

**POST GRADUATE DIPLOMA IN INFORMATION SYSTEMS**

**MODULE**

**CIS 7206/ IS 509 DATA COMMUNICATION & NET ASSIGNMENT 01**

**SUBMITED DATE: 27TH JULY, 2023**

**LECTURER**

**Dr. JOHNSON MWEBAZE**

**STUDENT**

AKIKI WILFRED OLIRUS

**REG NO JAN23/PGDIS/1907U**

TEL: 0772099572, 0756602444

**JAN23 INTAKE**

**Data Communication and Networking Assignment 01**

**Question 1**

a) To calculate how much of UTAMU's network capacity students are consuming when watching YouTube, we need to calculate the data rate of the YouTube video streaming.

Data Rate of YouTube Video: Each frame is represented by 480 \* 360 \* 24 bits (pixel size) = 4,147,200 bits Number of frames per second = 30 frames per second

Data rate per second = 4,147,200 bits/frame \* 30 frames/second = 124,416,000 bits/second

Considering the 10:1 compression ratio: Effective Data rate per second = 124,416,000 bits/second / 10 = 12,441,600 bits/second

b) To calculate what percentage of available bus cycles on students' computers are consumed when watching YouTube, we need to compare the data rate of the YouTube video with the available bus speed.

Bus Speed of Students' Computers: Bus Speed = 400 MHz = 400,000,000 cycles/second Bus width = 32 bits

Data rate of the bus = 400,000,000 cycles/second \* 32 bits/cycle = 12,800,000,000 bits/second

Percentage of available bus cycles consumed by YouTube = (Data rate of YouTube / Data rate of the bus) \* 100 Percentage of available bus cycles consumed by YouTube = (12,441,600 bits/second / 12,800,000,000 bits/second) \* 100 ≈ 0.097%

c) Now, let's evaluate the impact of 500 students watching YouTube and a faster UTAMU Internet connection:

If there are 500 students on UTAMU network: Total data rate of YouTube streaming for 500 students = 500 \* 12,441,600 bits/second ≈ 6,220,800,000 bits/second

Total data rate of YouTube streaming for 500 students is about 6.22 Gbps (Gigabits per second).

If the UTAMU Internet connection is 45 Mbps (Mega-bits per second): UTAMU Internet connection speed = 45 Mbps = 45,000,000 bits/second

In this scenario, the total data rate of YouTube streaming for 500 students (6.22 Gbps) is much higher than the UTAMU Internet connection speed (45 Mbps). This means that the UTAMU Internet connection is insufficient to support all 500 students watching YouTube simultaneously, and it would lead to network congestion and poor video streaming performance.

Therefore, based on the calculations, it might be reasonable to impose some restrictions on YouTube streaming during peak hours or consider upgrading the Internet connection to accommodate the high data demand of students.

**Question 2**

Discuss the pros and cons of merging the OSI model's session, presentation, and application layers into a single Internet-centric application layer.

Merging the OSI model's session, presentation, and application layers into a single Internet-centric application layer has its own set of advantages and disadvantages. Let's discuss the pros and cons of this approach:

**Pros:**

1. Simplification: Combining the three layers into one can simplify the networking architecture and reduce complexity. It would streamline the communication process and make it easier to understand and implement applications.
2. Reduced Overhead: With fewer layers, there would be less encapsulation and de-encapsulation overhead, leading to more efficient data transfer.
3. Faster Communication: The reduced number of layers can potentially lead to faster communication between applications since there are fewer intermediaries.
4. Improved Performance: By eliminating the need for additional layers, the overall system performance may improve, especially in resource-constrained environments.
5. Internet-Centric Focus: The merged application layer would be specifically tailored for internet-based applications, potentially optimizing the protocols and functionalities for modern internet usage.

**Cons:**

1. Loss of Modularity: The OSI model's layering provides modularity, which makes it easier to update and maintain individual layers independently. Merging layers would result in a loss of this modularity, making it harder to make targeted changes or updates.
2. Lack of Flexibility: Combining session, presentation, and application layers could limit the flexibility in designing and implementing various application functionalities.
3. Compatibility Issues: Existing applications and protocols designed based on the traditional OSI model may not be directly compatible with the new merged layer. This could lead to the need for extensive modifications or even a complete redesign of existing applications.
4. Complexity for Developers: Merging multiple layers into one might make it more challenging for developers to understand and work with the application layer, especially for complex applications.
5. Increased Vulnerability: By combining different functionalities into one layer, there is a risk of increased security vulnerabilities. In the OSI model, separation of concerns helps in isolating and addressing security issues more effectively.
6. Limited Extensibility: The merged layer may not support easy extensibility for future application needs or the incorporation of new protocols.
7. Lack of Standardization: The OSI model's separation of layers allows for clear standardization at each level. Merging layers could lead to a lack of standardized protocols and conventions.

**Question 3**

1. Each station sends 5 frames: Since there are 4 other stations apart from the main station, a total of 5 frames will be sent by each station (including the main station).
2. Data frame size: Each data frame can hold 599 bytes of information.
3. Size of poll, ACK, and NAK frames: Each of the poll, ACK, and NAK frames is 64 bytes in length.

Now, let's calculate the total bytes transmitted for each station:

Total bytes transmitted by each station:

* Data frames: 5 data frames \* 599 bytes each = 5 \* 599 = 2995 bytes
* Poll frames: 5 poll frames \* 64 bytes each = 5 \* 64 = 320 bytes
* ACK frames: 5 ACK frames \* 64 bytes each = 5 \* 64 = 320 bytes
* NAK frames: 5 NAK frames \* 64 bytes each = 5 \* 64 = 320 bytes

Total bytes transmitted by each station = 2995 + 320 + 320 + 320 = 3955 bytes

There are a total of 5 stations including the main station,

Total bytes transmitted by all stations = 5 stations \* 3955 bytes each = 5 \* 3955 = 19775 bytes

Therefore, a total of 19775 bytes are transmitted in this scenario.

**Question 4**

In a CSMA/CD network with a data rate of 10 Mbps, the minimum frame size is found to be 512 bits for the correct operation of the collision detection process. What should be the minimum frame size if we increase the data rate to 100 Mbps? To 1 Gbps? To 10 Gbps?

In a CSMA/CD (Carrier Sense Multiple Access with Collision Detection) network, the minimum frame size is determined by the time required for collision detection. For CSMA/CD to work correctly, the frame must be long enough to ensure that a station can detect a collision before it has finished transmitting the entire frame.

The minimum frame size (in bits) required for collision detection is calculated using the following formula:

Minimum Frame Size = (2 \* Round-trip propagation delay \* Data rate) + 64 bits

Where:

* Round-trip propagation delay is the time it takes for a signal to travel from one end of the network to the other and back.
* Data rate is the transmission rate of the network in bits per second.

Therefore, calculating the minimum frame size for data rates of 100 Mbps, 1 Gbps, and 10 Gbps:

1. For a data rate of 100 Mbps: Minimum Frame Size = (2 \* Round-trip propagation delay \* 100,000,000) + 64 bits
2. For a data rate of 1 Gbps: Minimum Frame Size = (2 \* Round-trip propagation delay \* 1,000,000,000) + 64 bits
3. For a data rate of 10 Gbps: Minimum Frame Size = (2 \* Round-trip propagation delay \* 10,000,000,000) + 64 bits

The round-trip propagation delay will remain the same regardless of the data rate because it depends on the physical distance and the speed of light. However, it's important to note that as the data rate increases, the time to transmit a given number of bits decreases.

To ensure that the collision detection process works correctly, the minimum frame size must be such that the transmission time is longer than the round-trip propagation delay. As the data rate increases, the minimum frame size needs to increase to maintain a sufficient transmission time.

Since the round-trip propagation delay is a fixed value, we can conclude that the minimum frame size should increase as the data rate increases.

**Question 5**

A datagram is transmitted from host A to host B. The datagram is never delivered to Host B, and Host A is never informed that the transmission was unsuccessful. It would be helpful to hear two distinct interpretations of what took place

**Interpretation 1: The Datagram was Lost or Dropped**

In this interpretation, the datagram was successfully transmitted from host A to the network, but somewhere along the network path, it was lost or dropped before reaching host B. Several factors could contribute to this loss, such as network congestion, hardware failures, or routing issues. Host A, being the sender, has no direct knowledge of what happened to the datagram after it left its network interface. Since there is no feedback mechanism in this scenario, Host A remains unaware that the transmission was unsuccessful, and it assumes that the datagram reached its destination successfully.

**Interpretation 2: Successful Transmission, but No Acknowledgmen**t

In this interpretation, the datagram was successfully transmitted from host A to host B's network interface without any loss or errors. However, the acknowledgment or response from host B to host A acknowledging the successful reception was either lost or not sent back. This lack of acknowledgment could be due to issues at host B's end, such as the host being busy or encountering internal processing delays. Host A, having sent the datagram successfully, expects an acknowledgment from host B but does not receive one. Consequently, Host A remains unaware that the transmission was unsuccessful from the perspective of host B.

In both interpretations, there is no indication of failure or error at the sender's end (host A). The difference lies in where the failure or lack of acknowledgment is assumed to have occurred - either within the network path (interpretation 1) or at the receiving host B (interpretation 2). In either case, the lack of feedback or acknowledgment prevents Host A from knowing that the transmission was unsuccessful. This situation highlights the importance of feedback mechanisms in networking protocols, which help ensure reliable data transmission and provide information about the status of transmitted data.

**Question 6**

A data packet consisting of 36 bytes is sent down from the upper layer to an Ethernet MAC sublayer. How many bytes of padding are supposed to be inserted on top of the data?

In Ethernet, there is a minimum frame size requirement to ensure proper collision detection and transmission efficiency. If a data packet is smaller than this minimum size, padding is added to meet the minimum frame size.

The minimum frame size for Ethernet is 64 bytes (including the frame check sequence). If the data packet from the upper layer is smaller than this (36 bytes in this case), padding needs to be added.

Padding Size = Minimum Frame Size - Data Packet Size Padding Size = 64 bytes - 36 bytes Padding Size = 28 bytes

So, 28 bytes of padding are supposed to be inserted on top of the data to create an Ethernet frame of the minimum required size.

**Question 7**

A data packet consisting of 300 bytes is sent down from the upper layers to an Ethernet MAC sublayer. Is it possible to enclose all the data in a single frame? If this is not the case, how many frames must be sent? What is the total amount of data that is contained in each frame?

To determine if the data packet consisting of 300 bytes can be enclosed in a single Ethernet frame, we need to consider the maximum frame size supported by Ethernet.

In Ethernet, the maximum frame size for standard Ethernet (without jumbo frames) is 1500 bytes (including the frame check sequence). This is also known as the Maximum Transmission Unit (MTU) for Ethernet.

If the data packet size is 300 bytes, it is smaller than the maximum frame size (1500 bytes). Meaning therefore, it is possible to enclose all the data in a single frame.

The total amount of data contained in each frame will be the size of the data packet of 300 bytes.

So, only one frame is needed to send the entire 300 bytes of data.

**Question 8**

The IP address 192.168.12.56/16 represents a host within a Class B private IP address block. The "/16" indicates that the first 16 bits of the IP address represent the network portion, and the remaining 16 bits represent the host portion.

1. Network Address, to find the network address, we need to set all the host bits to 0. In this case, the first 16 bits are the network portion, so the network address would be: Network Address = 192.168.0.0
2. Broadcast Address, To find the broadcast address, we need to set all the host bits to 1. In this case, the last 16 bits are the host portion, so the broadcast address would be: Broadcast Address = 192.168.255.255
3. Other IP Addresses in this Block, Since the network portion is fixed (192.168.0.0), we can change the host portion to get other IP addresses within the same block.

Here are three other IP addresses that are part of this block:

* 192.168.12.1
* 192.168.12.100
* 192.168.12.254

These addresses have the same network portion (192.168.0.0) as the given IP address and fall within the range of assignable host addresses (from 192.168.0.1 to 192.168.255.254) in the /16 subnet.

**Question 9**

a) Find the Subnet Mask:

To create 1024 subnets from the IP address block 10.56.0.0/16, we need to borrow some bits from the host portion to extend the subnet mask. The formula to find the number of bits required for 1024 subnets is:

Number of Subnet Bits = log2(Number of Subnets)

Number of Subnet Bits = log2(1024) = 10 bits

Since the original subnet mask is /16, it has 16 network bits. To create 1024 subnets, we need to add 10 more bits to the subnet mask, making the new subnet mask:

New Subnet Mask = /16 + 10 = /26

b) Find the Number of Addresses in Each Subnet:

With the new subnet mask of /26, we have 32 - 26 = 6 host bits remaining. The number of addresses in each subnet is calculated using the formula:

Number of Addresses = 2^(Number of Host Bits) - 2

Number of Addresses = 2^6 - 2 = 64 - 2 = 62 addresses

c) Find the First and Last Addresses in Subnet 1:

Subnet 1 is the first subnet, and its subnet number is 10.56.0.0. To find the first address in this subnet, set all the host bits to 0:

First Address in Subnet 1 = 10.56.0.0

To find the last address in this subnet, set all the host bits to 1:

Last Address in Subnet 1 = 10.56.0.63

d) Find the First and Last Addresses in Subnet 1024:

Subnet 1024 is the last subnet, and its subnet number is 10.56.15.0 (since we are using 10 bits for subnetting). To find the first address in this subnet, set all the host bits to 0:

First Address in Subnet 1024 = 10.56.15.0

To find the last address in this subnet, set all the host bits to 1:

Last Address in Subnet 1024 = 10.56.15.63

**Question 10**

To distribute the IP address block 150.80.0.0/16 to 2000 customers as specified, we need to create subblocks for each group based on their address requirements.

a) For the first group with 200 universities, each needing 128 addresses: The closest power of 2 greater than or equal to 128 is 256 (2^8). So, each university will be allocated a subblock of /24, which provides 256 addresses per subnet (including network and broadcast addresses).

Total addresses for the first group = 200 universities \* 128 addresses each = 200 \* 128 = 25,600 addresses Subblock for each university = /24

b) For the second group with 400 secondary schools, each needing 16 addresses: The closest power of 2 greater than or equal to 16 is 16 (2^4). So, each secondary school will be allocated a subblock of /28, which provides 16 addresses per subnet (including network and broadcast addresses).

Total addresses for the second group = 400 secondary schools \* 16 addresses each = 400 \* 16 = 6,400 addresses Subblock for each secondary school = /28

c) For the third group with 1400 SMEs, each needing 4 addresses: The closest power of 2 greater than or equal to 4 is 8 (2^3). So, each SME will be allocated a subblock of /29, which provides 8 addresses per subnet (including network and broadcast addresses).

Total addresses for the third group = 1400 SMEs \* 4 addresses each = 1400 \* 4 = 5,600 addresses Subblock for each SME = /29

To calculate the total number of addresses used by these subblocks:

Total addresses used = Total addresses for the first group + Total addresses for the second group + Total addresses for the third group Total addresses used = 25,600 + 6,400 + 5,600 = 37,600 addresses

To find out how many addresses are still available after these allocations:

Total addresses in the original block = 2^16 = 65,536 addresses Remaining addresses = Total addresses in the original block - Total addresses used Remaining addresses = 65,536 - 37,600 = 27,936 addresses

After these allocations, there are 27,936 addresses still available in the IP address block 150.80.0.0/16.

**Question 11**

To find the shortest form of the given IPv6 addresses, we can eliminate leading zeros and use the "::" notation for consecutive blocks of zeros.

a) A000:0000:0000:0000:0000:2340:1ABC:119A Shortest form: A000::2340:1ABC:119A

b) 119A:A001:0000:2340:0000:0000:0000:0000 Shortest form: 119A:A001:0:2340:: (Note: We can collapse consecutive blocks of zeros using "::" only once in an IPv6 address.)

c) 3400:0000:0000:0000:0000:0000:0000:0000 Shortest form: 3400::

To show the original (unabbreviated) form of the given IPv6 addresses, we expand the "::" notation and add leading zeros to each block as necessary.

a) 0::0 Original form: 0000:0000:0000:0000:0000:0000:0000:0000

b) b.0:AA::0 Original form: 0B00:00AA:0000:0000:0000:0000:0000:0000

c) 0:1234::3 Original form: 0000:1234:0000:0000:0000:0000:0000:0003

d) 123::1:2 Original form: 0123:0000:0000:0000:0000:0000:0001:0002

**Question 12**

List three transition strategies to move from IPv4 to IPv6. When is each strategy used?

Transitioning from IPv4 to IPv6 is essential due to the exhaustion of available IPv4 addresses and the need to support the growing number of devices connected to the internet. Several transition strategies facilitate the migration from IPv4 to IPv6. Here are three common strategies and when they are typically used:

1. Dual Stack:

Strategy: The dual-stack strategy involves running both IPv4 and IPv6 protocols simultaneously on devices and network infrastructure. This means that devices and routers are configured with both IPv4 and IPv6 addresses, and they can communicate using either protocol.

**Usage**: Dual stack is widely used during the transition period when IPv4 and IPv6 coexist. It allows for gradual adoption of IPv6 without disrupting existing IPv4 services. Devices and networks can communicate with both IPv4 and IPv6 hosts during the transition, allowing for a seamless transition for end-users and minimizing service disruption.

1. Tunneling:

Strategy: Tunneling involves encapsulating IPv6 packets within IPv4 packets to transport them over an IPv4 network. This enables IPv6 communication between two endpoints that are not directly connected by an IPv6-capable network.

**Usage:** Tunneling is used when there is no native IPv6 support in some parts of the network infrastructure. It creates a virtual IPv6 link over the existing IPv4 network, allowing communication between IPv6-enabled endpoints across IPv4-only regions. Tunneling can be configured between routers or implemented using tunnel brokers to enable IPv6 connectivity over an IPv4 network.

1. Network Address Translation 64 (NAT64) and DNS64:

Strategy: NAT64 and DNS64 work together to allow IPv6-only devices to communicate with IPv4-only devices and services. NAT64 translates IPv6 addresses to IPv4 addresses, and DNS64 synthesizes IPv6 addresses for IPv4-only domains.

**Usage:** NAT64 and DNS64 are used to facilitate communication between IPv6-only devices and IPv4-only servers. As more devices transition to IPv6, some legacy systems and services may still be IPv4-only. NAT64 and DNS64 provide a mechanism for these devices to communicate with each other until they fully transition to IPv6.