E-mail Addressing, Message Formats, MIME and Access Protocols
Chapter - 2

E-mail Addressing, Message Formats, MIME and Access Protocols

Introduction

This chapter presents a review of several vital aspects of e-mail system. It has been divided into four sections. The first section besides discussing TCP/IP e-mail addressing also introduces addressing used in some other non TCP/IP based networks. Second and third sections pertain to e-mail message formats and Multipurpose Internet Message Extensions respectively. The mail access protocols particularly POP, IMAP, direct mail access and access through WWW are given in section 4.

2.1. E-mail Addressing

All communication on an internetwork requires some way of specifying the identity of the intended recipient of the communication [2.1]. One element that sets e-mail apart from many other internetworking applications like FTP is that it is user-oriented [2.2]. E-mail is sent from one user's machine to another, unlike a file which is transferred using FTP from one machine to another. The e-mail address should identify user mailbox located on the user's SMTP server which is permanently connected to the Internet and not the user specific location at any particular time. It is so because: a) E-mail is indirect and based on the concept of a user's local Simple Mail Transfer Protocol (SMTP) server holding received messages until they can be retrieved. b) The machine employed by the user to access e-mail may not even routinely be connected to the Internet and thus not easy to identify and c) Ability to be accessed from multiple machines.

An electronic address book is usually implemented to allow e-mail software to associate a name with an e-mail address. Multiple Recipients Addressing allows easy specification of multiple recipients e.g. <user1@domain1>, <user2@domain2>, <user3@domain3>. In this case a separate copy is mailed to each recipient. Of course, aliases and/or address books can be used to specify each recipient here as well, making this even simpler. In large groups, a mailing list is used to make sending mail easier to multiple recipients. Many thousands of mailing lists are in use on the Internet, covering every subject imaginable [2.3, 2.4]. Each list differs in a number of ways,
including the following: a) implementation, b) subscription rules and technique, c) management method and style, d) culture and e) special features like ability to subscribe in digest mode and message access through the web.

2.1.1 TCP/IP Based E-mail Addresses

In TCP/IP protocol based systems the Domain Name System (DNS) is used for identifying servers. The machines on the Internet are arranged into device naming hierarchy using a domain name which consists of a series of text labels separated by dots. The complete e-mail address consists of two components viz; a user name speciation and a domain name specification which are connected together using the @ symbol like <username>@<domainname>. The format of <domainname> follows the syntax rules of DNS. The format of <username> allows special characters such as the underscore (_) and other special characters and spaces are also allowed in the <username> if they are surrounded by quotation marks or otherwise marked as being part the name, such as through the use of an escape character. Depending on the system, domain names may or may not be case-sensitive. It is also possible to specify an e-mail address using an Internet-standard Uniform Resource Locator (URL) like mailto:sgmrb@yahoo.com thereby allowing a link to be embedded in a hypertext (Web) document which when clicked invokes an e-mail client [2.3]. The DNS system includes special Mail eXchange (MX) record that can be set up to specify which SMTP server should be used for mail arriving at a particular domain name. DNS is also significant in that its MX resource records eliminate the need to relay e-mail from one SMTP server to the next to deliver it. In modern TCP/IP, it is possible to send e-mail directly from the sender’s SMTP server to the recipient’s SMTP server, making communication faster and more efficient. This provides enough flexibility to handle the following situations: a) Use of generic addresses that do not specify the name of the SMTP server to handle e-mail, to make it easier for senders to remember an e-mail address, b) Changing mail server temporarily to user and c) Use of multiple mail servers.

2.1.2. Other Types of E-mail Addresses

Several other networks have developed e-mail systems which are different from TCP/IP based e-mail systems and their addressing mechanisms are not directly compatible with that of
TCP/IP systems. Gatewaying is used for necessary conversions between SMTP e-mail system and non-SMTP based e-mail systems. Some of the non-SMTP based e-mail addresses are FidoNet addressing, Unix-to-Unix Copy Protocol (UUCP)-Style addressing and addressing for Gateway. The FidoNet e-mail system is a worldwide network connected using modems and proprietary protocols. In this system users are indentified using four numbers that specify the FidoNet, zone, net, node, and point (connection point) which is completely different from TCP/IP. However, to communication between TCP/IP and FidoNet a gateway system that allows mail to be sent to FidoNet using TCP/IP style domain names can be used for conversions. In UUCP system users are assigned addresses as a path of hosts separated by exclamation marks (!). The path dictates the route that mail takes to get to a particular user, passing through a series of intermediate machines running UUCP. One may encounter e-mail addresses that appear as if multiple TCP/IP addresses have been nested using unusual punctuation. For example, something like: user%domain1.com@subdomain2.edu. This is a way of addressing sometimes seen when e-mail gateways are used. It will cause the mail to be sent to user%domain1.com at subdomain.domain2.edu. The address then is interpreted as user domain@domain1.com.

2.2. Message Formats and Processing

Many types of headers that can be included in e-mail message are mandatory while others are not mandatory but are usually present, because they are fundamental to describing the message. Headers are organized into categories called header field groups as detailed in table 3.

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Appearance and No. of Occurrences per message</th>
<th>Occurrences Per Message</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Field Group: Origination Date</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Date:</td>
<td>Mandatory</td>
<td>1</td>
<td>Indicates date and time that the message was made available for delivery by the mail transport system. This is commonly the date/time that the user tells her e-mail clients to send the message.</td>
</tr>
</tbody>
</table>
### Field Name | Appearance and No. of Occurrences per message | Occurrences Per Message | Description
--- | --- | --- | ---
**Field Group: Origination Fields**

From: | Mandatory | 1 | E-mail address of the user sending the message, who should be the person who is the source of the message.

Sender: | Optional | 1 | E-mail address of the person sending the e-mail, if different from the message originator. For example, if person B is sending an e-mail containing message from person A on A's behalf, person A's address goes in the From: header and person B's in the Sender: header. If the originator and the sender are the same (commonly the case), this field is not present.

Reply-To: | Optional | 1 | Tells the recipient of the message, the address the originator would like the recipient to use for replies. If absent, replies are normally sent back to the From: address.

**Field Group: Destination Address Fields**

To: | Normally | 1 | A list of primary recipients of the message.

Cc: | Optional | 1 | A list of recipients to receive a copy of the message (cc stands for carbon copy, as used in old typewriters). There is no technical difference between how a message is sent to someone listed in the Cc: header and someone in the To: header. The difference is only in how the recipients interpret the message. The person in the To: list is usually the main recipient of the message, while the person in the Cc: list is being copied on the message for informational purpose.

Bcc: | Optional | 1 | Contains a list of recipients to receive a "blind" copy of the message without other recipients knowing they have received it. For examples, if person X is specified in the To: line person Y is in the cc: line and person Z in the Bcc: line all three would get a copy of the message, but X and y would not know Z had received a copy. This is done by either removing the Bcc, line before message delivery or altering in contents.

**Field Group: Identification Fields**

Message-ID: | Should be presents | 1 | Provides a unique for identifying a message; normally generated when a message is sent.

In-Reply-To: | Optional | 1 | When a message is sent in reply to another, the
<table>
<thead>
<tr>
<th>Field Name</th>
<th>Appearance and No. of Occurrences per message</th>
<th>Occurrences Per Message</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>normally presents for replies</td>
<td></td>
<td>message ID: field of the original message is specified in this field, to tell the recipients of the reply to what original message the reply pertains.</td>
</tr>
<tr>
<td>References:</td>
<td>Optional</td>
<td>1</td>
<td>Identifies other documents related to this message, such as other e-mail message.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Field Group: Informational Fields</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Subject:</td>
<td>Normally present</td>
<td>1</td>
<td>Describes the subject or topic of the message.</td>
</tr>
<tr>
<td>Comments:</td>
<td>Optional</td>
<td>Unlimited</td>
<td>Contains summarized comments about the message.</td>
</tr>
<tr>
<td>Keyword:</td>
<td>Optional</td>
<td>Unlimited</td>
<td>Contains a list of comma-separated keywords that may be useful to the recipients. May be used optionally when searching for message on a particular subject matter.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Field Group: Resent Fields</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Resent-date:</td>
<td></td>
<td></td>
<td>For each resent-block, Resent-Date: and resent-sender: are required; others are optional.</td>
</tr>
<tr>
<td>Resent-From:</td>
<td></td>
<td></td>
<td>Special fields used only when a message is resent by the original recipients to someone else, called forwarding. For example, person X may send a message to person Y, who forward it to Z. In that case, the original Date; From; and other headers are as they were when persons X sent the message. The Resent Date; Resent From: and other resent headers are used to indicate the date, originator recipients and other characteristics of the resent message.</td>
</tr>
<tr>
<td>Sender:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Resent-To:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Resent-cc:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Resent-Bcc:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Resent-Message-ID:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Field Group: Trace Fields</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Received:</td>
<td>Inserted by e-mail system</td>
<td>Unlimited</td>
<td>Inserted by computers as they process a message and transport it from the originator to the recipients. Can be used to trace the path a message has taken through the e-mail system.</td>
</tr>
<tr>
<td>Return-Path:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

SMTP also describes the protocol for transporting *mail objects*. A mail object consisting of two components: a message and an envelope. The e-mail message includes both message header and body and the envelope contains all the information necessary to accomplish transport of the
message. The envelope is not the same as the message headers. However, each message includes the recipients and other information needed for mail transport. E-mail software's at sender's client and server can process and interpret the message to construct the necessary envelope for SMTP to transport the message to its destination mailbox. The sender's client parses the intended recipient list to create envelop for transmission of the message using SMTP. The sender's server adds certain headers, especially trace headers such as Received and Return-Path, as they transport the message. These are generally pre-pended to the beginning of the message to ensure that existing headers are not rearranged or modified. However when gatewaying occurs between e-mail systems, certain headers must actually be changed to ensure that the message is compatible with non-TCP/IP e-mail software.

2.3. Introduction to MIME

The RFC2822 e-mail message format uses simple ASCII text which makes it easy to create, process, and read e-mail messages. To allow e-mail to carry multimedia information, arbitrary files, and messages in languages using character sets other than ASCII, the MIME standard was created. MIME has been adopted by other application protocols like Hypertext Transfer Protocols (HTTP), which uses MIME headers for indicating the characteristics of data being transferred. HTTP only uses some elements of MIME, therefore, HTTP messages are not compliant to MIME encoded e-mail message. MIME also defines a structure that allows multiple files to be encoded into a single e-mail message, including files of different types. Headers are added to the e-mail message to indicate how the information is encoded. This process is transparent to SMTP and the encoded e-mail message can then be sent through the system like any other e-mail message. However, both the sender and receiver e-mail clients must support MIME to encode and decode the message.

MIME was introduced in June 1992 in two standards RFC 1431 and RFC 1342 which were updated to RFC 1521 and RFC 1522 in September 1993. A supplementary standard RFC 1590 was published in March 1994. Some MIME standards are defined as part of other technologies like HTTP standard defined in RFC 1945. MIME was completely restructured in November 1996 and standards thereof RFC 2045, RFC 2046, RFC 2047, RFC 2048 and RFC 2049 were published. Since then various other extensions have been published, which include RFC 2183 and RFC
Chapter - 2: E-mail Addressing, Message Formats, MIME and Access Protocols

2557. Security features have been added to MIME to form (Secure) S/MIME through RFC 3851 in July 2004 [2.5].

MIME messages have a simple structure that carry a single media type, such as a text message or a graphical image, use either a simple structure or a complex structure that carry a composite media type, which allows multiple different media to be contained in a single message, such as a text message and a graphical image, or which allows the e-mail to encapsulate another e-mail message in it entirely. These structures are respectively called Discrete Media and Composite Media Structures. On recipient of an MIME message the headers are inspected to determine whether the message uses a simple or complex structure. For complex structure, the body of the message is parsed and each individual body part including its individual headers are intercepted individually. Mime-Version, Content Type, Content-Transfer-Encoding, Content-ID and Content-Description are the five primary MIME headers that communicate basic information about the content of each MIME entity (message or body part). MIME allows defining additional headers; however, both the sender and recipient must support these custom headers. Some new MIME headers have been defined in various RFCs that include Content-Disposition, Content- Location and Content-Length. Content Type header describes the nature of the data that is encoded in the MIME entity. This header specifies a content type and a content subtype, which are separated by a slash character. For example, a message containing an HTML document might have a Content-Type header of text/html, where a message containing a JPEG graphical file might be specified as image/jpeg. For a composite MIME type, the Content-Type header of the whole message will contain something like multipart/mixed or multipart/alternative, and each body part will contain individual Content-Type headers such as text/html or image/jpeg. This header is optional. When not present, the default of a regular ASCII text message is assumed. MIME uses special techniques to encode different types of media into ASCII text form, such as graphical images, sound files, video clips, applications programs, compressed data files, and many others. A complete list of MIME Medias organized by top level media type is listed on IANA's website at http://www.iana.org/assignments/media-types/index.html. MIME discrete media types allow MIME to represent hundreds of different kinds of data in e-mail messages. MIME standard also defines composite media types that allow MIME to perform even more spectacular feats, such as sending many types of data at
once or encapsulating other messages or information into e-mail. The use of a MIME composite media type is indicated via the Content-Type header of an RFC 822 message.

**2.4. Introduction to Protocols for E-mail Access and Retrieval**

SMTP is used for sending e-mail from the sender's client to the sender's server and it is also used for communication between sender's SMTP server and recipient's SMTP server. However, SMTP is not used for accessing the recipient's mailbox. This is owing to a variety of reasons that include:

i. Incorporating multiple access protocol in SMTP would require additional functionalities leading to complexity of SMTP.

ii. SMTP works on a push model as the transactions are initiated by the sender. For incorporating mailbox access functionalities, it needs to access to requests from recipients' clients. This would again make SMTP difficult rather than simple.

iii. Incorporating mailbox access functionalities within SMTP would limit its flexibility for access using diverse technologies.

iv. Further, this incorporation would also limit users to access e-mail from specific clients and thus shall pose severe difficulties to users who may need access to their mailboxes from different clients in different parts of the Globe.

RFC 1733, "Distributed Electronic Mail Models in IMAP4", describes three different paradigms, or models for mail access and retrieval. These are online access, offline access and disconnected access models.

In online access users have direct online access to their mailboxes. However, this is possible only if users' machines are connected permanently to the internet and are configured as SMTP servers. This is impractical baring some special examples like for users who run their own SMTP servers.

In offline access user's client computer establishes a connection to the server where his mailbox is located using some protocol. The mail is downloaded from the user mailbox on the server to the mailbox on the user client and the mail is deleted from the user's server mailbox.
of the mail is performed on the user's client computer and thus does not require a continuous connection to the Internet.

Disconnected access is a hybrid of online and offline access models. Users download mails from the server, and manipulate them without requiring a continuous connection to the server. However, the mail is not deleted from the server, as in the offline model. Periodically, users connect to the server and synchronize mailboxes on their server and client computers.

Online access has the main benefits of instant speed and universal access from any location. But requires a user to be online and often requires UNIX e-mail clients. Offline access has the main advantages of simplicity and short connection time requirements. However, this method is inflexible and poorly suited to access from different machines. Still, it is currently the most popular access method because simplicity is important; it is best typified by POP. The advantages of this method are: ability to access mail quickly and use offline mail processing. In recent years, a somewhat new mailbox access method has become popular: e-mail access using the World Wide Web. This technique allows a user to access his mailbox from any computer with an Internet connection and a web browser.

2.4.1. Introduction to Post Office Protocol (POP)

The simplicity and ease of implementation of offline e-mail access model are key to its success despite its limited features in comparison to other access models. Post Office Protocol (POP) which was published in 1984 and described in RFC 918 is a simple offline e-mail access model. It provided a simple way for a client computer to retrieve e-mail from a mailbox on SMTP server. It describes a simple sequence of operations in which a user provides name and password for authentication and then downloads the entire contents of a mailbox. Basic POP has been revised by POP2 and POP3 since its development. Several RFCs that include RFC 1081, RFC 1225, RFC 1460, RFC 1725 and RFC 1939 that describe revisions, extensions and additions to the basic POP protocol have been published.

POP3 is a TCP/IP client/server protocol and to provide access to mailboxes. POP3 server software must be installed and running continuously on the server on which the mailboxes are located and POP3 client software must be connected on clients. POP3 clients are regular end-
user e-mail programs that make connections to POP3 server to get mail; examples include Microsoft Outlook and Eudora E-mail. POP3 servers listen on well-known port number 110 for incoming connection requests from POP3 clients. Once a TCP connection is established, the POP3 session is activated. The client sends commands to the server, which replies with responses and/or e-mail message contents. POP3 commands are three or four letters long and are case-insensitive. They are all sent in plain ASCII text and terminated with a CRLF sequence. POP3 replies are textual and include only two basic responses. These are +OK: A positive response, sent when a command or action is successful or -ERR: A negative response indicates occurrence of an error.

These messages may be accompanied by explanatory text, especially in the case of an ERR response, to provide more information about the nature of the error. POP3 is described in terms of a Finite State Machine (FSM), with a session transitioning only once through three states namely Authorization, Transaction and Session during the course of its lifetime as shown in figure 8.
A session between a POP3 client and a POP3 server begins when the client sends a TCP connection request to the server. The connection is established using the standard TCP three-way handshake and the POP3 session commences. The first of the three states of a POP3 session, the Authorization state, is responsible for authenticating the POP3 client with the server. The process is illustrated in figure 9.

Once the POP3 client has successfully authenticated the user who is performing mailbox access, the session transitions from the Authorization state to the Transaction state. In Transaction state, the POP3 client issues the commands that perform mailbox access and message retrieval transaction. Table 4 lists commands defined in POP3 which are valid in Transaction State.
Table: POP3 Transaction Commands

<table>
<thead>
<tr>
<th>Command Code</th>
<th>Command</th>
<th>Parameters</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>STAT</td>
<td>Status</td>
<td>None</td>
<td>Requests status information for the mailbox. The server will normally respond, telling the client the number of messages in the mailbox and the number of bytes of data it contains. Optionally, more information may also be returned.</td>
</tr>
<tr>
<td>UST</td>
<td>List Messages</td>
<td>Optional messages number</td>
<td>Lists information for the messages in a mailbox; generally this means showing the message number and its size. If a message number is given, only that message’s information is provided; otherwise, the full contents of the mailbox are described, one line at a time, with a line containing just a single period at the end.</td>
</tr>
<tr>
<td>RETR</td>
<td>Retrieve</td>
<td>Message number</td>
<td>Retrieves a particular message from the mailbox. The server responds with a standard +OK message and then immediately sends the message in RFC 822 format, one line at a time. A line with a single period is sent after the last line.</td>
</tr>
<tr>
<td>DELE</td>
<td>Delete</td>
<td>Message number</td>
<td>Marks a message as deleted. Once deleted, any further attempt to access a message (using LIST or RETR, for example) results in an error.</td>
</tr>
<tr>
<td>NOOP</td>
<td>No Operation</td>
<td>None</td>
<td>Does nothing; the server just returns a +OK reply.</td>
</tr>
<tr>
<td>RSET</td>
<td>Reset</td>
<td>None</td>
<td>Resets the session to the state it was in upon entry to the Transaction state. This includes un-deleting any messages already marked for deletion.</td>
</tr>
<tr>
<td>TOP</td>
<td>Retrieve Message Top</td>
<td>Message number and number of lines</td>
<td>Allows a client to retrieve only the beginning of a message. The server returns the headers of the message and only the first N lines, where N is the number of lines specified. This command is optional and may not be supported by all servers.</td>
</tr>
<tr>
<td>UIDL</td>
<td>Unique ID</td>
<td>Optional message number</td>
<td>If a message number was specified, returns a unique identification code for that message; otherwise, returns an identification code for each message in the mailbox. This command is optional and may not be supported by all servers.</td>
</tr>
</tbody>
</table>

A typical POP3 mail exchange is shown in figure 10.
Some POP3 client may be configured not to delete messages from server mailboxes after retrieving them. This is useful, for example, when Web-based access is being combined with a conventional e-mail client program. Once the POP3 client has completed all the e-mail message access and retrieval transactions the session transactions from Transaction state to the Update state, to perform various housekeeping functions, after which both the POP3 session and the underlying TCP connection are terminated. The TCP connection between the client and server is then inhibited and the session is concluded. A POP3 mail-retrieval session normally lasts a few seconds or minutes, but it can take many minutes if the mailbox is large and the connection between the client and server are slow.

2.4.2. Introduction to Interactive Mail Access Protocol (IMAP)

Interactive Mail Access Protocol later renamed as Internet Message Access Protocol (IMAP) is flexible as it can operate in all three types of access modes i.e. online, offline and disconnected...
access. Of these, the online and disconnected access modes are of interest to most users of the protocol; offline access is similar to how POP works.

IMAP allows a user to a) Access and retrieve mail from a remote server so that it can be used locally while retaining it on the server, b) Set message flags so that the user can keep track of which messages have already been seen, already answered and so on, c) Manage multiple mailboxes and transfer messages from one mailbox to another. Mail can be organized into categories, which is useful for working on multiple projects or on various mailing lists, d) Determine information about a message prior to downloading it, to decide whether or not to retrieve it, e) Download only portions of a message, such as one body part from a MIME multipart message. This can be quite helpful in cases where large multimedia files are combined with short text elements in a single message and f) Manage documents other than e-mail. For example, IMAP can be used to access Usenet messages.

IMAP was first published as RFC 1064 in 1988. Later, it was updated by RFC 1176. Another version of IMAP named IMAP3 was published as RFC 1203 in February 1991. IMAP version 4 named IMAP4 has been described by RFC 1730 in December 1994. This has undergone several revisions through RFC 1731, RFC 2060 and RFC 3501. Further, secure version of MIME named S/MIME has been developed.

Like POP3, IMAP4 is also a standard client/server protocol. For the protocol to function, an IMAP4 server must be operating on the server where user mailboxes are located. Again, as with POP3, this does not necessarily need to be the same physical server that provides SMTP service. The mailbox must in some way be made accessible to both SMTP for incoming mail and to IMAP4 for message retrieval and modification. A mechanism for ensuring exclusive access to avoid interference between the various protocols is also needed. IMAP4 uses the Transmission Control Protocol (TCP) for communication. This ensures that all commands and data are sent reliably and received in the correct order. IMAP4 server listen on well-known port number 143 for incoming connection requests from IMAP4 clients. After a TCP connection is established, the IMAP4 session begins.
The session between an IMAP4 client and server is described in the IMAP standards using an FSM as given in figure 11.

**Figure 11: The IMAP FSM**

The IMAP states in the usual sequence in which they occur for a session are Not Authenticated State, Authenticated State, Selected State and Logout State.

Of the four IMAP states, only the first-three are interactive, i.e. commands are actively issued by the client and responses provided by the server. Some IMAP commands can be used while the session is in any state and other commands are state-specific.

Once an IMAP session is established, all communication between the client and server takes place in the form of commands sent by the client and responses returned by the server. Like
Chapter - 2: E-mail Addressing, Message Formats, MIME and Access Protocols

POP3, commands and responses are sent as strings of ASCII text and terminated with a CRLF sequence, making them compatible with the way data is sent using the Telnet Protocol. However, IMAP has a few differences from POP and many other TCP/IP application protocols. The first interesting thing about IMAP commands is that most are not abbreviated into codes of three or four letters – they are spelled out in full. So where POP3 has a STAT command, the command in IMAP is called STATUS. Commands are normally shown in uppercase, but they are case-insensitive.

IMAP commands are organized into four groups. a) "Any State" Commands: A small number of commands that can be used at any time during an IMAP session, b) Not Authenticated State Commands: Commands that can be used only in the Not Authenticated state. They are usually used for authentication, of course, c) Authenticated State Commands: Commands used to perform various actions on mailboxes and d) Selected State: A set of commands for accessing and manipulating individual messages that can be used only in the selected states.

Each command sent by the IMAP client elicits some sort of action from the IMAP server. The server takes action based on what the client requested and then returns one or more text strings to indicate what occurred. The server can send two types of replies namely Result and Response after a command is received. A Result is a reply usually indicating the status or disposition of a command. It may be tagged with the command tag of the command whose result it is communicating, or it may be a general message that is not tagged. The Result commands are OK, NO, BAD, PREAUTH and BYE. Any type of information that is being sent by the server to the client is called a Response. It is usually not tagged with a command tag and is not specifically intended to indicate server status. The response commands are ALERT, BADCHARSET, CAPABILITY, PARSE, PERMANENTFLAGS, READ-ONLY, READ-WRITE, TRYCREATE, UIDNEXT, UIDVALIDITY and UNSEEN.

The IMAP4 standard defines three different mechanisms by which a client may authenticate itself. These are plain login, TLS login and Negotiated Authentication methods.
Plain Login: This is the typical user name/password technique, using the LOGIN command by itself. This is similar to the simple scheme used in POP3, except that in IMAP4 one command is used to send both user name and password. It is the least secure method of authentication as the command and its parameters are sent in plain text. Plain Login is not recommended unless some other means is used in conjunction.

TLS Login: This is a secure login where the Transport Layer Security (TLS) protocol is first enabled with the STARTTLS command and then the LOGIN command is used securely. The STARTTLS does not itself cause the IMAP client to be authenticated and instead only causes the TLS negotiation to begin. Either LOGIN or AUTHENTICATE command must be used for secure login.

Negotiated Authentication Method: The AUTHENTICATE command allows the client and server to use any authentication scheme that they both support. The server may indicate which schemes it supports in response to a CAPABILITY command. After specifying the authentication mechanism to be used, the server and client exchange authentication information as required by the mechanism specified. This may require one or more additional lines of data to be sent.

Once the IMAP client has been authorized to access the server, it enters the Authenticated state, where it is allowed to execute tasks on whole mailboxes. Since IMAP allows multiple mailboxes to be manipulated, message-specific commands cannot be used until the client informs the server which mailbox it wants to work with. Only one mailbox can be accessed at a time in a given session. After the SELECT or EXAMINE command is successfully issued, the session enters in the selected state. In this state, the full palette of message and mailbox commands is available to the client. Most of IMAP's message-specific commands do not include a mailbox name as parameters, since the server knows automatically that the commands apply to whatever mailbox was selected in the Authenticated state. The session remains in the selected state for as long as the client continues to have work to do with the particular selected (or examined) mailbox. Three different actions can cause a transition out of the selected state. These are: a) If the client has nothing more to do when it is done with the current mailbox, it can use the LOGOUT command to end the session, b) The client can use the CLOSE command to tell the...
server it is finished with the current mailbox but keep the session active. The server will close
the mailbox, and the session will go back to the Authenticated state and c) The client can issue a
new SELECT or EXAMINE command, which will implicitly close the current mailbox and then
open the new one. The transition in this case is from the selected state back to the selected state,
but with a new current mailbox.

2.4.3. Introduction to Direct Mail Access

An e-mail client can gain access to server mailbox using one of several generic methods. These
methods are all variations of the online e-mail access model and work by establishing direct
access to the server where the mailbox is located. The mailbox is implemented as a file on a mail
server, so if that file can be made available, it can be viewed and manipulated like any other file
using an e-mail client program. Following are some of these:

Using the SMTP Server Directly: The simplest method for gaining access to the mailbox is to log
on to the mail server itself but for security reasons it is not an option. However, some people do
run their own SMTP servers, giving them considerable control and access to their mailboxes
directly.

File Sharing Access: Using a protocol such as Network File System (NFS), it is possible to have a
mailbox mounted on a user’s client machine where it can be accessed as if it were a local file.
The mail is still on the mail server and not the client machine, but the communication between
the client and the server occurs transparently to both the user and the e-mail client software.

Dial-Up Remote Server Access: A user on a client machine can dial up a mail server where
mailbox is located and logs in to it. The user then can issue commands to access mail on that
server directly. However, the mail server needs to be configured to handle this situation.

Telnet Remote Server Access: Instead of dialing in to the server as in case of dial-up remote
server access, a user can connect to it for remote access using the Telnet Protocol.

Direct server access is a method that has been around for decades. At one time, this was how
the majority of people accessed e-mail, for two main reasons. i) Protocols like POP or IMAP had
not by then been developed, and ii) The general way that e-mail and networks were used years ago was different from what it is today.

Direct e-mail access if not the only method but was the most preferred method for e-mail access in early days of e-mail. However, with newer and more secure methods of e-mail access direct access is rarely used nowadays.

2.4.4. Introduction to Mail Access using WWW

Most e-mail users prefer to take the advantages of online access, especially the ability to read mail from a variety of different machines but with tools that are easy to use. The mail access using WWW exploits the flexibility of the Hypertext Transfer Protocol (HTTP) to tunnel e-mail from a mailbox server to the client. A web browser (client) is opened and given a URL for a special web server document that accesses the user's mailbox. The web server reads information from the mailbox and sends it to the web browser, where it is displayed to the user. This method uses the online access model like direct server access. The web based e-mail is much easier to use than other online access methods. Web-based mail is easy to use, but inflexible; the users do not have direct access to their mailbox and can use only whatever features the provider's website implements. Use of conventional direct access methods is quick and easy to read through a mailbox and the same is true for access using POP3 after the mail is downloaded. In contrast, Web based mail services mean each request requires another HTTP request/response cycle. The fact that many Web-based services are free often means server overload that exacerbates the speed issue. When Web-based mail is combined with other methods such as POP3, care must be taken to avoid strange results. If the Web interface does not provide all the features of the conventional e-mail client, certain changes made by the client may not show up when Web-based access is used and even mail may have deleted if not configured otherwise. Also, mail retrieval using POP3 by default removes the mail from the server. Many e-mail client programs now allow mail to remain intact on the server after retrieving it using POP3.
Summary

Different Internet applications have different addressing mechanisms, however, for e-mail unlike most other Internet applications, addresses are aimed to recipients rather than machine specific. Further, some networks on Internet not based on TCP/IP have different addressing mechanisms which are not directly compatible with TCP/IP based e-mail addressing. E-mail message has a specific format that comprises of a header and a body which is allowed to carry multimedia information, arbitrary files, and messages in languages using character sets other than ASCII by the use of Multimedia Internet Mail Extensions (MIME) standard. SMTP is not used for accessing the recipient's mailbox and instead three different paradigms, or models for mail access and retrieval vis-à-vis online access, offline access and disconnected access models have been proposed each having its relative merits.
References


[2.4] Artur Miguel Dias, Moving from mailing lists to web-based discussion forums, ITiCSE '05: Proceedings of the 10th annual SIGCSE conference on Innovation and technology in computer science education, June, 2005.