I.P. PROTOCOL AND NETWORK APPLICATIONS

Authors

Mayur Patel **Jitendra Shishangiya**

I.P. PROTOCOL AND NETWORK APPLICATIONS

Introduction to Network Layer Logical Addressing

- Communication at the network layer is host-to-host (computer-to-computer).
- A computer somewhere in the world needs to communicate with another computer somewhere else in the world.
- For this communication, we need a global addressing scheme, called "logical addressing"
- Today, IP addresses are used to provide logical addresses in the network layer of the TCP/IP protocol suite.
  IPV4
- The Internet addresses are 32 bits in length; this gives us a maximum of 2^{32} addresses.
- These addresses are referred to as IPv4 (IP version 4) addresses or simply IP addresses.

- The need for more addresses, in addition to other concerns about the IP layer, motivated a new design of the IP layer called the new generation of IP or IPv6 (IP version 6).

- In this version, the Internet addresses are 128 bits in length; this gives us a maximum of 2^{128} addresses.
- 128-bit addresses give much greater flexibility in address allocation. These addresses are referred to as IPv6 (IP version 6) addresses.
- In this chapter, we first discuss IPv4 addresses, which are currently being used in the Internet. We then discuss the IPv6 addresses, which may become dominant in the future.

1 IP Protocol – IP v4, IP v6.

IP Protocol

IP protocol works at network layer, It is unreliable and connectionless protocol. It doesn't facilitate for error checking. It has neither error control nor flow control. IP uses only error detection mechanism and discards the corrupted packets.

IP does its best service for to packet transmission, but it doesn't guarantee for it. For data transmission, reliability is important; So IP must be paired with a reliable protocol TCP.

IP is a connectionless protocol. This means each datagram is handled independently. Each datagram can follow different routes. IP doesn't keep track of the routes and has no facility for recording datagram, because it is connectionless. So IP uses higher level (TCP) protocol to take care of all the problems. There are two versions of IP protocol. 1) IPv4 and 2) IPv6

1) IPV4

It stands for Internet Protocol version 4. An IPv4 address is a 32-bit address that *uniquely* and *universally* defines the connection of a device (for example, a computer or a router) to the Internet.

IPv4 addresses are unique and universal. They are unique in the sense that each address defines one, and only one, connection to the Internet. Two devices on the Internet can never have the same address at the same time.

Address Space

IPv4 protocol defines several addresses has an address space. "An address space is the total number of addresses used by the protocol". If a protocol uses *N* bits to define an address, the address space is 2^N because each bit can have two different values (0 or 1) *N* bits can have 2^N values. IPv4 uses 32-bit addresses, which means that the address space is 2^{32} or 4,294,967,296 (more than 4 billion).

Dotted-Decimal Notation

To make the IPv4 address more compact and easier to read, Internet addresses are usually written in decimal form with a decimal point (dot) separating the bytes. Now, as we know IPv4 addresses are 32 bits long, these 32 bits can be divided into 4 parts, each with 8 bits.

So, for 32 bits available range for the addresses is as follows:

- In binary notation, the range is from

00000000 00000000 00000000 00000000 to 11111111 11111111 11111111 11111111

- In decimal notation, the range is from

0.0.0.0 to 255.255.255.255

Ex. For an IP address in dotted decimal notation : 192.168.1.1. This IP address in binary notation can be written as 11000000 10101000 00000001 00000001

Figure 1 shows an IPv4 address in both binary and dotted-decimal notation. Note that because each byte is 8 bits, each number in dotted-decimal notation is a value ranging from 0 to 255.

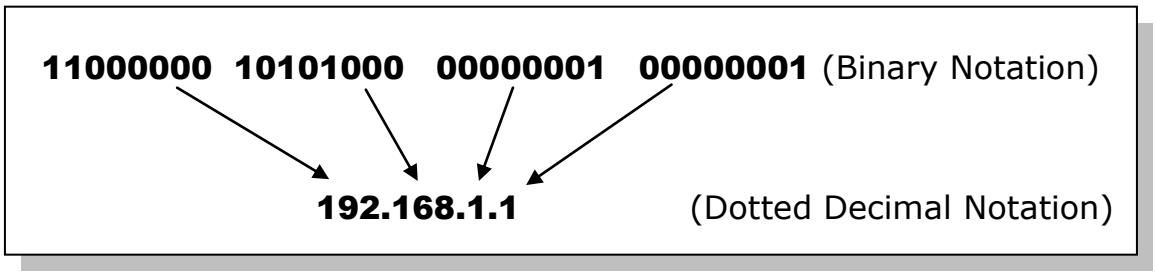


Figure 1 shows an IPv4 address in both binary and dotted-decimal notation.

HPv4 Header

IP header includes many relevant information including Version Number, which, in this context, is 4. IPv4 Header is show in Figure 2.

Version**:** Version no. of Internet Protocol used (e.g. 4 for IPv4)

IHL**:** Internet Header Length, Length of entire IP header

ToS**:** The ToS field could specify a datagram's priority and request a route for low-delay, high-throughput, or highly-reliable service. Based on these ToS values, a packet would be placed in a prioritized outgoing queue and take a route with appropriate latency, throughput, or reliability.

In practice, the ToS field is not used widely.

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 no. to identify original IP packet they belong to.

Flags: Flags. This is a 3-bit field.
- The first bit is reserved for future use and it is always '0'.
- The second bit is called the "*don't fragment*" bit. If its value is 1, the machine must not fragment the datagram. If its value is 0, the datagram can be fragmented if necessary.
- The third bit is called the "*more fragment*" bit. If its value is 1, it means the datagram is not the last fragment and more fragments are yet to be received. Thus, there are more fragments after this one. If its value is 0, it means this is the last or only fragment.

Fragment Offset: This offset tells the exact position of the fragment in the original IP Packet.

TTL: TTL stands for Time to Live. It is used 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.

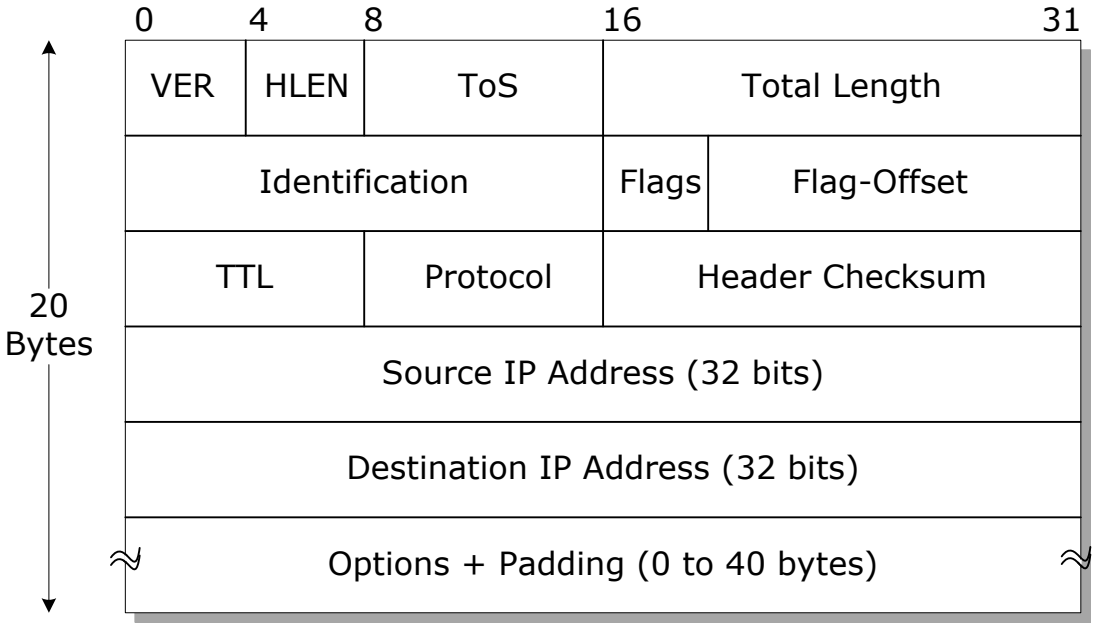


Figure 2 IPv4 Header

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 TCP is 6, UDP is 17 and ICMP is 1.

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 an 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.

2) IPv6

Internet Protocol version 6 (IPv6) is the latest version of the Internet Protocol (IP). It provides an identification and location system for computers on networks and routes traffic across the Internet. IPv6 was developed to deal with the long-anticipated problem of IPv4 address exhaustion. As IPv6 uses 128 bits for address, it allows 2^{128} different addresses.

In IPv6 new features are added. It has a large address space and a new efficient header. IPv6 is intended to replace IPv4, which still carries more than 96% of Internet traffic worldwide as of May 2014.

Every device on Internet is assigned an IP address for identification and location definition. IPv6 provides other technical benefits in addition to a large addressing space. The use of multicast addressing is expanded and simplified. It also provides additional optimization for the delivery of services.

IPv6 addresses are represented by 8 groups of 4 hexadecimal digits, separated by colons for ex. 3FFE:085B:1F1F:0000:0000:0000:00A9:1234. IPv6 header differs from IPv4 header. Figure 3 shows IPv6 Header. Various fields has significant meaning in the header as given below.

Ver: It is a 4 bits version field. It is used to identify the version of the IP. Here, it is set to 6.

Traffic class: This field is 4-bit priority field it defines the priority of the packet with respect to traffic congestion. The traffic class field is used to distinguish between packets with different real time delivery requirements.

Flow label: The flow label is a 3-byte (24-bit) field that is designed to provide special handling for a particular flow of data. We will discuss this field later.

Payload length: The 2-byte payload length field defines the length of the IP datagram excluding the base header.

Next header: The next header is an 8-bit field defining the header that follows the base header in the datagram. The next header is either one of the optional extension headers used by IP or the header of an encapsulated packet such as UDP or TCP.

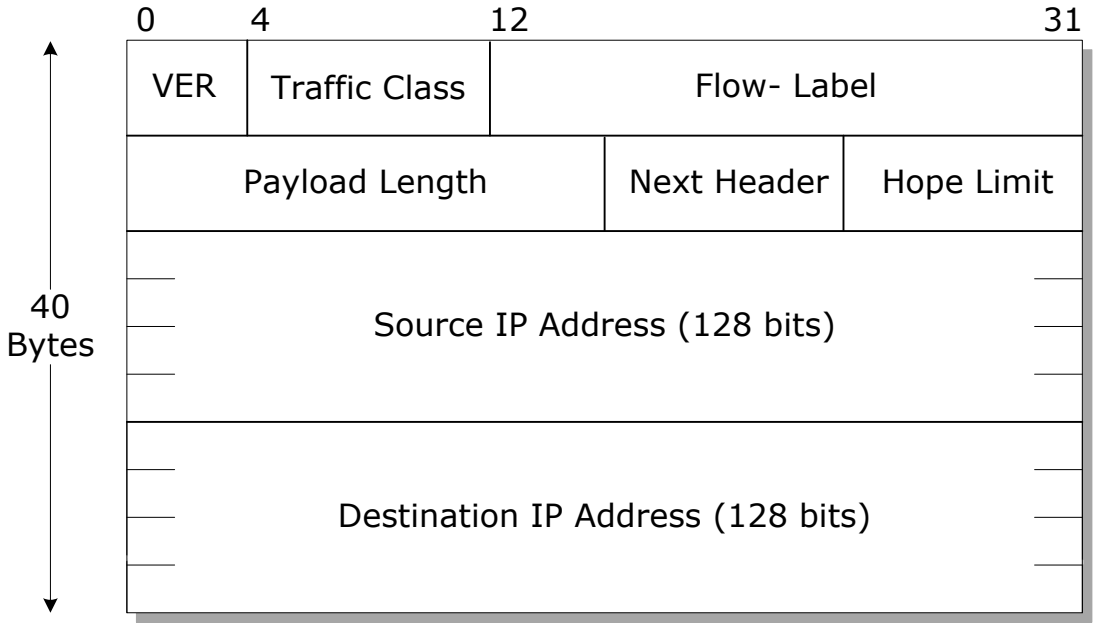


Figure 3 IPv6 Header

Hop limit: It indicates maximum number of links over which IPv6 packet can travel before being discarded. When the hope limit becomes 0, the packet is discarded. This 8-bit hop limit field serves the same purpose as the TTL field in IPv4.

Source address: The source address field is a 128-bit Internet address that identifies the original source of the datagram.

Destination address: The destination address field is a 16-byte (128-bit) Internet address that usually identifies the final destination of the datagram. However, if source routing is used, this field contains the address of the next router.

Thus, IPv6 header differs from that of IPv4 header. In IPv6 header some fields are removed like Header length, total length, option field, header checksum etc. while some fields are added to support routing, authentication, encapsulation and Confidentiality. The difference between IPv4 and IPv6 are shown in Table 1.

IPV4	IPV6
1) Source and Destination Addresses are 32 bits long.	1) Source and Destination Addresses are 128 bits long.
2) It includes Header Checksum.	2) The header checksum is eliminated because the checksum is provided by upper-layer Protocols.
3) Header includes Options.	3) Header doesn't include Options. But Extension headers are available.
4) Header Length field is there. (IHL)	4) Header Length field is removed because the length of the header is fixed in this version.
5) Time to Live Field is there. (TTL)	5) Time to Live Field is replaced by Hop Limit.
6) Total Length field is there.	6) Total Length field is replaced by payload length field.
7) Types of Services field is there. (TOS)	7) Types of Services field is replaced by Priority and Flow Label Fields.
8) Protocol field is there.	8) Protocol field is replaced by Next header field.
9) Identification, flags and Offset fields are there.	9) Identification, flags and Offset fields are eliminated. (They are included in Fragmentation Extension Header).
10) Network security is not integrated into design of IPv4 Architecture.	10) Network security is integrated into design of IPv6 Architecture.

Table 1 Difference between IPv4 and IPv6

2 Addressing Schemes

The Two Parts of an IP Address

An IP address consists of two parts, one identifying the network and one identifying the node, or host. The Class of the address determines which part belongs to the network address and which part belongs to the node address. All nodes on a given network share the same network prefix but must have a unique host number.

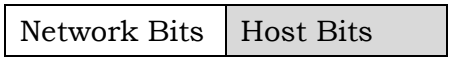


Figure 4 Two parts of an IP address

IP addresses are divided into 5 classes. Class A, class B, class C, class D and class E. Each class has different number of Network bits and Host bits. Each class has its own use. Class A, class B and class C are used for unicasting. Class D is used for multicasting and class E is reserved for future research and development. Various classes can be differentiated by their initial bits of the IP address as shown in the Figure 5

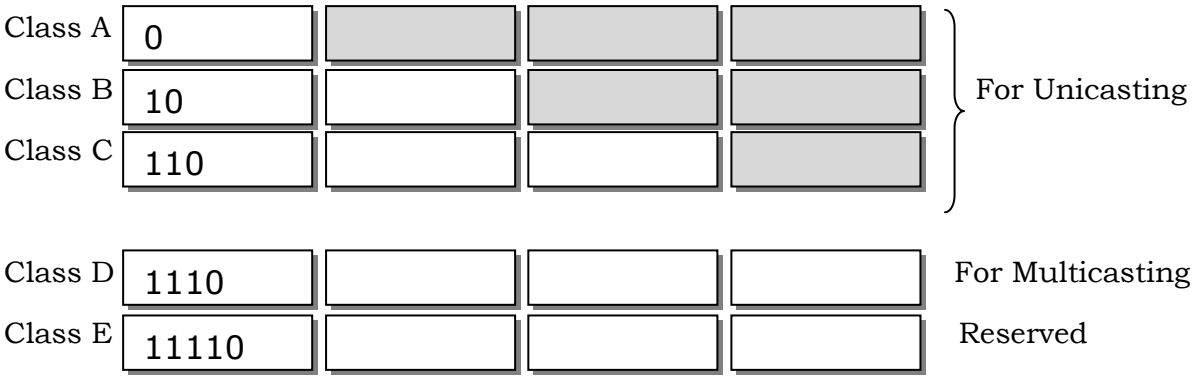


Figure 5 Different classes of IP Address

Based on the initial bits for the IP address in various classes, Range of the IP addresses also differs. It is shown in the in the table 2. The table also shows the number of network bits, host bits and the maximum number of networks (without sub netting) possible in each class.

Class A Network

As show in the table, In a Class A Network, there are 8 network bits and address always starts with 0. Thus, first position in class A network is fixed. For remaining 7 positions 2^7 = 128 no. of different networks can be constructed. Therefore for the first byte, the decimal number can be anywhere from 0 to 127.

In class A network, first 8 bits identify the network and the remaining 24 bits indicate the host within the network. An example of a Class A IP address is 102.168.212.226, where "102" identifies the network and "168.212.226" identifies the host on that network. Class A addresses are assigned to networks with a very large number of hosts.

Class	**Leftmost Bits**	**Network Bits**	**Host Bits**	**No. of Networks**	**IP address Range**
Class A	0	8	24	2^7	**0**.0.0.0 **to** **127**.255.255.255
Class B	10	16	16	2^{14}	**128**.0.0.0 **to** **191**.255.255.255
Class C	110	24	8	2^{21}	**192**.0.0.0 **to** **223**.255.255.255
Class D	1110	Reserved for Multicasting			**224**.0.0.0 **to** **239**.255.255.255
Class E	1111	Reserved for Research and development			**240**.0.0.0 **to** **255**.255.255.255

Table 2 Different classes of IP Address

Note: Class A IP address 127.x.y.z is reserved for loopback testing and is used for internal testing on the local machine.

Class B Network

As show in the table, In a Class B Network, there are 16 network bits and address always starts with 10. Thus, first two positions in class B network are fixed. For remaining 14 positions 2^{14} (= 16,384) no. of different networks can be constructed. Therefore for the first byte the decimal number can be anywhere from 128 to 191.

In class B network, The first 16 bits (the first two octets) identify the network and the remaining 16 bits indicate the host within the network. An example of a Class B IP address is 168.212.226.204 where "168.212" identifies the network and "226.204" identifies the host on that network. Class B addresses are assigned to medium to large sized networks.

Class C Network

As show in the table, In a Class C Network, there are 24 network bits and binary address always starts with 110. Thus, first three positions in class C network are fixed. For remaining 21 positions 2^{21} (= 16,384) no. of different networks can be constructed. Therefore for the first byte the decimal number can be anywhere from 192 to 223.

The first 24 bits (the first three octets) identify the network and the remaining 8 bits indicate the host within the network. An example of a Class C IP address is 200.168.212.226 where "200.168.212" identifies the network and "226" identifies the host on that network. Class C addresses are used for small networks.

Class D Network

As show in the table, Class D Network, doesn't distinguish between Network bits and Host bits. Class D networks are used to support multicasting. In a Class D Network, binary addresses start with 1110, therefore, for the first byte, the decimal number can be anywhere from 224 to 239. An example of a Class D IP address is 224.100.200.220

Class E Network

In a Class E Network, binary addresses start with 1111. Therefore for the first byte, the decimal number can be anywhere from 240 to 255. Class E

networks are used for experimentation. They have never been documented or utilized in a standard way. An example of a Class E IP address is 240.240.240.240

Network Address

Network address is a special address that defines the network itself. It cannot be assigned to any host. In network address all the host Id bits are zeros. In other words network address is the first address of the block. A router can route a packet based on the network address. Ex. 192.168.2.0 is a class C network address and it supports up to 254 (2^8 -2) hosts. Two is subtracted because first and last addresses (all 0s and all 1s) cannot be used. First address is assigned to the network and the last address is reserved for special purpose.

3 Subnet & masking

IP addressing is designed with two-level of hierarchy. To reach a host on the Internet, we must first reach to the network using first portion of the IP address i.e. network Id. Then we must reach to the host by using the second portion i.e. host Id.

Network Id → Host Id

Sometimes two level of hierarchy is not suitable to the organization, so at that point network needs to be divided into several small networks. “The further division of the network into sunbnetworks is called sub network”.

For example, there is a university, for which there can be a single network address. If there are many departments in the university, then different subnetworks can be assigned to different departments.

When we divide a network into several subnets, we have 3 levels of hierarchy as below.

Network Id → Subnetwork Id → Host Id

Subnet example for class B

As we know, class B addresses have 16 network bits and 16 host bits. The network bits always start with 10. So maximum 2^{14} number of networks are possible. If the need arises to have even more networks then, some of the host bits can be used as network bits. This is called subnetting.

For example, a class B network address 128.128.0.0 various subnetworks can be created based on number of host bits used for subnets.

Decimal notation 128.128.0.0 can be represented in binary notation as follows:

10000000.10000000.00000000.0000000

Here, some of the last 16 host bits may be used as the subnet bits. To create 2 subnets, 1 host bit can be used. Which can allow 32,766 hosts per subnet.

Masking

When a router receives a packet with a destination address, it needs to route a packet. The routing is based on the Network address and subnetwork address. The routers outside the organization (network) routes the packet based on Network Address. The router inside the organization routes the packet based on subnetwork address.

How can a router find network address and subnetwork address? Here is the answer: The network administrator knows the network address and subnetwork address, but router doesn't. So Router uses a process known as 'Masking'.

"Masking is the process that extracts the network address from an IP address."

Masking can be done with and without subnetting. If we have not subnetted the network, masking extracts the network address from an IP address. If the network is divided into several subnetworks then masking extracts the subnet address from an IP address.

Default Subnet Masks for class A, B and C are as follows:

Class A – 255.0.0.0

Class B – 255.255.0.0

Class C – 255.255.255.0

Masking without subnetting is shown in the following figure.

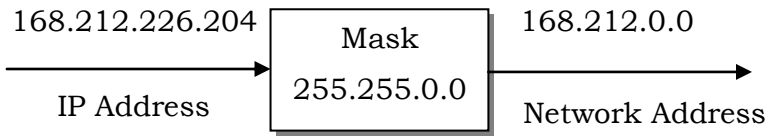


Figure 6 Masking of an IP address without subnetting

Masking with subnetting is shown in the following figure.

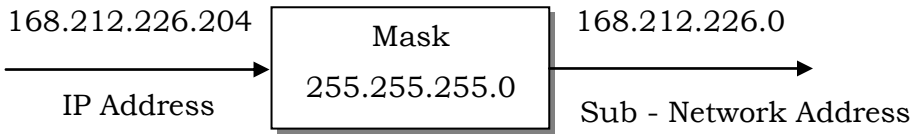


Figure 7 Masking of an IP address with subnetting

4 DNS

DNS stands for Domain Name System. To identify an entity, TCP/IP protocols use the IP address, which uniquely identifies the connection of a host to the Internet. However, people prefer to use names instead of numeric addresses. Therefore, we need a system that can map a name to an address or an address to a name. DNS does this mapping. DNS transfers the name into IP address. For example if a user types "www.google.com", DNS will map this site's name into its IP address.

Domain Name Space

A domain name space was designed, to have a hierarchical name space. In this design the names are defined in an inverted-tree structure with the root at the

top. The tree can have only 128 levels: level 0 (root) to level 127 (see Figure 2). Each node in the tree has a label; the root label is a null string (empty string). DNS requires that children of a node should have different labels, which guarantees the uniqueness of the domain names.

A domain is a "sub tree of the domain name space". The name of the domain is the domain name of the node at the top of the sub tree.

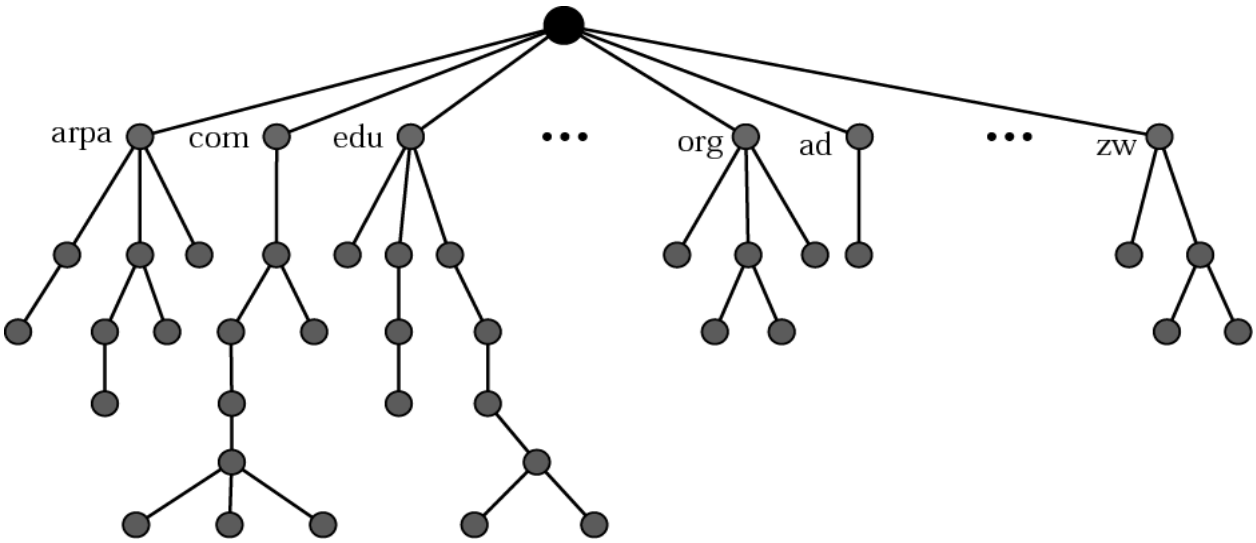


Figure 8 Domain name space

Distribution of Name Space

"The information stored in the domain name spaces are distributed among many computers called DNS servers". The whole space is divided into many domains based on the first level. As DNS allows domains to be divided further into smaller domains (subdomains). Thus, we have a hierarchy of servers in the same way that we have a hierarchy of names as shown in figure 9.

Root Server

A root server is a server whose zone (a name server) consists of the whole tree. A root server usually does not store any information about domains but delegates its authority to other servers, keeping references to those servers.

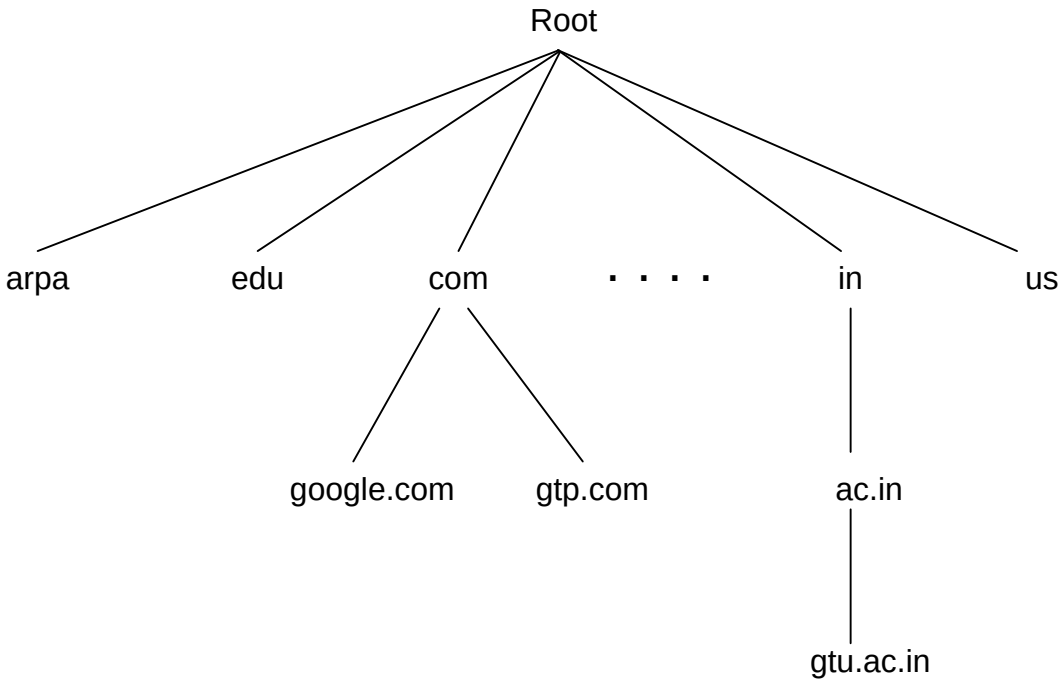


Figure 9 Hierarchy of name servers

Primary and Secondary Servers

DNS defines two types of servers: primary and secondary. A primary server is a server that stores a file about the zone for which it is an authority. A secondary server is a server that transfers the complete information about a zone from another server (primary or secondary) and stores the file on its local disk. The secondary server neither creates nor updates the zone files. If updating is required, it must be done by the primary server, which sends the updated version to the secondary server. The primary and secondary servers are both authoritative for the zones they serve. When the secondary downloads information from the primary server, it is called zone transfer.

Types of DNS

The domain name space (tree) is divided into three different sections: generic domains, country domains, and the inverse domain.

1) Generic Domains

The generic domains define registered hosts according to their generic behavior. Each node in the tree defines a domain, which is an index to the domain name space database. The labels of the nodes describe the type of the organization as listed in Table 3.

Label	***Description***
Aero	Airlines and aerospace companies
Biz	Businesses or firms (similar to "com")
Com	Commercial organizations
Coop	Cooperative business organizations
Edu	Educational institutions
Gov	Government institutions
Info	Information service providers
Int	International organizations
Mil	Military groups
Museum	Museums and other nonprofit organizations
Name	Personal names (individuals)
Net	Network support centers
Org	Nonprofit organizations
Pro	Professional individual organizations

Table 3 Generic domain labels

2) Country Domains

The country domains section uses two-character country abbreviations. (e.g., in for India). Second labels can be organizational, or they can be more specific, national designations. For example, In India, for any non-profit organization .org.in abbreviation is used. Following are some examples of the country domains.

- .in (available to anyone; used by companies, individuals, and organizations in India)
- .co.in (originally for banks, registered companies, and trademarks)
- .firm.in (originally for shops, partnerships, liaison offices,sole proprietorships)
- .net.in (originally for Internet service providers)
- .org.in (originally for non-profit organisations)
- .gen.in (originally for general/miscellaneous use)
- .ind.in (originally for individuals)

Six zones are reserved for use by qualified institutions in India:

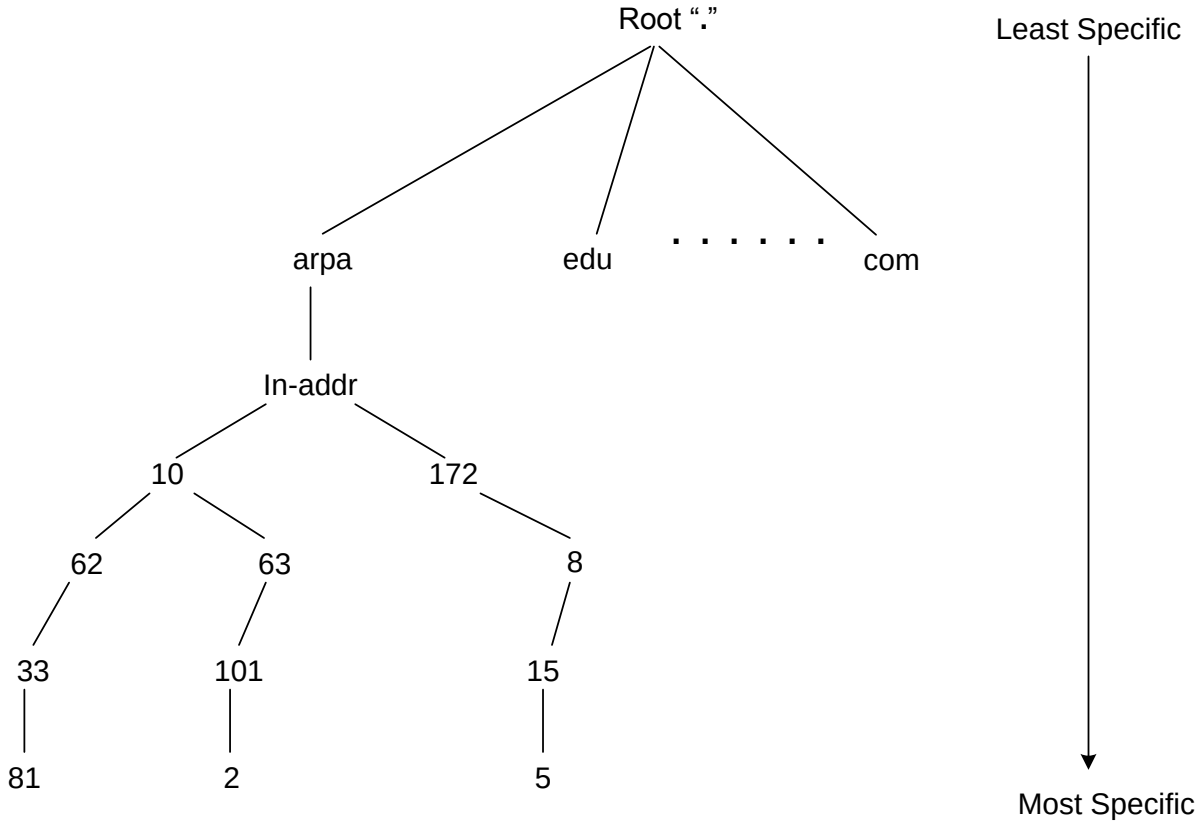
- .ac.in (Academic institutions)
- .edu.in (Educational institutions)
- .res.in (Indian research institutes)
- .ernet.in (Older, for both educational and research institutes)
- .gov.in (Indian government)
- .mil.in (Indian military)

3) Inverse Domain

"The inverse domain is used to map an address to a name". This may happen, for example, when a server has received a request from a client to do a task. Although the server has a file that contains a list of authorized clients, only the IP address of the client (extracted from the received IP packet) is listed.

The server asks its resolver to send a query to the DNS server to map an address to a name to determine if the client is on the authorized list. This type of query is called an inverse or pointer query.

To handle a pointer query, the inverse domain is added to the domain name space with the first-level node called *arpa*. The second level is also one single node named *in-addr* (for inverse address).



IP Address	DNS Name
10.62.33.81	81.33.62.10.in-addr.arpa.
10.63.101.2	2.101.63.10.in-addr.arpa.
172.8.15.5	5.15.8.172.in-addr.arpa.

Figure 10 Example of inverse domains and the Domain Name Space

The rest of the domain defines IP addresses. The servers that handle the inverse domain are also hierarchical. This means the netid part of the address should be at a higher level than the subnetid part, and the subnetid part higher than the hostid part. This configuration makes the domain look inverted. The figure 10 illustrates the inverse domain.

5 Email

One of the most popular Internet services is electronic mail (e-mail). Its architecture consists of several components. At the beginning of the Internet era, the messages sent by electronic mail were short and consisted of text only.

Today, electronic mail is much more complex. It allows a message to include text, audio, and video. It also allows one message to be sent to one or more recipients.

E-mail system includes the three main components: user agent, message transfer agent, and message access agent.

User Agent

The first component of an electronic mail system is the user agent. It provides service to the user to make the process of sending and receiving a message easier. A user agent is a software package (program) that reads, composes, replies to and forwards messages. It also handles mailboxes. Figure 10 shows the services of a typical user agent.

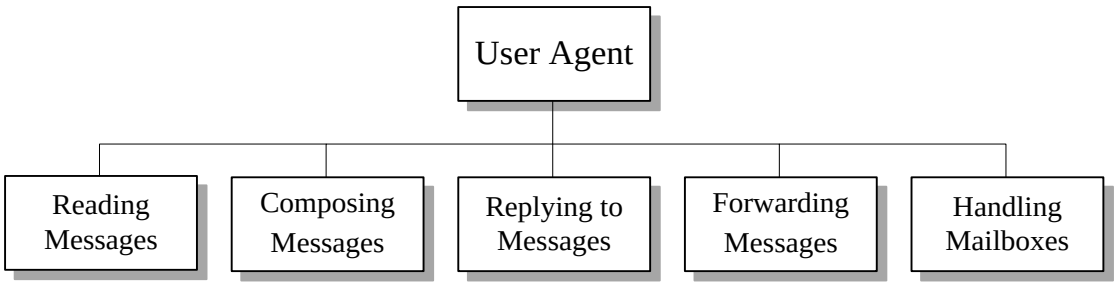


Figure 11 Services of user agent

1) Reading Messages

It is possible to read incoming messages using user agent. When a user invokes a user agent, it first checks the mail in the incoming mailbox. Most user agents show a one-line summary of each received mail. Each e-mail contains the following fields.

- The sender.
- The optional subject field.
- A flag field that shows the status of the mail such as new, already read but not
- replied to, or read and replied to.
- The size of the message.
- A number field.

2) Composing Messages

A user agent allows users to compose the e-mail message to be sent out. Most user agents provide a template on the screen to be filled in by the user. Some even have a built-in editor that can do spell checking, grammar checking, and other tasks expected from a sophisticated word processor.

A user, of course, could alternatively use his or her favorite text editor or word processor to create the message and import it, or cut and paste it, into the user agent template.

3) Replying to Messages

After reading a message, a user can use the user agent to reply to a message. A user agent usually allows the user to reply to the original sender or to reply to all recipients of the message. The reply message may contain the original message (for quick reference) and the new message.

4) Forwarding Messages

Replying is defined as sending a message to the sender or recipients of the copy. *Forwarding* is defined as sending the message to a third party. A user agent allows the receiver to forward the message, with or without extra comments, to a third party.

Example: If Alice receives a message from Bob and like to send the same message to Robert, then she can use the facility of forwarding message.

5) Handling Mailboxes

A user agent normally creates two mailboxes: an inbox and an outbox. Each box is a file with a special format that can be handled by the user agent. The inbox keeps all the received e-mails until they are deleted by the user.

The outbox keeps all the sent e-mails until the user deletes them. Most user agents today are capable of creating customized mailboxes.

Sending Mail

Header	**Meaning**
To:	E-mail addresses of primary recipient(s)
Cc:	E-mail addresses of secondary recipient(s)
Bcc:	E-mail addresses for blind carbon copies
From:	Person or people who created the message
Sender:	E-mail address of the actual sender

Table 4 E-mail Header Fields

To send mail, the user, through the UA, creates mail that looks very similar to postal mail. It has an envelope and a message. The header contains the fields shown in the table 4

To: The e-mail addresses of the primary recipients are written in this field.

Cc: Carbon copy : E-mail addresses of secondary recipient(s) are written here. Many email clients will mark email in one's inbox differently depending on whether they are in To: or Cc: list.

Bcc: Blind carbon copy
Addresses added to the SMTP delivery list but not (usually) listed in the message data, remaining invisible to other recipients.

E-mail envelope usually contains the sender and the receiver addresses. Message The message contains the header and the body. The header of the message defines the sender, the receiver, the subject of the message, and some other information. The body of the message contains the actual information to be read by the recipient.

Receiving Mail

The user agent is triggered by the user (or a timer). If a user has mail, the *VA* informs the user with a notice. If the user is ready to read the mail. A list is displayed in which each line contains a summary of the information about a particular message in the mailbox. The summary usually includes the sender mail address, the subject, and the time the mail was sent or received. The user can select any of the messages and display its contents on the screen.

6 FTP

Transferring files from one computer to another is one of the most common tasks expected from a networking or internetworking environment. FTP is a popular protocol involved in transferring files. FTP stands for File Transfer Protocol.

File Transfer Protocol (FTP) is the standard mechanism provided by *TCP/IP* for copying a file from one host to another. Although transferring files from one system to another seems simple and straightforward, some problems must be dealt with first. For example, two systems may use different file name conventions. Two systems may have different ways to represent text and data. Two systems may have different directory structures. All these problems have been solved by FTP in a very simple and elegant approach.

FTP differs from other client/server applications in that it establishes two connections between the hosts. One connection is used for data transfer, the other for control information (commands and responses). Separation of commands and data transfer makes FTP more efficient. The control connection uses very simple rules of communication.

We need to transfer only a line of command or a line of response at a time. FTP uses two well-known TCP ports: port 20 is used for the data connection and port 21 is used for the control connection.

Figure 12 shows the basic model of FTP. The client has three components: user interface, client control process, and the client data transfer process. The server has two components: the server control process and the server data transfer

process. The control connection is made between the control processes. The data connection is made between the data transfer processes.

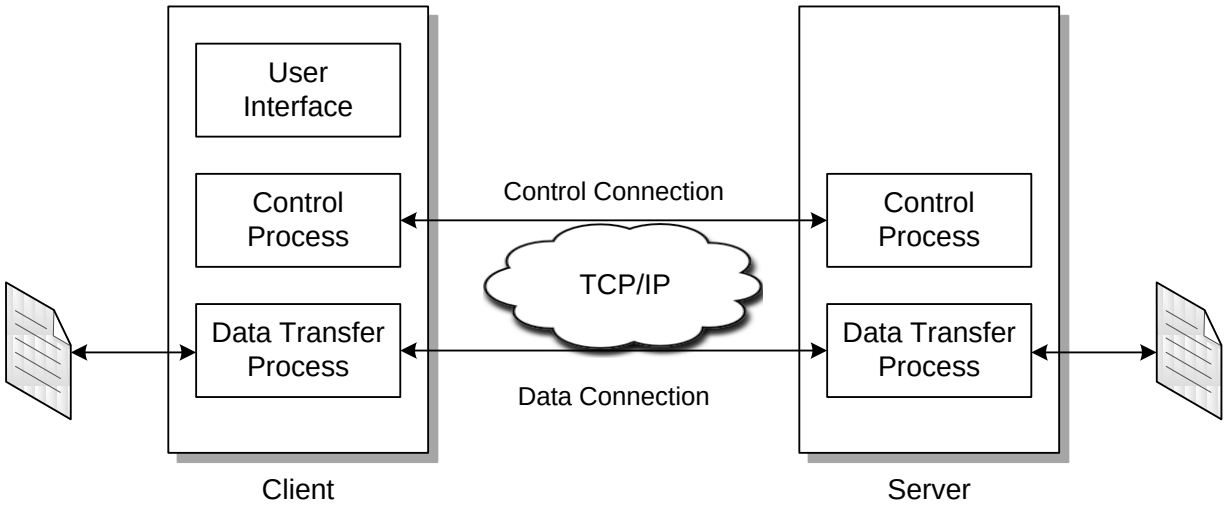


Figure 12 FTP

The control connection remains connected during the entire interactive FTP session. The data connection is opened and then closed for each file transferred. It opens each time commands that involve transferring files are used, and it closes when the file is transferred. In other words, when a user starts an FTP session, the control connection opens. While the control connection is open, the data connection can be opened and closed multiple times if several files are transferred.

Trivial FTP

TFTP stands for Trivial File Transfer Protocol. It is very simple to implement. One of its primary uses is in the early stages of nodes booting from a Local Area Network. TFTP allows only unidirectional transfer of files. It doesn't provide authentication. TFTP depends on UDP, requires less overhead and provides virtually no control. TFTP uses UDP port number 69 for file transfer. Table 5 shows the difference between FTP and TFTP.

FTP	TFTP
1) File Transfer Protocol	1) Trivial File Transfer Protocol
2) FTP is more complex than TFTP.	2) Very simple to implement.
3) FTP is a complete, session-oriented, general purpose file transfer protocol	3) One of its primary uses is in the early stages of nodes booting from a Local Area Network.
4) FTP can ne used interactively.	4) TFTP allows only unidirectional transfer of files.
5) FTP provides authentication.	5) TFTP doesn't provide authentication
6) FTP depends on TCP, is connection, and provides reliable control	6) TFTP depends on UDP, requires less overhead and provides virtually no control.
7) FTP uses well known TCP port numbers: 20 for data, 21 for connection dialog.	7) TFTP uses UDP port number 69 for file transfer.

Table 5 Difference between FTP and TFTP

7 HTTP

The Hypertext Transfer Protocol (HTTP) is a protocol used mainly to access data on the World Wide Web. HTTP functions as a combination of FTP and SMTP. It is similar to FTP because it transfers files and uses the services of TCP. However, it is much simpler than FTP because it uses only one TCP connection. There is no separate control connection; only data are transferred between the client and the server.

HTTP is like SMTP because the data transferred between the client and the server look like SMTP messages. In addition, the format of the messages is controlled by MIME-like headers.

Unlike SMTP, the HTTP messages are not destined to be read by humans; they are read and interpreted by the HTTP server and HTTP client (browser). SMTP messages are stored and forwarded, but HTTP messages are delivered immediately.

The commands from the client to the server are embedded in a request message. The contents of the requested file or other information are embedded in a response message. HTTP uses the services of TCP on well-known port 80.

HTTP Transaction

Figure 13 illustrates the HTTP transaction between the client and server. Although HTTP uses the services of TCP, HTTP itself is a stateless protocol. The client initializes the transaction by sending a request message. The server replies by sending a response.

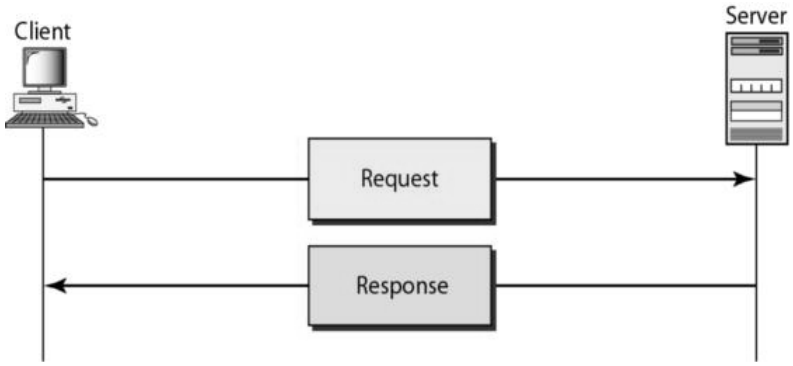


Figure 13 HTTP transaction

The formats of the request and response messages are similar; both are shown in Figure 14. A request message consists of a request line, a header, and sometimes a body.

The first line in a request message is called a request line; the first line in the response message is called the status line.

Request type: This field is used in the request message. In version 1.1 of HTTP, several request types are defined. The request type is categorized into *methods* as defined in Table 6.

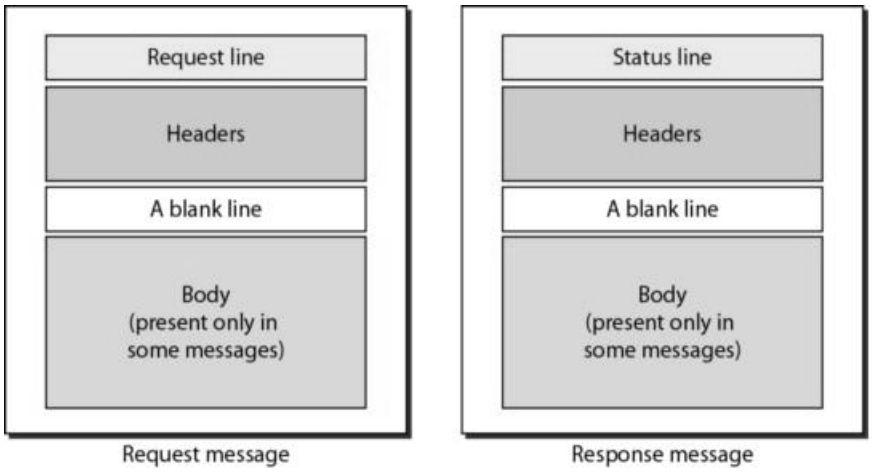


Figure 14 Request and response messages

Method	Action
GET	Requests a document from the server
HEAD	Requests information about a document but not the document itself
POST	Sends some information from the client to the server
PUT	Sends a document from the server to the client
TRACE	Echoes the incoming request
CONNECT	Reserved
OPTION	Inquires about available options

Table 6 HTTP Methods

* * * * * * * *