CELEBAL TECHNOLOGY INTERNSHIP (CSI)

Name: Harsh Tongariya

College: Arya College of Engineering Information Technology

Domain: Cloud Infra & Security

Student ID: CT_CSI_CI_1160

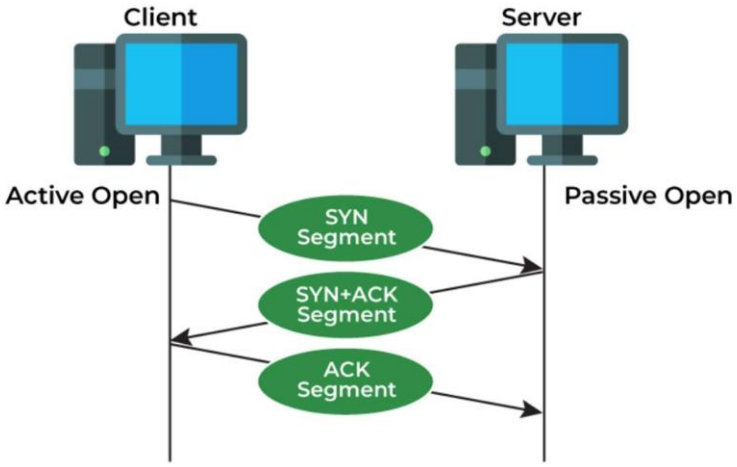
Working of TCP & UDP Protocols | Working of HTTP, HTTPS & ICMP Protocols

---

1. Introduction

Communication over the Internet is made possible through a wide range of protocols, each designed to address specific requirements like data transfer, security, reliability, and control. Two of the core transport-layer protocols are **TCP (Transmission Control Protocol)** and **UDP (User Datagram Protocol)**, which govern how data packets are transmitted between computers. Higher-layer protocols like **HTTP**, **HTTPS**, and **ICMP** build on these transport protocols to provide functionality such as web browsing, secure communication, and diagnostics.

This document details the **working principles**, **key features**, and **use cases** of these protocols to provide a deeper understanding of their roles in network communication.



---

2. Working of TCP and UDP Protocols

2.1 TCP (Transmission Control Protocol)

Overview:
TCP is a **connection-oriented** and **reliable** transport-layer protocol. It ensures that data is delivered accurately and in the correct order from sender to receiver.

Key Features:

1. **Three-Way Handshake Process:**
    - **SYN (Synchronize):** Initiated by the client to start the connection.
    - **SYN-ACK:** Server acknowledges the request and responds with synchronization.
    - **ACK (Acknowledge):** Client confirms, and the connection is established.
2. **Reliable Data Transmission:**
    - **Sequencing:** Each packet is assigned a sequence number to track ordering.
    - **Acknowledgment (ACK):** Receiver confirms successful receipt of data.
    - **Retransmission:** If packets are lost or corrupted, TCP retransmits them.
3. **Flow Control:**
    - Uses **sliding window** mechanism to ensure the sender does not overwhelm the receiver.
4. **Congestion Control:**
    - Implements algorithms like **Slow Start**, **Congestion Avoidance**, **Fast Retransmit**, and **Fast Recovery** to maintain network stability.

Use Cases:

- Web browsing (HTTP/HTTPS)
- Email (SMTP, IMAP)
- File transfers (FTP)
- Remote administration (SSH, Telnet)

2.2 UDP (User Datagram Protocol)

Overview:
UDP is a **connectionless** and **unreliable** transport-layer protocol. It is used where speed is more critical than reliability.

Key Features:

1. **No Connection Setup:**
    - UDP does not establish a session before sending data. Each packet (datagram) is sent independently.
2. **No Reliability:**
    - No acknowledgment or retransmission.
    - Packets may be lost, duplicated, or received out of order.
3. **Minimal Overhead:**
    - Faster due to lack of flow/congestion control and session handling.

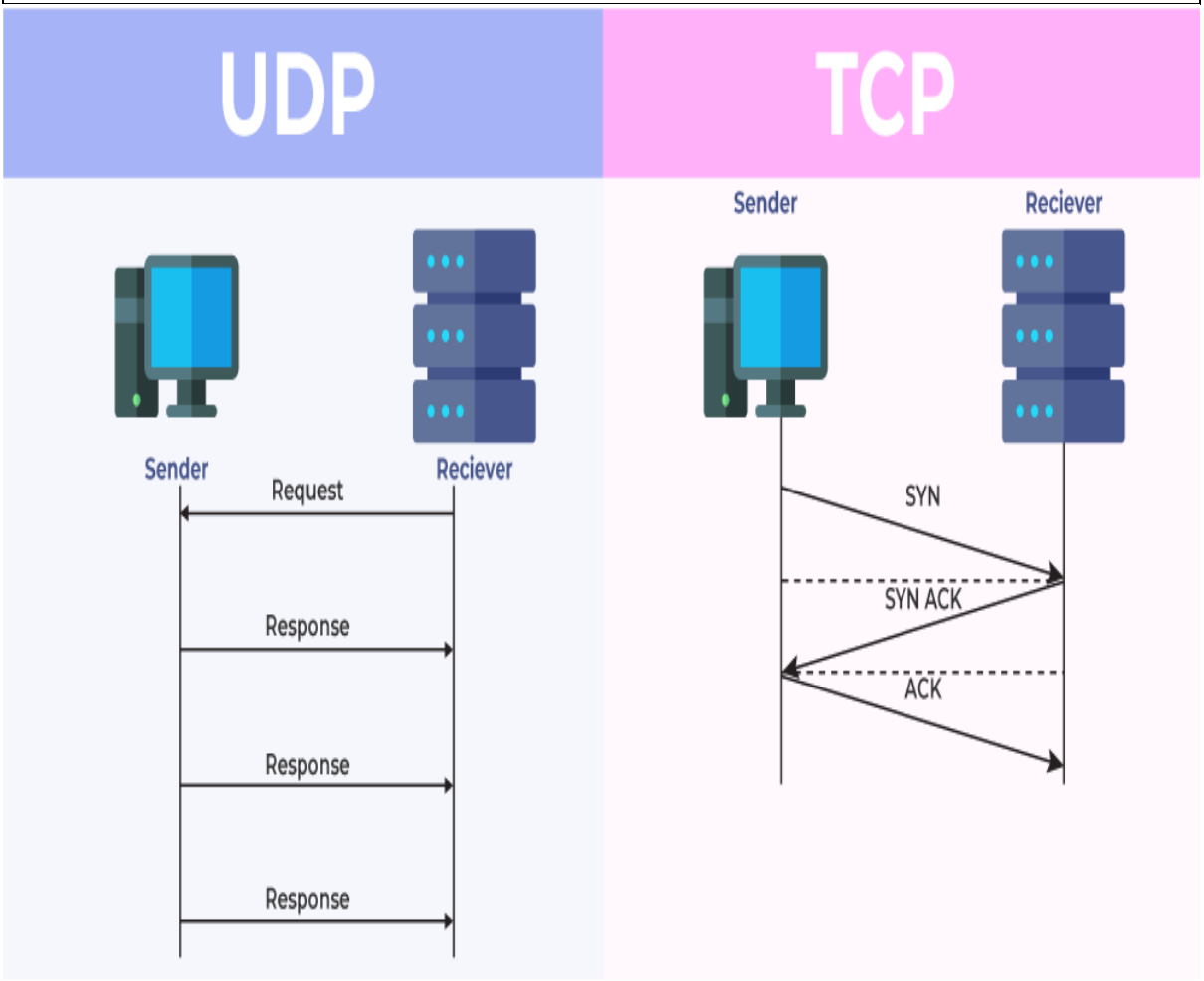
Use Cases:

- Live video/audio streaming
- Online multiplayer games
- DNS lookups
- VoIP (Voice over IP)

2.3 Comparison Table: TCP vs UDP

Feature	**TCP**	**UDP**
Protocol Type	Connection-Oriented	Connectionless
Reliability	High – ensures delivery and order	Low – no guarantee of delivery
Error Checking	Yes, with acknowledgments	Yes, but no retransmission
Order of Packets	Maintained	Not maintained
Overhead	High	Low

Feature	TCP	UDP
Speed	Slower due to reliability features	Faster due to minimal processing
Use Case Examples	HTTP, FTP, SMTP	VoIP, DNS, Live Streaming



---

3. Working of HTTP, HTTPS & ICMP Protocols

3.1 HTTP (HyperText Transfer Protocol)

Overview:
HTTP is an **application-layer** protocol used for transferring data over the web.

Working Mechanism:

- When a user accesses a website, the web browser (client) sends an **HTTP request** to the server.
- The server responds with an **HTTP response**, which includes the requested content (HTML, CSS, images, etc.).

Key Characteristics:

- **Stateless:** Each request is treated independently with no memory of previous requests.
- **Uses TCP:** HTTP operates over TCP (typically port 80).

Common Methods:

- GET – Retrieve data
- POST – Submit data
- PUT – Update data
- DELETE – Remove data

3.2 HTTPS (HyperText Transfer Protocol Secure)

Overview:
HTTPS is the **secure version of HTTP**. It encrypts data using **SSL/TLS**, ensuring confidentiality, integrity, and authentication.

Working Mechanism:

1. **TLS/SSL Handshake:**
    - Client and server agree on encryption parameters.
    - Server sends its **digital certificate** (public key).
2. **Session Key Exchange:**
    - Client and server generate and agree on a shared session key.
3. **Secure Data Transmission:**
    - All subsequent HTTP communication is encrypted.

Security Features:

- **Encryption:** Prevents eavesdropping.

- **Authentication:** Verifies server identity via certificates.
- **Integrity:** Detects if data has been tampered with.

Uses TCP on Port 443

Use Cases:

- Online banking
- E-commerce
- Login pages
- Secure APIs

3.3 ICMP (Internet Control Message Protocol)

Overview:
ICMP is a **network-layer protocol** used for diagnostics and error reporting in IP networks. It is not used to transmit data between applications.

Working Mechanism:

- ICMP messages are encapsulated within IP packets.
- Used to indicate errors (e.g., "Destination Unreachable") or informational messages (e.g., ping).

Common Message Types:

- **Echo Request/Reply (Type 8/0):** Used by ping to check reachability.
- **Destination Unreachable (Type 3):** Informs the sender that the destination cannot be reached.
- **Time Exceeded (Type 11):** Used in traceroute to map the path packets take.

Use Cases:

- ping command to test host availability.
- traceroute for route diagnostics.
- Routers and gateways use ICMP to report problems.

---

4. Conclusion

TCP and UDP are foundational protocols for Internet communication, with TCP ensuring reliable, ordered delivery, while UDP offers fast, low-latency data transfer. HTTP and HTTPS use TCP to transmit web content, with HTTPS adding a critical layer of encryption. ICMP, on the other hand, is essential for diagnostic operations and managing network errors.

Understanding these protocols helps network engineers and developers design applications that are efficient, secure, and suited to their operational needs.

References

- Diffen.com. “TCP vs UDP.” https://www.diffen.com/difference/TCP_vs_UDP