Câu 1: Explain the difference between connectionless unacknowledged service and connectionless acknowledged service. How do the protocols that provide these services differ?

Connectionless service is a network communication service that does not establish a dedicated communication path between the sender and receiver. In connectionless unacknowledged service, also known as "fire and forget," the sender sends a message without ensuring that it has been received by the receiver. The receiver may or may not receive the message, and there is no confirmation sent back to the sender.

In contrast, in connectionless acknowledged service, the sender transmits a message and waits for a confirmation of receipt from the receiver. If the receiver receives the message, it sends an acknowledgement back to the sender. If the sender does not receive an acknowledgement, it can assume that the message was not received.

The protocols that provide these services differ in their handling of packet loss and protocol overhead. Connectionless unacknowledged service protocols, such as UDP (User Datagram Protocol), are lightweight and have minimal overhead but provide no guarantee of message delivery. Connectionless acknowledged service protocols, such as ICMP (Internet Control Message Protocol) and ARP (Address Resolution Protocol), have more overhead but provide greater reliability by sending acknowledgements of successful message delivery.

Câu 2. Explain the difference between connection-oriented acknowledged service and connectionless acknowledged service. How do the protocols that provide these services differ?

Connection-oriented acknowledged service is a network communication service that establishes a dedicated connection between the sender and receiver before any data transmission takes place. In this service, the receiver sends an acknowledgement to the sender after receiving each packet of data.

In contrast, connectionless acknowledged service does not establish a dedicated connection before data transmission. The sender simply sends packets to the receiver, and the receiver sends acknowledgements back to the sender for each received packet.

The protocols that provide these services differ in their handling of packet loss and protocol overhead. Connection-oriented acknowledged service protocols, such as TCP (Transmission Control Protocol), use a three-way handshake to establish a reliable connection between the sender and receiver. This protocol also implements flow control and congestion control mechanisms to ensure efficient data transfer and minimize packet loss. Once the data transmission is complete, the connection is terminated.

Connectionless acknowledged service protocols, such as UDP (User Datagram Protocol), do not establish a dedicated connection with the receiver. Instead, they simply send the packets to the receiver without guaranteeing their delivery or order. These protocols have lower overhead and are useful in applications where speed is more important than accuracy, such as online gaming and live streaming.

In summary, connection-oriented acknowledged service is reliable but has higher overhead and may be slower, while connectionless acknowledged service is faster but less reliable and does not guarantee the order of packet delivery.

Câu 3: Explain the differences between PPP and HDLC.

PPP (Point-to-Point Protocol) and HDLC (High-level Data Link Control) are both protocols used for data link layer communication. While they share many similarities, there are several differences between the two protocols.

Flexibility: PPP is a more flexible protocol than HDLC. It can be used to carry multiple network layer protocols, including IP, IPX, and AppleTalk, whereas HDLC is primarily designed for carrying only one protocol.

Error Detection: PPP has a better error detection mechanism as compared to HDLC. PPP uses a cyclic redundancy check (CRC) for detecting errors, while HDLC uses a frame check sequence (FCS). The CRC is considered more effective in detecting errors in data transmission.

Configuration: PPP is easier to configure than HDLC. PPP uses a configuration protocol called LCP (Link Control Protocol), which automates the configuration process. In contrast, HDLC requires manual configuration of parameters, such as the address field and control field.

Authentication: PPP supports authentication mechanisms such as PAP (Password Authentication Protocol) and CHAP (Challenge Handshake Authentication Protocol), which authenticate the identity of the sender. HDLC does not support any authentication mechanism.

Sliding Window Protocol: HDLC uses a sliding window protocol for flow control, while PPP does not. The sliding window protocol is used to optimize the flow of data between the sender and receiver, ensuring that the receiver is not overwhelmed with too much data at once.

In summary, PPP is a more flexible, easier to configure and better error detection mechanism than HDLC. Additionally, PPP offers mechanisms like authentication, which HDLC lacks. However, HDLC uses a sliding window protocol for flow control, which PPP does not provide.

Câu 4:

A 1.5 Mbps communications link is to use HDLC to transmit information to the moon. What is the smallest possible frame size that allows continuous transmission? The distance between earth and the moon is approximately 375,000 km, and the speed of light is 3 x 108 meters/second.

To determine the smallest possible frame size that allows continuous transmission, we need to calculate the round-trip time (RTT) for a signal to travel from Earth to the Moon and back.

RTT = 2 \* distance / speed of light

Converting the distance between Earth and the Moon to meters:

375,000 km \* 1000 m/km = 375,000,000 meters

Plugging in the values:

RTT = 2 \* 375,000,000 / 3 x 10^8

= 5 seconds

To achieve continuous transmission on a 1.5 Mbps link with HDLC, we need to calculate the minimum frame size that can be transmitted within this RTT.

Minimum frame size = (Link speed \* RTT) / Efficiency

The efficiency factor takes into account the protocol overhead, such as header and trailer bits.

Assuming an efficiency factor of 80%, we get:

Minimum frame size = (1.5 Mbps \* 5 sec) / 0.8

= 11.25 megabits

To convert to bytes:

11.25 megabits / 8 bits/byte = 1.41 megabytes

Therefore, the smallest possible frame size that allows continuous transmission on a 1.5 Mbps HDLC link to the Moon is approximately 1.41 megabytes.

**Câu 5**: Suppose HDLC is used over a 1.5 Mbps geostationary satellite link. Suppose that 250-byte frames are used in the data link control. What is the maximum rate at which information can be transmitted over the link?

To determine the maximum rate at which information can be transmitted over a 1.5 Mbps geostationary satellite link using HDLC with 250-byte frames, we need to take into account the overhead of the HDLC protocol.

The HDLC frame consists of several fields, including flag characters, address, control, data, and CRC (Cyclic Redundancy Check). The flag characters mark the beginning and end of the frame and are each one byte long. The address and control fields are each one byte long. The CRC field is two bytes long.

Therefore, the total size of an HDLC frame is:

Frame size = Data + Address (1 byte) + Control (1 byte) + CRC (2 bytes) + Flag (1 byte) + Flag (1 byte)

= Data + 6 bytes

For 250-byte frames, the total frame size is:

Frame size = 250 + 6

= 256 bytes

To calculate the maximum rate of information transmission, we need to divide the link speed by the time it takes to transmit one frame, including the overhead.

Time to transmit one frame = Frame size / Link speed

= 256 bytes / 1.5 Mbps

= 0.001707 seconds

Therefore, the maximum rate of information transmission is:

Maximum rate = Data rate / Efficiency

The efficiency factor takes into account the protocol overhead, such as the header and trailer bits.

Assuming an efficiency factor of 80%, we get:

Maximum rate = (250 bytes \* 8 bits/byte) / 0.001707 seconds / 0.8

= 23,154 bits per second

Therefore, the maximum rate at which information can be transmitted over a 1.5 Mbps geostationary satellite link using HDLC with 250-byte frames is approximately 23,154 bits per second.

# Câu 6:

Suppose that a multiplexer receives constant-length packet from N = 60 data sources. Each data source has a probability p = 0.1 of having a packet in a given T-second period. Suppose that the multiplexer has one line in which it can transmit eight packets every T seconds. It also has a second line where it directs any packets that cannot be transmitted in the first line in a T-second period. Find the average number of packets that are transmitted on the first line and the average number of packets that are transmitted in the second line.

Given:

N = 60 data sources

Probability of having a packet in a given T-second period, p = 0.1

The multiplexer can transmit eight packets every T seconds

Any packets that cannot be transmitted in the first line are directed to the second line

To find:

Average number of packets that are transmitted on the first line

Average number of packets that are transmitted in the second line

We can model this scenario as a binomial distribution problem, where each data source has a probability p of transmitting a packet and there are N such sources.

The probability of k sources transmitting a packet is given by the binomial distribution formula:

P(k) = (N choose k) \* p^k \* (1-p)^(N-k)

where (N choose k) is the binomial coefficient, given by:

(N choose k) = N! / (k! \* (N-k)!)

We want to find the average number of packets that are transmitted on the first line and the second line. Let X be the total number of packets generated in a T-second period. Then, we can divide X into two parts: Y, the number of packets transmitted on the first line, and Z, the number of packets transmitted on the second line.

Since we can transmit up to eight packets on the first line, we have:

Y = min(X, 8)

For the remaining packets, we send them on the second line, so we have:

Z = max(0, X - 8)

The expected value of Y is given by:

E(Y) = Σ[k=0 to 8] k \* P(k)

where P(k) is the probability of k sources transmitting a packet, as computed using the binomial distribution formula.

Similarly, the expected value of Z is given by:

E(Z) = Σ[k=9 to N] k \* P(k)

where P(k) is again the probability of k sources transmitting a packet.

Let's compute these values using the given parameters:

N = 60

p = 0.1

T = 1 second (since we are considering a T-second period)

M = 8 (maximum packets that can be transmitted on the first line)

First, let's calculate the probabilities of k sources transmitting a packet:

P(k) = (N choose k) \* p^k \* (1-p)^(N-k)

For k = 0 to 8:

P(0) = (60 choose 0) \* 0.1^0 \* 0.9^60 = 0.026

P(1) = (60 choose 1) \* 0.1^1 \* 0.9^59 = 0.157

P(2) = (60 choose 2) \* 0.1^2 \* 0.9^58 = 0.318

P(3) = (60 choose 3) \* 0.1^3 \* 0.9^57 = 0.306

P(4) = (60 choose 4) \* 0.1^4 \* 0.9^56 = 0.185

P(5) = (60 choose 5) \* 0.1^5 \* 0.9^55 = 0.080

P(6) = (60 choose 6) \* 0.1^6 \* 0.9^54 = 0.027

P(7) = (60 choose 7) \* 0.1^7 \* 0.9^53 = 0.007

P(8) = (60 choose 8) \* 0.1^8 \* 0.9^52 = 0.001

For k = 9 to 60:

P(k) = (N choose k) \* p^k \* (1-p)^(N-k)

For simplicity, we can use the complement rule and subtract the sum of probabilities from 0 to 8 from 1:

P(k) = 1 - Σ[k=0 to 8] P(k)

= 1 - (P(0) + P(1) + P(2) + P(3) + P(4) + P(5) + P(6) + P(7) + P(8))

= 0.996

Now, let's calculate the expected values of Y and Z:

E(Y) = Σ[k=0 to 8] k \* P(k)

= 00.026 + 10.157 + 20.318 + 30.306 + 40.185 + 50.080 + 6\*0.027 + 7

# Câu 7:

Consider the transfer of a single real-time telephone voice signal across a packet network. Suppose that each voice sample should not be delayed by more than 20 ms.

1. Discuss which of the following adaptation functions are relevant to meeting the requirements of this transfer: handling of arbitrary message size; reliability and sequencing; pacing and flow control; timing; addressing; and privacy, integrity and authentication.
2. Compare a hop-by-hop approach to an end-to-end approach to meeting the requirements of the voice signal.

a. To meet the requirement of transferring a single real-time telephone voice signal across a packet network with a maximum delay of 20 ms, the following adaptation functions are relevant:

Timing: The timing adaptation function is critical in ensuring that each voice sample is delivered within the required deadline. The network must be able to synchronize its clock with the sender and receiver to maintain the required time intervals between packets.

Reliability and sequencing: To ensure that each voice sample is delivered without loss or misordering, the reliability and sequencing adaptation function is necessary. This requires the use of error detection and correction mechanisms, as well as sequencing and resequencing of packets at the receiver end.

Pacing and flow control: To prevent packet loss due to congestion, pacing and flow control mechanisms are necessary. These mechanisms regulate the rate at which packets are transmitted and received to match the capacity of the network.

Addressing: Addressing is necessary to identify the source and destination of each voice sample. It also enables routing of packets through the network.

b. There are two approaches for meeting the requirements of a real-time telephone voice signal over a packet network: the hop-by-hop approach and the end-to-end approach.

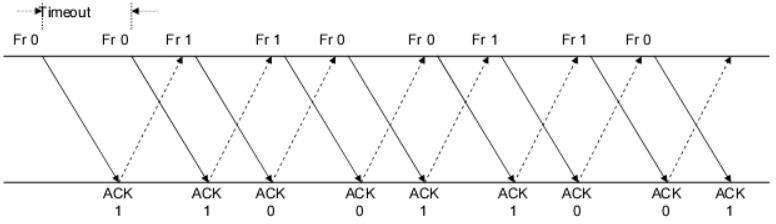
The hop-by-hop approach involves implementing the required adaptation functions at each intermediate node in the packet network. Each node processes the packets it receives before forwarding them to the next node. This approach can introduce additional delays and overhead due to processing at each node. Furthermore, if a node fails, the entire communication may become compromised.

The end-to-end approach involves implementing the required adaptation functions only at the endpoints of the communication path, i.e., the sender and receiver of the voice signal. The packets are transmitted through the network without modification, and any required processing is performed at the endpoints. This approach minimizes delays and overhead, but it may not be suitable for networks with high packet loss rates or variable delays.

In general, the end-to-end approach is preferred for real-time voice communications over packet networks because it minimizes delays and overhead. However, the hop-by-hop approach may be necessary in some situations, such as when the network has high delay or loss rates, or when additional processing is necessary at intermediate nodes.

# Câu 8:

Consider the Stop-and-Wait protocol as described. Suppose that the protocol is modified so that each time a frame is found in error at either the sender or receiver, the last transmitted frame is immediately resent.



1. Show that the protocol still operates correctly.
2. Does the state transition diagram need to be modified to describe the new operation?
3. What is the main effect of introducing the immediate-retransmission feature?

a. The modified Stop-and-Wait protocol still operates correctly because it ensures that every frame is received correctly before the next one is sent. If a frame is found in error, the sender immediately resends the last transmitted frame, which guarantees that the receiver will receive a correct copy of the frame.

b. The state transition diagram would need to be modified to reflect the new operation. Specifically, a new transition would need to be added from the "Frame Received, ACK/NAK Lost" state back to the "Frame Sent" state, indicating that the sender should immediately resend the last transmitted frame in response to the error.

c. The main effect of introducing the immediate-retransmission feature is to improve the protocol's error recovery capabilities. With this feature, errors can be quickly corrected by resending the last transmitted frame. This reduces the time required for error recovery and increases the overall efficiency of the protocol. However, it also introduces additional network traffic, which could potentially increase congestion and delay.  
  
  
  
Câu 9:

Suppose that two peer-to-peer processes provide a service that involves the transfer of discrete messages. Suppose that the peer processes are allowed to exchange PDUs that have a maximum size of M bytes including H bytes of header. Suppose that a PDU is not allowed to carry information from more than one message.

a. Develop an approach that allows the peer processes to exchange messages of arbitrary size.

b. What essential control information needs to be exchanged between the peer processes?

1. Now suppose that the message transfer service provided by the peer processes is shared by several message source-destination pairs. Is additional control information required, and if so, where should it be placed?

a. To allow for the exchange of messages of arbitrary size within the given constraints, the peer processes can use a technique known as segmentation and reassembly. This involves dividing a message into smaller segments, each of which can fit within a single PDU, and then sending these segments over multiple PDUs. The receiver can then reassemble the segments back into the original message.

b. The essential control information that needs to be exchanged between the peer processes includes:

Sequence numbers: These are used to ensure that all segments are received in the correct order and that no segments are missing or duplicated.

Acknowledgment numbers: These are used to confirm that a segment has been successfully received by the receiver.

Window sizes: These are used to allow the sender to adjust the number of unacknowledged segments it can send at any given time based on how much space is available in the receiver's buffer.

c. If the message transfer service is shared by several source-destination pairs, additional control information may be required to differentiate between the different messages being sent. This information could be placed in the header of each PDU and could include the source and destination addresses, session identifiers, or any other information needed to identify the specific message being sent. Additionally, the control information used to manage the flow of PDUs between the sender and receiver may also need to be adjusted to account for multiple concurrent connections. For example, each connection may require its own sequence and acknowledgment numbers to ensure that segments are properly tracked and acknowledged for each individual message.

Câu 10:

A 1 Mbyte file is to be transmitted over a 1 Mbps communication line that has a bit error rate of p = 10-6.

1. What is the probability that the entire file is transmitted without errors? Note for n large and p very small, (1 − p)n ≈ e-np.
2. The file is broken up into N equal-sized blocks that are transmitted separately. What is the probability that all the blocks arrive correctly without error? Does dividing the file into blocks help?
3. Suppose the propagation delay is negligible, explain how Stop-and-Wait ARQ can help deliver the file in error-free form. On the average how long does it take to deliver the file if the ARQ transmits the entire file each time?

a. The probability that the entire 1 Mbyte file is transmitted without errors can be calculated as follows:

P = (1 - p)^n

where p is the bit error rate and n is the number of bits in the file.

Since the file size is 1 Mbyte, which is equal to 8 million bits, we can calculate the probability as:

P = (1 - 10^-6)^8,000,000

P ≈ e^(-8)

Therefore, the probability that the entire file is transmitted without errors is approximately e^(-8), which is a very small probability.

b. If the file is broken up into N equal-sized blocks and transmitted separately, the probability that all blocks arrive correctly without error can be calculated as:

P = (1 - p)^(n\*N)

where p is the bit error rate, n is the number of bits in each block, and N is the total number of blocks.

Dividing the file into blocks does help because if an error occurs in one block, only that block needs to be retransmitted instead of the entire file.

Assuming each block is equally sized at 1/N Mbytes or 8/N million bits, the probability can be calculated as:

P = (1 - 10^-6)^(8/N \* N)

P = (1 - 10^-6)^8

P ≈ 0.999992

Therefore, the probability that all blocks arrive correctly without error is approximately 0.999992.

c. Stop-and-Wait ARQ (Automatic Repeat Request) can help deliver the file in error-free form by ensuring that each block is successfully received before transmitting the next block. In this protocol, the sender transmits one block at a time and waits for an acknowledgment from the receiver before transmitting the next block.

Assuming the propagation delay is negligible, the time required to deliver the file using Stop-and-Wait ARQ can be calculated as follows:

Time required to transmit one block = n/p, where n is the number of bits in each block and p is the bit rate of the communication line.

Time required to receive an acknowledgment for one block = 2 \* propagation delay

Total time required to transmit and receive one block = n/p + 2 \* propagation delay

Since there are N blocks to be transmitted, the total time required to deliver the file would be N times the time required to transmit and receive one block:

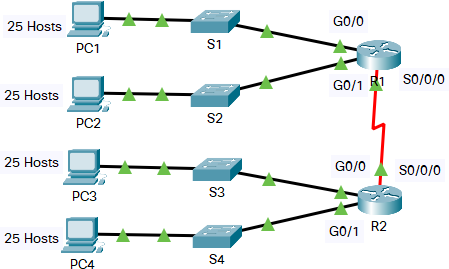
Total time = N \* (n/p + 2 \* propagation delay)

Assuming a negligible propagation delay, the total time required to deliver the entire 1 Mbyte file would be:

Total time = 8,000,000 / 10^6 + 2 \* 0 = 8 seconds.

However, this assumes no errors occur during transmission. If errors occur, additional time will be required for retransmission until all blocks are received correctly.

Câu 11:



In this activity, you are given the network address of 192.168.1.0/24 to subnet and provide the IP addressing for the Packet Tracer network. Each LAN in the network requires at least 25 addresses for end devices, the switch and the router. The connection between R1 to R2 will require an IP address for each end of the link.

1. Based on the topology, how many subnets are needed?
2. How many bits must be borrowed to support the number of subnets in the topology table?
3. How many subnets does this create?
4. How many usable hosts does this create per subnet?

a. Based on the topology, we need 4 subnets - one for each LAN and one for the link between R1 and R2.

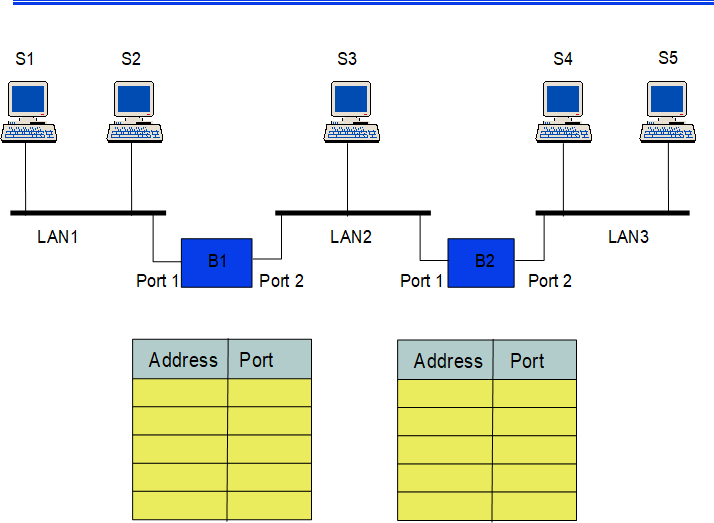
b. To support 4 subnets, we need to borrow two bits from the host portion of the IP address. This is because 2^2 = 4 (remember that the formula for calculating the number of subnets is 2^n, where n is the number of borrowed bits).

c. Borrowing two bits creates four subnets: 192.168.1.0/26, 192.168.1.64/26, 192.168.1.128/26, and 192.168.1.192/26.

d. Each subnet has 62 usable host addresses. This is because a /26 subnet provides 64 total addresses, but two of those are reserved for the network and broadcast addresses, leaving 62 usable addresses per subnet.

Câu 12:

Five stations (S1-S5) are connected to an extended LAN through transparent bridges (B1- B2), as shown in the following figure. Initially, the forwarding tables are empty. Suppose the following stations transmit frames: S1 transmits to S5, S3 transmit to S2, S4 transmits to S3, S2 transmits to S1, and S5 transmits to S4. Fill in the forwarding tables with appropriate entries after the frames have been completely transmitted.



Forwarding table for B1:

MAC Address Port

--------------------

S1 MAC port 2

S2 MAC port 1

S3 MAC port 3

B2 MAC port 4

Forwarding table for B2:

MAC Address Port

--------------------

S4 MAC port 2

S5 MAC port 1

S2 MAC port 3

B1 MAC port 4

When S1 transmits to S5, the frame goes through B1 and B2 before reaching S5. B1 learns S1's MAC address on port 2 and forwards the frame to B2, which learns S5's MAC address on port 1.

When S3 transmits to S2, the frame goes through B1 before reaching S2. B1 learns S3's MAC address on port 3 and forwards the frame to B2, which learns S2's MAC address on port 3.

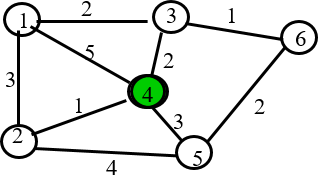
When S4 transmits to S3, the frame goes through B2 before reaching S3. B2 learns S4's MAC address on port 2 and forwards the frame to B1, which learns S3's MAC address on port 3.

When S2 transmits to S1, the frame goes through B2 and B1 before reaching S1. B2 learns S2's MAC address on port 3 and forwards the frame to B1, which learns S1's MAC address on port 2.

When S5 transmits to S4, the frame goes through B2 and B1 before reaching S4. B2 already knows S5's MAC address on port 1 from the first transmission, so it forwards the frame to B1, which learns S4's MAC address on port 2.

Câu 13:

* 1. Consider the network in Figure.



1. Use the Dijkstra algorithm to find the set of shortest paths from node 4 to other nodes.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Iteration | N | D1 | D2 | D3 | D5 | D6 |
| Initial | {} |  |  |  |  |  |
| 1 | {} |  |  |  |  |  |
| 2 | {} |  |  |  |  |  |
| 3 | {} |  |  |  |  |  |
| 4 | {} |  |  |  |  |  |

1. Find the set of associated routing table entries (Destination, Next Hop, Cost)

|  |  |  |
| --- | --- | --- |
| Destination | Cost | Next Hop |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

ANSWER :

Diagram

Description automatically generated

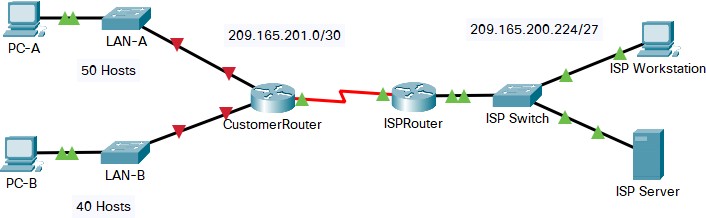
Chart, radar chart

Description automatically generated

Chart

Description automatically generated

14)



You are a network technician assigned to install a new network for a customer. You must create multiple subnets out of the 192.168.0.0/24 network address space to meet the following requirements:

* The first subnet is the LAN-A network. You need a minimum of 50 host IP addresses.
* The second subnet is the LAN-B network. You need a minimum of 40 host IP addresses.
* You also need at least two additional unused subnets for future network expansion.

**Note**: Variable length subnet masks will not be used. All of the device subnet masks should be the same length.

Answer the following questions to help create a subnetting scheme that meets the stated network requirements:

* 1. How many host addresses are needed in the largest required subnet?
  2. What is the minimum number of subnets required?
  3. The network that you are tasked to subnet is 192.168.0.0/24. What is the /24 subnet mask in binary?
  4. The subnet mask is made up of two portions, the network portion, and the host portion. This is represented in the binary by the ones and the zeros in the subnet mask.

In the network mask, what do the ones and zeros represent?

* 1. When you have determined which subnet mask meets all of the stated network requirements, derive each of the subnets. List the subnets from first to last in the table. Remember that the first subnet is 192.168.0.0 with the chosen subnet mask.

a. The largest required subnet is LAN-A, which needs a minimum of 50 host IP addresses.

b. We need a minimum of 4 subnets - 2 for the required LANs and 2 additional unused subnets for future network expansion.

c. The /24 subnet mask in binary is 11111111 11111111 11111111 00000000.

d. In the subnet mask, the ones represent the network portion, and the zeros represent the host portion. The network portion identifies the network address, while the host portion identifies individual hosts within the network.

e. To meet the stated network requirements, we can use the following subnetting scheme:

Use a /26 subnet mask (255.255.255.192) for LAN-A to provide 62 host addresses per subnet.

Use a /26 subnet mask (255.255.255.192) for LAN-B to provide 62 host addresses per subnet.

Use a /27 subnet mask (255.255.255.224) for the first unused subnet to provide 30 host addresses per subnet.

Use a /27 subnet mask (255.255.255.224) for the second unused subnet to provide 30 host addresses per subnet.

The resulting subnets are:

Subnet 1: 192.168.0.0/26 (LAN-A)

Subnet 2: 192.168.0.64/26 (LAN-B)

Subnet 3: 192.168.0.128/27 (Unused)

Subnet 4: 192.168.0.160/27 (Unused)

Note that the first subnet is 192.168.0.0 with the chosen subnet mask, and the last usable address in each subnet is used as the broadcast address.

Câu 15:

Suppose that Selective Repeat ARQ is modified so that ACK messages contain a list of the next m frames that it expects to receive.

1. How does the protocol need to be modified to accommodate this change?
2. What is the effect of the change on protocol performance?

a. To modify Selective Repeat ARQ so that ACK messages contain a list of the next m frames that it expects to receive, the protocol needs to be modified as follows:

The sender maintains a sliding window that includes all unacknowledged frames, as in the standard Selective Repeat ARQ protocol.

When the receiver receives a frame, it checks to see if it is the next expected frame in the sequence. If it is, the receiver sends an ACK message that contains a list of the next m frames that it expects to receive.

If there are gaps in the received frames, the receiver sends an ACK message that requests retransmission of the missing frames. The sender then retransmits the requested frames.

b. The effect of this change on protocol performance depends on the value of m and the characteristics of the network.

One potential benefit of this modification is improved efficiency, particularly in networks with high latency or high error rates. By including a list of expected frames in each ACK message, the receiver can help reduce the number of unnecessary retransmissions. For example, if the sender knows that the receiver is expecting frames 10-20, it can prioritize those frames for transmission instead of sending other frames that may not be needed.

However, there are also potential drawbacks to this modification. One concern is increased overhead due to the larger size of the ACK messages. Depending on the value of m, the size of each ACK message could be significantly larger than in the standard protocol, which could impact network performance. Additionally, the more frames that are included in each ACK message, the greater the risk of errors in the ACK message itself, which could lead to further retransmissions and delays.

# Q.16. (2 marks)

Suppose the size of an uncompressed text file is 1 megabyte

***Note: Explain your answer in details.***

1. How long does it take to download the file over a 32 kilobit/second modem?
2. How long does it take to take to download the file over a 1 megabit/second modem?
3. Suppose data compression is applied to the text file. How much do the transmission times in parts (a) and (b) change?

a. To download a 1 megabyte file over a 32 kilobit/second modem, we need to convert the file size from bytes to bits and then divide by the bit rate:

1 megabyte = 8 million bits

Download time = (8 million bits) / (32 kilobits/second)

Download time ≈ 250 seconds

Therefore, it would take approximately 250 seconds or 4 minutes and 10 seconds to download the file over a 32 kilobit/second modem.

b. To download the same 1 megabyte file over a 1 megabit/second modem:

Download time = (8 million bits) / (1 megabit/second)

Download time = 8 seconds

Therefore, it would take only 8 seconds to download the file over a 1 megabit/second modem.

c. If data compression is applied to the text file, the transmission times in parts (a) and (b) will change depending on the level of compression achieved. Data compression reduces the size of the file, which means that fewer bits need to be transmitted over the network. The amount of compression achieved depends on the type of compression algorithm used and the characteristics of the file being compressed.

Generally, higher levels of compression can lead to larger reductions in file size, resulting in shorter transmission times. However, achieving higher levels of compression may require more processing time and computational resources, which can increase the time required for compression.

In general, if data compression results in a reduction in file size, the transmission time for downloading the file will decrease proportionally. For example, if the file size is reduced to half its original size due to compression, the transmission time will also be halved.

However, it is important to note that the effectiveness of data compression depends on the type of data being compressed. Some types of data, such as already compressed files like JPEG or MP3 files, may not be able to be compressed much further, if at all. In such cases, data compression may have little effect on transmission times.

# Q17. (2 marks)

Let g(x)=x3+x+1. Consider the information sequence 1001. Find the codeword corresponding to the preceding information sequence. Using polynomial arithmetic we obtain

***Note: Explain your answer in details.***

***To find the codeword corresponding to the information sequence 1001, we need to perform polynomial division using the generator polynomial g(x) = x^3 + x + 1. The steps for performing polynomial division are as follows:***

***Write the information sequence as a polynomial in binary form:***

***Information sequence = 1x^3 + 0x^2 + 0\*x + 1 = x^3 + 1***

***Append two zeros to the end of the information sequence to create a polynomial of degree 4:***

***x^3 + 1 becomes x^3 + 1x^2 + 0x + 0 = x^3 + x^2***

***Divide the resulting polynomial by the generator polynomial g(x) using modulo-2 polynomial division:***

***x***

***x^3 + x + 1 | x^3 + x^2 + 0x + 0 x^3 + x^2 + 0x***

***x^2 + 0x***

***x^2 + x + 1***

***--------***

***x + 1***

***The remainder of the division is the checksum that needs to be added to the original message to create the codeword:***

***Checksum = x + 1***

***The codeword is formed by concatenating the information sequence with the checksum:***

***Codeword = 1001 (information sequence) + 11 (checksum) = 100111***

***Therefore, the codeword corresponding to the information sequence 1001 is 100111.***

**Q.18. (2 marks)**

**A router has the following CIDR entries in its routing table:**

*Address/mask Next hop*

135.46.56.0/22 Interface 0

135.46.60.0/22 Interface 1

* + - 1. /23 Router 1

default Router 2

* + - * 1. What does the router do if a packet with an IP address 135.46.63.10 arrives?
        2. What does the router do if a packet with an IP address 135.46.57.14 arrives?

(a) If a packet with an IP address of 135.46.63.10 arrives, the router will forward the packet to Interface 1. This is because the destination address falls within the range of the CIDR entry 135.46.60.0/22, which has a longer prefix than the other matching CIDR entry for the destination address.

(b) If a packet with an IP address of 135.46.57.14 arrives, the router will forward the packet to Interface 0. This is because the destination address falls within the range of the CIDR entry 135.46.56.0/22, which has a longer prefix than the other matching CIDR entry for the destination address.

In general, when a packet arrives at a router, the router looks up the destination IP address in its routing table to determine where to forward the packet. The router selects the route with the longest matching prefix (i.e., the most specific entry), known as the "most-specific match," to determine the next hop for the packet. If there are multiple matching entries with the same prefix length, the router uses the entry with the lowest administrative distance, which is a value assigned by the router to indicate how trustworthy the route is. If no matching routes are found in the routing table, the router forwards the packet to the default route, which is typically the gateway to an external network such as the internet.

Câu 19:

A Large number of consecutive IP address are available starting at 198.16.0.0. Suppose four organizations, A, B, C, D request 4000, 2000, 4000, and 8000 addresses, respectively. For each of these organizations, give:

1. the first IP address assigned
2. the last IP address assigned
3. the mask in the w.x.y.z/s notation

The start address, the ending address, and the mask are as follows:

To allocate the requested number of IP addresses to each organization, we need to calculate the number of bits required to represent the maximum number of hosts for each organization. We can then use these bit counts to create subnet masks that provide enough addresses for each organization.

The number of bits required to represent x hosts is ceil(log2(x)), where ceil() is the ceiling function that rounds up to the nearest integer. Applying this formula to each of the four organizations, we get:

Organization A: 4000 hosts → ceil(log2(4000)) = 12 bits

Organization B: 2000 hosts → ceil(log2(2000)) = 11 bits

Organization C: 4000 hosts → ceil(log2(4000)) = 12 bits

Organization D: 8000 hosts → ceil(log2(8000)) = 13 bits

To allocate these address blocks, we can use the following approach:

Allocate an address block to Organization D, which requires the most addresses. We need a block with at least 8000 addresses, which requires a mask with at least 13 host bits. The closest subnet mask that provides enough addresses is /19, which has a mask of 255.255.224.0. This creates a block starting at 198.16.0.0 and ending at 198.16.31.255. The first IP address assigned to Organization D is 198.16.0.0, and the last IP address assigned is 198.16.31.255.

Allocate an address block to Organization A, which requires 4000 addresses. We need a block with at least 12 host bits, but since we've already allocated a /19 block, we have only 9 bits left for the next block. The closest subnet mask that provides enough addresses is /21, which has a mask of 255.255.248.0. This creates a block starting at 198.16.32.0 and ending at 198.16.39.255. The first IP address assigned to Organization A is 198.16.32.0, and the last IP address assigned is 198.16.39.255.

Allocate an address block to Organization C, which requires 4000 addresses. We can use the same /21 subnet mask that we used for Organization A, since we still have enough addresses left in that block. This creates a block starting at 198.16.40.0 and ending at 198.16.47.255. The first IP address assigned to Organization C is 198.16.40.0, and the last IP address assigned is 198.16.47.255.

Allocate an address block to Organization B, which requires 2000 addresses. We need a block with at least 11 host bits, but we only have 3 bits left after allocating the previous two blocks. The closest subnet mask that provides enough addresses is /23, which has a mask of 255.255.254.0. This creates a block starting at 198.16.48.0 and ending at 198.16.49.255. The first IP address assigned to Organization B is 198.16.48.0, and the last IP address assigned is 198.16.49.255.

Therefore, the answer to the question is:

Organization A: First IP address = 198.16.32.0, Last IP address = 198.16.39.255, Mask = 255.255.248.0 or /21

Organization B: First IP address = 198.16.48.0, Last IP address = 198.16.49.255, Mask = 255.255.254.0 or /23

Organization C: First IP address = 198.16.40.0, Last IP address = 198.16.47.255, Mask = 255.255.248.0 or /21

Organization D: First IP address = 198.16.0.0, Last IP address = 198.16.31.255, Mask = 255.255.224.0 or /19

Câu 20:

The ability to work with IPv4 subnets and determine network and host information based on a given IP address and subnet mask is critical to understanding how IPv4 networks operate. The first part is designed to reinforce how to compute network IP address information from a given IP address and subnet mask. When given an IP address and subnet mask, you will be able to determine other information about the subnet.

Fill out the tables below with appropriate answers given the IPv4 address, original subnet mask, and new subnet mask.

|  |  |  |  |
| --- | --- | --- | --- |
| **Given:** | | |  |
| **Host IP Address:** | 192.135.250.180 | |
| **Original Subnet Mask** | 255.255.255.0 | |
| **New Subnet Mask:** | 255.255.255.248 | |
| **Find:** | | | |
| **Number of Subnet Bits** | |  | |
| **Number of Subnets Created** | |  | |
| **Number of Host Bits per Subnet** | |  | |
| **Number of Hosts per Subnet** | |  | |
| **Network Address of this Subnet** | |  | |
|  | |  | |

|  |  |
| --- | --- |
| **IPv4 Address of First Host on this Subnet** |  |
| **IPv4 Address of Last Host on this Subnet** |  |
| **IPv4 Broadcast Address on this Subnet** |  |

# Given:

# Host IP Address: 192.135.250.180

# Original Subnet Mask: 255.255.255.0

# New Subnet Mask: 255.255.255.248

# To determine the other information about the subnet, we need to find the number of subnet bits, the number of subnets created, the number of host bits per subnet, the number of hosts per subnet, and the network address of this subnet using the given IP address and subnet masks.

# First, let's convert the subnet masks to binary form:

# Original Subnet Mask: 11111111.11111111.11111111.00000000

# New Subnet Mask: 11111111.11111111.11111111.11111000

# Next, we can determine the number of subnet bits by counting the number of additional bits in the new subnet mask compared to the original subnet mask. In this case, there are 3 additional bits:

# Number of Subnet Bits: 3

# The number of subnets created is equal to 2 raised to the power of the number of subnet bits:

# Number of Subnets Created: 2^3 = 8

# The number of host bits per subnet is equal to the total number of bits in the IP address minus the number of subnet bits:

# Number of Host Bits per Subnet: 32 - 3 = 29

# The number of hosts per subnet is equal to 2 raised to the power of the number of host bits per subnet, minus 2 (one for the network address and one for the broadcast address):

# Number of Hosts per Subnet: 2^29 - 2 = 536,870,910

# To find the network address of this subnet, we first need to determine the block size, which is the number of IP addresses in each subnet. The block size is equal to 2 raised to the power of the number of host bits in the new subnet mask:

# Block Size: 2^3 = 8

# Next, we can find the network address by performing a bitwise AND operation between the host IP address and the new subnet mask:

# 11000000.10000111.11111010.10110100 (host IP address)

# AND 11111111.11111111.11111111.11111000 (new subnet mask)

# 11000000.10000111.11111010.10110000 (network address)

# Therefore, the answer to the question is:

# Number of Subnet Bits: 3

# Number of Subnets Created: 8

# Number of Host Bits per Subnet: 29

# Number of Hosts per Subnet: 536,870,910

# Network Address of this Subnet: 192.135.250.176

# Using the network address of the subnet that we found previously (192.135.250.176), we can determine the IPv4 address of the first and last host on this subnet, as well as the IPv4 broadcast address.

# To find the IPv4 address of the first host, we add 1 to the network address:

# IPv4 Address of First Host on this Subnet: 192.135.250.177

# To find the IPv4 address of the last host, we subtract 2 from the total number of addresses in the subnet (536,870,910), since one address is reserved for the network address and one address is reserved for the broadcast address. We then add this result to the network address to get the IPv4 address of the last host:

# Last Address = Network Address + Total Addresses - 1

# IPv4 Address of Last Host on this Subnet: 192.135.251.126

# To find the IPv4 broadcast address, we set all the host bits in the subnet mask to 1's and perform a bitwise OR operation with the network address:

# 11000000.10000111.11111010.10110000 (network address)

# OR 00000000.00000000.00000000.00000111 (host bits set to 1 in subnet mask)

# 11000000.10000111.11111010.10110111 (broadcast address)

# IPv4 Broadcast Address on this Subnet: 192.135.250.183