Introduction to Packet Switching

Hui Chen
Department of Computer & Information Science
CUNY Brooklyn College

Review and Discussion

- What have we discussed so far?

Review and Discussion

- Design objectives of computer networks
  - General purpose
  - Cost-effective network sharing
  - Fair network link allocation
  - Robust connectivity
- Building computer networks from ground-up: direct link networks
  - Smallest network
  - Problems to solve
    - Encoding
    - Framing
    - Error detection and correction
    - Reliable delivery
    - Media access control
  - Example
    - Ethernet

Review and Discussion

- Building computer networks from ground-up: direct link networks
  - Smallest network
  - Problems to solve
    - Encoding
    - Framing
    - Error detection and correction
    - Reliable delivery
    - Media access control
  - Example
    - Ethernet
- What are the limitations of the direct link networks?

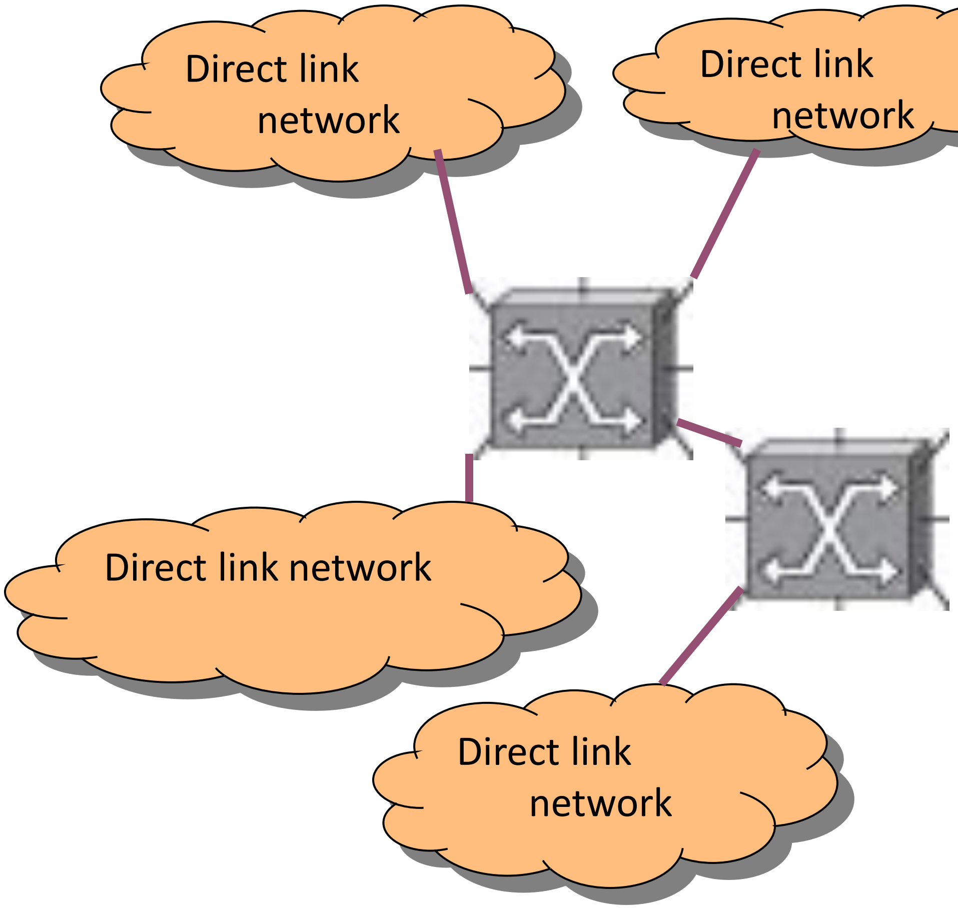
Review and Discussion

- Building computer networks from ground-up: direct link networks
  - Smallest network
  - Problems to solve
    - Encoding
    - Framing
    - Error detection and correction
    - Reliable delivery
    - Media access control
  - Example
    - Ethernet
- What are the limitations of the direct link networks?
  - Scalability (Size): how many users can the network accommodate?
  - Heterogeneity: how do we connect different types of direct link networks?

Lecture Outline

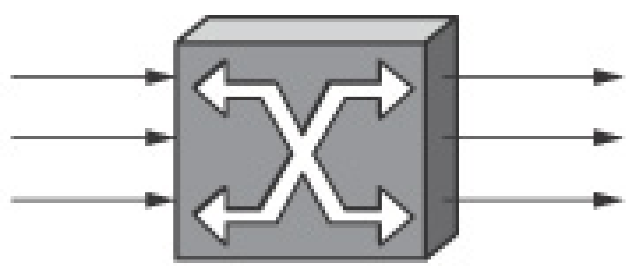
- Addressing the scalability problem: design scalable networks
- Mechanism (algorithms): switching
  - Datagram switching
  - Virtual Circuit
  - Source routing

The Design: Switched Networks



Switches

- Special node that forwards packets/frames
  - Multiple-input-multiple-output devices
  - Forward packets/frames from input port to output port
  - Switches can connect to each others
  - Output port selected based on destination address in packet/frame header
  - Provide high aggregate throughput
  - Layer 2 switches
    - Each link runs data link protocol



Switches

- Think about how telephone networks (circuit-switched networks) work
  - How switching (data forwarding) is performed?
    - A physical circuit is established → someone has to help you.
      - Someone = a real person or a computer
    - The circuit is dedicated to one connection
    - Each link can be shared (multiplex) a fixed number of connections (TDM or FDM)

Central office distribution frame

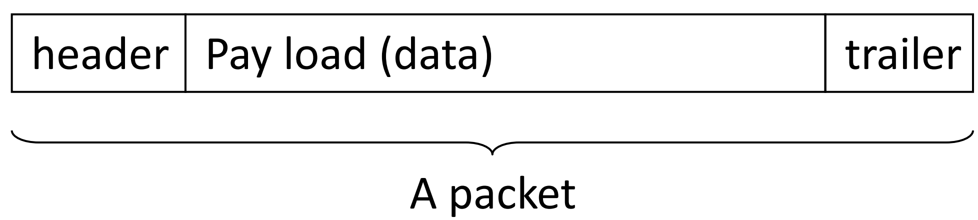
Computer networks are packet switched networks

Data are divided into frames/packets

Still, one has to decide which port (of a link) to forward a frame/packet

Packet-switched Networks

- Data are divided and sent using *packets*
  - A packet has a header and trailer which contain control information
- *Store-and-forward*
  - Each packet is passed from node to node along **some** path through the network
  - At each node, the entire packet is received, stored briefly, and then forwarded to the next node
- Statistical multiplexing
  - No capacity is allocated for packets



Switching Algorithms

- Q: how does a switch decide on which output port (on a link) to place a packet?

Switching Approaches

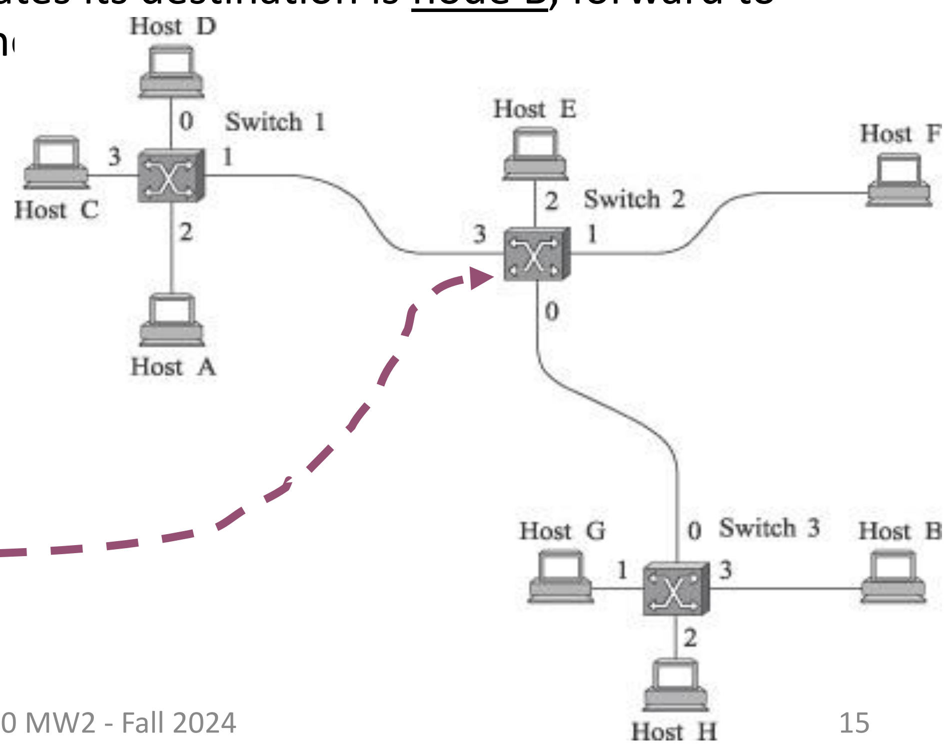
- Datagram switching
  - Connectionless model
- Virtual circuit switching
  - Connection-oriented model
- Source routing
- Common properties
  - Switches have identifiable ports
  - Hosts/nodes are identifiable

Datagram Switching: Data Structure

- Each switch maintains a forwarding table
- Frame header contains the identifier of destination node
- Forward packets/frames based on the table
  - Example: if frame header indicates its destination is <u>node B</u>, forward to <u>port 0</u> → done by looking up th

Forwarding/Routing Table for **Switch 2**

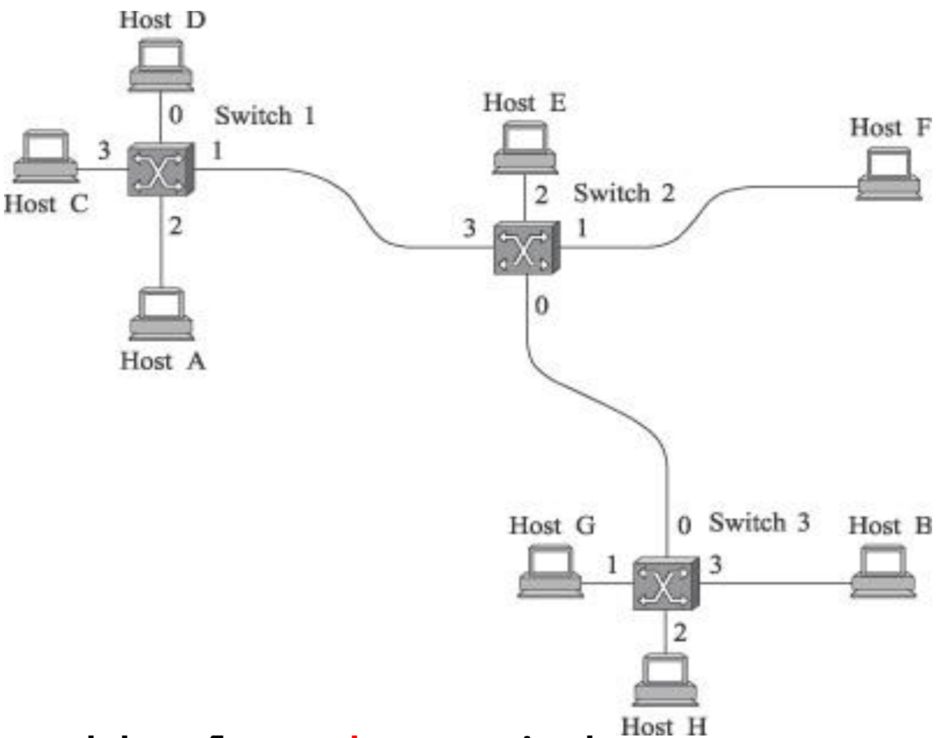
Destination	Port
A	3
B	0
C	3
D	3
E	2
F	1
G	0
H	0



Exercise 1

Forwarding/Routing Table for Switch 2

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



- Construct the forwarding tables for other switches (switches 1 & 3)

Datagram Switching: Discussion

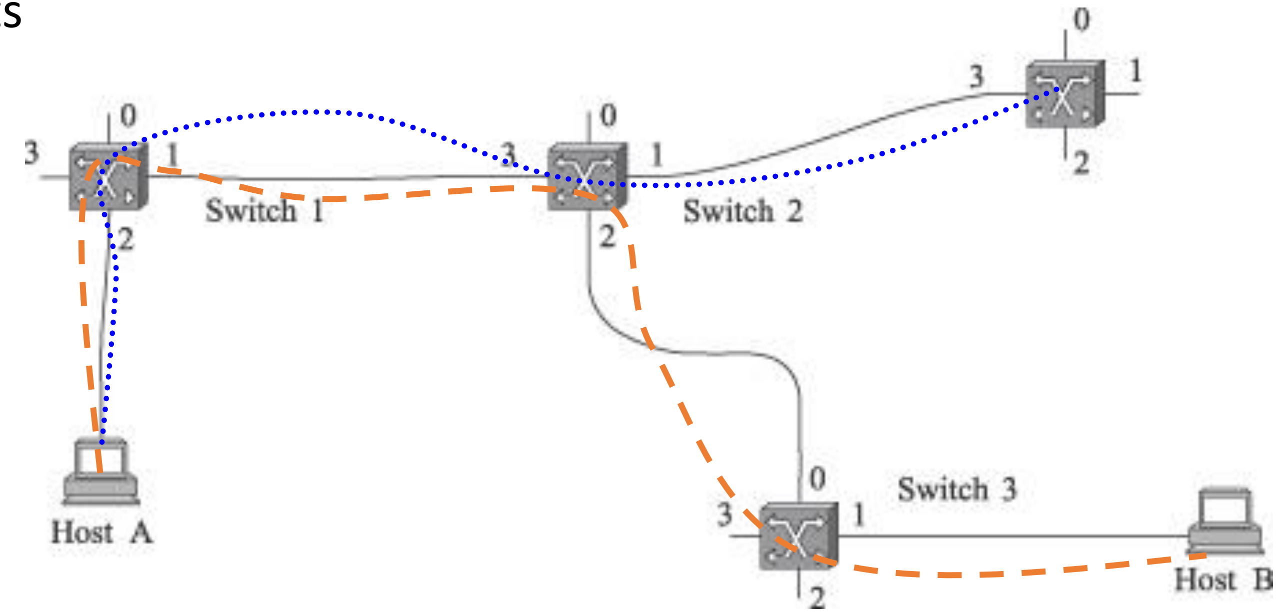
- Each node maintains a forwarding table
- No connection setup
- Hosts/switches sends/forwards packets independently
- Hosts/switches do not know if the network can deliver a packet to its destination
- A switch/link failure might not be catastrophic
  - Find an alternate route and update forwarding table

Virtual Circuit Switching

- Connection-oriented model
  - Connection setup → establish "virtual circuit (VC)"
  - Data transfer → subsequent packets follow same circuit
  - Tear down VC
- Each switch maintains a VC table
  - An entry (row) in VC table must have
    - VCI: identify connection at this switch **within** a link → a different VCI will be used for outgoing packets
    - Incoming interface, e.g., a port for receiving packets
    - Outgoing interface, e.g., a port for forwarding packets
- Frame header contains VC number (VCI value) of **next link** along a VC

Virtual Circuit Switching: Example

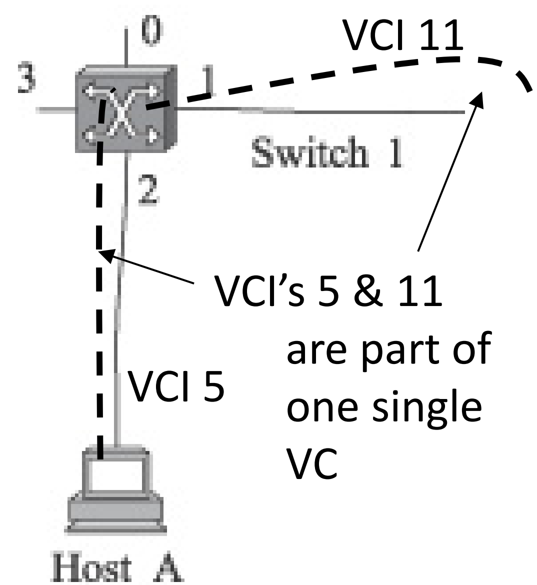
- Example: host A → host B
  - Switches needed?
    - switches 1, 2, and 3
  - Network do not explicitly maintain global information about virtual circuits



Two planned virtual circuits in red dashed line and blue dotted line

Virtual Circuit Switching: Example: VC Table

- Setup phase (could be performed manually for a network administrator) → permanent VC→ Establish VC table for each switch
- Example: Switch 1
  - When host A sends out a frame, it places the VCI (i.e. 5) of next link into the frame header
  - Switch 1 looks up an entry based on both incoming interface (i.e., 2) and the VCI (i.e., 5) in the frame header to determine outgoing port (i.e., 1) and VCI (i.e., 11)
  - The scope of VCI values is links
    - Unused VCI value on the link (Host A to Switch 1)
    - VCI can be duplicated on different link



Virtual circuit table entry for **switch 1**

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

Virtual Circuit Switching: Example: VC Table

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

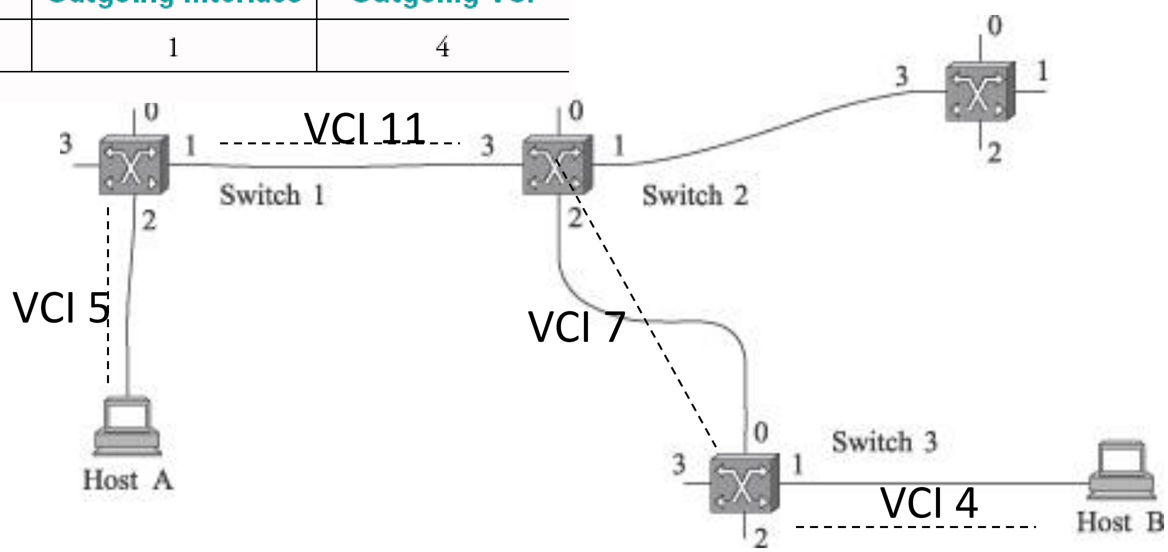
Virtual circuit table entry for switch 1

Incoming Interface	Incoming VCI	Outgoing Interface	Outgoing VCI
3	11	2	7

VC table entry at switch 2

Incoming Interface	Incoming VCI	Outgoing Interface	Outgoing VCI
0	7	1	4

VC table entry at switch 3

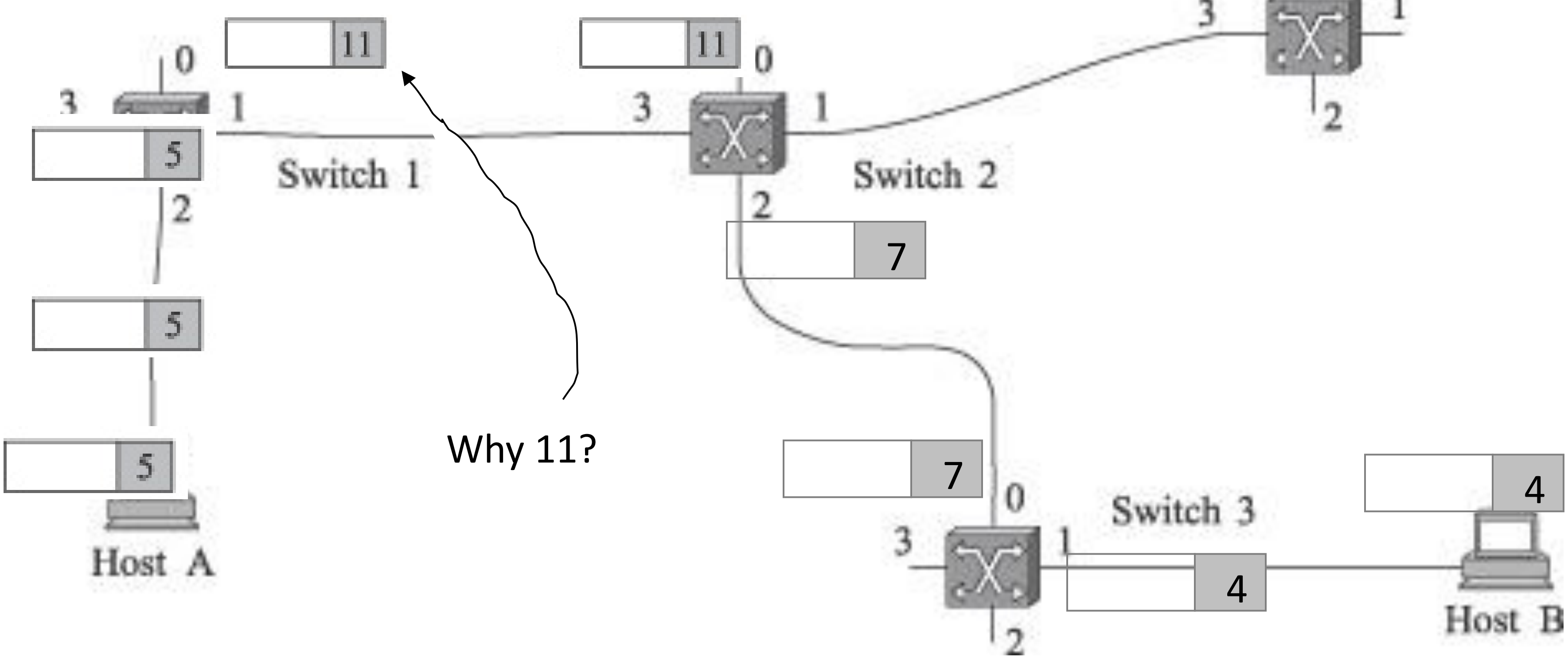


Virtual Circuit Switching: Example

- Host A sends a frame to host B

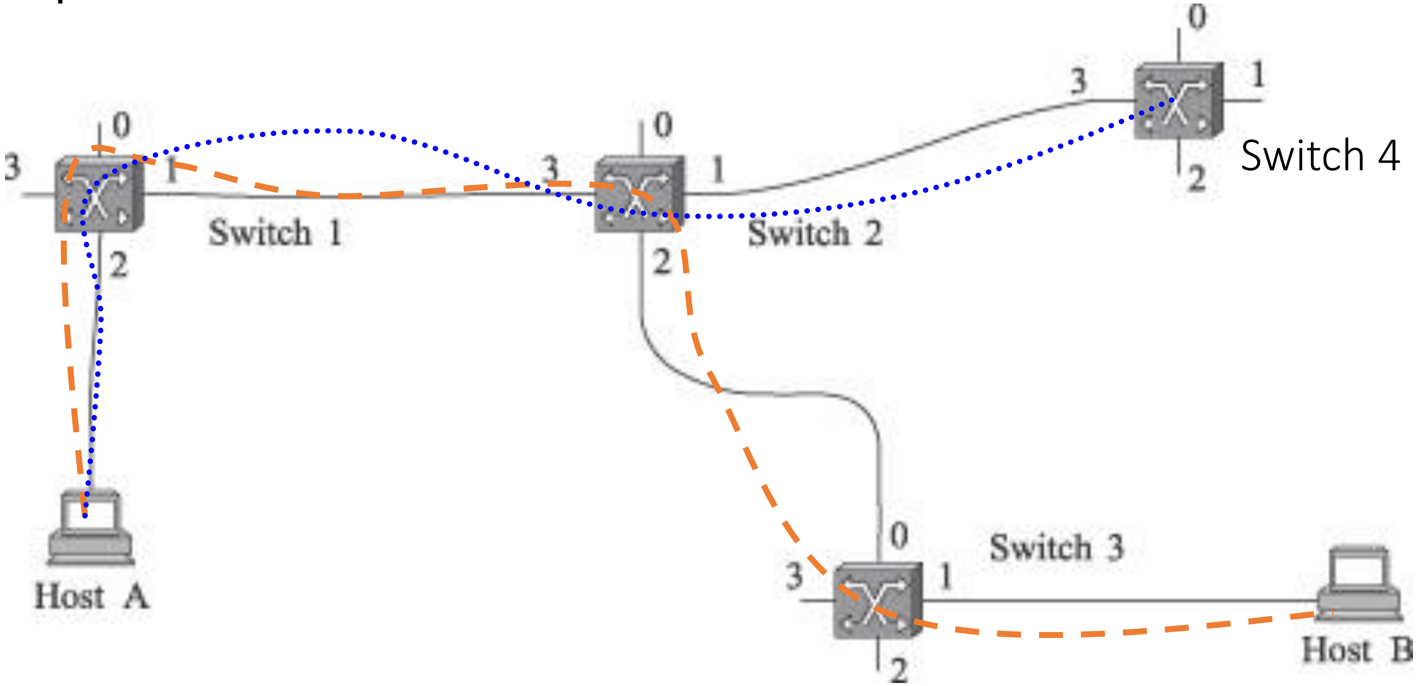
Switch 1

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



Exercise 2

- Construct Virtual Circuit (VC) table entry for all the switches on the Virtual Circuit for both red and blue Virtual Circuits
- List VC tables for switches 1, 2, 3, and 4. You may make necessary assumptions.



Virtual Circuit Switching: Connection Setup

- Connection setup
  - Permanent virtual circuit (PVC): manual configured → unmanageable for great number of nodes
  - Switched virtual circuit (SVC): automatically configured via signaling
    - A process similar to datagram model

Virtual Circuit: Discussion

- Connection setup takes 1 RTT minimally
- VCI number typically needs less memory space. Per-packet overhead is less than that of the datagram model
- Need VC re-setup in case of a connection failure
- Possible to allocate network resources during VC setup

Comparison of Datagram and Virtual Circuit Switching

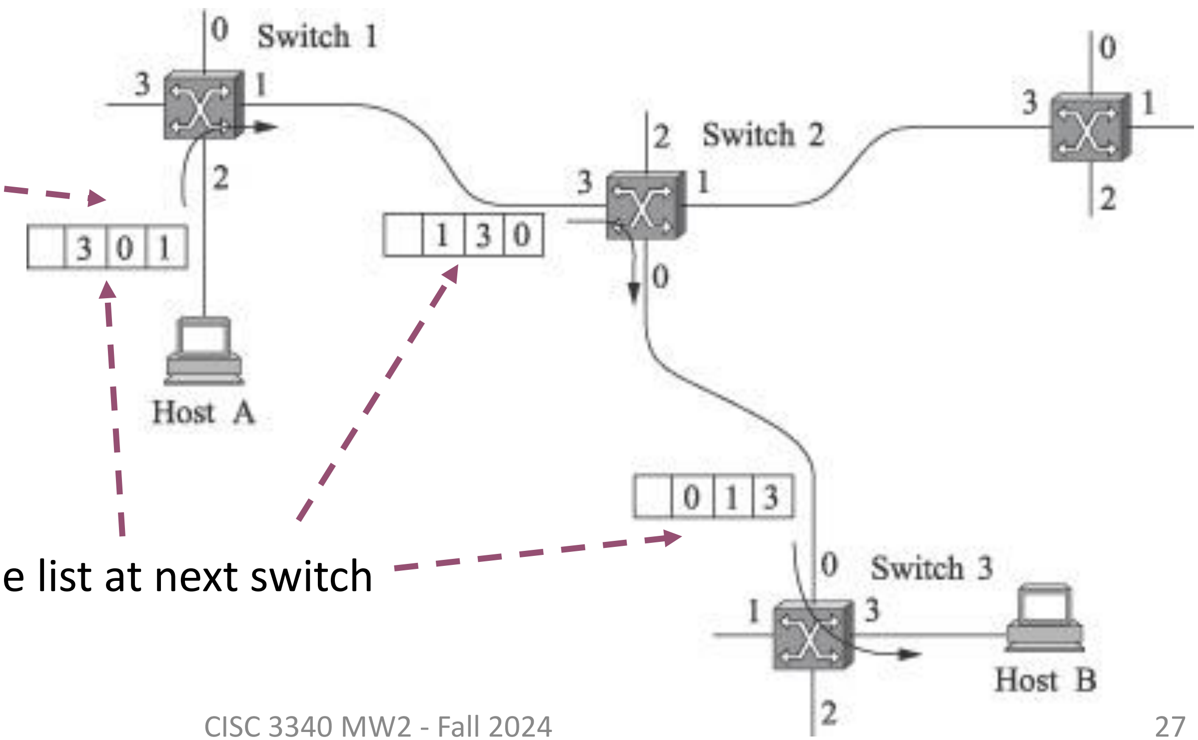
- Virtual Circuit
  - Need connection setup
    - Typically wait full RTT for connection setup before sending first data packet.
  - While the connection request contains the full address for destination, each data packet contains only a small identifier, making the per-packet header overhead small.
    - In datagram switching: forwarding table contains entries for every host → large table → more memory, slow lookup
  - Delivery assurance or failure
    - If a switch or a link in a connection fails, the connection is broken and a new one needs to be established.
  - Connection setup provides an opportunity to reserve resources → Quality of Service (QoS)

- Datagram
  - No connection setup
    - There is no RTT delay waiting for connection setup; a host can send data as soon as it is ready.
  - Since every packet must carry the full address of the destination, the overhead per packet is higher than for the connection-oriented model.
    - In virtual circuit switching: VC table contains only "circuits" to be used → smaller table → less memory, fast lookup
  - Delivery assurance or failure
    - Source host has no way of knowing if the network is capable of delivering a packet or if the destination host is even up.
  - Since packets are treated independently, it is possible to route around link and node failures → difficult to satisfy QoS

Source Routing

- Source host knows network topology to deliver a packet/frame
- Source host places output ports of each switch along the route into the frame header
  - Example: Host A sends a frame to host B

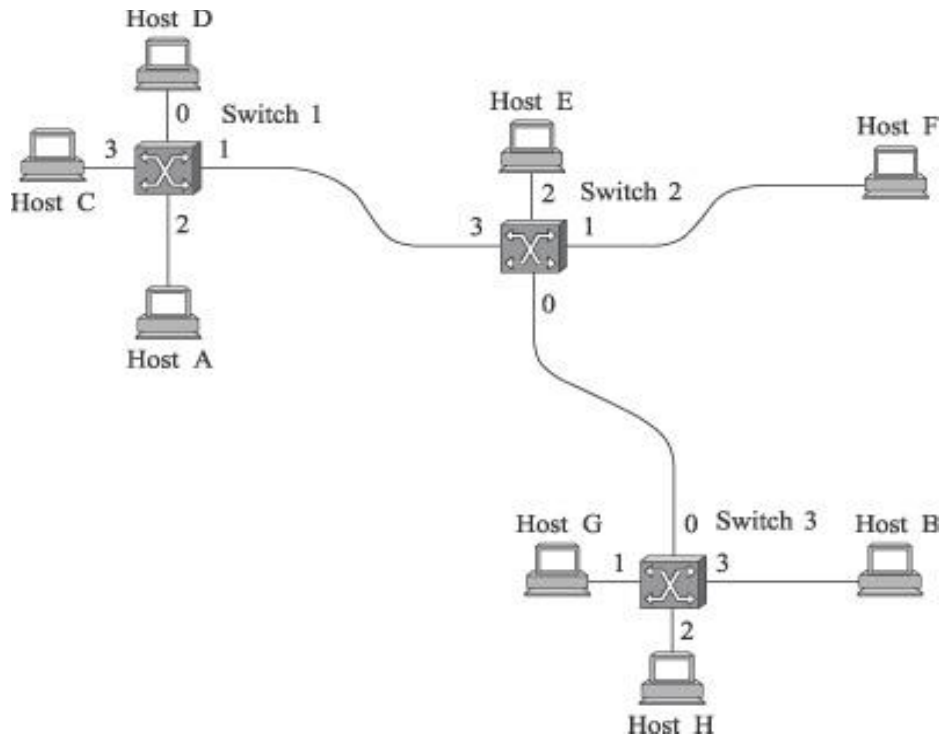
Ordered list of outputs of switches in header



Rotate the list at next switch

Exercise 3

- Assume source routing presented in previous slide is used, show headers of a frame leaves from Host H and arrives at Host D at each switches along the path



Summary

- Switches → scalable networks
- Datagram switching
- Virtual circuit switching
- Source routing
- *Q: Example in practice?*
  - *Ethernet*