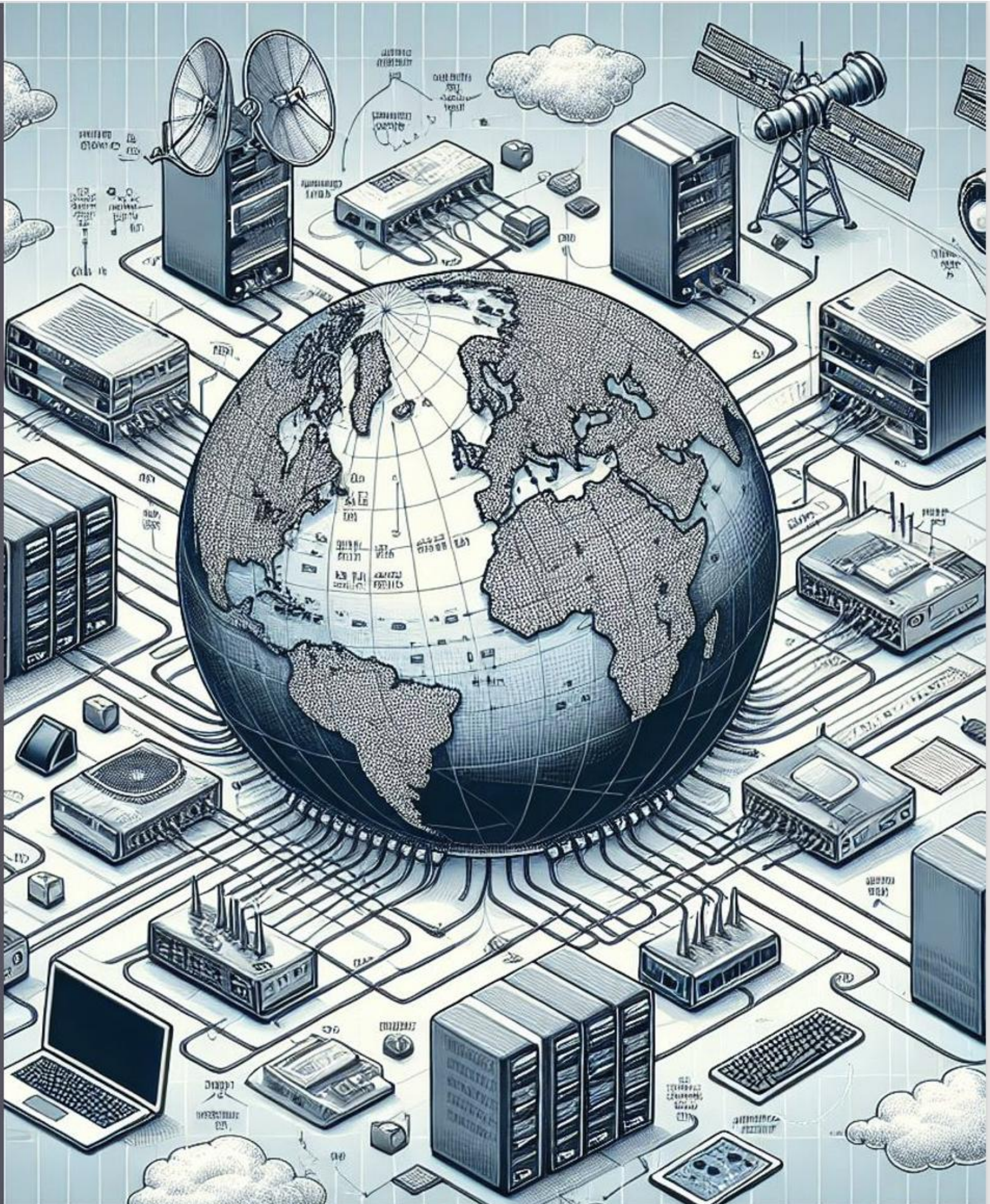
COLLEGE OF ARTS AND SCIENCES
The University of Alabama at Birmingham
Department of Computer Science

CS 334/534 NETWORKING

Dr. Ragib Hasan

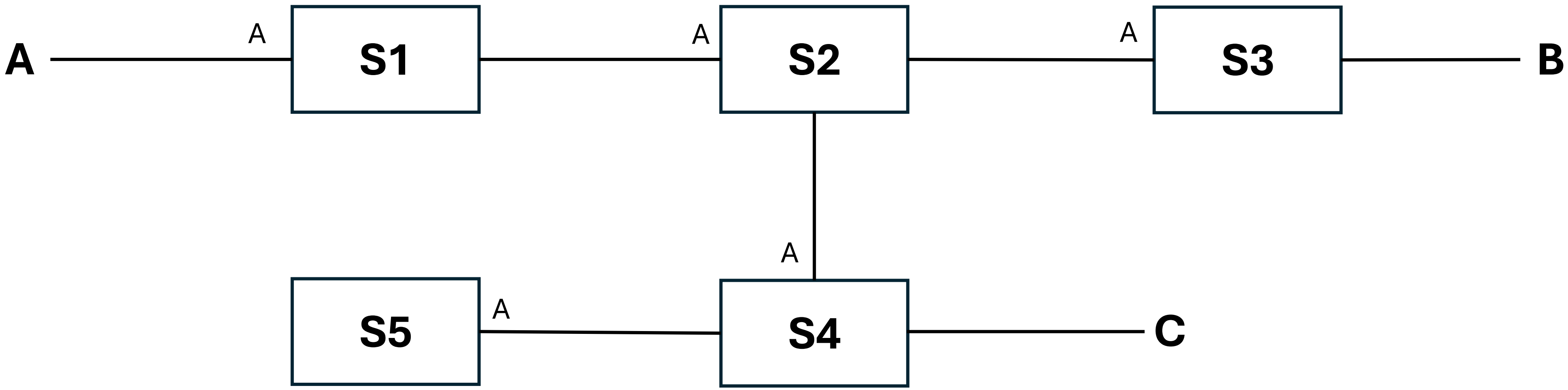
Lecture 3.1:
Spanning Tree Protocol



Lecture goals

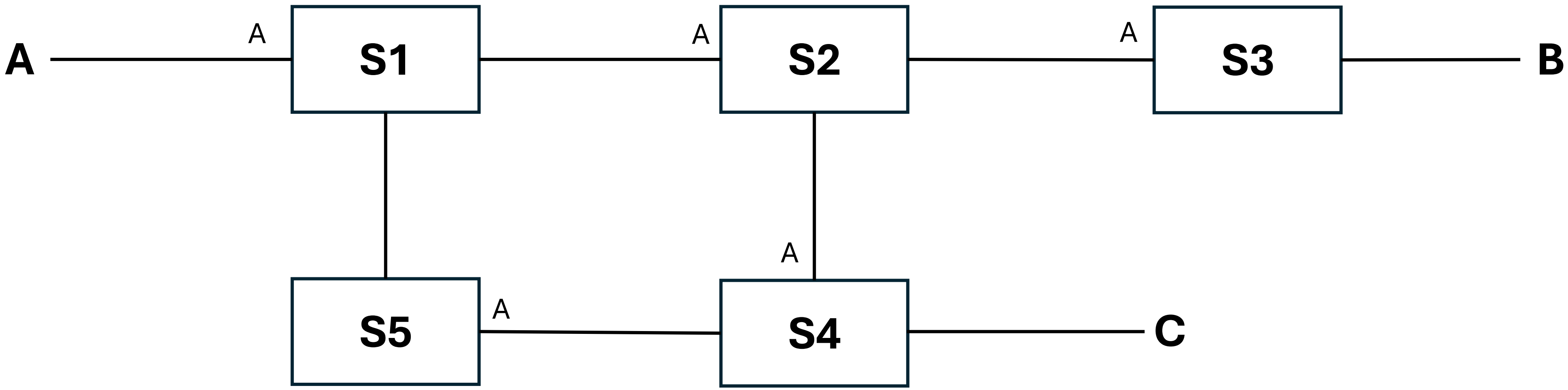
- Learning about Spanning Tree Protocol

- Book reference: Chapter 3, section 3.1



First packet: A sends to B; all switches learn where A is

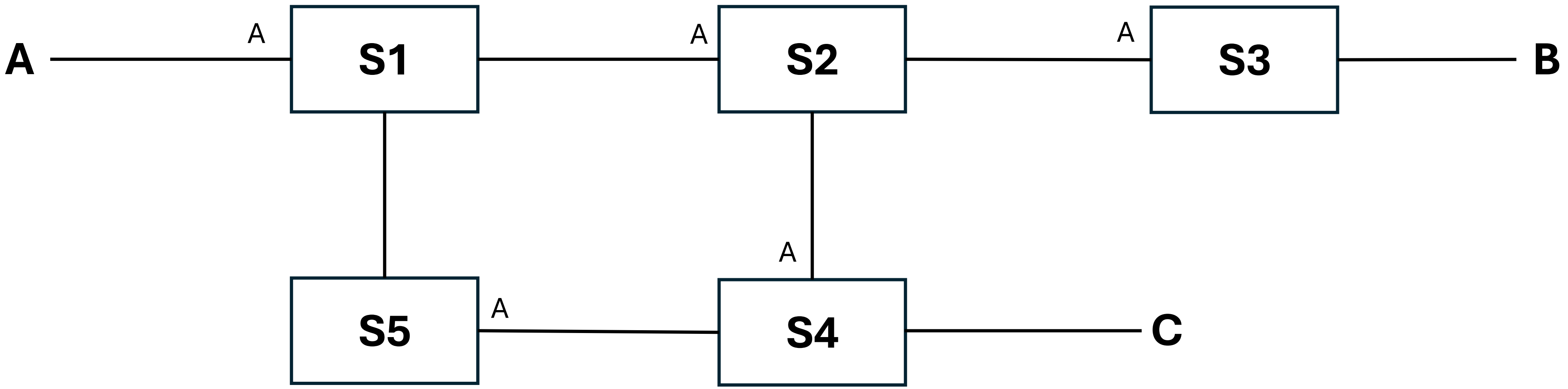
Multiple Switch Learning Process (Recap)



First packet: A sends to B; all switches learn where A is

What if?

- S5 is also connected with S1?



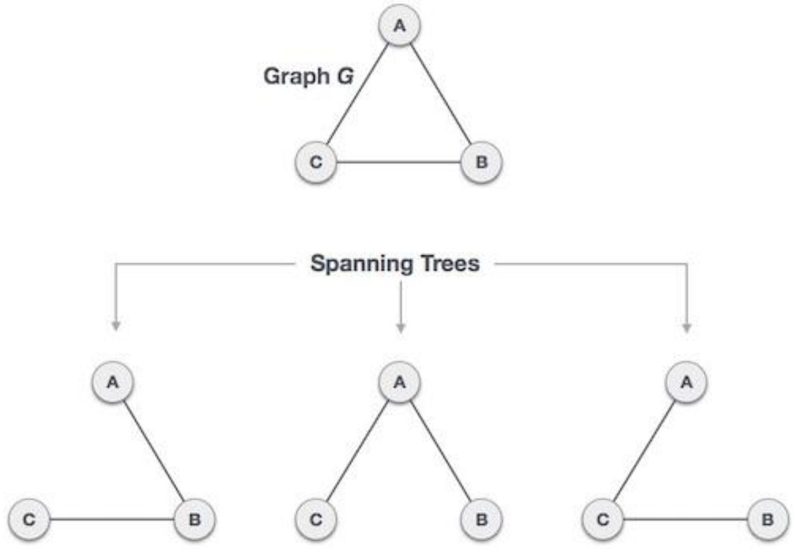
First packet: A sends to B; all switches learn where A is

What if?

- S5 is also connected with S1?
- It will form a loop.
- Loop provides redundancy, ensuring connectivity if a link fails
- But, the broadcast packet will circulate endlessly, Consuming bandwidth and leading to network failure

Spanning Tree

- The Spanning Tree Protocol (STP), is a network protocol used to **prevent loops** in Ethernet networks while ensuring redundancy and connectivity.
- It makes a **loop-free subset** of the network.
- Links **not part** of the spanning tree are **disabled**, even if they offer more efficient paths.
- If a link in the spanning tree fails, the network dynamically **recomputes the tree**, potentially **reactivating previously disabled links.**



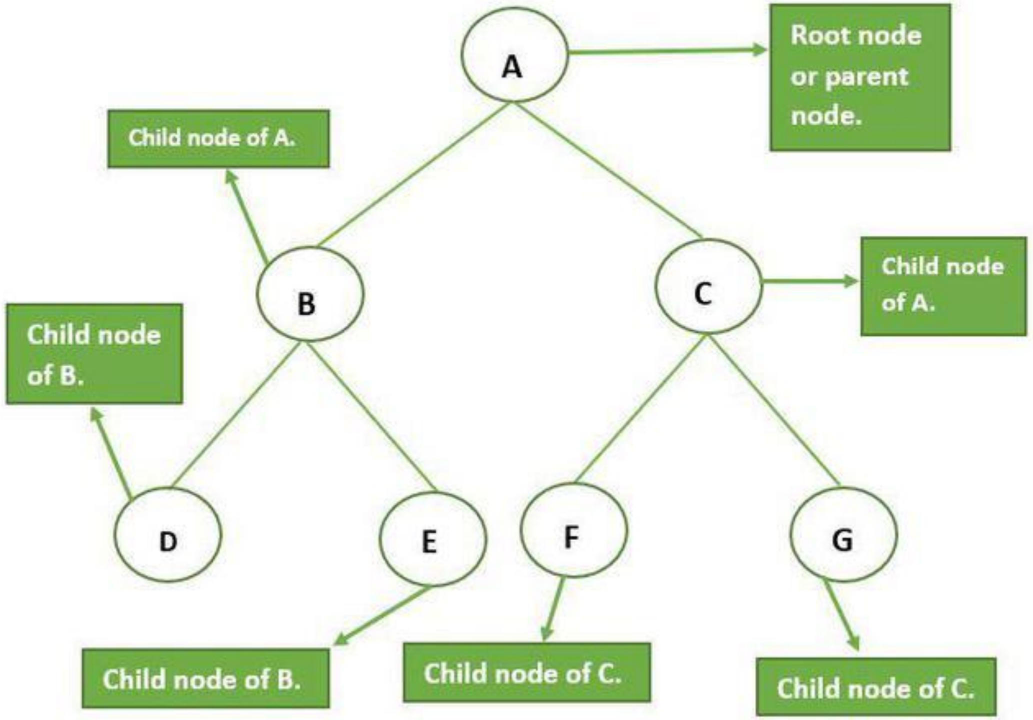
Spanning Tree

- Each switch has a unique ID (e.g., smallest Ethernet MAC address).
- Each edge (device) attaches to a switch via a numbered interface.
- Algorithm: Developed by Radia Perlman ([RP85]).
- **Outage Handling**: Spanning tree is recomputed if an outage occurs.
- **Partition Handling**: Both network pieces build spanning trees if partitioned.

Spanning Tree

Algorithm:

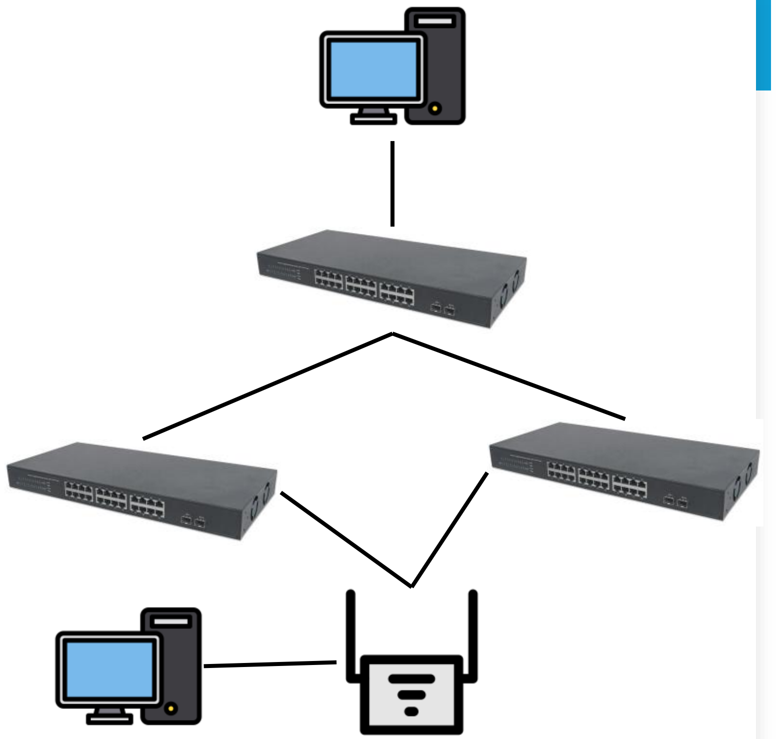
- Root Node: Switch with the smallest ID.
- Path Selection:
  - Shorter path to the root is preferred.
  - In case of ties, the switch with the smaller ID is used.
  - Path cost is typically the number of hops or inversely proportional to bandwidth.



Spanning Tree

Bridge Protocol Data Units (BPDUs):

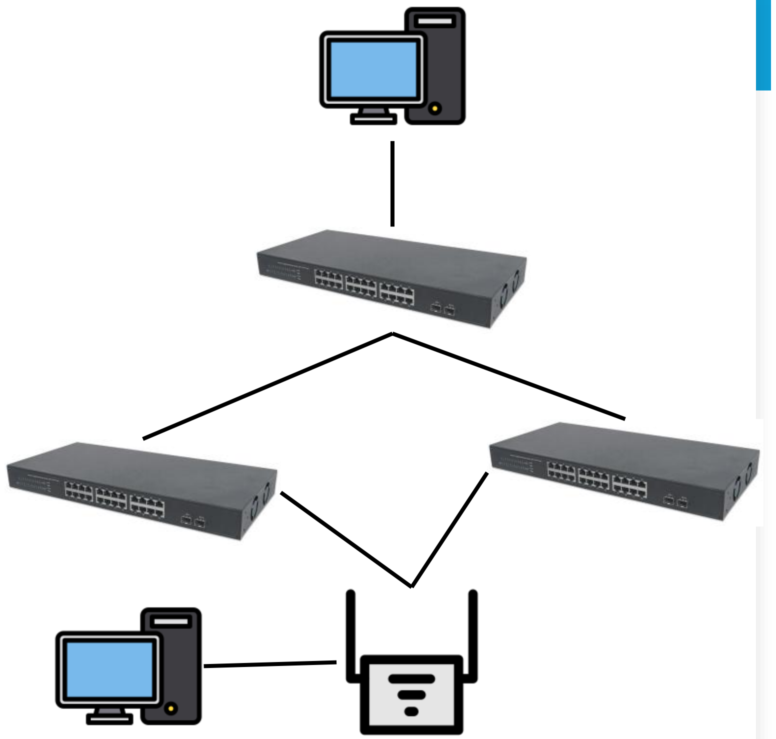
- All switches send out regular messages on all interfaces called BPDU (“Hello” message)
- Multicast Address: 01:80:c2:00:00:00. (which is processed only by **switches** and **not forwarded to end devices**.)
- BPDU Contents:
  - Switch ID
  - Root node ID (Probable)
  - Path cost to the root



Spanning Tree

How Switch process BPDU?

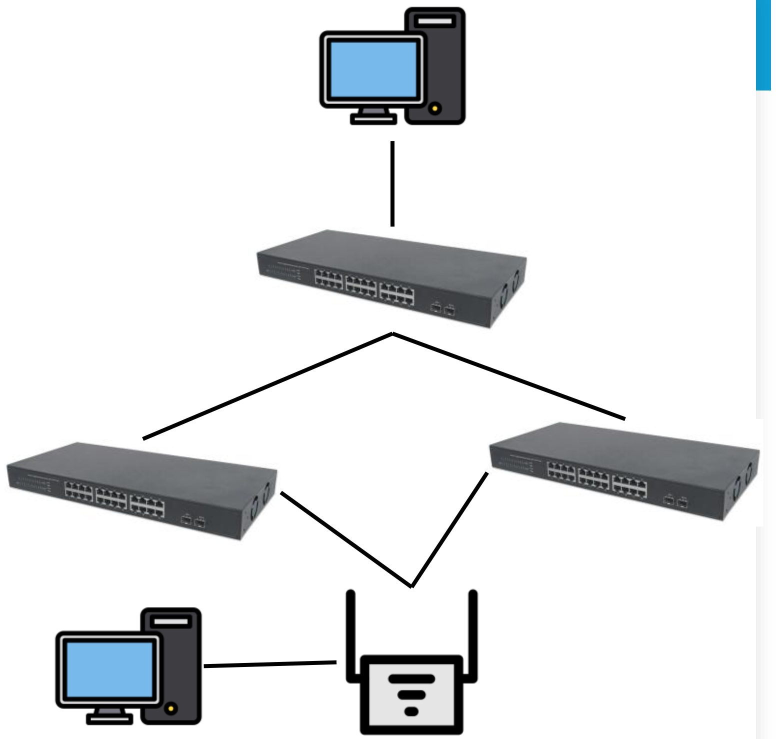
- New root candidate with a lower ID.
- Shorter path to the existing root.
- Equal-length path via a neighbor switch with a lower ID (tie-breaker rule).
- If there are two ports that connect to that switch, the port number is used as an additional tie-breaker.



Spanning Tree

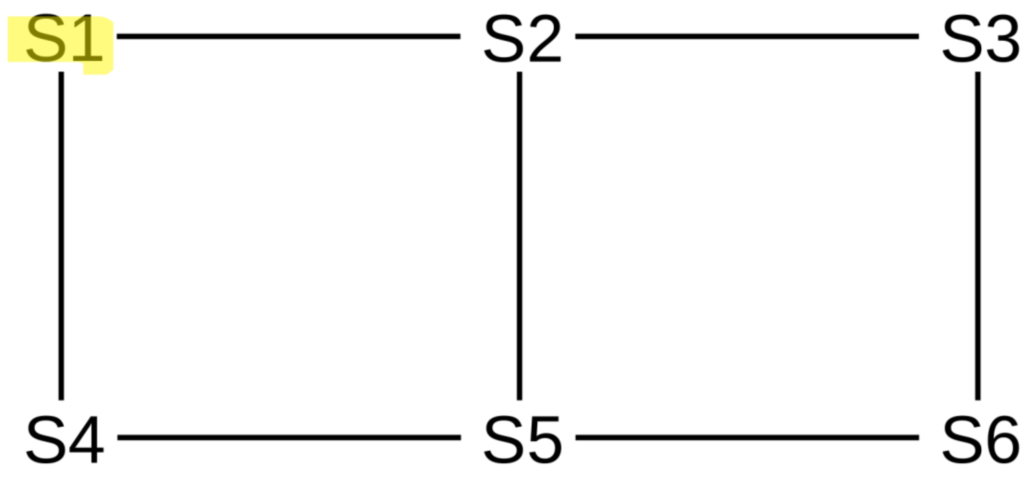
Interface Pruning Rules

1. Enable the port that connects to the root switch.
2. Enable ports that other switches use to connect to the root.
3. Enable ports that connect to segments with other switches if:
    1. This switch has the lowest cost to the root.
    2. If tied, this switch has the smallest ID.
    3. If still tied, the port with the smallest ID is kept.
4. Enable ports that connect directly to a host or segment (no other switches).



Example: Switch only

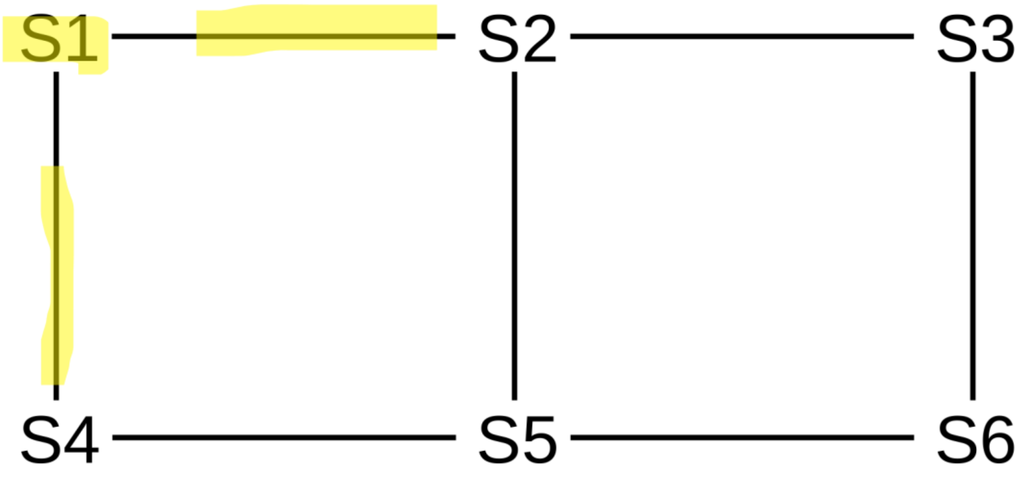
- **S1** has the lowest ID, and so becomes the root.



* Considering switch numbers represent their IDs

Example: Switch only

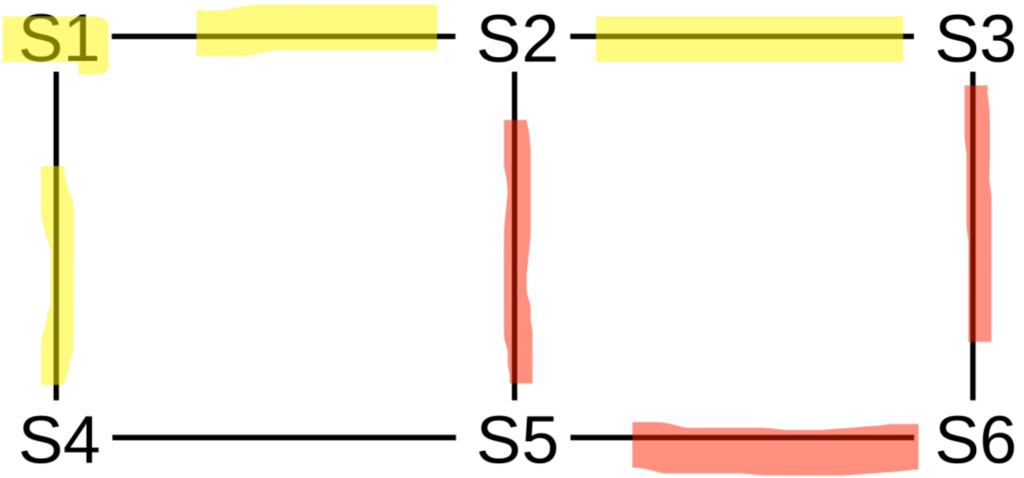
- **S1** has the lowest ID, and so becomes the root.
- **S2** and **S4** are directly connected, so they will enable the interfaces by which they reach **S1** (Rule 1).
- **S1** will enable its interfaces by which **S2** and **S4** reach it (Rule 2).



* Considering switch numbers represent their IDs

Example: Switch only

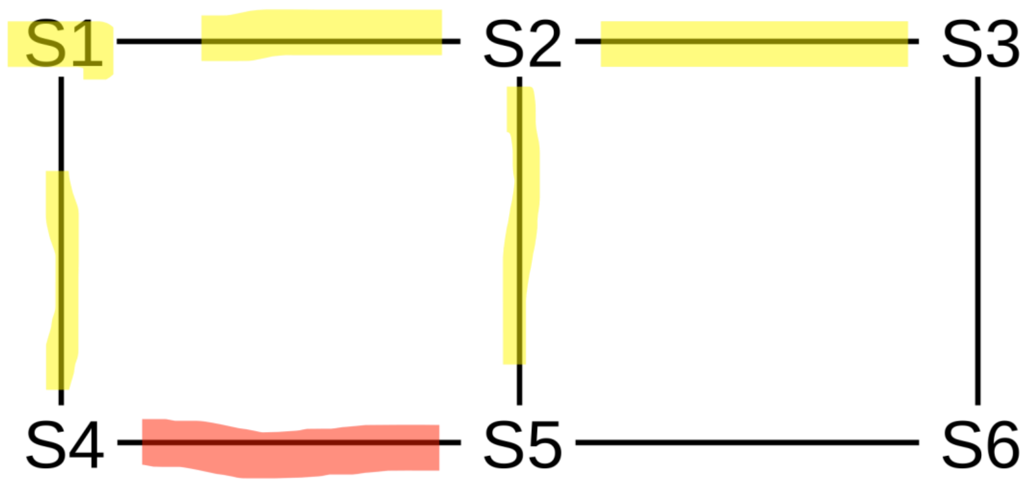
- **S1** has the lowest ID, and so becomes the root.
- **S2** and **S4** are directly connected, so they will enable the interfaces by which they reach **S1** (Rule 1).
- **S1** will enable its interfaces by which **S2** and **S4** reach it (Rule 2).
- **S3** has a unique lowest-cost route to **S1**, and so again by Rule 1 it will enable its interface to **S2**.
- **S2** will enable its interface to **S3** (Rule 2)



* Considering switch numbers represent their IDs

Example: Switch only

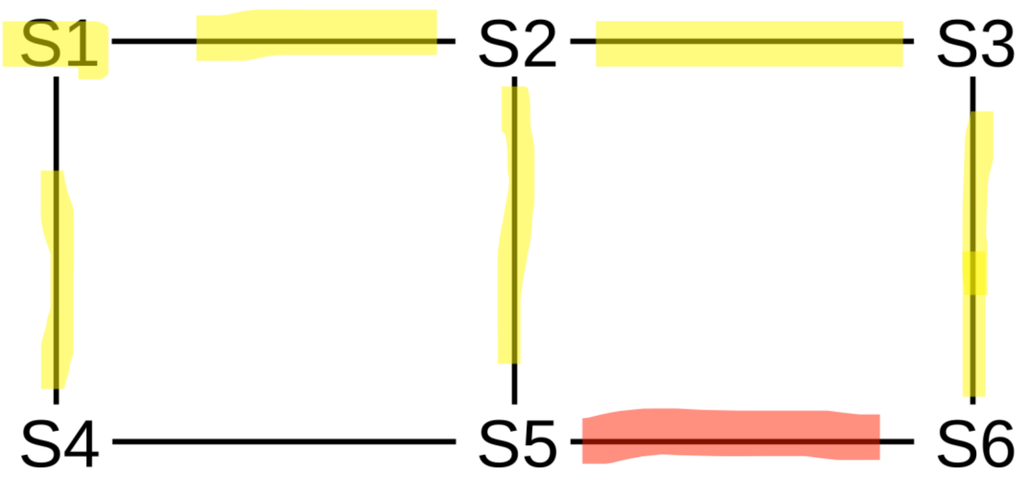
- **S5** has two choices;
  - It hears of equal-cost paths to the root from both **S2** and **S4**.
  - It picks the lower-numbered neighbor **S2**;
- The interface to **S4** will never be enabled. Similarly, **S4** will never enable its interface to **S5**.



* Considering switch numbers represent their IDs

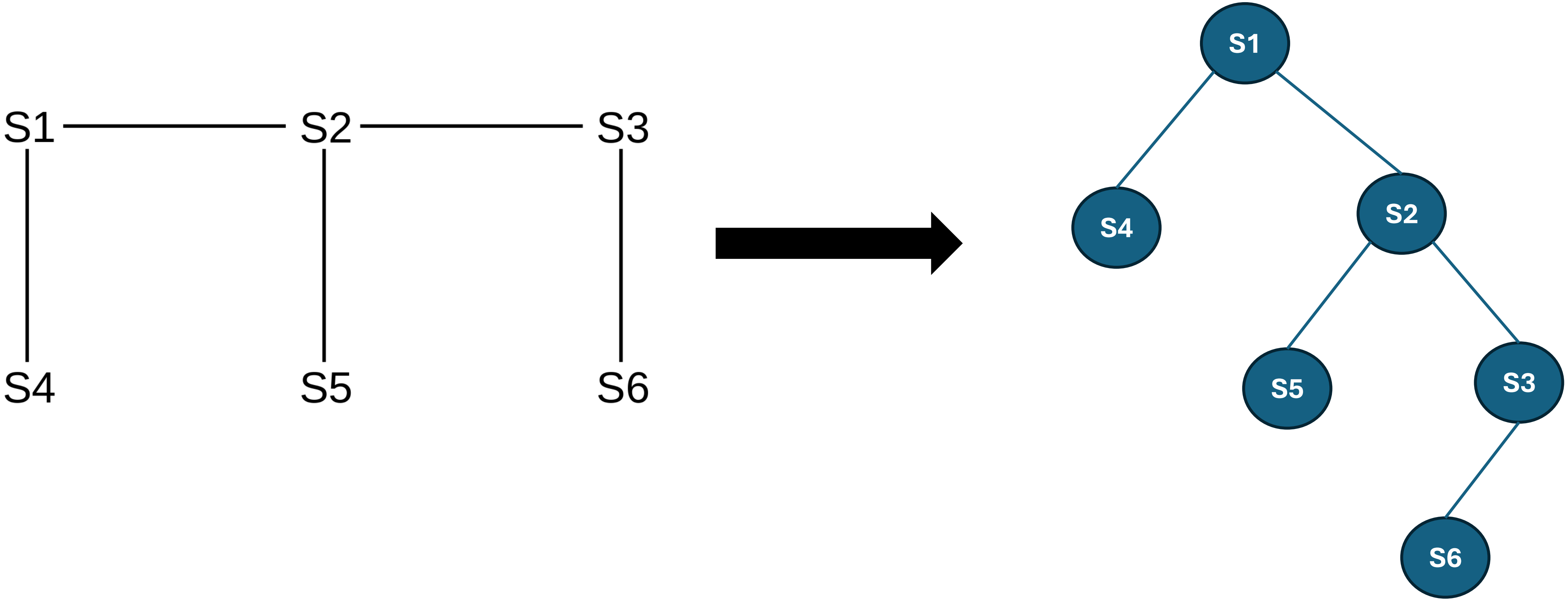
Example: Switch only

- **S5** has two choices;
    - It hears of equal-cost paths to the root from both **S2** and **S4**.
    - It picks the lower-numbered neighbor **S2**;
- The interface to **S4** will never be enabled. Similarly, **S4** will never enable its interface to **S5**.
- Similarly, **S6** has two choices; it selects **S3**.



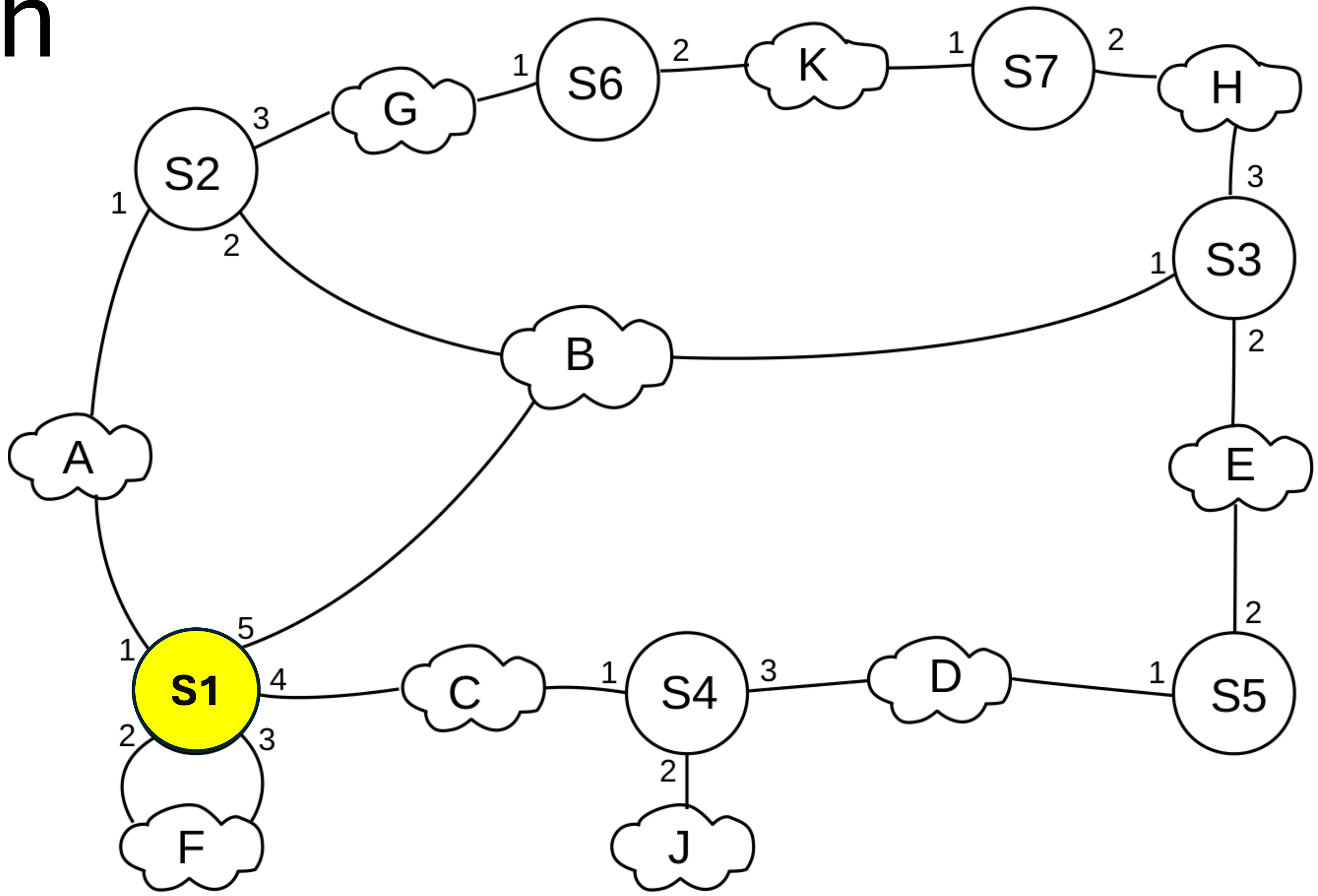
* Considering switch numbers represent their IDs

Example: Switch only



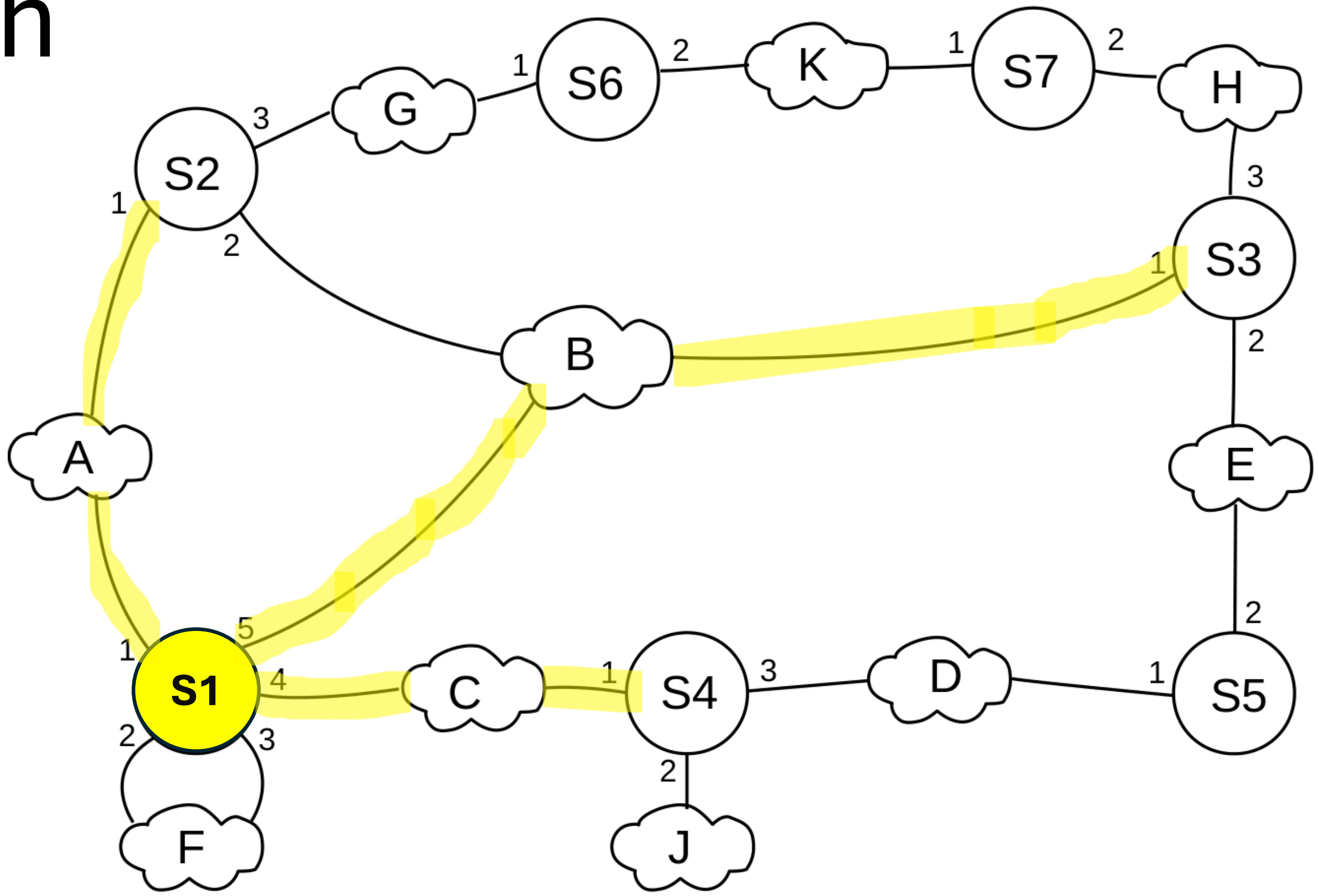
Example: Switch with Segments

- S1 is the root (because 1 is the smallest)
- S2, S3 and S4 are one (unique) hop away; S5, S6 and S7 are two hops away.



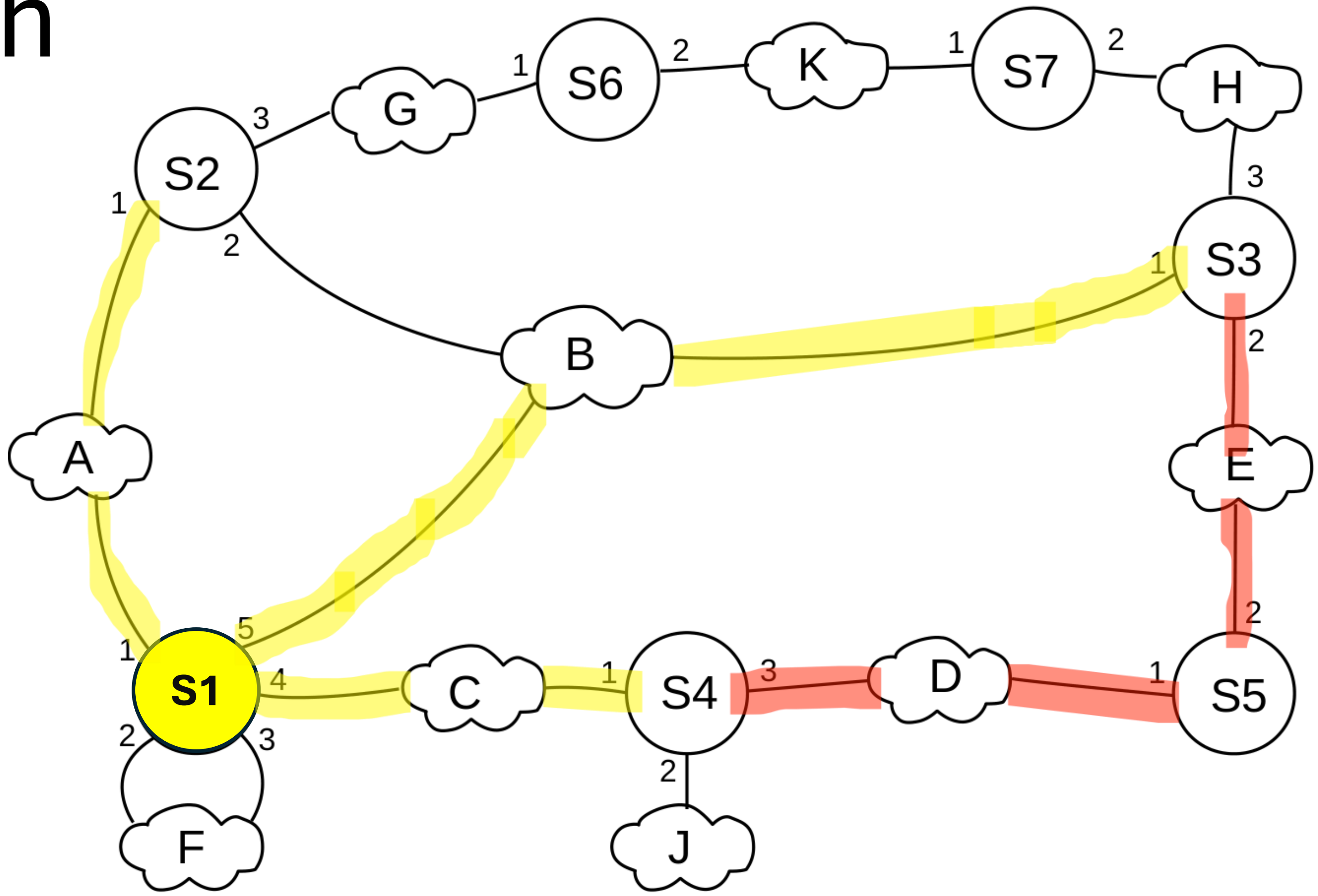
Example: Switch with Segments

- S1 is the root (because 1 is the smallest)
- S2, S3 and S4 are one (unique) hop away; S5, S6 and S7 are two hops away.
- Rule 1 enables S2's port 1 (Why not 2?), S3's port 1, and S4's port 1.
- Rule 2 enables the corresponding ports on S1: ports 1, 5 and 4 respectively



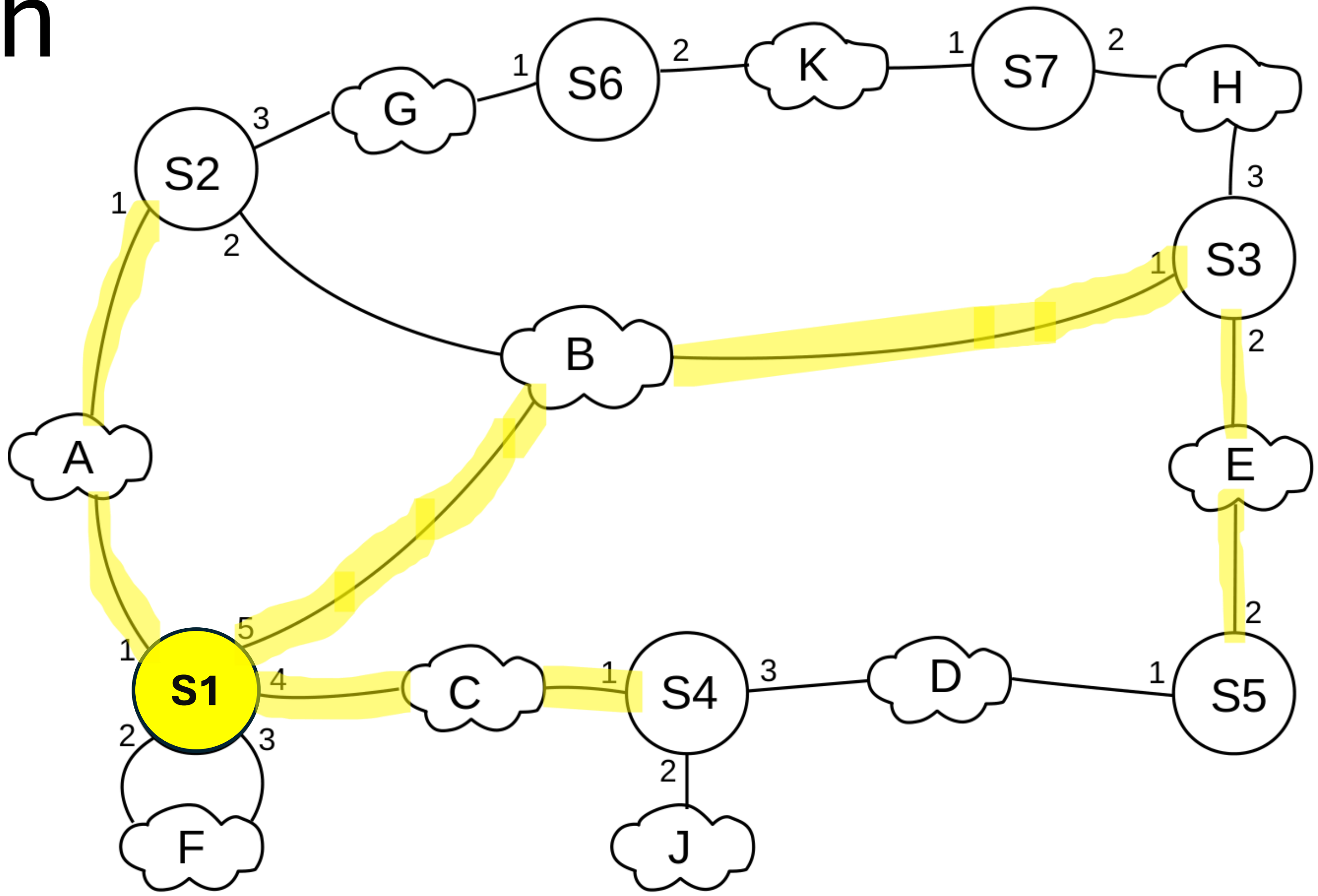
Example: Switch with Segments

- S5 has two equal-cost paths to the root: S5=>S4=>S1 and S5=>S3=>S1.

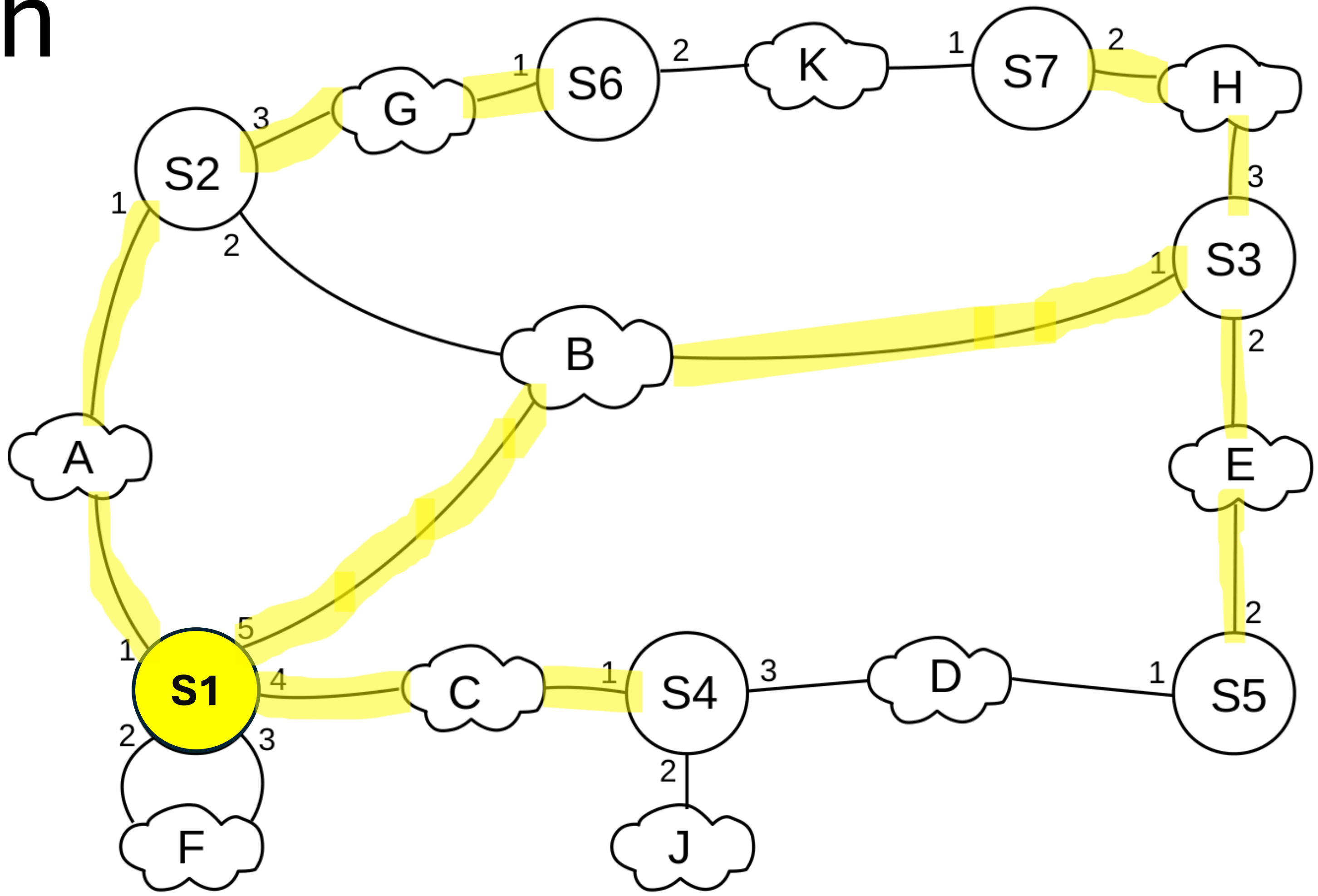


Example: Switch with Segments

- S5 has two equal-cost paths to the root: S5=>S4=>S1 and S5=>S3=>S1.
- S3 is the switch with the lower ID; its port 2 is enabled and S5 port 2 is enabled.



Example: Switch with Segments

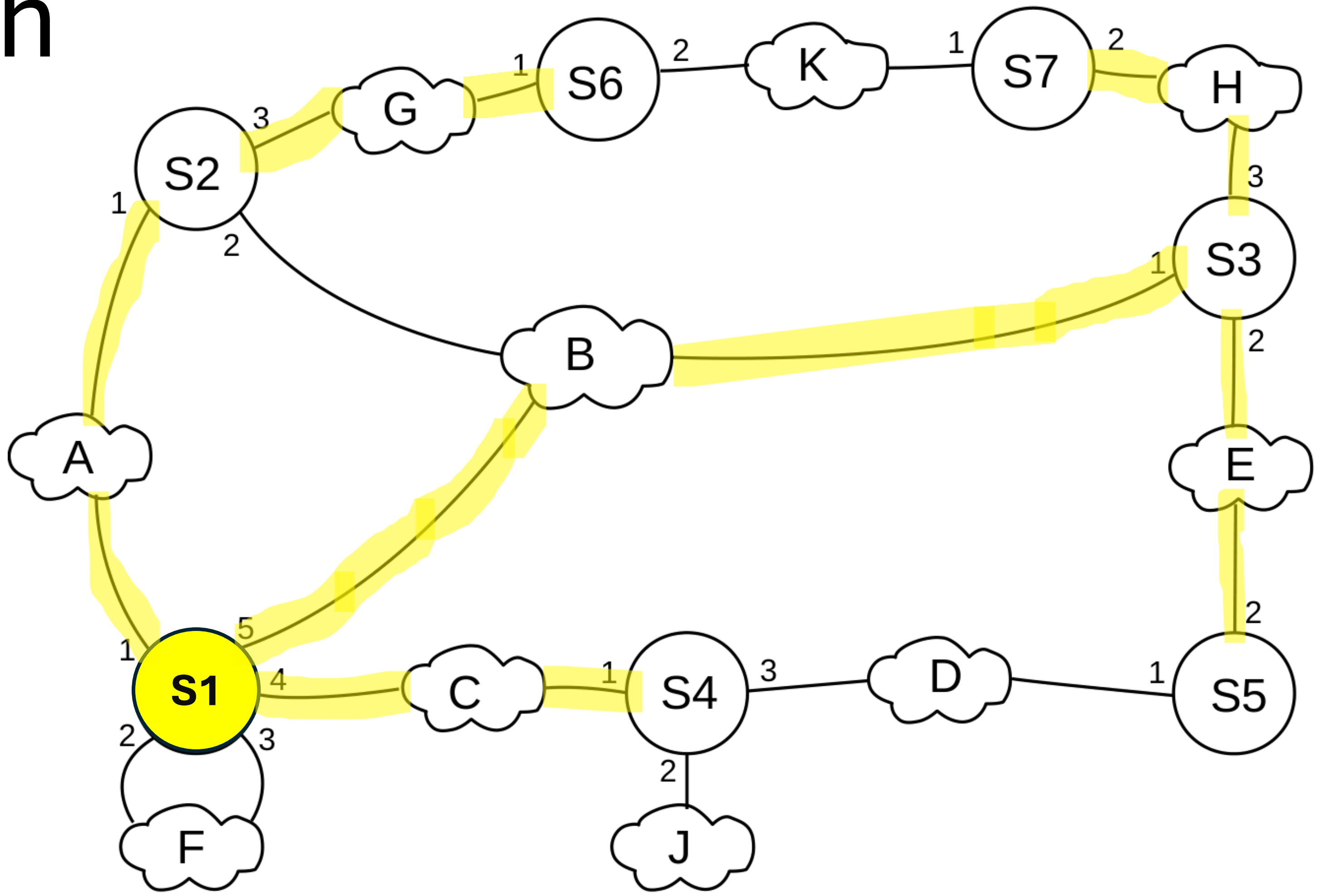


- S5 has two equal-cost paths to the root: S5=>S4=>S1 and S5=>S3=>S1.
- S3 is the switch with the lower ID; its port 2 is enabled and S5 port 2 is enabled.
- S6 and S7 reach the root through S2 and S3 respectively;
- It enable S6 port 1, S2 port 3, S7 port 2 and S3 port 3.

Example: Switch with Segments

According to Rule 3:

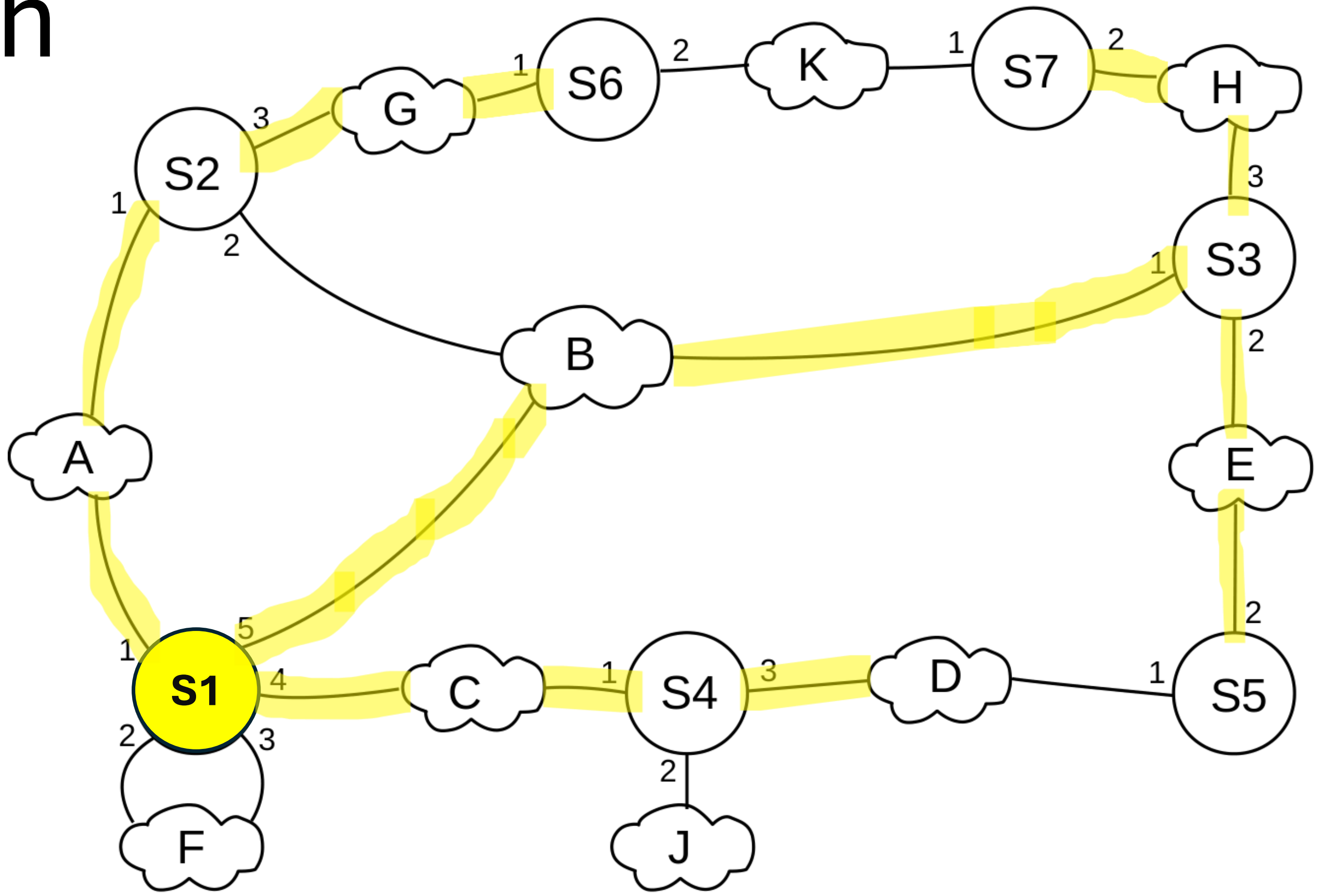
- S2 port 2: Not enabled because network B connects directly to the root (S1).



Example: Switch with Segments

According to Rule 3:

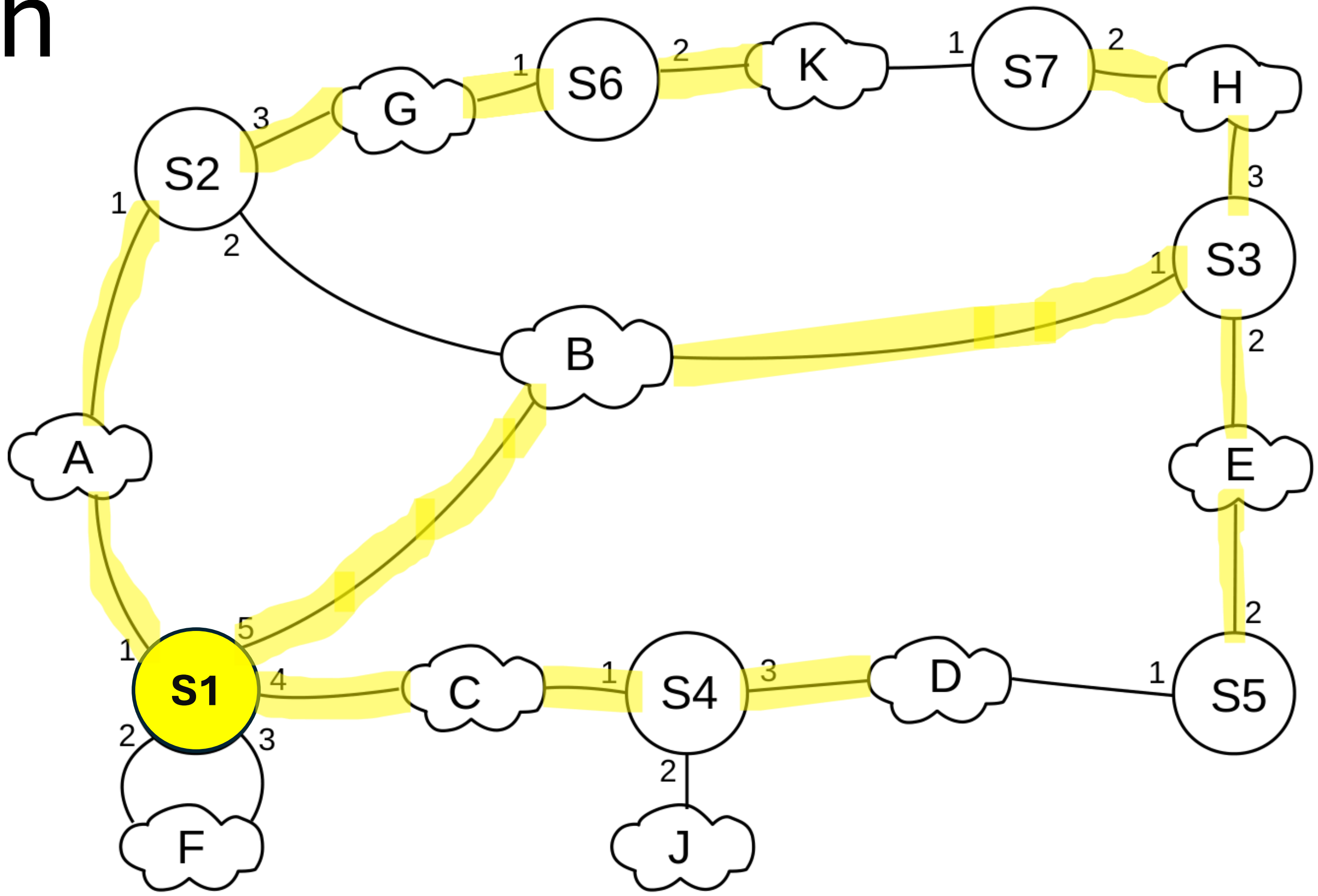
- S2 port 2: Not enabled because network B connects directly to the root (S1).
- S4 port 3: Enabled because S4 is closer to the root, allowing network D to connect. S5 port 1 is not enabled.



Example: Switch with Segments

According to Rule 3:

- S2 port 2: Not enabled because network B connects directly to the root (S1).
- S4 port 3: Enabled because S4 is closer to the root, allowing network D to connect. S5 port 1 is not enabled.
- S6 and S7: Both have the same path length to the root, but S6 has a smaller ID, so S6 enables port 2. S7 port 1 is not enabled.



Example: Switch with Segments

According to Rule 4:

- S4 enables port 2 for host J and S1 port 2 for network F (lower-numbered connection
- **Disabled Ports**: S1 port 3, S2 port 2, S5 port 1, and S7 port 1.