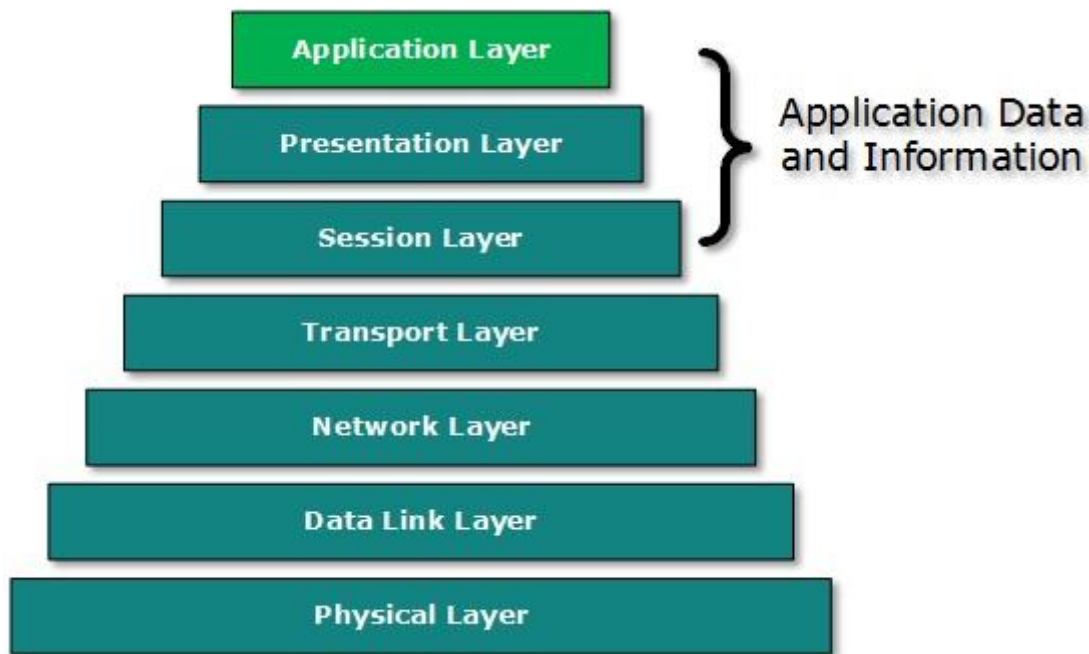


# Application Layer Introduction

Application layer is the highest most layer in OSI and TCP/IP layered model and. This layer exists in both layered Models because of its significance which is interacting with user and user applications. This layer is for applications which are involved in communication system.

A user may or may not directly interact with these applications. Application layer is where the actual communication is initiated and reflects. Because this layer is on the top of the layer stack it does not serve any other layers. Application layer takes the help of Transport and all layers below it to communicate or transfer its data to the remote host.

When an application layer protocol wants to communicate with its peer application layer protocol on remote hosts it hands over the data or information to the Transport layer. The transport layer does the rest of the things with help of all layers below it.



[Image: Application Layer]

There's an ambiguity in understanding Application Layer and its protocol. Not every user application can be put into Application Layer. Only application which interacts with the communication system. For example, a designing software or text-editor cannot be considered as application layer programs.

On the other hand when we use a Web Browser, which is actually using HTTP (Hyper Text Transfer Protocol) to interact with the network. So in this case, HTTP is Application Layer protocol which we take into consideration when we study layered models.

Another example is File Transfer Protocol, which helps a user to transfer a text based or binary file across the network. A user can use this protocol in either GUI based software like FileZilla or CuteFTP and the same user can use FTP in Command Line mode.

So it is not important what software you use, it the protocol which is considered at Application Layer used by that software. DNS is a protocol which helps user application protocols like HTTP to accomplish its work.

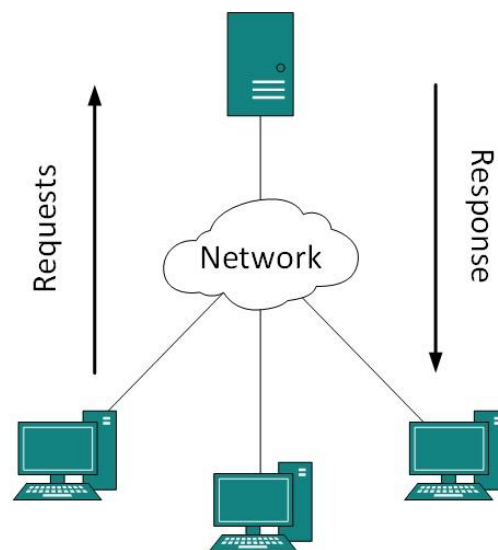
## Client Server Model

### Introduction

Two remote application process can communicate in mainly two different fashions:

- **Peer-to-peer:** Both remote processes are at same level and exchange data using some shared resource.
- **Client-Server:** One the remote process acts as Client and requests some resource from another application process acting as Server.

In client-server model, any process can act as Server or Client. This not the machine or size of the machine or its computing power which makes it server but it is the feature of serving request that makes it server.



[Image: Client Server Model]

A system can act as Server and Client simultaneously. That is, one process is acting as Server and another is acting as a client. This may also happen that both client and server processes reside on the same machine.

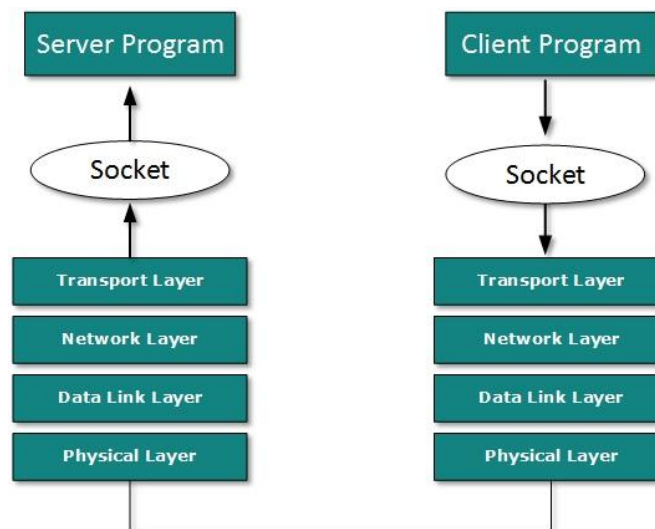
## Communication:

Two processes in client-server model can interact in various ways:

- Sockets
- Remote Procedure Calls *RPC*

## Sockets:

In this paradigm, the process acting as Server opens a socket using a well-known *or known by client* port and waits until some client request comes. The second process acting as Client also opens a socket but instead of waiting for an incoming request, client process 'requests first'.



[Image: Socket]

When the request is reached to server, it is served. It can either be an information sharing or resource request.

## Remote Procedure Call

This is a mechanism where one process interacts with another by means of procedure calls. One process *client* calls the procedure lying on remote host. The process on remote host is said to be Server. Both processes are allocated stubs. This communication happens in the following way:

- The client process calls the client stub. It passes all the parameters pertaining to program local to it.
- All parameters are then packed *marshalling* and a system call is made to send them to other side of the network is made.
- Kernel sends the data over the network and the other end receives it.
- The remote host passes data to the server stub where it is unmarshalled.

- The parameters are passed to the procedure and the procedure is then executed.
- The result is sent back to the client in the same manner.

# Application Protocols

## Introduction

There are several protocols which work for users in Application Layer. Application layer protocols can be divided into two categories broadly:

- Protocols which are used by users; email for example.
- Protocols which help and support protocols used by users; DNS for example.

Few of Application layer protocols are described below:

## Domain Name System

DNS works on Client Server model. It uses UDP protocol for transport layer communication. DNS uses hierarchical domain based naming scheme. The DNS server is configured with FQDN *Fully Qualified Domain Names* and email addresses mapped with their respective Internet Protocol addresses.

A DNS server is requested with FQDN and it responds back with the IP address mapped with it. DNS uses UDP port 53.

## Simple Mail Transfer Protocol

SMTP is used to transfer electronic mail from one user to another. This task is done by means of email client software *User Agents* the user is using. User Agents help the user to type and format the email and store it until internet is available. When an email is submitted to send, the sending process is handled by Message Transfer Agent which is normally comes inbuilt in email client software.

Message Transfer Agent uses SMTP to forward the email to another Message Transfer Agent *Server side*. While SMTP is used by end user to only send the emails, Servers normally use SMTP to send as well as receive emails. SMTP uses TCP port number 25 and 587.

Client software uses IMAP or POP protocols to receive emails.

## **File Transfer Protocol**

FTP is the most widely used protocol for file transfer over the network. FTP uses TCP/IP for communication and it works on TCP port 21. FTP works on Client/Server Model where a client requests file from Server and server sends requested resource back to the client.

FTP uses out-of-band controlling i.e. FTP uses TCP port 20 for exchanging controlling information and the actual data is sent over TCP port 21.

The client requests the server for a file. When the server receives a request for a file it opens a TCP connection for the client and transfers the file. After the transfer is complete the server closes the connection. For a second file, client requests again and the server open a new TCP connection again.

## **Post Office Protocol**

POP *version3* is a simple mail retrieval protocol used by User Agents *client email software* to retrieve mails from mail server.

When a client needs to retrieve mails from server it opens a connection with the server on TCP port 110. User can then access his mails and download them to local computer system. POP3 works in two modes. The most common mode *delete mode* is to delete the emails from remote server after they are downloaded to local machines. The second mode *keep mode* does not delete the email from mail server and gives the user an option to access mails later on mail server.

## **Hyper Text Transfer Protocol**

HTTP is the foundation of World Wide Web. Hypertext is well organized documentation system which uses hyperlinks to link to another page in the text documents. HTTP works on client server model. When a client *user* wants to access any HTTP page on the internet it initiates TCP connection to server on port 80. When the server accepts client request, the client is then authorized to access web pages.

To access web pages a client normally uses web browsers, who are responsible for initiating, maintaining and closing TCP connections. HTTP is a stateless protocol, which means the Server maintains no information about earlier requests by clients.

HTTP versions:

- HTTP 1.0 uses non persistent HTTP. At most one object can be sent over a single TCP connection.
- HTTP 1.1 uses persistent HTTP. In this version, multiple objects can be sent over a single TCP connection.

# Network Services

## Introduction

Computer systems and computerized systems help human beings to work efficiently and explore the unthinkable. When these devices are connected together to form a network, the capabilities are enhanced multi-times. Below mentioned are some basic services which any computer network can offer.

## Directory Services

These services are mapping between name and its value, which can be variable value or fixed. This software system helps store information, organizes them and provides various means of access.

- **Accounting**

In an organization, lots of users have their user name and password mapped to them. Directory Services provides means of storing these information in cryptic form and make available when requested.

- **Authentication & Authorization**

User credentials can be checked to authenticate a user at the time of login and/or periodically. User accounts can be set into hierarchical structure and their access to resources can be controlled using authorization schemes.

- **Domain Name Services**

DNS is widely used and one of the essential services on which internet works. This system maps IP addresses to domain names, which are easier to remember and recall than IP addresses. Because network works on IP addresses and humans tend to remember website names, when a website is requested, in the back-end DNS provides website's IP address which is mapped to its name.

## File Services

- **File Sharing**

One of the reasons which gave birth to networking was file sharing. File sharing enables its users to share their data with other users. User can upload the file to a specific server, which is accessible by all intended users. As an alternative, user can make its file shared on its own computer and provides access to intended users.

- **File Transfer**

This is an activity to copy or move file from one computer to another computer or to multiple computers, with help of underlying network. Networks enable its user to locate other users in the network and transfer files.

## **Communication Services**

- **Email**

Electronic mail is a communication method and something a computer user cannot live without. This is the bases of today's internet features. Email system has one or more email servers. All its users are provided with unique IDs. When a user sends email to other user, it is actually transferred between users with help of email server.

- **Social Networking**

Recent technologies have made technical life social. People, computer savvy, can find other known people or friends, can add them into their own account and can share thoughts, pictures and videos.

- **Internet Chat**

Internet chat provides instant text transfer services between two hosts. Two or more people can communicate with each other user Internet Relay Chat services which is text based. These days, voice chat and video chat are common too.

- **Discussion Boards**

Discussion boards provide a mechanism to connect multiple peoples with same interests. It enables its users to put queries, questions, suggestions which can be seen by all other users. Other may respond as well.

- **Remote Access**

This service enables user to access their own data lying on their computer, remotely. This feature is known as Remote desktop. This can be done via some remote device, e.g. mobile phone or home computer.

## **Application Services**

- **Resource Sharing**

To use resources efficiently and economically, network provides a mean to share them. This may include Servers, Printers, and Storage Media etc.

- **Databases**

This application service is one of the most important one. It stores data and information, processes it and enables its users to retrieve in very efficient way by using queries. Databases help organizations to make decisions based on statistics.

- **Web Services**

World Wide Web has become the synonym for Internet. It is used to connect to the Internet and access files and information services provided by the internet servers.