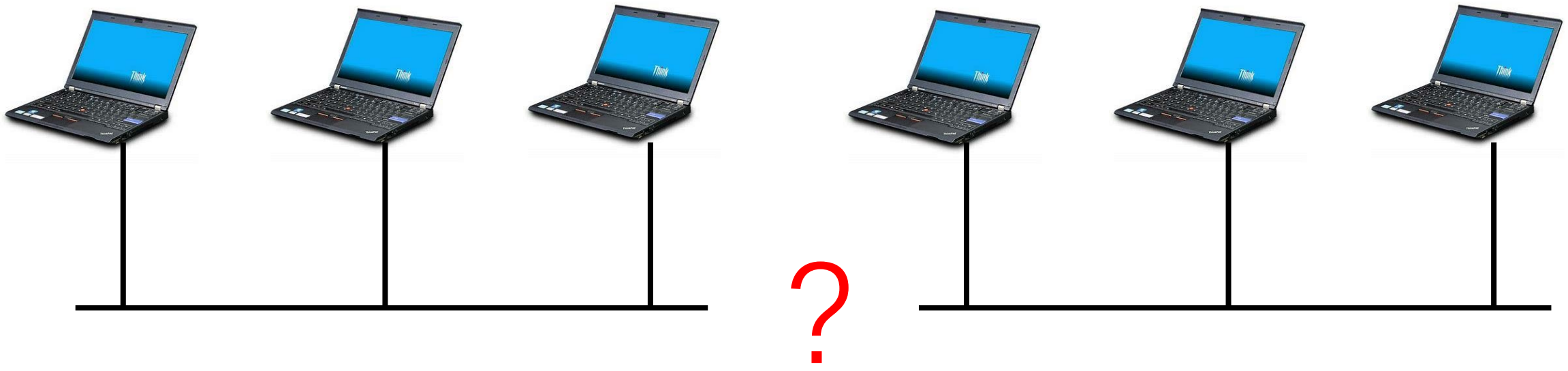
CS120: Computer Networks

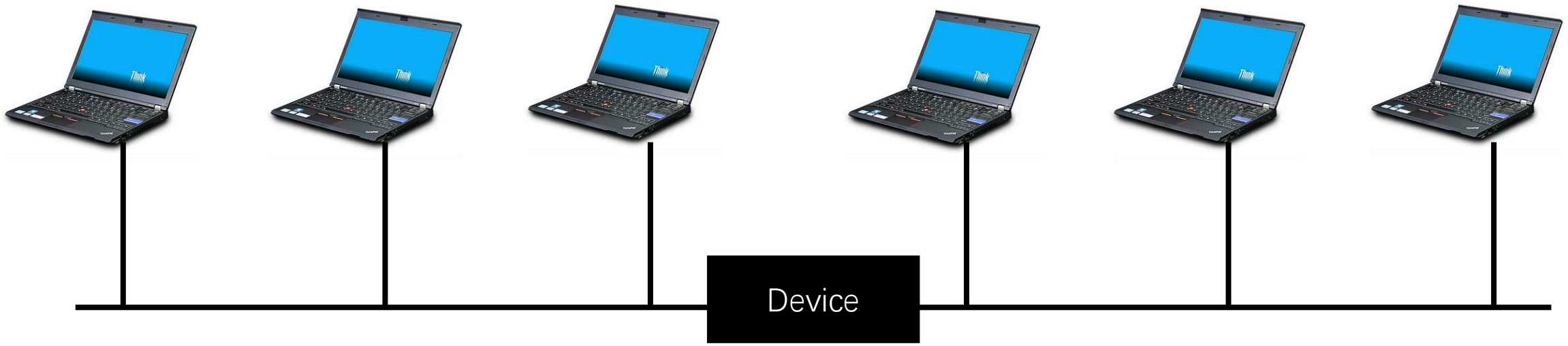
Lecture 8. Switching

Zhice Yang

How to Extend the Ethernet ?



How to Extend the Ethernet ?



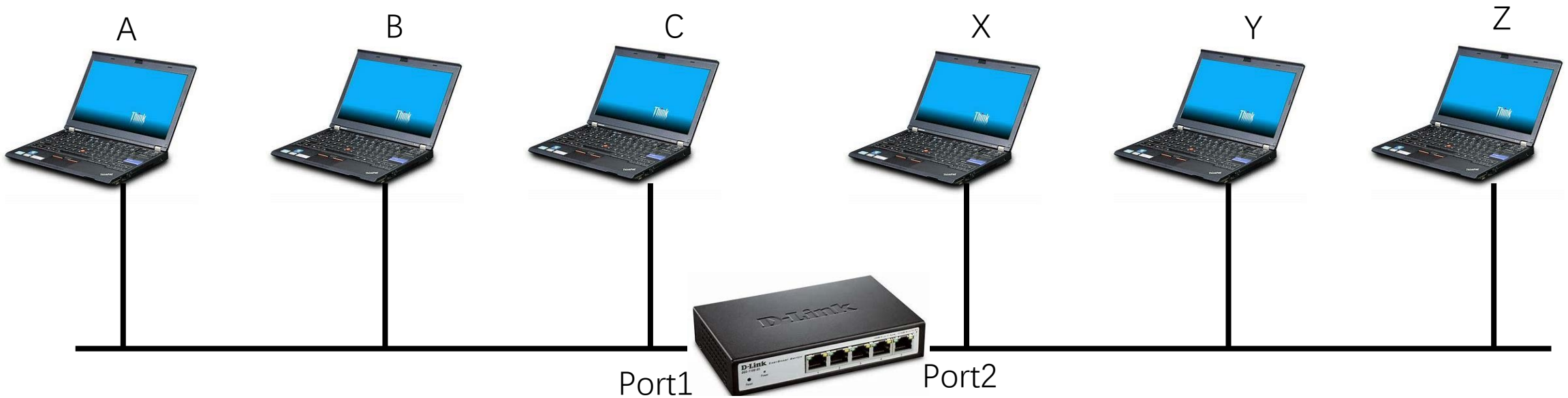
Ethernet Bridge/Switch

Store and Forward

How to Extend the Ethernet ?

- Simplest Strategy
  - Accept LAN frames on inputs and forward them out to **all** other outputs
- Better Strategy: learning Bridge
  - Observation: No need to forward frames to all outputs
  - Forwarding Table

How to Extend the Ethernet ?



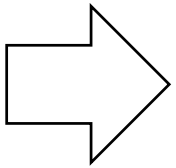
Ethernet Bridge/Switch

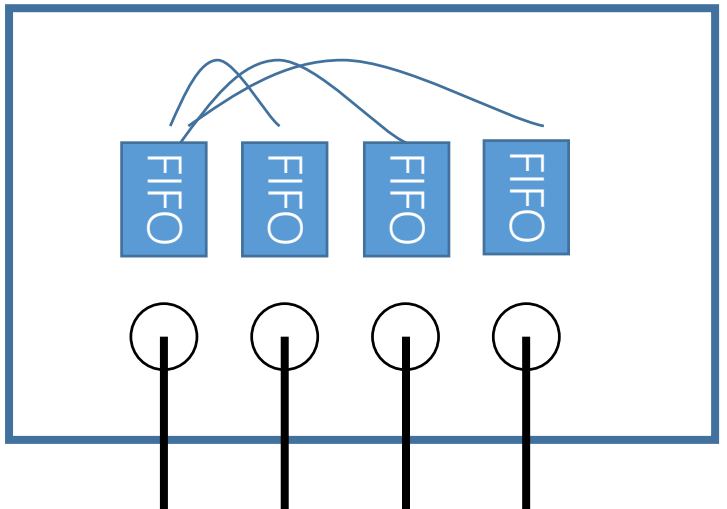
Forward

Host	Port
A	1
B	1
C	1
X	2
Y	2
Z	2

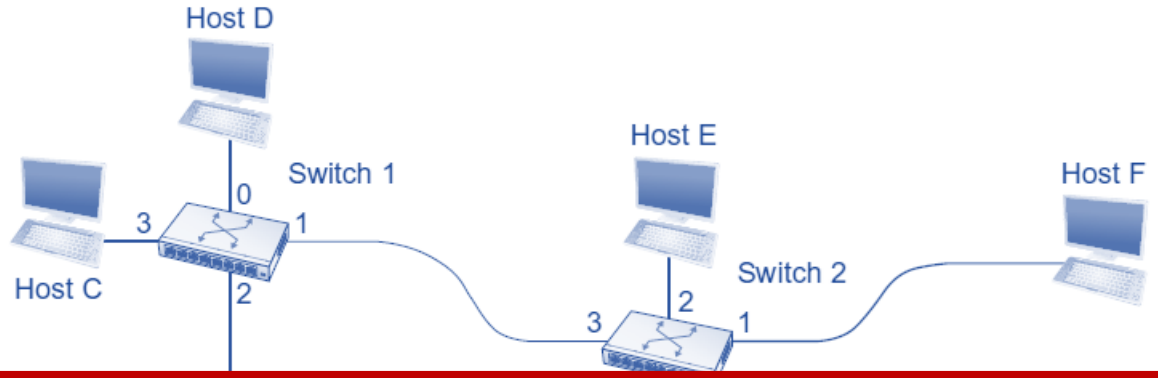
Switch

- A multi-input, multi-output device
  - Function: transfers packets from an input to one or more outputs
  - Ports can be connected to hosts
  - Ports can be connected to other switches
  - Performance: more ports in use => higher network throughput
- A device to form Ethernet to a large network





Larger Network with Switches

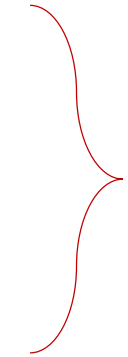


How to Find the Right Path to the Destination
-> Routing

How to Guide the Packet in the Right Path
-> Switching

Switching Methods

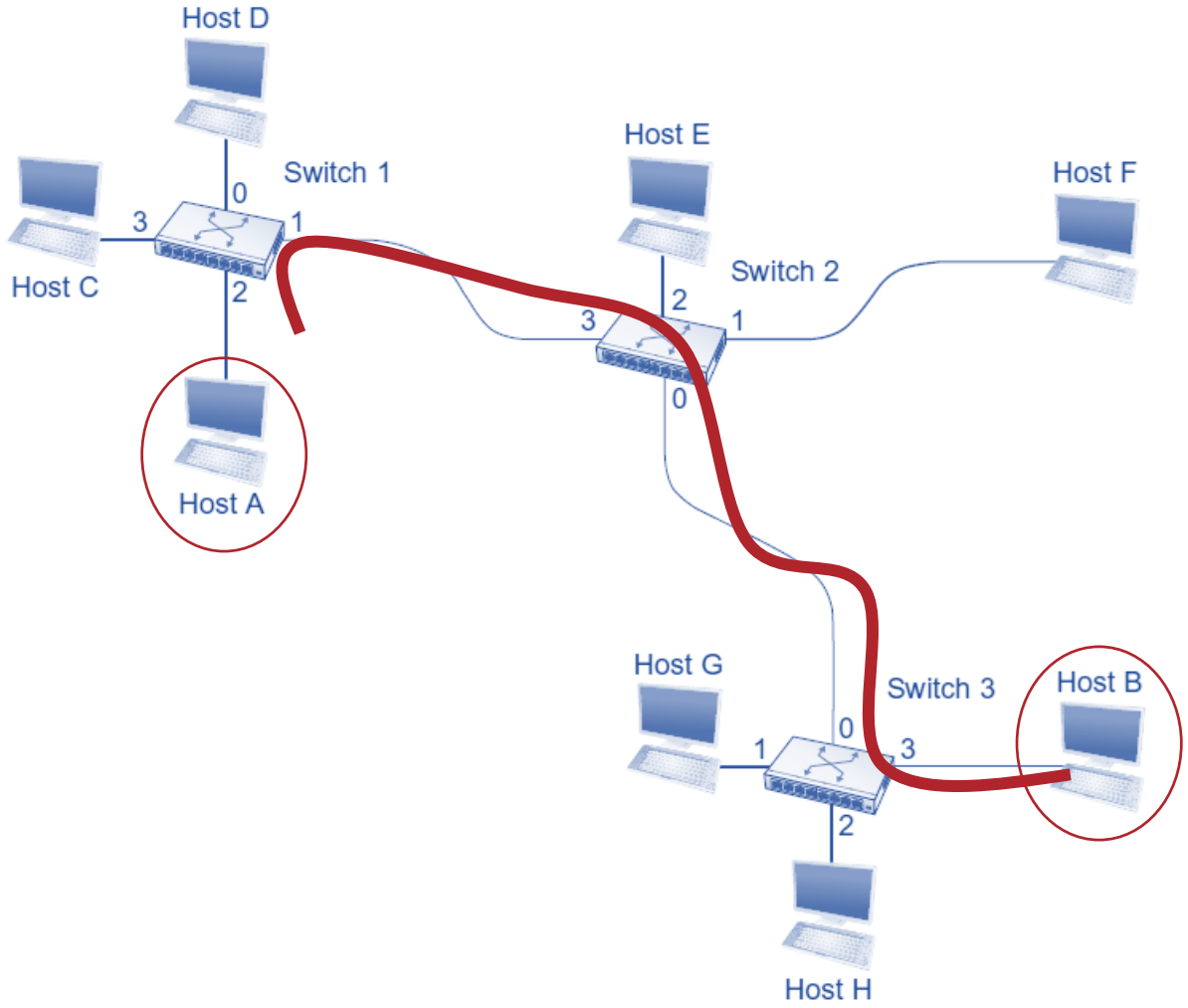
- Datagram/Connectionless
  - e.g. IP
- Virtual Circuit(VC)/Connection
  - e.g. X.25, ATM
- Source Routing



Router Determines the Path

Source Host Determines the Path

Datagram



Forwarding Table

Switch1	
Dest	Port
A	2
B	1
C	3
D	0
E	1
F	1
G	1
H	1

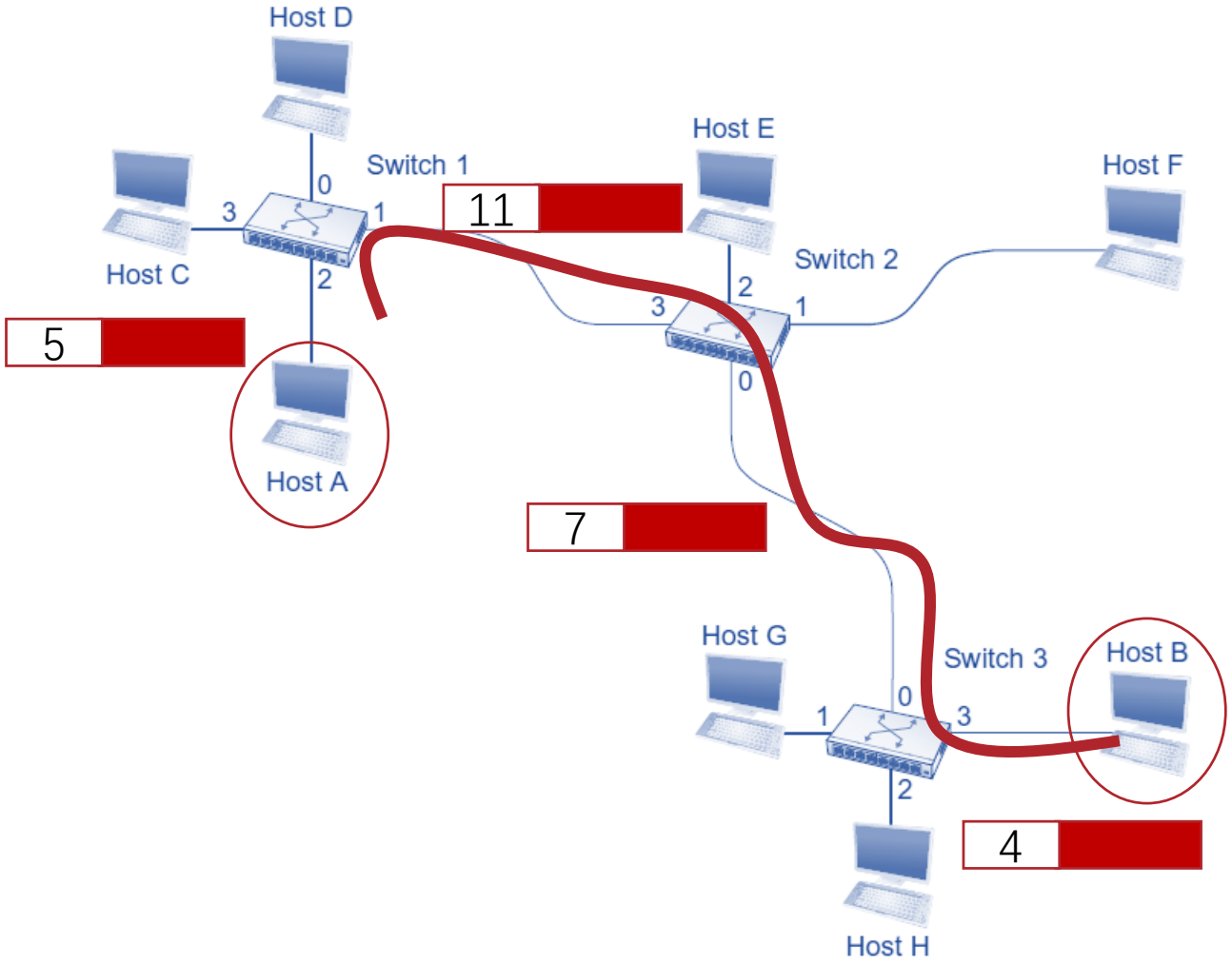
Switch2	
Dest	Port
A	3
B	0
C	3
D	3
E	2
F	1
G	0
H	0

Switch3	
Dest	Port
A	0
B	3
C	0
D	0
E	0
F	0
G	1
H	2

Datagram

- Elastic Service
  - Send at any time
- No Guarantee for
  - Success delivery
  - Performance
    - Delay, Throughput
  - Packet Order

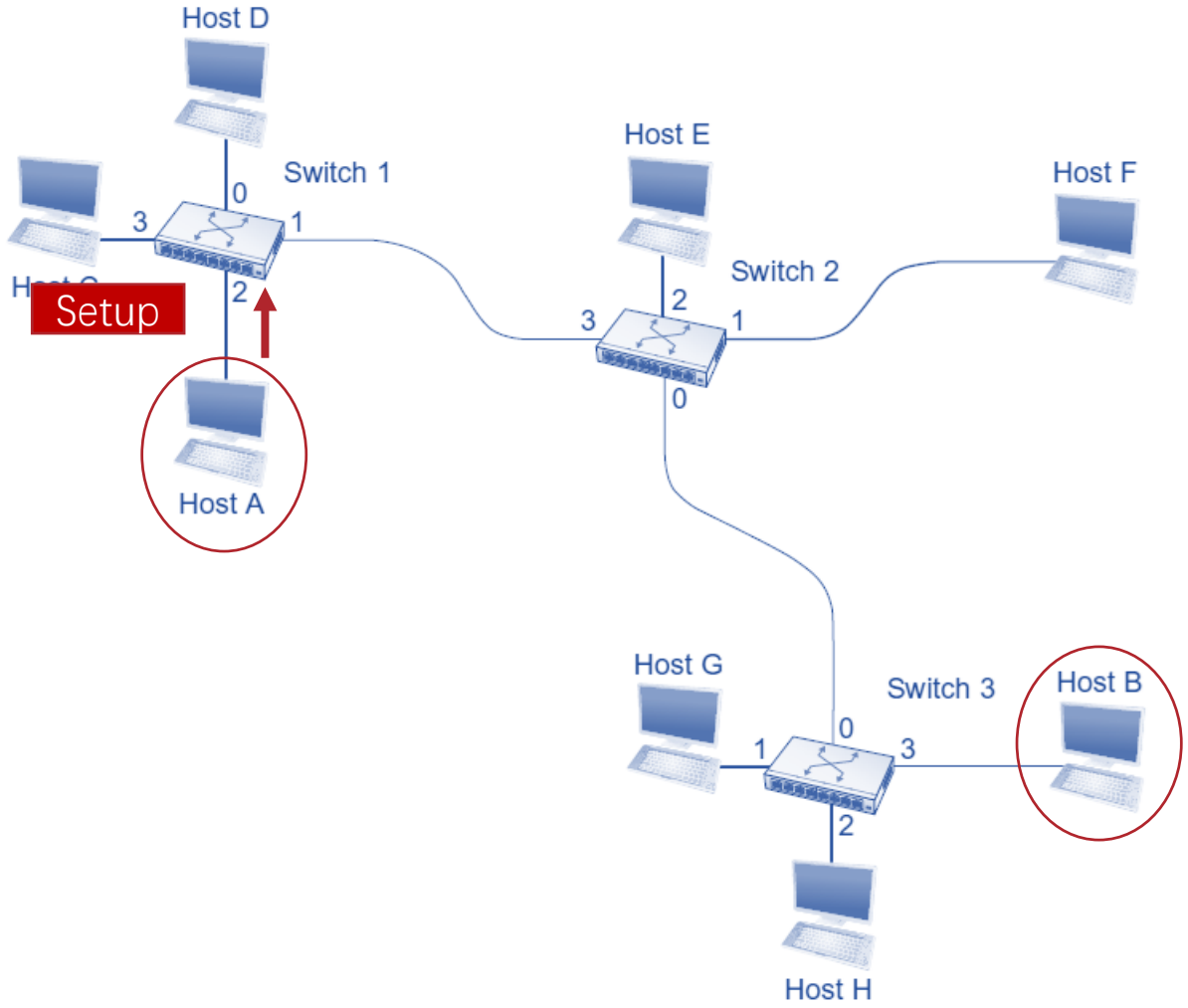
Virtual Circuit



Virtual Circuit Table

Switch1			
Incoming Interface	Incoming VCI	Outgoing Interface	Outgoing VCI
2	5	1	11
Switch2			
Incoming Interface	Incoming VCI	Outgoing Interface	Outgoing VCI
3	11	0	7
Switch3			
Incoming Interface	Incoming VCI	Outgoing Interface	Outgoing VCI
0	7	3	4

Virtual Circuit



Virtual Circuit Table

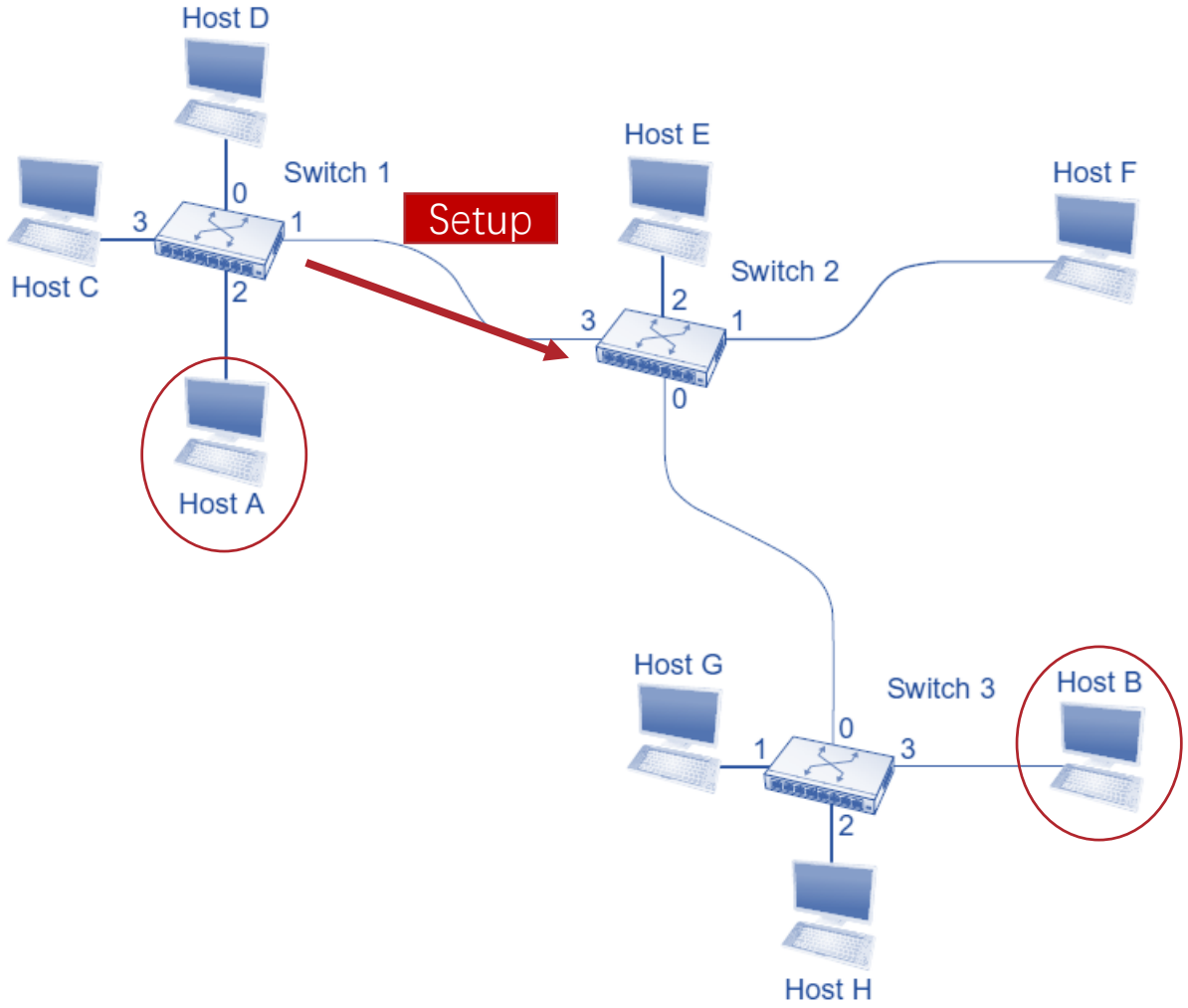
Switch1			
Incoming Interface	Incoming VCI	Outgoing Interface	Outgoing VCI
2	5		

Switch2			
Incoming Interface	Incoming VCI	Outgoing Interface	Outgoing VCI

Switch3			
Incoming Interface	Incoming VCI	Outgoing Interface	Outgoing VCI

Host A		Host B	
Destination	Outgoing VCI	Source	Incoming VCI

Virtual Circuit



Virtual Circuit Table

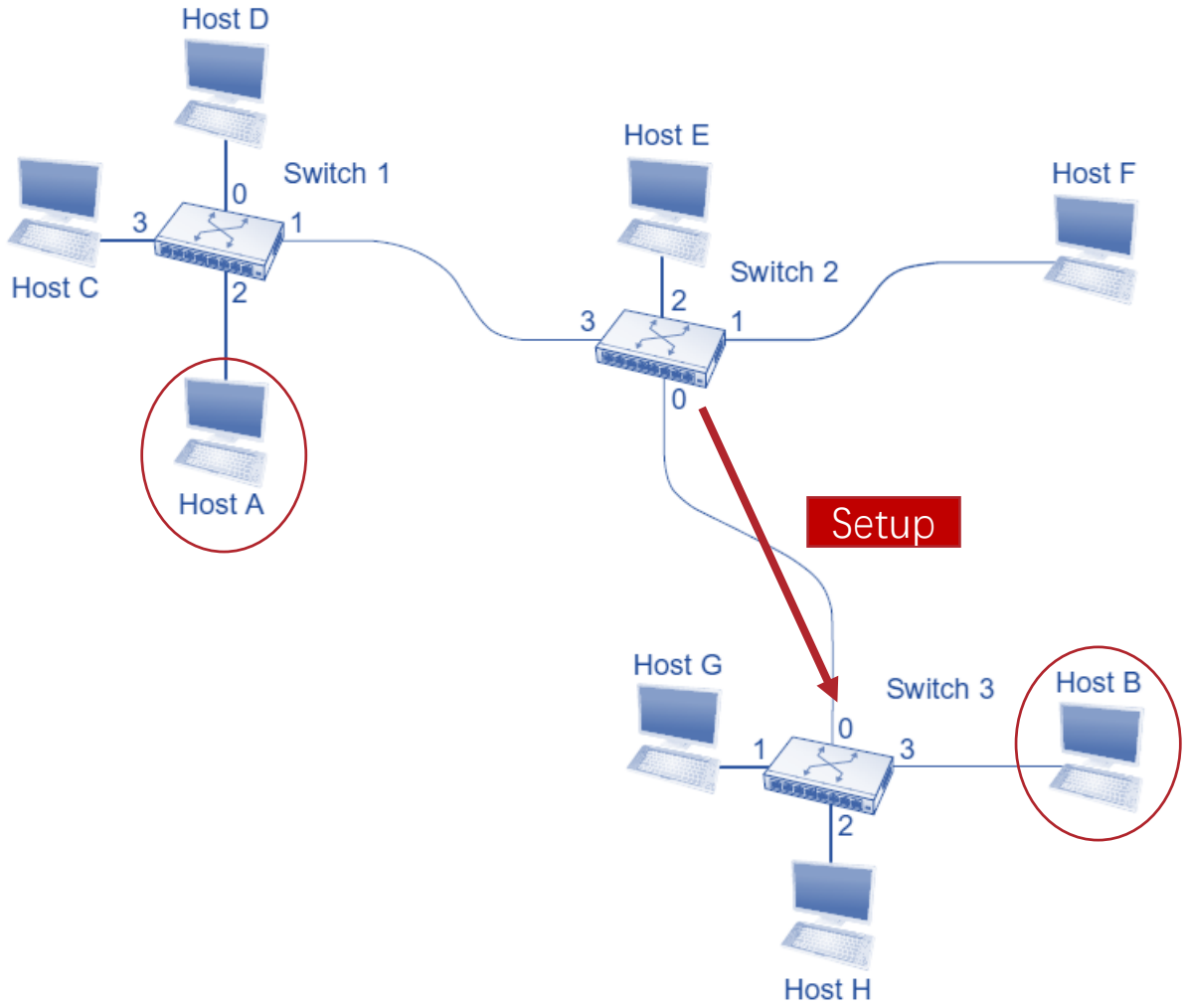
Switch1			
Incoming Interface	Incoming VCI	Outgoing Interface	Outgoing VCI
2	5		

Switch2			
Incoming Interface	Incoming VCI	Outgoing Interface	Outgoing VCI
3	11		

Switch3			
Incoming Interface	Incoming VCI	Outgoing Interface	Outgoing VCI

Host A		Host B	
Destination	Outgoing VCI	Source	Incoming VCI

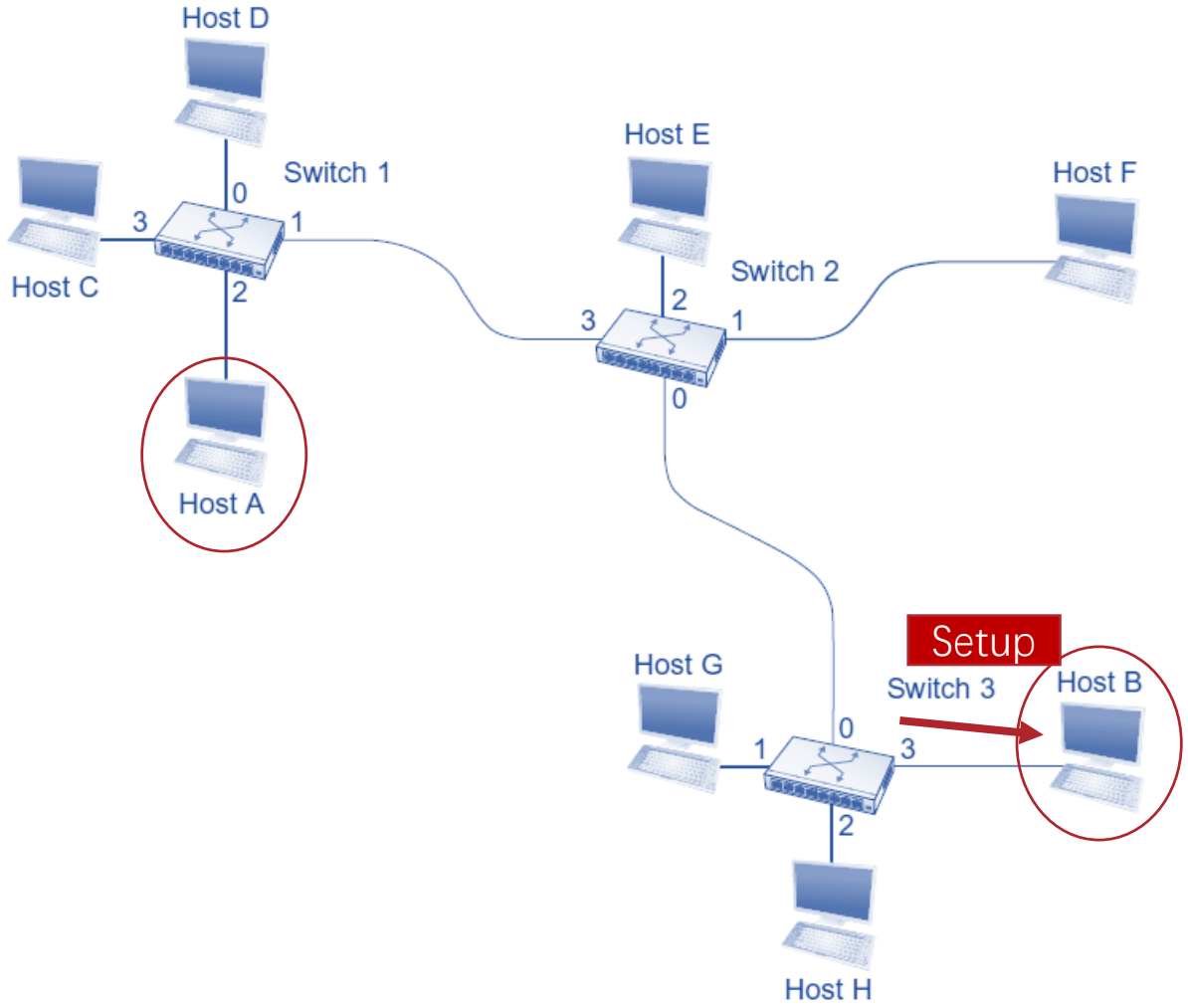
Virtual Circuit



Virtual Circuit Table

Switch1			
Incoming Interface	Incoming VCI	Outgoing Interface	Outgoing VCI
2	5		
Switch2			
Incoming Interface	Incoming VCI	Outgoing Interface	Outgoing VCI
3	11		
Switch3			
Incoming Interface	Incoming VCI	Outgoing Interface	Outgoing VCI
0	7		
Host A		**Host B**	
Destination	Outgoing VCI	Source	Incoming VCI

Virtual Circuit



Virtual Circuit Table

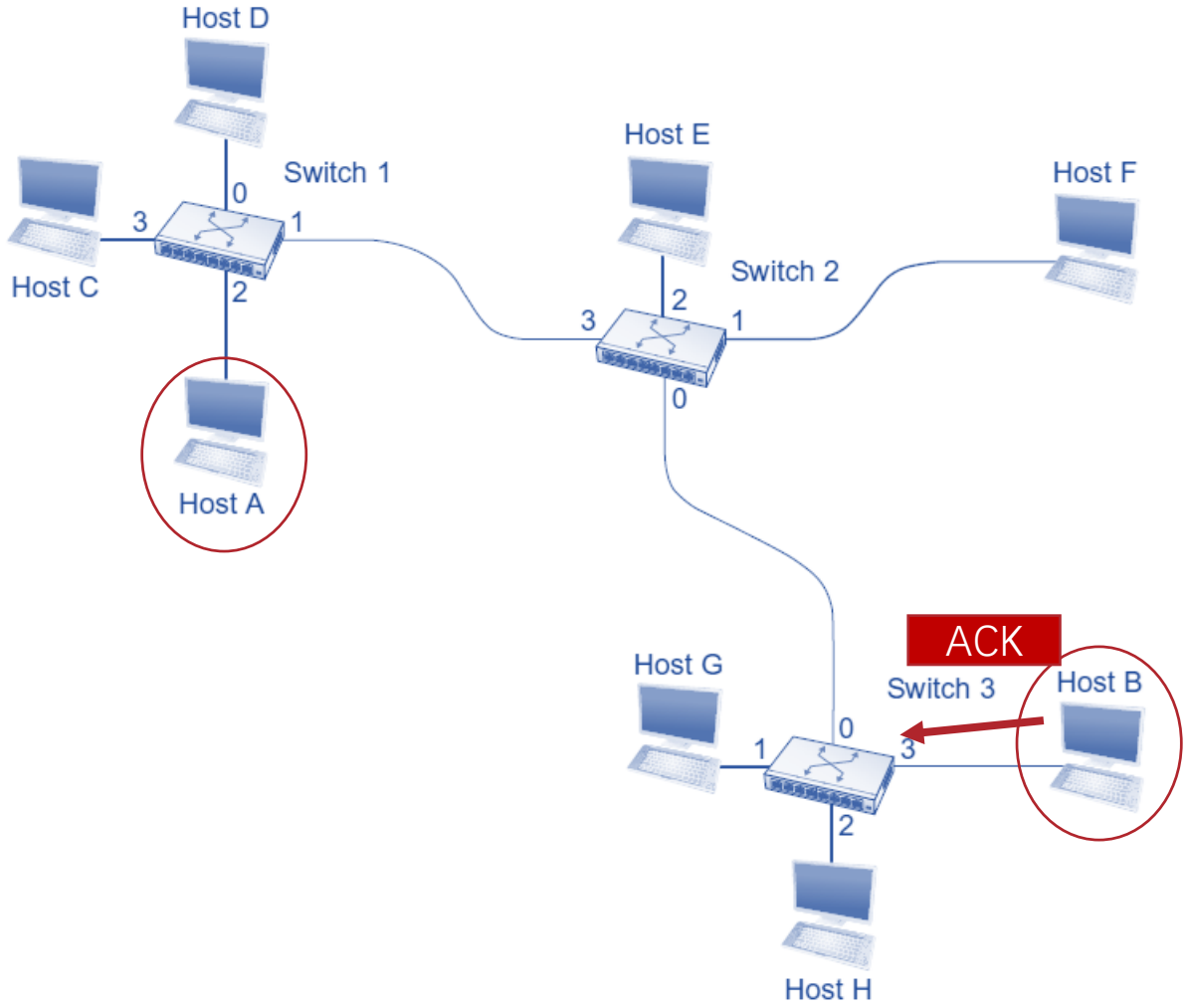
Switch1			
Incoming Interface	Incoming VCI	Outgoing Interface	Outgoing VCI
2	5		

Switch2			
Incoming Interface	Incoming VCI	Outgoing Interface	Outgoing VCI
3	11		

Switch3			
Incoming Interface	Incoming VCI	Outgoing Interface	Outgoing VCI
0	7		

Host A		Host B	
Destination	Outgoing VCI	Source	Incoming VCI
		From A	4

Virtual Circuit



Virtual Circuit Table

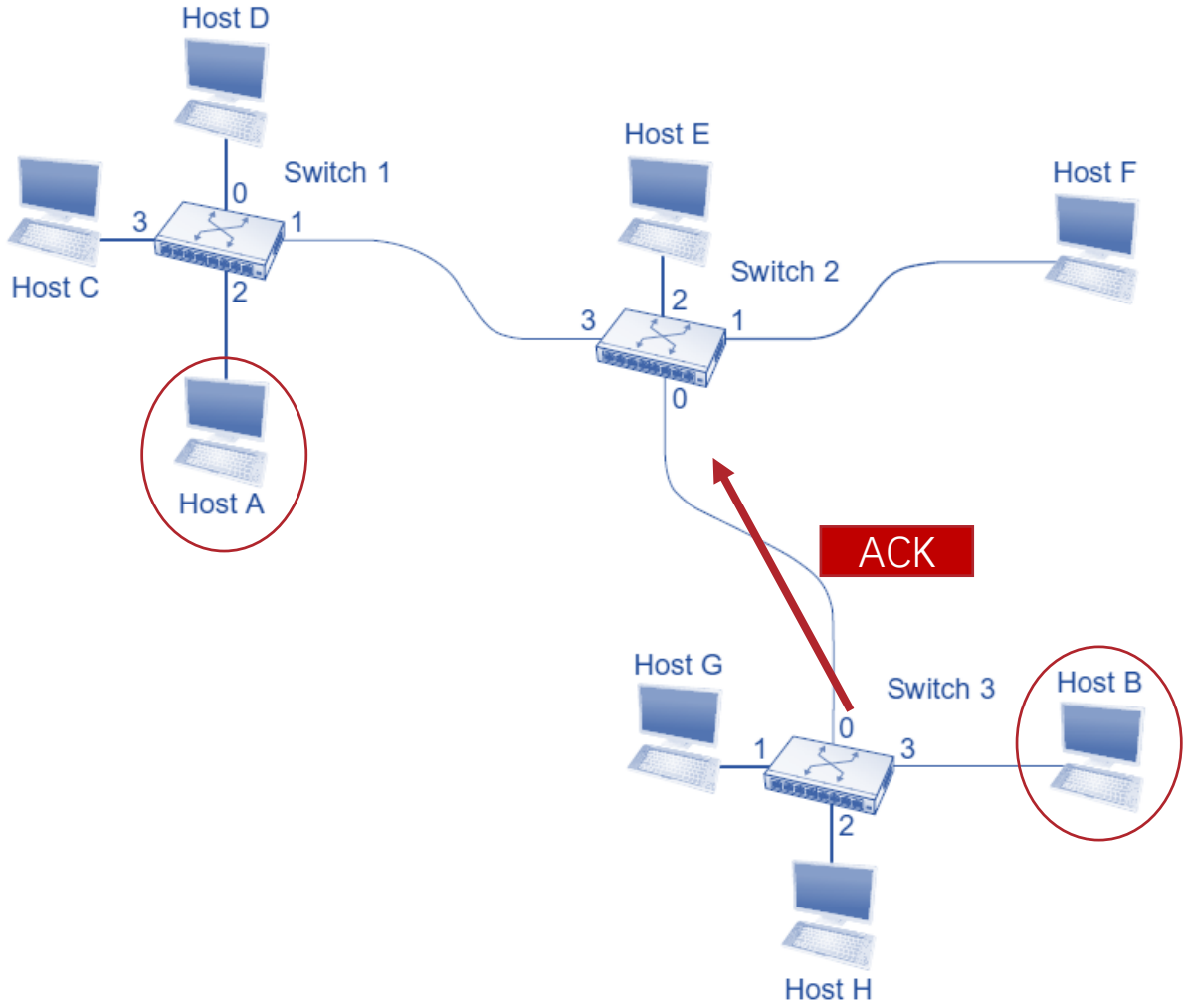
Switch1			
Incoming Interface	Incoming VCI	Outgoing Interface	Outgoing VCI
2	5		

Switch2			
Incoming Interface	Incoming VCI	Outgoing Interface	Outgoing VCI
3	11		

Switch3			
Incoming Interface	Incoming VCI	Outgoing Interface	Outgoing VCI
0	7	3	4

Host A		Host B	
Destination	Outgoing VCI	Source	Incoming VCI
		From A	4

Virtual Circuit



Virtual Circuit Table

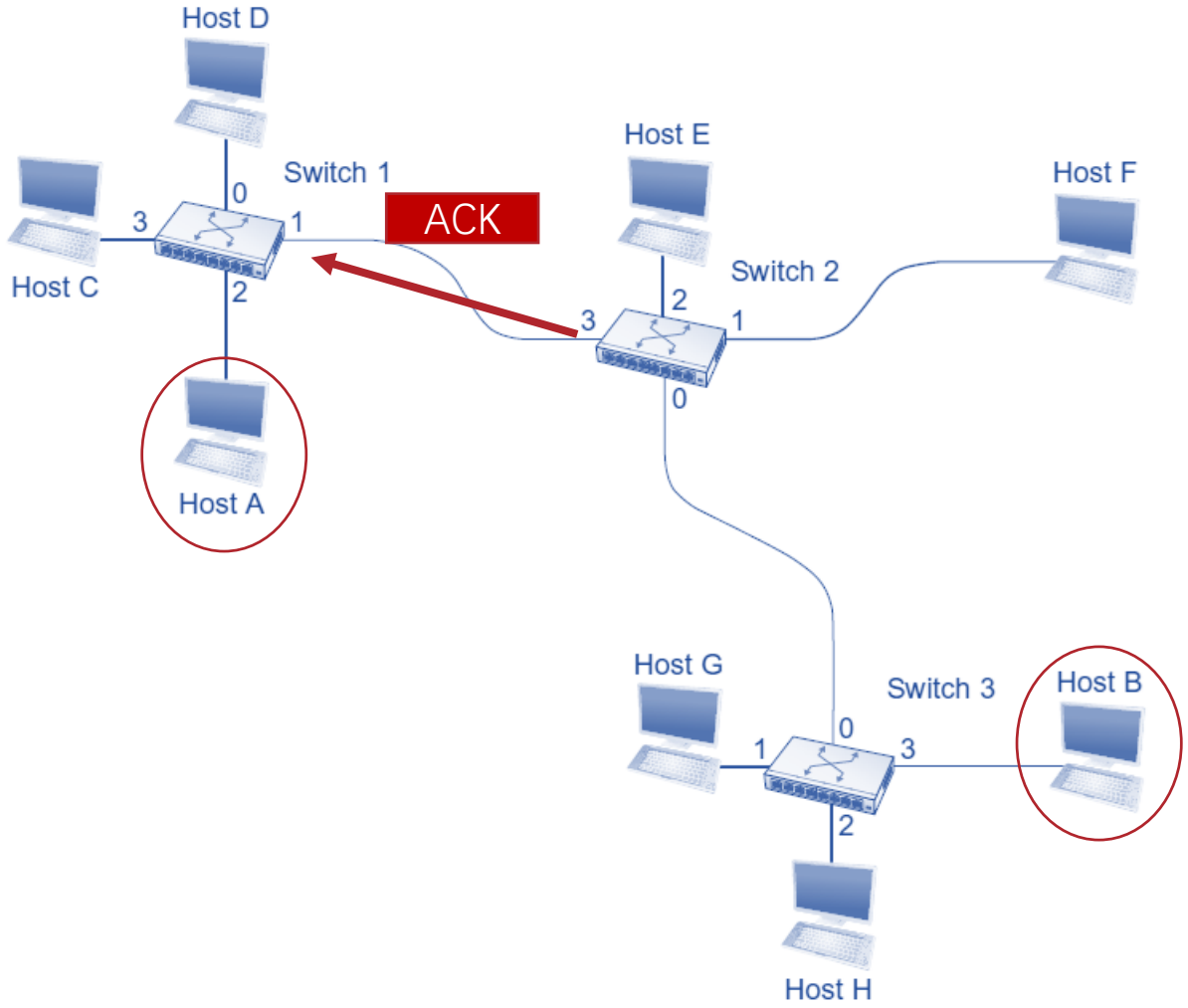
Switch1			
Incoming Interface	Incoming VCI	Outgoing Interface	Outgoing VCI
2	5		

Switch2			
Incoming Interface	Incoming VCI	Outgoing Interface	Outgoing VCI
3	11	0	7

Switch3			
Incoming Interface	Incoming VCI	Outgoing Interface	Outgoing VCI
0	7	3	4

Host A		Host B	
Destination	Outgoing VCI	Source	Incoming VCI
		From A	4

Virtual Circuit



Virtual Circuit Table

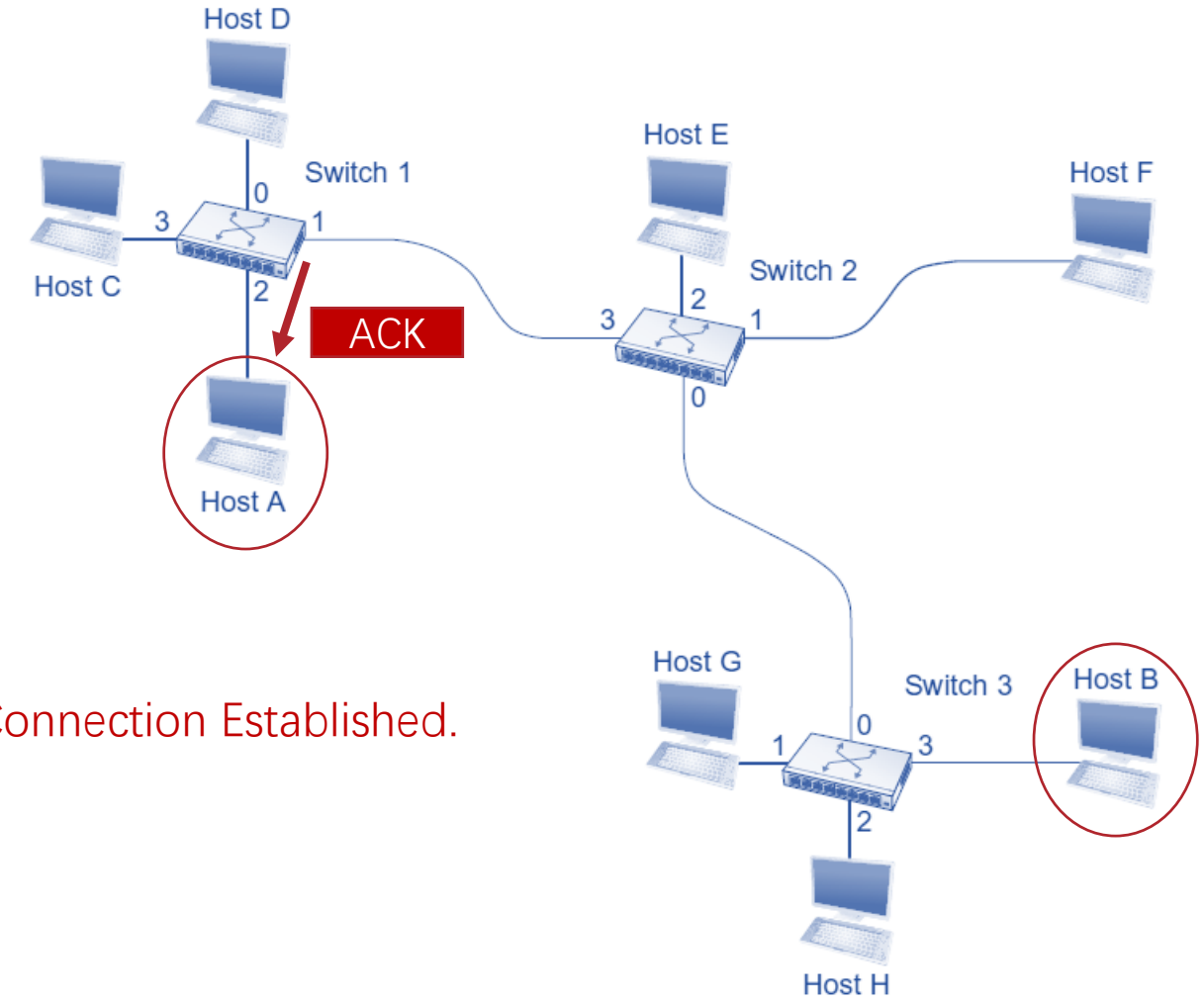
Switch1			
Incoming Interface	Incoming VCI	Outgoing Interface	Outgoing VCI
2	5	1	11

Switch2			
Incoming Interface	Incoming VCI	Outgoing Interface	Outgoing VCI
3	11	0	7

Switch3			
Incoming Interface	Incoming VCI	Outgoing Interface	Outgoing VCI
0	7	3	4

Host A		Host B	
Destination	Outgoing VCI	Source	Incoming VCI
		From A	4

Virtual Circuit



Connection Established.

Virtual Circuit Table

Switch1			
Incoming Interface	Incoming VCI	Outgoing Interface	Outgoing VCI
2	5	1	11

Switch2			
Incoming Interface	Incoming VCI	Outgoing Interface	Outgoing VCI
3	11	0	7

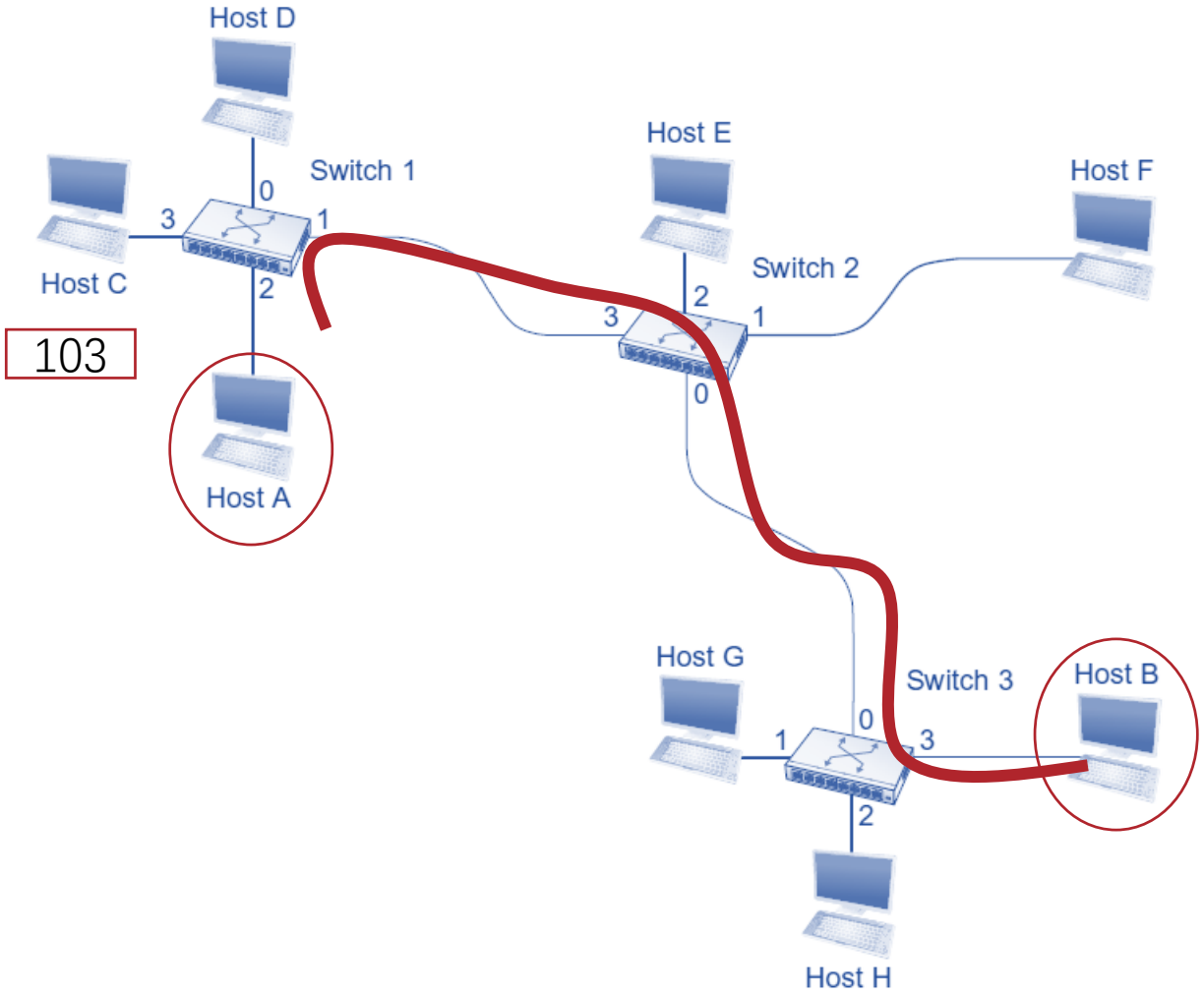
Switch3			
Incoming Interface	Incoming VCI	Outgoing Interface	Outgoing VCI
0	7	3	4

Host A		Host B	
Destination	Outgoing VCI	Source	Incoming VCI
To B	5	From A	4

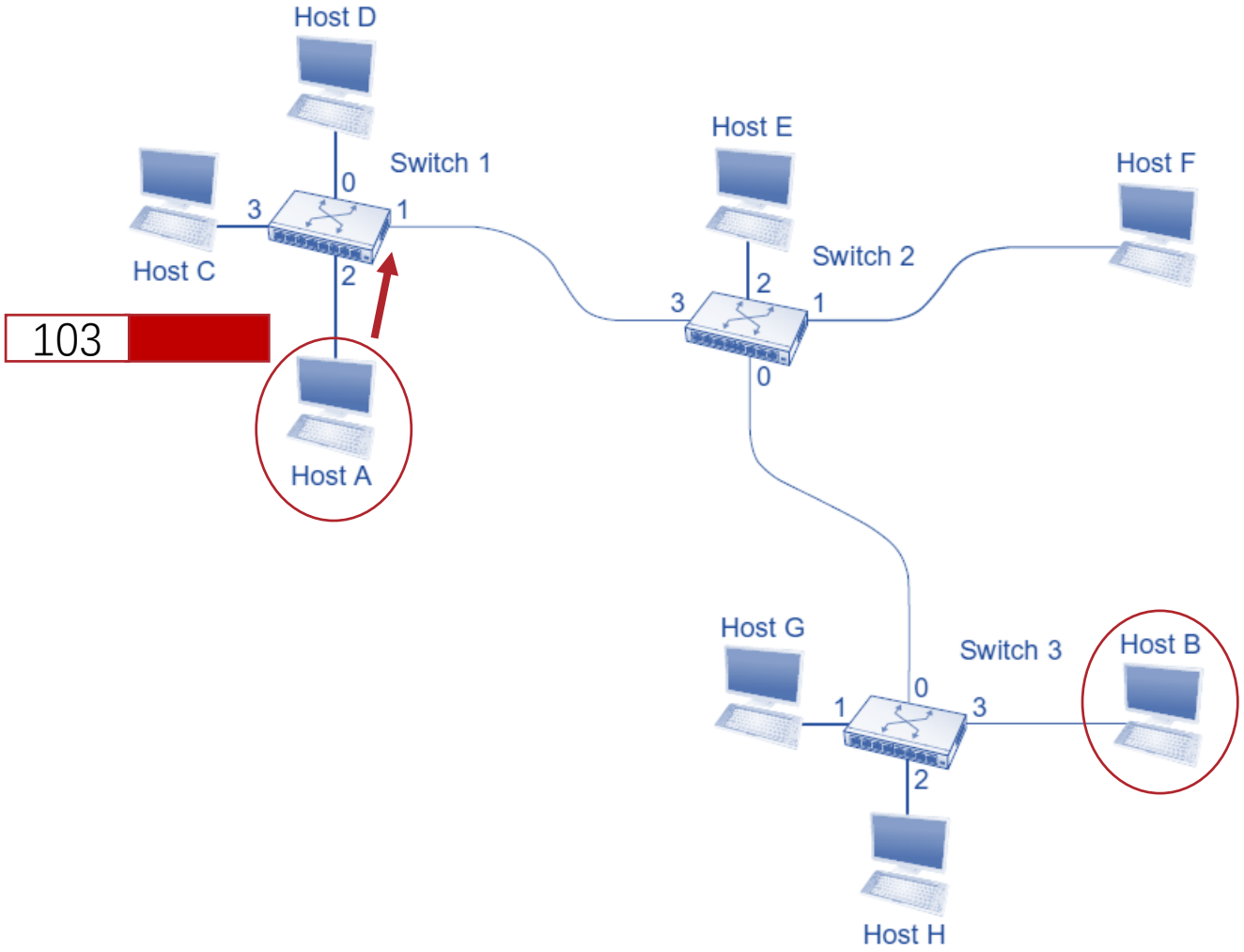
Virtual Circuit

- Reservation Service
  - Reserve Before Sending
- Guaranteed Service
  - Bitrate, Delay, etc.
  - Performance
    - Through reserving buffer, connection bandwidth, etc.

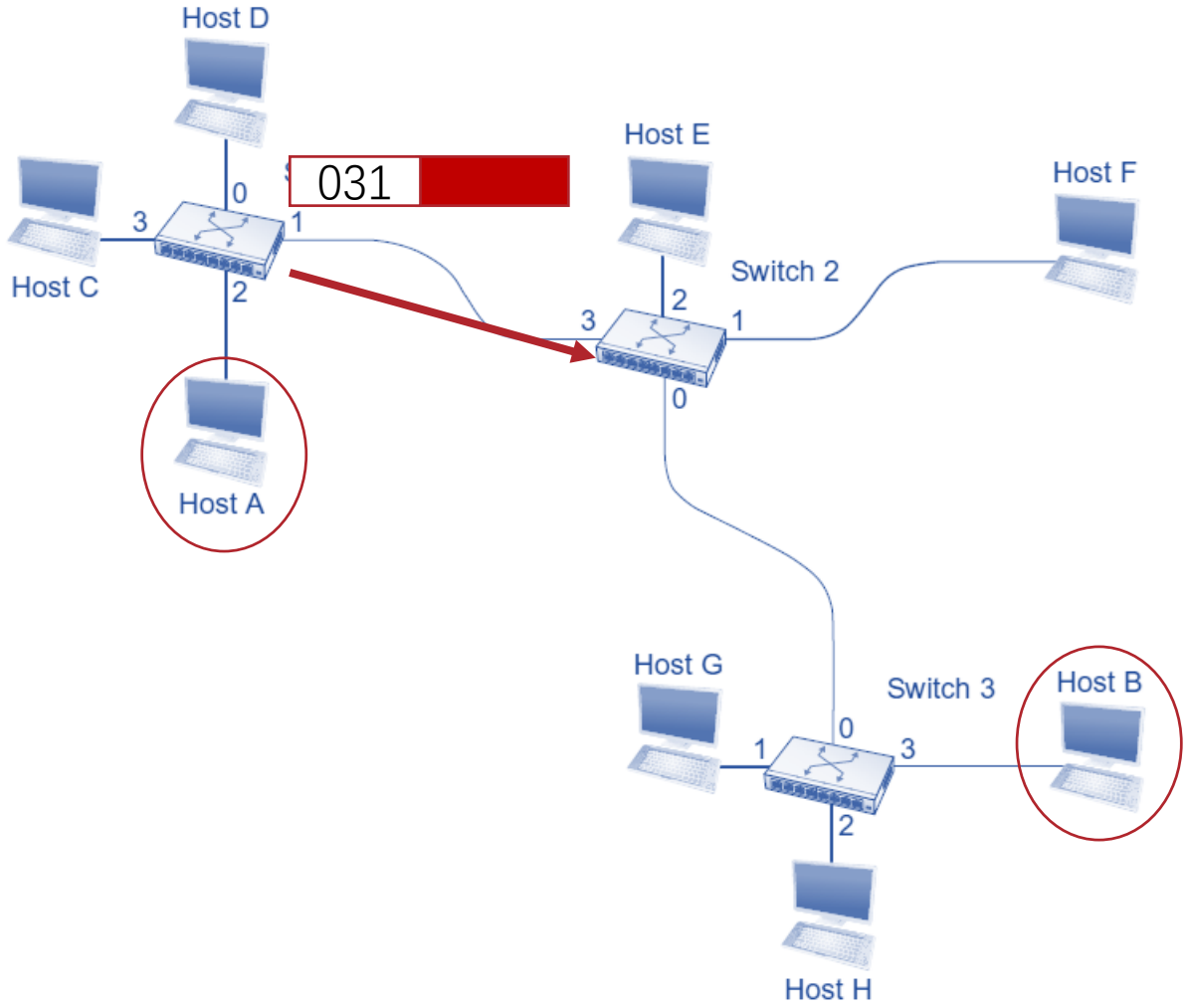
Source Routing



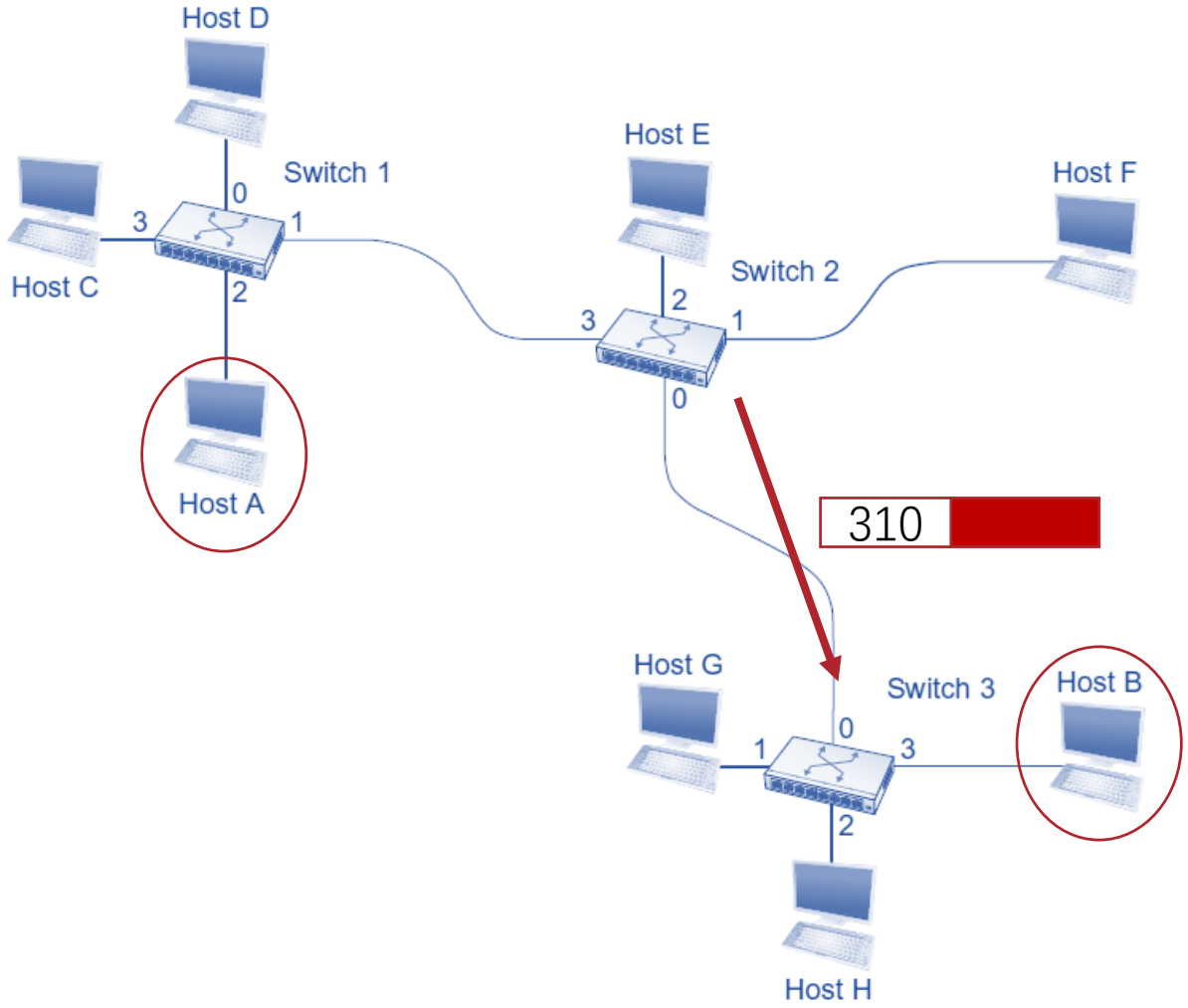
Source Routing



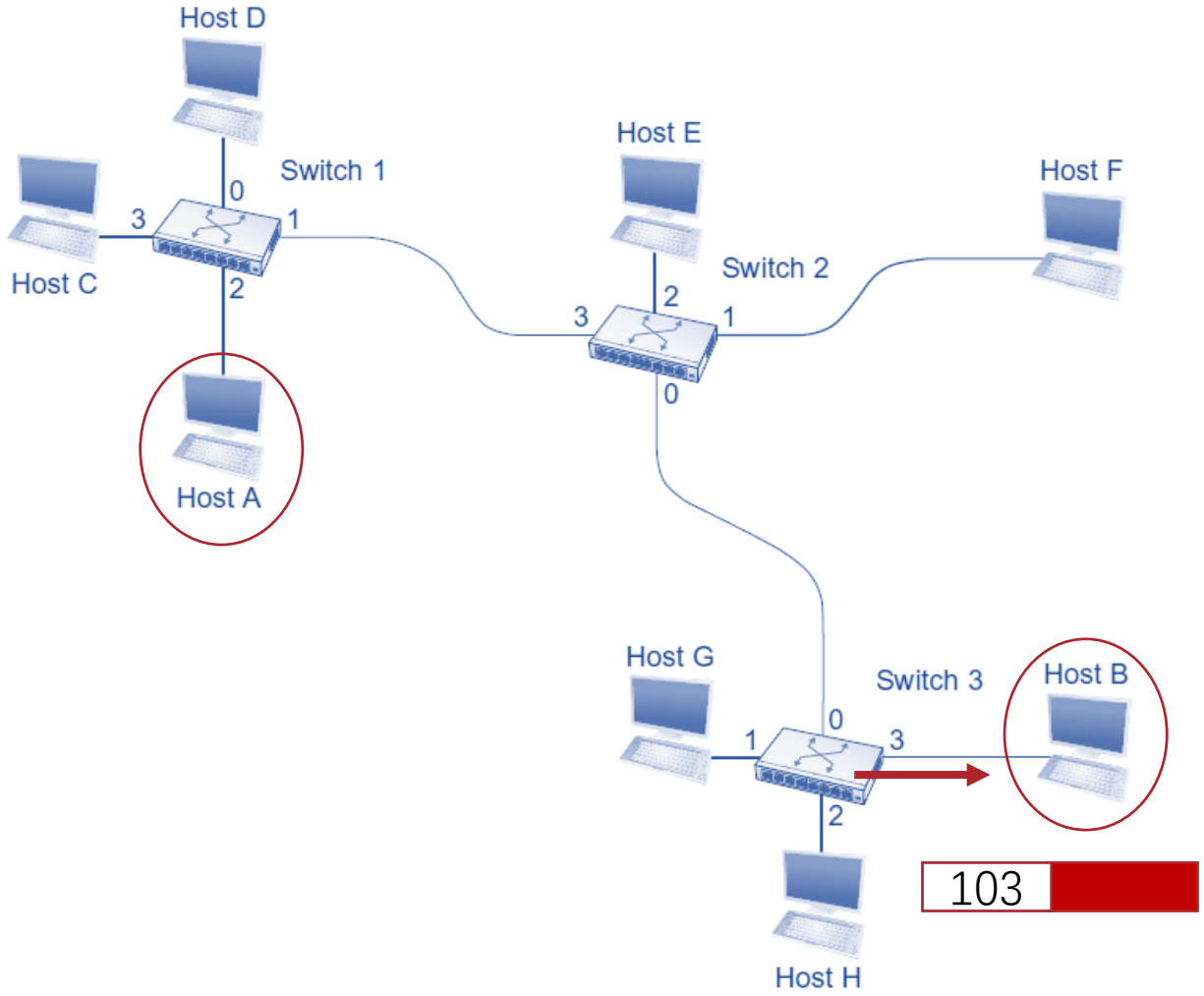
Source Routing



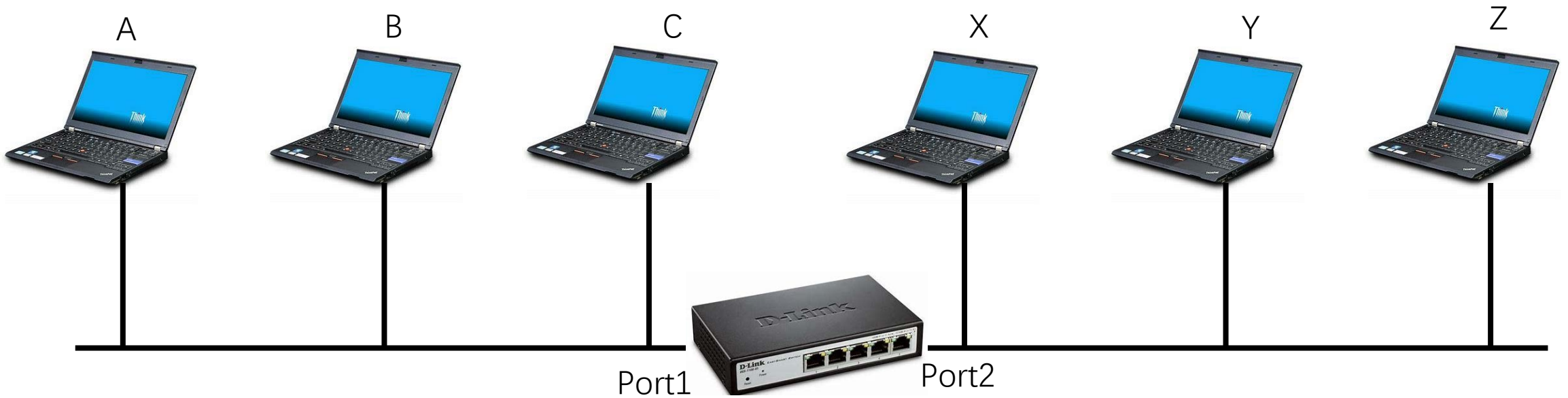
Source Routing



Source Routing



How to Extend the Ethernet ?

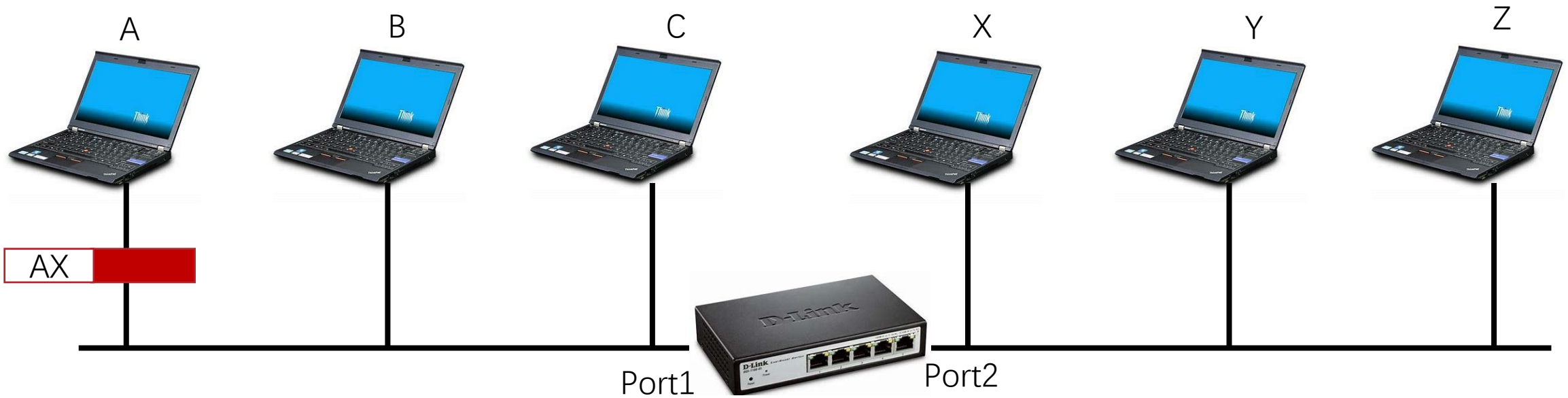


Ethernet Bridge/Switch

Host	Port
X	2
Y	2
Z	2

How to Obtain the Forwarding Table

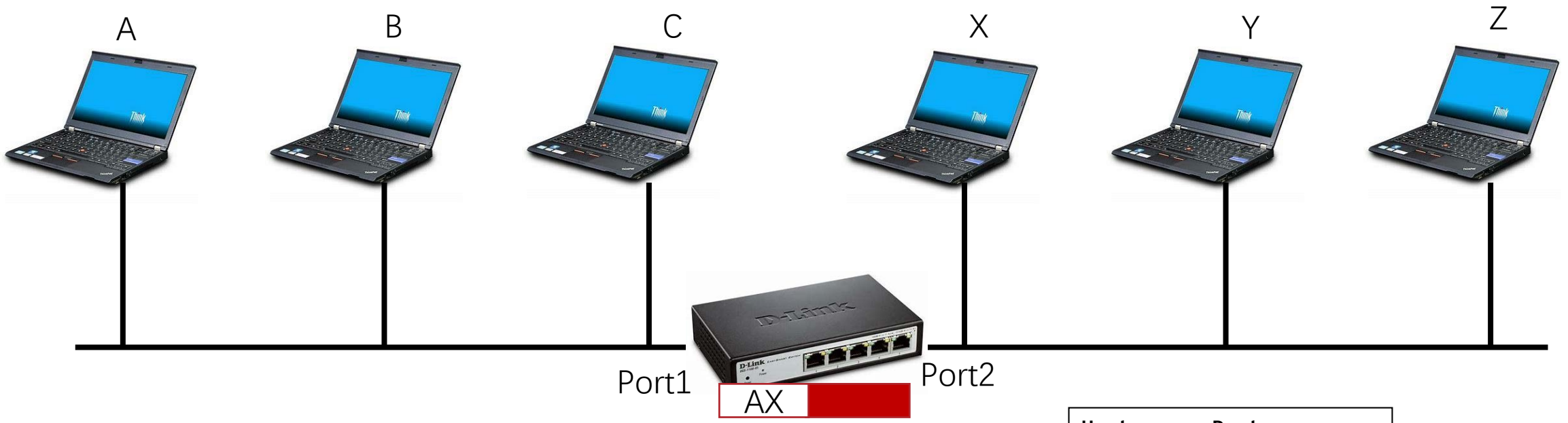
Learning Switch



Ethernet Bridge/Switch

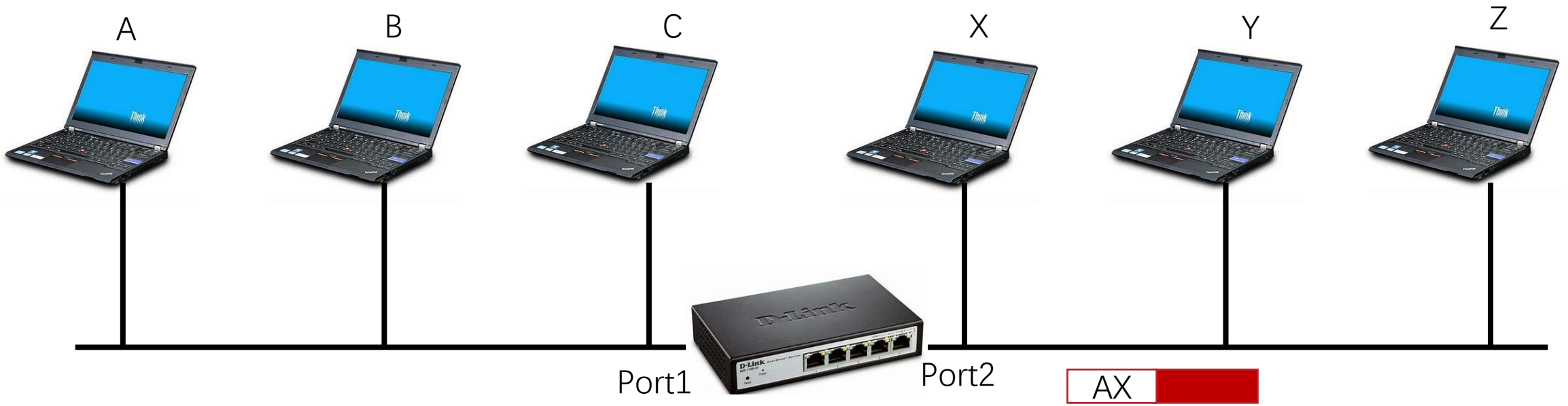
Host	Port
Null	Null

Learning Switch



Host	Port
A	1

Learning Switch

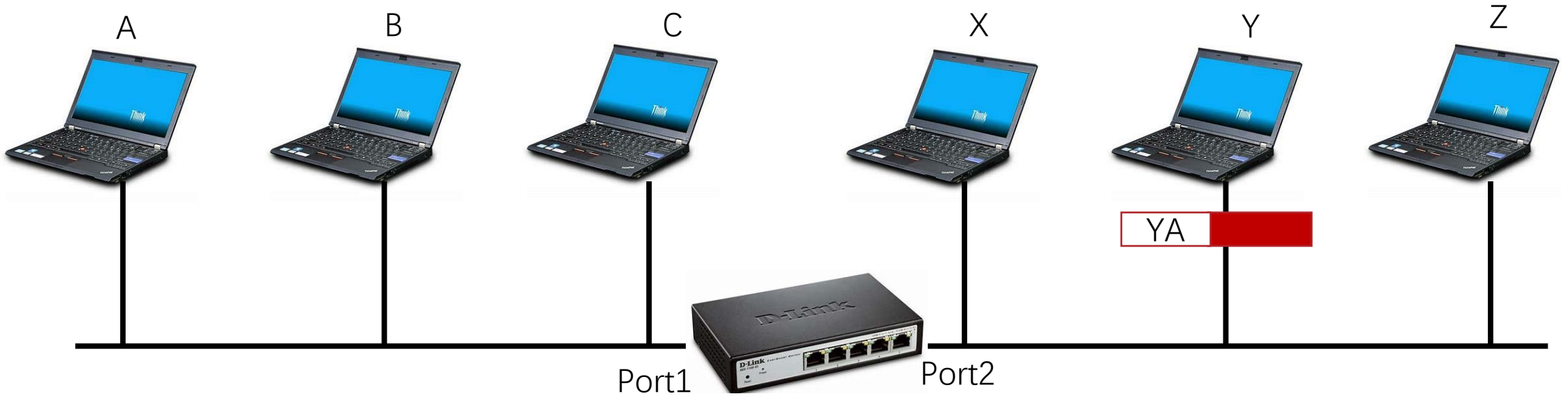


Ethernet Bridge/Switch

AX

Host	Port
A	1

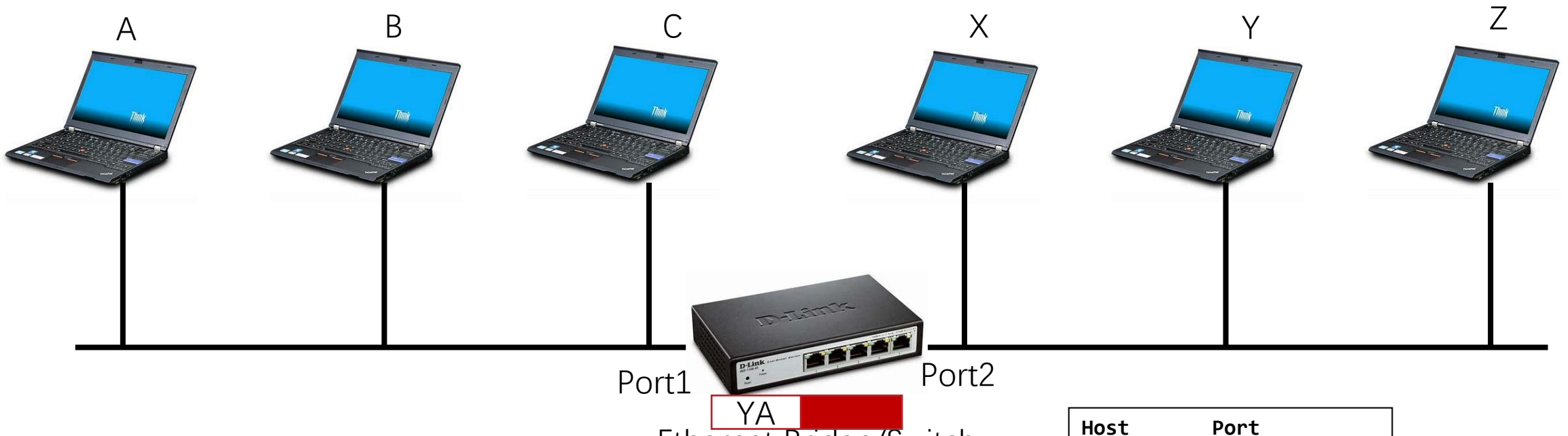
Learning Switch



Ethernet Bridge/Switch

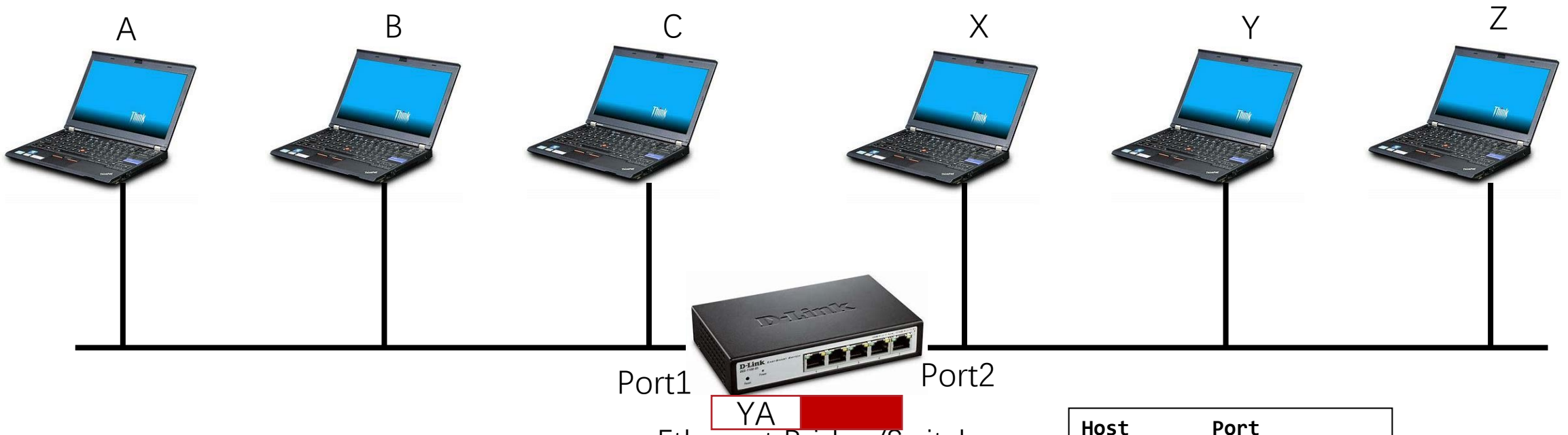
Host	Port
A	1

Learning Switch



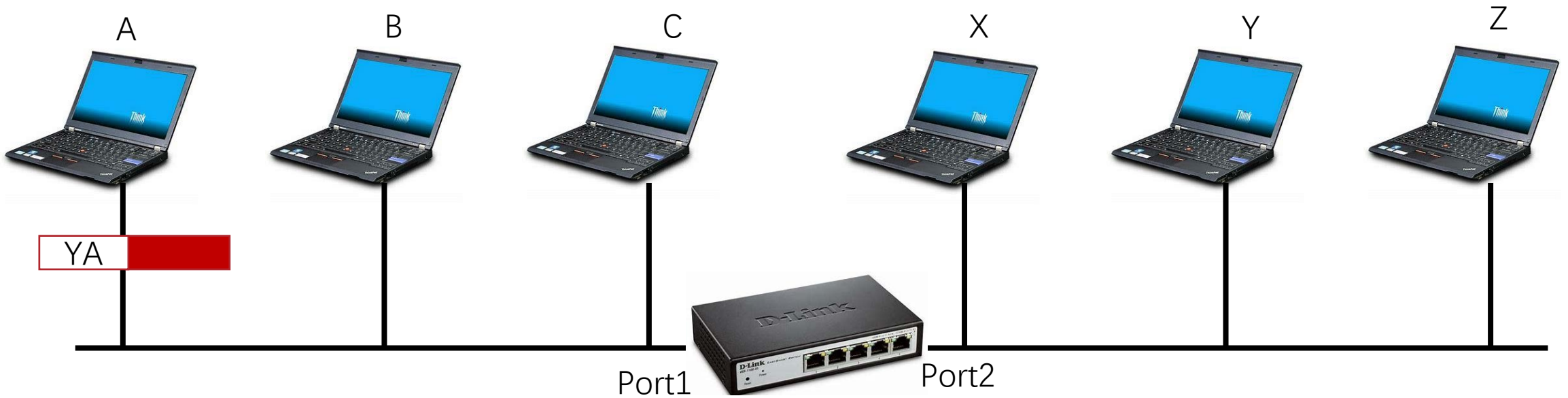
Host	Port
A	1
Y	2

Learning Switch



Host	Port
A	1
Y	2

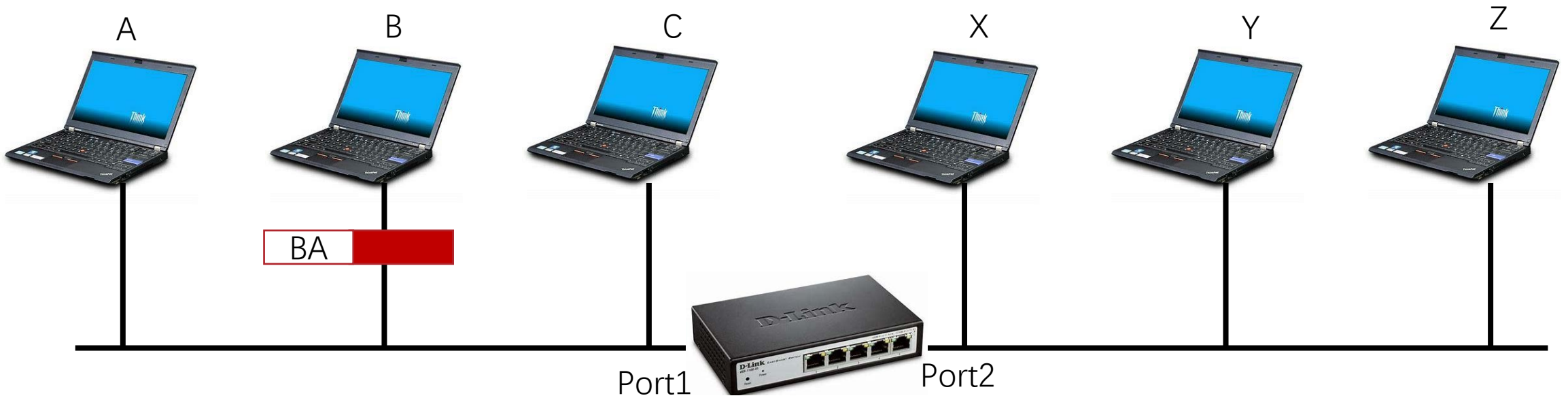
Learning Switch



Ethernet Bridge/Switch

Host	Port
A	1
Y	2

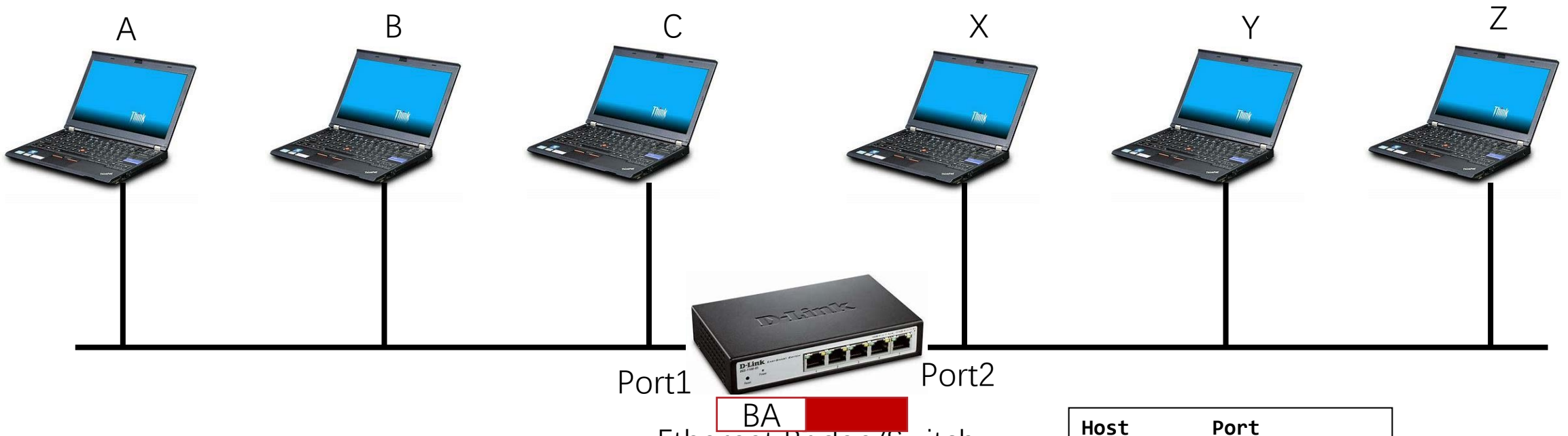
Learning Switch



Ethernet Bridge/Switch

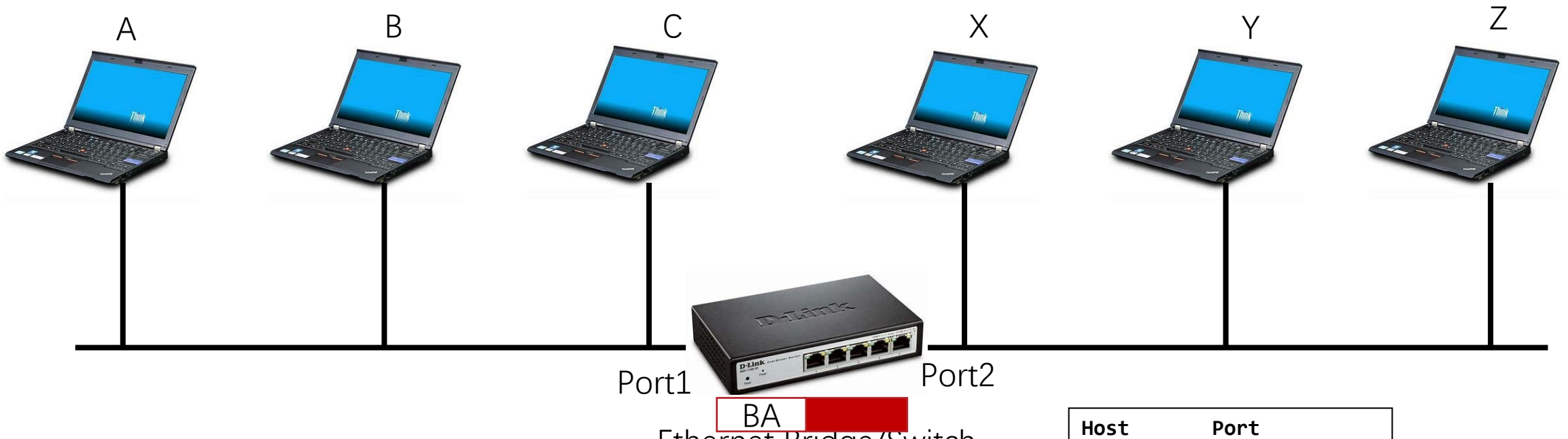
Host	Port
A	1
Y	2

Learning Switch



Host	Port
A	1
Y	2
B	1

Learning Switch



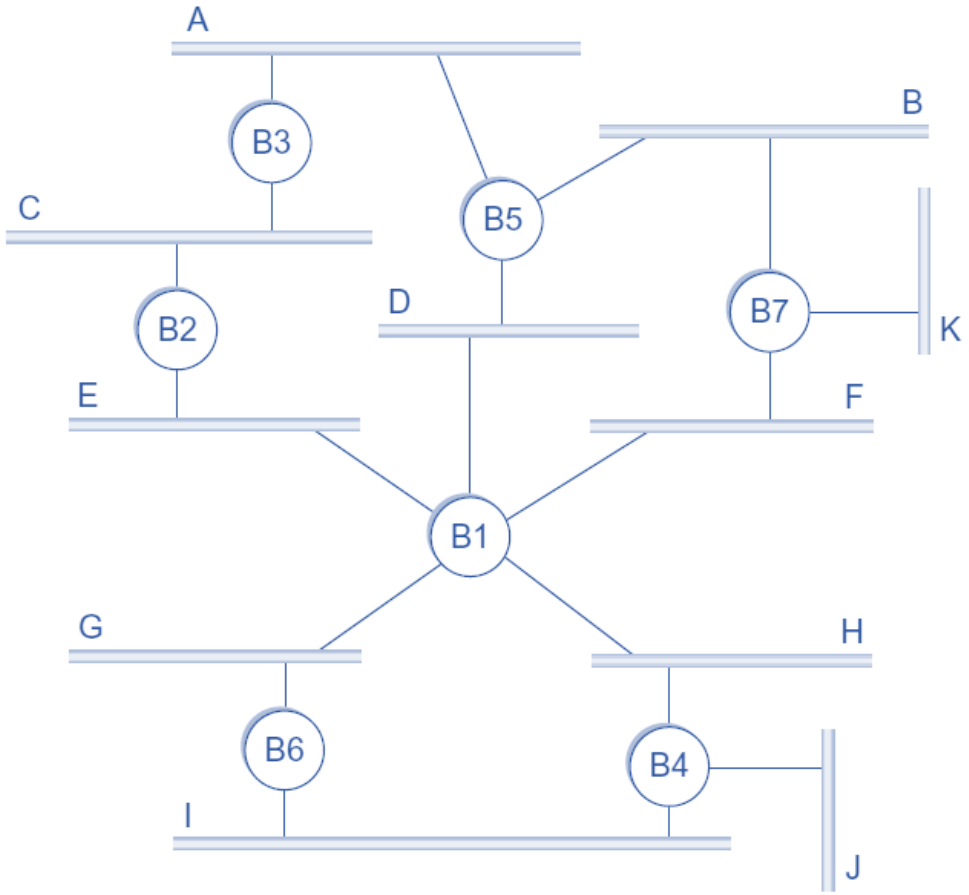
Host	Port
A	1
Y	2
B	1

Learning Switch

- When packet is received at switch
  - Record incoming port, source address
  - Index forwarding table using destination address
    - if destination exists
      - if destination on port from which packet arrived
        - drop
      - else
        - forward packet on port indicated by entry
    - else
      - forward on all ports except arriving port

Network with Switches

- A network of Ethernet

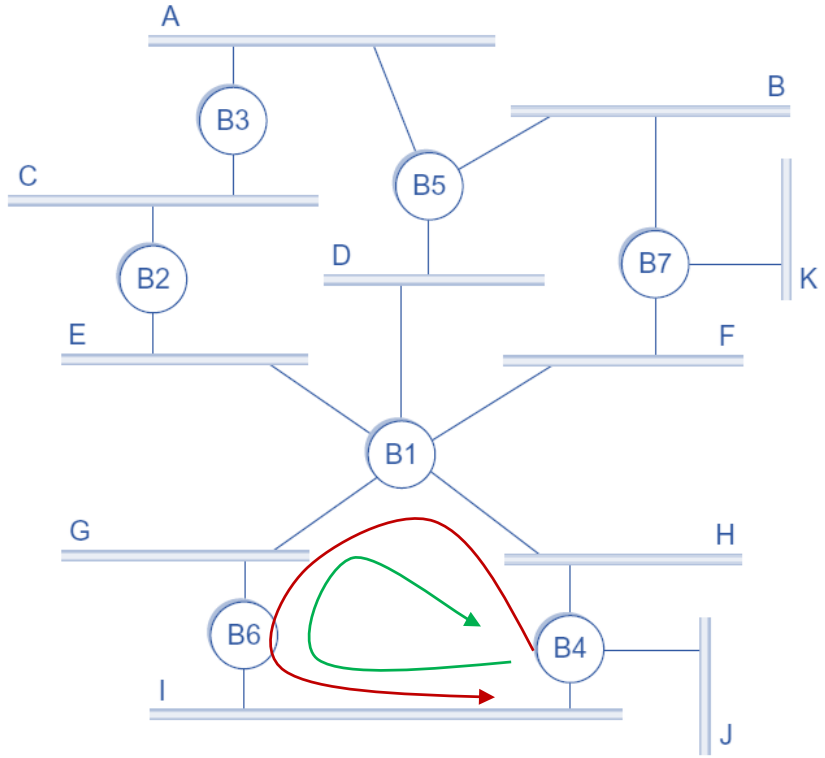


Cycles in the Network of Ethernet

- Possible Reasons
  - On purpose: Introduce redundancy
    - Cycles in network enable recovery from single link failure
  - Not on purpose: easy network management
    - Network manager dose not have the entire view of the network
- Problem
  - Broadcast storm

Broadcast Storm

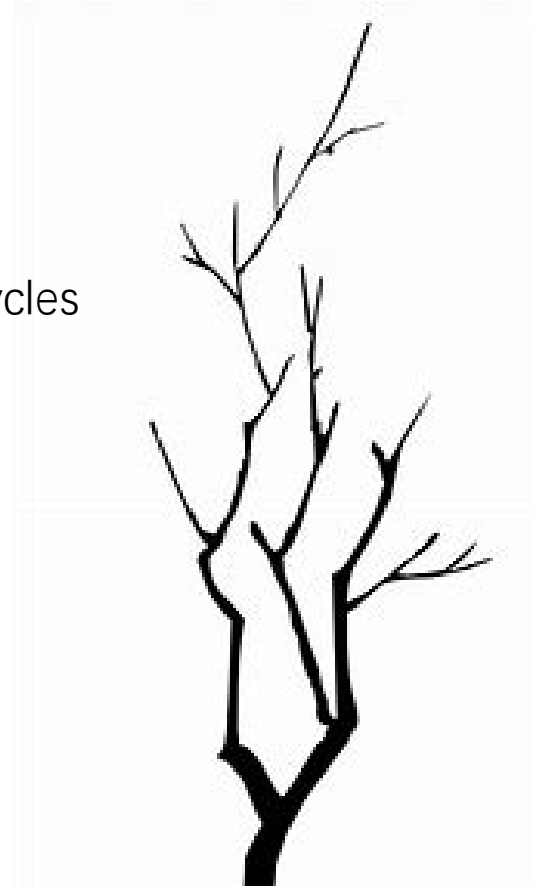
- Network J sends a packet to Network A, but B1,B4,B6 has no entry about Network A

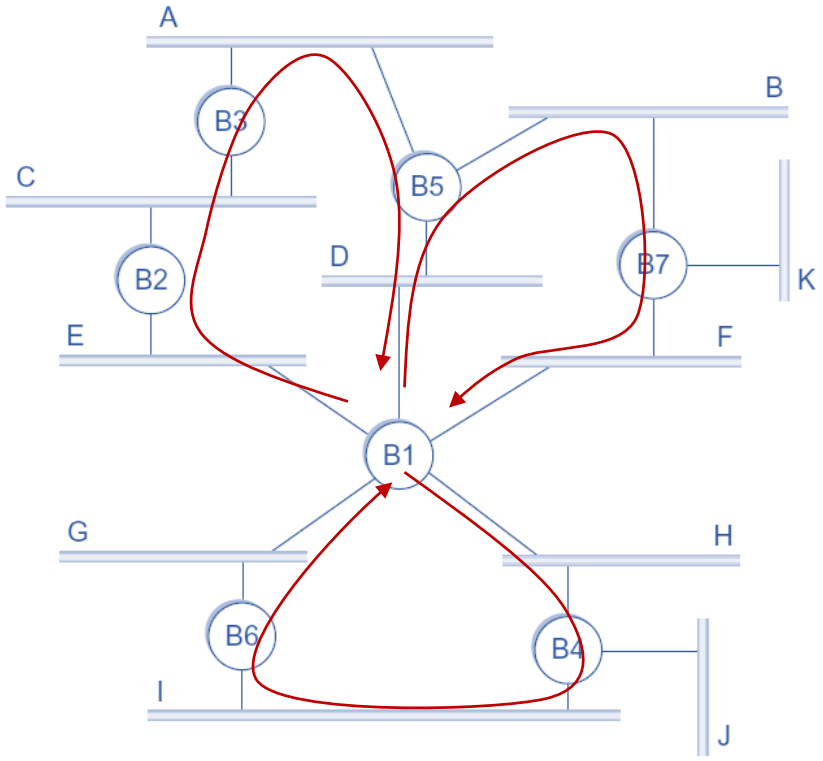


Handling Cycles

- Break the Cycles

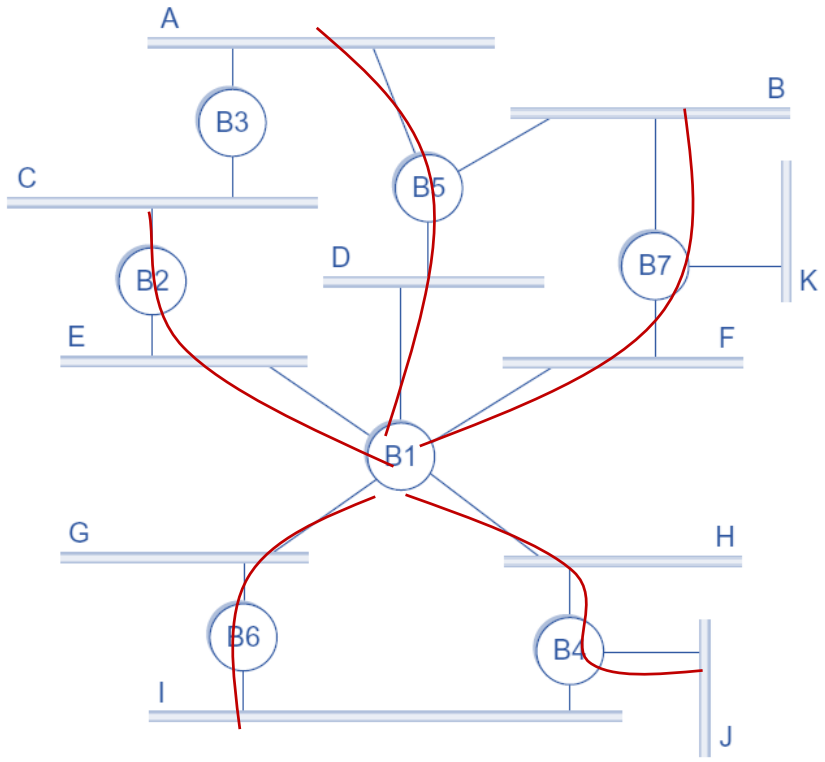
Tree has no cycles





Distributed Spanning Tree Algorithm

- Each switch is a vertex
- Each connected port of a switch is an edge
- Goal: A spanning tree is a sub-graph of this graph that covers all the vertices but contains no cycles
  - Each switch decides the ports over which it is and is not willing to forward frames



Distributed Spanning Tree Algorithm

- Step 1: Elect the Root Switch
  - Each switch has a unique identifier
    - B1, B2, B3,⋯and so on
  - Broadcast the configuration message to neighbors
    - (RootID)
  - Initially each switch thinks it is the root, and sends
    - (SelfID)
  - Update configuration message if
    - RxRootID < RootID
      - RootID = Rx_RootID

Distributed Spanning Tree Algorithm

- Step2: Elect the Root Port
  - A switch's Root Port is the port closest to the Root Switch
    - Every non-Root switch will select one and only one Root Port
  - Non-Root switches forward configuration message from Root Switch
    - Receive messages (RootID, SelfID, Dis) from Root Switch
    - Replace SelfID and calculate Distance to the Root Switch, and forward to other ports
    - Choose the Root Port
      - Rx_Dis_to_Root+1 < Dis_to_Root
        - Dis_to_Root = Rx_Dis_to_Root + 1
        - Set **Root Port** to the port that receives the message
      - Rx_Dis_to_Root +1 == Dis_to_Root && SelfD > RxID
        - Set **Root Por**t to the port that receives the message

Distributed Spanning Tree Algorithm

- Step3: Elect the Designated Port
  - A Designated Port functions as the single port that connect the connected Ethernet segment the Root Switch
  - Elect according to distance to Root Switch and the SelfID
    - Receive messages (RootID, SelfID, Dis) from neighbor switches
      - If Rx_Dis_to_Root > Dis_to_Root
      - or Rx_Dis_to_Root == Dis_to_Root && SelfD < RxID
        - Set this port as Designated Port
          - Forward frames from this port to the Root Switch
      - else
        - Close this port
          - Do not forward frames from this port

Distributed Spanning Tree Algorithm

- Rx_RootID == RootID && Rx_Dis_to_Root < Dis_to_Root+1
  - Dis_to_Root = Rx_Dis_to_Root + 1
  - Set output port to the port that receives the message
- Rx_RootID == RootID && Rx_Dis_to_Root == Dis_to_Root+1 && SelfD > RxID
  - Set output port to the port that receives the message
- Use the selected output port to forward packets

Reference

- Textbook 3.1