CSC175 Spring 2026 Sample Exam Questions ANS

Z. Gu

Spring 2026

Lecture 3 - Links

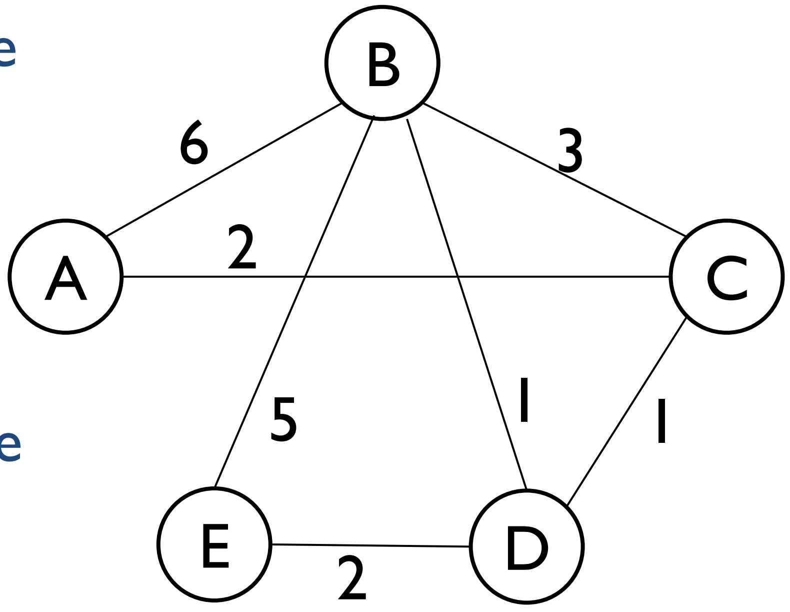
- Consider a network path between a source host (A) and a destination host (B) that passes through two routers (R1 and R2) connected in series. The routers use store-and-forward packet switching. You are given the following network parameters:
  - Packet size (L): 1,500 bytes
  - Transmission rate (R): 10 Mbps for all three links
  - Link lengths (d):
    - Link 1 (A to R1): 4,000 km
    - Link 2 (R1 to R2): 2,000 km
    - Link 3 (R2 to B): 1,000 km
  - Propagation speed (s): 2×10^8 m/s for all links
  - Processing and Queuing delays: Assume both are negligible (0 ms)
- Questions:
- a) What is the transmission delay (d_{trans}) for a single link in milliseconds?
- b) What is the total propagation delay (d_{prop}) across all three links combined in milliseconds?
- c) Calculate the total end-to-end delay from the moment Host A begins transmitting the first bit until Host B receives the last bit of the packet.

Lecture 3 - Links ANS

- **Part 1: Single Link Transmission Delay**
  - **Formula:** $d_{trans} = \frac{L}{R}$
  - Convert bytes to bits: $L = 1{,}500$ bytes $\times$ 8 bits/byte $= 12{,}000$ bits
  - Convert Mbps to bps: $R = 10$ Mbps $= 10{,}000{,}000$ bps
  - $d_{trans} = \frac{12{,}000}{10{,}000{,}000} = 0.0012$ seconds $= \mathbf{1.2\ ms}$
- **Part 2: Total Propagation Delay**
  - **Formula:** $d_{prop} = \frac{d}{s}$
  - Total distance $d = 4{,}000 + 2{,}000 + 1{,}000 = 7{,}000$ km $= 7{,}000{,}000$ m
  - Total $d_{prop} = \frac{7{,}000{,}000 \text{ m}}{2\times10^8 \text{ m/s}} = 0.035$ seconds $= \mathbf{35\ ms}$
- **Part 3: Total End-to-End Delay**
  - **Formula:** $d_{total} = N \times d_{trans} +$ Total d_{prop} (where N is the number of links).
  - Because the two routers use store-and-forward packet switching, the packet must be fully received before it is transmitted again. Across 3 links, the packet is transmitted 3 separate times.
  - Total Transmission Delay: 3×1.2 ms $= 3.6$ ms
  - Total Delay: $d_{total} = 3.6$ ms $+ 35$ ms $= \mathbf{38.6\ ms}$

Lecture 5.1 Shortest Paths Algorithms

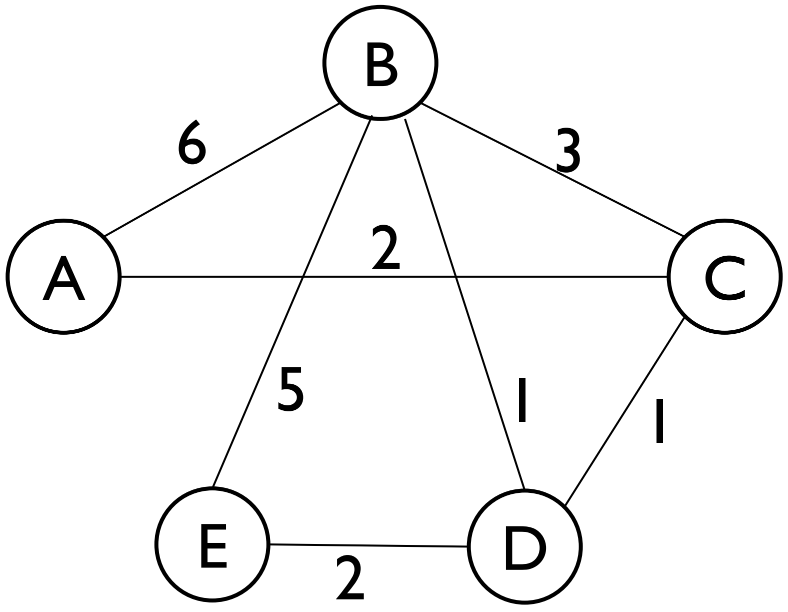
- Consider an Autonomous System with 5 routers (A, B, C, D, E) connected by bidirectional (undirected) point-to-point links. The link costs are shown in the figure:
  - Use Dijkstra's Algorithm to compute the shortest path tree from the **Source Router A**.
  - List the **Visit Order** of the routers.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



Lecture 5.1 Shortest Paths Algorithms ANS

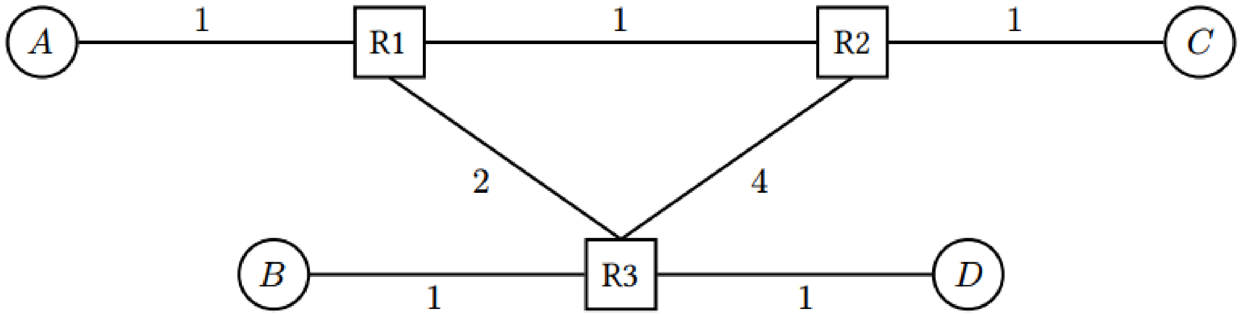
- **Visit Order:** A, C, D, B, E
- Router **B** gets updated *twice*.
- First, it is discovered directly by A (Cost 6).
- Then, visiting C finds a better path (2 + 3 = 5).
- Finally, visiting D finds the *optimal* path (3 + 1 = 4).

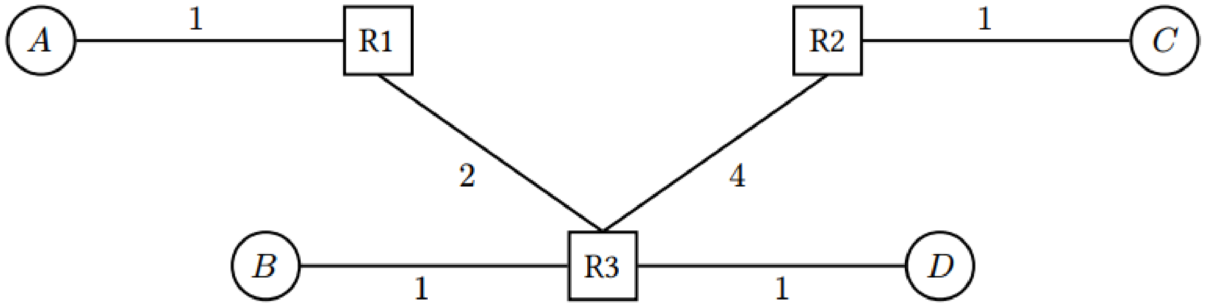
Node	SD	PN
A	0	
B	~~6~~ ~~5~~ 4	~~A~~ ~~C~~ D
C	2	A
D	3	C
E	5	D



Lecture 5.2 - Distance-Vector

- Alice (A), Bob (B), Connie (C), and Diego (D) are connected to the local network, which runs the distance-vector algorithm. Assumptions:
  - Static routes are installed at time t = 0.
  - Routers send periodic advertisements every 2 seconds, starting at t = 0. (Simplifying assumption assuming synchronized clocks for all routers.)
  - Routing table entries expire after TTL=10 seconds of receiving no advertisements.
  - Every second, each router (1) expires routes based on TTL, then (2) processes advertisements and updates its table, then (3) sends out advertisements if t is even (every 2 seconds).
  - Link costs correspond to packet travel times (in seconds). Ignore processing and queuing delays.
- Q1 Fill in R1's table at steady state. If a host is directly connected, the next hop is "Direct"
- After the network converges, the R1 - R2 link is broken, and split horizon is enabled on all routers.
- Q2 Fill in R1's table at the new steady state after the R1 - R2 link is broken.
- Q3 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 Q2? Assume split horizon but no route poisoning.
- Q4 Redo Q3 assuming both split horizon and route poisoning



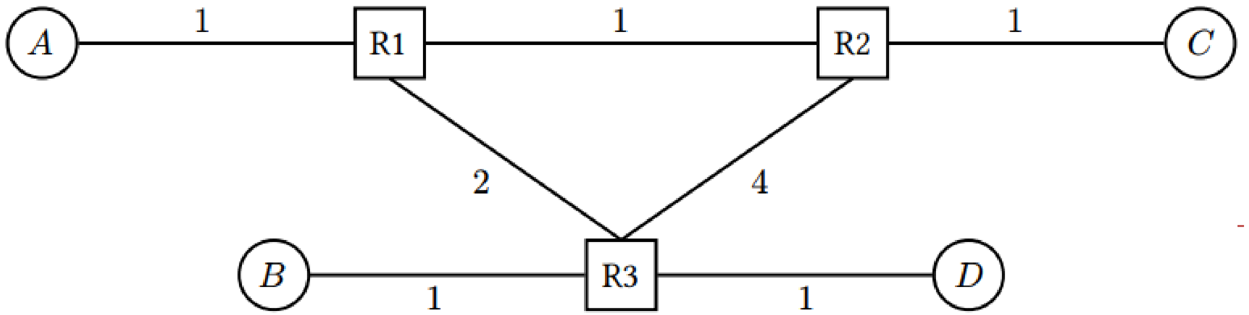


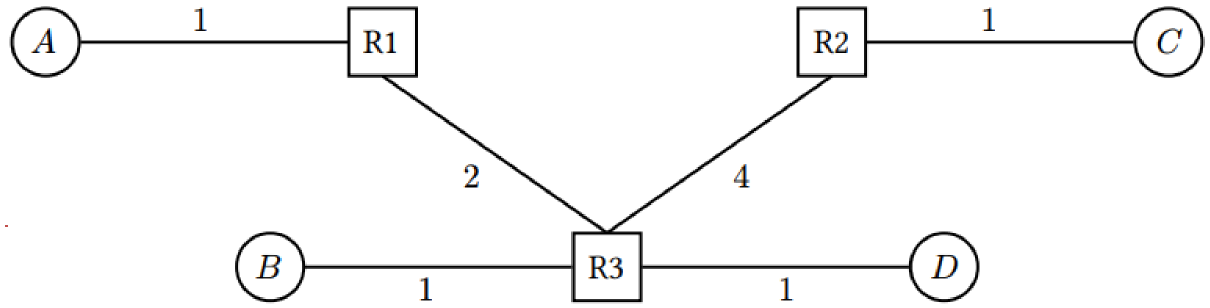
Lecture 5.2 - Distance-Vector ANS

- Q1 Fill in R1's table at steady state. If a host is directly connected, the next hop is "Direct" (table on left)
- After the network converges, the R1 - R2 link is broken, and split horizon is enabled on all routers.
- Q2 Fill in R1's table at the new steady state after the R1 - R2 link is broken. (table on right)
- Q3 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 Q2?
- ANS: The route to C is the only route that changes for R1. R1's route to C expires at t = 20, this means that the last possible advertisement for the route to C was sent from R1 to R3 at t = 18. (After t = 20, R1 cannot advertise the old route to C anymore; it must wait for a new usable route to come back from R3.) It reaches R3 after 2 seconds at t = 20. R3 had a route to C via R1 (R3 – R2 – R2 - C), which will expire at t = 20 + TTL = 30 at the earliest when it will accept new route to C via R2 from R2. Then R3 advertises back to R1 (2 seconds) to reach steady state at t = 32.

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

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

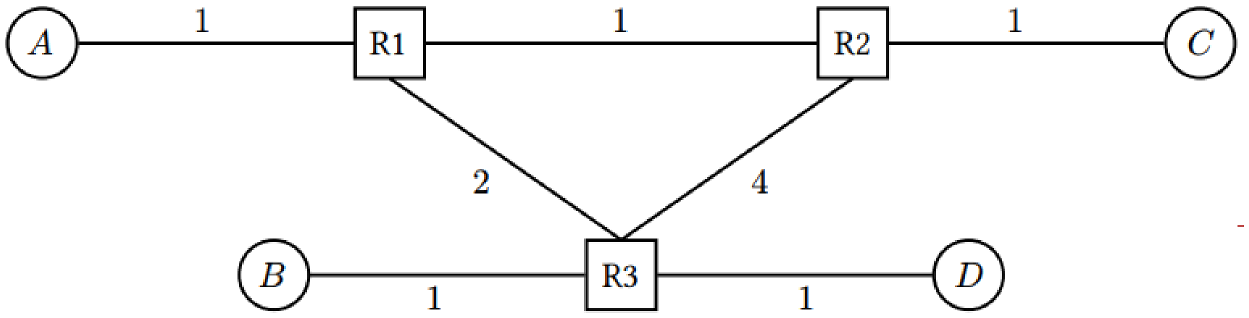




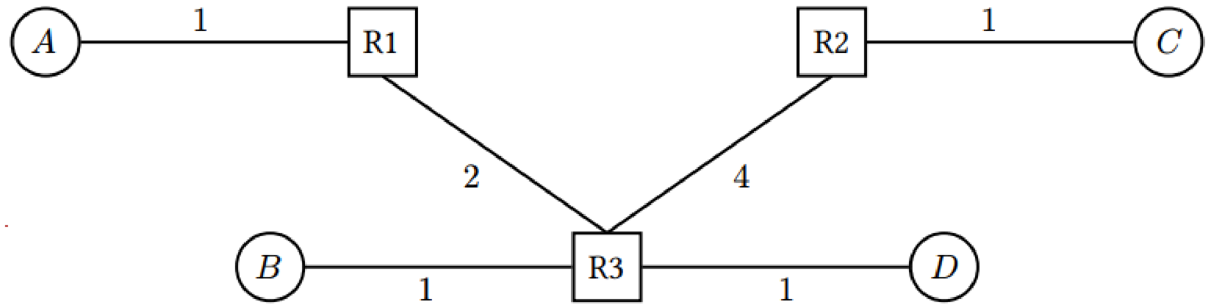
Lecture 5.2 - Distance-Vector ANS

- Q4 Redo Q3 assuming both split horizon and route poisoning
- ANS: For **split horizon + route poisoning**, R1 does **not** need to wait until R3's old route to C times out at t = 30. The update happens sooner:
  - t = 20: R1's route to C expires, and R1 advertises C as unreachable.
  - t = 22: R3 receives that poison from R1, and also still has the valid C advertisement from R2. R3 switches to C via R2 with cost 5 and advertises it to R1.
  - t = 24: R1 receives that update from R3 and installs C via R3 with cost 7.
- So R1 reaches the new steady state at t = 24, not t = 32.

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

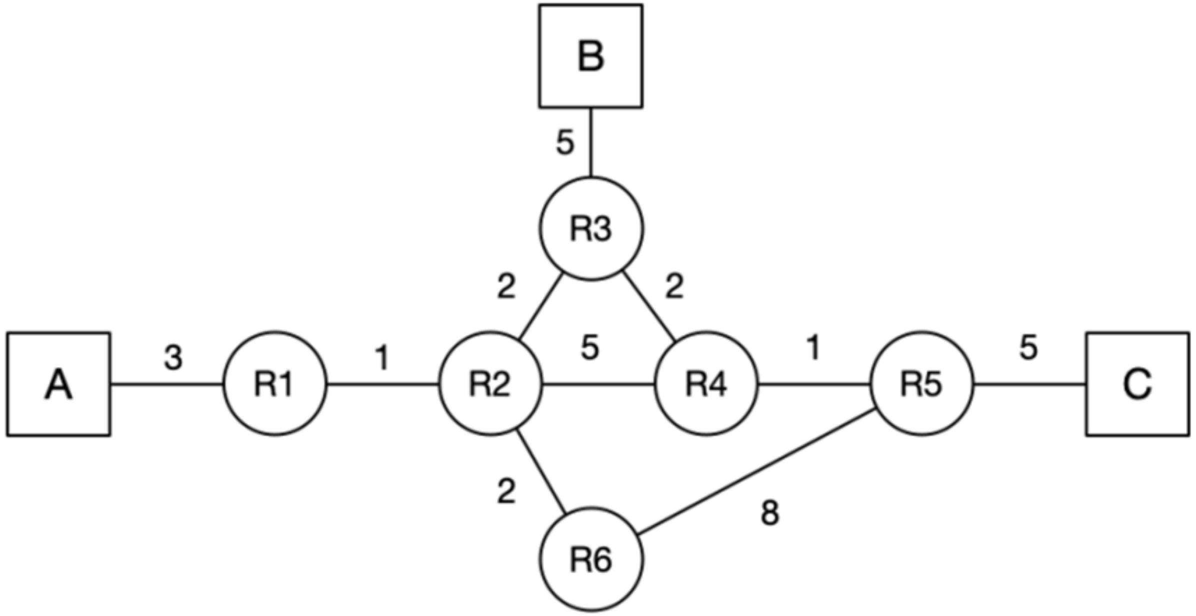


To	Next Hop Router	Cost
A	Direct	1
B	R3	3
C	R3	7
D	R3	3



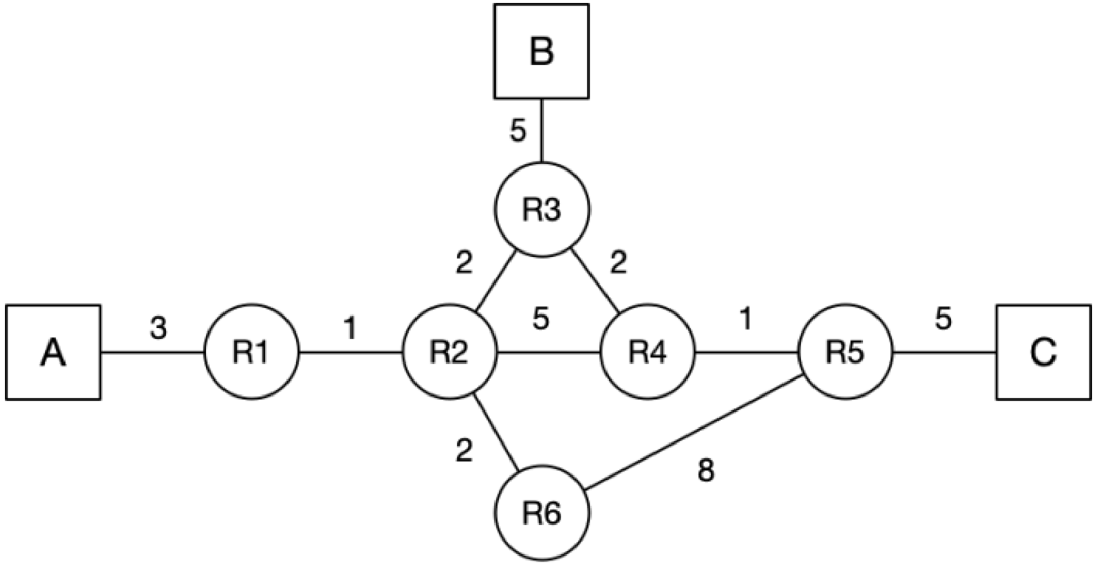
Lecture 5.3 - Link-State

- 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).
- Q1. Suppose that the link between R3 and R4 goes down. R3 and R4 have recomputed their routes, but they have not yet sent updates. What route will a packet from A to C take?
- Q2. Suppose that the link between R2 and R3 goes down. R2 and R3 have recomputed their routes, but have not yet sent updates. What route will a packet from A to C take?
- Q3. Assume that A sends a packet to C, and at time t=0, it arrives at R1. At t=0.5s, the link between R4 and R5 goes down, and R4 and R5 instantaneously recognize and recompute their routes. Assume that link-state advertisements are processed and propagated instantaneously. A link's propagation delay is equal to the link costs in the diagram (in seconds), i.e. R1 - R2 has a 1-second delay, R2 - R3 has 2-second delay, etc). You can ignore all processing and queuing delays. Does the packet reach its destination? If so, write down the route the packet from A to C takes.



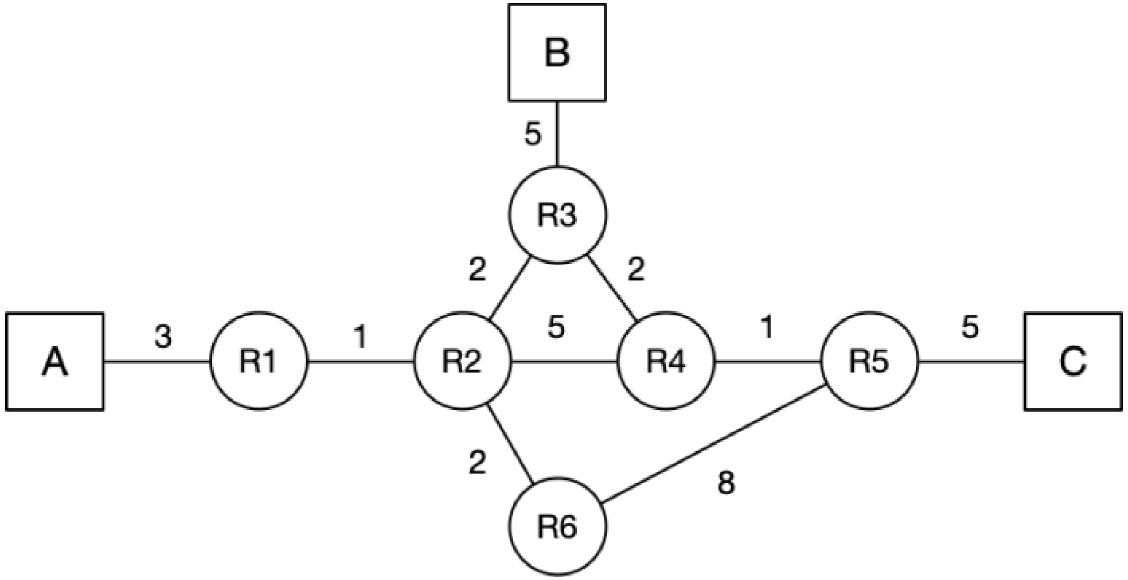
Lecture 5.3 - Link-State ANS

- Q1. Suppose that the link between R3 and R4 goes down. R3 and R4 have recomputed their routes, but they have not yet sent updates. What route will a packet from A to C take?
- ANS: A → R1 → R2 → R3 → R2 → R3 (loop)
- Explanation: Because R3 and R4 have not yet sent their Link-State Advertisements (LSAs), R1 and R2 are unaware of the failure and continue forwarding the packet toward R3, assuming it is the shortest path. However, R3 knows its link to R4 is down, so it has locally recomputed its next-best path to C, which happens to go back through R2 (R3 → R2 → R4 → R5 → C). This creates a temporary routing loop until R1 and R2 receive the updates
- Q2. Suppose that the link between R2 and R3 goes down. R2 and R3 have recomputed their routes, but have not yet sent updates. What route will a packet from A to C take?
- ANS: A → R1 → R2 → R4 → R5 → C
- Explanation: R1 is unaware of the failure and sends the packet to R2. However, R2 is aware that its link to R3 is down. R2 locally recalculates an alternate route to C via R4 and R5. Because this new path does not bounce back to an uninformed router that points back to R2, the packet successfully reaches its destination without looping.

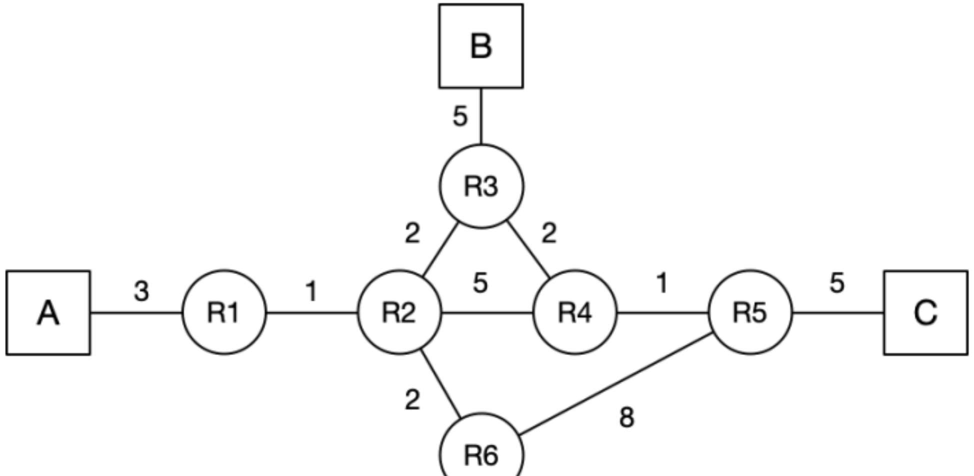


Lecture 5.3 - Link-State ANS

- Q3. ANS: A → R1 → R2 → R3 → R2 →R6 → R5 → C
- Explanation: Because LSA propagation delays are tied to link costs, the failure updates spread across the network at the exact same time the packet is traveling. As the packet advances (A → R1 → R2 → R3), the routers it hits are actively updating their tables based on newly arriving LSAs. When the packet hits R3, R3 has just learned of the failure and redirects it back to R2. R2 has also just updated its table and dynamically reroutes the packet through R6 and R5 to safely reach C.



Lecture 5.3 - Link-State ANS

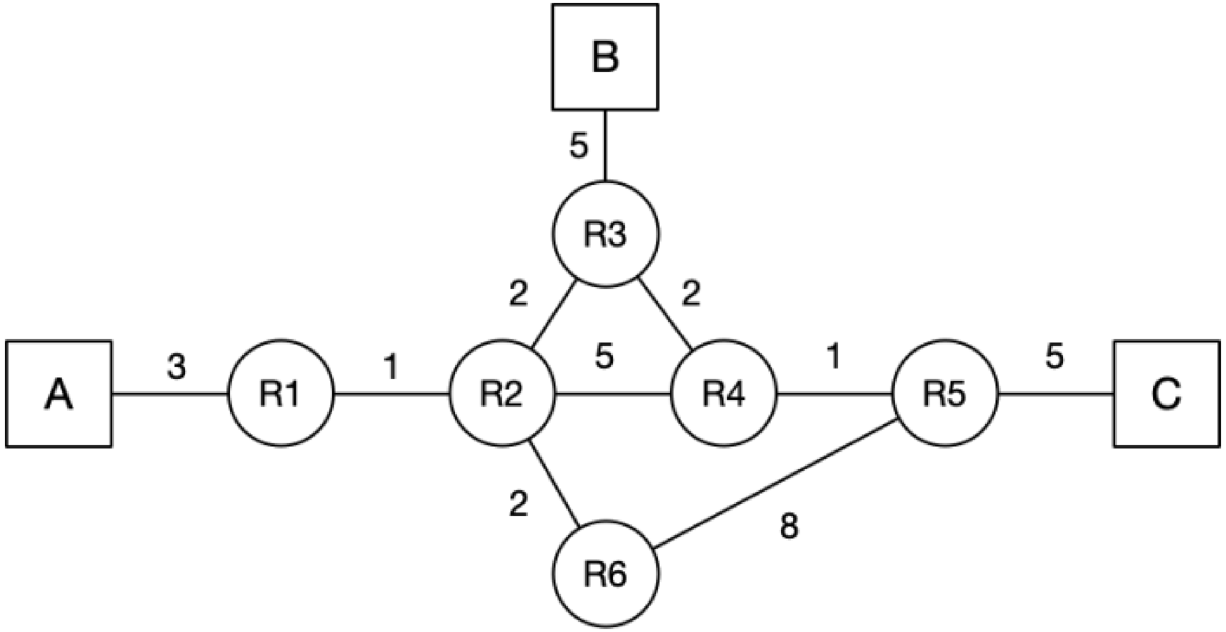


- Q3. in detail. Assume that A sends a packet to C, and at time t=0, it arrives at R1. At t=0.5s, the link between R4 and R5 goes down.
- ANS: A → R1 → R2 → R3 → R2 →R6 → R5 → C
- Explanation: Here is the detailed timeline of events for Q4.5, which is driven by the "race" between the packet's physical propagation delay and the Link-State Advertisement (LSA) propagation delay. Both delays are strictly tied to the link costs.
- t = 0.0s. Packet from host A toward Host C arrives at R1, and R1 forwards it to R2. The propagation delay for the R1-R2 link is 1 second.
- t = 0.5s. While the packet is mid-flight between R1 and R2, the link between R4 and R5 suddenly fails. R4 and R5 immediately recognize the failure and generate LSAs detailing the broken link. These LSAs begin propagating across the network at a speed dictated by the link costs.
- t = 1.0s. Packet arrives at R2. LSA Status: The failure LSA from R4 has not yet reached R2 because it has only been propagating for 0.5 seconds, and the delay (cost) from R4 to R2 is longer than that. Action: Because R2 is unaware of the broken link, its routing table still indicates that the shortest path to C is via R3. R2 forwards the packet to R3. The delay for the R2-R3 link is 2 seconds.
- t = 1.0s – 3.0s. Packet is in transit between R2 and R3. Meanwhile, the LSAs are continuing to propagate across the network toward R2 and R3.
- t = 3.0s. Packet arrives at R3. LSA Status: By this time, the LSA from R4 reaches R3 at time 2.5s. R3 processes it instantaneously and updates its routing table. Action: R3 recalculates its shortest path to C because it now knows the R4-R5 link is down. The new shortest path routes back through R2 (R3 → R2 → R6 → R5 → C). R3 forwards the packet back to R2, which takes another 2 seconds.
- t = 3.0s – 5.0s. Packet travels from R3 back to R2. During this window, the LSA from R4 reaches R2 at time 4.5s, and R2 instantaneously updates its own routing table.
- t = 5.0s. Packet arrives back at R2. LSA Status: R2 is now fully aware of the R4-R5 link failure. Action: R2 recalculates its next-best path to C. With the old route invalidated, R2 determines the new optimal route is via R6 (R2 → R6 → R5 → C). R2 forwards the packet to R6.
- t > 5.0s. The packet successfully continues along the updated path, safely traveling from R6 → R5 → Host C.

t (s)	Event
0	Packet arrives at R1
0.5	R4-R5 link breaks
1	Packet arrives at R2
2.5	LSA from R4 arrives at R3. R3 reconfigures its route to C as R3-R2-R6-R5-C
3	Packet arrives at R3
4.5	LSA from R4 arrives at R2. R2 reconfigures its route to C as R2-R6-R5-C
5	Packet arrives at R2
7	Packet arrives at R6
15	Packet arrives at R5
20	Packet arrives at C

Lecture 5.3 - Link-State ANS

- Q3. continued. Why does the LSA from R4 reach R2 at time 4.5s?
- ANS: LSAs are flooded across all paths. So R4 and R5 both send our LSAs with the R4-R5 link failure to all their neighbors.
  - Through path R4 → R3 → R2: LSA arrives at R2 at time 4.5s.
  - Through path R4 → R2: LSA arrives at R2 at time 5.5s.
  - Through path R5 → R6 → R2: LSA arrives at R2 at time 10.5s.
- Therefore, the earliest time that LSA from R4 reaches R2 is at time 4.5s, making R2 aware that the R4-R5 link is dead, and allowing it to correctly recalculate the route and forward the packet to R6.



Lecture 6 - IP Addressing

- Suppose hofstra.edu is the provider AS for Engineering (EE), CS, Math, and Biology, and needs to assign IPv4 addresses to them. Assume that CIDR (Classless Inter-Domain Routing) addressing is used, and that hofstra.edu has the prefix: 198.51.0.0/16. The address space is allocated as follows:
  - Chemistry: 198.51.0.0/18
  - Math: 198.51.128.0/18
  - The block 198.51.192.0/18 is reserved for EE and CS
  - The block 198.51.64.0/18 is currently unassigned
- Questions
- 1) Which addresses are included in the Math department's prefix? How many addresses are in this range?
- 2) The block 198.51.192.0/18 is reserved for EE and CS. Assign equal halves of this address space to the two departments.
- 3) You want to start a new department of Data Science, and assign it an unused address range. You foresee that no more than 90 people will enroll. Assuming one address per person, what prefix would you assign to minimize unused/wasted addresses?
- 4) After assigning the Data Science prefix, suppose Biology needs an address block for 30 people.
  What is the smallest unused prefix you can assign to Biology without overlapping any existing assignments?

Decimal	Binary
198	11000110
51	00110011
192	11000000
224	11100000
64	01000000
128	10000000

Lecture 6 - IP Addressing ANS

- **Math:** 198.51.128.0/18
  - Binary: 11000110 . 00110011 . 10 000000 . 00000000
  - Range: 198.51.128.0 to 198.51.191.255
  - Addresses: 16384
- **EE / CS split of 198.51.192.0/18:**
  - Binary: 11000110 . 00110011 . 11 000000 . 00000000
  - EE: 198.51.192.0/19
  - CS: 198.51.224.0/19
  - Binary:
    - 11000110 . 00110011 . 110 00000 . 00000000
    - 11000110 . 00110011 . 111 00000 . 00000000
- **Data Science:** 198.51.64.0/25
  - Binary: 11000110 . 00110011 . 01000000 . 00000000
- **Biology:** 198.51.64.128/27
  - Binary: 11000110 . 00110011 . 01000000 . 10000000

Decimal	Binary
198	11000110
51	00110011
192	11000000
224	11100000
64	01000000
128	10000000

Lecture 6 - IP Addressing ANS Details

- Suppose hofstra.edu is the provider AS for Engineering (EE), CS, Math, and Biology, and needs to assign IPv4 addresses to them. Assume that CIDR (Classless Inter-Domain Routing) addressing is used, and that hofstra.edu has the prefix: 198.51.0.0/16. The address space is allocated as follows:
  - Chemistry: 198.51.0.0/18
  - Math: 198.51.128.0/18
  - The block 198.51.192.0/18 is reserved for EE and CS
  - The block 198.51.64.0/18 is currently unassigned
- ANS: 198.51.0.0/16 in binary:198 = 1100011051 = 00110011So the reserved /16 block is:198.51.0.0/16= 11000110 . 00110011 . 00000000 . 00000000
- 1) Which addresses are included in the Math department's prefix? How many addresses are in this range?
  - Math is assigned: 198.51.128.0/18 = 11000110 . 00110011 . 10 000000 . 00000000
  - Since a /18 fixes the first 18 bits, the host bits are the remaining 14 bits. So the address range is:
  - 11000110 . 00110011 . 10 000000 . 00000000
    through
    11000110 . 00110011 . 10 111111 . 11111111
  - In decimal: 198.51.128.0 through 198.51.191.255
  - **Number of addresses:** A /18 has: 2^(32 - 18) = 2^14 = 16384
- 2) The block 198.51.192.0/18 is reserved for EE and CS. Assign equal halves of this address space to the two departments.
  - The block is: 198.51.192.0/18 = 11000110 . 00110011 . 11 000000 . 00000000
  - To split a /18 into two equal halves, use two /19 prefixes.
  - **EE** 198.51.192.0/19 = 11000110 . 00110011 . 110 00000 . 00000000
  - **CS** 198.51.224.0/19 = 11000110 . 00110011 . 111 00000 . 00000000

Lecture 6 - IP Addressing ANS Details

- 3) You want to start a new department of Data Science, and assign it an unused address range. You foresee that no more than 90 people will enroll. Assuming one address per person, what prefix would you assign to minimize unused/wasted addresses?
  - Need at least 90 addresses. Smallest power of 2 that is ≥ 90 is: 128 = 2^7. So the network prefix length is: 32 - 7 = /25. The unused block is:
  - 198.51.64.0/18 = 11000110 . 00110011 . 01 000000 . 00000000
  - A valid minimal assignment is: 198.51.64.0/25 = 11000110 . 00110011 . 01000000 . 00000000
  - This gives 128 addresses, which is enough for 90 students. (You can use arbitrary bits for bits 19-25.)
- 4) After assigning the Data Science prefix, suppose Biology needs an address block for 30 people. What is the smallest unused prefix you can assign to Biology without overlapping any existing assignments?
- Need at least 30 addresses. Smallest power of 2 that is ≥ 30 is 32 = 2^5. So the network prefix length is: 32 - 5 = /27
- After assigning: 198.51.64.0/25 = 11000110 . 00110011 . 01000000 . 00000000, the next aligned unused /27 block is: 198.51.64.128/27 = 11000110 . 00110011 . 01000000 . 10000000
  - That is a valid unused prefix and does not overlap any existing assignment. (The last octet can be 1xx00000, as long as it starts with a 1 (xx can be arbitrary bits).)

Lecture 7 - Routers

- Consider a router in a network that uses a least-cost routing protocol, with ties broken by taking the route from the link with the smallest port number. The router has 4 ports and its default route sends all traffic onto port 1. Table 1 lists the routes that our router sees advertised at each port. You can find some useful binary conversions in the table below. For the following 7 subparts, determine which ports the packets with the following destinations are forwarded to based on the advertisements given above. Give brief explanation for each.
- Q1 A packet with destination 3.4.0.1
- Q2 A packet with destination 4.0.0.1
- Q3 A packet with destination 2.2.208.1
- Q4 A packet with destination 2.3.0.10
- Q5 A packet with destination 2.2.204.13
- Q6 A packet with destination 1.1.21.7
- Q7 A packet with destination 2.2.96.22

Port	Destination	Cost
1	1.0.0.0/8	10
	2.1.0.0/16	15
	2.2.192.0/20	12
	4.0.0.0/8	10
	2.2.96.0/17	15
2	1.1.0.0/16	8
	2.2.128.0/17	14
	4.0.0.0/8	8
3	3.0.0.0/8	10
	2.2.204.0/20	13
	1.0.10.0/24	8
4	3.4.0.0/16	11
	1.1.0.0/16	8
	2.2.0.0/17	14

Table 1: Routes Advertised at each port

Decimal	Binary
192	11000000
128	10000000
96	01100000
204	11001100
208	11010000
64	01000000
32	00100000

Lecture 7 - Routers ANS

- **Three rules:**
  - **Longest Prefix Match (LPM):** Select the route with the most specific subnet mask (highest /X number).
  - **Lowest Cost:** If multiple routes tie for the longest prefix, choose the one with the lowest cost.
  - **Smallest Port Number:** If costs are also identical, break the tie by choosing the lower port number.
    If no prefixes match, use the Default Route (Port 1).
- **Q1 (3.4.0.1):** Matches 3.0.0.0/8 (Port 3) and 3.4.0.0/16 (Port 4). Port 4 wins via **LPM** (/16 is more specific than /8).
- **Q2 (4.0.0.1):** Ties on LPM between 4.0.0.0/8 on Port 1 (Cost 10) and Port 2 (Cost 8). Port 2 wins via **Lowest Cost**. (Such duplicate routing entries should not exist in a real router, since the higher cost route will never be taken.)
- **Q3 (2.2.208.1):** 208 (11010000 in binary) only matches 2.2.128.0/17 on Port 2 because it successfully checks the first bit (1) of the 3rd octet. It fails the more specific /20 checks, 2.2.192.0/20, 2.2.204.0/20, since 1101 does not match 1100 in the 3rd octet. So Port 2 wins via **LPM**.
- **Q4 (2.3.0.10):** Does not match any of the advertised 2.1 or 2.2 prefixes. It falls back to the **Default Route** on Port 1.
- **Q5 (2.2.204.13):** 204 (11001100) matches two /20 subnets:, 2.2.192.0/20, 2.2.204.0/20, since they all have 1100 in the 3rd octet. Among Port 1 (Cost 12) and Port 3 (Cost 13), Port 1 wins via **Lowest Cost**.
- **Q6 (1.1.21.7):** Ties on LPM between two 1.1.0.0/16 routes with equal costs (Cost 8 on both Port 2 and Port 4). Port 2 wins via the **Smallest Port Number** tie-breaker.. (Such duplicate routing entries should not exist in a real router.)
- **Q7 (2.2.96.22):** 96 (01100000) matches two overlapping /17 routes 2.2.96.0/17, 2.2.0.0/17, (not 2.2.128.0/17), since both check that the first bit is 0: Port 1 (Cost 15) and Port 4 (Cost 14). Port 4 wins via **Lowest Cost**.

Port	Destination	Cost
1	1.0.0.0/8	10
	2.1.0.0/16	15
	2.2.192.0/20	12
	4.0.0.0/8	10
	2.2.96.0/17	15
2	1.1.0.0/16	8
	2.2.128.0/17	14
	4.0.0.0/8	8
3	3.0.0.0/8	10
	2.2.204.0/20	13
	1.0.10.0/24	8
4	3.4.0.0/16	11
	1.1.0.0/16	8
	2.2.0.0/17	14

Table 1: Routes Advertised at each port

Decimal	Binary
192	11000000
128	10000000
96	01100000
204	11001100
208	11010000
64	01000000
32	00100000

Lecture 7 – Routers 2

- Consider a router running longest prefix matching to forward packets.
- Q1 Given the current routing table, use route aggregation to build a new table, such that both tables produce the same forwarding decisions, and every IPv4 address matches only one prefix. Write one IP prefix per box.

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

Destination	Port

New Table

Lecture 7 – Routers 2

- Q2 Using binary tries to run longest prefix matching. 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. Draw the resulting trie. What is the height of the resulting binary trie?
- Q3 Using binary tries to run longest prefix matching. Consider building a binary trie out of a forwarding table with these two prefixes: 17.0.0.0/8, 18.0.0.0/8. What is the height of the resulting binary trie? What is the earliest branching point?

Lecture 7 – Routers 2 ANS

- Consider a router running longest prefix matching to forward packets.
- Q1 Given the current routing table, use route aggregation to build a new table, such that both tables produce the same forwarding decisions, and every IPv4 address matches only one prefix. Write one IP prefix per box.
- ANS: First row and third row: stay the same.
- Second row: We want to use a prefix that 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 (3rd octet from the left) for both prefixes are 00000001 and 00000010, respectively. Therefore, bits 1 to 22 are the same for the two IP addresses, so we can have the IP prefix 128.1.0.0/22.

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

Destination	Port
128.1.0.0/24	1
128.1.1.0/22	2
128.1.3.0/24	3

New Table

Lecture 7 – Routers 2 ANS

- Q2 Using binary tries to run longest prefix matching. 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? Draw or verbally describe the resulting trie.
- ANS: A binary trie for IP prefixes branches one bit per level. Therefore, the height = length of the longest prefix. Given prefixes
- 17.0.0.0/8, 17.1.0.0/16, 17.1.1.0/24
- The longest prefix is /24, so: Height = 24.
- The trie is one long path (no branching), because each prefix extends the previous one. /8 is a prefix of /16; /16 is a prefix of /24. (See next page for picture drawing, which is optional.)

Lecture 7 – Routers 2 ANS

Step 1: Common prefix

All three prefixes start with **17**, whose first 8 bits are:

```
00010001  (this is 17 in binary)
```

So the trie begins as a **single chain of 8 levels**:

```
(root)
  |
  0
  |
  0
  |
  0
  |
  1
  |
  0
  |
  0
  |
  0
  |
  1   ← depth 8 → matches 17.0.0.0/8
```

Step 2: Extend to /16 (17.1.0.0/16)

Next 8 bits correspond to the second octet (1):

```
00000001
```

So we extend another chain:

```
(depth 8)
    |
    0
    |
    0
    |
    0
    |
    0
    |
    0
    |
    0
    |
    0
    |
    1   ← depth 16 → matches /16
```

Step 3: Extend to /24 (17.1.1.0/24)

Next 8 bits (third octet = 1):

```
00000001
```

Continue:

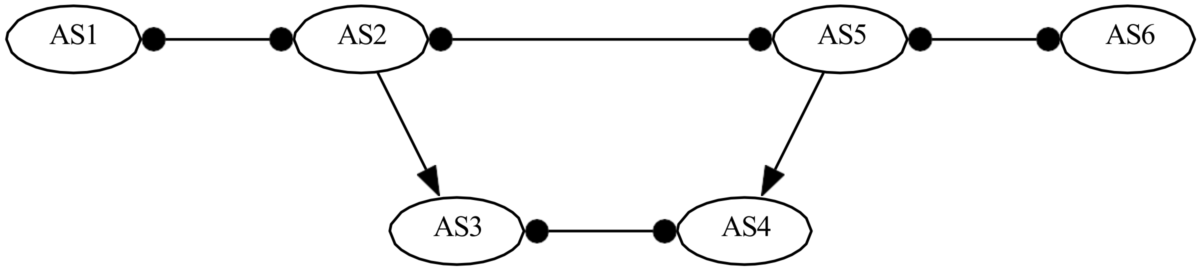
```
(depth 16)
    |
    0
    |
    0
    |
    0
    |
    0
    |
    0
    |
    0
    |
    0
    |
    1   ← depth 24 → matches /24
```

Lecture 7 – Routers 2 ANS

- Q3 Using binary tries to run longest prefix matching. Consider building a binary trie out of a forwarding table with these two prefixes: 17.0.0.0/8, 18.0.0.0/8. What is the height of the resulting binary trie? Draw or verbally describe the resulting trie.
- ANS: Trie height = length of the longest prefix. Given prefixes: 17.0.0.0/8, 18.0.0.0/8. The longest prefix is /8, so: Height = 8.
- Convert first octet to binary: 17 = 00010001, 18 = 00010010. They share the first 6 bits, then split at the 7th bit.
  - First 6 levels: shared path. At level 7: branch (0 vs 1). Then each continues to depth 8. (Picture drawing is optional.)

Lecture 8 - Inter-Domain Routing

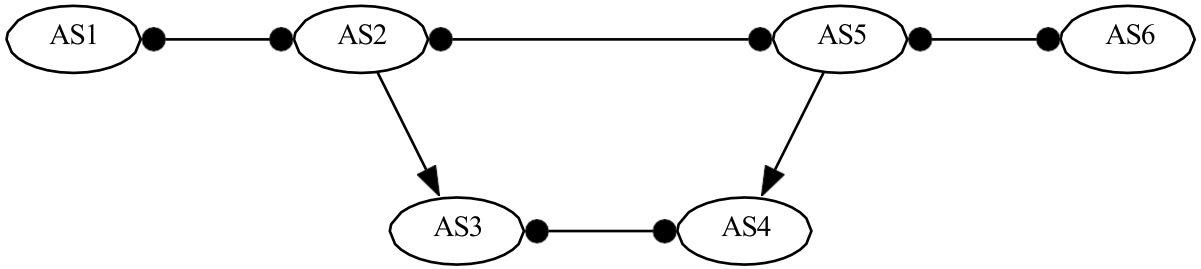
- Consider the AS graph below, where each AS follows the Gao-Rexford import and export policies. For each source/destination pair, select whether it is possible for packets to be sent from the source AS to the destination AS. In other words, is there an AS path from source to destination where all intermediate ASes agree to export the path?
- Q1 Source AS1, destination AS3.
- Q2 Source AS1, destination AS4.
- Q3 Source AS2, destination AS4.
- Q4 Why is reachability not guaranteed in this AS graph?
- Q5 On the graph below, draw at most 3 extra links, such that the resulting AS graph provides reachability.



- Under Gao-Rexford, a route is **legal** if it is **valley-free** and respects export preference rules:
  - an AS may learn a route from a **customer**, **peer**, or **provider**;
  - it prefers **customer routes** over **peer routes** over **provider routes**;
  - it exports **customer-learned routes** to everyone, but **peer/provider-learned routes only to customers**.
  - Every intermediate AS on a legal path has at least one customer neighbor along that path.

Lecture 8 - Inter-Domain Routing ANS

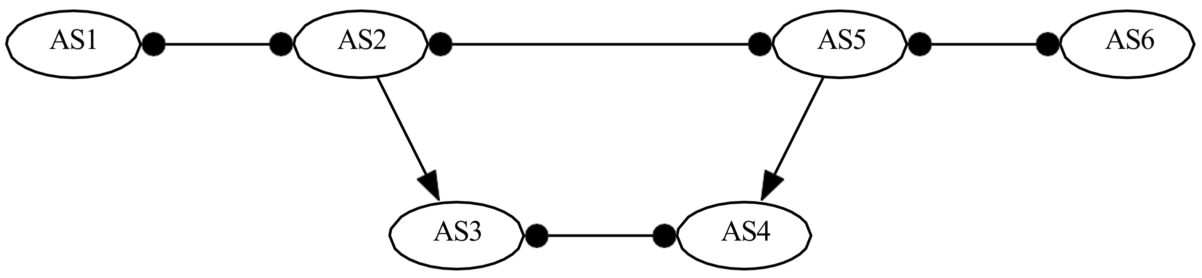
- Q1 Source AS1, destination AS3.
  - ANS: There are two possible paths:
  - AS1 → AS2 → AS3: This path **works** because the intermediate AS AS2 has a customer neighbor AS3 along that path. In other words, AS3 will advertise to AS2, and AS2 will advertise to AS1, creating the path.
  - AS1 → AS2 → AS5 → AS4 → AS3: This path **does not work** because AS2 does not have a customer neighbor along that path (both neighbors are peers).
- Q2 Source AS1, destination AS4.
  - ANS: There are two possible paths:
  - AS1 → AS2 → AS3 → AS4: This path **does not work** because AS3 does not have a customer neighbor along that path.
  - AS1 → AS2 → AS5 → AS4: This path **does not work** because AS2 does not have a customer neighbor along that path.
  - Since none of the paths are valid, it's not possible to send packets from AS1 to AS4.
- Q3 Source AS2, destination AS4.
  - ANS: There are two possible paths:
  - AS2 → AS3 → AS4: This path **doesn't work** because AS3 does not have a customer neighbor along that path.
  - AS2 → AS5 → AS4: This path **works** because AS5 has a customer neighbor AS4 along that path. In other words, AS4 will advertise to AS5, and AS5 will advertise to AS2, creating the path..



- Under Gao-Rexford, a route is **legal** if it is **valley-free** and respects export preference rules:
  - an AS may learn a route from a **customer**, **peer**, or **provider**;
  - it prefers **customer routes** over **peer routes** over **provider routes**;
  - it exports **customer-learned routes** to everyone, but **peer/provider-learned routes only to customers**.
  - Every intermediate AS on a legal path has at least one customer neighbor along that path.

Lecture 8 - Inter-Domain Routing ANS

- Q4 Why is reachability not guaranteed in this AS graph?
  - ANS: One of the requirements for reachability is that all Tier 1 ASes (with no provider) form a fully connected peering clique. However, this graph does not meet this requirement, e.g. AS1 is not connected to AS6.
- Q5 On the graph below, draw at most 3 extra links, such that the resulting AS graph provides reachability.
  - The simplest solution is to connect all the Tier 1 ASes to every other Tier 1 AS:
  - Peering link from AS1–AS5
  - Peering link from AS1–AS6
  - Peering link from AS2–AS6
  - Alternate solutions may exist, although they would not be able to use the conditions 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.



- Under Gao-Rexford, a route is **legal** if it is **valley-free** and respects export preference rules:
  - an AS may learn a route from a **customer**, **peer**, or **provider**;
  - it prefers **customer routes** over **peer routes** over **provider routes**;
  - it exports **customer-learned routes** to everyone, but **peer/provider-learned routes only to customers**.
  - Every intermediate AS on a legal path has at least one customer neighbor along that path.