CS 168 Summer 2025

Introduction to the Internet Midterm

Solutions last updated: July 28, 2025

PRINT Your Name: ____________________

PRINT Your Student ID: ____________________

You have 110 minutes. There are 6 questions of varying credit. (100 points total)

Question:	1	2	3	4	5	6	Total
Points:	14	21	16	15	16	18	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)

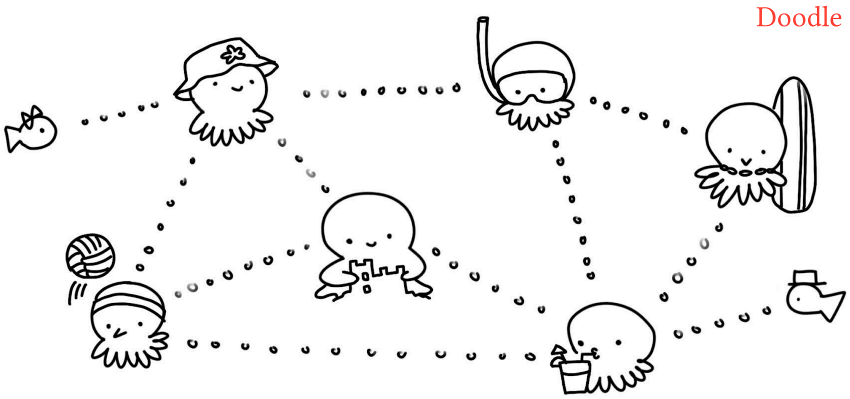
Anything you write outside the answer boxes or you ~~cross out~~ will not be graded. If you write multiple answers, your answer is ambiguous, or the bubble/checkbox is not entirely filled in, we will grade the worst interpretation.

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: Andrea Lou



Q1 *Potpourri* **(14 points)**

Q1.1 (2 points) Which of the following layers must be implemented in routers?

- [x] Layer 1: Physical
- [x] Layer 3: Internet
- [ ] Layer 7: Application
- [x] Layer 2: Link
- [ ] Layer 4: Transport
- ( ) None of the above

Solution: Only the physical layer, link layer and IP layer must be implemented in routers.

Q1.2 (1 point) Reliability is only implemented in routers, not end hosts.

- ( ) True
- (●) False

Solution: False. The end-to-end principle says that end hosts alone are sufficient to guarantee reliability. Routers may, but do not need to implement reliability.

Q1.3 (2 points) Which of the following is the "narrow waist" protocol?

- ( ) WiFi
- ( ) BGP
- ( ) HTTP
- ( ) TCP
- (●) IP
- ( ) None of the above

Solution: The narrow waist protocol is the IP layer since everyone operating on the Internet must speak IP.

Q1.4 (2 points) What does separating systems into layers directly help with?

- [ ] Reliability
- [x] Modularity
- [ ] Transfer Speed
- [x] Abstraction
- [ ] Addressing
- ( ) None of the above

Solution: Layering provides modularity and abstractions. Reliability and addressing need to be handled within a layer, it is not provided by layers. Transfer speed might actually be hurt by layering since it adds overhead.

Q1.5 (1 point) Circuit switching gives applications a straightforward abstraction to guarantee bandwidth.

- (●) True
- ( ) False
- ( ) Not enough information

Solution: True. Circuit switching allows applications to reserve bandwidth to match their needs.

Q1.6 (2 points) When a layer 4 protocol receives a packet, what can be used to demultiplex?

- ☐ Physical port
- ☐ IPv4 header
- ■ Layer 4 header
- ■ Logical port
- ☐ ICMP header
- ☐ Layer 3 header

Solution: A layer four protocol such as TCP/UDP uses logical ports to demultiplex and send data to an application. Logical ports are located in the Layer 4 header.

Q1.7 (1 point) What is the packet delay for a 200-bit packet between two nodes connected by a link with bandwidth 100 Kbps, and a propagation delay of 1 second?

- ● 1.002 seconds
- ○ 1.0002 seconds
- ○ 1 second
- ○ 0.002 seconds
- ○ 1.02 seconds
- ○ 0.02 seconds

Solution: (Transmission delay = 200 bits x (1s / 100000 bits)) = 0.002 seconds

Propagation delay = 1 second

This sums to 0.002 + 1 = 1.002 seconds.

Q1.8 (1 point) In the traceroute project, when a traceroute program receives an ICMP response with type "Destination Unreachable", it is from a router along the path.

- ○ True
- ● False
- ○ Not enough information

Solution: False. "Destination Unreachable" comes from the final destination. A router along the way should return "Time Exceeded" because the TTL would've expired.

Q1.9 (1 point) A router changes the TTL on a data packet from 1 to 0. What type of traffic is this?

- ○ User traffic
- ○ Control Plane traffic
- ● Punt traffic

Solution: When the TTL changes from 1 to 0 there has been a timeout, so the packet should be punted to the CPU for error handling.

Q1.10 (1 point) Longest prefix matching is typically implemented in software on a router's linecard.

- ○ True
- ● False
- ○ Not enough information

Solution: False. LPM is usually implemented in hardware due to the fact that hardware is much faster than software (albeit less flexible).

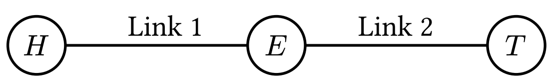
Q2 *Pipes: EvanBot Strikes Back* (21 points)

EvanBot is a Galactic Empire™ engineer who uses a network connecting three planets.
EvanBot sends data from Hoth (H), through Endor (E), to Tatooine (T).

For the entire question: EvanBot is the only user on the network. E has infinite queuing capacity, and E processes its queue in FIFO order.

Topology and setup for **Q2.1 to Q2.6** only:

- Starting at $t = 0$, EvanBot sends 10 payloads, one after the other, from H to T. Each payload is 20 Gbits.
- E must receive all bits of a payload before sending that payload along the next link.



	Link 1	Link 2
Propagation Delay	2 sec	10 sec
Bandwidth	20 Gbps	10 Gbps

Q2.1 (1 point) What is the queuing delay of the first payload?

Note: For all queuing delay questions, count from the time the last bit of the payload arrives at E, to the time the first bit of the payload is sent out of E.

0 sec

Q2.2 (1 point) At what time does T receive the last bit of the first payload?

$t = 15$ sec

Q2.3 (2 points) What is the queuing delay of the second payload?

1 sec

Q2.4 (2 points) At what time does T receive the last bit of the second payload?

$t = 17$ sec

Q2.5 (2 points) What is the queuing delay of the 10th payload?

9 sec

Q2.6 (2 points) At what time does T receive the last bit of the 10th payload?

$t = 33$ sec

Solution: Queuing delay questions:

This table shows the time each payload arrives at E (time of last bit arriving), and the time each payload is sent from E (time of first bit sent out).

Payload number	Time of arrival at E	Time of sending from E	Queuing delay of payload
1	3	3	0
2	4	5	1
3	5	7	2
4	6	9	3
5	7	11	4
6	8	13	5
7	9	15	6
8	10	17	7
9	11	19	8
10	12	21	9

To derive the arrival column:

- The first bit of the first payload is transmitted at H at $t = 0$.
- The payload is 20 Gbit, and the bandwidth of Link 1 is 20 Gbps, so it takes 1 second to transmit the payload. The last bit of the first payload is transmitted at H at $t = 1$.
- Link 1 has a 2-second propagation delay, so the last bit of the first payload arrives at E at $t = 3$.
- At this point, a constant stream of payloads is sent from H to E. Each payload takes 1 second to transmit, which is why the times of arrivals are $3, 4, 5, \ldots$ (1 second per packet).

To derive the sending column:

- Payload 1's last bit arrives at $t = 3$, so its first bit can be forwarded starting at $t = 3$.
- The payload is 20 Gbit, and the bandwidth of Link 2 is 10 Gbps, so it takes 2 seconds to transmit the payload.
- This is why the times of sending are $3, 5, 7, \ldots$ (2 seconds per packet).

The queuing delay is the difference between the time of last bit arriving, and the time of first bit sending.

Solution: Time of arrival questions:

This table shows the time each payload arrives at E (time of last bit arriving), and the time each payload is sent from E (time of first bit sent out).

Payload number	Time of arrival at E	Time of sending from E	Time of arrival at T
1	3	3	15
2	4	5	17
3	5	7	19
4	6	9	21
5	7	11	23
6	8	13	25
7	9	15	27
8	10	17	29
9	11	19	31
10	12	21	33

To derive the arrival at E and sending from E columns, see the previous solution.

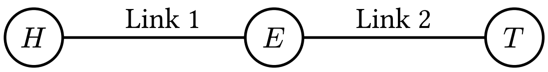
When the first bit of a payload is sent out from E, the last bit gets sent out 2 seconds later. This is because Link 2 has bandwidth 10 Gbps, and the payload is 20 Gbits.

The last bit then takes another 10 seconds to arrive at T. This is the propagation delay of Link 2.

Thus, the time the last bit of a payload arrives at T is 12 seconds after the time the first bit is sent out of E.

Topology and setup for **Q2.7 to Q2.9** only:

- Starting at $t = 0$, EvanBot sends an endless stream of data, at constant rate 20 Gbps, from H to T.
- At E, each bit can be forwarded independently of other bits.



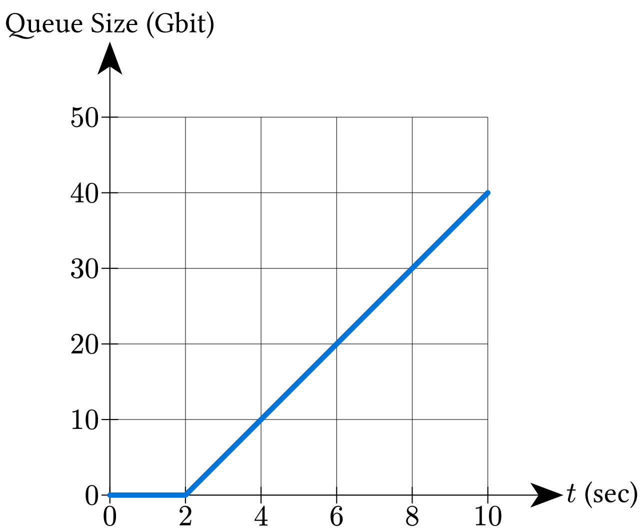
	Link 1	**Link 2**
Propagation Delay	2 sec	10 sec
Bandwidth	20 Gbps	*see subparts*

Q2.7 (4 points) For this subpart, Link 2 has a constant bandwidth of 15 Gbps.

What are the queue sizes at the following times?

Time	Queue Size
2 sec	Gbits
4 sec	Gbits
6 sec	Gbits
8 sec	Gbits
10 sec	Gbits

Solution:



The first bit is sent at $t = 0$, and arrives at E at $t = 2$. This is the 2-second propagation delay of Link 1.

This means that from $t = 0$ to $t = 2$, no data arrives at E, so there is no queue.

Starting at $t = 2$, a steady stream of 20 Gbps is arriving at E, and a steady stream of 15 Gbps is leaving E.

This means that every second, 20 Gbits enter and 15 Gbits leave. This leaves $20 - 15 = 5$ Gbits, which get added to the queue in every second.

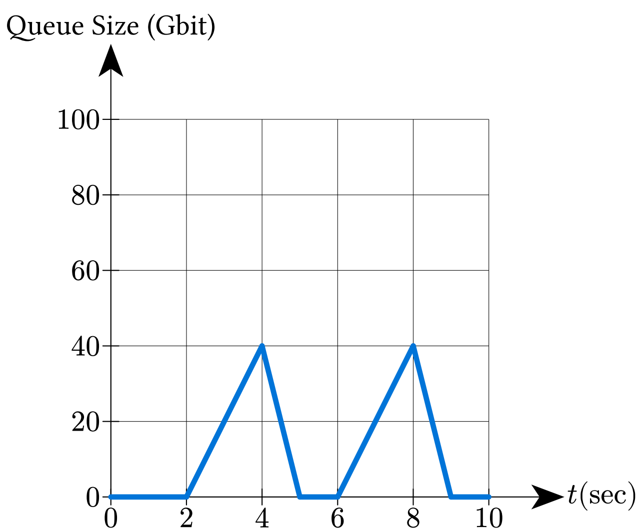
Q2.8 (4 points) For this subpart, the bandwidth of Link 2 alternates forever between 60 and 0 Gbps, changing every 2 seconds:

- From $t = 0$ to $t = 2$, the bandwidth is 60 Gbps.
- From $t = 2$ to $t = 4$, the bandwidth is 0 Gbps, and so on.

What are the queue sizes at the following times?

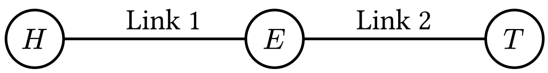
Time	Queue Size
2 sec	Gbits
4 sec	Gbits
6 sec	Gbits
8 sec	Gbits
10 sec	Gbits

Solution:



Topology and setup for **Q2.7 to Q2.9**, reprinted:

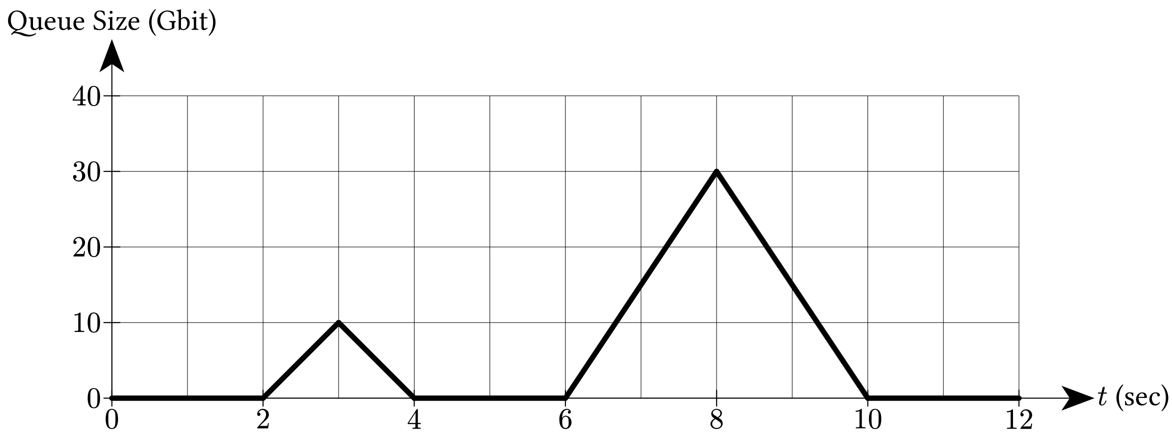
- Starting at $t = 0$, EvanBot sends an endless stream of data, at constant rate 20 Gbps, from H to T.
- At E, each bit can be forwarded independently of other bits.



	Link 1	**Link 2**
Propagation Delay	2 sec	10 sec
Bandwidth	20 Gbps	*see subparts*

Q2.9 (3 points) For this subpart, the bandwidth of Link 2 now changes in an unknown pattern.

The graph below shows the size of E's queue changing over time.



Use the graph to determine the bandwidth of Link 2 at the following times.

Write your answer as a range. Examples:

- If the bandwidth must be exactly 3 Gbps, write `[3, 3]`.
- If the exact bandwidth cannot be determined, but must be at least 3 Gbps, write `[3, ∞]`.
- If there is not enough information to determine any range, write `[0, ∞]`.

Time	Bandwidth of Link 2
5 sec	Gbps
7 sec	Gbps
9 sec	Gbps

Solution: If the queue is empty, the bandwidth must be at least 20Gbps to keep up with the incoming stream. At t=7, since the queue is increasing with a rate of 30Gbit/2sec=15Gbps the second link is only handling 5Gbps of the 20Gbps. At t=9, the queue is draining at a rate of 15Gbps, so the link is able to handle 15+20Gbps = 35Gbps.

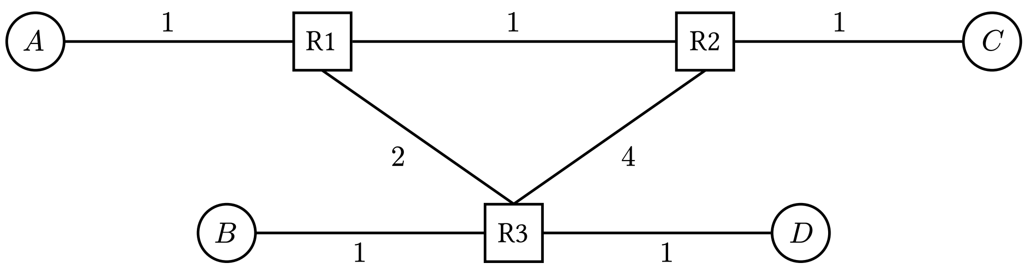
Time	Bandwidth of Link 2
5 sec	$[20, \infty)$ Gbps
7 sec	$[5, 5]$ Gbps
9 sec	$[35, 35]$ Gbps

Q3 *Distance-Vector: Soda Shenanigans* **(16 points)**

Alice (A), Bob (B), Connie (C), and Diego (D) are connected to the Soda Hall network, which runs the distance-vector algorithm from lecture.

Assumptions:

- Static routes are installed at time $t = 0$.
- Routers send periodic advertisements every 2 seconds, starting at $t = 0$.
- Routing table entries expire after 10 seconds of receiving no advertisements.
- Every second, each router (1) expires routes, then (2) processes advertisements and updates its table, then (3) sends out advertisements if t is even.
- Link costs correspond to packet travel times (in seconds). Ignore processing and queuing delays.
- Poison reverse, split horizon, and route poisoning are disabled.



Q3.1 (4 points) Fill in R1's table at steady state. If a host is directly connected, the next hop is "Direct".

Dest.	Next Hop	Cost
A		
B		

Dest.	Next Hop	Cost
C		
D		

Solution: A is directly connected to R1 with a cost of 1. From R1 to B, there are two paths via R2 with cost 6 and via R3 with cost 3. From R1 to C, there are two paths via R2 with cost 2 and via R3 with cost 7. From R1 to D, there are two path via R3 with cost 3 and via R2 with cost 6.

R1		
To	**Next Hop Router**	**Cost**
A	Direct	1
B	R3	3
C	R2	2
D	R3	3

Q3.2 (3 points) Which topology changes could cause a routing loop? Consider each choice independently.

- ■ *A* leaves.
- ■ Link R1—R3 breaks.
- ■ Link B—R3 breaks.
- ■ Links R1—R2 and R2—R3 both break.

Solution: If Alice *A* leaves, R1's route to A be deleted and will be replaced by a longer route from R2. R2 sends packets bound for A to R1 and R1 sends packets bound for A to R2, a loop! You would get the same behavior if Link B-R3 breaks.

If Link R1-R3 breaks, R1 will expire routes with next hop R3 to B and D and R3 will expire routes with next hop R1 to A and C. They will then accept advertisements to replace these routes from R2. Since R2 had routes to B and D through R1, R1 sends traffic to B and D through R2 and R2 sends traffic to B and D through R1, a loop!

If Link R1-R2 and R2-R3 breaks, R1 will expire the route with next hop R2 to C and R2 will expire all routes expect the direct connection to C, since it is now cut off from the topology. R1 will then accept a false route to C from R3 through itself. R1 sends traffic to C through R3 and R3 sends traffic to C through R1, a loop!

Q3.3 (3 points) What distance-vector optimizations could help deal with at least one of the routing loop(s) from Q3.2? Consider each choice independently.

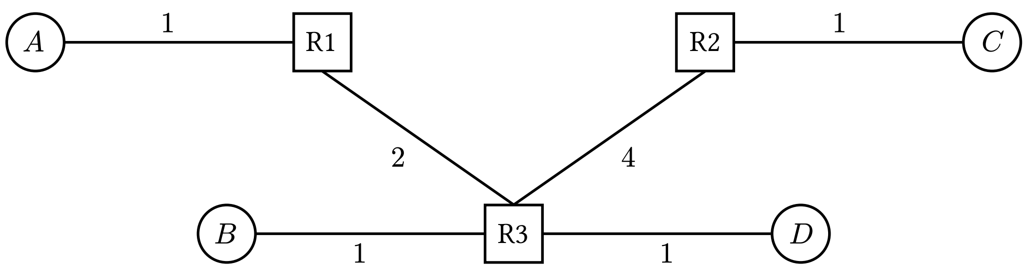
- [x] Poison Expired Routes
- [x] Poison Reverse
- [ ] Eventful Updates
- [x] Split Horizon
- [x] Count to Infinity
- ◯ None of the above

Solution: Poison Expired Routes can help convergence occur faster, reducing the amount of time a loop is present. Split horizon helps prevent loops between two routers by not advertising back to where a route came from. Poison reverse helps prevent loops between two routers by advertising poison back to where a route came from. This fixes the loop if Link R1-R3 breaks and if Link R1-R2 and R2-R3 breaks. Count to infinity prevents routing loops by stopping routes with costs that increment higher and higher. It does this by setting the cost to infinity when larger than a certain number (commonly set to 16). Count to infinity fixes loops between R1,R2,R3 when end-hosts leave.

After the network converges, Connie unplugs a mysterious cable and presses a mysterious button. As a result, **the R1—R2 link is removed**, and **split horizon is enabled** on all routers.

The assumptions and topology are reprinted below, with Connie's changes:

- Static routes are installed at time $t = 0$.
- Routers send periodic advertisements every 2 seconds, starting at $t = 0$.
- Routing table entries expire after 10 seconds of receiving no advertisements.
- Every second, each router (1) expires routes, then (2) processes advertisements and updates its table, then (3) sends out advertisements if t is even.
- Link costs correspond to packet travel times (in seconds). Ignore processing and queuing delays.
- Poison reverse and route poisoning are disabled. Split horizon is enabled.



Q3.4 (4 points) Fill in R1's table at the new steady state (after Connie's changes).

Dest.	Next Hop	Cost
A		
B		

Dest.	Next Hop	Cost
C		
D		

Solution:

A is directly connected to R1 with a cost of 1. From R1 to B, there is now only one path via R3 with cost 3. From R1 to C, there is now only one path via R3 with cost 7. From R1 to D, there is now only one path via R3 with cost 3.

R1		
To	**Next Hop Router**	**Cost**
A	Direct	1
B	R3	3
C	R3	7
D	R3	3

Q3.5 (2 points) Suppose R1's original table entry for destination C expires at $t = 20$.

At what time step does R1's table reach the new steady state in Q3.4?

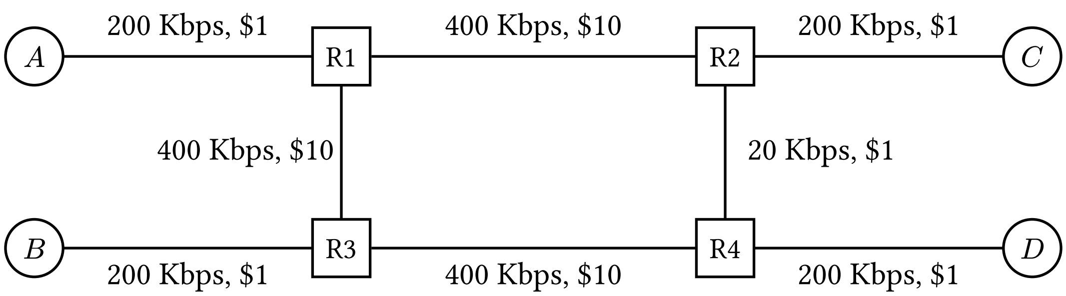
$t = 32$

Solution: The route to C is the only route that changes for R1. R1's route for C expires at t=20, this means that the last possible advertisement was sent from R1 to R3 at t=18. It reaches R3 after 2 seconds at t=20. R3's route to C will then expires at t=30 at the earliest when it will accepts new route from R2. Then R3 advertises back to R1 (2 seconds) to reach steady state at t=32.

Q4 *Link-State: Config Chaos* **(15 points)**

You are the first network engineer at a small company, and no one knows how the routers are set up. They ask you to check it out.

You notice that links in the network are labeled with different bandwidths and dollar amounts. This is shown in the topology with labels (in the form "bandwidth, dollar amount"):



All the routers are running a link-state protocol. All the routers learn the complete global topology, but do not know what algorithms other routers are running.

Q4.1 (2 points) In total, how many initial "hello" advertisements get sent between the routers? Do not count periodically re-sent advertisements or advertisements flooded on behalf of another router.

Solution: R1->R2 & R4, R2-> R1 & R4, R3-> R1 & R4, R4 -> R3 & R2 so 8 in total

For subparts Q4.2 to Q4.5, you notice that:

- R1, R3, and R4 optimize for paths with the **lowest** total dollar amount (i.e., sum of dollars along path).
- R2 optimizes for paths with the **highest** bottleneck bandwidth (bottleneck bandwidth is the smallest bandwidth on the path).

Q4.2 (1 point) Does traffic from A to C reach its destination?

- ● Yes
- ○ No
- ○ Not enough information

Q4.3 (1 point) Does traffic from A to D reach its destination?

- ○ Yes
- ● No
- ○ Not enough information

Q4.4 (1 point) Does traffic from B to C reach its destination?

- ● Yes
- ○ No
- ○ Not enough information

Solution: Yes - Traffic from A to C would get routed from R1 to R2 since $11 < 22$, then to C. No - Traffic from A to D would get routed from R1 to R2 since $12 < 21$, then R2 would send back to R1 because $20 < 200$, a loop. Yes - Traffic from B to C would be routed from R3 to R4 since $12 < 21$ and from R4 to R2, then to C.

Q4.5 (4 points) Does this topology converge to a valid routing state?

○ Yes ● No ○ Not enough information

Why or why not? If there are any loops or dead ends, list at least one and explain why it occurs. Answer in 20 words or fewer.

Solution: This is not a valid routing state since there is a loop between R1 and R2 when routing from A to D. Traffic from C to D also results in a loop between R1 and R2. Occurs since routers are optimizing for different things.

Q4.6 (4 points) Now you reprogram all routers to use the lowest bottleneck bandwidth as a cost metric. You notice that R1, R3, and R4 use Dijkstra's algorithm and R2 uses the Bellman-Ford algorithm.

Note: Both these algorithms compute the optimal shortest path. You do not need any more information about either algorithm to answer this question.

Does this topology converge to a valid routing state?

- (●) Yes
- ( ) No
- ( ) Not enough information

Why or why not? Answer in 20 words or fewer.

Solution: Yes, it convergences. Either algorithm can be used as long as the optimal shortest path is found for the same cost metric.

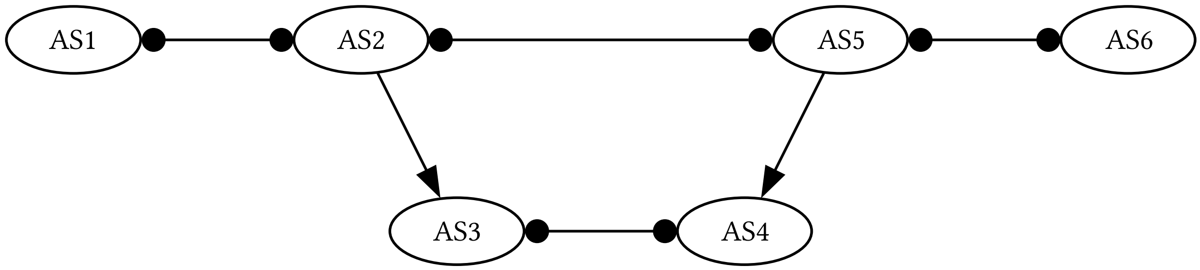
Q4.7 (2 points) Which features of the link-state protocol will prevent infinite flooding? Consider each choice independently.

- [ ] Routers send poison advertisements.
- [ ] Each table entry has a TTL field, and the entry is deleted when TTL reaches 0.
- [ ] Advertisements are periodically re-sent.
- [x] Packets contain timestamps.
- ( ) None of the above

Solution: Timestamps are used to prevent flooding since routers can use them to know if they have already received that advertisement.

Q5 *BGP: Follow the Money* **(16 points)**

Consider the AS graph below, where each AS follows the Gao-Rexford import and export policies.



For each source/destination pair, select whether it is possible for packets to be sent from the source AS to the destination AS. In other words, is there an AS path from source to destination where all intermediate ASes agree to export the path?

Q5.1 (1 point) Source AS1, destination AS3.

● Possible ○ Not possible

Solution:

There are two possible paths:

AS1 → AS2 → AS3: This path works because AS2 makes money from its customer (AS3). In other words, AS3 will advertise to AS2, and AS2 will advertise to AS1, creating the path.

AS1 → AS2 → AS5 → AS4 → AS3: This path doesn't work because AS2 will not agree to participate (both neighbors are peers).

Since we have at least one valid path, it's possible to send packets from AS1 to AS3.

Q5.2 (1 point) Source AS1, destination AS4.

○ Possible ● Not possible

Solution:

There are two possible paths:

AS1 → AS2 → AS3 → AS4: This path doesn't work because AS3 will not agree to participate (neither neighbor is a customer).

AS1 → AS2 → AS5 → AS4: This path doesn't work because AS2 will not agree to participate (both neighbors are peers).

Since none of the paths are valid, it's not possible to send packets from AS1 to AS4.

Q5.3 (1 point) Source AS2, destination AS4.

- (●) Possible
- ( ) Not possible

Solution:

There are two possible paths:

AS2 → AS3 → AS4: This path doesn't work because AS3 will not agree to participate (neither neighbor is a customer).

AS2 → AS5 → AS4: This path works because AS5 makes money from its customer (AS4). In other words, AS4 will advertise to AS5, and AS5 will advertise to AS2, creating the path.

Since we have at least one valid path, it's possible to send packets from AS2 to AS4.

Reachability is not guaranteed in this AS graph. In other words, for some source/destination pairs, no AS path exists where all intermediate ASes agree to export the path.

Q5.4 (2 points) Why is reachability not guaranteed in this AS graph?

- ( ) Because the graph has a loop (AS2–AS3–AS4–AS5–AS2).
- (●) Because the graph does not have a set of Tier 1 ASes all connected to each other.
- ( ) Because AS1 and AS6 have no customers, and all Tier 1 ASes must have a customer.
- ( ) Because it is always impossible to send packets to AS6.

Solution:

(A) is false. In AS graphs, a loop creates an invalid graph when there are directed edges forming a loop. However, some of the edges in AS2–AS3–AS4–AS5–AS2 are not directed, so this example does not make the graph invalid. In other words, removing this loop (e.g. by removing the AS2–AS5 link) would not guarantee reachability.

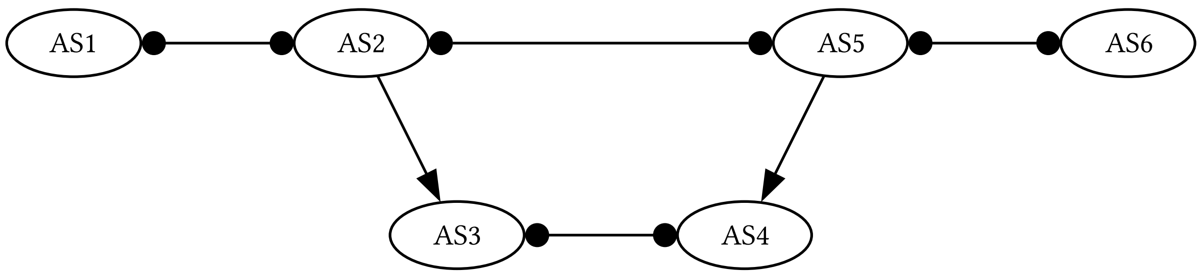
(B) is true. One of the requirements for reachability is for each Tier 1 AS (an AS with no provider) to be connected to every other Tier 1 AS. However, in this graph, we don't have this requirement, e.g. AS1 is not connected to AS6.

(C) is false. A Tier 1 AS is not required to have a customer.

(D) is false. For example, AS4 can send packets to AS6 using the AS4–AS5–AS6 path.

Q5.5 (3 points) On the graph below, draw at most 3 extra links, such that the resulting AS graph provides reachability.

Use dots and arrows to clearly label your added links. You may not change existing links, and you may not add multiple links between the same two ASes.



Solution:

The simplest solution is to connect all the Tier 1 ASes to every other Tier 1 AS:

1. Peering link from AS1–AS5
2. Peering link from AS1–AS6
3. Peering link from AS2–AS5

Alternate solutions may exist, although they would not be able to use the proof from lecture to prove reachability. For these alternate solutions, you would have to manually check all pairs of ASes to ensure that there is a valid path between every pair of ASes.

For the rest of the question, consider the original graph again, without any of the extra links you drew.

In each subpart, some ASes' policies are modified. Select whether the AS graph provides reachability under the modified policies. Each subpart is independent.

Q5.6 (2 points) AS3 imports all received paths (from customers/peers/providers), and exports paths to everybody.

○ Reachable ● Not reachable

Solution:

There is no path from AS1 to AS6.

AS1—AS2—AS5—AS6 is invalid because of the peering links: AS2 and AS5 both would not participate since they wouldn't earn money in the exchange.

AS1—AS2—AS3—AS4—AS5—AS6 is invalid because AS4 would not agree to participate. AS4 would not advertise the path (AS4—AS5—AS6) to AS3 because AS3 is AS4′s peer, and following Gao-Rexford rules, AS4 wouldn't advertise a route that came from its provider to a peer. Therefore, AS3 would never have the chance to import a route via AS4 to AS6 since AS4 never exports AS4 -> AS5 -> AS6 to AS3.

Q5.7 (2 points) AS3 and AS4 both import all received paths, and export paths to everybody.

● Reachable ○ Not reachable

Solution:

AS1—AS2—AS3—AS4—AS5—AS6 is valid (since AS4 exports paths to everybody, we don't run into the same situation as in the previous subpart).

Therefore, we also know that AS1 can reach AS5 (AS1—AS2—AS3—AS4—AS5) and AS2 can reach AS6 (AS2—AS3—AS4—AS5—AS6). Using the same strategy, we can extrapolate to all other sender-destination pairs to show that they're valid.

Q5.8 (2 points) AS2 and AS5 both import all received paths, and export paths to everybody.

● Reachable ○ Not reachable

Solution:

AS1—AS2—AS5—AS6 is valid because now AS5 will export paths to everybody (including its peer, AS2). AS2 imports paths from everybody (including its peer AS5). Therefore, AS5 exports AS5—AS6 to AS2, and AS2 exports this route.

We can extrapolate and show that AS1 to AS5 as well as AS2 to AS6 are valid (by using a subset of the path defined above).

Routes to AS3 are also valid because they need to get to AS2, who then forwards the packet on to AS3 (again, the importing and exporting of routes via the AS2 – AS5 link allow this to happen).

A symmetric argument for routes to AS4 being valid applies (here, AS5 forwards the packets on to AS4).

Q5.9 (2 points) AS2 and AS5 both use this modified policy:

- For paths from customers, export to everybody (unchanged from Gao-Rexford).
- For paths from peers, export to peers only.
- For paths from providers, export to customers only (unchanged from Gao-Rexford).

○ Reachable ● Not reachable

Solution: Since a path from peers will only be exported to peers, A5 and AS2 will not export routes to AS4 or AS3.

Q6 *Routers: Don't Trie Me, Patricia* **(18 points)**

Consider a router running longest prefix matching to forward packets.

Q6.1 (6 points) Fill in the new table below, such that both tables produce the same forwarding decisions, and every IPv4 address matches only one prefix. Write one IP prefix per box.

Original Table:

Destination	Port Number
128.1.0.0/24	1
128.1.1.0/24	2
128.1.2.0/24	2
128.1.3.0/24	3

→

New Table:

Destination	Port Number
	1
	2
	3

Solution:

First row: We can still allow for IP prefix 128.1.0.0/24 to map to Port 1 like usual.

Second row: We want to use a prefix that effectively allows for both IP prefixes 128.1.1.0/24 and 128.1.2.0/24 to map to port 2. We see that bits 17 to 24 (starting from the left) for both prefixes are 00000001 and 0000010, respectively. Therefore, bits 1 to 22 are the complete same for the two IP addresses, meaning we can have the IP prefix 128.1.0.0/22.

Third row: Similar to port 1, we can still allow for IP prefix 128.1.3.0/24 to map to Port 3 like usual.

Now, consider using binary tries to run longest prefix matching.

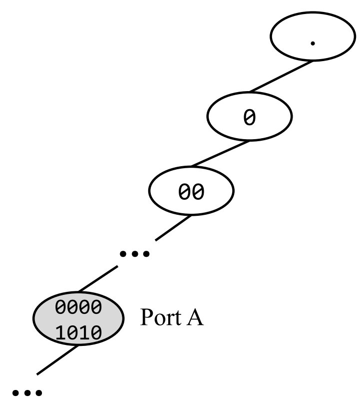
Q6.2 (2 points) Consider building a binary trie out of a forwarding table with these three prefixes:

17.0.0.0/8 17.1.0.0/16 17.1.1.0/24

What is the height of the resulting binary trie?

○ 3 ○ 8 ○ 16 ● 24 ○ 32

Solution: We first have the no-bit node, and then similar to the last problem, if our trie only had the valid port nodes, it would take at least 8 more nodes to reach the first port number, and at least 16 more nodes for the next two port numbers.



Therefore, you would need at least 25 nodes in order to construct this regular binary trie for this forwarding table.

Q6.3 (2 points) What is the maximum height that *any* binary trie can have for IPv4 addresses?

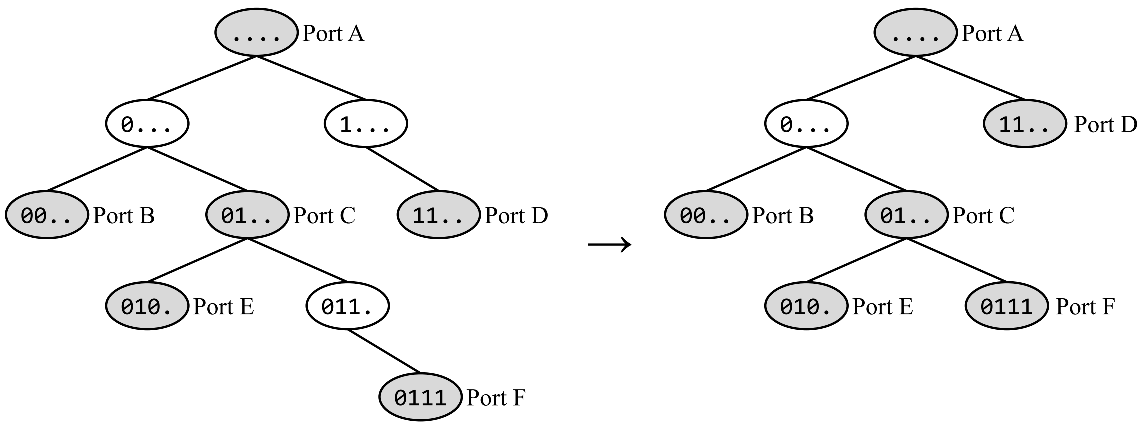
○ 3 ○ 8 ○ 16 ○ 24 ● 32

Solution: Since IPv4 addresses are 32 bits long, and each level represents 1 bit of the address in a binary trie, the maximum amount of levels a regular binary trie can have is 32.

One downside of binary tries is that they can get too tall. To fix this, we will design a new data structure.

In a *Patricia trie*, all nodes that are not assigned a port (colored white below) with a single parent and a single child are compressed.

Example of converting a binary trie into its corresponding Patricia trie:



Binary trie Patricia trie

Q6.4 (1 point) What part of the router is responsible for building Patricia tries?

○ Data Plane ● Control Plane ○ Management Plane

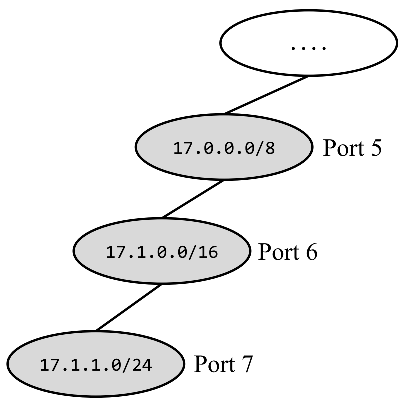
> **Solution:** Since the control plane is in charge of performing routing protocols, it will in turn be responsible for building this Patricia trie for a router.

Q6.5 (5 points) In the box below, draw the Patricia trie for the forwarding table below.

Your diagram should look similar to the tries on the previous page. You can skip the shading, and you can write IP prefixes in the nodes (instead of bitstrings).

Destination	Port Number
17.0.0.0/8	5
17.1.0.0/16	6
17.1.1.0/24	7

Solution:



As we can see, the corresponding Patricia trie only requires four nodes instead of the 24 nodes from the regular binary trie!

Q6.6 (2 points) Select all true statements.

- [x] A Patricia trie could look exactly the same as its corresponding binary trie in some cases.
- [ ] A Patricia trie could have more nodes than its corresponding binary trie in some cases.
- [x] Inserting a node for a Patricia trie can be more computationally intense than for a binary trie.
- [x] A Patricia trie can directly connect the default route root node with a node with 32 bits fixed.
- ○ None of the above

Solution:

(A) is true. For instance, if there was only one node within the trie, then the Patricia trie and the corresponding binary trie would be the exact same since there are no nodes to compress.

(B) is false. Since Patricia tries are always attempting to compress nodes, they will always have at most the number of nodes in their corresponding binary tries.

(C) is true. Unlike binary tries where nodes are added in a "top down" perspective, one can insert nodes in a Patricia trie anywhere, and therefore the trie may need to be restructured so that the Patricia trie's compression rule still holds.

(D) is true. Say our forwarding table is such that matching to no bits means sending to Port A, and matching to exactly 17.17.17.17 means sending to port B. In that case, we'll have two nodes, one for a no-bit match, and one for a 32-bit match, with an edge going from the no-bit match to the 32-bit match. It's worth realizing that the corresponding binary trie will have 33 nodes.

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.