# WORKING of SEVERAL PROTOCOLS

# ABSTRACT

The R&D document discusses TCP, UDP, HTTP, HTTPS, and ICMP protocols, which are the important transport and application protocols for internal communication. It explains how each protocol works, shows differences in performance,and an example workflow to show how they are integrated in the TCP/IP model. An explanation of the differences between TCP and UDP is also there to make it understandable.

# OBJECTIVE

* To learn the internal operation of TCP, UDP, HTTP, HTTPS, and ICMP protocols.
* To recognize the main differences between TCP and UDP.
* To learn how HTTP/HTTPS protocols operate over TCP and how ICMP is placed in the Internet Layer.

# INTRODUCTION

On the internet, protocols play the main role in directing how devices transfer data to each other. TCP and UDP are part of the Transport Layer; TCP is responsible for reliable transfer, and UDP for connectionless and instant data transmission over the web. When it comes to Application Layer, HTTP and HTTPS are the main protocols needed for using the web. ICMP takes care of error reporting at the Internet Layer and monitors network performance through ping and traceroute. To sum up, these protocols ensure efficient, reliable, and secure communication all the way from one system to another over the internet.

## TRANSMISSION CONTROL PROTOCOL (TCP)

**Layer:** Transport Layer

TCP protocol helps two systems on a network to exchange their data securely and properly. It forms a connection for a session using a three-way handshake and allows checking for errors and retransmitting packets.

Working:

* The process starts with SYN, continues with SYN-ACK and finishes with ACK known as three way handshake.
* Split the data into segments and give each portion a sequence number.
* Make sure the information is guaranteed to be delivered by using acknowledgments.
* It retransmits lost packets and also reorder the packets if out of sequence.
* The connection closes with the help of both FIN-ACK termination.

Example:

* No matter if the connection is steady or not, TCP makes certain that all packets are received in their correct order and none are missing when downloading a file.

## User Datagram Protocol (UDP)

**Layer:** Transport Layer

UDP is used to send small chunks of data called datagrams in a connectionless way. It has a quicker speed but cannot be trusted, since there is guarantee for deliveries or correct order.

Working:

* Here data is broken down into smaller datagrams.
* Every datagram is sent on its own without being grouped with others.
* There is no handshaking or any retransmission in this kind of data transfer.
* It is most convenient for real time applications that need quicker speed.

Example:

To watch videos online or play games over the internet, people accept occasional data loss but low latency is essential.

## TCP vs UDP

| **Feature** | **TCP** | **UDP** |
| --- | --- | --- |
| Connection | Connection-oriented | Connectionless |
| Reliability | Guaranteed | Not guaranteed |
| Speed | Slower | Faster |
| Use cases | File transfer, email,  HTTP | Video call, DNS, live games |
| Header size | 20 Bytes | 8 Bytes |
| Acknowledgement | Yes | No |
| Congestion control | Yes | No |

HYPERTEXT TRANSFER PROTOCOL (HTTP)

**Layer:** Application Layer

**Underlying Protocols:** TCP:Port-80

When we use HTTP, it’s an application layer protocol that passes HTML, CSS, JavaScript, images, and text between a web browser and a web server. Since HTTP is a stateless and unprotected protocol, it sends data in plain text and each request is dealt separately, so it is less secure than HTTPS.

Working:

* Using the method GET(to fetch data) or POST(to send data), the client sends an HTTP request to the server.
* They communicate through Port 80 of TCP.
* The server works on the request and answers back with an HTTP response that has:
* You can see a status code (e.g.,200 OK, 404 Not Found).
* The requested content e.g.,JSON,HTML etc..
* Because sessions are not stored with HTTP, so each HTTP request operates separately unless you add cookies or tokens.

Example:

Indicating http://example.com in a browser sets up an HTTP GET request. As a response, the server gives the webpage to the browser. If the web page cannot be located or found, a 404 Not Found error is sent.

### HYPERTEXT TRANSFER PROTOCOL SECURE (HTTPS)

**Layer:** Application Layer  
**Underlying Protocols:** TCP + TLS/SSL:port 443

HTTPS is the secure version of HTTP, meant to transfer information on the web from a client (browser) to a server. It applies SSL/TLS encryption so that users can Communicate on a network, ensuring their data privacy and authentication.

Ensuring privacy for sensitive information such as passwords, personal information, and bank details is very important.

Working:

* Visiting an https:// website causes the browser and the server to carry out a TLS handshake.
* The server sends an SSL certificate, which proves that its identity is correct.
* As soon as the certificate is verified, the client and server will use encryption keys for a secure exchange of data.
* All after that, HTTP requests and replies are encrypted with these keys and are sent securely on TCP Port 443.
* If anybody attempts to intercept the information, it will still be unreadable due to the encryption.

Example:

Before your details reach the server, the website encrypts your username and password when you log in using https://www.amazon.in. No matter if someone intercepts your network data, they won’t be able to see or take your information.

A padlock icon in the browser’s address bar is a sign that the site is protected with HTTPS.

## INTERNET CONTROL MESSAGE PROTOCOL (ICMP)

**Layer:** Network Layer

ICMP is a network layer protocol which is not used for data transfer but it does help in detecting network issues. It serves for identifying errors, monitoring networks, and doing tests to check their status and traffic flows.

Like IP, it does its work separate from ports, and is essential for important tools such as ping and traceroute.

Working:

* ICMP does not send user data, but it sends control message that help report problems such as:
* Not able to find a destination.
* Packet timeout.
* When lots of traffic fills the network or a routing error arises.
* ICMP messages get triggered automatically whenever they face an error.
* One of the well-known ICMP types is:
* Echo Request/Reply used in ping.
* The time exceeded(used in traceroute).
* Since ICMP runs over IP, without using TCP or UDP and thus port numbers are not used.

Example:

The command ping [google.com](http://google.com) sends ICMP Echo requests to google server. When the server is reachable or found, it answers with ICMP Echo Reply packets. It lets determine whether the server works and, if so, how much time it takes to respond to requests.

If all the messages go unanswered, it could mean that the server is not available or the network is not working.

| Protocol | Layer | Reliable | Connection | Example |
| --- | --- | --- | --- | --- |
| HTTP | Application | Yes via TCP | Yes | Website access(non-secure) |
| HTTPS | Application | Yes via TCP | Yes | Online banking |
| ICMP | Network | No | NO | Ping,network diagnostic |