4.1 Application Layer services:
- **4.1.1 concepts of email**
- **4.1.2 working of email account and services**
- **4.1.3 URL and URL types (Absolute, Relative)**

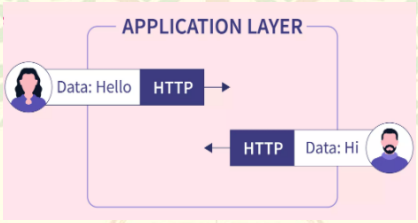
4.2 Case study of email:
- **4.2.1 From sender to receiver (Mailer, Mail Server, Mailbox)**
- **4.2.2 Functionality and use of protocols at different layers**

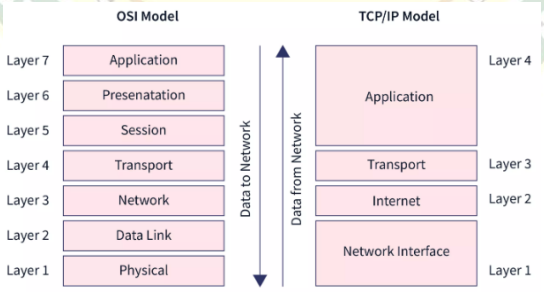
4.3 Case study of locating Website:
- **4.3.1 URL and locating URL**
- **4.3.2 Steps and protocols involved in accessing URL**
- **4.3.3 Concepts of search engine and purpose.**

4.1 Application Layer services:

What is the Application Layer?

The application layer is the topmost layer of the OSI model and the TCP/IP model. In TCP/IP model, the application layer is formed by combining the top three layers, i.e., the application layer, the presentation layer, and the session layer. It is the layer closest to the end-user, implying that the application layer and the end-user can interact directly with the software application.





It does not provide service to other layers because it is the topmost layer. The Application layer uses Transport and any levels below it to communicate with or transfer data to a remote host.

Consumers frequently require protocols from the Application Layer. One of the most often used application protocols is HTTP (HyperText Transfer Protocol), the foundation for the World Wide Web. Domain Name System (DNS), Simple Mail Transfer Protocol (SMTP), File Transfer Protocol (FTP), Trivial File Transfer Protocol (TFTP), and TELNET are some of the protocols used in the application layer.

What Are the Services Provided by the Application Layer?

The application layer provides the following services.

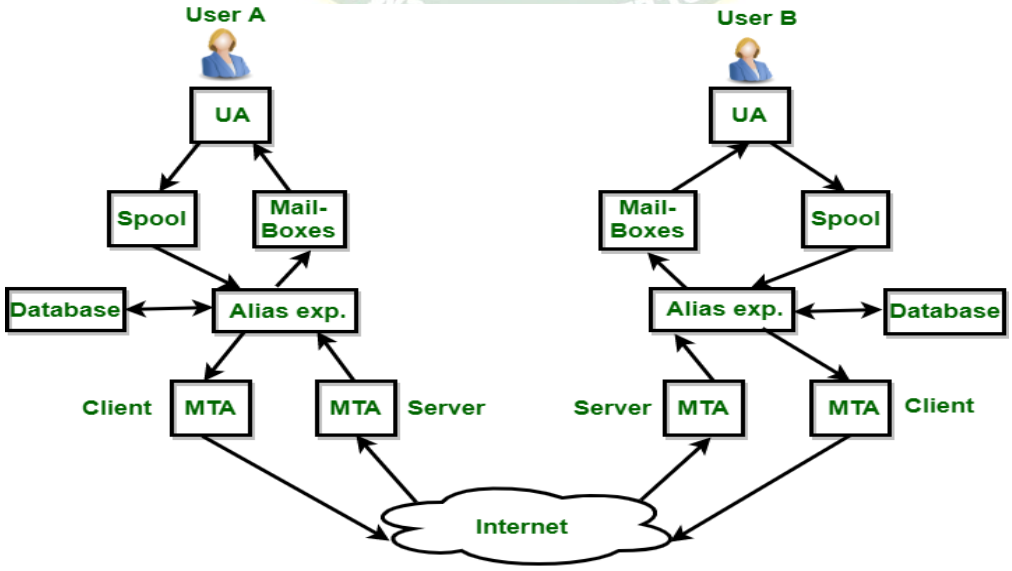
1. The application layer guarantees that the receiver is recognized, accessible, and ready to receive data from the sender.
2. It enables authentication between devices for an extra layer of network security.
3. It determines the protocol and data syntax rules at the application level.
4. The protocols of the application layer also define the basic syntax of the message being forwarded or retrieved.
5. It also checks whether the sender's computer has the necessary communication interfaces, such as an Ethernet or Wi-Fi interface.
6. Finally, the data on the receiving end is presented to the user application.

4.1.1 Concepts of email

Electronic Mail (e-mail) is one of most widely used services of Internet. This service allows an Internet user to send a **message in formatted manner (mail)** to the other Internet user in any part of world. Message in mail not only contain text, but it also contains images, audio and videos data. The person who is sending mail is called **sender** and person who receives mail is called **recipient**. It is just like postal mail service.

Components of E-Mail System :
The basic components of an email system are : User Agent (UA), Message Transfer Agent (MTA), Mail Box, and Spool file. These are explained as following below.

1. **User Agent (UA) :**
   The UA is normally a program which is used to send and receive mail. Sometimes, it is called as mail reader. It accepts variety of commands for composing, receiving and replying to messages as well as for manipulation of the mailboxes.
2. **Message Transfer Agent (MTA) :**
   MTA is actually responsible for transfer of mail from one system to another. To send a mail, a system must have client MTA and system MTA. It transfer mail to mailboxes of recipients if they are connected in the same machine. It delivers mail to peer MTA if destination mailbox is in another machine. The delivery from one MTA to another MTA is done by Simple Mail Transfer Protocol.



3. **Mailbox :**
   It is a file on local hard drive to collect mails. Delivered mails are present in this file. The user can read it delete it according to his/her requirement. To use e-mail system each user must have a mailbox . Access to mailbox is only to owner of mailbox.

4. **Spool file :**
   This file contains mails that are to be sent. User agent appends outgoing mails in this file using SMTP. MTA extracts pending mail from spool file for their delivery. E-mail allows one name, an **alias**, to represent several different e-mail addresses. It is known as **mailing list**, Whenever user have to sent a message, system checks recipients's name against alias database. If mailing list is present for defined alias, separate messages, one for each entry in the list, must be prepared and handed to MTA. If for defined alias, there is no such mailing list is present, name itself becomes naming address and a single message is delivered to mail transfer entity.

Services provided by E-mail system :

- **Composition –**
  The composition refer to process that creates messages and answers. For composition any kind of text editor can be used.
- **Transfer –**
  Transfer means sending procedure of mail i.e. from the sender to recipient.
- **Reporting –**
  Reporting refers to confirmation for delivery of mail. It help user to check whether their mail is delivered, lost or rejected.
- **Displaying –**
  It refers to present mail in form that is understand by the user.
- **Disposition –**
  This step concern with recipient that what will recipient do after receiving mail i.e save mail, delete before reading or delete after reading.

Advantages of email:

1. Convenient and fast communication with individuals or groups globally.
2. Easy to store and search for past messages.
3. Ability to send and receive attachments such as documents, images, and videos.
4. Cost-effective compared to traditional mail and fax.
5. Available 24/7.

Disadvantages of email:

1. Risk of spam and phishing attacks.
2. Overwhelming amount of emails can lead to information overload.
3. Can lead to decreased face-to-face communication and loss of personal touch.

4. Potential for miscommunication due to lack of tone and body language in written messages.
5. Technical issues, such as server outages, can disrupt email service.
6. It is important to use email responsibly and effectively, for example, by keeping the subject line clear and concise, using proper etiquette, and protecting against security threats.

1. Email Protocols

Protocols are standardized rules that govern email transmission and retrieval.

- **SMTP (Simple Mail Transfer Protocol)**
  - Used to **send** emails from a client to a server or between servers.
  - Port: 25 (or 587 for secure sending)
- **POP3 (Post Office Protocol version 3)**
  - Used to **retrieve** emails from the server to the client.
  - Downloads and often **deletes** messages from the server.
  - Port: 110 (or 995 for secure)
- **IMAP (Internet Message Access Protocol)**
  - Also used to **retrieve** emails but keeps messages on the server.
  - Allows **synchronization** across multiple devices.
  - Port: 143 (or 993 for secure)

---

2. Email Components

- **Email Client**: Software used to read, write, and send emails (e.g., Outlook, Thunderbird, Apple Mail).
- **Mail Server**: Hosts mailboxes and handles mail delivery (e.g., Microsoft Exchange, Postfix).
- **Email Address**: Identifies the sender/receiver, typically in the format: username@domain.com

---

3. Structure of an Email Message

- **Header**: Contains metadata like:
  - From
  - To
  - Subject
  - Date
  - CC/BCC

- **Body**: Main content of the message (can be plain text or HTML)
- **Attachments**: Files included with the message

---

4. DNS and Email

- **MX (Mail Exchange) Records** in DNS point to the mail servers responsible for receiving email on behalf of a domain.

---

5. Security in Email

- **TLS/SSL**: Encrypts emails in transit.
- **SPF (Sender Policy Framework)**: Prevents spoofing by verifying sending servers.
- **DKIM (DomainKeys Identified Mail)**: Uses cryptographic authentication.
- **DMARC (Domain-based Message Authentication, Reporting, and Conformance)**: Combines SPF and DKIM to protect domains from spoofing.
- **Spam Filters**: Use rules and AI to block junk email.

---

6. Email Storage and Management

- Emails are stored in **mailboxes** on the mail server.
- Managed via:
  - **Local storage** (POP3)
  - **Server-side storage** (IMAP)

---

7. Webmail vs Email Client

Feature	Webmail	Email Client
Access	Browser-based (e.g., Gmail)	Installed application
Storage	On server	Local or synced storage
Flexibility	Accessible anywhere	Advanced features, offline use

4.1.2 working of email account and services

An **email account** allows a user to send, receive, store, and manage email messages over the Internet. This involves **email clients**, **servers**, **protocols**, and **network infrastructure**.

Components of an Email:

1. **Sender:** The sender creates an email in which he records the information that needs to be transferred to the receiver.

2. **Receiver:** The receiver gets the information sent by the sender via email.
3. **Email address:** An email address is just like a house address where the communication arrives for the sender and receiver and they communicate with each other.
4. **Mailer:** The mailer program contains allows the ability to read, write, manage and delete the emails like Gmail, Outlook, etc.
5. **Mail Server:** The mail server is responsible for sending, receiving, managing, and recording all the data proceeded by their respective mail programs and then processing them to their respective users.
6. **SMTP:** SMTP stands for Simple mail transfer protocol. SMTP basically uses the internet network connection to send and receive email messages over the Internet.

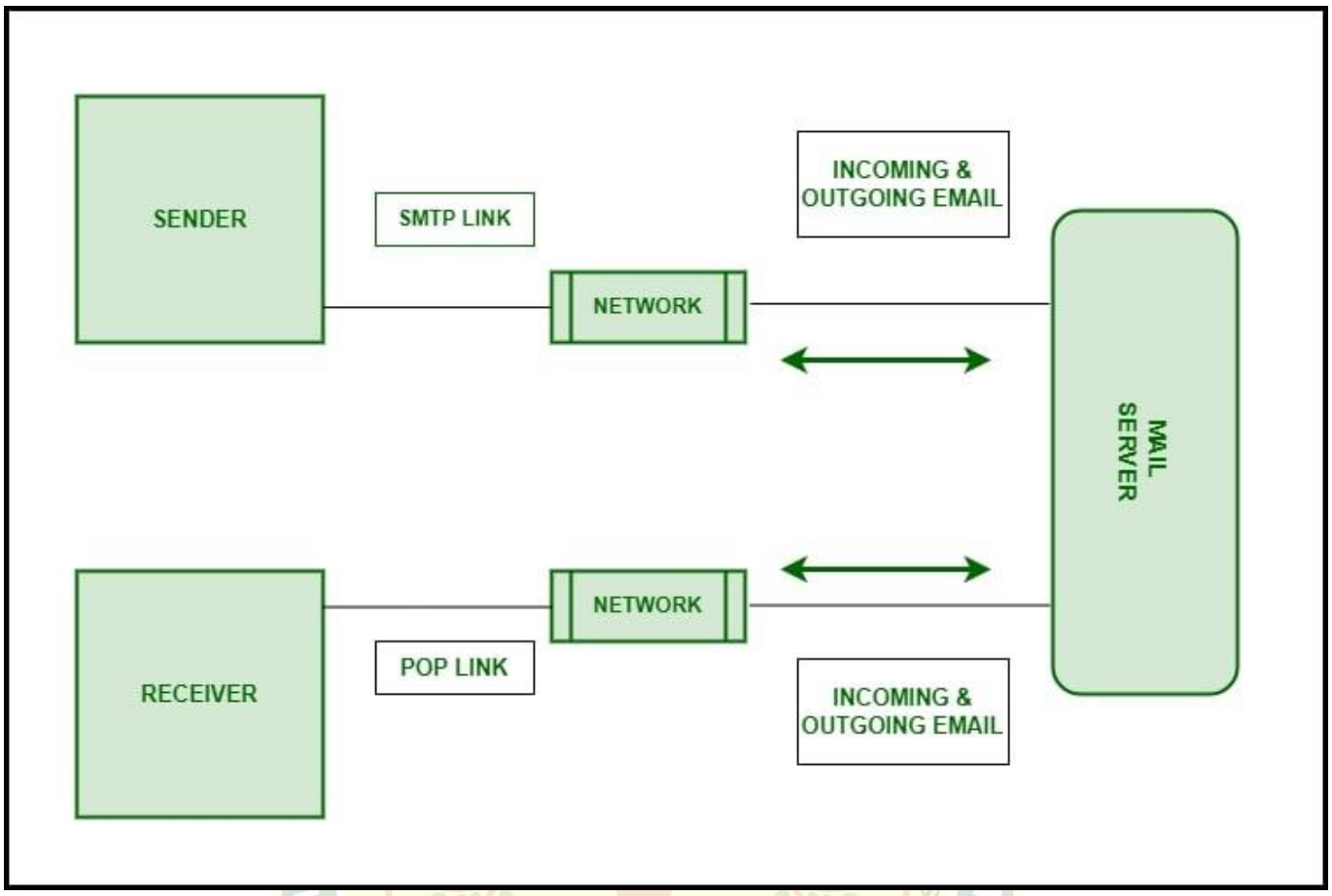
Protocols of Email:

Emails basically use two types of standard protocols for communication over the Internet. They are:-

1. **POP:** POP stands for post office protocol for email. Similar to a post office, our approach is just to drop the email over the service mail provider and then leave it for services to handle the transfer of messages. We can be even disconnected from the Internet after sending the email via POP. Also, there is no requirement of leaving a copy of the email over the web server as it uses very little memory. POP allows using concentrate all the emails from different email addresses to accumulate on a single mail program. Although, there are some disadvantages of POP protocol like the communication medium is unidirectional, i.e it will transfer information from sender to receiver but not vice versa.
2. **IMAP:** IMAP stands for Internet message access protocol. IMAP has some special advantages over POP like it supports bidirectional communication over email and there is no need to store conversations on servers as they are already well-maintained in a database. It has some advanced features like it tells the sender that the receiver has read the email sent by him.

Working of Email:

1. When the sender sends the email using the mail program, then it gets redirected to the simple mail transfer protocol which checks whether the receiver's email address is of another domain name or it belongs to the same domain name as that of the sender (Gmail, Outlook, etc.). Then the email gets stored on the server for later purposes transfer using POP or IMAP protocols.
2. If the receiver has another domain name address then, the SMTP protocol communicates with the DNS (domain name server) of the other address that the receiver uses. Then the SMTP of the sender communicates with the SMTP of the receiver which then carries out the communication and the email gets delivered in this way to the SMTP of the receiver.
3. If due to certain network traffic issues, both the SMTP of the sender and the receiver are not able to communicate with each other, the email to be transferred is put in a queue of the SMTP of the receiver and then it finally gets receiver after the issue resolves. And if due to very bad circumstances, the message remains in a queue for a long time, then the message is returned back to the sender as undelivered.



From Sender to Receiver:

The sender first needs the email address of the receiver to send the information to be communicated via email. When the sender writes all the information in the email along with the email address of the receiver and clicks on the send button, the mail program transfers the message to the MTA (Mail Transfer Agent) which is transferred from the local computer of the sender to the mail server via the SMTP protocol.

Then the webmail server looks out for the similar mail transfer agent of the receiver and locates it whether it is using the same DNS (domain name server) or a different service. The DNS looks for the mail exchanger service of the receiver. Now, the SMTP protocol transfers the message between both mail servers through their mailing agents. Then the receiver's MTA finally transfers this message to the receiver's local computer.

In case, the receiver uses POP protocol then when he receives the email, then the copy of the email at the webserver will get deleted. And if he uses IMAP then the copy of the email gets stored on the webserver and it can be changed at any time by the user.

📧 **What is an Email Account?**

An **email account** is a service that allows a user to send, receive, and manage electronic messages (emails) over a network like the Internet.

📌 **Example**:
user@example.com

- user = username
- example.com = domain name (managed by the mail server)

⚙️ How an Email Account Works – Step by Step

1. User Login

- The user logs in using:
  - **Webmail** (e.g., Gmail, Yahoo Mail)
  - **Email client** (e.g., Microsoft Outlook, Thunderbird)
- The device connects to a **mail server** using the Internet.

---

2. Composing and Sending Email

- User writes a message and clicks **Send**.
- The message is sent to an **SMTP server** (outgoing mail server).

3. SMTP – Sending the Email

- **SMTP (Simple Mail Transfer Protocol)** forwards the email to the **recipient's mail server**.
- The recipient's mail server is found using **DNS** (Domain Name System), which locates the correct **MX (Mail Exchange) record**.

---

4. Email Travels Across the Network

- The email is transmitted over the Internet to the recipient's **mail server** (e.g., Yahoo, Outlook, Gmail).
- The server stores the message in the recipient's **mailbox**.

---

5. Receiving the Email

- The recipient logs in to check mail.
- The email client (or webmail) retrieves the message using:
  - **IMAP (Internet Message Access Protocol)** – Views/syncs email without downloading.
  - **POP3 (Post Office Protocol 3)** – Downloads email and removes it from the server (optional).

---

6. Reading and Replying

- The recipient reads the message.
- If they reply, the whole process is repeated using SMTP to send and POP3/IMAP to receive.

🔁 Summary of Email Services and Protocols

Service/Protocol	Purpose
SMTP	Sends emails (client → server → server)
IMAP	Reads and syncs emails (keeps on server)
POP3	Downloads and may delete email from server
DNS	Finds domain and mail server via MX record
Mail Server	Stores and manages email accounts/messages
Email Client	Software to send/receive emails
Webmail	Browser-based email interface

📨 Example Scenario

Alice (alice@gmail.com) sends an email to **Bob** (bob@yahoo.com):

1. Alice types the email and hits Send.
2. Gmail's SMTP server sends the email using DNS to find Yahoo's server.
3. Bob's Yahoo server receives and stores the email.
4. Bob logs into Yahoo Mail and reads the email using IMAP.
5. Bob replies, and the process repeats in reverse.

🔐 Security in Email Services

- **TLS/SSL** – Encrypts emails during transmission.
- **Authentication** – Verifies sender and receiver.
- **Spam Filters** – Block unwanted emails.
- **SPF, DKIM, DMARC** – Prevent fraud, spoofing, phishing.

✅ Conclusion

The working of an email account is based on:

- **Client-server communication**
- Standard protocols: **SMTP**, **POP3**, **IMAP**
- Reliable internet infrastructure and security mechanisms

It enables fast, secure, and global communication over a computer network.

4.1.3 URL and URL types (Absolute, Relative)

URL stands for **Uniform Resource Locator**.
It is the **address** used to access resources (like web pages, images, videos, files) on the **Internet** or a **local network**.

📌 A URL tells the browser:

- **Where** to go (location)
- **What protocol** to use (e.g., HTTP, HTTPS, FTP)

🔧 **Basic Structure of a URL**

scheme://domain/path/resource

🔸 Example:

https://www.example.com/folder/page.html

Part	Description
https://	Protocol
www.example.com	Domain name
/folder/page.html	Path to resource

🔹 **Types of URLs**

There are two main types of URLs:

---

1. Absolute URL

An **absolute URL** contains the **full path** to a resource, including:

- Protocol (http, https, ftp, etc.)
- Domain name
- Full directory path

🔸 **Example:**

https://www.example.com/images/photo.jpg

📌 This link can be accessed from **any location**, and it always points to the same resource.

---

2. Relative URL

A **relative URL** refers to a resource **relative to the current page or directory**.

🔸 **Example:**

/images/photo.jpg

📌 This link depends on the location of the current file. If the current page is on https://www.example.com, then the full path becomes:

https://www.example.com/images/photo.jpg

🔚 **Difference Between Absolute and Relative URLs**

Feature	Absolute URL	Relative URL
Contains Protocol	Yes	No
Contains Domain	Yes	No
Location	Fixed – works from anywhere	Depends on current page's location
Example	https://example.com/page.html	page.html or /folder/page.html

---

URL stands for Uniform Resource Locator. Any internet location available on the server is called a web URL, web address, or website. Each website or webpage has a unique address called URL. e.g., the website of **geeksforgeeks** website has an address or URL called https://www.geeksforgeeks.org/

Basic Structure of URL

Types of URL: URL gives the address of files created for webpages or other documents like an image, pdf for a doc file, etc.

Type:

- It specifies the type of the server in which the file is located.
- **address:** It specifies the address or location of the internet server.
- **path:** It specifies the location of the file on the internet server.
- There are two types of URLs:
  - Absolute URL
  - Relative URL

- **Absolute URL**

  This type of URL contains both the domain name and directory/page path. An absolute URL gives complete location information. It begins with a protocol like "http://" and continues, including every detail. An absolute URL typically comes with the following syntax.

  Syntax:

  protocol://domain/path

  For web browsing, absolute URLs are types in the address bar of a web browser. For example, if it is related to our project page link on **geeksforgeeks** website, the URL should be mentioned as https://www.geeksforgeeks.org/gate/computer-science-projects/ this gives the complete information about the file location path.

  Note: The protocol may be of the following types.

  http://, https://, ftp://, gopher://, etc.

- **Relative URL**

  This type of URL contains the path excluding the domain name. Relative means "in relation to", and a relative URL tells a URL location on terms of the current location. Relative path is used for reference to a given link of a file that exist within the same domain.

  Let us assume a web developer setting up a webpage and want to link an image called "geeksforgeeks.jpg".

  <img src="geeksforgeeks.jpg">

  It would internally be interpreted like the following.

  <img src="./geeksforgeeks.jpg">

  The dot(.) before the "/" in the *src* attribute is a "special character". It means the location should be started from the current directory to find the file location.

4.2 Case study of email:

Here is a **simple and effective case study** of **email** that demonstrates its **real-world use, components, protocols, and impact** in a computer network environment.

Case Study: Implementation of Email System in a Corporate Office

---

Organization:

TechNova Solutions Pvt. Ltd. – A mid-sized IT company with 200 employees.

Objective:

To implement a **reliable, secure, and efficient email communication system** for internal and external business communication.

Key Requirements:

- Unique email accounts for all employees.
- Internal communication between departments.
- External communication with clients and vendors.
- Access through webmail and desktop/mobile clients.
- Spam protection and email security.
- Centralized storage and backup of emails.

Technical Setup:

Component	Description
Domain Name	`technova.com`
Email Server	Microsoft Exchange Server (self-hosted)
Protocols Used	SMTP (for sending), IMAP (for reading/syncing), POP3 (optional)
Webmail Access	Enabled via Outlook Web Access (OWA)
Email Clients	Microsoft Outlook on desktop, mobile mail apps
Security Features	SSL/TLS encryption, spam filters, SPF, DKIM, DMARC
Storage	Mailboxes with 5 GB per user, backed up nightly

How the Email System Works – Example Use Case

Scenario:

- Employee **Amit** (amit@technova.com) from HR sends an interview schedule to a candidate (**nina.khan@gmail.com**).

Step-by-Step Flow:

1. **Amit** composes the email using Microsoft Outlook.
2. Outlook connects to the company's **SMTP server**.
3. The server checks the recipient domain `gmail.com`, uses **DNS** to find the **MX record**.
4. The email is sent to Gmail's mail server via the Internet.
5. Gmail stores the email in Nina's mailbox.

6. Nina opens the email using Gmail's webmail interface (via IMAP).

Email successfully delivered using standard email protocols across different domains.

Security Measures in Place

- **Spam filters** block junk emails.
- **Antivirus scanner** checks attachments.
- **TLS encryption** secures email in transit.
- **Authentication** prevents unauthorized access.

Results and Impact

Area	Outcome
Communication Speed	Instant messaging across teams and departments
Productivity	Improved due to centralized communication
Customer Response	Faster and more reliable replies to clients
Security	Reduced phishing and spam emails by over 90%
Cost	Lower than using third-party mailing services over time

Functionality and Use of Protocols at Different Layers

The use of protocols across a layered network model, such as the TCP/IP model, ensures that data is formatted, addressed, transmitted, and received reliably.

1. Application Layer

- Functionality:

  This layer enables software applications to communicate with each other and exchange data over the network.

- Use:

Protocols at this layer define the rules for specific services.

- HTTP (Hypertext Transfer Protocol): Used by web browsers and servers to transmit web pages and other hypermedia documents.
- DNS (Domain Name System): Translates human-readable domain names (like google.com) into machine-readable IP addresses.
- FTP (File Transfer Protocol): Used for uploading and downloading files between computers.
- SMTP (Simple Mail Transfer Protocol): Manages email communication.

2. Transport Layer

- Functionality:

Provides reliable or unreliable data delivery between applications on different hosts.

- Use:

Protocols manage data segmentation, reassembly, and error control.

- TCP (Transmission Control Protocol): Guarantees that data is delivered in the correct order and without errors, establishing a reliable, two-way connection.
- UDP (User Datagram Protocol): A faster, less reliable protocol that does not require a full connection, used for real-time applications like online gaming, video streaming, and DNS lookups.

3. Network Layer (or Internet Layer)

- Functionality:

Handles the logical addressing (IP addresses) and routing of data packets across different networks.

- Use:

Protocols determine the path for data packets to travel from the source to the destination.

- IP (Internet Protocol): The core protocol for addressing and routing packets across the internet.
- ICMP (Internet Control Message Protocol): Used for sending error messages and operational information between devices, such as ping requests.

4. Network Access Layer (or Data Link Layer)

- Functionality:

Manages communication and data transfer between devices on the same physical network segment (LAN).

- Use:

Protocols handle physical addressing and error detection.

- Ethernet: A common protocol for local area networks that defines how devices access the network medium and use MAC addresses to identify devices.
- ARP (Address Resolution Protocol): Maps an IP address to a physical MAC (hardware) address for devices within a local network.

4.3 Case Study: Locating a Website

This case study explains how a user accesses a website, including how URLs work, the steps involved, and the role of search engines.

- Scenario:

A user wants to visit the website https://www.nationalgeographic.com to read an article.

4.3.1 URL and Locating URL

What is a URL?

A Uniform Resource Locator (URL) is the web address used to locate a website or resource on the internet.

Example URL Breakdown:

https://www.nationalgeographic.com/articles/explore-wildlife

Part	Meaning
https://	Protocol (secure HTTP)
www.nationalgeographic.com	Domain name (website address)
/articles/explore-wildlife	Path to specific page

So, this URL tells the browser:

Use HTTPS to go to the server at nationalgeographic.com, then fetch the page at /articles/explore-wildlife.

4.3.2 Steps and Protocols Involved in Accessing a URL

Steps Involved:

User types the URL in the browser.
Browser checks cache to see if it has the IP address.
If not, it sends a DNS request to translate nationalgeographic.com into an IP address.
Browser initiates a TCP connection with the server using the IP.
Using HTTPS, the browser sends a GET request to the server.
The web server responds with the HTML content.
The browser renders the webpage for the user to see.

Protocols Used:

Protocol	Purpose
DNS	Converts domain name to IP address
TCP/IP	Manages data transfer between devices
HTTPS (Secure HTTP)	Encrypts and transmits the website request
SSL/TLS	Secures the connection (part of HTTPS)

4.3.3 Search Engine and Its Purpose

What is a Search Engine?

A search engine is a software system that helps users find information on the internet using keywords.

Purpose:

Allows users to discover websites without knowing the exact URL.
Ranks results by relevance using search algorithms.
Makes web navigation faster and easier.

How It Helps in Locating a Website:

Instead of typing the full URL, a user can search:
"National Geographic wildlife articles"
Google (or another engine) shows the most relevant pages with links.

Behind the Scenes:

Search engine crawlers scan websites.
Index stores website content.
Algorithms rank and show results.

****

Old year Questions:-

1. Explain Working of Email Account and it's Services.
2. Different between Absolute and Relative URL.
3. Explain Case Study of Email.
4. Explain use of protocols at different layers of email.
5. What is Application layer? Explain it's services.