Abubakar Munirat Mikail

VUG/CSC/22/8526

Cyber security

23rd April, 2024.

1. **Components Of Socket Programming**

* **IP Address**: Every device connected to a network has a unique IP address, which is used to identify it on the network.
* **Port**: Ports are used to differentiate between different services running on the same device. A port is a number that is associated with a specific process or service. For example, web servers typically listen on port 80 for HTTP requests, while email servers listen on port 25 for SMTP (Simple Mail Transfer Protocol) traffic.
* **Server Socket**: A server socket is a combination of an IP address and a port number that the server listens on for incoming connections. When a client wants to communicate with the server, it initiates a connection to the server's IP address and port number.
* **Client Socket**: A client socket is a combination of the client's IP address and a randomly assigned port number used to communicate with the server. When a client wants to connect to a server, it establishes a connection to the server's IP address and port number.
* **Communication Protocol**: Socket programming can use different communication protocols such as TCP (Transmission Control Protocol) or UDP (User Datagram Protocol). TCP provides reliable, connection-oriented communication, while UDP is faster but less reliable and connectionless.

**Steps involved in Socket Programming**

**Creating a Socket**: Both the client and server create sockets using appropriate functions provided by the programming language or library being used.

**Binding**: The server binds its socket to a specific IP address and port number, so that it can listen for incoming connections.

**Listening**: The server socket enters a listening state, waiting for incoming connection requests from clients.

**Accepting Connections**: When a client initiates a connection request, the server accepts the connection, creating a new socket dedicated to that client.

**Data Transfer**: Once the connection is established, both the client and server can send and receive data through their respective sockets.

**Closing Connections**: After the communication is complete, both the client and server close their sockets to release the resources.

1. **15 Commonly Used Network Ports**

Transport Protocols: The Digital Highways

TCP: The Reliable Courier

UDP: The Speedy Messenger

Common TCP and UDP Default Ports

1. 21 (TCP, UDP) – FTP
2. 22 (TCP, UDP) – SSH
3. 23 (TCP) – Telnet
4. 25 (TCP) – SMTP
5. 53 (TCP, UDP) – DNS
6. 67,68 (UDP) – DHCP
7. 80 (TCP) – HTTP
8. 110 (TCP) – POP3
9. 111 (TCP, UDP) – Portmapper
10. 137 (TCP, UDP) – NetBIOS
11. 143 (TCP, UDP) – IMAP
12. 161, 162 (UDP) – SNMP
13. 443 (TCP) – HTTPS
14. 587 (TCP) – SMTP
15. 993 (TCP) – IMAPS
16. **How a Port can be Left Open**
17. Server Application Running: A port is typically opened by a server application or service that is running on a device. For example, a web server might open port 80 to listen for HTTP requests, or an SSH server might open port 22 to allow secure shell connections.
18. Binding to the Port: When a server application starts, it binds its socket to a specific port number, effectively reserving that port for its use. This tells the operating system that any incoming network traffic on that port should be directed to the server application.
19. Listening for Connections: Once the server application has bound its socket to a port, it enters a listening state, waiting for incoming connection requests. This allows clients to establish connections with the server by sending network packets to the server's IP address and port number.
20. Accepting Connections: When a client initiates a connection request to the server's port, the server accepts the connection and creates a new socket dedicated to that client. The server can then communicate with the client over this connection.
21. Closing the Port: A port is considered open as long as the server application is actively listening for connections. If the server application is stopped or the port is closed explicitly, the port is considered closed and no further connections can be accepted on that port.
22. Different Protocols used in the 7 OSI Model of a Computer Network.

**Physical Layer (Layer 1):**

The physical layer deals with the transmission and reception of raw data bits over a physical medium, such as cables or wireless signals.

Protocols: Ethernet, Wi-Fi (IEEE 802.11), Fiber Channel, DSL, Bluetooth.

**Data Link Layer (Layer 2):**

The data link layer provides error detection and correction, as well as flow control, to ensure reliable data transmission between directly connected devices.

Protocols: Ethernet (IEEE 802.3), Wi-Fi (IEEE 802.11), PPP (Point-to-Point Protocol), HDLC (High-Level Data Link Control).

**Network Layer (Layer 3):**

The network layer is responsible for addressing, routing, and forwarding data packets between different networks.

Protocols: IP (Internet Protocol), ICMP (Internet Control Message Protocol), ARP (Address Resolution Protocol), OSPF (Open Shortest Path First), BGP (Border Gateway Protocol).

**Transport Layer (Layer 4):**

The transport layer ensures reliable and efficient data transfer between end systems. It provides mechanisms for error recovery, flow control, and segmentation of data.

Protocols: TCP (Transmission Control Protocol), UDP (User Datagram Protocol), SCTP (Stream Control Transmission Protocol).

**Session Layer (Layer 5):**

The session layer establishes, manages, and terminates connections (sessions) between applications on different devices. It also provides mechanisms for synchronization and checkpointing during data exchange.

Protocols: Not commonly associated with specific protocols; functions may be implemented within application layer protocols.

**Presentation Layer (Layer 6):**

The presentation layer is responsible for data formatting, encryption, and compression to ensure that data sent by one application can be understood by another application.

Protocols: SSL/TLS (Secure Sockets Layer/Transport Layer Security), MIME (Multipurpose Internet Mail Extensions).

**Application Layer (Layer 7):**

The application layer provides network services directly to end-users and applications. It includes protocols for tasks such as email, file transfer, web browsing, and remote access.

Protocols: HTTP (Hypertext Transfer Protocol), FTP (File Transfer Protocol), SMTP (Simple Mail Transfer Protocol), DNS (Domain Name System), SSH (Secure Shell), SNMP (Simple Network Management Protocol).