IP Protocols

Definition of IPv4

An IPv4 address is a 32- bit binary value, which can be displayed as four decimal digits. The IPv4 address space offers about 4.3 billion addresses. Only 3.7 billion addresses can only be assigned out of 4.3 billion address. The other addresses are conserved for specific purposes such as multicasting, private address space, loopback testing, and research.
IP version 4 (IPv4) uses Broadcasting for transferring packets from one computer to all computers; this probably generates problems sometimes.

Dotted-Decimal Notation of IPv4
128.11.3.31

Definition of IPv6

An IPv6 address is a 128-bit binary value, which can be displayed as 32 hexadecimal digits. Colons isolate entries in a sequence of 16-bit Hexadecimal fields.

It provides 3.4×10^{38} IP addresses. This version of IP addressing is designed to fulfill the needs of exhausting IP's and providing sufficient addresses for future Internet growth requirements.

As IPv4 uses two-level address structure where the use of address space is insufficient. That was the reason for proposing the IPv6, to overcome the deficiencies IPv4. The format and the length of the IP addresses were changed along with the packet format and protocols were also modified.

Hexadecimal Colon Notation of IPv6
FDEC:BA98:7654:3210:ADBF:BBFF:2922:FFFF

IP provides three major things that are:

I. Specification of the exact format of all data.
II. It performs routing function and chooses path for sending the data.
III. It involves a collection of rules that support the idea of unreliable packet delivery.

Comparison Chart

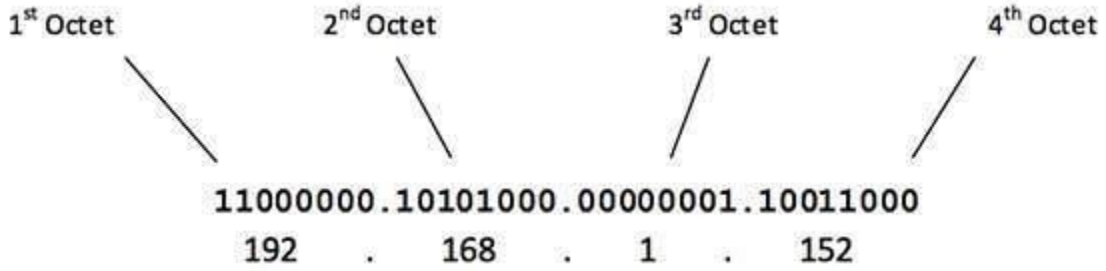
Basis of comparison	**IPv4**	**IPv6**
Address Configuration	Supports Manual and DHCP configuration.	Supports Auto-configuration and renumbering
End-to-end connection integrity	Unachievable	Achievable
Address Space	It can generate 4.29 x 10^9 addresses.	It can produce quite a large number of addresses, i.e., 3.4 x 10^{38}.
Security features	Security is dependent on application	IPSEC is inbuilt in the IPv6 protocol
Address length	32 bits (4 bytes)	128 bits (16 bytes)
Address Representation	In decimal	In hexadecimal
Fragmentation performed by	Sender and forwarding routers	Only by the sender
Packet flow identification	Not available	Available and uses flow label field in the header
Checksum Field	Available	Not available
Message Transmission Scheme	Broadcasting	Multicasting and Any casting
Encryption and Authentication	Not Provided	Provided

CLASSES IPV4

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.

1. 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

2. 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 – **10**111111
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

3. Class C Address

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

11000000 – **110**11111
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

4. 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 – **1110**1111
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.

5. 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.

IP addresses: Networks and hosts

An IP address is a 32-bit number that uniquely identifies a host (computer or other device, such as a printer or router) on a TCP/IP network.

IP addresses are normally expressed in dotted-decimal format, with four numbers separated by periods, such as 192.168.123.132. To understand how subnet masks are used to distinguish between hosts, networks, and sub networks, examine an IP address in binary notation.

For example, the dotted-decimal IP address 192.168.123.132 is (in binary notation) the 32 bit number 11000000010100011110111000100. This number may be hard to make sense of, so divide it into four parts of eight binary digits.

These eight bit sections are known as octets. The example IP address, then, becomes 11000000.10101000.01111011.10000100. This number only makes a little more sense, so for most uses, convert the binary address into dotted-decimal format (192.168.123.132). The decimal numbers separated by periods are the octets converted from binary to decimal notation.

For a TCP/IP wide area network (WAN) to work efficiently as a collection of networks, the routers that pass packets of data between networks do not know the exact location of a host for which a packet of information is destined. Routers only know what network the host is a member of and use information stored in their route table to determine how to get the packet to the destination host's network. After the packet is delivered to the destination's network, the packet is delivered to the appropriate host.

For this process to work, an IP address has two parts. The first part of an IP address is used as a network address, the last part as a host address. If you take the example 192.168.123.132 and divide it into these two parts you get the following:

192.168.123. Network .132 Host

-or-

192.168.123.0 - network address. 0.0.0.132 - host address.

Subnet mask

The second item, which is required for TCP/IP to work, is the subnet mask. The subnet mask is used by the TCP/IP protocol to determine whether a host is on the local subnet or on a remote network.

In TCP/IP, the parts of the IP address that are used as the network and host addresses are not fixed, so the network and host addresses above cannot be determined unless you have more information. This information is supplied in another 32-bit number called a subnet mask. In this example, the subnet mask is 255.255.255.0. It is not obvious what this number means unless you know that 255 in binary notation equals 11111111; so, the subnet mask is:

11111111.11111111.11111111.0000000

Lining up the IP address and the subnet mask together, the network and host portions of the address can be separated:

11000000.10101000.01111011.10000100 -- IP address (192.168.123.132)
11111111.11111111.11111111.00000000 -- Subnet mask (255.255.255.0)

The first 24 bits (the number of ones in the subnet mask) are identified as the network address, with the last 8 bits (the number of remaining zeros in the subnet mask) identified as the host address. This gives you the following:

11000000.10101000.01111011.00000000 -- Network address (192.168.123.0)

00000000.00000000.00000000.10000100 -- Host address (000.000.000.132)

So now you know, for this example using a 255.255.255.0 subnet mask, that the network ID is 192.168.123.0, and the host address is 0.0.0.132. When a packet arrives on the 192.168.123.0 subnet (from the local subnet or a remote network), and it has a destination address of 192.168.123.132, your computer will receive it from the network and process it.

Almost all decimal subnet masks convert to binary numbers that are all ones on the left and all zeros on the right. Some other common subnet masks are:

Decimal	**Binary**
255.255.255.192	1111111.11111111.1111111.11000000
255.255.255.224	1111111.11111111.1111111.11100000

Subnetting

A Class A, B, or C TCP/IP network can be further divided, or subnetted, by a system administrator. This becomes necessary as you reconcile the logical address scheme of the Internet (the abstract world of IP addresses and subnets) with the physical networks in use by the real world.

A system administrator who is allocated a block of IP addresses may be administering networks that are not organized in a way that easily fits these addresses. For example, you have a wide area network with 150 hosts on three networks (in different cities) that are connected by a TCP/IP router. Each of these three networks has 50 hosts. You are allocated the class C network 192.168.123.0. (For illustration, this address is actually from a range that is not allocated on the Internet.) This means that you can use the addresses 192.168.123.1 to 192.168.123.254 for your 150 hosts.

Two addresses that cannot be used in your example are 192.168.123.0 and 192.168.123.255 because binary addresses with a host portion of all ones and all zeros are invalid. The zero address is invalid because it is used to specify a network without specifying a host. The 255 address (in binary notation, a host address of all ones) is used to broadcast a message to every host on a network. Just remember that the first and last address in any network or subnet cannot be assigned to any individual host.

You should now be able to give IP addresses to 254 hosts. This works fine if all 150 computers are on a single network. However, your 150 computers are on three separate physical networks. Instead of requesting more address blocks for each network, you divide your network into subnets that enable you to use one block of addresses on multiple physical networks.

In this case, you divide your network into four subnets by using a subnet mask that makes the network address larger and the possible range of host addresses smaller. In other words, you are 'borrowing' some of the bits usually used for the host address, and using them for the network portion of the address. The subnet mask 255.255.255.192 gives you four networks of 62 hosts each. This works because in binary notation, 255.255.255.192 is the same as 1111111.11111111.1111111.11000000. The first two digits of the last octet become network addresses, so you get the additional networks 00000000 (0), 01000000 (64), 10000000 (128) and 11000000 (192). (Some administrators will only use two of the subnetworks using 255.255.255.192 as a subnet mask. For more information on this topic, see RFC 1878.) In these four networks, the last 6 binary digits can be used for host addresses.

Using a subnet mask of 255.255.255.192, your 192.168.123.0 network then becomes the four networks 192.168.123.0, 192.168.123.64, 192.168.123.128 and 192.168.123.192. These four networks would have as valid host addresses:

192.168.123.1-62
192.168.123.65-126
192.168.123.129-190
192.168.123.193-254

Remember, again, that binary host addresses with all ones or all zeros are invalid, so you cannot use addresses with the last octet of 0, 63, 64, 127, 128, 191, 192, or 255.

You can see how this works by looking at two host addresses, 192.168.123.71 and 192.168.123.133. If you used the default Class C subnet mask of 255.255.255.0, both addresses are on the 192.168.123.0 network. However, if you use the subnet mask of 255.255.255.192, they are on different networks; 192.168.123.71 is on the 192.168.123.64 network, 192.168.123.133 is on the 192.168.123.128 network.

DNS

The DNS provides mapping between human-readable names (like www.amazon.com) and their associated IP addresses (like 205.251.242.103). DNS can be best **compared to a phone book** where you look up the phone numbers listed by easier-to-remember names. DNS comes under the application layer protocol.

A user types *www.amazon.com* in his browser, which then queries the DNS server for amazon.com's IP addresses. The servers return Amazon's address so the browser can request data from Amazon's web host, which returns the elements necessary to build their home page in the local browser.

How DNS Works: Domain Name System Terminology

Domain Names

A domain name is a **human-readable name**—like amazon.com—that we type in a web browser URL field. The Internet Corporation for Assigned Names and Numbers manages these domain names

Top Level Domain (TLD)

TLD refers to the last part of a domain name. For example, the **.com** in amazon**.com** is the Top Level Domain. The most common TLDs include .com, .net, org, and .info. Country code TLDs represents specific geographic locations. For example: .in represents India. Here are some more examples:

- **com** – Commercial businesses.

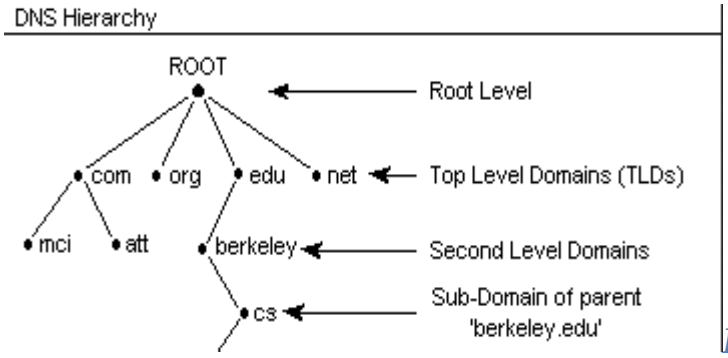
- **gov** – U.S. government agencies.
- **edu** – Educational institutions such as universities.
- **org** – Organizations (mostly non-profit).
- **mil** – Military.
- **net** – Network organizations.
- **eu** – European Union.

Second Level Domain

This is the part of a domain name which comes **right before** the TLD—**amazon**.com—for example.

Sub Domain

A subdomain can be created to **identify unique content areas** of a web site. For example, the aws of **aws**.amazon.com.



Domain Name Registrar

By managing domain name reservations, name registrars are critical to how DNS works. ICANN currently grants permission to organizations to act as domain name registrars for **specific higher level domains**.

Name Server

Like a phone book, the name server is a **collection of domain names** matched to IP addresses.

How DNS Works: Domain Name System record types

A Record

Address record. A Records map server IP addresses to domain names. For example, 72.21.206.6 to amazon.com.

CNAME

Canonical Name record. A CNAME record establishes one domain as an alias to another (thereby routing all traffic addressed to the alias to the target; the canonical address).

Alias Record

Like a CNAME record, Alias records can be used to map one address to another. But Aliases can coexist with other records using the same name.

MX Record

Mail Exchange Record. These records will redirect a domain's email to the servers hosting the domain's user accounts. Mail exchange records are used for determining the priority of email servers for a domain.

How DNS Works

When a user types a human-readable address into the browser, the operating system's DNS client will check for information in a local cache. If the requested address isn't there, it will look for a Domain Name System server in the local area network (LAN). When the local DNS server receives the query, and the requested domain name is found, it will return the result.

If the name is not found, the local server will forward the query to a DNS cache server, often provided by the Internet Service Provider (ISP). Since the DNS server's cache contains a temporary store of DNS records, it will quickly respond to requests. These DNS cache servers are called ***not authoritative DNS servers*** as they provide request resolution based in a cached value acquired from *authoritative DNS servers*.

An **Authoritative Root Name Server** maintains and provides a list of authoritative name servers for each of the top-level domains (.com, .org, etc.).

An **Authoritative Top Level Domain Name Server** maintains and provides a list of authoritative **name servers** for all domains (gmail.com, wikipedia.org, etc.). Its job is to query name servers to find and return the authoritative name server for the requested domain.

Email

The term “email” stands for “electronic mail”. The electronic mail is introduced first in the 1960s, however it became available in the current structure in the 1970s. Let us take a look at how email actually works.

Protocols used in email systems

The email communication is done via three protocols in general. They are listed below.

- IMAP
- POP
- SMTP

IMAP

The IMAP stands for Internet Mail Access Protocol. This protocol is used while receiving an email. When one uses IMAP, the emails will be present in the server and not get downloaded to the user’s mail box and deleted from the server. This helps to have less memory used in the local computer and server memory is increased.

POP

The POP stands for Post Office Protocol. This protocol is also used for incoming emails. The main difference with the both protocols is that POP downloads the entire email into the local computer and deletes the data on the server once it is downloaded. This is helpful in a server with less free memory. Current version of POP is POP3.

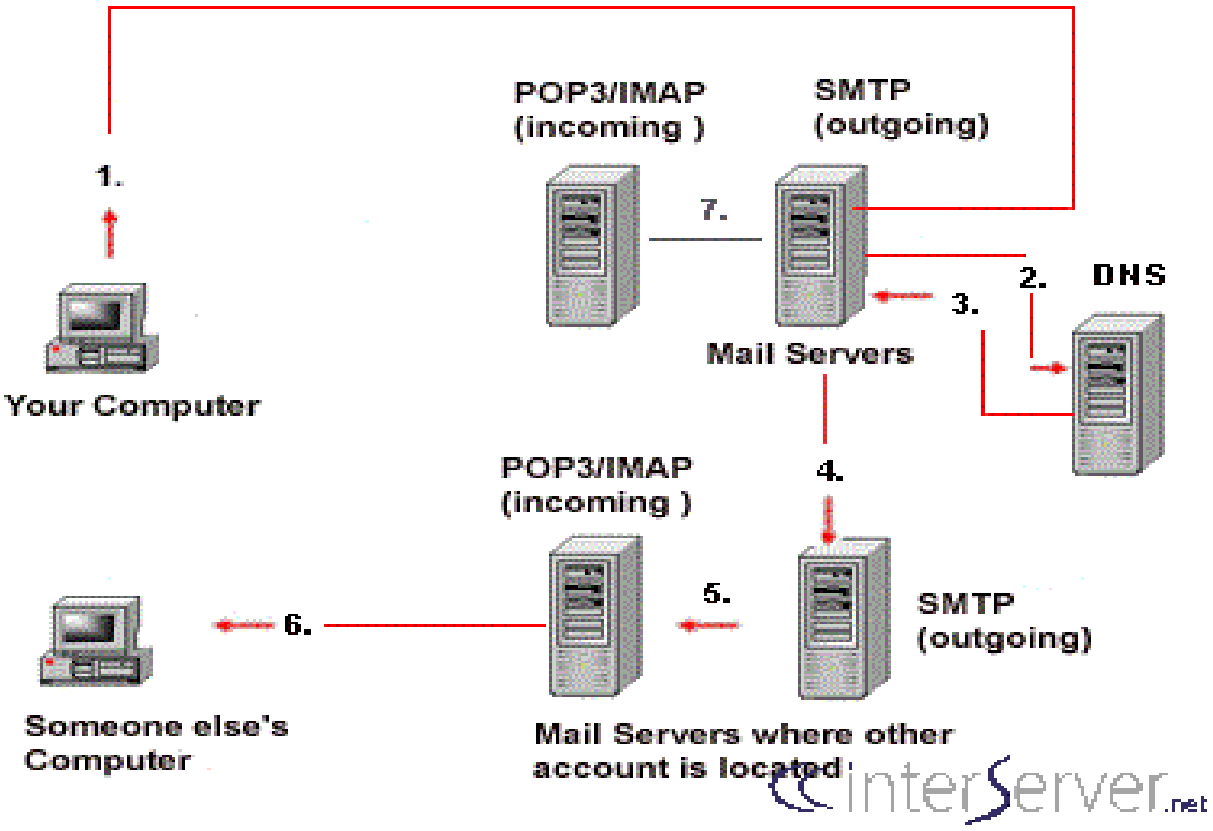
SMTP

The SMTP stands for Simple Mail Transfer Protocol. Email is sent using this protocol.

How does email work?

The diagram down below describes the path that email takes from your computer to the intended recipient . This shows the path of the email from sending to

receiving ends. There are also many logical machines in the email delivery process. Please have a look at the diagram before proceeding.



FTP (File Transfer Protocol)

File Transfer Protocol (FTP) is a standard Internet protocol for transmitting files between computers on the Internet over TCP/IP connections. FTP is a client-server protocol where a client will ask for a file, and a local or remote server will provide it.

The end-users machine is typically called the local host machine, which is connected via the internet to the remote host—which is the second machine running the FTP software.

Anonymous FTP is a type of FTP that allows users to access files and other data without needing an ID or password. Some websites will allow visitors to use a guest ID or password- anonymous FTP allows this.

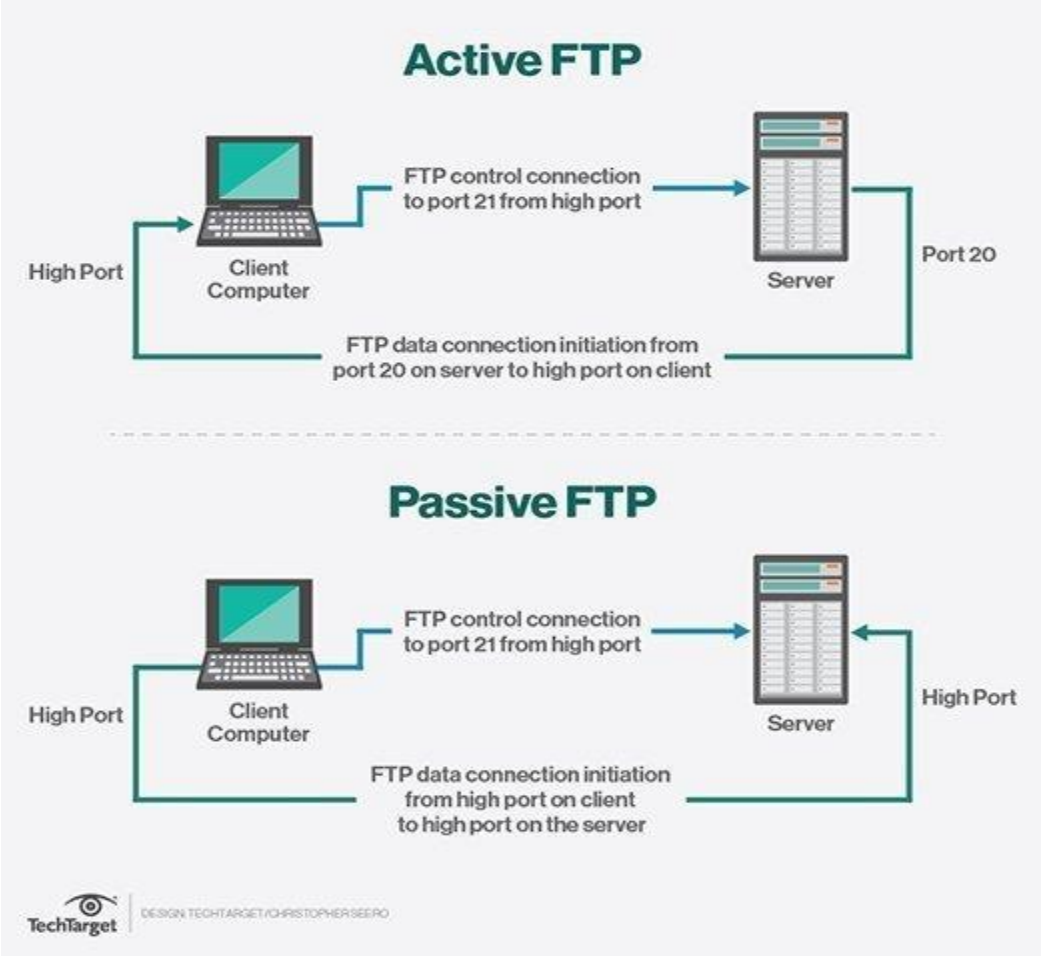
Although a lot of file transfer is now handled using HTTP, FTP is still commonly used to transfer files "behind the scenes" for other applications -- e.g., hidden behind the user interfaces of banking, a service that helps build a website, such as

Wix or SquareSpace, or other services. It is also used, via Web browsers, to download new applications.

How FTP works

FTP is a client-server protocol that relies on two communications channels between client and server: a command channel for controlling the conversation and a data channel for transmitting file content. Clients initiate conversations with servers by requesting to download a file. Using FTP, a client can upload, download, delete, rename, move and copy files on a server. A user typically needs to log on to the FTP server, although some servers make some or all of their content available without login, known as anonymous FTP.

FTP sessions work in passive or active modes. In active mode, after a client initiates a session via a command channel request, the server initiates a data connection back to the client and begins transferring data. In passive mode, the server instead uses the command channel to send the client the information it needs to open a data channel. Because passive mode has the client initiating all connections, it works well across firewalls and Network Address Translation (NAT) gateways.



Active FTP and passive FTP compared

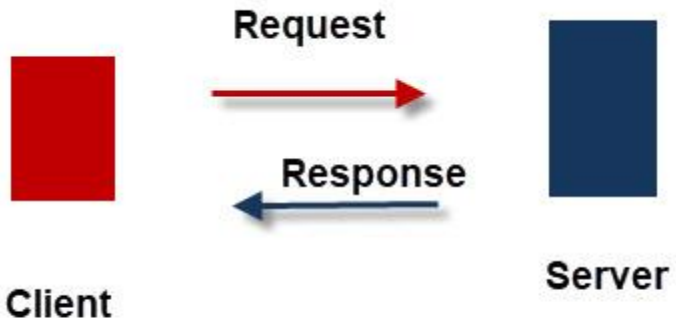
HTTP (HyperText Transfer Protocol)

HTTP is a protocol which allows the fetching of resources, such as HTML documents. It is the foundation of any data exchange on the Web and it is a client-server protocol, which means requests are initiated by the recipient, usually the Web browser. A complete document is reconstructed from the different sub-documents fetched, for instance text, layout description, images, videos, scripts, and more.

Clients and servers communicate by exchanging individual messages (as opposed to a stream of data). The messages sent by the client, usually a Web browser, are called *requests* and the messages sent by the server as an answer are called *responses*.

How It Works

Like most of the Internet protocols **http** it is a **command** and **response text based** protocol using a **client server** communications model.



HTTP Protocol Basics

The client makes a request and the server responds.

The **HTTP protocol** is also a **stateless protocol** meaning that the server isn't required to store session information, and each request is independent of the other.

- All requests originate at the client (your browser)
- The server responds to a request.
- The requests (commands) and responses are in readable text.
- The requests are independent of each other and the server **doesn't need to track** the requests.

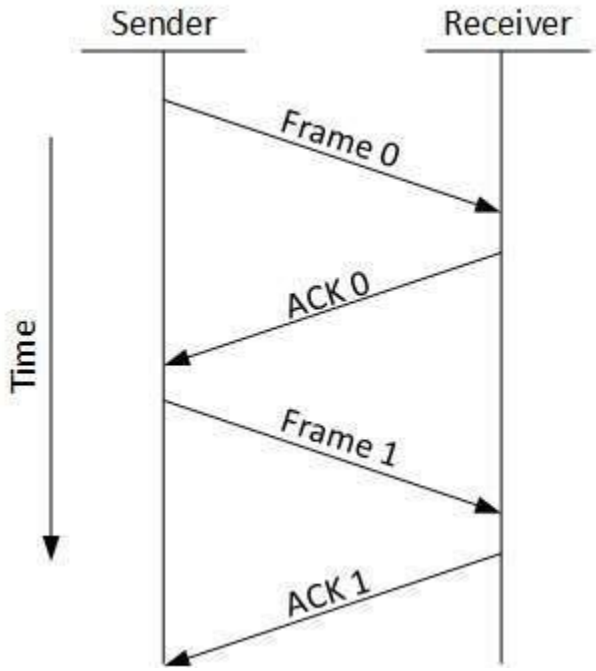
Flow Control

When a data frame (Layer-2 data) is sent from one host to another over a single medium, it is required that the sender and receiver should work at the same speed. That is, sender sends at a speed on which the receiver can process and accept the data. What if the speed (hardware/software) of the sender or receiver differs? If sender is sending too fast the receiver may be overloaded, (swamped) and data may be lost.

Two types of mechanisms can be deployed to control the flow:

- **Stop and Wait**

  This flow control mechanism forces the sender after transmitting a data frame to stop and wait until the acknowledgement of the data-frame sent is received.



- **Sliding Window**

  In this flow control mechanism, both sender and receiver agree on the number of data-frames after which the acknowledgement should be sent. As we learnt, stop and wait flow control mechanism wastes resources, this protocol tries to make use of underlying resources as much as possible.

Error Control

When data-frame is transmitted, there is a probability that data-frame may be lost in the transit or it is received corrupted. In both cases, the receiver does not receive the correct data-frame and sender does not know anything about any loss.In such case, both sender and receiver are equipped with some protocols which helps them to detect transit errors such as loss of data-frame. Hence, either the sender retransmits the data-frame or the receiver may request to resend the previous data-frame.

Requirements for error control mechanism:

- **Error detection** - The sender and receiver, either both or any, must ascertain that there is some error in the transit.
- **Positive ACK** - When the receiver receives a correct frame, it should acknowledge it.
- **Negative ACK** - When the receiver receives a damaged frame or a duplicate frame, it sends a NACK back to the sender and the sender must retransmit the correct frame.
- **Retransmission:** The sender maintains a clock and sets a timeout period. If an acknowledgement of a data-frame previously transmitted does not arrive before the timeout the sender retransmits the frame, thinking that the frame or it's acknowledgement is lost in transit.

Describe basic routing concepts

Routing is a path to finding from one end to the other and routing occurs at layer 3 and bridging occurs at layer 2. Routing Process to forward packets to destination networks.

Routers don't really care about hosts they care only about networks and the best path to each network. The logical network address of the destination host is used to get packets to a network through a routed network, and then the hardware address of the host is used to deliver the packet from a router to the correct destination host.

Routing table is used to find best path to destination. Forwarding decisions based on Layer 3. IP performs search for a matching host address, search for a matching network address, and search for a default entry, Routing done by IP router, when it searches the routing table and decides which interface to end a packet out.

When a router receives a packet, it examines the destination IP address. If the destination IP address does not belong to any of the router's directly connected networks, the router must forward this packet to another router.

IP Routing

IP routing—the process of forwarding IP packets—delivers packets across entire TCP/IP networks, from the device that originally builds the IP packet to the device that is supposed to receive the packet. In other words, IP routing delivers IP packets from the sending host to the destination host.

Routers don't really care about hosts they care only about networks and the best path to each network. The logical network address of the destination host is used to get packets to a network through a routed network, and then the hardware address of the host is used to deliver the packet from a router to the correct destination host.

Routing table is used to find best path to destination. Forwarding decisions based on Layer 3. IP performs search for a matching host address, search for a matching network address, and search for a default entry, Routing done by IP router, when it searches the routing table and decides which interface to end a packet out.

When a router receives a packet, it examines the destination IP address. If the destination IP address does not belong to any of the router's directly connected networks, the router must forward this packet to another router.

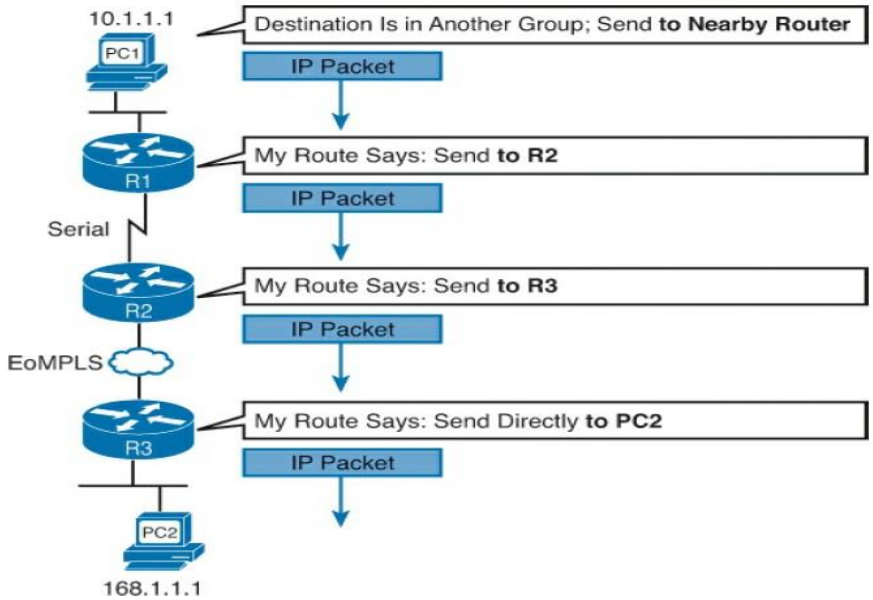


Fig shows Basic IP Routing.

There are two types of Routing:

1. Static Routing.

2. Dynamic Routing.

Static Routing:

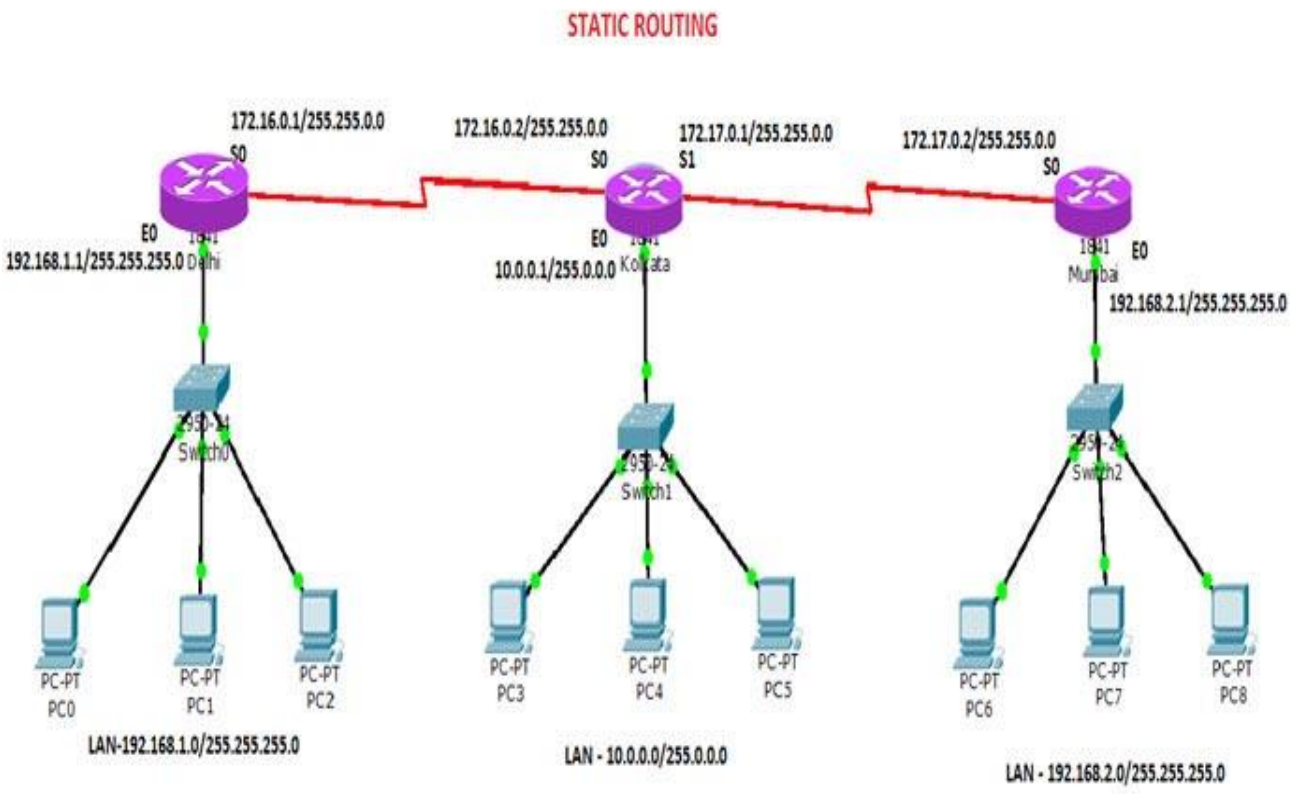
Static Routing is typically used in hosts Enter subnet mask, router (gateway), IP address Perfect for cases with few connections, doesn’t change much. Specifies network address and subnet mask of remote network, and IP address of next hop router or exit interface.

Static Routing is Easy to configure and Easier for administrator to understand.

A static route includes the network address and subnet mask of the remote network, along with the IP address of the next-hop router or exit interface. Static routes should be used in a large network is configured in a hub-and-spoke topology, a network is connected to the Internet only through a single ISP, a network consists of only a few routers.

The static routing method requires someone to hand-type all network locations into the routing table. The network administrator manually enters the routing information in the router. The static routing is used to test a particular link in a network.

Drawback for Static Routing is a network consists of only a few routers and requires complete knowledge of the whole network for proper implementation and does not scale well with growing networks.



Dynamic Routing:

Most routers use dynamic routing automatically builds the routing tables, there are two major approaches Link State Algorithms and Distance Vector Algorithms. Dynamic Routing is used by routers to share information about the reachability and status of remote network. In dynamic routing, a routing protocol on one router communicates with the same routing protocol running on neighbor routers. The routers then update each other about all the networks they know about and place this information into the routing table.

Dynamic Routing has less administrative overhead when adding or deleting a network Protocols automatically react to the topology changes and it is more scalable. Dynamic routing protocols are used by routers to share information about the reachability and status of remote networks.

The dynamic routing is using routing protocol to update routing information. The dynamic routing used as a protocol on one router communicates with the same protocol running on neighboring routers.

Drawback for Dynamic Routing is more administrator knowledge is required for configuration, verification and troubleshooting and Router resources are used (CPU cycles, memory and link bandwidth).