# CISC 7332X T6

# Packet Switching

Hui Chen

Department of Computer & Information Science

CUNY Brooklyn College

# Acknowledgements

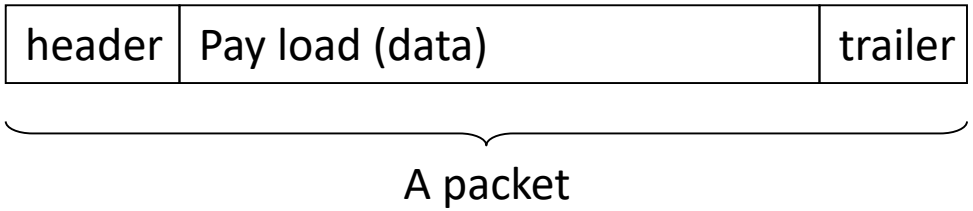

- Some pictures used in this presentation were obtained from the Internet
- The instructor used the following references
  - Andrew S. Tanenbaum, Computer Networks, 5th Edition, Prentice-Hall, 2010
  - Larry L. Peterson and Bruce S. Davie, Computer Networks: A Systems Approach, 5th Edition, Elsevier, 2011

# Design Issues

- Store-and-forward packet switching
- Connectionless service (datagrams)
- Connection-oriented service (virtual circuits and source routing)
- Comparison of virtual-circuits and datagrams
- Source routing

# Packet-switched Networks

- Data are divided and sent using *packets*
  - A packet has a header and trailer which contain control information
    - *e.g., destination and source addresses*
- *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



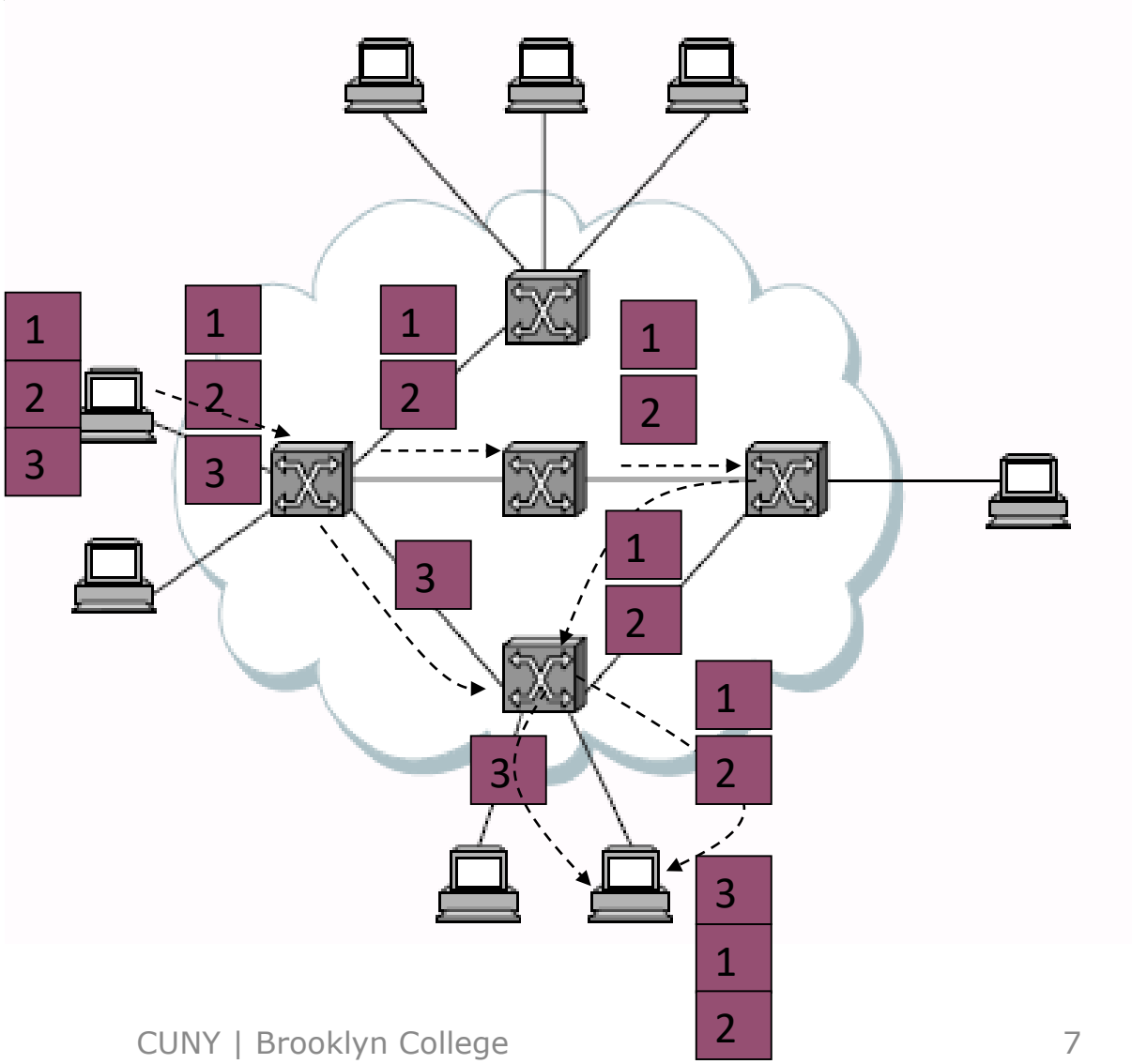
# Switching Approaches

- Datagram switching
  - Connectionless service
- Virtual circuit switching
  - Connection-oriented  service
- Source routing

# Datagram Packet Switching

- Network nodes process each packet independently
- Two consecutively-sent packets can take different routes.
- Implications:
  - A sequence of packets can be received in a different order than they were sent
  - Each packet header  must contain full address of the destination
- Example of networks using packet switching
  - Extended Ethernet LAN, a data link layer protocol
  - The Internet Protocol, a network layer protocol

# Example

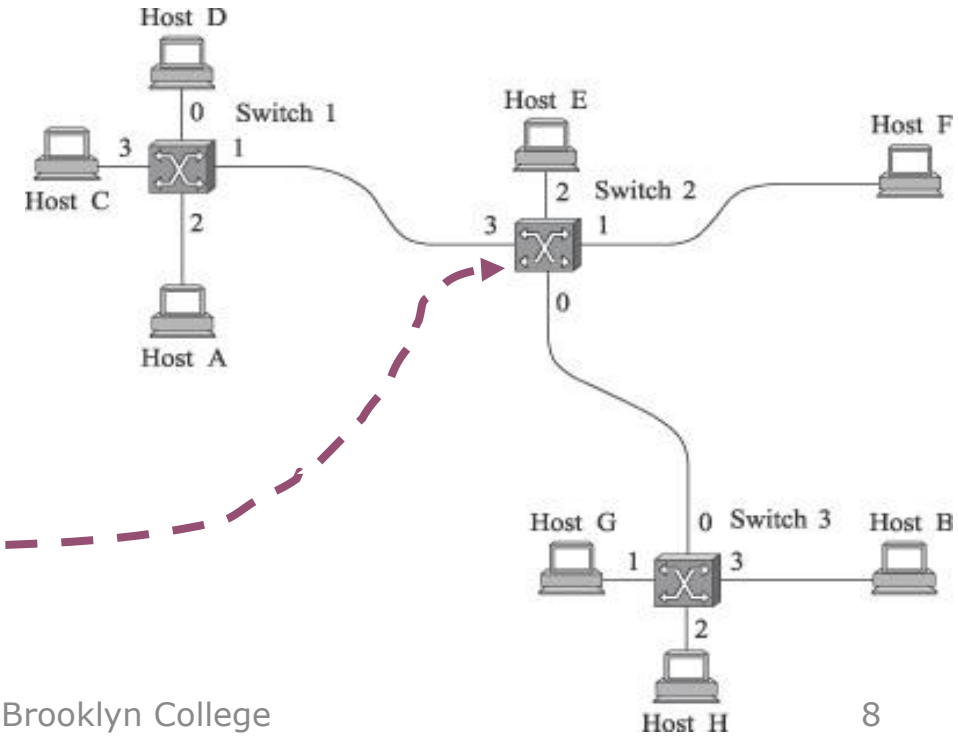


# Datagram Switching

- 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 node B, forward to port 0 → done by looking up the table

Forwarding/Routing Table for **Switch 2**

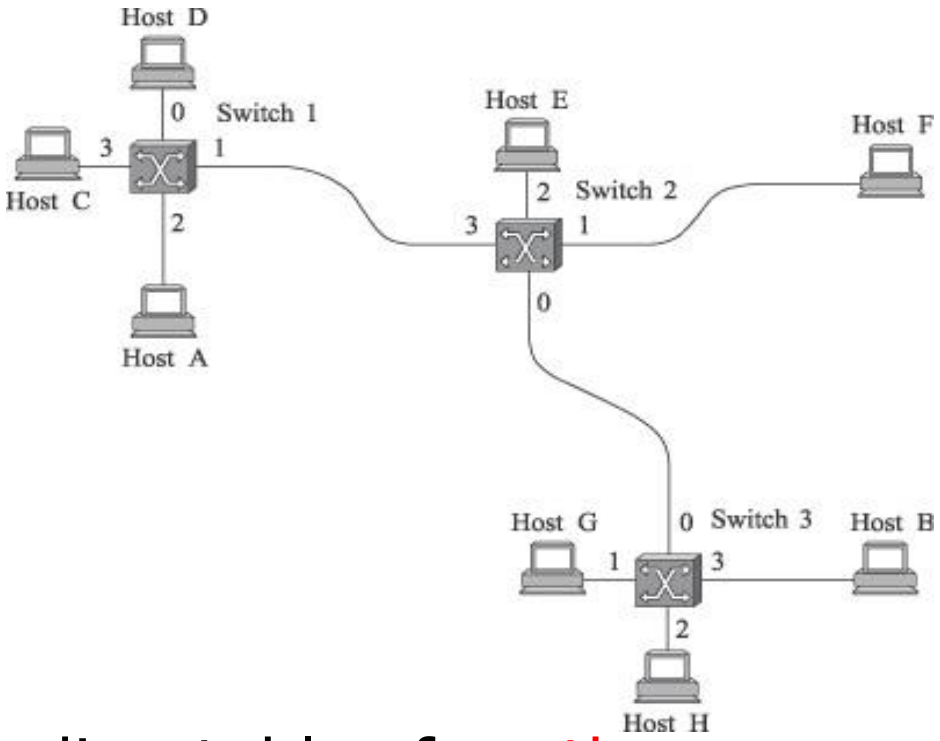
| Destination | Port |
|---|---|
| A | 3 |
| B | 0 |
| C | 3 |
| D | 3 |
| E | 2 |
| F | 1 |
| G | 0 |
| H | 0 |



# 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

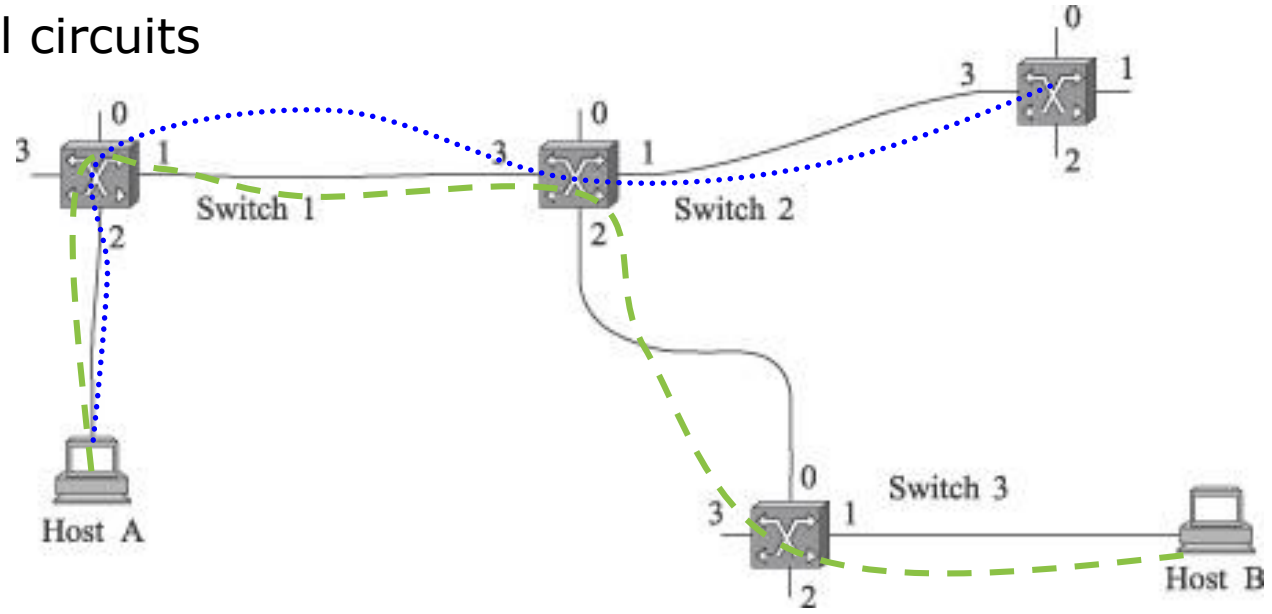
- Each node maintains a forwarding table
- No connection setup
- Hosts/switches sends/forwards packets independently
- Hosts/switches do not know if the network can deliver a packet to its destination
- A switch/link failure might not be catastrophic
  - Find an alternate route and update forwarding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

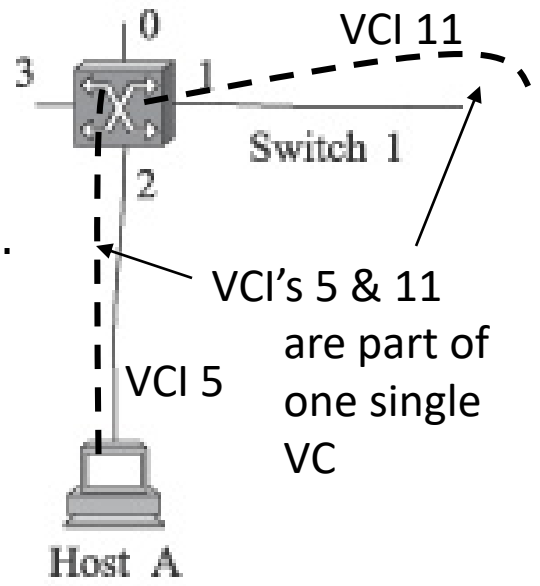
- Example: host A → host B
  - Switches needed? Switches 1, 2, and 3
  - Network do not explicitly maintain global information about virtual circuits




Two planned virtual circuits in dashed/dotted lines

# 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 |

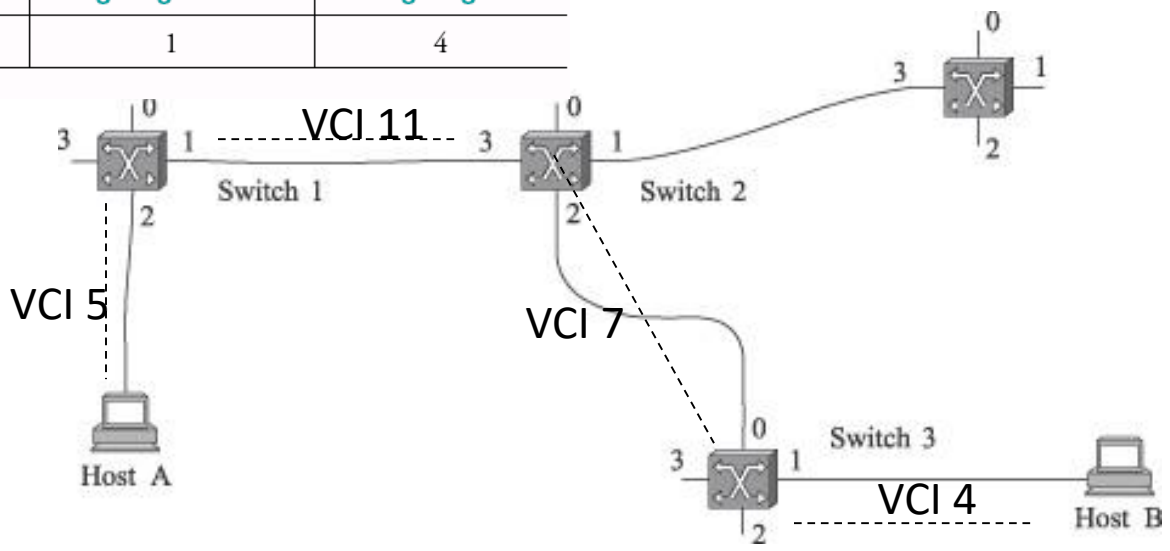
Virtual circuit table entry for switch 1

| 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

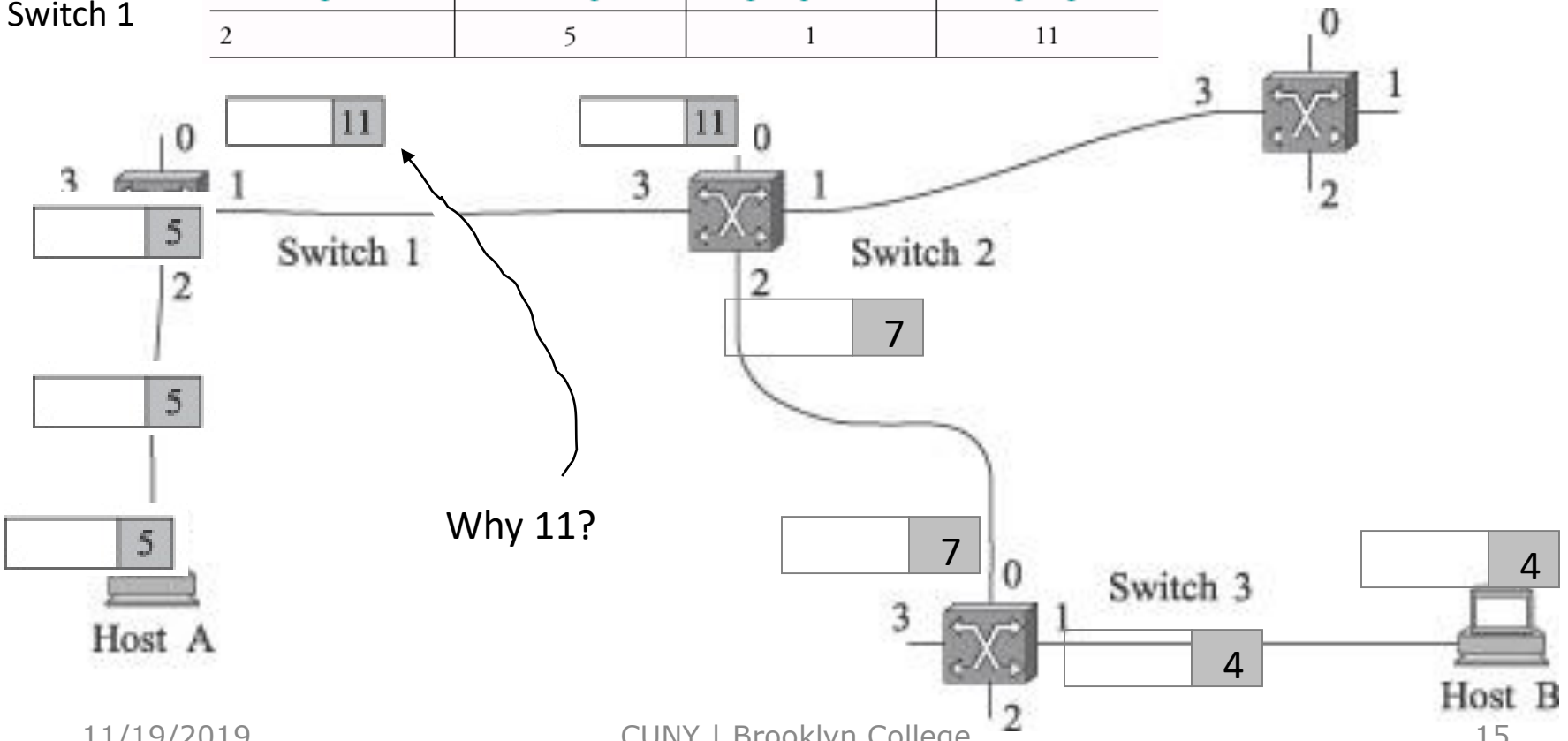


# Virtual Circuit Switching: Example

- Host A sends a frame to host B

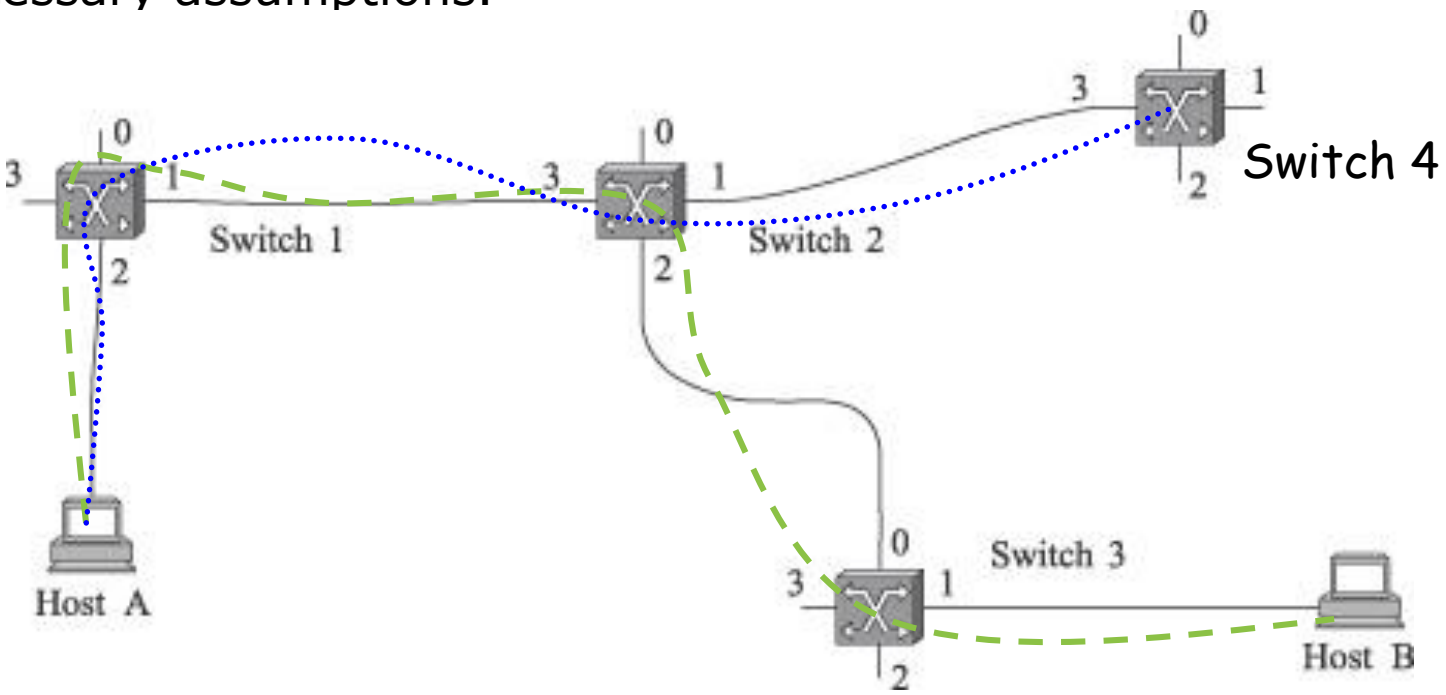
Switch 1

| 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 Virtual Circuits (dash or dotted lines)
- 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

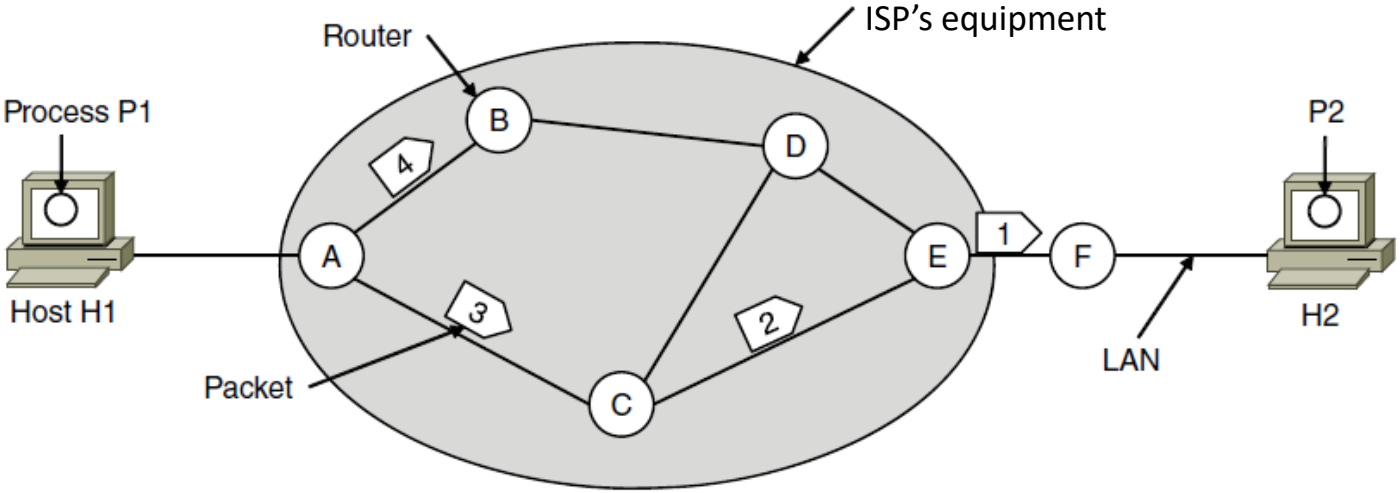
- Connection setup takes 1 RTT minimally
- VCI number typically needs less memory space. Per-packet overhead is less than that of the datagram model
- Need VC re-setup in case of a connection failure
- Possible to allocate network resources during VC setup

# Comparison of Virtual-Circuit & Datagram Switching

| Issue | Datagram network | Virtual-circuit network |
|---|---|---|
| Circuit setup | Not needed | Required |
| Addressing | Each packet contains the full source and destination address | Each packet contains a short VC number |
| State information | Routers do not hold state information about connections | Each VC requires router table space per connection |
| Routing | Each packet is routed independently | Route chosen when VC is set up; all packets follow it |
| Effect of router failures | None, except for packets lost during the crash | All VCs that passed through the failed router are terminated |
| Quality of service | Difficult | Easy if enough resources can be allocated in advance for each VC |
| Congestion control | Difficult | Easy if enough resources can be allocated in advance for each VC |

# Connectionless Service: Datagram Switching

- Different packets may take different paths




A's table (initially)

| | |
|---|---|
| A | |
| B | B |
| C | C |
| D | B |
| E | C |
| F | C |

A's table (later)

| | |
|---|---|
| A | |
| B | B |
| C | C |
| D | B |
| E | D |
| F | D |

C's Table

| | |
|---|---|
| A | A |
| B | A |
| C | |
| D | E |
| E | E |
| F | E |

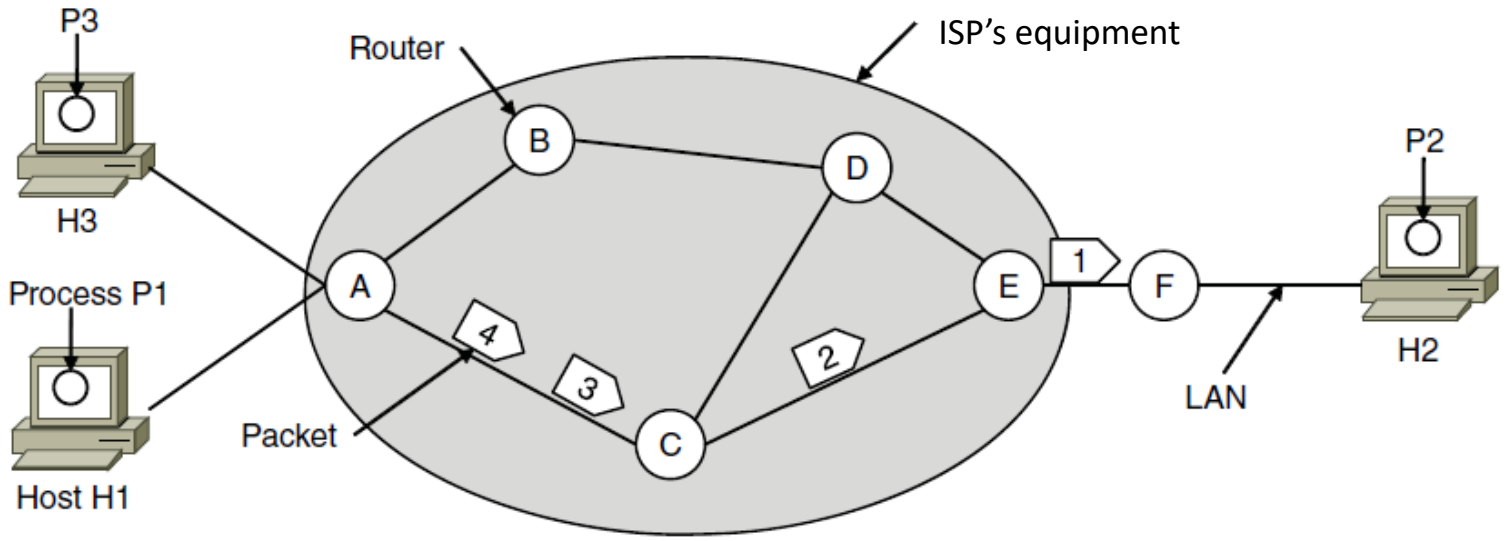
E's Table

| | |
|---|---|
| A | C |
| B | D |
| C | C |
| D | D |
| E | |
| F | F |

Dest. Line

# Connection-Oriented Service: Virtual Circuit Switching

- Packet is forwarded along a virtual circuit




A's table

| In: Line | Tag | Line | Tag: Out |
|---|---|---|---|
| H1 | 1 | C | 1 |
| H3 | 1 | C | 2 |

C's Table

| | | | |
|---|---|---|---|
| A | 1 | E | 1 |
| A | 2 | E | 2 |

E's Table

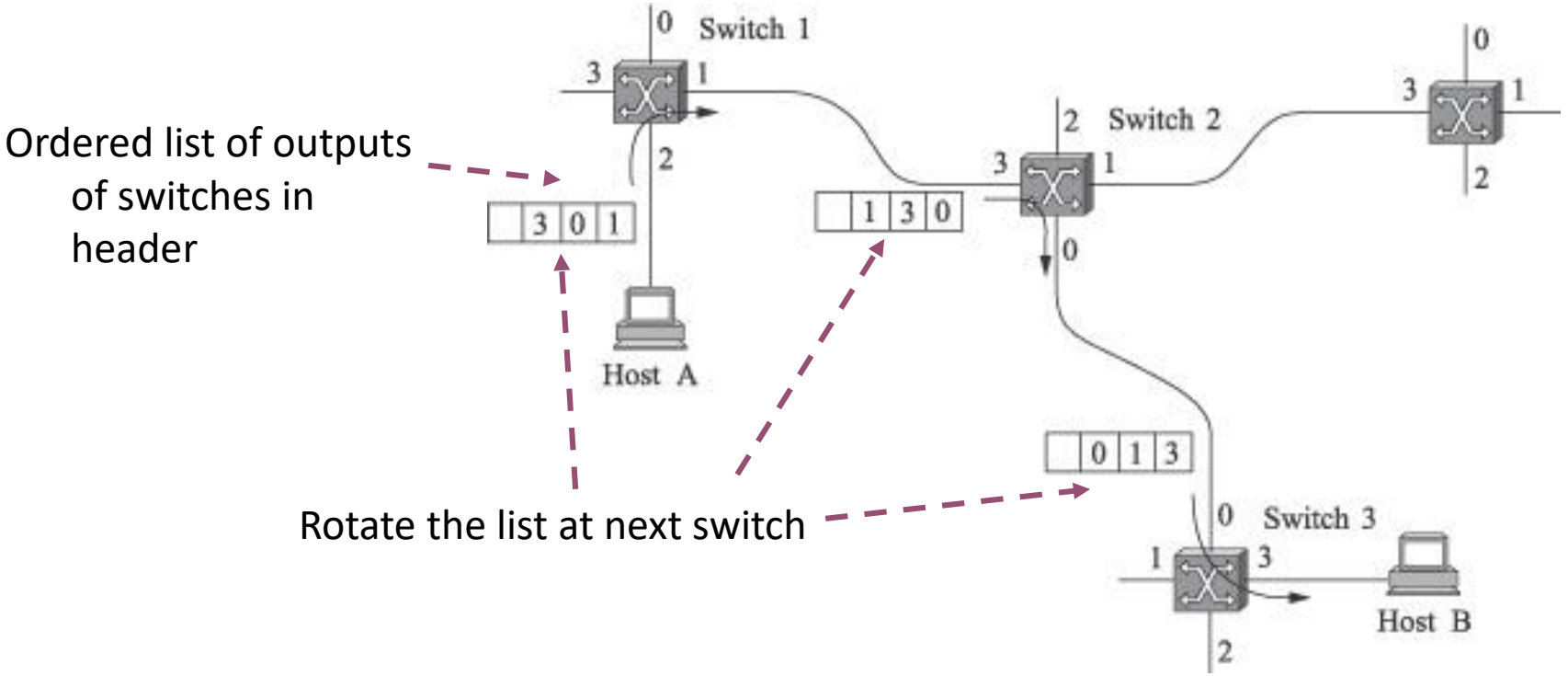
| | | | |
|---|---|---|---|
| C | 1 | F | 1 |
| C | 2 | F | 2 |

# 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

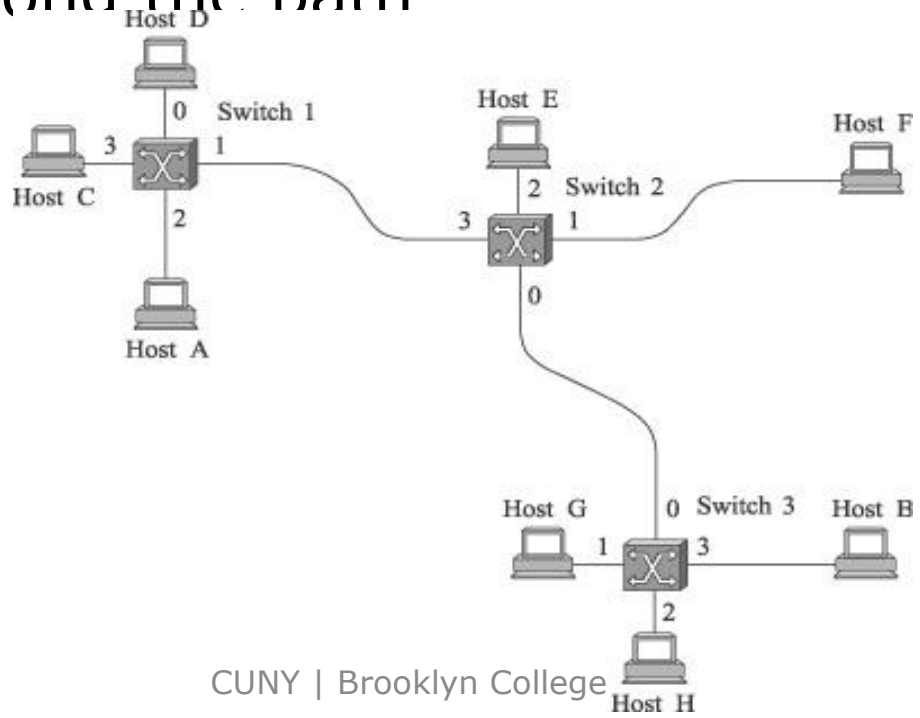
# Source Routing: Example

- Example: Host A sends a frame to host B



# 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



# Questions?

- Packet switching
- Datagram switching
  - *Example in practice?*
    - *Ethernet and learning bridge*
- Virtual circuit switching
- Source routing