CS 168 Fall 2026

Introduction to the Internet Midterm

Solutions last updated: Friday, October 23, 2026

PRINT Your Name: _____

PRINT Your Student ID: _____

You have 110 minutes. There are 7 questions of varying credit. (100 points total)

Question:	1	2	3	4	5	6	7	Total
Points:	11	10	14	16	21	18	10	100

For questions with **circular bubbles**, you may select only one choice.

- Ⓐ Unselected option (Completely unfilled)
- Ⓑ Don't do this (it will be graded as incorrect)
- ● Only one selected option (completely filled)

For questions with **square checkboxes**, you may select one or more choices.

- ■ You can select
- ■ multiple squares
- ☑ Don't do this (it will be graded as incorrect)

Anything you write outside the answer boxes or you ~~cross out~~ will not be graded. If you write multiple answers, your answer is ambiguous, or the bubble/checkbox is not entirely filled in, we will grade the worst interpretation.

Honor Code: Read the honor code below and sign your name.

I understand that I may not collaborate with anyone else on this exam, or cheat in any way. I am aware of the Berkeley Campus Code of Student Conduct and acknowledge that academic misconduct will be reported to the Center for Student Conduct and may further result in, at minimum, negative points on the exam.

SIGN your name: _____



Doodle credit: Valerie Eng

Q1 *Potpourri* **(11 points)**

Q1.1 (2 points) In Project 1 (Traceroute), which of these best describes the result of using an unreachable destination port in each probe?

- (A) The destination sends an ICMP Port Unreachable message.
- (B) The destination sends an ICMP Time Exceeded message.
- (C) Intermediate routers each send an ICMP Port Unreachable message.
- (D) Replies to the probes are sent to an unreachable port that is only used for traceroute.

Solution: If we send a probe to an unreachable port, every intermediate router whose TTL expires will send back an ICMP Time Exceeded message. The destination will instead send an ICMP Port Unreachable message, confirming that the probe reached the destination rather than being lost along the way.

Q1.2 (2 points) In Project 2 (Routing), which of these best describes the result of setting the expiry time of poisoned routes to `api.current_time() + self.ROUTE_TTL`, instead of deleting the routes right away?

- (A) The router periodically advertises poison for expired routes.
- (B) The router continues forwarding packets along the expired route until a new route is found.
- (C) The router's neighbors forward packets along the expired route until a new route is found.
- (D) The router forwards the poison to all neighbors except the next-hop router.

Solution: (A) is correct. Keeping the poisoned entry means it gets re-advertised periodically, so neighbors that missed the first poison still learn the route is gone.

(B) and (C) are false because routes with latency `INFINITY` are never used to forward packets.

(D) is false because poison is advertised to all neighbors.

Q1.3 (3 points) CS 168 owns the IPv4 address range `10.168.0.0/16`, and wants to allocate a block of 400 IP addresses in this range, starting at `10.168.0.0`.

What is the smallest IP prefix that fits this block? Write your answer in the form `a.b.c.d/n`.

`10.168.0.0/23`

Solution: The block must contain at least 400 addresses. A prefix of length n leaves $32 - n$ host bits, giving 2^{32-n} addresses. $2^8 = 256$ is too few, and $2^9 = 512 \geq 400$ is enough, so we need 9 host bits. The prefix length is $32 - 9 = 23$.

The block starts at 10.168.0.0, which is a multiple of 512, so it lines up with a /23 boundary.

Q1.4 (2 points) Select all true statements about router hardware.

[A] The data plane computes routes and installs forwarding table entries.

[B] The fabric allows data to travel between linecards.

[C] The data plane is used every time a user packet arrives.

[D] Operators typically interact with routers using the control plane.

(E) None of the above

Solution: A is false as the control plane computes routes and installs forwarding table entries. B and C are true. D is false as operators interact with routers via the management plane.

Q1.5 (2 points) In BGP, why does an AS reject a route advertisement if its own AS number already appears in the advertisement's ASPATH?

(A) To prevent routing loops.

(B) To select the lowest-cost route.

(C) To prevent multiple routes to the same destination.

(D) To select the route with the fewest AS hops.

Solution: BGP uses the AS-PATH attribute to track which ASes an advertised route has traversed.

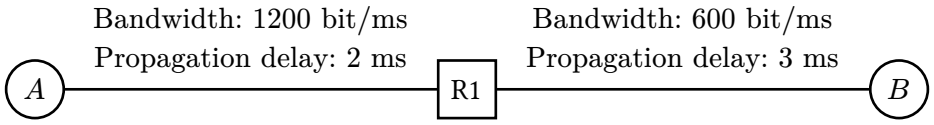
If an AS sees its own AS number in the AS-PATH, accepting that route could create a routing loop.

Rejecting such advertisements prevents these AS-level routing loops.

Q2 *Are The Links Green?* **(10 points)**

Consider the topology below. Assumptions:

- A router must receive all bits of a payload before sending that payload along the next link.
- Queue sizes are infinite, and queues are processed in FIFO order.
- A packet is considered to "arrive" at a router/host when the last byte of the packet is received by that router/host.



Host A sends a packet to host B using UDP-over-IPv4. The packet consists of:

- An application payload of size 800 bits.
- A UDP header of size 100 bits.
- An IPv4 header of size 150 bits.
- A link-layer header of size 150 bits.

Q2.1 (1 point) What is the total size of the packet, including all headers?

1200 bits

Solution: The packet contains the application message, UDP header, IPv4 header, and link-layer header:

$$800 + 100 + 150 + 150 = 1200 \quad \text{bits}$$

Q2.2 (2 points) Based on lecture (i.e. no router optimizations or extra features), which parts of the packet does R1 modify? Select all that apply.

- [A] Application payload
- [B] UDP header
- [C] IPv4 header
- [D] Link-layer header
- (E) None of these

Solution: The application payload and UDP header remain unchanged.

The IPv4 header changes because R1 decrements the TTL and updates the IPv4 header checksum.

The link-layer header changes because the packet is transmitted over a different link.

Q2.3 (4 points) A transmits the first bit of the packet at $t = 0$. What time does the packet arrive at B?

$t = 8$ ms

Solution: From the previous subpart, the packet size is 1200 bits.

The transmission delays are:

$$\frac{1200}{1200} = 1 \text{ ms}$$

and

$$\frac{1200}{600} = 2 \text{ ms}$$

The propagation delays are 2 ms and 3 ms.

Since R1 must receive the entire packet before forwarding, the total delay is:

$$1 + 2 + 2 + 3 = 8 \text{ ms}$$

Q2.4 (3 points) This subpart is independent of the previous subparts.

A transmits two packets to B, one after the other. Each packet is 600 bits (including all headers).

A transmits the first bit of the first packet at $t = 0$. What time does the **second** packet arrive at B?

(A) 4 ms (C) 7 ms (E) 8 ms (G) 10 ms

(B) 6.5 ms (D) 7.5 ms (F) 9 ms (H) 16 ms

Solution:

The transmission delay of each packet is 0.5 ms on Link 1 and 1 ms on Link 2.

The first packet reaches R1 at 2.5 ms and finishes transmission on Link 2 at 3.5 ms.

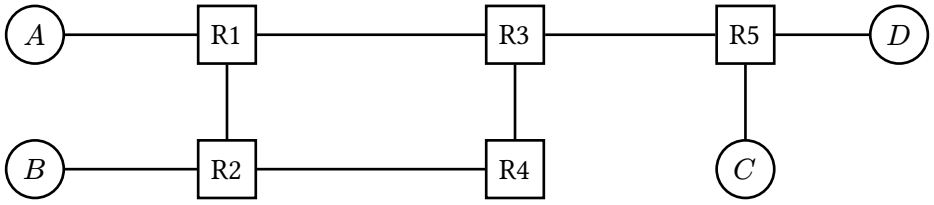
The second packet reaches R1 at 3 ms, so it waits until 3.5 ms before being transmitted on Link 2. It finishes transmission at 4.5 ms, then incurs 3 ms of propagation delay.

Therefore, B receives the last bit at

$$4.5 + 3 = 7.5 \text{ ms.}$$

Q3 *Distance-Vector* **(14 points)**

Consider running the distance-vector algorithm from lecture on this topology. All links cost 1.



Assumptions:

- At $t = 0$, all forwarding tables are empty.
- At $t = 1$, static routes are installed.
- At every time step starting at $t = 2$, each router (1) removes expired routes, then (2) processes advertisements and updates its table, then (3) sends out advertisements.
- Transmission delay is negligible, i.e. an advertisement sent out at $t = 2$ can be processed at $t = 3$.
- When two routes have the same cost, tiebreak by preferring the router with lower number.
- Routing table entries expire after 5 time steps of receiving no advertisements.
- Unless otherwise specified, split horizon, poison reverse, and route poisoning are disabled.
- All costs 16 or greater are considered infinite.

Q3.1 (2 points) What is the first time step where, after processing advertisements at that time step, R2 has an entry for every destination?

Ⓐ $t = 1$ Ⓑ $t = 2$ Ⓒ $t = 3$ Ⓓ $t = 4$ Ⓔ $t = 5$ Ⓕ $t = 6$

Solution:

t = 1: R2 installs its static route to B.

t = 3: R2 learns A from R1 (R1 advertised its static route at t = 2).

t = 4: R1 learns C and D from R3 (R3 learned them from R5 at t = 3). R4 also learns C and D from R3. R2 does not have them yet.

t = 5: R2 learns C and D from R1 (and from R4) at cost 4. R2 now has all four destinations.

A destination whose static route is h router hops away first appears at t = 2 + h. C and D are 3 router hops from R2 (R2 ⟶ R1 ⟶ R3 ⟶ R5), so R2 learns them at t = 5.

Q3.2 (4 points) Fill in R3's forwarding table immediately after processing advertisements at $t = 3$.

If no route has been learned for a destination, write "None" in both columns.

R3's Forwarding Table		
Destination	**Next Hop**	**Cost**
A		
B		
C		
D		

Solution:

Destination	Next hop	Cost
A	R1	2
B	None	None
C	R5	2
D	R5	2

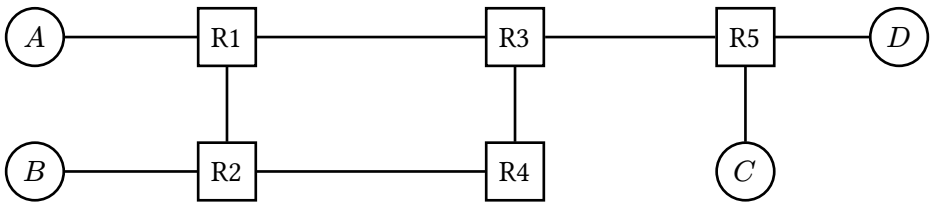
At $t = 3$, R3 processes the advertisements sent at $t = 2$.

At $t = 2$, the only routes in the network are the static routes installed at $t = 1$ (R1 to A, R2 to B, R5 to C and D).

R3 hears from R1 (A) and R5 (C, D). Each cost is 1 (link to the neighbor) + 1 (cost advertised by the neighbor) = 2.

R3 is not adjacent to R2, and R4 had no routes to advertise at $t = 2$, so R3 has not learned a route to B yet.

The diagram, reprinted for your convenience:



Q3.3–Q3.5 are independent, i.e. each subpart starts from the original network.

Q3.3 (3 points) The network converges. Then the R3–R5 link goes down.

Which router(s) eventually have a cost-16 entry for D? Select all that apply.

[A] R1 [B] R2 [C] R3 [D] R4 [E] R5 (F) None

Solution: R5 is still directly connected to D, so its static route (cost 1) is unaffected. R5 never has a cost 16 entry.

R3 stops receiving advertisements from R5, so its route to D expires 3 seconds later. R1 and R4 still have stale routes to D via R3, and with split horizon disabled they advertise them back to R3. R3 has no route, so it accepts one (cost 4). R3 is now the next hop of a router whose next hop is R3, so the two advertise to each other. Each must accept the other's increasing cost (Rule 2, update from next hop), and the costs count up to 16.

R4 uses R3 as its next hop, and R2 uses R1, so both follow the increasing costs up to 16 as well. No router can reach D anymore except R5, so nothing stops the counting until the costs hit 16.

Q3.4 (2 points) In this subpart only, R3 uses poison reverse and every other router uses split horizon.

The network converges. Then the R3–R5 link goes down.

True or false: At some later time step t, packets destined for D are forwarded in a loop.

(A) True (B) False

Solution: R3's poison reverse only sends ∞ to R3's own next hop for D. In a 4-router loop, that next hop's own next hop is a different router, not R3. R4's next hop is R2, so R4 ignores the ∞.

Q3.5 (3 points) In this subpart only, you may choose to enable poison reverse on exactly one router. All other routers use split horizon.

The network converges. Then the R3–R5 link goes down.

Which router should have poison reverse enabled, so that the network does not suffer from the count-to-infinity problem?

(A) R1 (B) R2 (C) R3 (D) R4 (E) R5

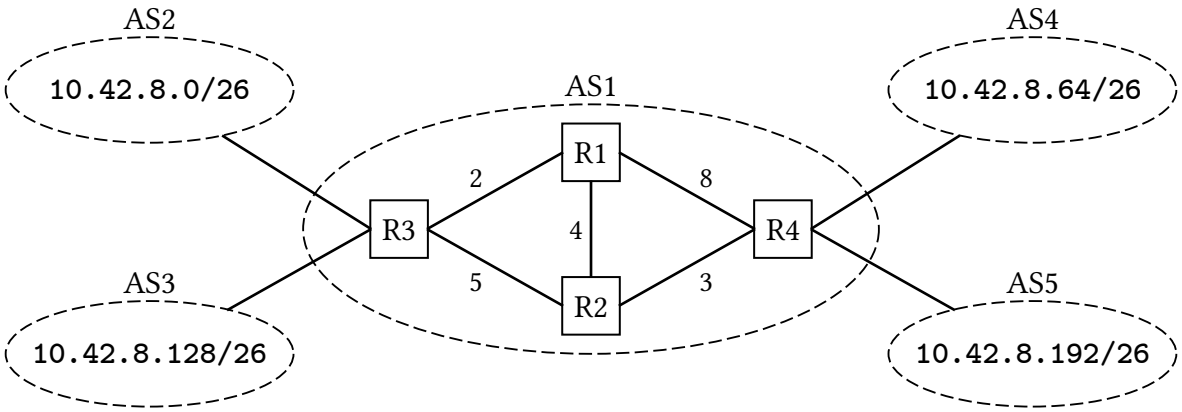
(F) None; the count-to-infinity problem occurs no matter which router you choose.

Solution: With split horizon, a router never advertises a route back to its own next hop, so two neighboring routers can't form a loop by routing through each other. The question states that the network still counts to infinity with split horizon enabled everywhere, so the loop doing the counting must contain at least 3 routers. In this graph, the only such cycle is R1–R3–R4–R2.

Poison reverse adds one thing to split horizon: instead of staying silent, a router advertises ∞ to its own next hop. That only matters if the next hop routes back through the sender, which is a 2-router loop. In a loop of 3 or more routers, the next hop routes through a different router, so it ignores the ∞: the ∞ is not a better route, and it does not come from that router's own next hop. So enabling poison reverse on any single router cannot stop counting to infinity.

Q4 *Link-State + IP Addressing* **(16 points)**

Consider the topology below, with five ASes:



AS1 runs link-state as its intra-domain routing protocol.

Q4.1 (2 points) At convergence, what is the path cost for the following source-destination pairs? Write one number in each box.

R1 to R3:	2	R1 to R4:	7
R2 to R3:	5	R2 to R4:	3

Solution: R1-R3 costs 2 and R2-R4 costs 3. R1-R2-R4 costs $4 + 3 = 7$, less than R1-R4's cost 8. R2-R3 costs 5, less than R2-R1-R3's cost $4 + 2 = 6$.

Q4.2 (2 points) For this subpart only, the R1–R4 link has cost 7. All other costs are unchanged.

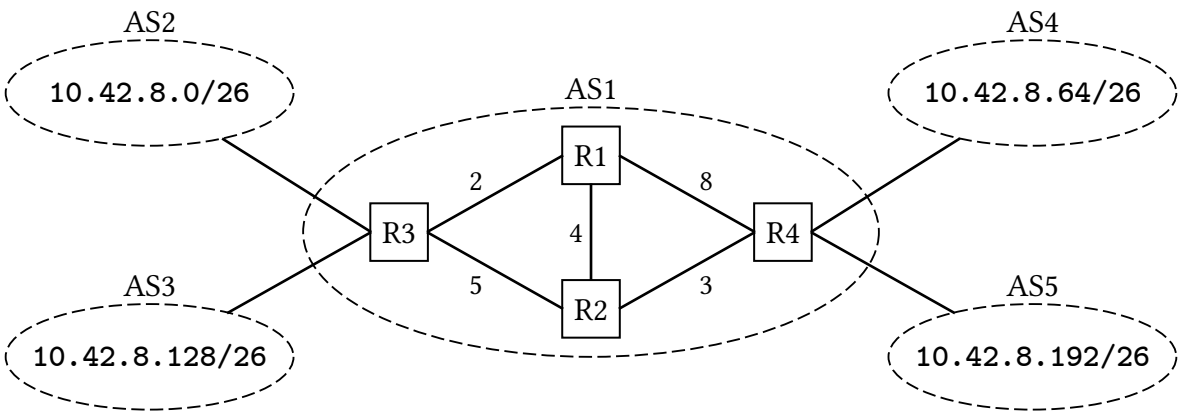
The routers run link-state routing, tiebreaking by preferring the lower-numbered next-hop router.

After convergence, which next-hop does R1 use to forward packets toward R4?

- (A) R2
- (B) R3
- (C) R4
- (D) No valid route exists.

Solution: False. R1 has two shortest paths to R4: R1-R4 and R1-R2-R4, both with cost 7. Due to the tie-breaking scheme, R1 chooses R2 as its next hop.

The diagram, reprinted for your convenience:



AS1's border routers run eBGP to learn about routes to AS2, AS3, AS4, and AS5.

Then, AS1 uses iBGP to advertise routes to AS2, AS3, AS4, and AS5 to the internal routers.

Because of a bug in iBGP, not all routers have the same information about egress routers:

R1's iBGP Table	
Destination	**Egress Router**
`10.42.8.0/24`	R3

R2's iBGP Table	
Destination	**Egress Router**
`10.42.8.0/24`	R4

R3 and R4's iBGP Table	
Destination	**Egress Router**
`10.42.8.0/26`	R3
`10.42.8.64/26`	R4
`10.42.8.128/26`	R3
`10.42.8.192/26`	R4

In Q4.3–Q4.4, select the first four routers visited by each packet. Do **not** include the starting router.

Choose one option in each row. If the packet leaves AS1, choose "Exited" for all remaining rows.

Q4.3 (4 points) A packet starts at R1 with destination `10.42.8.70`. After R1, the next routers are:

First hop:	Ⓐ R1	Ⓑ R2	**Ⓒ R3**	Ⓓ R4	Ⓔ Exited
Second hop:	Ⓐ R1	**Ⓑ R2**	Ⓒ R3	Ⓓ R4	Ⓔ Exited
Third hop:	Ⓐ R1	Ⓑ R2	Ⓒ R3	**Ⓓ R4**	Ⓔ Exited
Fourth hop:	Ⓐ R1	Ⓑ R2	Ⓒ R3	Ⓓ R4	**Ⓔ Exited**

Solution: BGP tables used in both packet traces. A BGP lookup maps the destination prefix to an **egress router**. Link-state routing then supplies the next hop toward that egress.

R3 reaches egress R4 through R2; R4 reaches egress R3 through R2. When the current router is the selected egress, the packet exits AS1 toward the corresponding external router.

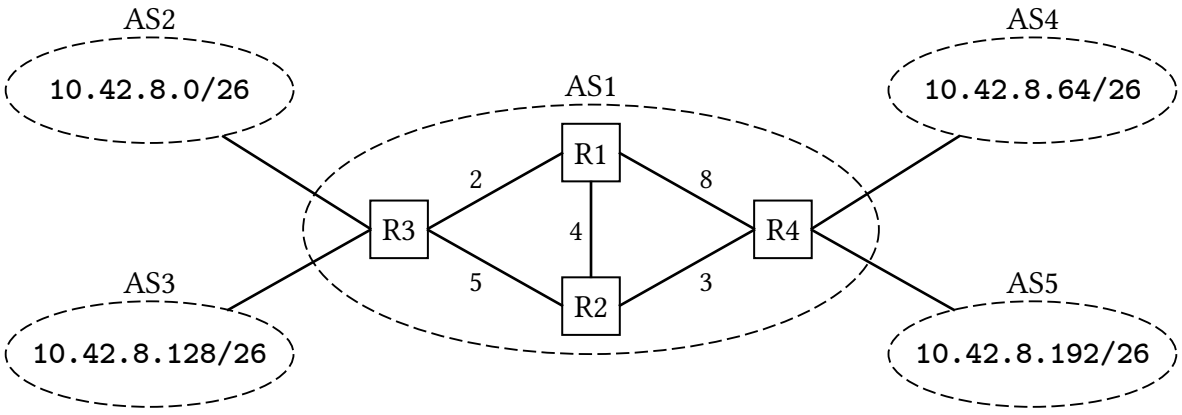
`10.42.8.70` is in R4's `10.42.8.64/26` subnet. At R1, the address matches the aggregated /24 entry, which sends the packet to R3. R3 keeps its original /26 entry: its shortest path to R4 is R3-R2-R4, with cost $5 + 3 = 8$, so it forwards the packet to R2. At R2, the address matches the aggregated /24 entry pointing to R4. R2 forwards the packet to R4, where it is delivered.

Q4.4 (4 points) A packet starts at R2 with destination `10.42.8.150`. After R2, the next routers are:

First hop:	Ⓐ	R1	Ⓑ	R2	Ⓒ	R3	Ⓓ	R4	Ⓔ Exited
Second hop:	Ⓐ	R1	Ⓑ	R2	Ⓒ	R3	Ⓓ	R4	Ⓔ Exited
Third hop:	Ⓐ	R1	Ⓑ	R2	Ⓒ	R3	Ⓓ	R4	Ⓔ Exited
Fourth hop:	Ⓐ	R1	Ⓑ	R2	Ⓒ	R3	Ⓓ	R4	Ⓔ Exited

Solution: `10.42.8.150` is in R3's `10.42.8.128/26` subnet. At R2, the address matches the aggregated /24 entry, which sends the packet to R4. R4 keeps its original /26 entry: its shortest path to R3 is R4-R2-R3, with cost $3 + 5 = 8$, so it forwards the packet back to R2. R2 uses its same /24 entry and sends the packet to R4 again. Neither router's forwarding entry changes, so they continue sending the packet back to each other.

Diagram and tables, reprinted for your convenience:



R1's iBGP Table	
Destination	**Egress Router**
`10.42.8.0/24`	R3

R2's iBGP Table	
Destination	**Egress Router**
`10.42.8.0/24`	R4

R3 and R4's iBGP Table	
Destination	**Egress Router**
`10.42.8.0/26`	R3
`10.42.8.64/26`	R4
`10.42.8.128/26`	R3
`10.42.8.192/26`	R4

In Q4.5–Q4.6, a new link with positive integer cost k is added between R3 and R4.

All routers in AS1 learn about the new link through link-state routing and recompute their shortest paths. The iBGP tables remain unchanged from above.

Assume all routers tiebreak by preferring the lower-numbered next-hop router.

Q4.5 (2 points) Our goal is to allow any packet, originating at any router in AS1, to be delivered to any other AS.

What are all values of k that achieve this goal? Write an integer in the box.

$k \leq 7$

Solution: $k \leq 7$. The original shortest path between R3 and R4 is R3-R2-R4, with cost $5 + 3 = 8$. For $k < 8$, R3 and R4 use the new direct link, so all packets reach their destinations. For $k \geq 8$, R4 forwards packets for R3's subnets through R2, but R2's /24 entry sends them back to R4, creating a forwarding loop.

Q4.6 (2 points) Our goal is to allow any packet, originating at any router in AS1, to be delivered to any other AS. Also, we want every packet to follow a least-cost path to an egress router within AS1.

What are all values of k that achieve these goals? Write an integer in the box.

$k \leq 2$

Solution: $k \leq 2$. R1 sends packets for R4's subnets along R1-R3-R4 at cost $2 + k$, compared with R1-R2-R4 at cost 7. This is shortest when $k \leq 5$. R2 sends packets for R3's subnets along R2-R4-R3 at cost $3 + k$, compared with R2-R3 at cost 5. This is shortest when $k \leq 2$.

Q5 *Routers: Trick or Trie!* **(21 points)**

Consider a router, as introduced in class.

Q5.1 (1 point) Which of these planes operates on the slowest time scale?

(A) Management (B) Data (C) Control

Solution: Management.

Q5.2 (2 points) Which of these router components does punt traffic pass through? Select all that apply.

[A] Forwarding Chip [C] Fabric [E] Fabric Chips

[B] Control Processor [D] Physical Ports (F) None of the above

Solution: Ports, Forwarding Chip, then "punted" to Control Processor without touching the Fabric Chip or the Fabric.

Q5.3 (2 points) Which of these address ranges are fully contained within the range `169.229.0.0/16`? Select all that apply.

[A] `169.229.129.0/24` [C] `169.0.0.0/8` [E] `169.245.154.0/24`

[B] `169.229.226.23/32` [D] `169.229.0.0/32` (F) None of the above

Solution: Match the first fixed 16 bits.

Fun fact - this is the UC Berkeley IPv4 prefix.

Q5.4 (3 points) Fill in the new table with three prefixes, such that all routing decisions stay the same as the original table.

Original Table:

Destination	Port Number
`128.32.128.0/24`	1
`128.32.129.0/24`	1
`128.32.16.0/24`	2
`128.32.17.0/24`	2
`128.32.40.1/32`	3

$\longrightarrow$

New Table:

Destination	Port Number
	1
	2
	3

Solution: Condensed Table:

Destination	Port Number
128.32.128.0/23	1
128.32.16.0/23	2
128.32.40.1/32	3

In Q5.5–Q5.8, consider using a binary trie to represent the following forwarding table:

Fixed Bits	Port Number
`0110 0010`	3
`0010 0101`	2
`0110 1110`	5

Q5.5 (2 points) How many total nodes does the resulting binary trie have?

20

Solution: Draw the tree for each of the IPs and notice that only the left side of the tree is needed. Counting the nodes should result in 20 with the root node.

Q5.6 (1 point) What is the depth of the resulting binary trie? (A trie with only a root node has depth 0.)

Ⓐ 1 Ⓑ 2 Ⓒ 4 Ⓓ 6 Ⓔ 8 Ⓕ 16 Ⓖ 24 Ⓗ 32

Solution: The depth is the number of fixed bits in a binary trie.

One trick to reduce depth is to use a *multi-bit trie*, which uses each edge to examine multiple bits at once.

The *stride* is the number of bits checked on each edge of the trie. For example, in a standard trie from lecture, the stride is 1.

Q5.7 (2 points) If we use a stride-4 trie for the above table, how many nodes are in the resulting trie?

6

Solution: Because the first row and the third row are the same, there are only two nodes at the first level, and two under `0110` and one under `0010` for a total of 6 with the root node.

Q5.8 (1 point) If we use a stride-4 trie for the above table, what is the depth of the resulting trie?

Ⓐ 1 Ⓑ 2 Ⓒ 4 Ⓓ 6 Ⓔ 8 Ⓕ 16 Ⓖ 24 Ⓗ 32

Solution: The depth of a multi-bit trie is fixed bits / stride = 8 / 4 = 2

In Q5.9–Q5.11, consider a trie for an arbitrary IPv4 forwarding table (not necessarily the table above).

Q5.9 (2 points) What is the maximum depth of a trie with stride 1?

32

Solution: The maximum depth is the maximum number of fixed bits in IPv4, 32

Q5.10 (2 points) What is the maximum depth of a trie with stride 8?

Ⓐ 1 Ⓑ 2 Ⓒ 4 Ⓓ 6 Ⓔ 8 Ⓕ 16 Ⓖ 24 Ⓗ 32

Solution: The maximum number of fixed bits / stride = 32 / 8 = 4

Q5.11 (3 points) Suppose we take an arbitrary forwarding table and build two tries: one with stride 1, and one with a larger stride. Select all true statements about the resulting two tries.

[A] The trie with larger stride could require traversing more edges.

[B] The trie with larger stride could have more nodes.

[C] In the trie with larger stride, updating a route could require updating more nodes.

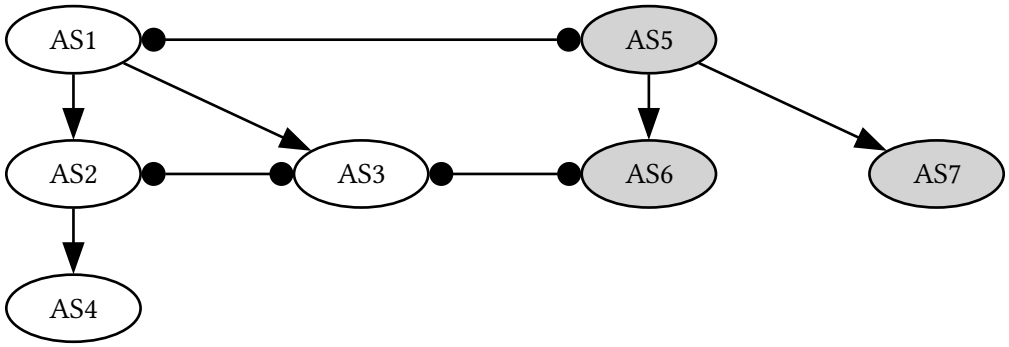
Ⓓ None of the above

Solution:

A) False, the trie with larger stride will have a smaller depth B) True, a larger stride might need prefix expansion, if prefixes are not a multiple of the stride. C) True, because of the expansion one route can occupy several nodes.

Q6 *BGP and Alliances* **(18 points)**

In Q6.1–Q6.4, consider the AS graph below:



All ASes follow Gao-Rexford import and export policies, except:
- There are two *alliances*: {AS1, AS2, AS3, AS4} are in one alliance, and {AS5, AS6, AS7} are in another.
- An AS does not export any routes to ASes in the other alliance (unless otherwise specified).

Q6.1 (2 points) AS3 sends a packet to AS4. Which of these paths does the packet take?

(A) AS3—AS2—AS4

(B) AS3—AS1—AS2—AS4

(C) AS3—AS6—AS5—AS1—AS2—AS4

(D) There is no route from AS3 to AS4.

Solution: AS4 advertises its own prefix to AS2. AS2 learns this route from a customer, so it exports it to both its allied peer AS3 and its allied provider AS1. AS1 also advertises its route to its customer AS3. AS3 prefers the peer-learned route through AS2 over the provider-learned route through AS1. The packet follows **AS3 - AS2 - AS4**.

Q6.2 (2 points) AS6 sends a packet to AS4. Which of these paths does the packet take?

(A) AS6—AS3—AS2—AS4

(B) AS6—AS5—AS1—AS2—AS4

(C) AS6—AS3—AS1—AS2—AS4

(D) There is no route from AS6 to AS4.

Solution: The only links between the alliances are AS1–AS5 and AS3–AS6. Neither link carries advertisements under the original rules. AS6 therefore never learns a route to AS4, even though physical paths connect the two ASes.

In Q6.3–Q6.4, consider the same graph as above, but AS1 modifies its export rule:
Routes from AS1's customers will be exported to AS5.

Q6.3 (2 points) AS6 sends a packet to AS4. With the modified rule, which path does the packet take?

Ⓐ AS6—AS3—AS2—AS4

Ⓒ AS6—AS3—AS1—AS2—AS4

Ⓑ AS6—AS5—AS1—AS2—AS4

Ⓓ There is no route from AS6 to AS4.

Solution: AS1 learns its route to AS4 from its customer AS2, so the exception permits AS1 to advertise this route to AS5. AS5 learns the route from its peer AS1 and may export it to its allied customer AS6. The advertisement propagates **AS4 - AS2 - AS1 - AS5 - AS6**; the packet travels in the opposite direction: **AS6 - AS5 - AS1 - AS2 - AS4**. The AS3–AS6 link still carries no advertisements.

Q6.4 (2 points) AS4 sends a packet to AS7. With the modified rule, which path does the packet take?

Ⓐ AS4—AS2—AS1—AS5—AS7

Ⓒ AS4—AS2—AS1—AS3—AS6—AS5—AS7

Ⓑ AS4—AS2—AS3—AS6—AS5—AS7

Ⓓ There is no route from AS4 to AS7.

Solution: AS7 advertises its own prefix to AS5. However, AS5 is still forbidden from advertising any route to AS1, and the AS3–AS6 link also remains blocked for advertisements. No route to AS7 reaches Alliance A. The exception permits advertisements only from AS1 to AS5; it does not create a route in the reverse direction to destinations in Alliance B.

In Q6.5–Q6.7, consider some arbitrary AS graphs (not necessarily the graph on the previous page).

- Every AS belongs to either the *red* alliance or the *blue* alliance.
- As before, an AS does not export any routes to ASes in the other alliance (unless otherwise specified).

Note: A "valid" AS graph is a graph with no cycles and a hierarchy with Tier 1 AS(es) on top.

Q6.5 (2 points) We take a valid AS graph, and randomly assign every AS to be either red or blue. Select all true statements about the resulting graph.

☐ A A red AS can send packets to any blue AS.

☐ B A blue AS can send packets to any red AS.

☐ C A blue AS can send packets to any blue AS.

☐ D A red AS can send packets to any red AS.

● E None of the above

Solution:

Choices (A) and (B):

ASes do not advertise any routes across alliance boundaries, including routes to their own prefixes.

Therefore, no AS can learn a route to a destination in the other alliance, even if a physical path exists.

Without a route to the destination, packets cannot be forwarded successfully.

Choices (C) and (D):

Belonging to the same alliance does not guarantee that a BGP route exists between two ASes.

For example, in the original topology, suppose AS4 were reassigned to Alliance B. AS4's only neighbor is AS2, which belongs to Alliance A.

Since AS2 and AS4 cannot exchange route advertisements, AS4 cannot learn routes to AS5, AS6, or AS7, despite sharing their alliance.

Conversely, under the original alliance assignment, AS5 and AS6 can reach each other using their customer-provider relationship.

Thus, reachability within an alliance is possible but not guaranteed.

Q6.6 (4 points) Consider two valid AS graphs: one with all red ASes, and one with all blue ASes.

Each graph only has a single Tier 1 AS. We add a peer link between the Tier 1 ASes.

Then, we modify the blue Tier 1 AS's export rules:
- Routes from the blue Tier 1 AS's customers will be exported to the red Tier 1 AS.

Select all true statements about the resulting graph.

[A] A red AS can send packets to any blue AS.

[B] A blue AS can send packets to any red AS.

[C] A blue AS can send packets to any blue AS.

[D] A red AS can send packets to any red AS.

(E) None of the above

Solution: The blue Tier-1 AS learns customer routes to all destinations within its network.

It advertises these routes to the red Tier-1 AS, which can then export them to its customers. The routes propagate throughout the white network.

Therefore, every red AS learns routes to all blue destinations.

However, the red Tier-1 AS exports no routes to the blue Tier-1 AS. Consequently, the blue network cannot learn any routes to red destinations.

Reachability is therefore one-way: red to blue.

Q6.7 (4 points) Consider two valid AS graphs: one with all red ASes, and one with all blue ASes.

Each graph has multiple Tier 1 ASes. We add a single peer link between one red Tier 1 AS and one blue Tier 1 AS.

Then, we modify the export rules of only these two Tier 1 ASes:
- Routes from the red AS's customers will be exported to the blue AS.
- Routes from the blue AS's customers will be exported to the red AS.

Select all true statements about the resulting graph.

[A] A red AS can send packets to any blue AS. [C] A blue AS can send packets to any blue AS.

[B] A blue AS can send packets to any red AS. [D] A red AS can send packets to any red AS.

(E) None of the above

Solution: Consider the red and blue Tier-1 ASes connected by the new peering link.

The blue Tier-1 AS learns routes to other blue Tier-1 ASes through peer relationships.

Under Gao-Rexford export policies, peer-learned routes cannot be exported to another peer.

Therefore, the connected blue Tier-1 AS cannot advertise routes to the other blue Tier-1 ASes across the new link.

The same restriction applies in the opposite direction for the red Tier-1 ASes.

Thus, even though both endpoints exchange routes, neither alliance obtains full reachability to the other.

Q7 *TCP Reliability* **(10 points)**

In Q7.1–Q7.3, consider the TCP design from lecture.

Q7.1 (2 points) Which of these best describes the reliability guarantee of TCP?

Ⓐ The receiver sees each byte exactly once.

Ⓑ The receiver provides the sent bytes in-order to the application.

Ⓒ The receiver sees every byte sent, even if the network goes down.

Ⓓ The receiver sees TCP segments in the order they were sent.

Solution: TCP provides the application with a reliable, in-order byte stream. The underlying network may lose, duplicate, delay, or reorder packets; TCP hides these effects from the application.

Q7.2 (2 points) Which of these best describes why TCP uses sequence numbers?

Ⓐ To detect payload corruption.

Ⓑ To act as a retransmission timer.

Ⓒ To detect duplicates and reorder data.

Ⓓ To avoid packets traveling in a loop.

Solution: Sequence numbers identify where data belongs in the byte stream. This allows TCP to recognize duplicate data and correctly handle data that arrives out of order.

Q7.3 (2 points) A packet is transmitted, but the ACK is lost. Which of these best describes what happens?

Ⓐ The receiver times out and resends ACK.

Ⓑ The sender retransmits the packet.

Ⓒ The entire TCP connection fails.

Ⓓ The receiver sends a negative ACK.

Solution: Because the sender does not receive the ACK, its retransmission timer may expire and cause it to retransmit the segment. Sequence numbers allow the receiver to recognize that the retransmitted bytes are duplicates.

In Q7.4–Q7.7, Alice sends data to Bob. Each segment contains 4 bytes of data.

Alice sends the following three segments, in order:

Segment	Sequence number	Result
S1	100	Received by Bob
S2	104	Lost in the network
S3	108	Received by Bob

Each subpart continues from the previous subparts.

Q7.4 (1 point) Bob receives S1. Which ACK number should he send?

Ⓐ ACK = 100 Ⓑ ACK = 104 Ⓒ ACK = 108 Ⓓ ACK = 112

Solution: The segment contains bytes 100 through 103, so the next byte Bob is waiting for is byte 104.

Q7.5 (1 point) Bob receives S3. Which ACK number should he send?

Ⓐ ACK = 104 Ⓑ ACK = 107 Ⓒ ACK = 108 Ⓓ ACK = 112

Solution: Bob is still missing bytes 104 through 107. Since the ACK indicates the next byte Bob is waiting for, Bob sends ACK = 104 even though he has received the later bytes.

Q7.6 (1 point) After Bob receives S1 and S3, which bytes can be delivered to the application?

Ⓐ Bytes 100–111

Ⓑ Bytes 100–103

Ⓒ Bytes 100–103 and 108–111

Ⓓ Bytes 108–111

Solution: TCP presents an in-order byte stream to the application. Bob can therefore deliver bytes 100 through 103, but the gap at bytes 104 through 107 prevents him from delivering the later bytes. He can buffer bytes 108 through 111 until the gap is filled.

Q7.7 (1 point) Alice retransmits S2, and Bob receives S2. Which ACK number should he send?

Ⓐ ACK = 104 Ⓑ ACK = 108 Ⓒ ACK = 112 Ⓓ ACK = 116

Solution: Bob now has bytes 100 through 111 with no gaps. The next byte he is waiting for is therefore byte 112.

Comment Box

Congrats for making it to the end of the exam! Leave any thoughts, comments, feedback, or doodles here.

Nothing in the comment box will affect your grade.

Ambiguities

If you feel like there was an ambiguity on the exam, you can put it in the box below.

For ambiguities, you must qualify your answer and provide an answer for both interpretations. For example, "if the question is asking about A, then my answer is X, but if the question is asking about B, then my answer is Y." You will only receive credit if it is a genuine ambiguity and both of your answers are correct. We will only look at this box if you request a regrade.