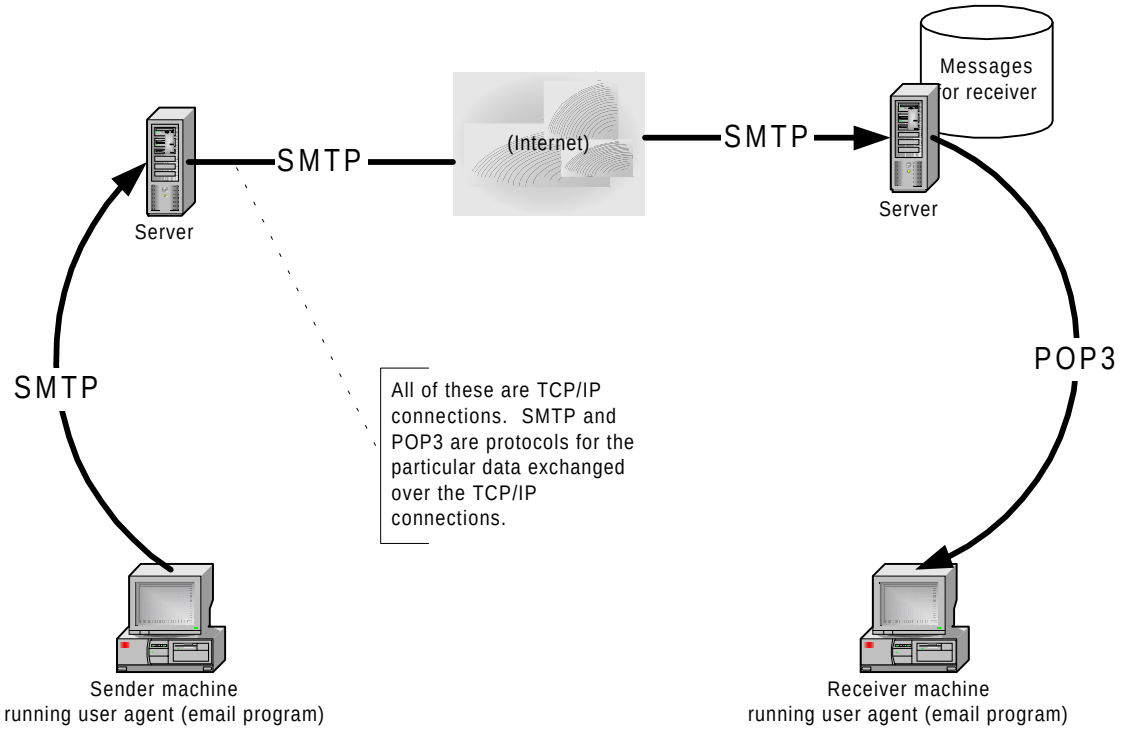
Just Enough about Electronic Mail Protocols and Terminology

For OMSE 532, Spam filter design exercise

This document attempts to provide, very briefly, an overview of the major internet electronic mail protocols that one would need to understand to design e-mail tools. It is designed to be "just enough" for the spam filter design exercise; really building tools would require a more detailed understanding. The material described here is drawn primarily from internet requests for comments (RFCs), which are available from http://www.ietf.org.

There are two major groups of protocols that govern electronic mail on the internet. One group of protocols describes the format of electronic mail messages, including fields that specify delivery addresses and other "header" information. The second set of protocols governs the transport of messages. They are not completely independent, since the header fields in a message includes the addresses that guide its transport, and since the transport protocols can insert additional header fields to record the actions taken in transport.

The following diagram is a conceptual overview of the transport of an electronic mail message from a sender to a receiver; it is discussed below.



The sender is running a program like Eudora, Outlook, or Netscape mail; this program is called a "user agent." The user composes a message using the user agent and sends it. The user agent program connects to a mail server using the Simple Mail Transport Protocol (SMTP). The server may be on a different machine, as illustrated in the diagram, using a local area network or a dial-up connection, or it could be on the same physical machine. In any case, the SMTP protocol is

used over a TCP/IP network connection. From the user agent's perspective, all TCP/IP connections are the same, regardless of the underlying transport mechanism.

The SMTP protocol is used to forward the email message until it reaches its destination on the mail server of the receiver, where it is stored in a local database. (Typically this database is just a text file.) The receiver is also running a "user agent" program. When the user invokes the "check mail" function of the user agent program, the user agent connects to the receiver's mail server using the POP3 protocol (also over a TCP/IP connection), and uses the POP3 protocol to transfer the waiting message from the server to the receiver's machine.

Note that none of these machines need be distinct. The sequence of transfers would be exactly the same if the sender and receiver shared the same mail server and even if they ran their mail agent programs on the same machine; only the underlying transport mechanism would be different, and that is invisible to the user and to the user agent software. If Joe@cs.uoregon.edu sends an electronic message to Joe@cs.uoregon.edu while logged in to cs.uoregon.edu, the message is transported through these same steps.

The message format protocols

This basic encoding scheme for internet mail messages, as well as definitions of some required and optional fields, was set forth in internet request for comment number 822 (http://www.ietf.org/rfc/rfc0822.txt) and now the common name for this scheme is "RFC 822 format." RFC 822 has been augmented and partly superseded by the "MIME" standards, which provide ways to encapsulate complex non-textual objects in electronic mail, but the basic scheme remains the same.

An RFC 822 format e-mail message text[1] divided into a set of fields. One of these fields is the message body. The other fields are called "header fields." Each header field is divided into a field-name and a field-body, separated by a colon. For example:

Subject: How the elephant got its trunk

The field-name of this field is "Subject", and the field-body is "How the elephant got its trunk." We use the field-name of a field to refer to it; e.g., we would refer to the line above as "the subject field."

There are a number of standard header fields such as subject, from, to, and mime-version. Some of these are required, and others are optional. Here is an example set of header fields:

```
Return-Path: <bounces-registration@findmail.com>
Received: from findmail.com (m4.egroups.com [207.138.41.151])
        by cs.uoregon.edu (8.9.1a/8.9.1) with SMTP id RAA26090
        for <michal@cs.uoregon.edu>; Sat, 3 Apr 1999 17:11:04 -0800 (PST)
Received: (qmail 5209 invoked by uid 505); 4 Apr 1999 01:11:03 -0000
Date: 4 Apr 1999 01:11:03 -0000
Message-ID: <19990404011103.5208.qmail@findmail.com>
From: "eGroups.com" <support-sub--roadcoders-palmos--251272@egroups.com>
Subject: Verify subscription to eGroup: roadcoders-palmos
Reply-To: support-sub--roadcoders-palmos--251272@egroups.com
To: michal@cs.uoregon.edu
Status:
```

[1] Strictly speaking, internet mail messages are encoded as text for transport, but the text may be a representation of non-textual data such as graphics, audio, etc.

The “Received” field may be of particular interest. An email message may pass through several mail servers between the original sender’s mail host and the receiver’s mail host. The “Received” field provides a kind of “audit trail” of its path. Spammers typically don’t want the receiver to know how the message was delivered, so they place inaccurate information in this field, often including the addresses of machines that don’t exist. Note that some fields, including “Received,” can appear more than once in a single e-mail message.

In addition to the standard fields, RFC 822 provides a way to include additional, non-standard information in the same format. To prevent possible clashes between non-standard header fields and standard fields that are introduced later, it is required that every non-standard header field name begins with “X-“. For example:

> From: "brucem" <censored@censored.com>
> To: "Michal Young" <michal@cs.uoregon.edu>
> Subject: Re: Setting your email options
> Date: Fri, 2 Apr 1999 11:17:25 -0500
> X-Mailer: Microsoft Outlook Express 5.00.0810.800
> X-MimeOLE: Produced By Microsoft MimeOLE V5.00.0810.800

This message contains a number of non-standard fields which have been set by the sender’s user agent, which we can see is Microsoft Outlook Express. Additional non-standard fields can be inserted at any point in the mail transport. For example, the receiver’s mail server could insert additional fields. Often these fields are not displayed by the user agent unless specifically requested (e.g., by clicking the “blah blah blah” button in Eudora), but they can be used by the user agent to filter and classify messages.

Retrieving Messages

The mail agent uses POP3, the Post Office Protocol, version 3 (ftp://ftp.isi.edu/in-notes/rfc1939.txt) to retrieve mail messages from the user’s mail server. (It may also use the newer IMAP protocol, but POP3 is simpler and still more widely used, so we will limit our attention here to POP3.) The POP3 protocol is simply a set of textual commands from the user agent and responses from the mail server. Responses consist of a status indicator (either “+OK” or “-ERR” and a keyword, possibly followed by other information. A sequence of commands and responses is called a “session.” Here is a typical session that might occur when you click the “check mail” button in your e-mail program.

- The user agent[2] identifies the user to the POP3 mail server (similar to logging in).
- The user agent uses the UIDL (unique identifier list) command to request a list of email messages stored on the server. The list consists of pairs, one element of each pair being an integer identifier used to refer to the message in the current session, and the other element being the unique identifier of the message. These identifiers are “persistent” across sessions, i.e., a message that remains on the server will always have the same unique identifier, no matter how long it remains. In this way, the user agent can determine which messages have already been transferred by referring to a log of unique ids in its own local storage.
- The user agent requests parts or all of the new messages be transmitted. The TOP command can be used to retrieve only a portion of a message, e.g., “TOP 12 200” is a command to transmit the first 200 characters of message 12. The RETR command

[2] The POP3 protocol refers to the user agent as the “client.” I will use “user agent” here for consistency with other usage in this document.

requests transmission of the whole message, and also causes the mail server to set the status field of the message to “R” for “has been read.”

- The user agent marks some messages for deletion, using the DELE command. It might delete all messages after transmission, or it might delete only messages that were older than some limit set by the user.

- The user agent uses the QUIT command to terminate the POP3 session. It is at this point that all actions are “committed”, and in particular, this is when the effect of message deletions becomes permanent in the mail server. If the user agent terminates the session by closing the TCP/IP connection without issuing the QUIT command, the mail server reverts to its prior state without deleting any messages.

Additional notes

Note that SMTP and POP3 are “on the wire” protocols, i.e., there is no application programming interface other than whatever interface the operating system provides for reading and writing text through TCP/IP. One consequence of this is that one can sometimes write a “proxy” program that mediates between a client and a server, masquerading as a server to the client and as a client to the server. The advantage of creating a proxy for an “on the wire” protocol is that it can be done without any modification to existing servers and without any knowledge of proprietary application interfaces.