ISSN NO. 2320-5407

Journal homepage: http://www.journalijar.com
Journal DOI: 10.21474/IJAR01

INTERNATIONAL JOURNAL OF ADVANCED RESEARCH

RESEARCH ARTICLE

COMPUTER NETWORK -- APPLICATION LAYER PROTOCOL.

Rajesh Kumar[1], Vivek Kumar Vaidya[2].

1. Senior Lecturer, Department of Computer Science and Engineering, Govt. Polytechnic College, Ujjain-456001 (M.P.)-India.
2. Senior Lecturer, Department of Computer Science and Engineering, Jija Mata Govt. Polytechnic College, Burhanpur-450331 (M.P.)-India

Manuscript Info

Manuscript History:

Received: 12 April 2016
Final Accepted: 19 May 2016
Published Online: June 2016

Key words:
HTTP, DNS, TELNET, FTP, SMTP.

***Corresponding Authors**

Rajesh Kumar,
Vivek Kumar Vaidya.

Abstract

A majority of the internet uses a protocol suite called the Internet Protocol Suite also known as the TCP/IP protocol suite. This suite is a combination of protocols which encompasses a number of different protocols for different purpose and need. Because the two major protocols in this suites are TCP (Transmission Control Protocol) and IP (Internet Protocol), this is commonly termed as TCP/IP Protocol suite. This protocol suite has its own reference model which it follows over the internet. In contrast with the OSI model, this model of protocols contains less layers. An application layer is an abstraction layer that specifies the shared protocols and interface methods used by hosts in a communications network. The application layer abstraction is used in both of the standard models of computer networking; the Internet Protocol Suite (TCP/IP) and the Open Systems Interconnection model (OSI model). There are several protocols which work for users in Application Layer. Application layer protocols can be broadly divided into two categories.

Introduction:-

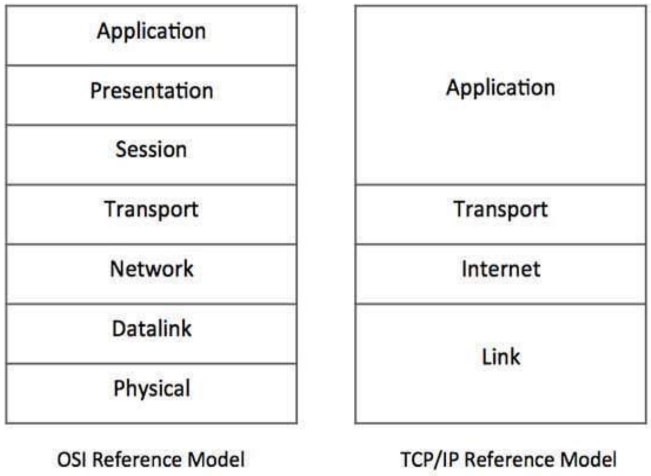
An application layer is an abstraction layer that specifies the shared protocols and interface methods used by hosts in a communications network. The application layer abstraction is used in both of the standard models of computer networking; the Internet Protocol Suite (TCP/IP) and the Open Systems Interconnection model (OSI model).

Although both models use the same term for their respective highest level layer, the detailed definitions and purposes are different.

In TCP/IP, the application layer contains the communications protocols and interface methods used in process-to-process communications across an Internet Protocol (IP) computer network. The application layer only standardizes communication and depends upon the underlying transport layer protocols to establish host-to-host data transfer channels and manage the data exchange in a client-server or peer-to-peer networking model. Though the TCP/IP application layer does not describe specific rules or data formats that applications must consider when communicating, the original specification (in RFC 1123) does rely on and recommend the robustness principle for application design.[1]

In the OSI model, the definition of the application layer is narrower in scope. The OSI model defines the application layer as the user interface responsible for displaying received information to the user. In contrast, the Internet Protocol model does not concern itself with such detail. OSI also explicitly distinguishes additional functionality below the application layer, but above the transport layer at two additional levels; the session layer and the presentation layer. OSI specifies a strict modular separation of functionality at these layers and provides protocol implementations for each layer.

A majority of the internet uses a protocol suite called the Internet Protocol Suite also known as the TCP/IP protocol suite. This suite is a combination of protocols which encompasses a number of different protocols for different purpose and need. Because the two major protocols in this suites are TCP (Transmission Control Protocol) and IP (Internet Protocol), this is commonly termed as TCP/IP Protocol suite. This protocol suite has its own reference model which it follows over the internet. In contrast with the OSI model, this model of protocols contains less layers.



This model is indifferent to the actual hardware implementation, i.e. the physical layer of OSI Model. This is why this model can be implemented on almost all underlying technologies.

Transport and Internet layers correspond to the same peer layers. All three top layers of OSI Model are compressed together in single Application layer of TCP/IP Model.

HTTP:-

The Hypertext Transfer Protocol (HTTP) is an application protocol for distributed, collaborative, hypermedia information systems.[1] HTTP is the foundation of data communication for the World Wide Web. Hypertext is structured text that uses logical links (hyperlinks) between nodes containing text. HTTP is the protocol to exchange or transfer hypertext.The standards development of HTTP was coordinated by the Internet Engineering Task Force (IETF) and the World Wide Web Consortium(W3C), culminating in the publication of a series of Requests for Comments (RFCs). The first definition of HTTP/1.1, the version of HTTP in common use, occurred in RFC 2068 in 1997, although this was obsoleted by RFC 2616 in 1999.A later version, the successor HTTP/2, was standardized in 2015, then supported by major web browsers and already supported by major web servers.

HTTP functions as a request–response protocol in the client–server computing model. A web browser, for example, may be the client and an application running on a computer hosting a web site may be the server. The client submits an HTTP request message to the server. The server, which provides resources such as HTML files and other content, or performs other functions on behalf of the client, returns a response message to the client. The response contains completion status information about the request and may also contain requested content in its message body.

A web browser is an example of a user agent (UA). Other types of user agent include the indexing software used by search providers (web crawlers),voice browsers, mobile apps, and other software that accesses, consumes, or displays web content.

HTTP is designed to permit intermediate network elements to improve or enable communications between clients and servers. High-traffic websites often benefit from web cache servers that deliver content on behalf of upstream servers to improve response time. Web browsers cache previously accessed web resources and reuse them when possible to reduce network traffic. HTTP proxy servers at private network boundaries can facilitate communication for clients without a globally routable address, by relaying messages with external servers.

HTTP is an application layer protocol designed within the framework of the Internet Protocol Suite. Its definition presumes an underlying and reliable transport layer protocol,[2] and Transmission Control Protocol (TCP) is commonly used. However HTTP can be adapted to use unreliable protocols such as the User Datagram Protocol (UDP), for example in HTTPU and Simple Service Discovery Protocol (SSDP).

HTTP resources are identified and located on the network by uniform resource locators (URLs), using the uniform resource identifier (URI) schemes http and https. URIs and hyperlinks inHypertext Markup Language (HTML) documents form inter-linked hypertext documents.

HTTP/1.1 is a revision of the original HTTP (HTTP/1.0). In HTTP/1.0 a separate connection to the same server is made for every resource request. HTTP/1.1 can reuse a connection multiple times to download images, scripts, stylesheets, etc after the page has been delivered. HTTP/1.1 communications therefore experience less latency as the establishment of TCP connections presents considerable overhead.

The Hyper Text Transfer Protocol (HTTP) is the foundation of World Wide Web. Hypertext is well organized documentation system which uses hyperlinks to link the pages in the text documents. HTTP works on client server model. When a user wants to access any HTTP page on the internet, the client machine at user end initiates a TCP connection to server on port 80. When the server accepts the client request, the client is authorized to access web pages.

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

HTTP versions:-

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

DNS:-

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

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

The Domain Name System (DNS) is a hierarchical decentralized naming system for computers, services, or any resource connected to the Internet or a private network. It associates various information with domain names assigned to each of the participating entities. Most prominently, it translates more readily memorized domain names to the numerical IP addresses needed for the purpose of locating and identifying computer services and devices with the underlying network protocols. By providing a worldwide, distributed directory service, the Domain Name System is an essential component of the functionality of the Internet.

The Domain Name System delegates the responsibility of assigning domain names and mapping those names to Internet resources by designating authoritative name servers for each domain. Network administrators may delegate authority over sub-domains of their allocated name space to other name servers. This mechanism provides distributed and fault tolerant service and was designed to avoid a single large central database.

The Domain Name System also specifies the technical functionality of the database service which is at its core. It defines the DNS protocol, a detailed specification of the data structures and data communication exchanges used in the DNS, as part of the Internet Protocol Suite. Historically, other directory services preceding DNS were not scalable to large or global directories as they were originally based on text files, prominently the HOSTS.TXT resolver. The Domain Name System has been in use since the 1980s.

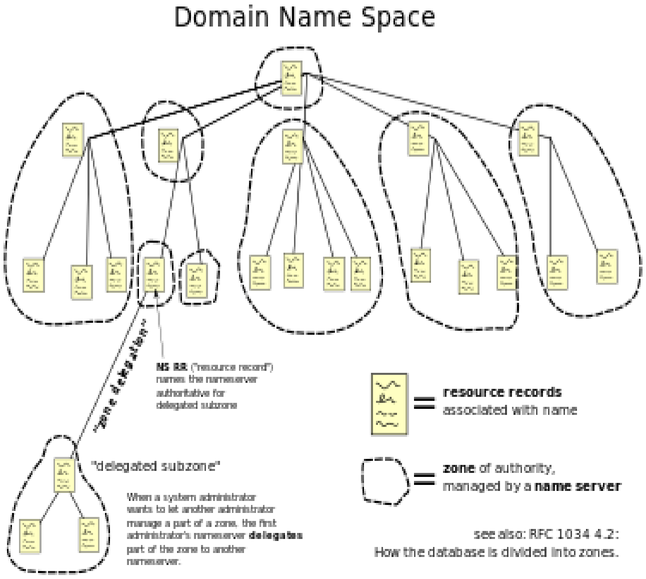
The Internet maintains two principal namespaces, the domain name hierarchy[1] and the Internet Protocol (IP) address spaces.[2] The Domain Name System maintains the domain name hierarchy and provides translation services between it and the address spaces. Internet name servers and a communication protocol implement the Domain Name System.[3] A DNS name server is a server that stores the DNS records for a domain; a DNS name server responds with answers to queries against its database.

The most common types of records stored in the DNS database are for DNS zone authority (SOA), IP addresses (A and AAAA), SMTP mail exchangers (MX), name servers (NS), pointers for reverse DNS lookups (PTR), and domain name aliases (CNAME). Although not intended to be a general purpose database, DNS can store records for other types of data for either automatic lookups, such as DNSSEC records, or for human queries such as responsible person (RP) records. As a general purpose database, the DNS has also been used in combatingunsolicited email (spam) by storing a real-time blackhole list. The DNS database is traditionally stored in a structured zone file.

Structure:-

Domain Name Space:-

The domain name space consists of a tree data structure. Each node or leaf in the tree has a label and zero or more resource records (RR), which hold information associated with the domain name. The domain name itself consists of the label, possibly concatenated with the name of its parent node on the right, separated by a dot.[14] The tree sub-divides into zones beginning at the root zone. A DNS zone may consist of only one domain, or may consist of many domains and sub-domains, depending on the administrative choices of the zone manager. DNS can also be partitioned according to class; the separate classes can be thought of as an array of parallel namespace trees.[15]


Domain Name Space
"zone delegation"
NS RR ("resource record") names the nameserver authoritative for delegated subzone
= resource records associated with name
= zone of authority, managed by a name server
"delegated subzone"
When a system administrator wants to let another administrator manage a part of a zone, the first administrator's nameserver delegates part of the zone to another nameserver.
see also: RFC 1034 4.2:
How the database is divided into zones.

The hierarchical Domain Name System for class Internet, organized into zones, each served by a name server

Administrative responsibility over any zone may be divided by creating additional zones. Authority over the new zone is said to be delegated to a designated name server. The parent zone ceases to be authoritative for the new zone.

Domain Name Syntax:-

The definitive descriptions of the rules for forming domain names appear in RFC 1035, RFC 1123, and RFC 2181. A domain name consists of one or more parts, technically called labels, that are conventionally concatenated, and delimited by dots, such as example.com.

The right-most label conveys the top-level domain; for example, the domain name www.example.com belongs to the top-level domain com.

The hierarchy of domains descends from right to left; each label to the left specifies a subdivision, or subdomain of the domain to the right. For example: the label example specifies a subdomain of the com domain, and www is a subdomain of example.com. This tree of subdivisions may have up to 127 levels.

A label may contain zero to 63 characters. The null label, of length zero, is reserved for the root zone. The full domain name may not exceed the length of 253 characters in its textual representation.[1] In the internal binary representation of the DNS the maximum length requires 255 octets of storage, since it also stores the length of the name.[3]

Although domain names may theoretically consist of any character representable in an octet, host names use a preferred format and character set. The characters allowed in their labels are a subset of the ASCII character set, consisting of characters a through z, A through Z, digits 0 through 9, and hyphen. This rule is known as the LDH rule (letters, digits, hyphen). Domain names are interpreted in case-independent manner.[16] Labels may not start or end with a hyphen.[17] An additional rule requires that top-level domain names should not be all-numeric.[17]

Internationalized Domain Names:-

The limited set of ASCII characters permitted in the DNS prevented the representation of names and words of many languages in their native alphabets or scripts. To make this possible, ICANN approved the Internationalizing Domain Names in Applications (IDNA) system, by which user applications, such as web browsers, map Unicode strings into the valid DNS character set using Punycode. In 2009 ICANN approved the installation of internationalized domain name country code top-level domains (ccTLDs). In addition, many registries of the existing top level domain names (TLDs) have adopted the IDNA system.

Name servers:-

The Domain Name System is maintained by a distributed database system, which uses the client–server model. The nodes of this database are the name servers. Each domain has at least one authoritative DNS server that publishes information about that domain and the name servers of any domains subordinate to it. The top of the hierarchy is served by the root name servers, the servers to query when looking up (resolving) a TLD.

Authoritative name server[edit]:-

An authoritative name server is a name server that gives answers that have been configured by an original source, for example, the domain administrator or by dynamic DNS methods, in contrast to answers that were obtained via a regular DNS query to another name server. An authoritative-only name server only returns answers to queries about domain names that have been specifically configured by the administrator.

In other words, an authoritative name server lets recursive name servers know what DNS data (the IPv4 IP, the IPv6 IP, a list of incoming mail servers, etc.) a given host name (such as "www.example.com") has. As just one example, the authoritative name server for "example.com" tells recursive name servers that "www.example.com" has the IPv4 IP address 192.0.43.10.

An authoritative name server can either be a master server or a slave server. A master server is a server that stores the original (master) copies of all zone records. A slave server uses an automatic updating mechanism of the DNS protocol in communication with its master to maintain an identical copy of the master records.

A set of authoritative name servers has to be assigned for every DNS zone. An NS record about addresses of that set must be stored in the parent zone and servers themselves (as self-reference).

When domain names are registered with a domain name registrar, their installation at the domain registry of a top level domain requires the assignment of a primary name server and at least one secondary name server. The requirement of multiple name servers aims to make the domain still functional even if one name server becomes inaccessible or inoperable.[18] The designation of a primary name server is solely determined by the priority given to the domain name registrar. For this purpose, generally only the fully qualified domain name of the name server is required, unless the servers are contained in the registered domain, in which case the corresponding IP address is needed as well.

Primary name servers are often master name servers, while secondary name servers may be implemented as slave servers.

An authoritative server indicates its status of supplying definitive answers, deemed authoritative, by setting a software flag (a protocol structure bit), called the Authoritative Answer (AA) bit in its responses.[3] This flag is usually reproduced prominently in the output of DNS administration query tools (such as dig) to indicate that the responding name server is an authority for the domain name in question.[3]

SMTP:-

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

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

Client software uses Internet Message Access Protocol (IMAP) or POP protocols to receive emails.

Simple Mail Transfer Protocol (SMTP) is an Internet standard for electronic mail (email) transmission. First defined by RFC 821 in 1982, it was last updated in 2008 with the Extended SMTP additions by RFC 5321—which is the protocol in widespread use today.

SMTP by default uses TCP port 25. The protocol for mail submission is the same, but uses port 587. SMTP connections secured by SSL, known as SMTPS, default to port 465 (nonstandard, but sometimes used for legacy reasons).

Although electronic mail servers and other mail transfer agents use SMTP to send and receive mail messages, user-level client mail applications typically use SMTP only for sending messages to a mail server for relaying. For retrieving messages, client applications usually use either POP3or IMAP.

Although proprietary systems (such as Microsoft Exchange and IBM Notes) and webmail systems (such as Outlook.com, Gmail and Yahoo! Mail) use their own non-standard protocols to access mail box accounts on their own mail servers, all use SMTP when sending or receiving email from outside their own systems.

Mail processing model:-

Email is submitted by a mail client (MUA, mail user agent) to a mail server (MSA, mail submission agent) using SMTP on TCP port 587. Most mailbox providers still allow submission on traditional port 25. From there, the MSA delivers the mail to its mail transfer agent (MTA,mail transfer agent). Often, these two agents are just different instances of the same software launched with different options on the same machine. Local processing can be done either on a single machine, or split among various appliances; in the former case, involved processes can share files; in the latter case, SMTP is used to transfer the message internally, with each host configured to use the next appliance as a smart host. Each process is an MTA in its own right; that is, an SMTP server.

The boundary MTA has to locate the target host. It uses the Domain name system (DNS) to look up the mail exchanger record (MX record) for the recipient's domain (the part of the email address on the right of @). The returned MX record contains the name of the target host. The MTA next connects to the exchange server as an SMTP client. (The article on MX record discusses many factors in determining which server the sending MTA connects to.)

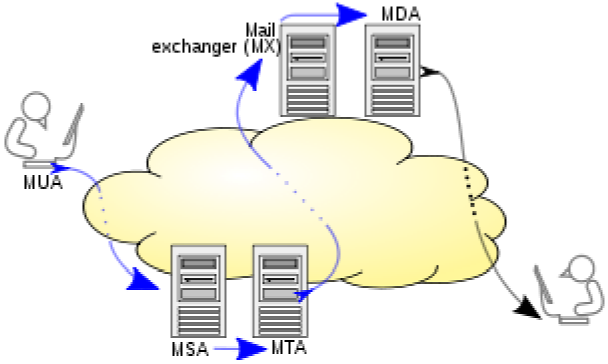
Message transfer can occur in a single connection between two MTAs, or in a series of hops through intermediary systems. A receiving SMTP server may be the ultimate destination, an intermediate "relay" (that is, it stores and forwards the message) or a "gateway" (that is, it may forward the message further using some protocol other than

SMTP). Each hop implies a formal handoff of responsibility for the message, whereby the receiving server must either deliver the message or properly report the failure to do so.[14]

Once the final hop accepts the incoming message, it hands it to a mail delivery agent (MDA) for local mail delivery. An MDA is able to save messages in the relevant mailbox format. Again, mail reception can be done using many computers or just one —the picture displays two nearby boxes in either case. An MDA may deliver messages directly to storage, or forward them over a network using SMTP, or any other means, including the Local Mail Transfer Protocol (LMTP), a derivative of SMTP designed for this purpose.

Once delivered to the local mail server, the mail is stored for batch retrieval by authenticated mail clients (MUAs). Mail is retrieved by end-user applications, called email clients, using Internet Message Access Protocol (IMAP), a protocol that both facilitates access to mail and manages stored mail, or the Post Office Protocol (POP) which typically uses the traditional mbox mail file format or a proprietary system such as Microsoft Exchange/Outlook or Lotus Notes/Domino. Webmail clients may use either method, but the retrieval protocol is often not a formal standard.

SMTP defines message transport, not the message content. Thus, it defines the mail envelope and its parameters, such as the envelope sender, but not the header (except trace information) nor the body of the message itself. STD 10 and RFC 5321 define SMTP (the envelope), while STD 11 and RFC 5322 define the message (header and body), formally referred to as the Internet Message Format.



Protocol overview:-

SMTP is a connection-oriented, text-based protocol in which a mail sender communicates with a mail receiver by issuing command strings and supplying necessary data over a reliable ordered data stream channel, typically a Transmission Control Protocol (TCP) connection. An SMTP session consists of commands originated by an SMTP client (the initiating agent, sender, or transmitter) and corresponding responses from the SMTP server (the listening agent, or receiver) so that the session is opened, and session parameters are exchanged. A session may include zero or more SMTP transactions. An SMTP transaction consists of three command/reply sequences (see example below.) They are:

1. MAIL command, to establish the return address, a.k.a. Return-Path,[15] reverse-path,[16] bounce address, m from, or envelope sender.
2. RCPT command, to establish a recipient of this message. This command can be issued multiple times, one for each recipient. These addresses are also part of the envelope.
3. DATA to signal the beginning of the message text; the content of the message, as opposed to its envelope. It consists of a message header and a message body separated by an empty line. DATA is actually a group of commands, and the server replies twice: once to the DATA command proper, to acknowledge that it is ready to receive the text, and the second time after the end-of-data sequence, to either accept or reject the entire message.

Besides the intermediate reply for DATA, each server's reply can be either positive (2xx reply codes) or negative. Negative replies can be permanent (5xx codes) or transient (4xx codes). A **reject** is a permanent failure by an SMTP server; in this case the SMTP client should send a bounce message. A **drop** is a positive response followed by message discard rather than delivery.

The initiating host, the SMTP client, can be either an end-user's email client, functionally identified as a mail user agent (MUA), or a relay server's mail transfer agent (MTA), that is an SMTP server acting as an SMTP client, in the relevant session, in order to relay mail. Fully capable SMTP servers maintain queues of messages for retrying message transmissions that resulted in transient failures.

A MUA knows the outgoing mail SMTP server from its configuration. An SMTP server acting as client, i.e. relaying, typically determines which SMTP server to connect to by looking up the MX (Mail eXchange) DNS resource record for each recipient's domain name. Conformant MTAs (not all) fall back to a simple A record in case no MX record can be found. Relaying servers can also be configured to use a smart host.

An SMTP server acting as client initiates a TCP connection to the server on the "well-known port" designated for SMTP: port 25. MUAs should use port 587 to connect to an MSA. The main difference between an MTA and an MSA is that SMTP Authentication is mandatory for the latter only.

Smtp transport example:-

Typical example of sending a message via SMTP to two mailboxes (alice and theboss) located in the same mail domain (example.com or localhost.com) is reproduced in the following session exchange. (In this example, the conversation parts are prefixed with S: and C:, for server and client, respectively; these labels are not part of the exchange.)

After the message sender (SMTP client) establishes a reliable communications channel to the message receiver (SMTP server), the session is opened with a greeting by the server, usually containing its fully qualified domain name (FQDN), in this case smtp.example.com. The client initiates its dialog by responding with a HELO command identifying itself in the command's parameter with its FQDN (or an address literal if none is available

```
S: 220 smtp.example.com ESMTP Postfix
C: HELO relay.example.org
S: 250 Hello relay.example.org, I am glad to meet you
C: MAIL FROM:<bob@example.org>
S: 250 Ok
C: RCPT TO:<alice@example.com>
S: 250 Ok
C: RCPT TO:<theboss@example.com>
S: 250 Ok
C: DATA
S: 354 End data with <CR><LF>.<CR><LF>
C: From: "Bob Example" <bob@example.org>
C: To: "Alice Example" <alice@example.com>
C: Cc: theboss@example.com
C: Date: Tue, 15 January 2008 16:02:43 -0500
C: Subject: Test message
C:
C: Hello Alice.
C: This is a test message with 5 header fields and 4 lines in the message body.
C: Your friend,
C: Bob
C: .
S: 250 Ok: queued as 12345
C: QUIT
S: 221 Bye
{The server closes the connection}
```

The client notifies the receiver of the originating email address of the message in a MAIL FROM command. In this example, the email message is sent to two mailboxes on the same SMTP server: one for each recipient listed in the To and Cc header fields. The corresponding SMTP command is RCPT TO. Each successful reception and execution of a command is acknowledged by the server with a result code and response message (e.g., 250 Ok).

The transmission of the body of the mail message is initiated with a DATA command after which it is transmitted verbatim line by line and is terminated with an end-of-data sequence. This sequence consists of a new-line (<CR><LF>), a single full stop (period), followed by another new-line. Since a message body can contain a line with just a period as part of the text, the client sends two periods every time a line starts with a period; correspondingly, the server replaces every sequence of two periods at the beginning of a line with a single one. Such escaping method is called dot-stuffing.

The server's positive reply to the end-of-data, as exemplified, implies that the server has taken the responsibility of delivering the message. A message can be doubled if there is a communication failure at this time, e.g. due to a power shortage: Until the sender has received that 250 reply, it must assume the message was not delivered. On the other hand, after the receiver has decided to accept the message, it must assume the message has been delivered to it. Thus, during this time span, both agents have active copies of the message that they will try to deliver.[22] The probability that a communication failure occurs exactly at this step is directly proportional to the amount of filtering that the server performs on the message body, most often for anti-spam purposes. The limiting timeout is specified to be 10 minutes.[23]

The QUIT command ends the session. If the email has other recipients located elsewhere, the client would QUIT and connect to an appropriate SMTP server for subsequent recipients after the current destination(s) had been queued. The information that the client sends in the HELO and MAIL FROM commands are added (not seen in example code) as additional header fields to the message by the receiving server. It adds a Received and Return-Path header field, respectively.

Some clients are implemented to close the connection after the message is accepted (250 Ok: queued as 12345), so the last two lines may actually be omitted. This causes an error on the server when trying to send the 221 reply.

File Transfer Protocol (FTP):-

The **File Transfer Protocol** (**FTP**) is a standard network protocol used to transfer computer files between a client and server on a computer network.

FTP is built on a client-server model architecture and uses separate control and data connections between the client and the server.[1] FTP users may authenticate themselves with a clear-text sign-in protocol, normally in the form of a username and password, but can connect anonymously if the server is configured to allow it. For secure transmission that protects the username and password, and encrypts the content, FTP is often secured with SSL/TLS (FTPS). SSH File Transfer Protocol (SFTP) is sometimes also used instead, but is technologically different.

The first FTP client applications were command-line programs developed before operating systems had graphical user interfaces, and are still shipped with most Windows, Unix, and Linux operating systems.[2][3] Many FTP clients and automation utilities have since been developed for desktops, servers, mobile devices, and hardware, and FTP has been incorporated into productivity applications, such as Web page editors.

Protocol overview:-

Communication & Data Transfer:-

FTP may run in active or passive mode, which determines how the data connection is established.[5] In both cases, the client creates a TCP control connection from a random, usually an unprivileged, port N to the FTP server command port 21.

In active mode, the client starts listening for incoming data connections from the server on port M. It sends the FTP command PORT M to inform the server on which port it is listening. By default, M=N. The server then initiates a data channel to the client from its port 20, the FTP server data port.

In situations where the client is behind a firewall and unable to accept incoming TCP connections, passive mode may be used. In this mode, the client uses the control connection to send a PASV command to the server and then receives a server IP address and server port number from the server,[5][6] which the client then uses to open a data connection from an arbitrary client port to the server IP address and server port number received.[7]

Both modes were updated in September 1998 to support IPv6. Further changes were introduced to the passive mode at that time, updating it to extended passive mode.[8]

The server responds over the control connection with three-digit status codes in ASCII with an optional text message. For example, "200" (or "200 OK") means that the last command was successful. The numbers represent the code for the response and the optional text represents a human-readable explanation or request (e.g. <Need account for storing file>).[1] An ongoing transfer of file data over the data connection can be aborted using an interrupt message sent over the control connection.

While transferring data over the network, four data representations can be used:[2][3][4]

ASCII mode: Used for text. Data is converted, if needed, from the sending host's character representation to "8-bit ASCII" before transmission, and (again, if necessary) to the receiving host's character representation. As a consequence, this mode is inappropriate for files that contain data other than plain text.

Image mode (commonly called Binary mode): The sending machine sends each file byte for byte, and the recipient stores the bytestream as it receives it. (Image mode support has been recommended for all implementations of FTP).

EBCDIC mode: Used for plain text between hosts using the EBCDIC character set.

Local mode: Allows two computers with identical setups to send data in a proprietary format without the need to convert it to ASCII.

For text files, different format control and record structure options are provided. These features were designed to facilitate files containing Telnet or ASA.

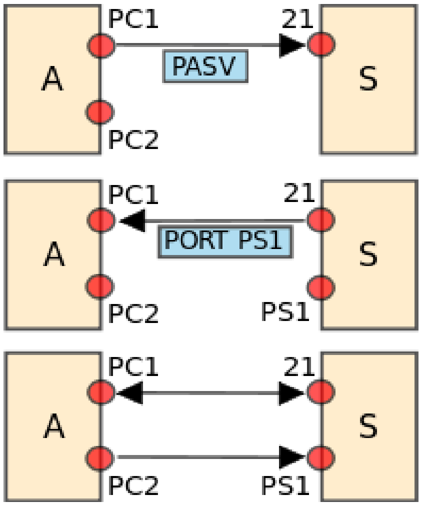
Data transfer can be done in any of three modes:[1][2]

Stream mode: Data is sent as a continuous stream, relieving FTP from doing any processing. Rather, all processing is left up to TCP. No End-of-file indicator is needed, unless the data is divided into records.

Block mode: FTP breaks the data into several blocks (block header, byte count, and data field) and then passes it on to TCP.[4]

Compressed mode: Data is compressed using a simple algorithm (usually run-length encoding).

Some FTP software also implements a DEFLATE-based compressed mode, sometimes called "Mode Z" after the command that enables it. This mode was described in an Internet Draft, but not standardized.[9]

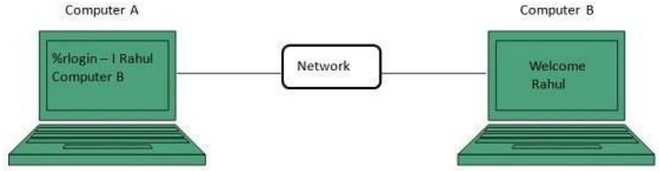


Login:-

FTP login utilizes a normal username and password scheme for granting access.[2] The username is sent to the server using the USER command, and the password is sent using the PASS command.[2] If the information provided by the client is accepted by the server, the server will send a greeting to the client and the session will commence.[2] If the server supports it, users may log in without providing login credentials, but the same server may authorize only limited access for such sessions.[2]

TELNET:-

Telnet is a protocol used to log in to remote computer on the internet. There are a number of Telnet clients having user friendly user interface. The following diagram shows a person is logged in to computer A, and from there, he remote logged into computer B.



Telnet is an application layer protocol used on the Internet or local area networks to provide a bidirectional interactive text-oriented communication facility using a virtual terminal connection. User data is interspersed in-band with Telnet control information in an 8-bit byte oriented data connection over the Transmission Control Protocol (TCP).

Telnet was developed in 1969 beginning with RFC 15, extended in RFC 854, and standardized as Internet Engineering Task Force (IETF) Internet Standard STD 8, one of the first Internet standards.

Historically, Telnet provided access to a command-line interface (usually, of an operating system) on a remote host, including most network equipment and operating systems with a configuration utility (including systems based on Windows NT). However, because of serious security concerns when using Telnet over an open network such as the Internet, its use for this purpose has waned significantly in favor of SSH.

The term telnet is also used to refer to the software that implements the client part of the protocol. Telnet client applications are available for virtually all computer platforms. Telnet is also used as a verb. To telnet means to establish a connection with the Telnet protocol, either with command line client or with a programmatic interface. For example, a common directive might be: "To change your password, telnet to the server, log in and run the passwd command." Most often, a user will be telnetting to a Unix-like server system or a network device (such as a router) and obtaining a login prompt to a command line text interface or a character-based full-screen manager.

Security:-

When Telnet was initially developed in 1969, most users of networked computers were in the computer departments of academic institutions, or at large private and government research facilities. In this environment, security was not nearly as much a concern as it became after the bandwidth explosion of the 1990s. The rise in the number of people with access to the Internet, and by extension the number of people attempting to hack other people's servers, made encrypted alternatives necessary.

Experts in computer security, such as SANS Institute, recommend that the use of Telnet for remote logins should be discontinued under all normal circumstances, for the following reasons:

Telnet, by default, does not encrypt any data sent over the connection (including passwords), and so it is often feasible to eavesdrop on the communications and use the password later for malicious purposes; anybody who has access to a router, switch, hub or gateway located on the network between the two hosts where Telnet is being used can intercept the packets passing by and obtain login, password and whatever else is typed with a packet analyzer.

Most implementations of Telnet have no authentication that would ensure communication is carried out between the two desired hosts and not intercepted in the middle.

Several vulnerabilities have been discovered over the years in commonly used Telnet daemons.

These security-related shortcomings have seen the usage of the Telnet protocol drop rapidly. especially on the public Internet, in favor of the Secure Shell (SSH) protocol, first released in 1995. SSH provides much of the functionality of telnet, with the addition of strong encryption to prevent sensitive data such as passwords from being intercepted, and public keyauthentication, to ensure that the remote computer is actually who it claims to be. As has happened with other early Internet protocols, extensions to the Telnet protocol provide Transport Layer Security (TLS) security and Simple Authentication and Security Layer (SASL) authentication that address the above concerns. However, most Telnet implementations do not support these extensions; and there has been relatively little interest in implementing these as SSH is adequate for most purposes.

It is of note that there are a large number of industrial and scientific devices which have only Telnet available as a communication option. Some are built with only a standard RS-232 port and use a serial server hardware appliance to provide the translation between the TCP/Telnet data and the RS-232 serial data. In such cases, SSH is not an option unless the interface appliance can be configured for SSH.

Conclusion:-

The lower layers of TCP/IP protocol suit are primarily concerned with formatting, encapsulating and transmitting data across the network. These layers are closely associated with the underlying network hardware and network infrastructure devices.

The topmost layer, Application Layer of TCP/IP protocol suit suit is concerned mainly with human interaction and the implementation of software applications and related protocols.

TCP/IP protocol suit is included with a large number of applications and application protocols. Using these applications and application protocols, data can be moved between hosts, and remote users can communicate.

The Application layer provides applications the ability to access the services of the other layers and defines the protocols that applications use to exchange data. There are many Application layer protocols and new protocols are always being developed.

The most widely-known Application layer protocols are those used for the exchange of user information:

- The Hypertext Transfer Protocol (HTTP) is used to transfer files that make up the Web pages of the World Wide Web.
- The File Transfer Protocol (FTP) is used for interactive file transfer.
- The Simple Mail Transfer Protocol (SMTP) is used for the transfer of mail messages and attachments.
- Telnet, a terminal emulation protocol, is used for logging on remotely to network hosts.

- Additionally, the following Application layer protocols help facilitate the use and management of TCP/IP networks:
- The Domain Name System (DNS) is used to resolve a host name to an IP address.
- The Routing Information Protocol (RIP) is a routing protocol that routers use to exchange routing information on an IP internetwork.
- The Simple Network Management Protocol (SNMP) is used between a network management console and network devices (routers, bridges, intelligent hubs) to collect and exchange network management information.

References:-

1. http://inetcore.com/project/ipv4ec/index_en.html.
2. http://www.omnisecu.com/tcpip/ipv6/differences-between-ipv4-and- ipv6.php.
3. "IPv6 Headers", Online: http://www.cu.ipv6tf.org/literatura/chap3.pdf, chapter 3, pp. 40-55, Des 12 1997.
4. T. Dunn, “The IPv6 Transition,” IEEE Internet Computing, Vol.6, No.3, May/June 2002, pp.11-13
5. IPv6 users’ site: http://www.ipv6.org.
6. http://www.juniper.net/techpubs/en_US/junose14.2/information-products/topic-collections/swconfig-ip ipv6/index.html? topic-64529.html.
7. http://ipv6security.wikia.com/wiki/Ipv6_header
8. IETF IPv6 Transition Working Group, http://www.6bone.net/ngtrans.
9. http://en.wikipedia.org.
10. http://www.cybertelecom.org/dns/ipv6_transition.htm.
11. RFC 4213, Basic Transition Mechanisms for IPv6 Hosts and Routers.
12. http://www.gao.gov/new.items/d05471.pdf.
13. RFC 3513: Internet Protocol Version 6 (IPv6) Addressing Architecture .
14. RFC 2893: Transition Mechanisms for IPv6 Hosts and Routers.
15. RFC 3596: DNS Extensions to Support IP Version 6 .
16. www.linecity.de/INFOTECH_ACS_SS04/acs4_top_4.pdf.
17. Ali, AmerNizar Abu. "Comparison study between IPV4 & IPV6." (2012).
18. Dutta, Chiranjit, and Ranjeet Singh. "Sustainable IPv4 to IPv6 Transition."International Journal 2.10 (2012).
19. Doshi, Jinesh, et al. "A Comparative Study of IPv4/IPv6 Co-existence Technologies."