Computer Network

Internetworking

Problems

- In Chapter 2 we saw how to connect one node to another, or to an existing network.
- Chapter 3 challenges:
  1. How do we connect these smaller similar types of networks together?
     1. Hubs, bridges and switches
  2. How do we interconnect different types of networks to build a large global network?
     1. Gateways/routers
  3. How do we build networks of global scale?

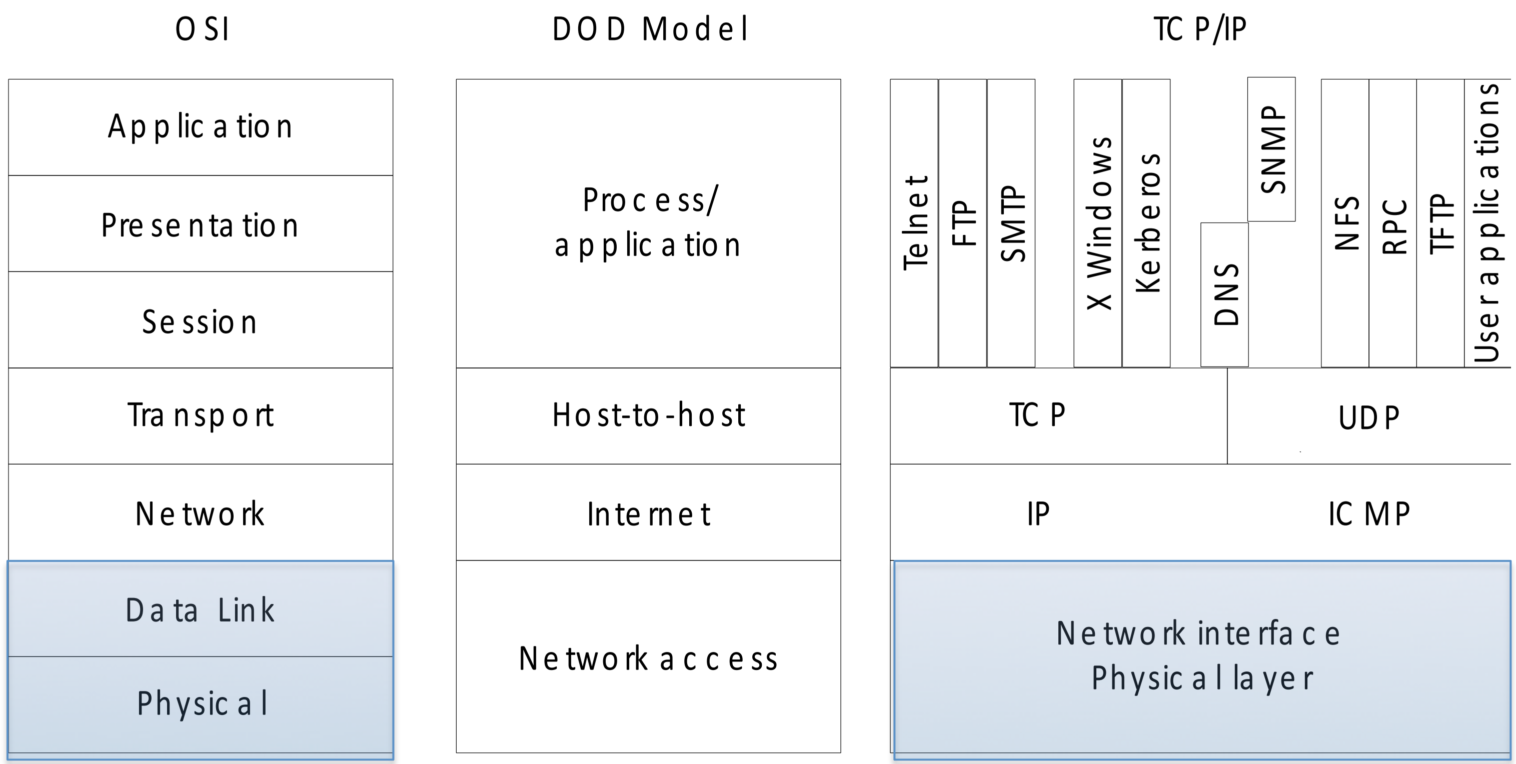
Chapter Outline

- Switching and Bridging
- Basic Internetworking (IP)
- Routing

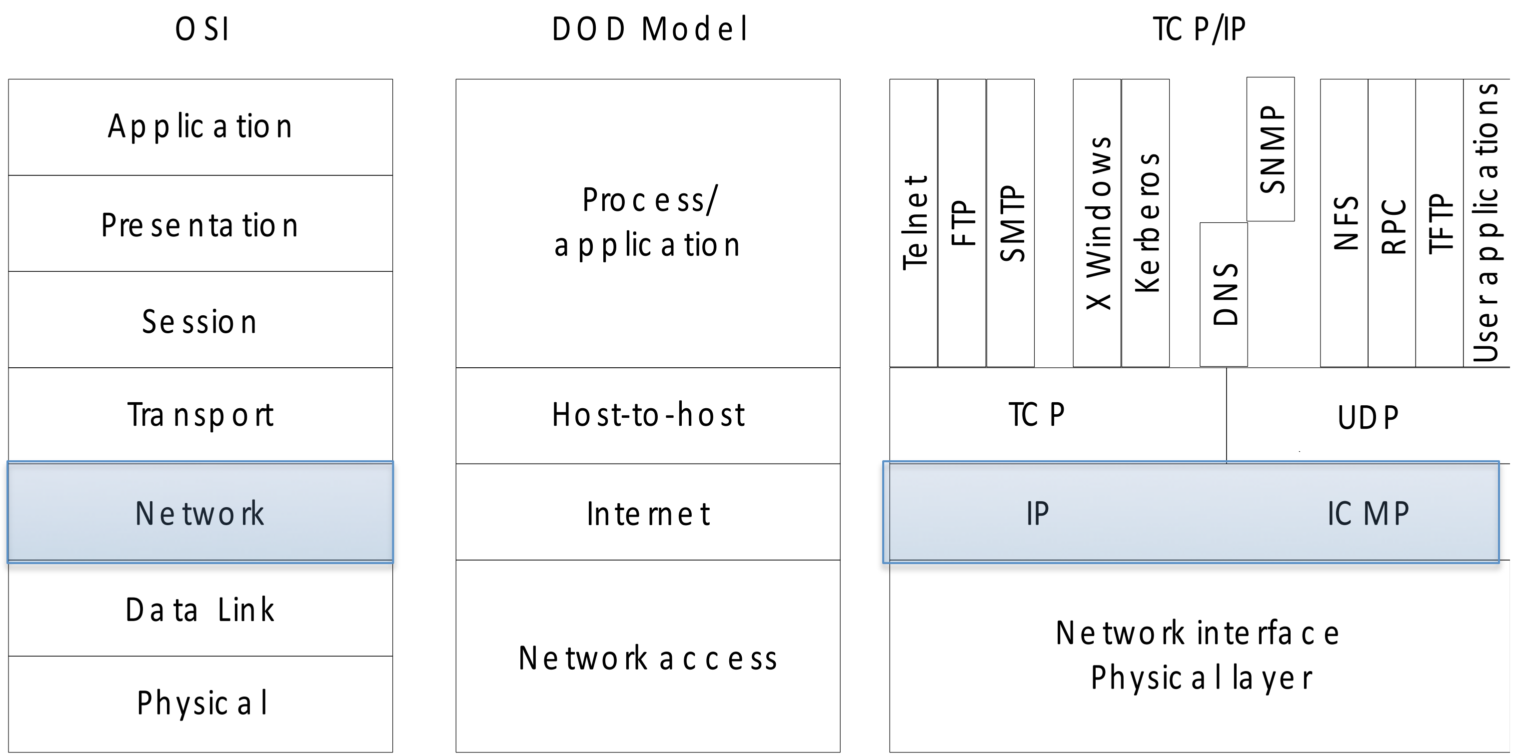
Chapter Goal

- Understanding the functions of switches, bridges and routers (gateways)
- Discussing Internet Protocol (IP) for interconnecting networks
- Understanding the concept of routing

How TCP/IP Corresponds to OSI



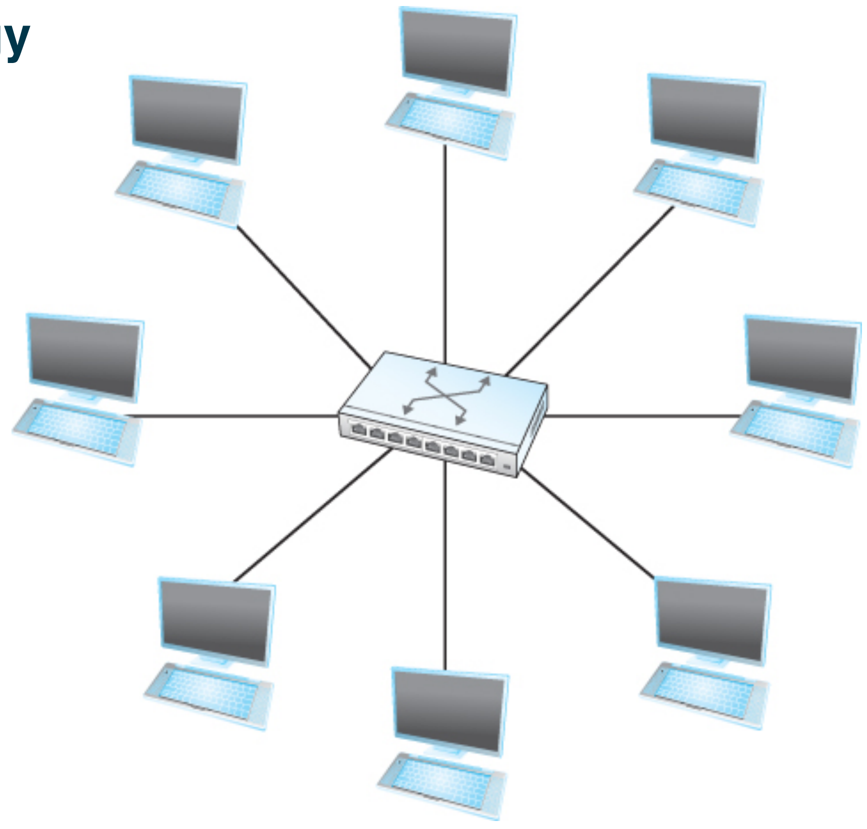
How TCP/IP Corresponds to OSI



Switching and Forwarding

- Switch
  – A multi-input, multi-output device

star topology

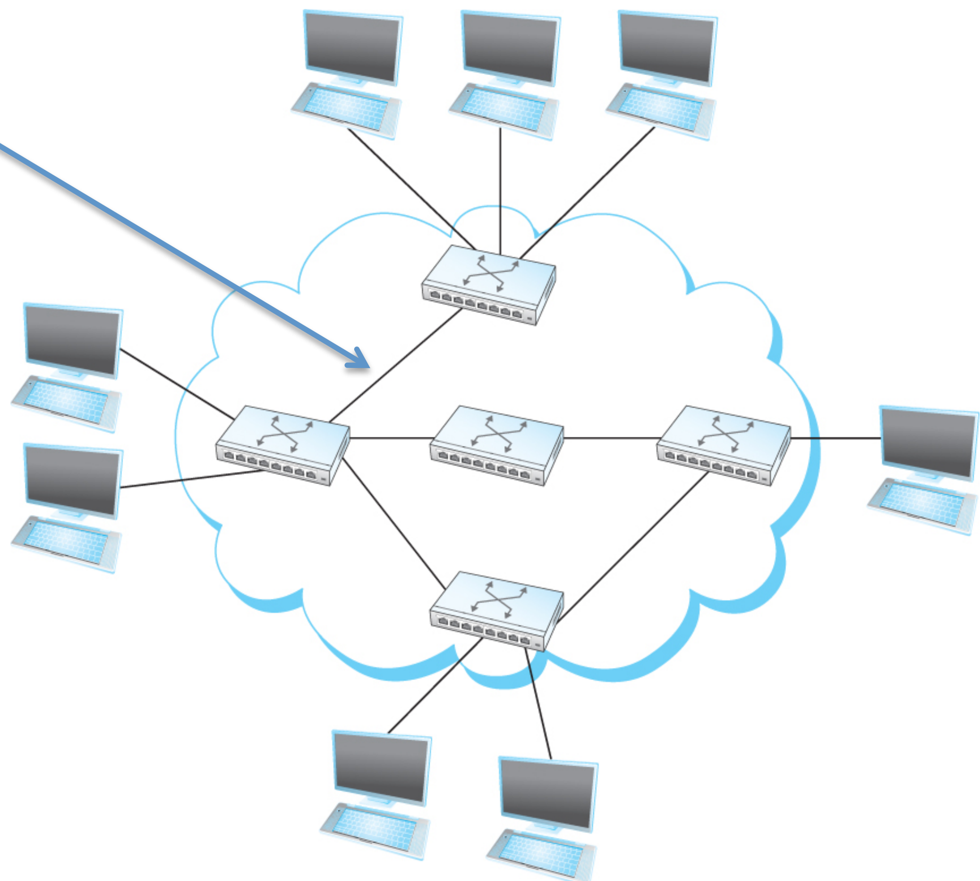


Switching and Forwarding

- Switching and forwarding
  - Interconnect links to form a large network

Point-to-point link

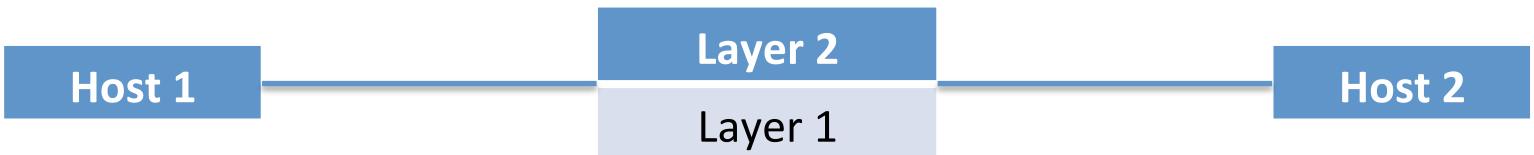
bus - Ethernet =>



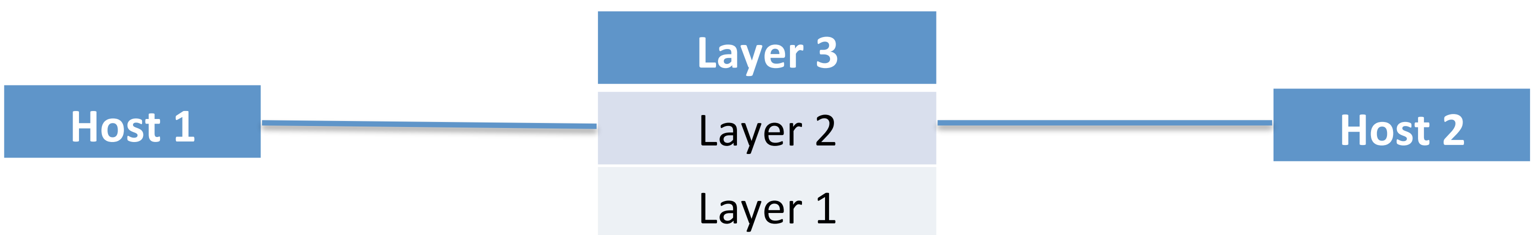
Switching and ~~Forwarding~~

****important**

- Layer 2 Switching vs. Layer 3 Switching



Layer 2 Switch



Layer 3 Switch

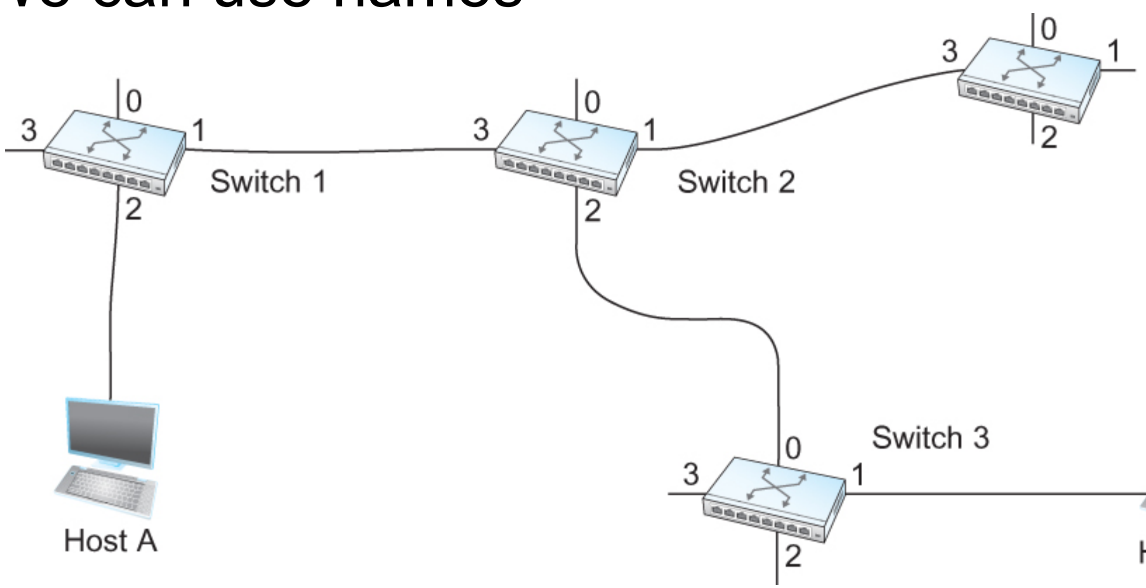
(Layer 3 based on IP, layer 2 switch based on MAC)

Switching and ~~Forwarding~~

- How does the switch decide which output port to place each packet on?
- Two common approaches
  1. *Datagram or Connectionless approach*
  2. *Virtual circuit or Connection-oriented approach*
- A third approach *source routing* is less common

Switching and Forwarding

- Assumptions
  - Each host has a globally unique address (IP address)
  - There is some way to identify the input and output ports of each switch
    - We can use numbers
    - We can use names

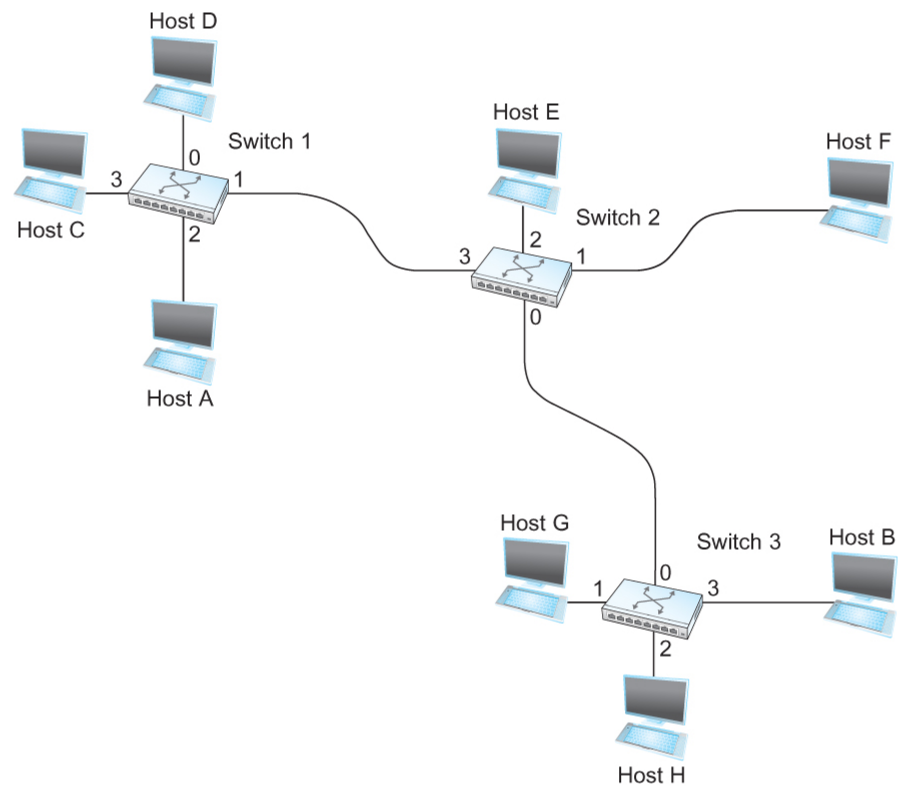


Switching and Forwarding

- Datagrams (connectionless approach)
  - Key Idea
    - Every packet contains enough information to enable any switch to decide how to get it to the destination (or one hop closer to the destination)
      - Every packet contains the complete destination address
    - Each switch
      - Decodes this destination address
      - Uses this address to Look up a table for a port number
      - Sends out the packet through the port number

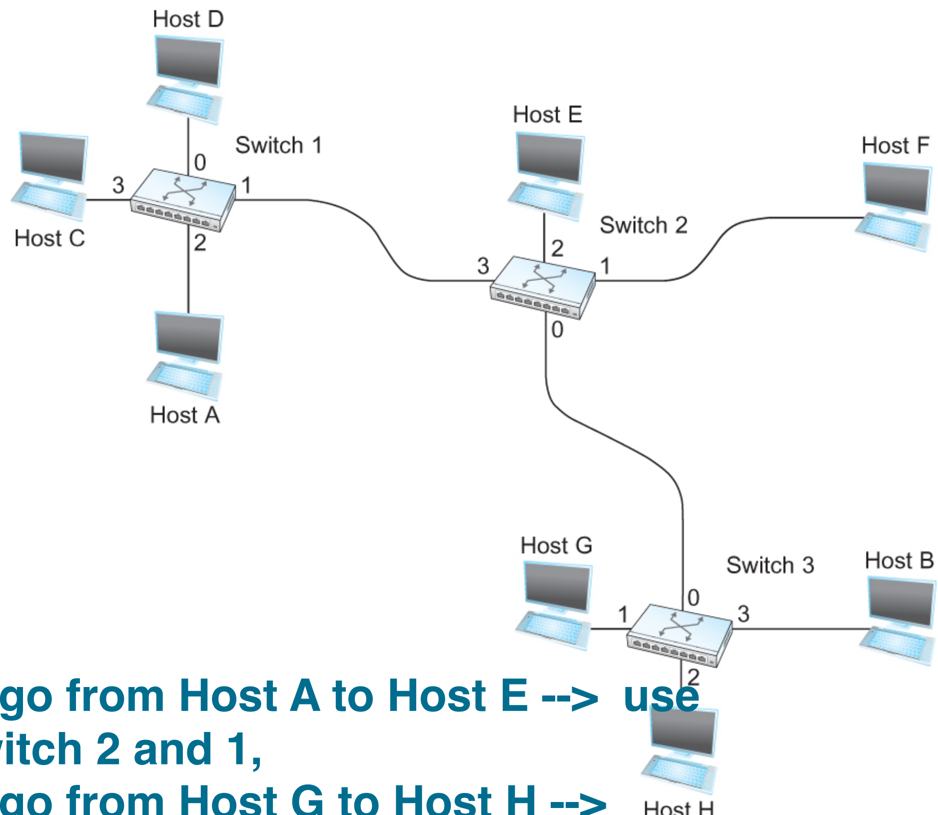
Switching and Forwarding (Datagrams/Connectionless approach)

An example network



- To decide how to forward a packet, a switch consults a *forwarding table* (sometimes called a *routing table*)

Switching and Forwarding (Datagrams/Connectionless approach)



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

Forwarding Table for Switch 2

to go from Host A to Host E --> use switch 2 and 1,
to go from Host G to Host H --> use switch 1 and 2?

each switch has each forwarding table

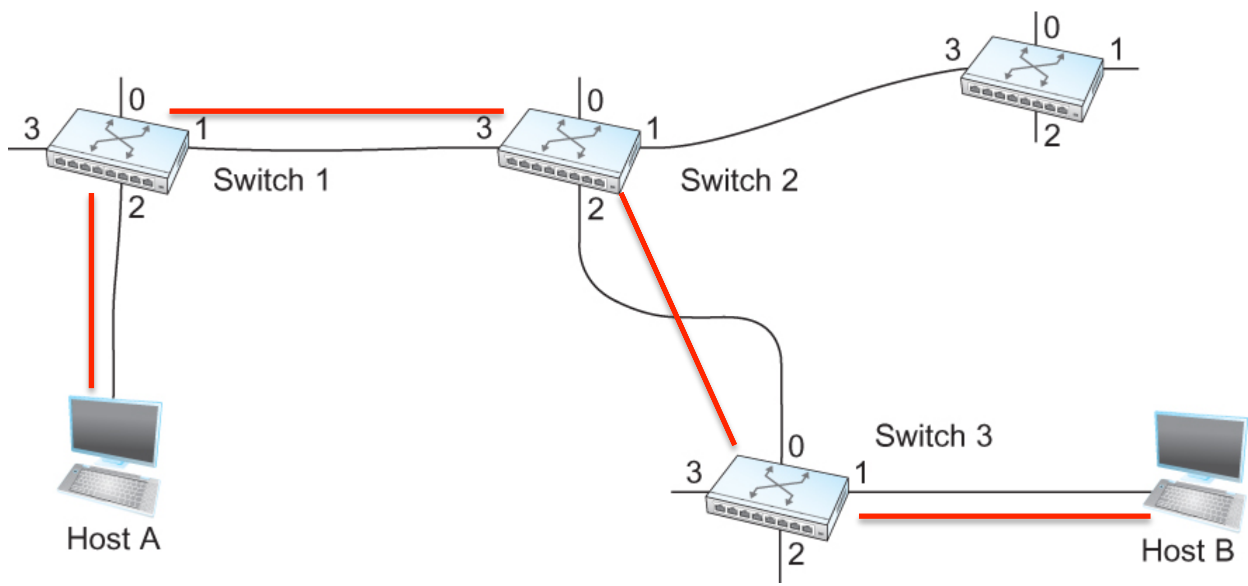
Switching and Forwarding (Datagrams/Connectionless approach)

- A host can send a packet anywhere at any time, since any packet that turns up at the switch can be immediately forwarded
  - Assuming a correctly populated forwarding table
- When a host sends a packet, it has no way of knowing if the network is capable of delivering it or if the destination host is even up and running
- Each packet is forwarded independently of previous packets that might have been sent to the same destination.
  - Thus two successive packets from host A to host B may follow completely different paths
- A switch or link failure might not have any serious effect on communication if it is possible to find an alternate route around the failure and update the forwarding table accordingly

Switching and Forwarding

Virtual Circuit Switching Approach

– Widely used technique for packet switching
  - Uses the concept of *virtual circuit* (VC)
  - Also called a connection-oriented model
  - First sets up a virtual connection from the source host to the destination host and then sends the data (two stage process)



Switching and Forwarding

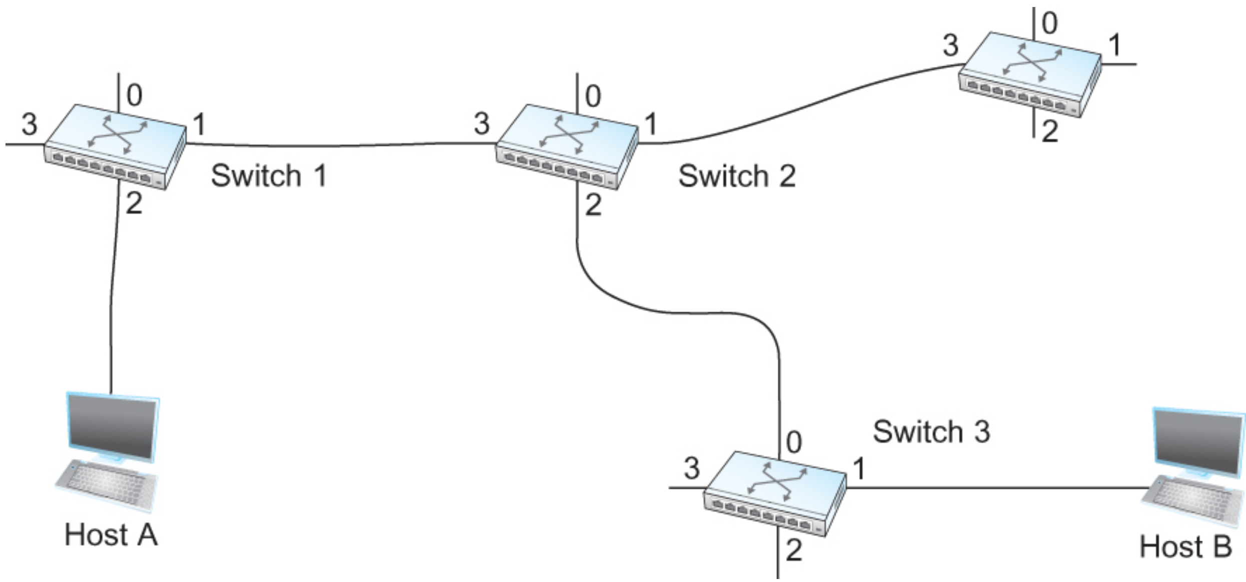
Virtual Circuit Switching Approach

- The semantics for one such entry is
  - If a packet arrives on the designated incoming interface/port and that packet contains the designated VC value in its header, then the packet should be sent out the specified outgoing port assigned to that VC with the specified outgoing VC value
  - A table at each switch with the following entries with the following fields for each entry
    - Incoming interface/port
    - Incoming VC value
    - Outgoing interface/port
    - Outgoing VC value

how to calculate them? see pg- 177 textbook

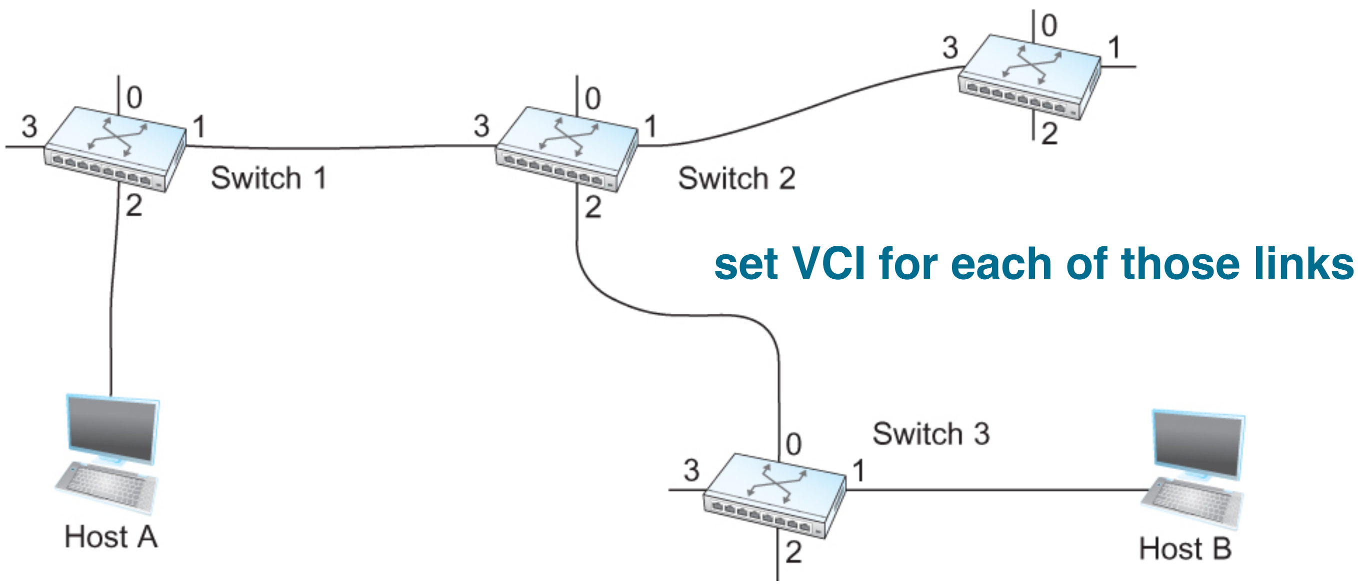
Switching and Forwarding (Virtual Circuit Switching)

Host A wants to send packets to host B



Switching and Forwarding (Virtual Circuit Switching)

Host A wants to send packets to host B



Incoming Interface	Incoming VC	Outgoing Interface	Outgoing VC
2	5	1	11

Virtual Circuit Table at Switch 1

Switching and Forwarding (Virtual Circuit Switching)

- An entry in the VC table on a single switch contains
  - A virtual circuit identifier (VCI) that uniquely identifies the connection at this switch
  - An incoming interface on which packets for this VC arrive at the switch
  - An outgoing interface in which packets for this VC leave the switch
  - A potentially different VCI that will be used for outgoing packets

Incoming Interface	Incoming VC	Outgoing Interface	Outgoing VC
2	5	1	11

Switching and Forwarding (Virtual Circuit Switching)

Note:

- The combination of the VCI of the packets as they are received at the switch and the interface on which they are received uniquely identifies the virtual connection
- There may be many virtual connections established in the switch at one time
- Incoming and outgoing VCI values are not generally the same
  - VCI is not a globally significant identifier for the connection; rather it has significance only on a given link
- Whenever a new connection is created, we need to assign a new VCI for that connection on each link that the connection will traverse
  - We also need to ensure that the chosen VCI on a given link is not currently in use on that link by some existing connection.

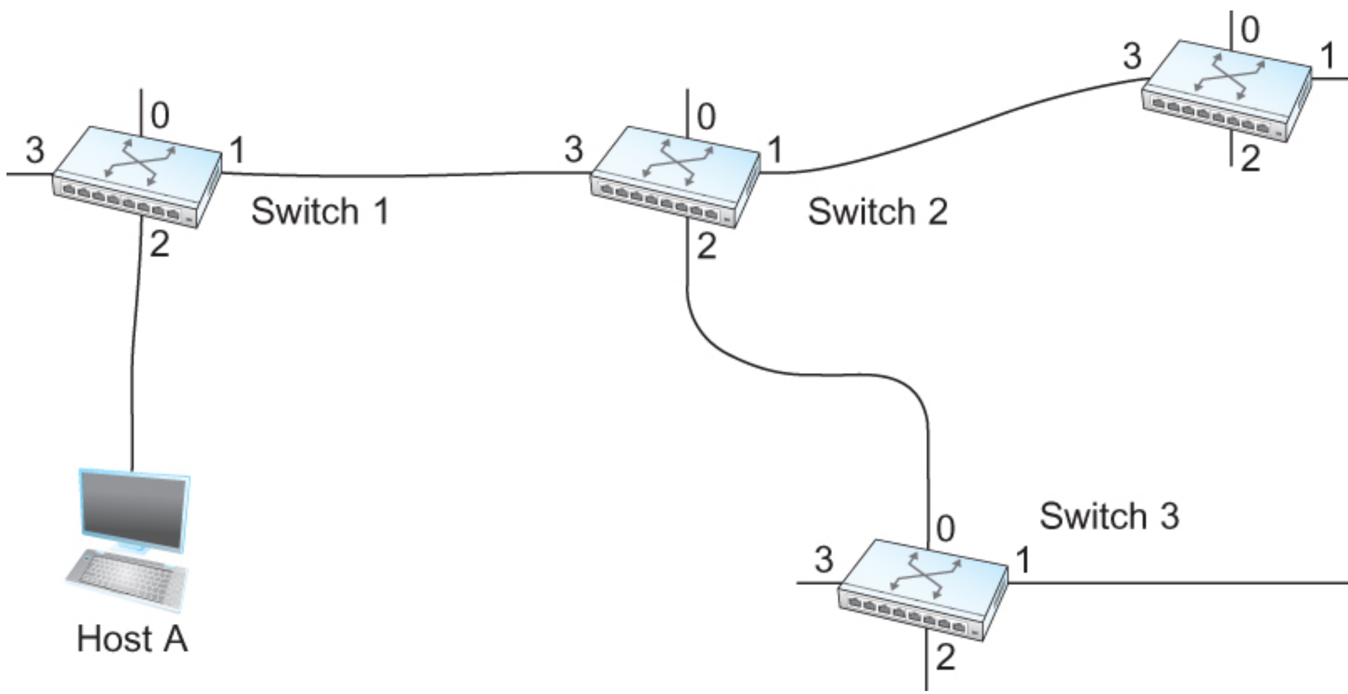
Switching and Forwarding (Virtual Circuit Switching)

- Two broad classes of approach to establishing connection state
  1. Network Administrator will configure the state
  2. A host can send messages into the network to cause the state to be established

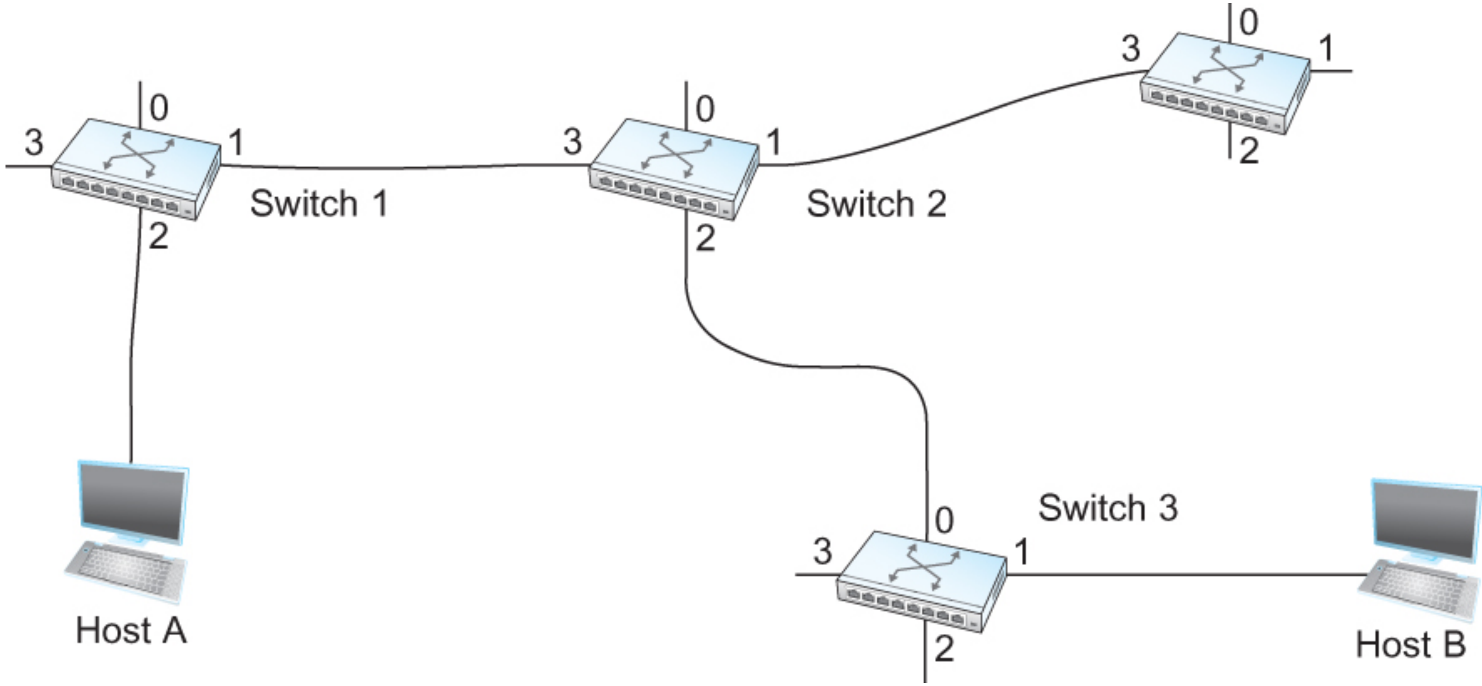
Switching and Forwarding (Virtual Circuit Switching)

Let's assume that a network administrator wants to manually create a new virtual connection from host A to host B

- First the administrator identifies a path through the network from A to B



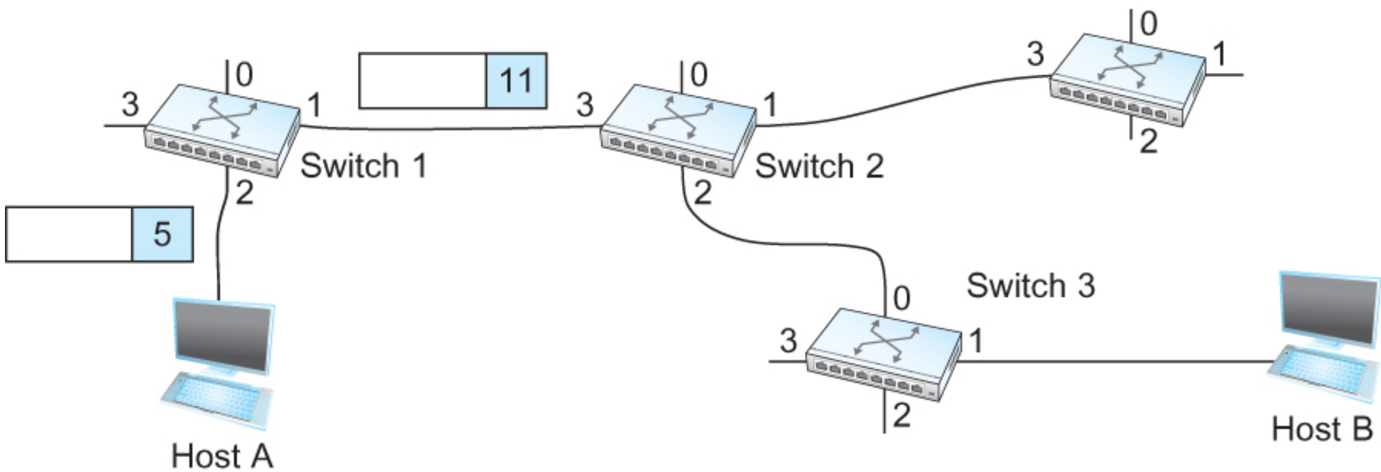
Switching and Forwarding (Virtual Circuit Switching)



Incoming Interface	Incoming VC	Outgoing Interface	Outgoing VC
2	5	1	11

Connection state on switch 1

Switching and Forwarding (Virtual Circuit Switching)

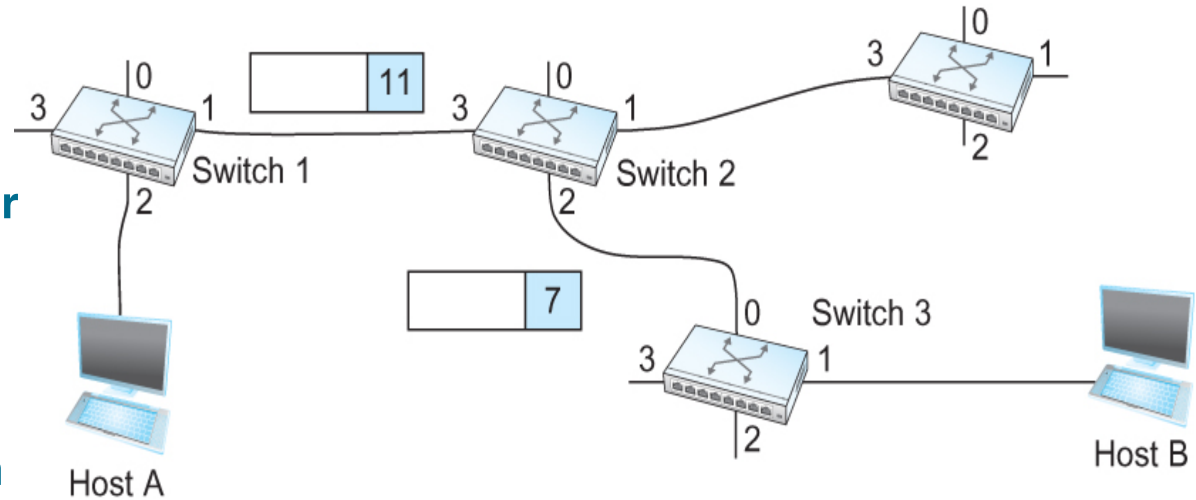


Incoming Interface	Incoming VC	Outgoing Interface	Outgoing VC
2	5	1	11

Connection state on switch 1

Switching and Forwarding (Virtual Circuit Switching)

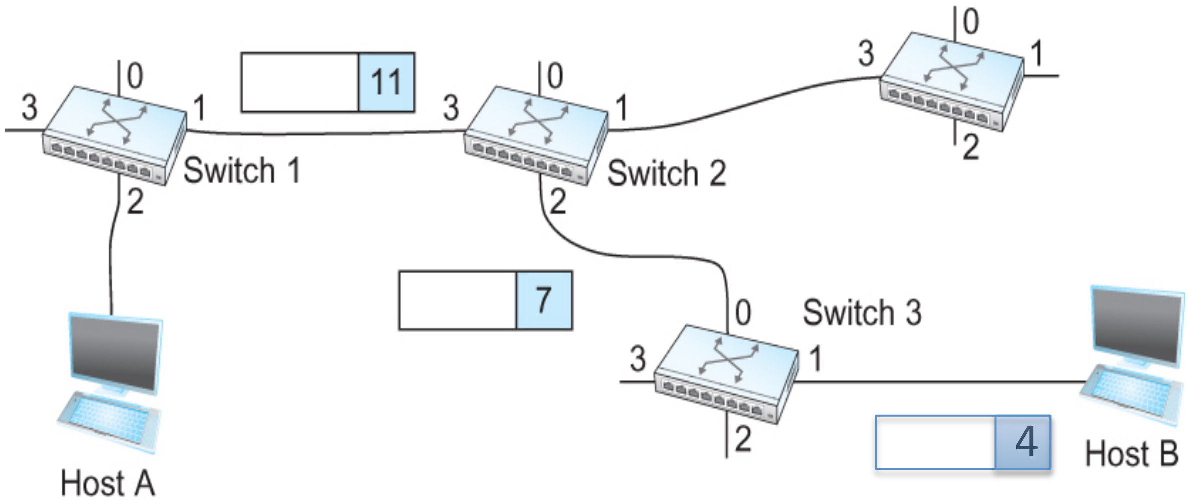
Each switch can pick any number it likes, as long as that number is not currently in use for some other connection on that port of that switch.



Incoming Interface	Incoming VC	Outgoing Interface	Outgoing VC
3	11	2	7

Connection state on switch 2

Switching and Forwarding (Virtual Circuit Switching)



Incoming Interface	Incoming VC	Outgoing Interface	Outgoing VC
0	7	1	4

Connection state on switch 3

Switching and Forwarding (Virtual Circuit Switching)

- Configuring VC tables correctly in a large number of switches not an easy task
  - Thus, some sort of signaling is almost always used, even when setting up "permanent" VCs
  - In case of PVCs, signaling is initiated by the network administrator
  - SVCs are usually set up using signaling by one of the hosts

Switching and Forwarding (Virtual Circuit Switching)

- How does the signalling work
  - To start the signalling process, host A sends a setup message into the network (i.e. to switch 1)
    - The setup message contains (among other things) the complete destination address of B.
    - The setup message needs to get all the way to B to create the necessary connection state in every switch along the way
    - It is like sending a datagram to B where every switch knows which output to send the setup message so that it eventually reaches B
    - Assume that every switch knows the topology to figure out how to do that
  - When switch 1 receives the connection request, in addition to sending it on to switch 2, it creates a new entry in its VC table for this new connection
    - The entry is exactly the same shown in the previous table
    - Switch 1 picks the value 5 for this connection

Switching and Forwarding (Virtual Circuit Switching)

- How does the signalling work (contd.)
  - When switch 2 receives the setup message, it performs the similar process and it picks the value 11 as the incoming VCI
  - Similarly switch 3 picks 7 as the value for its incoming VCI
    - Each switch can pick any number it likes, as long as that number is not currently in use for some other connection on that port of that switch
  - Finally the setup message arrives at host B.
  - Assuming that B is healthy and willing to accept a connection from host A, it allocates an incoming VCI value, in this case 4.
    - This VCI value can be used by B to identify all packets coming from A

Switching and Forwarding (Virtual Circuit Switching)

- Now to complete the connection, everyone needs to be told what their downstream neighbor is using as the VCI for this connection
  - Host B sends an acknowledgement of the connection setup to switch 3 and includes in that message the VCI value that it chose (4)
  - Switch 3 completes the VC table entry for this connection and sends the acknowledgement on to switch 2 specifying the VCI of 7
  - Switch 2 completes the VC table entry for this connection and sends acknowledgement on to switch 1 specifying the VCI of 11
  - Finally switch 1 passes the acknowledgement on to host A telling it to use the VCI value of 5 for this connection

Virtual Circuit Identifier

Switching and Forwarding (Virtual Circuit Switching)

- When host A no longer wants to send data to host B, it tears down the connection by sending a teardown message to switch 1
- The switch 1 removes the relevant entry from its table and forwards the message on to the other switches in the path which similarly delete the appropriate table entries
- At this point, if host A were to send a packet with a VCI of 5 to switch 1, it would be dropped as if the connection had never existed

Switching and Forwarding (Virtual Circuit Switching)

- Characteristics of VC
  - There is at least one RTT of delay before data is sent
  - Per-packet overhead caused by the header is reduced relative to the datagram model (higher transmission speed)
  - If a switch or a link in a connection fails, the connection is broken and a new one will need to be established.
  - The link selection decision to forward the connection request has similarities with the function of a routing algorithm.

Switching and Forwarding (Virtual Circuit Switching)

- Good Properties of VC
  - By the time the host gets the go-ahead to send data, it knows quite a lot about the network-
  - It is also possible to allocate resources to the virtual circuit at the time it is established

Switching and Forwarding (Virtual Circuit Switching)

- Comparison with the Datagram Model
  - Datagram network has no connection establishment phase and each switch processes each packet independently
  - Each arriving packet competes with all other packets for buffer space
  - If there are no buffers, the incoming packet must be dropped
- In VC, we could imagine providing each circuit with a different quality of service (QoS)
  - The network gives the user some kind of performance related guarantee
    - Switches set aside the resources they need to meet this guarantee
      - For example, a percentage of each outgoing link's bandwidth
      - Delay tolerance on each switch
- Most popular examples of VC technologies are Frame Relay and ATM
  - One of the applications of Frame Relay is the construction of VPN

Switching and Forwarding (Virtual Circuit Switching)

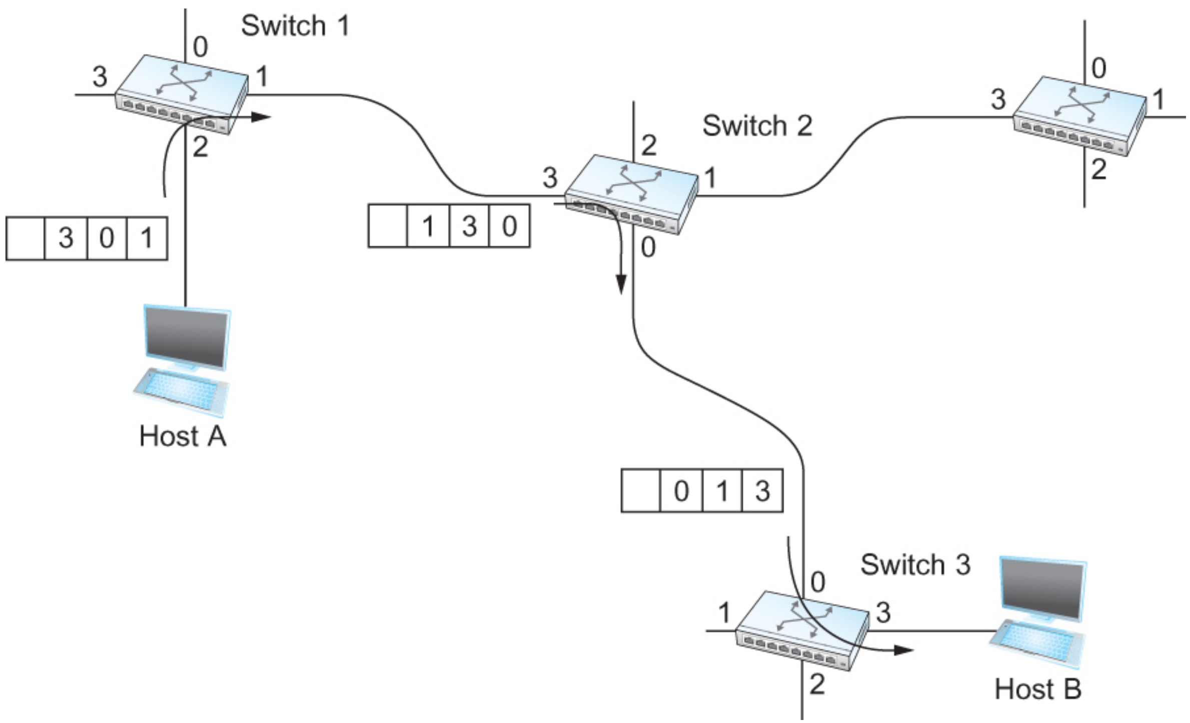
- ATM (Asynchronous Transfer Mode)
  - Connection-oriented packet-switched network
  - Packets are called cells (fixed size each)
    - 5 byte header + 48 byte payload = 53-bytes ATM cell
  - Contain only very basic routing info (since uses VC switching – path is already known and set)
    - Therefore high transmission speed (speech, video, graphics, audio, etc.)
  - Fixed length packets are easier to switch in hardware
    - Simpler to design
    - Enables parallelism

Switching and Forwarding (Source Routing)

How does the switch decide which output port to place each packet on?

3rd approach less common one is

- Source Routing
  - All the information about network topology that is required to switch a packet across the network is provided by the source host



Switching and Forwarding (Source Routing)

- Other approaches in Source Routing