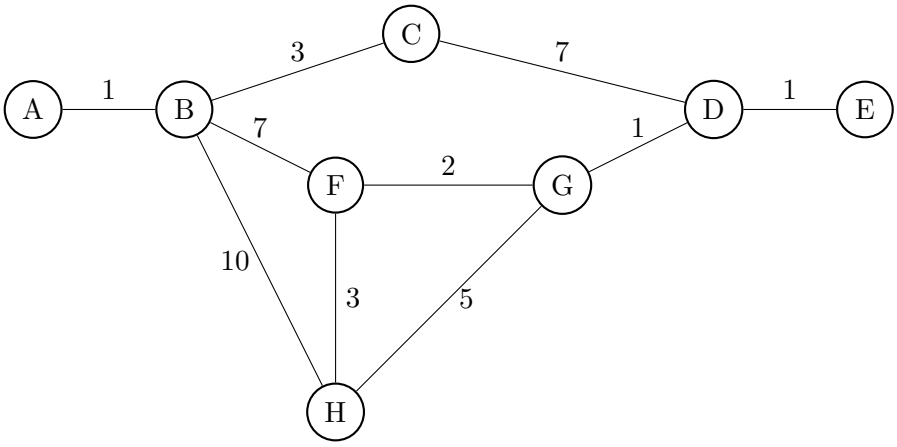


Figure 4: The network in need of load balancing.

(ii) You now try to see if the load balancing worked and run a traceroute from A to E. Assume that the IP address of node A is `a.a.a.a`, node B is `b.b.b.b`, etc. and that the RTT increases by 10 ms for each hop. Briefly explain how the following traceroute output where some nodes show up on multiple lines with different RTTs can happen.

(2 Points)

```
traceroute to E (e.e.e.e), 30 hops max, 60 byte packets
1  B (b.b.b.b) 10ms 10ms 10ms
2  C (c.c.c.c) 20ms F (f.f.f.f) 20ms C (c.c.c.c) 20ms
3  D (d.d.d.d) 30ms 30ms G (g.g.g.g) 30ms
4  E (e.e.e.e) 40ms D (d.d.d.d) 40ms 40ms
```

Solution: The reason this output is possible because of the way traceroute works. It sends out individual packets with different TTLs so every response can take a different path. This is the reason that both C and F aswell as D and G can show up in the same line.

(iii) Considering the traceroute output from above: are the correct two paths between A and E used? Explain your answer. (1 Point)

Solution: The traceroute output shows that the correct two paths are used. The first set of packets traverse path A-B-C-D-E while the other two sets traverse a mix of A-B-C-D-E and A-B-F-G-D-E.

c) Reverse Dijkstra (14 Points)

leave blank

You are looking at a drawing of your network (Figure 5) but it clearly seems some links are missing. Table 4 shows the *partial* output of the Dijkstra's algorithm (some cells are empty) as performed in the lecture starting from the node **U**. For each iteration, the table indicates the weights of the shortest paths found so far. If after one iteration there are multiple nodes with an equally-shortest path, the algorithm continues with the node which comes first in the alphabet. There is at most one link between two nodes and each link has a strictly positive weight $[1, \infty]$.

You feel confident that, with the information from Figure 5 and the partial output from Dijkstra's algorithm in Table 4, you can find some of those missing links and link weights.

(i) First, fill in the highlighted cells in Table 4 using the information available in Figure 5. Write your answers directly in Table 4. (3 Points)

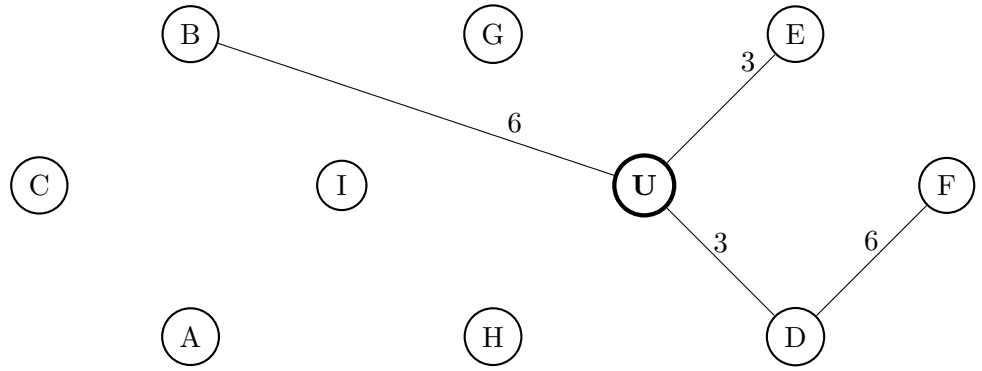


Figure 5: A network consisting of 10 nodes with missing links and link weights. (You can find a copy for taking notes on page 32.)

Solution: Table 5

(ii) Now that you have completed Table 4 you can use this information to piece together the missing links in Figure 5. Draw all the links and corresponding weights you can identify directly in Figure 5. (6 Points)

Solution: Figure 6

(iii) You now wonder whether you actually found all the links or not. For all the following links, mark if they could exist or not. (5 Points)

Use the following format:

- If the link could exist, indicate the range of weights this link is allowed to have;
- If the link could not exist, write down the iteration in which the link would have showed up.

#	**U**	A	B	C	D	E	F	G	H	I
1	0	1		∞			∞	∞	∞	∞
2	0	1	5	2	3	3	∞	∞	∞	∞
3	0	1	4	2	3	3	∞	∞	∞	∞
4	0	1	4	2	3	3		∞	8	∞
5	0	1	4	2	3	3	5	∞	8	∞
6	0	1	4	2	3	3	5	10	8	∞
7	0	1	4	2	3	3	5	9	8	∞
8	0	1	4	2	3	3	5	9	8	13
9	0	1	4	2	3	3	5	9	8	12
10	0	1	4	2	3	3	5	9	8	12

Table 4: For each iteration (1 to 10) the table shows the shortest path found by Dijkstra's algorithm performed on node **U** towards all other nodes.

#	**U**	A	B	C	D	E	F	G	H	I
1	0	1	**6**	∞	**3**	**3**	∞	∞	∞	∞
2	0	1	5	2	3	3	∞	∞	∞	∞
3	0	1	4	2	3	3	∞	∞	∞	∞
4	0	1	4	2	3	3	**9**	∞	8	∞
5	0	1	4	2	3	3	5	∞	8	∞
6	0	1	4	2	3	3	5	10	8	∞
7	0	1	4	2	3	3	5	9	8	∞
8	0	1	4	2	3	3	5	9	8	13
9	0	1	4	2	3	3	5	9	8	12
10	0	1	4	2	3	3	5	9	8	12

Table 5: Completed output of Dijkstra's algorithm with the help of the links in the graph.

Solution:

Link `U-C`: cannot exist, since C would need to show up in the first iteration.
Link `B-E`: can exist, weight $[1, \infty]$

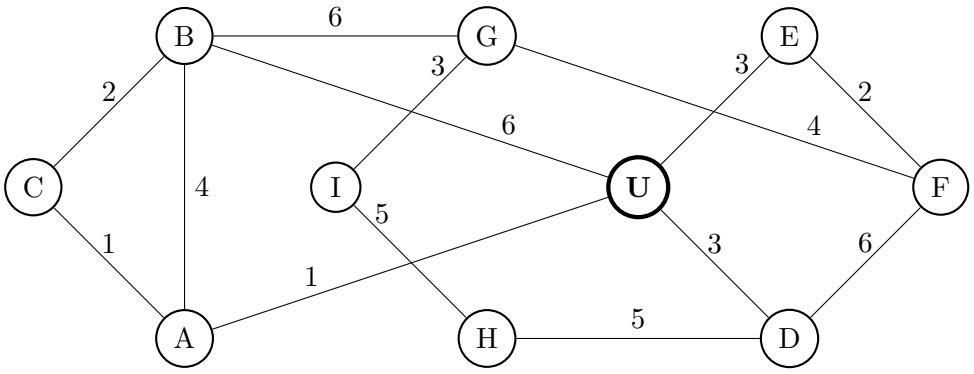


Figure 6: The completed view of the original network.

Link `U-I`: cannot exist, since I would need to show up in the first iteration.
Link `A-D`: can exist, weight $[2, \infty]$
Link `E-G`: cannot exist, since G would need to show up in the 5. iteration.

leave blank

leave blank

Task 3: Inter-domain routing **38 Points**

a) BGP 201 **(10 Points)**

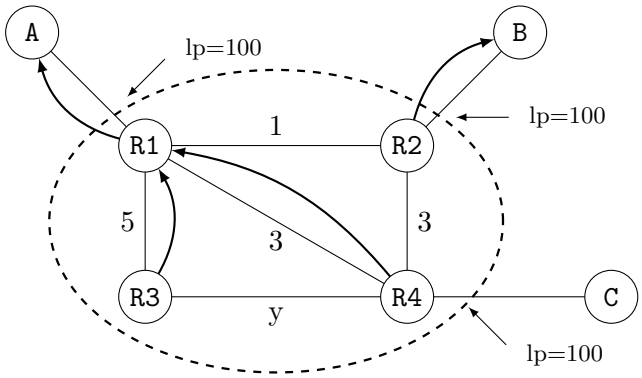


Figure 7: A BGP network of 4 internal routers and 3 external neighbors.

Figure 7 shows a BGP network with 4 internal routers and 3 external neighbors. Each IGP link between two internal routers has a weight. The network forms an iBGP **full-mesh** (not shown in the figure). Each external neighbor has an eBGP session with one border router. Each border router sets the local preference (lp) to 100 for any BGP route received from its external neighbor. The network does not have any other BGP policy configured.

`A`, `B`, `C` announce the same prefix p to the network. The announcements are such that `R1`, `R3`, and `R4` select the route from `A`, while `R2` selects the route from `B`. (The thick arrows in the figure represent the current forwarding paths each internal router takes to reach p.)

(i) Let $len(x)$ be the AS path length of the BGP route x announces to the network. Given the forwarding paths depicted above, briefly explain whether each of the following relation is possible or not. (2 Points)

Solution:

$len(A) > len(B)$: Impossible. In the BGP decision process, when two routes have the same local preference, the route with the shorter AS path length is preferred. Therefore, if $len(A) > len(B)$, both `R1` and `R2` will prefer `B`'s route.

$len(B) = len(C)$: Impossible. If $len(C) = len(A)$, `R4` would select `C`'s route. If $len(C) < len(A)$, since $len(A) = len(B)$ must be the case, all routers would prefer `C`'s route. If $len(C) > len(A)$, all routers would prefer `A`'s route.

One can also argue that `A` and `C` belong to the same AS and `A` sends a lower MED, in this case $len(A) = len(C)$ is possible.

(ii) Let y be the IGP weight of link `R3` – `R4`. Given the forwarding paths shown in the figure, what is the complete range of possible values for y? (1 Point)

Solution: $y >= 2$.

If $y < 2$, the IGP path `R3` $\rightarrow$ `R4` $\rightarrow$ `R2` has a cost lower than the current path `R3` $\rightarrow$ `R1`.

If $y = 2$, since `R4` prefers `R1` when both `R1` and `R2` announce to it a route with the same IGP cost, `R3` will also prefers `R1` even when two IGP paths have the same cost.

(iii) Assume A and B belong to the same AS X. How can AS X attract traffic to p via B rather than via A? List two distinct approaches. (2 Points)

Solution:

- Approach 1: AS X can set a lower MED on B's route.
- Approach 2: AS X can prepend AS path on A's route.
- Approach 3: AS X can announce more specific prefixes via B.
- Approach 4: AS X can fail the link between R1 and A.
- Approach 5: AS X can perform As poisoning to R1.

(iv) Assume R4 increases the local preference of the routes learned on session R4 – C to 200. Select the BGP message type that will be seen on each of the following BGP session. Select N/A if the BGP session will not see any message following the change.

(2 Points)

Solution:

	UPDATE	WITHDRAW	N/A
R1 → A:	■	■	■
R1 → R2:	□	■	□
R3 → R4:	□	□	■
R4 → C:	■	■	■

(v) Assuming equal local preferences again (100 for all learned routes), we consider the case where A withdraws its route to p. Briefly explain whether a blackhole can occur for p during the first and the last BGP message the network will see for this event. Recall that a blackhole occurs if a router receives a packet but does not know how to forward it. (3 Points)

Solution: No packet loss will occur.

R1 also knows a backup route to reach p via R2. Therefore, when R1 receives the withdrawal message from A, R1 will start forwarding packets to R2.

Before R1 sends out the withdrawal message to R3 and R4, these two routers continue forwarding packets to R1 and R1 forwards them to R2.

Once R4 receives R1's withdrawal message, it will start forwarding packets to R2. R3 still forwards packets to R1 after it receives the withdrawal message.

Another possible full-point solution is a packet loss can occur between the gap when R1 has withdrawn the route but has not switched to the backup route.

b) Mini Mini-Internet **(18 Points)**

leave blank

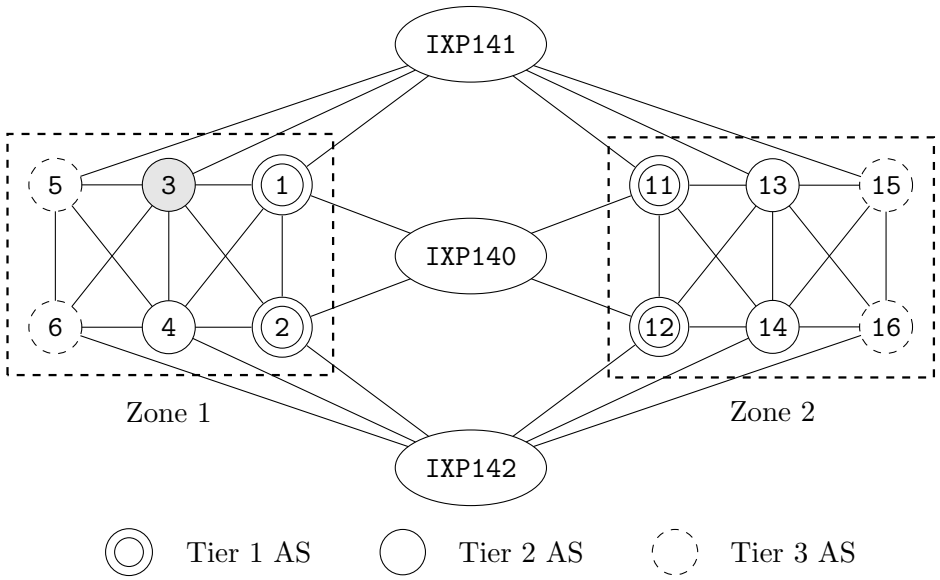


Figure 8: A mini-mini-internet with 2 zones and 3 IXPs.

Figure 8 shows a "mini mini-Internet" with 2 zones and 3 IXPs. Each zone contains 6 ASes acting at different tiers. Within each zone, each AS belonging to tier k is a provider of all the tier $k+1$ ASes, a customer of all the tier $k-1$ ASes, and a peer of all the same-tier ASes. For instance, AS 3 (a tier-2 AS) is a provider of AS 5 and AS 6, a customer of AS 1 and AS 2, and a peer of AS 4; AS 5 has no customer; and AS 1 has no provider. Each AS also maintains a **peer session** with at least one IXP, as drawn in the figure.

Each AS X announces its prefix `X.0.0.0/8` onto **all** connected sessions. Each IXP announces **any** prefix it receives from one zone to the **other** zone. For instance, IXP 141 announces any prefix its receives from AS 5 to AS 11, 13 and 15. An IXP does **not** append its number to the AS path when announcing prefixes.

All ASes strictly follow the classical business relationships as seen in the lecture defined over providers, peers and customers. Note that an IXP is also a **peer**. When two routes are equally preferred, each AS selects the route with the **smallest** next-AS in the AS path (note that IXPs do not appear in the AS path) as a final tie-breaker.

(i) List the AS path that AS 3 selects to reach the following 3 prefixes. (3 Points)

Solution:
`11.0.0.0/8`: [3, 11]
`12.0.0.0/8`: [3, 1, 12]
`14.0.0.0/8`: [3, 11, 14]

(ii) Which IXP is the most important to maintain the connectivity between the two zones? Justify your answer. (3 Points)

Solution: The most important IXP: `IXP 140`.

Justification: There is a full connectivity between the two zones even if there is only IXP 140. When IXP 140 is down, each tier 1 AS cannot be fully reached by the other zone even if all other IXPs are up. E.g., AS 12 cannot reach `1.0.0.0/8`.

(iii) Assume now that some ASes violate business relationships. Table 6 lists the AS paths for **all** the routes that some border router in AS 3 receives for `16.0.0.0/8`. The symbol `>` indicates the route that the router selects as best.

Based on these received routes, list 3 ASes that do not strictly follow business relationships and justify how each AS selects or exports routes incorrectly. (4 Points)

	Network	AS Path
*>	16.0.0.0/8	1 4 6 16
*		2 4 6 16
*		6 16
*		11 13 16

Table 6: All routes AS 3 receives for `16.0.0.0/8`.

Solution:

- Violating AS: 3
  Justification: When AS 3 receives the route from both AS 6 and AS 1, it selects the route from AS 1, which violates the business relationship that AS 3 should prefer the route from its customer to its provider.
- Violating AS: 6
  Justification: AS 6 should not have announced the route it receives from IXP 142 to its providers AS 3 and AS 4.
- Violating AS: 4
  Justification: AS 4 should have announced the route it receives from AS 6 to AS 3.
- Violating AS: 13
  Justification: AS 13 does not announce `16.0.0.0/8` received from AS 16 to IXP 141.

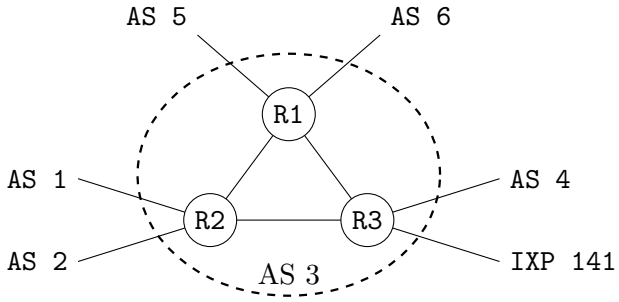


Figure 9: The internal topology of AS 3.

We now consider again the case in which all ASes in Figure 8 properly follow the classical business relationships. Figure 9 shows the internal topology of AS 3 and its external connections. AS 3 runs an iBGP full-mesh.

(iv) Assume both `R2` and `R3` can reach `14.0.0.0/8`, but `R1` cannot reach it. What could be the reason for this issue and how can you confirm it. (2 Points)

Solution: Reason: the iBGP session between `R1` and `R2` or between `R1` and `R3` is not properly configured, e.g., missing `next-hop-self`, or no session set up at all.

How to confirm: Check the iBGP configuration and link status connected to `R1`, and check the session configuration at `R2` and `R3`.

(v) Assume no router in AS 3 can ping `14.0.0.0/8`. You contacted the operator of AS 14 and she confirmed that AS 14 has received your packets. What could be the reason for this issue and how can you confirm it. (3 Points)

Solution: Reason 1: The best routing path AS 3 and AS 14 should select to reach each other is asymmetric, and there could be some gray failure occurring on the return path from AS 14 to AS 3 (AS 14 $\rightarrow$ AS 11 $\rightarrow$ AS 12 $\rightarrow$ AS 3), i.e., link congestion or temporary link failure.

Reason 2: Some AS hijacks AS 3's route.

How to confirm: AS 3 should first make sure AS 14 learns `3.0.0.0/8`, if so AS 14 could temporarily switch to a sub-best route to see if the packets can reach AS 3.

(vi) Assume `R2` accidentally stops advertising `3.0.0.0/8` to AS 1 and AS 2. Can AS 1 and AS 2 still reach `3.0.0.0/8`? Justify your answer.

Hint: You do not necessarily need to consider all possible routing paths. (3 Points)

Solution: Can AS 1 or AS 2 reach it: None of them can reach it.

Justification: We show by contradiction. Assume AS 1 learns the route for `3.0.0.0/8` from AS X, then AS X is either a peer or a customer of AS 1 since AS 1 is a tier 1 AS.

In either case, AS X only announces this route when itself receives it recursively from a customer or AS 3 itself (i.e., the base case) since all ASes follow business relationships.

However, it contradicts the fact that AS 3 does not announce the route to either provider. The same argument applies to AS 2.

c) Amongst us **(10 Points)**

leave blank

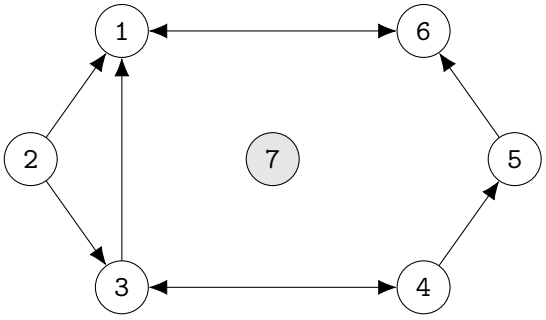


Figure 10: An internet of 7 ASes.

Figure 10 depicts an Internet of 7 ASes. Each single-headed arrow points **from** a provider **to** a customer (AS 3 is a provider of AS 1), each double-headed arrow points between two peers (AS 3 is a peer of AS 4).

All ASes *except AS 7* (highlighted) strictly follow the classical business relationships defined over providers, peers and customers.

Each AS X announces its prefix `X.0.0.0/8` to **all** connected sessions. Each AS has registered and published a Route Origin Authorization (ROA) for its own prefix in the Resource Public Key Infrastructure (RPKI) repository. **Each AS filters any invalid route using RPKI**.

(i) Describe one generic attack that an AS using RPKI-based filtering can defend against, and one attack that it cannot defend against. (2 Points)

Solution:

Defend against: the AS can defend against prefix hijacking attacks for the prefixes registered in the ROA.

Cannot defend against: the AS cannot defend against prefix hijacking attacks for the prefixes not registered in the ROA; or the AS cannot defend against BGP route manipulation attacks, e.g., AS path poisoning.

(ii) Assume AS 7 can establish one or more BGP sessions of any type with any AS and consistently announce a route for `7.0.0.0/8` on the established session(s). Is it possible for AS 7 to trigger a BGP oscillation for `7.0.0.0/8` amongst AS 1, AS 2, and AS 3, i.e., that **none** of AS 1, AS 2 or AS 3 can converge to a stable state? Justify your answer. (3 Points)

Solution: Not possible.

Justification: The Gao-Rexford model guarantees the convergence of the BGP. In this example, An oscillation occurs when each AS prefers the route announced by a counterclockwise neighbor over the direct route, which is not possible since the business relationship guarantees the valley-free property.

(iii) Assume AS 7 wants to hijack the traffic for `3.0.0.0/8`. To do that, AS 7 needs to establish BGP sessions with other ASes and to announce routes. (5 Points)

What are **all** possible ASes that AS 7 can hijack traffic from without alarming AS 3? An AS is alarmed if it receives **any** route (valid or invalid) for its own prefix.

Solution:

ASes that AS 7 can hijack traffic from: AS 1, AS 5 and AS 6.

AS 7 cannot hijack traffic from AS 2 since AS 2 always prefers the route from AS 3.

AS 7 cannot hijack traffic from AS 4, because if AS 4 selects the fake route from AS 5, AS 4 will alarm AS 3. If AS 4 selects the route from AS 7, then either AS 7 is a customer of AS 4, in which case AS 4 will alarm AS 3, or AS 7 is a peer of AS 4, in which case AS 4 will not prefer the route from AS 3 for short AS path.

If the answer assumes AS 7 can advertise more specific prefix, then AS 4 can also be hijacked.

What is the **minimum** number of BGP sessions that AS 7 needs to establish to hijack all this traffic? For each BGP session AS 7 establishes: identify the session type (provider-to-customer, customer-to-provider or peer-to-peer) and specify the AS path it announces on that session.

Hint: You do not necessarily need to fill all the sessions.

Solution:

If the answer assumes AS 7 can only announce `3.0.0.0/8`, then the solution is following:

Session 1: AS 6 $\rightarrow$ AS 7

AS path announced to Session 1: [3, 4, 7]

If the answer assumes AS 7 can announce more specific prefixes, then the solution is following:

Session 1: AS 1 peer with AS 7

AS path announced to Session 1: [3, 7]

Session 2: AS 1 peer with AS 4 or is a provider of AS 4.

AS path announced to Session 2: [3, 7]

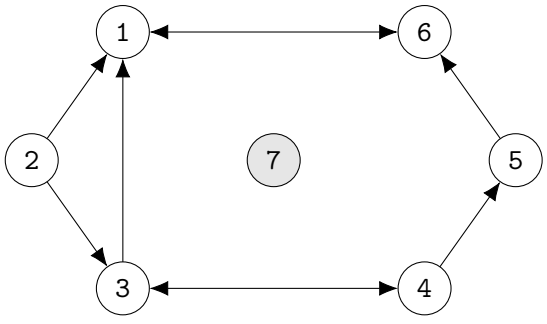


Figure 11: Copy of Figure 10.

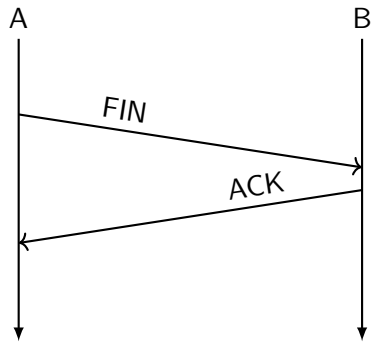
leave blank

leave blank

Task 4: Reliable transport **35 Points**

a) Warm-Up **(6 Points)**

(i) Consider a TCP connection between host A and B. Give a minimal sequence of packets such that host A considers the connection *half-closed*. Draw the packet exchange in the diagram below. (1 Point)



(ii) Consider a TCP implementation that immediately removes a socket after having sent the final ACK. Briefly describe two problems that can happen. (2 Points)

Solution: There are two problems. First, the timeout is needed to guarantee that the ACK is received, and resend it if lost. Secondly, if the socket is closed immediately, we risk to mix delayed packets from the now closed connection with a new connection.

(iii) Consider a TCP connection between two hosts. Explain why the three-way handshake is not used to negotiate the receive window size. Justify your answer with an example. (2 Points)

Solution: The receive window can change throughout the connection. Thus, it's necessary to advertise it in every packet. Example: The receiving host is running at full capacity, such that the receiving application starts consuming data more slowly. If the sender doesn't throttle down the sending rate, the receiver's buffer will overflow. Thus, the receiver advertises the remaining buffer size in every ACK.

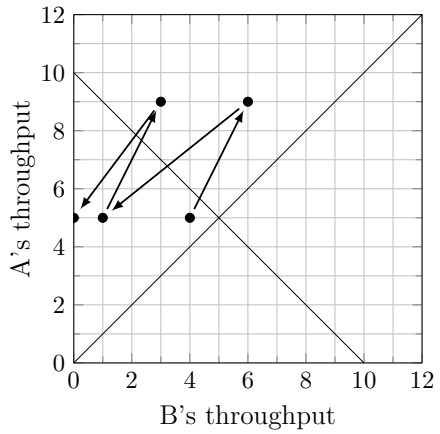
(iv) Consider two web servers. Web server A listens on port number 443, whereas web server B listens on port number 444. Explain what information a user needs to enter into a web browser to access each website. (1 Point)

Solution: Web server A uses a well-known port number, the default port number for HTTPS. Thus, a browser only requires the IP address to access the webpage. Web server B uses a non-standard port number. Thus, the web browser needs an IP address and the port number.

b) Efficiency and Fairness **(12 Points)**

leave blank

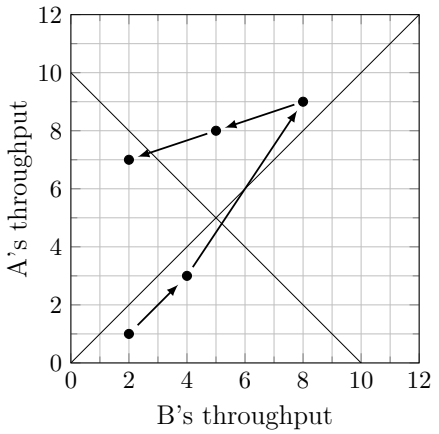
(i) Below are multiple system trajectory plots. For each plot, indicate which congestion control algorithm is shown. **Select one algorithm per plot**. (4 Points)



■ AIAD ☐ AIMD

☐ MIAD ☐ MIMD

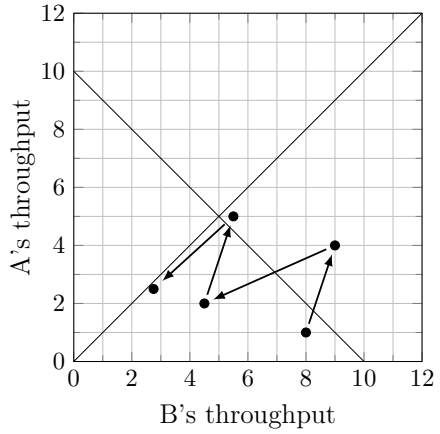
$\alpha_A = 3, \beta_A = 6, \alpha_B = 2, \beta_B = 4$



☐ AIAD ☐ AIMD

■ MIAD ☐ MIMD

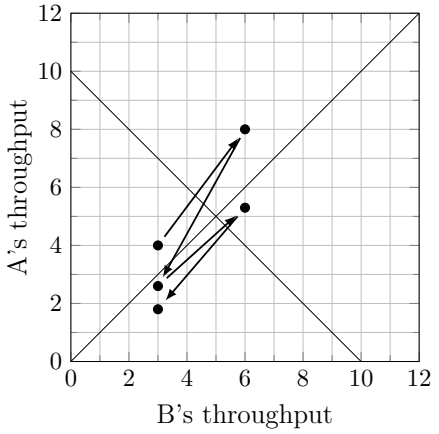
$\alpha_A = 3, \beta_A = 1, \alpha_B = 2, \beta_B = 3$



☐ AIAD ■ AIMD

☐ MIAD ☐ MIMD

$\alpha_A = 3, \beta_A = 2, \alpha_B = 1, \beta_B = 2$



☐ AIAD ☐ AIMD

☐ MIAD ■ MIMD

$\alpha_A = 2, \beta_A = 2, \alpha_B = 2, \beta_B = 3$

(ii) Two independent flows with AIMD CC algorithms share the same link. You notice that no matter how often you restart the flows, flow A *always* ends up starving flow B. What could be the reason for this behavior? Could this problem also arise in the public internet? *Justify your answer.* (4 Points)

Solution:

The two flows likely use different parameters α and β, so that they converge to an unfair allocation. This problem could potentially also arise in the public internet. But it requires the usage of a custom TCP stack.

(iii) Two independent flows with MIAD CC algorithms share the same link. As a reminder, MIAD is defined in the following way:

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

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

Given an initial bandwidth allocation $(\texttt{cwnd}_0^A, \texttt{cwnd}_0^B)$, is there a set of parameters (i.e., α_A, α_B, β_A and β_B) for which the flows will *always* converge to a fair bandwidth allocation? Justify your answer.

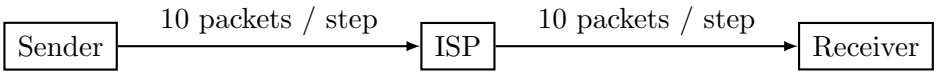
Hint: Perform a case distinction, $\texttt{cwnd}_0^A = \texttt{cwnd}_0^B$, and $\texttt{cwnd}_0^A \neq \texttt{cwnd}_0^B$. (4 Points)

Solution: For the case $x = y$, the two flows remain fair if the parameters are $\alpha_A = \alpha_B$ and $\beta_A = \beta_B$. For the case $x \neq y$, the two flows will always converge to an unfair allocation. In order to temporarily end up with a fair allocation, it is necessary that at least $\alpha_A \neq \alpha_B$ or $\beta_A \neq \beta_B$. But due to the different parameters, they will deviate from a fair allocation again.

leave blank

c) Congestion Control **(17 Points)**

In this question, we explore the impact of fluctuating network conditions on congestion control algorithms. We will model the network as a discrete event system, consisting of a sender, an ISP, and a receiver:



Both parties have an access link with a bandwidth of 10 packets per step. We assume that the sender operates in the following, fixed order. At each step t, it first processes incoming ACKs and checks for packet losses, then it updates its sending rate (y_t), retransmission timeout ($\texttt{RTO}_t$), and slow-start threshold ($\texttt{ssthresh}_t$) accordingly, before finally sending y_t packets. The sender computes the sending rate y_t as follows:

$$y_t = \begin{cases} 1 & \textit{Timeout:} \text{ if a timeout occurs in step } t \\ 2y_{t-1} & \textit{Slow start:} \text{ if } y_{t-1} < \texttt{ssthresh}_t \\ y_{t-1} + 1 & \text{otherwise} \end{cases}$$

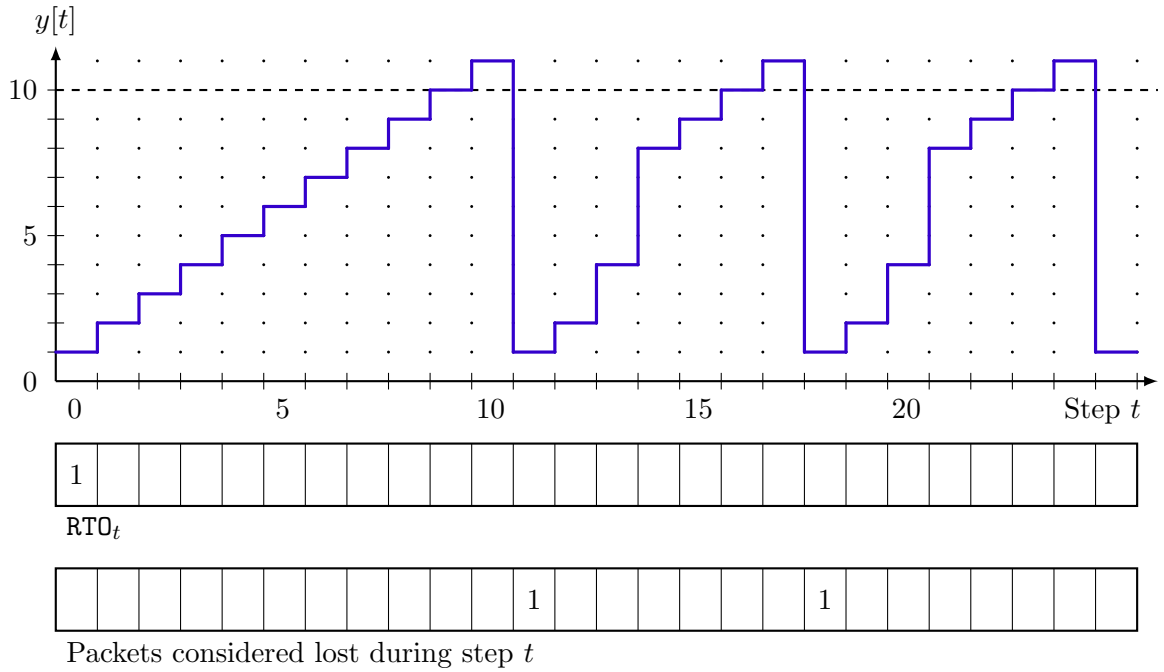
For each packet sent at step t, the sender sets a timer that times out in $\texttt{RTO}_t$ steps. If a timeout occurs, and the respective packet has not been ACK'ed yet, the sender considers the packet lost. In that case, **all timers are reset**, and $\texttt{ssthresh}_t$ is updated as follows:

$$\texttt{ssthresh}_t = \begin{cases} \left\lfloor \max\left(y_{t-1}/2, 1\right) \right\rfloor & \textit{Timeout:} \text{ if a timeout occurs in step } t \\ \texttt{ssthresh}_{t-1} & \text{otherwise} \end{cases}$$

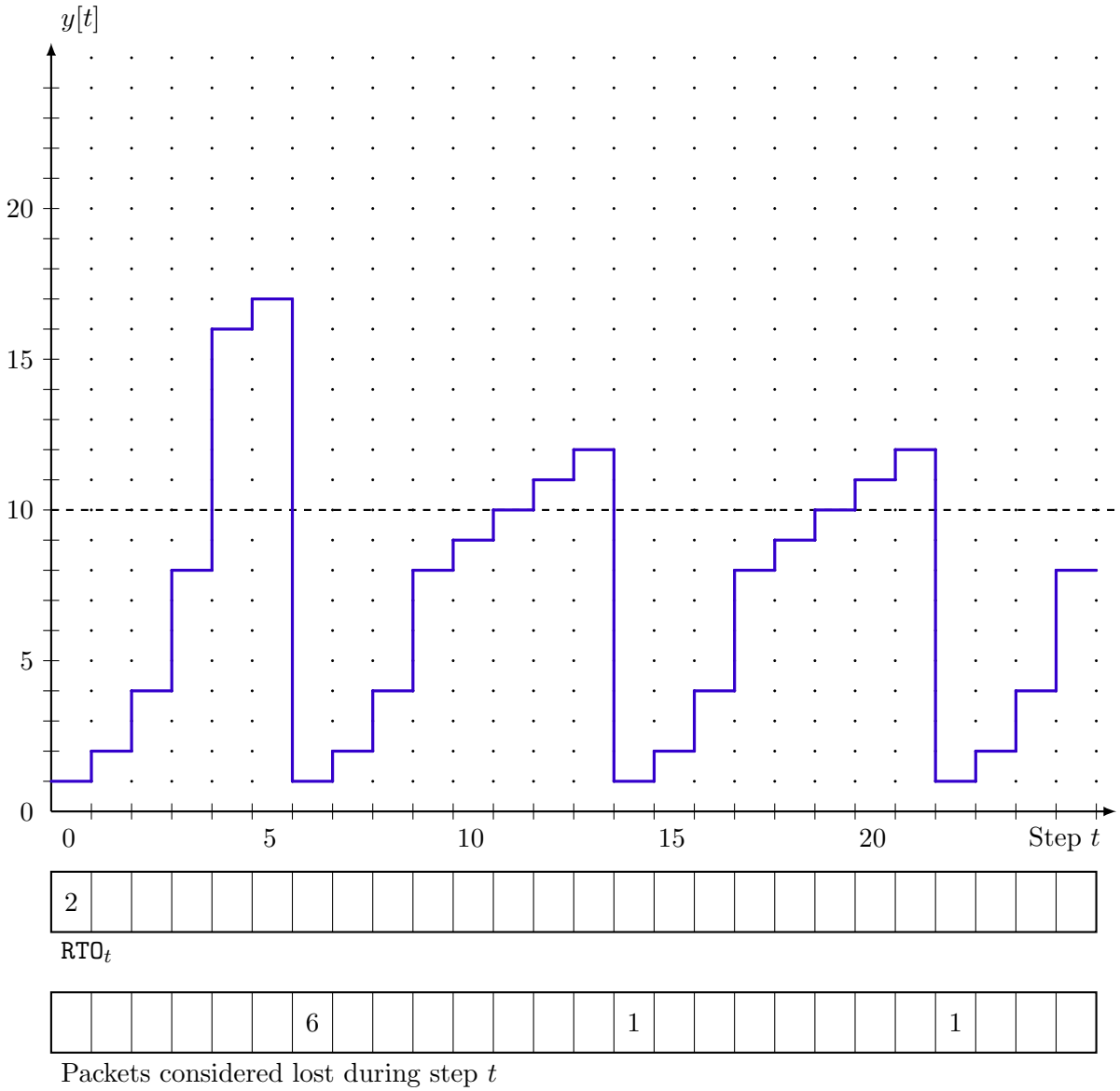
In this question, you will draw multiple throughput diagrams that consist of three parts:

1. The number of packets y_t sent by the sender at step t.
2. The **given** $\texttt{RTO}_t$ timeout for packets sent at step t (empty means unchanged).
3. The number of packets considered lost at step t (empty means zero).

As an example, here is the throughput diagram for $\texttt{ssthresh}_0 = 0$ and a fixed $\texttt{RTO} = 1$:

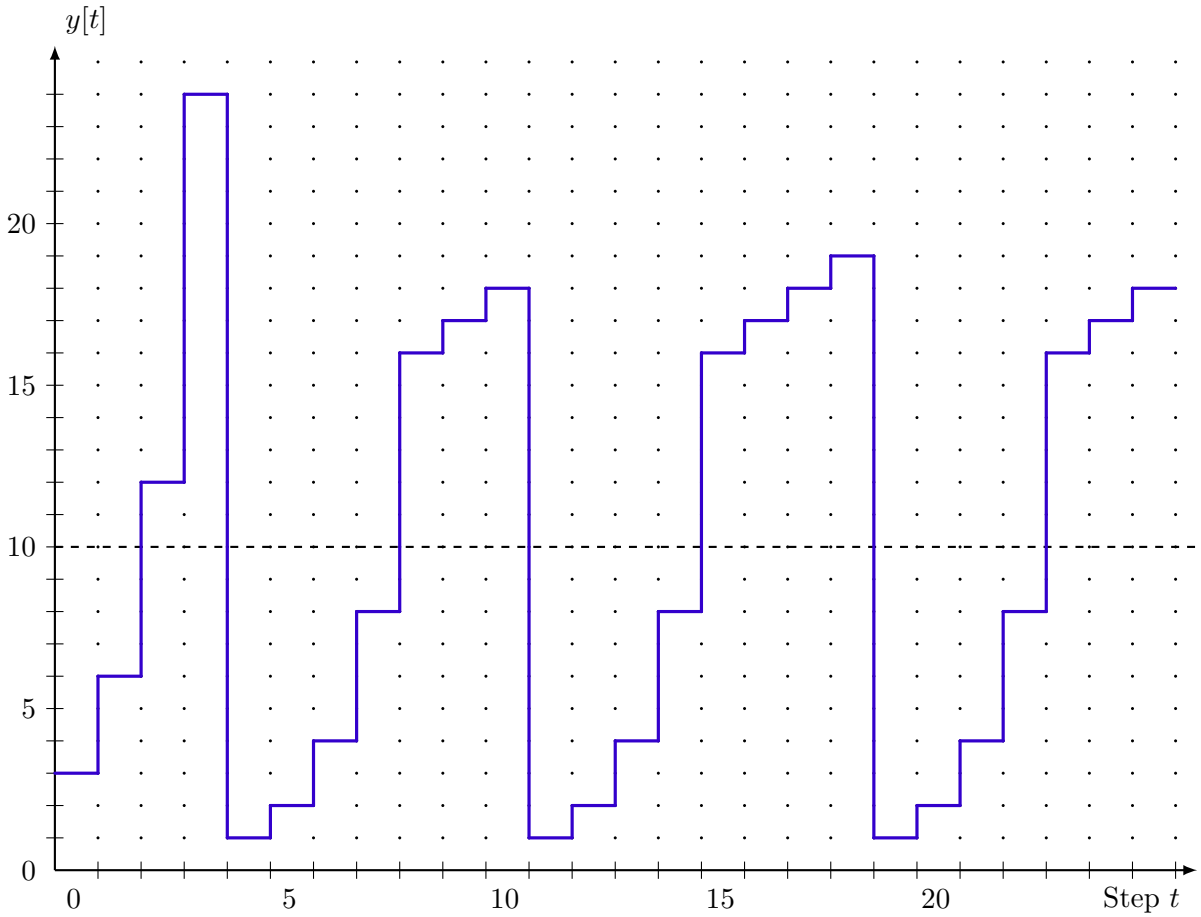


(i) For this question, consider a sender that uses a fixed $\texttt{RTO}_t = 2$ for every step t. Draw the throughput diagram for $\texttt{ssthresh}_0 = 10$ and the given $\texttt{RTO}_t$. Make sure to include the number of packets considered lost. (8 Points)



- 0 - 4: Slow start due to $y < \texttt{ssthresh}_t = 10$
- 5: Transition to AIMD because $y_5 = 16 > \texttt{ssthresh}_5 = 10$
- 6: Timeout, because at $s = 6 - \texttt{RTO} = 4$ six packets were sent and dropped. Update $\texttt{ssthresh}_t = 8$.
- 6-13: Slow start until $y = 8$, then transition to AIMD.
- 14: Timeout, because at $s = 14 - \texttt{RTO} = 12$ a packet was sent and dropped. Update $\texttt{ssthresh}_t = 6$.
- 14-21: Slow start until $y = 6$, then transition to AIMD, because $y_{17} = 8 > \texttt{ssthresh}_{17} = 6$.

(ii) Now, consider a sender that approximates $\texttt{RTO}_t$ at each step (e.g. at $t = 7$, $\texttt{RTO}_7$ is updated to 3). Draw the throughput diagram for $\texttt{ssthresh}_0 = \infty$ and the given $\texttt{RTO}_t$. Make sure to include the number of packets considered lost. (9 Points)



- 0 - 4: Slow start due to $y < \texttt{ssthresh}_t = \infty$
- 5: Timeout, because at $s = 5 - \texttt{RTO}_2 = 3$ two packets were sent and dropped. Update $\texttt{ssthresh}_t = 12$.
- 5-10: Slow start until $y = 8$, then transition to AIMD, because $y_8 = 16 > \texttt{ssthresh}_8 = 12$.
- 8-10: Packets are being dropped, but $\texttt{RTO}_t = 3$, so no timeout occurs just yet.
- 11: Timeout, because at $s = 11 - \texttt{RTO}_8 = 8$ 6 packets were sent and dropped. Update $\texttt{ssthresh}_t = 9$.
- 11-18: Slow start until $y = 8$, then transition to AIMD, because $y_{15} = 16 > \texttt{ssthresh}_{15} = 9$.
- 19: Timeout, because at $s_1 = 19 - \texttt{RTO}_{15} = 15$ and at $s_2 = 19 - \texttt{RTO}_{17} = 17$, 6 and 8 packets, respectively, were sent and dropped. Update $\texttt{ssthresh}_t = 9$.

leave blank

leave blank

Task 5: Applications **19 Points**

a) Warm-Up **(6 Points)**

(i) Feeling adventurous, you type:

`https://comm-net.ethz.ch:4242/pdfs/exam_august_24.pdf`

in your browser's address bar, and press enter. (4 Points)

Write down the DNS requests issued by your local DNS server, assuming its cache is completely empty. For each request, indicate: the queried name, the server being queried (describe the server in English, e.g. "the DNS server responsible for `a.b.c`)", alongside with the DNS query type. You can assume that each DNS request is replied to correctly, and you do *not* need to write down the DNS replies.

Hint: You do not necessarily need to fill in all the entries.

Solution:
DNS request #1
- queried name: .ch or comm-net.ethz.ch
- queried server: One of the root servers
- query type: A

- DNS request #2
- queried name: ethz.ch or comm-net.ethz.ch
- queried server: One of the DNS servers responsible for .ch
- query type: A

- DNS request #3
- queried name: comm-net.ethz.ch
- queried server: One of the DNS servers responsible for ethz.ch
- query type: A

Write down a source and a destination port that could be used by the corresponding TCP connection.

Solution: - src port: XXX (any random, non-assigned port)
dst port: 4242

Write down the content of the HTTP `GET` query your browser will send to the server. (Only include the mandatory fields in your answer.)

Solution: - GET /pdfs/exam_august_24.pdf HTTP/1.1
Host: comm-net.ethz.ch

(ii) Briefly explain one way to automatically redirect users browsing an unsecure version of a website (e.g. `http://ubs.ch`) to the secure version (e.g. `https://ubs.ch`).
(2 Points)

Solution:

The server replies with an HTTP reply with status "301 Moved Permanently", and location "https://ubs.ch".

b) Anycast strikes back (6 Points)

leave blank

Inspired by DNS root servers, you would like to rely on BGP anycast for hosting your website `routing.is.fun` which currently sits behind `129.132.19.216` (you own `129.132.19.0/24`) and `2001:67c::200` (you own `2001:67c::/8`). Your goal is to deploy and serve content from three replicas servers located in New York (AS 10), Zürich (AS 20), and Singapore (AS 30).

(i) Describe the steps necessary to realize this from a routing viewpoint. Be precise. (2 Points)

Solution:

All three ASes host one replica with the IP address 129.132.19.216 / 2001:67c::200. All three ASes advertize "129.132.19.0/24" and "2001:67c::/8" to their external neighbors using BGP. BGP routers around the world select one of the routes as best and direct their traffic to the corresponding replica.

(ii) Assuming no DNS records exist for your website yet, write down the DNS record(s) you would need to add to the DNS server responsible for `is.fun`. *Hint:* You might not need all 3 records. (2 Points)

Solution:

- "routing.is.fun", "A", "129.132.19.216"
- "routing.is.fun", "AAAA", "2001:67c::200"

(iii) A typical challenge when using anycast for load-balancing web traffic is that web traffic relies on reliable transport (it uses TCP), whereas DNS does not (it uses UDP). Explain why this could be a problem for `routing.is.fun` with a concrete example. (2 Points)

Solution:

- challenge: TCP connections have state, so e.g., if communication switches from one server to another, one needs to re-open a connection
- problematic situation: you have connected to a server in AS 10, but then the path changes to a server in AS 20, as the result of AS 10 withdrawing its advertizements for "129.132.19.0/24" (due to a faulty re-configuration)

c) Fun with proxies (7 Points)

leave blank

Consider that ETH Zürich has 10 Gbps of access capacity towards the Internet which it uses to serve the web (HTTP) requests of 20 000 users. Each user issues 4 HTTP requests per second on average, and each requested HTTP object is exactly 100 kbit. The ETH network operators notice two problems: *(i)* frequent overloads of the access link; and *(ii)* continuous slow loading times for the users, *even when the access link is lightly loaded.*

ETH is considering setting up a proxy server acting as a cache to reduce the access link load and loading times. Concretely, whenever an ETH user would request an HTTP object, the request would first go to the proxy server which will serve the object from its cache, if it can, or contact the origin server on behalf of the user, if it cannot.

(i) To avoid congestion, the ETH operators would like to maintain the average access link load below 20% (2 Gbps). What should the minimum cache hit rate be in order to guarantee this? Describe your computation. (3 Points)

Solution:

- Minimum cache hit rate: 75%.
- 20 000 users * 4 requests per second * 100 kbit = 8 Gbps.
- We need to bring down the load to 2 Gbps, meaning 25% of the requests should go to the Internet, meaning that we need a hit rate of 75% at least.

(ii) Consider that the round trip time (RTT) between an ETH user and the proxy is 10 msec, and the RTT between the proxy and any Internet server is 150 msec. What is the average delay experienced by users assuming a cache hit rate of 60%? Describe your computation. (2 Points)

Solution:

Here, we could accept two answers, depending on how you interpret the RTT:
- 40% of the requests takes (150 + 10) = 160 msec
- 60% of the requests takes 10 msec
- 0.4*160 + 0.6*10 = 70 msec

and: - 40% of the requests takes (150 + 10)*2 = 320 msec
- 60% of the requests takes 10*2 msec = 20 msec
- 0.4*320 + 0.6*20 = 140 msec

(iii) Analyzing the traffic going over the access link, the ETH operators realize that most of the HTTP-based video traffic (such as Netflix's) is *not* cached, yet they know that a lot of students are watching the same shows at exactly the same time. The ETH operators suspect that Netflix is using strategies to prevent its video chunks from being cached. (You can assume that video chunks are sent unencrypted.) Explain two distinct techniques Netflix could use to achieve this.

(2 Points)

Solution:

- technique #1: in the HTTP reply, server hints that object expired/not cacheable.
- technique #2: Netflix use different URL/URIs for the same content.

Copy of Figure 2 (not graded)

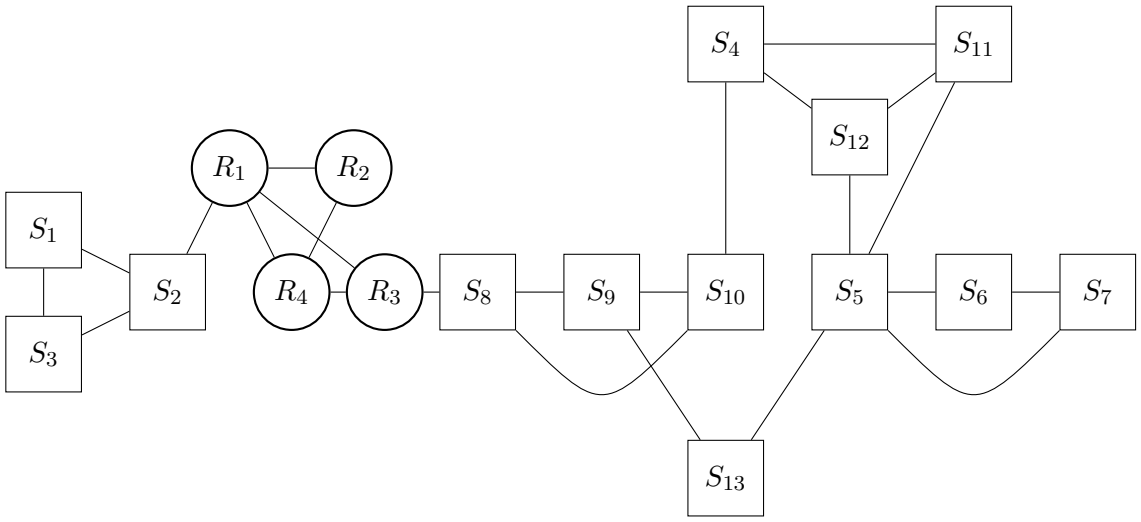


Figure 12: Topology of layer-2 switches (squares) and layer-3 routers (circles).

Copy of Figure 3 (not graded)

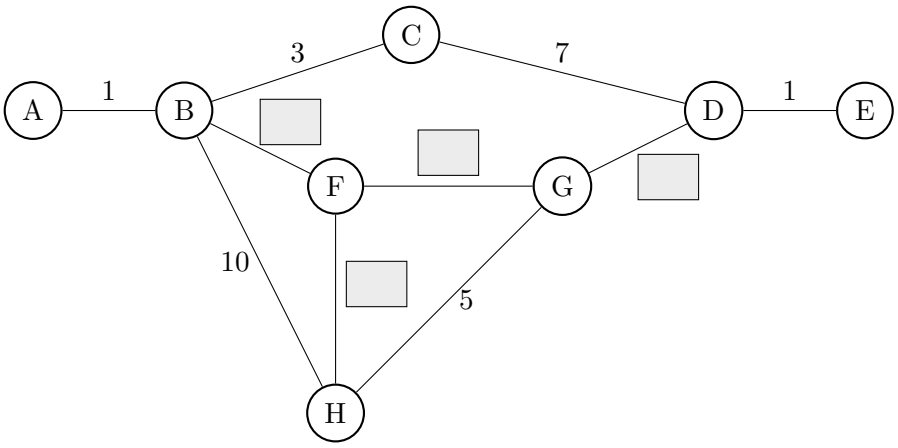


Figure 13: The network to load balance.

Copy of Figure 5 (not graded)

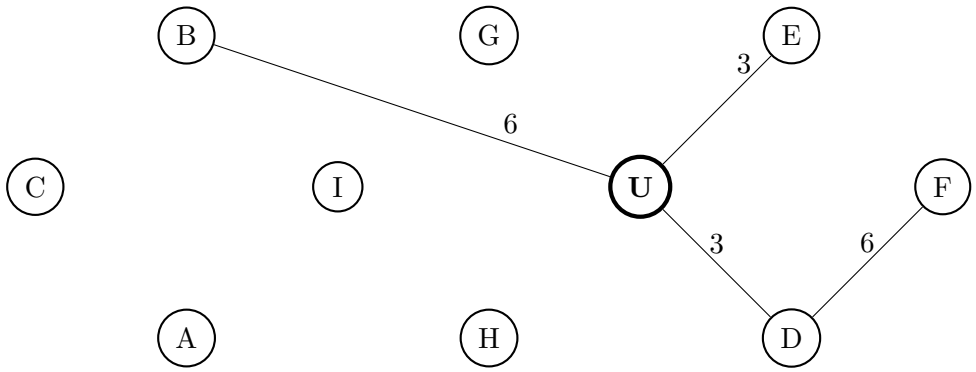


Figure 14: A network consisting of 10 nodes with missing links and link weights.

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: _____

Extra Sheet 2

Task: _____

Task: _____

Extra Sheet 3

Task: _____

Task: _____