SMTP(Simple Mail Transfer Protocol)

Name	C. Yeswanth
Pin NO:	19005A0505

History

In the earlier days of the Internet E-mail was sent and received through the protocol MTP. MTP describes a set of commands and procedures by which two devices can connect using TCP to exchange e-mail messages. Its operation is described largely using elements borrowed from two early TCP/IP application protocols that were already in use at that time:. There was nothing wrong with basing e-mail on something like FTP, but using these services to render another service made mail transfer something like a hack, something that was just extended from another protocol without fully defining it.

Sending mails this way also restricted the functionality of E-mails. As FTP and Telnet were file transfer protocols it was not possible to include features in the protocol that were specific to sending and receiving mail.

Due to the importance of e-mail, a specific protocol designed for the purpose of delivering e-mail was warranted. This protocol was first defined in RFC 788, published in November 1981: the Simple Mail Transfer Protocol (SMTP)

SMTP was made for elegance, it was specially designed for the transfer of mail, it is an independent protocol that ran over TCP/IP.

Since then development of both the E-mail messages and the SMTP protocol continued. And in August 1982, a mile stone in TCP/IP e-mail was achieved when RFC's 821 and 822 published.

> RFC 821 became the defining standard for SMTP

Fundamentals

It is an application layer protocol that is used for transferring e-mails on the internet. It is used as the common mechanism for transporting electronic mail among different hosts within the transmission control protocol/Internet protocol (TCP/IP) suite. Under SMTP a client SMTP process opens a TCP connection to a server SMTP process on a remote host and attempts to send mail across the connection. The server SMTP listens for a TCP connection on a specific port and the client SMTP process initiates a connection on that port.

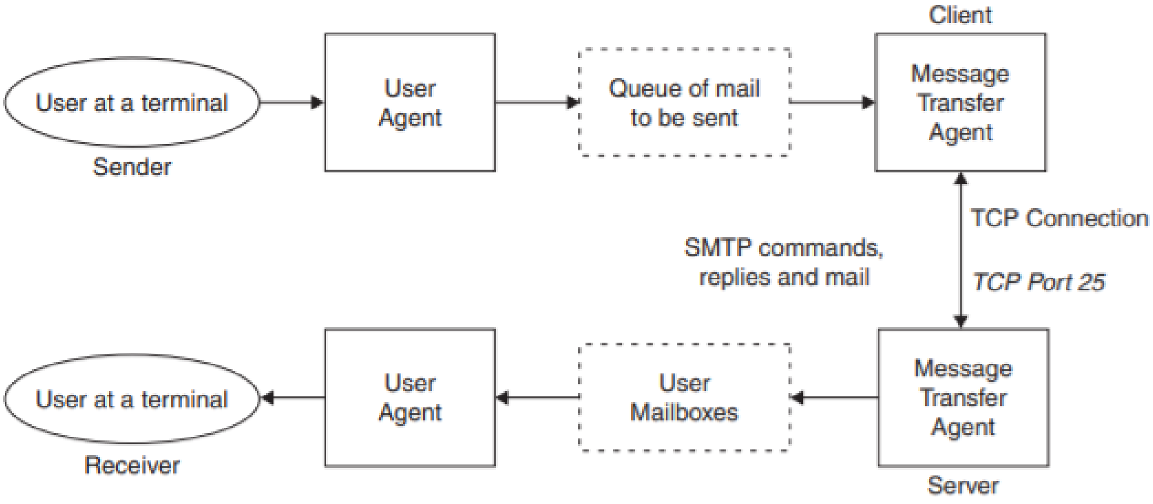
> The port used by SMTP is Port 25

When the TCP connection is successful, the two processes execute a simple request-response dialogue, defined by the SMTP protocol, in which the client process transmits the mail address of the originator and the recipient for a message. When the server process accepts this mail addresses the client process transmits the e-mail instant message. This mail must contain a message body and the message text is formatted according to the RFC 822.

Mail that arrives via SMTP is forwarded to a remote server, or it is delivered to mailboxes on the local server. POP3 or IMAP allow users download mail that is stored on the local server. The SMTP model (RFC 821) supports both end-to-end (no intermediate message transfer agents [MTAs]) and store-and-forward mail delivery methods.

The end-to-end method is used between organizations, and the store-and forward method is chosen for operating within organizations that have TCP/IP and SMTP-based networks.
A SMTP client will contact the destination host's SMTP server directly to deliver the mail. It will keep the mail item from being transmitted until it has been successfully copied to the recipient's SMTP. This is different from the store-and-forward principle that is common in many other electronic mailing systems, where the mail item may pass through a number of intermediate hosts in the same network on its way to the destination and where successful transmission from the sender only indicates that the mail item has reached the first intermediate hop .

Generally the complete process in the RFC 821 defines the SMTP as the client server protocol, where the client is the one who initiates the session and the server is the one that responds to the session request. It is also often referred to as the sender receiver protocol, where the sender sends a mail via a SMTP and the receiver receives the receiver SMTP.



Basic SMTP Model

An SMTP-based process can transfer electronic mail to another process on the same network or to another network via a relay or gateway process accessible to both networks.

An e-mail message may pass through a number of intermediate relay or gateway hosts on its path from a sender to a recipient. The receiver SMTP could either be the actual destination or an intermediate destination, known as the mail gateway. The sender SMTP will generate commands which are replied to by the receiver SMTP.

Uses Of SMTP(User Agents)

Any protocol's reason to be created is to be used. What use will the protocols be of if they aren't used what is the purpose of them. As we have seen the basic purpose of SMTP is sending and receiving E-mails.
The applications that make use of the SMTP are called the **User Agents**

Some of the most famous SMTP user agents are

- Gmail
- Microsoft Outlook
- Yahoo

Sending e-Mail

Electronic mail is sent by a series of request–response transactions between a client and a server. An SMTP transaction consists of the **envelope** and message, which is composed of **header** (with From: and To: fields) and **body** (text after headers sent with the DATA command).

There are 3 commands that facilitate the transfer of mails via SMTP, they inform the receiver SMTP of the semantics of this transaction

1. **Send** <SP> FROM: <reverse-path> <CRLF>
   The Send command is used to deliver the mail to the users terminal as long as he is active. If the user is not active it relays with a 450 error message. The mail transaction is successful if the message is delivered to the terminal.
2. **SOML** <SP> FROM: <reverse-path> <CRLF>
   The Send or Mail command is used to deliver the mail to the users terminal if the user is active , if he is not active it will be delivered to his inbox. The transaction is successful if the mail reaches either the user's terminal or their inbox. The mail here is transferred to either the users mailbox or his terminal but not both.
3. **SAML** <SP> FROM: <reverse-path> <CRLF>
   The Send and Mail command is used to deliver the mail to the users inbox as well as his terminal if he is active. The transaction is deemed successful it the mail is delivered to the mail box.

Receiving e-Mail.

The UA periodically checks the content of the mailboxes. It informs the user about mail arrival by giving a special notice. When the user tries to read the mail, a list of arrived mail packages is displayed. Each line of the list contains a brief summary of the information about a particular package in the mailbox. The summary may include the sender mail address, the subject, and the time the mail was received or sent. By selecting any of the
packages, the user can view its contents on the terminal display.

Mail Header Format

The header includes a number of key words and values that define the sending date, sender's address where replies should go and some other information. The header is a list of lines of the form
field-name: field-value
Consider the following mail id: **chalayeshwanth@gmai.com**

The string "chalayeshwanth" shows the name of the recipient and the mail box owner.
@ is used to divide the name and the domain.
gmail.com, that is whatever comes after the @ symbol specifies the Domain name, that is associated to the User Agent.

The form of addressing followed is closely related to the domain name system concept. SMTP also uses the DNS to determine the IP address of the destination mailbox.

Delayed Delivery

SMTP also has the ability of a delayed delivery. The messages can be delayed at the sender site, receiver site and even at any Intermediate gateway.

In the case of delaying at the sender site, the client has to accommodate a spooling system, in which e-mail messages are stored before being sent. A message created by the user agent is delivered to the spool storage. The
client mail transfer agent periodically (usually every 10 to 30 minutes) checks the spool to find the mail that can be sent. The mail will be sent only if the receiver is ready and the IP address of the server has been obtained though DNS. If a message cannot be delivered in the timeout period (usually about 3 to 5 days), the mail returns to the sender.

SMTP Mail Transaction Flow

The SMTP protocol defines how commands and responses must be sent by Mail Transfer Agents. The client sends commands to the server and the server responds with numeric reply codes and optional human-readable strings. There are a small number of commands that can be utilized for this purpose.

The SMTP Transaction Progresses like this.

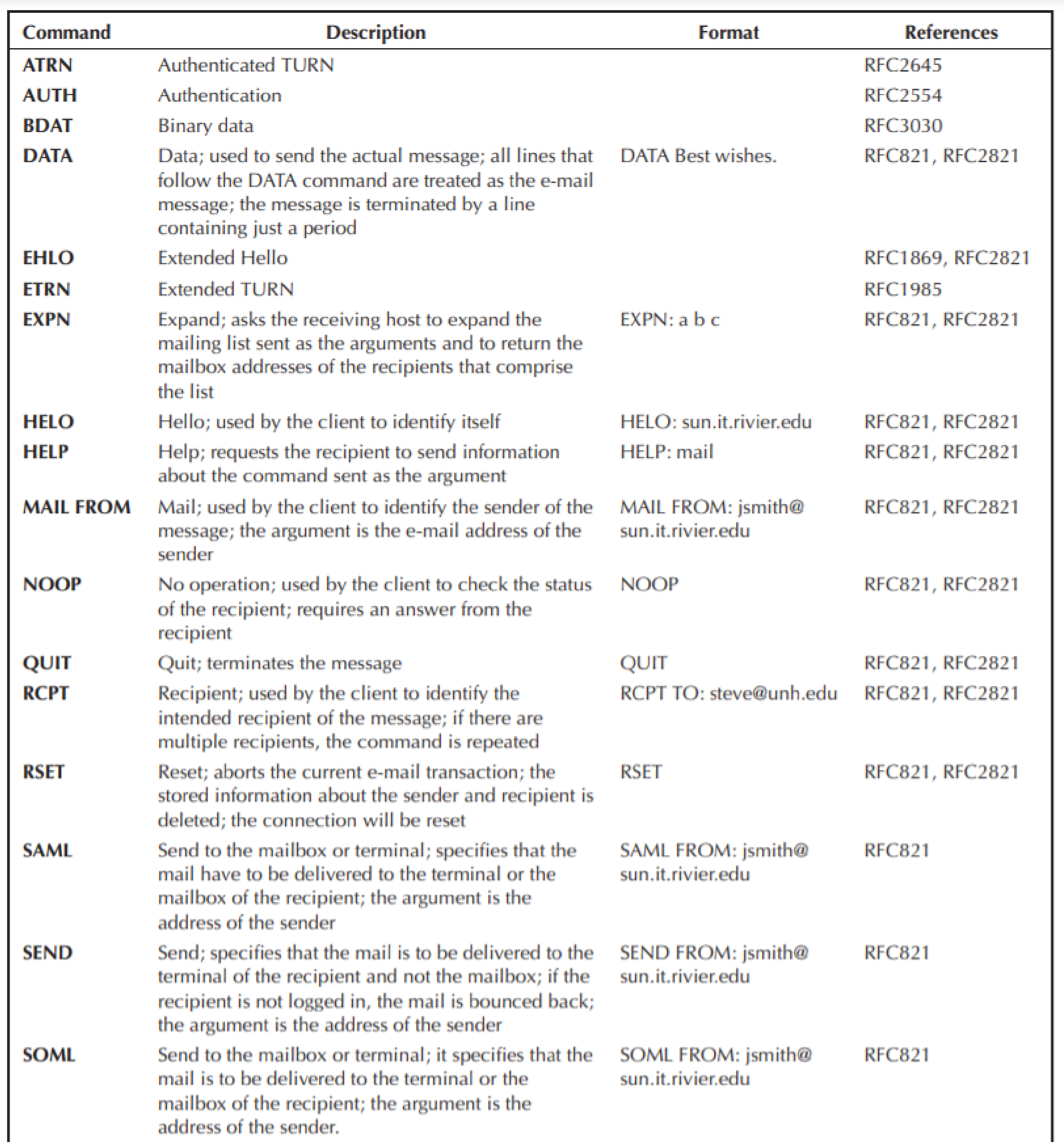
1. The client establishes a TCP connection with the destination SMTP and waits for the server to send a ready message or not available message according to it's situation.
2. The **HELO** command is sent, and the receiver is forced to identify himself by sending back its domain name. The client can use this information to verify if it contacted the right destination SMTP
3. If the receiver SMTP supports the extensions sent by the client, it will respond . If it doesn't support it will respond with a 500 syntax error command

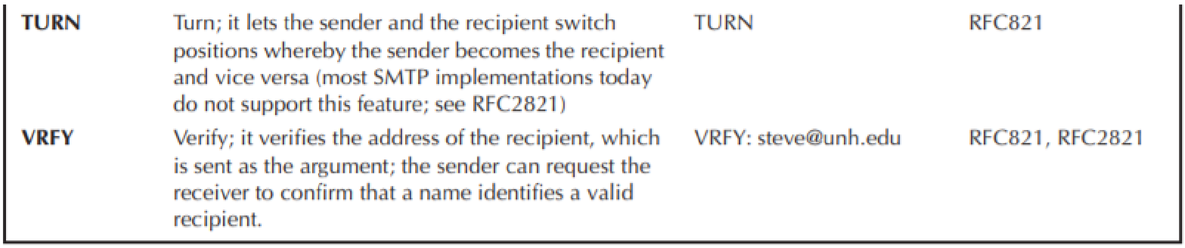
unrecognized error.

4. Once the connection is established, the client initiates the start of a mail transaction by sending a **MAIL** command to the receiver. This command contains the reverse-path, which can be used to report errors.
5. The next step is providing the server SMTP with the destinations for the message. If there is such a destination, a 250 OK message will be sent else a 500 Not Found message will be relayed.
6. Next the client issues a **DATA** command to notify the server that the message contents are following.
7. Then the client sends the data line by line, ending with the sequence <CRLF>, <CRLF>, line on which the receiver acknowledges with a 250 OK or any other relevant error.
8. Once the complete process is over, the connection is closed, if there are any other messages to be relayed, the same procedure will be followed.

SMTP Commands

There are some commands that the SMTP utilizes to send mails from the client to server, they are formed with ASCII and allow the building of mail clients on any platforms and servers. The complete list is provided below.





Command	Description	Format	References
ATRN	Authenticated TURN		RFC2645
AUTH	Authentication		RFC2554
BDAT	Binary data		RFC3030
DATA	Data; used to send the actual message; all lines that follow the DATA command are treated as the e-mail message; the message is terminated by a line containing just a period	DATA Best wishes.	RFC821, RFC2821
EHLO	Extended Hello		RFC1869, RFC2821
ETRN	Extended TURN		RFC1985
EXPN	Expand; asks the receiving host to expand the mailing list sent as the arguments and to return the mailbox addresses of the recipients that comprise the list	EXPN: a b c	RFC821, RFC2821
HELO	Hello; used by the client to identify itself	HELO: sun.it.rivier.edu	RFC821, RFC2821
HELP	Help; requests the recipient to send information about the command sent as the argument	HELP: mail	RFC821, RFC2821
MAIL FROM	Mail; used by the client to identify the sender of the message; the argument is the e-mail address of the sender	MAIL FROM: jsmith@ sun.it.rivier.edu	RFC821, RFC2821
NOOP	No operation; used by the client to check the status of the recipient; requires an answer from the recipient	NOOP	RFC821, RFC2821
QUIT	Quit; terminates the message	QUIT	RFC821, RFC2821
RCPT	Recipient; used by the client to identify the intended recipient of the message; if there are multiple recipients, the command is repeated	RCPT TO: steve@unh.edu	RFC821, RFC2821
RSET	Reset; aborts the current e-mail transaction; the stored information about the sender and recipient is deleted; the connection will be reset	RSET	RFC821, RFC2821
SAML	Send to the mailbox or terminal; specifies that the mail have to be delivered to the terminal or the mailbox of the recipient; the argument is the address of the sender	SAML FROM: jsmith@ sun.it.rivier.edu	RFC821
SEND	Send; specifies that the mail is to be delivered to the terminal of the recipient and not the mailbox; if the recipient is not logged in, the mail is bounced back; the argument is the address of the sender	SEND FROM: jsmith@ sun.it.rivier.edu	RFC821
SOML	Send to the mailbox or terminal; it specifies that the mail is to be delivered to the terminal or the mailbox of the recipient; the argument is the address of the sender.	SOML FROM: jsmith@ sun.it.rivier.edu	RFC821
TURN	Turn; it lets the sender and the recipient switch positions whereby the sender becomes the recipient and vice versa (most SMTP implementations today do not support this feature; see RFC2821)	TURN	RFC821
VRFY	Verify; it verifies the address of the recipient, which is sent as the argument; the sender can request the receiver to confirm that a name identifies a valid recipient.	VRFY: steve@unh.edu	RFC821, RFC2821

Future of SMTP

E-mails never go out of fashion. And so does the protocol that runs them. One great disadvantage of SMTP are it's vulnerabilities. There are many of them like the Mail

Buffer overflow, server buffer overflow, relaying vulnerability.
Most of them affect the business prospects of SMTP. It is widely believed that SMTP has a great many disadvantages for the business use. The future of SMTP would be dedicated to overcoming these major vulnerabilities and making SMTP more secure.

As long as e-mails run the Simple Mail Transfer Protocol Will Keep improving and enhancing.