NETWORK ARCHITECTURE MODELS

There are two models in Network Architecture and are:

1. OSI model
   OSI stands for Open Systems Interconnection. It is a conceptual framework that standardizes the functions of a communication system or network into seven distinct layers. Each layer has a specific purpose and interacts with adjacent layers to facilitate communication between devices. The OSI model serves as a guideline for designing and understanding network protocols and systems.
2. TCP/IP model
   The TCP/IP model, also known as the Internet Protocol Suite, is another network reference model. It is named after two of its prominent protocols, the Transmission Control Protocol (TCP) and the Internet Protocol (IP), which play crucial roles in data transmission and addressing within the model.

OSI Model Layers and Functions:

The OSI model consists of seven (7) layers:

1. Physical Layer:

Responsible for the physical transmission of raw data over the network medium (e.g., copper cables, optical fibers).
Defines characteristics like voltage levels, data rates, and connector types.
Transmits raw bits.

2. Data Link Layer:

Handles framing, addressing, and flow control to ensure error-free transmission.
Divided into LLC (Logical Link Control) and MAC (Media Access Control) sub-layers.
Manages access to the physical medium and handles link establishment and termination.
Transmits frames.

3. Network Layer:

Routes data packets across different networks to reach their destination.
Uses logical addressing (IP addresses) to identify devices.
Handles packet forwarding, routing decisions, and fragmentation/reassembly.
Transmits packets.

4. Transport Layer:

Ensures end-to-end communication reliability and data integrity.
Segments and reassembles data, providing error checking and flow control.
Commonly uses TCP (reliable, connection-oriented) and UDP (unreliable, connectionless) protocols.
Transmits segments or datagrams.

5. Session Layer:

Establishes, maintains, and terminates communication sessions between devices.

Provides synchronization points for data exchange and handles session recovery.
Transmits data between applications.

6. Presentation Layer:

Translates, encrypts, and compresses data for proper interpretation by applications.

Converts data between different formats, ensuring compatibility between systems.
Transmits formatted data.

7. Application Layer:

Provides network services directly to end-user applications.
Contains application protocols like HTTP, FTP, SMTP, and more.
Supports user communication and data exchange.
Transmits data between applications.

OSI Model vs. TCP/IP Model:

The OSI model and the TCP/IP model have different numbers of layers, but they serve similar purposes:
OSI Model: Provides a theoretical framework with seven layers, focusing on the logical flow of data.
TCP/IP Model: Offers a more streamlined approach with four layers, closely aligned with the functionality of the modern internet.

Layers Typically Targeted by Hackers:

Hackers often target specific layers to exploit vulnerabilities:

1. Application Layer:

Hackers use techniques like SQL injection, cross-site scripting (XSS), and remote code execution to compromise applications and steal data.

2. Transport Layer:

Man-in-the-Middle (MitM) attacks involve intercepting and altering communication between devices.
Hackers exploit weak encryption or authentication mechanisms.

3. Network Layer:

IP spoofing attacks involve using forged IP addresses to impersonate legitimate devices.
Distributed Denial of Service (DDoS) attacks overwhelm networks, rendering services inaccessible.

4. Data Link Layer:

MAC address spoofing and attacks on switches can allow unauthorized access to a network.
ARP spoofing redirects network traffic, enabling attackers to intercept data.

5. Physical Layer:

Physical theft or tampering with networking equipment can lead to unauthorized access or data breach.

In conclusion, understanding the different layers of network models, their functions, and the vulnerabilities associated with them is crucial for designing secure and resilient network architectures. Staying informed about potential threats at each layer is essential for maintaining network security.

OSI Model Layer Data

1. Physical Layer:

Bits: Raw binary data with no structure. It's the fundamental unit of data transmission at this layer.

2. Data Link Layer:

Frames: Structured units that include the data being transmitted, control information (such as error-checking codes), and addressing information (MAC addresses).

3. Network Layer:

Packets: Logical units of data that include the payload (actual data being transmitted) along with headers containing source and destination IP addresses.

4. Transport Layer:

Segments (TCP) / Datagrams (UDP): Segments in TCP or datagrams in UDP include the transported data, along with headers containing source and destination port numbers.

5. Session Layer:

Data: The actual data generated by the application and being transmitted between communicating systems.

6. Presentation Layer:

Data: Here, data still refers to the application's payload, but this layer is responsible for data translation, encryption, and compression if needed.

7. Application Layer:

Messages or Data: This layer deals with the actual data structures that applications generate, whether they are messages, files, or any other application-specific information.

TCP/IP Model Layer Data

1. Network Interface Layer:

Frames: Similar to the OSI Data Link Layer, this layer deals with the transmission of frames containing data, error-checking codes, and MAC addresses.

2. Internet Layer:

Packets: Similar to the OSI Network Layer, packets here include the data being sent, along with IP headers containing source and destination IP addresses.

3. Transport Layer:

Segments (TCP) / Datagrams (UDP): These units include the transported data, along with headers that provide information like port numbers.

4. Application Layer:

Messages or Data: This layer involves the actual data generated by applications, whether it's a web page, an email, or any other application-specific content.

In Summary:

Each layer of the network models deals with specific units of data or frames tailored to its purpose in the communication process. Understanding these units of data is essential for comprehending how information flows through the layers and how each layer contributes to successful communication while also providing opportunities for security vulnerabilities.

Data Transmission in OSI Model:

1. Application Layer:

Sends a Data Message (high-level data generated by an application).

2. Presentation Layer:

Encrypts and applies necessary formatting to the data, creating Encrypted Data with Formatting.

3. Session Layer:

Adds synchronization tokens and manages session-related information, resulting in Tokens & Synchronization.

4. Transport Layer:

Divides the data into smaller units called Segments (TCP) / Datagrams (UDP) for efficient transmission.

5. Network Layer:

Adds routing headers and creates IP Packets that include the source and destination IP addresses.

6. Data Link Layer:

Further encapsulates the packets into Frames, attaching MAC addresses for source and destination.

7. Physical Layer:

The frames are converted into electrical signals or light pulses for actual transmission over the physical medium.

Data Transmission in TCP/IP Model:

1. Application Layer:

Generates a Data Message representing the data intended for transmission.

2. Transport Layer:

Segments the data into Segments (for TCP) or Datagrams (for UDP) and adds port information.

3. Internet Layer:

Takes the segments/datagrams and encapsulates them into Packets with source and destination IP addresses.

4. Network Interface Layer:

Creates Frames by adding MAC addresses and error-checking information to the packets.

5. Physical Layer:

Converts the frames into appropriate signals or pulses for physical transmission through the network medium.

At the receiving end, the process reverses as data travels upwards through the layers. Each layer extracts and processes the relevant information, progressively reconstructing the original data generated by the application.

Please note that this description provides a simplified overview of data transmission, and in real-world scenarios, there can be additional complexities and protocols involved at each layer.

Ports in Networking

In networking, a port is a numerical identifier that directs data to specific applications or services on a device. Ports work alongside IP addresses to ensure accurate data routing. They are crucial for allowing various applications to communicate simultaneously on a single network connection.

Types of Ports:

1. Well-Known Ports (0-1023):

Reserved for standardized services and protocols.

Frequently used ports include HTTP (port 80), HTTPS (port 443), and FTP (port 21).

2. Registered Ports (1024-49151):

Used by various applications but aren't standardized like well-known ports.

Often employed for custom applications and services.

3. Dynamic/Private Ports (49152-65535):

Used temporarily, typically by client applications initiating connections.

4. Port Numbers and Communication:

Port numbers, combined with IP addresses, form a socket, a unique communication identifier.

Data sent to an IP address is directed to a specific application based on the corresponding port number.

Importance of Ports

1. Multiplexing and Demultiplexing:

Multiplexing combines multiple data streams into one connection.

Demultiplexing splits combined data streams into individual streams.

Ports enable this process, letting different applications share a connection.

2. Concurrent Communication:

Ports enable hosting multiple applications on one device, communicating concurrently.

E.g., a web server can use ports 80 for HTTP and 443 for HTTPS simultaneously.

3. Firewall and Security:

Firewalls use port numbers to allow/restrict communication.

Securely configuring firewalls enhances network security by controlling open ports.

4. Load Balancing:

Load balancers distribute requests among multiple servers using port numbers.

Common Port Numbers:

Port numbers vary based on protocols/services. Common ones include:

HTTP: 80

HTTPS: 443

FTP: 21

SSH: 22

SMTP: 25

DNS: 53

IMAP: 143

POP3: 110

Total Number of Ports:

The total number of ports available is 65536 (2^16).

5. Web Server Reference

Web servers often use port 80 for HTTP and port 443 for HTTPS. When encountering an invalid or non-existent page request on a web server, such as a "404 Error," the server responds with an HTTP status code 404, indicating that the requested resource was not found.

In summary, ports are essential for organized networking, allowing applications to communicate effectively. Port numbers, along with IP addresses, direct data to the right destinations, enabling seamless communication across various services and devices.

Protocols in Networking

In networking, a protocol is a set of rules that govern how data is exchanged between devices. Protocols ensure consistent and reliable communication by defining the format, timing, sequencing, and error handling of data transmission.

Types of Protocols

1. Communication Protocols:

Specify how devices communicate and exchange data.

- TCP (Transmission Control Protocol): Provides reliable, connection-oriented communication. It ensures data integrity and ordered delivery, making it suitable for applications that require accuracy, such as web pages and emails.
- UDP (User Datagram Protocol): Offers fast, connectionless communication. While it lacks error-checking and sequencing like TCP, it is used when speed is prioritized, as in video streaming and online gaming.

2. Routing Protocols:

Determine the best path for data to travel across a network.

3. RIP (Routing Information Protocol): A simple protocol used in small networks. It exchanges routing information regularly.
4. OSPF (Open Shortest Path First): A more advanced protocol suitable for larger networks. It calculates the shortest path based on various factors.
5. Application Layer Protocols:

Govern how applications interact and exchange data.

- HTTP (Hypertext Transfer Protocol): Used for web page retrieval.
- HTTPS (HTTP Secure): Adds encryption for secure data exchange.
- SMTP (Simple Mail Transfer Protocol): Sends email.
- IMAP (Internet Message Access Protocol): Retrieves email and maintains email on the server.
- POP3 (Post Office Protocol): Retrieves and deletes email from a mail server.

Protocol Components

- Message Format:
  Defines how data is structured within messages, including headers, body, and sometimes trailers.
- Addressing and Routing:
  Specifies how devices are addressed and how data is routed to its destination.
- Handshaking:

Describes the process of establishing, maintaining, and terminating connections between devices.

- Flow Control:
  Manages the pace of data transmission to prevent congestion and ensure efficient communication.
- Error Detection and Correction:

Implements methods for detecting and handling errors in data transmission to ensure data integrity.

Importance of Protocols

- Interoperability:

Protocols enable devices from different manufacturers to communicate effectively.

- Efficiency:

Standardized protocols reduce complexity, ensuring smooth and efficient communication.

- Reliability:

Protocols provide error detection and correction mechanisms, enhancing data integrity.

- Scalability:

Well-defined protocols allow networks to grow and adapt without disrupting communication.

Common Network Protocols:

- TCP/IP:
  The foundation of the internet, consisting of protocols like TCP, UDP, and IP (Internet Protocol). IP defines how devices are addressed and how data is routed.
- HTTP/HTTPS:
- HTTP (Hypertext Transfer Protocol) is used for web page retrieval.
- HTTPS (HTTP Secure) adds encryption for secure data exchange.
- SMTP/IMAP/POP3:
- SMTP (Simple Mail Transfer Protocol) sends email.
- IMAP (Internet Message Access Protocol) and POP3 (Post Office Protocol) retrieve email.
- DNS (Domain Name System):

Translates domain names into IP addresses, enabling human-readable web addresses.

Protocol Evolution:

Protocols continue to evolve to meet the changing needs of networks and technologies.

New protocols, like IPv6 (Internet Protocol version 6), are developed to address the limitations of older ones (e.g., IPv4).

In summary, protocols serve as the foundation of network communication, enabling devices to exchange data reliably and efficiently. They establish a common language that devices understand, ensuring seamless interaction and data transfer across diverse network environments.