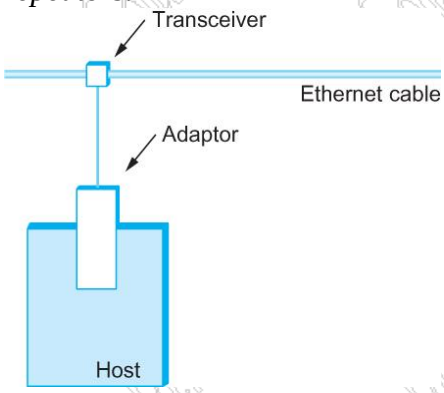
UNIT II

MEDIA ACCESS & INTERNETWORKING

The technology behind the Ethernet is Carrier Sense, Multiple Access with Collision Detect (CSMA/CD). As indicated by the CSMA name, the Ethernet is a multiple-access network, meaning that a set of nodes sends and receives frames over a shared link. The "carrier sense" in CSMA/CD means that all the nodes can distinguish between an idle and a busy link, and "collision detect" means that a node listens as it transmits and can therefore detect when a frame it is transmitting has interfered (collided) with a frame transmitted by another node.

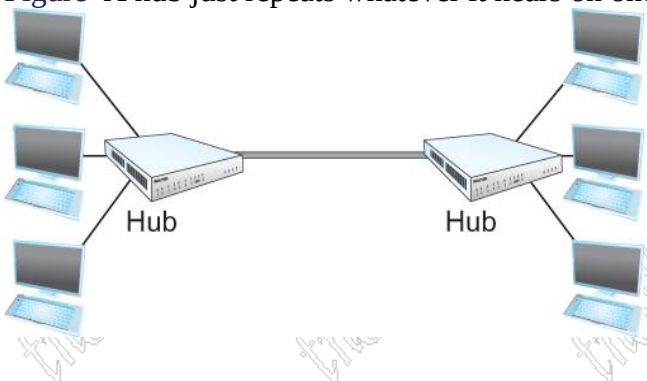
Physical Properties

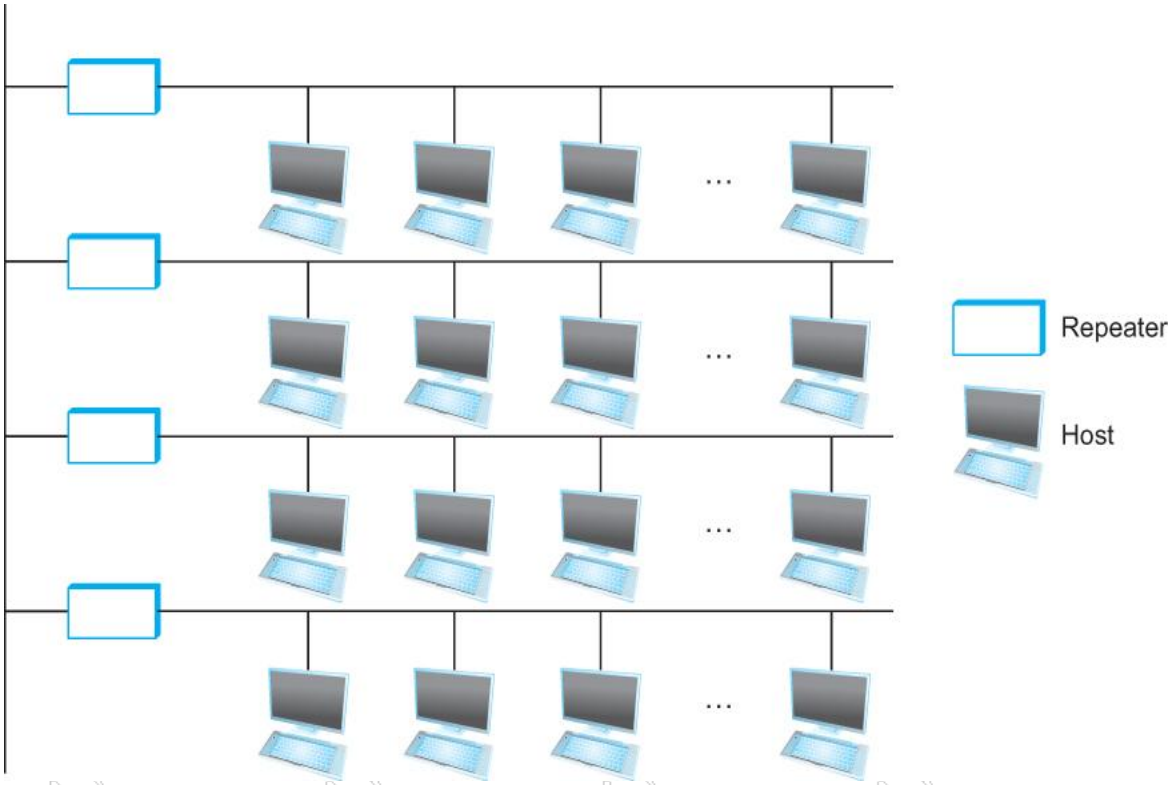
Ethernet segments were originally implemented using coaxial cable of length up to 500 m. Hosts connected to an Ethernet segment by tapping into it. A *transceiver*, a small device directly attached to the tap, detected when the line was idle and drove the signal when the host was transmitting. It also receives incoming signals. The transceiver, in turn, connected to an Ethernet adaptor, which was plugged into the host. Multiple Ethernet segments can be joined together by *repeaters*.



A repeater is a device that forwards digital signals, much like an amplifier forwards analog signals. Repeaters understand only bits, not frames; however, no more than four repeaters could be positioned between any pair of hosts, meaning that a classical Ethernet had a total reach of only 2500 m.

It's also possible to create a multiway repeater, sometimes called a *hub*, as illustrated in Figure A hub just repeats whatever it hears on one port out all its other ports.





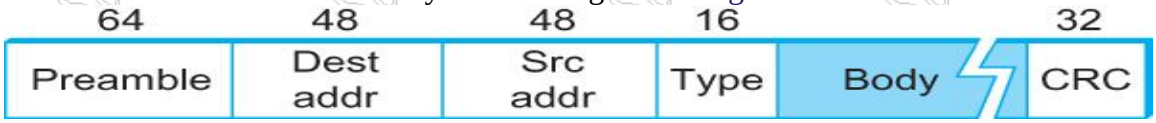
Any signal placed on the Ethernet by a host is broadcast over the entire network; that is, the signal is propagated in both directions, and repeaters and hubs forward the signal on all outgoing segments. Terminators attached to the end of each segment absorb the signal and keep it from bouncing back and interfering with trailing signals.

Access Protocol

An algorithm commonly called Ethernet's *media access control* (MAC) implemented in hardware on the network adaptor controls access to a shared Ethernet link.

Frame Format

Each Ethernet frame is defined by the format given in Figure



The 64-bit preamble allows the receiver to synchronize with the signal; it is a sequence of alternating 0s and 1s. Both the source and destination hosts are identified with a 48-bit address. The packet type field serves as the demultiplexing key; it identifies to which of possibly many higher level protocols this frame should be delivered. Each frame contains up to 1500 bytes of data. Minimally, a frame must contain at least 46 bytes of data, even if this means the host has to pad the frame before transmitting it. The reason for this minimum frame size is that the frame must be long enough to detect a collision; we discuss this more below. Finally, each frame includes a 32-bit CRC.

ADDRESSES

Every Ethernet host in the world has a unique Ethernet address. The address belongs to the adaptor, not the host. Ethernet addresses are sequence of six numbers separated by colons. Each number corresponds to 1 byte of the 6-byte address and is given by a pair of hexadecimal digits, one for each of the 4-bit nibbles in the byte; leading 0s are dropped. For example, 8:0:2b:e4:b1:2 is the human-readable representation of Ethernet address

00001000 00000000 00101011 11100100 10110001 00000010

Each frame transmitted on an Ethernet is received by every adaptor connected to that Ethernet. Each adaptor recognizes those frames addressed to its address and passes only those frames on to the host. (An adaptor can also be programmed to run in *promiscuous* mode, in which case it delivers all received frames to the host, but this is not the normal mode.) In addition to these *unicast* addresses, an Ethernet address consisting of all 1s is treated as a *broadcast* address; all adaptors pass frames addressed to the broadcast address up to the host. Similarly, an address that has the first bit set to 1 but is not the broadcast address is called a *multicast* address. A given host can program its adaptor to accept some set of multicast addresses. Multicast addresses are used to send messages to some subset of the hosts on an Ethernet, an Ethernet adaptor receives all frames and accepts

- Frames addressed to its own address
- Frames addressed to the broadcast address
- Frames addressed to a multicast address, if it has been instructed to listen to that address
- All frames, if it has been placed in promiscuous mode

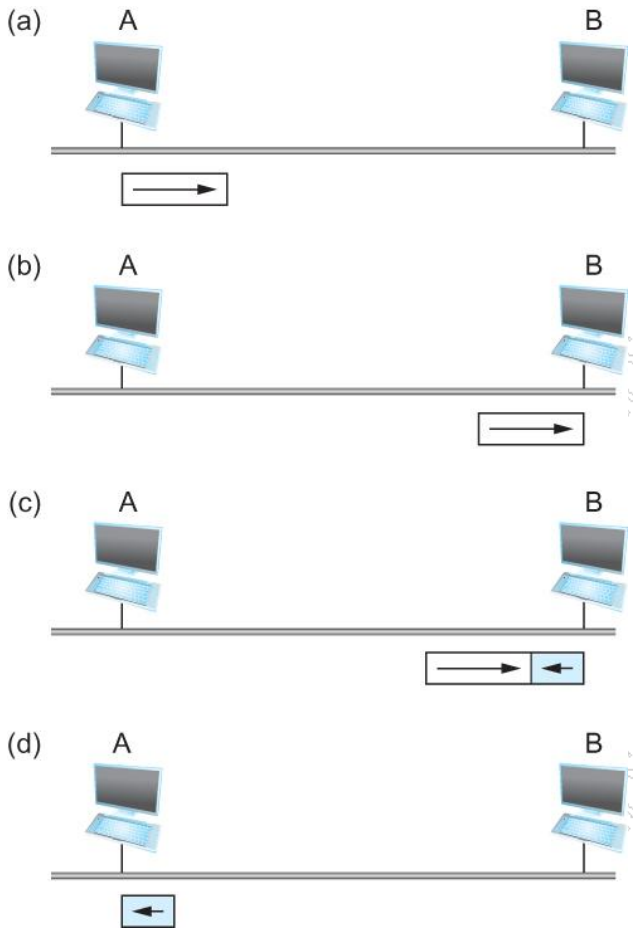
It passes to the host only the frames that it accepts.

TRANSMITTER ALGORITHM

When the adaptor has a frame to send and the line is idle, it transmits the frame immediately; there is no negotiation with the other adaptors. The upper bound of 1500 bytes in the message means that the adaptor can occupy the line for only a fixed length of time.

When an adaptor has a frame to send and the line is busy, it waits for the line to go idle and then transmits immediately. The Ethernet is said to be a *1-persistent* protocol because an adaptor with a frame to send transmits with probability 1 whenever a busy line goes idle. In general, a *p-persistent* algorithm transmits with probability $0 \leq p \leq 1$ after a line becomes idle and defers with probability $q = 1- p$.

There is no centralized control it is possible for two (or more) adaptors to begin transmitting at the same time, either because both found the line to be idle or because both had been waiting for a busy line to become idle. When this happens, the two (or more) frames are said to *collide* on the network. Each sender, because the Ethernet supports collision detection, is able to determine that a collision is in progress. At the moment an adaptor detects that its frame is colliding with another, it first makes sure to transmit a 32-bit jamming sequence and then stops the transmission. Thus, a transmitter will minimally send 96 bits in the case of a collision: 64-bit preamble plus 32-bit jamming sequence. One way that an adaptor will send only 96 bits—which is sometimes called a *runt frame*—is if the two hosts are close to each other. Had the two hosts been farther apart, they would have had to transmit longer, and thus send more bits, before detecting the collision. In fact, the worst-case scenario happens when the two hosts are at opposite ends of the Ethernet.



Worst-case scenario: (a) A sends a frame at time *t*; *(b) A's frame arrives* at B at time *t + d; (c) B begins transmitting at time t + d and collides with A's frame;* (d) B's runt (32-bit) frame arrives at A at time *t + 2d*.

WIRELESS

Spread Spectrum technique

- Idea is to spread the signal over a wider frequency band So as to minimize the impact of interference from other devices
- Originally designed for military use

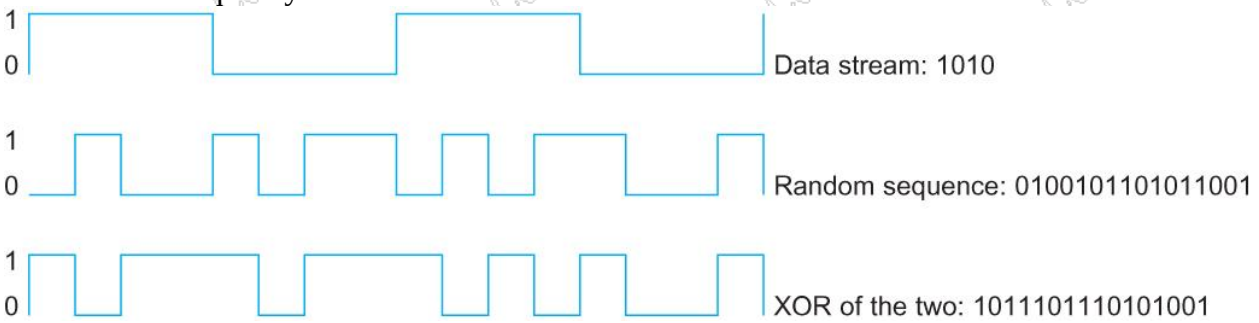
Frequency hopping

- Transmitting signal over a random sequence of frequencies
- First transmitting at one frequency, then a second, then a third…
- The sequence of frequencies is not truly random, instead computed algorithmically by a pseudorandom number generator
- The receiver uses the same algorithm as the sender, initializes it with the same seed, and is Able to hop frequencies in sync with the transmitter to correctly receive the frame

Direct sequence

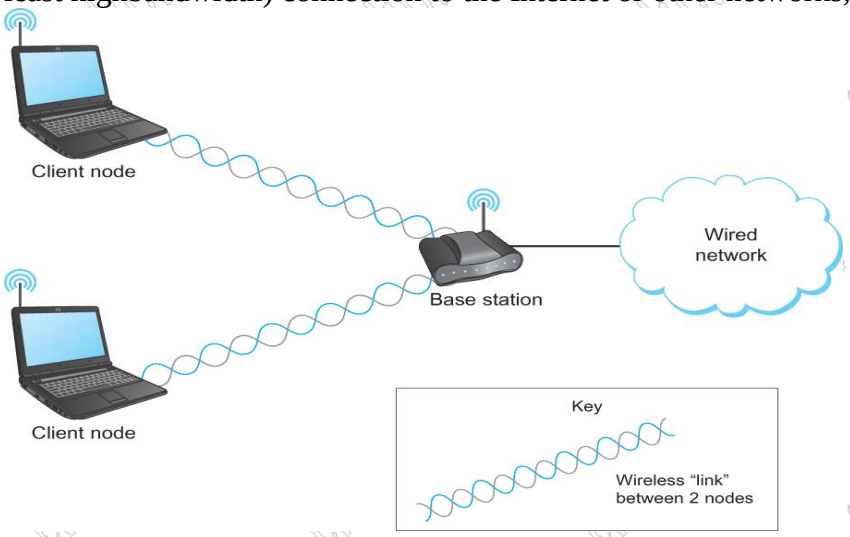
- Represents each bit in the frame by multiple bits in the transmitted signal.
- For each bit the sender wants to transmit

- It actually sends the exclusive OR of that bit and *n* random bits
- The sequence of random bits is generated by a pseudorandom number generator known to both the sender and the receiver.
- The transmitted values, known as an *n*-bit chipping code, spread the signal across a frequency band that is *n* times wider



Example 4-bit chipping sequence

In many wireless networks today we observe that there are two different classes of endpoints. One endpoint, sometimes described as the *base station*, usually has no mobility but has a wired (or at least highbandwidth) connection to the Internet or other networks, as shown in Figure .

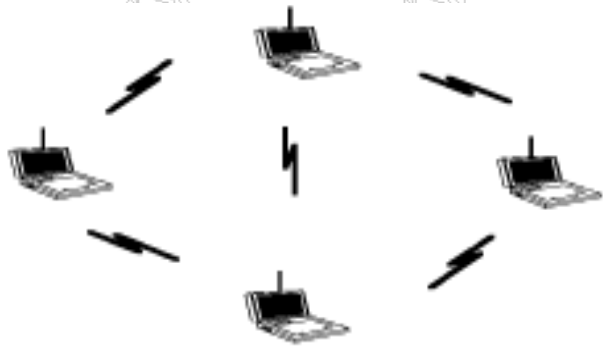


A wireless network using a base station

The node at the other end of the link—shown here as a client node—is often mobile and relies on its link to the base station for all of its communication with other nodes. in Figure a wavy pair of lines to represent the wireless "link" abstraction provided between two devices (e.g., between a base station and one of its client nodes). One of the interesting aspects of wireless communication is that it naturally supports point-to multipoint communication, because radio waves sent by one device can be simultaneously received by many devices. communication between non-base (client) nodes is routed via the base station. This is in spite of the fact that
radio waves emitted by one client node may well be received by other client nodes

MESH OR AD-HOC NETWORK

In a wireless mesh, nodes are peers; that is, there is no special base station node. Messages may be forwarded via a chain of peer nodes as long as each node is within range of the preceding node. This is illustrated in Figure



802.11/Wi-Fi

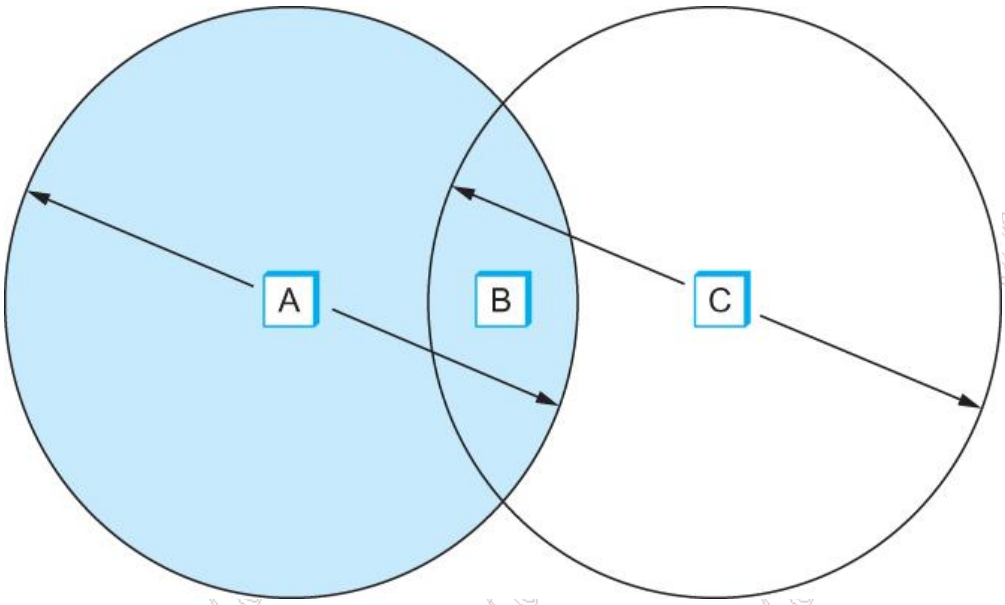
Physical Properties

The original 802.11 standard defined two radio-based physical layers standards, one using frequency hopping (over 79 1-MHz-wide frequency bandwidths) and the other using direct sequence spread spectrum (with an 11-bit chipping sequence). Both provided data rates in the 2 Mbps range.

Collision Avoidance

Consider the situation in the following figure where each of four nodes is able to send and receive signals that reach just the nodes to its immediate left and right

- For example, B can exchange frames with A and C, but it cannot reach D
- C can reach B and D but not A
- Suppose both A and C want to communicate with B and so they each send it a frame.
- A and C are unaware of each other since their signals do not carry that far
- These two frames collide with each other at B
- But unlike an Ethernet, neither A nor C is aware of this collision
- A and C are said to *hidden nodes* with respect to each other

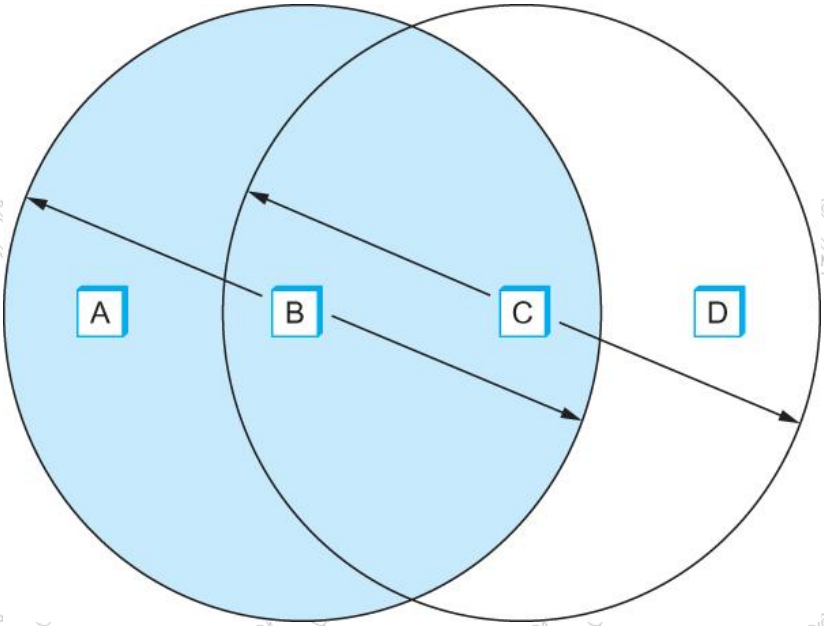


The "Hidden Node" Problem. Although A and C are hidden from each other, their signals can collide at B. (B's reach is not shown.)

exposed node problem

- Suppose B is sending to A. Node C is aware of this communication because it hears B's transmission.
- It would be a mistake for C to conclude that it cannot transmit to anyone just because it can hear B's transmission.
- Suppose C wants to transmit to node D.
- This is not a problem since C's transmission to D will not interfere with A's ability to receive from B.

tnScholars.com



Exposed Node Problem. Although B and C are exposed to each other's signals, there is no interference if B transmits to A while C transmits to D. (A and D's reaches are not shown.)

802.11 addresses these two problems with an algorithm called Multiple Access with Collision Avoidance (MACA).

KEY IDEA

- Sender and receiver exchange control frames with each other before the sender actually transmits any data.
- This exchange informs all nearby nodes that a transmission is about to begin
- Sender transmits a *Request to Send* (RTS) frame to the receiver.
- The RTS frame includes a field that indicates how long the sender wants to hold the medium- Length of the data frame to be transmitted
- Receiver replies with a *Clear to Send* (CTS) frame
- This frame echoes this length field back to the sender
- Any node that sees the CTS frame knows that it is close to the receiver, therefore cannot transmit for the period of time it takes to send a frame of the specified length
- Any node that sees the RTS frame but not the CTS frame is not close enough to the receiver to interfere with it, and so is free to transmit

- Using ACK in MACA
- Proposed in MACAW: MACA for Wireless LANs
- Receiver sends an ACK to the sender after successfully receiving a frame
- All nodes must wait for this ACK before trying to transmit
- If two or more nodes detect an idle link and try to transmit an RTS frame at the same time,Their RTS frame will collide with each other
- 802.11 does not support collision detection So the senders realize the collision has happened when they do not receive the CTS frame after a period of time
- In this case, they each wait a random amount of time before trying again.
- The amount of time a given node delays is defined by the *exponential backoff* algorithm

Distribution System

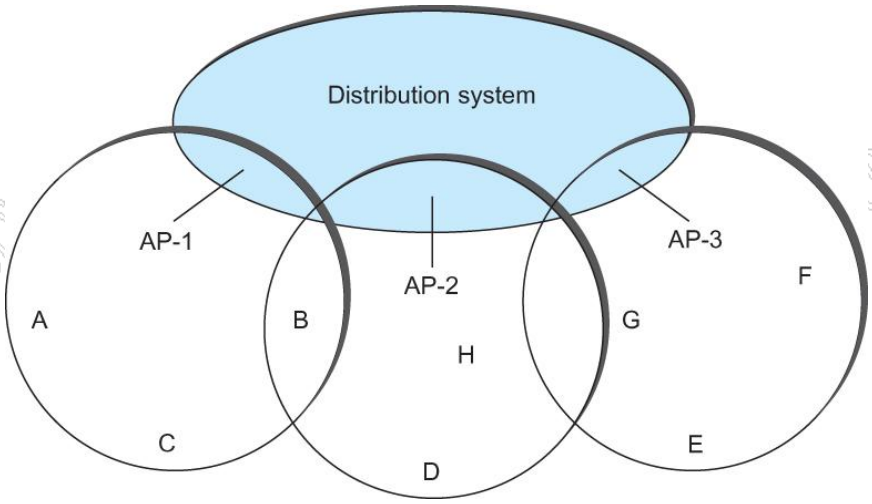
802.11 is suitable for an ad-hoc configuration of nodes that may or may not be able to communicate with all other nodes. Nodes are free to move around. The set of directly reachable nodes may change over time

To deal with this mobility and partial connectivity, 802.11 defines additional structures on a set of nodes. Instead of all nodes being created equal, some nodes are allowed to roam some are connected to a wired network infrastructure - they are called *Access Points* (AP) and they are connected to each other by a so-called *distribution system*

Following figure illustrates a distribution system that connects three access points, each of which services the nodes in the same region Each of these regions is analogous to a cell in a cellular phone system with the APIs playing the same role as a base station

The distribution network runs at layer 2 of the ISO architecture

tnScholars.com



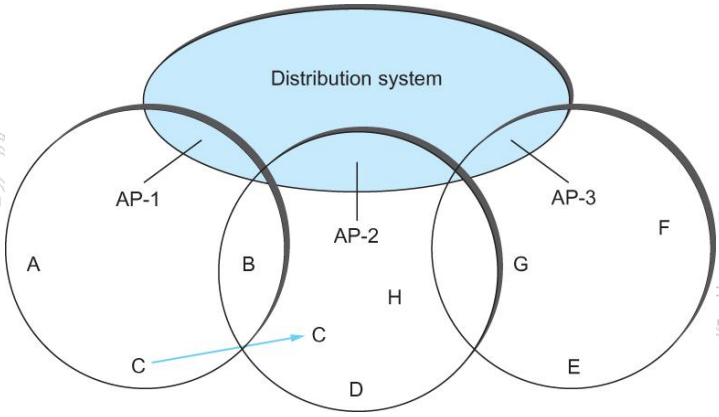
Fig, Access points connected to a distribution network

Although two nodes can communicate directly with each other if they are within reach of each other, the idea behind this configuration is, Each nodes associates itself with one access point. For node A to communicate with node E, A first sends a frame to its AP-1 which forwards the frame across the distribution system to AP-3, which finally transmits the frame to E

The technique for selecting an AP :

- The technique for selecting an AP is called *scanning*
- The node sends a *Probe* frame
- All APs within reach reply with a *Probe Response* frame
- The node selects one of the access points and sends that AP an *Association Request* frame
- The AP replies with an *Association Response* frame
- A node engages this protocol whenever it joins the network, as well as when it becomes unhappy with its current AP
- This might happen, for example, because the signal from its current AP has weakened due to the node moving away from it
- Whenever a node acquires a new AP, the new AP notifies the old AP of the change via the distribution system
- Consider the situation shown in the following figure when node C moves from the cell serviced by AP-1 to the cell serviced by AP-2.
- As it moves, it sends *Probe* frames, which eventually result in *Probe Responses* from AP-2.
- At some point, C prefers AP-2 over AP-1 , and so it associates itself with that access point.

- This is called *active scanning* since the node is actively searching for an access point



Node Mobility

- APs also periodically send a *Beacon* frame that advertises the capabilities of the access point; these include the transmission rate supported by the AP. This is called *passive scanning*
- A node can change to this AP based on the *Beacon* frame simply by sending it an *Association Request* frame back to the access point.

Frame Format

diagram(copy from note)

- Source and Destinations addresses: each 48 bits
- Data: up to 2312 bytes
- CRC: 32 bit
- Control field: 16 bits
- Contains three subfields (of interest)
    - 6 bit **Type** field: indicates whether the frame is an RTS or CTS frame or being used by the scanning algorithm
    - A pair of 1 bit fields : called **ToDS** and **FromDS**
- Frame contains four addresses
    - these addresses are interpreted depends on the settings of the **ToDS** and **FromDS** bits in the frame's Control field

- This is to account for the possibility that the frame had to be forwarded across the distribution system which would mean that, the original sender is not necessarily the same as the most recent transmitting node ,Same is true for the destination address

- Simplest case
    - When one node is sending directly to another, both the DS bits are 0, Addr1 identifies the target node, and Addr2 identifies the source node
- Most complex case
    - Both DS bits are set to 1
        - Indicates that the message went from a wireless node onto the distribution system, and then from the distribution system to another wireless node
    - With both bits set,
        - Addr1 identifies the ultimate destination,
        - Addr2 identifies the immediate sender (the one that forwarded the frame from the distribution system to the ultimate destination)
        - Addr3 identifies the intermediate destination (the one that accepted the frame from a wireless node and forwarded across the distribution system)
        - Addr4 identifies the original source
        - Addr1: E, Addr2: AP-3, Addr3: AP-1, Addr4: A

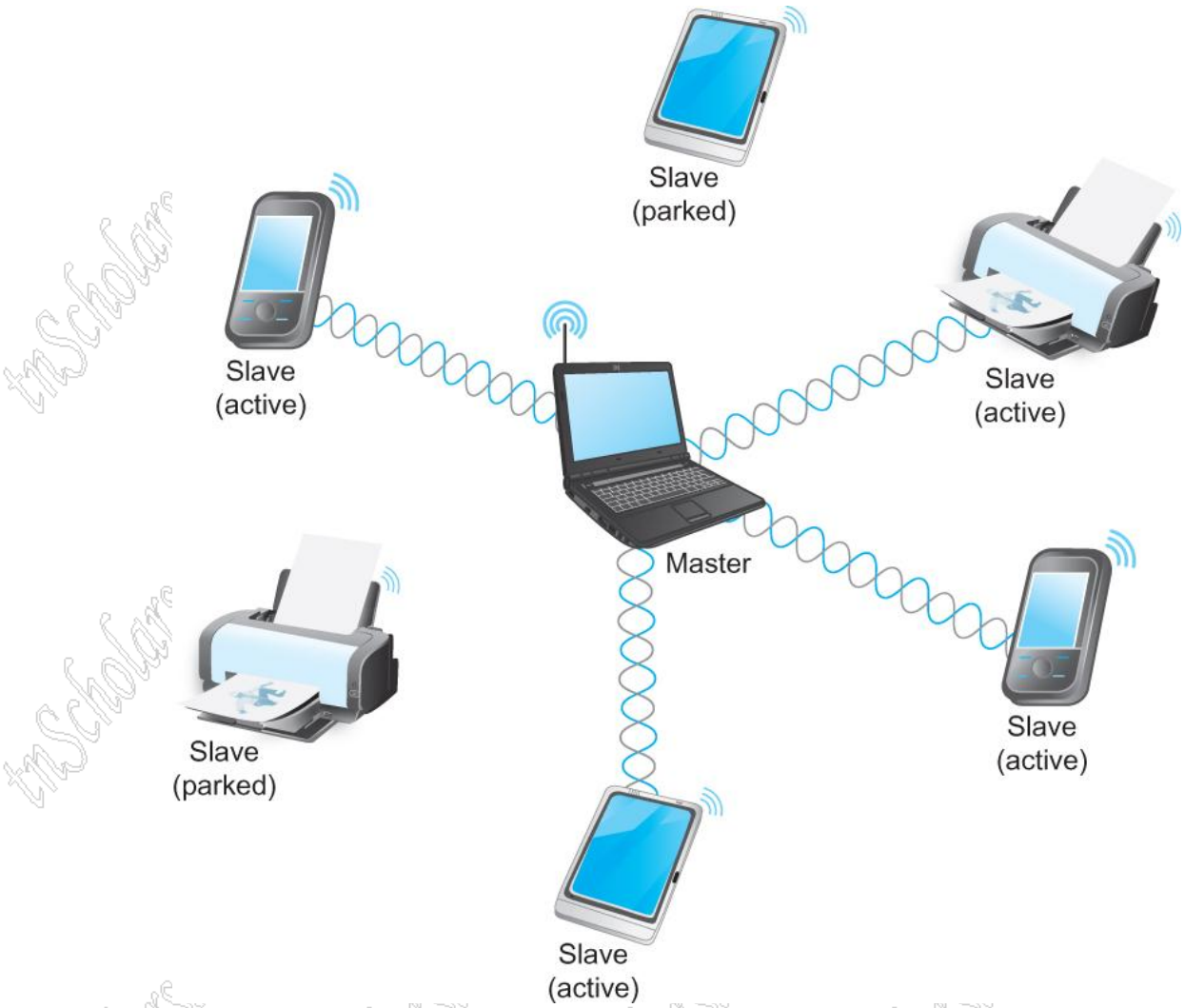
BLUETOOTH

Used for very short range communication between mobile phones, PDAs, notebook computers and other personal or peripheral devices,

- Operates in the license-exempt band at 2.45 GHz
- Has a range of only 10 m
- Communication devices typically belong to one individual or group
- Sometimes categorized as Personal Area Network (PAN)
- Version 2.0 provides speeds up to 2.1 Mbps
- Power consumption is low

- Bluetooth is specified by an industry consortium called the Bluetooth Special Interest Group
- It specifies an entire suite of protocols, going beyond the link layer to define application protocols, which it calls *profiles*, for a range of applications
- There is a profile for synchronizing a PDA with personal computer
- Another profile gives a mobile computer access to a wired LAN
- The basic Bluetooth network configuration is called a *piconet*
- Consists of a master device and up to seven slave devices
- Any communication is between the master and a slave
- The slaves do not communicate directly with each other
- A slave can be *parked*: set to an inactive, low-power state

tnScholars.com



SWITCHING AND BRIDGING

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

properties of star topology

- 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

- switches can be connected 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
- It is impossible for two hosts on the same Ethernet to transmit continuously at 10Mbps because they share the same transmission medium
- Every host on a switched network has its own link to the switch
  So it may be entirely possible for many hosts to transmit at the full link speed (bandwidth) provided that the switch is designed with enough aggregate capacity
- 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

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*

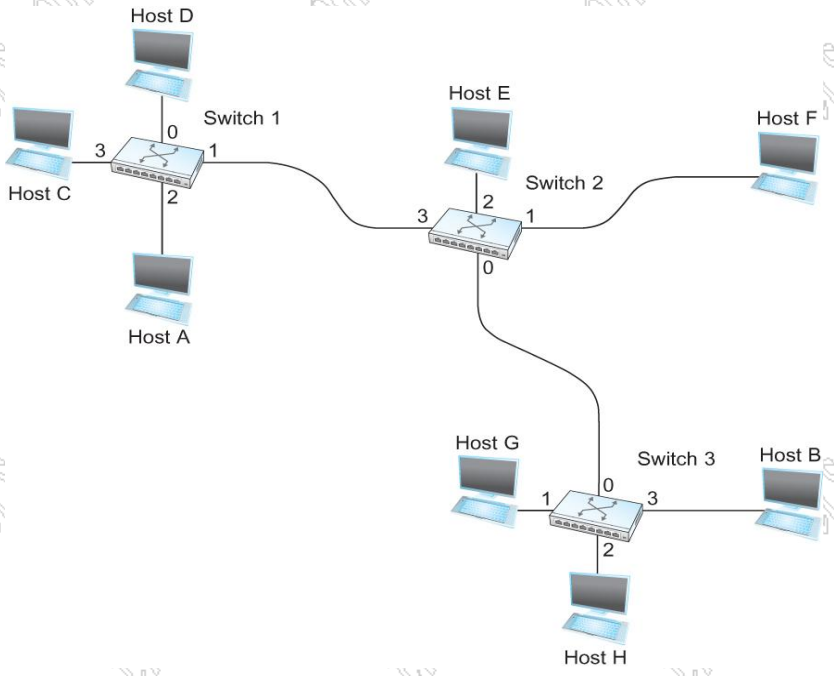
Assumptions

- Each host has a globally unique address
- There is some way to identify the input and output ports of each switch

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



An example network

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

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

Forwarding Table for Switch 2

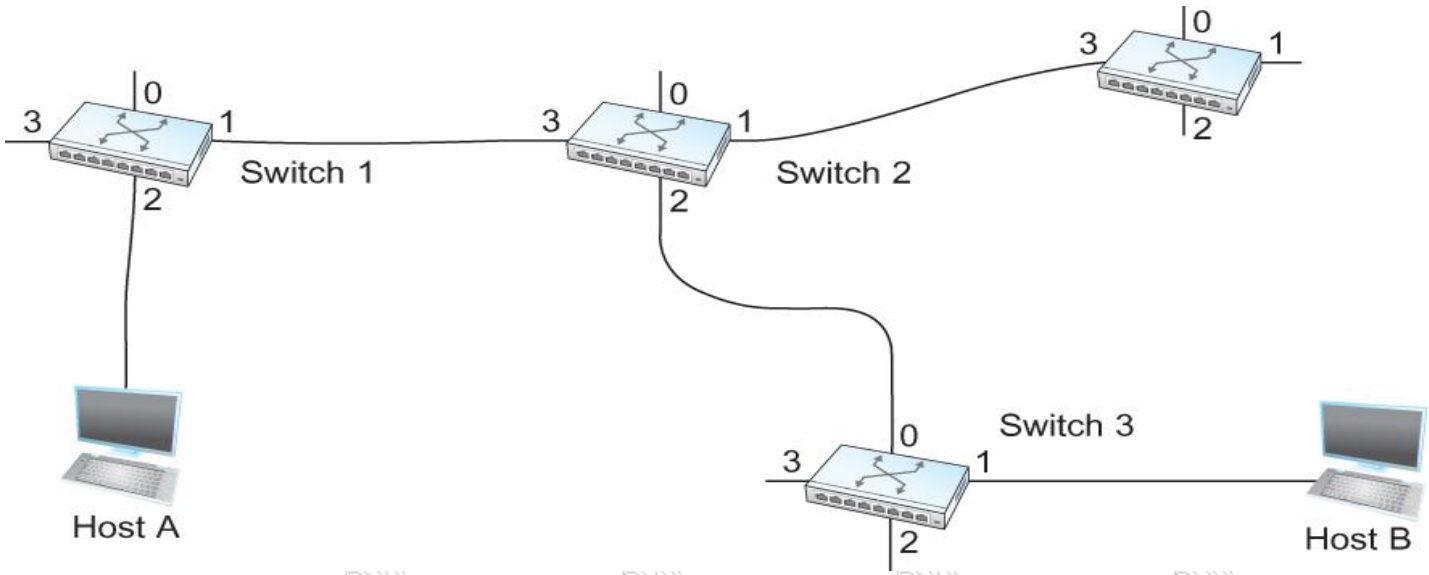
Consider the example network illustrated in Figure, in which the hosts have addresses A, B, C, and so on. To decide how to forward a packet, a switch consults a *forwarding table* (sometimes called a *routing table*), an example of which is depicted in Table, this particular table shows the forwarding information that switch 2 needs to forward datagrams

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

VIRTUAL CIRCUIT SWITCHING

This approach is also referred to as a *connection-oriented model*, requires setting up a virtual connection from the source host to the destination host before any data is sent.



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

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
- 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.

Two broad classes of approach to establishing connection state

- **Network Administrator** will configure the state
  - The virtual circuit is permanent (PVC)
  - The network administrator can delete this
  - Can be thought of as a long-lived or administratively configured VC
- A **host** can send messages into the network to cause the state to be established
  - This is referred as signalling and the resulting virtual circuit is said to be switched (SVC)
  - A host may set up and delete such a VC dynamically without the involvement of a network administrator
    - 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

      The administrator then picks a VCI value that is currently unused on each link for the connection
      - **For our example**,
      - Suppose the VCI value 5 is chosen for the link from host A to switch 1
      - 11 is chosen for the link from switch 1 to switch 2
      - So the switch 1 will have an entry in the VC table

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

Similarly, suppose

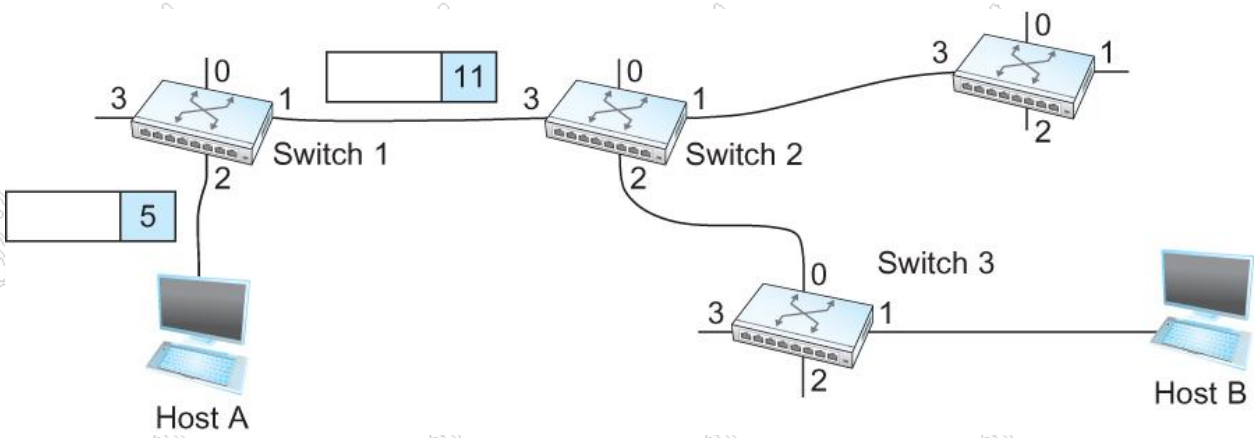
- VCI of 7 is chosen to identify this connection on the link from switch 2 to switch 3
- VCI of 4 is chosen for the link from switch 3 to host B
- Switches 2 and 3 are configured with the following VC table

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

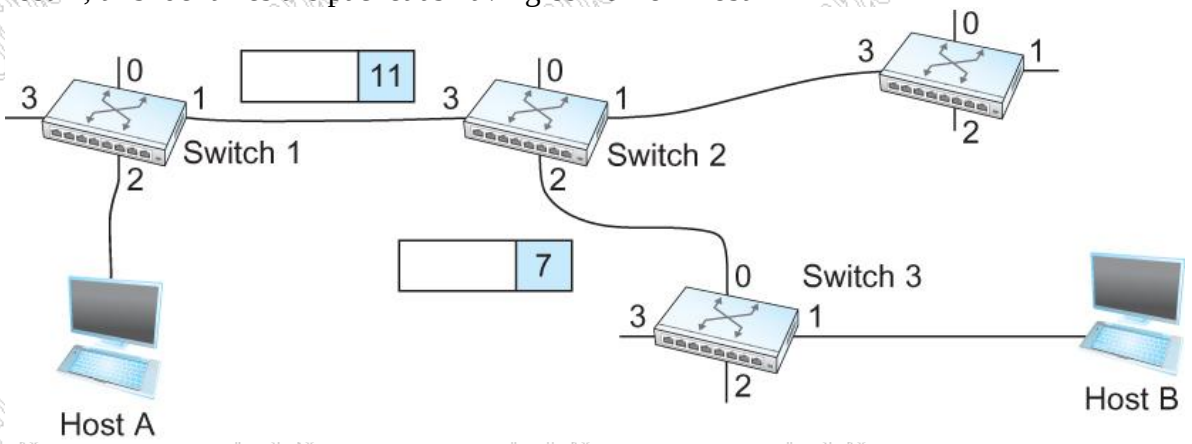
Incoming Interface	Incoming VC	Outgoing Interface	Outgoing VC
0	7	1	4

- 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



- 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



- 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

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
- 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
- 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
- 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

Characteristics of VC

- Since host A has to wait for the connection request to reach the far side of the network and return before it can send its first data packet, there is at least one RTT of delay before data is sent
- While the connection request contains the full address for host B (which might be quite large, being a global identifier on the network), each data packet contains only a small identifier, which is only unique on one link.
- Thus the per-packet overhead caused by the header is reduced relative to the datagram model
- If a switch or a link in a connection fails, the connection is broken and a new one will need to be established.
- Also the old one needs to be torn down to free up table storage space in the switches

- The issue of how a switch decides which link to forward the connection request on has similarities with the function of a routing algorithm
- **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

An X.25 network – a packet-switched network that uses the connection-oriented model – employs the following three-part strategy

- Buffers are allocated to each virtual circuit when the circuit is initialized
- The sliding window protocol is run between each pair of nodes along the virtual circuit, and this protocol is augmented with the flow control to keep the sending node from overrunning the buffers allocated at the receiving node
- The circuit is rejected by a given node if not enough buffers are available at that node when the connection request message is processed

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

Most popular examples of VC technologies are Frame Relay and ATM

- One of the applications of Frame Relay is the construction of VPN

■ ATM

generic flow control (GFC) bits, which never saw much use, and start with the 24 bits that are labelled VPI (virtual path identifier—8 bits) and VCI (virtual circuit identifier—16 bits). If you consider these bits together as a single 24-bit 8-bit cyclic redundancy check (CRC), known as the *header error check* (HEC)

4 8 16 3 1

8 384

GFC	VPI	VCI	Type	CLP	HEC(crc)	Payload

ATM cell format

■ **SOURCE ROUTING**

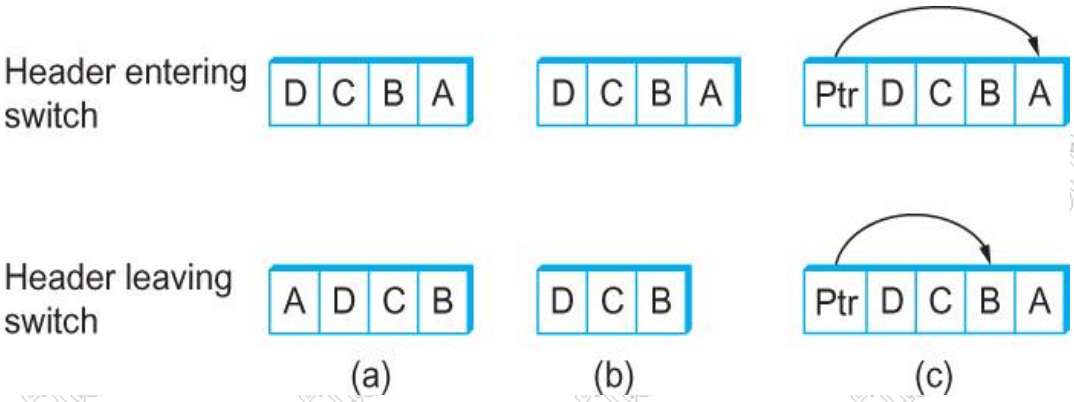
■ All the information about network topology that is required to switch a packet across the network is provided by the source host

There are various ways to implement source routing. One would be to assign a number to each output of each switch and to place that number in the header of the packet. The switching function is then very simple: For each packet that arrives on an input, the switch would read the port

number in the header and transmit the packet on that output. However, since there will in general be more than one switch in the path between the sending and the receiving host, the header for the packet needs to contain enough information to allow every switch in the path to determine which output the packet needs to be placed on. One way to do this would be to put an ordered list of switch ports in the header and to rotate the list so that the next switch in the path is always at the front of the list. Figure illustrates this idea.

FIG

In this example, the packet needs to traverse three switches to get from host A to host B. At switch 1, it needs to exit on port 1, at the next switch it needs to exit at port 0, and at the third switch it needs to exit at port 3. Thus, the original header when the packet leaves host A contains the list of ports (3, 0, 1), where we assume that each switch reads the rightmost element of the list. To make sure that the next switch gets the appropriate
information, each switch rotates the list after it has read its own entry. Thus, the packet header as it leaves switch 1 en route to switch 2 is now (1, 3, 0); switch 2 performs another rotation and sends out a packet with (0, 1, 3) in the header. Although not shown, switch 3 performs yet another rotation, restoring the header to what it was when host A sent it. There are several things to note about this approach. First, it assumes that host A knows enough about the topology of the network to form a header that has all the right directions in it for every switch in the path. This is somewhat analogous to the problem of building the forwarding tables in a datagram network or figuring out where to send a setup packet in a virtual circuit network. Second, observe that we cannot predict how big the header needs to be, since it must be able to hold one word of information for every switch on the path. This implies that headers are probably of variable length with no upper bound, unless we can predict with absolute certainty the maximum number of switches through which a packet will ever need to pass. Third, there are some variations on this approach. For example, rather than rotate the header, each switch could just strip the first element as it uses it. Rotation has an advantage over stripping, however: Host B gets a copy of the complete header, which may help it figure out how to get back to host A. Yet another alternative is to have the header carry a pointer to the current “next port” entry, so that each switch just updates the pointer rather than rotating the header; this may be more efficient to implement. We show these three approaches in fig



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
- A pair of Ethernets 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**

Simplest Strategy for Bridges

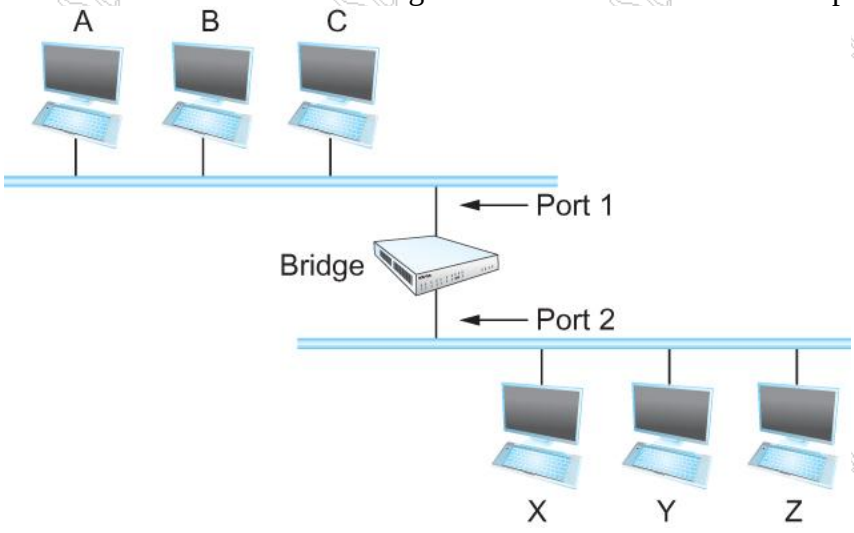
- Accept LAN frames on their inputs and forward them out to all other outputs

Learning Bridges

There is no need to forward all the frames that a bridge receives

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?

Solution

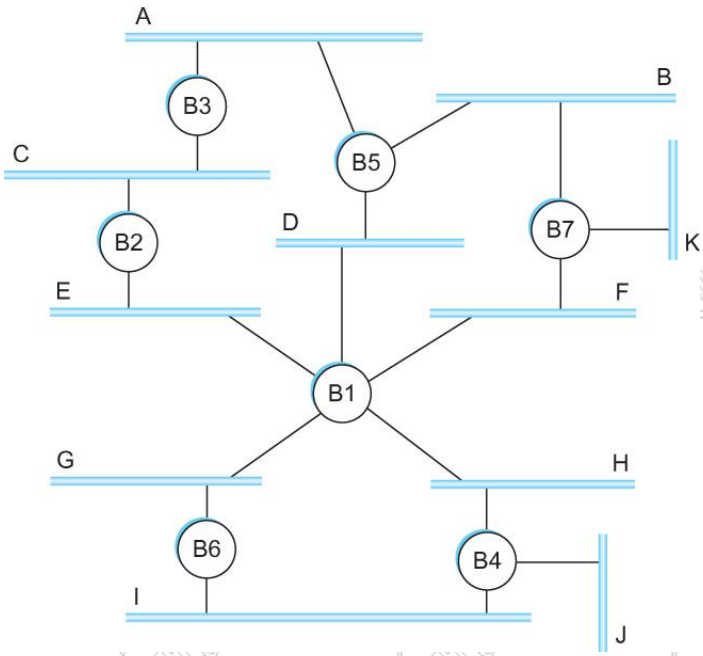
- Download a table into the bridge

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

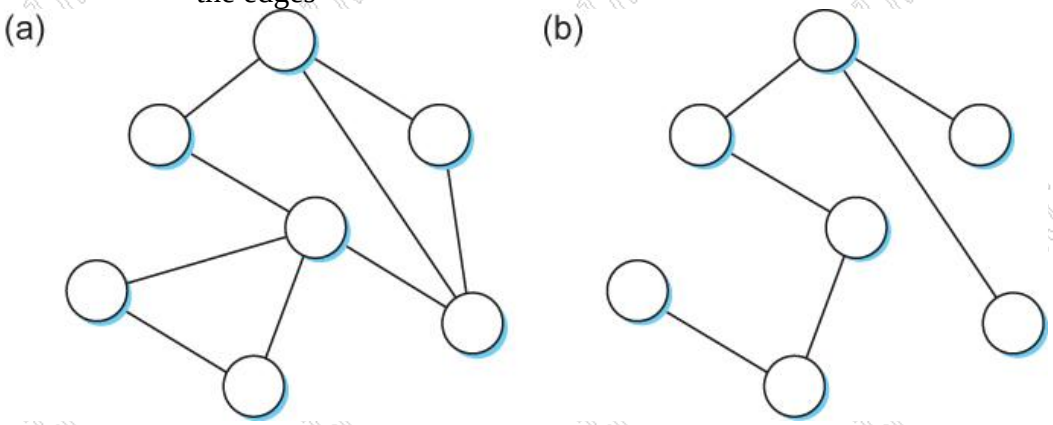
  - Who does the download?
  - Human
  - Too much work for maintenance
- Can the bridge learn this information by itself?
  - Yes
  - How

    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
- Strategy works fine if the extended LAN does not have a loop in it
- Why?
  - Frames potentially loop through the extended LAN forever

tnScholars.com



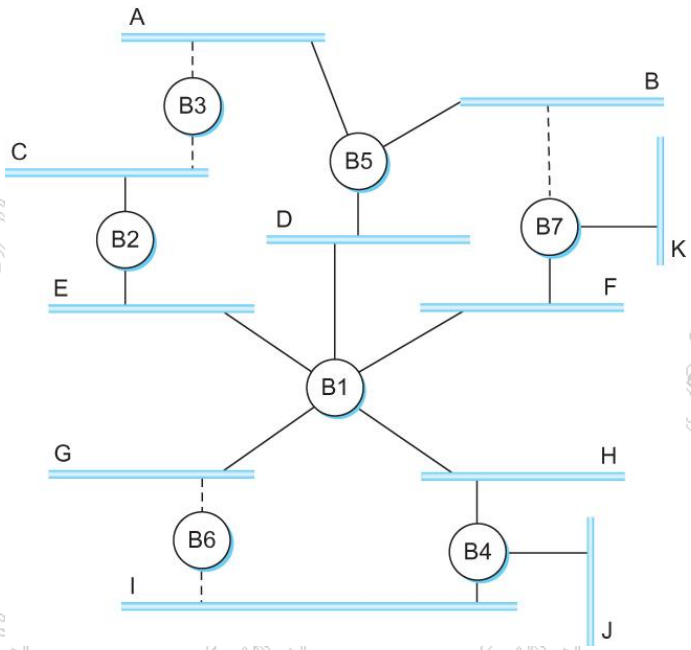
- Bridges B1, B4, and B6 form a loop
- How does an extended LAN come to have a loop in it?
  - Network is managed by more than one administrator
    - For example, it spans multiple departments in an organization
    - It is possible that no single person knows the entire configuration of the network
      - A bridge that closes a loop might be added without anyone knowing
  - Loops are built into the network to provide redundancy in case of failures
- **Solution**
  - **Distributed Spanning Tree Algorithm**
- Think of the extended LAN as being represented by a graph that possibly has loops (cycles)
- A spanning tree is a sub-graph of this graph that covers all the vertices but contains no cycles
  - Spanning tree keeps all the vertices of the original graph but throws out some of the edges



Example of (a) a cyclic graph; (b) a corresponding spanning tree.

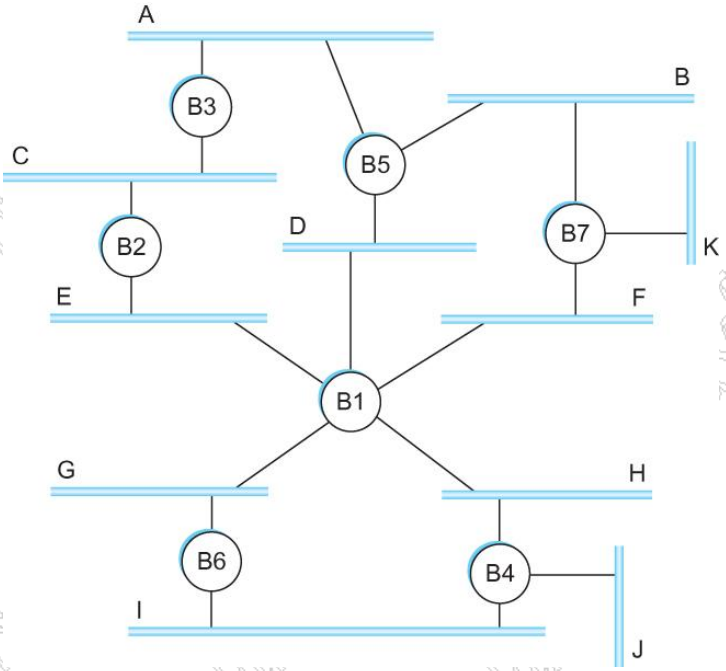
- Each bridge decides the ports over which it is and is not willing to forward frames
    - In a sense, it is by removing ports from the topology that the extended LAN is reduced to an acyclic tree
    - It is even possible that an entire bridge will not participate in forwarding frames
- Algorithm is dynamic
    - The bridges are always prepared to reconfigure themselves into a new spanning tree if some bridges fail
    - Main idea
    - Each bridge selects the ports over which they will forward the frames
- Algorithm selects ports as follows:
    - Each bridge has a unique identifier
        - B1, B2, B3,…and so on.
    - Elect the bridge with the smallest id as the root of the spanning tree
    - The root bridge always forwards frames out over all of its ports
    - Each bridge computes the shortest path to the root and notes which of its ports is on this path
        - This port is selected as the bridge's preferred path to the root
    - Finally, all the bridges connected to a given LAN elect a single *designated bridge* that will be responsible for forwarding frames toward the root bridge
- Each LAN's designated bridge is the one that is closest to the root
- If two or more bridges are equally close to the root,
    - Then select bridge with the smallest id
- Each bridge is connected to more than one LAN
    - So it participates in the election of a designated bridge for each LAN it is connected to.
    - Each bridge decides if it is the designated bridge relative to each of its ports
    - The bridge forwards frames over those ports for which it is the designated bridge
- B1 is the root bridge
- B3 and B5 are connected to LAN A, but B5 is the designated bridge
- B5 and B7 are connected to LAN B, but B5 is the designated bridge

tnScholars.com

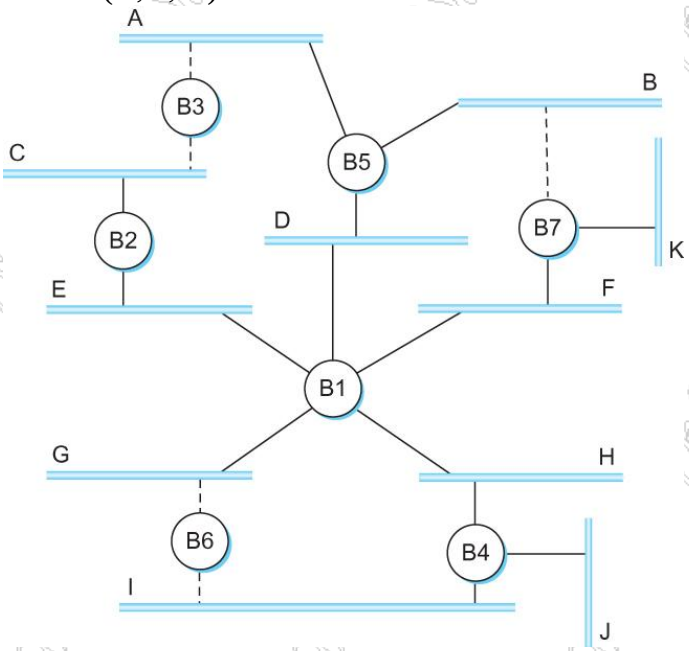


- Initially each bridge thinks it is the root, so it sends a configuration message on each of its ports identifying itself as the root and giving a distance to the root of 0
- Upon receiving a configuration message over a particular port, the bridge checks to see if the new message is *better* than the current best configuration message recorded for that port
- The new configuration is better than the currently recorded information if
  - It identifies a root with a smaller id or
  - It identifies a root with an equal id but with a shorter distance or
  - The root id and distance are equal, but the sending bridge has a smaller id
- If the new message is better than the currently recorded one,
  - The bridge discards the old information and saves the new information
  - It first adds 1 to the distance-to-root field
  - When a bridge receives a configuration message indicating that it is not the root bridge (that is, a message from a bridge with smaller id)
  - The bridge stops generating configuration messages on its own
  - Only forwards configuration messages from other bridges after 1 adding to the distance field
- When a bridge receives a configuration message that indicates it is not the designated bridge for that port

  => a message from a bridge that is closer to the root or equally far from the root but with a smaller id
  - The bridge stops sending configuration messages over that port
- When the system stabilizes,
  - Only the root bridge is still generating configuration messages.
  - Other bridges are forwarding these messages only over ports for which they are the designated bridge
- Consider the situation when the power had just been restored to the building housing the following network



- All bridges would start off by claiming to be the root
- Denote a configuration message from node X in which it claims to be distance d from the root node Y as (Y, d, X)



- Consider the activity at node B3
- B3 receives (B2, 0, B2)
- Since 2 < 3, B3 accepts B2 as root
- B3 adds 1 to the distance advertised by B2 and sends (B2, 1, B3) to B5
- Meanwhile B2 accepts B1 as root because it has the lower id and it sends (B1, 1, B2) toward B3
- B5 accepts B1 as root and sends (B1, 1, B5) to B3
- B3 accepts B1 as root and it notes that both B2 and B5 are closer to the root than it is.

- Thus B3 stops forwarding messages on both its interfaces
- This leaves B3 with both ports not selected

- Even after the system has stabilized, the root bridge continues to send configuration messages periodically
  - Other bridges continue to forward these messages
- When a bridge fails, the downstream bridges will not receive the configuration messages
- After waiting a specified period of time, they will once again claim to be the root and the algorithm starts again
- Note
  - Although the algorithm is able to reconfigure the spanning tree whenever a bridge fails, it is not able to forward frames over alternative paths for the sake of routing around a congested bridge
- Limitation of Bridges
  - Do not scale
    - Spanning tree algorithm does not scale
    - Broadcast does not scale
  - Do not accommodate heterogeneity

TEXT BOOK:

1. Larry L. Peterson, Bruce S. Davie, ―Computer Networks: A Systems Approach‖, Fifth Edition, Morgan Kaufmann Publishers, 2011.

REFERENCES:

1. James F. Kurose, Keith W. Ross, ―Computer Networking - A Top-Down Approach Featuring the Internet‖, Fifth Edition, Pearson Education, 2009.
2. Nader. F. Mir, ―Computer and Communication Networks‖, Pearson Prentice Hall Publishers, 2010.
3. Ying-Dar Lin, Ren-Hung Hwang, Fred Baker, ―Computer Networks: An Open Source Approach‖, Mc Graw Hill Publisher, 2011.
4. Behrouz A. Forouzan, ―Data communication and Networking‖, Fourth Edition, Tata McGraw – Hill, 2011.

tnScholars.com