SAYAN CHATTERJEE
2K19/CO/348

Q1.

The transport layer links the network and user support layers by segmenting and rearranging the data using the TCP and UDP protocols.

Q2.

The network physical layer is the lowest network layer in the Open System Interconnections (OSI) model.

- The primary concern of this layer is transmission of individual bits from one node to another over a physical medium.
- Mainly the design issues here deal with electrical, mechanical, timing interfaces, and the physical transmission medium, which lies below the physical layer.
- Design issue has to do with making sure that when 1 bit send from one side, it is received 1 bit by the other side also not as a 0 bit.

Q3.

Data Link Layer

Data Link Layer is the second layer of OSI Layered Model. It is responsible for converting data streams to signals bit by bit and to send that over the underlying hardware.
The **functions** of the Data Link layer are:

1. **Framing:** Framing is a function of the data link layer. It provides a way for a sender to transmit a set of bits that are meaningful to the receiver. This can be accomplished by attaching special bit patterns to the beginning and end of the frame.
2. **Physical addressing:** After creating frames, the Data link layer adds physical addresses (MAC address) of the sender and/or receiver in the header of each frame.
3. **Error control:** Data link layer provides the mechanism of error control in which it detects and retransmits damaged or lost frames.
4. **Flow Control:** The data rate must be constant on both sides else the data may get corrupted thus, flow control coordinates the amount of data that can be sent before receiving acknowledgement.
5. **Access control:** When a single communication channel is shared by multiple devices, the MAC sub-layer of the data link layer helps to determine which device has control over the channel at a given time.

Network layer

Network layer is the third layer of the OSI Layered Model. It manages options pertaining to host and network addressing, managing sub-networks, and internetworking.
Network layer takes the responsibility for routing packets from source to destination within or outside a subnet.
The **functions** of the Network layer are :

1. **Routing:** The network layer protocols determine which route is suitable from source to destination. This function of the network layer is known as routing.
2. **Logical Addressing:** In order to identify each device on internetwork uniquely, the network layer defines an addressing scheme. The sender & receiver's IP addresses are placed in the header by the network layer. Such an address distinguishes each device uniquely and universally.

Q4.

In the Data Link layer, the stream of bits from the physical layer are divided into data frames. The data frames can be of fixed length or variable length. Each frame begins and ends with a special bit pattern, 01111110 (7E in hexadecimal notation) called a **flag** byte. In variable-length framing, the size of each frame to be transmitted may be different. So, a pattern of bits is used as a delimiter to mark the end of one frame and the beginning of the next frame. To separate one frame from the next, an 8-bit (or 1-byte) flag is added at the beginning and the end of a frame. But the problem with that is, any pattern used for the flag could also be part of the information. There are two ways to overcome this problem – **Bit stuffing** and **Byte stuffing**.

Bit stuffing: Also called Bit-oriented framing, in this, whenever five consecutive 1's are encountered in the data, a '0' bit is automatically stuffed into the outgoing bit stream. When the receiver sees five consecutive incoming 1 bits, followed by a 0 bit, it automatically destuffs (i.e., deletes) the 0 bit.

For example, input bit stream = 01111101011111. After bit-stuffing, output stream = 01111100010111110.

Q5.

Both Bit rate and Baud rate are terms associated with data communication. However, there are certain differences between them are as follows:

Bit rate	**Baud rate**
Refers to the number of bits transmitted per unit time.	Refers to the number of signals transmitted per unit time.

Determines the number of bits traveled per second.	Determines the number of times the state of a signal is changing while transmission.
Mainly focuses on the efficiency of a computer.	Concerns the transmission of data over a given channel.
Cannot be used to determine the bandwidth of a channel.	It can determine the bandwidth of the channel.
Measured in bits/sec or 'bps'.	Measured in pulses/sec.

Bit rate and Baud rate are related by the equation:

Baud rate = Bit rate*(Number of bits in 1 baud)

Q6.

Both **ARP (Address Resolution Protocol)** and **RARP (Reverse Address Resolution Protocol)** are protocols of the network layer. Whenever a host needs to send an IP datagram to another host, the sender requires both the logical address and physical address of the receiver. The dynamic mapping provides two protocols ARP and RARP.

Some key differences between the two are listed here –

ARP	RARP
Retrieves the receiver's physical address from its logical address.	Retrieves the receiver's logical address from its physical address.

Maps 32-bit IP address to 48-bit MAC address.	Maps 48-bit MAC address to 32-bit IP address.
MAC addresses are used for broadcasting.	IP addresses are used for broadcasting.
ARP table is maintained by local host.	RARP table is maintained by RARP server.
Used by hosts and routers for knowing the MAC address of other hosts and routers in the networks.	Used by small users having less facilities.

Q7.

On a network, multicast communication refers to communication between a single sender and numerous recipients. Multicast can be one-to-many or many-to-many distribution. Some **examples** are - Mobile workers may be updated from a home office, and online newsletters can be sent out on a regular basis. Multicast is one of the packet types in Internet Protocol Version 6 along with anycast and unicast.

Motivation for developing Multicast:

Multicast is used for sending data from a source to some specific receivers that are interested in the data. Unicast could not be employed for this task as the source would have to send duplicated copies of data to each receiver which requires a large bandwidth and more load on traffic. Similarly, the Broadcast method has the disadvantage of sending the data to all the hosts on a network, whether they require the data or not. This causes hosts to receive unnecessary data packets and also increases the load on traffic. Thus, Multicast was developed to tackle the problems in Unicast and broadcast communication.

Q8.

The Open Systems Interconnect (OSI) model is a conceptual framework for networking and telecommunications systems that divides them into seven layers. Each layer has its own set of ISO standards, which are published independently. It is referred to as OSI as it deals with connection open systems. That is the systems are open for communication with other systems.

7 Layers of OSI Model:

1. **Physical Layer:** The physical layer is in charge of the network nodes' actual cable or wireless connections. It specifies the connection, electrical cable, or wireless technology that connects the devices, and is in charge of transmitting raw data, which is just a series of 0s and 1s, as well as bit rate control.
2. **Data Link Layer:** A connection between two physically linked nodes on a network is established and terminated via the data link layer. It divides packets into frames and transmits them from one location to another. This layer is divided into two sections: LLC, which detects network protocols, does error checking, and synchronizes frames, and MAC, which connects devices and defines rights to transmit and receive data using MAC addresses.
3. **Network Layer:** The network layer has two purposes. One method is to split segments into network packets and then reassemble them on the receiving end. The alternative method is to route packets over a physical network by determining the optimum path. To route packets to a target node, the network layer utilises network addresses (usually Internet Protocol addresses).
4. **Transport Layer:** The transport layer takes data transferred in the session layer and breaks it into “segments” on the transmitting end. It's in charge of reassembling the segments on the receiving end and converting them back into data that the session layer can use. The transport layer handles flow control, which involves providing data at a pace that matches the receiving device's connection speed, as well as error control.
5. **Session Layer:** The session layer establishes communication channels between devices, known as sessions. It's in charge of starting sessions, making sure they stay open and functional while data is transmitted, and terminating them after the connection is through.
6. **Presentation Layer:** Data is prepared for the application layer by the presentation layer. It specifies how data should be encoded, encrypted, and compressed between two devices so that it is appropriately received on the other end.
7. **Application Layer:** The application layer is used by end-user software such as web browsers and email clients. It defines protocols that allow software to communicate and receive data and present it to users in a meaningful way. The Hypertext Transfer Protocol (HTTP), File Transfer Protocol (FTP), Post Office Protocol (POP), Simple Mail Transfer Protocol (SMTP), and Domain Name System are all examples of application layer protocols (DNS).

Q9.

When the receiver's information does not match the sender's information, this is known as an Error. Digital signals are subject to noise during transmission, which can cause mistakes in the binary bits as they travel from sender to receiver.

To avoid receiving corrupted data, we use error-detecting codes which are additional data added to a given digital message to help us detect if any error has occurred during transmission of the message.
Three main techniques used for Error-Detection are :-

1. Parity Check
2. Checksum
3. Cyclic Redundancy Cycle (CRC)

In Error Correction the receiver needs to guess the Original codeword that is sent. In this way, Error Correction is much more difficult than Error Detection.
There are two ways to handle the error correction:

1. Whenever an error is discovered, the receiver can have the sender in order to retransmit the entire data unit. This technique is known as the Backward Error correction technique. This technique is simple and inexpensive in the case of wired transmission like fiber optics; there is no expense in retransmitting the data. In the case of wireless transmission, retransmission costs too much thus forward error correction technique is used then.
2. The receiver can use an error-correcting code that automatically contains certain errors. This technique is known as the Forward Error Correction technique.

Some Popular Codes used for Handling Error correction are : - Hamming codes, Binary convolutional codes, Reed-Solomon codes, Low-Density Parity Check codes.

Q10.

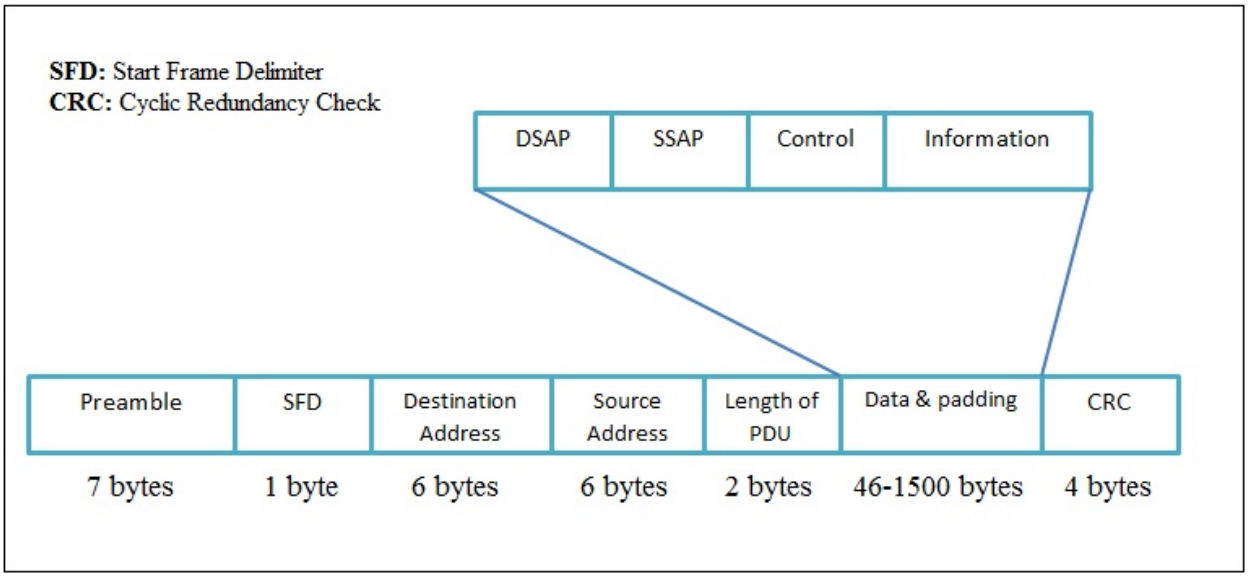
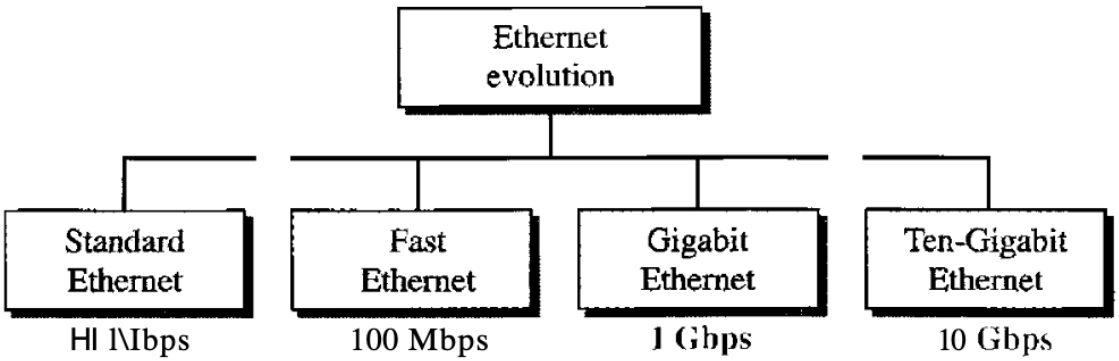


Fig: IEEE 802.3 MAC Frame Format

Preamble: It consumes 7 bytes of alternating 0's and 1's and alerts the receiving entity about the incoming frame. It also gives enough time to the receiver to synchronize its clock with that of the transmitter. It is added at the physical layer.
Start Frame Delimiter (SFD): It is 1 byte long and signals the receiver about the beginning of the frame. It also provides the last chance for synchronization. The last 2 bits of this field (10101011) is 11 to tell the receiver that the next bit is the start of the destination address.
Destination Address (DA): It is 6 bytes long and it contains the physical address of the destination station or the next station to receive the packet.
Source Address (SA): It is 6 bytes long and it contains the physical address of the source station or the sender of the packet.
Length/Type of PDU: It is 2 bytes long. It defines the length or type of PDU. If its value is less than in 1518 then it is considered as the length field and if its value is greater than 1536 then it is considered as the type field.
Data: This field can be split up into two parts Data (0-1500 bytes) and padding (0- 46 bytes). This field carries the data sent by the upper layer. This field may have a minimum length of 46 bytes and a maximum length of 1500 bytes.
Cyclic Redundancy Check (CRC): This field is 4 bytes long and it contains the error detection code for the frame.

Q11.
Ethernet is a family of wired computer networking technologies commonly used in local area networks (LAN), metropolitan area networks (MAN) and wide area networks (WAN). The original Ethernet was created in 1976 at Xerox's Palo Alto Research Center (PARC). It was commercially introduced in 1980 and first standardized in 1983 as IEEE 802.3. Since then, it has gone through four generations: Standard Ethernet (lot Mbps), Fast Ethernet (100 Mbps), Gigabit Ethernet (l Gbps), and Ten-Gigabit Ethernet (l0 Gbps), as shown in Figure.



Q12.

Network designers have developed two basic strategies for dealing with errors. Both add redundant information to the data that is sent.

One strategy is to include enough redundant information to enable the receiver to deduce what the transmitted data must have been. The other is to include only enough redundancy to allow the receiver to deduce that an error has occurred and have it request a retransmission. The former strategy uses **error-correcting codes** and the latter uses **error-detecting codes**.

Error Correction can be handled in two ways:

- **Backward error correction:** Once the error is discovered, the receiver requests the sender to retransmit the entire data unit.

- **Forward error correction:** In this case, the receiver uses the error-correcting code which automatically corrects the errors.

We will examine two different **error-correcting codes**:

1. Hamming codes.
2. Reed-Solomon codes.

1. Hamming Codes

Hamming code is a block code that can identify and repair single-bit faults while detecting up to two simultaneous bit errors.The source encodes the message using this coding method by introducing superfluous bits into the message. Extra bits are created and injected at certain locations in the message to facilitate error detection and repair. When the destination receives this message, it does recalculations in order to discover faults and determine which bit position is incorrect.

The procedure used by the sender to encode the message encompasses the following steps –

- **Step 1** – Calculation of the number of redundant bits.
- **Step 2** – Positioning the redundant bits.
- **Step 3** – Calculating the values of each redundant bit.

Once the redundant bits are embedded within the message, this is sent to the user.

Step 1 – Calculation of the number of redundant bits.

If the message contains *m* number of data bits, *r* number of redundant bits are added to it so that *mr* is able to indicate at least (*m* + *r* + 1) different states. Here, (*m* + *r*) indicates location of an error in each of (*m* + *r*) bit positions and one additional state indicates no error. Since, *rr* bits can indicate 2^r states, 2^r must be at least equal to (*m* + *r* + 1). Thus the following equation should hold $2^r \geq m+r+1$

Step 2 – Positioning the redundant bits.

The *r* redundant bits placed at bit positions of powers of 2, i.e. 1, 2, 4, 8, 16 etc. They are referred to in the rest of this text as *r1* (at position 1), *r2* (at position 2), *r3* (at position 4), *r4* (at position 8) and so on.

Step 3 – Calculating the values of each redundant bit.

The redundant bits are parity bits. A parity bit is an extra bit that makes the number of 1s either even or odd. The two types of parity are −

- **Even Parity** − Here the total number of bits in the message is made even.
- **Odd Parity** − Here the total number of bits in the message is made odd.

Each redundant bit, ri, is calculated as the parity, generally even parity, based upon its bit position. It covers all bit positions whose binary representation includes a 1 in the ith position except the position of ri. Thus −

- r1 is the parity bit for all data bits in positions whose binary representation includes a 1 in the least significant position excluding 1 (3, 5, 7, 9, 11 and so on)
- r2 is the parity bit for all data bits in positions whose binary representation includes a 1 in the position 2 from right except 2 (3, 6, 7, 10, 11 and so on)
- r3 is the parity bit for all data bits in positions whose binary representation includes a 1 in the position 3 from right except 4 (5-7, 12-15, 20-23 and so on)

Decoding a message in Hamming Code

Once the receiver gets an incoming message, it performs recalculations to detect errors and correct them. The steps for recalculation are −

- **Step 1** − Calculation of the number of redundant bits.
- **Step 2** − Positioning the redundant bits.
- **Step 3** − Parity checking.
- **Step 4** − Error detection and correction

Step 1 – Calculation of the number of redundant bits

Using the same formula as in encoding, the number of redundant bits are ascertained.

$2^r \geq m + r + 1$ where *m* is the number of data bits and *r* is the number of redundant bits.

Step 2 – Positioning the redundant bits

The *r* redundant bits placed at bit positions of powers of 2, i.e. 1, 2, 4, 8, 16 etc.

Step 3 – Parity checking

Parity bits are calculated based upon the data bits and the redundant bits using the same rule as during generation of c1,c2 ,c3 ,c4 etc. Thus

c1 = parity(1, 3, 5, 7, 9, 11 and so on)

c2 = parity(2, 3, 6, 7, 10, 11 and so on)

c3 = parity(4-7, 12-15, 20-23 and so on)

Step 4 – Error detection and correction

The decimal equivalent of the parity bits binary values is calculated. If it is 0, there is no error. Otherwise, the decimal value gives the bit position which has an error. For example, if c1c2c3c4 = 1001, it implies that the data bit at position 9, decimal equivalent of 1001, has error. The bit is flipped to get the correct message.

Example :- The first step is to identify the bit position of the data & all the bit positions which are powers of 2 are marked as parity bits (e.g. 1, 2, 4, 8, etc.). The following image will help in visualizing the received hamming code of 7 bits.

D7	D6	D5	P4	D3	P2	P1
1	0	1	1	0	1	1

First, we need to detect whether there are any errors in this received hamming code.
Step 1: For checking parity bit P1, use check one and skip one method, which means, starting from P1 and then skip P2, take D3 then skip P4 then take D5, and then skip D6 and take D7, this way we will have the following bits,

D7	D5	D3	P1
1	1	0	1

As we can observe the total number of bits are odd so we will write the value of parity bit as P1 = 1. This means error is there.

Step 2: Check for P2 but while checking for P2, we will use check two and skip two method, which will give us the following data bits. But remember since we are checking for P2, so we have to start our count from P2 (P1 should not be considered).

D7	D6	D3	P2
1	0	0	1

As we can observe that the number of 1's are even, then we will write the value of P2 = 0. This means there is no error.
Step 3: Check for P4 but while checking for P4, we will use check four and skip four method, which will give us the following data bits. But remember since we are checking for P4, so we have started our count from P4(P1 & P2 should not be considered).

D7	D6	D5	P4
1	0	1	1

As we can observe that the number of 1's are odd, then we will write the value of P4 = 1. This means the error is there.
Since we found that received code has an error, so now we must correct them. To correct the errors, use the following steps:
Now the error word E will be:

P4	P2	P1
1	0	1

Now we have to determine the decimal value of this error word 101 which is 5 (22 *1 + 21 * 0 + 20 *1 = 5).
We get E = 5, which states that the error is in the fifth data bit. To correct it, just invert the fifth data bit.
So the correct data will be:

D7	D6	D5	P4	D3	P2	P1
1	0	0	1	0	1	1

2. Reed Solomon Codes:

Parameters of Reed - Solomon Codes

- A Reed-Solomon code is specified as RS(n,k).
- Here, n is the block length which is recognizable by symbols, holding the relation, $n = 2^m - 1$.
- The message size is of k bits.
- So the parity check size is ($n - k$) bits
- The code can correct up to (t) errors in a codeword, where ($2t = n - k$).

Generator Polynomial of Reed Solomon Code

In coding systems with block codes, valid code words consist of polynomials that are divisible by another fixed polynomial of short length. This fixed polynomial is called generator polynomial.

In Reed Solomon code, generator polynomials with factors are constructed where each root is a consecutive element in the Galois field. The polynomial is of the form −

$g(x) = (x - \alpha)(x - \alpha^2)(x - \alpha^3) \ldots\ldots (x - \alpha^{2t})$where α is a primitive element.

Encoding using Reed Solomon Code

The method of encoding in Reed Solomon code has the following steps −

- The message is represented as a polynomial p(x), and then multiplied with the generator polynomial g(x).
- The message vector $[x_1, x_2, x_3 \ldots\ldots x_k]$ is mapped to a polynomial of degree less than *k* such that $p_x(\alpha_i) = x_i$ for all i = 1,...k
- The polynomial is evaluated using interpolation methods like Lagrange Interpolation.
- Using this polynomial, the other points $\alpha_{k+1} \ldots \alpha_n$, are evaluated.
- The encoded message is calculated as s(x) = p(x) * g(x). The sender sends this encoded message along with the generator polynomial g(x).

Decoding using Reed Solomon Code

At the receiving end, the following decoding procedure done −

- The receiver receives the message r(x) and divides it by the generator polynomial g(x).
- If r(x)/g(x)=0, then it implies no error.
- If r(x)/g(x)≠0, then the error polynomial is evaluated using the expression: r(x) = p(x) * g(x) + e(x)
- The error polynomial gives the error positions.

Example : - A popular Reed-Solomon code is RS(255,223) with 8-bit symbols. Each codeword contains 255 code word bytes, of which 223 bytes are data and 32 bytes are parity. For this code:

n = 255, k = 223, s = 8

2t = 32, t = 16

> The decoder can correct any 16 symbol errors in the code word: i.e. errors in up to 16 bytes anywhere in the codeword can be automatically corrected.

Given a symbol size s, the maximum codeword length (n) for a Reed-Solomon code is **$n = 2^s - 1$**

For example, the maximum length of a code with 8-bit symbols (s=8) is 255 bytes.

Reed-Solomon codes may be shortened by (conceptually) making a number of data symbols zero at the encoder, not transmitting them, and then re-inserting them at the decoder.

We will examine three different **error-detecting codes**. They are all linear, systematic block codes:

1. Parity.
2. Checksums.
3. Cyclic Redundancy Checks (CRCs).

1. **Parity -** In this method a single parity bit is appended to the data. The parity bit is chosen so that the number of 1 bits in the codeword is even (or odd). Doing this is equivalent to computing the (even) parity bit as the modulo 2 sum or XOR of the data bits.

   Example - When 1011010 is sent in even parity, a bit is added to the end to make it 10110100. With odd parity 1011010 becomes 10110101. A code with a single parity bit has a distance of 2, since any single-bit error produces a codeword with the wrong parity. This means that it can detect single-bit errors.

2. **Checksums -** The word "checksum" is often used to mean a group of check bits associated with a message, regardless of how they are calculated. A group of parity bits is one example of a checksum.

   Example - An example of a checksum is the 16-bit Internet checksum used on all Internet packets as part of the IP protocol. This checksum is a sum of the message bits divided into 16-bit words.

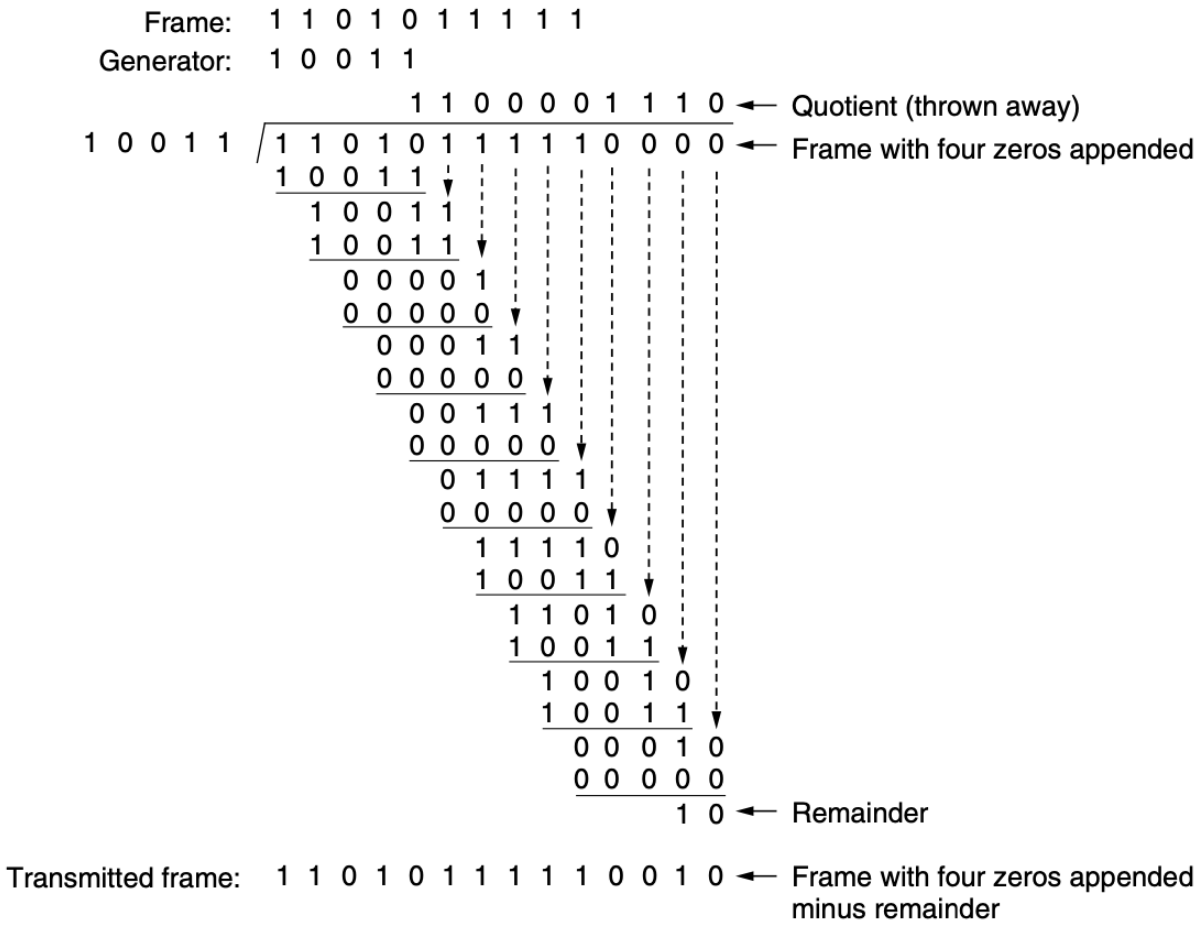
3. **Cyclic Redundancy Checks (CRCs) -** CRC (also known as a polynomial code) is based upon treating bit strings as representations of polynomials with coefficients of 0 and 1 only.

   The sender and receiver must agree upon a generator polynomial, G(x), in advance. Both the high- and low- order bits of the generator must be 1. To compute the CRC for some frame with m bits corresponding to the polynomial M(x), the frame must be longer than the generator polynomial. The idea is to append a CRC to the end of the frame in such a way that the polynomial represented by the checksummed frame is divisible by G(x). When the receiver gets the checksummed frame, it tries dividing it by G(x). If there is a remainder, there has been a transmission error.

   The algorithm for computing the CRC is as follows:

   1. Let r be the degree of G(x). Append r zero bits to the low-order end of the frame so it now contains m + r bits and corresponds to the polynomial xrM(x).
   2. Divide the bit string corresponding to G(x) into the bit string corresponding to $x^rM(x)$, using modulo 2 division.
   3. Subtract the remainder (which is always r or fewer bits) from the bit string corresponding to $x^rM(x)$ using modulo 2 subtraction. The result is the checksummed frame to be transmitted. Call its polynomial T(x).

   Example -



Q13.

Flow control is basically a technique that gives permission to two stations that are working and processing at different speeds to just communicate with one another. It is a technique that generally observes proper flow of data from sender to receiver. It is very essential because it is possible for the sender to transmit data or information at a very fast rate and hence the receiver can receive this information and process it. This can happen only if the receiver has a very high load of traffic as compared to the sender, or if the receiver has less processing power as compared to sender.

Flow Control is **classified into two categories**:

- **Feedback-based Flow Control:** In this control technique, sender simply transmits data or information or frame to receiver, then receiver transmits data back to sender and also allows sender to transmit more amount of data or tell sender about how receiver is

processing or doing. This simply means that the sender transmits data or frames after it has received acknowledgments from user.

- **Rate-based Flow Control:** In this control technique, usually when sender sends or transfer data at faster speed to receiver and receiver is not being able to receive data at the speed, then mechanism known as built-in mechanism in protocol will just limit or restricts overall rate at which data or information is being transferred or transmitted by sender without any feedback or acknowledgment from receiver.

Q14.

In the Data Link layer, the stream of bits from the physical layer are divided into data frames. There are four different methods of framing:

1. Byte count.
2. Flag bytes with byte stuffing.
3. Flag bits with bit stuffing.
4. Physical layer coding violations.

1. **Byte count -** This method uses a field in the header to specify the number of bytes in the frame. When the data link layer at the destination sees the byte count, it knows how many bytes follow and hence where the end of the frame is.
   Example -

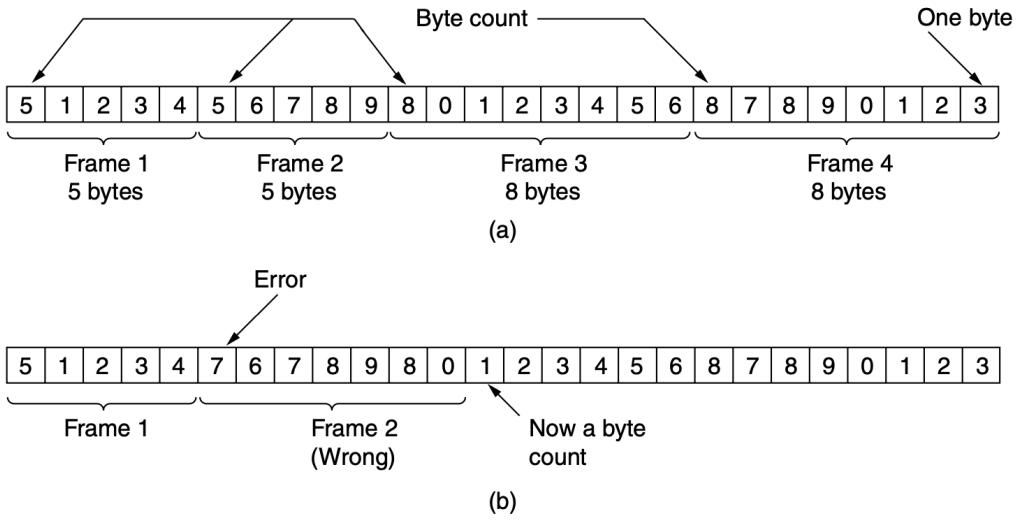


Figure 3-3. A byte stream. (a) Without errors. (b) With one error.

2. **Flag bytes with byte stuffing** - In byte stuffing (or character stuffing), a special byte is added to the data section of the frame when there is a character with the same pattern as the flag. The data section is stuffed with an extra byte. This byte is usually called the escape character (ESC), which has a predefined bit pattern. Whenever the receiver encounters the ESC character, it removes it from the data section and treats the next character as data, not a delimiting flag. If the escape character is part of the text, an extra one is added to show that the second one is part of the text.

3. **Flag bits with bit stuffing** - In this method each frame begins and ends with a special bit pattern, 01111110 or 0x7E in hexadecimal. This pattern is a flag byte. Whenever the sender's data link layer encounters five consecutive 1s in the data, it automatically stuffs a 0 bit into the outgoing bit stream. This bit stuffing is analogous to byte stuffing, in which an escape byte is stuffed into the outgoing character stream before a flag byte in the data. It also ensures a minimum density of transitions that help the physical layer maintain synchronization. When the receiver sees five consecutive incoming 1 bits, followed by a 0 bit, it automatically destuffs (i.e., deletes) the 0 bit.

4. **Physical layer coding violations -** This method of framing uses a shortcut from the physical layer. The encoding of bits as signals often includes redundancy to help the receiver. This redundancy means that some signals will not occur in regular data. We can use some reserved signals to indicate the start and end of frames. In effect, we are using "coding violations" to delimit frames. The beauty of this scheme is that, because they are reserved signals, it is easy to find the start and end of frames and there is no need to stuff the data.

Q15.

Network software is an extremely broad term for a range of software aimed at the design, operation, implementation and monitoring of modern networks. Various types of network software support the creation, calibration and operation of networks.

The **Software Defined Networking** framework has **three layers**:
Application Layer: SDN applications reside in the Application Layer. The applications convey their needs for resources and services to the control layer through APIs.
Control Layer: The Network Control Software, bundled into the Network Operating System, lies in this layer. It provides an abstract view of the underlying network

infrastructure. It receives the requirements of the SDN applications and relays them to the network components.

Infrastructure Layer: Also called the Data Plane Layer, this layer contains the actual network components. The network devices reside in this layer that shows their network capabilities through the Control to data-Plane Interface.

Functions of Network Software:

- When using network software, the size and breadth of a network are important factors to consider when making decisions.
- Other network software resources assist administrators and security workers in monitoring a network in order to protect it from a variety of threats, prevent data breaches, and limit unwanted access.
- Other tools help to make network operations work better.
- Network virtualization is a different type of network software. Various tools replace old legacy systems based on physical hardware setups with modern virtual networks.

In general, network software should be used in accordance with security requirements. This can include things like whether or not a network is connected to the Internet as well as specific usage goals and objectives.