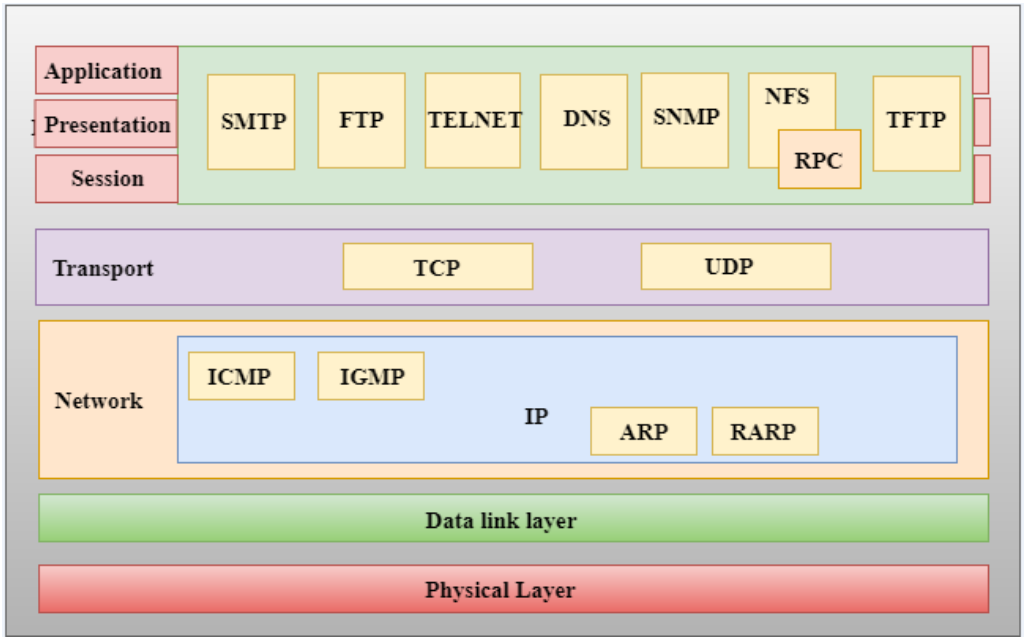
Unit :2

TCP/IP model

- The TCP/IP model was developed prior to the OSI model.
- The TCP/IP model is not exactly similar to the OSI model.
- The TCP/IP model consists of five layers: the application layer, transport layer, network layer, data link layer and physical layer.
- The first four layers provide physical standards, network interface, internetworking, and transport functions that correspond to the first four layers of the OSI model and these four layers are represented in TCP/IP model by a single layer called the application layer.
- TCP/IP is a hierarchical protocol made up of interactive modules, and each of them provides specific functionality.

Here, hierarchical means that each upper-layer protocol is supported by two or more lower-level protocols.

Functions of TCP/IP layers:



Network Access Layer

- A network layer is the lowest layer of the TCP/IP model.

- A network layer is the combination of the Physical layer and Data Link layer defined in the OSI reference model.
- It defines how the data should be sent physically through the network.
- This layer is mainly responsible for the transmission of the data between two devices on the same network.
- The functions carried out by this layer are encapsulating the IP datagram into frames transmitted by the network and mapping of IP addresses into physical addresses.
- The protocols used by this layer are ethernet, token ring, FDDI, X.25, frame relay.

Internet Layer

- An internet layer is the second layer of the TCP/IP model.
- An internet layer is also known as the network layer.
- The main responsibility of the internet layer is to send the packets from any network, and they arrive at the destination irrespective of the route they take.

Following are the protocols used in this layer are:

IP Protocol: IP protocol is used in this layer, and it is the most significant part of the entire TCP/IP suite.

Following are the responsibilities of this protocol:

- **IP Addressing:** This protocol implements logical host addresses known as IP addresses. The IP addresses are used by the internet and higher layers to identify the device and to provide internetwork routing.
- **Host-to-host communication:** It determines the path through which the data is to be transmitted.
- **Data Encapsulation and Formatting:** An IP protocol accepts the data from the transport layer protocol. An IP protocol ensures that the data is sent and received securely, it encapsulates the data into message known as IP datagram.
- **Fragmentation and Reassembly:** The limit imposed on the size of the IP datagram by data link layer protocol is known as Maximum Transmission unit (MTU). If the size of IP datagram is greater than the MTU unit, then the IP protocol splits the datagram into smaller units so that they can travel over the local network. Fragmentation can be done by the sender or intermediate router. At the receiver side, all the fragments are reassembled to form an original message.

- **Routing:** When IP datagram is sent over the same local network such as LAN, MAN, WAN, it is known as direct delivery. When source and destination are on the distant network, then the IP datagram is sent indirectly. This can be accomplished by routing the IP datagram through various devices such as routers.

ARP Protocol

- ARP stands for **Address Resolution Protocol**.
- ARP is a network layer protocol which is used to find the physical address from the IP address.
- **The two terms are mainly associated with the ARP Protocol:**
    - **ARP request:** When a sender wants to know the physical address of the device, it broadcasts the ARP request to the network.
    - **ARP reply:** Every device attached to the network will accept the ARP request and process the request, but only recipient recognize the IP address and sends back its physical address in the form of ARP reply. The recipient adds the physical address both to its cache memory and to the datagram header

ICMP Protocol

- **ICMP** stands for Internet Control Message Protocol.
- It is a mechanism used by the hosts or routers to send notifications regarding datagram problems back to the sender.
- A datagram travels from router-to-router until it reaches its destination. If a router is unable to route the data because of some unusual conditions such as disabled links, a device is on fire or network congestion, then the ICMP protocol is used to inform the sender that the datagram is undeliverable.
- An ICMP protocol mainly uses two terms:
    - **ICMP Test:** ICMP Test is used to test whether the destination is reachable or not.
    - **ICMP Reply:** ICMP Reply is used to check whether the destination device is responding or not.
- The core responsibility of the ICMP protocol is to report the problems, not correct them. The responsibility of the correction lies with the sender.

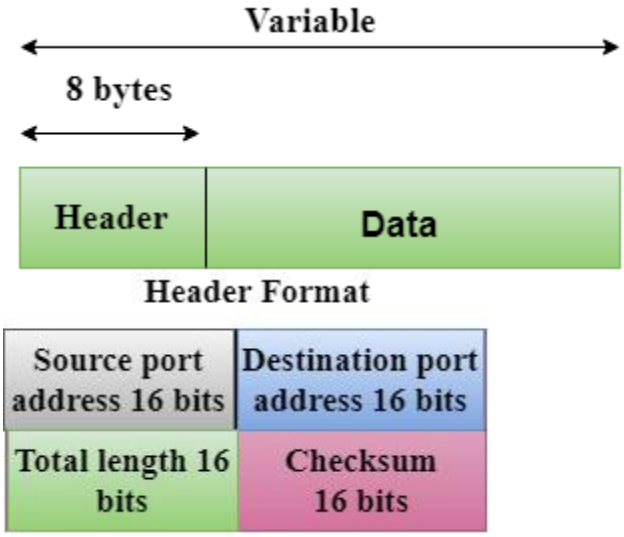
- ICMP can send the messages only to the source, but not to the intermediate routers because the IP datagram carries the addresses of the source and destination but not of the router that it is passed to.

Transport Layer

The transport layer is responsible for the reliability, flow control, and correction of data which is being sent over the network.

The two protocols used in the transport layer are **User Datagram protocol and Transmission control protocol**.

- **User Datagram Protocol (UDP)**
  - It provides connectionless service and end-to-end delivery of transmission.
  - It is an unreliable protocol as it discovers the errors but not specify the error.
  - User Datagram Protocol discovers the error, and ICMP protocol reports the error to the sender that user datagram has been damaged.
  - **UDP consists of the following fields:**
    Source port address: The source port address is the address of the application program that has created the message.
    Destination port address: The destination port address is the address of the application program that receives the message.
    Total length: It defines the total number of bytes of the user datagram in bytes.
    Checksum: The checksum is a 16-bit field used in error detection.
  - UDP does not specify which packet is lost. UDP contains only checksum; it does not contain any ID of a data segment.



- **Transmission Control Protocol (TCP)**
  - It provides a full transport layer services to applications.
  - It creates a virtual circuit between the sender and receiver, and it is active for the duration of the transmission.
  - TCP is a reliable protocol as it detects the error and retransmits the damaged frames. Therefore, it ensures all the segments must be received and acknowledged before the transmission is considered to be completed and a virtual circuit is discarded.
  - At the sending end, TCP divides the whole message into smaller units known as segment, and each segment contains a sequence number which is required for reordering the frames to form an original message.
  - At the receiving end, TCP collects all the segments and reorders them based on sequence numbers.

Application Layer

- An application layer is the topmost layer in the TCP/IP model.
- It is responsible for handling high-level protocols, issues of representation.
- This layer allows the user to interact with the application.
- When one application layer protocol wants to communicate with another application layer, it forwards its data to the transport layer.

- There is an ambiguity occurs in the application layer. Every application cannot be placed inside the application layer except those who interact with the communication system. For example: text editor cannot be considered in application layer while web browser using **HTTP** protocol to interact with the network where **HTTP** protocol is an application layer protocol.

Following are the main protocols used in the application layer:

- **HTTP:** HTTP stands for Hypertext transfer protocol. This protocol allows us to access the data over the world wide web. It transfers the data in the form of plain text, audio, video. It is known as a Hypertext transfer protocol as it has the efficiency to use in a hypertext environment where there are rapid jumps from one document to another.
- **SNMP:** SNMP stands for Simple Network Management Protocol. It is a framework used for managing the devices on the internet by using the TCP/IP protocol suite.
- **SMTP:** SMTP stands for Simple mail transfer protocol. The TCP/IP protocol that supports the e-mail is known as a Simple mail transfer protocol. This protocol is used to send the data to another e-mail address.
- **DNS:** DNS stands for Domain Name System. An IP address is used to identify the connection of a host to the internet uniquely. But, people prefer to use the names instead of addresses. Therefore, the system that maps the name to the address is known as Domain Name System.
- **TELNET:** It is an abbreviation for Terminal Network. It establishes the connection between the local computer and remote computer in such a way that the local terminal appears to be a terminal at the remote system.
- **FTP:** FTP stands for File Transfer Protocol. FTP is a standard internet protocol used for transmitting the files from one computer to another computer.

UDP Protocol

In computer networking, the UDP stands for User Datagram Protocol. The David P. Reed developed the UDP protocol in 1980. It is defined in RFC 768, and it is a part of the TCP/IP

protocol, so it is a standard protocol over the internet. The UDP protocol allows the computer applications to send the messages in the form of datagrams from one machine to another machine over the Internet Protocol (IP)

network. The UDP is an alternative communication protocol to the TCP protocol (transmission control protocol). Like TCP, UDP provides a set of rules that governs how the data should be exchanged over the internet. The UDP works by encapsulating the data into the packet and providing its own header information to the packet. Then, this UDP packet is encapsulated to the IP packet and sent off to its destination. Both the TCP and UDP

protocols send the data over the internet protocol network, so it is also known as TCP/IP

and UDP/IP. There are many differences between these two protocols. UDP enables the process to process communication, whereas the TCP provides host to host communication. Since UDP sends the messages in the form of datagrams, it is considered the best-effort mode of communication. TCP

sends the individual packets, so it is a reliable transport medium. Another difference is that the TCP is a connection-oriented protocol whereas, the UDP is a connectionless protocol as it does not require any virtual circuit to transfer the data.

UDP also provides a different port number to distinguish different user requests and also provides the checksum capability to verify whether the complete data has arrived or not; the IP

layer does not provide these two services.

Features of UDP protocol

The following are the features of the UDP protocol:

- **Transport layer protocol**

UDP is the simplest transport layer communication protocol
It contains a minimum amount of communication mechanisms. It is considered an unreliable protocol, and it is based on best-effort delivery services. UDP provides no acknowledgment mechanism, which means that the receiver does not send the acknowledgment for the received packet, and the sender also does not wait for the acknowledgment for the packet that it has sent.

- **Connectionless**

The UDP is a connectionless protocol as it does not create a virtual path to transfer the data. It does not use the virtual path, so packets are sent in different paths between the sender and the receiver, which leads to the loss of packets or received out of order.

Ordered delivery of data is not guaranteed.

In the case of UDP, the datagrams are sent in some order will be received in the same order is not guaranteed as the datagrams are not numbered.

- **Ports**

The UDP protocol uses different port numbers so that the data can be sent to the correct destination. The port numbers are defined between 0 and 1023.

- **Faster transmission**

UDP enables faster transmission as it is a connectionless protocol, i.e., no virtual path is required to transfer the data. But there is a chance that the individual packet is lost, which affects the transmission quality. On the other hand, if the packet is lost in TCP connection, that packet will be resent, so it guarantees the delivery of the data packets.

- **Acknowledgment mechanism**

The UDP does have any acknowledgment mechanism, i.e., there is no handshaking between the UDP sender and UDP receiver. If the message is sent in TCP, then the receiver acknowledges that I am ready, then the sender sends the data. In the case of TCP, the handshaking occurs between the sender and the receiver, whereas in UDP, there is no handshaking between the sender and the receiver.

- **Segments are handled independently.**

Each UDP segment is handled individually of others as each segment takes different path to reach the destination. The UDP segments can be lost or delivered out of order to reach the destination as there is no connection setup between the sender and the receiver.

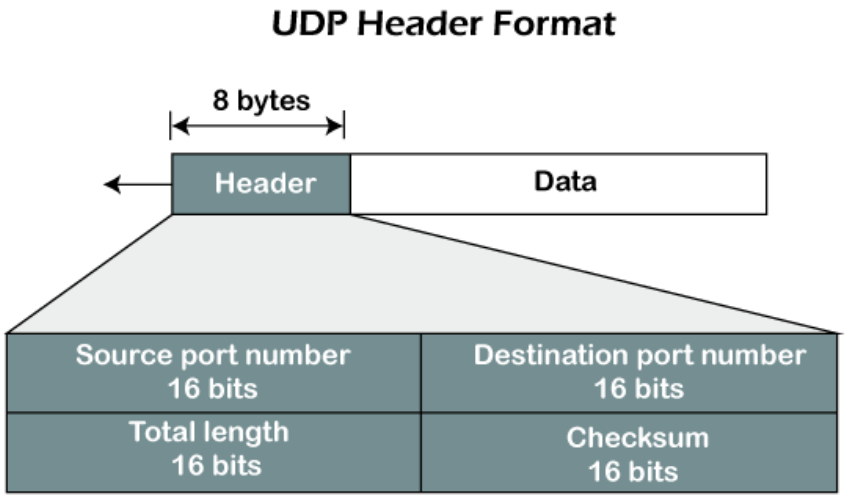
- **Stateless**

It is a stateless protocol that means that the sender does not get the acknowledgement for the packet which has been sent.

Why do we require the UDP protocol?

As we know that the UDP is an unreliable protocol, but we still require a UDP protocol in some cases. The UDP is deployed where the packets require a large amount of bandwidth along with the actual data. For example, in video streaming, acknowledging thousands of packets is troublesome and wastes a lot of bandwidth. In the case of video streaming, the loss of some packets couldn't create a problem, and it can also be ignored.

UDP Header Format

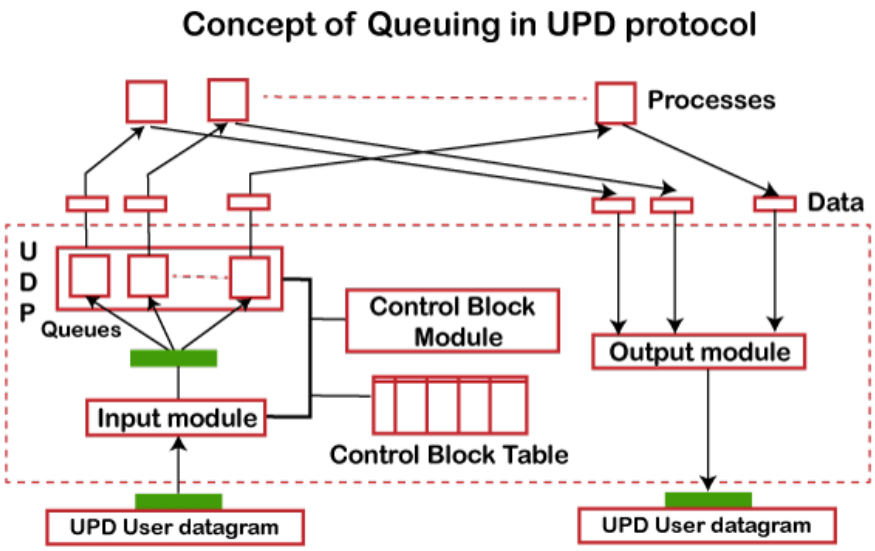


In UDP, the header size is 8 bytes, and the packet size is upto 65,535 bytes. But this packet size is not possible as the data needs to be encapsulated in the IP datagram, and an IP packet, the header size can be 20 bytes; therefore, the maximum of UDP would be 65,535 minus 20. The size of the data that the UDP packet can carry would be 65,535 minus 28 as 8 bytes for the header of the UDP packet and 20 bytes for IP header.

The UDP header contains four fields:

- **Source port number:** It is 16-bit information that identifies which port is going t send the packet.
- **Destination port number:** It identifies which port is going to accept the information. It is 16-bit information which is used to identify application-level service on the destination machine.
- **Length:** It is 16-bit field that specifies the entire length of the UDP packet that includes the header also. The minimum value would be 8-byte as the size of the header is 8 bytes.
- **Checksum:** It is a 16-bits field, and it is an optional field. This checksum field checks whether the information is accurate or not as there is the possibility that the information can be corrupted while transmission. It is an optional field, which means that it depends upon the application, whether it wants to write the checksum or not. If it does not want to write the checksum, then all the 16 bits are zero; otherwise, it writes the checksum. In UDP, the checksum field is applied to the entire packet, i.e., header as well as data part whereas, in IP, the checksum field is applied to only the header field.

Concept of Queuing in UDP protocol



In UDP protocol, numbers are used to distinguish the different processes on a server and client. We know that UDP provides a process to process communication. The client generates the processes that need services while the server generates the processes that provide services. The queues are available for both the processes, i.e., two queues for each process. The first queue is the incoming queue that receives the messages, and the second one is the outgoing queue that sends the messages. The queue functions when the process is running. If the process is terminated then the queue will also get destroyed.

UDP handles the sending and receiving of the UDP packets with the help of the following components:

- **Input queue:** The UDP packets uses a set of queues for each process.
- **Input module:** This module takes the user datagram from the IP, and then it finds the information from the control block table of the same port. If it finds the entry in the control block table with the same port as the user datagram, it enqueues the data.
- **Control Block Module:** It manages the control block table.
- **Control Block Table:** The control block table contains the entry of open ports.
- **Output module:** The output module creates and sends the user datagram.

Several processes want to use the services of UDP. The UDP multiplexes and demultiplexes the processes so that the multiple processes can run on a single host.

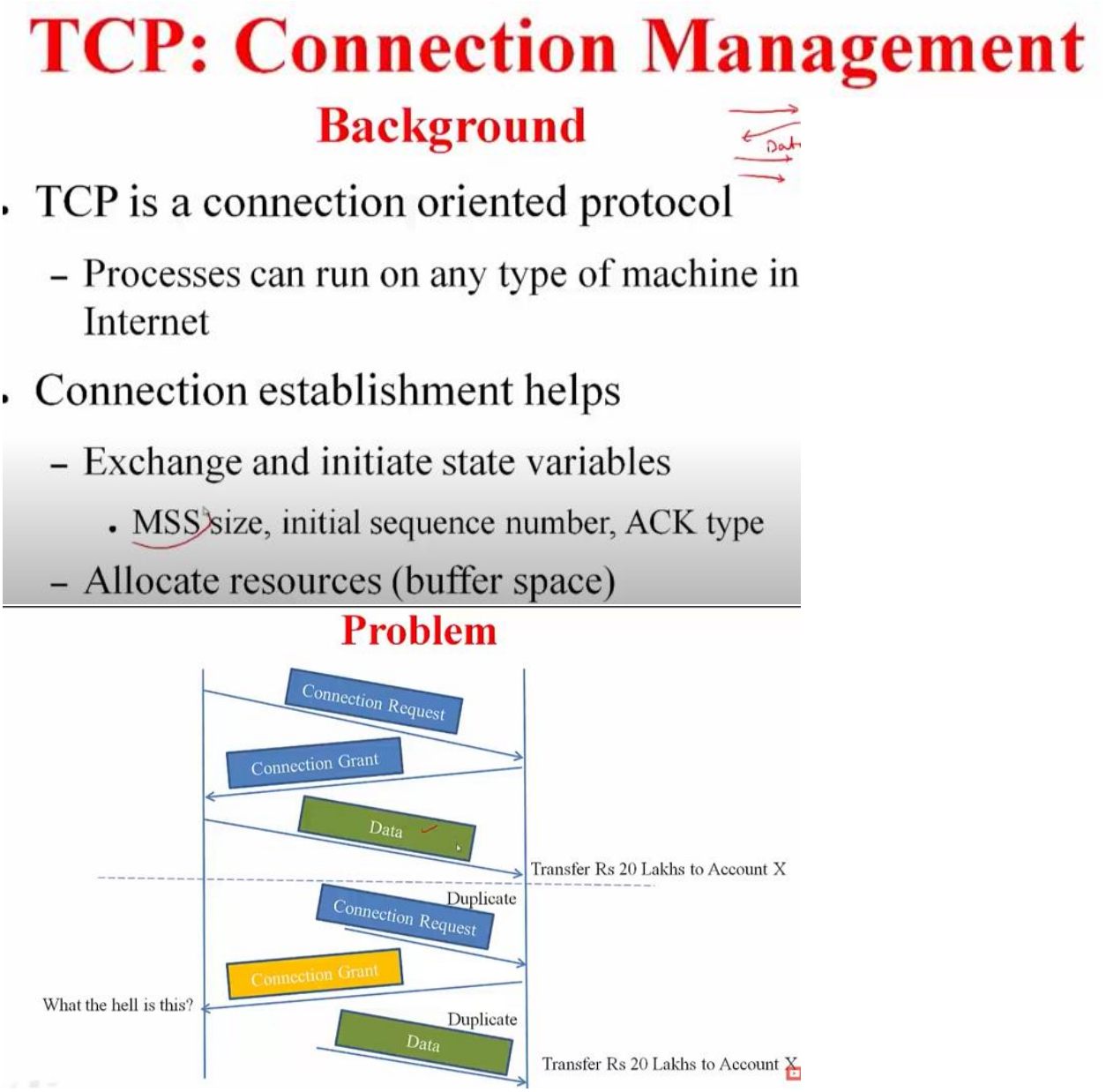
Limitations

- It provides an unreliable connection delivery service. It does not provide any services of IP except that it provides process-to-process communication.

- The UDP message can be lost, delayed, duplicated, or can be out of order.
- It does not provide a reliable transport delivery service. It does not provide any acknowledgment or flow control mechanism. However, it does provide error control to some extent.

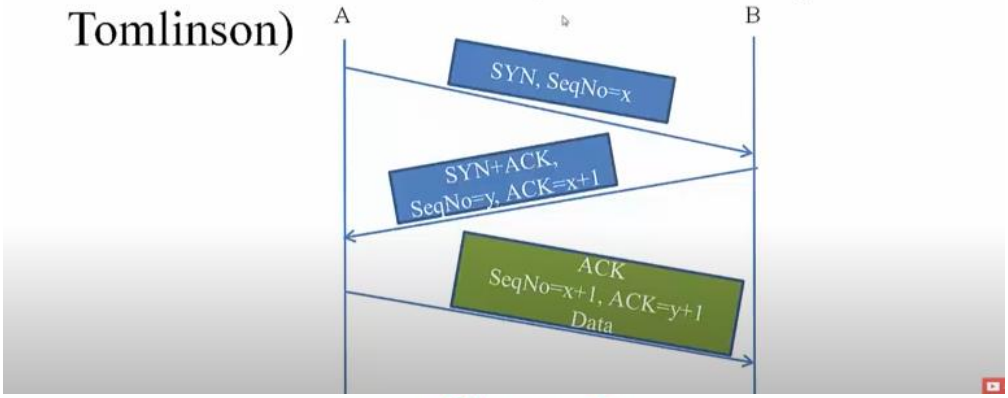
Advantages

- It produces a minimal number of overheads.

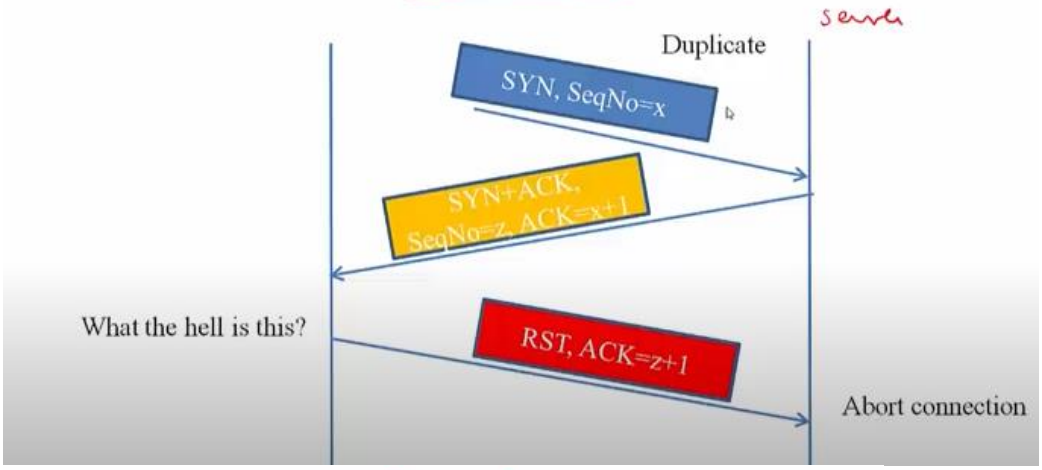


Solution

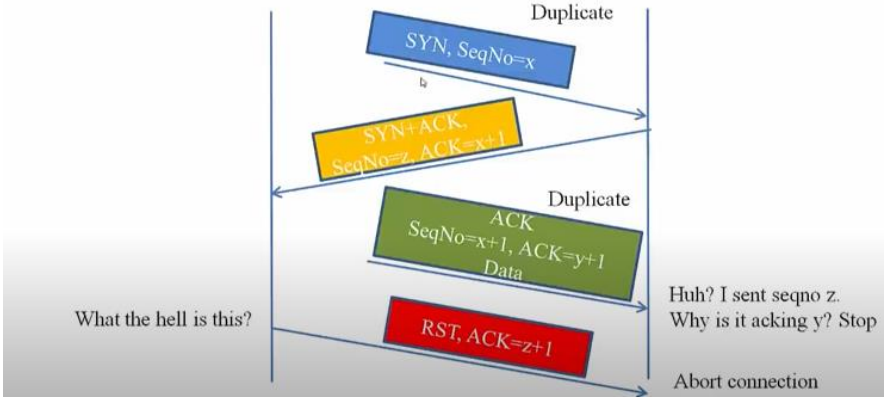
- TCP's famous three-way handshake (idea from Tomlinson)



Case-1



Case-2



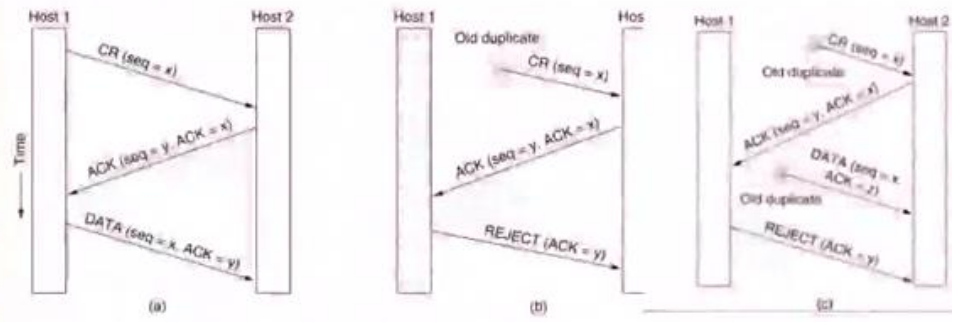
Initial Sequence Number (ISN)

- Why not start with Seqno zero?
- Segments from different connections can get mixed up
- Security risk when ISN's are predictable
- Original solution: Use a clock (e.g. increments every 4 microsec) to choose ISN
  - 32 bit sequence number wraps around in 4 hrs
- Current implementations use random ISN

CONNECTION MANAGEMENT

- Connection Establishment
- Connection Release

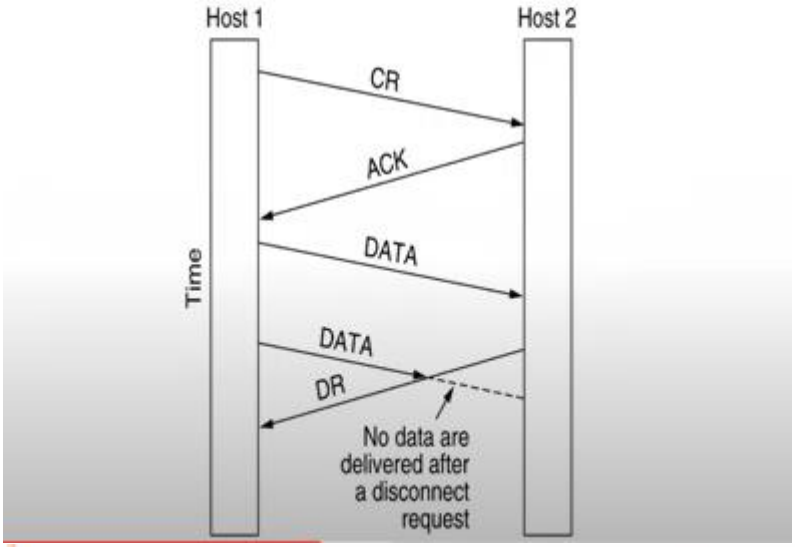
CONNECTION ESTABLISHMENT



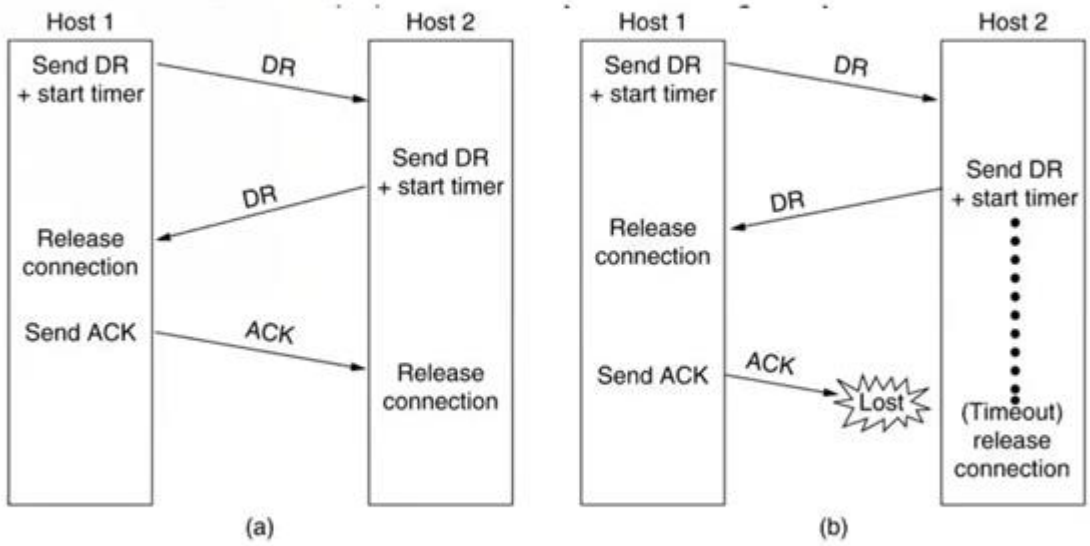
Three protocol scenarios for establishing a connection using a **three-way handshake**. CR denotes CONNECTION REQUEST.
(a) Normal operation.
(b) Old CONNECTION REQUEST appearing out of nowhere.
(c) Duplicate CONNECTION REQUEST and duplicate ACK.

CONNECTION RELEASE

ASYMMETRIC RELEASE

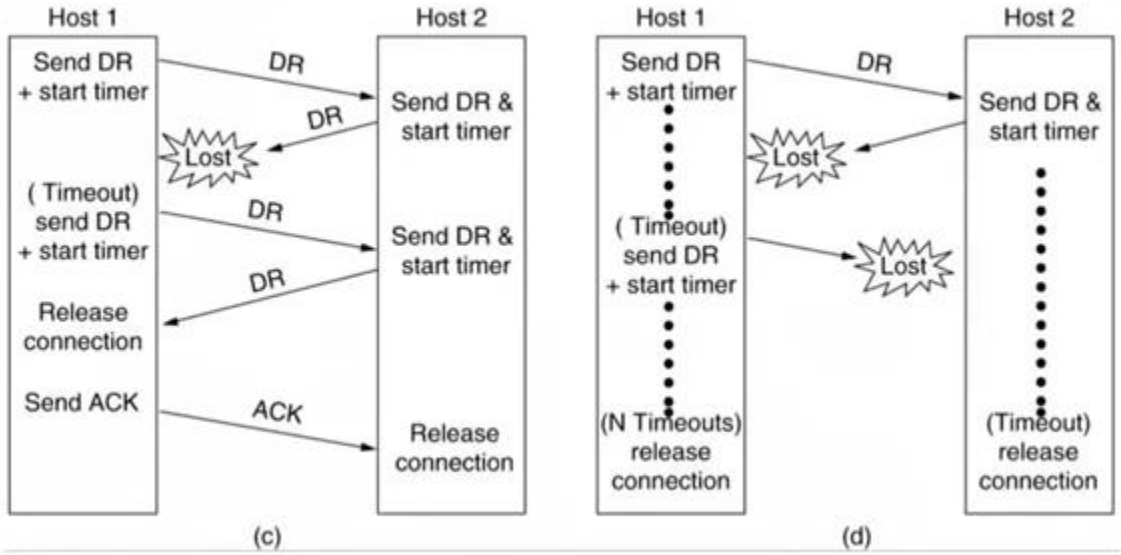


CONNECTION RELEASE



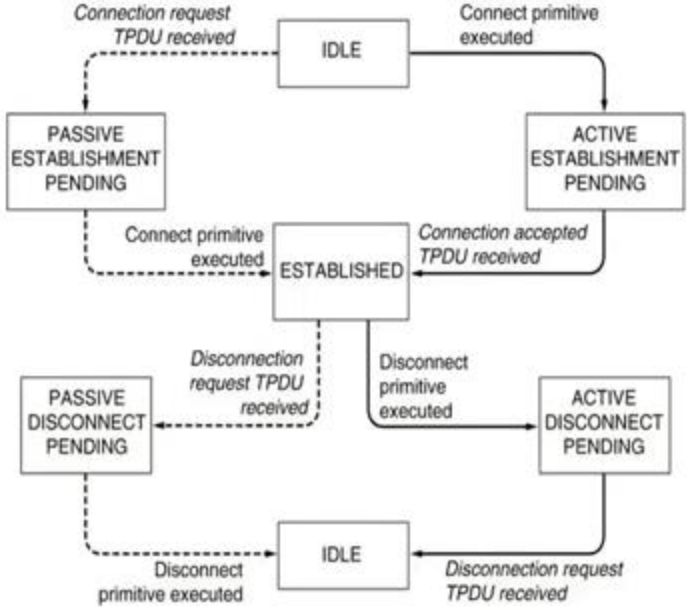
Four protocol scenarios for releasing a connection.
(a) Normal case of a three-way handshake. (b) final ACK lost.

Connection Release



(c) Response lost. (d) Response lost and subsequent DRs lost.

Transport Service Primitives



•A state diagram for a simple connection management scheme. Transitions labeled in italics are caused by packet arrivals.

•The solid lines show the client's state sequence.

•The dashed lines show the server's state sequence.

.SESSION LAYER

- The Session Layer allows users on different machines to establish active communication sessions between them.
- It's main aim is to establish, maintain and synchronize the interaction between communicating systems.
- Session layer manages and synchronize the conversation between two different applications.
- In Session layer streams of data are marked and are resynchronized properly, so that the ends of the messages are not cut prematurely and data loss is avoided.

FUNCTIONS :

1. **Dialog Control :**

   This layer allows two systems to start communication with each other in half-duplex or full-duplex.

2. **Token Management:**

   This layer prevents two parties from attempting the same critical operation at the same time.

3. **Synchronization :**

This layer allows a process to add checkpoints which are considered as synchronization points into stream of data

.Eg: If a system is sending a file of 800 pages, adding checkpoints after every 50 pages is recommended. This ensures that 50 page unit is successfully received and acknowledged. This is beneficial

DESIGN ISSUES :

- To allow machines to establish sessions between them in a seamless fashion.
- Provide enhanced services to the user.
- To manage dialog control.
- To provide services such as Token management & synchronisation

Design issues in Session Layer

Session Layer is one of the Seven Layers of OSI Model. Physical layer, Data Link Layer and Network Layer lack some services such as establishment of a session between communicating systems. This is managed by Session Layer which particularly behaves as a dialog controller between communicating system thus facilitating interaction between them.
Before looking into design issues, here are some of functions of Session Layer:

1. **Dialog Control –**
   Session layer allows two systems to enter into a dialog exchange mechanism which can either be full or half-duplex.
2. **Managing Tokens –**
   The communicating systems in a network try to perform some critical operations and it is Session Layer
   which prevents collisions which might occur while performing these operations which would otherwise result in a loss.

3. **Synchronization –**
   Checkpoints are the midway marks that are added after a particular interval during stream of data
   transfer. These points are also referred to as synchronization points. The Session layer permits process to add these checkpoints.
   For example, suppose a file of 400 pages is being sent over a network, then it is highly beneficial to set up a checkpoint after every 50 pages so that next 50 pages are sent only when previous pages are received and acknowledged.

Design Issues with Session Layer :

1. **Establish sessions between machines –**
   The establishment of session between machines is an important service provided by session layer. This session is responsible for creating a dialog between connected machines. The Session Layer provides mechanism for opening, closing and managing a session between end-user application processes, i.e. a semi-permanent dialogue. This session consists of requests and responses that occur between applications.
2. **Enhanced Services –**
   Certain services such as checkpoints and management of tokens are the key features of session layer and thus it becomes necessary to keep enhancing these features during the layer's design.
3. **To help in Token management and Synchronization –**
   The session layer plays an important role in preventing collision of several critical operation as well as ensuring better data transfer over network by establishing synchronization points at specific intervals. Thus it becomes highly important to ensure proper execution of these services.

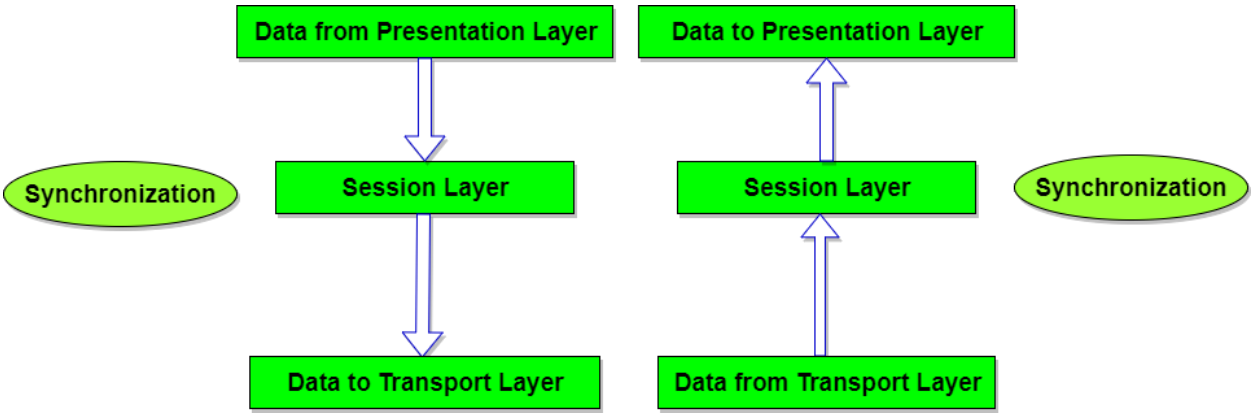
The Session Layer allows users on different machines to establish active communication sessions between them.

It's main aim is to establish, maintain and synchronize the interaction between communicating systems. Session layer manages and synchronize the conversation between two different applications. In Session layer, streams of data are marked and are resynchronized properly, so that the ends of the messages are not cut prematurely and data loss is avoided.

Functions of Session Layer

1. **Dialog Control :** This layer allows two systems to start communication with each other in half-duplex or full-duplex.
2. **Token Management:** This layer prevents two parties from attempting the same critical operation at the same time.
3. **Synchronization :** This layer allows a process to add checkpoints which are considered as synchronization points into stream of

data. Example: If a system is sending a file of 800 pages, adding checkpoints after every 50 pages is recommended. This ensures that 50 page unit is successfully received and acknowledged. This is beneficial at the time of crash as if a crash happens at page number 110; there is no need to retransmit 1 to100 pages.



Design Issues with Session Layer

- To allow machines to establish sessions between them in a seamless fashion.
- Provide enhanced services to the user.
- To manage dialog control.
- To provide services such as **Token management** and **Synchronization**.

What is Session Layer in the Computer Network?

t is one of the parts of the upper layers. The Application Layer, Presentation Layer, and Session Layers form the OSI reference model's upper layers. They provide user-oriented services. The session layer provides a defined set of services to the presentation layer

Presentation Layer uses session layer protocols and transport services. The session layer service's actual user is the application layer through which the presentation layer lies between them. It has relatively few features as compared to lower layers. It controls structures and the interaction between the application programme.

It synchronizes the interaction between elements and controls the direction of information flow, but it is not taken. It has nothing to do with error detection and correction functions. It provides services to the presentation layer. The service definition and protocol specifications are defined in ISO 8326 and ISO 8327.

Design Issues

The session layer is the thinnest layer with the most negligible numbers of protocols in the OSI model. The session layer objective is to create, maintain and synchronize dialogs between transmitting upper layers. Communication can take place between either users or applications.

Session to Transport Communication

The session layer helps coordinate connection and release of dialogs connections between the communicating applications. It communicates with the transport layer. The communication may be one to one, many to one and one to many. In one to one, one session layer connection establishes for each transport layer connection.

In many to one, multiple session layer connections are shared with the services of one transport layer connection. The one to many connection communication is set up when one session layer connection calls for many transport layer connections to handle the service.

Dialog Management

The session layer aims to decide whose turn it is to talk. Some of the applications operate in half-duplex mode. The half-duplex provides two sides alternate communication between sending and receiving messages and never sending data simultaneously.

The dialog management is implemented using a data token transmitted back and forth to provide a user with a right to transmit only when it possesses the token.

Activity Management

The session layer enables the user to delimit data into logical units called activities. Each activity is treated as a separate activity and independent from the preceding and following activities to that activity.

Activities are used to delimit files of a multi-file transfer. Activities are used for quarantining, collecting all the data of a multi-message exchange together before processing them. The receiving application begins processing data only after all the data arrives. This ensures that all or none of a set of operations is performed.

For example, a bank transaction may involve locking a record, updating a value, and unlocking the record. When an application processes the first operation but could not receive the remaining operations due to the client or network failures. The record will remain locked forever. Quarantining solves this problem.

Exception handling

It is a general-purpose mechanism for reporting errors.

Remote Procedure Call

- Basic RPC operation

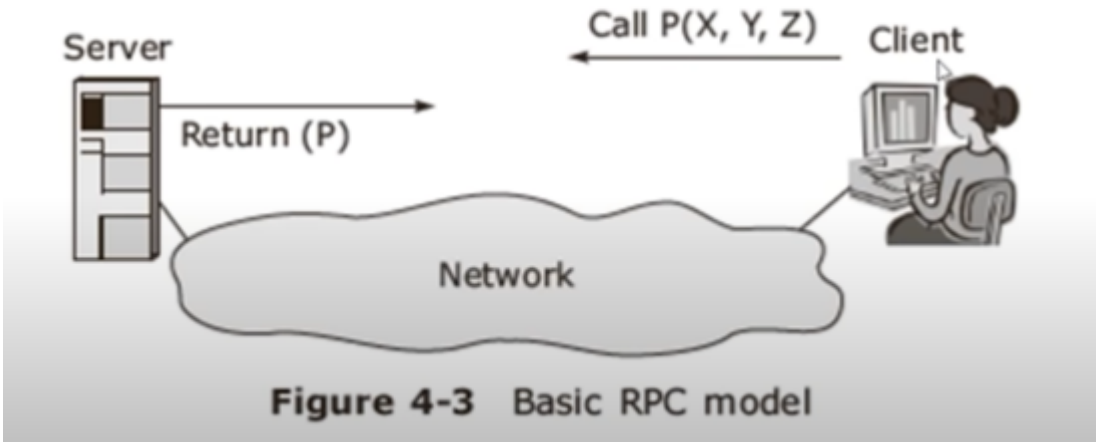


Figure 4-3 Basic RPC model

Elements of RPC mechanism implementation

- Client
- Client stub
- RPC Runtime
- Server stub
- Server

RPC Execution

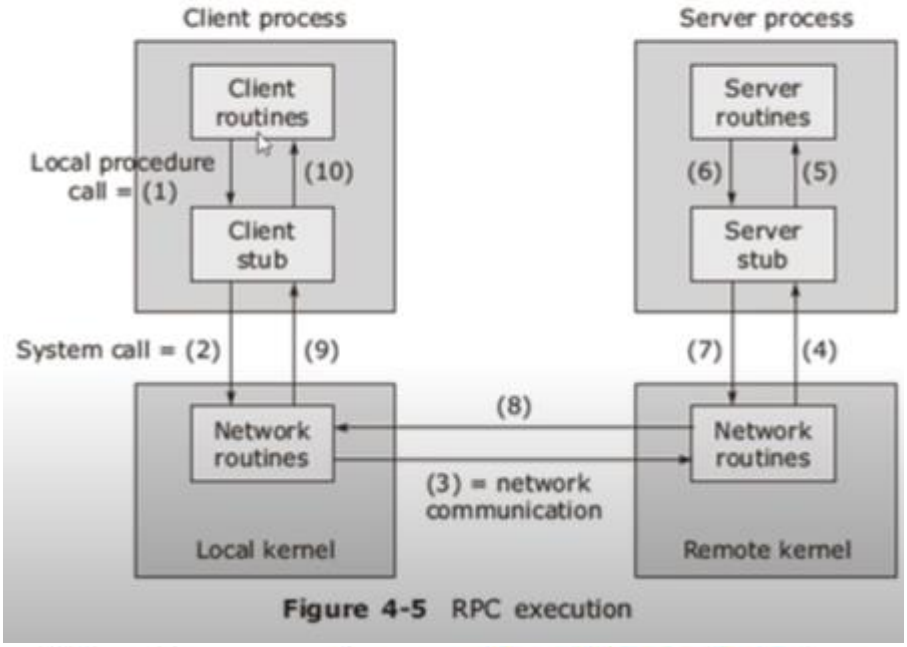


Figure 4-5 RPC execution

- Client procedure calls client stub in normal way
- Client stub builds message, calls local OS
- Client's OS sends message to remote OS
- Remote OS gives message to server stub
- Server stub unpacks parameters, calls server
- Server does work, returns result to the stub
- Server stub packs it in message, calls local OS
- Server's OS sends message to client's OS
- Client's OS gives message to client stub
- Stub unpacks result, returns to client

RPC implementation

RPC messages:

- Call / Request
- Reply

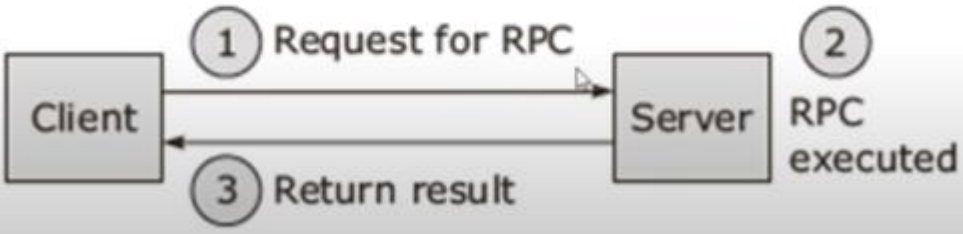


Figure 4-7 RPC messages

RPC Call/ Request message

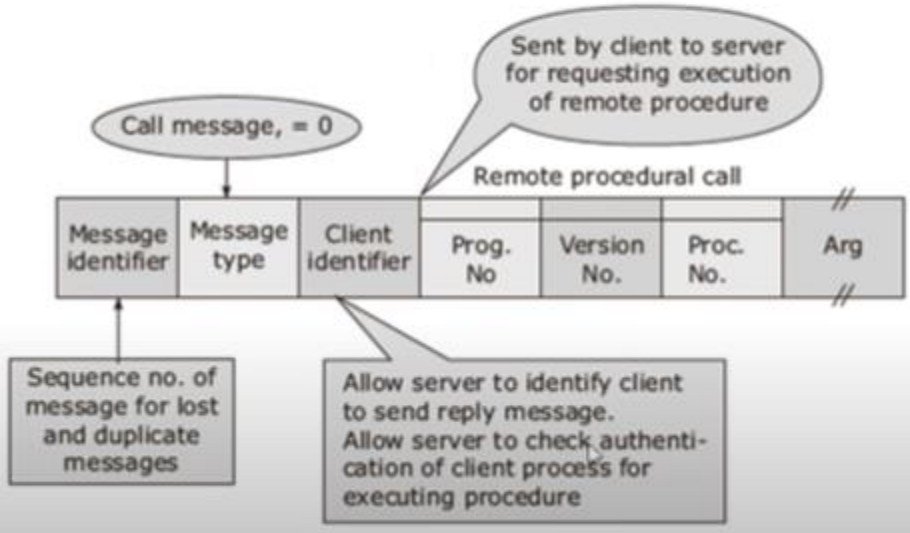


Figure 4-8 RPC call/request message format

RPC reply message

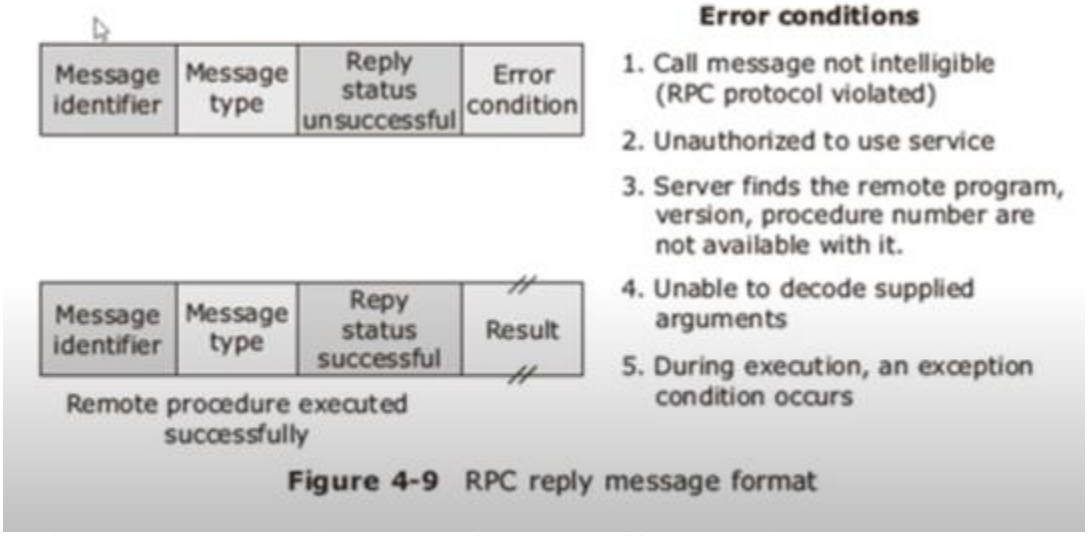


Figure 4-9 RPC reply message format

Parameter Passing Semantics

- Call-by-value semantic
  - Marshalling
- Call-by –reference semantic
- Call-by-copy/restore semantic

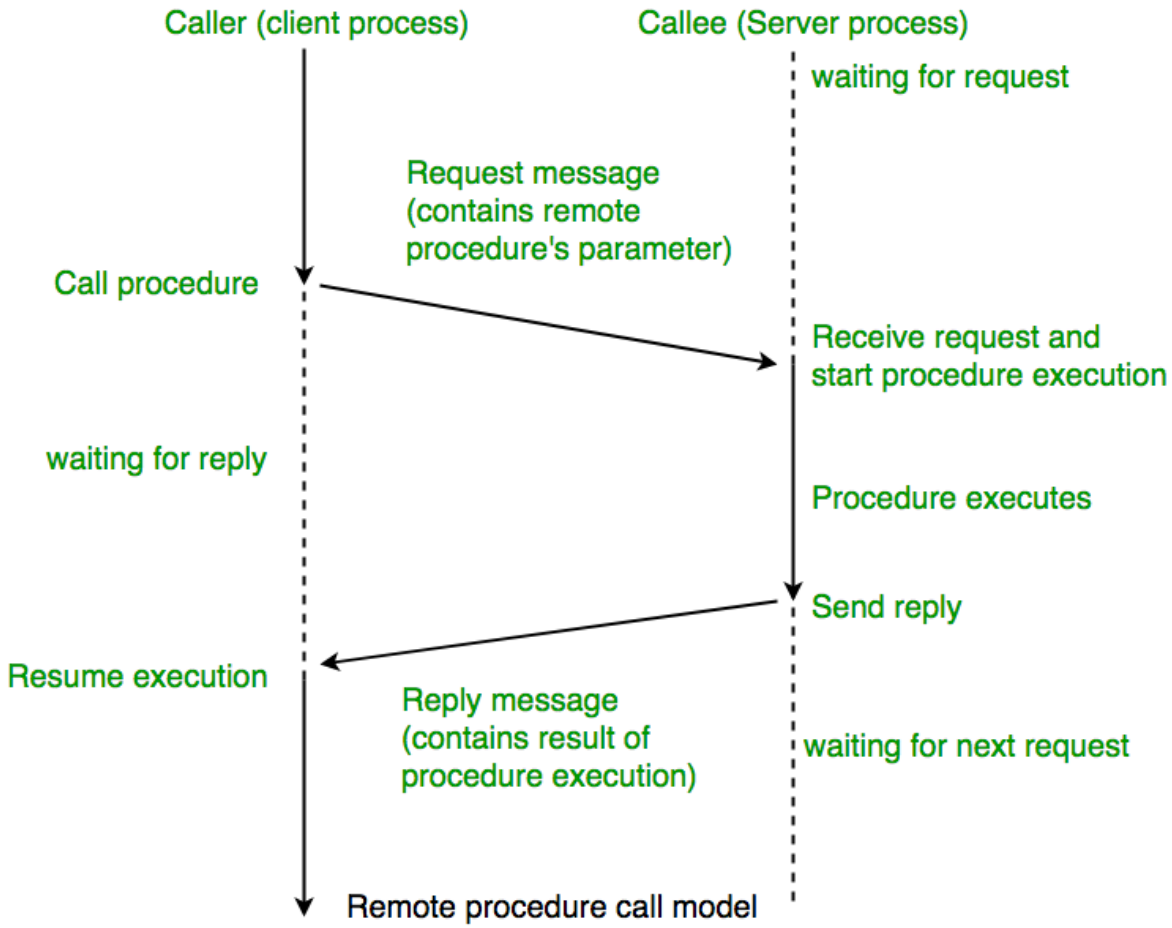
Call-by-value copies all parameters into a message before transmission . Call-by-reference passes pointers to the parameters that are passed from the client to the server. Call-by-copy/restore uses temporary storage accessible to both programs

Remote Procedure Call (RPC) in Operating System

Remote Procedure Call (RPC) is a powerful technique for constructing **distributed, client-server based applications**. It is based on extending the conventional local procedure calling so that the **called procedure**

need not exist in the same address space as the calling procedure. The two processes may be on the same system, or they may be on different systems with a network connecting them.

When making a Remote Procedure Call:

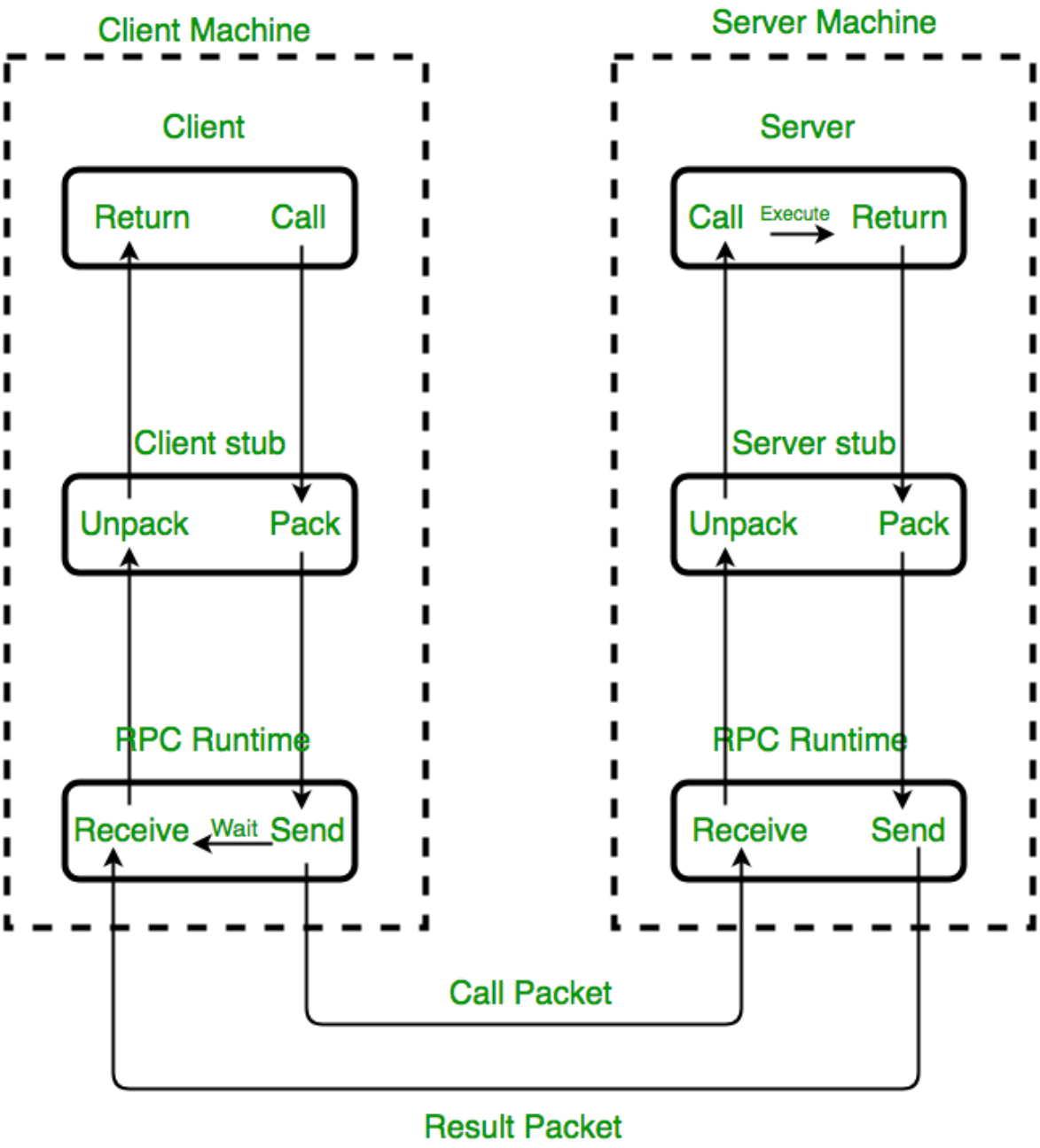


1. The calling environment is suspended, procedure parameters are transferred across the network to the environment where the procedure is to execute, and the procedure is executed there.

2. When the procedure finishes and produces its results, its results are transferred back to the calling environment, where execution resumes as if returning from a regular procedure call.

NOTE: RPC is especially well suited for client-server **(e.g. query-response)** interaction in which the flow of control **alternates between the caller and callee**. Conceptually, the client and server do not both execute at the same time. Instead, the thread of execution jumps from the caller to the callee and then back again.

Working of RPC



Implementation of RPC mechanism

The following steps take place during a RPC :

1. A client invokes a **client stub procedure**, passing parameters in the usual way. The client stub resides within the client's own address space.
2. The client stub **marshalls(pack)** the parameters into a message. Marshalling includes converting the representation of the parameters into a standard format, and copying each parameter into the message.
3. The client stub passes the message to the transport layer, which sends it to the remote server machine.

4. On the server, the transport layer passes the message to a server stub, which **demarshalls(unpack)** the parameters and calls the desired server routine using the regular procedure call mechanism.
5. When the server procedure completes, it returns to the server stub **(e.g., via a normal procedure call return)**, which marshalls the return values into a message. The server stub then hands the message to the transport layer.
6. The transport layer sends the result message back to the client transport layer, which hands the message back to the client stub.
7. The client stub demarshalls the return parameters and execution returns to the caller.

RPC ISSUES :

Issues that must be addressed:

1. RPC Runtime:

RPC run-time system is a library of routines and a set of services that handle the network communications that underlie the RPC mechanism. In the course of an RPC call, client-side and server-side run-time systems' code handle **binding, establish communications over an appropriate protocol, pass call data between the client and server, and handle communications errors.**

2. Stub:

The function of the stub is to **provide transparency to the programmer-written application code**.

- **On the client side**, the stub handles the interface between the client's local procedure call and the run-time system, marshalling and unmarshalling data, invoking the RPC run-time protocol, and if requested, carrying out some of the binding steps.
- **On the server side**, the stub provides a similar interface between the run-time system and the local manager procedures that are executed by the server.

3. Binding: How does the client know who to call, and where the service resides?

The most flexible solution is to use dynamic binding and find the server at run time when the RPC is first made. The first time the client stub is invoked, it contacts a name server to determine the transport address at which the server resides.

Binding consists of two parts:

- Naming:
- Locating:

1. **A Server** having a service to offer exports an interface for it. Exporting an interface registers it with the system so that clients can use it.
2. **A Client** must import an (exported) interface before communication can begin.

4. The call semantics associated with RPC :

It is mainly classified into following choices-

- **Retry request message –**
  Whether to retry sending a request message when a server has failed or the receiver didn't receive the message.
- **Duplicate filtering –**
  Remove the duplicate server requests.
- **Retransmission of results –**
  To resend lost messages without re-executing the operations at the server side.

ADVANTAGES :

1. RPC provides **ABSTRACTION** i.e message-passing nature of network communication is hidden from the user.
2. RPC often omits many of the protocol layers to improve performance. Even a small performance improvement is important because a program may invoke RPCs often.
3. RPC enables the usage of the applications in the distributed environment, not only in the local environment.
4. With RPC code re-writing / re-developing effort is minimized.
5. Process-oriented and thread oriented models supported by RPC.

Presentation Layer in OSI model

This layer is also known as Translation layer, as this layer serves as a data translator for the network. The data which this layer receives from the Application Layer is extracted and manipulated here as per the required format to transmit over the network. The main responsibility of this layer is to provide or define the data format and encryption. The presentation layer is also called as Syntax layer since it is responsible for maintaining the proper syntax of the data which it either receives or transmits to other layer(s).

- Presentation layer format and encrypts data to be sent across the network.
- This layer takes care that the data is sent in such a way that the receiver will understand the information (data) and will be able to use the data efficiently and effectively.
- This layer manages the abstract data structures and allows high-level data structures (example- banking records), which are to be defined or exchanged.
- This layer carries out the encryption at the transmitter and decryption at the receiver.
- This layer carries out data compression to reduce the bandwidth of the data to be transmitted (the primary goal of data compression is to reduce the number of bits which is to be transmitted).
- This layer is responsible for interoperability (ability of computers to exchange and make use of information) between encoding methods as different computers use different encoding methods.
- This layer basically deals with the presentation part of the data.
- Presentation layer, carries out the data compression (number of bits reduction while transmission), which in return improves the data throughput.

- This layer also deals with the issues of string representation.
- The presentation layer is also responsible for integrating all the formats into a standardized format for efficient and effective communication.
- This layer encodes the message from the user-dependent format to the common format and vice-versa for communication between dissimilar systems.
- This layer deals with the syntax and semantics of the messages.
- This layer also ensures that the messages which are to be presented to the upper as well as the lower layer should be standardized as well as in an accurate format too.
- Presentation layer is also responsible for translation, formatting, and delivery of information for processing or display.
- This layer also performs serialization (process of translating a data structure or an object into a format that can be stored or transmitted easily).

Features of Presentation Layer in the OSI model: Presentation layer, being the 6th layer in the OSI model, plays a vital role while communication is taking place between two devices in a network.
List of features which are provided by the presentation layer are:

- Presentation layer could apply certain sophisticated compression techniques, so fewer bytes of data are required to represent the information when it is sent over the network.
- If two or more devices are communicating over an encrypted connection, then this presentation layer is responsible for adding encryption on the sender's end as well as the decoding the encryption on the receiver's end so that it can represent the application layer with unencrypted, readable data.
- This layer formats and encrypts data to be sent over a network, providing freedom from compatibility problems.
- This presentation layer also negotiates the Transfer Syntax.
- This presentation layer is also responsible for compressing data it receives from the application layer before delivering it to the session layer (which is the 5th layer in the OSI model) and thus improves the speed as well as the efficiency of communication by minimizing the amount of the data to be transferred.

Working of Presentation Layer in the OSI model :
Presentation layer in the OSI model, as a translator, converts the data sent by the application layer of the transmitting node into an acceptable and compatible data format based on the applicable network protocol and architecture. Upon arrival at the receiving computer, the presentation layer translates data into an acceptable format usable by the application layer. Basically, in other words, this layer takes care of any issues occurring when transmitted data must be viewed in a format different from the original format. Being the functional part of the OSI mode, the presentation layer performs a multitude (large number of) data conversion algorithms and character translation functions. Mainly, this layer is responsible for managing two network characteristics: protocol (set of rules) and architecture.

Presentation Layer Protocols :
Presentation layer being the 6th layer, but the most important layer in the OSI model performs several types of functionalities, which makes sure that data which is being

transferred or received should be accurate or clear to all the devices which are there in a closed network.
Presentation Layer, for performing translations or other specified functions, needs to use certain protocols which are defined below –

- **Apple Filing Protocol (AFP):** Apple Filing Protocol is the proprietary network protocol (communications protocol) that offers services to macOS or the classic macOS. This is basically the network file control protocol specifically designed for Mac-based platforms.
- **Lightweight Presentation Protocol (LPP):** Lightweight Presentation Protocol is that protocol which is used to provide ISO presentation services on the top of TCP/IP based protocol stacks.
- **NetWare Core Protocol (NCP):** NetWare Core Protocol is the network protocol which is used to access file, print, directory, clock synchronization, messaging, remote command execution and other network service functions.
- **Network Data Representation (NDR):** Network Data Representation is basically the implementation of the presentation layer in the OSI model, which provides or defines various primitive data types, constructed data types and also several types of data representations.
- **External Data Representation (XDR):** External Data Representation (XDR) is the standard for the description and encoding of data. It is useful for transferring data between computer architectures and has been used to communicate data between very diverse machines. Converting from local representation to XDR is called encoding, whereas converting XDR into local representation is called decoding.
- **Secure Socket Layer (SSL):** The Secure Socket Layer protocol provides security to the data that is being transferred between the web browser and the server. SSL encrypts the link between a web server and a browser, which ensures that all data passed between them remains private and free from attacks.

Design issues with Presentation Layer :

1. **Standard way of encoding data –**
   The presentation layer follows a standard way to encode data when it needs to be transmitted. This encoded data is represented as character strings, integers, floating point numbers, and data structures composed of simple components. It is handled differently by different machines based on the encoding methods followed by them.
2. **Maintaining the Syntax and Semantics of distributed information –**
   The presentation layer manages and maintains the syntax as well as logic and meaning of the information that is distributed.
3. **Standard Encoding on the wire –**
   The data structures that are defined to be exchanged need to be abstract along with the standard encoding to be used “on the wire”.

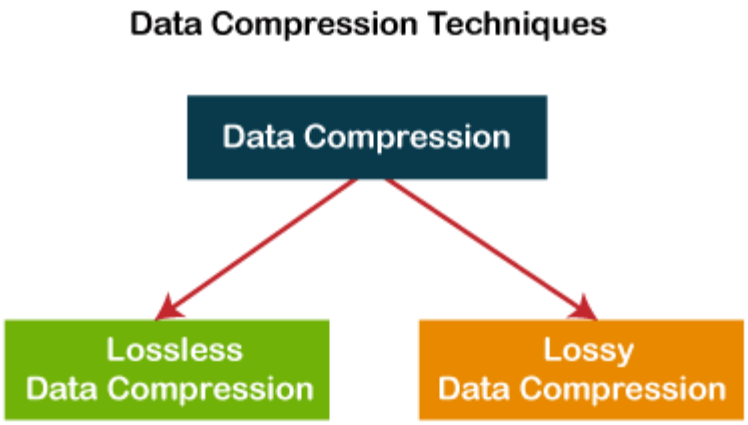
What is Data Compression

Data Compression is also referred to as **bit-rate reduction** or **source coding**. This technique is used to reduce the size of large files.

The advantage of data compression is that it helps us save our disk space and time in the data transmission.

There are mainly two types of data compression techniques -

1. Lossless Data Compression
2. Lossy Data Compression

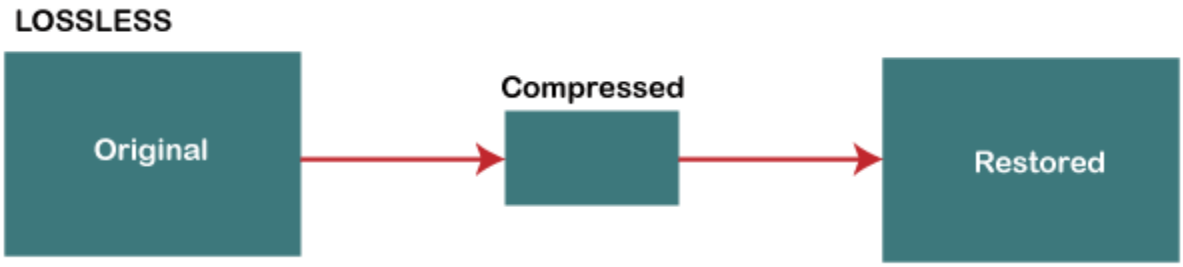


What is Lossless data compression

Lossless data compression is used to compress the files **without losing an original file's quality and data**. Simply, we can say that in lossless data compression, file size is reduced, but the quality of data remains the same.

The main advantage of lossless data compression is that we can restore the original data in its original form after the decompression.

Lossless data compression mainly used in the sensitive documents, confidential information, and PNG, RAW, GIF, BMP file formats.



Some most important Lossless data compression techniques are -

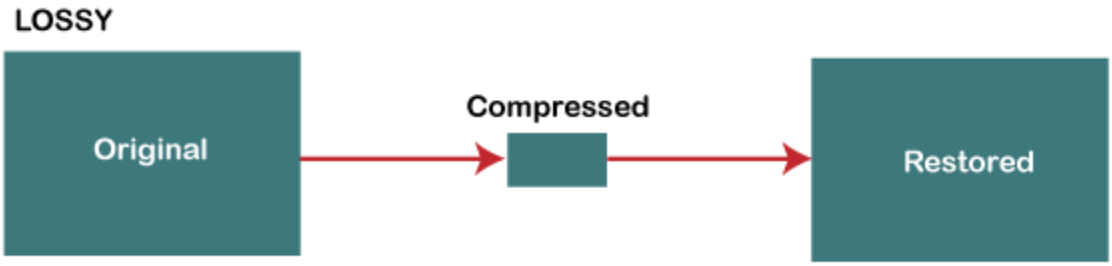
1. Run Length Encoding (RLE)
2. Lempel Ziv - Welch (LZW)
3. Huffman Coding
4. Arithmetic Coding

What is Lossy data compression

Lossy data compression is used to compress larger files into smaller files. In this compression technique, some specific amount of **data and quality are removed (loss) from the original file**. It takes less memory space from the original file due to the loss of original data and quality. This technique is generally useful for us when the quality of data is not our first priority.

Note: The human eye does not measure the loss of data.

Lossy data compression is most widely used in JPEG images, MPEG video, and MP3 audio formats.



Some important Lossy data compression techniques are -

1. Transform coding
2. Discrete Cosine Transform (DCT)
3. Discrete Wavelet Transform (DWT)

Difference between lossless and lossy data compression

As we know, both lossless and lossy data compression techniques are used to compress data form its original size. The main difference between lossless and lossy data compression is that we can restore the lossless data in its original form after the decompression, but lossy data can't be restored to its original form after the decompression.

The below table shows the difference between lossless and lossy data compression -

S.No	Lossless data compression	Lossy data compression
1.	In Lossless data compression, there is no loss of any data and quality.	In Lossy data compression, there is a loss o quality and data, which is not measurable.
2.	In lossless, the file is restored in its original form.	In Lossy, the file does not restore in its original form.
3.	Lossless data compression algorithms are Run Length Encoding, Huffman encoding, Shannon fano encoding, Arithmetic encoding, Lempel Ziv Welch encoding, etc.	Lossy data compression algorithms are Transform coding, Discrete Cosine Transform, Discrete Wavelet Transform fractal compression, etc.
4.	Lossless compression is mainly used to compress text-sound and images.	Lossy compression is mainly used to compress audio, video, and images.
5.	As compare to lossy data compression, lossless data compression holds more data.	As compare to lossless data compression lossy data compression holds less data.
6.	File quality is high in the lossless data compression.	File quality is low in the lossy data compression.
7.	Lossless data compression mainly supports RAW, BMP, PNG, WAV, FLAC, and ALAC file types.	Lossy data compression mainly supports JPEG, GIF, MP3.

What is The Purpose of Cryptography?

Cryptography aims to keep data and messages private and inaccessible to possible threats or bad actors. It frequently works invisibly to encrypt and decrypt the data you send through email, social media, applications, and website interactions.

There are several uses for symmetric cryptography, including:

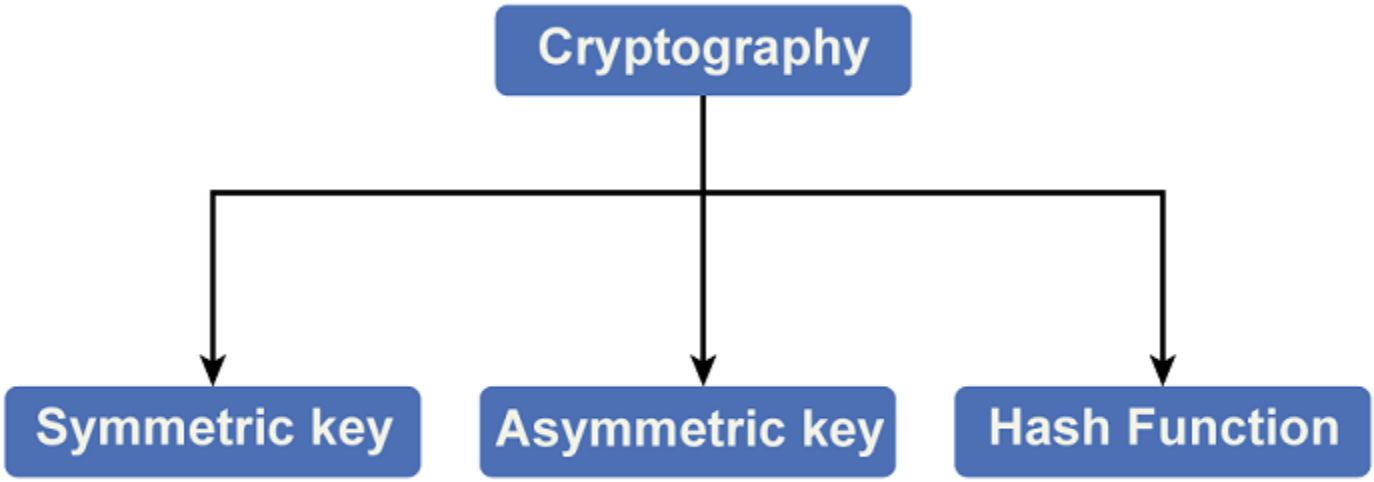
- Payment applications and card transactions

- Random number generation
- Verify the sender's signature to be sure they are who they claim they are

There are several uses for asymmetric cryptography, including:

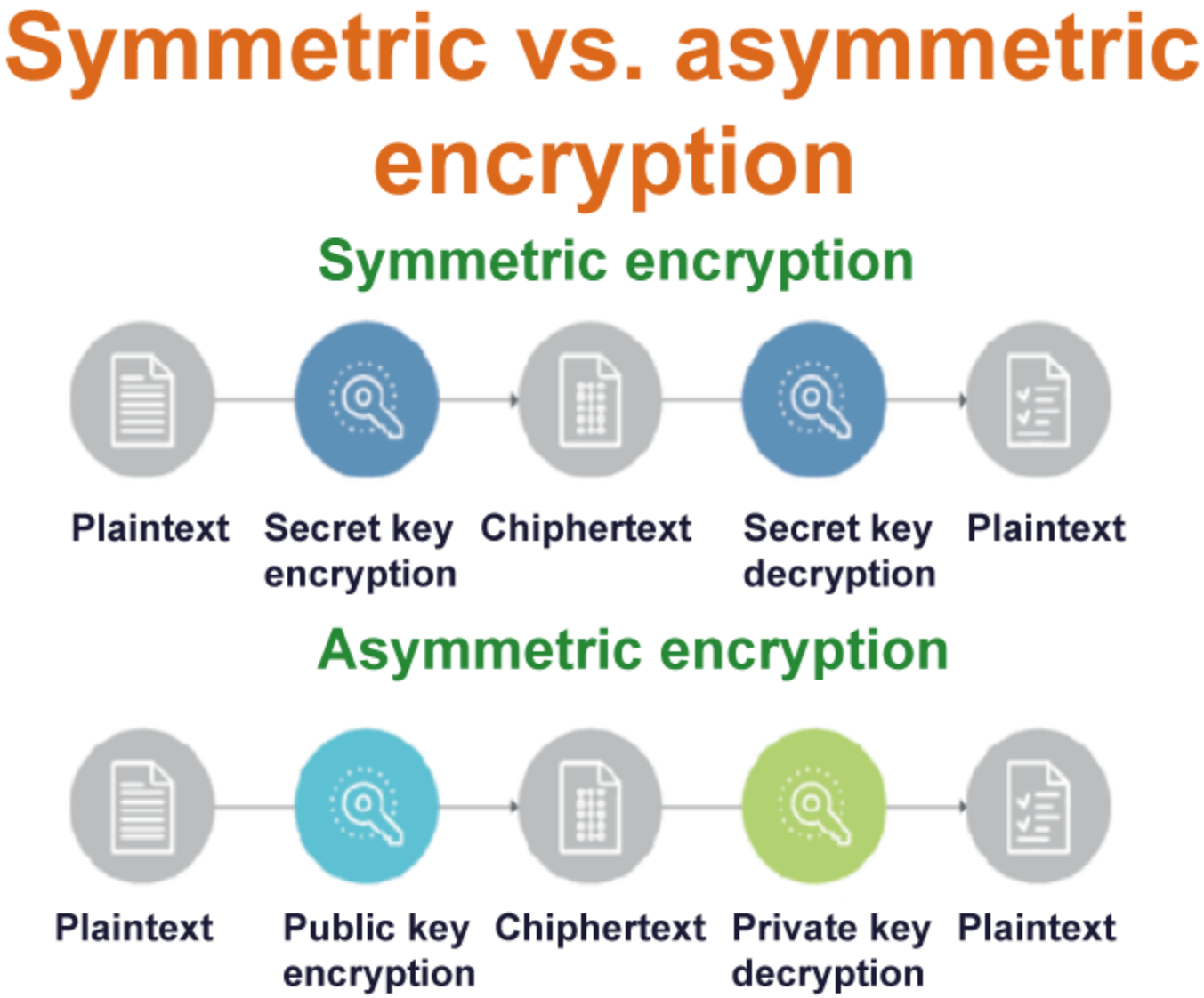
- Email messages
- SIM card authentication
- Web security
- Exchange of private keys

Types of Cryptography



There are three main types of cryptography:

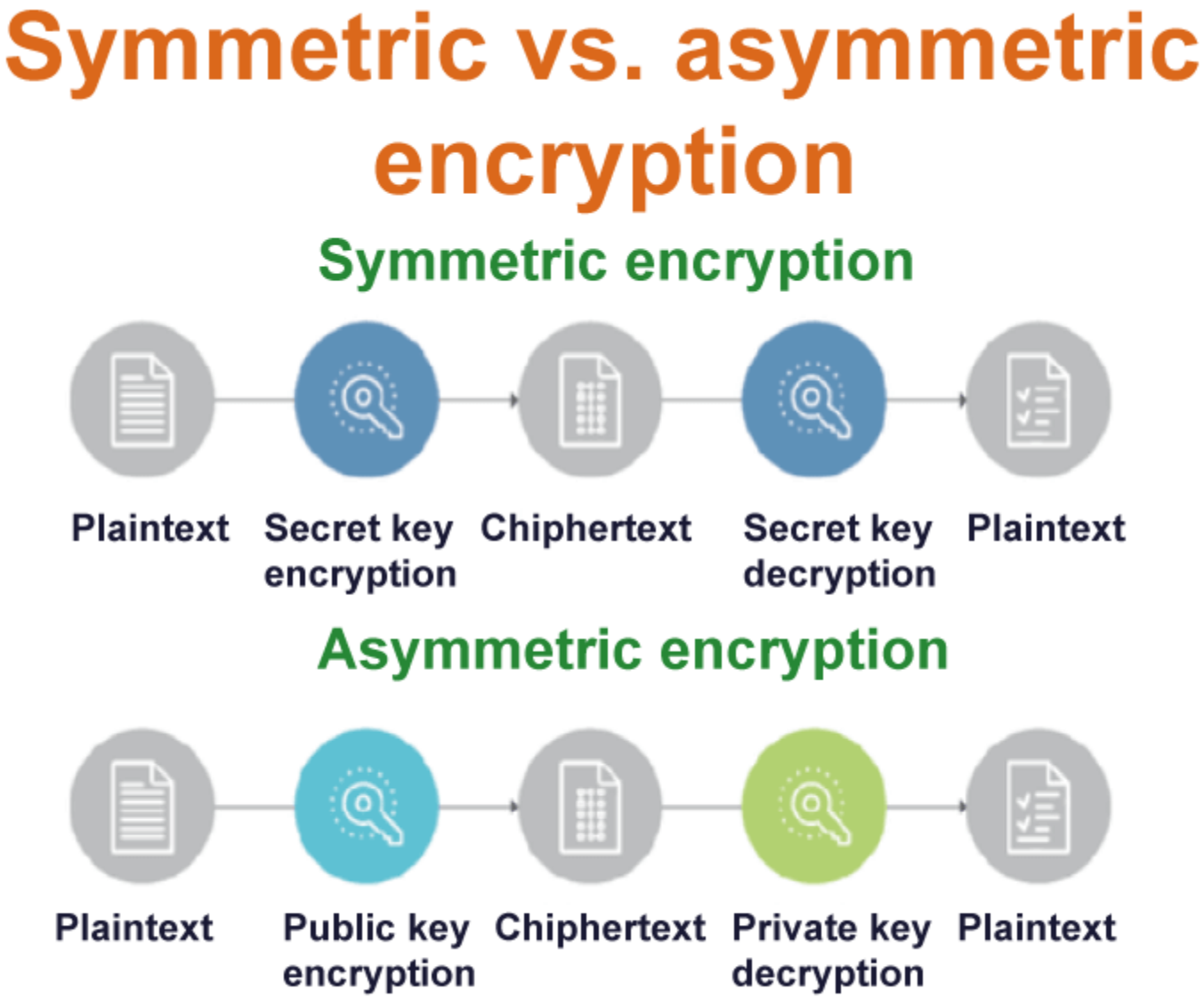
Symmetric key Cryptography: With the encryption technique, the sender and the recipient use the same shared key to encrypt and decrypt messages.



Although symmetric key systems are quicker and easier to use, they have the drawback of requiring a secure key exchange between the sender and the receiver. Data Encryption System (DES) is the most widely used symmetric key encryption method.

Hash Functions: In this algorithm, no key is used. The plain text is used to produce a hash value that has a fixed length, making it challenging to retrieve the plain text's information. Hash functions are widely used by operating systems to encrypt passwords.

Asymmetric Key Cryptography: This approach uses a set of keys to encrypt and decrypt data. Public keys are used for encryption, whereas private keys are used for decryption.



The Public Key and Private Key are different from one another. Even if everyone knows the public key, only the intended recipient may decode the message since only he can access the private key.

Features of Cryptography

Cryptography has the following features:

- **Confidentiality:** The only person who can access information is the one it is intended for, which is the primary feature of cryptography.
- **Integrity:** Information cannot be altered while it is being stored or sent from the sender to the intended destination without the recipient spotting the addition of new information in Cryptography.

- **Non-repudiation:** The creator/sender of a message cannot deny his intent to send information at a future point.
- **Authentication:** The identities of the sender and the recipient have been confirmed. Furthermore, the information's source and final destination are confirmed.
- **Availability:** It also ensures that the required information is available to authorized users at the appropriate time.
- **Key Management:** The creation, distribution, storage, and alteration of cryptographic keys take place in this process.
- **Algorithm:** Mathematical formulae are used in cryptography to encrypt and decrypt messages.
- **Digital Signatures:** A signature that can be applied to messages to protect the message's authenticity and sender identification.

Applications of Cryptography

- **Computer passwords:** Cryptography is frequently used in computer security, especially when creating and managing passwords. When users log in, their password is hashed and contrasted with the previously saved hash. To store them, passwords are first hashed and encrypted. This method encrypts the passwords so that even if hackers can access the password database, they can't comprehend the passwords.
- **Digital Currencies:** Cryptography is also used by digital currencies like Bitcoin to secure transactions and prevent fraud. Since advanced algorithms and cryptographic keys safeguard transactions, tampering with or creating fake transactions is practically impossible.
- **Secure web browsing:** Cryptography protects users from eavesdropping in on their conversations and man-in-the-middle attacks and provides online browsing security. The Secure Sockets Layer (SSL) and Transport Layer Security (TLS) protocols use public key cryptography to encrypt data between the web server and the client, creating a secure communication channel.
- **Digital signatures:** Digital signatures are used to sign papers and act as the handwritten signature's digital copy. Cryptography is used to create digital signatures, and public key cryptography is used to verify them. Digital signatures are becoming more widely used, and many countries have laws that make them legally binding.
- **Authentication:** When logging into a computer, cryptography is employed as the authentication method, for example, a bank account or a secure network. The authentication protocols use cryptographic techniques to validate the user's identity and possession of the necessary access privileges to the resource.

- **Cryptocurrencies:** Cryptocurrencies like Bitcoin and Ethereum largely rely on cryptography to protect transactions, prevent fraud, and uphold the integrity of the network. Transactions are protected by complicated algorithms and cryptographic keys, making it nearly impossible to tamper with or fake transactions.
- **End-to-End Encryption:** Email, instant messages, and video chats are all examples of two-way communications protected by end-to-end encryption. Even if a message is encrypted, this guarantees that only the intended recipients can decode it. End-to-end encryption is frequently employed in messaging apps like WhatsApp and Signal, offering users high protection and anonymity.

Application Layer

Application Layer provides a facility by which users can forward several emails and it also provides a storage facility.

- This layer allows users to access, retrieve and manage files in a remote computer.
- It allows users to log on as a remote host.
- This layer provides access to global information about various services.
- This layer provides services which include: e-mail, transferring files, distributing results to the user, directory services, network resources and so on.
- It provides protocols that allow software to send and receive information and present meaningful data to users.
- It handles issues such as network transparency, resource allocation and so on.
- This layer serves as a window for users and application processes to access network services.
-

Working of Application Layer in the OSI model :
In the OSI model, this application layer is narrower in scope.
The application layer in the OSI model generally acts only like the interface which is responsible for communicating with host-based and user-facing applications. This is in contrast with TCP/IP protocol, wherein the layers below the application layer, which is Session Layer and Presentation layer, are clubbed together and form a simple single layer which is responsible for performing the functions, which includes controlling the dialogues between computers, establishing as well as maintaining as well as ending a particular session, providing data compression and data encryption and so on.

Design Issues with Application Layer

In the design and implementation of **Application Layer** protocols occurring problems and these problems can be addressed by patterns from several different pattern languages:

- Design (pattern) Language for Application-level Communication Protocols
- Service Design Patterns
- Enterprise Application Architecture's Patterns
- Pattern-Oriented Software Architecture

Functionalities of the Application layer

Specific **functionalities of the Application layer** are as follows:

1. Network Virtual terminal

- The application layer is the software version of a physical terminal and this layer permitted to a user to log on to a remote host.
- For this, an application creates a software emulation of a terminal at the remote host. By this user's computer can communicate with the software terminal, which in turn, communicates with the host.
- It is shown that the remote host is communicating with one of its terminals, so it allows the user to log on.

2. File Transfer, Access, and Management (FTAM)

- An application permits a user to access files in a remote computer, to retrieve files from a computer and to manage files on a remote computer.
- FTAM is concerned with a hierarchical virtual file in terms of file attributes, file structure and the types of operations performed on the files and their attributes.

3. Addressing

- To achieve communication between client and server system, there is a need for addressing.
- When a request is sent from the client side to the server side, this request contains the server address and its own address.
- The server answered to the client request, this request contains the destination address, i.e., client address. DNS is used to achieve this type of addressing.

4. Mail Services

Email forwarding and storage of e-mails provided by an application layer.

5. Directory Services

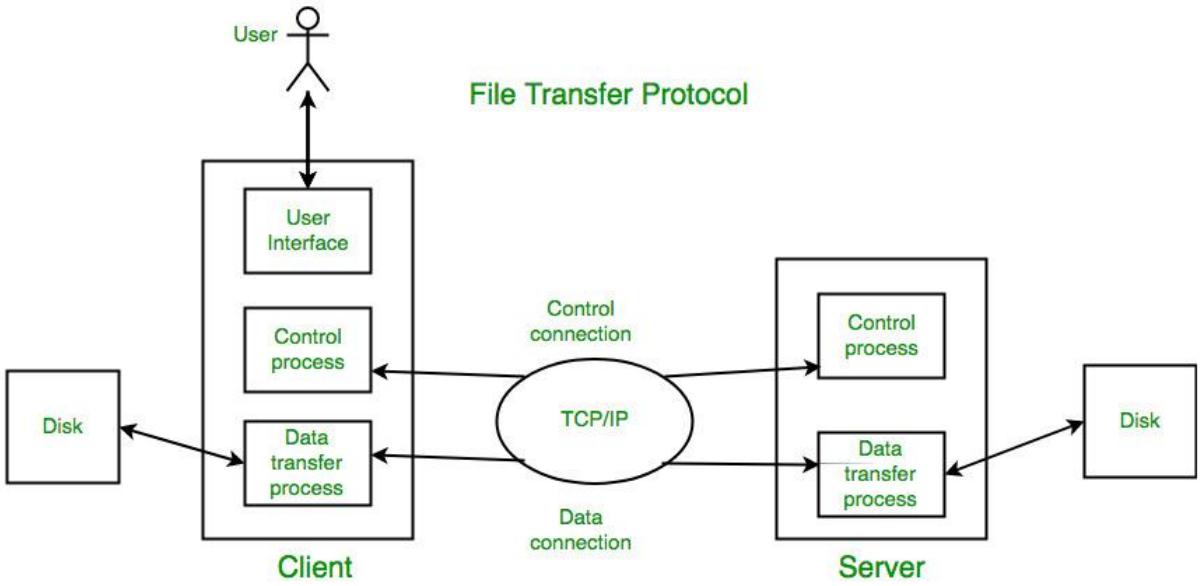
A distributed database is contained by an application that provides access for global information about various objects and services.

6. Authentication

It provides authentication to occur between devices for an extra layer of security and it authenticates the sender or receiver's message or both.

File Transfer Protocol (FTP) in Application Layer

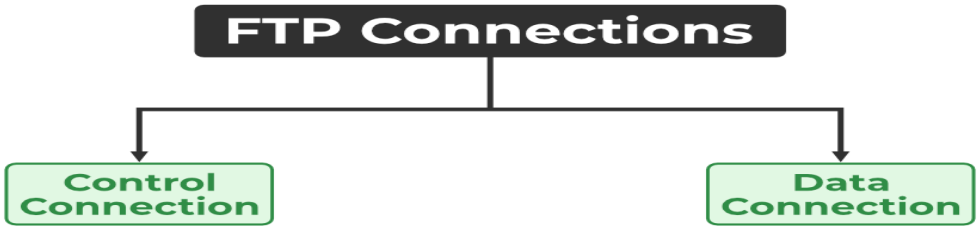
File Transfer Protocol(FTP) is an application layer protocol that moves files between local and remote file systems. It runs on top of TCP, like HTTP. To transfer a file, 2 TCP connections are used by FTP in parallel: control connection and data connection.



Mechanism of File Transfer Protocol

Types of Connection in FTP

1. Control Connection
2. Data Connection



Types of Connection in FTP

1. Control Connection: For sending control information like user identification, password, commands to change the remote directory, commands to retrieve and store files, etc., FTP makes use of a control connection. The control connection is initiated on port number 21.
2. Data connection: For sending the actual file, FTP makes use of a data connection. A data connection is initiated on port number 20.
FTP sends the control information out-of-band as it uses a separate control connection. Some protocols send their request and response header lines and the data in the same TCP connection. For this reason, they are said to send their control information in-band. HTTP and SMTP are such examples.

FTP Session

When an FTP session is started between a client and a server, the client initiates a control TCP connection with the server side. The client sends control information over this. When the server receives this, it initiates a data connection to the client side. Only one file can be sent over one data connection. But the control connection remains active throughout the user session. As we know HTTP is stateless i.e. it does not have to keep track of any user state. But FTP needs to maintain a state about its user throughout the session.

FTP Clients

FTP works on a client-server model. The FTP client is a program that runs on the user's computer to enable the user to talk to and get files from remote computers. It is a set of commands that establishes the connection between two hosts, helps to transfer the files, and then closes the connection.

FTP Data Structures

FTP allows three types of data structures :

1. **File Structure:** In file structure, there is no internal structure and the file is considered to be a continuous sequence of data bytes.
2. **Record Structure:** In record structure, the file is made up of sequential records.
3. **Page Structure:** In page structure, the file is made up of independent indexed pages.

FTP Commands

Some of the FTP commands are:

- USER – This command sends the user identification to the server.
- PASS – This command sends the user password to the server.
- CWD – This command allows the user to work with a different directory or dataset for file storage or retrieval without altering his login or accounting information.
    - RMD – This command causes the directory specified in the path name to be removed as a directory.

Advantages of FTP

- Speed is one of the advantages of FTP(File Transfer Protocol).
- File sharing also comes in the category of advantages of FTP in this between two machines files can be shared on the network.
- Efficiency is more in FTP.

Disadvantages of FTP

- File size limit is the drawback of FTP only 2 GB size files can be transferred.
- Multiple receivers are not supported by the FTP.
- FTP does not encrypt the data this is one of the biggest drawbacks of FTP.
- FTP is unsecured we use login IDs and passwords making it secure but they can be attacked by hackers.

Virtual terminal definition

A virtual terminal is a text-based interface within a graphical user interface (GUI) or another program that enables users to access a computer or server. Virtual terminals essentially replicate the functionality of hardware terminals without the need for physical devices. So, the process usually occurs remotely over a network connection. A virtual terminal has a command-line interface (CLI) and uses protocols such as SSH or Telnet so two different operating systems can communicate. Users usually use virtual terminals alongside operating systems like Unix, Linux, and Windows for system administration tasks, running command-line tools, and managing servers. In e-commerce, a virtual terminal refers to the software that processes card payments.

Virtual terminal examples

- **PuTTY.** An open-source software initially created for Windows that provides network tools for secure remote access and file transfers through protocols such as SSH and Telnet.

- **SecureCRT.** A commercial emulator that supports several network protocols, including Rlogin, SSH, and Telnet.
- **Tera Term.** An open-sourced, free terminal emulator that works on Windows and supports SSH, Telnet, and other protocols.
- **iTerm2.** An open-sourced, free emulator that is designed for macOS, supporting a line of network protocols.

Virtual terminal benefits

- Convenient access and management of systems and networks from anywhere.
- Can be integrated with other software applications and tools, such as code editors.
- Help businesses reduce maintenance costs because they don’t require physical terminals.
- Increase productivity by allowing more efficient work and remote access to data.
- Allow users to open several terminal sessions in one tab and manage multiple terminal sessions at once.
- May offer customizable services, such as configuring the terminal window size, font, and keyboard shortcuts.