R&D on TCP & UDP Protocols, working of HTTP, HTTPs & ICMP Protocol

Prepared by:

Virat Pandey

Celebal Technology

Cloud Infrastructure & Security Internship

Research & Development Document

June 8, 2025

Table of contents

1. Introduction

Communication over networks is made possible through a set of well-defined protocols that govern how data is packaged, transmitted, and interpreted across systems. These protocols operate at different layers of the **TCP/IP model** and ensure the seamless flow of information in local, wide, and global networks like the internet.

In this report, we focus on five core protocols that play essential roles in modern network communication:

- **TCP** and **UDP**, which are Transport Layer protocols responsible for data delivery between devices
- **HTTP** and **HTTPS**, which function at the Application Layer and enable web-based communication
- **ICMP**, a Network Layer protocol used primarily for diagnostics and network health monitoring

These protocols are not just theoretical constructs they form the backbone of everyday technologies including **web browsing, streaming, cloud services, system monitoring**, and more. Understanding how they work individually and interact collectively is crucial for managing, securing, and optimizing modern IT infrastructure.

This report will explore the **working principles, use cases, and real-world importance** of each protocol while aligning with industry references such as **Microsoft Learn** and cloud infrastructure practices.

2. Transport Layer Protocol

2.1 Transmission Control Protocol (TCP)

TCP (Transmission Control Protocol) is a **connection-oriented** protocol that ensures **reliable and ordered delivery** of data between devices. It operates at the **Transport Layer** of the TCP/IP model and is one of the most widely used network protocols in the world.

TCP uses a **3-way handshake** mechanism to establish a connection before transmitting data. It also verifies that all packets are received correctly and in order. If any packets are lost or damaged, TCP **retransmits** them.

- **Key Features:**

- **Connection-oriented:** Establishes a session before transmitting data
- **Reliable delivery:** Retransmits lost or corrupted packets
- **Ordered data:** Maintains the correct sequence of packets
- **Flow control & congestion control:** Avoids overwhelming the receiver or the network

- **Common Use Cases:**

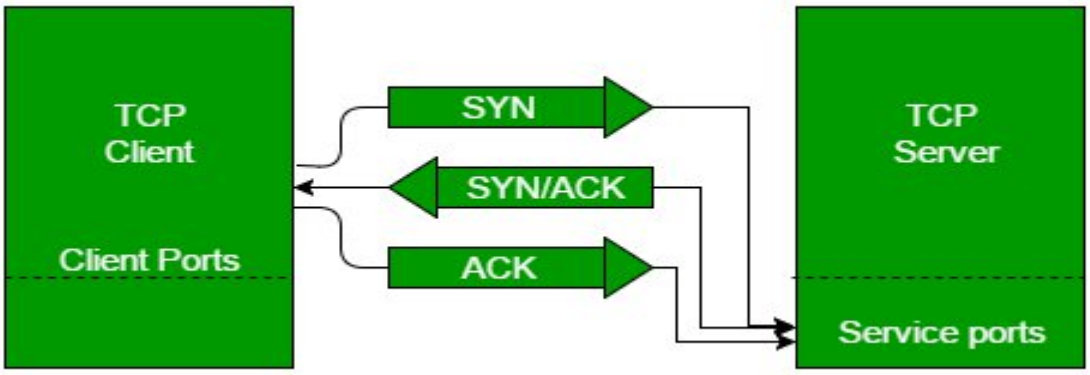
- **Web traffic:** HTTP/HTTPS
- **Email transmission:** SMTP, IMAP
- **File transfers:** FTP
- **Cloud APIs:** REST APIs over HTTPS

- **How It Works:**

1. **3-Way Handshake:**
    - Client → SYN → Server
    - Server → SYN-ACK → Client
    - Client → ACK → Server → Connection established

2. **Data Transmission:**
    - Divides large data into segments
    - Adds sequence numbers and acknowledgment numbers
3. **Connection Termination:**
    - Gracefully closes connection using FIN/ACK packets

"TCP is used when applications require reliable and error-free transmission, such as when downloading files or browsing secure websites." — Microsoft Learn [1]



2.2 User Datagram Protocol (UDP)

UDP (User Datagram Protocol) is a **connectionless** protocol used when speed is prioritized over reliability. It also operates at the **Transport Layer** of the TCP/IP model, but unlike TCP, it does not establish a connection before sending data and does not guarantee delivery, order, or error correction.

UDP is lightweight and faster than TCP, making it ideal for **real-time communication**, where slight data loss is acceptable but low latency is essential.

- **Key Features:**

- **Connectionless:** No handshake or session establishment
- **Fast transmission:** Minimal protocol overhead
- **No guarantee of delivery or order**
- **No built-in error correction or retransmission**

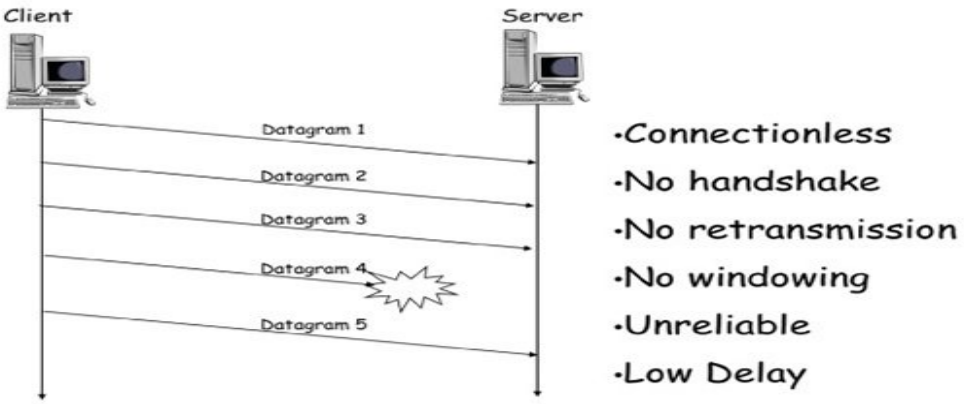
- **Common Use Cases:**

- **Live video/audio streaming** (e.g., Zoom, YouTube Live)
- **Online gaming** (e.g., PUBG, Valorant)

- **Voice over IP (VoIP)** (e.g., Skype, WhatsApp calls)
- **DNS lookups** (quick one-request–one-response)

  - **How It Works:**
- Data is packaged into **datagrams**
- Sent directly to the recipient's IP and port without prior communication
- If packets are lost or arrive out of order, UDP does not attempt to fix them

> "UDP is useful for applications where speed is critical and occasional data loss is acceptable, such as video conferencing or online gaming." — Microsoft Learn [1]



3. Application Layer Protocol

3.1 Hypertext Transfer Protocol (HTTP)

HTTP (Hypertext Transfer Protocol) is an application layer protocol used for transferring data on the web. It is the foundation of any data exchange on the World Wide Web and follows a request-response model a client (like a browser) sends a request, and the server responds with the required data (like a webpage or file).

HTTP works over TCP to ensure reliable delivery of resources, but by itself, HTTP is stateless, meaning each request is treated independently, without knowledge of previous interactions.

- **Key Features:**

- Client-server communication: Browser (client) communicates with a web server
- Text-based and stateless: Every request is independent of the last
- Supports multiple methods: GET, POST, PUT, DELETE, etc.
- Uses TCP for delivery (typically port 80)

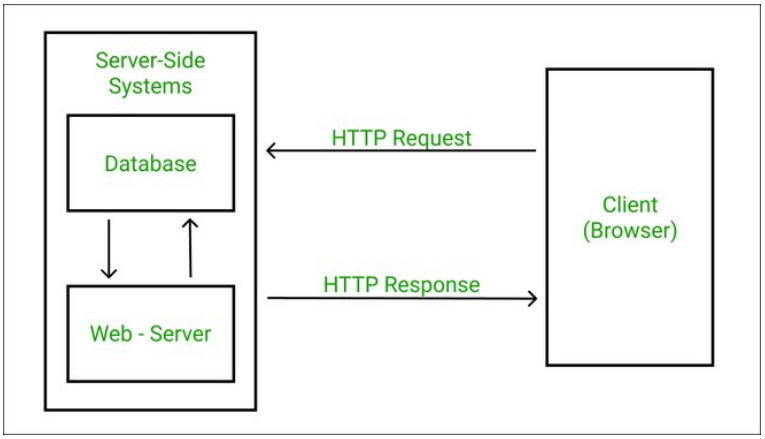
- **Common Use Cases:**

- Accessing websites via web browsers
- Requesting content from RESTful APIs
- Downloading files and documents
- Communicating between web-based front-end and back-end systems

- **How It Works:**

1. Client sends an HTTP request (e.g., GET /index.html)
2. Server receives it and processes the request
3. Server responds with an HTTP response (e.g., a web page or error code)
4. The connection may be closed, or reused if supported (keep-alive)

> “HTTP is the protocol used by the World Wide Web to define how messages are formatted and transmitted, and how servers and browsers should respond to various commands.” — Microsoft Learn [1]



3.2 Hypertext Transfer Protocol Secure (HTTPS)

HTTPS (Hypertext Transfer Protocol Secure) is the **secure version of HTTP**. It combines the standard HTTP protocol with **SSL/TLS encryption** to ensure that all communication between the client and server is **encrypted, authenticated, and protected from tampering**.

HTTPS is critical for securing sensitive transactions like **login pages, online banking, and any form of data submission**. It operates over **TCP port 443** and uses certificates to verify the identity of the server.

- **Key Features:**

- **Encryption:** Prevents third parties from reading the data in transit
- **Authentication:** Verifies the identity of the server via SSL/TLS certificates
- **Data integrity:** Ensures that content is not modified during transmission
- **Built on HTTP + SSL/TLS**

- **Common Use Cases:**

- Online payments and e-commerce platforms
- Secure APIs for mobile and web applications
- Any form of data login, upload, or exchange requiring confidentiality

- **How It Works:**

1. Client initiates a connection via HTTPS
2. Server presents a **digital certificate (SSL/TLS)**
3. A **secure handshake** is performed to establish encryption keys
4. Encrypted HTTP data is exchanged using **TLS over TCP**

> “HTTPS encrypts HTTP messages using SSL/TLS to prevent unauthorized access or tampering during transmission.” — Microsoft Learn [1]

4. Network Layer Protocol

4.1 Internet Control Message Protocol (ICMP)

ICMP (Internet Control Message Protocol) is a **network layer protocol** used for **diagnostics, error reporting**, and **network communication troubleshooting**. Unlike TCP or UDP, ICMP is **not used to send data between applications**, but to **send control messages** such as reporting unreachable destinations, latency issues, or packet loss.

One of the most common uses of ICMP is the **ping command**, which tests the reachability of a host and measures round-trip time for messages.

- **Key Features:**

- Operates at the **Network Layer**
- Used for **error handling**, diagnostics, and control signalling
- Works **alongside IP**, not over TCP/UDP
- Contains **message types** like Echo Request/Reply, Destination Unreachable, Time Exceeded

- **Common Use Cases:**

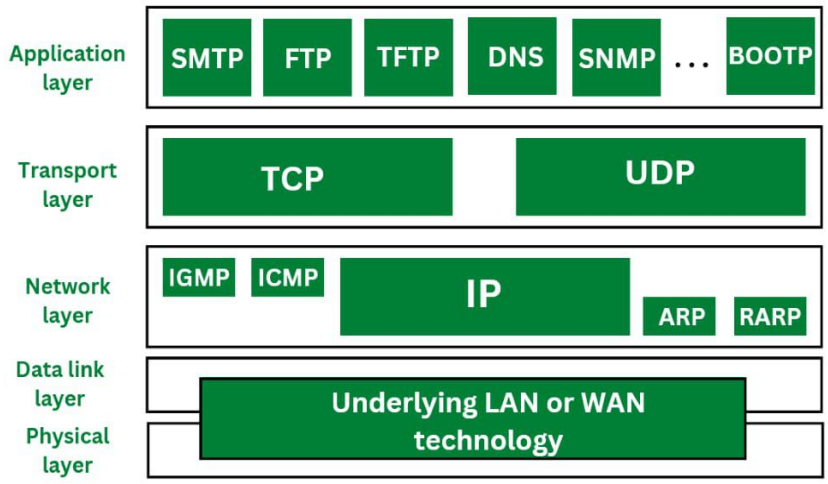
- **Ping:** Tests whether a device is reachable and how fast it responds
- **Traceroute:** Identifies the path data takes through the network
- **Error reporting:** Informs source hosts of network issues (e.g., unreachable routers)

- **How It Works:**

1. Host A sends an **ICMP Echo Request** to Host B

2. If Host B is reachable, it replies with an **ICMP Echo Reply**
3. If a router cannot forward the packet, it may send an **ICMP Destination Unreachable** message back to the source

> "ICMP is not a transport protocol, but an integral part of IP that helps in network diagnostics and error handling." — Microsoft Learn [1]



5. Comparison & Use Cases

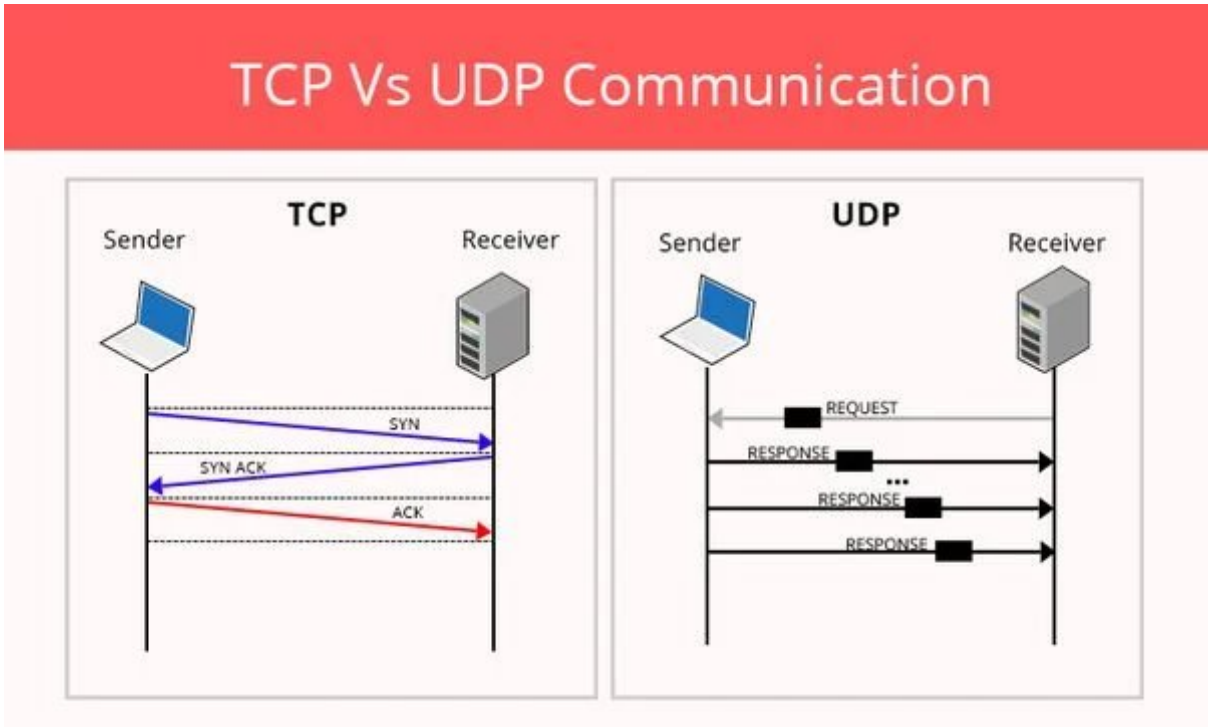
TCP vs UDP

Feature	TCP (Transmission Control Protocol)	UDP (User Datagram Protocol)
Connection Type	Connection-oriented (reliable, establishes connection)	Connectionless (no connection setup)
Reliability	Guaranteed delivery with acknowledgments	No guarantee; best-effort delivery
Flow Control	Yes, uses congestion and flow control mechanisms	No flow control
Ordering	Ensures packets arrive in order	No ordering; packets may arrive out of sequence

Feature	TCP (Transmission Control Protocol)	UDP (User Datagram Protocol)
Speed	Slower due to overhead	Faster, less overhead
Use Cases	File transfers, emails, web browsing (HTTPS)	Streaming, gaming, VoIP, DNS lookups

HTTP, HTTPS & ICMP Overview

- **HTTP (Hypertext Transfer Protocol):**
  Protocol for transferring web pages and resources over the internet. Operates on TCP port 80. It is stateless and text-based, enabling browsers and servers to communicate.
- **HTTPS (HTTP Secure):**
  Secure version of HTTP that encrypts data using SSL/TLS protocols, ensuring confidentiality and data integrity. Operates on TCP port 443. Widely used for secure transactions like banking and online shopping.
- **ICMP (Internet Control Message Protocol):**
  Used for network diagnostics and error reporting (e.g., ping and traceroute commands). Operates at the network layer, not for data transport but for control messages between hosts and routers.



6. Conclusion

The TCP/IP suite is fundamental to modern networking, providing flexible protocols suited to diverse application needs. **TCP** ensures reliable, ordered data transmission essential for critical applications, while **UDP** offers low-latency communication suited for real-time services. **HTTP** and **HTTPS** enable web-based content delivery, with HTTPS adding crucial security layers to protect user data. Meanwhile, **ICMP** supports network health monitoring and troubleshooting, helping maintain network reliability.

Together, these protocols form the backbone of the internet, enabling seamless, efficient, and secure communication across diverse devices and networks worldwide. Mastery of these protocols is essential for network engineers, developers, and cybersecurity professionals to design, manage, and secure modern digital infrastructures. The ongoing evolution of these protocols also reflects the growing demands for speed, reliability, and security in global communications.

7. Reference

[1] Microsoft. (2024). The OSI Model. Microsoft Learn. Retrieved from

https://learn.microsoft.com/en-us/troubleshoot/windows-client/networking/tcpip-addressing-and-subnetting

[2] Geeksforgeeks : https://www.geeksforgeeks.org/differences-between-tcp-and-udp/