Prepared by: Nandini Goyal
Intern at Celebal Technologies
Department: Cloud Infra & Security
Date: 6 June 2025

R&D Document

Working of TCP, UDP, HTTP, HTTPs & ICMP Protocol

Table of Contents

Introduction

This document explores the working and functionality of key networking protocols: TCP, UDP, HTTP, HTTPS, and ICMP. These protocols operate at different layers of the OSI and TCP/IP models, enabling various aspects of network communication, from reliable data transfer to web browsing and network diagnostics.

Overview

The protocols covered in this document serve distinct purposes within the networking ecosystem:

- **TCP (Transmission Control Protocol)**: A reliable, connection-oriented protocol at the Transport Layer, ensuring error-free and ordered data delivery, ideal for applications like web browsing and email.
- **UDP (User Datagram Protocol)**: A lightweight, connectionless protocol at the Transport Layer, prioritizing speed over reliability, suitable for real-time applications like streaming and gaming.
- **HTTP (HyperText Transfer Protocol)**: An Application Layer protocol for transferring hypertext, enabling web communication through a request-response model.
- **HTTPS (HyperText Transfer Protocol Secure)**: A secure version of HTTP, using encryption to protect web data, critical for secure transactions like online banking.
- **ICMP (Internet Control Message Protocol)**: A Network/Internet Layer protocol for diagnostic and error-reporting functions, supporting tools like ping and traceroute. This document details the working mechanisms, features, use cases, advantages, disadvantages, and future trends of these protocols, providing a comprehensive understanding of their roles in networking.

1. TCP (Transmission Control Protocol)

- **Layer:** Transport Layer (OSI/TCP-IP)
- **Functionality:** TCP is a connection-oriented protocol that ensures reliable, ordered, and error-checked data delivery.
- **Working:**
  - Establishes a connection using a three-way handshake (SYN, SYN-ACK, ACK).
  - Segments data, adds sequence numbers, and reassembles it at the destination.

- Uses acknowledgments to confirm receipt and retransmits lost packets.
- Implements flow control (via windowing) and congestion control to prevent network overload.

- **Key Features:**
  - Reliable: Ensures no data loss or duplication.
  - Ordered: Delivers data in the correct sequence.
  - Connection-Oriented: Maintains a connection until data transfer is complete.
- **Use Cases:** Web browsing (HTTP/HTTPS), email (SMTP), file transfer (FTP).
- Advantages:
  - Guarantees data delivery and order.
  - Handles error correction and retransmission.
- **Disadvantages:**
  - Slower due to overhead from reliability mechanisms.
  - Higher resource usage compared to UDP.

2. UDP (User Datagram Protocol)

- **Layer:** Transport Layer (OSI/TCP-IP)
- **Functionality:** UDP is a connectionless protocol that provides fast, lightweight data transfer without reliability guarantees.
- **Working:**
  - Sends datagrams without establishing a connection.
  - Adds minimal headers (source/destination ports, length, checksum) to data.
  - Does not guarantee delivery, order, or error correction.
  - Relies on higher layers or applications for reliability if needed.
- **Key Features:**
  - **Fast:** Minimal overhead makes it quicker than TCP.
  - **Connectionless:** No handshake or connection maintenance.
  - **Unreliable:** No retransmission or ordering of packets.
  - **Use Cases:** Video streaming, online gaming, DNS queries, VoIP.
- **Advantages:**
  - Low latency, ideal for real-time applications.

- Simple and efficient for small, frequent data transfers.

- **Disadvantages:**
  - No reliability or error recovery.
  - Data may arrive out of order or be lost.

3. HTTP (HyperText Transfer Protocol)

- **Layer:** Application Layer (OSI/TCP-IP)
- **Functionality:** HTTP is a protocol for transferring hypertext, enabling web browsing by facilitating communication between clients (browsers) and servers.
- **Working:**
  - Uses a request-response model: clients send HTTP requests (e.g., GET, POST), and servers respond with status codes (e.g., 200 OK, 404 Not Found) and content.
  - Operates over TCP for reliable data transfer.
  - Stateless: Each request is independent unless managed by cookies or sessions.
  - Supports methods like GET (retrieve data), POST (send data), PUT (update data), and DELETE.
- **Key Features:**
  - Text-Based: Uses human-readable messages.
  - Flexible: Supports various data types (HTML, images, JSON).
  - Use Cases: Web page loading, API communication.
- **Advantages:**
  - Simple and widely supported.
  - Enables dynamic web content delivery.
- **Disadvantages:**
  - Unencrypted, vulnerable to interception.
  - Statelessness requires additional mechanisms for session management.

4. HTTPS (HyperText Transfer Protocol Secure)

- **Layer:** Application Layer (OSI/TCP-IP)

- **Functionality:** HTTPS is the secure version of HTTP, using encryption to protect data during web communication.
- **Working:**
  - Builds on HTTP but uses SSL/TLS protocols for encryption and authentication.
  - Establishes a secure connection via a handshake, where the server presents a certificate verified by a Certificate Authority (CA).
  - Encrypts data to ensure confidentiality and integrity.
  - Operates over TCP, typically on port 443.
- **Key Features:**
  - **Secure:** Protects against eavesdropping and man-in-the-middle attacks.
  - **Authenticated:** Verifies server identity via certificates.
  - **Use Cases:** Online banking, e-commerce, secure API communication.
- **Advantages:**
  - Enhances user trust with secure data transfer.
  - Required for modern web standards (e.g., SEO, browser security).
- **Disadvantages:**
  - Slightly slower due to encryption overhead.
  - Requires certificate management and costs.

5. ICMP (Internet Control Message Protocol)

- **Layer:** Network Layer (OSI), Internet Layer (TCP-IP)
- **Functionality:** ICMP is used for diagnostic and error-reporting functions in IP networks.
- **Working:**
  - Sends control messages (e.g., error notifications, diagnostic queries) between devices.
  - Does not carry application data but supports network operations.
  - Common messages include Echo Request/Reply (used by ping) and Destination Unreachable.
  - Operates over IP, encapsulated in IP packets.
- **Key Features:**
  - **Diagnostic:** Helps troubleshoot network issues.
  - **Error Reporting:** Notifies senders of issues like unreachable hosts.

- **Use Cases:** Network troubleshooting (ping, trace route), error reporting.

- **Advantages:**
  - Essential for network diagnostics and monitoring.
  - Lightweight with minimal overhead.
- **Disadvantages:**
  - Limited to control messages, not for data transfer.
  - Can be blocked by firewalls, limiting diagnostic capabilities.

Comparison of TCP and UDP

Points	Transmission Control Protocol (TCP)	User Datagram Protocol or Universal Datagram Protocol (UDP)
Connection	Transmission Control Protocol is a connection-oriented protocol.	User Datagram Protocol is a connectionless protocol.
Function	As a message makes its way across the internet from one computer to another. This is connection based.	UDP is also a protocol used in message transport or transfer. This is not connection based which means that one program can send a load of packets to another and that would be the end of the relationship.
Usage	TCP is suited for applications that require high reliability, and transmission time is relatively less critical.	UDP is suitable for applications that need fast, efficient transmission, such as games. UDP's stateless nature is also useful for servers that answer small queries from huge numbers of clients.

Use by other protocols	HTTP, HTTPs, FTP, SMTP, Telnet	DNS, DHCP, TFTP, SNMP, RIP, VOIP.
Ordering of data packets	TCP rearranges data packets in the order specified.	UDP has no inherent order as all packets are independent of each other. If ordering is required, it has to be managed by the application layer.
Speed of transfer	The speed for TCP is slower than UDP.	UDP is faster because error recovery is not attempted. It is a "best effort" protocol.
Reliability	There is absolute guarantee that the data transferred remains intact and arrives in the same order in which it was sent.	There is no guarantee that the messages or packets sent would reach at all.
Header Size	TCP header size is 20 bytes	UDP Header size is 8 bytes.
Common Header Fields	Source port, Destination port, Check Sum	Source port, Destination port, Check Sum
Streaming of data	Data is read as a byte stream, no distinguishing indications are transmitted to signal message (segment) boundaries.	Packets are sent individually and are checked for integrity only if they arrive. Packets have definite boundaries which are honored upon receipt, meaning a read operation at the receiver socket will yield an entire message as it was originally sent.

Weight	TCP is heavy-weight. TCP requires three packets to set up a socket connection, before any user data can be sent. TCP handles reliability and congestion control.	UDP is lightweight. There is no ordering of messages, no tracking connections, etc. It is a small transport layer designed on top of IP.
Data Flow Control	TCP does Flow Control. TCP requires three packets to set up a socket connection, before any user data can be sent. TCP handles reliability and congestion control.	UDP does not have an option for flow control
Error Checking	TCP does error checking and error recovery. Erroneous packets are retransmitted from the source to the destination.	UDP does error checking but simply discards erroneous packets. Error recovery is not attempted.
Fields	1. Sequence Number, 2. AcK number, 3. Data offset, 4. Reserved, 5. Control bit, 6. Window, 7. Urgent Pointer 8. Options, 9. Padding, 10. Check Sum, 11. Source port, 12. Destination port	1. Length, 2. Source port, 3. Destination port, 4. Check Sum
Acknowledgement	Acknowledgement segments	No Acknowledgment
Handshake	SYN, SYN-ACK, ACK	No handshake (connectionless protocol)

Future Aspects

- **TCP/UDP:** TCP will evolve with optimisations for high-speed networks (e.g., QUIC, which combines TCP reliability with UDP speed). UDP will remain critical for real-time applications like AR/VR and IoT.
- **HTTP/HTTPS:** HTTP/3, built on QUIC, will enhance web performance with faster, more reliable connections. HTTPS will adopt post-quantum cryptography to counter future quantum computing threats.
- **ICMP:** Enhanced ICMP versions (e.g., ICMPv6) will support IPv6 networks, improving diagnostics for IoT and 5G/6G environments.

Conclusion

TCP, UDP, HTTP, HTTPS, and ICMP are foundational protocols enabling diverse network functions, from reliable data transfer to secure web communication and diagnostics. Their adaptability ensures relevance in future networking technologies like 5G, IoT, and quantum communication.

References

- Study-CCNA. (n.d.). OSI & TCP/IP Models. Retrieved from https://study-ccna.com/osi-tcp-ip-models/
- Diffen. (n.d.). TCP vs UDP. Retrieved from https://www.diffen.com/difference/TCP_vs_UDP
- Webopedia. (n.d.). The 7 Layers of the OSI Model. Retrieved from https://www.webopedia.com/definitions/7-layers-of-osi-model/
- Kurose, J. F., & Ross, K. W. (2020). Computer Networking: A Top-Down Approach (8th ed.). Pearson.
- Cisco Networking Academy. (2023). Introduction to Networks: Protocols and Models.