[CO1 and CO2]

1. **Define mobility binding w.r.t IPv6.**

   In IPv6, a mobility binding is the relationship between a mobile node and its home agent, allowing the MN to maintain its IPv6 address as it moves to a new network. The MN obtains a temporary address known as a care-of address in the new network and informs its HA of its new CoA. The HA then creates a mobility binding that associates the MN's home address with its CoA. Mobility bindings are important for maintaining end-to-end connectivity for mobile nodes in IPv6 networks, allowing them to continue to use their home address and maintain active sessions with other nodes as they move between networks.

2. **Why and how would you secure binding messages?**

   Securing binding messages is important in IPv6 mobility to prevent unauthorized nodes from modifying or hijacking mobility bindings between a mobile node (MN) and its home agent (HA).

   Binding messages are used to create, update, or remove a mobility binding between an MN and its HA. If these messages are not secured, an attacker could intercept and modify them to create false bindings or hijack existing bindings, leading to security breaches and network disruptions. To secure binding messages, various security mechanisms can be used, such as:

   1. IPsec: IPsec can be used to provide authentication and encryption of binding messages between the MN and HA, ensuring that only authorized parties can access and modify the mobility bindings.

   2. Cryptographically Generated Addresses (CGA): CGA is a technique that uses a public key infrastructure to generate unique IPv6 addresses for mobile nodes, which can be used to verify the authenticity of the MN and protect against address spoofing attacks.

   3. Authentication Header (AH): AH is a security protocol that provides data authentication and integrity for IPv6 packets. It can be used to protect binding messages from tampering and ensure that the MN and HA can verify each other's identities.

   By securing binding messages, an MN and its HA can ensure that only authorized parties can access and modify the mobility bindings, protecting the integrity and availability of the network.

3. **Draw the internet protocol stack used for progressive download based applications.**

   The Internet Protocol stack used for progressive download-based applications:

   Application layer -> HTTP/HTTPS (HyperText Transfer Protocol/Secure)

   Transport layer -> TCP (Transmission Control Protocol)

   Network layer -> IP (Internet Protocol), ICMP (Internet Control Message Protocol) (optional)

   Data link layer -> Ethernet (or other data link protocol)

   Physical layer -> Electrical, optical, or wireless transmission medium

   This protocol stack is used for delivering content through progressive download, where the video or audio file is downloaded to the client device and played back locally. The HTTP/HTTPS protocol is used to transfer the file from the server to the client over the internet. TCP is used as the transport protocol to ensure reliable delivery of the file data, and IP is used to route the data between the server and the client. The data link layer is responsible for transmitting the data over the physical medium, such as Ethernet, and the physical layer is the actual transmission medium itself.

4. **State the different addressing formats of IPv6. Which ones of them are utilized in supporting mobility? How?**

   IPv6 supports three different addressing formats, as follows:
   Unicast Address: A unicast address is an IPv6 address that identifies a single interface on a network. Unicast addresses can be global or link-local.
   Multicast Address: A multicast address is an IPv6 address that identifies a group of interfaces on a network. Packets sent to a multicast address are delivered to all interfaces that belong to the multicast group.
   Anycast Address: An anycast address is an IPv6 address that identifies a group of interfaces on different nodes. When a packet is sent to an anycast address, it is delivered to the nearest interface that belongs to the anycast group.
   Both unicast and anycast addresses are used to support mobility in IPv6 networks.
   Unicast addresses are used to identify the mobile node's home address, which remains constant regardless of the node's current location. When the mobile node moves to a new network, it obtains a temporary address known as a care-of address. The mobile node then uses its home address as the source address for packets, which are routed to its care-of address using IPv6 routing mechanisms.
   Anycast addresses are used to identify the closest access point to a mobile node's current location. When a mobile node moves to a new network, it sends a router solicitation message using an anycast address to discover the closest access point. The anycast address is associated with a group of routers, and the router closest to the mobile node responds to the solicitation message, providing the mobile node with the information it needs to establish a new care-of address.
   By using unicast and anycast addresses in this way, IPv6 mobility allows mobile nodes to maintain active sessions with other nodes and services, even as they move between networks, without the need for reconfiguration or interruption of service.

5. **What will happen if (i) the correspondent node fails to receive a binding update message, (ii) a home agent fails to receive a binding update message?**

   If a correspondent node fails to receive a binding update message, the node will continue to use the old binding information for the mobile node, even though it is no longer valid. This may result in the node sending packets to the mobile node's previous location, causing packet loss or delay. To avoid this, the correspondent node should send a binding update to the mobile node's new address, which can be obtained by sending a neighbor solicitation message.
   If a home agent fails to receive a binding update message, the agent will continue to use the old binding information to forward packets to the mobile node's care-of address. This may result in packet loss or delay, as the packets will not be delivered to the mobile node's new location. To avoid this, the home agent should periodically send binding updates to the mobile node's new care-of address, even if it has not received a binding update from the mobile node.
   In both cases, failure to receive binding update messages can result in disruptions to network connectivity, leading to loss of packets and delay. Therefore, it is important to ensure that binding updates are sent and received in a timely and reliable manner to maintain uninterrupted network connectivity.

6. **Do you think proxy neighbour advertisement mechanism is better than address resolution protocol? Give reasons in the context of node mobility.**

   Proxy Neighbor Advertisement (PNA) mechanism is a better alternative to the Address Resolution Protocol (ARP) in the context of node mobility for several reasons.

   When a mobile node changes its network, it is assigned a new IP address, and the previous address becomes invalid. If other nodes in the network have cached the MAC address of the mobile node based on its previous IP address, they will continue to send packets to the old address, resulting in a loss of connectivity.

   In such scenarios, ARP would not be able to resolve the new MAC address of the mobile node because the cache on the other nodes has not been updated. This problem is known as the "ARP cache problem" and can lead to significant disruptions in the network.

   In contrast, the PNA mechanism allows the home agent to act as a proxy for the mobile node and respond to neighbor solicitation messages on behalf of the mobile node, even if the mobile node is not on the same network. This ensures that other nodes in the network can update their cache with the mobile node's new MAC address, even when it changes networks, ensuring uninterrupted connectivity.

   Moreover, the PNA mechanism also allows for better security by ensuring that only authorized nodes can access the mobile node's new MAC address.

   Therefore, in the context of node mobility, the PNA mechanism is a better alternative to the ARP, as it ensures uninterrupted connectivity and better security.

7. **Name one application each that uses push and pull based protocols. Justify your answer.**

   An example of an application that uses push-based protocol is web push notifications. In push-based protocol, the server initiates the communication and sends data or information to the client without any request from the client. Web push notifications enable websites to send real-time notifications to users even when the website is not open.

   An example of an application that uses pull-based protocol is email. In pull-based protocol, the client initiates the communication and requests data or information from the server. Email clients such as Gmail, Yahoo Mail, etc. use pull-based protocol to retrieve emails from the server. The client periodically checks for new emails by requesting data from the server.

8. **Describe a streaming based protocol.**

   One example of a streaming-based protocol is the Real-Time Messaging Protocol (RTMP). It is a protocol developed by Adobe Systems for the streaming of audio, video and data over the Internet between a Flash player and a server.

   RTMP operates over Transmission Control Protocol (TCP) and works by establishing a persistent connection between a client and a server. The protocol allows for the delivery of real-time data, which is particularly important for live streaming.

   The protocol works by breaking the data into small chunks, called packets, and sending them over the network to the client. The client then reassembles the packets into a continuous stream of data. RTMP also includes mechanisms for error correction and bandwidth control, which help ensure a smooth streaming experience for the user.

   In addition to the basic functionality of delivering audio, video and data, RTMP also supports a number of advanced features such as encryption, authentication and adaptive streaming. These features make it a popular choice for many different types of streaming applications, from live sports events to video-on-demand services.

9. **State the features of adaptive streaming that are similar to the progressive download and the features dissimilar to progressive download**.

Adaptive streaming shares some similarities with progressive download, such as the ability to start playback before the entire video file is downloaded, enabling faster startup times and reducing buffering. Both techniques allow for seeking and pause/resume functionality, and can utilize a content delivery network (CDN) to improve streaming performance.

However, adaptive streaming differs from progressive download in several ways. With adaptive streaming, the video is encoded at multiple bitrates and resolutions, allowing for the client device to switch between streams to adapt to changes in network conditions. This allows for a more seamless viewing experience and can help avoid buffering and interruptions due to insufficient bandwidth. In contrast, progressive download streams the video at a fixed bitrate and resolution, regardless of network conditions.

Adaptive streaming also requires specialized servers and software to encode the video at multiple bitrates and resolutions and to manage the switching between streams, while progressive download can be implemented with standard HTTP servers and video players. Additionally, adaptive streaming can require more processing power and storage on the client device to manage the switching between streams.

10. **How does streaming of audio differ from video if quality of experience is a prime concern?**

When it comes to streaming, the quality of experience (QoE) is a prime concern for both audio and video content. However, there are some differences in how streaming of audio differs from video in terms of QoE.

Firstly, video streaming requires more bandwidth than audio streaming to maintain the same level of quality. This means that video streaming is more prone to buffering and interruptions due to fluctuations in network conditions. Audio streaming, on the other hand, is more resilient to network fluctuations and can maintain a consistent level of quality even with lower bandwidth.

Secondly, the impact of interruptions on QoE is more noticeable in video streaming than in audio streaming. A brief interruption in a video stream can result in a loss of synchronization between audio and video, leading to a poor viewing experience. In contrast, a brief interruption in an audio stream may be less noticeable or disruptive to the listening experience.

Finally, the QoE considerations for video streaming often extend beyond just the quality of the video itself. Factors such as the playback speed, resolution, and aspect ratio can also affect the viewing experience. With audio streaming, the focus is primarily on the quality of the audio itself, such as the bit rate and sample rate.

Overall, while both audio and video streaming require attention to QoE, video streaming can be more complex due to the higher bandwidth requirements and additional factors that can impact the viewing experience.

11. Why is UDP not sufficient for video streaming applications? Does TCP address those issues?

UDP (User Datagram Protocol) is not sufficient for video streaming applications because it does not provide any reliability or error correction mechanisms. UDP is a connectionless protocol that does not guarantee delivery of packets, so packets may be lost, duplicated, or arrive out of order. This can result in a poor quality viewing experience for users, with video stuttering or freezing due to missing or out-of-order frames.

TCP (Transmission Control Protocol) can address some of the issues with UDP for video streaming applications. TCP provides reliable, ordered delivery of packets and includes error detection and correction mechanisms. This means that lost or corrupted packets are retransmitted, and packets are delivered in the correct order. This results in a more stable and consistent viewing experience for users, with fewer interruptions due to packet loss or corruption.

However, TCP also has some limitations that can impact video streaming performance. TCP introduces more latency due to the additional overhead of connection setup and error correction, which can result in a delay between the time the user requests the video and the time it starts playing. TCP also has higher bandwidth requirements due to the additional packet overhead, which can result in slower download speeds for large video files.

Overall, while TCP can address some of the issues with UDP for video streaming applications, it may not be the optimal solution for all scenarios. Other protocols and techniques, such as adaptive streaming and content delivery networks, may be used in conjunction with TCP to provide a better viewing experience for users.

12. What is the most important quality of service parameter for multimedia streaming? How does RTP address that?

One of the most important quality of service (QoS) parameters for multimedia streaming is latency or delay. Latency refers to the time it takes for a packet to travel from the source to the destination, and it can have a significant impact on the viewing or listening experience for multimedia content.

RTP (Real-time Transport Protocol) is a protocol designed to address latency and other QoS issues for multimedia streaming. RTP is a transport protocol that provides end-to-end delivery of real-time audio and video data over IP networks. It includes mechanisms for timestamping packets and calculating inter-arrival time, which can be used to estimate the end-to-end delay and adjust the playout time at the receiver.

In addition, RTP can be used in conjunction with other protocols and techniques to further improve QoS for multimedia streaming. For example, RTP can be used with RTCP (Real-time Transport Control Protocol) to provide feedback on packet loss, jitter, and other QoS metrics. RTP can also be used with techniques such as forward error correction (FEC) and retransmission to improve reliability and reduce the impact of packet loss on the viewing experience.

Overall, RTP is an important protocol for multimedia streaming because it addresses the key QoS parameter of latency and provides a framework for other QoS techniques to be used in conjunction with it.

13. Is RTSP push based or pull based? Give reasons.

RTSP (Real-Time Streaming Protocol) is a pull-based protocol, which means that the client initiates a request to the server to receive the stream data. The server responds to the client's request by sending the data over the network.

The reason RTSP is pull-based is because it was designed to be a control protocol that allows the client to request and control the playback of multimedia streams. The client sends RTSP requests to the server to initiate playback or to modify playback parameters such as the playback rate or the position within the stream. The server responds to these requests by sending the appropriate data to the client.

In contrast, a push-based protocol would involve the server sending data to the client without the client requesting it. This approach would not be well-suited to RTSP because it would not allow the client to control the playback of the stream or to request specific data from the server.

Overall, RTSP's pull-based approach allows for greater control and flexibility in streaming multimedia content over the network, and is well-suited to the protocol's primary function as a control protocol.

14. For pull based streaming applications, which features need to be measured/monitored by the client?

For pull-based streaming applications, the client needs to monitor the buffering time, available bandwidth, and buffer level. The buffering time is the time required to fill the buffer. Available bandwidth is the maximum amount of data that can be downloaded per unit time. Buffer level is the amount of data currently stored in the buffer. By monitoring these features, the client can adapt the streaming rate to avoid buffering and maintain smooth playback.

15. How are the movie fragments randomly accessed following Microsoft smooth streaming protocol?

Microsoft Smooth Streaming protocol uses a technique called chunked transfer encoding to divide a video into small fragments or chunks. Each chunk is typically a few seconds long and can be easily downloaded by the client device.

The chunks are stored on the server as individual files and are made available to the client on-demand. The client can request a specific chunk or segments of chunks, allowing for random access to any part of the video.

To support random access, the server maintains a manifest file that contains metadata about each chunk, such as its duration, bitrate, and location. The client uses this metadata to construct a request for the desired chunk or segment of chunks.

The server then sends the requested chunks to the client in real-time, adjusting the bitrate based on the current network conditions and available bandwidth. This adaptive bitrate streaming technique ensures that the video is delivered smoothly and without interruption, even if the network conditions are fluctuating.

Overall, Microsoft Smooth Streaming protocol provides a seamless streaming experience to the end-user by allowing them to access any part of the video on-demand while optimizing the video quality based on the network conditions.

16. **How does RTP handle audio and video data?**

RTP (Real-time Transport Protocol) is a protocol designed for transmitting real-time multimedia data such as audio and video over IP networks. RTP handles audio and video data by dividing it into small, manageable packets that can be transmitted across the network. Each RTP packet contains a sequence number, a timestamp, and a payload. The payload contains the audio or video data itself. The sequence number is used to detect packet loss and to reorder packets that arrive out of order. The timestamp is used to synchronize the audio and video streams at the receiver.

In addition to RTP, a companion protocol called RTCP (RTP Control Protocol) is used to provide feedback on the quality of the transmission. RTCP packets are sent periodically and contain information about packet loss, delay, and other statistics. This feedback can be used to adjust the transmission parameters to improve the quality of the audio and video streams.

17. **What are the three classes of multimedia applications? Give 1 example each.**

The three classes of multimedia applications are:

Linear Presentation: These are time-based applications that present multimedia content in a linear sequence. Examples include movies, documentaries, and video lectures.

Interactive Multimedia: These are non-linear applications that allow users to interact with multimedia content. Examples include video games, educational software, and simulation programs.

Virtual Reality: These are immersive applications that create a 3D environment for users to interact with. Examples include virtual reality games, virtual tours, and training simulations.

18. **Discuss adaptive streaming**.

Adaptive streaming adjusts the quality of video streaming based on available network bandwidth and device capabilities to provide the best possible viewing experience while avoiding buffering and interruptions due to insufficient bandwidth. The video is encoded at multiple bitrates and resolutions, and the client selects the appropriate stream based on current network conditions. There are two main approaches to adaptive streaming: server-side and client-side. In server-side adaptive streaming, the server selects the appropriate video stream based on the client's network conditions and device capabilities, while in client-side adaptive streaming, the client device is responsible for selecting the appropriate video stream. Adaptive streaming is used in popular video streaming services and allows for a better viewing experience on a variety of devices and network conditions.

19. **Does HTTP 2.0 help in multimedia streaming? Justify.**

Yes, HTTP 2.0 helps in multimedia streaming. HTTP 2.0 is a major revision of the HTTP network protocol, and it is designed to improve website and application performance. It introduces several new features that are particularly useful for multimedia streaming, including multiplexing, server push, and binary framing. Multiplexing allows multiple requests and responses to be sent and received simultaneously over a single connection, which can reduce latency and improve throughput. Server push allows the server to initiate the transmission of resources that the client hasn't requested yet, which can further reduce latency and improve performance. Binary framing is a new way of encoding data that is more efficient than the text-based format used in HTTP 1.1, which can improve performance and reduce bandwidth usage. Overall, these features make HTTP 2.0 a better protocol for multimedia streaming than HTTP 1.1.

20. Compare between progressive download and adaptive streaming.

Progressive download and adaptive streaming are two methods used for delivering multimedia content over the internet.

Progressive download is a technique in which a multimedia file is downloaded to the user's device and can be played as it is being downloaded. The user does not have to wait for the entire file to be downloaded before playing it. However, if the user's internet connection is slow, the playback may be choppy or stop frequently. An example of progressive download is downloading a video file from a website and playing it using a media player.

Adaptive streaming, on the other hand, is a technique in which the multimedia content is divided into small chunks and each chunk is delivered to the user based on the user's internet speed and device capabilities. The user's device automatically selects the best quality based on the available bandwidth and the device's capabilities. This allows for a smoother playback experience, as the user can switch between different qualities seamlessly. An example of adaptive streaming is Netflix, where the quality of the video adjusts automatically based on the user's internet speed and device capabilities.

21. Which frames constitute a GoP and what are their purpose?

In video coding, a Group of Pictures (GoP) is a sequence of consecutive frames in a video stream that are used for inter-frame prediction. A GoP typically consists of the following types of frames:

I-frames (Intra-coded frames): These are keyframes that are encoded without any reference to other frames. I-frames are fully coded and contain all the necessary information to reconstruct a single frame of the video. They are used as reference frames for inter-frame prediction.

P-frames (Predictive-coded frames): These frames are encoded based on the difference between the current frame and the previous I-frame or P-frame. They only contain the changes in the image data from the previous frame and require less data to transmit than I-frames.

B-frames (Bidirectional-coded frames): These frames are encoded based on the differences between the previous and next I-frames or P-frames. They can provide better compression than P-frames because they can use information from both past and future frames to predict the current frame.

The purpose of using a GoP is to improve compression efficiency by exploiting the temporal redundancy in a video stream. By predicting the content of frames based on previous frames, fewer bits are required to represent the video, resulting in a smaller file size and faster transmission.

22. What is the significance of GoP for streaming applications?

A Group of Pictures (GoP) is a collection of successive frames in a video stream that are interdependent and share information. The frames within a GoP include an I-frame (intra-coded frame), which is a complete image, and several B-frames (bi-directional frames) and P-frames (predictive frames) that reference the I-frame and each other to form a compressed video stream.

The significance of GoP for streaming applications is that it allows for efficient compression of video data by exploiting the temporal redundancy between frames. By referencing previous frames within a GoP, the encoder can reduce the amount of data needed to represent subsequent frames, resulting in smaller file sizes and faster transmission over limited bandwidth connections. Additionally, GoP can help improve video quality by reducing artifacts and increasing the accuracy of motion prediction.

23. Compare between push and pull based applications when the server is hosted using some cloud service and the clients are smartphones.

An example of an application that uses push-based protocol is web push notifications. In push-based protocol, the server initiates the communication and sends data or information to the client without any request from the client. Web push notifications enable websites to send real-time notifications to users even when the website is not open.

An example of an application that uses pull-based protocol is email. In pull-based protocol, the client initiates the communication and requests data or information from the server. Email clients such as Gmail, Yahoo Mail, etc. use pull-based protocol to retrieve emails from the server. The client periodically checks for new emails by requesting data from the server.

24. Compare between push and pull based applications when the server is hosted using some cloud service and the clients are smartphones.

In the context of cloud-hosted server and smartphone clients, push-based applications are generally more suitable for real-time or near real-time updates, where the server needs to send data updates to the clients as soon as they are available. Push-based applications typically require a persistent connection between the server and the clients, which can be more challenging to maintain in a cloud-based environment.

On the other hand, pull-based applications are better suited for scenarios where the clients need to retrieve data on demand, as they can make a request to the server only when they need data. Pull-based applications do not require a persistent connection and can be implemented using standard request-response HTTP protocols. However, pull-based applications may not be as efficient as push-based applications for real-time updates, as clients may need to wait for a server response before, they can retrieve data.

Overall, the choice between push and pull-based applications depends on the specific requirements of the application and the nature of the data being transmitted.

[CO3]

25. "HTTP is a client driven protocol."-Discuss about one strategy adopted to overcome this limitation, especially for smartphone-based clients.

One strategy adopted to overcome the client-driven limitation of HTTP, especially for smartphone-based clients, is the use of server push or HTTP/2 push. With server push, the server can proactively send resources to the client without the client having to make a request for each resource individually.

In HTTP/2, server push is implemented by allowing the server to send multiple resources in response to a single client request. For example, if a client requests a webpage, the server can send the HTML, CSS, and JavaScript resources for that page in a single response, rather than requiring the client to make individual requests for each resource. This can help to reduce the latency and improve the performance of the application, especially on slow or unreliable mobile networks.

Another advantage of server push is that it can reduce the number of round trips required between the client and server, which can help to improve the responsiveness of the application. By proactively pushing resources to the client, the server can minimize the amount of time the client spends waiting for resources to be downloaded, which can improve the overall user experience.

Overall, server push is a strategy that can help to overcome the client-driven limitation of HTTP by allowing the server to proactively push resources to the client. This can help to improve the performance and responsiveness of the application, especially on mobile networks where latency and reliability can be a challenge.

26. Discuss about the significance of MIME types for HTTP based applications. Give corresponding code snippets of at least one type.

MIME (Multipurpose Internet Mail Extensions) types are an important aspect of HTTP-based applications because they allow servers to indicate the type of content that is being transmitted in a message. This is important because it allows the client to know how to handle the content, such as rendering it in a web page or downloading it as a file.

MIME types are specified using a string that consists of a type and subtype separated by a slash (/) character. For example, the MIME type for HTML documents is "text/html", while the MIME type for JPEG images is "image/jpeg".

In HTTP, MIME types are typically included in the "Content-Type" header of a message. For example, the following code snippet shows how the server might specify the MIME type of an HTML document in a response:

```
HTTP/1.1 200 OK
Content-Type: text/html

<!DOCTYPE html>
<html>
<head>
  <title>Example HTML Page</title>
</head>
<body>
  <h1>Hello, world!</h1>
</body>
  </html>
```

```
HTTP/1.1 200 OK
Content-Type: application/javascript

function myFunction() {
  alert("Hello, world!");
}
```

27. **How is caching decided by the different versions of the HTTP protocol? What do you store in cache here?**

    HTTP caching allows clients to reuse previously obtained responses for subsequent requests, reducing network traffic and improving performance. In HTTP/1.0, caching is controlled by the "Expires" header, while HTTP/1.1 uses the "Cache-Control" header. Cached responses include the status code, headers, and content of the response. However, cached responses are only valid for a specific request, which is identified by the request method, URI, and headers. Caching is useful for frequently accessed resources that do not change frequently.

28. **How does HTTP and websocket relate in order to establish a connection? Give code snippets to explain the connection setup.**

    HTTP and WebSocket are related in that WebSocket connections begin with an HTTP-based handshake. Here's an example of how the connection setup works:

    1. The client sends an HTTP request to the server with an "Upgrade" header indicating the desire to use WebSockets:

    GET /chat HTTP/1.1
    Host: example.com
    Upgrade: websocket
    Connection: Upgrade
    Sec-WebSocket-Key: dGhlIHNhbXBsZSBub25jZQ==
    Sec-WebSocket-Version: 13

    2. The server responds with an HTTP response indicating that it is willing to switch to WebSockets:

    HTTP/1.1 101 Switching Protocols
    Upgrade: websocket
    Connection: Upgrade
    Sec-WebSocket-Accept: s3pPLMBiTxaQ9kYGzzhZRbK+xOo=

    3. The connection is now established and data can be exchanged using the WebSocket protocol.

    The "Sec-WebSocket-Key" and "Sec-WebSocket-Accept" headers are used for security purposes, as described in the WebSocket specification.

29. **State the motivation for using design patterns for developing web applications.**

    The main motivation for using design patterns in web application development is to provide proven solutions to common problems that arise during development. Design patterns provide a standard way of solving these problems, making it easier for developers to create high-quality, maintainable, and scalable code. Additionally, design patterns can help reduce the amount of code duplication and increase code reuse, resulting in a more efficient and effective development process. Finally, design patterns can make it easier for developers to communicate and collaborate on a project by providing a common vocabulary and shared understanding of best practices.

30. Given that a web application is developed using the Spring framework, give use cases corresponding to HTTP response codes 3XX, 4XX and 5XX.

HTTP response codes are used to indicate the status of a web request. In a web application developed using the Spring framework, some common use cases for different HTTP response codes are:

3XX (Redirection): This status code is typically used when the requested resource has been moved or renamed. Use cases could include redirecting a user to a new URL when a resource has been relocated or when a session has expired and the user needs to be redirected to the login page.

4XX (Client Error): This status code is used when there is an error on the client side. Use cases include returning a 404 status code when a resource is not found, returning a 403 status code when a user does not have permission to access a resource, or returning a 400 status code when the client's request is invalid.

5XX (Server Error): This status code is used when there is an error on the server side. Use cases include returning a 500 status code when there is an unexpected error in the application, or returning a 503 status code when the server is temporarily unavailable or overloaded.

31. Compare between websocket and HTTP2.0.

WebSocket and HTTP/2.0 are both protocols designed for efficient and low-latency communication between client and server, but they serve different purposes.

WebSocket is a bi-directional protocol that enables real-time communication between a client and server over a single, long-lived connection. It is commonly used for chat applications, online gaming, and other real-time applications.

HTTP/2.0, on the other hand, is an updated version of the HTTP protocol that improves performance by introducing features such as multiplexing, header compression, and server push. It is designed to address the limitations of HTTP/1.x, which relied on multiple connections to load resources and was subject to high latency.

In terms of use cases, WebSocket is well-suited for applications that require real-time, low-latency communication between client and server, while HTTP/2.0 is best suited for applications that require efficient delivery of static and dynamic content over a network.

32. State the relation among web application, www and the internet.

A web application is a software program that runs on a web server and allows users to interact with it using a web browser. The World Wide Web (WWW) is a global network of interconnected documents and resources, accessed via the internet using standardized protocols such as HTTP. The internet is a global network of computer networks that connects millions of devices and enables communication and information sharing.

In other words, a web application is a type of software that is accessed over the World Wide Web, which in turn is accessed via the internet. The internet is the underlying network infrastructure that enables communication and data transfer between devices connected to it, and the World Wide Web is a subset of that infrastructure that enables access to information and resources through standardized protocols such as HTTP.

33. Though HTTP is a stateless protocol, user behaviour can be tracked.

Yes, it is possible to track user behaviour even though HTTP is a stateless protocol. This is achieved through the use of various techniques, such as cookies, session management, and URL rewriting.

Cookies are small text files that are stored on the client-side and contain information such as the user's preferences or login credentials. By storing this information in a cookie, the server can recognize the user on subsequent requests and track their behaviour.

Session management is another technique that is commonly used to track user behaviour. A session is a logical connection between the client and the server, and it allows the server to maintain state information for a particular user. This information can include things like the user's login status, shopping cart contents, or preferences.

URL rewriting is a technique that involves appending session IDs or other information to the URL. This allows the server to track the user's behaviour by analysing the URLs in the request.

By using these techniques, web applications can effectively track user behaviour and provide a personalized experience for each user.

34. What are the differences between Microsoft smooth streaming and Apple HTTP Live streaming?

Microsoft Smooth Streaming and Apple HTTP Live Streaming (HLS) are both adaptive bitrate streaming protocols used for delivering multimedia content over the internet. However, there are several differences between the two:

Technology: Microsoft Smooth Streaming is based on the proprietary Microsoft Silverlight technology, whereas Apple HLS is based on open standards like HTTP, HTML5, and JavaScript.

Segmentation: In Smooth Streaming, the video is segmented into small files of fixed duration, while in HLS, the video is segmented into small chunks with a variable duration.

Protocol: Smooth Streaming uses its own proprietary protocol, while HLS uses HTTP protocol for data transfer.

Media Formats: Smooth Streaming supports only a limited number of media formats, while HLS supports a wide range of media formats including MPEG-2, H.264, and AAC.

Streaming Quality: Smooth Streaming has better quality for high-bitrate videos, but it requires more server resources. HLS, on the other hand, has lower quality for high-bitrate videos, but it requires fewer server resources.

Support: Smooth Streaming is supported only on Microsoft platforms, while HLS is supported on both iOS and Android platforms.

Overall, both protocols have their own advantages and disadvantages, and the choice between the two depends on the specific requirements of the application.

35. For an RTSP application, mention the different protocols associated with it and how many channels are established between the client and the server?

RTSP (Real-Time Streaming Protocol) is a network control protocol that is used to control the delivery of multimedia data in streaming applications. The different protocols associated with RTSP are TCP, UDP, and RTP.

When a client connects to the RTSP server, two channels are established between them - the control channel and the data channel. The control channel is used for sending RTSP commands and receiving responses, while the data channel is used for transmitting the multimedia data itself, usually via RTP. Some RTSP implementations also use a third channel for sending RTCP packets to monitor the quality of service.

36. UDP is not sufficient as a stand-alone transport protocol for multimedia streaming applications. -Give your justification.

UDP (User Datagram Protocol) is a lightweight transport protocol that provides low latency and low overhead data transmission. However, it does not provide reliability, congestion control, or flow control mechanisms, making it unsuitable for standalone multimedia streaming applications.

Reliability is a critical requirement for multimedia streaming because it ensures that the transmitted data is delivered correctly without loss or corruption. Congestion control and flow control are necessary to prevent network congestion and ensure that data is transmitted at a rate that the network can handle without dropping packets or causing delays.

Therefore, most multimedia streaming applications use a hybrid approach where UDP is used for the actual transmission of data, and a higher-level protocol, such as RTSP or HTTP, is used for control and signaling. This approach provides the necessary reliability, congestion control, and flow control mechanisms required for a robust multimedia streaming system.

37. Is there any relation between marker bit and payload type of an RTP header?

Yes, there is a relation between the marker bit and the payload type of an RTP header.

The marker bit (M-bit) is the last bit in the RTP header and is used to indicate the end of a frame or packet. When set to 1, it indicates that the current packet contains the last part of the frame.

The payload type field in the RTP header specifies the type of data that is carried in the payload of the packet. Different payload types are assigned to different types of data such as audio, video, and other application-specific data.

In some cases, the marker bit is used to indicate the start of a new frame, and the payload type is used to identify the type of data in that frame. For example, in H.264 video streams, the payload type 24 is used for STAP-A (single-time aggregation packet), and the marker bit is set to 1 in the first packet of the STAP-A to indicate the start of a new frame.

38. How is RTCP responsible for quality of experience as perceived by the user?

RTCP is responsible for measuring and reporting quality of experience (QoE) to participants in an RTP session. It provides feedback on network conditions, such as packet loss, delay, and jitter, which can affect the quality of the media stream. RTCP reports can be used to adjust the transmission parameters of the RTP stream, such as adjusting the bit rate or the packet size, in order to improve the QoE. By monitoring and adapting to network conditions, RTCP helps to ensure that the media stream is delivered with the best possible quality to the end user.

39. Differentiate between Http GET and POST requests.

The main differences between HTTP GET and POST requests are:

Parameters: GET requests include parameters in the URL or the query string, while POST requests include parameters in the request body.

Data type: GET requests can only send data in plain text format, while POST requests can send data in various formats like text, JSON, XML, etc.

Caching: GET requests are cacheable by default, while POST requests are not.

Security: GET requests are less secure compared to POST requests as the parameters are visible in the URL, while POST requests hide the parameters in the request body.

Idempotent: GET requests are idempotent, meaning that making the same request multiple times will have the same effect, while POST requests are not idempotent, and making the same request multiple times may have different effects.

40. What are the functionalities of a container?

In the context of multimedia applications, a container is a file format that is used to store and organize multimedia data, such as audio, video, and subtitles. The main functionalities of a container include:

Multiplexing: A container can hold multiple types of multimedia data in a single file, allowing them to be played back simultaneously.

Metadata management: Containers can store information about the multimedia content, such as the format, codec, duration, and other metadata.

Error handling: Containers provide a mechanism for error detection and correction, allowing for the recovery of corrupted data.

Compression: Some container formats support compression, which can reduce the size of the multimedia data without sacrificing quality.

Streaming: Containers are often used for streaming multimedia data over the internet, providing a way to send and receive multimedia content in real-time.

41. Differentiate between parameters and attributes.

In web development, "parameters" and "attributes" refer to different things depending on the context.

In general, "parameters" are values passed to a function or method, while "attributes" are properties of an object.

In the context of web development, "parameters" usually refer to values passed in the URL or as part of a form submission. They are used to specify additional information needed to process a request or to filter results. For example, in an HTTP GET request, parameters might be used to specify search criteria or sorting options.

On the other hand, "attributes" in web development usually refer to properties of an HTML element, such as the "class" or "id" attributes. These properties are used to apply styles or manipulate the element through JavaScript.

42. What is the relation between a web server and a web container?

A web server and a web container are two different components used in web application architecture. A web server is responsible for handling HTTP requests and responses, serving static resources such as HTML, CSS, and JavaScript files, and forwarding dynamic requests to the appropriate web container. On the other hand, a web container is responsible for managing the life cycle of servlets and JSP pages, managing and pooling resources such as database connections, and providing a runtime environment for web applications.

In other words, a web server is used to handle low-level HTTP communications, while a web container provides an execution environment for web applications, including servlets and JSP pages. A web server can host multiple web containers, and each web container can host multiple web applications. The web server and web container work together to deliver dynamic web content to clients.

43. How will you validate the data sent by the client for servlet-based web applications? Explain with an example.

In servlet-based web applications, data sent by the client can be validated using the javax.servlet.ServletRequest object. The ServletRequest object represents the client request and provides methods to access request parameters, headers, and other details.

For example, suppose we have a servlet that accepts a username and password as parameters from the client. We can validate the input using the following code snippet:

```
protected void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
   String username = request.getParameter("username");
   String password = request.getParameter("password");

   if (username == null || username.isEmpty() || password == null ||
password.isEmpty()) {
      // return an error response if either username or password is missing or empty
      response.sendError(HttpServletResponse.SC_BAD_REQUEST, "Username and
password are required");
      return;
   }

   // perform authentication and other operations as necessary
   // ...
}
```

In this example, we retrieve the username and password parameters from the request using the getParameter() method. We then check if either of them is null or empty, and if so, return an error response with an appropriate message. If both parameters are present and non-empty, we can perform further validation and authentication as necessary.

44. How to count the no of clients who are currently logged in?

To count the number of clients who are currently logged in to a web application, you can use a session tracking mechanism. In a servlet-based web application, you can create a session object for each client when they first access the web application.

You can use the HttpSession interface to create and manage sessions in servlet-based web applications. Each session has a unique session ID that is used to identify the client. You can store the session object in a collection, such as a HashMap, and increment a counter every time a new session is created.

the SessionCounterListener class implements the HttpSessionListener interface, which contains two methods that are called when a session is created or destroyed. The sessionCreated() method is called when a new session is created, and the sessionDestroyed() method is called when a session is destroyed.

the sessionCreated() method, the session object is stored in the sessionMap collection, and the activeSessions counter is incremented. In the sessionDestroyed() method, the session object is removed from the sessionMap collection, and the activeSessions counter is decremented.

To get the number of active sessions in your web application, you can call the getActiveSessions() method of the SessionCounterListener class, which returns the value of the activeSessions counter.

[CO5]

45. What is confidentiality?

Confidentiality is the property of data or information that ensures that it is not disclosed to unauthorized individuals or systems. It means that the data is protected from being accessed, viewed, or used by anyone who is not authorized to do so. Confidentiality is an important aspect of information security and is necessary to protect sensitive or classified data, such as personal information, financial data, trade secrets, or classified government information. There are various techniques used to ensure confidentiality, including encryption, access control, and secure communication protocols.

46. How could it be provided through the web container?

Confidentiality can be provided through the web container by enabling SSL/TLS (Secure Sockets Layer/Transport Layer Security) encryption. SSL/TLS provides secure communication between the client and server by encrypting the data transmitted between them. When SSL/TLS is enabled, the data is encrypted before it is sent over the network and decrypted at the receiver's end, ensuring that the data cannot be intercepted by a third party.

To enable SSL/TLS in a web container, you need to generate a digital certificate and configure the web container to use it. The digital certificate contains information about the website's identity, such as the website name and public key. The certificate is signed by a trusted Certificate Authority (CA), ensuring that the certificate is genuine.

In the context of Spring Framework, you can enable SSL/TLS by configuring the web container to use HTTPS protocol instead of HTTP protocol. Here is an example of how to configure SSL/TLS in Spring Boot:

```
server.port=8443
server.ssl.key-store-type=JKS
server.ssl.key-store=classpath:keystore.jks
server.ssl.key-store-password=changeit
 server.ssl.key-alias=tomcat
```

In the above example, the server.port property is set to 8443, which is the default port for HTTPS. The server.ssl.key-store-type property is set to JKS, which is the default keystore type used by Java. The server.ssl.key-store property is set to the path of the keystore file. The server.ssl.key-store-password property is set to the password for the keystore file. Finally, the server.ssl.key-alias property is set to the alias name for the key in the keystore file.

47. What is declarative security? is there any other way of securing web application?

Declarative security is a mechanism used in web applications to specify security constraints in a declarative manner, typically using XML or annotations. With declarative security, the security requirements are specified separately from the code, allowing for easier management of security policies.

In addition to declarative security, there is also programmatic security, which involves writing security code directly in the application. This approach can be more flexible but can also be more complex and harder to manage.

In the Spring framework, declarative security is provided by Spring Security, which allows developers to specify security constraints using annotations or XML configurations. This approach provides a flexible and powerful way to secure web applications.

48. Explain the role of 'http.csrf().disable();' w.r.t Spring security.

In Spring Security, the http.csrf().disable() method is used to disable Cross-Site Request Forgery (CSRF) protection.

CSRF is a type of attack where a malicious website can perform unwanted actions on behalf of an authenticated user. To prevent this, Spring Security includes CSRF protection by default, which involves generating a token on the server-side and including it in each form submission. The token is then checked on the server-side to ensure that the request was submitted from the same origin as the token.

However, in some cases, such as when using a stateless architecture, it may be unnecessary or impractical to use CSRF protection. In such cases, the http.csrf().disable() method can be used to disable the CSRF protection for the entire application.

It's important to note that disabling CSRF protection can make your application more vulnerable to CSRF attacks, so it should only be done if necessary and with careful consideration.

49. How are user session management and authentication related to each other? Discuss w.r.t. Spring framework.

In a web application, user session management and authentication are closely related to each other. A session is typically created for a user after they successfully authenticate themselves with the application. The session allows the user to remain authenticated across multiple requests without having to provide their credentials again.

In Spring Security, session management and authentication are handled by different components. The SessionManagementFilter is responsible for managing user sessions, while the AuthenticationFilter is responsible for authenticating users.

When a user logs in to the application, Spring Security creates an authentication object, which contains information about the user's identity and credentials. This authentication object is then stored in the user's session so that it can be retrieved on subsequent requests.

The SessionManagementFilter is responsible for enforcing session-related security policies, such as session timeouts, invalidating sessions, and preventing session fixation attacks. It also provides support for concurrent session control, which allows administrators to limit the number of concurrent sessions that a user can have.

In Spring Security, session management and authentication can be configured using the session-management and form-login elements in the security configuration file. For example:

```
<http>
  <session-management>
    <concurrency-control max-sessions="1" />
  </session-management>
  <form-login login-page="/login" />
</http>
```

This configuration limits the number of concurrent sessions that a user can have to 1 and specifies the login page to use for authentication.

Overall, user session management and authentication are essential components of web application security, and Spring Security provides a robust set of features for handling both aspects.

50. How are the key pairs kept for HTTPS services in the context of the Spring framework?

In the Spring framework, the key pairs for HTTPS services are typically kept in a Java KeyStore file. This file stores private keys and digital certificates for use in SSL and TLS protocols. The KeyStore file can be created using the keytool utility provided with the Java SDK. In Spring, the KeyStore file can be specified in the application configuration file and loaded at runtime using a Java SSLContext object. This SSLContext object can then be used to configure the server-side SSL/TLS settings for the HTTPS service, including the key pairs, certificate authorities, and cipher suites to be used.

51. How can injection attack happen through HTTP request-response?

Injection attacks happen when untrusted user input is not properly validated and is used to construct dynamic SQL queries, commands or code fragments. This can happen through HTTP request-response cycle in several ways such as:

SQL Injection: Attackers can craft malicious SQL queries and inject them into input fields that are not properly validated, allowing them to manipulate the database and potentially retrieve sensitive information.

Command Injection: Attackers can inject malicious commands into input fields that are not properly validated, allowing them to execute arbitrary commands on the server and potentially gain access to sensitive resources.

Cross-Site Scripting (XSS): Attackers can inject malicious scripts into input fields that are not properly validated, allowing them to steal sensitive data, such as user credentials, or take control of the user's session.

To prevent injection attacks, it is important to properly validate and sanitize all user input, and use parameterized queries or prepared statements to avoid directly embedding user input in SQL queries or other code fragments. Web application frameworks like Spring provide built-in protection against injection attacks by providing tools for input validation and sanitization.

52. How could Confidentiality be provided through the web container?

Confidentiality can be provided through the web container by enabling HTTPS protocol for secure communication between the client and the server. HTTPS (HTTP Secure) uses SSL/TLS encryption to ensure that the communication between the client and the server is secure and cannot be intercepted by a third party.

To enable HTTPS, a digital certificate must be obtained from a trusted certificate authority and installed on the web server. The web container can be configured to use HTTPS by modifying the web.xml configuration file and specifying the SSL connector properties, such as the keystore file path, password, and key alias.

Once HTTPS is enabled, all communication between the client and server will be encrypted, providing confidentiality for sensitive information such as passwords, credit card numbers, and personal information.

53. What is the difference between authentication and authorization? Are they handled separately in Spring? Explain with suitable code snippets.

Authentication and authorization are two important aspects of security in web applications. Authentication refers to the process of verifying the identity of a user, while authorization refers to the process of granting or denying access to resources based on the user's identity and privileges.

In Spring, authentication and authorization are handled separately. The Spring Security framework provides various mechanisms for implementing both authentication and authorization.

Here's an example of how to configure authentication using Spring Security:

```
@Configuration
@EnableWebSecurity
public class SecurityConfig extends WebSecurityConfigurerAdapter {

  @Override
  protected void configure(AuthenticationManagerBuilder auth) throws Exception {
    auth
      .inMemoryAuthentication()
        .withUser("user")
          .password("{noop}password")
          .roles("USER");
  }

  @Override
  protected void configure(HttpSecurity http) throws Exception {
    http
      .authorizeRequests()
        .anyRequest().authenticated()
        .and()
      .formLogin()
        .and()
      .httpBasic();
  }
}
```

In the above code snippet, we are configuring authentication using an in-memory user store. The withUser method is used to specify the username and password, while the roles method is used to specify the user's roles.

Here's an example of how to configure authorization using Spring Security:

```
@Configuration
@EnableWebSecurity
public class SecurityConfig extends WebSecurityConfigurerAdapter {

  @Override
  protected void configure(HttpSecurity http) throws Exception {
    http
      .authorizeRequests()
        .antMatchers("/admin/**").hasRole("ADMIN")
        .antMatchers("/user/**").hasRole("USER")
        .anyRequest().authenticated()
        .and()
      .formLogin()
        .and()
      .httpBasic();
  }
}
```

In the above code snippet, we are configuring authorization based on the user's roles. The hasRole method is used to specify the required role for accessing a particular URL pattern. In this example, users with the "ADMIN" role can access URLs starting with "/admin", while users with the "USER" role can access URLs starting with "/user". The authenticated method is used to specify that all authenticated users can access any other URL.