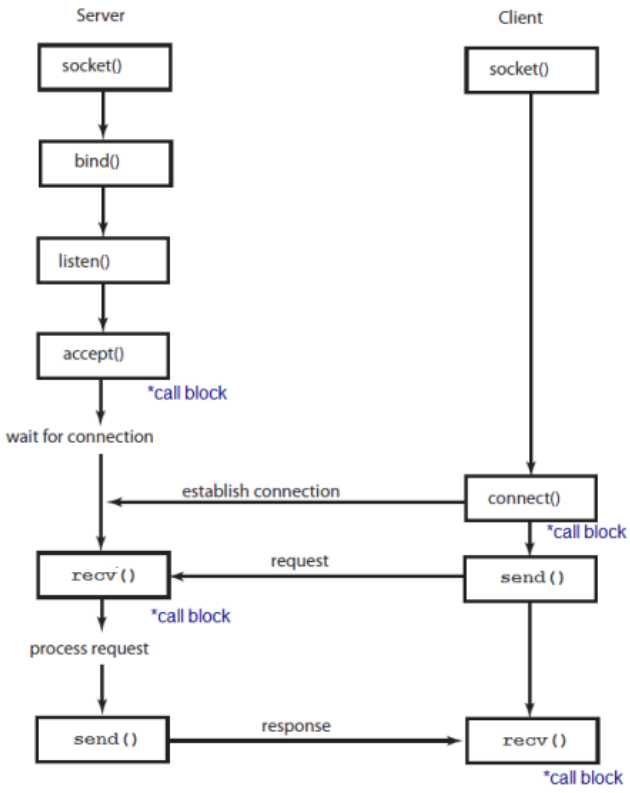


TCP/IP transfer control photocol

Application (Smtp), Transport(tcp,Udp), Netwwork (ip,ARP), Datalink(Eathernet), Hardware

Socket() End point for communication

Bind() Assign a unique telephone

Listen()wait for a call

Accept() Dial a Number

Send() Recev() Receive a call to talk

Close() Hangup.

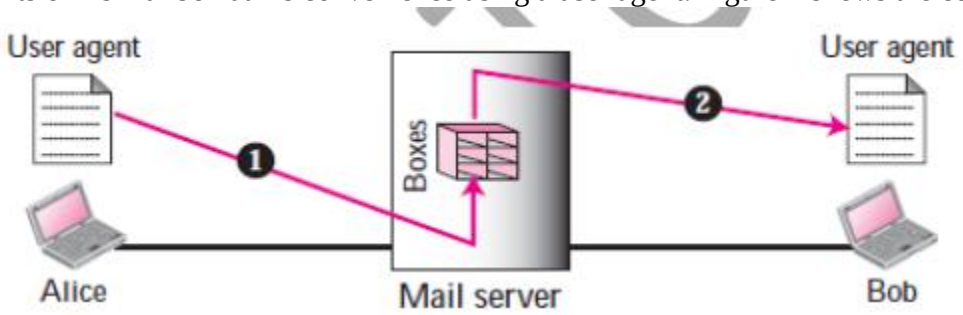
Electronic Mail: SMTP, POP, IMAP, and MIME

One of the most popular Internet services is electronic mail (e-mail). The designers of the Internet probably never imagined the popularity of this application program. **ARCHITECTURE –** To explain the architecture of e-mail, we give four scenarios.The fourth scenario is the most common in the exchange of e-mail.

First Scenario –
In the first scenario, the sender and the receiver of the e-mail are users (or application programs) on the same mail server; they are directly connected to a shared mail server. The administrator has created one mailbox for each user where the received messages are stored.

A *mailbox* is part of a local hard drive, a special file with permission restrictions. Only the owner of the mailbox has access to it. When Alice needs to send a message to Bob, she runs a *user agent* (*UA*) program to prepare the message and store it in Bob's mailbox.

The message has the sender and recipient mailbox addresses (names of files). Bob can retrieve and read the contents of his mailbox at his convenience using a user agent. Figure – shows the concept.



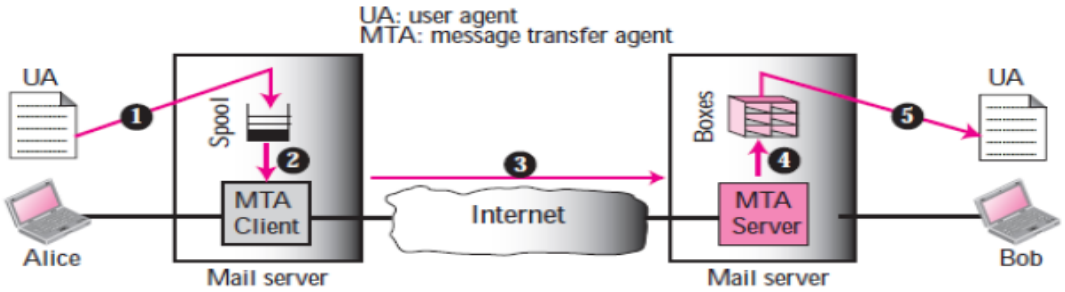
This is similar to the traditional memo exchange between employees in an office. There is a mail room where each employee has a mailbox with his or her name on it.

When Alice needs to send a memo to Bob, she writes the memo and inserts it into Bob's mailbox. When Bob checks his mailbox, he finds Alice's memo and reads it.

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

Second Scenario –

In the second scenario, the sender and the receiver of the e-mail are users (or application programs) on two different mail servers. The message needs to be sent over the Internet. Here we need **user agents (UAs)** and **message transfer agents (MTAs)** as shown in Figure –



Alice needs to use a user agent program to send her message to the mail server at her own site. The mail server at her site uses a queue (spool) to store messages waiting to be sent. Bob also needs a user agent program to retrieve messages stored in the mailbox of the system at his site.

The message, however, needs to be sent through the Internet from Alice's site to Bob's site. Here two message transfer agents are needed: one client and one server. Like most client-server programs on the Internet, the server needs to run all of the time because it does not know when a client will ask for a connection.

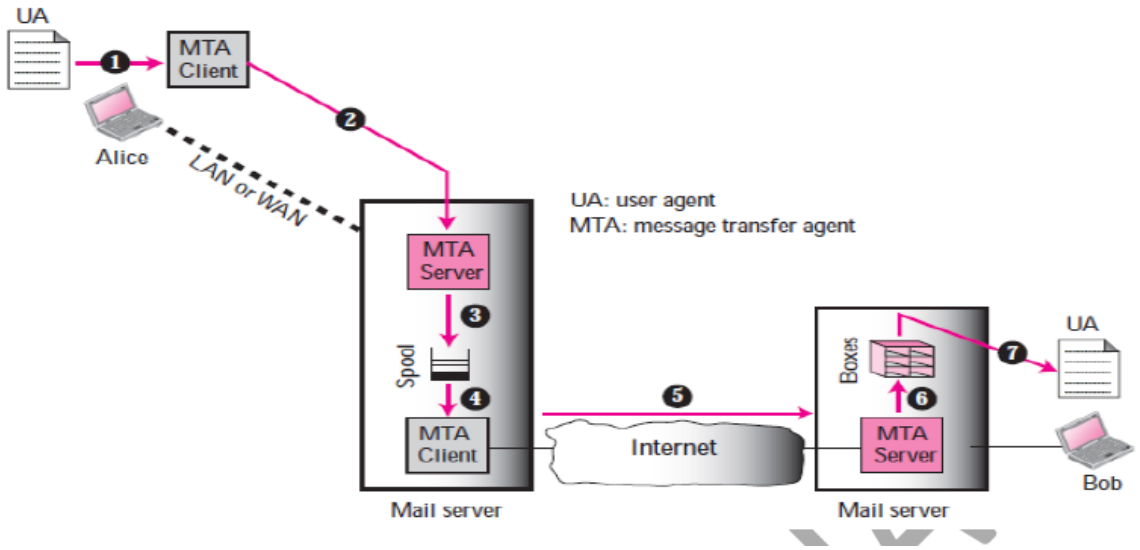
The client, on the other hand, can be triggered by the system when there is a message in the queue to be sent.

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

Figure – shows the third scenario. Bob, as in the second scenario, is directly connected to his mail server. Alice, however, is separated from her mail server. Alice is either connected to the mail server via a point-to-point WAN—such as a dial-up modem, a DSL, or a cable modem—or she is connected to a LAN in an organization that uses one mail server for handling e-mails; all users need to send their messages to this mail server.

Alice still needs a user agent to prepare her message. She then needs to send the message through the LAN or WAN. This can be done through a pair of message transfer agents (client and server). Whenever Alice has a message to send, she calls the user agent which, in turn, calls the MTA client.



The MTA client establishes a connection with the MTA server on the system, which is running all the time. The system at Alice's site queues all messages received. It then uses an MTA client to send the messages to the system at Bob's site; the system receives the message and stores it in Bob's mailbox.

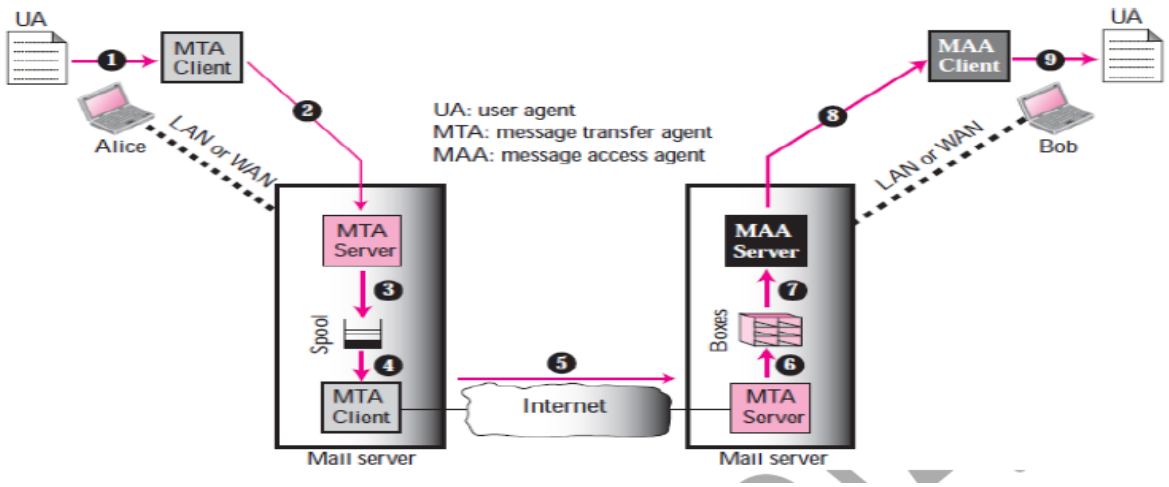
At his convenience, Bob uses his user agent to retrieve the message and reads it. Note that we need two pairs of MTA client-server programs.

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).

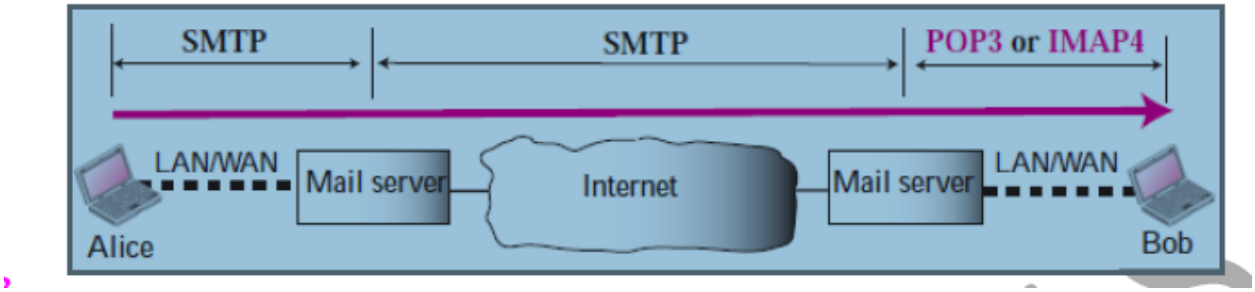
Fourth Scenario –

In the fourth and most common scenario, Bob is also connected to his mail server by a WAN or a LAN. After the message has arrived at Bob's mail server, Bob needs to retrieve it. Here, we need another set of client-server agents, which we call **message access agents (MAAs).** Bob uses an MAA client to retrieve his messages.

The client sends a request to the MAA server, which is running all the time, and requests the transfer of the messages. The situation is shown in Figure –



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.



Post Office Protocol (POP)*Internet Message Access Protocol* (IMAP) **Multipurpose Internet Mail Extensions (MIME)**

SIMPLE MAIL TRANSFER PROTOCOL (SMTP)

- In addition to message formats, the TCP/IP protocol suite specifies a standard for exchange of mail between machines.
- The standard specifies the exact format of messages a client on one machine uses to transfer mail to a server on another.
- SMTP focuses specifically on how the underlying mail delivery system passes messages across an internet from one machine to another.
- It does not specify how the mail system accepts mail from a user or how the user interface presents the user with incoming mail.

- Also it does not specify how mail is stored or how frequently the mail system attempts to send messages.
- communication between client and server consists of readable ASCII text.
- SMTP rigidly defines the command format, humans can easily read a transcript of interactions between a client and server.
- Initially a client establishes a reliable stream connection to the server and waits for the server to send a *220 READY FOR MAIL* message.
- If the server is overloaded, it may delay sending the *220* message temporarily.
- Upon receipt of *220* messages, the client sends a *HELLO* command.
- The end of line marks end of command.
- The server responds by identifying itself.
- Once communication has been established,the sender can transmit one or more mail messages, terminate the connection, or request the server to exchange the roles of sender and receiver so messages can flow in opposite direction.
- The receiver must acknowledge each message. It can also abort the entire connection or abort the current message transfer.
- Mail transactions begin with a *MAIL* command that gives sender identification as well as a *FROM:* field that contains the address to which errors should be reported.
- Response *250* means that all is well.The full response consists of the text *250 OK*.
- After a successful *MAIL* command, the sender issues a series of *RCPT* commands that identify recipients of mail message.
- The recipient must acknowledge each *RCPT* command by sending *250 OK* or by sending the error message *550 NO such user here*.
- After all *RCPT* commands have been acknowledged, the sender issues a *DATA* command. In essence, a *DATA* command informs the receiver that the sender is ready to transfer a complete mail message.
- The receiver responds with message 354 start mail input and specifies the sequence of characters used to terminate the mail message. It consists of 5 characters: carriage return, line feed,period,carriage return, and line feed.
- consider the following example:
- suppose user *smith* at host *Alpha.EDU* sends a message to users *jones,green and brown* at host *Beta.GOV*.

- The SMTP client software on host *Alpha.EDU* contacts the SMTP server software on host *Beta.GOV* and begins the exchange as shown:
-
- S: 220 Beta.GOV Simple Mail Transfer Service Ready
- C: HELO Alpha.EDU
- S: 250 Beta.GOV
- SMTP is much more complex because if a user has moved, the server may know the users new mailbox address. SMTP allows the server to inform the client about the new address so the client can use it in future. When informing the client about a new address, the server may choose to forward the mail that triggered the message, or it may request that the client take the responsibility for forwarding.

Mail Retrieval

- SMTP scheme- Server should remain ready to accept e-mails all the time.
- How can a user receive e-mail without a permanent connection?
  - User assigned a mail box on computer that has permanent internet connection.
  - User forms dialup connection and runs protocol that receives messages from permanent mailbox.

Post Office Protocol

- Protocol to transfer e-mail messages from mailbox to local computer.
- User invokes a POP3 client, which creates a TCP connection to a POP3 server on the mailbox computer.
- Computer with permanent mailbox must run 2 servers-
  - SMTP(Accepts mail sent to user)
  - POP3 (Allows to extract messages from mailbox).

IMAP4

- *Internet Message Access Protocol* is an alternative to POP3.
- User runs a IMAP4 client that contacts the server to retrieve messages.

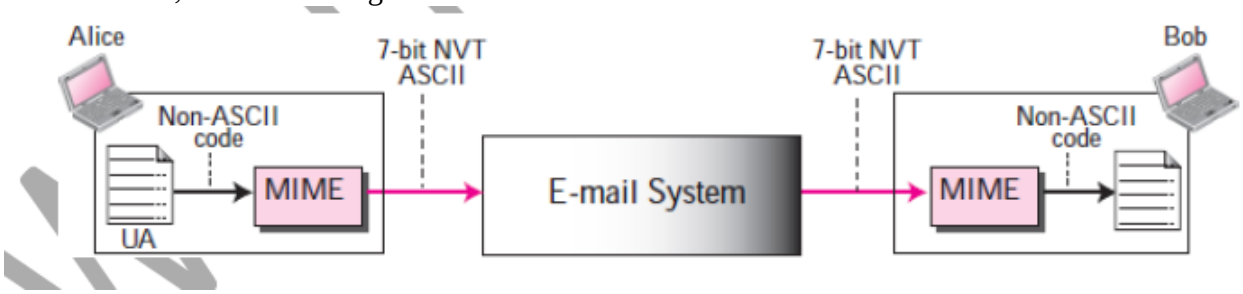
- IMAP4 allows user to dynamically create, delete or rename mail boxes.

Subtypes for Multipart Message

- ***Parallel***
  - *Permits message to include subparts that should be viewed together(Like video and audio).*
- ***Digest***
  - *Permits a message to contain set of other messages.*
  - *E.g., collection of e-mail messages from a discussion.*

MIME

Multipurpose Internet Mail Extensions (MIME) is a supplementary protocol that allows non-ASCII data to be sent through e-mail. MIME transforms non-ASCII data at the sender site to NVT ASCII data and delivers it to the client MTA to be sent through the Internet. The message at the receiving site is transformed back to the original data.
We can think of MIME as a set of software functions that transforms non-ASCII data to ASCII data and vice versa, as shown in Figure –



MIME Headers – MIME defines five headers that can be added to the original e-mail header section to define the transformation parameters:

1. MIME-Version

2. Content-Type

3. Content-Transfer-Encoding

4. Content-Id

5. Content-Description Figure – shows the MIME headers.

MIME-Version –\ This header defines the version of MIME used. The current version is 1.1.
Content-Type – This header defines the type of data used in the body of the message. The content

type and the content subtype are separated by a slash. Depending on the subtype, the header may contain other parameters. MIME allows seven different types of data, listed in Table

Type	*Subtype*	*Description*
Text	Plain	Unformatted
	HTML	HTML format (see Appendix E)
Multipart	Mixed	Body contains ordered parts of different data types
	Parallel	Same as above, but no order
	Digest	Similar to Mixed, but the default is message/RFC822
	Alternative	Parts are different versions of the same message
Message	RFC822	Body is an encapsulated message
	Partial	Body is a fragment of a bigger message
	External-Body	Body is a reference to another message
Image	JPEG	Image is in JPEG format
	GIF	Image is in GIF format
Video	MPEG	Video is in MPEG format
Audio	Basic	Single channel encoding of voice at 8 KHz
Application	PostScript	Adobe PostScript
	Octet-stream	General binary data (eight-bit bytes)

Content-Transfer-Encoding **–** This header defines the method used to encode the messages into 0s and 1s for transport: **Content- Transfer-Encoding: <type>** The five types of encoding methods are listed in Table –

Type	*Description*
7bit	NVT ASCII characters and short lines
8bit	Non-ASCII characters and short lines
Binary	Non-ASCII characters with unlimited-length lines
Base64	6-bit blocks of data are encoded into 8-bit ASCII characters
Quoted-printable	Non-ASCII characters are encoded as an equal sign plus an ASCII code

Content-Id **–** This header uniquely identifies the whole message in a multiple message environment.

Content-Description

This header defines whether the body is image, audio, or video.

Summary

- E-mail uses TCP/IP paradigm.
- ***SMTP***- Defines how a mail system on one machine transfers mail to server on another.
- ***POP3***- Specifies how user can retrieve contents of a mailbox.
- ***MIME***- Provides a mechanism that allows arbitrary data to be transmitted using SMTP.