) 1.1. True. Wireless links, such as cellular and Wi-Fi networks, are multiple access links because multiple devices can access and use the network simultaneously.

1.2. True. Routing determines how to forward messages or data packets toward the destination node based on its address. It involves making decisions on the best path to take to reach the desired destination.

1.3. True. Peer-to-peer (P2P) computing or networking is a distributed application architecture that partitions tasks or workloads between peers. It enables direct communication and collaboration between individual peers without relying on a central server.

1.4. False. The presentation layer in the networking protocol stack is responsible for the formatting and conversion of data between different entities, but it does not concern the router format exchanged between peers. The router format is handled by the network layer.

1.5. False. The application layer interacts with higher-level protocols such as HTTP, SMTP, FTP, etc., and is not directly involved in calculating IP addresses. The network layer is responsible for IP address calculations in the layered Internet architecture.

1.6. True. In the Internet architecture, the transport layer divides the data stream into small units called packets. These packets are sent across the network individually and reassembled at the receiving end.

1.7. True. A network link is a physical medium, such as cables or wireless channels, that carries signals in the form of electromagnetic waves. It allows transmission and communication between devices on a network.

1.8. False. Manchester encoding is a coding scheme that helps in synchronization and data transmission over a medium, and it does double the rate of signal transitions. It ensures that each bit is represented by a transition from high to low or low to high during each bit period.

1.9. False. The extra bits transmitted with the message for error detection and correction are called error-correcting codes or error-detecting codes. These codes are used to detect errors during data transmission and enable error correction if possible.

1.10. True. The goal of designing error detection algorithms is to maximize the probability of detecting errors using redundant bits. By including additional bits, the likelihood of detecting errors increases as they provide more information for error detection and correction.

1.11. True. The Internet is indeed a globally connected network system that uses the TCP/IP protocol suite for transmitting data. TCP/IP is the standard communication protocol used on the Internet and allows for the interconnection of various devices and media types.

1.12. False. Internet technology does not reveal the details of network hardware. Instead, it provides an abstraction layer that allows computers and devices to communicate independently of their physical network connections. Internet technology focuses on the logical and virtual aspects of network communication.

1.13. True. Routers make forwarding decisions based on the destination network, not the specific destination computer. They use routing tables to determine the next hop for a packet based on the destination network address.

1.14. False. Protocols provide a standardized set of rules and formats for network communication, abstracting away the details of specific network hardware. They allow different systems and devices to communicate regardless of their hardware differences.

1.15. False. An IP address identifies a device on a network, but it does not uniquely identify the source and destination of data transmitted with the Internet Protocol. The combination of the IP address and the port number identifies the source and destination of the data.

1.16. True. In subnetting, the subnet mask is used to distinguish the network and host IP addresses. It covers both the Internet (network) and the physical network portions of the address to determine how the IP address is divided into network and host components.

1.17. False. The IP address 196.168.1.0 is not a class B network address. It is a class C network address as per the classful addressing scheme. In classful addressing, the first octet of a class B address ranges from 128 to 191.

21:04 Lut Lat Aung (Alex) 1.18. True. In a classless address scheme, subnet addressing allows for flexibility in using the network prefix of the address. This means that the network prefix length can be varying and does not have to follow the fixed lengths defined in the classful addressing scheme.

1.19. True. In subnetting, the subnet mask is used to distinguish the network and host IP addresses. It covers both the Internet (network) and the physical network portions of the address, allowing for the division of the IP address into network and host components.

1.20. True. TCP/IP uses the term "host" to refer to an end system or device that connects to the internet. A host can be a computer, server, router, or any other device that participates in the TCP/IP network and communicates with other hosts.

Q2

The three main stakeholders required to develop a computer network are:

1. **Network Engineers/IT Professionals:** These are the individuals responsible for the design, implementation, and maintenance of the computer network. Their functions include:
   * **Planning and Design:** Network engineers analyze the requirements of the organization and design a network topology that suits their needs. This involves deciding on the type of network (e.g., LAN, WAN), selecting the appropriate hardware and software components, and establishing security protocols.
   * **Installation and Configuration:** Once the design is finalized, network engineers set up the physical infrastructure, install networking devices such as routers, switches, and firewalls, and configure them to work harmoniously within the network.
   * **Troubleshooting and Maintenance:** Network engineers continuously monitor the network's performance, identify and resolve issues like connectivity problems, bottlenecks, or security breaches. Regular maintenance and updates are also part of their responsibilities to ensure the network remains reliable and secure.
2. **Network Administrators:** Network administrators handle the day-to-day operations and management of the computer network. Their functions include:
   * **User Management:** They create and manage user accounts, providing appropriate permissions and access levels to network resources based on the users' roles and needs.
   * **Resource Allocation:** Network administrators oversee the distribution of network resources such as bandwidth, IP addresses, and file storage to ensure fair and efficient usage.
   * **Monitoring and Optimization:** They monitor network performance, track traffic patterns, and identify areas for improvement. This includes optimizing the network to ensure it operates at its best efficiency.
   * **Security Management:** Network administrators implement security measures like firewalls, intrusion detection systems, and access controls to protect the network from unauthorized access and cyber threats.
3. **End Users:** End users are the individuals who utilize the computer network to perform their daily tasks and access network resources. Their functions include:
   * **Resource Consumption:** End users use the network to access shared files, applications, databases, and other resources required for their work.
   * **Collaboration:** Through the network, end users can communicate and collaborate with colleagues, sharing information and working together on projects.
   * **Compliance:** End users need to follow network policies and security protocols to ensure data integrity, confidentiality, and the overall smooth functioning of the network.
   * **Feedback and Reporting:** End users play a role in reporting any network issues they encounter, assisting network administrators and engineers in resolving problems and improving the network's functionality.

Q3

To calculate the throughput of the channel, we can use the formula:

Throughput (in bits per second) = Bandwidth (in bits per second) \* Efficiency

Where Efficiency is a value between 0 and 1, representing the efficiency of the channel in transmitting data without any loss or overhead.

Given:

* Bandwidth = 10 Mbps (millions of bits per second)
* File size = 20 MB (megabytes)

First, we need to convert the file size from megabytes to bits, as the bandwidth is given in bits per second.

1 MB = 8 megabits (since 1 byte = 8 bits) 20 MB = 20 \* 8 megabits = 160 megabits

Now, we can calculate the throughput:

Throughput = 10 Mbps \* Efficiency

To find Efficiency, we can use the formula:

Efficiency = (File size in bits) / (Bandwidth in bits per second)

Efficiency = 160 megabits / 10 Mbps

Efficiency = 160 / 10 = 16

Now, the throughput is:

Throughput = 10 Mbps \* 16 = 160 Mbps

So, the throughput of the channel is 160 Mbps (megabits per second).

Q4

Non-Return to Zero (NRZ) encoding is a binary data transmission technique in which the presence of a signal during a specific time interval represents one bit, while the absence of a signal represents the other bit. While NRZ encoding is straightforward and easy to implement, it has some issues that can limit its practicality, especially in long-distance communication and high-speed data transmission scenarios. Here are the key issues with NRZ encoding:

1. **DC Component and Baseline Wander:** NRZ encoding does not have a guaranteed zero mean, which means that there is no balance between the number of 0s and 1s in the signal. As a result, the signal can have a significant DC (direct current) component. This leads to baseline wander, where the average signal level may drift over time, causing difficulties in accurate bit detection, especially in long sequences of 0s or 1s.
2. **Synchronization:** NRZ encoding relies on consistent timing for bit detection. If there are long sequences of 0s or 1s, it becomes challenging for the receiver to maintain synchronization with the transmitter, potentially leading to misinterpretation of data bits.
3. **Limited Error Detection:** NRZ encoding does not provide any inherent error-detection capability. In the absence of additional error-checking mechanisms, such as parity or checksum, it becomes difficult to detect and correct errors that might occur during transmission.
4. **Lack of Signal Transitions:** In NRZ encoding, consecutive identical bits result in continuous signal levels without any transitions. This lack of transitions can create problems for clock recovery and data regeneration at the receiver end, leading to potential bit slipping or loss of synchronization.
5. **Spectral Efficiency:** NRZ encoding utilizes a wide frequency range since signal transitions occur frequently. This can limit the spectral efficiency of the channel and may result in higher susceptibility to noise and interference.

Due to these issues, NRZ encoding is not commonly used in long-distance and high-speed data transmission applications. Other encoding schemes, such as Manchester encoding, Differential Manchester encoding, or other advanced modulation techniques, are often employed to overcome these limitations and provide better performance and reliability in various communication scenarios.

Q5

In Non-Return to Zero Inverted (NRZI) encoding, the signal level changes when a 1 bit is encountered, and it remains the same when a 0 bit is encountered. The initial signal level depends on the first bit of the data.

Let's apply NRZI encoding to the 16-bit data: 1010110011011101

1. Start with an initial signal level (usually high for NRZI): Assume we start with a high signal level (1) as our initial state.
2. Encode each bit according to the rules:
   * When encountering a 1 bit, invert the signal level (change from 1 to 0 or from 0 to 1).
   * When encountering a 0 bit, keep the current signal level.

Applying NRZI encoding to the given data:

Data: 1 0 1 0 1 1 0 0 1 1 0 1 1 1 0 1 NRZI: ↑ ↓ ↑ ↓ ↑ ↑ ↓ ↓ ↑ ↑ ↓ ↑ ↑ ↑ ↓ ↑

The NRZI encoded result of the 16-bit data "1010110011011101" is: ↑↓↑↓↑↑↓↓↑↑↓↑↑↑↓↑

Please note that the initial signal level (↑) depends on the first bit of the data. If the first bit were 0, the initial signal level would be ↓.

Q6

The 4B/5B encoding is a technique used to convert 4-bit data into 5-bit symbols to ensure a balanced number of 0s and 1s for better clock recovery and error detection in data transmission. Each 4-bit input is mapped to a unique 5-bit output using a predefined lookup table.

Here's the 4B/5B encoding for the 16-bit data "0110011101001010":

Data: 0 1 1 0 0 1 1 1 0 1 0 0 1 0 1 0 4B/5B: 1110 0001 0101 1001 1100 1100 0010 1010

The 4B/5B encoded result for the 16-bit data "0110011101001010" is "11100001010110011100110000101010".

Note: The 4B to 5B conversion table used to perform the encoding may vary depending on the specific encoding standard being used (e.g., 4B/5B, 8B/10B). The table used in this example is for illustrative purposes only and may not match any specific standard.

Q7

To find the checksum of the given 8-bit data, we'll perform a simple binary addition and take the one's complement of the sum. The checksum is used for error detection and is calculated by adding all the data bytes together.

Given data: 10101010 and 11010110

Step 1: Add the two 8-bit data together (discarding any carry):

markdown

10101010

+ 11010110

-----------

100000000

Step 2: Take the one's complement of the sum:

lua

100000000

-----------

011111111

The checksum of the given data "10101010" and "11010110" is "01111111".

Q8

To calculate the Cyclic Redundancy Check (CRC) polynomial, we perform polynomial division, also known as long division, with the message polynomial M(x) as the dividend and the channel divisor polynomial C(x) as the divisor. The remainder of this division operation will be the CRC polynomial.

Given: Message polynomial M(x) = x^7 + x^5 + x^4 + x^3 Channel divisor polynomial C(x) = x^3 + x^1 + 1

Let's perform polynomial division to find the CRC polynomial:

scss

x^4

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

x^3 + x^1 + 1 | x^7 + x^5 + x^4 + x^3

-(x^7 + x^5 + x^3)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

x^5 + x^4

-(x^5 + x^3 + x^1)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

x^4 + x^3 + x^1

-(x^4 + x^3 + x^1)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

0

The remainder is 0, which means there is no remainder. So, the CRC polynomial is 0.

The binary representation of the CRC polynomial is "00000000" since there are no non-zero terms in the polynomial representation.

Q9

Briefly describe how a network switch decides which output link to place each packet.

A network switch is a networking device that operates at the data link layer (Layer 2) of the OSI model and is responsible for forwarding data packets between multiple devices within a local area network (LAN). When a packet arrives at a switch, it needs to determine the appropriate output link through which the packet should be forwarded to reach its destination.

The process by which a network switch decides which output link to place each packet is called "switching." There are primarily two types of switching techniques employed by switches:

1. **Store-and-Forward Switching:** In store-and-forward switching, the switch first receives the entire data packet before it starts making any decisions about forwarding. The switch checks the packet's destination MAC address (Media Access Control) in the packet header and compares it to its internal MAC address table, which associates MAC addresses with the corresponding switch ports.

The steps involved in store-and-forward switching are as follows:

* + The switch reads the destination MAC address from the packet header.
  + It looks up the MAC address table to find the corresponding output port associated with that MAC address.
  + The packet is then forwarded to the identified output port.

Store-and-forward switching ensures error checking as it verifies the integrity of the received packet before forwarding, but it introduces a slight delay due to the entire packet being received and checked.

1. **Cut-Through Switching:** In cut-through switching, the switch starts forwarding the packet as soon as it reads the destination MAC address in the packet header, without waiting for the entire packet to arrive. This means that the switch only reads enough of the packet to determine the destination address before forwarding it.

The steps involved in cut-through switching are as follows:

* + The switch reads the destination MAC address from the packet header.
  + It looks up the MAC address table to find the corresponding output port associated with that MAC address.
  + The packet is immediately forwarded to the identified output port, without waiting for the entire packet to be received.

Cut-through switching reduces the latency introduced by store-and-forward switching, as packets are forwarded as soon as the destination address is known. However, it may lead to forwarding corrupted or erroneous packets since the switch does not verify the entire packet's integrity.

Modern switches typically use a combination of both store-and-forward and cut-through switching techniques, adapting their behavior based on factors such as packet size, network conditions, and quality of service requirements.

Q10

Consider a network with the address 211.165.13.0. Subnetting the network with the starting 4-bit value of the last octet of its address. Show all of its valid subnet addresses (in dotted decimal form). What is its subnet mask?

ChatGPT

To subnet the network with the starting 4-bit value of the last octet of its address (211.165.13.0), we will borrow 4 bits from the last octet to create subnets. This means we will have 2^4 = 16 subnets.

The subnet mask will be the binary representation of the borrowed bits filled with 1s and followed by 0s. In this case, the subnet mask will be "11111111.11111111.11111111.11110000" or in dotted decimal form "/28".

Let's calculate the valid subnet addresses:

1. Starting address: 211.165.13.0

The binary representation of the last octet "0000" means the last 4 bits are all zeros, representing subnet 0.

1. Increment the last 4 bits for each subsequent subnet:

Subnet 1: 211.165.13.16 Subnet 2: 211.165.13.32 Subnet 3: 211.165.13.48 ... Subnet 14: 211.165.13.224 Subnet 15: 211.165.13.240

1. Last subnet with all 1s in the last 4 bits: Subnet 16: 211.165.13.255

So, the valid subnet addresses are: 211.165.13.0 (Subnet 0) 211.165.13.16 (Subnet 1) 211.165.13.32 (Subnet 2) ... 211.165.13.240 (Subnet 15) 211.165.13.255 (Subnet 16)

The subnet mask is "255.255.255.240" or "/28".

Q11

Show all the /25 subnets in the 132.42.6.0/23 address block.

ChatGPT

To find all the /25 subnets within the 132.42.6.0/23 address block, we need to determine the valid subnet ranges by dividing the address block into smaller subnets with a subnet mask of /25.

A /25 subnet has 25 bits for the network portion and 302 - 25 = 7 bits for the host portion. Each subnet will have 2^7 = 128 IP addresses, with the first and last addresses reserved for the   
network address and broadcast address, respectively.

The address block 132.42.6.0/23 covers IP addresses from 132.42.6.0 to 132.42.7.255. To subnet it into /25 subnets, we will divide it as follows:

Subnet 1: 132.42.6.0/25

* Subnet Range: 132.42.6.0 to 132.42.6.127
* Usable IP addresses: 132.42.6.1 to 132.42.6.126 (excluding network and broadcast addresses)

Subnet 2: 132.42.6.128/25

* Subnet Range: 132.42.6.128 to 132.42.6.255
* Usable IP addresses: 132.42.6.129 to 132.42.6.254 (excluding network and broadcast addresses)

Subnet 3: 132.42.7.0/25

* Subnet Range: 132.42.7.0 to 132.42.7.127
* Usable IP addresses: 132.42.7.1 to 132.42.7.126 (excluding network and broadcast addresses)

Subnet 4: 132.42.7.128/25

* Subnet Range: 132.42.7.128 to 132.42.7.255
* Usable IP addresses: 132.42.7.129 to 132.42.7.254 (excluding network and broadcast addresses)

Please note that the subnet ranges and usable IP addresses listed above are exclusive of the network and broadcast addresses. The network address is the first address in each subnet range, and the broadcast address is the last address in each subnet range. These are reserved addresses and cannot be assigned to individual devices.

Q12

Consider a datagram network illustrated in the Figure below. In which the hosts have addresses A, B, C, and so on. Assume that Host A wants to send a packet to Host G. Construct the forwarding (routing) tables for the switches that support the packet transmission from A to G (assume that during a routing process, when a data packet turns up, the forwarding table will have the right information to forward/switch the packet).

ChatGPT

I apologize, but as an AI text-based model, I don't have access to external figures or images. Without the specific details of the network topology and the forwarding logic used by the switches in the network, I can't construct the forwarding (routing) tables.

However, I can provide you with a general understanding of how forwarding tables are constructed in such a scenario:

1. **Switches:** Each switch maintains a forwarding table that maps destination addresses (host addresses) to the corresponding output port. The forwarding table is used to determine where to forward incoming packets.
2. **Initial Configuration:** Initially, the forwarding tables in the switches are empty or have default entries to forward packets to all ports except the incoming port.
3. **Learning Phase:** The switches use a learning process to populate their forwarding tables as they receive packets from hosts. When a switch receives a packet, it learns the source MAC address and associates it with the incoming port in its forwarding table.
4. **Flooding:** When a switch receives a packet with a destination MAC address that is not present in its forwarding table, it floods the packet to all its ports (except the incoming port). This ensures that the packet reaches all parts of the network.
5. **Update Forwarding Tables:** As packets are transmitted between hosts, switches continue to learn the association between MAC addresses and ports. This enables the switches to update their forwarding tables dynamically, allowing them to forward future packets directly to the correct outgoing port based on their destination MAC addresses.
6. **Destination Address Lookup:** When a switch receives a packet with a destination MAC address present in its forwarding table, it uses the table to determine the appropriate output port and forwards the packet to that port.

Please note that the specific forwarding tables and routing logic will depend on the network's actual topology, the addressing scheme used, and the forwarding algorithms implemented by the switches. For a concrete example, the network topology and forwarding tables would need to be provided.

Q13

Consider a virtual circuit network illustrated in Figure below, in which the hosts have addresses A, B, C, and so on. Assume that Host A wants to send packets to Host G. Show 3 the virtual circuit (VC) table entry for switches that are involved in packet routing from A to G (assume your virtual circuit identifier (VCI) for each switch).

ChatGPT

I apologize, but as an AI text-based model, I don't have access to external figures or images, including the specific figure you mentioned. Therefore, I don't have the exact topology or details of the switches in the network to construct the virtual circuit (VC) table entries.

However, I can provide you with a general idea of how virtual circuit (VC) table entries might be set up in a virtual circuit network to route packets from Host A to Host G:

1. **Virtual Circuit Identifier (VCI):** Each virtual circuit is assigned a unique Virtual Circuit Identifier (VCI) that helps identify the specific virtual circuit among different connections in the network.
2. **VC Table Entry Format:** The VC table entry for each switch typically includes the following information:
   * Incoming VCI: The VCI of the incoming packet from the previous switch or host.
   * Outgoing VCI: The VCI to be used when forwarding the packet to the next switch or host.
   * Outgoing Port: The output port through which the packet should be forwarded.
3. **Virtual Circuit Setup:** Before communication begins, a virtual circuit is established between Host A and Host G. This involves setting up VC table entries in each intermediate switch to route the packets properly.
4. **VC Table Entries for Switches:** Assuming there are multiple switches between Host A and Host G, the VC table entries in each switch along the path might look something like this:
   * Switch 1 (connected to Host A):
     + Incoming VCI: Not Applicable (N/A) since it's the source.
     + Outgoing VCI: VCI\_1 (a unique VCI value for the connection to the next switch in the path).
     + Outgoing Port: The output port connecting to Switch 2.
   * Switch 2 (intermediate switch):
     + Incoming VCI: VCI\_1 (matching the VCI used by Switch 1).
     + Outgoing VCI: VCI\_2 (a unique VCI value for the connection to the next switch in the path).
     + Outgoing Port: The output port connecting to Switch 3.
   * Switch 3 (connected to Host G):
     + Incoming VCI: VCI\_2 (matching the VCI used by Switch 2).
     + Outgoing VCI: Not Applicable (N/A) since it's the destination.
     + Outgoing Port: The output port connecting to Host G.

Please note that the actual VC table entries will depend on the network's specific topology, the virtual circuit setup process, and the addressing and VCI schemes used. The VCI values and the specific connections in the network would need to be provided for a more concrete example.

Q14

As an AI language model, I don't have the capability to display images or access external figures directly. Since you mentioned Figure (a) and Figure 4 (b), I don't have access to those specific figures to refer to the internetwork topology and the IP datagrams you described.

However, I can provide a general explanation of how IP fragmentation works and how the data, flag, and offset values are used in fragmented and unfragmented IP datagrams.

**IP Fragmentation:** IP fragmentation is a process that occurs when a datagram is too large to be transmitted over a network with a smaller Maximum Transmission Unit (MTU). In such cases, the original datagram is divided into smaller fragments to fit within the MTU of the network.

**Data, Flag, and Offset Values:**

1. **Data:** The "data" field represents the actual payload of the IP datagram, including the higher-layer data (e.g., TCP or UDP segments) to be delivered to the destination.
2. **Flag:** The "flag" field in the IP header includes three bits used for fragmentation control:
   * Bit 0 (Reserved): Always set to 0.
   * Bit 1 (Don't Fragment - DF): If set to 1, it indicates that the datagram should not be fragmented, and if it cannot be delivered without fragmentation, it should be discarded.
   * Bit 2 (More Fragments - MF): If set to 1, it indicates that this is not the last fragment of the datagram. If set to 0, it means that this is the last fragment.
3. **Offset:** The "offset" field in the IP header specifies the position of the data in the original datagram. It is measured in units of 8 bytes (64 bits). The offset value indicates where this fragment's data begins relative to the original datagram.

Since I don't have access to the specific figures and datagram details you mentioned, I'm unable to provide the exact data, flag, and offset values for the fragmented and unfragmented IP datagrams in a, b, c, d, and e. If you can provide more specific details or the figures directly, I'd be happy to help analyze them in more detail.