Simple Mail Transfer Protocol (SMTP)

Email is emerging as one of the most valuable services on the internet today. Most internet systems use SMTP as a method to transfer mail from one user to another. SMTP is a push protocol and is used to send the mail whereas POP (post office protocol) or IMAP (internet message access protocol) are used to retrieve those emails at the receiver's side.

Fundamentals

SMTP is an application layer protocol. The client who wants to send the mail opens a TCP connection to the SMTP server and then sends the mail across the connection. The SMTP server is an always-on listening mode. As soon as it listens for a TCP connection from any client, the SMTP process initiates a connection through port 25. After successfully establishing a TCP connection the client process sends the mail instantly.

SMTP Protocol

The SMTP model is of two types:

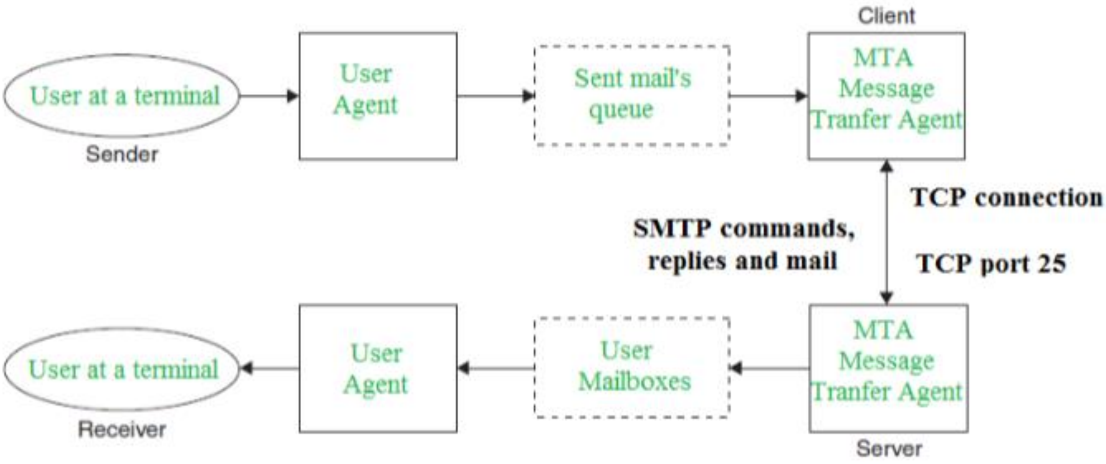
1. End-to-end method
2. Store-and- forward method

The end-to-end model is used to communicate between different organizations whereas the store and forward method is used within an organization. An SMTP client who wants to send the mail will contact the destination's host SMTP directly, in order to send the mail to the destination. The SMTP server will keep the mail to itself until it is successfully copied to the receiver's SMTP.

The client SMTP is the one that initiates the session so let us call it client- SMTP and the server SMTP is the one that responds to the session request so let us call it receiver-SMTP. The client-SMTP will start the session and the receiver-SMTP will respond to the request.

Model of SMTP system

In the SMTP model user deals with the user agent (UA), for example, Microsoft Outlook, Netscape, Mozilla, etc. In order to exchange the mail using TCP, MTA is used. The user sending the mail doesn't have to deal with MTA as it is the responsibility of the system admin to set up a local MTA. The MTA maintains a small queue of mails so that it can schedule repeat delivery of mails in case the receiver is not available. The MTA delivers the mail to the mailboxes and the information can later be downloaded by the user agents.



Both the SMTP-client and SMTP-server should have 2 components:

1. User-agent (UA)
2. Local MTA

Communication between sender and the receiver:

The sender's user agent prepares the message and sends it to the MTA. The MTA's responsibility is to transfer the mail across the network to the receiver's MTA. To send mails, a system must have a client MTA, and to receive mails, a system must have a server MTA.

SENDING EMAIL:

Mail is sent by a series of request and response messages between the client and the server. The message which is sent across consists of a header and a body. A null line is used to terminate the mail header and everything after the null line is considered as the body of the message, which is a sequence of ASCII characters. The message body contains the actual information read by the receipt.

RECEIVING EMAIL:

The user agent at the server-side checks the mailboxes at a particular time of intervals. If any information is received, it informs the user about the mail. When the user tries to read the mail it displays a list of emails with a short description of each mail in the mailbox. By selecting any of the mail users can view its contents on the terminal.

Some SMTP Commands:

- HELO – Identifies the client to the server, fully qualified domain name, only sent once per session
- MAIL – Initiate a message transfer, fully qualified domain of originator
- RCPT – Follows MAIL, identifies an addressee, typically the fully qualified name of the addressee, and for multiple addressees use one RCPT for each addressee

- DATA – send data line by line

What is DNS?

The domain name system (DNS) is a naming database in which internet domain names are located and translated into Internet Protocol (IP) addresses. The domain name system maps the name people use to locate a website to the IP address that a computer uses to locate that website.

For example, if someone types "example.com" into a web browser, a server behind the scenes maps that name to the corresponding IP address. An IP address is similar in structure to 203.0.113.72.

Web browsing and most other internet activities rely on DNS to quickly provide the information necessary to connect users to remote hosts. DNS mapping is distributed throughout the internet in a hierarchy of authority. Access providers and enterprises, as well as governments, universities and other organizations, typically have their own assigned ranges of IP addresses and an assigned domain name. They also typically run DNS servers to manage the mapping of those names to those addresses. Most Uniform Resource Locators (URLs) are built around the domain name of the web server that takes client requests.

How DNS works

DNS servers convert URLs and domain names into IP addresses that computers can understand and use. They translate what a user types into a browser into something the machine can use to find a webpage. This process of translation and lookup is called *DNS resolution*.

1. The basic process of a DNS resolution follows these steps:
2. The user enters a web address or domain name into a browser.
3. The browser sends a message, called a *recursive DNS query*, to the network to find out which IP or network address the domain corresponds to.
4. The query goes to a recursive DNS server, which is also called a *recursive resolver*, and is usually managed by the internet service provider (ISP). If the recursive resolver has the address, it will return the address to the user, and the webpage will load.
5. If the recursive DNS server does not have an answer, it will query a series of other servers in the following order: DNS root name servers, top-level domain (TLD) name servers and authoritative name servers.
6. The three server types work together and continue redirecting until they retrieve a DNS record that contains the queried IP address. It sends this information to the recursive DNS server, and the webpage the user is looking for loads. DNS root name servers and TLD servers primarily redirect queries and rarely provide the resolution themselves.
7. The recursive server stores, or caches, the A record for the domain name, which contains the IP address. The next time it receives a request for that domain name, it can respond directly to the user instead of querying other servers.
8. If the query reaches the authoritative server and it cannot find the information, it returns an error message.

The entire process querying the various servers takes a fraction of a second and is usually imperceptible to the user.

DNS servers answer questions from both inside and outside their own domains. When a server receives a request from outside the domain for information about a name or address inside the domain, it provides the authoritative answer.

When a server gets a request from within its domain for a name or address outside that domain, it forwards the request to another server, usually one managed by its ISP.

HTTP

The Hypertext Transfer Protocol (HTTP) is an application-level protocol for distributed, collaborative, hypermedia information systems. This is the foundation for data communication for the World Wide Web (i.e. internet) since 1990. HTTP is a generic and stateless protocol which can be used for other purposes as well using extensions of its request methods, error codes, and headers.

Basically, HTTP is a TCP/IP based communication protocol, that is used to deliver data (HTML files, image files, query results, etc.) on the World Wide Web. The default port is TCP 80, but other ports can be used as well. It provides a standardized way for computers to communicate with each other. HTTP specification specifies how clients' request data will be constructed and sent to the server, and how the servers respond to these requests.

Basic Features

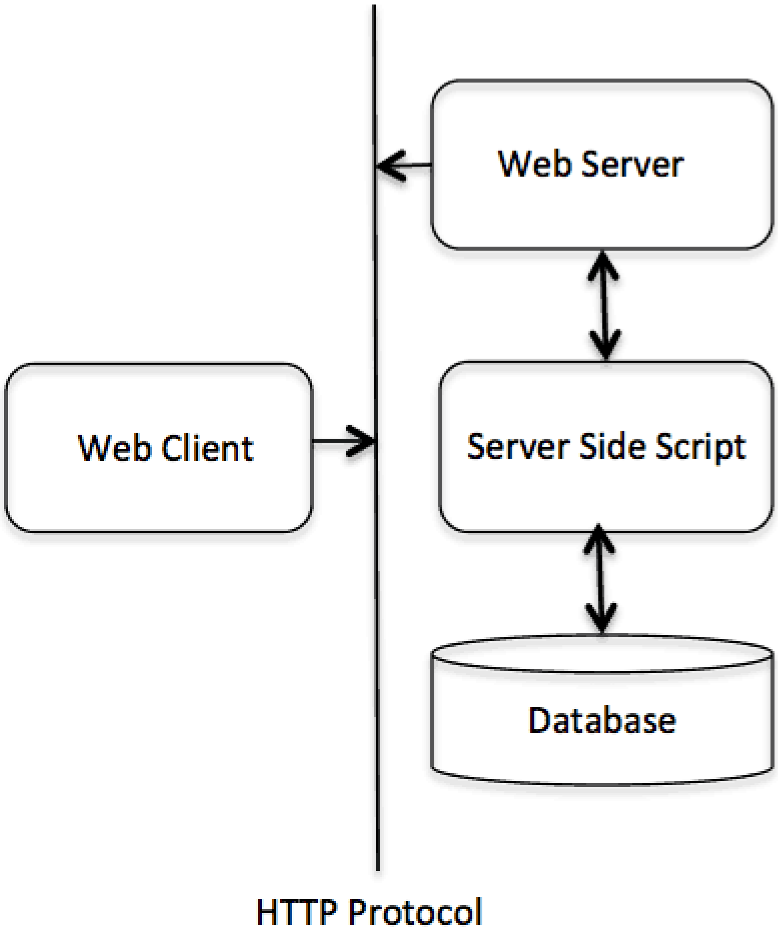
There are three basic features that make HTTP a simple but powerful protocol:

- **HTTP is connectionless:** The HTTP client, i.e., a browser initiates an HTTP request and after a request is made, the client waits for the response. The server processes the request and sends a response back after which client disconnect the connection. So client and server knows about each other during current request and response only. Further requests are made on new connection like client and server are new to each other.
- **HTTP is media independent:** It means, any type of data can be sent by HTTP as long as both the client and the server know how to handle the data content. It is required for the client as well as the server to specify the content type using appropriate MIME-type.
- **HTTP is stateless:** As mentioned above, HTTP is connectionless and it is a direct result of HTTP being a stateless protocol. The server and client are aware of each other only during a current request. Afterwards, both of them forget about each other. Due to this nature of the protocol, neither the client nor the browser can retain information between different requests across the web pages.

HTTP/1.0 uses a new connection for each request/response exchange, where as HTTP/1.1 connection may be used for one or more request/response exchanges.

Basic Architecture

The following diagram shows a very basic architecture of a web application and depicts where HTTP sits:



The HTTP protocol is a request/response protocol based on the client/server based architecture where web browsers, robots and search engines, etc. act like HTTP clients, and the Web server acts as a server.

Client

The HTTP client sends a request to the server in the form of a request method, URI, and protocol version, followed by a MIME-like message containing request modifiers, client information, and possible body content over a TCP/IP connection.

Server

The HTTP server responds with a status line, including the message's protocol version and a success or error code, followed by a MIME-like message containing server information, entity meta information, and possible entity-body content.