# UNIT 5

# APPLICATION LAYER

# Applications

- Traditional Applications
- Web Services
- Multimedia Applications
- Overlay Networks

# Traditional Applications

Electronic Mail

- SMTP
- MIME
- IMAP
- POP3

World Wide Web

- HTTP

Name Service

- DNS

Network Management

- SNMP

# Companion protocols for email message format

- RFC 822
- MIME

- RFC 822 –> a message: a header, a body.

- Some Header lines are-
  - To:
  - Subject:
  - Date:
  - From:
  - Received:

- Body assumed to be simple text, represented as ASCII value.

- MIME was augmented with RFC 822 to now include all kinds of data in body- text, image, audio, video, pdf, etc.

- MIME -> Three pieces:

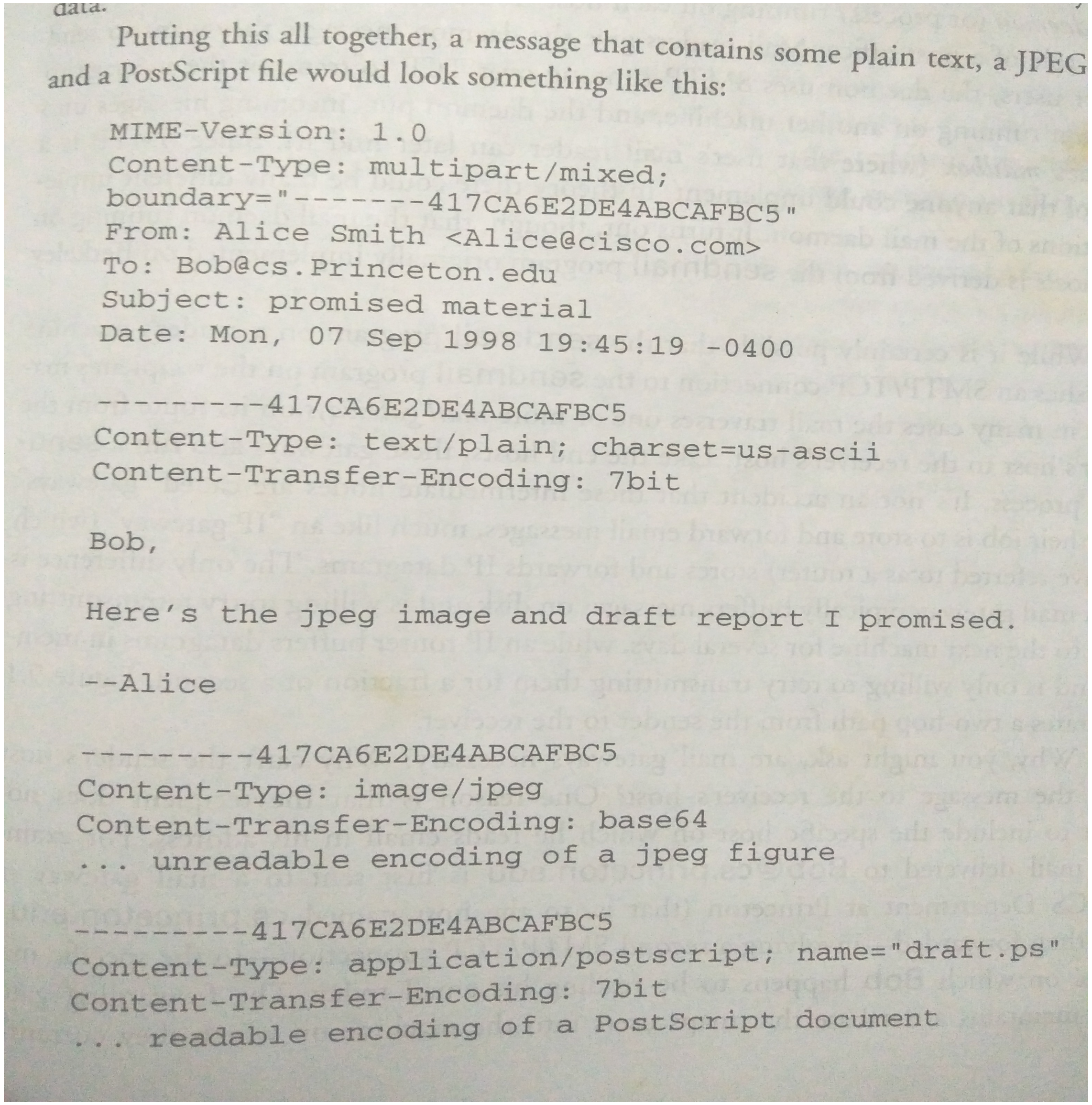
  1) Header Lines

  2)Definition for set of content types

  3)Way to encode the data types

- Header now includes:
  - MIME-Version
  - Content-Description
  - Content-Type
  - Content-Transfer-Encoding
- Content-Type can have value multipart with subtypes: mixed, signed, encrypted, etc.
- Content-Transfer-Encoding: base64, 7bit

# Multipurpose Internet Mail Extensions (Example)



data.

Putting this all together, a message that contains some plain text, a JPEG and a PostScript file would look something like this:

```
MIME-Version: 1.0
Content-Type: multipart/mixed;
boundary="-------417CA6E2DE4ABCAFBC5"
From: Alice Smith <Alice@cisco.com>
To: Bob@cs.Princeton.edu
Subject: promised material
Date: Mon, 07 Sep 1998 19:45:19 -0400

---------417CA6E2DE4ABCAFBC5
Content-Type: text/plain; charset=us-ascii
Content-Transfer-Encoding: 7bit

Bob,

Here's the jpeg image and draft report I promised.

--Alice

---------417CA6E2DE4ABCAFBC5
Content-Type: image/jpeg
Content-Transfer-Encoding: base64
... unreadable encoding of a jpeg figure

---------417CA6E2DE4ABCAFBC5
Content-Type: application/postscript; name="draft.ps"
Content-Transfer-Encoding: 7bit
... readable encoding of a PostScript document
```

Larry L. Peterson and Bruce S. Davie: COMPUTER NETWORKS – A Systems Approach

# Simple Mail Transfer Protocol(SMTP)

It is important

- (1) to distinguish the user interface (i. e., your mail reader) from the underlying message transfer protocols (such as SMTP or IMAP), and
- (2) to distinguish between this transfer protocol and a companion protocol (RFC 822 and MIME) that defines the format of the messages being exchanged

# Key Players in SMTP

First, users interact with a mail reader when they compose, file, search, and read their email.

- There are countless mail readers available, just like there are many Web browsers to choose from.
- In the early days of the Internet, users typically logged into the machine on which their mailbox resided, and the mail reader they invoked was a local application program that extracted messages from the file system.
- Today, of course, users remotely access their mailbox from their laptop or smart phone,  they do not first log into the host that stores their mail (a mail server).

Second, there is a mail daemon (or process) running on each host that holds a **mailbox.**

- You can think of this process, also called a message transfer agent (MTA), as playing the role of a post office: users (or their mail readers) give the daemon messages they want to send to other users, the daemon uses SMTP running over TCP to transmit the message to a daemon running on another machine, and the daemon puts incoming messages into the user’s mailbox (where that user’s mail reader can later find it).
- Since SMTP is a protocol that anyone could implement, in theory there could be many different implementations of the mail daemon.

- RFC 821 defines STMP model and procedures.

- HELO <domain-name>
- MAIL FROM : <reverse-path> <CRLF>
- RCPT TO : <forward-path> <CRLF>
- DATA <CRLF>
- QUIT

- 250 OK
- 550 Failure
- 354 Intermediate Reply
- 221 Closing Connection
- 251 User not local; will try forwarding to <forward_path>
- 551 User not local; please try <forward_path>

Here we assume that host Alpha contacts host Beta directly.

```
S: HELO Alpha.ARPA
R: 250 Hello daemon@mail.Alpha.ARPA [128.12.169.24]
S: MAIL FROM:<Smith@Alpha.ARPA>
R: 250 OK
S: RCPT TO:<Jones@Beta.ARPA>
R: 250 OK
S: RCPT TO:<Green@Beta.ARPA>
R: 550 No such user here
S: RCPT TO:<Brown@Beta.ARPA>
R: 250 OK
S: DATA
R: 354 Start mail input; end with <CRLF>.<CRLF>
S: Blah blah blah...
S: ...etc. etc. etc.
S: <CRLF>.<CRLF>
R: 250 OK
S: QUIT
R: 221 Closing Connection
```

**RFC 821: STMP Procedures**

The final step is for the user to actually retrieve his or her messages from the mailbox, read them, reply to them, and possibly save a copy for future reference.

- The user performs all these actions by interacting with a **mail reader**.
- As pointed out earlier, this reader was originally just a program running on the same machine as the user's mailbox, in which case it could simply read and write the file that implements the mailbox.
- This was the common case in the pre-laptop era.
- Today, most often the user accesses his or her mailbox from a remote machine using yet another protocol, such as the Post Office Protocol (POP) or the Internet Message Access Protocol (IMAP).

# POP(Post Office Protocol)

- Post Office Protocol (POP) is an application-layer Internet standard protocol used by local e-mail clients to retrieve e-mail from a remote server over a TCP/IP connection.
- POP has been developed through several versions, with version 3 (POP3) being the last standard in common use.
- POP3 allows you to download email messages on your local computer and read them even when you are offline.

# States of POP3 Session

- Authorisation State
- Transaction State
- Update State

# Authorization State

- Connection request from the client should be accepted with an OK message.
- Authentication of user through
  - USER name
  - PASS string
- If authenticated, acquire lock on maildrop.Go to Transaction State.
- Else, QUIT, and server closes connection.
  - C: QUIT
  - S: +OK dewey POP3 server signing off

**RFC 1939: POP3**

# Transaction State

- Execute commands until QUIT. If QUIT, go to Update State.
- STAT
  - C: STAT
  - S: +OK 2 320
- RETR msg
  - C: RETR 1
  - S: +OK 120 octets
  - S: <the POP3 server sends the entire message here>
  - S: .

- LIST [msg]
  - C: LIST
  - S: +OK 2 messages (320 octets)
  - S: 1 120
  - S: 2 200
  - S: .
  - ...
  - C: LIST 2
  - S: +OK 2 200
  - ...
  - C: LIST 3
  - S: -ERR no such message, only 2 messages in maildrop

- DELE msg
  - C: DELE 1
  - S: +OK message 1 deleted
  - ...
  - C: DELE 2
  - S: -ERR message 2 already deleted
- NOOP
  - C: NOOP
  - S: +OK
- RSET
  - C: RSET
  - S: +OK maildrop has 2 messages (320 octets)

**RFC 1939: POP3**

# Update State

- At the end of Transaction State, all the mails marked to delete will be deleted from the maildrop.
- OK status will be returned on success.
- Else, -ERR some deleted msgs not removed.

**Source:**

1)Larry L. Peterson and Bruce S. Davie: COMPUTER NETWORKS – A Systems Approach

2)RFC 821

3)RFC 1939