CS 6676-XTIA Advanced Computer Networks T3 2023

Final Exam

**1 Computer Networks**

1. One example of a centralized network system is a client-server architecture.

In this architecture, there is one central server that manages and coordinates all communication between clients and provides services to clients upon request.

The communication between a client and a server typically follows a request-response pattern. The client sends a request message to the server, which contains information about the service the client wants to access. The server receives the request, processes it, and sends back a response message to the client containing the requested information.

1. One example of a decentralized network system is a peer-to-peer (P2P) network. In a P2P network, all nodes (or peers) are equal and can act as both clients and servers, sharing and requesting resources from each other.

In a P2P network, communication between nodes (client/server) is typically initiated by a peer requesting a resource from another peer. The requesting peer sends a message to the other peer over the network, asking for the requested resource.

The communication between nodes in a P2P network is based on a distributed model where each node can act as both a client and a server.

1. In a distributed system, middleware is a layer of software that sits between the application and the operating system, providing services that enable communication and coordination between distributed components.

The role of middleware in a distributed system is to abstract the underlying complexity of the network and provide a simplified, standardized interface for communication and coordination between components.

Location transparency: Location transparency hides the physical location of resources from the user, allowing them to access resources without knowing where they are located.

* Scenario: A distributed file system where users can access files stored across multiple servers. With location transparency, users can access files without needing to know the physical location of the servers storing the files. They can simply access the file system as if all files were stored on a single server.

Replication transparency: Replication transparency allows multiple copies of a resource to be created and managed as a single entity, hiding the fact that multiple copies exist from the user.

* Scenario: A distributed database that stores customer data for an e-commerce site. With replication transparency, multiple copies of the database can be created across multiple servers to improve performance and availability. Users can access the database as if it were a single entity, without knowing that multiple copies exist.

Failure transparency: Failure transparency hides the fact that components of a distributed system can fail, allowing the system to continue operating despite failures.

* Scenario: A web application that is deployed across multiple servers. If one of the servers fails, the application can continue running on the remaining servers without any interruption or downtime. This is achieved through failure transparency, which hides the fact that the system is running on multiple servers and allows the system to continue operating despite failures.

**2 Software Architecture**

2.1

i. Client-Server Architecture: Components in this architecture communicate through a network, with the client and server running on separate machines. The client and server communicate using a standardized protocol, such as HTTP, TCP/IP, or RPC. The client sends a request message to the server, which contains information about the service or resource that the client wants to access. The server receives the request, processes it, and sends back a response message to the client containing the requested information.

ii. Peer-to-Peer Architecture: Components in this architecture communicate directly with each other over the network, without the need for a centralized server. The communication between nodes in a peer-to-peer network is based on a distributed model, where each node can act as both a client and a server. Nodes in the network communicate using a standardized protocol, such as BitTorrent or Gnutella. The requesting peer sends a message to the other peer requesting the resource, and the other peer responds by providing the requested resource directly to the requesting peer over the network.

2.2

i. Napster is an early peer-to-peer (P2P) file sharing application. It was created by Shawn Fanning and Sean Parker. Napster allowed users to share and download music files over the internet.

ii. Napster is different from both the centralized model and fully decentralized model.

In a centralized model, all communication is routed through a central server, which acts as a mediator between clients. In Napster, while there is a central server that maintains a directory of shared files and user information, the actual file sharing occurs between clients directly, without the need for the central server to mediate the transfer.

In a fully decentralized model, there is no central server or directory. All communication occurs between nodes in the network directly, without any central authority to manage or mediate the transfer of data. Napster is not a fully decentralized model, as it still requires a central server to maintain a directory of shared files and user information.

iii. The steps to successful communication in Napster are:

* A user connects to the central Napster server.
* The user sends a search query to the server, requesting files with specific keywords, such as artist name, album name, or song title.
* The Napster server responds with a list of files that match the user's search query, along with information about the user who is sharing the file and the file's location on their computer.
* The user selects a file from the list and sends a request to the user who is sharing the file, asking to download the file.
* If the user who is sharing the file accepts the request, they send the file to the user who requested it, using a direct connection between the two users.
* Once the file has been successfully transferred, the connection is terminated, and the users can continue using the service.

**3 Networking and internetworking**

3.1

Virtual Private Network (VPN). A VPN is a network technology that creates a secure and private connection between two or more devices over the internet. It allows users to access and transmit data over a public network as if they were directly connected to a private network.

In my daily life, I use a VPN to create a private tunnel that secures my data from the possibility of a “bad actor” ready to exploit publicly accessed Wi-Fi.

3.2

No, messages are typically not sent as a whole from the source to the destination in a network. Instead, they are broken down into smaller packets, which are then transmitted individually through the network and reassembled at the destination. When the packet fails to arrive at destination then it is known as packet loss.

Steps:

* The sender creates a message and sends it to the network interface, which adds the necessary header information to the message, such as the sender and receiver addresses.
* The message is then broken down into smaller packets, each packet is assigned a unique identifier, which is used to reassemble the message at the destination.
* The packets are then sent to the network layer, which adds additional header information.
* The packets are then transmitted over the network, with each packet potentially taking a different path to reach its destination.
* As the packets arrive at the destination, they are reassembled into the original message based on their unique identifiers.
* Once all the packets have been received and reassembled, the message is delivered to the destination application or user.

**4. Inter-process communication, Remote Invocation**

**4.1**

Characteristics of synchronous communication:

* Requires both the sender and receiver to be active and available at the same time
* Sender waits for the response from the receiver before continuing
* Communication is typically faster than asynchronous communication, as there is less overhead
* Suitable for applications that require real-time interaction and immediate responses

Characteristics of asynchronous communication:

* Does not require both the sender and receiver to be active and available at the same time
* Sender does not wait for a response from the receiver before continuing
* Communication is typically slower than synchronous communication, as there is more overhead
* Suitable for applications that do not require real-time interaction and can handle delayed responses

The main difference between synchronous and asynchronous communication is the way in which they handle timing and availability. Synchronous communication requires both parties to be present and active at the same time, while asynchronous communication allows for one party to be offline or unresponsive.

1. Blocking in synchronous communication refers to the fact that the sender must wait for a response from the receiver before continuing. This can lead to a delay in the communication process if the receiver is busy or unavailable. While, non-blocking in asynchronous communication refers to the fact that the sender can continue with other tasks while waiting for a response from the receiver. This allows for more efficient use of resources and can reduce delays in the communication process.

**4.2**

1. Steps to perform communication between a client and a server using TCP stream communication:

* The client initiates the communication by sending a SYN (synchronize) message to the server.
* The server responds with a SYN-ACK (synchronize-acknowledgment) message to confirm that the communication is being established.
* The client sends an ACK (acknowledgment) message back to the server to confirm the connection.
* Data is then exchanged between the client and server using the established TCP connection.
* Each data packet sent by one party is acknowledged by the other party using ACK messages to ensure that the data is received correctly.
* Once the communication is complete, either the client or server can send a FIN (finish) message to indicate that the connection is being terminated.
* The other party responds with an ACK message to confirm the termination, and the connection is closed.

1. The role of the ACK (acknowledgment) message in TCP communication is to confirm that data has been received correctly. When one party sends a data packet to the other party, the receiving party sends an ACK message back to indicate that the data has been received and is error-free.

**4.3**

The role of the client stub and server stub is to provide an abstraction layer between the client and server applications, shielding them from the complexities of network communication. This allows the client and server applications to communicate with each other in a standardized manner, regardless of the underlying network protocol or hardware. The stubs also handle issues such as error detection and recovery, and can provide additional services such as security and authentication.

**4.4**

The overwhelming messages created by broadcasting in the network is called broadcast storm. A Broadcast Storm occurs when a significant number of broadcast packets are transmitted in a short period of time. The network quality suffers dramatically as a result of the broadcast storm. It causes a computer network's broadcast and multicast traffic to build up.

Steps to reduce Broadcast storm

* Antivirus Firewalls can be used to detect and eliminate malicious and purposefully created broadcast storms that cause network disruption.
* The more frequently ARP tables are cleansed, the more broadcast requests are generated.
* The broadcast packets can be limited using storm protocols and similar management approaches.
* On Layer 3 devices, turn off broadcasts. If a storm originates from the WAN, the problem can be resolved by turning off IP directed broadcasts.
* Diverting broadcast traffic can be made easier by splitting the broadcast domain. We can redirect more than half of the traffic to the other network by constructing a new VLAN network.
* It will significantly reduce the number of storms.
* Switches should be maintained on a regular basis to avoid hardware failure.
* Upstreaming the unmanaged switch is made easier by looking for loops in switches, In order to respond to broadcasts, unmanaged switches sometimes overload the network with needless traffic.

**5. Name services**

**5.1**

**Top domain:**

This is the main suffix that is used at the end of the name of the domain. For example, .edu, .com, .org, etc. Top domains can also be chosen from almost 1000 domains. These can be related to countries or regions. For example for the UK, it would be .uk.

Root domain:

This is a combination of a top domain and a unique domain. It is used to completely form a website address. It is taken as the highest page in the hierarchy. Subdomains or other pages emerge from this domain. For example, abc.net.

Subdomain:

This is the last level. It is added after the root domain. It is separated using a period. For example, def.abc.net. 'www' is the most common subdomain.

**5.2**

1. The home agent is used in Mobile IP. It is used in the form of a router in the home network. It is used to keep data related to the current location of the device. It serves the role of care-of-address**.**

2. When a node moves to a different network, the data will be sent by the sender using the home address of the node. This will be a normal routine activity.

If the node is moved to a foreign network, then the home agent will be sent the datagram and it will forward it to the foreign agent. The datagram is given to the respective node via a foreign agent then

**6. OmNeT++**

**6.1**

The initialize() method is used to set up initial state when it is created

initialize() is called at the beginning of the simulation, to bootstrap the simulation process.

It sends an initial message from one of the modules.

handleMessage() method is called whenever a message arrive at the module.

It is used to define how the response to incoming messages or events.

**6.2**

Forwarding function in this network, can be conducted by:

* Setting each node to have a unique identifier, to facilitate routing of messages between them. For example, we can assign the IDs 0 to 4 to the nodes.
* Node 0 should be designated as the central hub or server that will receive messages from the client nodes and forward them to the appropriate destination node.
* Each client node (1, 2, 3, and 4) will have a list of the other client nodes it can directly communicate with.
* When a client node wants to send a message to another client node, it should first send the message to the central hub (Node 0). The message should include the destination node ID
* When Node 0 receives the message, it should examine the destination ID and forward the message to the appropriate node in its list of connected client nodes. If the destination node is not in the list, Node 0 should send an error message back to the sender.
* When the destination node receives the message, it should process and generate a response message, if necessary.
* The response message should be sent back to Node 0, along with the original sender's ID and a response code indicating whether the message was successfully processed or not.
* Node 0 should forward the response message back to the original sender, using the sender's ID to determine the appropriate destination.
* If any errors occur during message forwarding or processing, appropriate error messages should be sent back to the original sender.