CS 168 — Introduction to the Internet

Summer 2025 — Midterm

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)

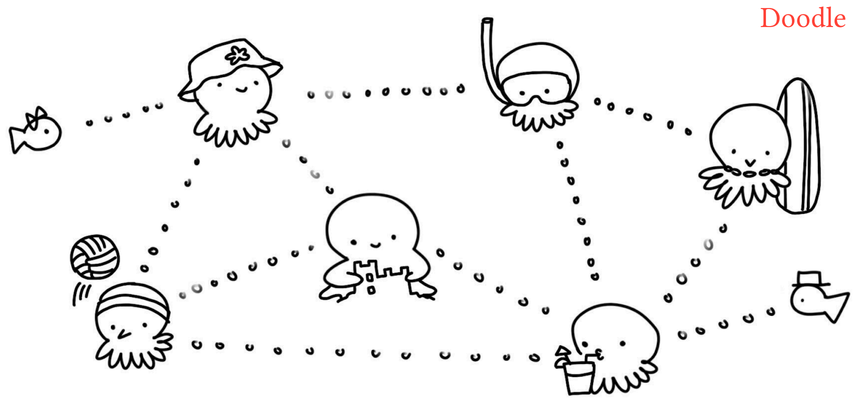
Anything you write outside the answer boxes or you ~~cross out~~ will not be graded. If you write multiple answers, your answer is ambiguous, or the bubble/checkbox is not entirely filled in, we will grade the worst interpretation.

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



This content is protected and may not be shared, uploaded, or distributed.

Q1 *Potpourri* (14 points)

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

- ☐ Layer 1: Physical
- ☐ Layer 3: Internet
- ☐ Layer 7: Application
- ☐ Layer 2: Link
- ☐ Layer 4: Transport
- ○ None of the above

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

- ○ True
- ○ False

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

- ○ WiFi
- ○ BGP
- ○ HTTP
- ○ TCP
- ○ IP
- ○ None of the above

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

- ☐ Reliability
- ☐ Modularity
- ☐ Transfer Speed
- ☐ Abstraction
- ☐ Addressing
- ○ None of the above

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

- ○ True
- ○ False
- ○ Not enough information

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

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

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

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

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

- ○ True
- ○ False
- ○ Not enough information

This content is protected and may not be shared, uploaded, or distributed.

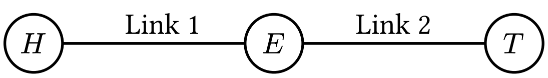
Q2 *Pipes: EvanBot Strikes Back* **(21 points)**

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.

sec

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

$t =$ sec

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

sec

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

$t =$ sec

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

sec

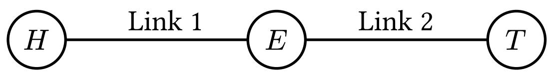
Q2.6 (2 points) At what time does T receive the last bit of the 10th payload?

$t =$ sec

This content is protected and may not be shared, uploaded, or distributed.

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

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.

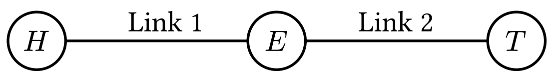
What are the queue sizes at the following times?

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

This content is protected and may not be shared, uploaded, or distributed.

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

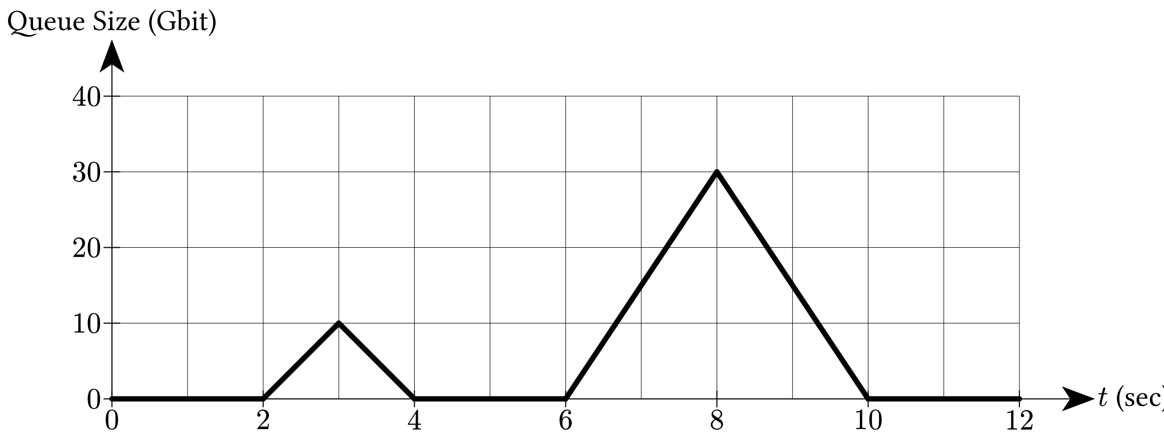
- Starting at $t = 0$, EvanBot sends an endless stream of data, at constant rate 20 Gbps, from H to T.
- At E, each bit can be forwarded independently of other bits.



	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

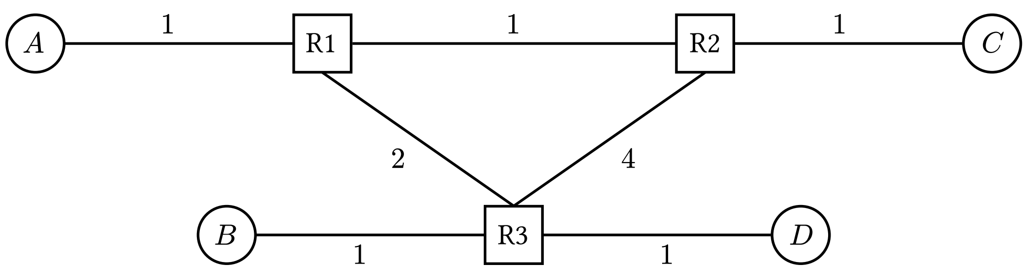
This content is protected and may not be shared, uploaded, or distributed.

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		

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.

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.

☐ Poison Expired Routes

☐ Poison Reverse

☐ Eventful Updates

☐ Split Horizon

☐ Count to Infinity

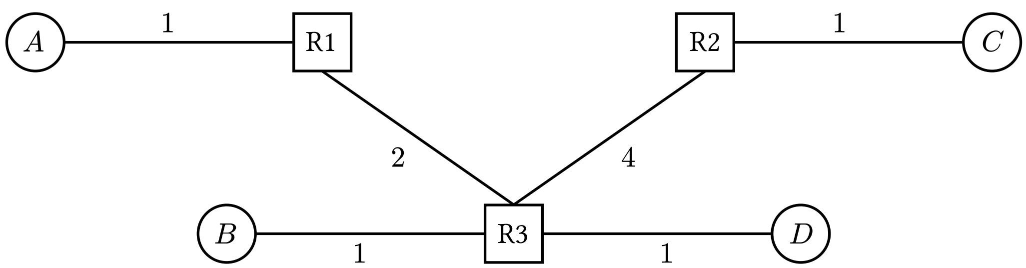
○ None of the above

This content is protected and may not be shared, uploaded, or distributed.

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		

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?

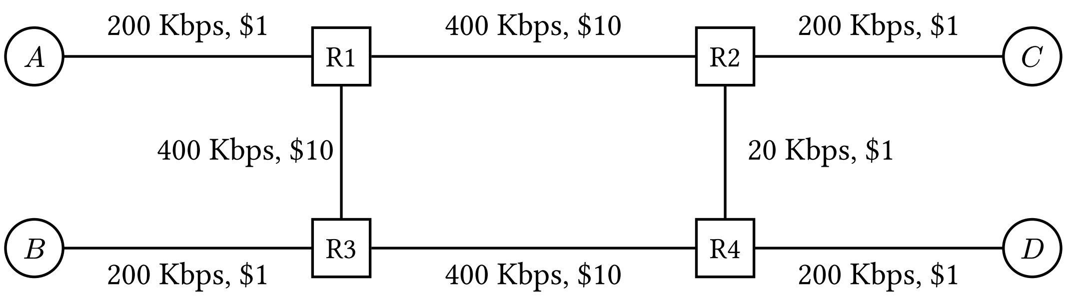
$t =$

This content is protected and may not be shared, uploaded, or distributed.

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.

[ ]

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

This content is protected and may not be shared, uploaded, or distributed.

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.

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.

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.

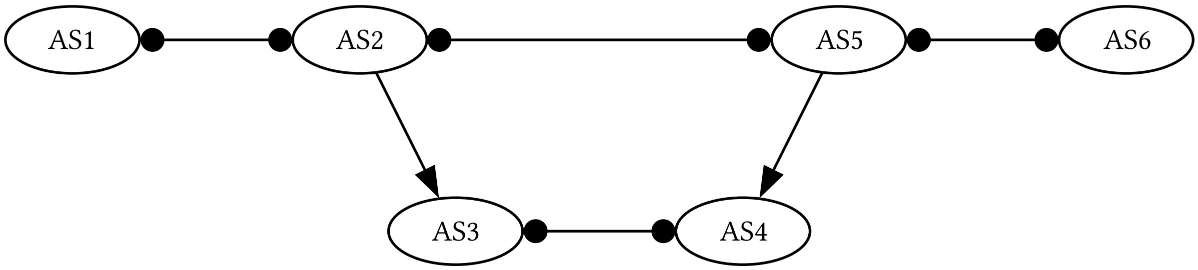
☐ Packets contain timestamps.

○ None of the above

This content is protected and may not be shared, uploaded, or distributed.

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

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

◯ Possible ◯ Not possible

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

◯ Possible ◯ Not possible

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.

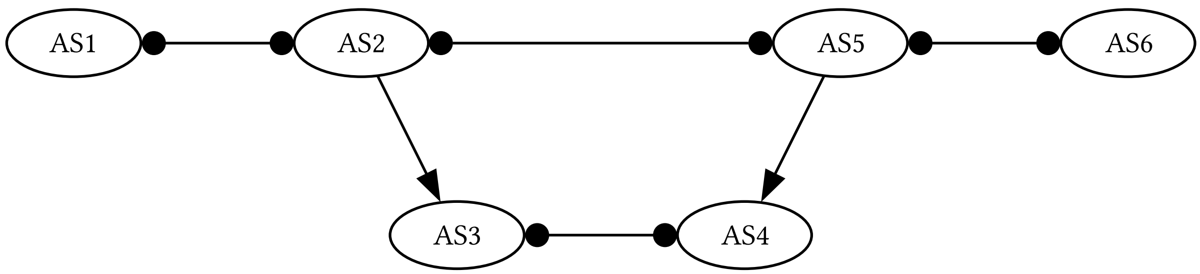
◯ Because AS1 and AS6 have no customers, and all Tier 1 ASes must have a customer.

◯ Because it is always impossible to send packets to AS6.

This content is protected and may not be shared, uploaded, or distributed.

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.



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

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

◯ Reachable ◯ Not reachable

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

◯ Reachable ◯ Not reachable

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

This content is protected and may not be shared, uploaded, or distributed.

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

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

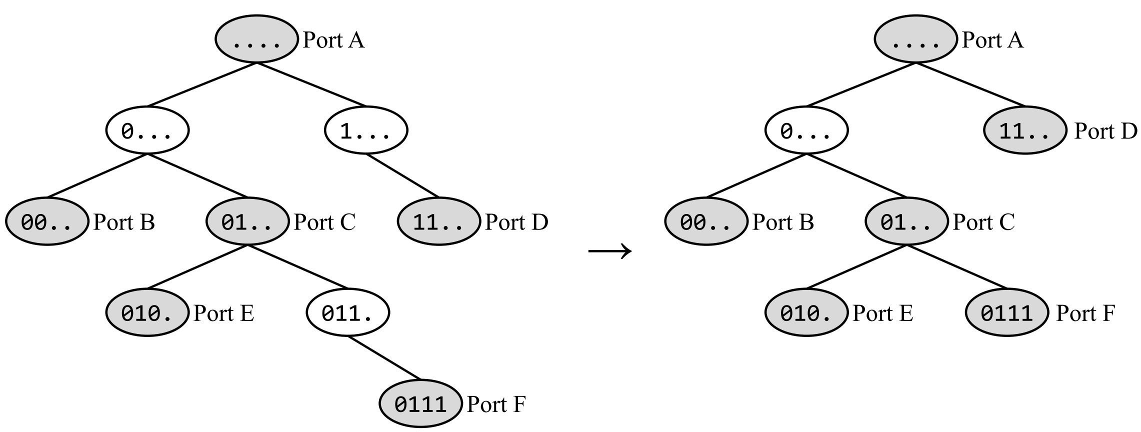
Q6.3 (2 points) What is the maximum height that *any* binary trie can have for IPv4 addresses?

○ 3 ○ 8 ○ 16 ○ 24 ○ 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

This content is protected and may not be shared, uploaded, or distributed.

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

◯ Data Plane ◯ Control Plane ◯ Management Plane

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

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

☐ 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.

☐ Inserting a node for a Patricia trie can be more computationally intense than for a binary trie.

☐ A Patricia trie can directly connect the default route root node with a node with 32 bits fixed.

◯ None of the above

This content is protected and may not be shared, uploaded, or distributed.

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.

This content is protected and may not be shared, uploaded, or distributed.