CS 168 Summer 2026

Introduction to the Internet Final Exam

PRINT Your Name: **Solutions last updated: August 14, 2026**

PRINT Your Student ID: _____

You have 170 minutes. There are 8 questions of varying credit. (100 points total)

Question:	1	2	3	4	5	6	7	8	Total
Points:	12	21	14	12	11	10	10	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)

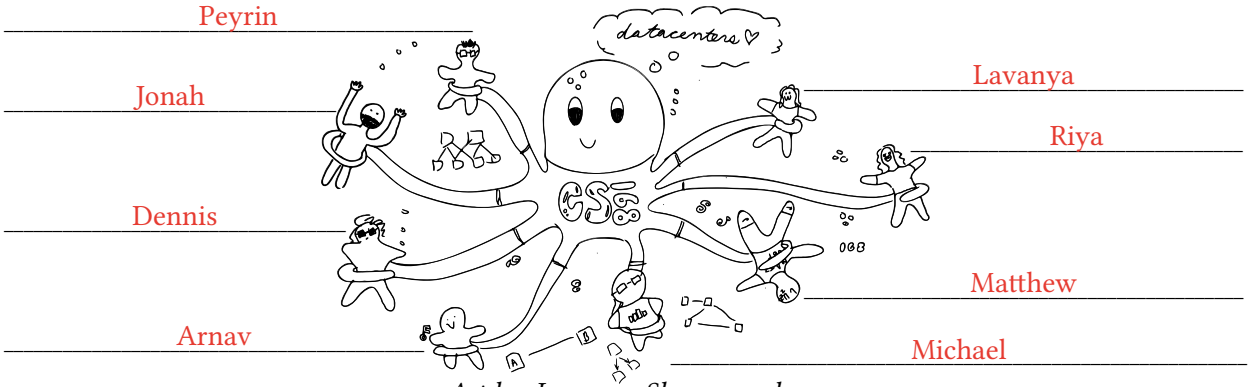
Anything you write outside the answer boxes or you ~~cross out~~ will not be graded. If you write multiple answers, your answer is ambiguous, or the bubble/checkbox is not entirely filled in, we will grade the worst interpretation.

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: _____

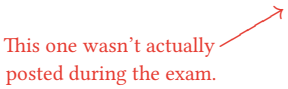
Pre-Exam Activity: Oh no! All of our instructors have been grabbed by an octopus... but who's who?



Art by: Lavanya Shyamsundar

Clarifications

- Q3 — In case of a tie, the router with the lower ID is considered “closer” to the root.
- Q4.1–4.5 — R1, R2, and R3 are not using NAT.
- The art on the front page was actually a collaborative effort by Lavanya, Riya.



Q1 *Projectpourri* **(12 points)**

For Q1.1–Q1.6, consider a working implementation of Project 3: Transport.

Q1.1 (2 points) When a data packet is first sent, what does the sender do? Select all that apply.

- [A] Tag the packet with a timestamp.
- [B] Add the packet to the receive queue.
- [C] Add the packet to the retransmit queue.
- [D] Send an ACK for the packet.
- (E) None of the above

Solution: When a data packet is first sent, the sender tags the data packet with a timestamp and added to the retransmit queue.

Q1.2 (2 points) Suppose the packet with the smallest sequence number in the receive queue has sequence number 120. If `self.rcv.nxt = 100`, what should the receiver do with this packet?

- (A) Call `handle_accepted_seg` on the packet, and advance `self.rcv.nxt` past 100.
- (B) Call `handle_accepted_seg` on the packet, but keep `self.rcv.nxt` at 100.
- (C) Keep the packet in the receive queue, and set a pending ACK.
- (D) Dequeue the packet from the receive queue, and wait for bytes 100–119 to arrive.

Solution: The receiver should not call `handle_accepted_seg` until it receives the packet with sequence number `self.rcv.nxt`, and should keep the packet in the receive queue.

Q1.3 (2 points) The receiver starts by advertising a large receive window, then later advertises a smaller receive window. How should this change affect the sender?

- (A) Reduce the amount of additional data allowed in flight.
- (B) Retransmit enough unacknowledged data to match the new window.
- (C) Decrease `self.snd.nxt` to fit the new window.
- (D) Drop enough unsent data to fit the new window.

Solution: Less data should be allowed in flight because the sender uses a smaller advertised window.

Q1.4 (2 points) A sender transmits three data packets: S1, S2, S3.

S2 and S3 arrive at the receiver, but S1 is lost. Later, the sender retransmits S1, and receives a cumulative ACK indicating that S1, S2, and S3 were received. What should the sender do?

(A) Retransmit S1, because S1 lacks an individual ACK.

(B) Remove S1, S2, and S3 from the retransmit queue.

(C) Retransmit S2 and S3.

(D) Keep S1, S2, and S3 in the retransmit queue until each of their retransmission timers expires.

Solution: The ACK cumulatively acknowledges all three packets. Therefore, S1, S2, and S3 are all acknowledged and should be removed from the retransmit queue.

Q1.5 (2 points) Suppose the receiver has already accepted bytes 100–109, and then receives a new packet whose payload is bytes 105–114. What should the receiver do?

(A) Keep the packet in the receive queue until the sender transmits 110–119.

(B) Call `handle_accepted_seg` on 100–114, and set `self.rcv.nxt` equal to 110.

(C) Remove 105–109 from the receive queue, and call `handle_accepted_seg` on 110–114.

(D) Replace 105–109 with the new packet's bytes, and call `handle_accepted_seg` on 110–114.

Solution: Bytes 105–109 have already been accepted, so they should not be processed again. The receiver removes the overlap and processes only the new bytes 110–114.

Q1.6 (2 points) A sender has data available in `self.tx_data`, but all bytes in the sender's currently advertised send window are in flight. What should `maybe_send` (from Stage 4) do?

(A) Send one packet if its payload length does not exceed the MSS.

(B) Retransmit the oldest packet to slide the window forward.

(C) Wait until ACKs or window updates slide the window forward.

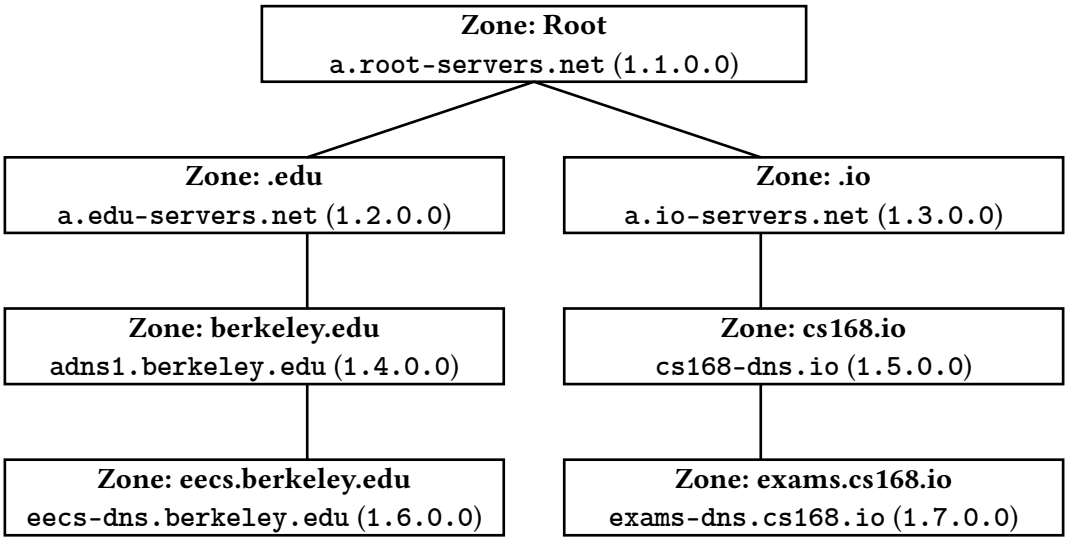
(D) Increase the send window to fit the remaining data.

Solution: The sender cannot send more data while the advertised send window is full. It must wait for ACKs or a window update to slide the window forward.

Q2 *DNS* (21 points)

Consider the DNS hierarchy below. Each box contains a zone, and the domain and IP of the authoritative name server for that zone. There are no other zones and name servers besides the ones shown.

Unless otherwise stated, assume no DNS or network errors occur.



In the berkeley.edu zone, we add the record:

```
cs168.berkeley.edu CNAME octobot.cs168.io
```

In the eecs.berkeley.edu zone, we add the record:

```
hi.eecs.berkeley.edu   A   1.6.1.0
```

For Q2.1–Q2.4, a recursive resolver starts with an empty cache. A host Octobot uses a stub resolver to forward queries to a recursive resolver.

Q2.1 (2 points) Octobot looks up **`hi.eecs.berkeley.edu`**. How many queries does Octobot's stub resolver need to make to look up the IP address?

Ⓐ 0 Ⓑ 1 Ⓒ 2 Ⓓ 3 Ⓔ 4 Ⓕ 5 Ⓖ 6 Ⓗ 7

Solution: The stub resolver sends out only one query to the recursive resolver.

Q2.2 (1 point) What record gets sent back to the stub resolver after looking up the IP address for `hi.eecs.berkeley.edu`?

Ⓐ `hi.eecs.berkeley.edu NS 1.6.1.0`

Ⓑ `hi.eecs.berkeley.edu A 1.6.1.0`

Ⓒ `hi.eecs.berkeley.edu CNAME 1.6.1.0`

Ⓓ `hi.eecs.berkeley.edu MX 1.6.1.0`

Ⓔ None of the above

Solution: Because the query asks for an IPv4 address, we send back an `A` type record.

Q2.3 (2 points) The recursive resolver queries for the IP address of `cs168-dns.io`. How many DNS queries does the recursive resolver need to make to look up the IP address of `cs168-dns.io`?

Ⓐ 0 Ⓑ 1 Ⓒ 2 Ⓓ 3 Ⓔ 4 Ⓕ 5 Ⓖ 6 Ⓗ 7

Solution: The recursive resolver has no relevant cache entries so it: (1) queries the root for .io, (2) queries .io for .cs168-dns.io and finds the record for it.

Q2.4 (2 points) Continuing from Q2.3 (i.e. caches not cleared), how many additional DNS queries does the recursive resolver need to make to look up the IP address of `cs168.berkeley.edu`?

Ⓐ 0 Ⓑ 1 Ⓒ 2 Ⓓ 3 Ⓔ 4 Ⓕ 5 Ⓖ 6 Ⓗ 7

Solution:

We dropped this subpart and gave everybody points due to two ambiguities in this question:
- We did not specify whether or not the records from Q2.1 or not.
- We were not clear enough whether Q2.3 is querying about the cs168.io zone, or the `cs168-dns.io` server specifically.

The query from Q2.1 may or may not cache the following records:
1. `a.root-servers.net` is authoritative for Root.
2. `a.root-servers.net` has IP `1.1.0.0`.
3. `a.edu-servers.net` is authoritative for .edu.
4. `a.edu-servers.net` has IP address `1.2.0.0`.
5. `adns1.berkeley.edu` is authoritative for berkeley.edu.
6. `adns1.berkeley.edu` has IP address `1.4.0.0`.
7. `eecs.berkeley.edu` is authoritative for eecs.berkeley.edu.
8. `eecs-dns.berkeley.edu` has IP address `1.6.0.0`.

If you assume the records from Q2.1 stay, you would include records 1–8. If you assume the cache clears, you would not include them.

The query from Q2.3 may or may not cache the following records:
9. `a.io-servers.net` is authoritative for .io.
10. `a.io-servers.net` has IP address `1.3.0.0`.
11. `cs168-dns.io` is authoritative for cs168.io.
12. `cs168-dns.io` has IP address `1.5.0.0`.

At this point, Q2.3 splits into two possible interpretations:
- If you thought the query was for the cs168.io zone, you would get two more records: 11 and 12.
- If you thought the query was just for `cs168-dns.io`, ignoring the fact that it is authoritative for the cs168.io zone, you would get only record 12, skipping record 11.

With two ambiguities, and two interpretations each, there are actually four possible solutions. (Given how complicated this is, we ultimately dropped the entire subpart.)

All four solutions involve querying for `cs168.berkeley.edu`, getting redirected, and querying again for `octobot.cs168.io`. The difference is how many records live in the cache at query time.

Solution 1: You assume that records are cached, and the Q2.3 query had zone information. This gives you records 1–12 in your cache.
- Ask `adns1.berkeley.edu` (whose IP you know from record 6) about the IP address of `cs168.berkeley.edu` (you know the name server is authoritative for this query because of record 5).
- The response triggers the redirect.
- Ask `cs168-dns.io` (whose IP you know from record 12) about the IP address of `octobot.cs168.io` (you know the name server is authoritative for this query because of record 11).

Answer: 2 queries.

(continued on the next page)

Solution: (continued)

Solution 2: You assume that records are *not* cached, and the Q2.3 query had zone information. This gives you records 9–12 in your cache.

- Ask `a.root-servers.net` about the IP address of `cs168.berkeley.edu`.
- Ask `a.edu-servers.net` about the IP address of `cs168.berkeley.edu`.
- Ask `adns1.berkeley.edu` about the IP address of `cs168.berkeley.edu`.
- The response triggers the redirect.
- Ask `cs168-dns.io` (whose IP you know from record 12) about the IP address of `octobot.cs168.io` (you know the name server is authoritative for this query because of record 11).

Answer: 4 queries.

Solution 3: You assume that records are cached, and the Q2.3 query *did not* have zone information. This gives you records 1–10 and 12 in your cache.

- Ask `adns1.berkeley.edu` (whose IP you know from record 6) about the IP address of `cs168.berkeley.edu` (you know the name server is authoritative for this query because of record 5).
- The response triggers the redirect, but at this point, even though you know `cs168-dns.io`'s IP address, you don't realize it's authoritative for cs168.io domains.
- Ask `a.io-servers.net` (whose IP you know from record 10) about the IP address of `octobot.cs168.io` (you know the name server is authoritative for this query because of record 9). This returns the missing record 11.
- Ask `cs168-dns.io` (whose IP you know from record 12) about the IP address of `octobot.cs168.io` (you know the name server is authoritative for this query because of record 11).

Answer: 3 queries.

Solution 4: You assume that records are *not* cached, and the Q2.3 query *did not* have zone information. This gives you records 9, 10, and 12 in your cache.

- Ask `a.root-servers.net` about the IP address of `cs168.berkeley.edu`.
- Ask `a.edu-servers.net` about the IP address of `cs168.berkeley.edu`.
- Ask `adns1.berkeley.edu` about the IP address of `cs168.berkeley.edu`.
- The response triggers the redirect.
- The response triggers the redirect, but at this point, even though you know `cs168-dns.io`'s IP address, you don't realize it's authoritative for cs168.io domains.
- Ask `a.io-servers.net` (whose IP you know from record 10) about the IP address of `octobot.cs168.io` (you know the name server is authoritative for this query because of record 9). This returns the missing record 11.
- Ask `cs168-dns.io` (whose IP you know from record 12) about the IP address of `octobot.cs168.io` (you know the name server is authoritative for this query because of record 11).

Answer: 5 queries.

For Q2.5–Q2.8, Octobot is trying to look up the IP addresses of different sites. Assuming each record exists, which name server sends the given record to the recursive resolver?

Q2.5 (2 points) `.edu NS a.edu-servers.net`

(A) Root name server
(B) `.edu` name server
(C) `.net` name server

Solution: The root nameserver knows who's authoritative for `.edu` and tells the recursive resolver "go ask a.edu-servers.net for anything.edu", so the root nameserver sends this NS record.

Q2.6 (1 point) `octobot.cs168.io A 1.5.28.5`

(A) `.io` name server
(B) `berkeley.edu` name server
(C) `cs168.io` name server

Solution: `octobot.cs168.io` is a host inside of the `cs168.io` zone, so the `cs168.io` name server sends the final A record.

Q2.7 (1 point) `soda.berkeley.edu CNAME soda.io`

(A) Root name server
(B) `.io` name server
(C) `berkeley.edu` name server

Solution: Since `soda.berkeley.edu` is a host inside of the `berkeley.edu` zone, a record about it, the CNAME in this case, would be sent by the `berkeley.edu` name server.

Q2.8 (2 points) `eecs.berkeley.edu NS eecs-dns.berkeley.edu`

(A) Root name server
(B) `.io` name server
(C) `berkeley.edu` name server

Solution: This records says that the `a.io-servers.net` name server is authoritative for the `eecs.berkeley.edu` zone, but this is false. Therefore, this record will not be sent by any of the name servers.

Q2.9 (2 points) Suppose that `eecs-dns.berkeley.edu`'s IP address changes from `1.6.0.0` to `1.6.5.0`. `eecs-dns.berkeley.edu` remains authoritative for the eecs.berkeley.edu zone. What zone(s) need to update their record as a result?

(A) Only root name server

(B) Only `.edu` name server

(C) Only `berkeley.edu` name server

(D) Only `eecs.berkeley.edu` name server

(E) Both `.edu` and `berkeley.edu` name servers

Solution: `eecs.berkeley.edu` is inside of `berkeley.edu`'s zone, and the new entry only needs to be updated in the zone it lives in.

Q2.10 (2 points) Octobot joins a different network and looks up `www.exams.cs168.io`.
How many queries does Octobot's recursive resolver need to make to look up the IP address for `www.exams.cs168.io`?

(A) 0 (C) 2 (E) 4

(B) 1 (D) 3 (F) Not enough information

Solution: Not enough information: It isn't known if the new network's recursive resolver (which may be different from Octobot's original recursive resolver) has anything cached. Recursive resolvers are typically shared among many users, so may have servers cached.

Q2.11 (2 points) Suppose that the root server goes down, and the recursive resolver cannot reach the root server.

Which modification to this setup would best help eliminate this issue?

Reminder: In the current setup, we assume that `a.root-servers.net` is the only root name server.

(A) Use anycast with physical mirrors.

(B) Add more recursive resolvers.

(C) Decrease the TTL on the root zone's records.

(D) Switch the DNS from UDP to TCP.

Solution: Anycast allows us to have multiple root servers that resolve to the same IP. This allows us to have redundancy in case any servers go down.

Q2.12 (2 points) What happens if all servers switch from using IPv4 addresses to using IPv6 addresses?

(A) New **`AAAA`** type records are added, instead of **`A`** type records.

(B) The recursive resolver empties its cache.

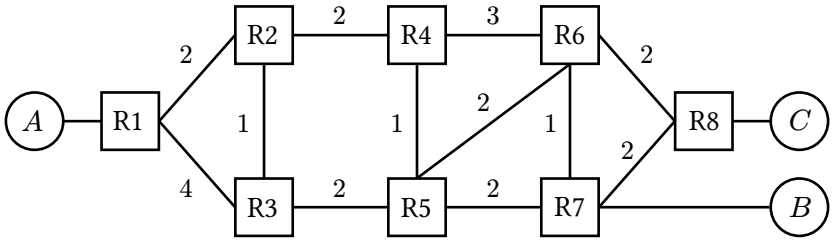
(C) **`NS`** type records are updated with the new IPv6 addresses.

(D) A flag bit is set in the DNS UDP packet.

Solution: **`AAAA`** type records are used instead of **`A`** records in IPv6.

Q3 *STP* **(14 points)**

Consider running STP on the following topology:



Switch IDs are ordered by their labels (R1 has the lowest ID, R2 has the next lowest ID, etc.).

Q3.1 (4 points) What is the state of the network after STP converges? Fill in the table below:

Switch	Total path cost to root	Switch at other end of root port
R5		
R6		
R7		
R8		

Solution:

Switch	Total path cost to root	Switch at other end of root port
R5	5	R3
R6	7	R4
R7	7	R5
R8	9	R6

Q3.2 (3 points) Select all switch ports that are blocked after STP converges.
For example, "R3 to R1" means R3's port on the R3–R1 link.

[A] R1 to R2	[F] R3 to R1	[K] R4 to R6	[P] R6 to R4	[U] R7 to R6
[B] R1 to R3	[G] R3 to R2	[L] R5 to R3	[Q] R6 to R5	[V] R7 to R8
[C] R2 to R1	[H] R3 to R5	[M] R5 to R4	[R] R6 to R7	[W] R8 to R6
[D] R2 to R3	[I] R4 to R2	[N] R5 to R6	[S] R6 to R8	[X] R8 to R7
[E] R2 to R4	[J] R4 to R5	[O] R5 to R7	[T] R7 to R5	(Y) None

Solution:

- R3 to R1: R3 gets a cheaper path through R2.
- R5 to R4: R5 chooses R3 after an equal-cost tie.
- R6 to R5: R6 chooses R4 after an equal-cost tie.
- R7 to R6: Both have root cost 7, so R6 wins the link because it has lower ID.
- R8 to R7: R8 has equal-cost paths through R6 and R7, so R6 (lower ID) becomes its root-port neighbor.

Q3.3 (2 points) Fill in the blanks about the BPDU messages exchanged between routers:

Before receiving any BPDUs, R3 advertises: "The root is ____ (i) and I can reach it with cost ____ (ii).

After convergence, R3 advertises: "The root is ____ (iii) and I can reach it with cost ____ (iv)."

(A) (i) R3	(ii) 1	(iii) R1	(iv) 3
(B) (i) R1	(ii) 0	(iii) R1	(iv) 3
(C) (i) R3	(ii) 0	(iii) R1	(iv) 3
(D) (i) R3	(ii) 0	(iii) R2	(iv) 1

Solution: Before receiving any BDPUs, R3 assumes it is the root and advertises root R3 with cost 0.

After convergence, R1 is the root. R3 reaches it through R2 with cost 1+2=3, so it advertises root R1 with cost 3.

Q3.4 (2 points) Suppose R6 receives only these two BPDUs:

- From R5: "The root is R2, and I can reach it with cost 3."
- From R7: "The root is R5, and I can reach it with cost 2."

After processing both BPDUs, what is R6's STP state?

Ⓐ Root R2 is 3 away.

Ⓑ Root R2 is 5 away.

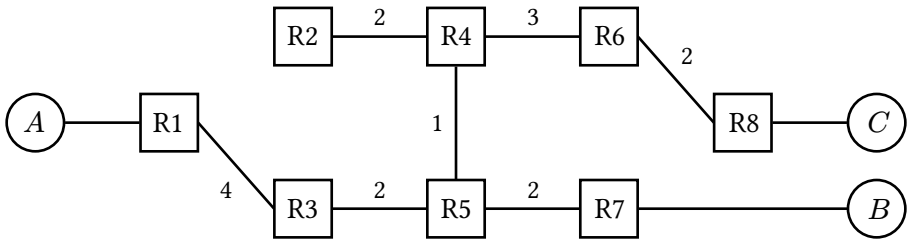
Ⓒ Root R5 is 2 away.

Ⓓ Root R5 is 3 away.

Solution: STP first prefers the BPDU advertising the lower root ID, so R6 chooses R2 over R5 no matter the path cost.

R5 advertises a cost of 3 to R2, and the R6-R5 link costs 2, so the total cost is 3+2=5.

Regardless of your answers to the previous subparts, suppose STP has converged on the topology below:



Assumptions:

- Switches R1 to R8 are all learning switches.
- All forwarding tables start out empty.

In each of the next subparts, select all of the switches that will receive the given packet.

Packets are sent one after the other, i.e. forwarding table entries in one subpart carry over to later subparts.

Q3.5 (1 point) B sends a packet to A.

A R1 B R2 C R3 D R4 E R5 F R6 G R7 H R8

Solution: Destination A is unknown, so the packet is flooded through the entire spanning tree.

Q3.6 (1 point) A sends a packet to C.

A R1 B R2 C R3 D R4 E R5 F R6 G R7 H R8

Solution: The switches learned B's location from the last packet but not C's. So the packet is flooded everywhere again.

Q3.7 (1 point) C sends a packet to A.

[A] R1 [B] R2 [C] R3 [D] R4 [E] R5 [F] R6 [G] R7 [H] R8

Solution: Every switch learned A's location in Q3.6, so this packet is unicasted along R8->R6->R4->R5->R3->R1

Q4 *End-to-End* **(12 points)**

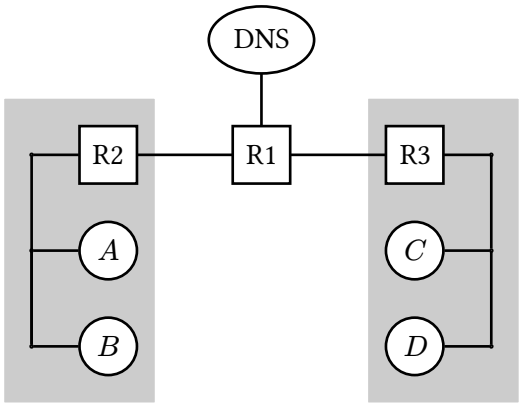
Consider the network topology to the right.

Host A, Host B, and R2 are all on the same subnet and connected by a single shared medium.

Likewise, Host C, Host D, and R3 are all on a different subnet and connected by another shared medium.

There is a single DNS resolver in the network, labeled DNS. Assume everyone uses the same DNS resolver.

For Q4.1–Q4.5, each subpart continues from the previous one.



Q4.1 (2 points) Host C joins the network for the first time with an empty cache and broadcasts a DHCP discover request. R3 responds with a DHCP offer with some addresses, and Host C accepts the offer.

What addresses are contained in the payload of R3's DHCP offer? Select all that apply.

- [A] R1's IP
- [B] DNS's MAC
- [C] C's MAC
- [D] DNS's IP
- [E] C's IP
- [F] D's IP
- [G] R3's MAC
- [H] R3's IP

Solution: A DHCP offer contains the following configuration information: the subnet mask, the host's IP address, the next-hop router's IP address, and the DNS resolver's IP address. In this case, that means C's IP, R3's IP, and DNS's IP are contained in the DHCP offer payload.

Q4.2 (2 points) Host C wants to communicate with Host B using the hostname `bee.cs168.org`.

What must Host C do first?

- (A) Send a DNS query with the DNS resolver as the IP destination.
- (B) Broadcast another DHCP discover request to receive a new IP address.
- (C) Send a DNS query with R3 as the IP destination, so R3 can forward the query.
- (D) Broadcast an ARP request to determine R3's MAC address.

Solution: In order to resolve the address that the hostname corresponds to, we must send a DNS query to the DNS resolver. However, to do so, we must first determine R3's MAC address so they can forward the query to the DNS resolver. We do this by broadcasting an ARP request.

Q4.3 (2 points) Later, Host C learns Host B's IP address. How can Host C determine whether Host B is in the same subnet as Host C?

Ⓐ Check if the IP addresses of Host B and Host C are equal.

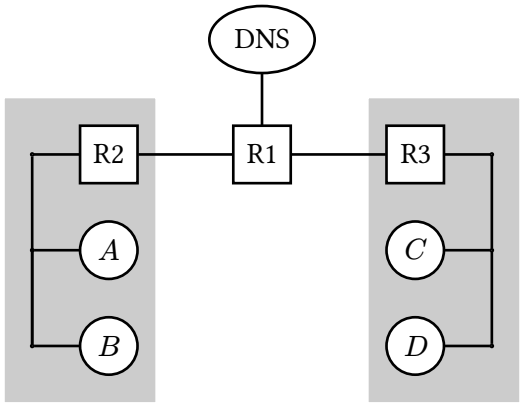
Ⓑ Apply Host C's subnet mask to the IP addresses of Host B and Host C.

Ⓒ Broadcast a DHCP discover request to learn Host B's MAC address.

Ⓓ Apply R2's subnet mask to the IP addresses of Host B and R2.

Solution:

The diagram, reprinted for your convenience:



Q4.4 (2 points) Later, Host C sends its first IP packet to Host B. Host B receives this packet and inspects the packet header.

Fill in the below header fields for this packet. In each box, write one of: A, B, C, D, R1, R2, R3, DNS.

	Source	Destination
MAC Address	R2	B
IP Address	C	B

Solution: MAC Address:
- Source: R2
- Destination: B

IP address:
- Source: C
- Destination: B

Q4.5 (2 points) Later, Host C wants to send another IP packet to Host B. Assume none of Host C's cached entries have expired.

Which steps must Host C perform again before sending the packet? Select all that apply.

- [A] Broadcast a DHCP discover request.
- [B] Broadcast an ARP request.
- [C] Send a DNS query to the resolver.
- [D] Send an ICMP echo request.
- (E) None of the above

Solution:

Q4.6 (2 points) Which of these are true about Network Address Translation (NAT)? Select all that apply.

- [A] Preserves the end-to-end principle between hosts.
- [B] Blocks unsolicited inbound traffic by default.
- [C] Hides internal IP addresses from external servers.
- [D] Encrypts packets between clients and servers.
- (E) None of the above

Solution:

Q5 *Datacenters* **(11 points)**

In this question, consider the datacenter overlay/underlay protocol from lecture.

A tenant uses virtual network ID 42 and has two VMs:
- VM 1 has virtual IP `192.0.2.0` and physical IP `1.1.1.1`.
- VM 2 has virtual IP `192.0.2.1` and physical IP `2.2.2.2`.

VM 1 sends a packet to VM 2.

Q5.1 (2 points) While the packet is sent along the underlay, what is the outermost destination IP address on the packet?

Ⓐ `1.1.1.1` Ⓑ `2.2.2.2` Ⓒ `192.0.2.0` Ⓓ `192.0.2.1`

Solution: The virtual switch encapsulates the original packet with an outer IP header for the underlay. Since VM 2 is hosted at physical IP `2.2.2.2`, the outer destination IP is `2.2.2.2`.

Q5.2 (2 points) When the packet from VM 1 reaches VM 2, the outer IP header is removed. Which destination IP address does the resulting packet contain?

Ⓐ `1.1.1.1` Ⓑ `2.2.2.2` Ⓒ `192.0.2.0` Ⓓ `192.0.2.1`

Solution: At VM 2's host, the outer IP header is removed. The original packet remains, so its destination is VM 2's virtual IP, `192.0.2.1`.

Q5.3 (2 points) Which of these are true? Select all that apply.

[A] VM 1 must know that VM 2 is hosted at physical address `2.2.2.2`.

[B] VM 2 must know that VM 1 is hosted at physical address `1.1.1.1`.

[C] Another tenant could reuse virtual IP `192.0.2.1` with a different virtual network ID.

[D] VM 2's virtual switch replaces a virtual IP address with a physical IP address.

Ⓔ None of the above

Solution: The VMs themselves do not need to know each other's physical addresses as the overlay handles the virtual-to-physical mapping. Virtual IP addresses can be reused by different tenants because the virtual network ID distinguishes their networks.

Q5.4 (2 points) VM 2 moves from physical address `2.2.2.2` to `4.4.4.4`. VM 2's virtual IP and virtual network ID do not change. The underlay already knows how to route to `4.4.4.4`.

Which of these are true after the move? Select all that apply.

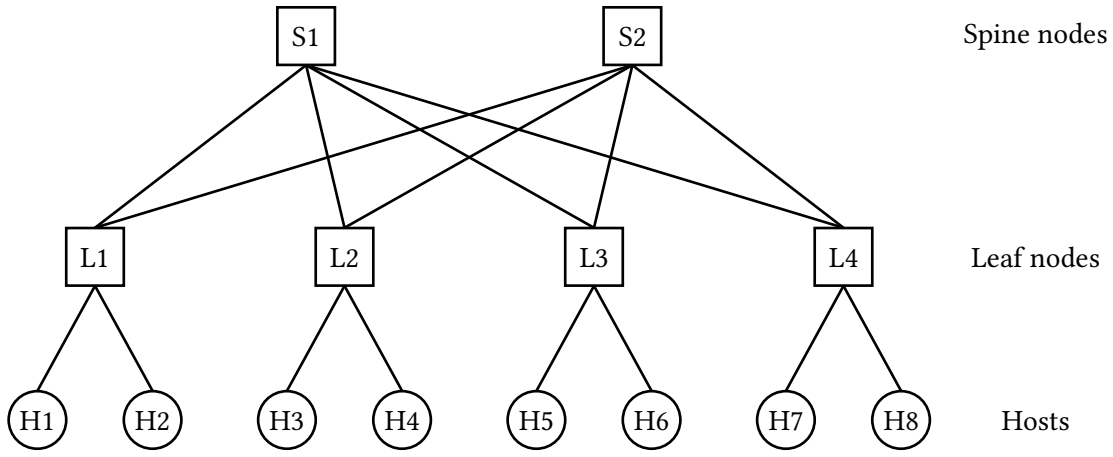
- [A] VM 2 must receive a new virtual IP address.
- [B] VM 2 must receive a new virtual network ID.
- [C] The underlay does not need to learn the virtual-to-physical address mapping.
- [D] Packets sent to VM 2's old physical address are redirected to `4.4.4.4` by the underlay.
- (E) None of the above

Solution: VM 2 keeps the same virtual IP and virtual network ID after moving. The overlay must update VM 2's mapping to `4.4.4.4`, while the underlay only needs to know how to route to that physical address.

(Question 5 continued...)

The following subpart is independent from the previous subparts.

Each host-to-leaf link is 20 Gbps, and each leaf-to-spine link is 10 Gbps.



Q5.5 (3 points) For the partition separating H1–H4 from H5–H8, fill in the following:

Bisection bandwidth:	40Gbps
Full bisection bandwidth:	80Gbps
Oversubscription ratio:	2:1

Solution: The cut has four 10 Gbps leaf-to-spine links, so the bisection bandwidth is 40 Gbps. Full bisection would require $4 \times 20 = 80$ Gbps, giving an oversubscription ratio of $80 : 40 = 2 : 1$.

Q6 *Multicast* **(10 points)**

Q6.1 (2 points) Why is DVMRP not scalable?

Ⓐ Routers store the entire network topology in memory.

Ⓑ Routers can never rejoin a tree once they have left.

Ⓒ Router forwarding tables require one entry per source, per destination group.

Ⓓ DVMRP is an extension of a unicast protocol.

Solution: DVMRP relies on building a separate shortest-path tree from every source to every destination group. As the number of sources and groups grows, the router's state table size increases multiplicatively (Source × Group), which scales poorly in large networks.

Q6.2 (2 points) In CBT, how does a host that is **not** a member of a group send a packet to the group?

Ⓐ The host broadcasts the packet.

Ⓑ The host unicasts an encapsulated packet directly to the core router.

Ⓒ The host must first join the group before the host is allowed to transmit data.

Ⓓ The host sends a special JOIN-ACK message to the nearest leaf router on the tree.

Solution: Because the sender is not on the CBT shared tree, it cannot simply forward the packet to a parent/child link. It must wrap the multicast packet in a unicast IP header destined for the Core. The Core unwraps it and distributes the payload down the group's shared tree.

Consider running CBT on the topology below. Each host belongs to group G1, group G2, or both.

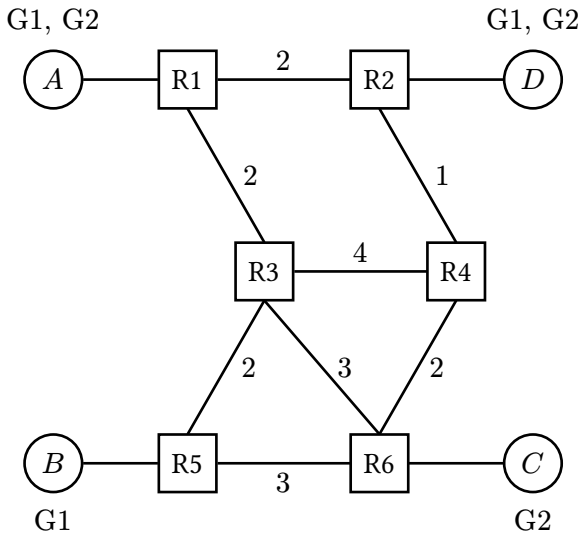
R3 is the designated core for G1, and R4 is the designated core for G2.

Q6.3 (2 points) Select all links that are part of the tree for group G1.

- [A] R1–R2
- [B] R1–R3
- [C] R2–R4
- [D] R3–R4
- [E] R3–R5
- [F] R3–R6
- [G] R4–R6
- [H] R5–R6

Q6.4 (2 points) Select all links that are part of the tree for group G2.

- [A] R1–R2
- [B] R1–R3
- [C] R2–R4
- [D] R3–R4
- [E] R3–R5
- [F] R3–R6
- [G] R4–R6
- [H] R5–R6



Solution: To find the shared tree for G1, we first determine the shortest unicast path from each member's local router to the Core (R3).

The hosts in G1 are A (at R1), B (at R5), and D (at R2).

- Path from R1 to R3: The direct link has cost 2. R1 sends a Join to R3. Tree edge: (R1, R3).
- Path from R5 to R3: The direct link has cost 2. R5 sends a Join to R3. Tree edge: (R5, R3).
- Path from R2 to R3: The shortest path is via R1 (cost 2 + 2 = 4), since via R4 is cost 1 + 4 = 5. R2 sends a Join to R1. Tree edge: (R1, R2).

The total shared tree for G1 consists of edges (R1, R3), (R5, R3), and (R1, R2).

- At R1, the tree edges connect to R2 and R3.
- At R2, the tree edge connects to R1.
- At R3, the tree edges connect to R1 and R5.
- At R5, the tree edge connects to R3.
- R4 and R6 are not on the tree, so their entries are None.

Solution: The hosts in G2 are A (at R1), C (at R6), and D (at R2).

- Path from R2 to R4: The direct link has cost 1. R2 sends a Join to R4. Tree edge: (R2, R4).
- Path from R6 to R4: The direct link has cost 2. R6 sends a Join to R4. Tree edge: (R6, R4).
- Path from R1 to R4: The shortest path is via R2 (cost 2 + 1 = 3). R1 sends a Join to R2. Tree edge: (R1, R2).

The total shared tree for G2 consists of edges (R2, R4), (R6, R4), and (R1, R2).

- At R1, the tree edge connects to R2.
- At R2, the tree edges connect to R1 and R4.
- At R4, the tree edges connect to R2 and R6.
- At R6, the tree edge connects to R4.
- R3 and R5 are not on the tree, so their entries are None.

Q6.5 (2 points) How many distinct CBT Join messages are sent across the network to build these two trees?

Do not count host-to-router messages, and do not count intermediate routers forwarding a Join message. Assume no network errors occur.

Ⓐ 4 Ⓑ 5 Ⓒ 6 Ⓓ 7 Ⓔ 8 Ⓕ Not enough information

Solution: For G1 tree:

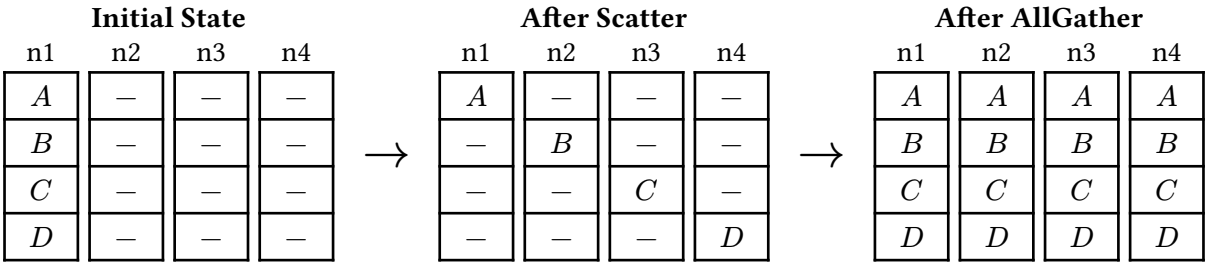
- R1 sends a join packet to R3.
- R2 sends a join packet to R3.
- R5 sends a join packet to R3.

For G2 tree:

- R2 sends join packet to R4.
- R1 sends join packet to R4.
- R6 sends join packet to R4.

Q7 *Collectives* **(10 points)**

Consider a collective operation with N nodes, each with an N-element vector.
We run a Scatter operation, followed by an AllGather operation. An example with $N = 4$ is shown below:



Q7.1 (2 points) True or false: Scatter and AllGather run computations on data while sending data across the network.

Ⓐ True Ⓑ False

Solution:

False. While some operations like AllReduce will aggregate data while it is being sent, these two operations leave the data unchanged.

For Q7.2–Q7.3, suppose we connect the N nodes in a full mesh (all-to-all) overlay topology.

Q7.2 (2 points) What is the minimum number of vector elements that **Node 1** needs to send through the network, across both steps combined?

Ⓐ 1 Ⓑ $N-1$ Ⓒ $2(N-1)$ Ⓓ $N(N-1)$ Ⓔ N^2

Solution: In the Scatter operation, Node 1 sends $N-1$ vector elements to the other $N-1$ nodes. In the AllGather operation, Node 1 needs to send its vector element to the other $N-1$ nodes, resulting in $2(N-1)$ total vector elements sent.

Q7.3 (2 points) What is the minimum number of vector elements that **Node 2** needs to send through the network, across both steps combined?

Ⓐ 1 Ⓑ $N-1$ Ⓒ $2(N-1)$ Ⓓ $N(N-1)$ Ⓔ N^2

Solution: In the Scatter operation, Node 2 receives a one vector element from Node 1 and sends 0 vector elements. In the AllGather step, it sends its vector element to the other $N-1$ nodes, so it sends a total of $N-1$ vector elements.

For Q7.4–Q7.5, suppose we connect the N nodes in a ring overlay topology. Then, we implement the AllGather operation in a similar way as the optimized AllReduce implementation from lecture:

- At the start, every node sends its element toward the left simultaneously.
- At each step, if a node receives a new element, the node sends that element to its left.
- The operation concludes when each node has the full vector.

Q7.4 (2 points) How many total vector elements are sent through the network in this AllGather implementation?

Ⓐ 1 Ⓑ $N-1$ Ⓒ $2(N-1)$ Ⓓ $N(N-1)$ Ⓔ N^2

Solution: At each time step, each node sends a vector element, which is N vector elements for $N-1$ timesteps, giving the answer $N(N-1)$

Q7.5 (2 points) True or false: For this AllGather implementation to work, the latencies between neighboring nodes must all be identical.

Ⓐ True Ⓑ False

Solution: False. The ring is an overlay topology, and there is no guarantee about the latencies of the physical underlay links between neighboring nodes. It is the collective operation's job to coordinate nodes so that the operation completes correctly, even if the latencies are not all identical.

Q8 *Wireless and Cellular Potpourri* **(10 points)**

Q8.1 (2 points) According to MACA (not MACAW), in which of these scenarios are you allowed to immediately send an RTS, without waiting for another transmission to finish? Select all that apply.

[A] You hear an RTS at $t = 1$ and want to send an RTS at $t = 2$.

[B] You hear an RTS at $t = 1$, hear nothing at $t = 2$, and want to send an RTS at $t = 3$.

[C] You hear an RTS at $t = 1$, hear a CTS at $t = 2$, and want to send an RTS at $t = 3$.

[D] You hear nothing at $t = 1$, hear a CTS at $t = 2$, and want to send an RTS at $t = 3$.

(E) None of the above

Solution:

(A): False. You are in range of the sender, so you need to be quiet for one time step, to allow the sender to hear the CTS.

(B): True. You are in range of sender, but not the receiver. Therefore, if you send an RTS, you won't clobber out the signal at the receiver.

(C): False. You are in range of both sender and receiver. Therefore, you need to be quiet, to allow the receiver to hear the data transmission.

(D): False. You are in range of the receiver (but not the sender). Therefore, you need to be quiet, to allow the receiver to hear the data transmission.

Q8.2 (2 points) In the free-space model for wireless connections (i.e. inverse-square law), if the receiver moves closer to the sender, the signal strength will ____ increase.

(A) Always (B) Sometimes (C) Never

Solution: In the free-space model, we assume there are no obstacles, so the signal strength is only a function of the distance between sender and receiver. The inverse-square law says that decreasing the distance causes the signal strength to increase.

Q8.3 (2 points) In the path-loss model for wireless connections (i.e. accounting for obstacles and signal collisions), if the receiver moves closer to the sender, the signal strength will ____ increase.

(A) Always (B) Sometimes (C) Never

Solution: In the path-loss model, the signal strength fluctuates as a function of distance. Moving closer may increase signal strength, but it may also decrease signal strength (e.g. if you move in front of an obstacle).

Q8.4 (1 point) True or false: If a single tunnel is configured between a user and the Internet, then a router that is part of that tunnel will see all packets being sent to/from this user.

(A) True (B) False

Solution: True. The single tunnel is the only path between the user and the Internet, so this router will see every packet being forwarded to/from this user.

Q8.5 (1 point) True or false: Routers in a cellular network must run a routing protocol (e.g. distance-vector) to find shortest paths through the network.

(A) True (B) False

Solution: False. The mobility manager creates tunnels through the network, and configures paths for the routers accordingly. The routers do not need to run a routing protocol.

If the routers did run a routing protocol, the routes would never converge, because hosts are constantly moving around.

Q8.6 (1 point) True or false: Routers in a cellular network must communicate with neighboring routers at specific frequencies and times allocated for that router.

(A) True (B) False

Solution: False. Routers in the cellular network are generally connected using wired links. Allocating specific frequencies and times is necessary in a wireless connection between the user's device and the cell tower, but it is not necessary in a wired connection between two routers.

Q8.7 (1 point) True or false: Routers in a cellular network could look at the destination IP header field to deduce whether two packets are being sent to the same cellular device.

(A) True (B) False

Solution: True. A user's IP address stays the same, even as they move around. Therefore, the router can look at the destination IP to deduce whether two packets are being sent to the same user.

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.