Network Troubleshooting and Security

Valiant Voora

Center of Excellence in Digital Forensics

CoEDF

E.R. Ramesh, M.C.A., M.Sc., M.B.A.,

Dr. M.G.R. Educational and Research Institute

University Decl. U/S 3 of UGC Act 1956

STRIVE TO EXCEL

Unit - 4

TCP/IP Application

Origins of TCP/ IP and evolution of Internet
IP Layers Vs OSI
IP number concepts
Network address
Classes of Networks
Subnet masking
Static and dynamic IP numbers
UDP
Establishing a TCP session (Three-way handshake)
Troubleshooting Physical Connectivity Problems
Name to address translation - Domain Name System

Unit - 4

TCP/IP Application

Transport layer protocols – TCP, UDP, ICMP, IGMP

The power of port numbers - registered ports

Connection status

Rules for determining good vs. bad communications

Common TCP/IP applications

- The world wide web
- Telnet
- Email
- FTP
- Internet applications

Origins of TCP/ IP and evolution of Internet

- The origins of the Transmission Control Protocol/Internet Protocol (TCP/IP) and the evolution of the Internet are deeply intertwined with the development of computer networks. Here's a brief overview:

Early Computer Networks (1960s):

1. The concept of computer networking began in the 1960s with the development of early experimental networks. One of the pioneering networks was the ARPANET (Advanced Research Projects Agency Network), funded by the United States Department of Defense's Advanced Research Projects Agency (ARPA). ARPANET, established in 1969, was a decentralized network designed to withstand partial outages (resilient to nuclear attacks).

Origins of TCP/ IP and evolution of Internet

Birth of TCP/IP (1970s):

1. In the early 1970s, Vinton Cerf and Bob Kahn developed the fundamental protocols that laid the foundation for the Internet. They introduced the Transmission Control Program (TCP), which provided reliable, end-to-end communication over the ARPANET.

TCP/IP Protocol Suite (1970s-1980s):

1. TCP was later split into two separate protocols: Transmission Control Protocol (TCP) for reliable data transmission and Internet Protocol (IP) for addressing and routing. This combination became known as the TCP/IP protocol suite. The specifications for TCP/IP were published in a series of documents known as the "Request for Comments" (RFCs).

Origins of TCP/ IP and evolution of Internet

Expansion Beyond ARPANET (1980s):

1. The TCP/IP protocol suite gained widespread adoption and was implemented on various networks. The National Science Foundation (NSF) played a crucial role in the 1980s by establishing the NSFNET, a high-speed backbone network that connected regional academic networks.

Commercialization and World Wide Web (1990s):

1. In the 1990s, the Internet transitioned from a government and academic research network to a commercialized and globally accessible infrastructure. Tim Berners-Lee's development of the World Wide Web in 1989 and the subsequent introduction of web browsers in the early 1990s played a pivotal role in making the Internet accessible to the general public.

Origins of TCP/ IP and evolution of Internet

Global Expansion (1990s-2000s):

1. The Internet continued to grow rapidly, connecting people and businesses worldwide. The development of technologies like broadband internet, improved network infrastructure, and the proliferation of internet service providers (ISPs) contributed to the expansion.

Mobile Internet and Broadband (2000s-Present):

1. The 2000s saw the rise of mobile internet with the proliferation of smartphones and mobile devices. Broadband technologies, such as DSL and cable, became more widespread, enhancing internet speeds and accessibility.

Origins of TCP/ IP and evolution of Internet

Emergence of New Technologies (2010s-Present):

1. The Internet of Things (IoT), cloud computing, and the deployment of high-speed networks like 5G have characterized the recent evolution of the Internet. These technologies have further transformed the way people and devices connect and interact.

- The development and evolution of the Internet have been driven by collaborative efforts, technological advancements, and the increasing demand for connectivity and information exchange on a global scale. The TCP/IP protocol suite remains the fundamental framework for communication on the Internet.

TCP/IP layers vs OSI

- The TCP/IP model and the OSI (Open Systems Interconnection) model are both conceptual frameworks used to understand and describe network protocols and communication processes. Each model consists of layers, each serving a specific purpose in the communication process. While they share similarities, they are not identical. Here's a comparison of the TCP/IP layers and the OSI layers:

TCP/IP Model:

1. **Link Layer (Network Interface Layer):**
    1. The TCP/IP model combines the OSI's Data Link and Physical layers into the Link Layer. It deals with the physical connection to the network and the low-level details of transmitting raw bits over a physical medium.

TCP/IP layers vs OSI

2. **Internet Layer:**
    1. Equivalent to the OSI Network Layer, the Internet Layer is responsible for logical addressing, routing, and fragmentation and reassembly of packets. The Internet Protocol (IP) operates at this layer.
3. **Transport Layer:**
    1. Similar to the OSI Transport Layer, it provides end-to-end communication, ensuring data integrity, sequencing, and flow control. Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) operate at this layer.
4. **Application Layer:**
    1. Combines elements of the OSI Session, Presentation, and Application layers. It encompasses the functionalities related to end-user services, such as file transfer, email, and remote login. Protocols like HTTP, SMTP, and FTP operate at this layer.

TCP/IP layers vs OSI

OSI Model:

1. **Physical Layer:**
   1. Concerned with the physical connection between devices, including hardware specifications such as cables and connectors.
2. **Data Link Layer:**
   1. Responsible for framing, addressing, and error detection within the data link. It includes the Logical Link Control (LLC) and Media Access Control (MAC) sub-layers.
3. **Network Layer:**
   1. Handles logical addressing, routing, and packet forwarding. The Internet Protocol (IP) is an example of a network layer protocol.
4. **Transport Layer:**
   1. Ensures end-to-end communication, providing error recovery, flow control, and retransmission of lost data. TCP and UDP operate at this layer.

TCP/IP layers vs OSI

5. **Session Layer:**
    1. Establishes, maintains, and terminates sessions or connections between applications. It manages dialog control and synchronization.
6. **Presentation Layer:**
    1. Deals with data format translation, encryption, and compression. It ensures that data sent from the application layer of one system can be properly read by the application layer of another.
7. **Application Layer:**
    1. Provides a network interface for end-user applications. It includes network-aware applications and application-level protocols like HTTP, FTP, and SMTP.

TCP/IP layers vs OSI

- While the OSI model has seven layers and the TCP/IP model has four, both serve as useful frameworks for understanding and designing network architectures.

- The TCP/IP model is more widely used in practice, especially on the Internet, while the OSI model is often used as a theoretical reference model for conceptual clarity.

IP number concepts

IP Addresses:

1. An IP address is a numerical label assigned to each device participating in a computer network that uses the Internet Protocol for communication. IPv4 addresses are written as four sets of decimal numbers separated by periods (e.g., 192.168.0.1), while IPv6 addresses are more complex and expressed as a series of hexadecimal numbers separated by colons.

IPv4 vs. IPv6:

1. IPv4 is the older version of the Internet Protocol and uses 32-bit addresses, limiting the number of unique addresses available. IPv6 was developed to address the exhaustion of IPv4 addresses by using 128-bit addresses, providing a vastly larger address space.

IP number concepts

Reserved IP Addresses:

1. Within both IPv4 and IPv6 address spaces, certain address ranges are reserved for specific purposes. For example, addresses in the range 192.168.0.0 to 192.168.255.255 are reserved for private networks in IPv4. IPv6 also has reserved address ranges.

Subnetting:

1. Subnetting involves dividing an IP network into sub-networks to improve performance and security. It allows organizations to create logical divisions within their networks. Subnet masks are used to determine which part of an IP address is the network and which part is the host.

IP number concepts

CIDR (Classless Inter-Domain Routing):

1. CIDR is a method for allocating IP addresses and IP routing. It allows for a more flexible allocation of IP addresses than the older class-based addressing methods.

Default Gateways:

1. A default gateway is the IP address of the router that a device uses to send data to a destination outside of its local network. It plays a crucial role in routing packets beyond the local subnet.

Classes of Network

- IP (Internet Protocol) networks can be classified based on the size, scope, and purpose of the network. The two main classes of IP networks are:

Classful IP Addressing:

- In the early days of IP networking, IP addresses were divided into three main classes: Class A, Class B, and Class C. Each class had a fixed portion for network and host addresses. Class D and Class E were reserved for multicast and experimental purposes, respectively. The classful system is largely obsolete, but it's important to understand its historical context:

Classes of Network

1. **Class A:**
   1. Range: 1.0.0.0 to 126.0.0.0
   2. Default Subnet Mask: 255.0.0.0
   3. Example: 10.0.0.0
2. **Class B:**
   1. Range: 128.0.0.0 to 191.255.0.0
   2. Default Subnet Mask: 255.255.0.0
   3. Example: 172.16.0.0
3. **Class C:**
   1. Range: 192.0.0.0 to 223.255.255.0
   2. Default Subnet Mask: 255.255.255.0
   3. Example: 192.168.0.0

Classes of Network

Classless IP Addressing (CIDR - Classless Inter-Domain Routing):

1. CIDR is a more flexible addressing scheme that allows for variable-length subnet masks, breaking away from the rigid class boundaries. IP addresses are expressed with a prefix length, denoted by the number of bits used for the network portion.
2. **CIDR Notation Examples:**
   1. 192.168.1.0/24 (Equivalent to a Class C address with a subnet mask of 255.255.255.0)
   2. 10.0.0.0/8 (Equivalent to a Class A address with a subnet mask of 255.0.0.0)
3. **CIDR** allows for more efficient use of IP addresses and is a fundamental part of modern IP addressing.

Classes of Network

Public and Private IP Addresses:

1. IP addresses are also classified as public or private based on their usability on the public internet:
   1. **Public IP Addresses:** Routable on the global internet. They are unique and globally accessible.
   2. **Private IP Addresses:** Reserved for use within private networks and are not routable on the public internet. Examples include the IP ranges reserved for local networks (e.g., 192.168.0.0/16, 172.16.0.0/12, 10.0.0.0/8).

Subnet masking

- Subnet masking is a technique used in IP networking to divide an IP address into two parts: the network address and the host address.

- It involves creating subnetworks or subnets within a larger network, allowing for better organization of IP addresses and efficient use of address space.

- Subnetting is often implemented using a subnet mask, which is a 32-bit number that specifies the network and host portions of an IP address.

Subnet masking

Here are the key concepts related to subnet masking:

Subnet Mask Format:

- A subnet mask is a 32-bit binary number expressed in dotted-decimal format, just like an IP address. The subnet mask consists of consecutive '1' bits followed by consecutive '0' bits. The '1' bits represent the network portion, and the '0' bits represent the host portion.
- Example: 255.255.255.0 (in binary: 11111111.11111111.11111111.00000000)

Subnet masking

Network Portion vs. Host Portion:

1. In an IP address, the network portion identifies the network itself, while the host portion identifies individual devices within that network. The subnet mask helps differentiate between these two parts.
2. Example:
   1. IP Address: 192.168.1.100
   2. Subnet Mask: 255.255.255.0
   3. Network Portion: 192.168.1.0
   4. Host Portion: 0.0.0.100

Subnet masking

CIDR Notation:

1. CIDR (Classless Inter-Domain Routing) notation is commonly used for specifying subnet masks. In CIDR notation, the number of '1' bits in the subnet mask is explicitly stated, followed by a forward slash and the prefix length.

2. Example:
    1. CIDR Notation: 192.168.1.0/24

Subnet masking

Subnetting Process:

1. To subnet a network, you borrow bits from the host portion of the IP address to create smaller subnets.
2. The number of borrowed bits determines the number of subnets and hosts per subnet.
3. For example, borrowing 3 bits allows for 8 subnets (2^3) with 2^5 - 2 hosts (32 - 2) in each.

Subnet masking

Subnet Masking Examples:

Consider a Class C IP address, 192.168.1.0, with a default subnet mask of 255.255.255.0 (or /24 in CIDR notation). If you decide to subnet further, you might use a subnet mask of 255.255.255.128 (or /25) to create two subnets, each with 126 usable host addresses.

Subnet 1:

1. Network Address: 192.168.1.0
2. Usable Host Range: 192.168.1.1 to 192.168.1.126
3. Broadcast Address: 192.168.1.127

Subnet masking

Subnet 2:

1. Network Address: 192.168.1.128
2. Usable Host Range: 192.168.1.129 to 192.168.1.254
3. Broadcast Address: 192.168.1.255

- Subnet masking allows network administrators to create smaller, more manageable networks within a larger address space.

- It is a crucial aspect of IP addressing and routing, contributing to the efficient use of IP addresses and improved network organization.

Static and Dynamic IP numbers

- Static and dynamic IP addresses refer to how IP addresses are assigned to devices on a network. Here's an overview of each:

1. Static IP Addresses:

A static IP address is manually assigned to a device and remains constant over time. It doesn't change unless manually reconfigured.

Static IP addresses are often used for devices that need a consistent and unchanging address, such as servers, routers, and network printers.

Static and Dynamic IP numbers

Characteristics:

1. **Permanence:** The IP address assigned to a device remains the same unless changed manually.
2. **Predictability:** It is known and can be easily identified in the network.
3. **Configuration:** Configured manually on the device or through a DHCP reservation.

Use Cases:

1. Servers: Web servers, email servers, DNS servers.
2. Network Devices: Routers, switches, printers.
3. Devices that require remote access.

Static and Dynamic IP numbers

Pros:

1. Predictable addressing for certain devices.
2. Easier to manage for specific applications or services.

Cons:

1. Requires manual configuration.
2. May lead to IP address conflicts if not managed properly.
3. Less flexible for devices that move between networks.

Static and Dynamic IP numbers

Dynamic IP Addresses:

A dynamic IP address is assigned to a device by a DHCP (Dynamic Host Configuration Protocol) server automatically. The assignment is temporary and may change over time. Dynamic addressing is more common in residential and small business networks.

Characteristics:

1. **Temporary:** The IP address is leased for a specific period and may change during renewal.
2. **Automatic:** Assigned by a DHCP server without manual configuration.
3. **Scalability:** Easily accommodates a large number of devices.

Static and Dynamic IP numbers

Use Cases:

1. Desktop computers, laptops, smartphones, and other devices in home or office networks.
2. Environments with a large number of devices that connect and disconnect frequently.

Pros:

1. Simplifies network management, especially in large networks.
2. Efficient use of IP addresses.
3. Supports mobile and transient devices.

Static and Dynamic IP numbers

Cons:

1. Address may change, making it harder to locate a specific device.
2. Not suitable for devices requiring a consistent, unchanging address.

Hybrid Approach:

- In some cases, a hybrid approach is used. Critical devices (e.g., servers) may have static IP addresses, while other devices receive dynamic IP addresses. This combines the predictability of static addressing with the scalability and ease of management of dynamic addressing.

Static and Dynamic IP numbers

Conclusion:

- The choice between static and dynamic IP addressing depends on the specific requirements of the network and the devices connected to it.

- Both approaches have their advantages and disadvantages, and the selection is often based on factors like network size, device mobility, and the need for consistent addressing.

UDP

- The User Datagram Protocol (UDP) is one of the core protocols of the Internet Protocol (IP) suite. It is a transport layer protocol that provides a simple, connectionless, and unreliable method of communication between devices on a network.

- UDP is often contrasted with the Transmission Control Protocol (TCP), another transport layer protocol that provides a reliable, connection-oriented communication.

- Here are key characteristics and features of UDP:

UDP

1. **Connectionless Protocol:**
   1. UDP is connectionless, meaning it does not establish a dedicated connection before sending data. Each UDP datagram is independent of the others.

2. **Unreliable Delivery:**
   1. Unlike TCP, UDP does not guarantee the delivery of packets. It does not use mechanisms such as acknowledgments, retransmissions, or flow control. Therefore, it is considered unreliable.

UDP

3. **Low Overhead:**

   1. Because UDP lacks the features for reliability, it has a lower overhead compared to TCP. This makes it faster and more suitable for applications where low latency is critical.

4. **No Handshake:**

   1. UDP does not perform a three-way handshake (like TCP) before transmitting data. This lack of handshake reduces the latency but also means that the sender doesn't know if the receiver is ready or capable of receiving the data.

UDP

5. **Simple Header Structure:**
    1. The UDP header is relatively simple, consisting of a source port, destination port, length, and checksum. The simplicity contributes to its efficiency.

6. **Datagrams:**
    1. UDP messages are called datagrams. Each datagram is an independent entity and may arrive out of order, duplicate, or not at all. Applications using UDP must manage these issues if required.

UDP

7. Use Cases:

1. Real-time applications: UDP is often used in real-time applications where low latency is critical, such as online gaming, video streaming, and VoIP (Voice over Internet Protocol).

2. Broadcast or multicast applications: UDP is suitable for scenarios where data needs to be sent to multiple recipients simultaneously.

UDP

8. **Examples of UDP-Based Protocols:**

   1. **DNS (Domain Name System):** DNS queries and responses are typically carried over UDP.
   2. **DHCP (Dynamic Host Configuration Protocol):** DHCP uses UDP for leasing and managing IP addresses in a network.
   3. **SNMP (Simple Network Management Protocol):** SNMP uses UDP for its transport.

UDP

9. **No Flow Control:**
   1. Unlike TCP, UDP does not provide flow control mechanisms. If a sender is transmitting data too fast for the receiver to process, there is no automatic way to slow down the sender.

- While UDP is not suitable for all types of applications, it is valuable in scenarios where low latency and simplicity are prioritized over guaranteed delivery.

- Developers often choose UDP for applications where occasional packet loss is acceptable, and retransmission or acknowledgment mechanisms would introduce too much overhead.

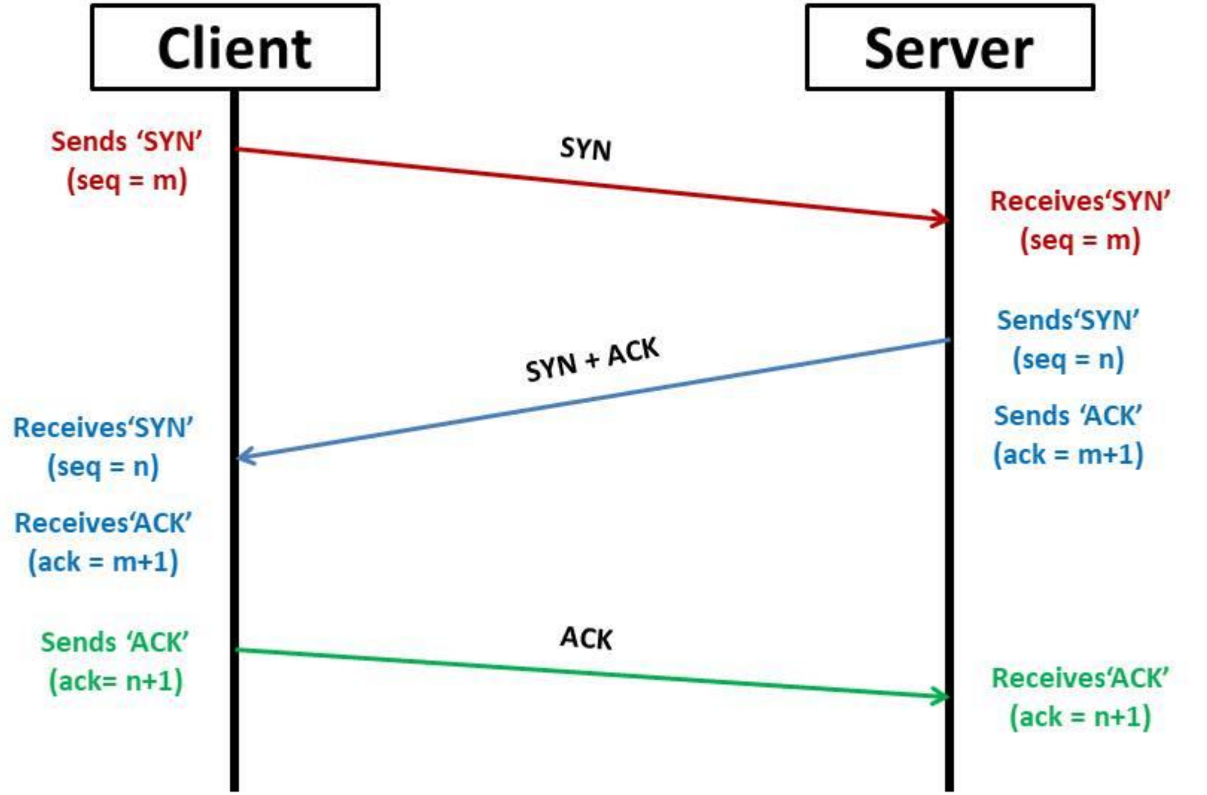
Establishing a TCP session (Three-way handshake)

- The three-way handshake is a fundamental process in the establishment of a TCP (Transmission Control Protocol) session between two devices.

- It is a method for initiating a reliable and connection-oriented communication channel. The three-way handshake involves three steps, where the client and server exchange specific packets to establish a connection.

- Here's an overview of each step:

Establishing a TCP session (Three-way handshake)



3-Way Handshaking(for establishing connection)

Establishing a TCP session (Three-way handshake)

1. **Step 1: SYN (Synchronize) from Client to Server:**
    1. The client initiates the connection by sending a TCP segment with the SYN flag set to the server. This initial segment contains the client's initial sequence number (ISN), which is a randomly generated number used to track the sequence of data.
    2. Format of the TCP segment from the client:
        1. Source Port: [Client's Port]
        2. Destination Port: [Server's Port]
        3. Sequence Number: [Client's ISN]
        4. Flags: [SYN = 1, ACK = 0]
        5. Other TCP header fields and options

Establishing a TCP session (Three-way handshake)

2. **Step 2: SYN-ACK (Synchronize-Acknowledge) from Server to Client:**
   1. Upon receiving the SYN segment from the client, the server responds with a TCP segment that has both the SYN and ACK flags set. The acknowledgment number (ACK) field in this segment is set to the client's ISN plus 1, acknowledging receipt of the client's SYN.
   2. Format of the TCP segment from the server:
      1. Source Port: [Server's Port]
      2. Destination Port: [Client's Port]
      3. Sequence Number: [Server's ISN]
      4. Acknowledgment Number: [Client's ISN + 1]

Establishing a TCP session (Three-way handshake)

5. Flags: [SYN = 1, ACK = 1]
6. Other TCP header fields and options

2. **Step 3: ACK (Acknowledge) from Client to Server:**

   1. In response to the server's SYN-ACK, the client sends an acknowledgment segment back to the server. The ACK flag is set, and the acknowledgment number field is set to the server's ISN plus 1, acknowledging receipt of the server's SYN.

Establishing a TCP session (Three-way handshake)

2. Format of the TCP segment from the client:
   1. Source Port: [Client's Port]
   2. Destination Port: [Server's Port]
   3. Sequence Number: [Client's ISN + 1]
   4. Acknowledgment Number: [Server's ISN + 1]
   5. Flags: [SYN = 0, ACK = 1]
   6. Other TCP header fields and options

Establishing a TCP session (Three-way handshake)

- Once this three-way handshake is complete, the TCP session is established, and both the client and server can begin exchanging data in a reliable manner.

- The use of sequence numbers and acknowledgments helps ensure that data is transmitted and received in the correct order, and any lost or corrupted packets can be identified and retransmitted if necessary.

Troubleshooting Physical Connectivity Problems

- Troubleshooting physical connectivity problems in computer networks involves identifying and resolving issues related to the physical components of the network infrastructure.

- Here are steps you can take to troubleshoot physical connectivity problems:

1. **Check Power and Lights:**
    1. Ensure that all network devices, including routers, switches, and modems, are powered on. Check for indicator lights on these devices to confirm power and proper operation.

Troubleshooting Physical Connectivity Problems

Troubleshooting Physical Connectivity Problems

2. **Inspect Cables:**

   1. Examine Ethernet cables for signs of damage, such as cuts, bends, or frayed ends. Make sure cables are securely connected to devices and that connectors are properly seated in ports.

3. **Verify Physical Connections:**

   1. Confirm that cables are connected to the correct ports on devices. Check both ends of the cable to ensure they are securely plugged into the appropriate ports.

Troubleshooting Physical Connectivity Problems

4. **Use Different Cables and Ports:**

    1. If possible, try using different Ethernet cables and ports on networking devices to rule out faulty cables or ports. This helps determine whether the issue is specific to a particular cable or port.

5. **Check for Bent or Broken Pins:**

    1. Inspect connectors on cables and network devices for bent or broken pins. Damaged pins can prevent a proper connection and may need to be repaired or replaced.

Troubleshooting Physical Connectivity Problems

6. **Restart Devices:**
   1. Power cycle or restart networking devices, including routers, switches, and modems. Sometimes, a simple reboot can resolve connectivity issues caused by temporary glitches.

7. **Use Loopback Testers:**
   1. For Ethernet cables, you can use loopback testers to check for continuity and ensure that the cable is functioning correctly. A loopback tester simulates the presence of a connected device at the other end of the cable.

Troubleshooting
Physical Connectivity Problems

8. **Verify Link Lights:**
   1. Check the link lights on networking devices. The link lights indicate whether a connection has been established between devices. If the link lights are not lit, it may indicate a problem with the physical connection.

9. **Check Network Interface Cards (NICs):**
   1. Ensure that network interface cards (NICs) on computers and other devices are operational. Check for any error messages related to the NIC in the device's operating system.

Troubleshooting Physical Connectivity Problems

10. Inspect Physical Environment:

1. Consider environmental factors that may affect physical connectivity, such as interference, electromagnetic fields, or physical obstructions. Relocate devices if necessary.

11.Test with Other Devices:

1. Test connectivity using different devices to determine whether the issue is specific to a particular device or if it affects multiple devices.

12.Consult Documentation:

1. Refer to the documentation for networking devices and cables to ensure that they are compatible with each other and configured correctly.

Troubleshooting Physical Connectivity Problems

- If, after going through these steps, the physical connectivity issue persists, it may be necessary to involve network administrators or technicians with expertise in diagnosing and resolving complex network problems.

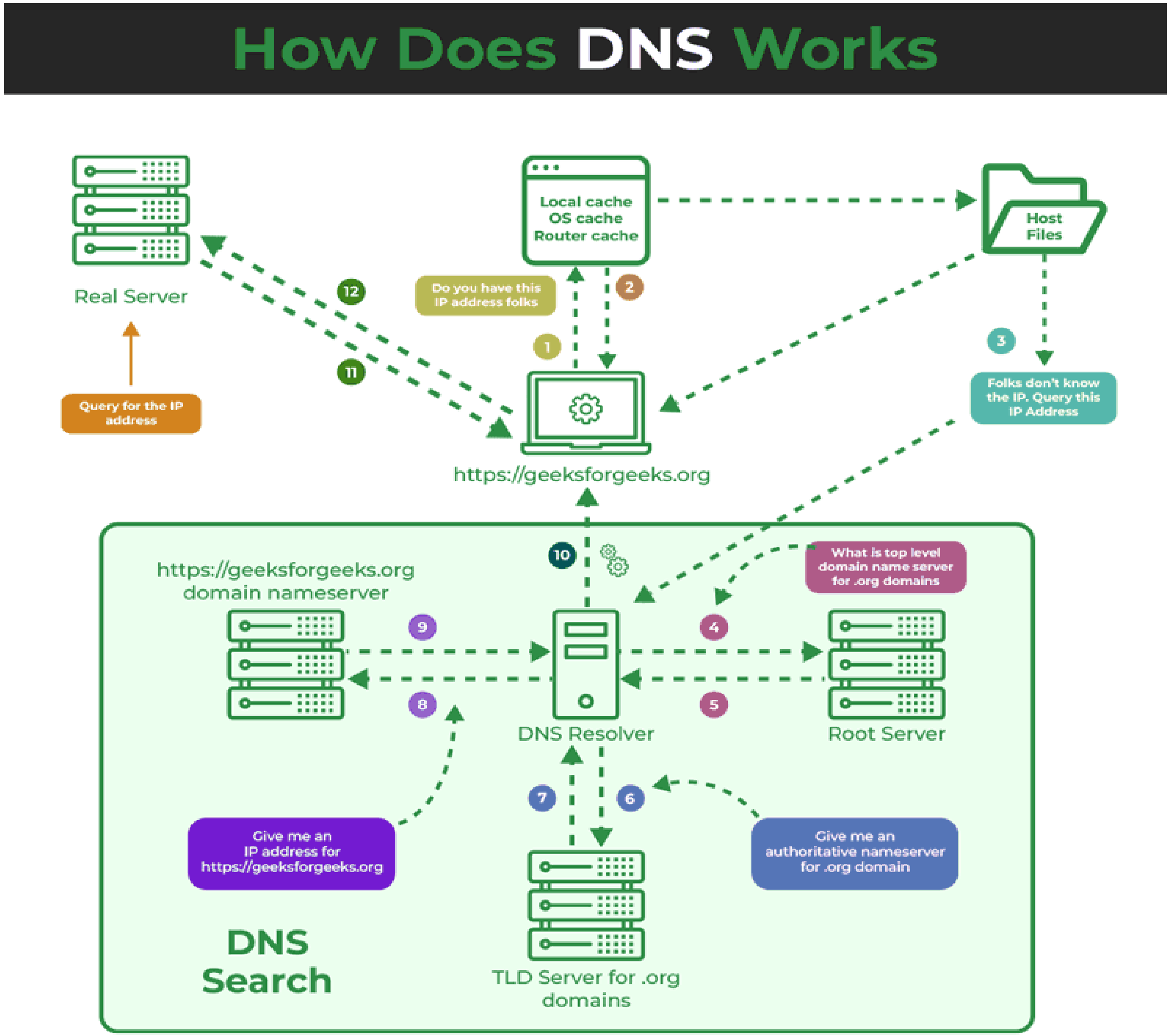
Name to address translation
Domain Name System

- The process of translating domain names into IP addresses is known as Domain Name System (DNS) resolution. DNS is a hierarchical system that provides the mapping between human-readable domain names and numerical IP addresses.

- The system allows users to access websites and other resources using easily remembered domain names instead of numerical IP addresses.

- Here's a breakdown of the key components involved in DNS name-to-address translation:

Name to address translation
Domain Name System


How Does DNS Works
Real Server
Query for the IP address
12
11
Local cache
OS cache
Router cache
Do you have this IP address folks
1
2
Host Files
3
Folks don't know the IP. Query this IP Address
https://geeksforgeeks.org
10
https://geeksforgeeks.org domain nameserver
9
8
DNS Resolver
What is top level domain name server for .org domains
4
5
Root Server
7
6
Give me an IP address for https://geeksforgeeks.org
Give me an authoritative nameserver for .org domain
TLD Server for .org domains
DNS Search

Name to address translation Domain Name System

- **Domain Name:**
  - A domain name is a human-readable label assigned to a specific IP address or a set of IP addresses associated with a particular resource on the internet. Examples include www.example.com or mail.google.com.

- **DNS Resolver:**
  - A DNS resolver is a component of the DNS client responsible for initiating DNS queries. When a user enters a domain name in a web browser, the resolver

Name to address translation Domain Name System

- **DNS Query:**
  - The DNS resolver sends a DNS query to the DNS server, requesting the IP address associated with the given domain name. The query is typically sent to the local DNS resolver or a DNS server specified in the system's configuration.

- **Root DNS Servers:**
  - The DNS query starts with the root DNS servers, which are authoritative servers that maintain information about top-level domains (TLDs), such as .com, .net, and .org. The root DNS servers direct the resolver to the authoritative DNS server for the appropriate TLD.

Name to address translation
Domain Name System

- **TLD DNS Servers:**
  1. The TLD DNS servers store information about domain names within their respective top-level domains. For example, the .com TLD DNS servers have information about domain names ending with .com.

- **Authoritative DNS Servers:**
  1. The authoritative DNS servers store the actual DNS records for specific domain names. These servers are responsible for providing the IP addresses associated with the requested domain names.

Name to address translation
Domain Name System

- **DNS Records:**
  - DNS records contain information mapping domain names to IP addresses. The most common type of DNS record is the A (Address) record, which associates a domain name with an IPv4 address. There are also AAAA records for IPv6 addresses, MX records for mail servers, CNAME records for aliasing, and others.
- **DNS Response:**
  - The authoritative DNS server responds to the DNS resolver's query with the IP address associated with the requested domain name. This response is then sent back to the client, allowing it to establish a connection to the desired resource.

Name to address translation
Domain Name System

- The entire process is transparent to users, and DNS resolution occurs automatically behind the scenes.

- The DNS system plays a crucial role in enabling users to access websites, send emails, and use various internet services by providing a way to translate human-friendly domain names into machine-readable IP addresses.

Transport layer protocols – TCP, UDP, ICMP, IGMP

- Transport layer protocols play a crucial role in the Internet Protocol (IP) suite, facilitating communication between applications running on devices connected to a network.

Here's an overview of four important transport layer protocols:

- TCP (Transmission Control Protocol)
- UDP (User Datagram Protocol)
- ICMP (Internet Control Message Protocol), and
- IGMP (Internet Group Management Protocol).

Transport layer protocols – TCP, UDP, ICMP, IGMP

TCP (Transmission Control Protocol):

1. **Connection-oriented:** TCP is a connection-oriented protocol that establishes a reliable and full-duplex communication channel between two devices before exchanging data. It ensures the ordered and error-checked delivery of data, handling retransmission of lost or corrupted packets.
2. **Features:**
    1. Reliable data delivery with error checking and correction.
    2. Flow control to manage the pace of data transmission.
    3. Connection establishment, maintenance, and termination through a three-way handshake.
    4. Sequencing of data to ensure proper order.

Transport layer protocols – TCP, UDP, ICMP, IGMP

3. **Use Cases:**
    1. File transfer (e.g., FTP - File Transfer Protocol).
    2. Web browsing (HTTP - Hypertext Transfer Protocol).
    3. Email (SMTP - Simple Mail Transfer Protocol).

UDP (User Datagram Protocol):

1. **Connectionless:** UDP is a connectionless protocol that provides a simple and lightweight communication method. It does not establish a connection before sending data and does not guarantee reliable delivery. UDP is often used in real-time applications where low latency is critical.

Transport layer protocols – TCP, UDP, ICMP, IGMP

2. **Features:**
    1. No connection setup or teardown.
    2. Unreliable and unordered delivery of data.
    3. Low overhead, making it faster than TCP.
    4. Suitable for real-time applications like streaming and gaming.
3. **Use Cases:**
    1. VoIP (Voice over Internet Protocol).
    2. Video streaming (e.g., YouTube).
    3. DNS (Domain Name System) queries.

Transport layer protocols – TCP, UDP, ICMP, IGMP

ICMP (Internet Control Message Protocol):

1. **Error Reporting and Network Management:** ICMP is a network layer protocol, but it is often associated with the transport layer due to its use in reporting errors and providing diagnostic information about network conditions.
2. **Features:**
   1. Error reporting (e.g., unreachable host or network).
   2. Network management and troubleshooting.
   3. Ping and traceroute utilities use ICMP.
3. **Use Cases:**
   1. Ping and traceroute.
   2. Network error reporting.

Transport layer protocols – TCP, UDP, ICMP, IGMP

IGMP (Internet Group Management Protocol):

1. **Multicast Group Management:** IGMP is used to manage multicast group memberships on a network. It enables hosts to join or leave multicast groups, allowing them to receive or stop receiving multicast traffic.
2. **Features:**
    1. Host membership management for multicast groups.
    2. Not used for actual data transmission but for group management.
3. **Use Cases:**
    1. IP television (IPTV) and video streaming.
    2. Multimedia applications that utilize multicast communication.

Transport layer protocols – TCP, UDP, ICMP, IGMP

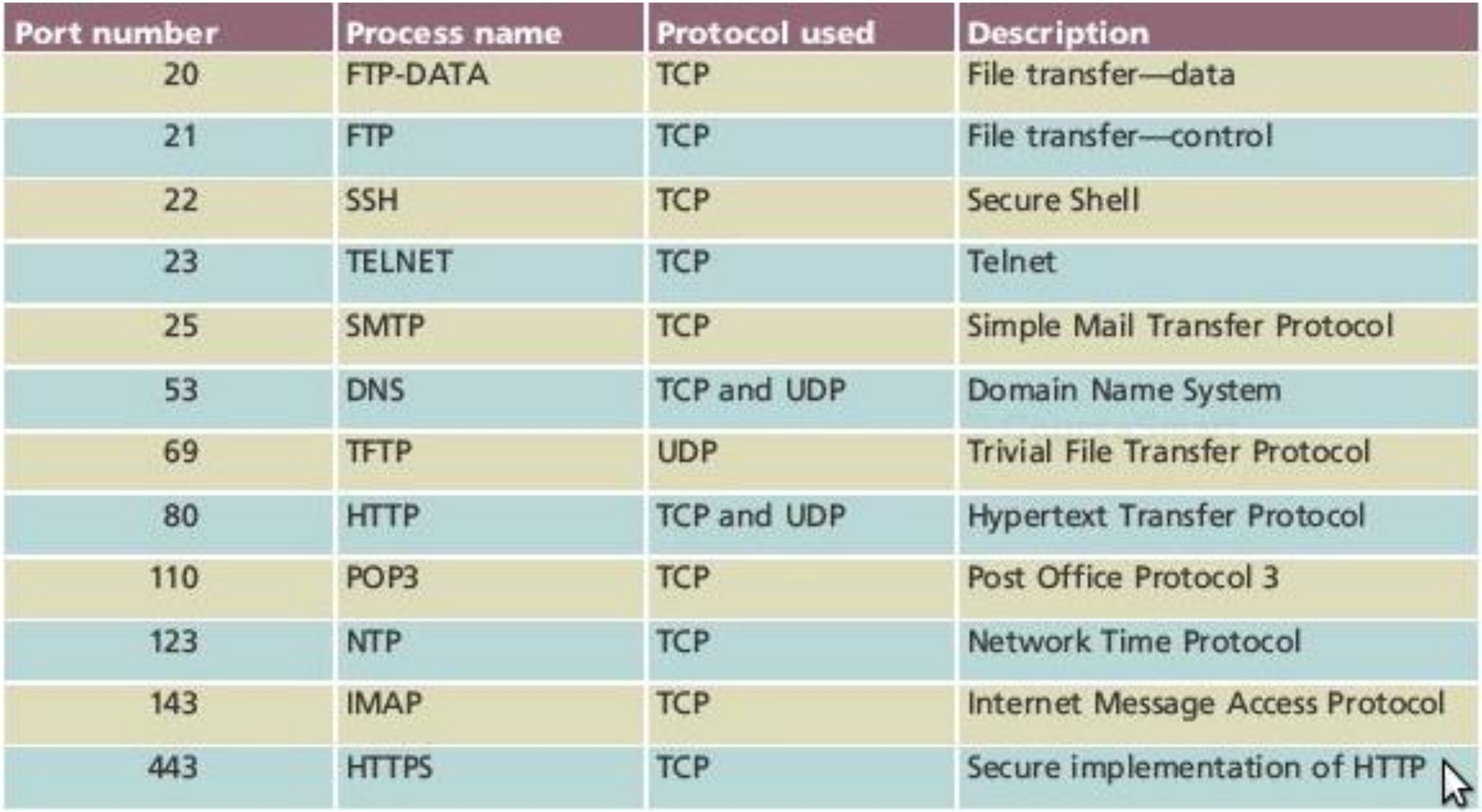
- These transport layer protocols serve different purposes and are chosen based on the specific requirements of the applications and services being used.

- TCP and UDP are the primary transport layer protocols for general data transmission, while ICMP and IGMP serve specific functions related to error reporting, network management, and multicast communication.

The power of port numbers – registered ports

- In computer networking, logical ports are communication endpoints associated with specific applications or services. These ports help facilitate the exchange of data between different applications running on devices within a network.
- Logical ports are identified by port numbers, and these port numbers are standardized to ensure consistency and interoperability across diverse networking environments. Registered ports are a subset of these logical ports.
- Here's an overview of logical ports, registered ports, and their roles in networking:

The power of port numbers – registered ports



Port number	Process name	Protocol used	Description
20	FTP-DATA	TCP	File transfer—data
21	FTP	TCP	File transfer—control
22	SSH	TCP	Secure Shell
23	TELNET	TCP	Telnet
25	SMTP	TCP	Simple Mail Transfer Protocol
53	DNS	TCP and UDP	Domain Name System
69	TFTP	UDP	Trivial File Transfer Protocol
80	HTTP	TCP and UDP	Hypertext Transfer Protocol
110	POP3	TCP	Post Office Protocol 3
123	NTP	TCP	Network Time Protocol
143	IMAP	TCP	Internet Message Access Protocol
443	HTTPS	TCP	Secure implementation of HTTP

The power of port numbers – registered ports

1. **Logical Ports:**
   1. **Definition:** Logical ports are virtual endpoints that allow networked devices to communicate with specific applications or services running on those devices.
   2. **Identification:** Logical ports are identified by port numbers ranging from 0 to 65,535.
   3. **Categorization:**
      1. Well-known ports (0 to 1023): Reserved for widely used and standardized services. Examples include HTTP (80), HTTPS (443), FTP (21), and Telnet (23).

The power of port numbers – registered ports

2. Registered ports (1024 to 49,151): Reserved for applications and services registered with the Internet Assigned Numbers Authority (IANA) to ensure uniqueness and prevent conflicts.
3. Dynamic or private ports (49,152 to 65,535): Used for ephemeral or temporary connections. These ports are typically dynamically assigned by the operating system.

2. **Registered Ports:**
   1. **Definition:** Registered ports are logical ports within the range of 1024 to 49,151. They are used by applications and services that have been officially registered with IANA.

The power of port numbers – registered ports

2. **IANA Registration:** Organizations or developers who create new networked applications can request the assignment of a registered port number from IANA to avoid conflicts with other applications.

3. **Examples:** Some examples of registered ports include:
   1. MySQL: 3306
   2. Oracle Database: 1521
   3. PostgreSQL: 5432
   4. Virtual Network Computing (VNC): 5900

The power of port numbers – registered ports

3. **Role of Ports in Networking:**

   1. **Communication:** Ports enable multiple applications on a device to communicate simultaneously over a network.
   2. **Packet Routing:** Logical ports are used by routers and switches to determine the destination application for incoming network packets.
   3. **Firewall Configuration:** Firewalls use port numbers to control and filter incoming and outgoing traffic based on specific rules for each application or service.

The power of port numbers – registered ports

4. **IANA (Internet Assigned Numbers Authority):**

    1. IANA is responsible for managing the assignment of port numbers to ensure that they are unique and standardized across the global internet.

    2. IANA maintains the Service Name and Transport Protocol Port Number Registry, which includes well-known, registered, and dynamic port assignments.

The power of port numbers – registered ports

- Understanding and properly managing logical ports, especially registered ports, is essential for maintaining order and preventing conflicts in network communications.

- Network administrators, application developers, and IANA work together to ensure that port assignments are well-coordinated and do not interfere with each other.

Connection status

- In computer networks, the connection status refers to the state or condition of a communication link between two devices or systems.

- Monitoring and understanding the connection status is crucial for network administrators to ensure proper functioning, troubleshoot issues, and optimize network performance.

- Here are common terms associated with connection status in computer networks:

Connection status

Connection status

1. **Established Connection:**
   1. An established connection indicates that communication has been successfully initiated and established between two devices or systems. In the context of the Transmission Control Protocol (TCP), a three-way handshake is used to establish a connection.

2. **Open Connection:**
   1. An open connection is a state where a communication link has been established, and data can be transmitted bidirectionally between the communicating devices.

Connection status

3. **Closed Connection:**
    1. A closed connection indicates that the communication link has been terminated or closed. In TCP, a connection can be closed gracefully using a four-way handshake.

4. **Listening State:**
    1. In the context of servers, a listening state means that the server is actively waiting for incoming connection requests from clients. Servers in the listening state are ready to establish new connections.

Connection status

5. **Half-Open Connection:**
   1. A half-open connection occurs when one side of the communication has terminated the connection, but the other side is still trying to send data. This situation can lead to communication issues and is typically addressed by ensuring proper connection termination.

6. **Dropped Connection:**
   1. A dropped connection refers to an unexpected termination of the communication link between devices. This can be caused by network issues, hardware failures, or software errors.

Connection status

7. **Timeout:**
   1. A timeout occurs when there is a delay in receiving expected data within a specified time frame. Timeouts are used to detect and handle situations where communication is taking longer than usual.

8. **Connection Refused:**
   1. A connection refused message indicates that the target device or service is actively rejecting the connection request. This can happen when a server reaches its maximum connection limit or when a service is not available.

Connection status

9. **Idle Connection:**
    1. An idle connection is in a state of inactivity, where no data is currently being transmitted. Idle connections are common in scenarios where devices periodically exchange small amounts of data.

10. **Failed Connection:**
    1. A failed connection indicates that the attempt to establish a connection was unsuccessful. Failures can result from issues such as network congestion, misconfigured settings, or hardware problems.

Connection status

11. Reconnecting:

1. Reconnecting refers to the process of attempting to establish a connection again after a previous connection has been closed or terminated.

- Monitoring tools, network protocols, and application-specific mechanisms are employed to track and manage connection status in computer networks.

- Understanding the different connection states is essential for diagnosing network issues, ensuring reliability, and maintaining the overall health of a network.

Rules for determining good vs. bad communications

- Determining whether network communications are good or bad involves assessing the performance, security, and reliability of the network. Here are some rules and criteria to consider when evaluating network communications:

1. Performance:

- **Throughput:** Measure the data transfer rate to ensure it meets the required speed for applications and services.
- **Latency:** Evaluate the delay in data transmission; lower latency is often preferred, especially for real-time applications.
- **Jitter:** Assess the variation in packet arrival times; consistent packet delivery is crucial for quality communication.

Rules for determining good vs. bad communications

2. Reliability:

- **Packet Loss:** Minimize packet loss to ensure that data is successfully transmitted without significant drops.
- **Availability:** Aim for high network availability to ensure that resources are consistently accessible.
- **Redundancy:** Implement redundancy mechanisms to prevent single points of failure and enhance reliability.

Rules for determining good vs. bad communications

3. Security:

- **Encryption:** Use encryption to secure data in transit and protect it from unauthorized access.
- **Access Control:** Implement proper access controls to restrict unauthorized users from accessing sensitive information.
- **Firewalls:** Employ firewalls to filter and monitor network traffic, blocking unauthorized access and potential threats.

Rules for determining good vs. bad communications

4. Scalability:

- **Capacity Planning:** Plan for future growth to accommodate an increasing number of users and devices.
- **Load Balancing:** Distribute network traffic across multiple servers to optimize resource utilization.
- **Scalable Protocols:** Choose protocols and technologies that can scale with the growth of the network.

Rules for determining good vs. bad communications

5. Monitoring and Management:

- **Network Monitoring:** Regularly monitor network performance, identify issues, and troubleshoot proactively.
- **Logging:** Maintain logs for auditing, troubleshooting, and security analysis.
- **Alerts:** Set up alerts for abnormal behavior or potential security incidents.

Rules for determining good vs. bad communications

6. Quality of Service (QoS):

- **Prioritization:** Implement QoS mechanisms to prioritize certain types of traffic, ensuring critical applications receive sufficient bandwidth.

- **Traffic Shaping:** Use traffic shaping to manage the flow of network traffic and prevent congestion.

Rules for determining good vs. bad communications

7. Compliance:

- **Regulatory Compliance:** Ensure that the network adheres to relevant regulatory requirements and standards.
- **Policy Compliance:** Enforce organizational policies related to network usage, security, and data protection.

8. Troubleshooting and Diagnostics:

- **Network Analysis Tools:** Use tools to analyze network traffic, diagnose issues, and identify bottlenecks.
- **Documentation:** Maintain accurate documentation to facilitate troubleshooting and network management.

Rules for determining good vs. bad communications

9. User Experience:

- **Accessibility:** Ensure that users can access resources without significant delays or disruptions.
- **User Feedback:** Gather feedback from users to understand their experience and address any concerns.

10. Cost-Effectiveness:

- **Resource Optimization:** Optimize resource utilization to avoid unnecessary costs.
- **Efficient Technologies:** Choose cost-effective technologies and solutions that meet the network's requirements.

Rules for determining good vs. bad communications

11. Adaptability:

- **Technology Evolution:** Keep abreast of technological advancements and be prepared to adapt the network to new requirements.
- **Agility:** Design the network with flexibility to accommodate changes in business needs and technological landscapes.

- Evaluating network communications involves a holistic approach, considering various factors to ensure that the network meets performance, reliability, security, and scalability requirements while adhering to compliance standards. Regular assessments, monitoring, and proactive management are essential to maintaining a healthy and effective network.

Common TCP/IP applications

- TCP/IP (Transmission Control Protocol/Internet Protocol) is the suite of protocols that forms the backbone of the Internet. It provides a set of communication protocols that allow computers to connect and exchange data over the Internet.

- Here are some common TCP/IP applications and their uses:
  - The world wide web
  - Telnet
  - Email
  - FTP
  - Internet applications

Common TCP/IP applications

1. **World Wide Web (WWW) - HTTP/HTTPS:**
   1. **Protocol:** HTTP (Hypertext Transfer Protocol) and HTTPS (Hypertext Transfer Protocol Secure).
   2. **Purpose:** Used for accessing and retrieving web pages on the World Wide Web. HTTPS adds a layer of security through encryption.

2. **Telnet:**
   1. **Protocol:** Telnet (Telnet Protocol).
   2. **Purpose:** Telnet allows a user to remotely log in to another computer on the Internet or a local network and use its resources as if they were physically present at that computer.

Common TCP/IP applications

3. **Email - SMTP, POP3, IMAP:**

   1. **Protocols:**
      1. SMTP (Simple Mail Transfer Protocol): Used for sending emails.
      2. POP3 (Post Office Protocol version 3): Used by email clients to retrieve emails from a mail server.
      3. IMAP (Internet Message Access Protocol): Also used by email clients to retrieve emails, offering more features than POP3.

Common TCP/IP applications

4. **FTP (File Transfer Protocol):**
   1. **Protocol:** FTP (File Transfer Protocol).
   2. **Purpose:** Used for transferring files between computers on a network. It allows users to upload and download files to and from remote servers.

5. **DNS (Domain Name System):**
   1. **Protocol:** DNS (Domain Name System).
   2. **Purpose:** Resolves domain names to IP addresses, enabling users to access resources on the Internet using human-readable domain names instead of numerical IP addresses.

Common TCP/IP applications

6. **DHCP (Dynamic Host Configuration Protocol):**
   1. **Protocol:** DHCP (Dynamic Host Configuration Protocol).
   2. **Purpose:** Automates the assignment of IP addresses and other network configuration parameters to devices in a network, simplifying network management.

7. **SNMP (Simple Network Management Protocol):**
   1. **Protocol:** SNMP (Simple Network Management Protocol).
   2. **Purpose:** Used for managing and monitoring network devices and their functions. SNMP enables the collection of information about network devices and the modification of their configuration.

Common TCP/IP applications

8. **SSH (Secure Shell):**
   1. **Protocol:** SSH (Secure Shell).
   2. **Purpose:** Provides secure access to a remote computer over a network. SSH encrypts the data transmitted, preventing unauthorized access and eavesdropping.

9. **NTP (Network Time Protocol):**
   1. **Protocol:** NTP (Network Time Protocol).
   2. **Purpose:** Synchronizes the time of computers on a network to a common time reference. It is crucial for maintaining accurate and consistent time across networked devices.

Common TCP/IP applications

10. **HTTP/2 and HTTP/3:**
    1. **Protocols:** HTTP/2 and HTTP/3.
    2. **Purpose:** Evolutions of the original HTTP protocol, designed to improve performance and reduce latency. HTTP/2 introduces features like multiplexing, while HTTP/3 uses a new transport protocol (QUIC) for faster and more secure communication.

- These applications showcase the versatility and functionality of TCP/IP protocols in enabling various communication services on the Internet. Each application serves a specific purpose and contributes to the overall functionality and usability of the global network.

Q & A

E.R. Ramesh, M.C.A., M.Sc., M.B.A.,
98410 59353, 98403 50547
rameshvani@gmail.com