

Computer Networks

COL 334/672

Local Area Network

Tarun Mangla

Slides adapted from KR

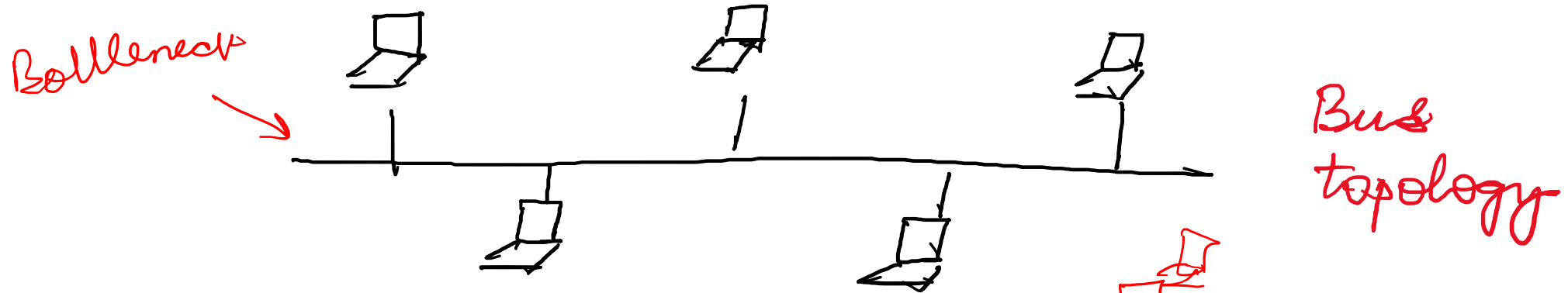
Sem 1, 2024-25

Recap

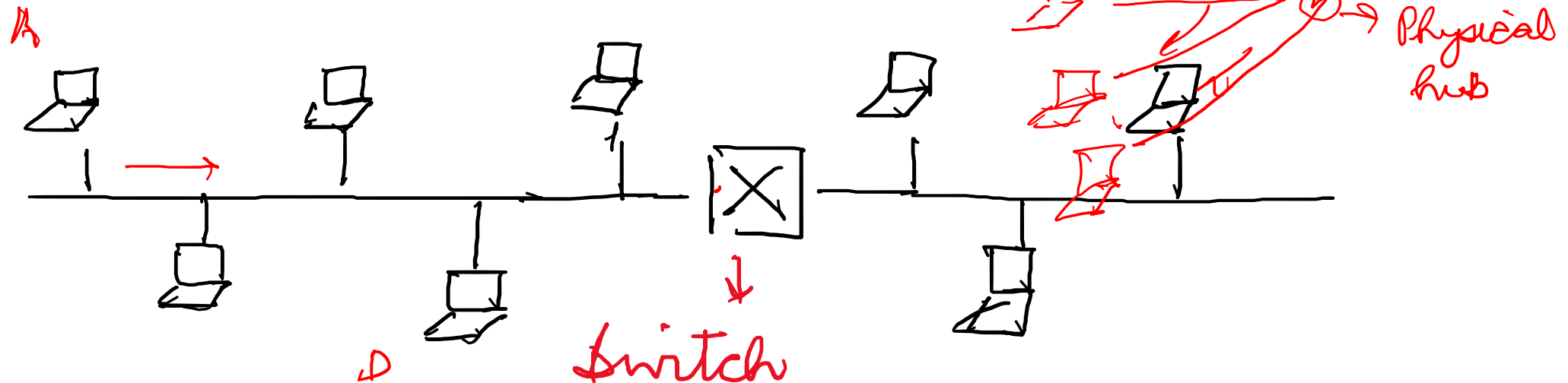
- Link-layer services
 - Framing
 - Error detection
 - Reliability
 - Medium Access Control (MAC)
- Ethernet protocol

So Far..

- We know how to communicate in case of a multiple-access link

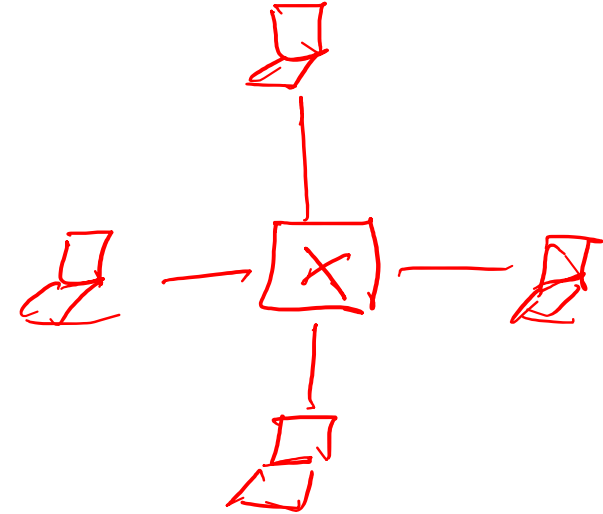


- How do we scale?



Switched Network

- Star topology
- Switches in both link layer or L2 (Ethernet protocol) and network layer or L3 (IP)
- Use store and forward approach
- L3 switches are also called routers, while L2 switches are also called bridges, ethernet switches
- **This lecture:** L2 switches or bridges or ethernet switch



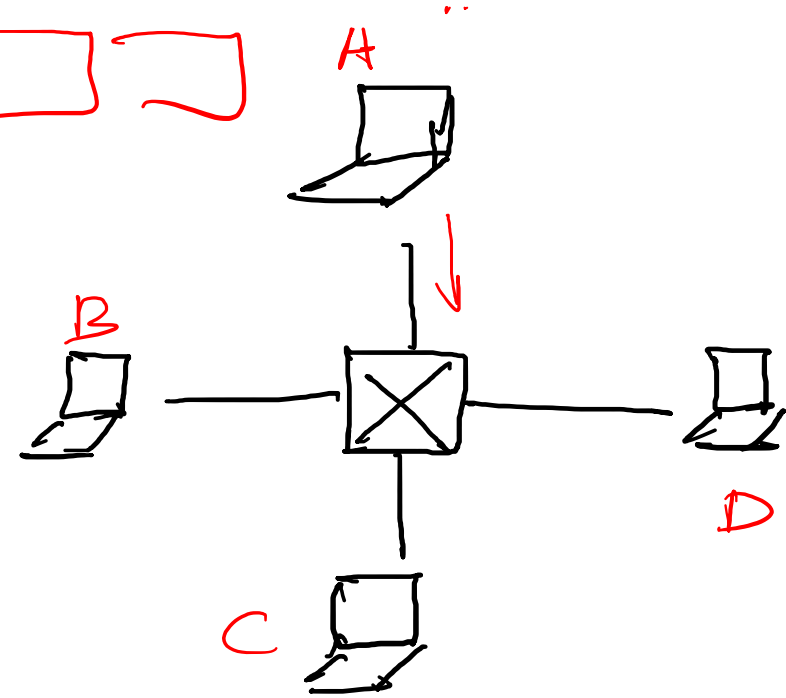
How does L2 switching work?

Requirement 1: Addressing

- Need a way to identify end nodes
- Ethernet uses a 48-bit MAC address burned in NIC ROM, also sometimes software settable
- E.g., 1A-2F-BB-76-09-AD

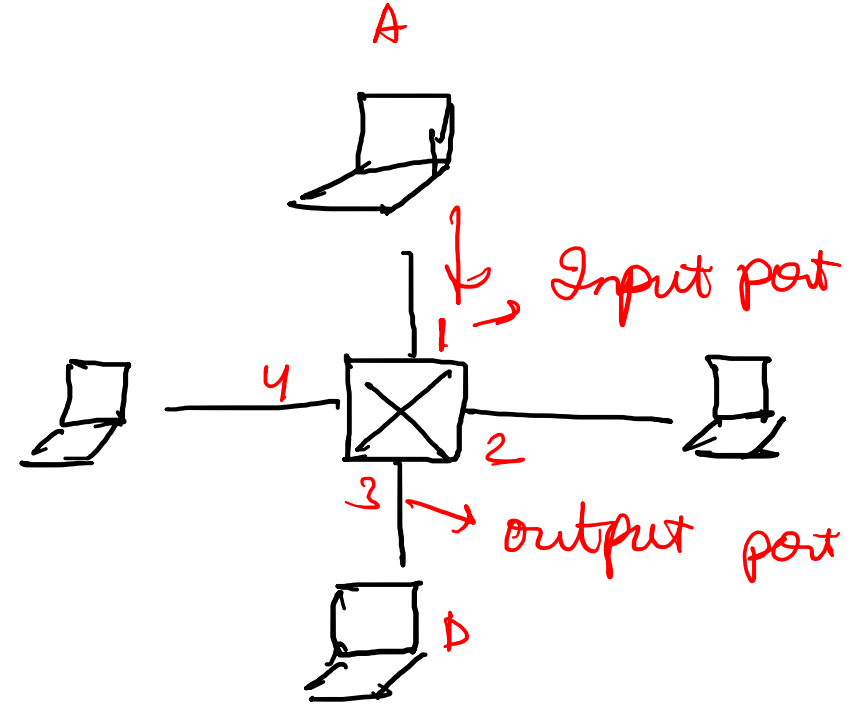
IP address: 32 bits

hexadecimal (base 16) notation
(each "numeral" represents 4 bits)



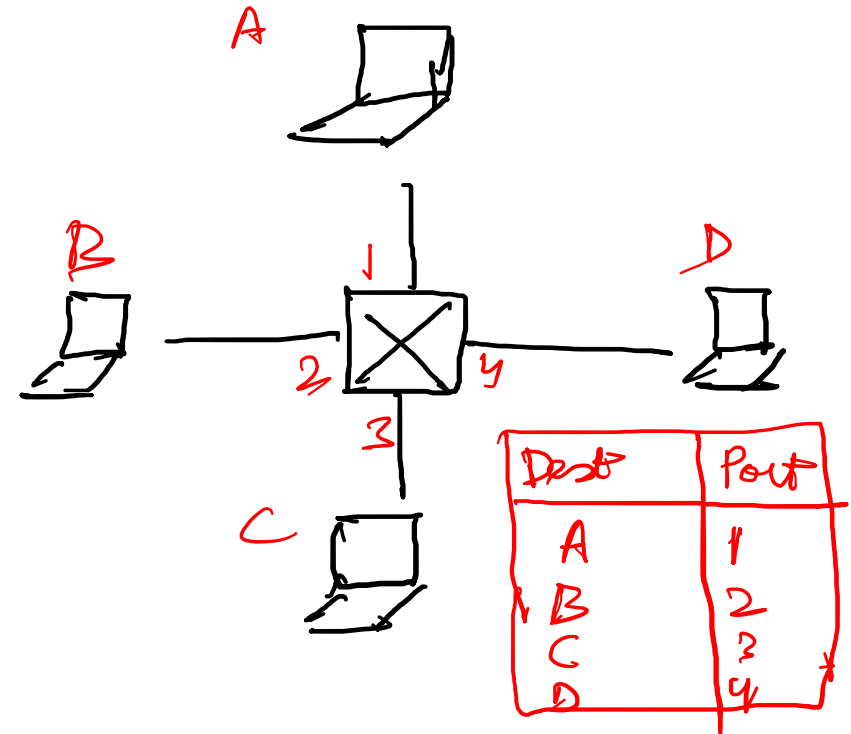
How does L2 switching work?

- **Requirement 1:** Addressing
- **Requirement 2:** Need a way to identify input and output ports on switch
 - Using a number
 - Name of the host it leads to



How does L2 switching work?

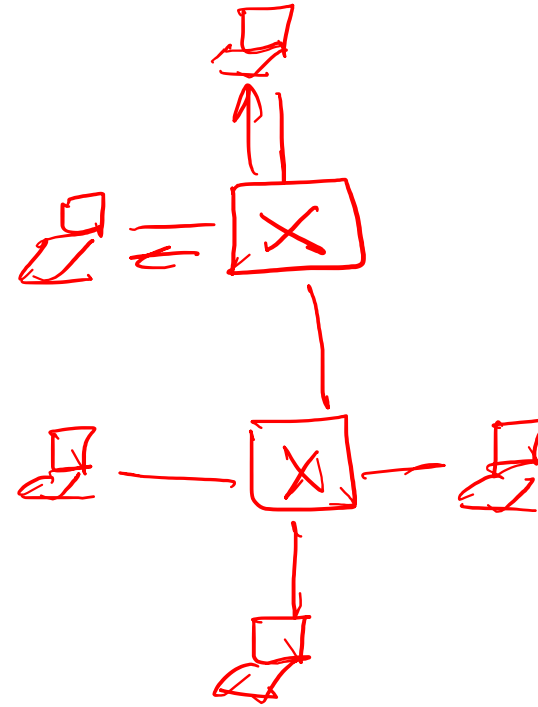
- **Requirement 1:** Addressing
- **Requirement 2:** Need a way to identify input and output ports on switch
- **Requirement 3:** Switching or forwarding
 - Moving packet from its input port to the appropriate output port
 - Forwarding table: map of destination address to output port



How does L2 switching work?

- **Requirement 1:** Addressing
- **Requirement 2:** Need a way to identify input and output ports on switch
- **Requirement 3:** Switching or forwarding → data plane
- **Requirement 4:** Need algorithms to fill the forwarding table, also known as routing

control plane

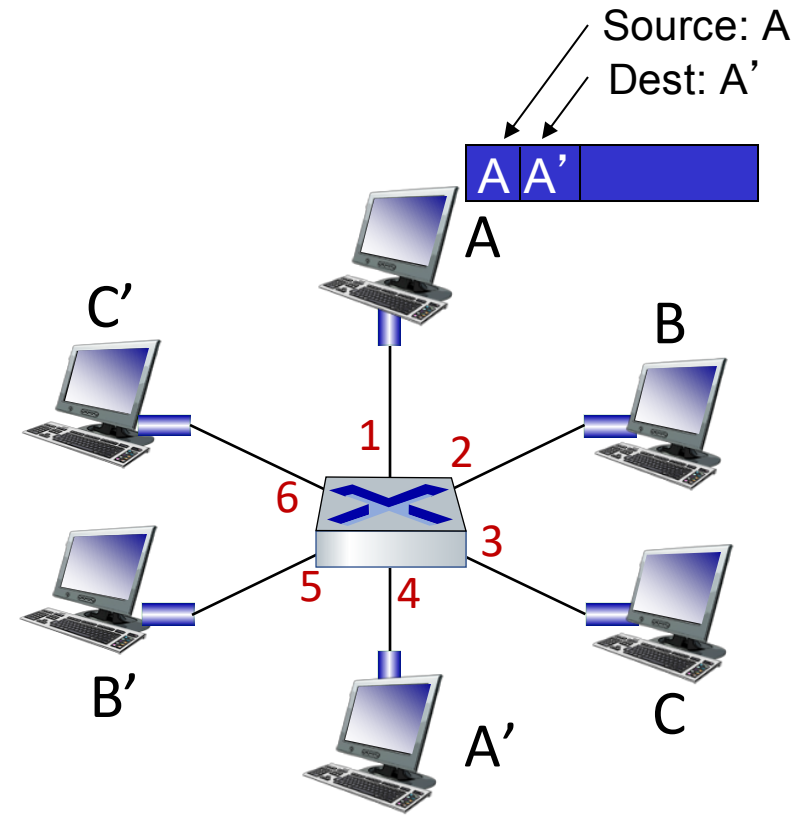


L2 Routing

- Using static tables added by network administrator
 - Difficult to support dynamic addition of nodes (end-hosts and switches)

L2 Routing: Self-learning Switch

- switch *learns* which hosts can be reached through which interfaces
 - when frame received, switch “learns” location of sender: incoming LAN segment
 - records sender/location pair in switch table



MAC addr	interface	TTL
A	1	60

*Switch table
(initially empty)*

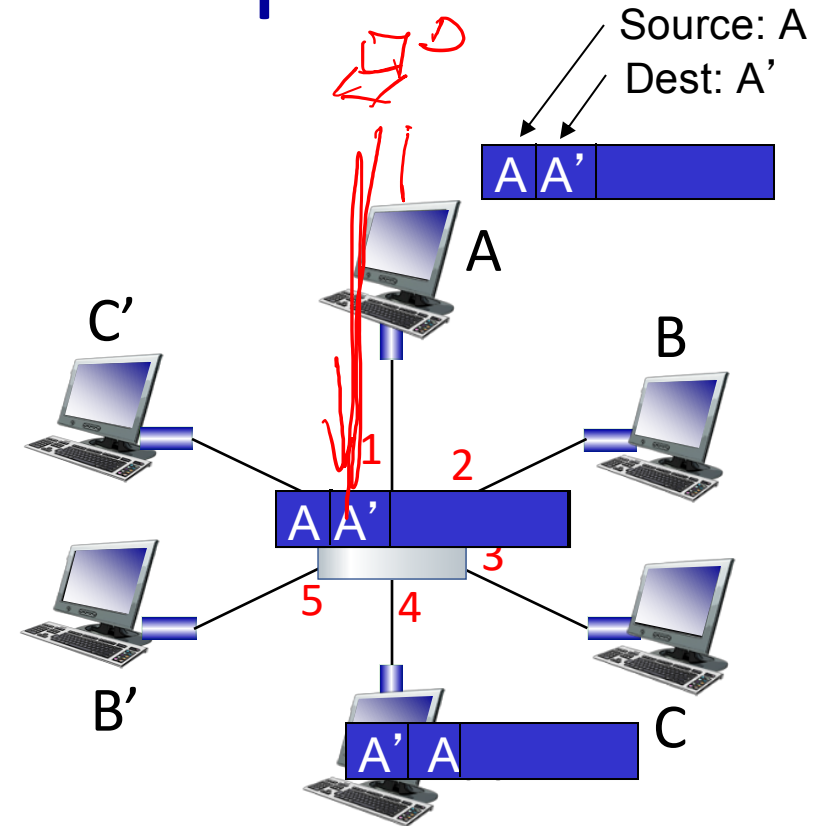
L2 Routing: Self-learning Switch

when frame received at switch:

1. record incoming link, MAC address of sending host
2. index switch table using MAC destination address
3. **if** entry found for destination
 then {
 if destination on segment from which frame arrived
 then drop frame
 else forward frame on interface indicated by entry
 }
 else flood /* forward on all interfaces except arriving interface */

Self-learning, Forwarding: example

- frame destination, A',
location unknown: **flood**
- destination A location
known: **selectively send**
on just one link

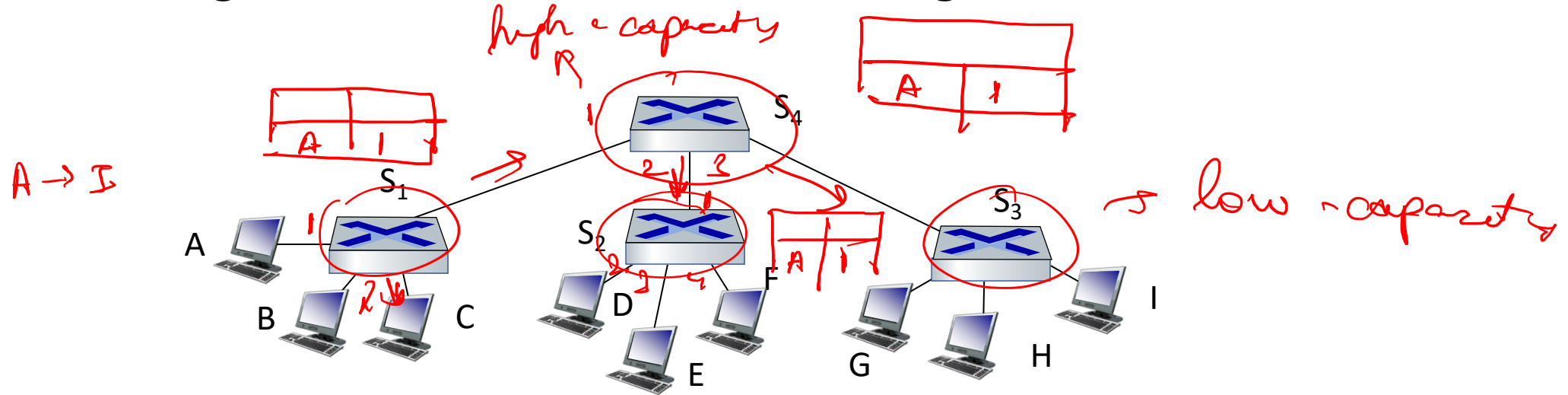


MAC addr	interface	TTL
A	1	60
A'	4	60

*switch table
(initially empty)*

Interconnecting switches

self-learning switches can be connected together:

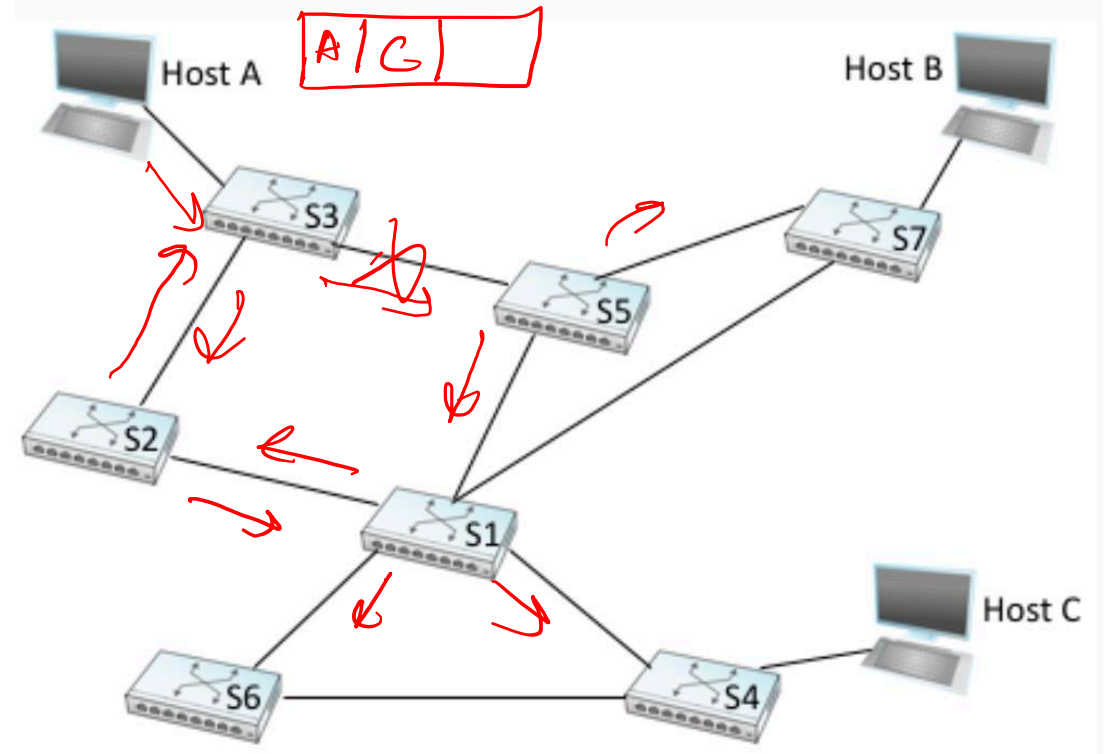


Self learning! (works exactly the same as in single-switch case!)

Q: show switch tables and packet forwarding in S₁, S₂, S₃, S₄

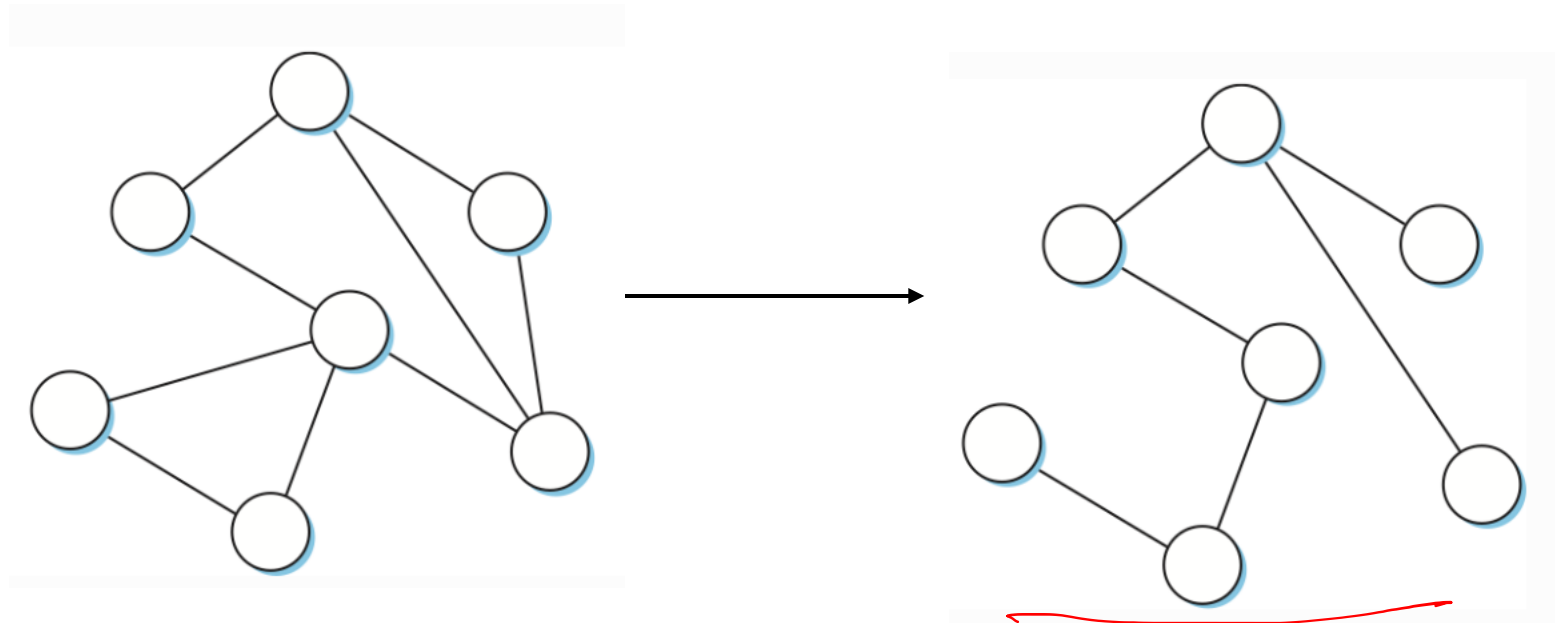
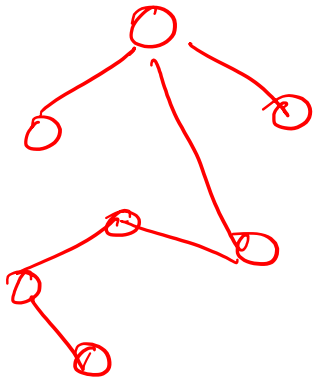
What happens in case of a loop?

- Packets will loop in both direction at S1, S4, and S6
- Why would switched Ethernet have loop?
 - Redundancy
 - By accident: network managed by more than one admins
- Loops are a waste of network bandwidth
- How do we prevent packets from looping?



Break the Loop!

- **Idea:** Create a logical **spanning tree** in the network
 - Spanning tree: subgraph that covers all vertices but contains no cycles
 - Switches will only forward on ports lying on the spanning tree



- **Challenge:** Need to create the same spanning tree in a distributed manner

Spanning Tree Protocol

- Invented by Radia Perlman from DEC

Algorhyme

*I think that I shall never see
A graph more lovely than a tree.*

*A tree whose crucial property
Is loop-free connectivity.*

*A tree which must be sure to span
So packets can reach every LAN.*

*First the Root must be selected.
By ID it is elected.*

*Least cost paths from Root are traced.
In the tree these paths are placed.*

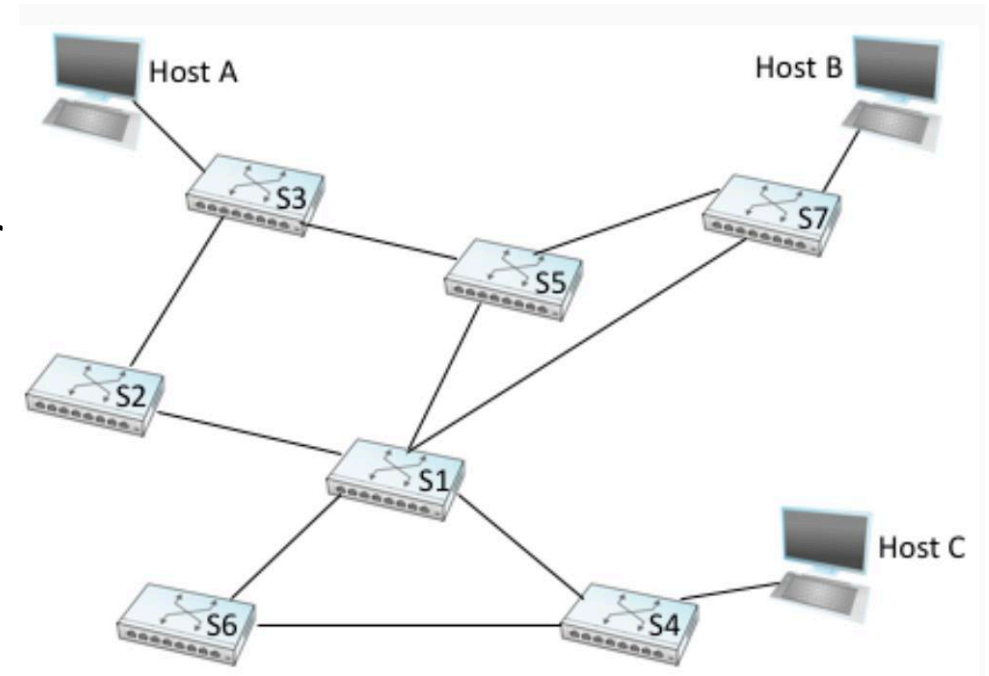
*A mesh is made by folks like me
Then bridges find a spanning tree.*

Spanning Tree Protocol

■ Algorithm

- Elect the switch with smallest ID as the root of the spanning tree
- Identify port that is closest to the root and assign it as a root port
- In case of a tie, select the port with smaller switch ID
- Any port that is not root port is disabled

■ How is it exactly done?

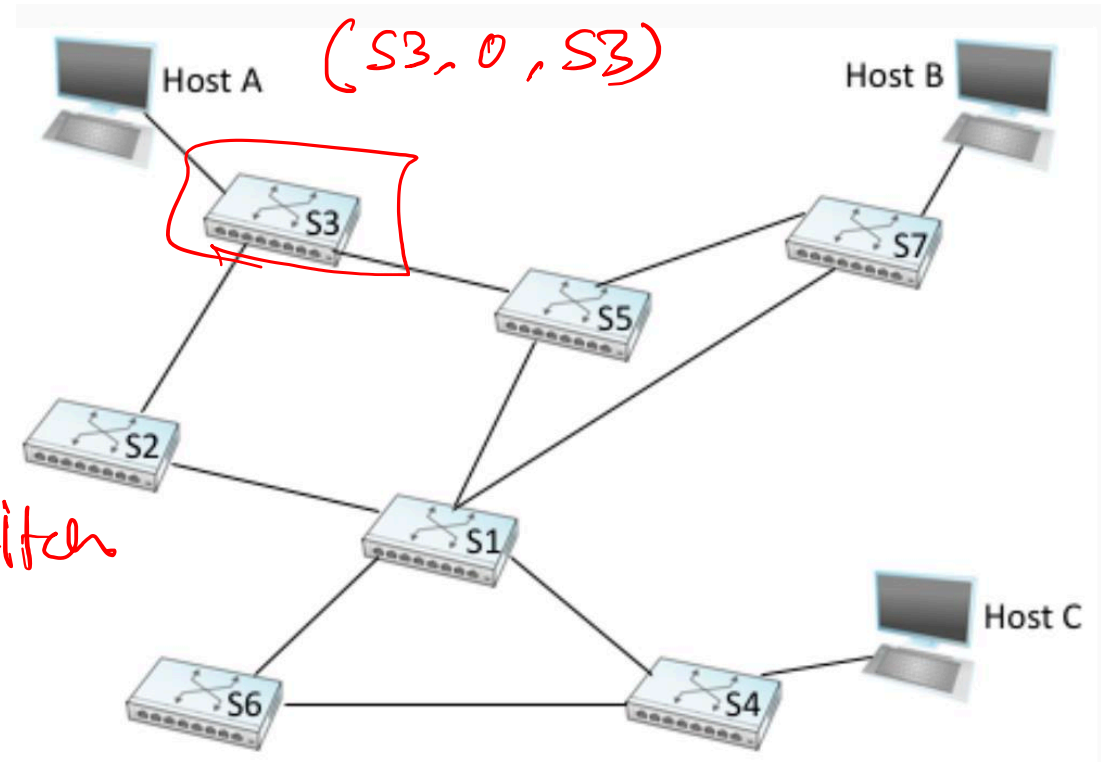


SPT (Details)

Switch

- Each bridge has an ID
 - 8 bytes: 2 bytes configurable, 6 bytes of MAC address
- Bridge X announces its bridge ID to its neighbors along with distance to root

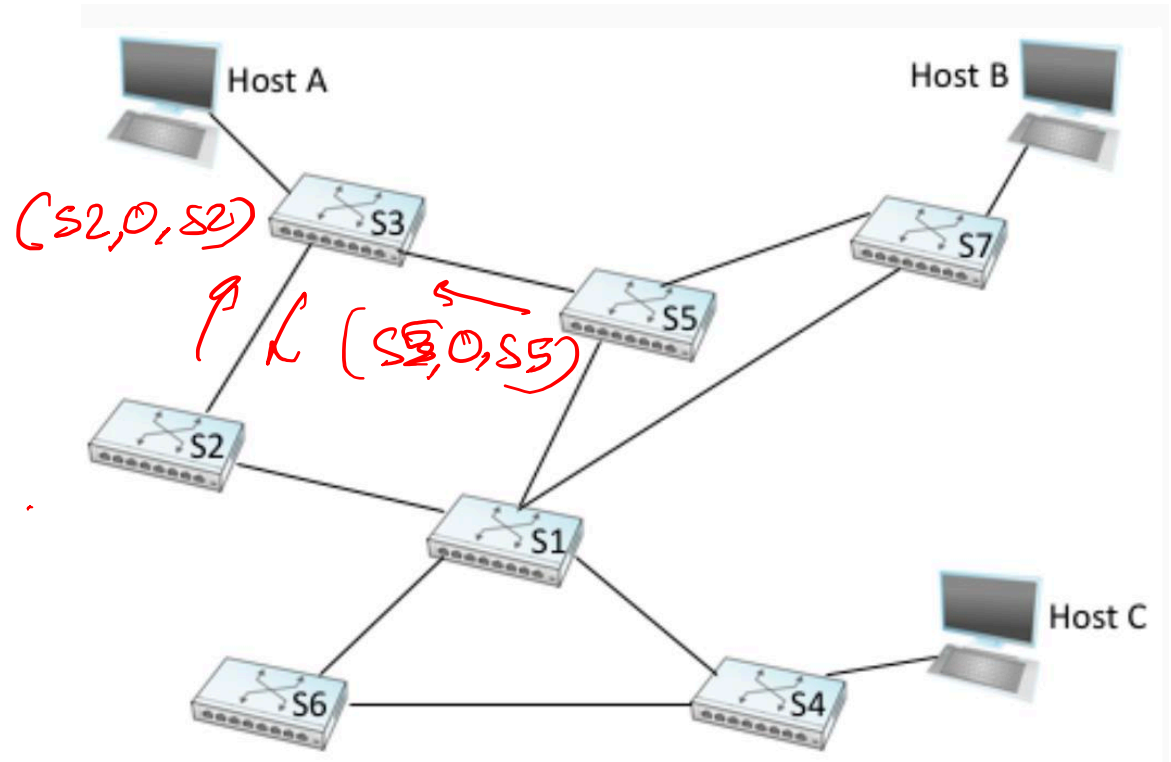
- (Y, d, X)
 - ID of root node,
 - d is distance
 - X: ID of the switch



SPT (Details)

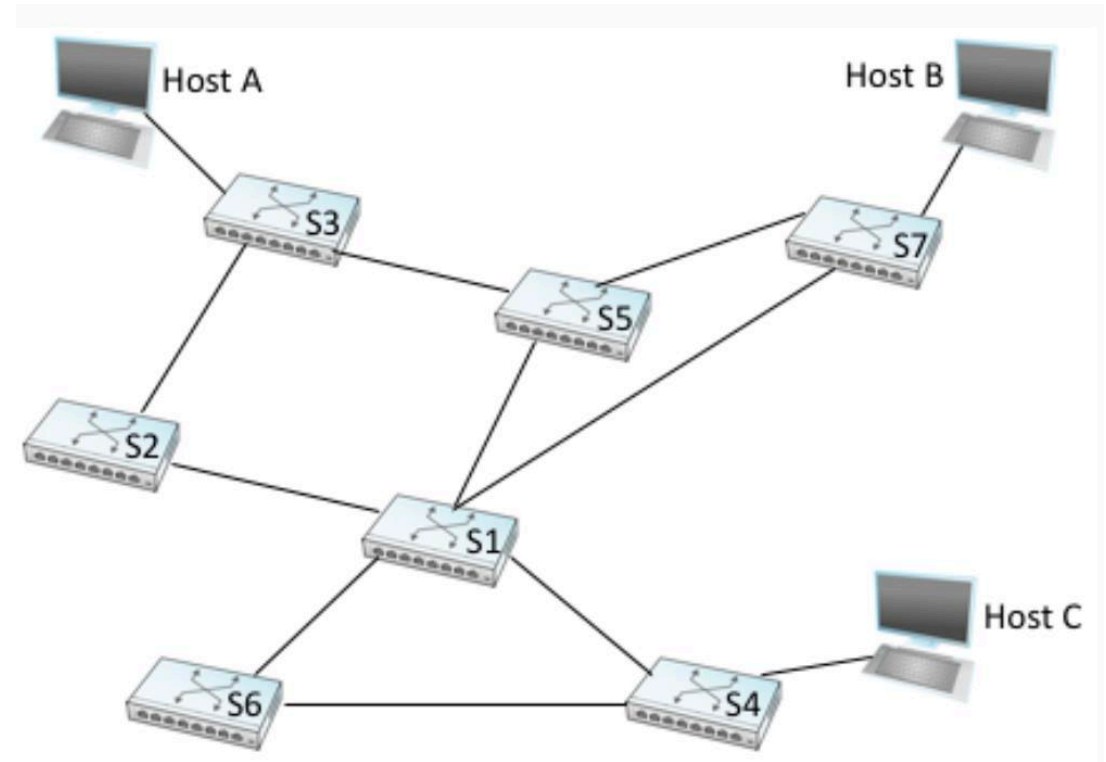
- Each bridge has an ID
 - 8 bytes: 2 bytes configurable, 6 bytes of MAC address
- Bridge X announces its bridge ID to its neighbors along with distance to root
 - (Y, d, X)
- Stop generating configuration messages *as root node*
 - When receives message from a smaller switch ID

$(S1, 0, S1)$



SPT (Details)

- Each bridge has an ID
 - 8 bytes: 2 bytes configurable, 6 bytes of MAC address
- Bridge X announces its bridge ID to its neighbors along with distance to root
 - (Y, d, X)
- Stop generating configuration messages
 - When receives message from a smaller switch ID
- Stop sending on a port
 - Message from a switch that is closer to root or
 - Equally far from the root but with smaller ID



Conclusion

- Link layer services
 - Framing
 - Error detection
 - Reliability
 - Medium Access Control (MAC)
- Ethernet protocol
- Forwarding in an Ethernet switched network
- A lot of interesting things happening at L2
 - MPLS, VLAN
 - Data center networks
 - ..