

Subject: Computer Networks [2140709] <u>UNIT – 5 – THE LINK LAYER AND LOCAL AREA NETWORKS</u>	
	<p>➤ <u>Hub:</u></p> <ul style="list-style-type: none"> ▪ A hub is the simplest devices. ▪ Any data packet coming from one port is sent to all other ports. That's the sole purpose of a hub. ▪ Imagine packets going through a hub as messages going into a mailing list. The mail is sent out to everyone and it is up to the receiving party to decide if it is of interest. ▪ The biggest problem with hubs is their simplicity. Since every packet is sent out to every computer on the network, there is a lot of wasted transmission. This means that the network can easily become bogged down (congestion in network). ▪ Hubs are typically used on small networks where the amount of data going across the network is never very high. <p>➤ <u>Bridge:</u></p> <ul style="list-style-type: none"> ▪ A bridge goes one step up on a hub in that it review the destination of the packet before sending. If the destination address is not on the other side of the bridge it will not transmit the data. ▪ A bridge only has one incoming and one outgoing port. ▪ To build on the email analogy above, the bridge is allowed to decide if the message should continue on. It reads the address bob@smith.com and decides if there is a bob@smith.com on the other side. If there isn't, the message will not be transmitted. ▪ Bridges are typically used to separate parts of a network that do not need to communicate regularly, but still need to be connected. <p>➤ <u>Switch:</u></p> <ul style="list-style-type: none"> ▪ A switch steps up on a bridge in that it has multiple ports. ▪ When a packet comes through a switch it is read to determine which computer to send the data to. <i>i.e.</i> one computer will be the sole receiver of the package. ▪ This leads to increased efficiency in that packets are not going to computers that do not require them. ▪ Now the email analogy has multiple people able to send email to multiple users. The switch can decide where to send the mail based on the address. ▪ Most large networks use switches rather than hubs to connect computers within the same subnet. <p>➤ <u>Router:</u></p>

	<ul style="list-style-type: none"> ▪ A router is similar in a switch in that it forwards packets based on address. But, instead of the MAC address that a switch uses, a router can use the IP address. ▪ Before forwarding a packet the router will review the destination IP address. This allows the network to go across different protocols. ▪ The most common home use for routers is to share a broadband internet connection. ▪ The router has a public IP address and that address is shared with the network. When data comes through the router it is forwarded to the correct computer.
Q.1	What are the Services Provided by the Link Layer?
Ans	<ul style="list-style-type: none"> ▪ Possible services that can be offered by a link-layer protocol include: ▪ <u>Framing</u>. Almost all link-layer protocols encapsulate each network-layer datagram within a link-layer frame before transmission over the link. A frame consists of a data field, in which the network-layer datagram is inserted, and a number of header fields. The structure of the frame is specified by the link-layer protocol. ▪ <u>Link access</u>. A medium access control (MAC) protocol specifies the rules by which a frame is transmitted onto the link. For point-to-point links that have a single sender at one end of the link and a single receiver at the other end of the link, the MAC protocol is simple (or nonexistent)—the sender can send a frame whenever the link is idle. The more interesting case is when multiple nodes share a single broadcast link—the so-called multiple access problem. Here, the MAC protocol serves to coordinate the frame transmissions of the many nodes. ▪ <u>Reliable delivery</u>. When a link-layer protocol provides reliable delivery service, it guarantees to move each network-layer datagram across the link without error. Recall that certain transport-layer protocols (such as TCP) also provide a reliable delivery service. ▪ Similar to a transport-layer reliable delivery service, a link-layer reliable delivery service can be achieved with acknowledgments and retransmissions link-layer reliable delivery service is often used for links that are prone to high error rates, such as a wireless link, with the goal of correcting an error locally—on the link where the error occurs-rather than forcing an end-to-end retransmission of the data by a transport- or application-layer protocol. ▪ However, link-layer reliable delivery can be considered an unnecessary overhead for low bit-error links, including fiber, coax, and many twisted-pair copper links. For this reason, many wired link-layer protocols do not provide a reliable delivery service. ▪ <u>Error detection and correction</u>. The link-layer hardware in a receiving node can incorrectly decide that a bit in a frame is zero when it was transmitted as a one, and vice versa. Such bit errors are introduced by signal attenuation and electromagnetic noise. ▪ Because there is no need to forward a datagram that has an error, many link-layer protocols provide a mechanism to detect such bit errors. This is done by having the transmitting node include error-detection bits in the frame, and having the receiving node perform an error check. ▪ Error detection in the link layer is usually more sophisticated and is implemented in

	<p>hardware. Error correction is similar to error detection, except that a receiver not only detects when bit errors have occurred in the frame but also determines exactly where in the frame the errors have occurred (and then corrects these errors).</p>
Q.2	Discuss where is the Link Layer Implemented?
Ans	<ul style="list-style-type: none"> Figure 1 shows a typical host architecture. For the most part, the link layer is implemented in a network adapter, also sometimes known as a network interface card (NIC). At the heart of the network adapter is the link-layer controller, usually a single, special-purpose chip that implements many of the link-layer services (framing, link access, error detection, and so on). Thus, much of a link-layer controller's functionality is implemented in hardware. most network adapters were physically separate cards (such as a PCMCIA card or a plug-in card fitting into a PC's PCI card slot) but increasingly, network adapters are being integrated onto the host's motherboard—a so-called LAN-on-motherboard configuration.

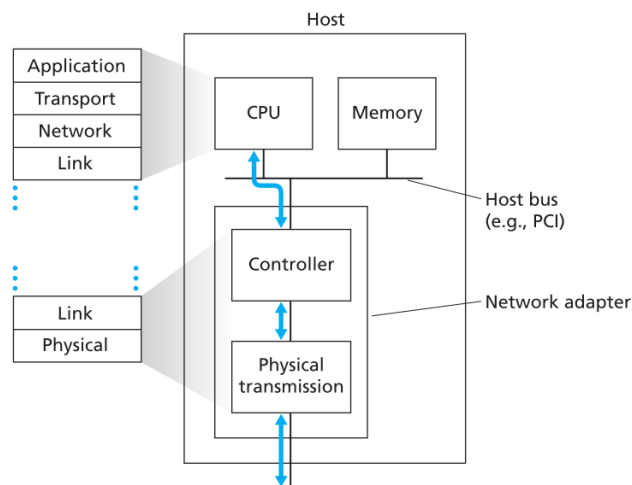
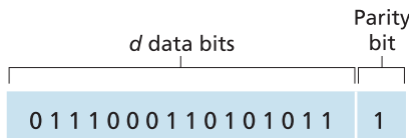


Figure 1 : Network adapter: its relationship to other host components and to protocol stack functionality

- On the sending side, the controller takes a datagram that has been created and stored in host memory by the higher layers of the protocol stack, encapsulates the datagram in a link-layer frame, and then transmits the frame into the communication link, following the link-access protocol.
- On the receiving side, a controller receives the entire frame, and extracts the network-layer datagram. If the link layer performs error detection, then it is the sending controller that sets the error-detection bits in the frame header and it is the receiving controller that performs error detection.
- Figure 1 shows a network adapter attaching to a host's bus (e.g., a PCI or PCI-X bus), where it looks much like any other I/O device to the other host components.

	<ul style="list-style-type: none"> Figure 1 also shows that while most of the link layer is implemented in hardware, part of the link layer is implemented in software that runs on the host's CPU. The software components of the link layer implement higher-level link-layer functionality such as assembling link-layer addressing information and activating the controller hardware. On the receiving side, link-layer software responds to controller interrupts (e.g., due to the receipt of one or more frames), handling error conditions and passing a datagram up to the network layer. Thus, the link layer is a combination of hardware and software—the place in the protocol stack where software meets hardware.
Q.3	Explain Error-Detection and -Correction Techniques performed at link layer using <i>parity checks</i> method.
Ans	<ul style="list-style-type: none"> Perhaps the simplest form of error detection is the use of a single parity bit. Suppose that the information to be sent, D in Figure 2, has d bits. In an even parity scheme, the sender simply includes one additional bit and chooses its value such that the total number of 1s in the d + 1 bits (the original information plus a parity bit) is even. For odd parity schemes, the parity bit value is chosen such that there is an odd number of 1s. Figure 2 illustrates an even parity scheme, with the single parity bit being stored in a separate field. <div style="text-align: center;">  </div> <p style="text-align: center;">Figure 2: One-bit even parity</p> <ul style="list-style-type: none"> Receiver operation is also simple with a single parity bit. The receiver need only count the number of 1s in the received d + 1 bits. If an odd number of 1-valued bits are found with an even parity scheme, the receiver knows that at least one bit error has occurred. More precisely, it knows that some odd number of bit errors have occurred. But what happens if an even number of bit errors occur? So this would result in an undetected error. If the probability of bit errors is small and errors can be assumed to occur independently from one bit to the next, the probability of multiple bit errors in a packet would be extremely small. In this case, a single parity bit might suffice. However, measurements have shown that, rather than occurring independently, errors are often clustered together in “bursts.” Under burst error conditions, the probability of undetected errors in a frame protected by single-bit parity can approach 50 percent. Clearly, a more robust error-detection scheme is needed. But before examining

error-detection schemes that are used in practice, let's consider a simple generalization of one-bit parity that will provide us with insight into error-correction techniques.

- Figure 3 shows a two-dimensional generalization of the single-bit parity scheme. Here, the d bits in D are divided into i rows and j columns. A parity value is computed for each row and for each column. The resulting $i + j + 1$ parity bits comprise the link-layer frame's error-detection bits.

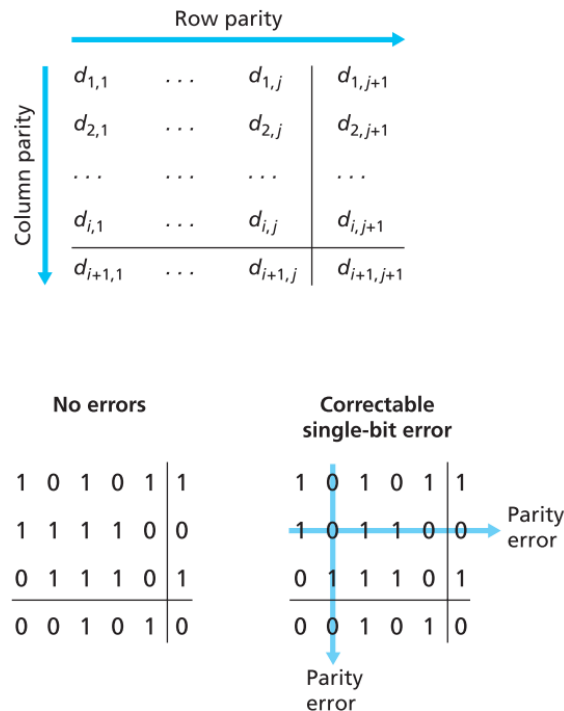


Figure 3: Two-dimensional even parity

- Suppose now that a single bit error occurs in the original d bits of information. With this two-dimensional parity scheme, the parity of both the column and the row containing the flipped bit will be in error.
- The receiver can thus not only detect the fact that a single bit error has occurred, but can use the column and row indices of the column and row with parity errors to actually identify the bit that was corrupted and correct that error! Figure 3 shows an example in which the 1-valued bit in position (2, 2) is corrupted and switched to a 0—an error that is both detectable and correctable at the receiver.

Q.4 Explain Cyclic Redundancy Check (CRC) with the help on an example.

- Ans**
- An error-detection technique used widely in today's computer networks is based on cyclic redundancy check (CRC) codes.
 - CRC codes operate as follows.
 - Consider the d -bit piece of data, D , that the sending node wants to send to the

receiving node. The sender and receiver must first agree on an $r + 1$ bit pattern, known as a generator, which we will denote as G . We will require that the most significant (leftmost) bit of G be a 1.

- The key idea behind CRC codes is shown in Figure 4.
- For a given piece of data, D , the sender will choose r additional bits, R , and append them to D such that the resulting $d + r$ bit pattern (interpreted as a binary number) is exactly divisible by G (i.e., has no remainder) using modulo-2 arithmetic.

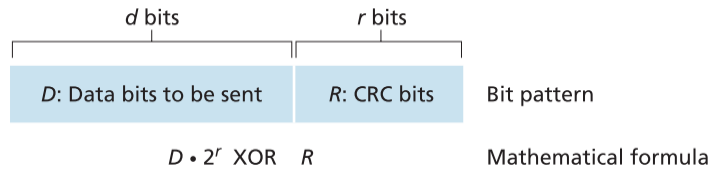


Figure 4: CRC

- The process of error checking with CRCs is thus simple: The receiver divides the $d + r$ received bits by G .
- If the remainder is nonzero, the receiver knows that an error has occurred; otherwise the data is accepted as being correct.
- This means that addition and subtraction are identical, and both are equivalent to the bitwise exclusive-or (XOR) of the operands. Thus, for example,

 $1011 \text{ XOR } 0101 = 1110$

 $1001 \text{ XOR } 1101 = 0100$
- Also, we similarly have

 $1011 - 0101 = 1110$

 $1001 - 1101 = 0100$
- That is, we want to choose R such that G divides into $D \cdot 2^r \text{ XOR } R$ without remainder. If we XOR (that is, add modulo-2, without carry) R to both sides of the above equation, we get

$$D \cdot 2^r \text{ XOR } R \text{ XOR } R$$

- This equation tells us that if we divide $D \cdot 2^r$ by G , the value of the remainder is precisely R . In other words, we can calculate R as

$$R = \text{remainder} \frac{D \cdot 2^r}{G}$$

- Below is the sample example of CRC calculation:

② Message = 101110
 $G = 1001$
 Sender:-

$$\begin{array}{r}
 1001 \overline{) 101110000} \quad (101011) \\
 \underline{1001} \\
 1010 \\
 \underline{1001} \\
 1100 \\
 \underline{1001} \\
 1010 \\
 \underline{1001} \\
 0011
 \end{array}$$

CRC = 011.

Fig 5.1 :A sample of CRC calculation at sender side.

Receiver:-

$$\begin{array}{r}
 1001 \overline{) 101110011} \quad (101011) \\
 \underline{1001} \\
 1000 \\
 \underline{1001} \\
 1101 \\
 \underline{1001} \\
 1001 \\
 \underline{1001} \\
 0000
 \end{array}$$

Fig 5.2: A sample CRC calculation at receiver side.

- In above both example $D=101110$, $d = 6$, $G = 1001$, and $r=3$.
- You should check these calculations for yourself and also check that indeed $D*2^r=101011 *G \text{ XOR } R$.
- Each of the CRC standards can detect burst errors of fewer than $r + 1$ bits. (This means that all consecutive bit errors of r bits or fewer will be detected.) Furthermore, under appropriate assumptions, a burst of length greater than $r + 1$ bits is detected with probability $1 - 0.5^r$. Also, each of the CRC standards can detect any odd number of bit errors.

Q.5	What is Multiple Access Links Protocols? Explain slotted ALOHA.
Ans	<ul style="list-style-type: none"> ▪ Multiple access protocols is a protocol by which nodes regulate their transmission into the shared broadcast channel. Multiple access protocols are needed in a wide variety of network settings, including both wired and wireless access networks, and satellite networks. ▪ Because all nodes are capable of transmitting frames, more than two nodes can transmit frames at the same time. When this happens, all of the nodes receive multiple frames at the same time; that is, the transmitted frames collide at all of the receivers. ▪ Typically, when there is a collision, none of the receiving nodes can make any sense of any of the frames that were transmitted. ▪ In order to ensure that the broadcast channel performs useful work when multiple nodes are active, it is necessary to somehow coordinate the transmissions of the active nodes. This coordination job is the responsibility of the multiple access protocol. ➤ Aloha: <ul style="list-style-type: none"> ▪ The slotted ALOHA protocol required that all nodes synchronize their transmissions to start at the beginning of a slot. The first ALOHA protocol was actually an unslotted, fully decentralized protocol. ▪ In pure ALOHA, when a frame first arrives (that is, a network-layer datagram is passed down from the network layer at the sending node), the node immediately transmits the frame in its entirety into the broadcast channel. If a transmitted frame experiences a collision with one or more other transmissions, the node will then immediately (after completely transmitting its collided frame) retransmit the frame with probability p. Otherwise, the node waits for a frame transmission time. After this wait, it then transmits the frame with probability p, or waits (remaining idle) for another frame time with probability $1 - p$. ▪ To determine the maximum efficiency of pure ALOHA, we focus on an individual node. We'll make the same assumptions as in our slotted ALOHA analysis and take the frame transmission time to be the unit of time. At any given time, the probability that a node is transmitting a frame is p. ▪ Suppose this frame begins transmission at time t_0. As shown in Figure 5.11, in order for this frame to be successfully transmitted, no other nodes can begin their transmission in the interval of time $[t_0 - 1, t_0]$. Such a transmission would overlap with the beginning of the transmission of node i's frame. ▪ The probability that all other nodes do not begin a transmission in this interval is $(1 - p)^{N-1}$. Similarly, no other node can begin a transmission while node i is transmitting, as such a transmission would overlap with the latter part of node i's transmission. The probability that all other nodes do not begin a transmission in this interval is also $(1 - p)^{N-1}$. ▪ Thus, the probability that a given node has a successful transmission is $p(1 - p)^{2(N-1)}$.

	By taking limits as in the slotted ALOHA case, we find that the maximum efficiency of the pure ALOHA protocol is only <u>$1/(2e)$—exactly half that of slotted ALOHA.</u>
Q.6	Explain slotted ALOHA channel access technique.
Ans	<ul style="list-style-type: none"> ▪ <u>The slotted ALOHA is a random access protocol.</u> In description of slotted ALOHA, we assume the following: <ol style="list-style-type: none"> 1. All frames consist of exactly <u>L</u> bits. 2. Time is divided into slots of size <u>L/R seconds</u> (that is, a slot equals the time to transmit one frame). 3. Nodes start to transmit frames only at the <u>beginnings of slots.</u> 4. The nodes are <u>synchronized</u> so that each node knows when the slots begin. 5. If two or more frames collide in a slot, then all the nodes detect the collision Event before the slot ends. ▪ Let <u>p be a probability</u>, that is, a <u>number between 0 and 1</u>. The operation of slotted ALOHA in each node is simple: <ol style="list-style-type: none"> 1. When the node has a fresh frame to send, it waits until the beginning of the next slot and transmits the entire frame in the slot. 2. If there isn't a collision, the node has successfully transmitted its frame and thus need not consider retransmitting the frame. (The node can prepare a new frame for transmission, if it has one.) 3. If there is a collision, the node detects the collision before the end of the slot. The node retransmits its frame in each subsequent slot with probability p until the frame is transmitted without a collision. ▪ By retransmitting with probability p, we mean that the node effectively tosses a biased coin; the event heads corresponds to “retransmit,” which occurs with probability p. ▪ The event tails corresponds to “skip the slot and toss the coin again in the next slot”; this occurs with probability $(1 - p)$. All nodes involved in the collision toss their coins independently. ▪ Slotted ALOHA would appear to have many advantages. Unlike channel partitioning, slotted ALOHA allows a node to transmit continuously at the full rate, R, when that node is the only active node. (A node is said to be active if it has frames to send.) ▪ Slotted <u>ALOHA is also highly decentralized</u>, because each node detects collisions and independently decides when to retransmit. <u>Slotted ALOHA is also an extremely simple protocol.</u> ▪ Slotted ALOHA works well when there is only one active node, but how efficient is it when there are multiple active nodes? There are two possible efficiency concerns here. ▪ First, as shown in Figure when there are multiple active nodes, a certain fraction of the slots will have collisions and will therefore be “wasted.” ▪ The second concern is that another fraction of the slots will be empty because all active nodes refrain from transmitting as a result of the probabilistic transmission policy.

- The only “unwanted” slots will be those in which exactly one node transmits. A slot in which exactly one node transmits is said to be a successful slot.
- The efficiency of a slotted multiple access protocol is defined to be the long-run fraction of successful slots in the case when there are a large number of active nodes, each always having a large number of frames to send.

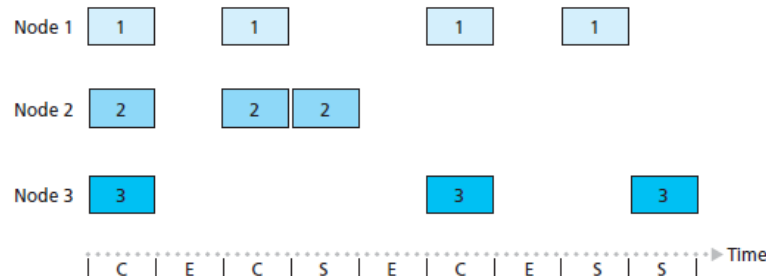


Figure 5: Nodes 1, 2, and 3 collide in the first slot. Node 2 finally succeeds in the fourth slot, node 1 in the eighth slot, and node 3 in the ninth slot

Key:

C = Collision slot

E = Empty slot

S = Successful slot

- If no forms of access control were used, and each node was to immediately retransmit after each collision, the efficiency would be zero. Slotted ALOHA clearly increases the efficiency beyond zero, but by how much?
- We now proceed to outline the derivation of the maximum efficiency of slotted ALOHA. To keep this derivation simple, let’s modify the protocol a little and assume that each node attempts to transmit a frame in each slot with **probability p** . (That is, we assume that each node always has a frame to send and that the node transmits with probability p for a fresh frame as well as for a frame that has already suffered a collision.)
- Suppose there are N nodes.
- Then the probability that a given slot is a successful slot is the probability that one of the nodes transmits and that the remaining $N - 1$ nodes do not transmit.
- The probability that a given node **transmits** is p .
- The probability that the remaining nodes **do not transmit** is $(1 - p)^{N-1}$.
- Therefore the **probability** a given node has a **success** is $p(1 - p)^{N-1}$. Because there are N nodes, the probability that **any one of the N nodes** has a success is $Np(1 - p)^{N-1}$.
- Thus, when there are N **active** nodes, the **efficiency of slotted ALOHA** is $Np(1 - p)^{N-1}$.
- To obtain the **maximum efficiency** for N active nodes, we have to find the p^* that maximizes this expression.
- And to obtain the **maximum efficiency** for a large number of active nodes, we take the limit of $Np^*(1 - p^*)^{N-1}$ as N approaches infinity.
- After performing these calculations, we’ll find that the maximum efficiency of the protocol is given by **$1/e = 0.37$** . That is, when a large number of nodes have many

	<p>frames to transmit, then (at best) only 37 percent of the slots do useful work.</p> <ul style="list-style-type: none"> ▪ Thus the effective transmission rate of the channel is not R bps but only $0.37 R$bps! A similar analysis also shows that 37 percent of the slots go empty and 26 percent of slots have collisions. ▪ Imagine the poor network administrator who has purchased a 100-Mbps slotted ALOHA system, expecting to be able to use the network to transmit data among a large number of users at an aggregate rate of, say, 80 Mbps! Although the channel is capable of transmitting a given frame at the full channel rate of 100 Mbps, in the long run, the successful throughput of this channel will be less than 37 Mbps.
Q.7	Explain Carrier Sense Multiple Access (CSMA).
Ans	<ul style="list-style-type: none"> ▪ In both slotted and pure ALOHA, a node's decision to transmit is made independently of the activity of the other nodes attached to the broadcast channel. In particular, a node neither pays attention to whether another node happens to be transmitting when it begins to transmit, nor stops transmitting if another node begins to interfere with its transmission. ▪ The first question that you might ask about CSMA is why, if all nodes perform carrier sensing, do collisions occur in the first place? After all, a node will refrain from transmitting whenever it senses that another node is transmitting. The answer to the question can best be illustrated using space-time diagrams. ▪ Figure 7 shows a space-time diagram of four nodes (A, B, C, D) attached to a linear broadcast bus. The horizontal axis shows the position of each node in space; the vertical axis represents time.

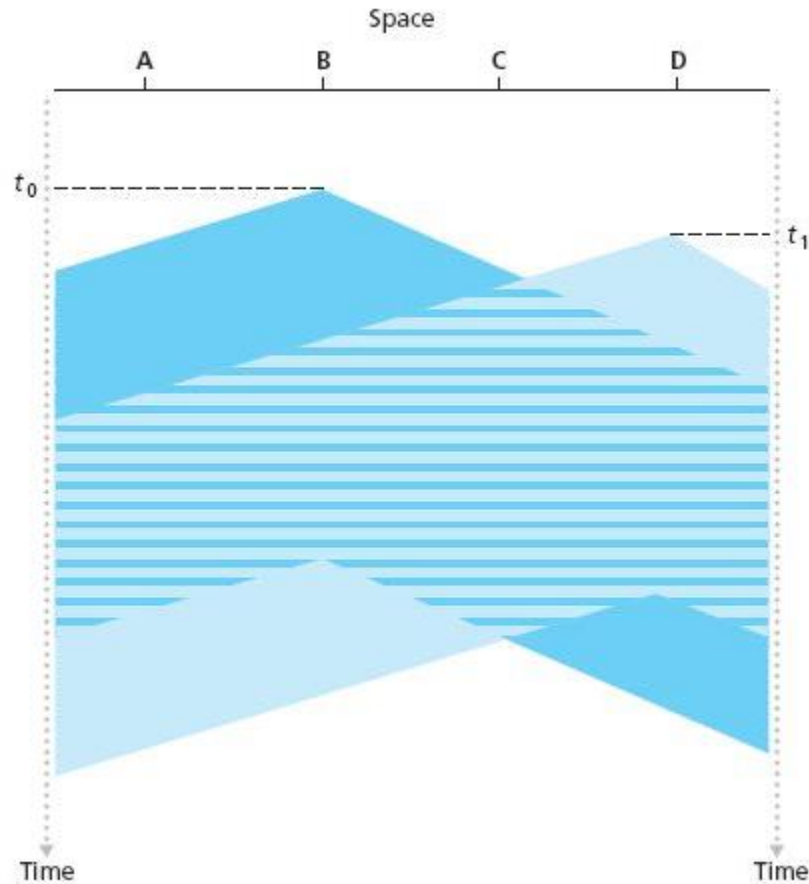


Figure 6: space-time diagram

- At time t_0 , node B senses the channel is idle, as no other nodes are currently transmitting. Node B thus begins transmitting, with its bits propagating in both directions along the broadcast medium. The downward propagation of B's bits in Figure 7 with increasing time indicates that a nonzero amount of time is needed for B's bits actually to propagate (albeit at near the speed of light) along the broadcast medium.
- At time t_1 ($t_1 > t_0$), node D has a frame to send. Although node B is currently transmitting at time t_1 , the bits being transmitted by B have yet to reach D, and thus D senses the channel idle at t_1 . In accordance with the CSMA protocol, D thus begins transmitting its frame.
- A short time later, B's transmission begins to interfere with D's transmission at D. From Figure 7, it is evident that the end-to-end channel propagation delay of a broadcast channel—the time it takes for a signal to propagate from one of the nodes to another—will play a crucial role in determining its performance. The longer this propagation delay, the larger the chance that a carrier-sensing node is not yet able to sense a transmission that has already begun at another node in the network.

Q.8	Explain CSMA/CD Protocol.
Ans	<ul style="list-style-type: none"> ▪ In Figure 7, nodes do not perform collision detection; both B and D continue to transmit their frames in their entirety even though a collision has occurred. When a node performs collision detection, it ceases transmission as soon as it detects a collision. ▪ Figure 8 shows the same scenario as in Figure 7, except that the two nodes each abort their transmission a short time after detecting a collision. ▪ Clearly, adding collision detection to a multiple access protocol will help protocol performance by not transmitting a useless, damaged (by interference with a frame from another node) frame in its entirety. ▪ Before analyzing the CSMA/CD protocol, let us now summarize its operation from the perspective of an adapter (in a node) attached to a broadcast channel: ▪ The adapter obtains a datagram from the network layer, prepares a link-layer frame, and puts the frame adapter buffer. ▪ If the adapter senses that the channel is idle (that is, there is no signal energy entering the adapter from the channel), it starts to transmit the frame. If, on the other hand, the adapter senses that the channel is busy, it waits until it senses no signal energy and then starts to transmit the frame. ▪ While transmitting, the adapter monitors for the presence of signal energy coming from other adapters using the broadcast channel. ▪ If the adapter transmits the entire frame without detecting signal energy from other adapters, the adapter is finished with the frame. If, on the other hand, the adapter detects signal energy from other adapters while transmitting, it aborts the transmission (that is, it stops transmitting its frame). ▪ After aborting, the adapter waits a random amount of time and then returns to step 2.

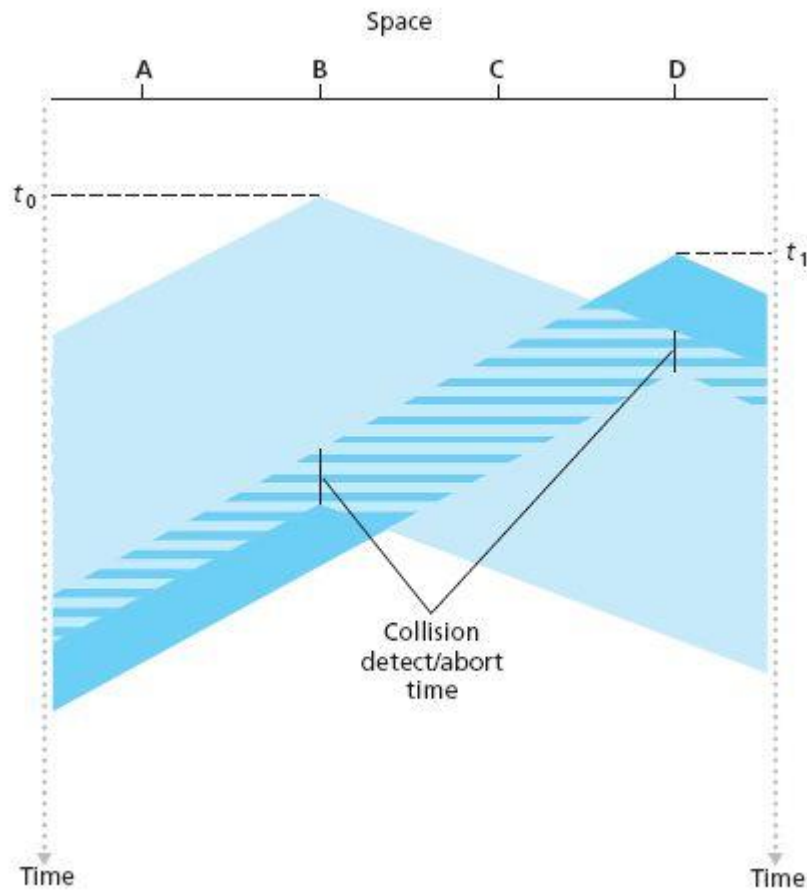


Figure 7 : CSMA with collision detection

- The need to wait a random (rather than fixed) amount of time is hopefully clear—if two nodes transmitted frames at the same time and then both waited the same fixed amount of time, they'd continue colliding forever.
- But what is a good interval of time from which to choose the random back off time? If the interval is large and the number of colliding nodes is small, nodes are likely to wait a large amount of time (with the channel remaining idle) before repeating the sense-and-transmit when- idle step.
- On the other hand, if the interval is small and the number of colliding nodes is large, it's likely that the chosen random values will be nearly the same, and transmitting nodes will again collide.
- What we'd like is an interval that is short when the number of colliding nodes is small, and long when the number of colliding nodes is large.
- The **binary exponential back off** algorithm, used in Ethernet as well as in DOCSIS cable network multiple access protocols.
- **Binary Exponential Back Off Algorithm**
 - Specifically, when transmitting a frame that has already experienced n collisions, a node chooses the value of K at random from $\{0, 1, 2, \dots, 2^{n-1}\}$. Thus, the more collisions experienced by a frame, the larger the interval from which K is chosen. For Ethernet, the actual amount of time a node waits is $K \cdot 512$ bit times (i.e. K times the

	<p>amount of time needed to send 512 bits into the Ethernet) and the maximum value that n can take is capped at 10.</p> <ul style="list-style-type: none"> Let's look at an example. Suppose that a node attempts to transmit a frame for the first time and while transmitting it detects a collision. The node then chooses K_0 with probability 0.5 or chooses K_1 with probability 0.5. If the node chooses K_0, then it immediately begins sensing the channel. If the node chooses K_1, it waits 512 bit times (e.g., 0.01 microseconds for a 100 Mbps Ethernet) before beginning the sense-and-transmit-when-idle cycle. After a second collision, K is chosen with equal probability from {0, 1, 2, 3}. After three collisions, K is chosen with equal probability from {0, 1, 2, 3, 4, 5, 6, 7}. After 10 or more collisions, K is chosen with equal probability from {0, 1, 2 . . . 1023}. Thus, the size of the sets from which K is chosen grows exponentially with the number of collisions; for this reason this algorithm is referred to as binary exponential back off. We also note here that each time a node prepares a new frame for transmission, it runs the CSMA/CD algorithm, not taking into account any collisions that may have occurred in the recent past. So it is possible that a node with a new frame will immediately be able to sneak in a successful transmission while several other nodes are in the exponential back off state. <p>➤ <u>CSMA/CD Efficiency:</u></p> <ul style="list-style-type: none"> When only one node has a frame to send, the node can transmit at the full channel rate (e.g., for Ethernet typical rates are 10 Mbps, 100 Mbps, or 1 Gbps). However, if many nodes have frames to transmit, the effective transmission rate of the channel can be much less. <i>"We define the efficiency of CSMA/CD to be the long-run fraction of time during which frames are being transmitted on the channel without collisions when there is a large number of active nodes, with each node having a large number of frames to send. "</i> In order to present a closed-form approximation of the efficiency of Ethernet, let d_{prop} denote the maximum time it takes signal energy to propagate between any two adapters. Let d_{trans} be the time to transmit a maximum-size frame (approximately 1.2 msec for a 10 Mbps Ethernet): $\text{Efficiency} = \frac{1}{1 + 5d_{prop}/d_{trans}}$ <ul style="list-style-type: none"> We see from this formula that as d_{prop} approaches 0, the efficiency approaches 1. This matches our intuition that if the propagation delay is zero, colliding nodes will abort immediately without wasting the channel. Also, as d_{trans} becomes very large, efficiency approaches 1. This is also intuitive because when a frame grabs the channel, it will hold on to the channel for a very long time; thus, the channel will be doing productive work most of the time.
Q.9	Explain both the Taking-Turns Protocols in details.

Ans	<ul style="list-style-type: none"> ▪ Two desirable properties of a multiple access protocol are ▪ When only one node is active, <u>the active node has a throughput of R bps</u>, and ▪ When M nodes are active, and then each active node has <u>a throughput</u> of nearly <u>R/M bps</u>. ▪ This properties of protocols are called the taking-turns protocols. ▪ There are various categories of the taking-turns protocols, ▪ Polling Protocol ▪ Token-Passing Protocol <ul style="list-style-type: none"> ▪ <u>Polling Protocol:</u> ▪ The polling protocol requires one of the nodes to be designated as a <u>master node</u>. The master node polls each of the nodes in a <u>round-robin fashion</u>. In particular, the master node first sends a message to node 1, saying that it (node 1) can transmit up to some maximum number of frames. After node 1 transmits some frames, the master node tells node 2 it (node 2) can transmit up to the maximum number of frames. (The master node can determine when a node has finished sending its frames by observing the lack of a signal on the channel.) ▪ The procedure continues in this manner, with the master node polling each of the nodes in a cyclic manner. The polling protocol eliminates the collisions and empty slots that plague random access protocols. This allows polling to achieve a much higher efficiency. ▪ But it also has a few drawbacks. The first drawback is that the protocol introduces a <u>polling delay</u>—the amount of time required to notify a node that it can transmit. If, for example, only one node is active, then the node will transmit at a rate less than R bps, as the master node must poll each of the inactive nodes in turn each time the active node has sent its maximum number of frames. ▪ The second drawback, which is potentially more serious, is that if the master node fails, the entire channel becomes inoperative. <u>The second taking-turns protocol is the token-passing protocol.</u> ➤ <u>Token-Passing Protocol:</u> ▪ The second taking-turns protocol is the token-passing protocol. In this protocol there is <u>no master node</u>. ▪ <u>A small, special-purpose frame known as a token</u> is exchanged among the nodes in some fixed order. For example, node 1 might always send the token to node 2, node 2 might always send the token to node 3, and node N might always send the token to node 1. When a node receives a token, it holds onto the token only if it has some frames to transmit; otherwise, it immediately forwards the token to the next node. ▪ If a node does have frames to transmit when it receives the token, it sends up to a maximum number of frames and then forwards the token to the next node. ▪ <u>Token passing is decentralized and highly efficient</u>. But it has its problems as well. For example, the failure of one node can crash the entire channel. Or if a node accidentally neglects to release the token, then some recovery procedure must be
-----	--

	<p>invoked to get the token back in circulation.</p> <ul style="list-style-type: none"> Over the years many token-passing protocols have been developed, including the <u>fiber distributed data interface (FDDI) protocol</u> and the <u>IEEE 802.5 token ring protocol</u>.
Q.10	<p>Explain ARP and justify why ARP query sent within a broadcast frame and ARP response sent within a frame with specific destination MAC address?</p>
Ans	<ul style="list-style-type: none"> There are both network-layer addresses (for example, Internet IP addresses) and link-layer addresses (i.e., MAC addresses), it needed to translate between both the layers. For the Internet, this is the job of the Address Resolution Protocol (ARP). To understand the need for a protocol such as ARP, consider the network shown in Figure 9. <div data-bbox="513 785 1359 1289" data-label="Diagram"> </div> <p>Figure 8: Each interface on a LAN has an IP address and a MAC address</p> <ul style="list-style-type: none"> Now that we have explained what ARP does, let's look at how it works. Each host and router has an ARP table in its memory, which contains mappings of IP addresses to MAC addresses. Figure 10 shows how the ARP table in host 222.222.222.220 might look like. The ARP table also contains a time-to-live (TTL) value, which indicates when each mapping will be deleted from the table. Note that, it is not necessary that a table contain an entry for every host and router on the subnet; some may have never been entered into table, and others may have expired. A typical expiration time for an entry is 20 minutes from when an entry is placed in an ARP table.


IP Address	MAC Address	TTL
222.222.222.221	88-B2-2F-54-1A-0F	13:45:00
222.222.222.223	5C-66-AB-90-75-B1	13:52:00

Figure 9: A possible ARP table in 222.222.222.220

- Now suppose that host 222.222.222.220 wants to send a datagram that is IP addressed to another host or router on that subnet. The sending host needs to obtain the MAC address of the destination given the IP address.
- This task is easy if the sender's ARP table has an entry for the destination node. But what if the ARP table doesn't currently have an entry for the destination? In particular, suppose 222.222.222.220 wants to send a datagram to 222.222.222.222.
- In this case, the sender uses the ARP protocol to resolve the address.
 - a) First, the sender constructs a special packet called an **ARP packet**. An ARP packet has several fields, including the sending and receiving IP and MAC addresses.
 - b) Both ARP query and response packets have the same format. The purpose of the ARP query packet is to query all the other hosts and routers on the subnet to determine the MAC address corresponding to the IP address that is being resolved.
- Returning to our example, 222.222.222.220 passes an ARP query packet to the adapter along with an indication that the adapter should send the packet to the MAC broadcast address, namely, FF-FF-FF-FF-FF-FF. The adapter encapsulates the ARP packet in a link-layer frame, uses the broadcast address for the frame's destination address, and transmits the frame into the subnet.
- *The frame containing the ARP query is received by all the other adapters on the subnet, and (because of the broadcast address) each adapter passes the ARP packet within the frame up to its ARP module.*
- *Each of these ARP modules checks whether, its IP address matches the destination IP address in the ARP packet. The one with a match sends back to the querying host a response ARP packet with the desire mapping.*
- The querying host 222.222.222.220 can then update, its ARP table and send its IP datagram encapsulated in a link-layer frame whose destination MAC is that of the host or router responding to the earlier ARP query.
- So ARP protocol works in this manner, the query ARP message is sent within a broadcast frame, whereas the response ARP message is sent within a standard frame. So this we have seen in above discussion.
- ARP is known as **plug-and-play**; (i.e., an ARP table gets built automatically) it doesn't have to be configured by a system administrator. And if a host becomes disconnected from the subnet, its entry is eventually deleted from the other ARP tables in the subnet.

Q.11 How packet can be transmitted when destination is off the subnet?

	<p>OR</p> <p>How packet can be transmitted when destination is on other network?</p>
Ans	<ul style="list-style-type: none"> Let's look at the more complicated situation when a host on a subnet wants to send a network-layer datagram to a host off the subnet (that is, across a router onto another subnet). Let's discuss this issue in the context of Figure 5.19, which shows a simple network consisting of two subnets interconnected by a router. There are several interesting things to note about Figure 5.19. Each host has exactly one IP address and one adapter. A router has an IP address for each of its interfaces. For each router interface there is also an ARP module (in the router) and an adapter. Because the router has two interfaces, it has two IP addresses, two ARP modules, and two adapters. Each adapter in the network has its own MAC address. <div data-bbox="352 728 1440 1016" data-label="Diagram"> </div> <p style="text-align: center;">Figure 10: Two subnets interconnected by a router</p> <ul style="list-style-type: none"> Also note that Subnet 1 has the network address 111.111.111/24 and that Subnet 2 has the network address 222.222.222/24. Thus all of the interfaces connected to Subnet 1 have addresses of the form 111.111.111.xxx and all of the interfaces connected to Subnet 2 have addresses of the form 222.222.222.xxx. Now let's examine how a host on Subnet 1 would send a datagram to a host on Subnet 2. Suppose that host 111.111.111.111 wants to send an IP datagram to a host 222.222.222.222. The sending host passes the datagram to its adapter. But the sending host must also indicate to its adapter an appropriate destination MAC address. If the sending adapter want to use that MAC address, then none of the adapters on Subnet 1 would bother to pass the IP datagram up to its network layer. If the frame's destination address would not match the MAC address of any adapter on Subnet 1. The datagram would just die. If we look carefully at above Figure, we see that for a datagram to go from 111.111.111.111 to a host on Subnet 2, the datagram must first be sent to the router interface 111.111.111.110, which is the IP address of the first-hop router on the path to the final destination. Thus, the appropriate MAC address for the frame is the address of the adapter for router interface 111.111.111.110, namely, E6-E9-00-17-BB-4B. How does the sending host acquire the MAC address for 111.111.111.110?

	<ul style="list-style-type: none"> By using ARP. Once the sending adapter has this MAC address, it creates a frame (containing the datagram addressed to 222.222.222.222) and sends the frame into Subnet 1. The router adapter on Subnet 1 sees that the link-layer frame is addressed to it, and therefore passes the frame to the network layer of the router. Thus, the IP datagram has successfully been moved from source host to the router. But it is not finished yet. We should move the datagram from the router to the destination. The router should determine the correct interface on which the datagram is to be forwarded. The forwarding table tells the router that the datagram is to be forwarded via router interface 222.222.222.220. This interface then passes the datagram to its adapter, which encapsulates the datagram in a new frame and sends the frame into Subnet 2. This time, the destination MAC address of the frame is indeed the MAC address of the ultimate destination. And the router obtains destination MAC address using ARP protocol.
Q.12	Explain Ethernet Frame structure.
Ans	<div style="text-align: center;">  </div> <ul style="list-style-type: none"> Let's consider sending an IP datagram from one host to another host, with both hosts on the same Ethernet LAN Although the payload of our Ethernet frame is an IP datagram, we note that an Ethernet frame can carry other network-layer packets as well.) Let the sending adapter, adapter A, have the MAC address AA-AA-AA-AA-AA-AA and the receiving adapter, adapter B, have the MAC address BB-BB-BB-BB-BB-BB. The sending adapter encapsulates the IP datagram within an Ethernet frame and passes the frame to the physical layer. The receiving adapter receives the frame from the physical layer, extracts the IP datagram, and passes the IP datagram to the network layer. <p>➤ Data field (46 to 1,500 bytes).</p> <ul style="list-style-type: none"> This field carries the IP datagram. The maximum transmission unit (MTU) of Ethernet is 1,500 bytes. This means that if the IP datagram exceeds 1,500 bytes, then the host has to fragment the datagram, as discussed in Section 4.4.1. The minimum size of the data field is 46 bytes. This means that if the IP datagram is less than 46 bytes, the data field has to be “stuffed” to fill it out to 46 bytes. When stuffing is used, the data passed to the network layer contains the stuffing as well as an IP datagram. The network layer uses the length field in the IP datagram header to remove the stuffing.

	<ul style="list-style-type: none"> ➤ Destination address (6 bytes). <ul style="list-style-type: none"> ▪ This field contains the MAC address of the destination adapter, BB-BB-BB-BB-BB-BB. When adapter B receives an Ethernet frame whose destination address is either BB-BB-BB-BB-BB-BB or the MAC broadcast address, it passes the contents of the frame's data field to the network layer; if it receives a frame with any other MAC address, it discards the frame. ➤ Source address (6 bytes) <ul style="list-style-type: none"> ▪ This field contains the MAC address of the adapter that transmits the frame onto the LAN, in this example, AA-AA-AA-AA-AA-AA. ➤ Type field (2 bytes) <ul style="list-style-type: none"> ▪ The type field permits Ethernet to multiplex network-layer protocols. To understand this, we need to keep in mind that hosts can use other network-layer protocols besides IP. In fact, a given host may support multiple. <ul style="list-style-type: none"> ▪ Network-layer protocols using different protocols for different applications. ▪ For this reason, when the Ethernet frame arrives at adapter B, adapter B needs to know to which network-layer protocol it should pass (that is, de-multiplex) the contents of the data field. IP and other network-layer protocols (for example, Novell IPX or AppleTalk) each have their own, standardized type number. Furthermore, the ARP protocol (discussed in the previous section) has its own type number, and if the arriving frame contains an ARP packet (i.e., has a type field of 0806 hexadecimal), the ARP packet will be de-multiplexed up to the ARP protocol. ▪ Note that the type field is analogous to the protocol field in the network layer datagram and the port-number fields in the transport-layer segment; all of these fields serve to glue a protocol at one layer to a protocol at the layer above. ➤ Cyclic redundancy check (CRC) (4 bytes) <ul style="list-style-type: none"> ▪ The purpose of the CRC field is to allow the receiving adapter, adapter B, to detect bit errors in the frame. ➤ Preamble (8 bytes) <ul style="list-style-type: none"> ▪ The Ethernet frame begins with an 8-byte preamble field. Each of the first 7 bytes of the preamble has a value of 10101010; the last byte is 10101011. The first 7 bytes of the preamble serve to “wake up” the receiving adapters and to synchronize their clocks to that of the sender's clock. That adapter A aims to transmit the frame at 10 Mbps, 100 Mbps, or 1 Gbps, depending on the type of Ethernet LAN. However, because nothing is absolutely perfect, adapter A will not transmit the frame at exactly the target rate; there will always be some drift from the target rate, a drift which is not known a priori by the other adapters on the LAN. <ul style="list-style-type: none"> ▪ A receiving adapter can lock onto adapter A's clock simply by locking onto the bits in the first 7 bytes of the preamble. The last 2 bits of the eighth byte of the preamble (the first two consecutive 1s) alert adapter B that the “important stuff” is about to come.

Q.13	Explain working of Link-Layer Switches and also explain the self-learning properties of switches.												
Ans	<ul style="list-style-type: none">▪ The role of the switch is to receive incoming link-layer frames and forward them onto outgoing links. The switch itself is transparent to the hosts and routers in the subnet; that is, a host/router addresses a frame to another host/router (rather than addressing the frame to the switch) and happily sends the frame into the LAN, unaware that a switch will be receiving the frame and forwarding it.▪ The rate at which frames arrive to any one of the switch’s output interfaces may temporarily exceed the link capacity of that interface. To accommodate this problem, switch output interfaces have buffers, in much the same way that router output interfaces have buffers for datagrams. <p>➤ <u>Forwarding and Filtering:</u></p> <ul style="list-style-type: none">▪ Filtering is the switch function that determines whether a frame should be forwarded to some interface or should just be dropped.▪ Forwarding is the switch function that determines the interfaces to which a frame should be directed, and then moves the frame to those interfaces. Switch filtering and forwarding are done with a switch table.▪ The switch table contains entries for some, but not necessarily all, of the hosts and routers on a LAN. An entry in the switch table contains<ul style="list-style-type: none">i. a MAC address,ii. the switch interface that leads toward that MAC address, andiii. The time at which the entry was placed in the table. <table><tr><th>Address</th><th>Interface</th><th>Time</th></tr><tr><td>62-FE-F7-11-89-A3</td><td>1</td><td>9:32</td></tr><tr><td>7C-BA-B2-B4-91-10</td><td>3</td><td>9:36</td></tr><tr><td>....</td><td>....</td><td>....</td></tr></table>	Address	Interface	Time	62-FE-F7-11-89-A3	1	9:32	7C-BA-B2-B4-91-10	3	9:36
Address	Interface	Time											
62-FE-F7-11-89-A3	1	9:32											
7C-BA-B2-B4-91-10	3	9:36											
....											

Figure 11: Portion of a switch table for the uppermost switch in Fig-13

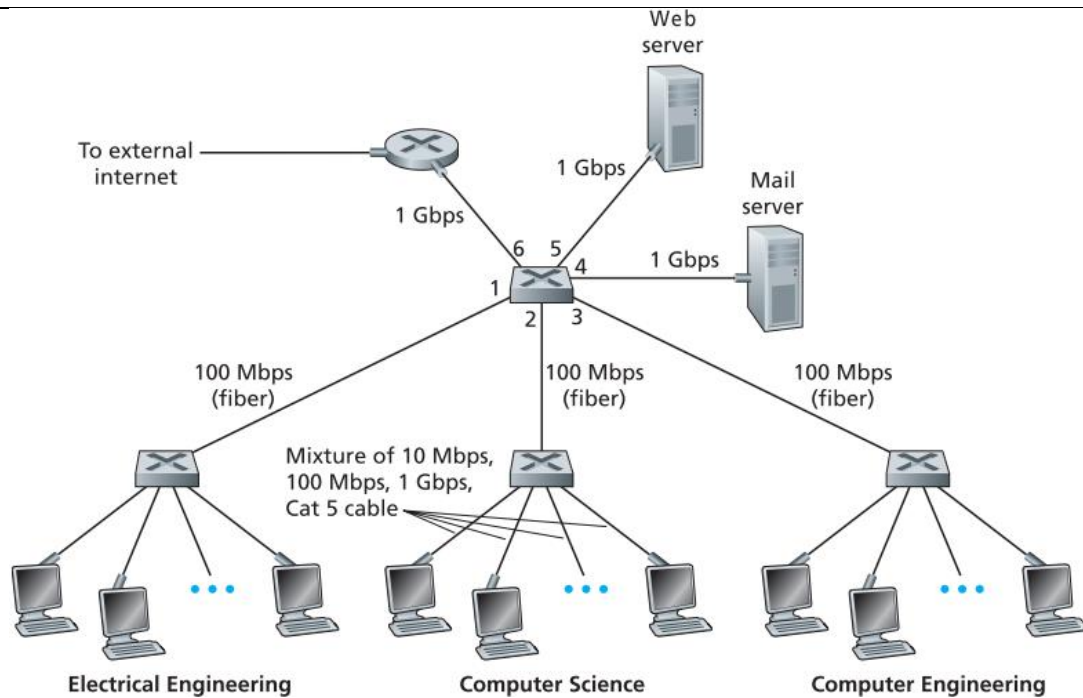


Figure 12: An institutional network connected together by four switches

- An example switch table for the uppermost switch in Figure 13 is shown in Figure 12. One important difference is that switches forward packets based on MAC addresses rather than on IP addresses. We will also see that a switch table is constructed in a very different manner from a router's forwarding table.
- To understand how switch filtering and forwarding work, suppose a frame with destination address DD-DD-DD-DD-DD-DD arrives at the switch on interface x. The switch indexes its table with the MAC address DD-DD-DD-DD-DD-DD. There are three possible cases:
 - ✓ There is no entry in the table for DD-DD-DD-DD-DD-DD. In this case, the switch forwards copies of the frame to the output buffers preceding all interfaces except for interface x. In other words, if there is no entry for the destination address, the switch broadcasts the frame.
 - ✓ There is an entry in the table, associating DD-DD-DD-DD-DD-DD with interface x. In this case, the frame is coming from a LAN segment that contains adapter DD-DD-DD-DD-DD-DD. There being no need to forward the frame to any of the other interfaces, the switch performs the filtering function by discarding the frame.
 - ✓ There is an entry in the table, associating DD-DD-DD-DD-DD-DD with interface y ≠ x. In this case, the frame needs to be forwarded to the LAN segment attached to interface y. The switch performs its forwarding function by putting the frame in an output buffer that precedes interface y.
- Let's walk through these rules for the uppermost switch in Figure 13 and its switch table in Figure 12. Suppose that a frame with destination address 62-FE-F7-11-89-A3

arrives at the switch from interface 1. The switch examines its table and sees that the destination is on the LAN segment connected to interface 1 (that is, Electrical Engineering).

- This means that the frame has already been broadcast on the LAN segment that contains the destination. The switch therefore filters (that is, discards) the frame. Now suppose a frame with the same destination address arrives from interface 2.
- The switch again examines its table and sees that the destination is in the direction of interface 1; it therefore forwards the frame to the output buffer preceding interface 1. It should be clear from this example that as long as the switch table is complete and accurate, the switch forwards frames towards destinations without any broadcasting.
- In this sense, a switch is “smarter” than a hub. But how does this switch table get configured in the first place.

➤ **Self-Learning:**

- A switch has the wonderful property (particularly for the already-overworked network administrator) that its table is built automatically, dynamically, and autonomously—without any intervention from a network administrator or from a configuration protocol.
- In other words, switches are self-learning. This capability is accomplished as follows:
 1. The switch table is initially empty.
 2. For each incoming frame received on an interface, the switch stores in its table
 - I. The MAC address in the frame’s source address field,
 - II. The interface from which the frame arrived, and
 - III. The current time.

In this manner the switch records in its table the LAN segment on which the sender resides. If every host in the LAN eventually sends a frame, then every host will eventually get recorded in the table.
 3. The switch deletes an address in the table if no frames are received with that address as the source address after some period of time (the aging time). In this manner, if a PC is replaced by another PC (with a different adapter), the MAC address of the original PC will eventually be purged from the switch table.

- Let’s walk through the self-learning property for the uppermost switch in Figure 13 and its corresponding switch table in Figure 12.
- Suppose at time 9:39 a frame with source address 01-12-23-34-45-56 arrives from interface 2. Suppose that this address is not in the switch table. Then the switch adds a new entry to the table, as shown in Figure 14.
- Continuing with this same example, suppose that the aging time for this switch is 60 minutes, and no frames with source address 62-FE-F7-11-89-A3 arrive to the switch between 9:32 and 10:32. Then at time 10:32, the switch removes this address from its table.

Address	Interface	Time
01-12-23-34-45-56	2	9:39
62-FE-F7-11-89-A3	1	9:32
7C-BA-B2-B4-91-10	3	9:36
....

Figure 13: Switch learns about the location of an adapter with address 01-12-23-34-45-56

- Switches are **plug-and-play devices** because they require no intervention from a network administrator or user. A network administrator wanting to install a switch need do nothing more than connect the LAN segments to the switch interfaces.
- The administrator need not configure the switch tables at the time of installation or when a host is removed from one of the LAN segments. Switches are also full-duplex, meaning any switch interface can send and receive at the same time.

Q.14 Explain the Properties of Link-Layer Switching.

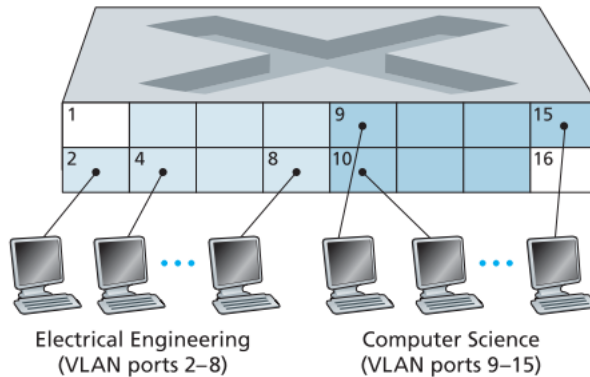
- Ans**
- Having described the basic operation of a link-layer switch, let's now consider their features and properties.
 - Elimination of collisions.** In a LAN built from switches (and without hubs), there is no wasted bandwidth due to collisions!
 - The switches buffer frames and never transmit more than one frame on a segment at any one time. As with a router, the maximum aggregate throughput of a switch is the sum of all the switch interface rates.
 - Thus, switches provide a significant performance improvement over LANs with broadcast links.
 - Heterogeneous links.** Because a switch isolates one link from another, the different links in the LAN can operate at different speeds and can run over different media.
 - For example, the uppermost switch in Figure 13 might have three 1 Gbps 1000BASE-T copper links, two 100 Mbps 100BASE-FX fiber links, and one 100BASE-T copper link. Thus, a switch is ideal for mixing legacy equipment with new equipment.
 - Management.** In addition to providing enhanced security (see sidebar on Focus on Security), a switch also eases network management. For example, if an adapter malfunctions and continually sends Ethernet frames (called a **jabbering adapter**), a switch can detect the problem and internally disconnect the malfunctioning adapter.
 - With this feature, the network administrator need not get out of bed and drive back to work in order to correct the problem.

Q.15 Explain the difference of Switches Versus Routers.

- Ans**
- Routers are store-and-forward packet switches that forward packets using network-layer addresses. Although a switch is also a store-and-forward packet switch, it is fundamentally different from a router in that it forwards packets using MAC

addresses.

- Whereas a router is a layer-3 packet switch, a switch is a layer-2 packet switch.
- Even though switches and routers are fundamentally different, network administrators must often choose between them when installing an interconnection device. For example, for the network in Figure 13, the network administrator could just as easily have used a router instead of a switch to connect the department LANs, servers, and internet gateway router.
- Indeed, a router would permit interdepartmental communication without creating collisions. Given that both switches and routers are candidates for interconnection devices, what are the pros and cons of the two approaches?
- First consider the pros and cons of switches. As mentioned above, switches are plug-and-play, a property that is cherished by all the overworked network administrators of the world. Switches can also have relatively high filtering and forwarding rates, switches have to process frames only up through layer 2, whereas routers have to process datagrams up through layer 3.
- On the other hand, to prevent the cycling of broadcast frames, the active topology of a switched network is restricted to a spanning tree. Also, a large switched network would require large ARP tables in the hosts and routers and would generate substantial ARP traffic and processing.
- Furthermore, switches are susceptible to broadcast storms—if one host goes haywire and transmits an endless stream of Ethernet broadcast frames, the switches will forward all of these frames, causing the entire network to collapse.
- Now consider the pros and cons of routers. Because network addressing is often hierarchical (and not flat, as is MAC addressing), packets do not normally cycle through routers even when the network has redundant paths. (However, packets can cycle when router tables are misconfigured, IP uses a special datagram header field to limit the cycling.) Thus, packets are not restricted to a spanning tree and can use the best path between source and destination.
- Because routers do not have the spanning tree restriction, they have allowed the Internet to be built with a rich topology that includes, for example, multiple active links between Europe and North America.
- Another feature of routers is that they provide firewall protection against layer-2 broadcast storms. Perhaps the most significant drawback of routers, though, is that they are not plug-and-play—they and the hosts that connect to them need their IP addresses to be configured. Also, routers often have a larger per-packet processing time than switches, because they have to process up through the layer-3 fields.

	<ul style="list-style-type: none"> Another feature of routers is that they provide firewall protection against layer-2 broadcast storms. Perhaps the most significant drawback of routers, though, is that they are not plug-and-play—they and the hosts that connect to them need their IP addresses to be configured. Also, routers often have a larger per-packet processing time than switches, because they have to process up through the layer-3 fields.
Q.16	Explain the working of Virtual Local Area Networks (VLANs).
Ans	<ul style="list-style-type: none"> Fortunately, each of these difficulties can be handled by a switch that supports virtual local area networks (VLANs). As the name suggests, a switch that supports VLANs allows multiple virtual local area networks to be defined over a single physical local area network infrastructure. Hosts within a VLAN communicate with each other as if they (and no other hosts) were connected to the switch. In a port-based VLAN, the switch's ports (interfaces) are divided into groups by the network manager. Each group constitutes a VLAN, with the ports in each VLAN forming a broadcast domain (i.e., broadcast traffic from one port can only reach other ports in the group). <div data-bbox="573 980 1159 1354" data-label="Diagram">  <p>The diagram shows a 4x4 grid representing a switch with 16 ports. Ports 2, 4, 8, 10, and 15 are highlighted in blue. Below the grid, there are two groups of computer icons. The first group, labeled 'Electrical Engineering (VLAN ports 2-8)', has three icons connected to ports 2, 4, and 8. The second group, labeled 'Computer Science (VLAN ports 9-15)', has three icons connected to ports 10 and 15. Ellipses indicate additional ports and hosts in each group.</p> </div> <p>Figure 15: A single switch with two configured VLANs</p> <ul style="list-style-type: none"> Figure 15 shows a single switch with 16 ports. Ports 2 to 8 belong to the EE VLAN, while ports 9 to 15 belong to the CS VLAN (ports 1 and 16 are unassigned). This VLAN solves all of the difficulties noted above—EE and CS VLAN frames are isolated from each other, the two switches in Figure 13 have been replaced by a single switch, and if the user at switch port 8 joins the CS Department, the network operator simply reconfigures the VLAN software so that port 8 is now associated with the CS VLAN. One can easily imagine how the VLAN switch is configured and operates—the network manager declares a port to belong to a given VLAN (with undeclared ports belonging to a default VLAN) using switch management software, a table of port-to-VLAN mappings is maintained within the switch; and switch hardware only delivers frames between ports belonging to the same VLAN. But by completely isolating the two VLANs, we have introduced a new difficulty! How

can traffic from the EE Department be sent to the CS Department? One way to handle this would be to connect a VLAN switch port (e.g., port 1 in Figure 15) to an external router and configure that port to belong both the EE and CS VLANs.

- In this case, even though the EE and CS departments share the same physical switch, the logical configuration would look as if the EE and CS departments had separate switches connected via a router.
- An IP datagram going from the EE to the CS department would first cross the EE VLAN to reach the router and then be forwarded by the router back over the CS VLAN to the CS host.
- Fortunately, switch vendors make such configurations easy for the network manager by building a single device that contains both a VLAN switch and a router, so a separate external router is not needed.
- Figure 16 shows a second 8-port switch, where the switch ports have been defined as belonging to the EE or the CS VLAN, as needed. But how should these two switches be interconnected?

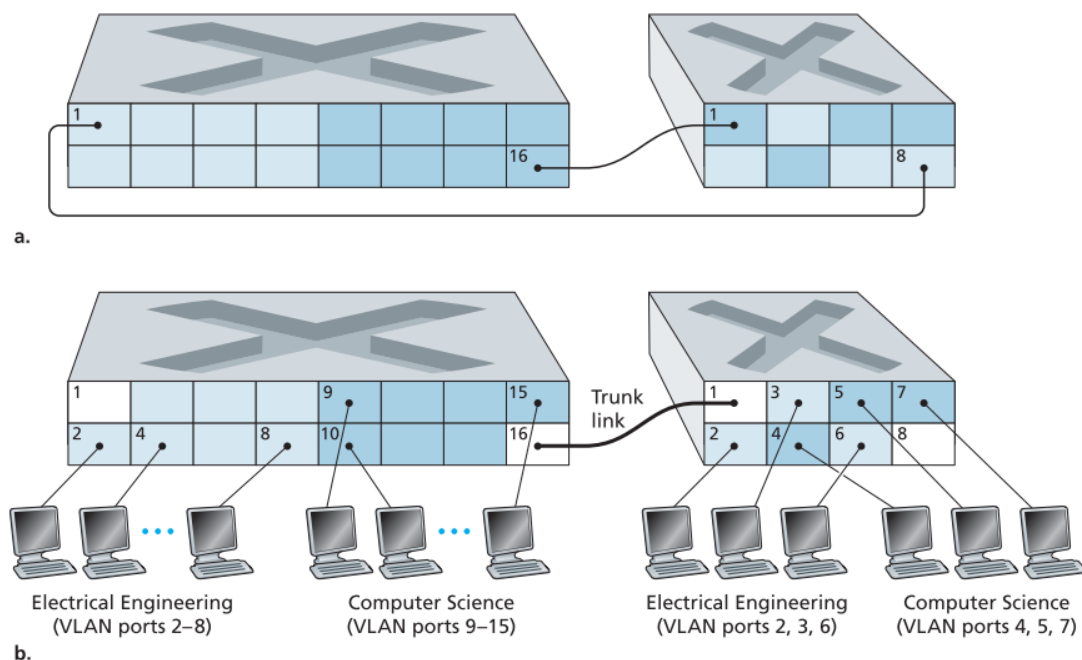


Figure 16: Connecting two VLAN switches with two VLANs: (a) twocables (b) trunked

- One easy solution would be to define a port belonging to the CS VLAN on each switch (similarly for the EE VLAN) and to connect these ports to each other, as shown in Figure 16(a).
- This solution doesn't scale, however, since N VLANs would require N ports on each switch simply to interconnect the two switches. A more scalable approach to interconnecting VLAN switches is known as **VLAN trunking**. In the **VLAN trunking** approach shown in Figure 16(b), a special port on each switch (port 16 on the left

switch and port 1 on the right switch) is configured as a trunk port to interconnect the two VLAN switches.

- The trunk port belongs to all VLANs, and frames sent to any VLAN are forwarded over the trunk link to the other switch. But this raises yet another question: How does a switch know that a frame arriving on a trunk port belongs to a particular VLAN? The IEEE has defined an extended Ethernet frame format, 802.1Q, for frames crossing a VLAN trunk.
- As shown in Figure 17, the 802.1Q frame consists of the standard Ethernet frame with a four-byte **VLAN tag** added into the header that carries the identity of the VLAN to which the frame belongs.
- The VLAN tag is added into a frame by the switch at the sending side of a VLAN trunk, parsed, and removed by the switch at the receiving side of the trunk. The VLAN tag itself consists of a 2-byte **Tag Protocol Identifier (TPID)** field (with a fixed hexadecimal value of 81-00), a 2-byte Tag Control Information field that contains a 12-bit VLAN identifier field, and a 3-bit priority field that is similar in intent to the IP datagram TOS field.

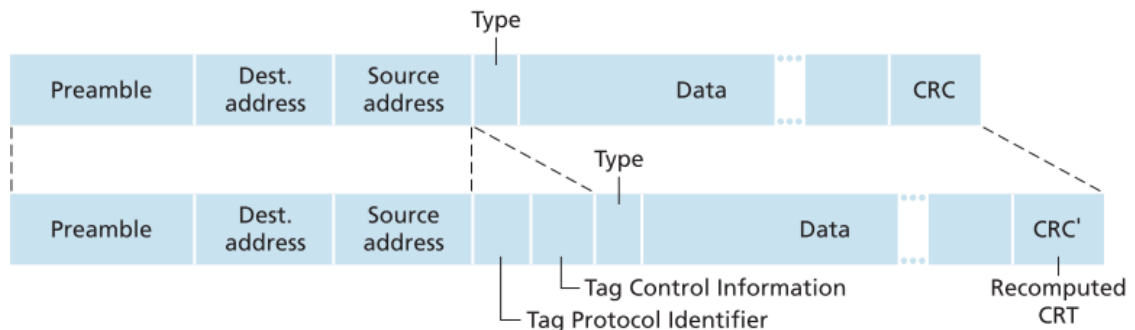


Figure 17: Original Ethernet frame (top), 802.1Q-tagged EthernetVLAN frame (below)

- In this discussion, we've only briefly touched on VLANs and have focused on port-based VLANs. We should also mention that VLANs can be defined in several other ways.
- In MAC-based VLANs, the network manager specifies the set of MAC addresses that belong to each VLAN; whenever a device attaches to a port, the port is connected into the appropriate VLAN based on the MAC address of the device. VLANs can also be defined based on network-layer protocols (e.g., IPv4, IPv6, or Appletalk) and other criteria.

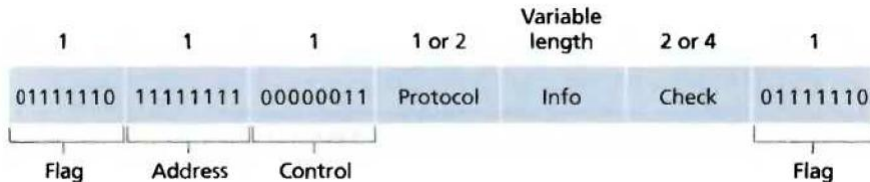
Q.18 What is PPP(Point-to-Point Protocol)? Also write the requirement satisfied by PPP?

Ans

- **PPP- the point-to-point protocol**, Because PPP is typically the protocol of choice for a dial-up link from a residential host, it is undoubtedly one of the most widely deployed link-layer protocols today.
- The other important link-layer protocol in use today is the high-level data link control (HDLC) protocol; see for a discussion of HDLC. Our discussion here of the simpler PPP

protocol will allow us to explore many of the most important features of a point-to-point link-layer protocol.

- As its name implies, the point-to-point protocol (PPP) is a link-layer protocol that operates over a point-to-point link—a link directly connecting two nodes, one on each end of the link.
- The point-to-point link over which PPP operates might be a serial dial-up telephone line (for example, a 56K modem connection), a **SONET/SDH** link, an **X.25** connection, or an ISDN circuit. As noted above, PPP is often the protocol of choice for connecting home users to their ISPs over a dial-up connection.
- **Packet framing.** The PPP protocol link-layer sender must be able to take a network-level packet and encapsulate it within the PPP link-layer frame such that the receiver will be able to identify the start and end of both the link-layer frame and the network-layer packet within the frame.
- **Transparency.** The PPP protocol must not place any constraints on data appearing on the network-layer packet (headers or data). Thus, for example, PPP cannot forbid the use of certain bit patterns in the network-layer packet. We'll return to this issue shortly in our discussion of byte stuffing.
- **Multiple network—layer protocols.** The PPP protocol must be able to support multiple network-layer protocols (for example, IP and DECnet) running over the same physical link at the same time. Just as the IP protocol is required to multiplex different transport-level protocols (for example, TCP and UDP) over a single end-to-end connection, so too must PPP be able to multiplex different network-layer protocols over a single point-to-point connection.
- This requirement means that at a minimum, PPP will likely require a protocol type field or some similar mechanism so the receiving-side PPP can demultiplex a received frame up to the appropriate network-layer protocol.
- **Multiple types of links.** In addition to being able to carry multiple higher-level protocols, PPP must also be able to operate over a wide variety of link types, including links that are either serial (transmitting a bit at a time in a given direction) or parallel (transmitting bits in parallel), synchronous (transmitting a clock signal along with the data bits) or asynchronous, low-speed or high-speed, electrical or optical.
- **Error detection.** A PPP receiver must be able to detect bit errors in the received frame.
- **Connection liveness.** PPP must be able to detect a failure at the link level (for example, the inability to transfer data from the sending side of the link to the receiving side of the link) and signal this error condition to the network layer.
- **Network-layer address negotiation.** PPP must provide a mechanism for the communicating network layers (for example, IP) to learn or configure each other's network—layer address.
- **Simplicity.** PPP was required to meet a number of additional requirements beyond those listed above. On top of all of these requirements, first and foremost is

	simplicity.
Q.19	Explain frame structure of PPP Data Framing.
Ans	<ul style="list-style-type: none"> Figure 19 shows a PPP data frame that uses HDLC-like framing. The ➤ PPP frame contains the following fields: ▪ Flag field. Every PPP frame begins and ends with a 1-byte flag field with a value of 01111110. ▪ Address field. The only possible value for this field is 11111111. ▪ Control field. The only possible value for this field is 00000011. Because both the address and control fields can take only a fixed value, you might wonder why the fields are defined in the first place. The PPP specification states that other values "may be defined at a later time," although none has been defined to date. Because these fields take fixed values, PPP allows the sender to simply not send the address and control bytes, thus saving 2 bytes of overhead in the PPP frame. <div style="text-align: center;">  <p>The diagram illustrates the structure of a PPP data frame. It consists of seven fields arranged horizontally. Above each field is its size in bytes: 1, 1, 1, 1 or 2, Variable length, 2 or 4, and 1. The fields are: Flag (01111110), Address (11111111), Control (00000011), Protocol, Info, Check, and Flag (01111110). Brackets below the first three fields label them as 'Flag', 'Address', and 'Control'. A bracket below the last field labels it as 'Flag'.</p> </div> <p style="text-align: center;">Figure 18: PPP data frame format</p> <ul style="list-style-type: none"> ▪ Protocol. The protocol field tells the PPP receiver the upper-layer protocol to which the received encapsulated data (that is, the contents of the PPP frame's information field) belongs. On receipt of a PPP frame, the PPP receiver will check the frame for correctness and then pass the encapsulated data on to the appropriate protocol. ▪ Of interest to us is the IP protocol (that is, the data encapsulated in the PPP frame is an IP datagram), which has a value of 21 hexadecimal; other network-layer protocols such as AppleTalk and DECnet. ▪ Information. This field contains the encapsulated packet (data) that is being sent by an upper-layer protocol (for example, IP) over the PPP link. The default maximum length of the information field is 1,500 bytes, although this can be changed when the link is first configured, as discussed below. ▪ Checksum. The checksum field is used to detect bit errors in a transmitted frame. It uses either a 2- or 4-byte HDLC-standard cyclic redundancy code.
Q.17	What is bit and byte stuffing? Explain with example.
Ans	<ul style="list-style-type: none"> While sending data over network, the data link layer divide into frames. Framing have several advantages than send raw very large data. It reduces the probability of error and reduces the amount of retransmission needed. There exist two general methods for framing: fixed size framing and variable size framing.

- Here the network system needs a mechanism to distinguish the end of a packet and beginning of another one. Two protocols are used for this purpose:

- ✓ **character oriented protocol and**
- ✓ **Bit oriented protocol.**

Byte stuffing:

- In character-oriented protocol, we add special characters to distinguish beginning and end of a frame. Usually flag has 8-bit length.
- The character-oriented protocols are popular only with text data. While using character-oriented protocol another problem arises.
- Let us consider a problem that arises when any protocol uses a specific bit pattern in a flag field to delineate the beginning or end of the frame. What happens if the flag pattern itself occurs elsewhere in the packet? For example, what happens if flag field value of 01111110 appears in the information field? Will the receiver incorrectly detect the end of PPP frame?
- One way to solve this problem would be for PPP to forbid the upper-layer protocol from sending data containing the flag bit pattern.
- The PPP requirement of transparency discussed above obviates this possibility. An alternative solution, and the one taken in PPP and many other protocols, is to use a technique known as byte stuffing (also known as character stuffing).
- PPP defines a special control escape byte, 01111110. If the flag sequence, 01111110, appears anywhere in the frame, except in the flag field, PPP precedes that instance of the flag pattern with the control escape byte.
- That is, it “stuffs” (adds) a control escape byte into the transmitted data stream, before the 01111110, to indicate that the following 01111110 is not a flag value but is, in fact, actual data.
- A receiver that sees a 01111110 preceded by a 01111110 will, of course, remove the stuffed control escape to reconstruct the original data.
- Similarly, if the control escape byte bit pattern itself appears as actual data, it too must be preceded by a stuffed control escape byte.
- Thus, when the receiver sees a single control escape byte by itself in the data stream, it knows that the byte was stuffed into the data stream.
- A pair of control escape bytes occurring back to back means that one instance of the control escape byte appears in the original data being sent.
- The following figure illustrates PPP stuffing.

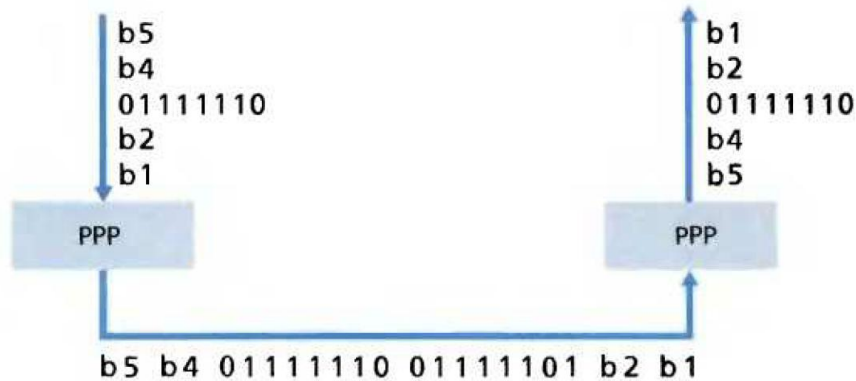


Figure 19: Byte stuffing

Bit stuffing:

- In a bit oriented protocol, the data to send is a series of bits. In order to distinguish frames, most protocols use a bit pattern of 8-bit length (01111110) as flag at the beginning and end of each frame.
- Here also cause the problem of appearance of flag in the data part to deal with this an extra bit added. This method is called bit stuffing.
- In bit stuffing, if a 0 and five successive 1 bits are encountered, an extra 0 is added. The receiver node removes the extra-added zero.
- Simply, **Bit stuffing** is a process of adding one extra 0 whenever five consecutive 1's follow a 0 in the data. **Byte stuffing** is the method of adding 1 extra byte if there is a flag or escape character in the part.