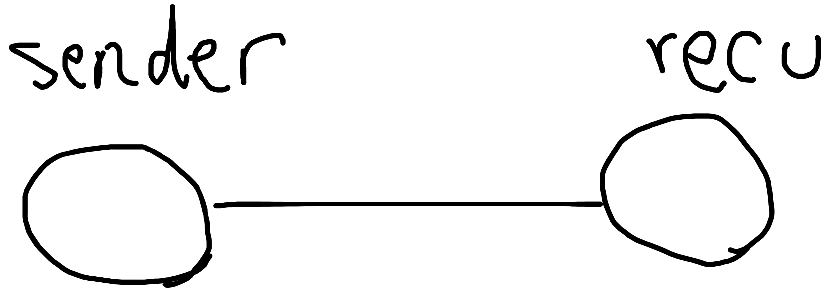
Tutorial #3

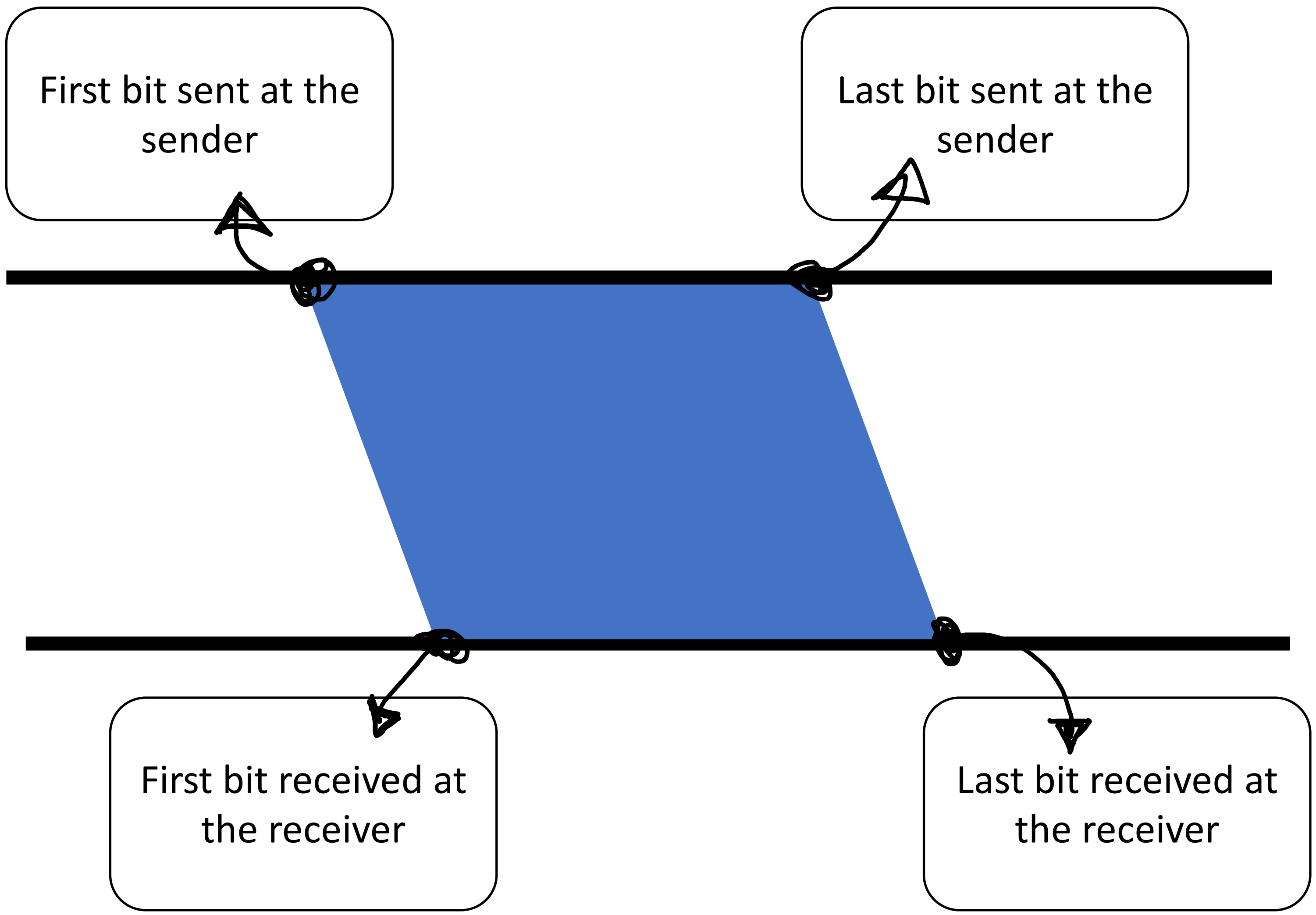
CSC458

Problem 1 – A

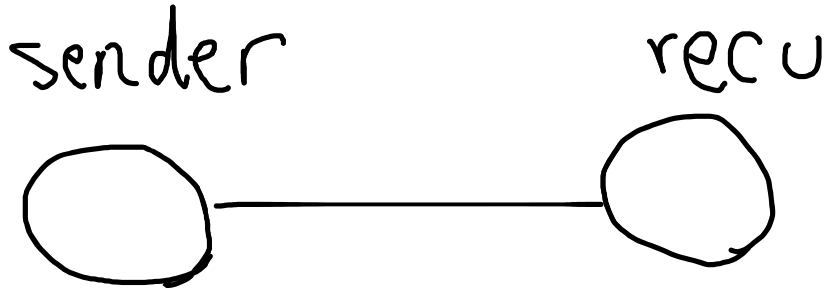


- We have a link, rate 100 Kbit/s, latency 1ms, MTU 100, sending 80 bytes of IP payload. How long does it take to transmit the data?
  - Ignore the Ethernet Header for now.

Propagation vs Transmission Delay



Problem 1 – A

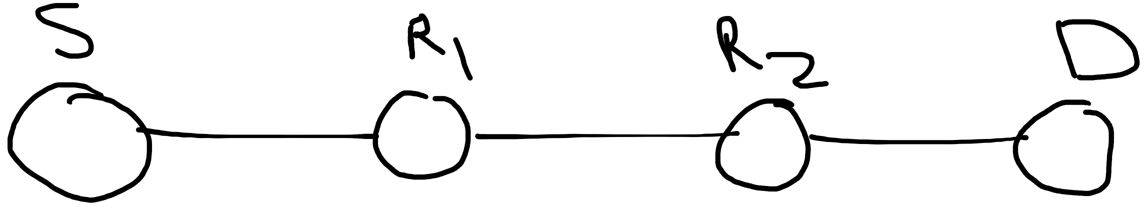


- **Step 1: Packet size**
  100 bytes × 8 = 800 bits

- **Step 2: Transmission time**
  t_tx = 800 / 100,000 = 0.008 s = 8 ms

- **Step 3: Add propagation delay**
  t_total = 8 ms + 1 ms = 9 ms

Problem 1 – B



- We have 3 back-to-back links, going through 2 intermediate switches. Similar numbers for the links. we have **store and forward** for the switches.

Problem 1 – B

- **Step 1: Packet size**
  100 B × 8 = 800 bits
- **Step 2: Per-link transmission time**
  t_tx = 800 / 100,000 = 0.008 s = 8 ms
- **Step 3: Per-link total delay**
  8 ms (tx) + 1 ms (prop) = 9 ms
- **Step 4: First packet arrival at destination**
  3 hops × 9 ms = 27 ms

Problem 1 – C

- Similar, but cut-through switching for the switches.

Problem 1 – C

- Packet size = 100 B → 800 bits. Header size = 20 B → 160 bits.
    - packet serialization time = 800/100,000 = 0.008 s = 8 ms
    - header serialization time = 160/100,000 = 0.0016 s = 1.6 ms.

- Source transmits at t = 0.
- Switch 1 begins forwarding at 2.6 ms, finishes at 10.6 ms.
- Switch 2 begins forwarding at 5.2 ms, finishes at 13.2 ms.
- Destination receives the last bit at 14.2 ms.

Problem 1 – D

- Let's go back to store and forward, Last link has MTU of 60.

Problem 1 – D

- **Links 1–2 (no fragmentation):**
  - On-wire size = 100 B → 100×8 = **800 bits**
  - Per-link tx time: **800/100,000 = 0.008 s = 8.0 ms**
  - Per-link total (tx + prop): **8.0 + 1.0 = 9.0 ms**
  - Arrival at Switch 1: **9.0 ms**; arrival at Switch 2: **9.0 + 9.0 = 18.0 ms**

- **Fragmentation for Link 3 (MTU 60):**
  - Each IP fragment must be ≤ 60 B **including** its 20 B IP header
  - Payload per fragment ≤ 40 B and (except maybe last) a multiple of 8 → **two fragments: 20+40 and 20+40 = 60 B each**

- **Link 3 transmissions:**
  - Each fragment: 60 B → 60×8 = **480 bits** → tx **480/100,000 = 0.0048 s = 4.8 ms**
  - Frag 1: starts at **18.0 ms**, finishes tx at **22.8 ms**, arrives (prop 1 ms) at **23.8 ms**
  - Frag 2: starts at **22.8 ms**, finishes tx at **27.6 ms**, arrives at **28.6 ms**

Problem 1 – other variations.

- Think about the other cases for the next session
  - Fragmentation happens at the second link, we have cut-through

- What if IP didn't support fragmentation? What would be the transmission time?

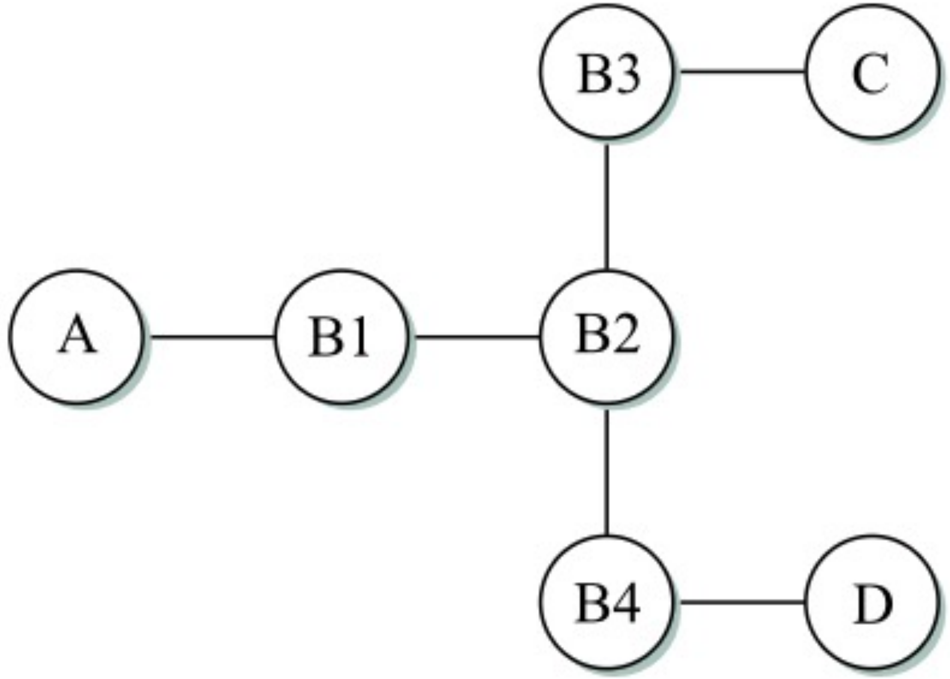
- What are the values of the fragmentation-related header fields?

Problem 2

- Learning bridges, Initially empty, sending these packets:

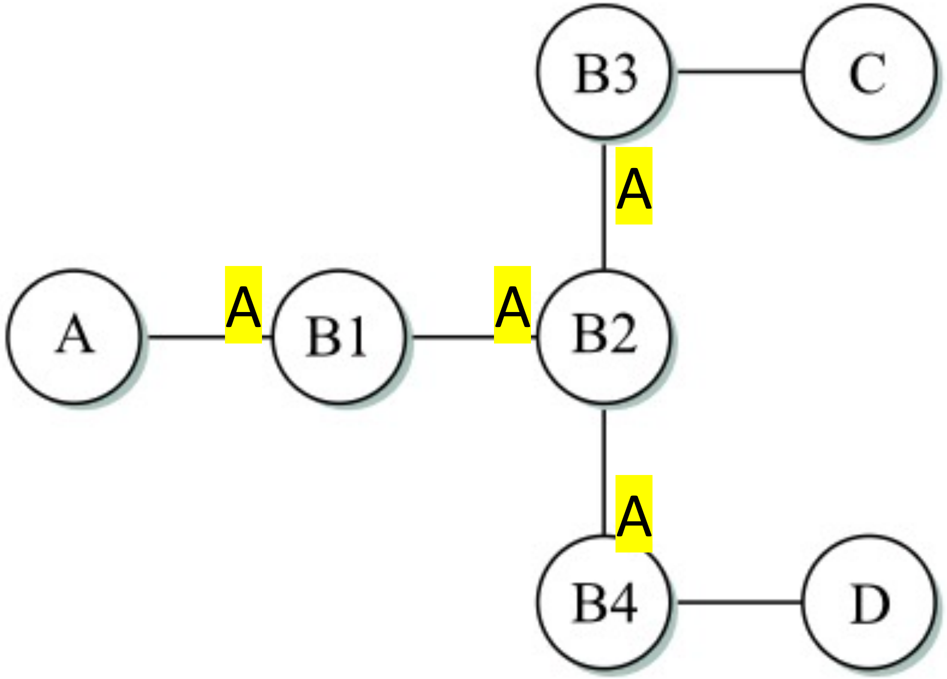
- A → C
- C → A
- D → C

What happens in the bridges?



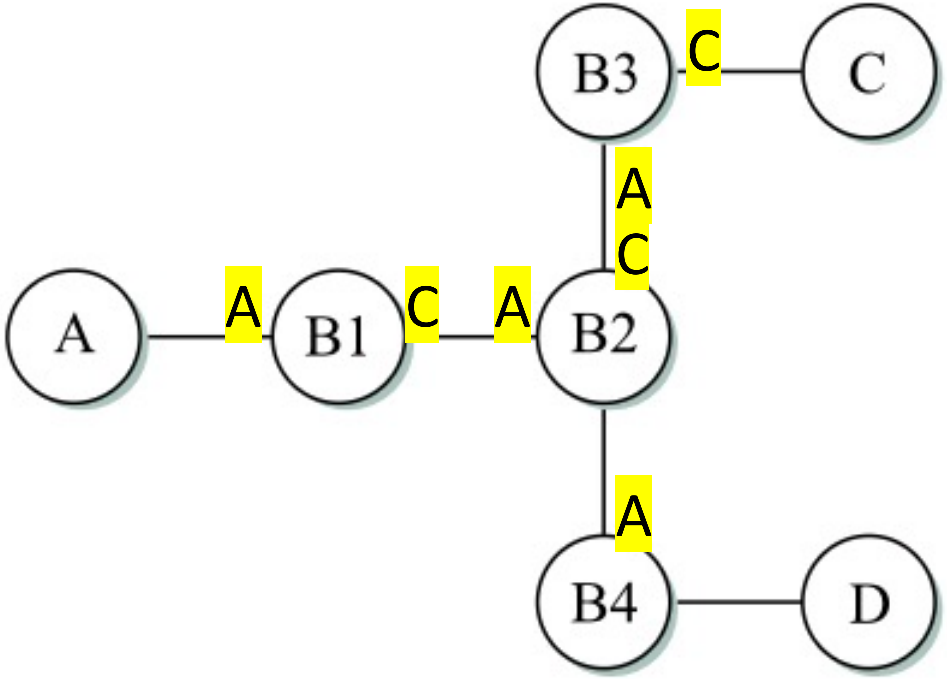
Problem 2

- A → C



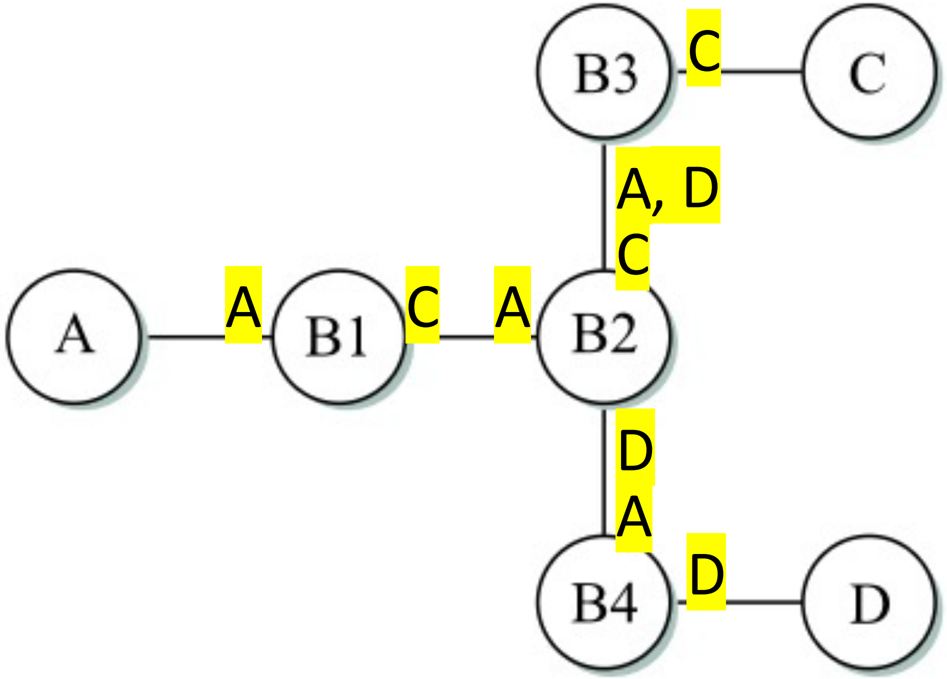
Problem 2

- A → C
- C → A



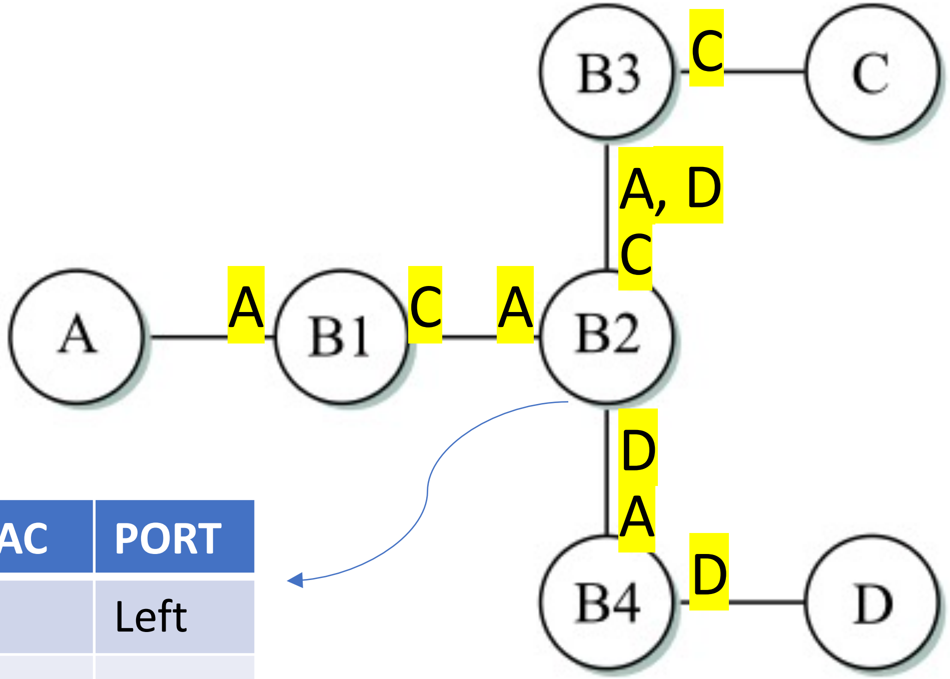
Problem 2

- A → C
- C → A
- D → C



Problem 2

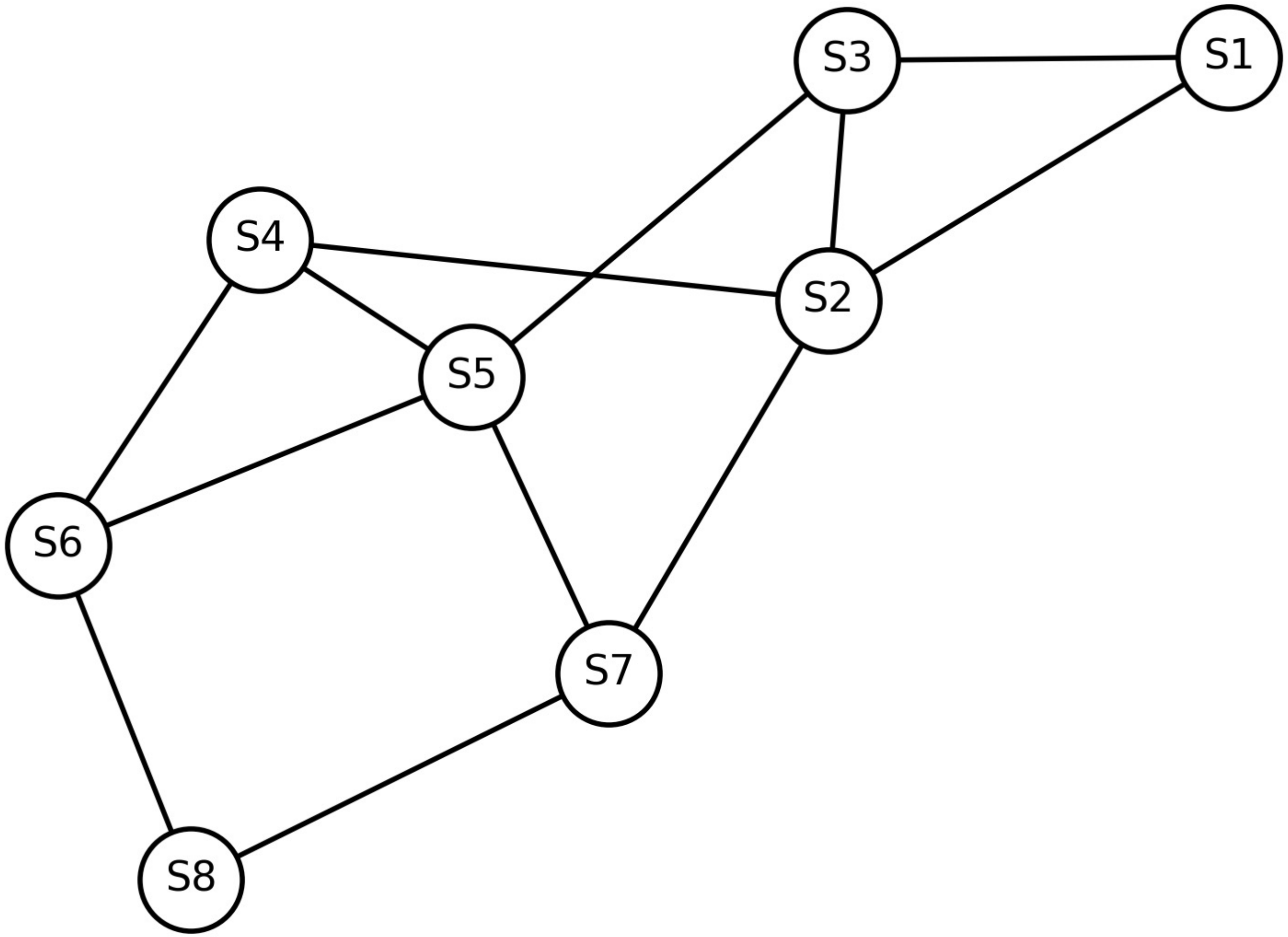
- A → C
- C → A
- D → C



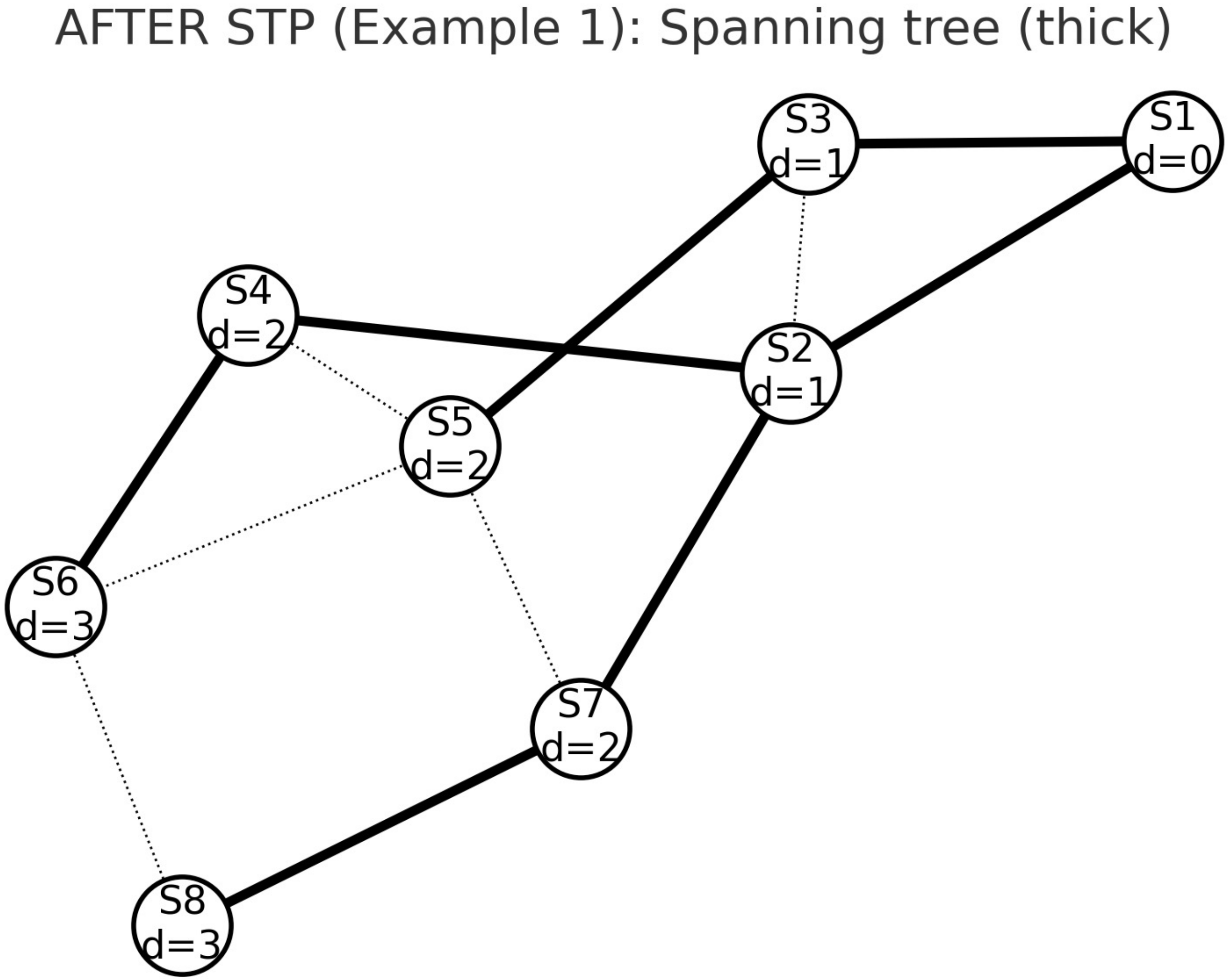
MAC	PORT
A	Left
C	Up
D	Down

Problem 2 – Spanning Tree

BEFORE STP (Example 1): All links forwarding

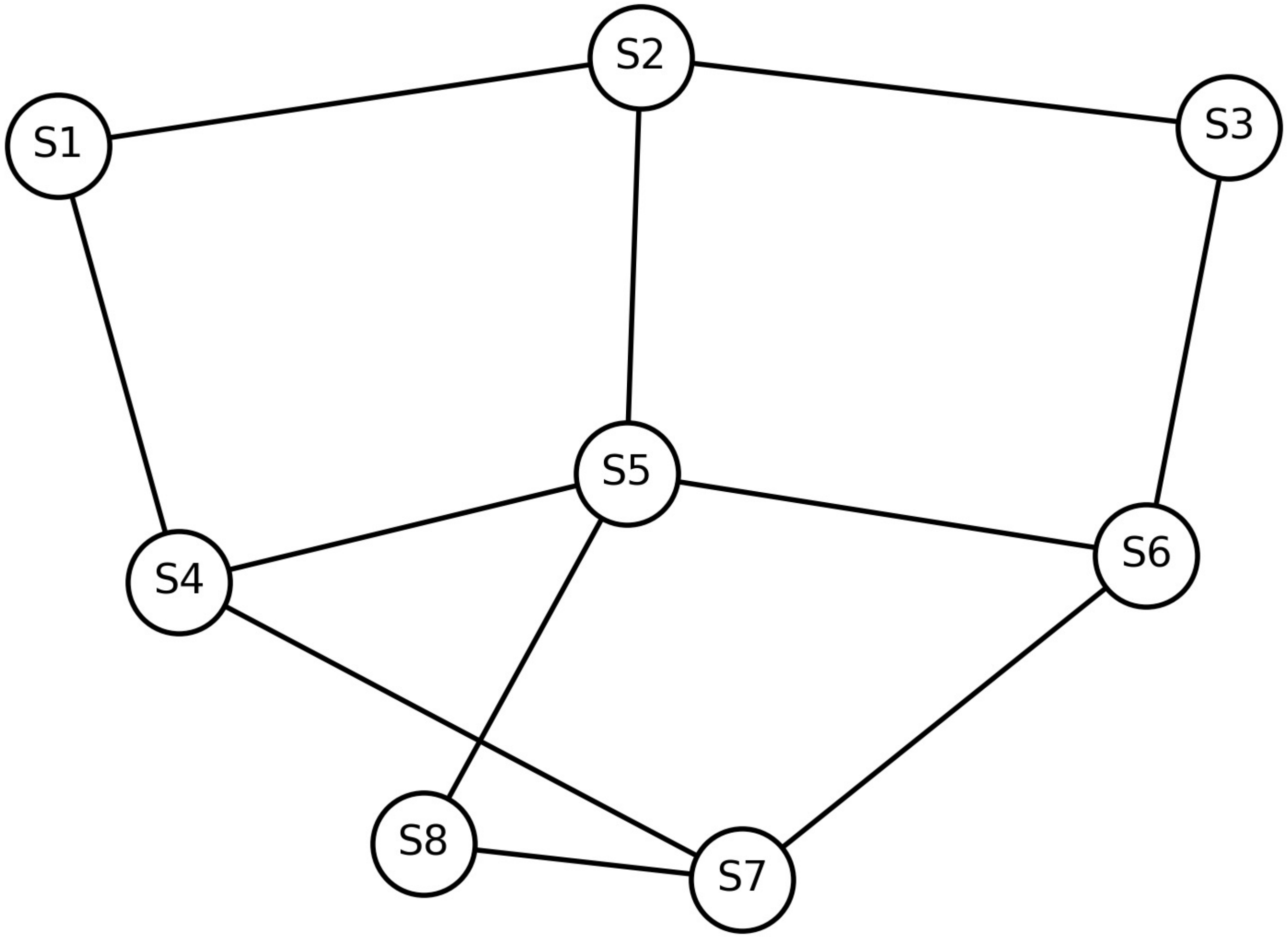


Problem 2 – Spanning Tree



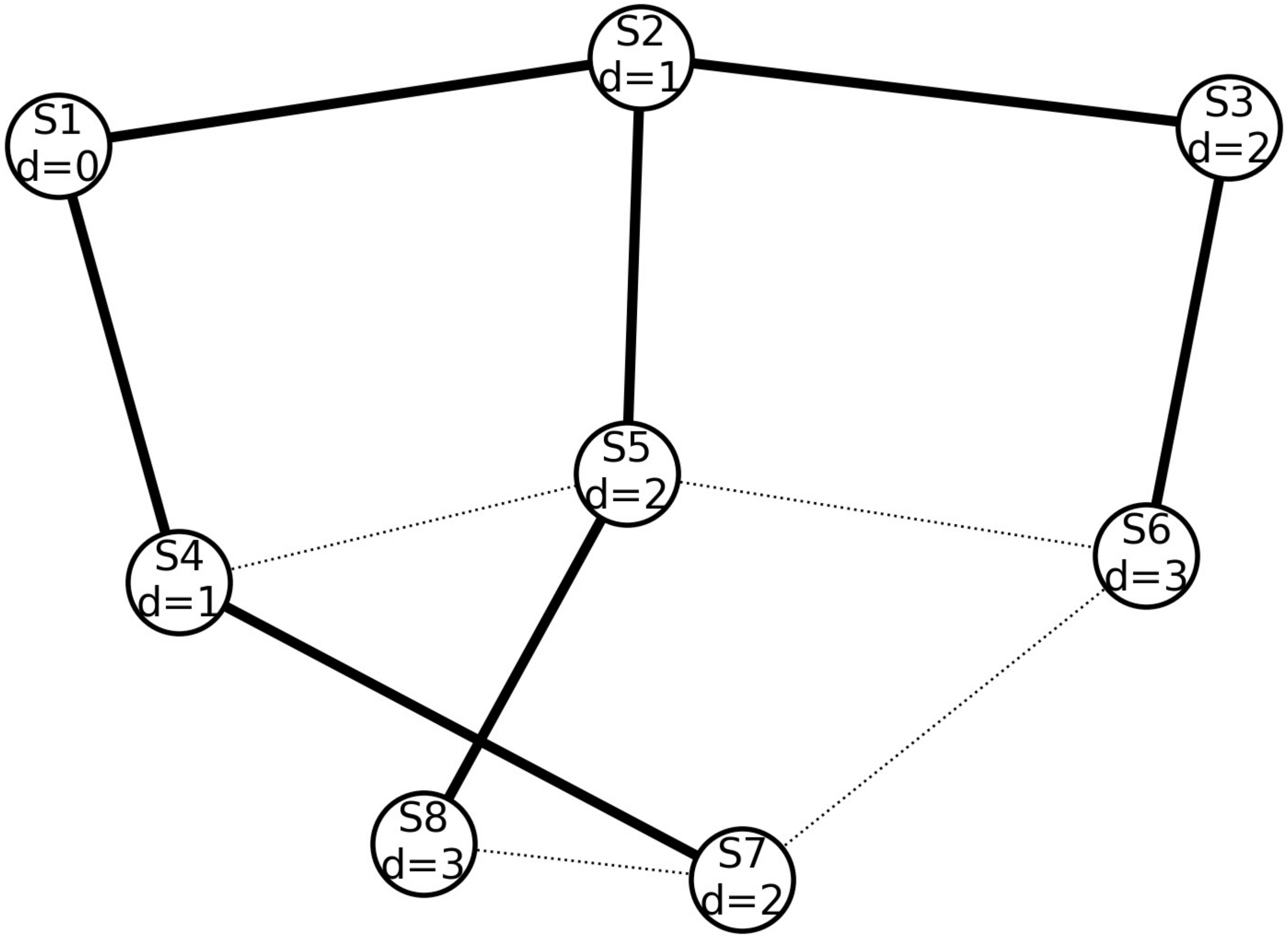
Problem 2 – Spanning Tree

BEFORE STP (Example 2): All links forwarding



Problem 2 – Spanning Tree

AFTER STP (Example 2): Spanning tree (thick)



Problem 2 – Spanning Tree

- What happens when the link costs are different?
- What happens when a new link is created or removed, or a node goes down?
- Is this a minimum spanning tree (MST)?
- What is the stretch factor for these examples? Will an MST create the lowest stretch factor?

Problem 3

- Assume we did distance vector.
- A network with 6 hosts, A to F.
- This is how the tables ended up at A and F.
- What does the network actually look like?

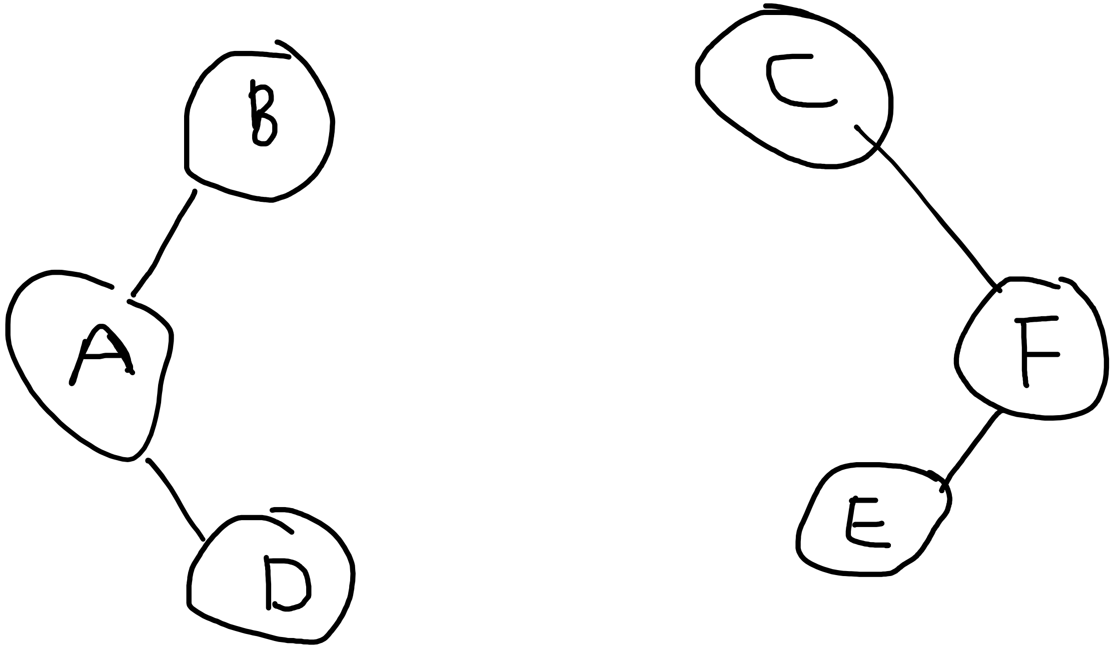
Node	Distance	Nexthop
B	1	B
C	2	B
D	1	D
E	2	B
F	3	D

Forwarding table on A

Node	Distance	Nexthop
A	3	E
B	2	C
C	1	C
D	2	E
E	1	E

Forwarding table F

Problem 3



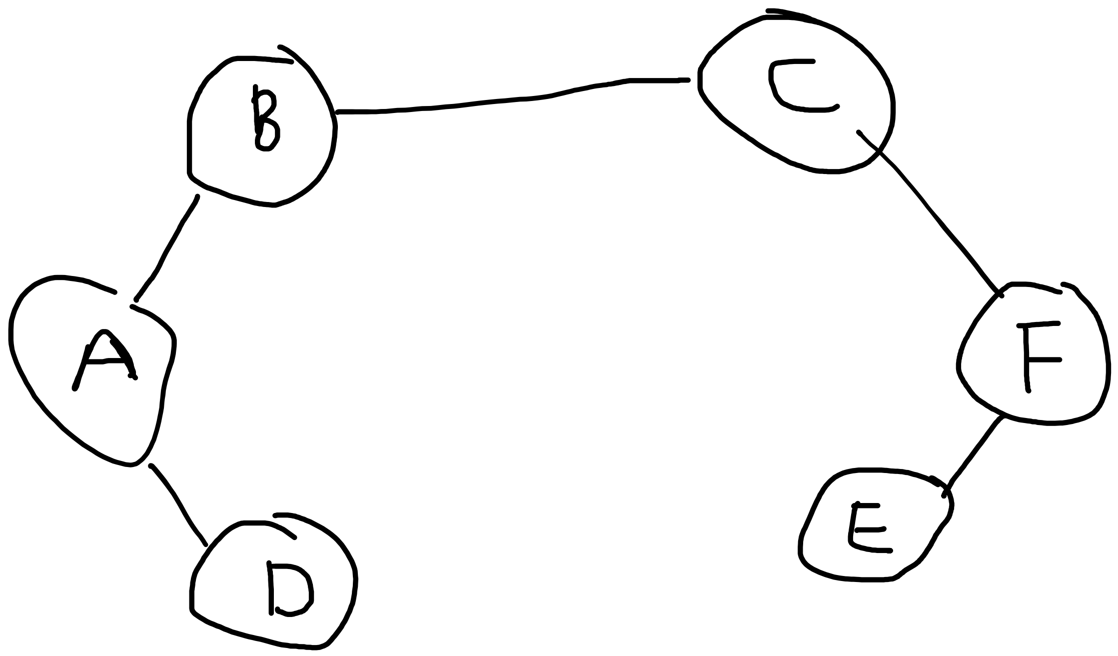
Node	Distance	Nexthop
B	1	B
C	2	B
D	1	D
E	2	B
F	3	D

Forwarding table on A

Node	Distance	Nexthop
A	3	E
B	2	C
C	1	C
D	2	E
E	1	E

Forwarding table F

Problem 3



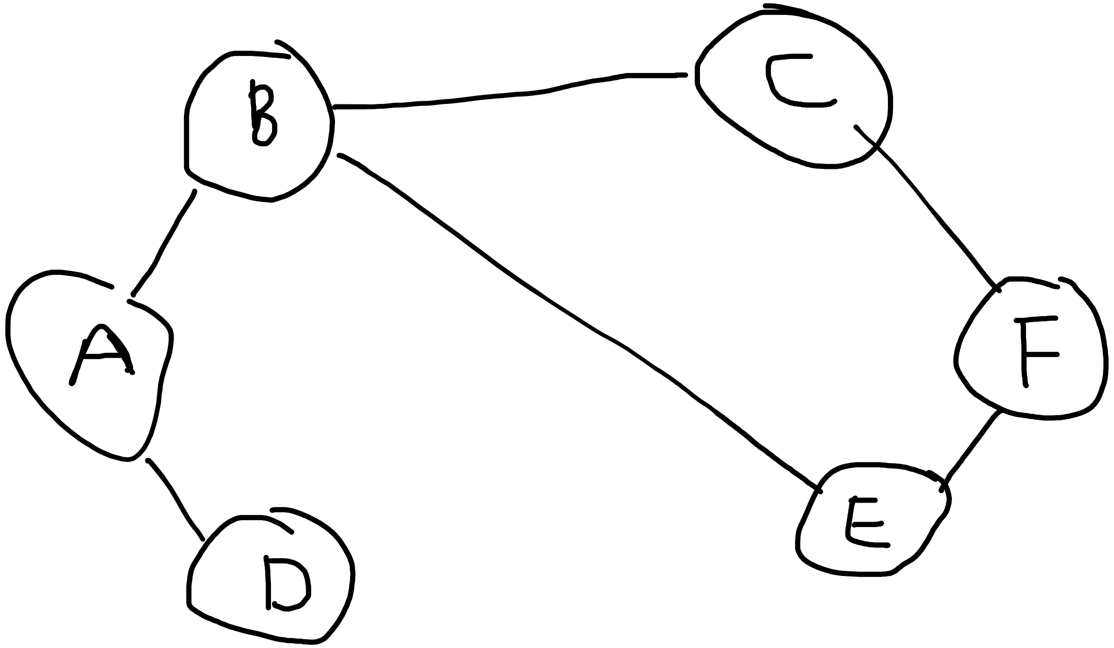
Node	Distance	Nexthop
B	1	B
C	2	B
D	1	D
E	2	B
F	3	D

Forwarding table on A

Node	Distance	Nexthop
A	3	E
B	2	C
C	1	C
D	2	E
E	1	E

Forwarding table F

Problem 3



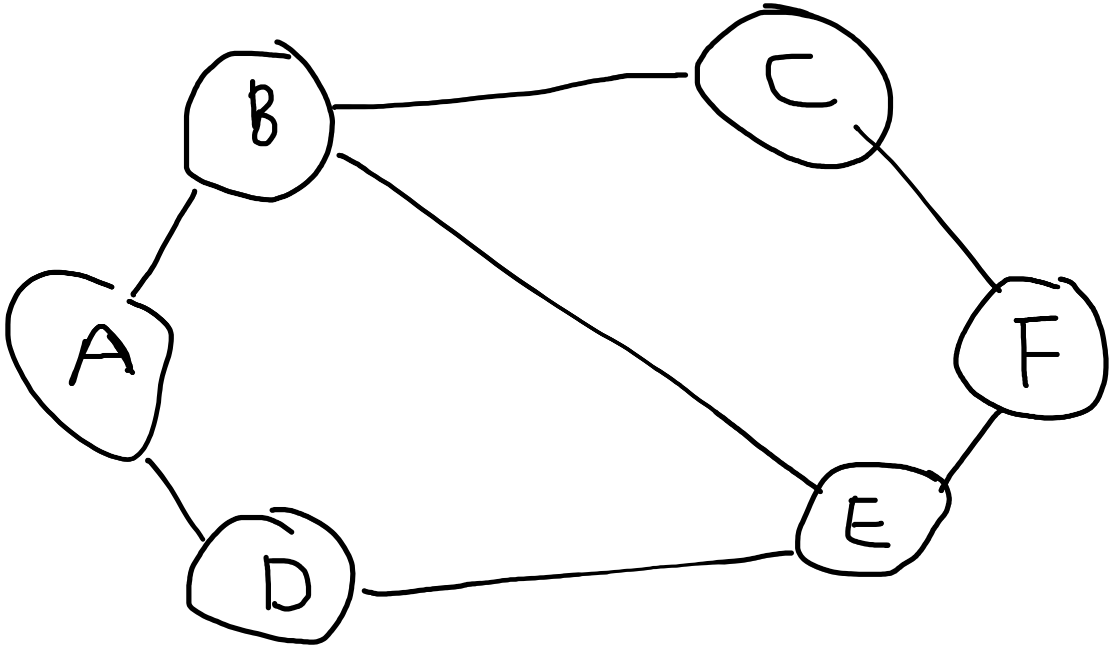
Node	Distance	Nexthop
B	1	B
C	2	B
D	1	D
E	2	B
F	3	D

Forwarding table on A

Node	Distance	Nexthop
A	3	E
B	2	C
C	1	C
D	2	E
E	1	E

Forwarding table F

Problem 3



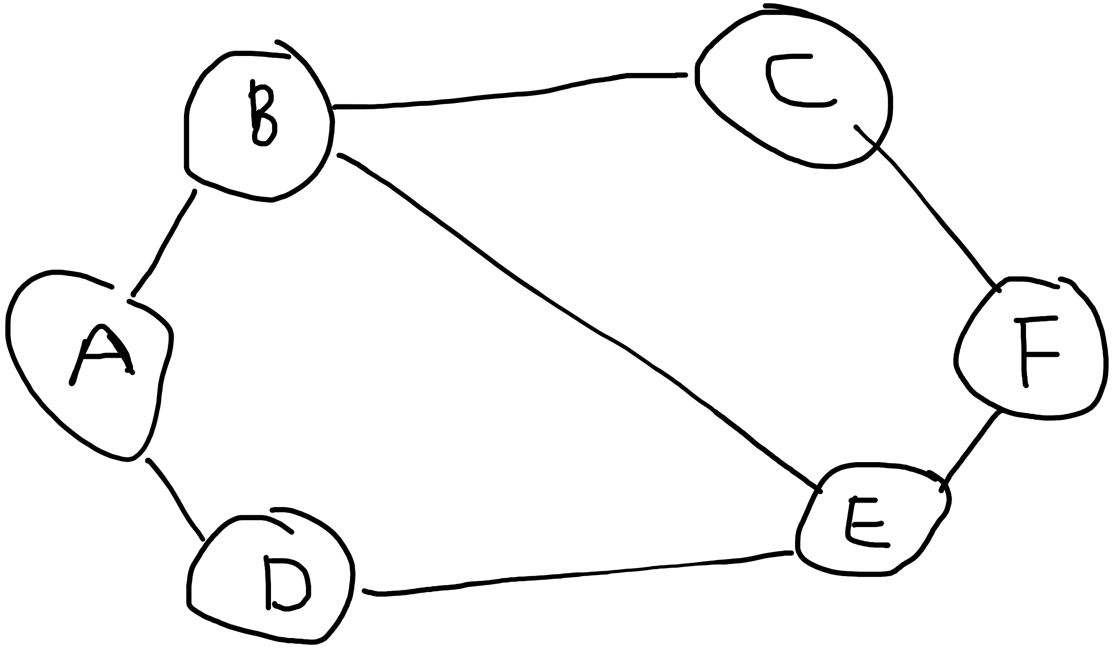
Node	Distance	Nexthop
B	1	B
C	2	B
D	1	D
E	2	B
F	3	D

Forwarding table on A

Node	Distance	Nexthop
A	3	E
B	2	C
C	1	C
D	2	E
E	1	E

Forwarding table F

Problem 3



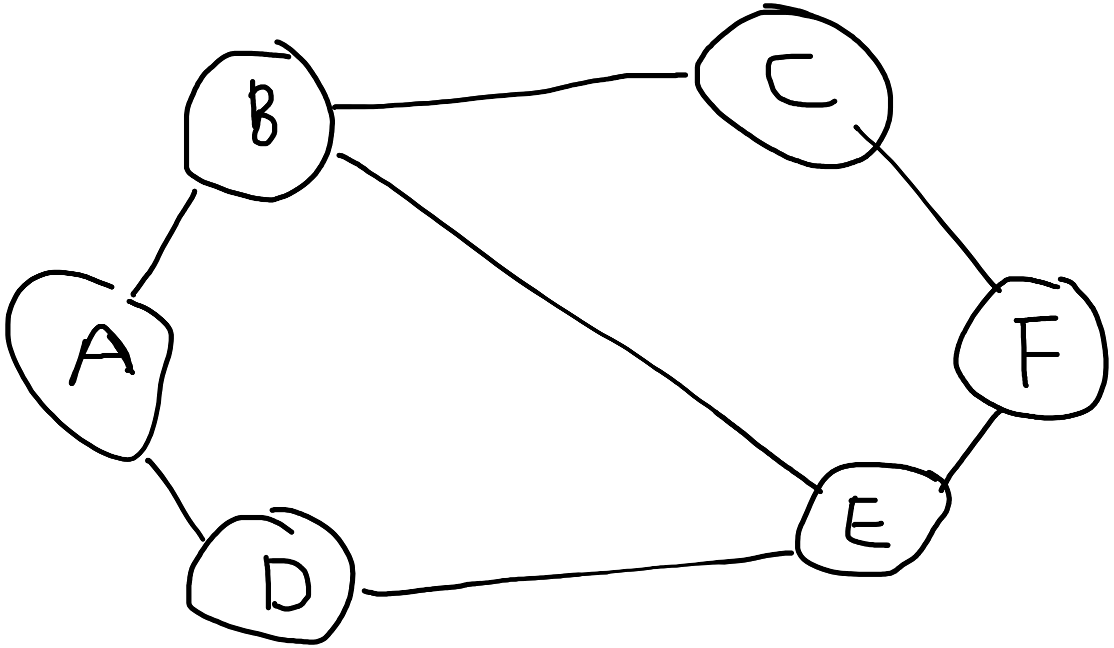
Node	Distance	Nexthop
B	1	B
C	2	B
D	1	D
E	2	B
F	3	D

Forwarding table on A

Node	Distance	Nexthop
A	3	E
B	2	C
C	1	C
D	2	E
E	1	E

Forwarding table F

Problem 3



Node	Distance	Nexthop
B	1	B
C	2	B
D	1	D
E	2	B
F	3	D

Forwarding table on A

Node	Distance	Nexthop
A	3	E
B	2	C
C	1	C
D	2	E
E	1	E

Forwarding table F

Problem 4

Where these packets will be routed based on Longest Prefix Matching?

- a) 10.1.129.70 → ___
- b) 10.1.129.10 → ___
- c) 10.1.130.5 → ___
- d) 10.2.3.4 → ___
- e) 11.0.0.1 → ___
- f) 10.1.0.1 → ___
- g) 10.1.128.200 → ___
- h) 10.1.255.255 → ___

Prefix	Next Hop
10.0.0.0/8	P
10.1.0.0/16	Q
10.1.128.0/17	R
10.1.128.0/24	S
10.1.129.64/26	T
* (Default)	U

Problem 4

Prefix	Next Hop
10.0.0.0/8	P
10.1.0.0/16	Q
10.1.128.0/17	R
10.1.128.0/24	S
10.1.129.64/26	T
* (Default)	U

Where these packets will be routed based on Longest Prefix Matching?

- a) 10.1.129.70 → **T** (matches 10.1.129.64/26; longest over /17, /16, /8)
- b) 10.1.129.10 → **R** (in /17, not /24)
- c) 10.1.130.5 → **R** (in 10.1.128.0 – 10.1.255.255 → /17)
- d) 10.2.3.4 → **P** (in 10.0.0.0/8)
- e) 11.0.0.1 → **U** (no 11.0.0.0/… entries; default)
- f) 10.1.0.1 → **Q** (in /16; not in /17)
- g) 10.1.128.200 → **S** (in /24; /24 outranks /17)
- h) 10.1.255.255 → **R** (in /17; not in /24 or /26)