AON Finals

Question 2:

1. Content Distribution Networks (CDNs) can improve web performance in two main ways. Firstly, by caching content at servers closer to the end user, CDNs can reduce the distance data needs to travel, resulting in faster load times. Secondly, CDNs can distribute traffic across multiple servers, reducing the load on any one server and improving the overall performance of the web service.
2. TCP’s reliable data transfer protocol is similar to Go-Back-N in that they both use sequence numbers to ensure that data is received in the correct order. However, TCP uses selective repeat, which allows it to retransmit only the missing packets, whereas Go-Back-N requires the entire window of packets to be resent.
3. SDN, or Software-Defined Networking, differs from the traditional per-router approach in that it centralizes network management and control in a software controller rather than distributing it across multiple routers. This allows for more efficient and flexible network management, as the controller can programmatically control network behavior and routing policies.
4. Traditional routing involves forwarding packets based on destination addresses, whereas generalized forwarding takes into account additional information such as the type of application or the quality of service required by the network traffic. Generalized forwarding allows for more intelligent and efficient use of network resources, but requires more complex routing algorithms and hardware.

2.
TCP and Go-Back-N both use sequence numbers to track packets for reliable data transfer, which enables the receiver to identify and eliminate duplicate packets. However, they differ in the way they handle retransmissions. Go-Back-N discards out-of-order packets and retransmits all packets from the lost one onward. In contrast, TCP uses selective retransmission to resend only the lost packet, thereby making better use of network resources.

3.
The traditional per-router approach to networking involves configuring each network device separately, while SDN centralizes network control through a software-based controller, which manages the network by configuring traffic flow and determining the best path for data to take. This makes network management

more efficient and flexible compared to the traditional approach where devices make decisions based on their individual knowledge of the network.

4.
Traditional routing and generalized forwarding are two different methods of network forwarding. In traditional routing, each network device makes individual forwarding decisions based on routing protocols and destination address.

In contrast, generalized forwarding allows for centralized control of network traffic through a software-based controller. Network devices are configured to forward packets based on matching header fields, promoting efficient use of network resources and enabling dynamic routing based on real-time network conditions. Generalized forwarding is more flexible, efficient, and adaptable than traditional routing, which is more rigid and limited in its ability to respond to changing network conditions and service demands.

Question 3:

1.
To calculate the round trip time (RTT), we need to consider the propagation delay and transmission delay for both links. Let's first calculate the transmission delay for a packet of size 1500 bytes:
Transmission delay (A-S) = (1500 x 8) bits / (3 x 10^8) m/s = 0.04 ms
Transmission delay (S-B) = (1500 x 8) bits / (2 x 10^8) m/s = 0.06 ms

The total transmission delay for the round trip is 0.1 ms. Now let's calculate the propagation delay:

Propagation delay (A-S) = 1000 m / (3 x 10^8) m/s = 0.00333 ms
Propagation delay (S-B) = 5000 m / (2 x 10^8) m/s = 0.025 ms

The total propagation delay for the round trip is 0.02833 ms. Therefore, the total round trip time is:

RTT = 2 x (transmission delay + propagation delay) = 2 x (0.1 + 0.02833) = 0.25666 ms

2.
With basic non-persistent HTTP, the client needs to establish a new TCP connection for each requested object. Therefore, for the 8 referenced objects, we will have 8 TCP connections. Each TCP connection requires a three-way handshake, which means three packets need to be exchanged between A and B (SYN, SYN-ACK, ACK) before the data transfer can start. In addition, for each TCP connection, an HTTP request and a response need to be exchanged. Therefore,

the total number of packets that need to be sent from A to B is:
Total packets = 8 x (3 + 2) + 1 + 1 = 28

Where 8 is the number of referenced objects, 3 is the number of packets for the TCP connection setup (SYN, SYN-ACK, ACK), 2 is the number of packets for HTTP request and response, and 1 is the number of packets for the base HTML file request and response.

Using the RTT calculated in part 1, we can calculate the total time to retrieve all objects as:

Total time = RTT x Total packets = 0.25666 ms x 28 = 7.18 ms

Therefore, it will take approximately 7.18 ms to retrieve all objects using basic non-persistent HTTP. Note that this calculation does not consider queueing or processing delays, which can add to the total time.

Question 4:

1. TCP slow start operates between the intervals of [1,6] and [23,26].

TCP congestion avoidance operates between the intervals of [6,16] and [17,22].

2. After round 16 of transmission, TCP utilizes a triple duplicate acknowledgement to detect segment loss. This is because TCP utilizes a fast retransmission technique that assumes a segment has been lost and immediately retransmits it upon receiving three duplicate acknowledgements.
If a segment is not received properly, the receiver sends an "ACK" (acknowledgment) to let the sender know. If the sender receives three duplicate ACKs, it assumes that a segment has been lost and retransmits it right away. In this particular case, after 16 rounds of transmission, if TCP detects three duplicate ACKs, it assumes a segment has been lost and triggers a quick retransmitting technique.

3. The initial value of ssthresh at transmission round 1 is 32. The value of ssthresh at transmission round 8 is 42.

4. When the loss happened, the threshold was set to half the congestion window's current value of 8 and the congestion window was set to the new threshold value plus 3 MSS. As a result, the threshold and window will now have new values of 4 and 7, respectively.

After the loss occurred, the threshold was adjusted to be equal to half of the current value of the congestion window, which was 8. The congestion window was then set to a value equal to the new threshold value plus 3 maximum segment sizes (MSS). This caused the threshold and window to have new values of 4 and 7, respectively.

5. round 17, 1 packet; round 18, 2 packets; round 19, 4 packets; round 20, 8 packets; round 21, 16 packets; round 22, 21 packets. So, the total number is 52.

Question 5:

1.
The purpose of using error detection techniques such as CRC or checksums in the Internet is to ensure the integrity of transmitted data. These techniques add redundancy to the data, allowing the receiver to detect and potentially correct errors that may occur during transmission. This helps to ensure that the data received is the same as the data that was transmitted.

1. The purpose of using error detection techniques such as CRC or checksums in the internet is to ensure that the data being transmitted from one computer to another arrives without any errors. When data is sent over a network, it travels through various devices such as routers, switches, and cables, which may introduce errors. Errors can occur due to various reasons like electromagnetic interference, faulty cables, etc. These errors can corrupt the data, and if not detected, they can lead to serious consequences like malfunctioning of the devices or data corruption. Error detection techniques like CRC and checksums are used to detect such errors and ensure that the data received is the same as the data transmitted.

2.
To calculate the CRC value, the user data is first left-shifted by 4 bits, and then the generator polynomial is divided into the resulting bit string using modulo-2 division. The remainder of this division is the CRC value, which is appended to the user data and transmitted as a single bit string.

In this scenario, the user data is 1100101. After left-shifting by 4 bits, it becomes 11001010000. The generator polynomial is 10011. So, performing modulo-2 division, we get the remainder 0001. This is the CRC value. The actual bit string transmitted is the user data followed by the CRC value, which is 11001010001.

3.
When the third bit from the left of the user data is inverted during transmission, the resulting bit string becomes 1110101. At the receiver's end, the same process

of CRC calculation is performed. When the generator polynomial is divided into this bit string using modulo-2 division, the remainder obtained is not equal to the previously calculated CRC value of 0001. This indicates that an error has occurred during transmission, and the receiver can request the sender to retransmit the data to ensure data integrity.

3. The receiver can detect the error by performing the same CRC calculation as the sender did and checking if the remainder obtained matches the CRC value received. If the remainder obtained is different from the CRC value received, it means that an error has occurred during transmission. In this case, the third bit from the left of the user data is inverted, which means that the received data will be 111010111. When the receiver calculates the CRC value using this received data, the remainder obtained will be different from the CRC value received. Hence, the receiver will detect the error and request the sender to retransmit the data.