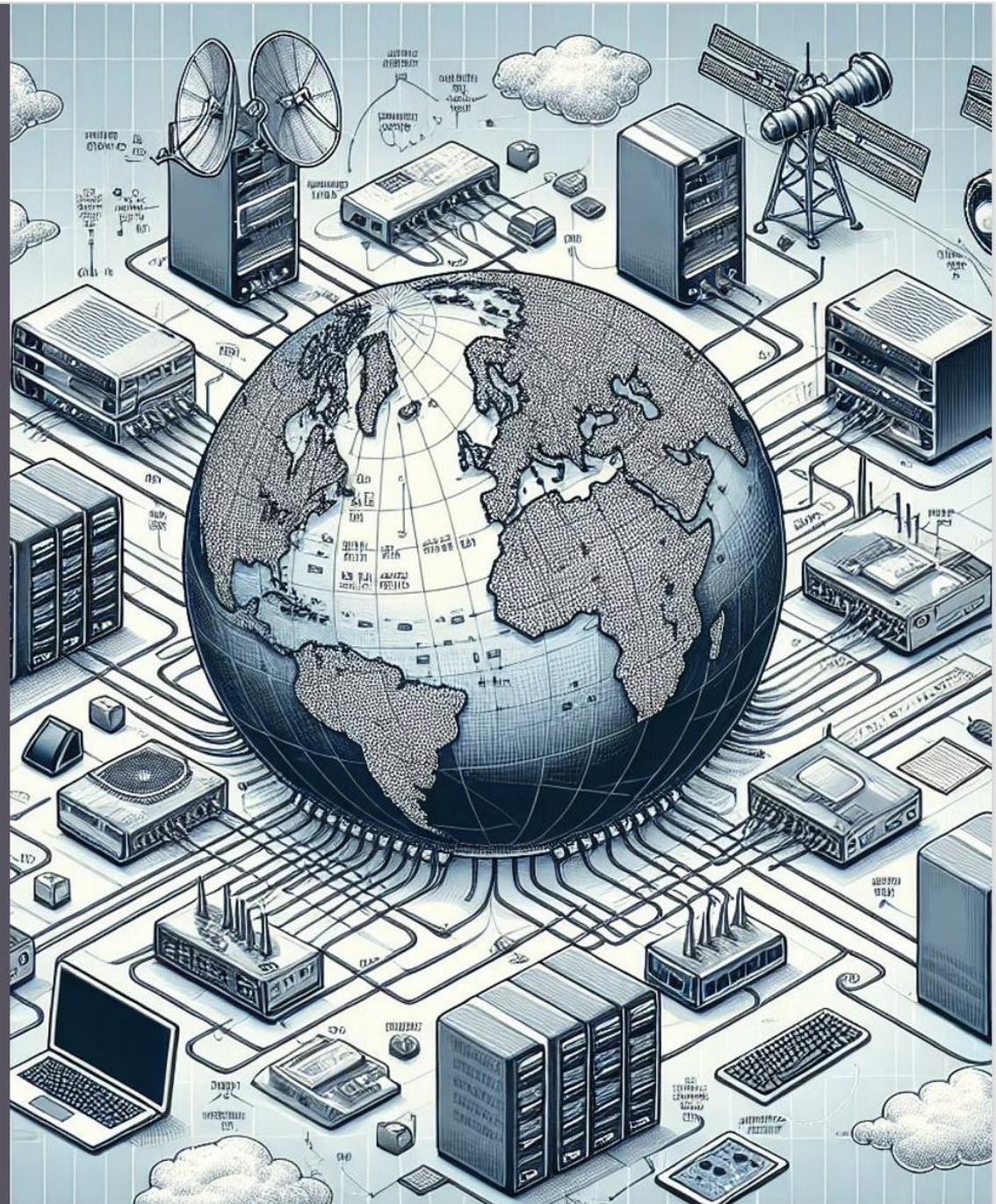
UAB COLLEGE OF ARTS AND SCIENCES
The University of Alabama at Birmingham
Department of Computer Science

CS 334/534 NETWORKING

Dr. Ragib Hasan

Lecture 2.3:
Communication Protocols and Switch



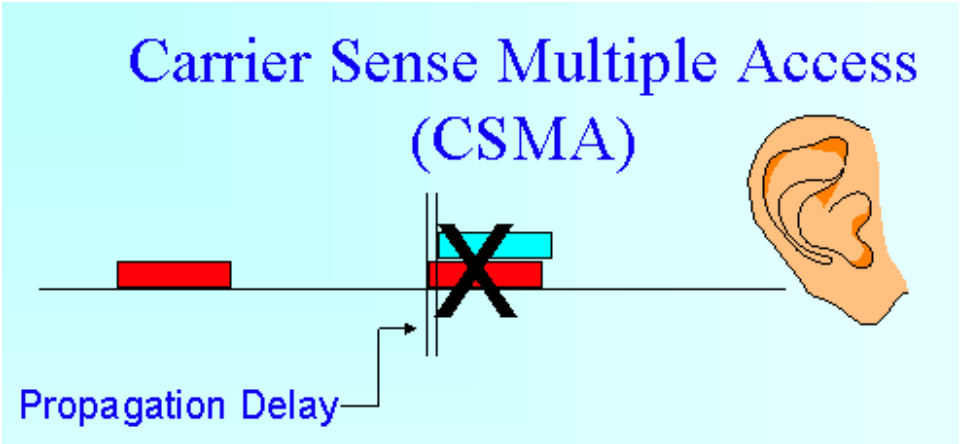
Lecture goals

- Learning about communication protocols:
  - CSMA
  - ALOHA
- Ethernet Switch
- Learning process of Switches

- Book reference: Chapter 2, section 2.1.10 to 2.4.1

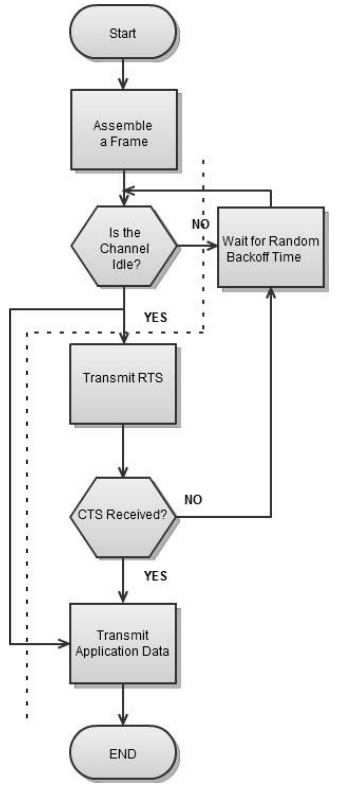
CSMA

- **Carrier Sense Multiple Access**
- It manages **access to a shared communication medium**.
- It ensures that multiple devices can share the same transmission channel efficiently by using a "**listen before you talk**" strategy.
- Propagation delay means two nodes may not hear each other's **just-started transmission.**



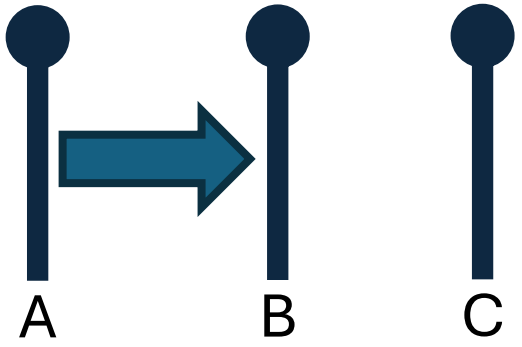
CSMA

- Algorithm
  - **Sense for idle**:
    - If the transmission channel is busy, wait for it to be idle.
    - Avoids collisions and time wastage.
  - **Transmit Frame and Monitor**:
    - No Collision: Frame is successfully delivered.
    - Collision Detected: Abort transmission immediately.
    - Reason: Prevents wasting time on already corrupted data.
  - **Handle Collision**:
    - Perform Exponential Backoff: Wait for a random interval based on the collision count.
    - Retry transmission.



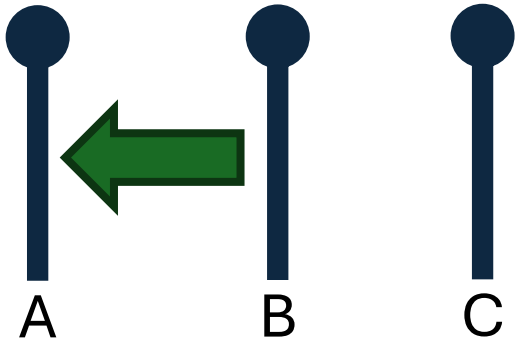
ALOHA

- **Additive Links On-Line Hawaii Area**
- A **multiple access protocol** for transmission of data via a shared network channel.
- Each node or station transmits a frame **without trying to detect** whether the transmission **channel is idle or busy**.
- If the channel is idle, then the frames will be successfully transmitted. Otherwise, **collision occurred.**
- This scenario mirrors **ground stations transmitting to a satellite** without seeing each other.
- Developed in the 70s for packet radio networks.



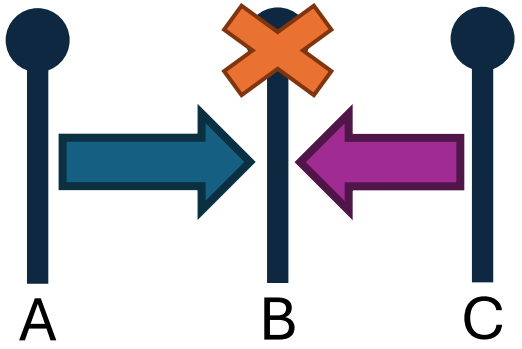
ALOHA

- **Additive Links On-Line Hawaii Area**
- A **multiple access protocol** for transmission of data via a shared network channel.
- Each node or station transmits a frame **without trying to detect** whether the transmission **channel is idle or busy**.
- If the channel is idle, then the frames will be successfully transmitted. Otherwise, **collision occurred.**
- This scenario mirrors **ground stations transmitting to a satellite** without seeing each other.
- Developed in the 70s for packet radio networks.



ALOHA

- **Additive Links On-Line Hawaii Area**
- A **multiple access protocol** for transmission of data via a shared network channel.
- Each node or station transmits a frame **without trying to detect** whether the transmission **channel is idle or busy**.
- If the channel is idle, then the frames will be successfully transmitted. Otherwise, **collision occurred.**
- This scenario mirrors **ground stations transmitting to a satellite** without seeing each other.
- Developed in the 70s for packet radio networks.

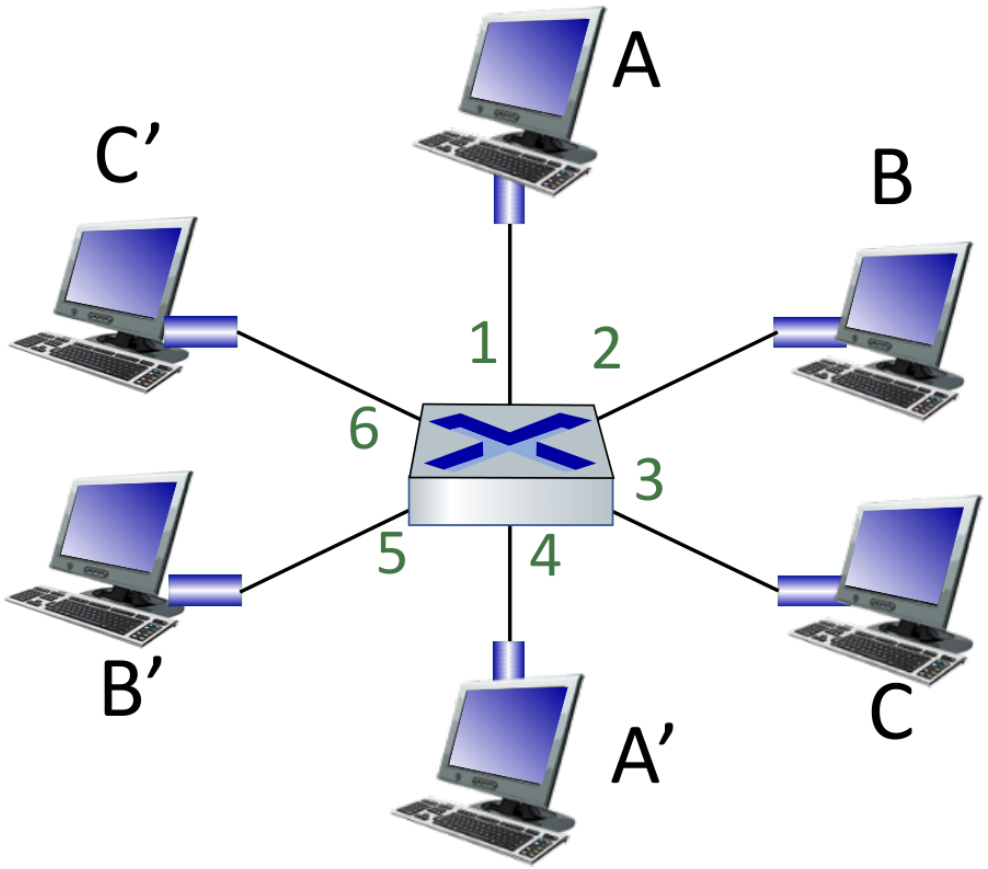


ALOHA

- **Pure ALOHA**
  - Start transmissions **immediately** after having a frame to send.
  - If there is collision and the frame is destroyed, wait **a random time** and retransmit.
- **Slotted ALOHA**
  - Start transmissions only at **fixed time slots.**
  - Significantly fewer collisions than pure ALOHA

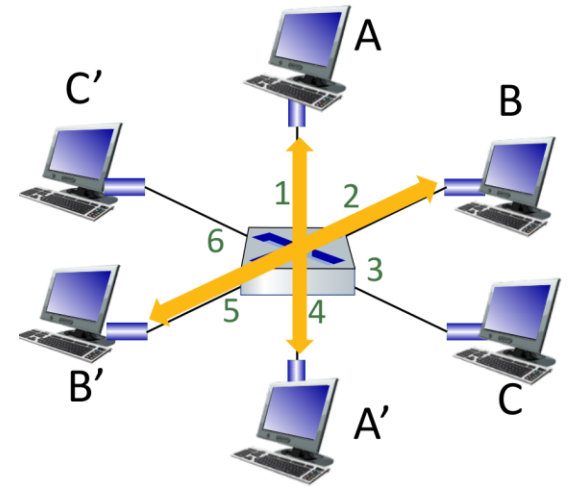
Switch

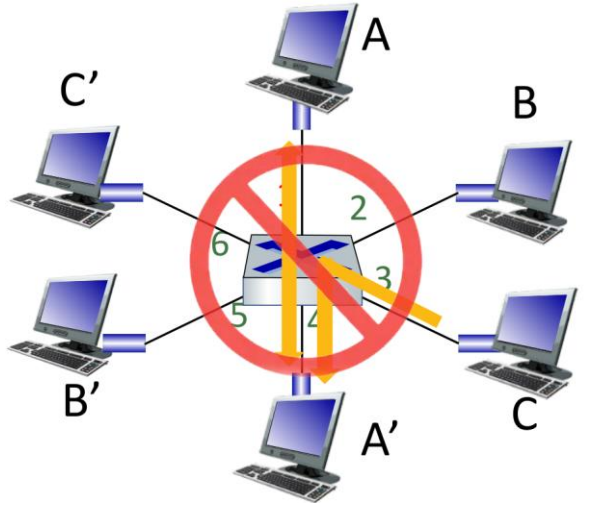
- Switch is a link-layer device; takes an active role
- **Store, forward** Ethernet (or other type of) frames
- **Examine** incoming frame's **MAC address**, selectively **forward** frame to One-or-more outgoing links
- When frame is to be forwarded on segment, uses CSMA/CD to access segment.



Switch

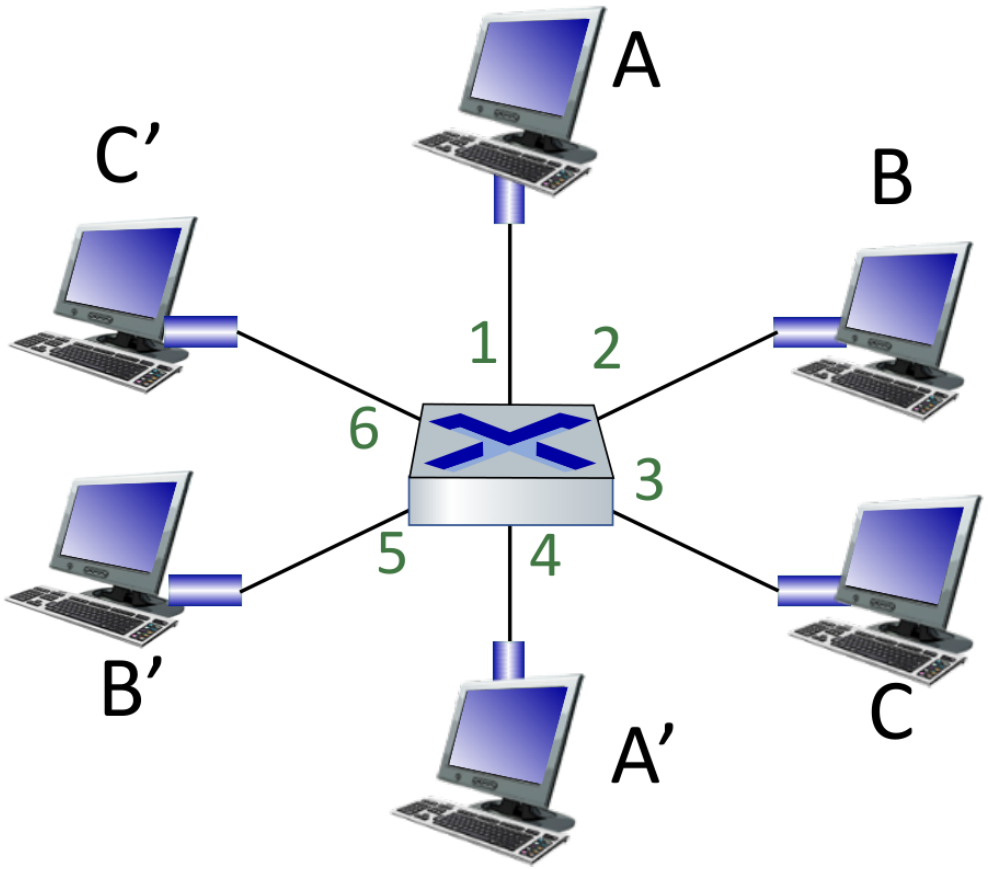
- **Transparent**: hosts unaware of presence of switches
- Hosts have **dedicated, direct connection** to switch
- **No collisions; full duplex**
  - A-to-A’ and B-to-B’ can transmit simultaneously
  - But A-to-A’ and C to A’ can not happen simultaneously
- **Plug-and-play, self-learning**
  - switches do not need to be configured





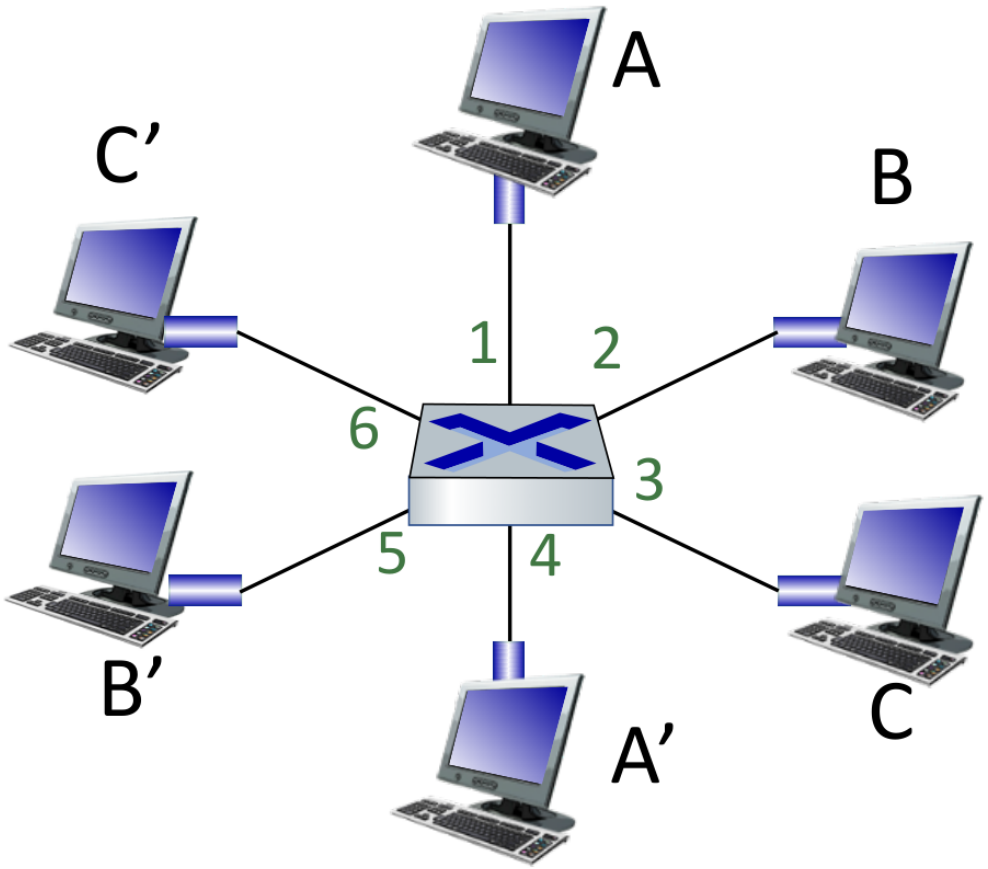
Switch

- **Q:** How does switch know A’ reachable via interface 4, B’ reachable via interface 5?



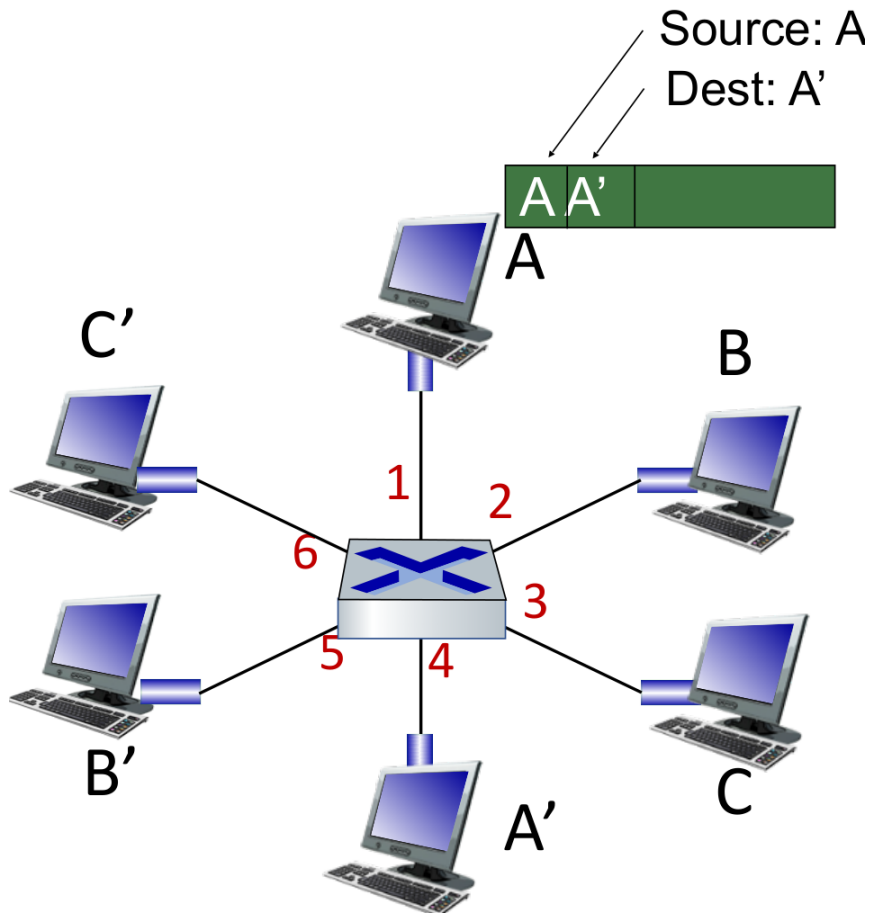
Switch

- **Q:** How does switch know A’ reachable via interface 4, B’ reachable via interface 5?
- **A:** Each switch has a switch table, each entry:
  - MAC address of host
  - Interface to reach host
  - Time stamp
- Looks like a routing table!



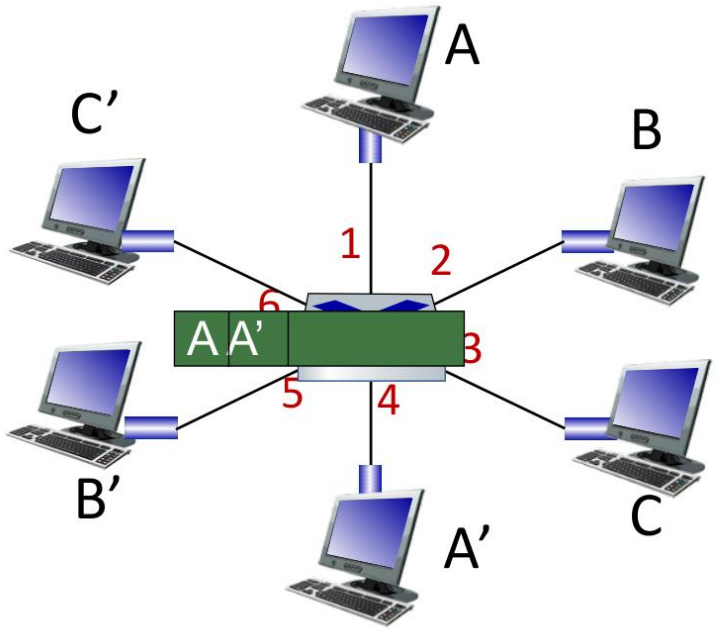
Switch: Self Learning

- Switch **learns** which hosts can be reached through which interfaces



Switch: Self Learning

- Switch **learns** which hosts can be reached through which interfaces
- When frame received, switch “learns” the **location of sender, incoming LAN segment**
- Initially the switch table is empty

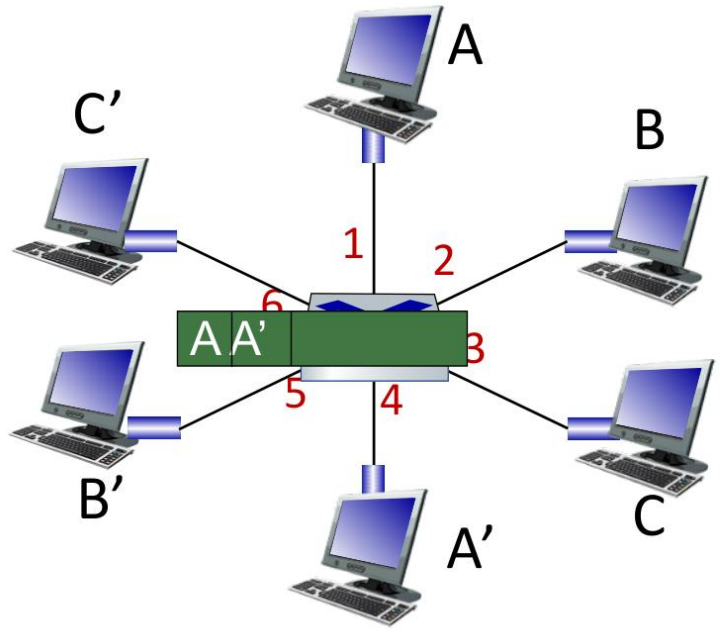


MAC addr	interface	TTL

Switch table (initially empty)

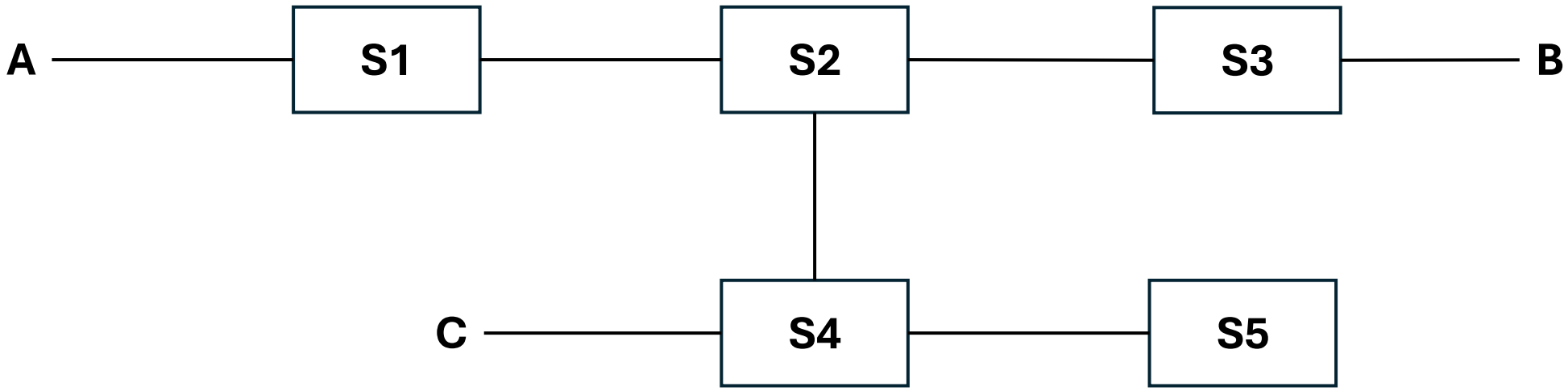
Switch: Self Learning

- Switch **learns** which hosts can be reached through which interfaces
- When frame received, switch “learns” the **location of sender, incoming LAN segment**
- Initially the switch table is empty
- Records sender/location pair in switch table



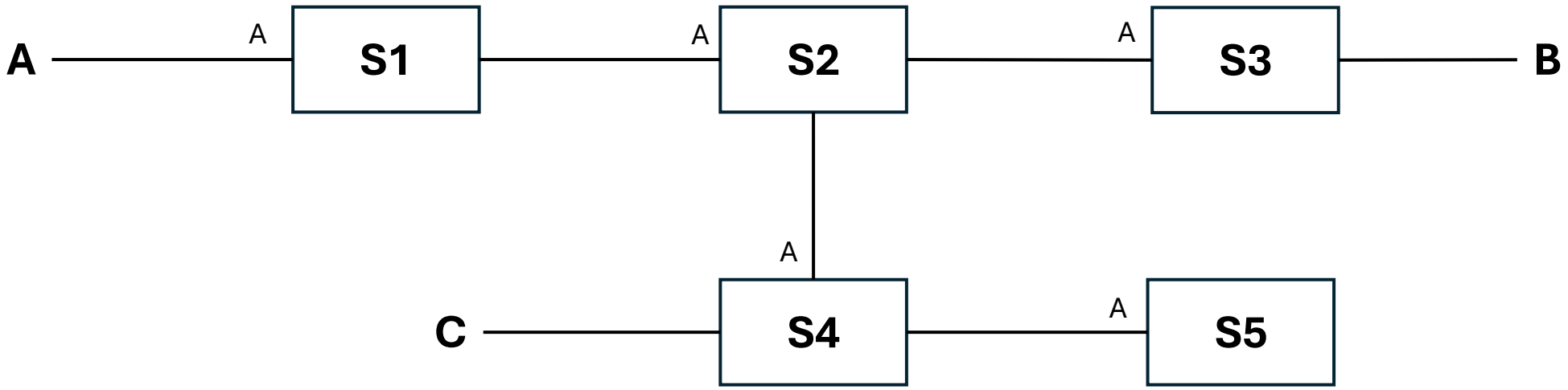
MAC addr	interface	TTL
A	*1*	*60*

Switch table (initially empty)



Multiple Switch

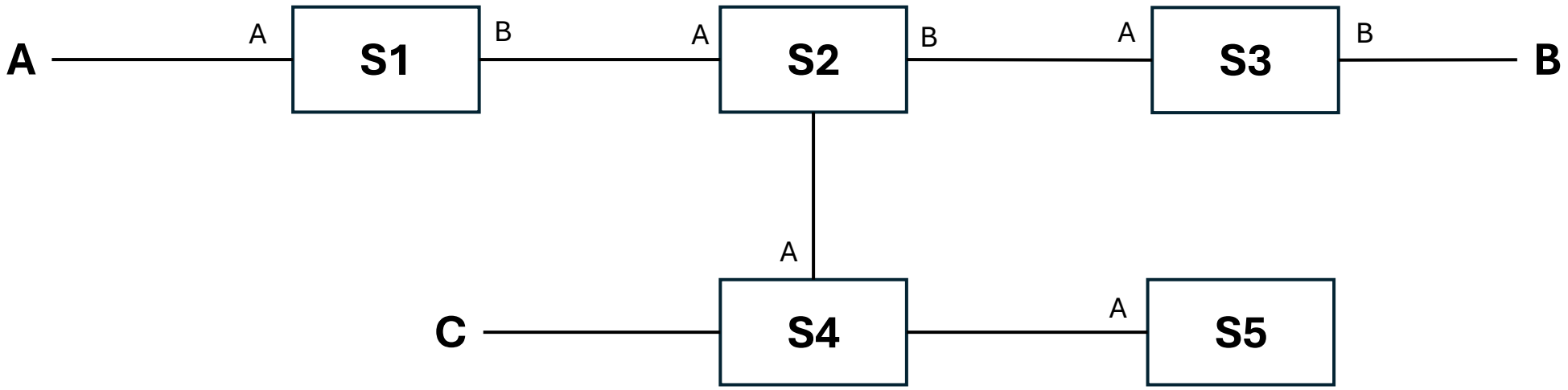
- When a switch gets a packet, it checks its forwarding table to find where to send it.
- If the destination is in the table, the switch sends the packet directly to the right connection.
- If the destination isn't in the table, the switch sends the packet out through all connections and store the source location.
- Over time, the switch learns where devices are and updates its table.



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

Multiple Switch

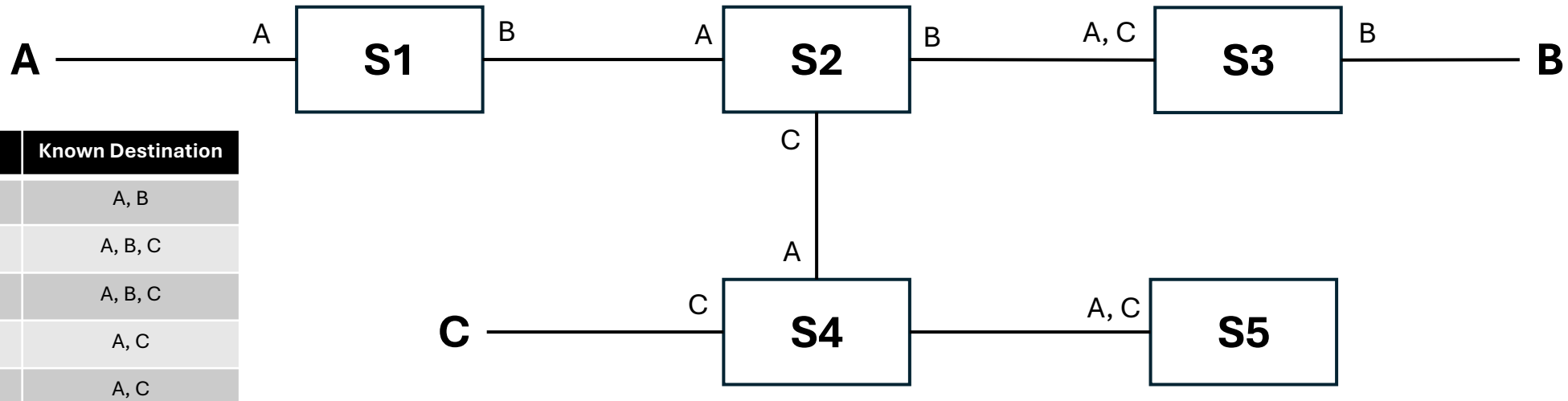
- When a switch gets a packet, it checks its forwarding table to find where to send it.
- If the destination is in the table, the switch sends the packet directly to the right connection.
- If the destination isn't in the table, the switch sends the packet out through all connections and store the source location.
- Over time, the switch learns where devices are and updates its table.



Second packet: B sends to A; this packet goes directly to A; only S3, S2 and S1 learn where B is

Multiple Switch

- When a switch gets a packet, it checks its forwarding table to find where to send it.
- If the destination is in the table, the switch sends the packet directly to the right connection.
- If the destination isn't in the table, the switch sends the packet out through all connections and store the source location.
- Over time, the switch learns where devices are and updates its table.

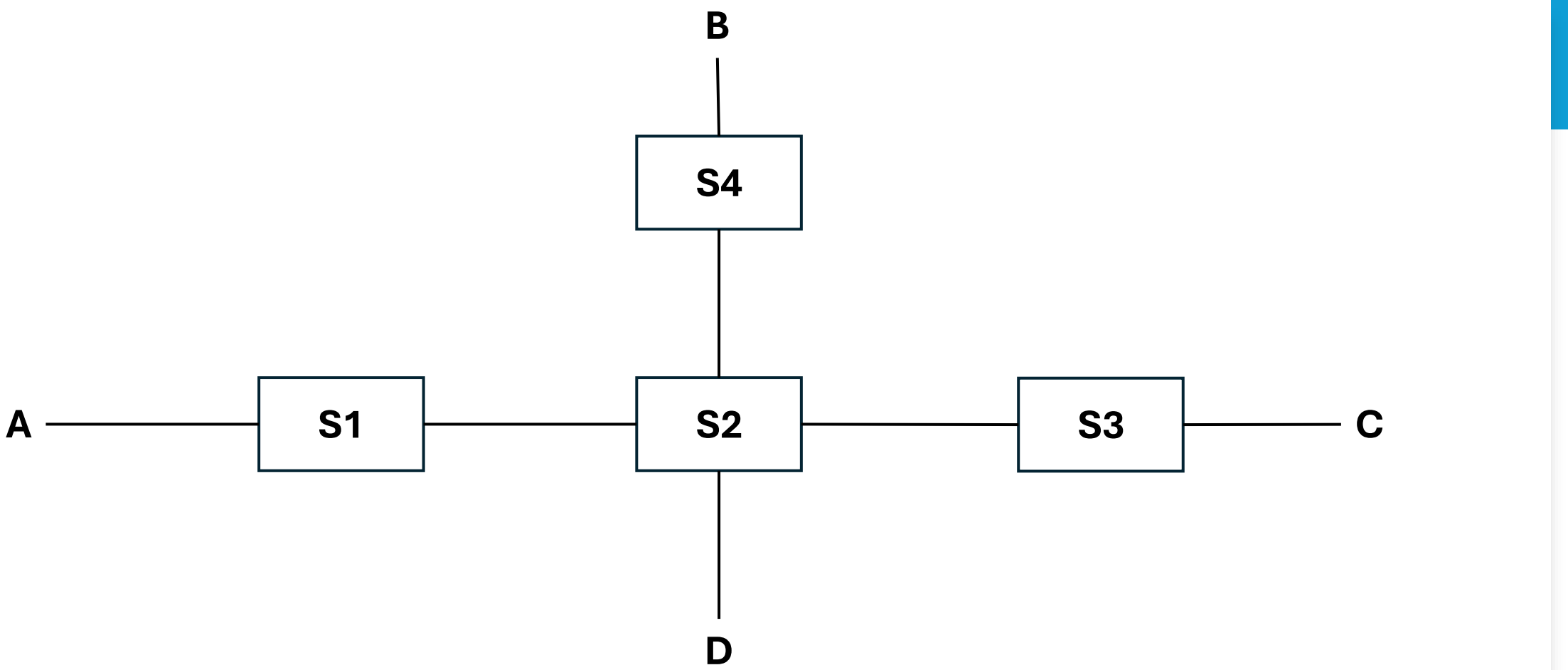


Switch	Known Destination
S1	A, B
S2	A, B, C
S3	A, B, C
S4	A, C
S5	A, C

Third packet: C sends to B; S4 does not know where B is so this packet goes to S5; S2 does know where B is so the packet does not go to S1.

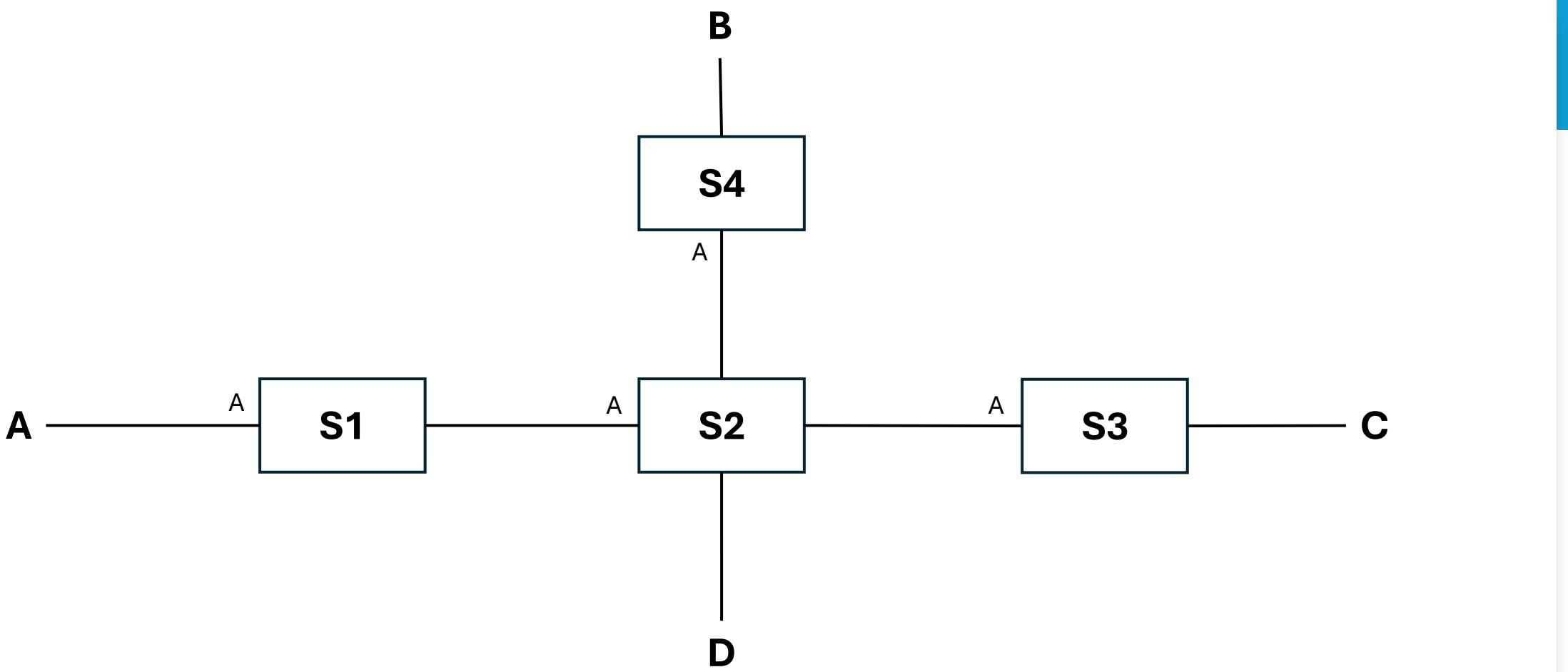
Multiple Switch

- When a switch gets a packet, it checks its forwarding table to find where to send it.
- If the destination is in the table, the switch sends the packet directly to the right connection.
- If the destination isn't in the table, the switch sends the packet out through all connections and store the source location.
- Over time, the switch learns where devices are and updates its table.



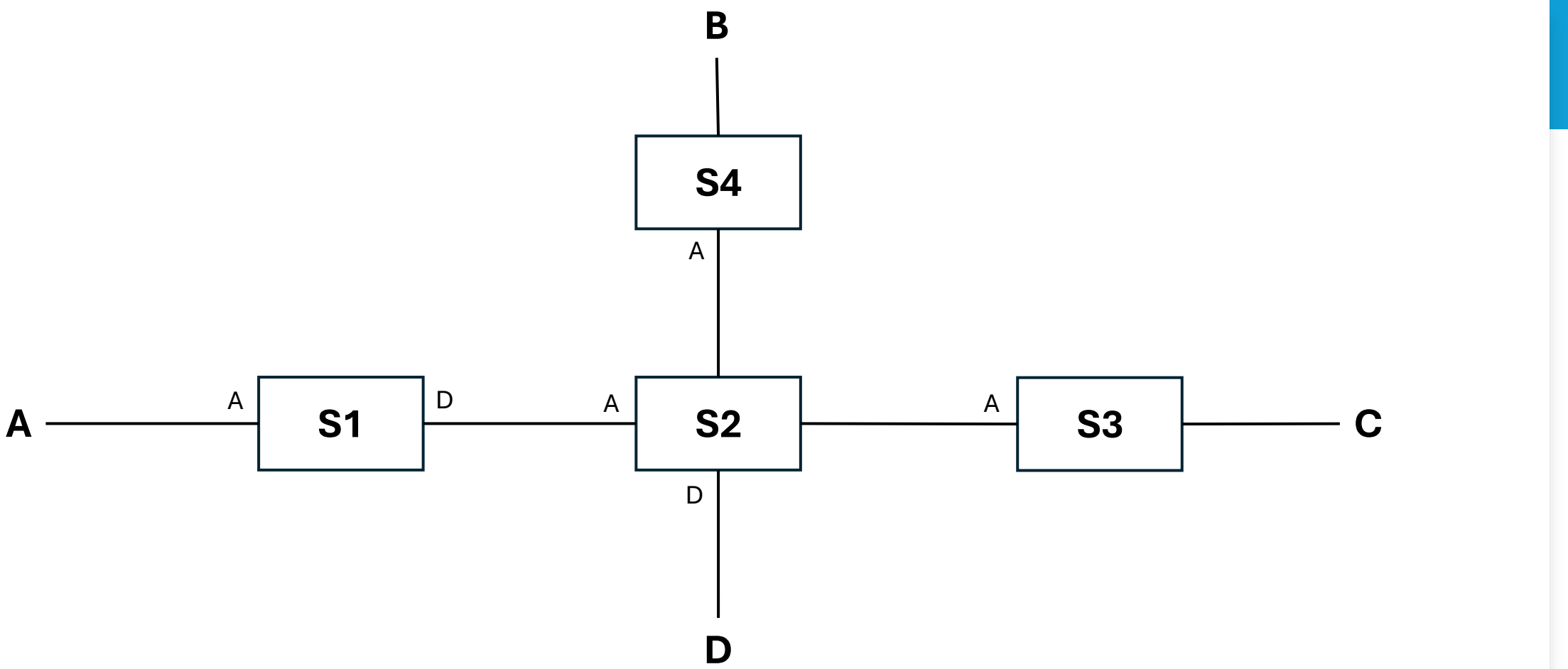
Exercise

- A sends to D
- D sends to A
- A sends to B
- B sends to D



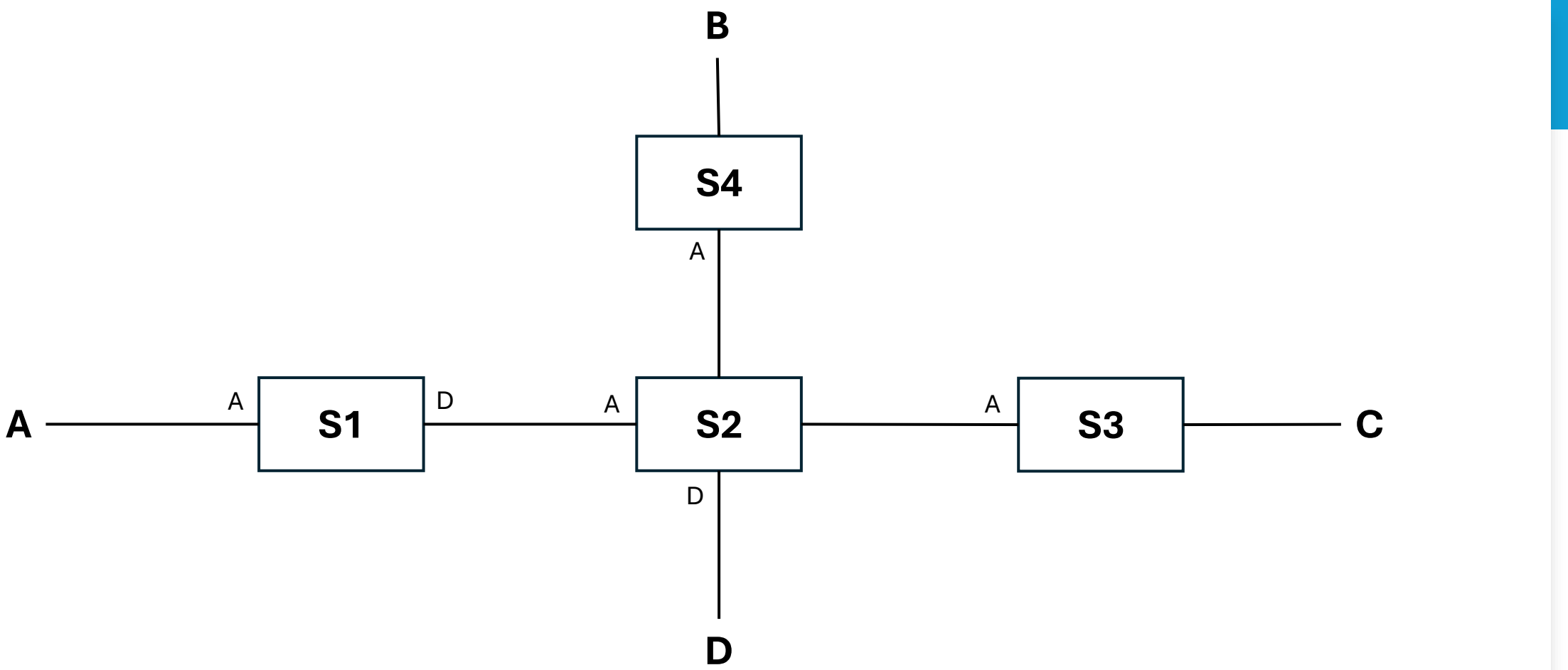
Exercise

- **A sends to D**
- **D sends to A**
- **A sends to B**
- **B sends to D**



Exercise

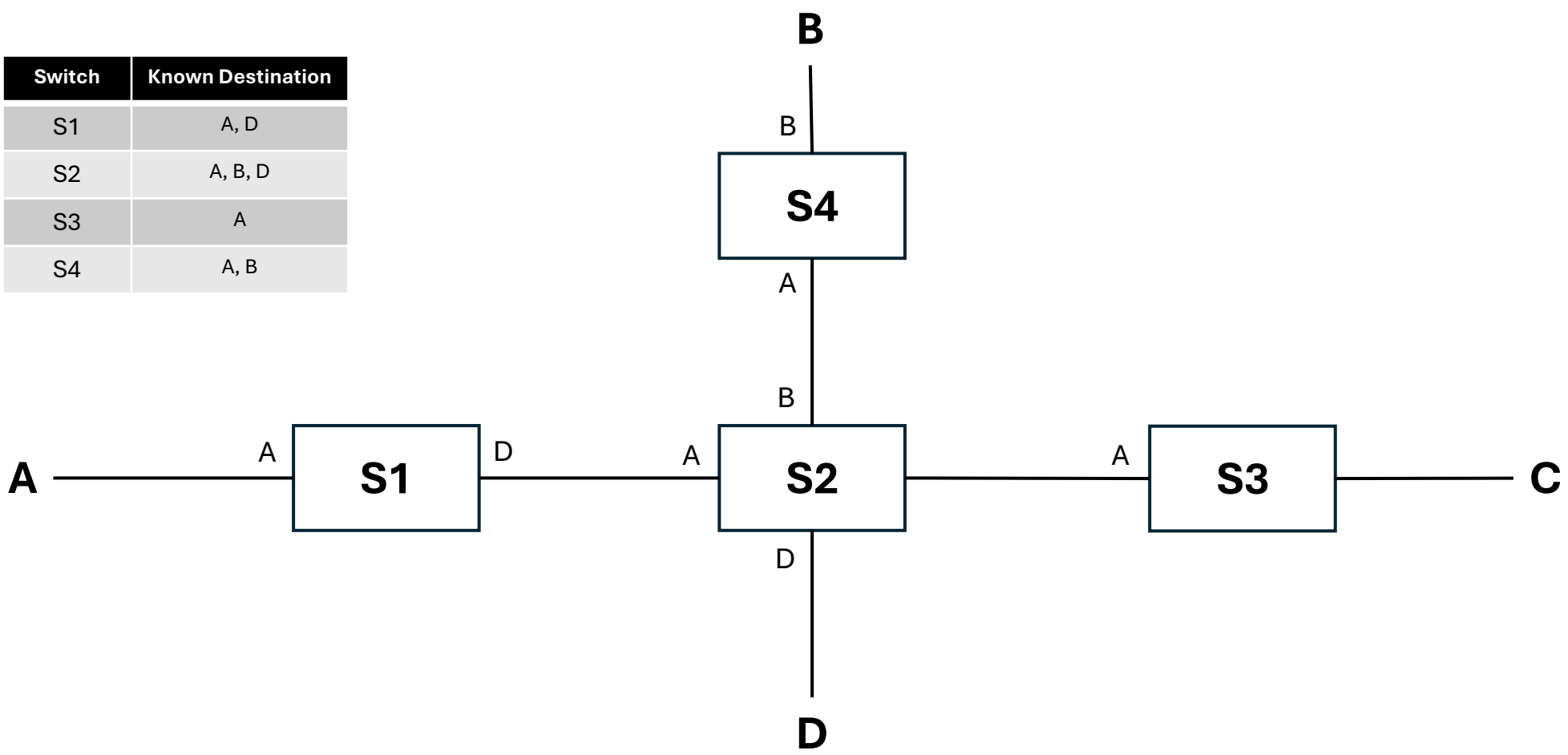
- **A sends to D**
- **D sends to A**
- **A sends to B**
- **B sends to D**



Exercise

- **A sends to D**
- **D sends to A**
- **A sends to B**
- **B sends to D**

Switch	Known Destination
S1	A, D
S2	A, B, D
S3	A
S4	A, B



Exercise

- **A sends to D**
- **D sends to A**
- **A sends to B**
- **B sends to D**