

CSC175 Data Communications & Networking

Spring 2026 Midterm Exam

Student Name: _____ ID: _____ _s

Total Points	
--------------	--

Note: This exam paper should be kept confidential and any dissemination violates copyright.

Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9
/10	/10	/10	/10	/15	/10	/10	/10	/5	/10

Q0 (10 points) Multiple-choice questions: enter your answer keys here:

1	2	3	4	5	6	7	8	9	10
C	B	B	B	C	B	B	A	B	B

For the following multiple-choice questions, each question has exactly one correct answer key. If multiple choices are correct, choose the option "All of the above". Fill in the answer keys in the table above. (Answer keys written in the question area will not be counted.)

1. The Internet is "Best-Effort" at Layer 3. What does this mean?

- A) The network guarantees 100% delivery of all packets
- B) The network will tell you immediately if a packet fails to arrive
- C) The network tries its best to deliver packets but makes no guarantees
- D) The network prioritizes important emails over video streaming

ANS: C

2. What is the main function of Layer 4 (Transport Layer)?

- A) To convert digital signals into analog waves
- B) To build reliable delivery (e.g., re-sending lost packets) on top of Layer 3
- C) To physically connect two machines with a wire
- D) To route packets between different towns

ANS: B

3 Which switching approach is more efficient for bursty traffic (e.g., web browsing)?

- A) Circuit switching, because it reserves capacity
- B) Packet switching, because it doesn't waste reserved capacity on idle periods
- C) Both are equally efficient
- D) Neither is efficient for bursty traffic

ANS: B

4. Why is Packet Switching better than Circuit Switching at handling network failures?

- A) Packet switching uses encryption to protect against failures
- B) Packet switching automatically reroutes packets; end hosts need not do anything extra
- C) Packet switching prevents routers from failing
- D) Circuit switching requires immediate manual intervention

ANS: B

5. For an 800-bit packet on a link with bandwidth = 1 Mbps and propagation delay = 1 ms, what is the total packet delay?

- A) 0.0008 ms
- B) 0.8 ms
- C) 1.8 ms
- D) 1.0008 ms

ANS: C

Total packet delay = transmission delay + propagation delay. Transmission delay = 800 bits / 1,000,000 bps = 0.0008 seconds = 0.8 ms. Propagation delay = 1 ms. Total = 0.8 + 1.0 = 1.8 ms.

6 What is destination-based forwarding?

- A) Forwarding packets only to their final destination without using routers
- B) Each router uses a table mapping destinations to next hops; forwarding decisions depend only on the destination address
- C) Forwarding packets based on the source address
- D) Forwarding all packets to a central routing server

ANS: B

7 During the edge relaxation step in Dijkstra's algorithm, which condition must be met to update the shortest distance (SD) to node v?

- A) $SD[v] < SD[u] + w(u,v)$
- B) $SD[v] > SD[u] + w(u,v)$
- C) $SD[u] < SD[v] + w(u,v)$
- D) $SD[v] == SD[u] + w(u,v)$

ANS: B

8 What is the 'Count-to-Infinity' problem?

- A) A loop where routing updates bounce back and forth, incrementing costs indefinitely
- B) When a router runs out of memory
- C) When a packet takes too many hops to reach the destination
- D) When the network bandwidth is infinite

ANS: A

9. Does Poisoned Reverse solve all counting-to-infinity loops?

- A) Yes, it solves all loops permanently
- B) No, it only works for loops involving 2 nodes
- C) No, it only works for loops involving 3 or more nodes
- D) It creates more loops

ANS: B

10. Why don't we use Link-State for the entire Internet (Inter-domain)?

- A) Dijkstra is too hard to spell
- B) Privacy and Scalability
- C) It is too fast
- D) IS-IS is patented

ANS: B

Q1. (10 pts) Links and End-to-End Delay

A packet of size **1,000 bits** is sent from Host A to Host D through three links (store-and-forward). Each link has different transmission rates and propagation speeds.

Link	Distance	Propagation Speed	Transmission Rate
A → B	1,000 km	2×10^8 m/s	1 Mbps
B → C	2,000 km	2×10^8 m/s	2 Mbps
C → D	1,000 km	1×10^8 m/s	1 Mbps

Compute the end-to-end delay (transmission + propagation), assuming store-and-forward switching and ignoring processing/queuing delays.

ANS: Packet size = 1000 bits

Transmission delays:

- 1 Mbps → $1000 / 10^6 = 1$ ms
- 2 Mbps → $1000 / (2 \times 10^6) = 0.5$ ms

Propagation delays:

- $1000 \text{ km} / (2 \times 10^8 \text{ m/s}) \rightarrow 5$ ms
- $2000 \text{ km} / (2 \times 10^8 \text{ m/s}) \rightarrow 10$ ms
- $1000 \text{ km} / (1 \times 10^8 \text{ m/s}) \rightarrow 10$ ms

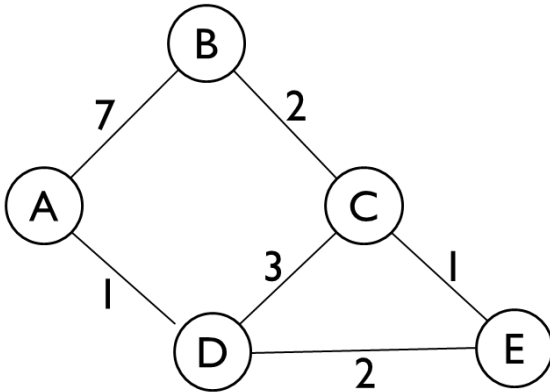
Transmission $1 + 0.5 + 1 = 2.5$ ms

Propagation: $5 + 10 + 10 = 25$ ms

Total delay = 27.5 ms

Q2. (10 pts) Dijkstra Shortest Paths

Consider an Autonomous System with 5 routers (A, B, C, D, E) connected by bidirectional (undirected) point-to-point links. The link costs are shown in the figure. Use Dijkstra's Algorithm to compute the shortest path tree from the Source Router A. Break ties in alphabetical order.



- List the Visit Order of the routers.
- Fill out the Routing Table, keeping track of the Shortest Distance (SD) and Previous Node (PN). When a shorter path is found to a router, cross out the old value and write the new one.

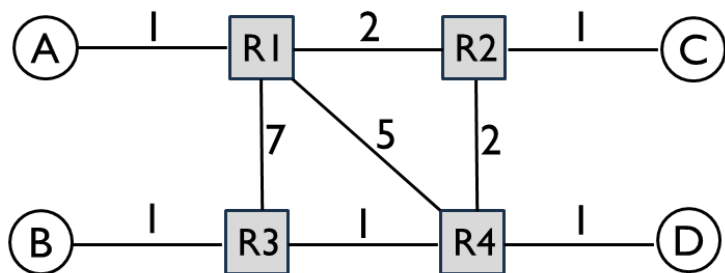
ANS:

- Visit order: A, D, E, C, B

Node	SD	PN
A	0	
B	7 6	A C
C	4	D
D	1	A
E	3	D

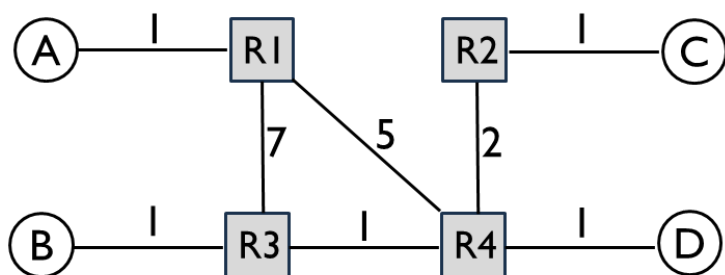
Q3. (10 pts) Distance-Vector Steady State and Failure

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

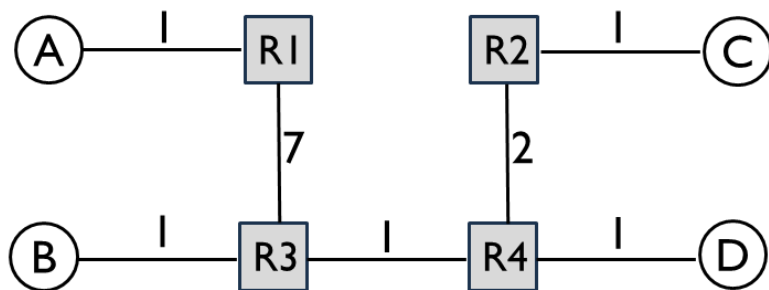


a) Fill in R1's table at steady state. If a host is directly connected, the next hop is "Direct"

b) After the network converges, the R1 - R2 link is broken. Fill in R1's table at the new steady state.



c) After the network converges, the R1 - R4 link is broken. Fill in R1's table at the new steady state.



ANS:

a)

To	Next-hop Router	Cost
A	Direct	1
B	R2	6

C	R2	3
D	R2	5

b)

To	Next-hop Router	Cost
A	Direct	1
B	R4	7
C	R4	8
D	R4	6

c)

To	Next-hop Router	Cost
A	Direct	1
B	R3	8
C	R3	11
D	R3	9

Q4. (15 pts) Link-State

Consider the following network graph with three hosts (A, B, C) and six routers (R1 - R6). For the following questions, assume that the routers run a link-state routing protocol and the routing state has converged. Every link is up unless otherwise noted. When picking between equal-cost paths, the routers pick the route through the neighbor with the lower ID number. For each answer, please provide a concise explanation. Note that all subparts are independent questions (changes made in one subpart do not affect the subsequent ones).

ANS:

a) A → R1 → R2 → R3 → R4 → R5 → C (no change)

b) A → R1 → R2 → R3 → R4 → R3 (routing loop between R3 and R4)

c) A → R1 → R2 → R3 → R4 → R3 → R2 → R6 → R5 → C

t (s)	Event
0	Packet arrives at R1
1	Packet arrives at R2
3	Packet arrives at R3
4.5	R4-R5 link breaks
5	Packet arrives at R4
6.5	LSA from R4 arrives at R3. R3 reconfigures its route to C as R3-R2-R6-R5-C
7	Packet arrives at R3
8.5	LSA from R4 arrives at R2. R2 reconfigures its route to C as R2-R6-R5-C
9	Packet arrives at R2
11	Packet arrives at R6
19	Packet arrives at R5
24	Packet arrives at C

Q5. (10 pts) CIDR Addressing

A university owns prefix 203.0.0.0/16. IP address allocation so far:

- Physics: 203.0.0.0/18
- Chemistry: 203.0.64.0/18
- 203.0.128.0/18 is reserved for CS and Math
- 203.0.192.0/18 is currently unassigned

- a)** (2 pts) What address range is included in Chemistry's prefix, and how many addresses does it contain?
- b)** (4 pts) Assign 3/4 of the address range within 203.0.128.0/18 to CS, 1/4 of the address range to Math. (Hint: you may need to assign multiple prefixes to the same department.)
- c)** (2 pts) A new Department of Anthropology needs at most 70 addresses. What prefix should you assign from the unused block to minimize waste?
- d)** (2 pts) After part (c), another Department of Fine Art needs 20 addresses. What is the smallest unused prefix you can assign to minimize waste?

(For each problem, choose the smallest unused prefix to assign next. You can find some useful binary conversions in the table below. You can write out the binary addresses to help you solve problems, but your solution will be graded based on the decimal values only.)

Decimal	Binary
20	00010100
40	00101000
43	00101011
64	01000000
127	01111111
128	10000000
160	10100000
168	10101000
172	10101100
176	10110000
192	11000000
200	11001000

ANS:

a) Chemistry's prefix 203.0.64.0/18 covers 203.0.64.0 through 203.0.127.255, and network prefix of /18 provides $2^{14} - 2 = 16,382$ usable host addresses. First address is the network address (203.0.64.0); last address is the broadcast address (203.0.127.255). (If you wrote $2^{14} = 16,384$ total addresses, it is also given full grade.)

b) The block 203.0.128.0/18 also has 16,384 total addresses, so CS should get 12,288 addresses and Math should get 4,096 addresses. A clean assignment is: CS = 203.0.128.0/19 and 203.0.160.0/20, Math = 203.0.176.0/20.

Focus on the 3rd octet for the 203.0.128.0/18 block:

The starting block 128.0/18 fixes the first 2 bits of the 3rd octet to 10: 10xx xxxx

To split this block, we assign the next bits (x denotes a host bit):

- CS gets half the space (bit 3 = 0): 100x xxxx → 203.0.128.0/19
- CS gets another quarter (bit 3 = 1, bit 4 = 0): 1010 xxxx → 203.0.160.0/20
- Math gets the final quarter (bit 3 = 1, bit 4 = 1): 1011 xxxx → 203.0.176.0/20

By doing this, CS holds 3/4 of the available combinations (starting with 100 and 1010), and Math holds the remaining 1/4 (starting with 1011).

c) The Department of Anthropology needing at most 70 addresses should get a /25, because /26 provides only $2^6 - 2 = 62$ usable hosts while /25 provides $2^7 - 2 = 126$ usable hosts. With /25, the 4th octet is 0xxx xxxx. Assign 203.0.192.0/25 from the unused 203.0.192.0/18 block.

d) After that, the Department of Fine Art needing 20 addresses should get a /27, because /28 provides only $2^4 - 2 = 14$ usable hosts while /27 provides $2^5 - 2 = 30$ usable hosts. Anthropology has the 4th octet 0xxx xxxx. /27 uses the first 3 bits of the 4th octet for the network. We assign Fine Art the very first available /27 block inside the remaining space, so the 4th octet starts with 1. With /27, the 4th octet is 100x xxxx. Since $1000\ 0000 = 128$, so Fine Art gets 203.0.192.128/27.

(Note: our lecture uses bold font or red color to denote network bits, and regular font to denote host bits. Here I use 0/1 to denote network bits, and x to denote host bits.)

Q6. (10 pts) Longest Prefix Match and Tie-Breaking

A router has 4 ports, with routing table shown below. Use these rules in order: longest prefix match, then lowest cost, then smallest port number. If no prefix matches, use the default route on port 4. Determine which ports the packets with the following destinations are forwarded to based on the advertisements given above. Give brief explanation for each.

Port	Destination	Cost
1	10.0.0.0/8	9
1	172.16.0.0/12	12
1	192.168.128.0/17	11
2	10.1.0.0/16	7
2	192.168.0.0/16	10
3	172.16.32.0/19	6
3	192.168.192.0/18	8
4	10.1.2.0/24	7
4	192.168.128.0/17	9

- a) 10.1.2.8
- b) 10.2.1.1
- c) 172.16.40.10

- d) 172.20.1.1
- e) 192.168.200.1

ANS:

a) 10.1.2.8 matches 10.0.0.0/8, 10.1.0.0/16, 10.1.2.0/24. The /24 is most specific, so choose port 4.

b) 10.2.1.1 matches only 10.0.0.0/8, so choose port 1.

c) 172.16.40.10 matches both 172.16.0.0/12 and 172.16.32.0/19. The /19 is more specific, so choose port 3.

172.16.40.10 in binary: 10101100 00010000 00101000 00001010

172.16.0.0/12 has 12 network bits: 10101100 0001. It matches the first 12 bits.

172.16.32.0/19 has 19 network bits: 10101100 00010000 001. It matches the first 19 bits.

d) 172.20.1.1 matches only 172.16.0.0/12, so choose port 1.

172.20.1.1 in binary: 10101100 00010100 00000001 0000 0001

172.16.0.0/12 has 12 network bits: 10101100 0001. It matches the first 12 bits.

172.16.32.0/19 has 19 network bits: 10101100 00010000 001. It does not match the first 19 bits.

e) 192.168.200.1 matches 192.168.0.0/16, 192.168.128.0/17, 192.168.192.0/18.

192.168.192.0/18 is most specific, so choose port 3.

192.168.200.1 in binary: 11000000 10101000 11001000 0000 0001.

192.168.0.0/16 has 16 network bits: 11000000 10101000

192.168.128.0/17 has 17 network bits: 11000000 10101000 1

192.168.192.0/18 has 18 network bits: 11000000 10101000 11

Q7. (10 pts) Route Aggregation

Consider a router running longest prefix matching to forward packets. Given the current routing table, use route aggregation to build a new table with the minimum number of entries, such that both tables produce the same forwarding decisions. Write one IP prefix per box.

ANS:

a) Current routing table:

Destination	Port
150.10.0.0/24	1
150.10.1.0/24	1
150.10.2.0/24	1

150.10.3.0/24	1
150.10.4.0/24	1
150.10.5.0/24	1

Merged routing table:

Destination	Port
150.10.0.0/21	1

Explanation: All six /24 routes map to the same port, so they can be summarized by a single prefix. The third octet values 0, 1, 2, 3, 4, 5 all begin with the same first 5 bits (00000), so they fit within 150.10.0.0/21 → 1. Since there are no conflicting routes with different ports, this one prefix preserves the forwarding behavior under longest prefix matching.

b) Current routing table:

Destination	Port
150.10.0.0/24	1
150.10.1.0/24	1
150.10.2.0/24	2
150.10.3.0/24	3
150.10.4.0/24	3
150.10.5.0/24	3

Merged routing table:

Destination	Port
150.10.0.0/23	1
150.10.2.0/24	2
150.10.0.0/21	3

Explanation: 150.10.0.0/24 and 150.10.1.0/24 share the first 7 bits in the third octet, and are aggregated into 150.10.0.0/23 → 1, while 150.10.3.0/24, 150.10.4.0/24, and 150.10.5.0/24 share the first 5 bits in the third octet, and are covered by 150.10.0.0/21 → 3. The entry 150.10.2.0/24 → 2 remains separate. By longest prefix matching, the more specific /23 entry overrides the /21, and the more specific /23 and /24 entries override the /21, preserving correct forwarding behavior.

Example: If a packet arrives for 150.10.0.10 (the .0 subnet), it matches the /21 (Port 3) AND the /23 (Port 1). Because /23 is longer than /21, it goes to Port 1.

If a packet arrives for 150.10.2.10 (the .2 subnet), it matches the /21 (Port 3) AND the /24 (Port 2). Because /24 is longer than /21, it goes to Port 2.

If a packet arrives for 150.10.3.10 (the .3 subnet), it only matches the /21 (Port 3). Therefore, it goes to Port 3 (Correct).

c) Current routing table:

Destination	Port
150.10.0.0/24	1
150.10.1.0/24	1
150.10.2.0/24	2
150.10.3.0/24	1
150.10.4.0/24	1
150.10.5.0/24	1

Merged routing table

Destination		Port
150.10.0.0/21		1
150.10.2.0/24		2

Explanation:

The subnets 150.10.0.0/24, 150.10.1.0/24, 150.10.3.0/24, 150.10.4.0/24, and 150.10.5.0/24 c so they can be covered by the aggregate prefix 150.10.0.0/21. This prefix also includes 150.10.2.0/24, which maps to port 2, so a more specific entry 150.10.2.0/24 → 2 is added. By longest prefix matching, traffic for 150.10.2.0/24 matches the /24 entry, while all other addresses match the /21 entry, preserving correct forwarding behavior.

Q8. (5 pts) Binary Tries

a) (2 pts) Build a binary trie for prefixes 25.0.0.0/15, 25.1.0.0/16, and 25.1.128.0/17. What is the trie height? Describe the structure.

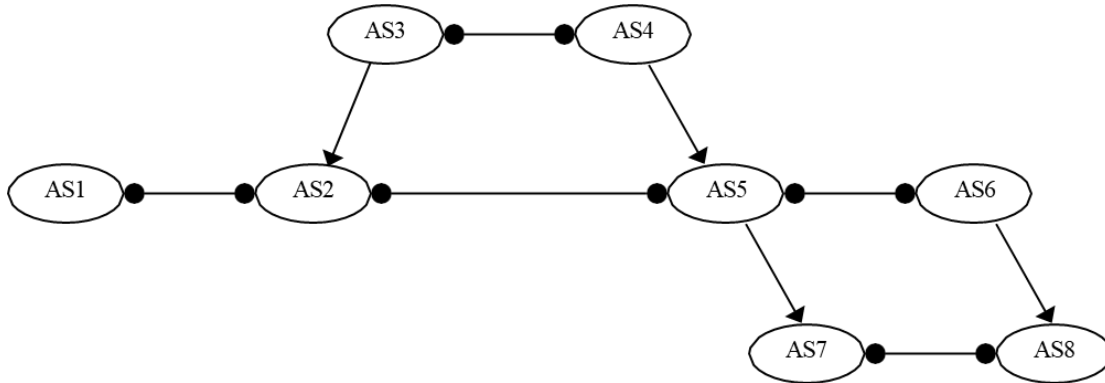
b) (3 pts) Build a binary trie for prefixes 40.0.0.0/8 and 43.128.0.0/8. What is the trie height? At what earliest level does branching happen?

ANS: (a), the longest prefix is /17, so the trie height is 17. Because /15 is a prefix of /16 and /16 is a prefix of /17, the trie is a single chain with no branching through depth 17.

(b), the longest prefix is /8, so the trie height is 8. 40 in binary is 00101000 and 43 in binary is 00101011. They share the first 6 bits, then split at the 7th bit.

Q9. (10 pts) Inter-Domain Routing under Gao-Rexford

Consider the AS graph below, where each AS follows the Gao-Rexford import and export policies. Provider to Customer relationship is denoted by arrows; peer to peer relationship is denoted by horizontal lines with dots. For each source/destination pair, select whether it is possible for packets to be sent from the source AS to the destination AS. Does the AS graph have full reachability from every AS to every other AS? If not, fill out the following table of reachability. (Hint: **Every intermediate AS on a legal path must have at least one customer neighbor along that path.**)



ANS:

Source AS	Reachable Destinations
AS1	AS2
AS2	AS1, AS3, AS4, AS5, and AS7
AS3	AS2, AS4, AS5, and AS7
AS4	AS2, AS3, AS5, and AS7
AS5	AS2, AS3, AS4, AS6, AS7, and AS8
AS6	AS5, AS7, and AS8
AS7	AS2, AS3, AS4, AS5, AS6, and AS8
AS8	AS5, AS6, and AS7

Explanations:

Because an AS will only forward transit traffic if it is getting paid, an **intermediate AS** (any router in the middle of a path) must be making money from at least one of its two immediate neighbors on that route. Since customers are the only ones who pay, the rule is simple: **look at the intermediate AS. If neither of its neighbors on the path is a customer, it drops the traffic.**

- **Valid Middlemen:** Provider ↔ **Intermediate** ↔ Customer | Peer ↔ **Intermediate** ↔ Customer
- **Invalid Middlemen (Free Transit):** Peer ↔ **Intermediate** ↔ Peer | Provider ↔ **Intermediate** ↔ Peer | Provider ↔ **Intermediate** ↔ Provider.

Step-by-Step Explanations

AS1's Reachability (Can reach: AS2)

- **AS2 (Valid):** Direct link (Peer to Peer).

- **Why not AS3? (Path: AS1 ↔ AS2 ← AS3)**
 - **Intermediate:** AS2.
 - **Neighbors:** AS1 (Peer) and AS3 (Provider).
 - **Has customer?** No. AS2 drops it.
- **Why not AS5? (Path: AS1 ↔ AS2 ↔ AS5)**
 - **Intermediate:** AS2.
 - **Neighbors:** AS1 (Peer) and AS5 (Peer).
 - **Has customer?** No. AS2 drops it.

AS2's Reachability (Can reach: AS1, AS3, AS4, AS5, AS7)

- **AS1, AS5 (Valid):** Direct (Peer to Peer).
- **AS3 (Valid):** Direct (Customer to Provider).
- **AS4 (Valid):** Path: AS2 ← AS3 ↔ AS4.
 - **Intermediate:** AS3.
 - **Neighbors:** AS2 (Customer) and AS4 (Peer).
 - **Has customer?** Yes (AS2). Valid!
- **AS7 (Valid):** Path: AS2 ↔ AS5 → AS7.
 - **Intermediate:** AS5.
 - **Neighbors:** AS2 (Peer) and AS7 (Customer).
 - **Has customer?** Yes (AS7). Valid!
- **Why not AS6? (Path: AS2 ↔ AS5 ↔ AS6)**
 - **Intermediate:** AS5.
 - **Neighbors:** AS2 (Peer) and AS6 (Peer).
 - **Has customer?** No. AS5 drops it.

AS3's Reachability (Can reach: AS2, AS4, AS5, AS7)

- **AS2 (Valid):** Direct (Provider to Customer).
- **AS4 (Valid):** Direct (Peer to Peer).
- **AS5 (Valid):** Path: AS3 ↔ AS4 → AS5.
 - **Intermediate:** AS4.
 - **Neighbors:** AS3 (Peer) and AS5 (Customer).
 - **Has customer?** Yes (AS5). Valid!
- **AS7 (Valid):** Path: AS3 ↔ AS4 → AS5 → AS7.
 - **Intermediates:** AS4 (has customer AS5) and AS5 (has customer AS7). Both valid!
- **Why not AS1? (Path: AS3 → AS2 ↔ AS1)**
 - **Intermediate:** AS2.
 - **Neighbors:** AS3 (Provider) and AS1 (Peer).
 - **Has customer?** No. AS2 drops it.

AS4's Reachability (Can reach: AS2, AS3, AS5, AS7)

- **AS3 (Valid):** Direct (Peer to Peer).
- **AS5 (Valid):** Direct (Provider to Customer).
- **AS2 (Valid):** Path: AS4 ↔ AS3 → AS2.
 - **Intermediate:** AS3.
 - **Neighbors:** AS4 (Peer) and AS2 (Customer).
 - **Has customer?** Yes (AS2). Valid!
- **AS7 (Valid):** Path: AS4 → AS5 → AS7.
 - **Intermediate:** AS5.
 - **Neighbors:** AS4 (Provider) and AS7 (Customer).

- **Has customer?** Yes (AS7). Valid!

AS5's Reachability (Can reach: AS2, AS3, AS4, AS6, AS7, AS8)

- **AS2, AS6 (Valid):** Direct (Peer to Peer).
- **AS4 (Valid):** Direct (Customer to Provider).
- **AS7 (Valid):** Direct (Provider to Customer).
- **AS3 (Valid):** Path: AS5 ← AS4 ↔ AS3.
 - **Intermediate:** AS4.
 - **Neighbors:** AS5 (Customer) and AS3 (Peer).
 - **Has customer?** Yes (AS5). Valid!
- **AS8 (Valid):** Path: AS5 ↔ AS6 → AS8.
 - **Intermediate:** AS6.
 - **Neighbors:** AS5 (Peer) and AS8 (Customer).
 - **Has customer?** Yes (AS8). Valid!
- **Why not AS1? (Path: AS5 ↔ AS2 ↔ AS1)**
 - **Intermediate:** AS2.
 - **Neighbors:** AS5 (Peer) and AS1 (Peer).
 - **Has customer?** No. AS2 drops it.

AS6's Reachability (Can reach: AS5, AS7, AS8)

- **AS5 (Valid):** Direct (Peer to Peer).
- **AS8 (Valid):** Direct (Provider to Customer).
- **AS7 (Valid):** Path: AS6 ↔ AS5 → AS7.
 - **Intermediate:** AS5.
 - **Neighbors:** AS6 (Peer) and AS7 (Customer).
 - **Has customer?** Yes (AS7). Valid!
- **Why not AS4? (Path: AS6 ↔ AS5 ← AS4)**
 - **Intermediate:** AS5.
 - **Neighbors:** AS6 (Peer) and AS4 (Provider).
 - **Has customer?** No. AS5 drops it.

AS7's Reachability (Can reach: AS2, AS3, AS4, AS5, AS6, AS8)

- **AS5 (Valid):** Direct (Customer to Provider).
- **AS8 (Valid):** Direct (Peer to Peer).
- **AS2, AS4, AS6 (Valid):** Because AS7 is a customer of AS5, whenever AS7 sends traffic *through* AS5, AS5's neighbors on that path are always AS7 (Customer) and someone else. Therefore, AS5 *always* has a customer and will legally route the traffic to its peers (AS2, AS6) and its provider (AS4).
- **AS3 (Valid):** Path: AS7 ← AS5 ← AS4 ↔ AS3.
 - **Intermediates:** AS5 (has customer AS7) and AS4 (has customer AS5). Both valid!
- **Why not AS1? (Path: AS7 ← AS5 ↔ AS2 ↔ AS1)**
 - **Intermediates:** AS5 (Valid, has customer AS7) and AS2.
 - **AS2's Neighbors:** AS5 (Peer) and AS1 (Peer).
 - **Does AS2 have a customer?** No. The path dies at AS2.

AS8's Reachability (Can reach: AS5, AS6, AS7)

- **AS6 (Valid):** Direct (Customer to Provider).
- **AS7 (Valid):** Direct (Peer to Peer).

- **AS5 (Valid):** Path: AS8 ← AS6 ↔ AS5.
 - **Intermediate:** AS6.
 - **Neighbors:** AS8 (Customer) and AS5 (Peer).
 - **Has customer?** Yes (AS8). Valid!
- **Why not AS4? (Path: AS8 ← AS6 ↔ AS5 ← AS4)**
 - **Intermediates:** AS6 (Valid, has customer AS8) and AS5.
 - **AS5's Neighbors:** AS6 (Peer) and AS4 (Provider).
 - **Does AS5 have a customer?** No. The path dies at AS5.
- **Why not AS4? (Path: AS8 ← AS7 ↔ AS5 ← AS4)**
 - **Does AS7 have a customer?** No. The path dies at AS7.