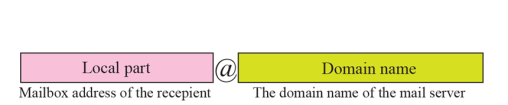
1. **mail Architecture and Services**
2. **Introduction**
   1. **What is E-mail?**

* Electronic mail, often abbreviated as **E-mail**, is a method of exchanging digital messages, designed primarily for human use. E-mail systems are based on a store & forward model in which E-mail server systems accept, forward, deliver and store messages on behalf of users, who only need to connect to the e-mail infrastructure.
* With e-mail, users create and send messages to one user, several users, or all users on a distribution list. Most e-mail software enables users to send text messages and attach files from word processors, spreadsheets, graphics programs, and so on. Many e-mail packages also permit you to filter or organize messages by priority.
  1. **Basic Functions of E-mail:** Typically, e-mail systems support **five** basic functions.

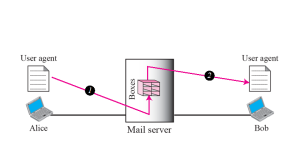
1. **Composition** refers to the process of creating messages and answers. Although any text editor can be used for the body of the message, the system itself can provide assistance with addressing and the numerous header fields attached to each message. For example, when answering a message, the e-mail system can extract the originator’s address from the incoming e-mail and automatically insert it into the proper place in the reply.
2. **Transfer** refers to moving messages from the originator to the recipient. In large part, this requires establishing a connection to the destination or some intermediate machine, outputting the message, and releasing the connection. The e-mail system should do this automatically, without bothering the user.
3. **Reporting** has to do with telling the originator what happened to the message. Was it delivered? Was it rejected? Was it lost? Numerous applications exist in which confirmation of delivery is important and may even have legal significance.
4. **Displaying** incoming messages is needed so people can read their e-mail. Sometimes conversion is required or a special viewer must be invoked, for example, if the message is a PostScript file or digitized voice. Simple conversions and formatting are sometimes attempted as well.
5. **Disposition** is the final step and concerns what the recipient does with the message after receiving it. Possibilities include throwing it away before reading, throwing it away after reading, saving it, and so on. It should also be possible to retrieve and reread saved messages, forward them, or process them in other ways.
   1. **E-mail Address**



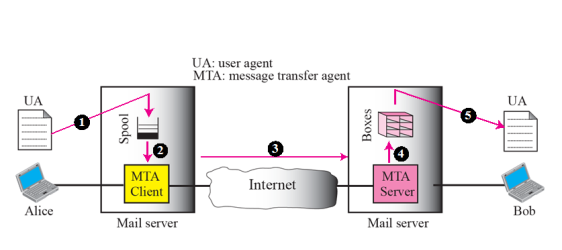
* 1. **Mail Protocols**
* **SMTP** - **Simple Mail Transport Protocol** is used on the internet, it is not a transport layer protocol but is an application layer protocol.
* **POP3** - **Post Office Protocol version 3** is used by clients to access an internet mail server to get mail. It is not a transport layer protocol.
* **IMAP4** - **Internet Mail Access Protocol version 4** is the replacement for POP3.
* **MIME** - **Multipurpose Internet Mail Extension** is the protocol that defines the way files are attached to SMTP messages.
* **X.400** - International Telecommunication Union standard defines transfer protocols for sending mail between mail servers.
* **MHS** - **Message Handling Service** by Novell is used for mail on Netware networks.

1. **Explanation of the architecture of E-mail using four scenarios**

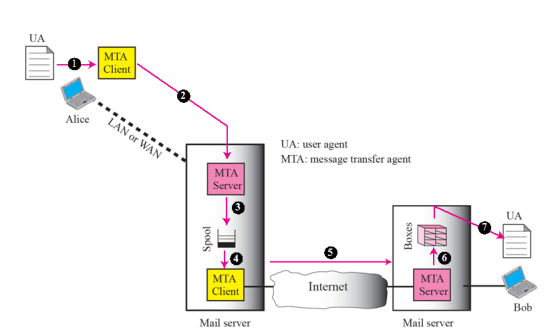
* **First Scenario:** When the **sender and the receiver** of an e-mail are on the **same mail server,**we need only two user agents.



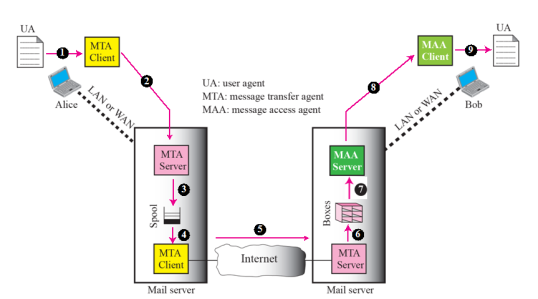
* **Second Scenario:** When the **sender and the receiver** of an e-mail are on **different mail servers**,we need two UAs and a pair of MTAs (client and server).



* **Third Scenario:**When the **sender is connected to the mail server via a LAN or a WAN**, we need two UAs and two pairs of MTAs (client and server).

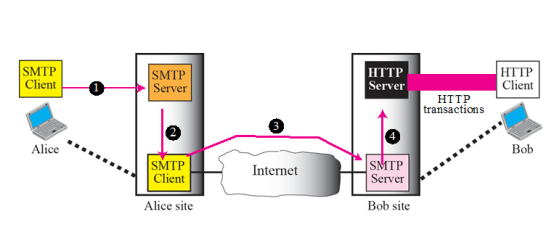


* **Fourth Scenario:**When **both sender and receiver** are connected to the **mail server via a LAN or a WAN**, we need two UAs, two pairs of MTAs (client and server), and a pair of MAAs (client and server). This is the most common situation today.

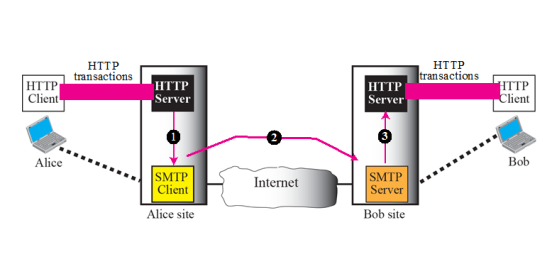


* **WEB-BASED MAIL**: E-mail is such a common application that some websites today provide this service to anyone who accesses the site. Three common sites are Hotmail, Yahoo, and Google. The idea is very simple. Let us go through two cases:

**Web-based e-mail, case 1**

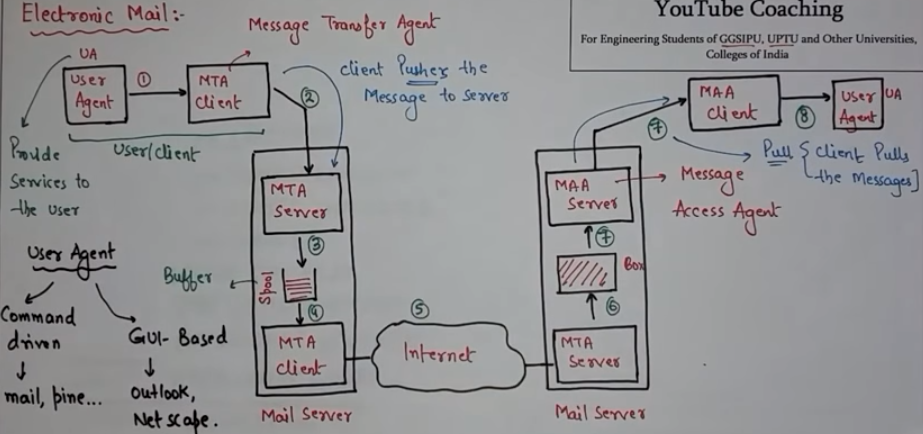


**Web-based e-mail, case 2**

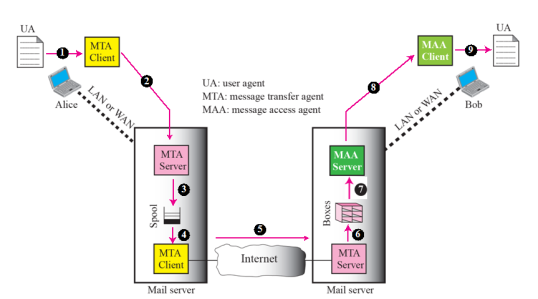


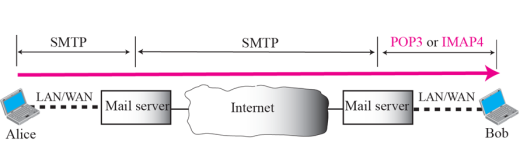
1. **Architecuure of E-mail System**
   1. **Components of Email System**

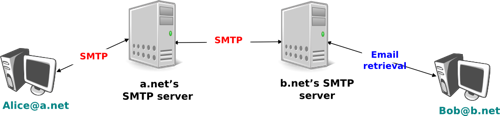
* **User Agent (UA) or Mail User Agent (MUA) or email client or email reader** is a software, **that** provides service to the user to make the process of sending and receiving a message easier. It allow people to read and send e-mail. The user agents are local programs that provide a command-based, menu-based, or graphical method for interacting with the e-mail system. Some examples of **command-driven user agents** are **mail, pine, and elm**. Examples of **GUI-based user agents** are **Eudora, Outlook and Netscape. T**he **user** usually **interacts directly** with a **MUA**..
* **Message/Mail Transfer Agent (MTA) or mail relay :** A **MTA** is a **software** that transfers electronic mail messages from one computer to another using a client–server application architecture. To send mail, a system must have the **client MTA**, and to receive mail, a system must have a **server MTA**. The formal protocol that defines the MTA client and server in the Internet is called Simple Mail Transfer Protocol (SMTP). A **MTA receives mail** from either **another MTA**, or a mail user agent (**MUA**). An MTA works in the background.
* **Message/Mail Delivery Agent (MDA) or Message/Mail Access Agent (MAA)** is a computer software component that is responsible for the delivery of e-mail messages to a local hosts mailbox.



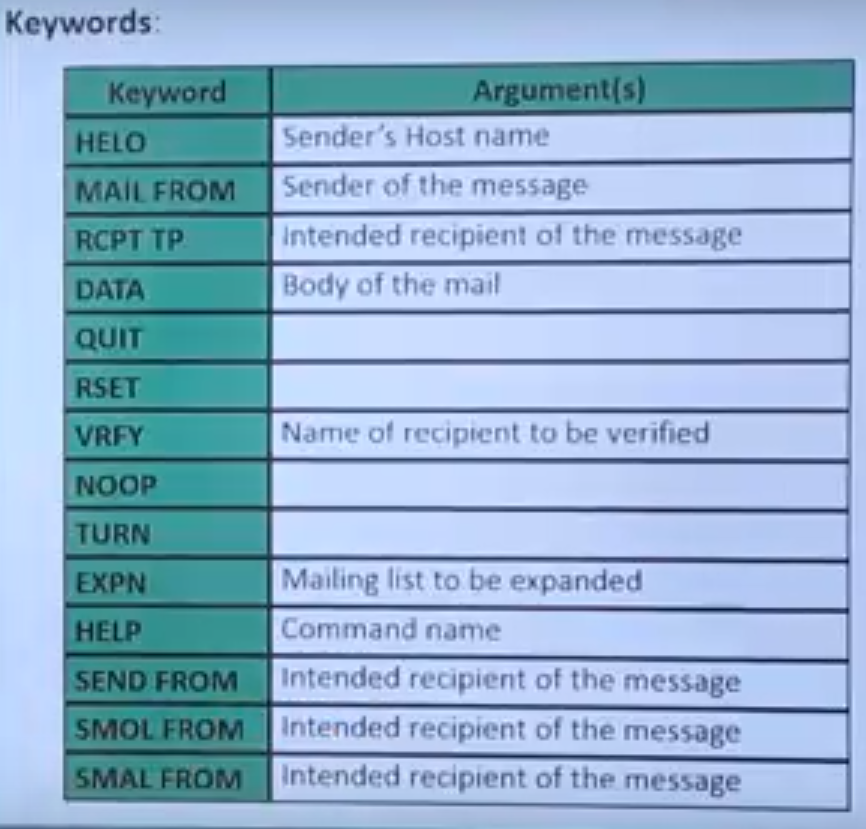
* 1. **Woring Principle of E-mail System**
* **The general architecture**







* 1. **SMTP**
* It is a standard protocol used to transfer e-mail across the network or internet.
* It is the protocol that defines MTA client and server in the intrnet.
* It helps in sending messages such as text or messages with attachment.
* First it is defined by RFC 821 in 1982, it was updated in 2008 with Extended SMTP additions by **RFC 5321**, which is client-server protocol protocol, in widespread use today.
* An MTA implements both the client (sending) and server (receiving) portions of the Simple Mail Transfer Protocol.
* Two pairs of MTA client/server programs are used in most common sitations.
* SMTP is used two times, between the sender and sender’s mail server and between the two mail servers. Another protocol (POP3 or ) is required between the server and the receiver.
* **SMTP Commands & responses:**SMTP simply defines how commands and responses must be sent back and forth between MTA client and MTA server in order to trannsfer data from MTA client to server.The client sends these commands to the mail server and the server replies with numerical codes and additional text messages after the numerical code. These reply codes tell the client if the previous sent command succeeded or failed.
* **SMTP Commands**: Commands are sent from MTA client to server. It considts of a key word followed by zero or more arguments. SMTP **defines 14 commands.** The **first five** are **mandatory**. The next three are **often used** and highly recommended. Th **last six** are **seldom used.**



Let Senders mail-id [abc@yahoo.com](mailto:abc@yahoo.com) Receivers mail-is [xyz@gmail.com](mailto:xyz@gmail.com)

|  |  |  |  |
| --- | --- | --- | --- |
| **Sl. No.** | **Command Name** | **Dscription** | **Example** |
| 1 | HELO | This command is used to identify the sender (client) to the SMTP server.  (Identify sender’s host name) | HELO: yahoo.com |
| 2 | MAIL FROM | Specifies the e-mail address of the sender.  (Identify sender) | MAIL FROM: abc@yahoo.com |
| 3 | RCPT TO | Specifies the e-mail address of the recipient.  (Indentify the intended receipient) | RCPT TO:xyz@gmail.com |
| 4 | DATA | Starts the transfer of the actual data (body text, attachments etc).  (Send the actual message) |  |
| 5 | RSET | Specifies that the current mail transaction will be aborted.  (To reset the connection) |  |
| 6 | VRFY | Asks the receiver to confirm that the argument identifies a user or mailbox.  (To verify the address of receipient) | VRFY:xyz@gmail.com |
| 7 | HELP | This command causes the server to send helpful information to the client. | HELP:mail |
| 8 | QUIT | Quits the session  (To terminate the message) |  |

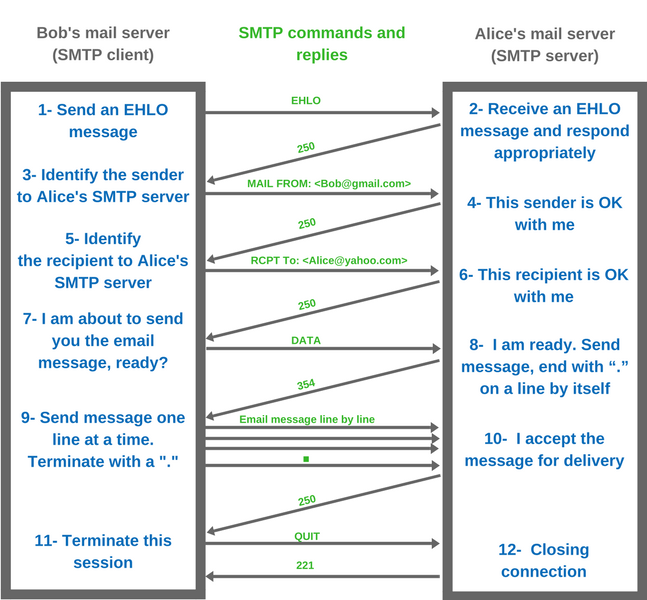
* **SMTP Responses**: Responses are sent from server to client. A response is a three digit code that may be followed by additional textual information.

1. **Positive condition reply**
2. - System Status
3. - Help
4. - Service Ready
5. - Service Closing
6. **Transient Negative Completion Reply**
7. - Service Not Available
8. - Mailbox not Available
9. - Command Aborted; Local Error
10. **Permanent Completion Reply**
11. - Syntax Error
12. - Transaction failed

**SMTP Possible Status Codes**

|  |  |
| --- | --- |
| **MTP Status Codes** | **Description** |
| **211** | **Help reply - system status** |
| **214** | **Help message** |
| **220** | **Service ready** |
| **221** | **Closing connection** |
| **250** | **Requested action okay** |
| **251** | **User not local - forwarding to &lt;path&gt;** |
| **354** | **Start mail input** |
| **421** | **Service not available** |
| **450** | **Action not taken - mailbox busy** |
| **451** | **Action aborted - local error** |
| **452** | **Action not taken - insufficient storage** |
| **500** | **Command unrecognized or syntax error** |
| **501** | **Syntax error in parameters or arguments** |
| **502** | **Command not supported** |
| **503** | **Bad sequence of commands (given out of order)** |
| **504** | **Command parameter not supported** |
| **550** | **Action not taken - mailbox unavailable** |
| **551** | **Not a local user** |
| **552** | **Aborted: Exceeded storage allocation** |
| **553** | **Action not taken - mailbox name not allowed** |
| 554 | Transaction failed |

* **Mail transfer Phases:** The process of transferring a mail message occures in three phases. Conncetion Establishment, Mail Transfer and Conncetion Termination.



### Connection Setup/establishment

An SMTP sender will attempt to set up a TCP connection with a target host when it has one or more mail message to deliver to that host. The following sequence occurs during connection setup:

1. The sender opens a TCP connection with the receiver.

2. Once the connection is established, the receiver identifies itself with '220 Service Ready".

3. The sender identifies itself with the HELO command.

4. The receiver accepts the sender's identification with "250 'OK".

5. If the mail service on the destination is not available, the destination host returns a "421 Service Not Available" reply in step 2 and the process is terminated.

### Mail transfer

• Once the connection has been established, the SMTP sender may send one or more messages to the SMTP receiver.

• There are three logical phases to the transfer of a message :

1. A MAIL command identifies the originator of the message.

2. One or more RCPT commands identify the recipients of this message.

3. A DATA command transfers the message text.

### Connection termination

• The SMTP sender closes the connection in the following manner:

1. The sender sends a QUIT command and waits for a reply.

2. Sender initiates TCP close operation for the TCP connection.

3. The receiver initiates its TCP close after sending is reply to the QUIT command.

**Sample-1**

C: HELO client.mydomain.com  
S: 250 Hello client.mydomain.com  
C: MAIL FROM:<mail@samlogic.com>  
S: 250 OK  
C: RCPT TO:<john@mail.com>  
S: 250 OK  
C: DATA  
S: 354 End data with <CR><LF>.<CR><LF>  
C: <The message data (body text, subject, e-mail header, attachments etc) is sent>  
C: .  
S: 250 OK, message accepted for delivery: queued as 12345  
C: QUIT  
S: 221 Bye

**Sample-2**

S: 220 smtp.example.com ESMTP Postfix

C: HELO relay.example.com

S: 250 smtp.example.com, I am glad to meet you

C: MAIL FROM:<bob@example.com>

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.com>

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

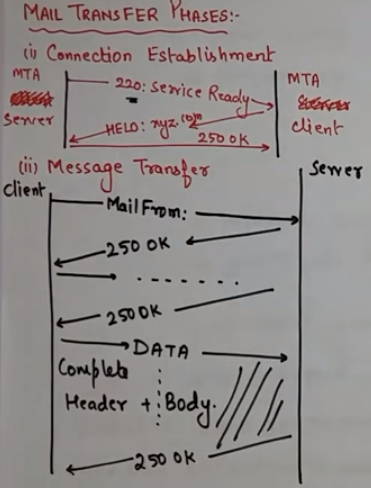
C: .

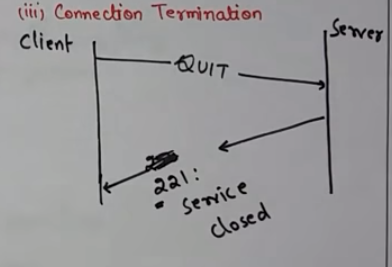
S: 250 Ok: queued as 12345

C: QUIT

S: 221 Bye

{The server closes the connection}





### For Extended Simple Mail Transfer Protocol (ESMTP) Refer <http://www.omnisecu.com/tcpip/smtp-status-codes-extended-simple-mail-transfer-protocol-esmtp.php>

Under SMTP, a user 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 well-known port (25), and the user SMTP process initiates a connection on that port. When the

TCP connection is successful, the two processes execute a simple request/response dialogue, defined

by the SMTP protocol, in which the user process transmits the mail addresses of the originator and

the recipient(s) for a message. When the server process accepts these mail addresses, the user process

transmits the message. The message must contain a message header and message text formatted in

accordance with RFC 822.