UNIT-IV

I. Network and Transport Layer

Network layer is the third layer in the OSI model of computer networks. Its main function is to transfer network packets from the source to the destination. It is involved both at the source host and the destination host. At the source, it accepts a packet from the transport layer, encapsulates it in a datagram and then deliver the packet to the data link layer so that it can further be sent to the receiver. At the destination, the datagram is decapsulated, the packet is extracted and delivered to the corresponding transport layer.

Features :

1. Main responsibility of Network layer is to carry the data packets from the source to the destination without changing or using it.
2. If the packets are too large for delivery, they are fragmented i.e., broken down into smaller packets.
3. It decides the route to be taken by the packets to travel from the source to the destination among the multiple routes available in a network (also called as routing).
4. The source and destination addresses are added to the data packets inside the network layer.

The **services** which are offered by the network layer protocol are as follows:

Packetizing

The process of encapsulating the data received from upper layers of the network (also called as payload) in a network layer packet at the source and decapsulating the payload from the network layer packet at the destination is known as packetizing.

The source host adds a header that contains the source and destination address and some other relevant information required by the network layer protocol to the payload received from the upper layer protocol, and delivers the packet to the data link layer.

The destination host receives the network layer packet from its data link layer, decapsulates the packet, and delivers the payload to the corresponding upper layer protocol. The routers in the path are not allowed to change either the source or the destination address. The routers in the path are not allowed to decapsulate the packets they receive unless they need to be fragmented.

Routing and forwarding –

These are two other services offered by the network layer. In a network, there are a number of routes available from the source to the destination. The network layer specifies has some strategies which find out the best possible route. This process is referred to as routing. There are a number of routing protocols which are used in this process and they should be run to help the routers coordinate with each other and help in establishing communication throughout the network.

Forwarding is simply defined as the action applied by each router when a packet arrives at one of its interfaces. When a router receives a packet from one of its attached networks, it needs to forward the packet to another attached network (unicast routing) or to some attached networks(in case of multicast routing).

Some of the other **services which are expected** from the network layer are:

1. **Error Control –**
   Although it can be implemented in the network layer, but it is usually not preferred because the data packet in a network layer maybe fragmented at each router, which makes error checking inefficient in the network layer.

2. **Flow Control –**
   It regulates the amount of data a source can send without overloading the receiver. If the source produces a data at a very faster rate than the receiver can consume it, the receiver will be overloaded with data. To control the flow of data, the receiver should send a feedback to the sender to inform the latter that it is overloaded with data.
   There is a lack of flow control in the design of the network layer. It does not directly provide any flow control. The datagrams are sent by the sender when they are ready, without any attention to the readiness of the receiver.

3. **Congestion Control –**
   Congestion occurs when the number of datagrams sent by source is beyond the capacity of network or routers. This is another issue in the network layer protocol. If congestion continues, sometimes a situation may arrive where the system collapses and no datagrams are delivered. Although congestion control is indirectly implemented in network layer, but still there is a lack of congestion control in the network layer.

Advantages of Network Layer Services :

- Packetization service in network layer provides an ease of transportation of the data packets.
- Packetization also eliminates single points of failure in data communication systems.
- Routers present in the network layer reduce network traffic by creating collision and broadcast domains.
- With the help of Forwarding, data packets are transferred from one place to another in the network.

Disadvantages of Network Layer Services :

- There is a lack of flow control in the design of the network layer.
- Congestion occurs sometimes due to the presence of too many datagrams in a network which are beyond the capacity of network or the routers. Due to this, some routers may drop some of the datagrams and some important piece of information maybe lost.
- Although indirectly error control is present in network layer, but there is a lack of proper error control mechanisms as due to presence of fragmented data packets, error control becomes difficult to implement.

****

II. Internet Protocol Version 4 (IPv4)

Internet Protocol is one of the major protocols in the TCP/IP protocols suite. This protocol works at the network layer of the OSI model and at the Internet layer of the TCP/IP model. Thus this protocol has the responsibility of identifying hosts based upon their logical addresses and to route data among them over the underlying network.

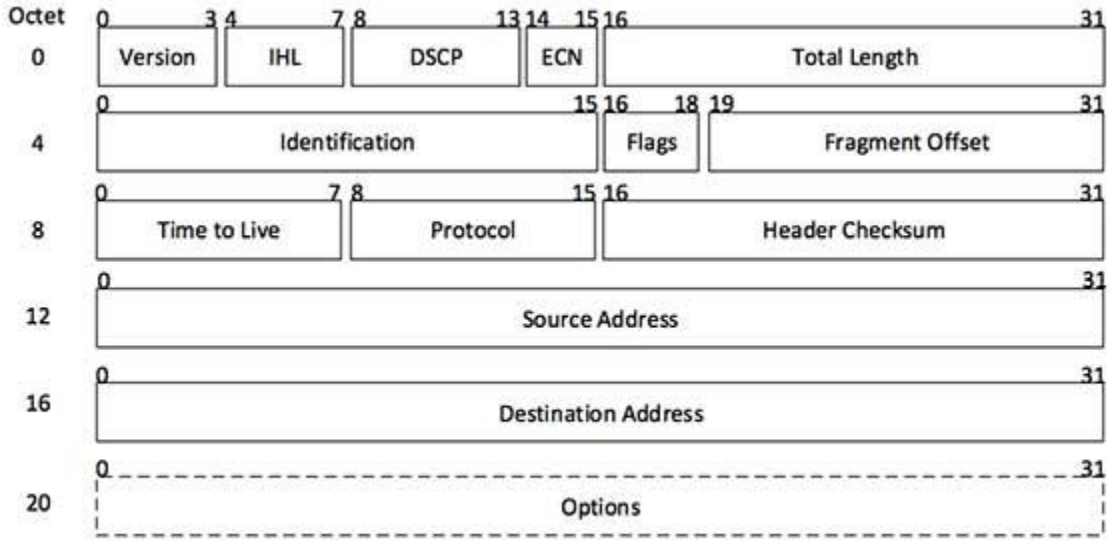
IP provides a mechanism to uniquely identify hosts by an IP addressing scheme. IP uses best effort delivery, i.e. it does not guarantee that packets would be delivered to the destined host, but it will do its best to reach the destination. Internet Protocol version 4 uses 32-bit logical address.

IPv4 - Packet Structure

Internet Protocol being a layer-3 protocol (OSI) takes data Segments from layer-4 (Transport) and divides it into packets. IP packet encapsulates data unit received from above layer and add to its own header information.

(IP Encapsulation)

The encapsulated data is referred to as IP Payload. IP header contains all the necessary information to deliver the packet at the other end.



[Image: IP Header]

IP header includes many relevant information including Version Number, which, in this context, is 4. Other details are as follows −

- **Version** − Version no. of Internet Protocol used (e.g. IPv4).
- **IHL** − Internet Header Length; Length of entire IP header.
- **DSCP** − Differentiated Services Code Point; this is Type of Service.
- **ECN** − Explicit Congestion Notification; It carries information about the congestion seen in the route.
- **Total Length** − Length of entire IP Packet (including IP header and IP Payload).

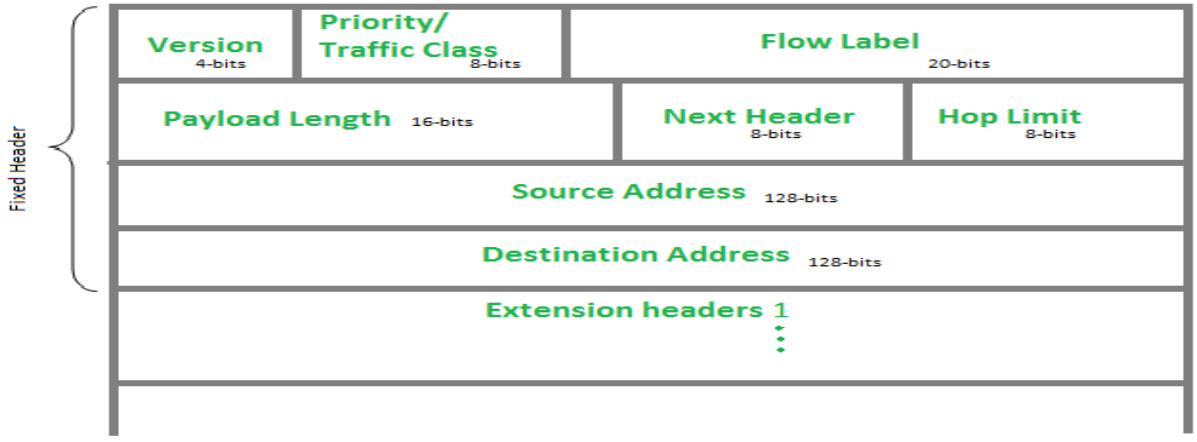
- **Identification** − If IP packet is fragmented during the transmission, all the fragments contain same identification number. to identify original IP packet they belong to.
- **Flags** − As required by the network resources, if IP Packet is too large to handle, these 'flags' tells if they can be fragmented or not. In this 3-bit flag, the MSB is always set to '0'.
- **Fragment Offset** − This offset tells the exact position of the fragment in the original IP Packet.
- **Time to Live** − To avoid looping in the network, every packet is sent with some TTL value set, which tells the network how many routers (hops) this packet can cross. At each hop, its value is decremented by one and when the value reaches zero, the packet is discarded.
- **Protocol** − Tells the Network layer at the destination host, to which Protocol this packet belongs to, i.e. the next level Protocol. For example protocol number of ICMP is 1, TCP is 6 and UDP is 17.
- **Header Checksum** − This field is used to keep checksum value of entire header which is then used to check if the packet is received error-free.
- **Source Address** − 32-bit address of the Sender (or source) of the packet.
- **Destination Address** − 32-bit address of the Receiver (or destination) of the packet.
- **Options** − This is optional field, which is used if the value of IHL is greater than 5. These options may contain values for options such as Security, Record Route, Time Stamp, etc.

****

III. Internet Protocol version 6 (IPv6)- Packet Structure

IP version 6 is the new version of Internet Protocol, which is way better than IP version 4 in terms of complexity and efficiency. Let's look at the header of IP version 6 and understand how it is different from the IPv4 header.

IP version 6 Header Format :



Version (4-bits): Indicates version of Internet Protocol which contains bit sequence 0110.

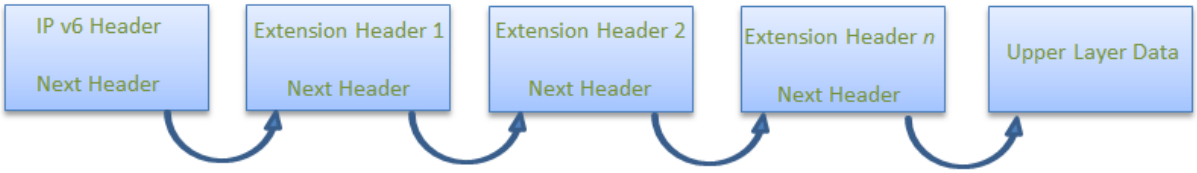
Traffic Class (8-bits): The Traffic Class field indicates class or priority of IPv6 packet which is similar to *Service Field* in IPv4 packet. It helps routers to handle the traffic based on the priority of the packet. If congestion occurs on the router then packets with the least priority will be discarded.
As of now, only 4-bits are being used (and the remaining bits are under research), in which 0 to 7 are assigned to Congestion controlled traffic and 8 to 15 are assigned to Uncontrolled traffic.
Priority assignment of Congestion controlled traffic :

Priority	Meaning
0	No Specific traffic
1	Background data
2	Unattended data traffic
3	Reserved
4	Attended bulk data traffic
5	Reserved
6	Interactive traffic
7	Control traffic

Uncontrolled data traffic is mainly used for Audio/Video data. So we give higher priority to Uncontrolled data traffic.
The source node is allowed to set the priorities but on the way, routers can change it. Therefore, the destination should not expect the same priority which was set by the source node.

Flow Label (20-bits): Flow Label field is used by a source to label the packets belonging to the same flow in order to request special handling by intermediate IPv6 routers, such as non-default quality of service or real-time service. In order to distinguish the flow, an intermediate router can use the source address, a destination address, and flow label of the packets. Between a source and destination, multiple flows may exist because many processes might be running at the same time. Routers or Host that does not support the functionality of flow label field and for default router handling, flow label field is set to 0. While setting up the flow label, the source is also supposed to specify the lifetime of the flow.
Payload Length (16-bits): It is a 16-bit (unsigned integer) field, indicates the total size of the payload which tells routers about the amount of information a particular packet contains in its payload. The payload Length field includes extension headers(if any) and an upper-layer packet. In case the length of the payload is greater than 65,535 bytes (payload up to 65,535 bytes can be indicated with 16-bits), then the payload length field will be set to 0 and the jumbo payload option is used in the Hop-by-Hop options extension header.
Next Header (8-bits): Next Header indicates the type of extension header(if present) immediately following the IPv6 header. Whereas In some cases it indicates the protocols contained within upper-layer packets, such as TCP, UDP.
Hop Limit (8-bits): Hop Limit field is the same as TTL in IPv4 packets. It indicates the maximum number of intermediate nodes IPv6 packet is allowed to travel. Its value gets decremented by one, by each node that forwards the packet and the packet is discarded if the value decrements to 0. This is used to discard the packets that are stuck in an infinite loop because of some routing error.

Source Address (128-bits): Source Address is the 128-bit IPv6 address of the original source of the packet.
Destination Address (128-bits): The destination Address field indicates the IPv6 address of the final destination(in most cases). All the intermediate nodes can use this information in order to correctly route the packet.
Extension Headers: In order to rectify the limitations of the *IPv4 Option Field*, Extension Headers are introduced in IP version 6. The extension header mechanism is a very important part of the IPv6 architecture. The next Header field of IPv6 fixed header points to the first Extension Header and this first extension header points to the second extension header and so on.



Differences between IPv4 and IPv6

IPv4 and IPv6 are internet protocol version 4 and internet protocol version 6, IP version 6 is the new version of Internet Protocol, which is way better than IP version 4 in terms of complexity and efficiency.

IPv4	IPv6
IPv4 has a 32-bit address length	IPv6 has a 128-bit address length
It Supports Manual and DHCP address configuration	It supports Auto and renumbering address configuration
In IPv4 end to end, connection integrity is Unachievable	In IPv6 end to end, connection integrity is Achievable
It can generate 4.29×10^{9} address space	Address space of IPv6 is quite large it can produce 3.4×10^{38} address space
The Security feature is dependent on application	IPSEC is an inbuilt security feature in the IPv6 protocol

IPv4	IPv6
Address representation of IPv4 is in decimal	Address Representation of IPv6 is in hexadecimal
Fragmentation performed by Sender and forwarding routers	In IPv6 fragmentation performed only by the sender
In IPv4 Packet flow identification is not available	In IPv6 packet flow identification are Available and uses the flow label field in the header
In IPv4 checksum field is available	In IPv6 checksum field is not available
It has broadcast Message Transmission Scheme	In IPv6 multicast and anycast message transmission scheme is available
In IPv4 Encryption and Authentication facility not provided	In IPv6 Encryption and Authentication are provided
IPv4 has a header of 20-60 bytes.	IPv6 has header of 40 bytes fixed
IPv4 consist of 4 fields which are separated by dot (.)	IPv6 consist of 8 fields, which are separated by colon (:)
IPv4's IP addresses are divided into five different classes. Class A , Class B, Class C , Class D , Class E.	IPv6 does not have any classes of IP address.
IPv4 supports VLSM(Variable Length subnet mask).	IPv6 does not support VLSM.
Example of IPv4: 66.94.29.13	Example of IPv6: 2001:0000:3238:DFE1:0063:0000:0000:FEFB

****

IV. IPv4 - Addressing

An IP address is a unique address that identifies a device on the internet or a local network. IP stands for "Internet Protocol," which is the set of rules governing the format of data sent via the internet or local network.

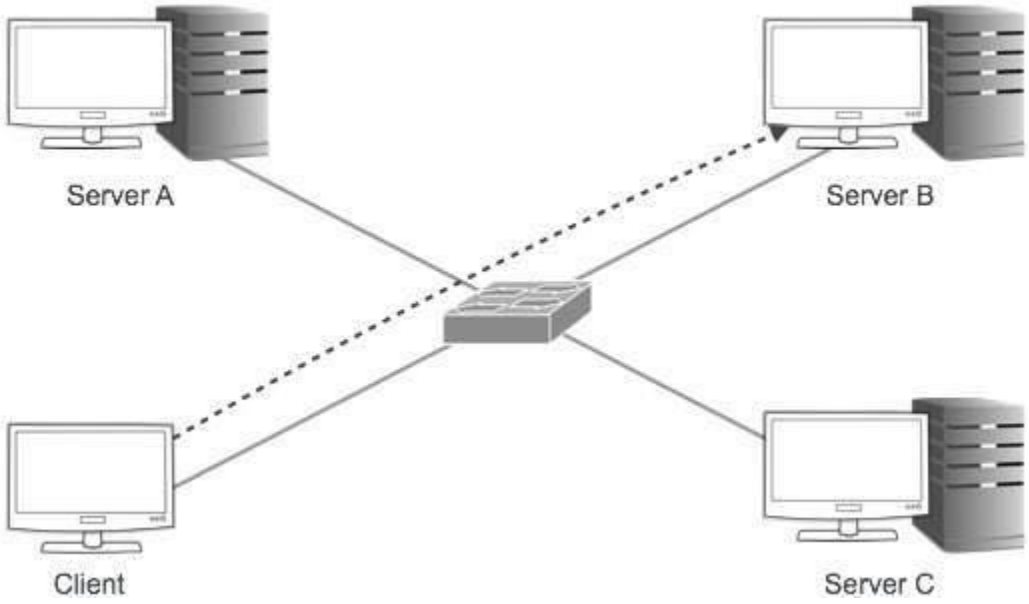
In essence, IP addresses are the identifier that allows information to be sent between devices on a network: they contain location information and make devices accessible for communication. The internet needs a way to differentiate between different computers, routers, and websites. IP addresses provide a way of doing so and form an essential part of how the internet works.

An IP address is a string of numbers separated by periods. IP addresses are expressed as a set of four numbers — an example address might be 192.158.1.38. Each number in the set can range from 0 to 255. So, the full IP addressing range goes from 0.0.0.0 to 255.255.255.255.

IPv4 supports three different types of addressing modes. −

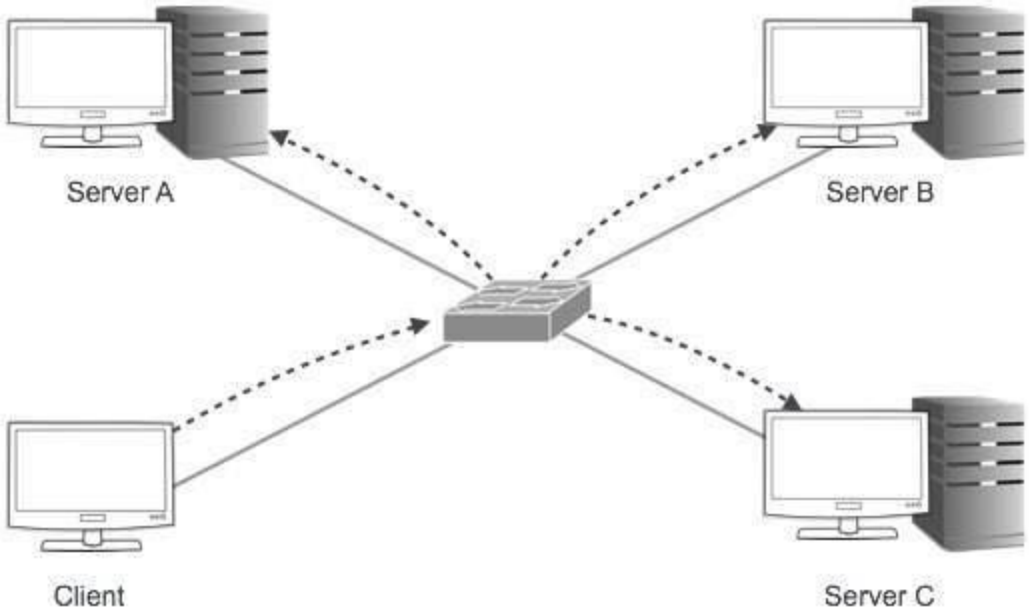
Unicast Addressing Mode

In this mode, data is sent only to one destined host. The Destination Address field contains 32- bit IP address of the destination host. Here the client sends data to the targeted server −



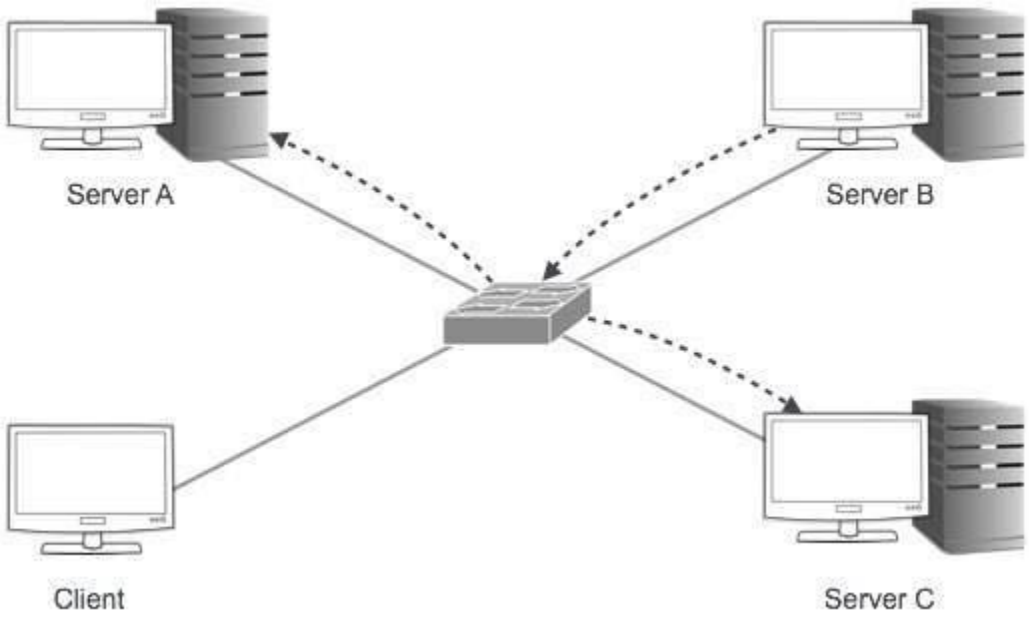
Broadcast Addressing Mode

In this mode, the packet is addressed to all the hosts in a network segment. The Destination Address field contains a special broadcast address, i.e. **255.255.255.255**. When a host sees this packet on the network, it is bound to process it. Here the client sends a packet, which is entertained by all the Servers −



Multicast Addressing Mode

This mode is a mix of the previous two modes, i.e. the packet sent is neither destined to a single host nor all the hosts on the segment. In this packet, the Destination Address contains a special address which starts with 224.x.x.x and can be entertained by more than one host.



Here a server sends packets which are entertained by more than one servers. Every network has one IP address reserved for the Network Number which represents the network and one IP address reserved for the Broadcast Address, which represents all the hosts in that network.

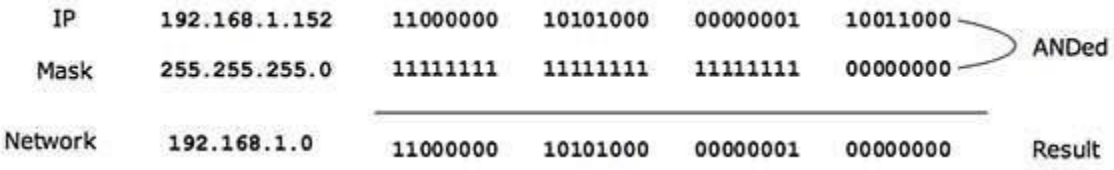
Hierarchical Addressing Scheme

IPv4 uses hierarchical addressing scheme. An IP address, which is 32-bits in length, is divided into two or three parts as depicted −

A single IP address can contain information about the network and its sub-network and ultimately the host. This scheme enables the IP Address to be hierarchical where a network can have many sub-networks which in turn can have many hosts.

Subnet Mask

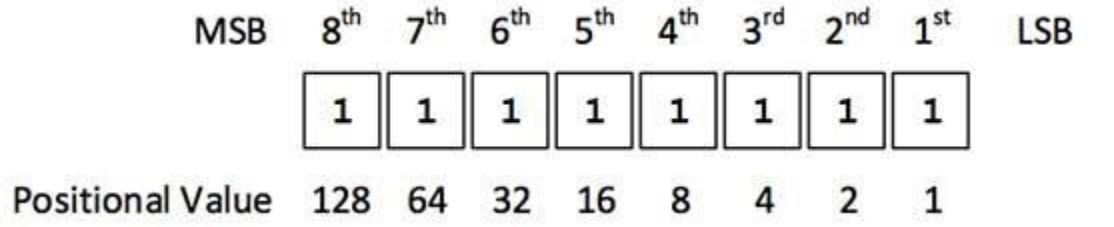
The 32-bit IP address contains information about the host and its network. It is very necessary to distinguish both. For this, routers use Subnet Mask, which is as long as the size of the network address in the IP address. Subnet Mask is also 32 bits long. If the IP address in binary is ANDed with its Subnet Mask, the result yields the Network address. For example, say the IP Address is 192.168.1.152 and the Subnet Mask is 255.255.255.0 then −



This way the Subnet Mask helps extract the Network ID and the Host from an IP Address. It can be identified now that 192.168.1.0 is the Network number and 192.168.1.152 is the host on that network.

Binary Representation

The positional value method is the simplest form of converting binary from decimal value. IP address is 32 bit value which is divided into 4 octets. A binary octet contains 8 bits and the value of each bit can be determined by the position of bit value '1' in the octet.



Positional value of bits is determined by 2 raised to power (position – 1), that is the value of a bit 1 at position 6 is 2^(6-1) that is 2^5 that is 32. The total value of the octet is determined by adding up the positional value of bits. The value of 11000000 is 128+64 = 192. Some examples are shown in the table below −

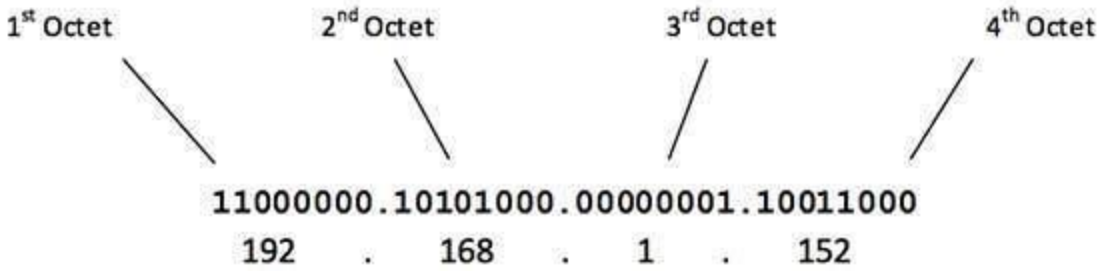
128	64	32	16	8	4	2	1	Value
0	0	0	0	0	0	0	1	1
0	0	0	0	0	0	1	0	2
0	0	0	0	0	0	1	1	3
0	0	0	0	0	1	0	0	4
0	0	0	0	0	1	0	1	5
0	0	0	0	0	1	1	0	6
0	0	0	0	0	1	1	1	7
0	0	0	0	1	0	0	0	8
0	0	0	0	1	0	0	1	9
0	0	0	0	1	0	1	0	10
0	0	0	1	0	0	0	0	16
0	0	1	0	0	0	0	0	32
0	1	0	0	0	0	0	0	64
0	1	1	0	0	1	0	0	100
0	1	1	1	1	1	1	1	127
1	0	0	0	0	0	0	0	128
1	0	1	0	1	0	0	0	168
1	1	0	0	0	0	0	0	192
1	1	1	1	1	1	1	1	255

IPv4 - Address Classes

Internet Protocol hierarchy contains several classes of IP Addresses to be used efficiently in various situations as per the requirement of hosts per network. Broadly, the IPv4 Addressing system is divided into five classes of IP Addresses. All the five classes are identified by the first octet of IP Address.

Internet Corporation for Assigned Names and Numbers is responsible for assigning IP addresses.

The first octet referred here is the left most of all. The octets numbered as follows depicting dotted decimal notation of IP Address −



The number of networks and the number of hosts per class can be derived by this formula −

Number of networks = 2^network_bits

Number of Hosts/Network = 2^host_bits – 2

When calculating hosts' IP addresses, 2 IP addresses are decreased because they cannot be assigned to hosts, i.e. the first IP of a network is network number and the last IP is reserved for Broadcast IP.

IP Class	Address Range	Maximum number of networks
Class A	0-126	126 (2^7-2)
Class B	128-191	16384
Class C	192-223	2097152
Class D	224-239	Reserve for multitasking
Class E	240-254	Reserved for Research and development

Class A Address

The first bit of the first octet is always set to 0 (zero). Thus the first octet ranges from 1 – 127, i.e.

00000001 – 01111111
1 – 127

Class A addresses only include IP starting from 1.x.x.x to 126.x.x.x only. The IP range 127.x.x.x is reserved for loopback IP addresses.

The default subnet mask for Class A IP address is 255.0.0.0 which implies that Class A addressing can have 126 networks (2^7-2) and 16777214 hosts (2^{24}-2).

Class A IP address format is thus: **0NNNNNNN**.HHHHHHHH.HHHHHHHH.HHHHHHHH

Class B Address

An IP address which belongs to class B has the first two bits in the first octet set to 10, i.e.

10000000 – 10111111
128 – 191

Class B IP Addresses range from 128.0.x.x to 191.255.x.x. The default subnet mask for Class B is 255.255.x.x.

Class B has 16384 (2^{14}) Network addresses and 65534 (2^{16}-2) Host addresses.

Class B IP address format is: **10NNNNNN.NNNNNNNN**.HHHHHHHH.HHHHHHHH

Class C Address

The first octet of Class C IP address has its first 3 bits set to 110, that is −

11000000 – 11011111
192 – 223

Class C IP addresses range from 192.0.0.x to 223.255.255.x. The default subnet mask for Class C is 255.255.255.x.

Class C gives 2097152 (2^{21}) Network addresses and 254 (2^{8}-2) Host addresses.

Class C IP address format is: **110NNNNN.NNNNNNNN.NNNNNNNN**.HHHHHHHH

Class D Address

Very first four bits of the first octet in Class D IP addresses are set to 1110, giving a range of −

11100000 – 11101111
224 – 239

Class D has IP address range from 224.0.0.0 to 239.255.255.255. Class D is reserved for Multicasting. In multicasting data is not destined for a particular host, that is why there is no need to extract host address from the IP address, and Class D does not have any subnet mask.

Class E Address

This IP Class is reserved for experimental purposes only for R&D or Study. IP addresses in this class ranges from 240.0.0.0 to 255.255.255.254. Like Class D, this class too is not equipped with any subnet mask.

IPv6: But, there is a problem with the IPv4 address. With IPv4, we can connect only the above number of 4 billion devices uniquely, and apparently, there are much more devices in the world

to be connected to the internet. So, gradually we are making our way to **IPv6 Address** which is a 128-bit IP address. In human-friendly form, IPv6 is written as a group of 8 hexadecimal numbers separated with colons(:). But in the computer-friendly form, it can be written as 128 bits of 0s and 1s. Since, a unique sequence of binary digits is given to computers, smartphones, and other devices to be connected to the internet. So, via IPv6 a total of (2^128) devices can be assigned with unique addresses which are actually more than enough for upcoming future generations.

IPv6 can be written as:

2011:0bd9:75c5:0000:0000:6b3e:0170:8394

IPv4 Address Allocation

The Internet Protocol address can be allocated to hosts or interfaces either manually or dynamically.

- **Static** – static IP address is set manually on the device. It is best practice to set static IP addresses on network devices, such as routers and switches, and on servers as well.
- **Dynamic** – dynamic IP address can be automatically allocated to a device via Dynamic Host Configuration Protocol (DHCP). Dynamic IP addresses are best to be used on end devices, such as PCs.

Types of IPv4 Addresses

We have two types of IP addresses, namely public IP addresses and private IP addresses.

- **Public IP address** – used to route Internet traffic. This is used on the Internet and is given out by Internet Service Providers (ISPs) to their customers.
- **Private IP address** – used in private networks for internal traffics within the LAN. Private addresses are not routable out the Internet.

****

V. Internetworking -TCP / IP

TCP Protocol stands for Transmission Control Protocol. It is a connection oriented and reliable protocol. Connection oriented mean the connection remains established until the message has been exchanged and after the complete exchange of packet the connection is terminated. It has been put under the Transport layer. It is responsible for breaking of data messages in to packets that are sent by using internet protocol. TCP is also used in remote login i.e. one can get access of the other computer for maintenance or trouble shooting purposes. It is also used in file transfers.

The message that is in bytes (form of 0 and 1) is broken up into chucks which are known as segments. After receiving the segments, the receiver send acknowledgements for segments. TCP

also maintains a timer. If the acknowledgement is not received in time, then the message is resent. Each byte that is transmitted through TCP protocol contains a sequence number. A TCP segment is made up of a segment header and a data section. The header contains 10 fields that must exist and an optional extension field. TCP provides the following facilities:-

1. It groups the bytes in TCP segments and then passes them to IP.

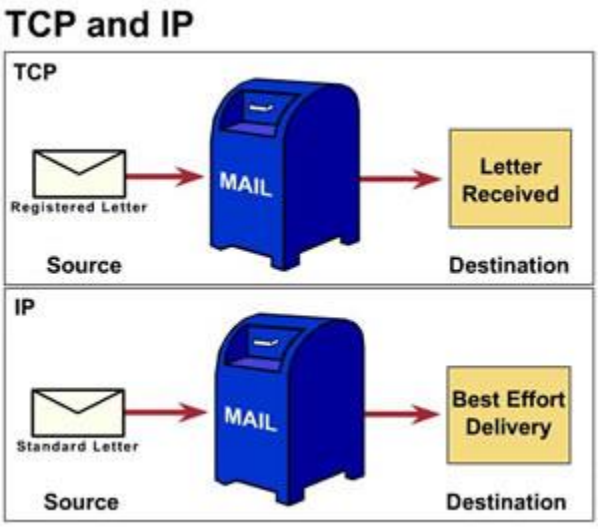
2. With the help of acknowledgments, it provides a greater reliability.

3.The flow of bytes transfer can also be informed with the help of acknowledgements that are sent by the receiver to the sender.

4. It allows multiplexing which means that many processes within a single host can use the facilities of TCP communication.

5. It provides a full duplex mechanism which means that the data can be transferred in both directions at the same time.

IP protocol stand for Internet protocol and it deefines the addresses that are necessary in order to send the data from the source to the destination. It was developed in the 1970s. IP address is used for providing the unique address for computers on a network. As the address is unique it can be used as an identifier to be connected to the other computers. Data is organized into packets also known as datagrams and each IP datagram contains the header and message data. IP protocol makes use of end to end principle in its design and therefore, the system is considered to be unreliable at any single network element or transmission medium. In absence of any central monitoring,the network tends to be more unreliable.



It is a connectionless protocol and it is prone to various error conditions like data corruption, packet loss, duplication and out of order delivery. Connectionless refers to the arrangement where the sender sends the data without determing the availability of receiver. Even if receiver is available, the receiver might not be ready to receive. Situations like this may lead to various problems. However, TCP/IP refers to the combination of TCP and IP protocols.

Some of the key differences have been listed in the table below:-

	TCP	IP
Definition	TCP provides the service of exchanging data between applications	IP handles addressing and routing messages to the computers across one or more networks
Connection	Connection Oriented	Connection less method
location	Transport	Internet
Reliability	Reliable	Unreliable
Transfer	Segments to internet layer	Datagrams to physical level
Flow control	Yes	No
Format	TCP segments have a 20 byte header with >= 0 bytes of data	IP datagrams contain a message, or one fragment of a message, that may be up to 65,535 bytes (octets) in length

****

VI. Network layer design issues

The network layer comes with some design issues they are described as follows:

1. Store and Forward packet switching:

The host sends the packet to the nearest router. This packet is stored there until it has fully arrived once the link is fully processed by verifying the checksum then it is forwarded to the next router till it reaches the destination. This mechanism is called "Store and Forward packet switching."

2. Services provided to Transport Layer**:**

Through the network/transport layer interface, the network layer transfers it's services to the transport layer. These services are described below.

But before providing these services to the transfer layer following goals must be kept in mind :-

- Offering services must not depend on router technology.

- The transport layer needs to be protected from the type, number and topology of the available router.
- The network addresses for the transport layer should use uniform numbering pattern also at LAN and WAN connections.

Based on the connections there are 2 types of services provided :

- **Connectionless –** The routing and insertion of packets into subnet is done individually. No added setup is required.
- **Connection-Oriented –** Subnet must offer reliable service and all the packets must be transmitted over a single route.

3. Implementation of Connectionless Service**:**

Packet are termed as "datagrams" and corresponding subnet as "datagram subnets". When the message size that has to be transmitted is 4 times the size of the packet, then the network layer divides into 4 packets and transmits each packet to router via. a few protocol.Each data packet has destination address and is routed independently irrespective of the packets.

4. Implementation of Connection Oriented service:

To use a connection-oriented service, first we establishes a connection, use it and then release it. In connection-oriented services, the data packets are delivered to the receiver in the same order in which they have been sent by the sender.

It can be done in either two ways :

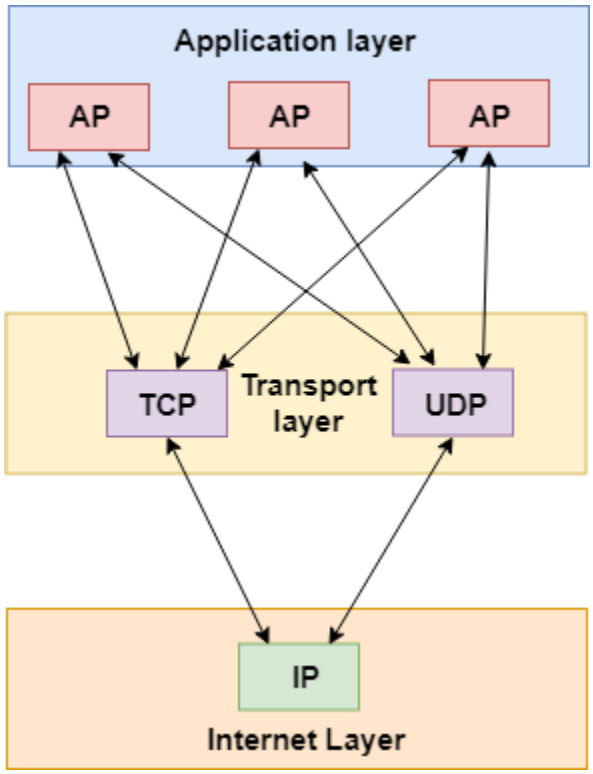
- **Circuit Switched Connection –** A dedicated physical path or a circuit is established between the communicating nodes and then data stream is transferred.
- **Virtual Circuit Switched Connection –** The data stream is transferred over a packet switched network, in such a way that it seems to the user that there is a dedicated path from the sender to the receiver. A virtual path is established here. While, other connections may also be using the same path.

****

VII. DESIGN ISSUES OF TRANSPORT LAYER:-

- The transport layer is a 4^{th} layer from the top.
- The main role of the transport layer is to provide the communication services directly to the application processes running on different hosts.
- The transport layer provides a logical communication between application processes running on different hosts. Although the application processes on different hosts are not physically connected, application processes use the logical communication provided by the transport layer to send the messages to each other.
- The transport layer protocols are implemented in the end systems but not in the network routers.
- A computer network provides more than one protocol to the network applications. For example, TCP and UDP are two transport layer protocols that provide a different set of services to the network layer.

- All transport layer protocols provide multiplexing/demultiplexing service. It also provides other services such as reliable data transfer, bandwidth guarantees, and delay guarantees.
- Each of the applications in the application layer has the ability to send a message by using TCP or UDP. The application communicates by using either of these two protocols. Both TCP and UDP will then communicate with the internet protocol in the internet layer. The applications can read and write to the transport layer. Therefore, we can say that communication is a two-way process.



Services provided by the Transport Layer

The services provided by the transport layer are similar to those of the data link layer. The data link layer provides the services within a single network while the transport layer provides the services across an internetwork made up of many networks. The data link layer controls the physical layer while the transport layer controls all the lower layers.

The services provided by the transport layer protocols can be divided into five categories:

- End-to-end delivery
- Addressing
- Reliable delivery

- Flow control
- Multiplexing

DESIGN ISSUES OF TRANSPORT LAYER:-

Transport layer is responsible for following issues:-

a) Accepting message segments from the application layer and to divide into packets.

b) End-to-End Delivery of the packet

c) Combining packets into message segment at receiver side.

d) Connection management.

In other words transport layer is responsible for two tasks:-

 Transport and regulate the flow of information from source to destination, reliably and accurately.

 The end-to-end control:

 Sliding windows.

 Sequencing numbers.

 Acknowledgments.

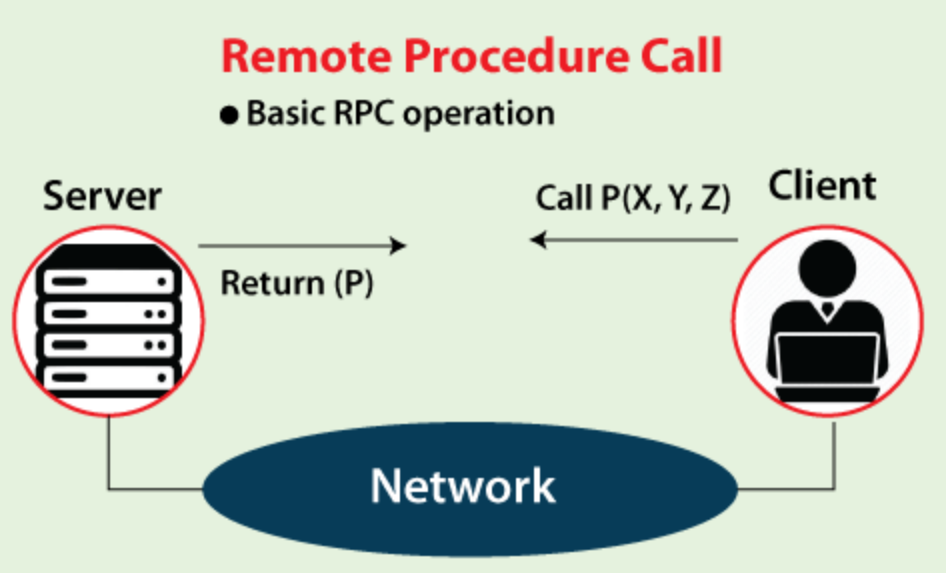
 Segmentation.

 Multiplexing.

****

VIII. What is RPC in Operating System?

Remote Procedure Call or RPC is a powerful technique for constructing distributed, client-server-based applications. It is also known as a function call or a subroutine call. A remote procedure call is when a computer program causes a procedure to execute in a different address space, coded as a local procedure call, without the programmer explicitly stating the details for the remote interaction. The programmer writes essentially the same code whether the subroutine is local to the executing program or remote. This is a form of client-server interaction implemented via a request-response message-passing system.

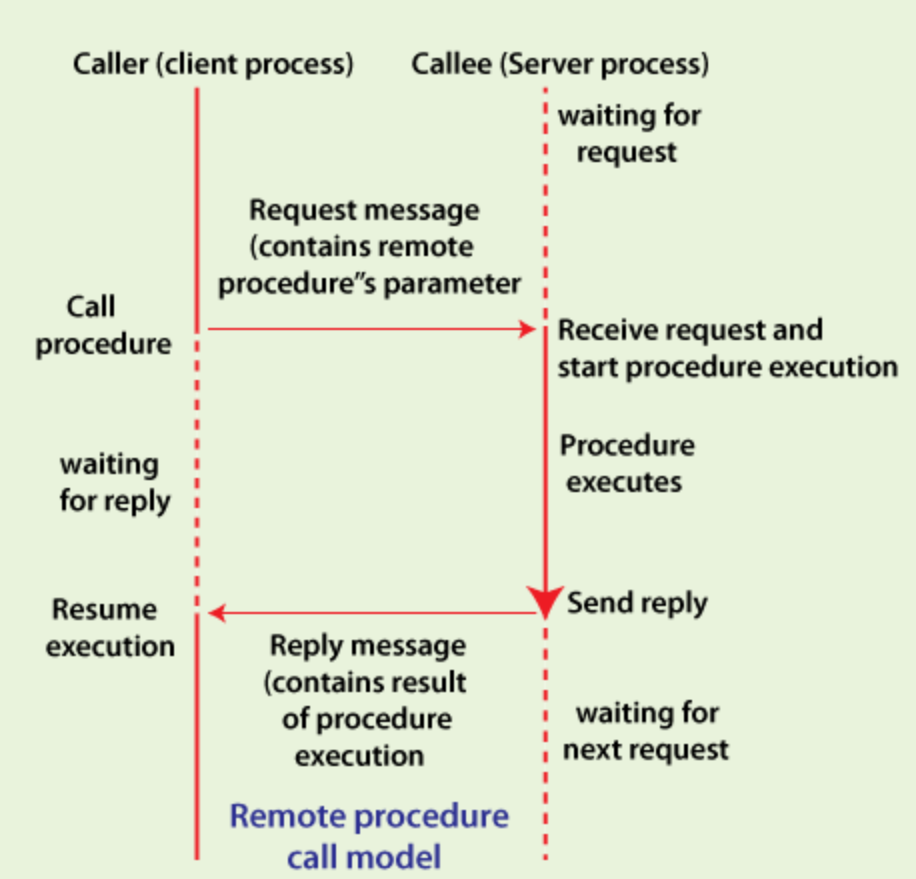


The RPC model implies ***location transparency*** that calling procedures are largely the same, whether local or remote. Usually, they are not identical, so that local calls can be distinguished from remote calls. Remote calls are usually orders of magnitude slower and less reliable than local calls, so distinguishing them is important.

RPCs are a form of inter-process communication (IPC), in that different processes have different address spaces. They have distinct virtual address spaces on the same host machine, even though the physical address space is the same. While if they are on different hosts, the physical address space is different.

How to Make a Remote Procedure Call

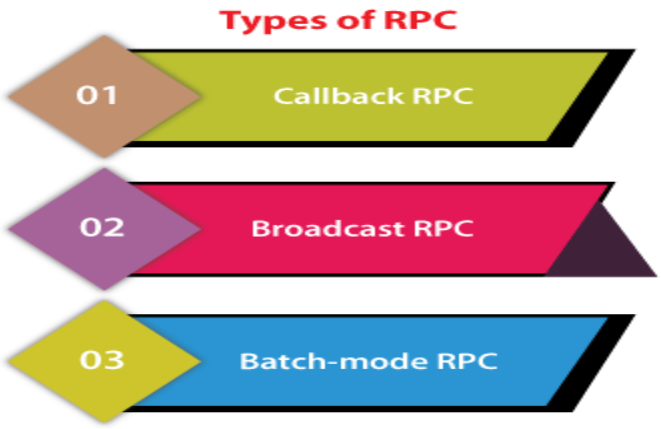
The calling environment is suspended, procedure parameters are transferred across the network to the environment where the procedure is to execute, and the procedure is executed there.



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

Types of RPC

There are three types of remote procedure call (RPC) in an operating system, such as:



1. Callback RPC

This type of RPC enables a P2P paradigm between participating processes. It helps a process to be both client and server services. Callback RPC has the following functions, such as:

- Remotely processed interactive application problems.
- Offers server with clients handle.
- Callback makes the client process wait.
- Manage callback deadlocks.
- It facilitates a peer-to-Peer paradigm among participating processes.

2. Broadcast RPC

Broadcast RPC is a client's request broadcast on the network, processed by all servers with the method for processing that request. Broadcast RPC has the following functions, such as:

- Allows you to specify that the client's request message has to be broadcast.
- You can declare broadcast ports.
- It helps to reduce the load on the physical network.

3. Batch-mode RPC

Batch-mode RPC helps to queue, separate RPC requests, in a transmission buffer, on the client-side, and then send them on a network in one batch to the server. Batch-mode RPC has the following functions, such as:

- It minimizes the overhead involved in sending a request by sending them over the network in one batch to the server.
- This type of RPC protocol is only efficient for an application that needs lower call rates.
- It needs a reliable transmission protocol.

What does RPC do?

When program statements using the RPC framework are compiled into an executable program, a stub is included in the compiled code representing the remote procedure code.

When the program is run and the procedure call is issued, the stub receives the request and forwards it to a client runtime program in the local computer. The first time the client stub is invoked, it contacts a name server to determine the transport address where the server resides.

The client runtime program knows how to address the remote computer and server application and sends the message across the network that requests the remote procedure. Similarly, the server

includes a runtime program and stub that interface with the remote procedure itself, and Response-request protocols are returned the same way.

Features of RPC

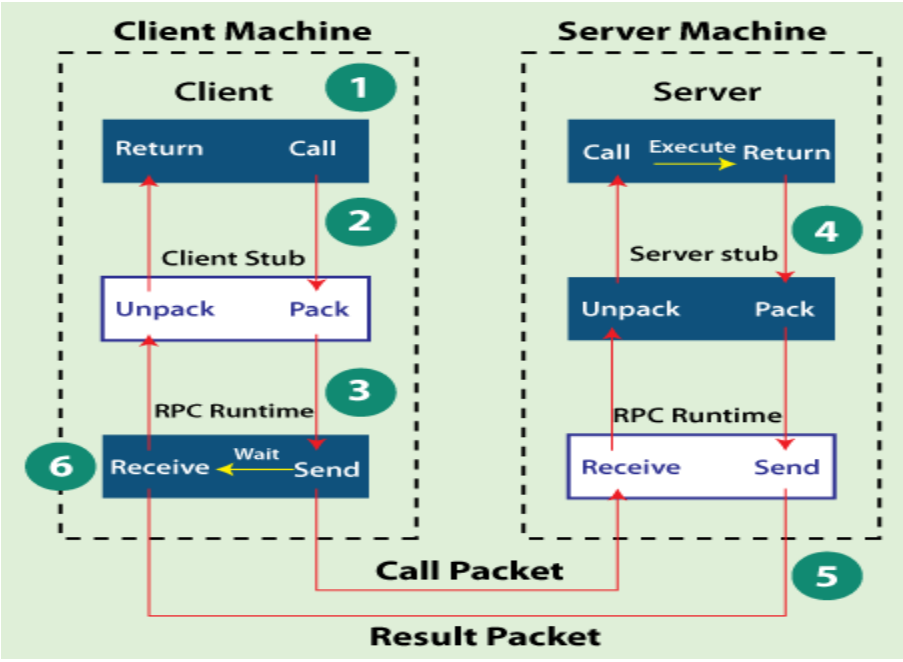
In an operating system, remote procedure call (RPC) has the following features, such as:

- RPC hides the complexity of the message passing process from the user.
- RPC only uses specific layers of the OSI model like the transport layer.
- Clients can communicate with the server by using higher-level languages.
- RPC works well with both local environments and remote environments.
- The program of RPC is written in simple code and is easily understood by the programmer.
- The operating system can handle processes and threads involved in RPC easily.
- The operating system hides the abstractions of RPC from the user.

How RPC works?

When a remote procedure call is invoked, the calling environment is suspended, the procedure parameters are transferred across the network to the environment where the procedure is to execute, and the procedure is then executed in that environment.

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



A remote procedure call (RPC) works in the following steps in an operating system:

Step 1: The client, client stub, and RPC run time execute on the client machine.

Step 2: A client starts a client stub process by passing parameters in the usual way. The packing of the procedure parameters is called ***marshalling***. The client stub stores within the client's own address space, and it also asks the local RPC Runtime to send back to the server stub.

Step 3: In this stage, the user can access RPC by making regular Local Procedural Call. RPC Runtime manages the transmission of messages between the network across client and server, and it also performs the job of retransmission, acknowledgment, routing, and encryption.

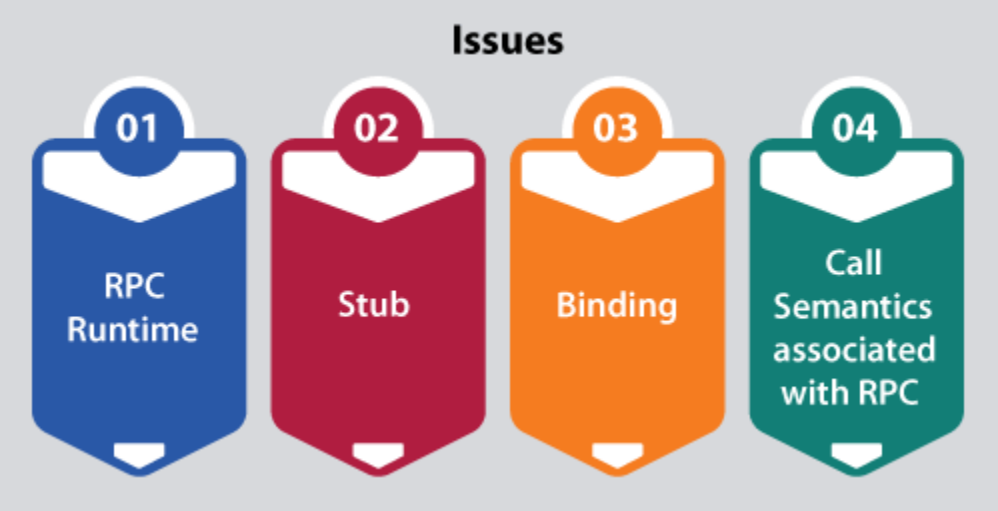
Step 4: After completing the server procedure, it returns to the server stub, which packs (marshalls) the return values into a message. The server stub then sends a message back to the transport layer.

Step 5: In this step, the transport layer sends back the result message to the client transport layer, which returns back a message to the client stub.

Step 6: In this stage, the client stub demarshalls (unpack) the return parameters in the resulting packet, and the execution process returns to the caller.

Issues of Remote Procedure Call (RPC)

In an operating system, Remote procedure call or RPC faced some issues that must be addressed, such as:



1. RPC Runtime

The RPC runtime system is a library of routines and services that handle the network communications that underlie the RPC mechanism. In the course of an RPC call, client-side and

server-side runtime 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 runtime system, marshaling and unmarshaling data, invoking the RPC runtime protocol, and if requested, carrying out some of the binding steps.
- **On the server-side:** The stub provides a similar interface between the runtime system and the local manager procedures 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. The binding consists of two parts:

- **Naming:** A Serverhaving a service to offer exports an interface for it. Exporting an interface registers it with the system so that clients can use it.
- **Locating:** A Clientmust import an (exported) interface before communication can begin.

4. The calling semantics associated with RPC

It is mainly classified into the 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.

Characteristics of RPC

Here are the essential characteristics of remote procedure call:

- The called procedure is in another process, which is likely to reside in another machine.

- The processes do not share address space.
- Parameters are passed only by values.
- RPC executes within the environment of the server process.
- It doesn't offer access to the calling procedure's environment.

Advantages of RPC

Here are some advantages or benefits of RPC, such as:

- RPC method helps clients to communicate with servers by the conventional use of procedure calls in high-level languages.
- The RPC method is modeled on the local procedure call, but the procedure is most likely to be executed in a different process and usually a different computer.
- RPC supports process and thread-oriented models.
- RPC makes the internal message passing mechanism hidden from the user.
- The effort needs to re-write and re-develop the code is minimum.
- Remote procedure calls can be used for distribution and the local environment.
- It commits many of the protocol layers to improve performance.
- RPC provides abstraction. For example, the message-passing nature of network communication remains hidden from the user.
- RPC allows the usage of the applications in a distributed environment that is not only in the local environment.
- With RPC code, re-writing and re-developing efforts are minimized.
- Process-oriented and thread-oriented models supported by RPC.

Disadvantages of RPC

Here are some disadvantages or drawbacks of using RPC, such as:

- Remote Procedure Call Passes Parameters by values only and pointer values, which is not allowed.
- Remote procedure calling (and return) time (i.e., overheads) can be significantly lower than a local procedure.

- This mechanism is highly vulnerable to failure as it involves a communication system, another machine, and another process.
- RPC concept can be implemented in different ways, which is can't standard.
- Not offer any flexibility in RPC for hardware architecture as It is mostly interaction-based.
- The cost of the process is increased because of a remote procedure call.

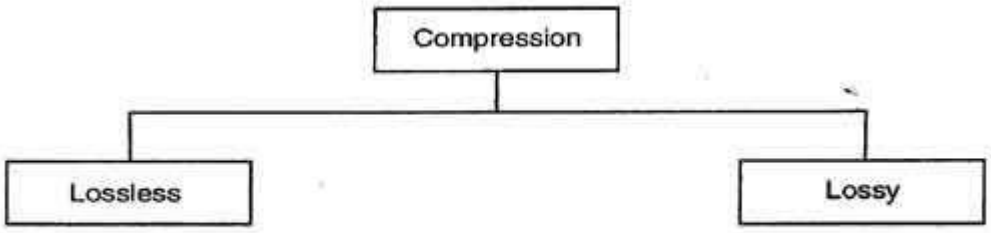
****

IX. Data compression

Data compression is the function of presentation layer in OSI reference model. Compression is often used to maximize the use of bandwidth across a network or to optimize disk space when saving data.

There are two general types of compression algorithms:

1. Lossless compression
2. Lossy compression



Types of Compression

Lossless Compression

Lossless compression compresses the data in such a way that when data is decompressed it is exactly the same as it was before compression *i.e.* there is no loss of data.

A lossless compression is used to compress file data such as executable code, text files, and numeric data, because programs that process such file data cannot tolerate mistakes in the data.

Lossless compression will typically not compress file as much as lossy compression techniques and may take more processing power to accomplish the compression.

Lossless Compression Algorithms

The various algorithms used to implement lossless data compression are :

1. Run length encoding

2. Differential pulse code modulation

3. Dictionary based encoding

1. Run length encoding

• This method replaces the consecutive occurrences of a given symbol with only one copy of the symbol along with a count of how many times that symbol occurs. Hence the names 'run length'.

• For example, the string AAABBCDDDD would be encoded as 3A2BIC4D.

• A real life example where run-length encoding is quite effective is the fax machine. Most faxes are white sheets with the occasional black text. So, a run-length encoding scheme can take each line and transmit a code for while then the number of pixels, then the code for black and the number of pixels and so on.

• This method of compression must be used carefully. If there is not a lot of repetition in the data then it is possible the run length encoding scheme would actually increase the size of a file.

2. Differential pulse code modulation

• In this method first a reference symbol is placed. Then for each symbol in the data, we place the difference between that symbol and the reference symbol used.

• For example, using symbol A as reference symbol, the string AAABBC DDDD would be encoded as AOOOl123333, since A is the same as reference symbol, B has a difference of 1 from the reference symbol and so on.

3. Dictionary based encoding

• One of the best known dictionary based encoding algorithms is Lempel-Ziv (LZ) compression algorithm.

• This method is also known as substitution coder.

• In this method, a dictionary (table) of variable length strings (common phrases) is built.

• This dictionary contains almost every string that is expected to occur in data.

• When any of these strings occur in the data, then they are replaced with the corresponding index to the dictionary.

• In this method, instead of working with individual characters in text data, we treat each word as a string and output the index in the dictionary for that word.

• For example, let us say that the word "compression" has the index 4978 in one particular dictionary; it is the 4978^{th} word is usr/share/dict/words. To compress a body of text, each time the string "compression" appears, it would be replaced by 4978.

Lossy Compression

Lossy compression is the one that does not promise that the data received is exactly the same as data send i.e. the data may be lost. This is because a lossy algorithm removes information that it cannot later restore. Lossy algorithms are used to compress still images, video and audio. Lossy

algorithms typically achieve much better compression ratios than the lossless algorithms. The Lossy compression method eliminates some amount of data that is not noticeable.

****

X. Cryptography in Computer Network

Cryptography refers to the science and art of transforming messages to make them secure and immune to attacks. It is a method of storing and transmitting data in a particular form so that only those for whom it is intended can read and process it. Cryptography not only protects data from theft or alteration but can also be used for user authentication.

Components

There are various components of cryptography which are as follows −

Plaintext and Ciphertext

The original message, before being transformed, is called plaintext. After the message is transformed, it is called ciphertext. An encryption algorithm transforms the plaintext into ciphertext; a decryption algorithm transforms the ciphertext back into plaintext. The sender uses an encryption algorithm, and the receiver uses a decryption algorithm.

Cipher

We refer to encryption and decryption algorithms as ciphers. The term cipher is also used to refer to different categories of algorithms in cryptography. This is not to say that every sender-receiver pair needs their very own unique cipher for secure communication. On the contrary, one cipher can serve millions of communicating pairs.

Key

A key is a number (or a set of numbers) that the cipher, as an algorithm, operates on. To encrypt a message, we need an encryption algorithm, an encryption key, and plaintext. These create the ciphertext. To decrypt a message, we need a decryption algorithm, a decryption key, and the ciphertext. These reveal the original plaintext.

Types

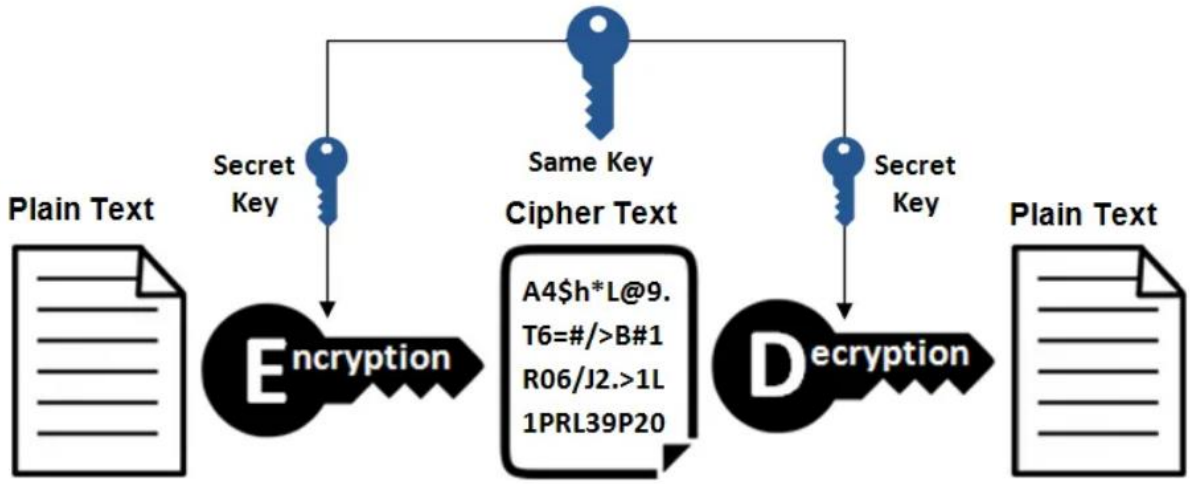
Types of cryptography

There are three types of cryptography:

- Symmetric key cryptography
- Asymmetric key cryptography
- Hash Function

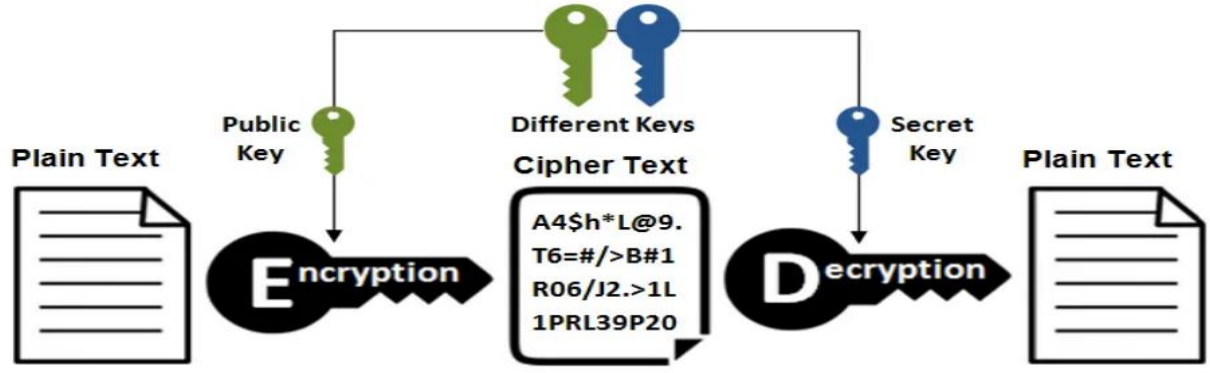
Symmetric key cryptography

Symmetric key cryptography is also known as secret-key cryptography, and in this type of cryptography, you can use only a single key. The sender and the receiver can use that single key to encrypt and decrypt a message. Because there is only one key for encryption and decryption, the symmetric key system has one major disadvantage: the two parties must exchange the key in a secure manner. An example of symmetric key cryptography is Blowfish.



Asymmetric key cryptography

Asymmetric key cryptography is also known as public-key cryptography, and it employs the use of two keys. This cryptography differs from and is more secure than symmetric key cryptography. In this system, each user encrypts and decrypts using two keys or a pair of keys (private key and public key). Each user keeps the private key secret and the public key is distributed across the network so that anyone can use those public keys to send a message to any other user. You can use any of those keys to encrypt the message and can use the remaining key for decryption. An RSA algorithm is an example of asymmetric key cryptography.



Hash Function

This algorithm makes no use of any keys. A hash value with a fixed length is calculated based on the plain text, making it impossible to recover the plain text's contents. Many operating systems encrypt passwords using hash functions.

Features Of Cryptography are as follows:

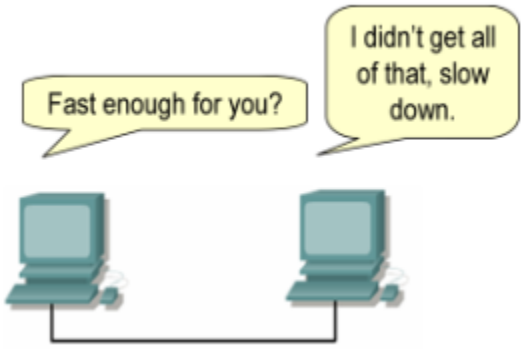
1. **Confidentiality:**
   Information can only be accessed by the person for whom it is intended and no other person except him can access it.
2. **Integrity:**
   Information cannot be modified in storage or transition between sender and intended receiver without any addition to information being detected.
3. **Non-repudiation:**
   The creator/sender of information cannot deny his intention to send information at later stage.
4. **Authentication:**
   The identities of sender and receiver are confirmed. As well as destination/origin of information is confirmed.

****

XI. Window management in TCP

Windowing and Window Size: Window management in TCP is an important concept that ensures reliability in packet delivery as well as reduce the wastage of time in waiting for the acknowledge after each packet.
Window size: window size determines the amount of data that you can transmit before receiving an acknowledgment. Sliding window refers to the fact that the window size is negotiated dynamically during the TCP session.

1. Expectational acknowledgment means that the acknowledgment number refers to the octet that is next expected
2. If the source receives no acknowledgment, it knows to retransmit at a slower rate.



The mechanism of the sliding window style may be understood easily with the help of below given
diagrams:

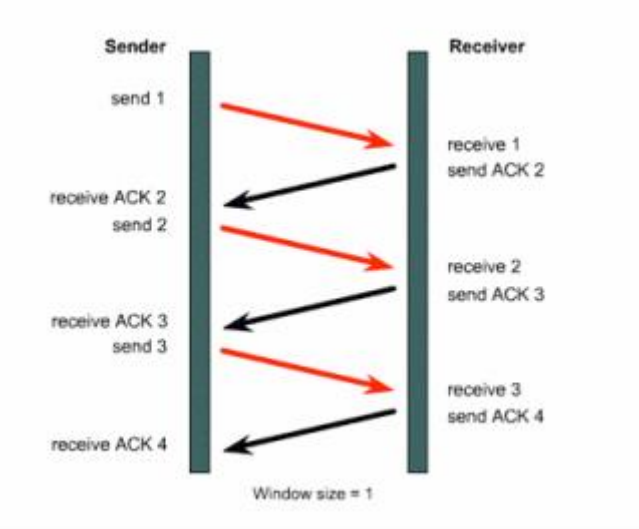


Figure one

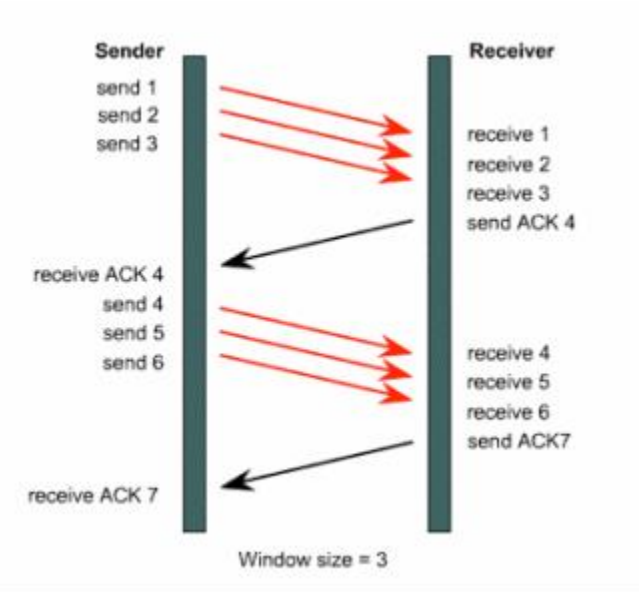
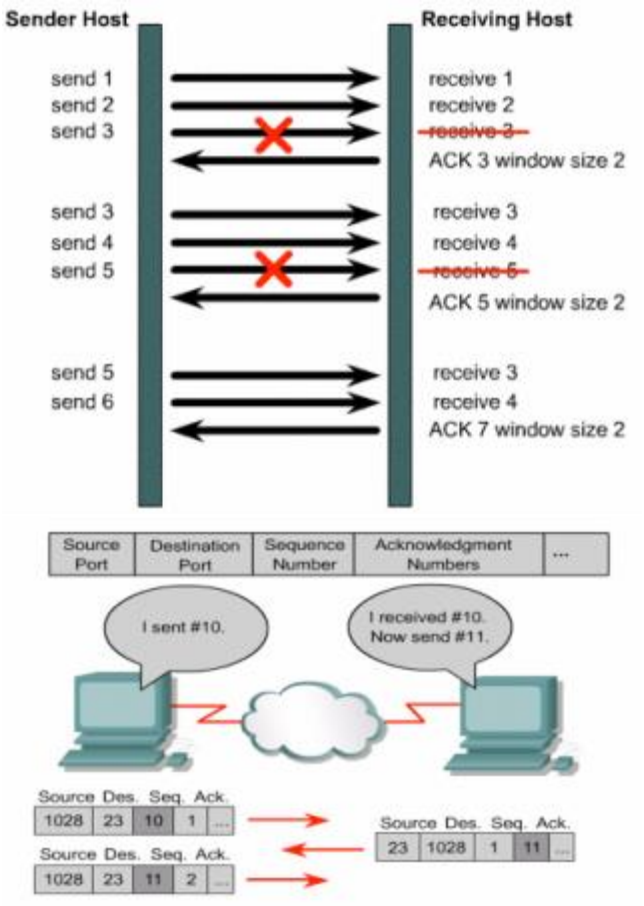


figure two



TCP keeps track the amount of data allowed to be sent at a given point; this is dictated by the host receiving the data, the host sending the data and the conditions of the network.

There are three TCP windows used in a TCP connection:

- ✓ Receive Window (RWIN)
- ✓ Send Window (SWIN)
- ✓ Congestion Window (CWIN)

Each window serves an important purpose for the flow of data between the TCP sender and TCP receiver. Here they are referred to as sender and receiver, in practice they are usually referred to as client and server. The reason for this is that data can flow in either direction client to server or server to client.

The RWIN dictates how much data a TCP receiver is willing to accept before sending an acknowledgement (ACK) back to the TCP sender. The receiver will advertise its RWIN to the sender and thus the sender knows it can't send more than an RWINs worth of data before receiving and ACK from receiver.

The SWIN dictates how much data a TCP sender will be allowed send before it must receive an ACK from the TCP receiver.

The CWIN is a variable that changes dynamically according to the conditions of the network. If data is lost or delivered out-of-order the CWIN is typically reduced.

The RWIN and SWIN are configurable values on the host, while the CWIN is dynamic and can't be configured.

The TCP window (RWIN/SWIN) is analogous to the number of seats on an airplane; the number of people filling the seats is analogous to bytes. Let's assume an airplane has 200 seats and its flying from New York to San Francisco which takes 6 hours or 12 hours there and back; 12 hours being our RTT (Round-Trip Time). It's a mid afternoon flight so only 50 seats were filled. If we assume the plane is only allowed to transport people in one direction and it must return to New York before transporting more people we can get our hourly average of people per hour.

********** All the Best**********