Internetworking
Lecture _2

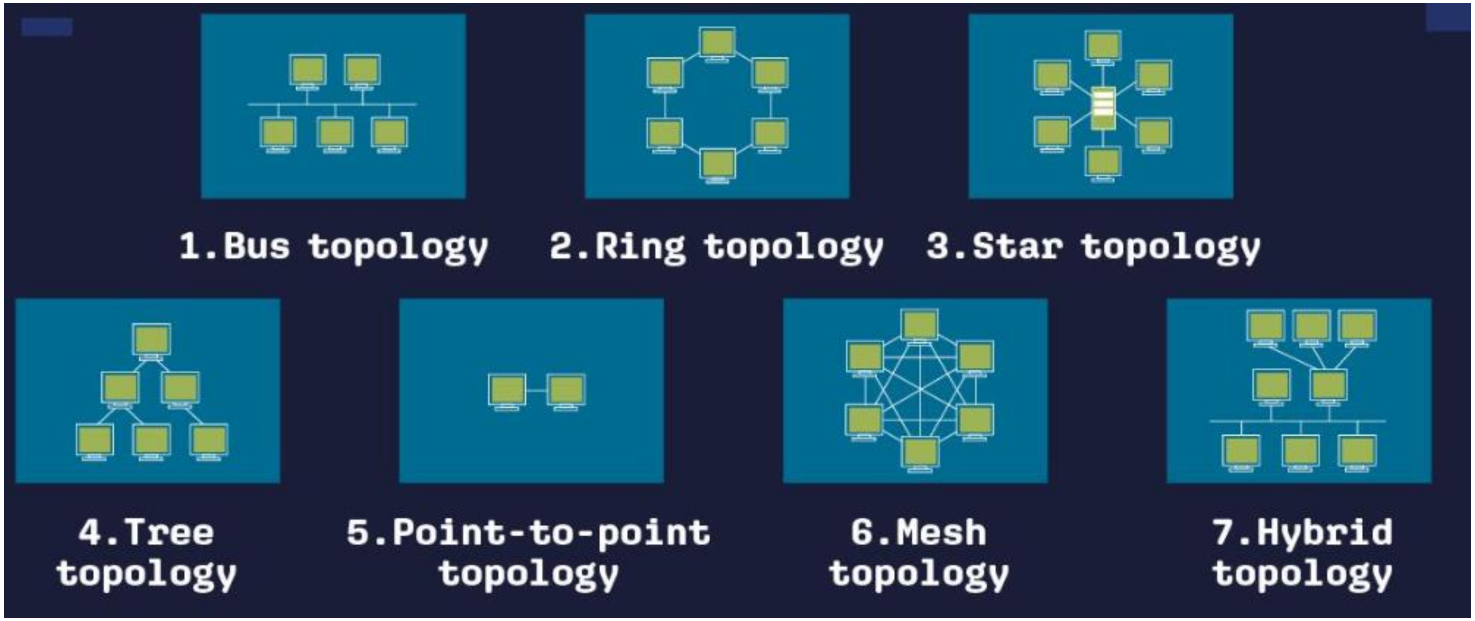
أ.م.د.عباس عبد العزيز عبد الحميد

كليـة العلـوم / قسم الحاسـوب

abbasabdulazeez@uomustansiriyah.edu.iq

What is network topology?

- Network topology refers to the physical or logical arrangement of nodes (such as computers, switches, routers, or other devices) and the connections between them in a computer network.
- The Role of Topology in Network Management By understanding the network topology, network administrators and engineers can plan, manage, and troubleshoot the network effectively. It helps them identify potential bottlenecks, optimize performance, and ensure reliable communication between devices.

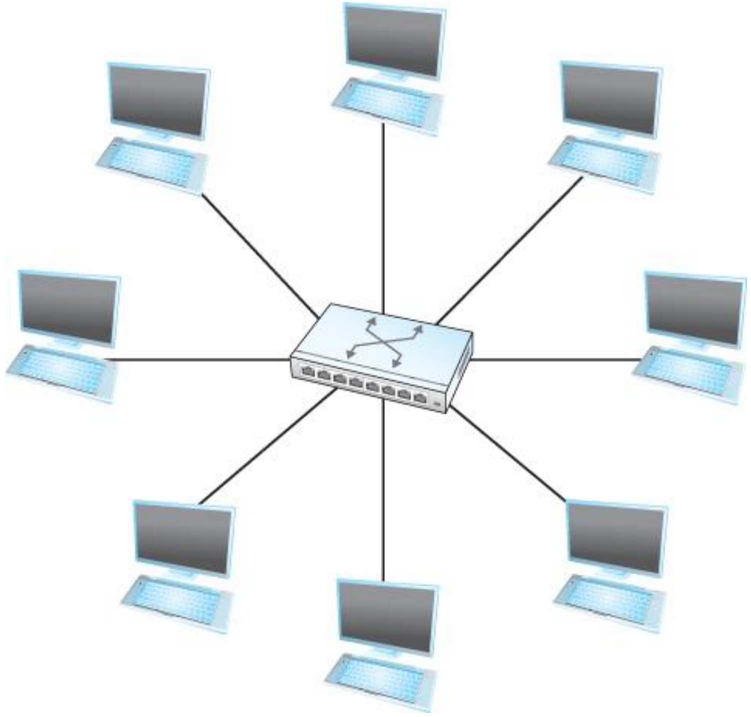


Switching and Forwarding

- Switch
  - A mechanism that allows us to interconnect links to form a large network
  - A multi-input, multi-output device which transfers packets from an input to one or more outputs

Switching and Forwarding

Adds the star topology to the point-to-point link, bus (Ethernet), and ring (802.5 and FDDI) topologies



Switching and Forwarding

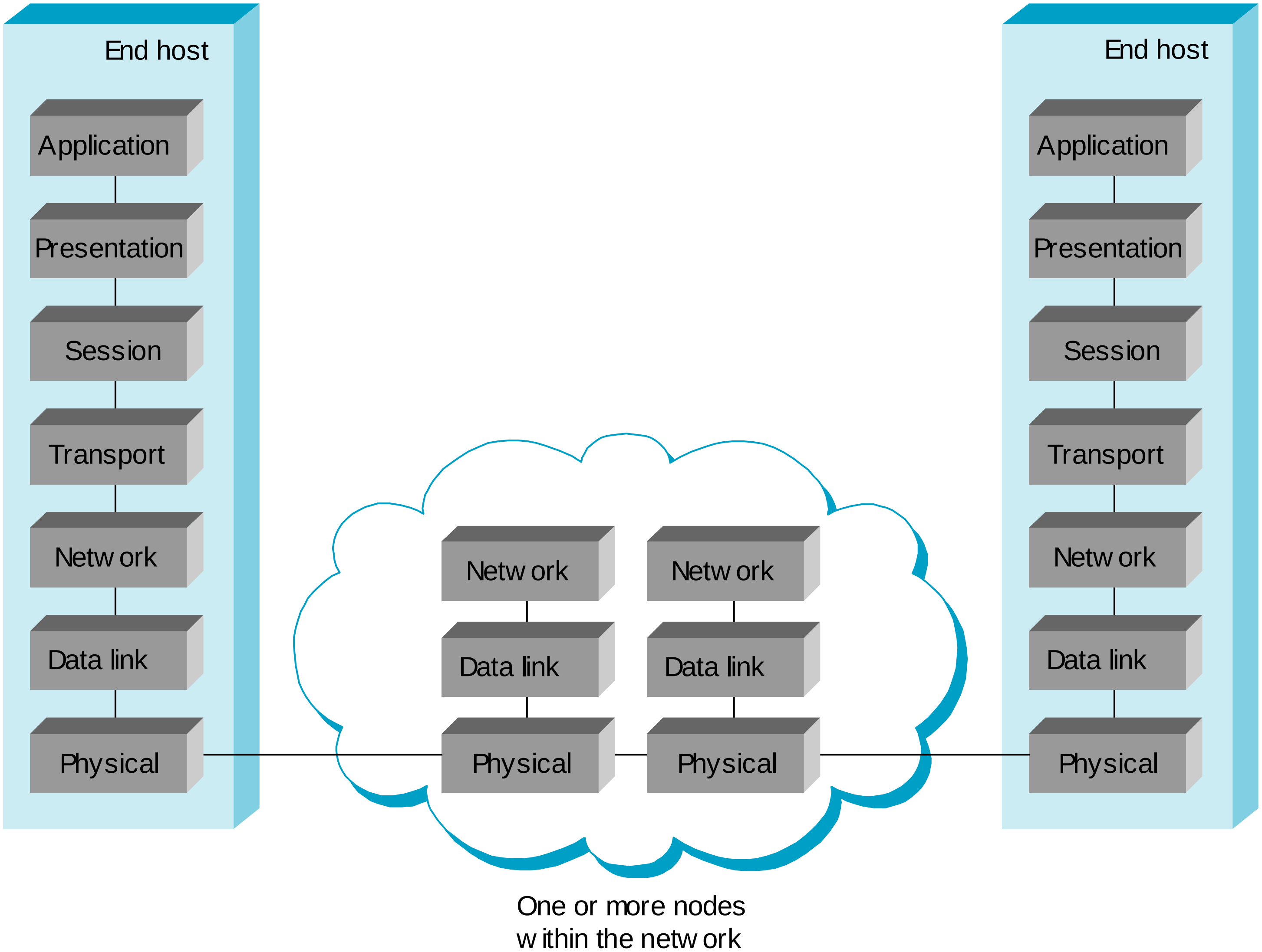
Properties of this star topology

- Even though a switch has a fixed number of inputs and outputs, which limits the number of hosts that can be connected to a single switch, large networks can be built by interconnecting a number of switches
- We can connect switches to each other and to hosts using point-to-point links, which typically means that we can build networks of large geographic scope
- Adding a new host to the network by connecting it to a switch does not necessarily mean that the hosts already connected will get worse performance from the network

Switching and Forwarding

- A switch is connected to a set of links and for each of these links, runs the appropriate data link protocol to communicate with that node
- A switch's primary job is to receive incoming packets on one of its links and to transmit them on some other link
  - This function is referred as *switching and forwarding*
  - According to OSI architecture this is the main function of the network layer

ISO Architecture



OSI protocol summary

Layer	*Description*	*Examples*
Application	Protocols that are designed to meet the communication requirements of specific applications, often defining the interface to a service.	HTTP, FTP, SMTP, CORBA IIOP
Presentation	Protocols at this level transmit data in a network representation that is independent of the representations used in individual computers, which may differ. Encryption is also performed in this layer, if required.	Secure Sockets (SSL),CORBA Data Rep.
Session	At this level reliability and adaptation are performed, such as detection of failures and automatic recovery.	
Transport	This is the lowest level at which messages (rather than packets) are handled. Messages are addressed to communication ports attached to processes, Protocols in this layer may be connection-oriented or connectionless.	TCP, UDP
Network	Transfers data packets between computers in a specific network. In a WAN or an internetwork this involves the generation of a route passing through routers. In a single LAN no routing is required.	IP, ATM virtual circuits
Data link	Responsible for transmission of packets between nodes that are directly connected by a physical link. In a WAN transmission is between pairs of routers or between routers and hosts. In a LAN it is between any pair of hosts.	Ethernet MAC, ATM cell transfer, PPP
Physical	The circuits and hardware that drive the network. It transmits sequences of binary data by analogue signalling, using amplitude or frequency modulation of electrical signals (on cable circuits), light signals (on fibre optic circuits) or other electromagnetic signals (on radio and microwave circuits).	Ethernet base- band signalling, ISDN

Switching and Forwarding

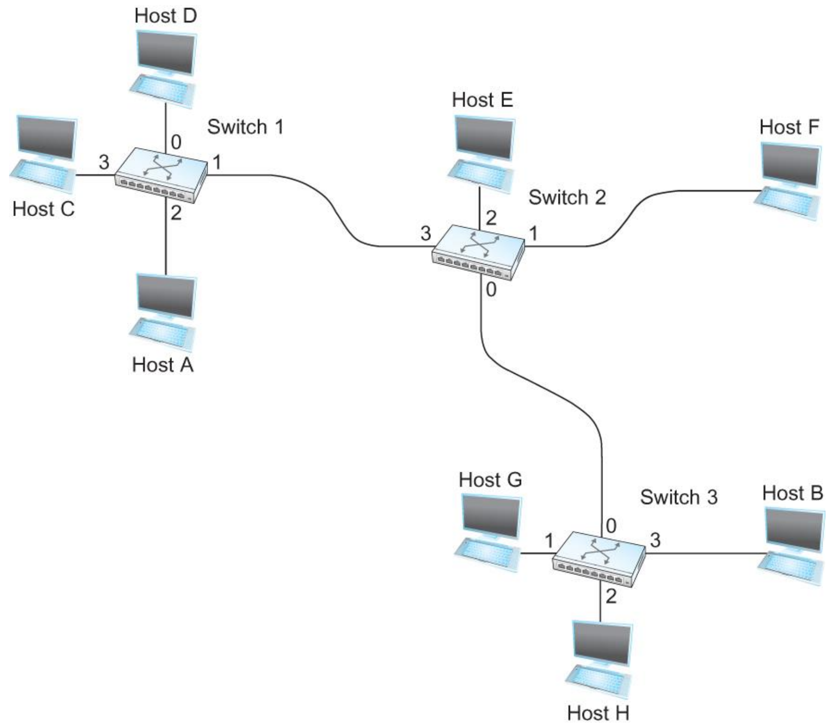
- How does the switch decide which output port to place each packet on?
  - It looks at the header of the packet for an identifier that it uses to make the decision
  - Two common approaches
    - *Datagram or Connectionless approach*
    - *Virtual circuit or Connection-oriented approach*
  - A third approach *source routing* is less common

Switching and Forwarding

- Datagrams
  - Key Idea
    - Every packet contains enough information to enable any switch to decide how to get it to destination
      - Every packet contains the complete destination address

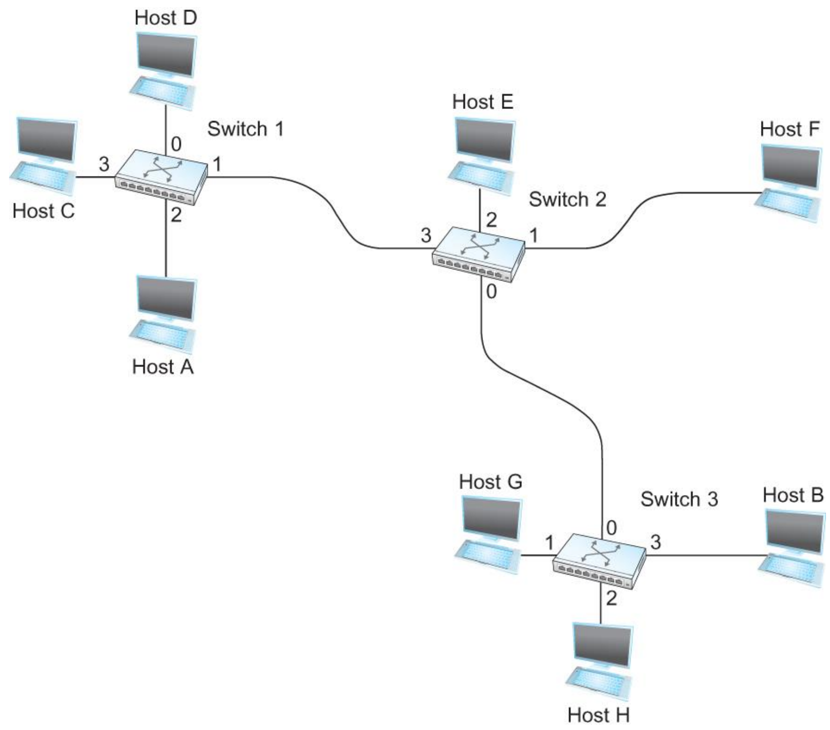
Switching and Forwarding

An example network



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

Switching and Forwarding



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

Forwarding Table for Switch 2

Switching and Forwarding

Characteristics of Connectionless (Datagram) Network

- 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

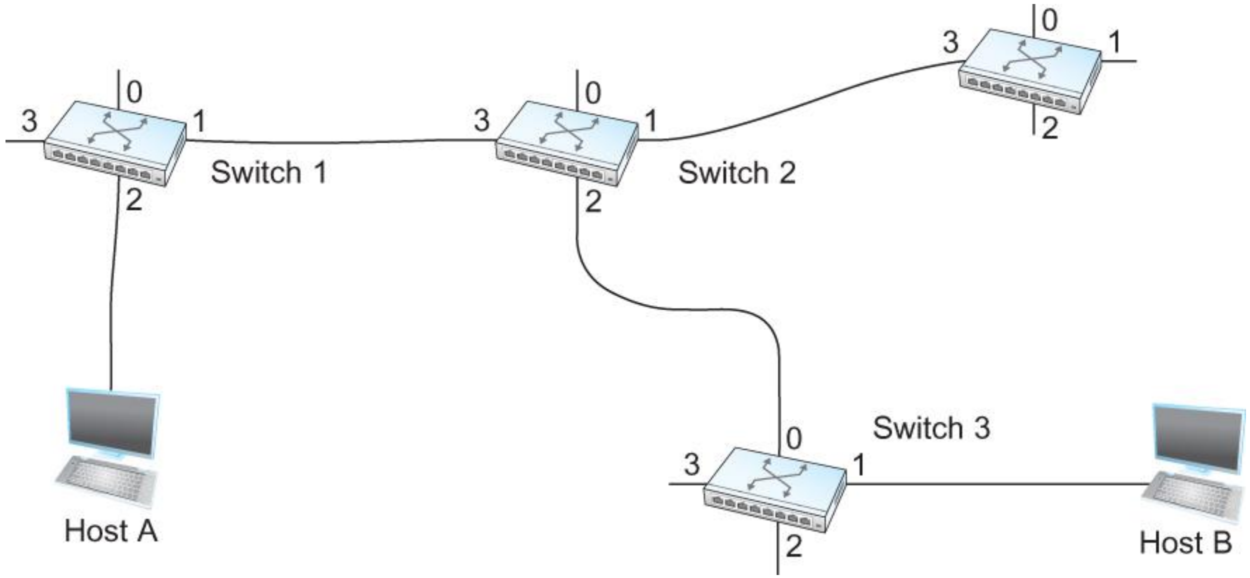
Switching and Forwarding

Virtual Circuit Switching

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

Switching and Forwarding

- Host A wants to send packets to host B



Switching and Forwarding

Two-stage process

- Connection setup
- Data Transfer

- Connection setup
  - Establish “connection state” in each of the switches between the source and destination hosts
  - The connection state for a single connection consists of an entry in the “VC table” in each switch through which the connection passes

Switching and Forwarding

One entry in the VC table on a single switch contains

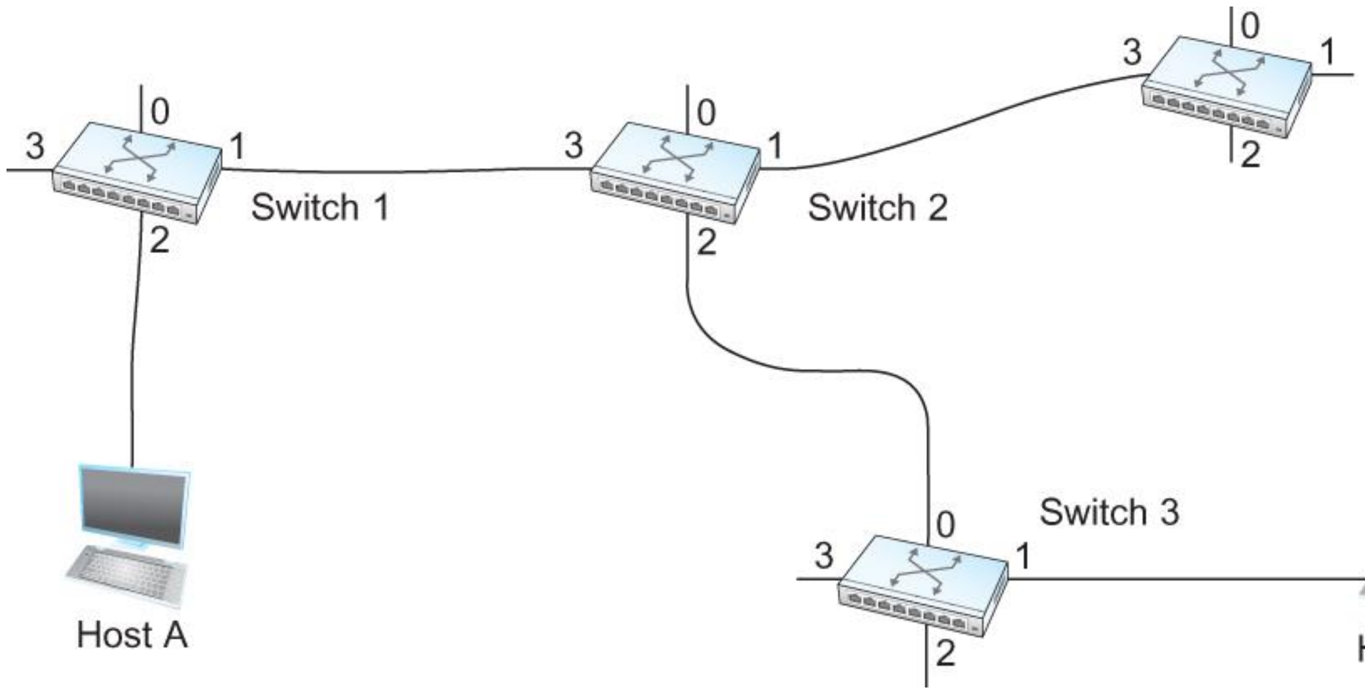
- A virtual circuit identifier (VCI) that uniquely identifies the connection at this switch and that will be carried inside the header of the packets that belong to this connection
- An incoming interface on which packets for this VC arrive at the switch
- An outgoing interface in which packets for this VC leave the switch
- A potentially different VCI that will be used for outgoing packets

- The semantics for one such entry is
  - If a packet arrives on the designated incoming interface and that packet contains the designated VCI value in its header, then the packet should be sent out the specified outgoing interface with the specified outgoing VCI value first having been placed in its header

Switching and Forwarding

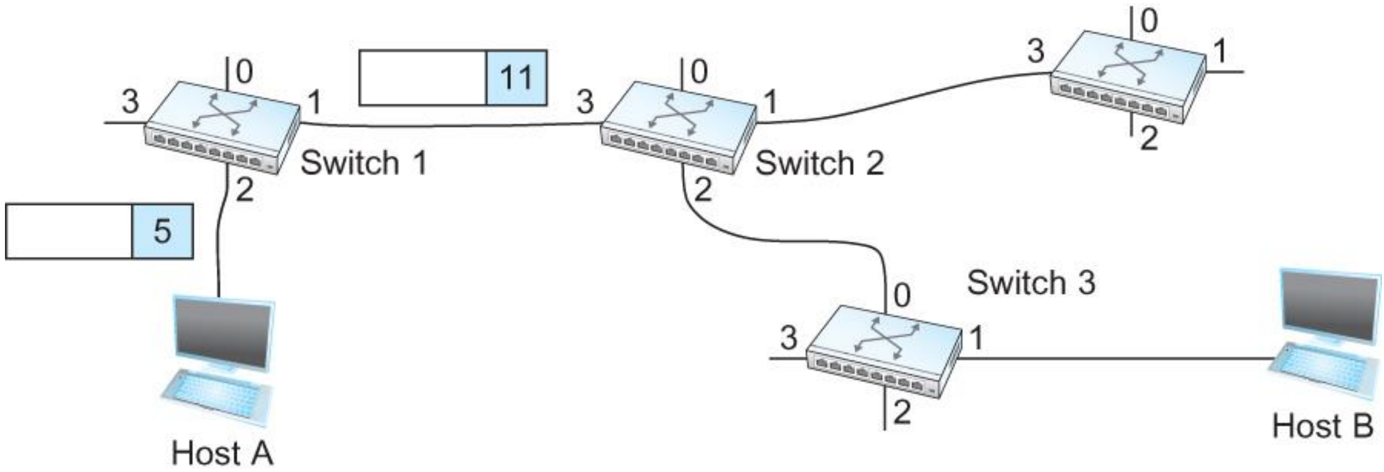
Let's assume that a network administrator wants to manually create a new virtual connection from host A to host B

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



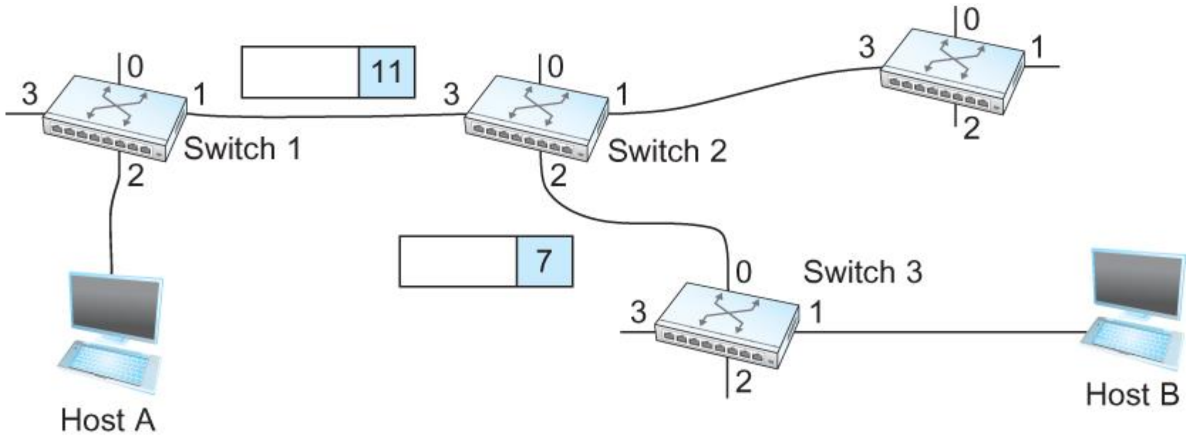
Switching and Forwarding

- For any packet that A wants to send to B, A puts the VCI value 5 in the header of the packet and sends it to switch 1
- Switch 1 receives any such packet on interface 2, and it uses the combination of the interface and the VCI in the packet header to find the appropriate VC table entry.
- The table entry on switch 1 tells the switch to forward the packet out of interface 1 and to put the VCI value 11 in the header



Switching and Forwarding

- Packet will arrive at switch 2 on interface 3 bearing VCI 11
- Switch 2 looks up interface 3 and VCI 11 in its VC table and sends the packet on to switch 3 after updating the VCI value appropriately
- This process continues until it arrives at host B with the VCI value of 4 in the packet
- To host B, this identifies the packet as having come from host A



Switching and Forwarding

- In real networks of reasonable size, the burden of configuring VC tables correctly in a large number of switches would quickly become excessive
  - Thus, some sort of signalling is almost always used, even when setting up “permanent” VCs
  - In case of PVCs, signalling is initiated by the network administrator
  - SVCs are usually set up using signalling by one of the hosts

Switching and Forwarding

- 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

Switching and Forwarding

- Good Properties of VC
  - By the time the host gets the go-ahead to send data, it knows quite a lot about the network-
    - For example, that there is really a route to the receiver and that the receiver is willing to receive data
  - It is also possible to allocate resources to the virtual circuit at the time it is established

Switching and Forwarding

- ATM (Asynchronous Transfer Mode)
  - Connection-oriented packet-switched network
  - Packets are called cells
    - 5 byte header + 48 byte payload
  - Fixed length packets are easier to switch in hardware
    - Simpler to design
    - Enables parallelism

Switching and Forwarding

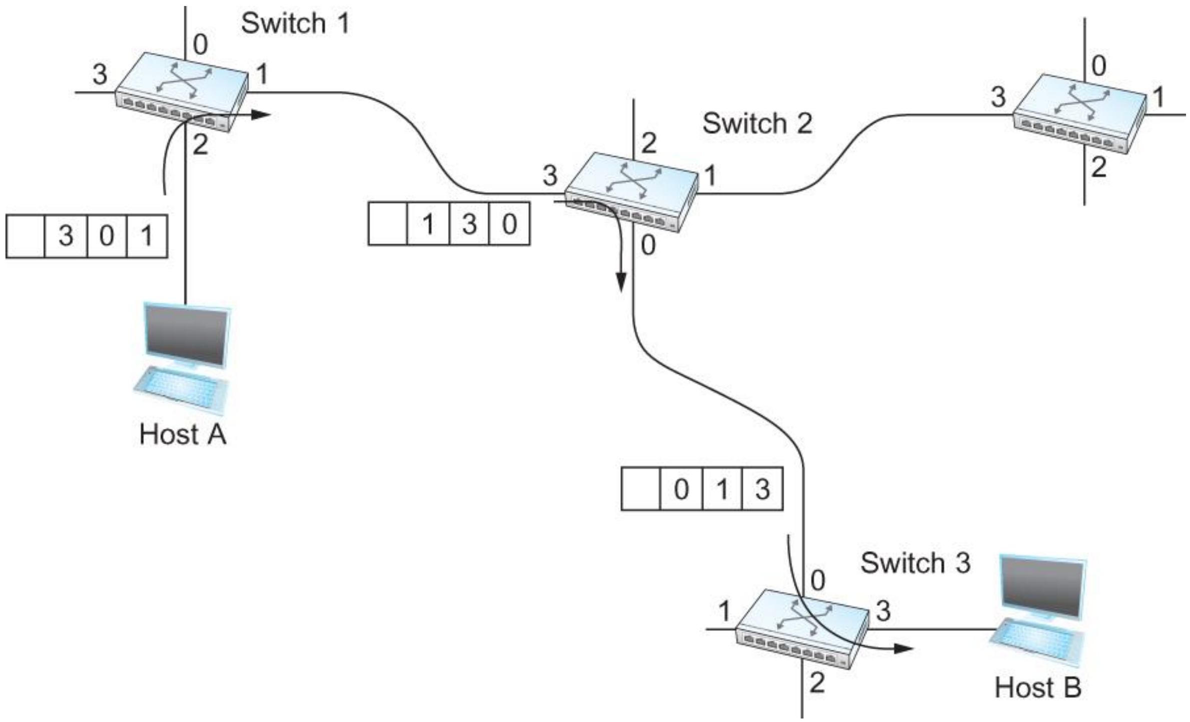
- ATM
  - User-Network Interface (UNI)
    - Host-to-switch format
    - GFC: Generic Flow Control
    - VCI: Virtual Circuit Identifier
    - Type: management, congestion control
    - CLP: Cell Loss Priority
    - HEC: Header Error Check (CRC-8)

4	8	16	3	1	8	384 (48 bytes)
GFC	VPI	VCI	Type	CLP	HEC (CRC-8)	Payload

  - Network-Network Interface (NNI)
    - Switch-to-switch format
    - GFC becomes part of VPI field

Switching and Forwarding

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



Bridges and LAN Switches

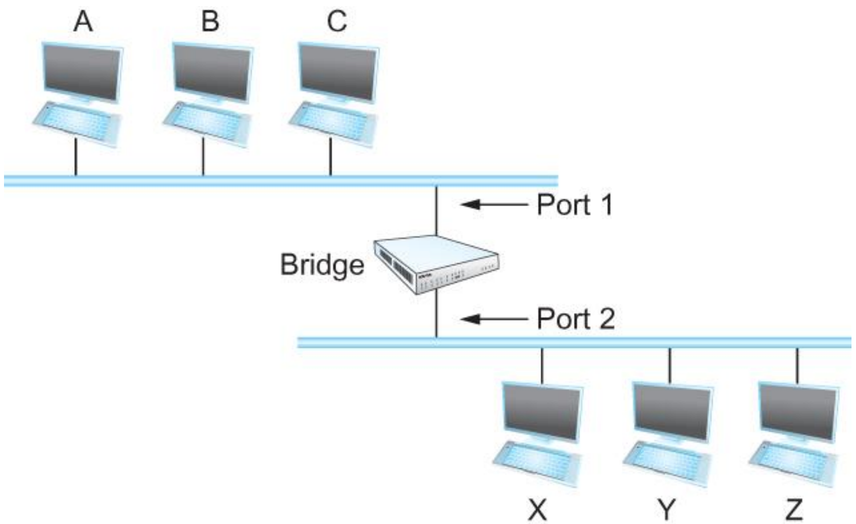
- Bridges and LAN Switches
  - Class of switches that is used to forward packets between shared-media LANs such as Ethernets
    - Known as LAN switches
    - Referred to as Bridges
  - Suppose you have a pair of Ethernets that you want to interconnect
    - One approach is put a repeater in between them
      - It might exceed the physical limitation of the Ethernet
        - No more than four repeaters between any pair of hosts
        - No more than a total of 2500 m in length is allowed
    - An alternative would be to put a node between the two Ethernets and have the node forward frames from one Ethernet to the other
      - This node is called a **Bridge**
      - A collection of LANs connected by one or more bridges is usually said to form an **Extended LAN**

Bridges and LAN Switches

- Simplest Strategy for Bridges
  - Accept LAN frames on their inputs and forward them out to all other outputs
  - Used by early bridges
- Learning Bridges
  - Observe that there is no need to forward all the frames that a bridge receives

Bridges and LAN Switches

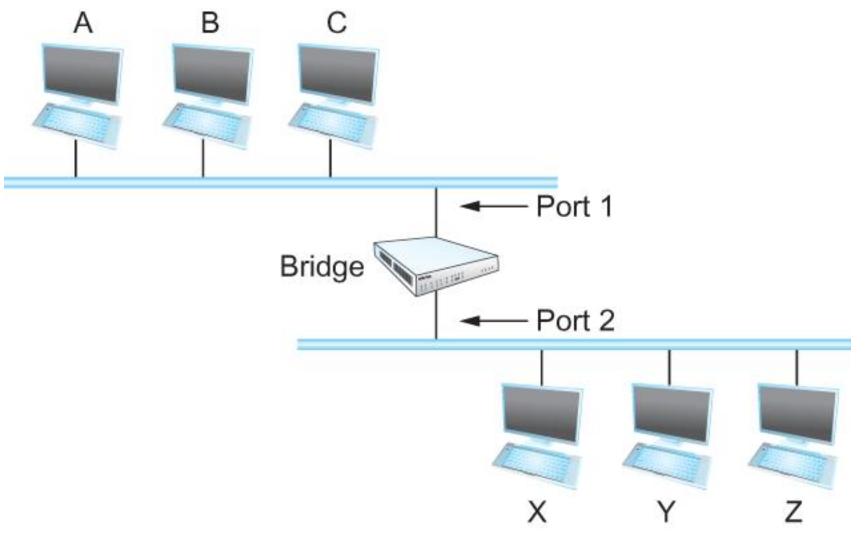
- Consider the following figure
  - When a frame from host A that is addressed to host B arrives on port 1, there is no need for the bridge to forward the frame out over port 2.



  - How does a bridge come to learn on which port the various hosts reside?

Bridges and LAN Switches

- Solution
  - Download a table into the bridge



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

  - Who does the download?
    - Human
      - Too much work for maintenance

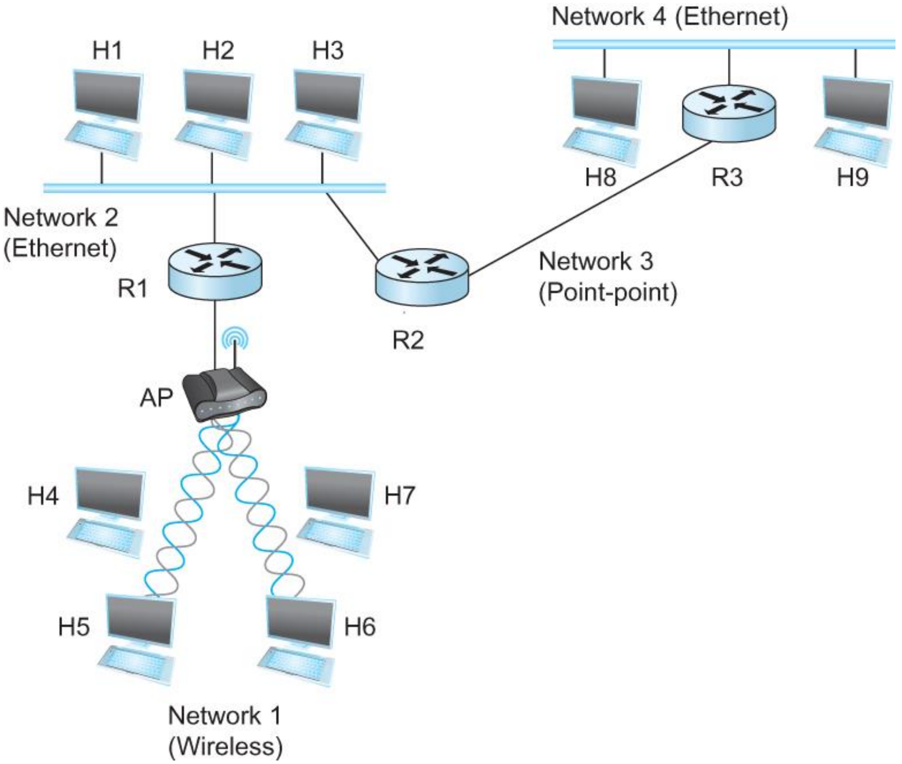
Bridges and LAN Switches

- Can the bridge learn this information by itself?
  - Yes

- How
  - Each bridge inspects the source address in all the frames it receives
  - Record the information at the bridge and build the table
  - When a bridge first boots, this table is empty
  - Entries are added over time
  - A timeout is associated with each entry
  - The bridge discards the entry after a specified period of time
    - To protect against the situation in which a host is moved from one network to another
- If the bridge receives a frame that is addressed to host not currently in the table
  - Forward the frame out on all other ports

Internetworking

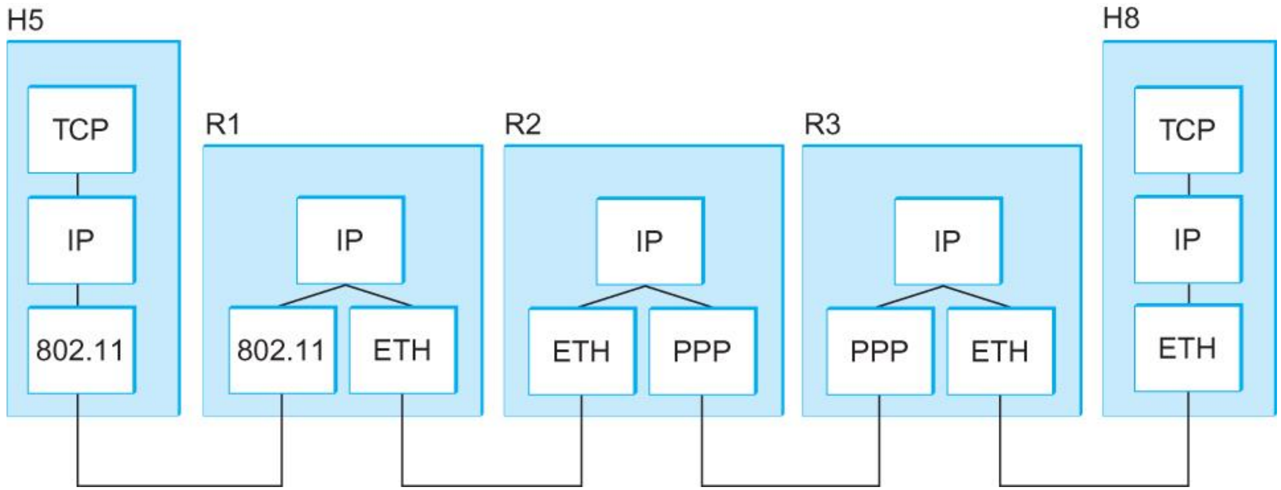
- What is internetwork
  - An arbitrary collection of networks interconnected to provide some sort of host-host to packet delivery service



A simple internetwork where H represents hosts and R represents routers

Internetworking

- What is IP
  - IP stands for Internet Protocol
  - Key tool used today to build scalable, heterogeneous internetworks
  - It runs on all the nodes in a collection of networks and defines the infrastructure that allows these nodes and networks to function as a single logical internetwork



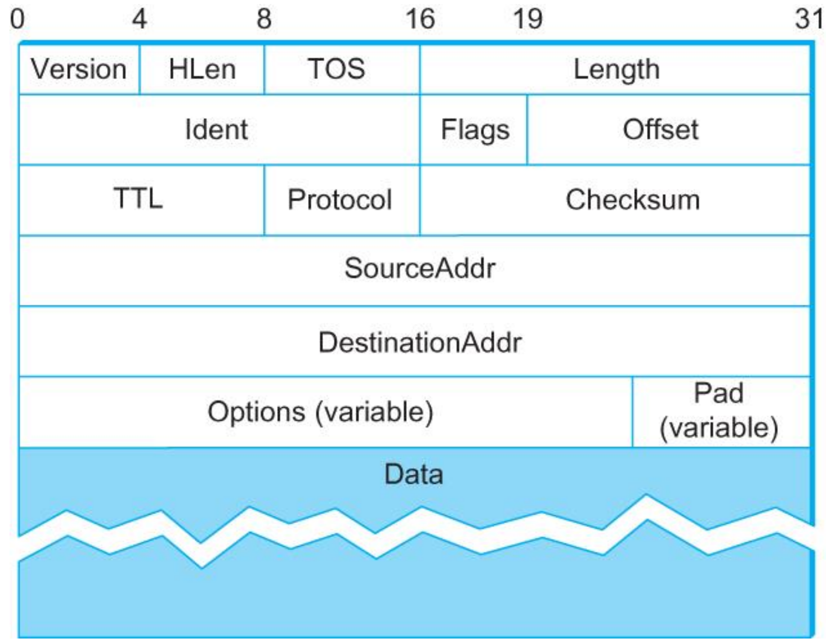
A simple internetwork showing the protocol layers

IP Service Model

- Packet Delivery Model
  - Connectionless model for data delivery
  - Best-effort delivery (unreliable service)
    - packets are lost
    - packets are delivered out of order
    - duplicate copies of a packet are delivered
    - packets can be delayed for a long time
- Global Addressing Scheme
  - Provides a way to identify all hosts in the network

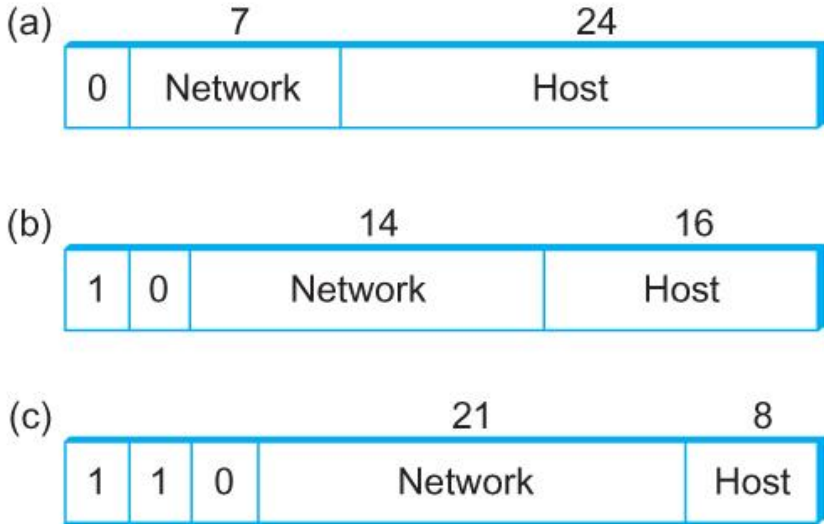
Packet Format

- Version (4): currently 4
- Hlen (4): number of 32-bit words in header
- TOS (8): type of service (not widely used)
- Length (16): number of bytes in this datagram
- Ident (16): used by fragmentation
- Flags/Offset (16): used by fragmentation
- TTL (8): number of hops this datagram has traveled
- Protocol (8): demux key (TCP=6, UDP=17)
- Checksum (16): of the header only
- DestAddr & SrcAddr (32)



Global Addresses

- Properties
  - globally unique
  - hierarchical: network + host
  - 4 Billion IP address, half are A type, ¼ is B type, and 1/8 is C type
- Format



- Dot notation
  - 10.3.2.4
  - 128.96.33.81
  - 192.12.69.77

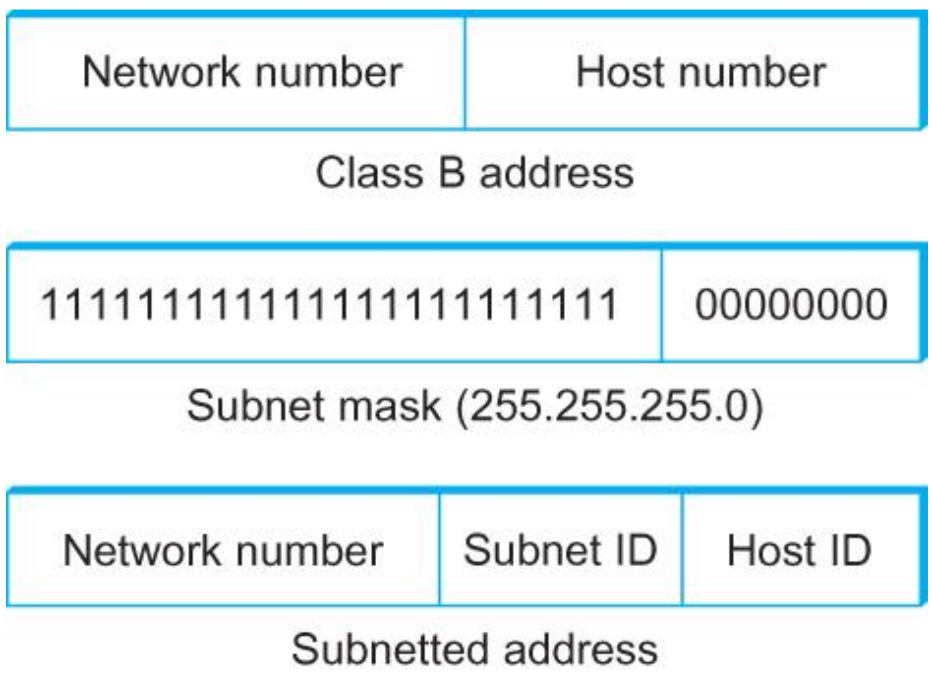
IP Datagram Forwarding

- Strategy
  - every datagram contains destination's address
  - if directly connected to destination network, then forward to host
  - if not directly connected to destination network, then forward to some router
  - forwarding table maps network number into next hop
  - each host has a default router
  - each router maintains a forwarding table
- Example (router R2)

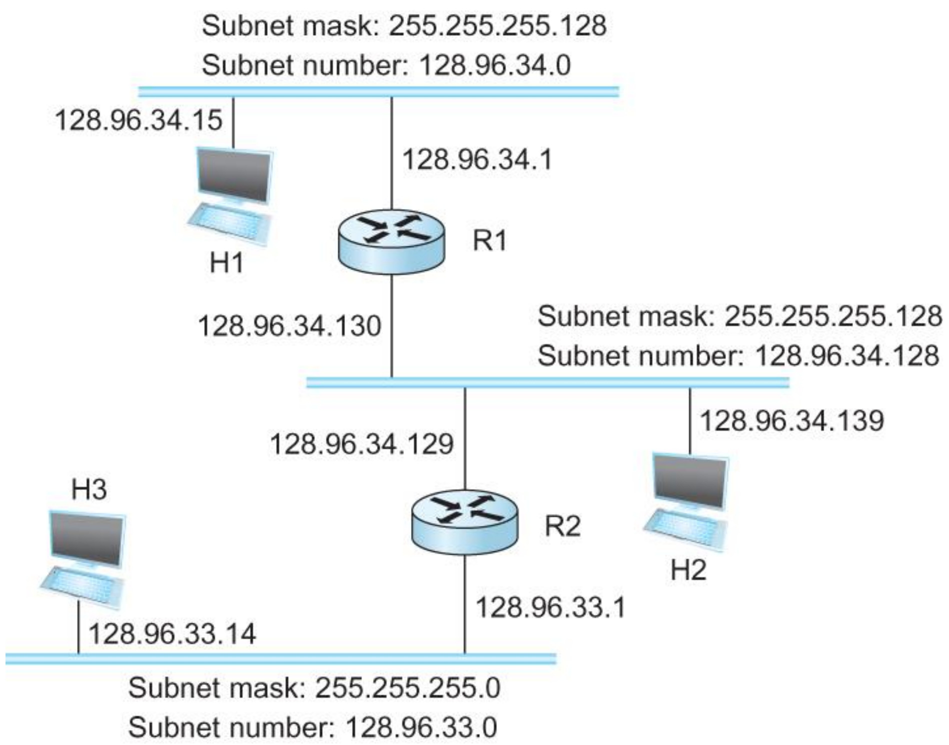
NetworkNum	NextHop
1	R1
2	Interface 1
3	Interface 0
4	R3

Subnetting

- Add another level to address/routing hierarchy: *subnet*
- *Subnet masks* define variable partition of host part of class A and B addresses
- Subnets visible only within site



Subnetting



- Forwarding Table at Router R1

SubnetNumber	SubnetMask	NextHop
128.96.34.0	255.255.255.128	Interface 0
128.96.34.128	255.255.255.128	Interface 1
128.96.33.0	255.255.255.0	R2

Classless Addressing

- Classless Inter-Domain Routing
  - A technique that addresses two scaling concerns in the Internet
    - The growth of backbone routing table as more and more network numbers need to be stored in them
    - Potential exhaustion of the 32-bit address space
  - Address assignment efficiency
    - Arises because of the IP address structure with class A, B, and C addresses
    - Forces us to hand out network address space in fixed-size chunks of three very different sizes
      - A network with two hosts needs a class C address
        - Address assignment efficiency = 2/255 = 0.78
      - A network with 256 hosts needs a class B address
        - Address assignment efficiency = 256/65535 = 0.39

Classless Addressing

- CIDR tries to balance the desire to minimize the number of routes that a router needs to know against the need to hand out addresses efficiently.

- CIDR uses aggregate routes
  - Uses a single entry in the forwarding table to tell the router how to reach a lot of different networks
  - Breaks the rigid boundaries between address classes

Classless Addressing

- Requires to hand out blocks of class C addresses that share a common prefix
- The convention is to place a /X after the prefix where X is the prefix length in bits
- For example, the 20-bit prefix for all the networks 192.4.16 through 192.4.31 is represented as 192.4.16/20
- By contrast, if we wanted to represent a single class C network number, which is 24 bits long, we would write it 192.4.16/24

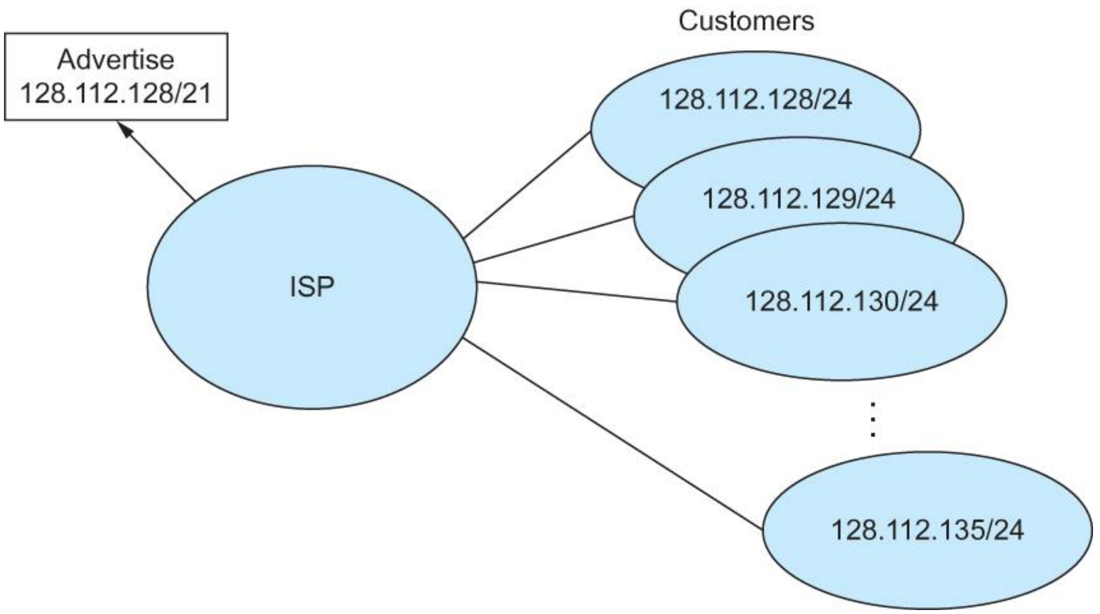
Classless Addressing

- How do the routing protocols handle this classless addresses
  - It must understand that the network number may be of any length
- Represent network number with a single pair

  `<length, value>`

- All routers must understand CIDR addressing

Classless Addressing



Route aggregation with CIDR

IP Forwarding Revisited

- IP forwarding mechanism assumes that it can find the network number in a packet and then look up that number in the forwarding table
- We need to change this assumption in case of CIDR
- CIDR means that prefixes may be of any length, from 2 to 32 bits

IP Forwarding Revisited

- It is also possible to have prefixes in the forwarding tables that overlap
  - Some addresses may match more than one prefix
- For example, we might find both 171.69 (a 16 bit prefix) and 171.69.10 (a 24 bit prefix) in the forwarding table of a single router
- A packet destined to 171.69.10.5 clearly matches both prefixes.
  - The rule is based on the principle of “longest match”
    - 171.69.10 in this case
- A packet destined to 171.69.20.5 would match 171.69 and not 171.69.10

Address Translation Protocol (ARP)

- Map IP addresses into physical addresses
  - destination host
  - next hop router
- Techniques
  - encode physical address in host part of IP address
  - table-based
- ARP (Address Resolution Protocol)
  - table of IP to physical address bindings
  - broadcast request if IP address not in table
  - target machine responds with its physical address
  - table entries are discarded if not refreshed

ARP Packet Format

<table>
<tr><td>0</td><td>8</td><td>16 … 31</td></tr>
<tr><td colspan="2">Hardware type = 1</td><td>ProtocolType = 0x0800</td></tr>
<tr><td>HLen = 48</td><td>PLen = 32</td><td>Operation</td></tr>
<tr><td colspan="3">SourceHardwareAddr (bytes 0–3)</td></tr>
<tr><td colspan="2">SourceHardwareAddr (bytes 4–5)</td><td>SourceProtocolAddr (bytes 0–1)</td></tr>
<tr><td colspan="2">SourceProtocolAddr (bytes 2–3)</td><td>TargetHardwareAddr (bytes 0–1)</td></tr>
<tr><td colspan="3">TargetHardwareAddr (bytes 2–5)</td></tr>
<tr><td colspan="3">TargetProtocolAddr (bytes 0–3)</td></tr>
</table>

- HardwareType: type of physical network (e.g., Ethernet)
- ProtocolType: type of higher layer protocol (e.g., IP)
- HLEN & PLEN: length of physical and protocol addresses
- Operation: request or response
- Source/Target Physical/Protocol addresses

Host Configurations

- Notes
  - Ethernet addresses are configured into network by manufacturer and they are unique
  - IP addresses must be unique on a given internetwork but also must reflect the structure of the internetwork
  - Most host Operating Systems provide a way to manually configure the IP information for the host
  - Drawbacks of manual configuration
    - A lot of work to configure all the hosts in a large network
    - Configuration process is error-prune
  - Automated Configuration Process is required

Internet Control Message Protocol (ICMP)

- Defines a collection of error messages that are sent back to the source host whenever a router or host is unable to process an IP datagram successfully
  - Destination host unreachable due to link /node failure
  - Reassembly process failed
  - TTL had reached 0 (so datagrams don't cycle forever)
  - IP header checksum failed
- ICMP-Redirect
  - From router to a source host
  - With a better route information

Internet Control Message Protocol (ICMP)

- Defines a collection of error messages that are sent back to the source host whenever a router or host is unable to process an IP datagram successfully
  - Destination host unreachable due to link /node failure
  - Reassembly process failed
  - TTL had reached 0 (so datagrams don't cycle forever)
  - IP header checksum failed
- ICMP-Redirect
  - From router to a source host
  - With a better route information

Routing

Forwarding versus Routing

- Forwarding:
  - to select an output port based on destination address and routing table
- Routing:
  - process by which routing table is built

Routing

- Forwarding table VS Routing table
  - Forwarding table
    - Used when a packet is being forwarded and so must contain enough information to accomplish the forwarding function
    - A row in the forwarding table contains the mapping from a network number to an outgoing interface and some MAC information, such as Ethernet Address of the next hop
  - Routing table
    - Built by the routing algorithm as a precursor to build the forwarding table
    - Generally contains mapping from network numbers to next hops

Routing

(a)

Prefix/Length	Next Hop
18/8	171.69.245.10

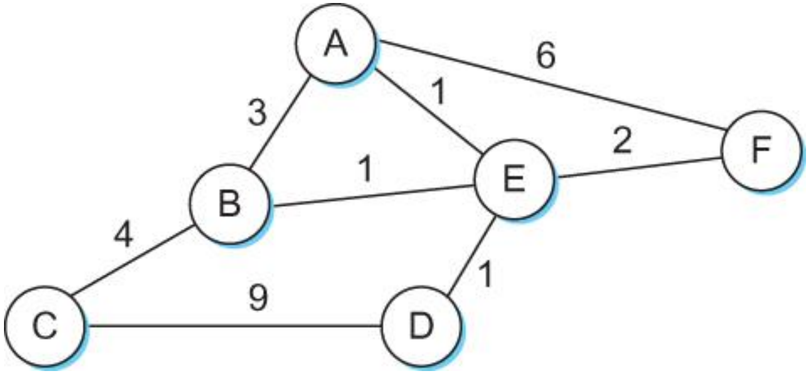
(b)

Prefix/Length	Interface	MAC Address
18/8	if0	8:0:2b:e4:b:1:2

Example rows from (a) routing and (b) forwarding tables

Routing

- Network as a Graph



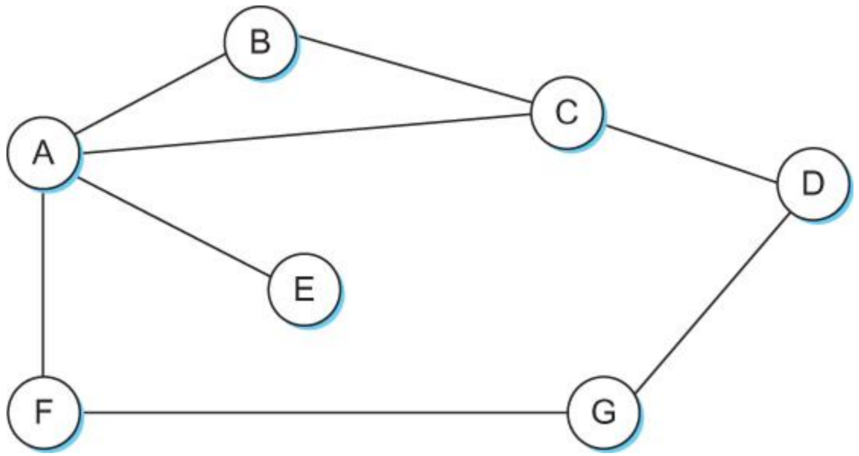
- The basic problem of routing is to find the lowest-cost path between any two nodes
  - Where the cost of a path equals the sum of the costs of all the edges that make up the path

Routing

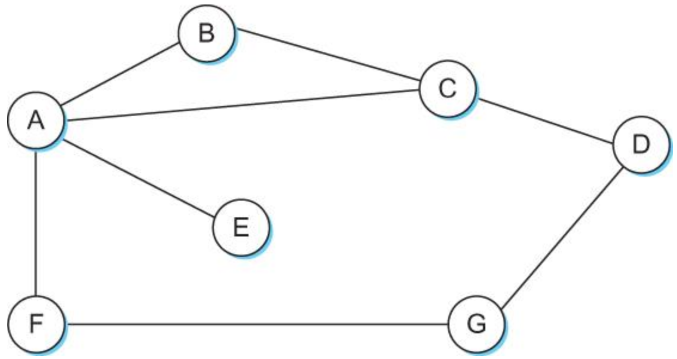
- For a simple network, we can calculate all shortest paths and load them into some nonvolatile storage on each node.
- Such a static approach has several shortcomings
  - It does not deal with node or link failures
  - It does not consider the addition of new nodes or links
  - It implies that edge costs cannot change
- What is the solution?
  - Need a distributed and dynamic protocol
  - Two main classes of protocols
    - Distance Vector
    - Link State

Distance Vector

- Each node constructs a one dimensional array (a vector) containing the “distances” (costs) to all other nodes and distributes that vector to its immediate neighbors
- Starting assumption is that each node knows the cost of the link to each of its directly connected neighbors



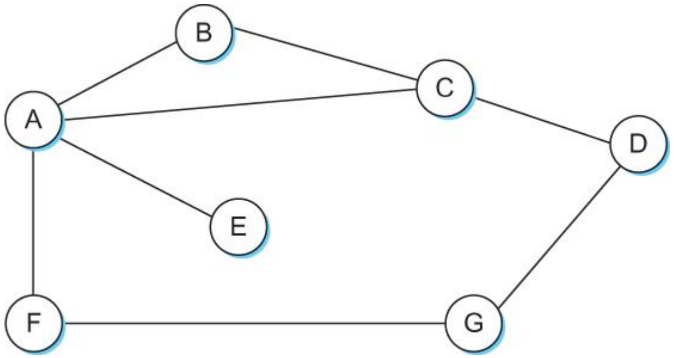
Distance Vector



Information Stored at Node	Distance to Reach Node						
	A	B	C	D	E	F	G
A	0	1	1	∞	1	1	∞
B	1	0	1	∞	∞	∞	∞
C	1	1	0	1	∞	∞	∞
D	∞	∞	1	0	∞	∞	1
E	1	∞	∞	∞	0	∞	∞
F	1	∞	∞	∞	∞	0	1
G	∞	∞	∞	1	∞	1	0

Initial distances stored at each node (global view)

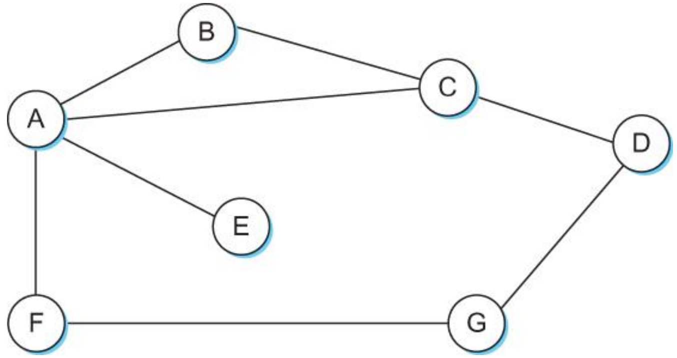
Distance Vector



Destination	Cost	NextHop
B	1	B
C	1	C
D	∞	—
E	1	E
F	1	F
G	∞	—

Initial routing table at node A

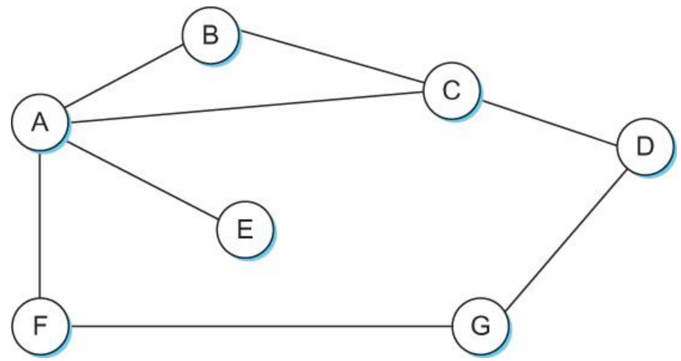
Distance Vector



Destination	Cost	NextHop
B	1	B
C	1	C
D	2	C
E	1	E
F	1	F
G	2	F

Final routing table at node A

Distance Vector



Information Stored at Node	Distance to Reach Node						
	A	B	C	D	E	F	G
A	0	1	1	2	1	1	2
B	1	0	1	2	2	2	3
C	1	1	0	1	2	2	2
D	2	2	1	0	3	2	1
E	1	2	2	3	0	2	3
F	1	2	2	2	2	0	1
G	2	3	2	1	3	1	0

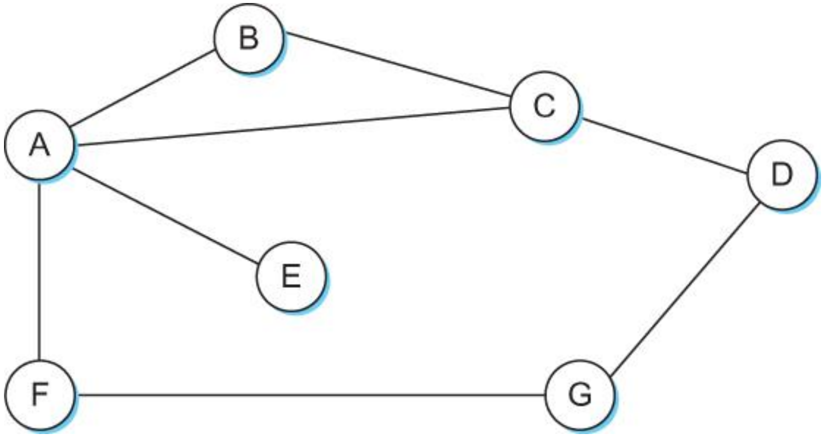
Final distances stored at each node (global view)

Distance Vector

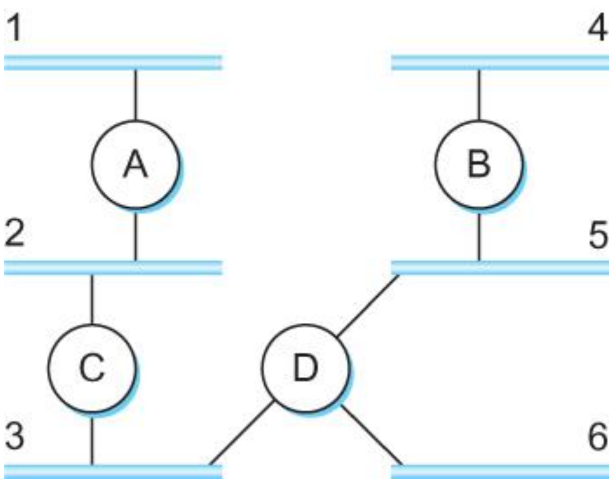
- The distance vector routing algorithm is sometimes called as Bellman-Ford algorithm
- Every T seconds each router sends its table to its neighbor each each router then updates its table based on the new information
- Problems include fast response to good new and slow response to bad news. Also too many messages to update

Distance Vector

- When a node detects a link failure
  - F detects that link to G has failed
  - F sets distance to G to infinity and sends update to A
  - A sets distance to G to infinity since it uses F to reach G
  - A receives periodic update from C with 2-hop path to G
  - A sets distance to G to 3 and sends update to F
  - F decides it can reach G in 4 hops via A



Routing Information Protocol (RIP)



Example Network running RIP

0	8	16 31
Command	Version	Must be zero
Family of net 1		Route Tags
Address prefix of net 1		
Mask of net 1		
Distance to net 1		
Family of net 2		Route Tags
Address prefix of net 2		
Mask of net 2		
Distance to net 2		

RIPv2 Packet Format

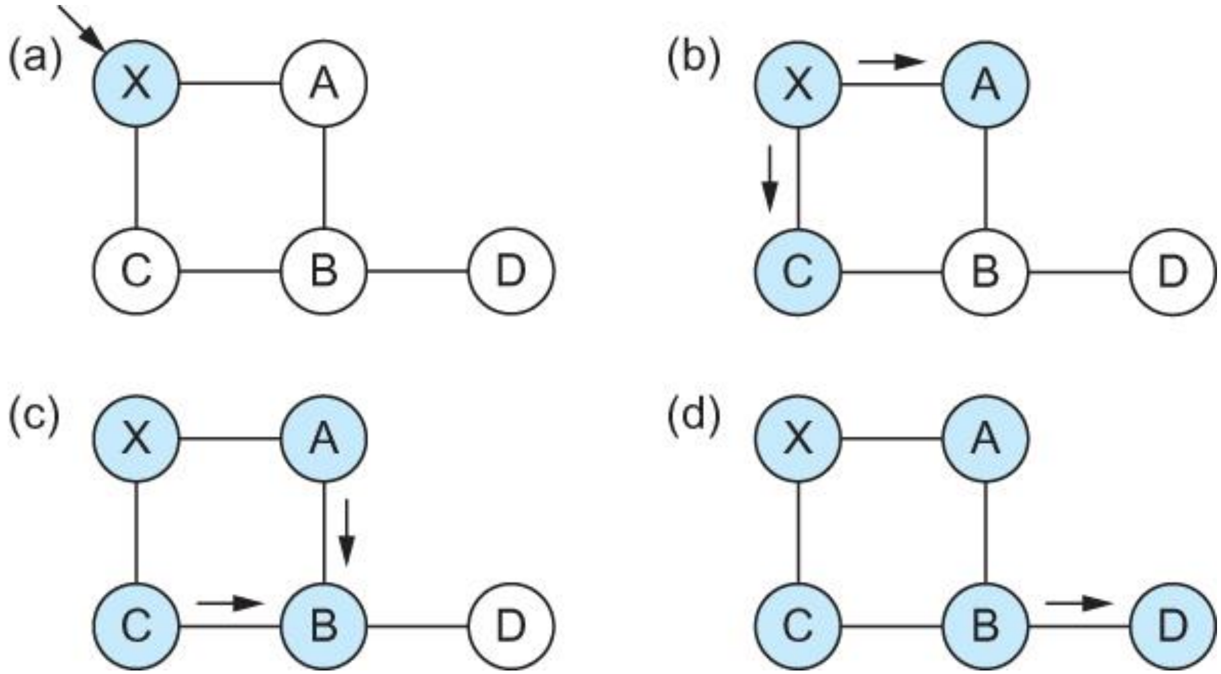
Link State Routing

Strategy: Send to all nodes (not just neighbors) information about directly connected links (not entire routing table).

- Link State Packet (LSP)
  - id of the node that created the LSP
  - cost of link to each directly connected neighbor
  - sequence number (SEQNO)
  - time-to-live (TTL) for this packet
- Reliable Flooding
  - store most recent LSP from each node
  - forward LSP to all nodes but one that sent it
  - generate new LSP periodically; increment SEQNO
  - start SEQNO at 0 when reboot
  - decrement TTL of each stored LSP; discard when TTL=0

Link State

Reliable Flooding



Flooding of link-state packets. (a) LSP arrives at node X; (b) X floods LSP to A and C; (c) A and C flood LSP to B (but not X); (d) flooding is complete