SMTP

George Porter
CSE 124
February 12, 2015
Announcements

- Homework 2 out
- Project 2 checkpoint due Tuesday
Traditional Applications

Two of the most popular—
  • The World Wide Web and
  • Email.

Broadly speaking, both of these applications use the request/reply paradigm—users send requests to servers, which then respond accordingly.
It is important to distinguish between application programs and application protocols.

For example, the HyperText Transport Protocol (HTTP) is an application protocol that is used to retrieve Web pages from remote servers.

There can be many different application programs—that is, Web clients like Internet Explorer, Chrome, Firefox, and Safari—that provide users with a different look and feel, but all of them use the same HTTP protocol to communicate with Web servers over the Internet.
SMTP

- SMTP: Simple Mail Transfer Protocol is used to exchange electronic mail.
Email is one of the oldest network applications.

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.

Analogy to HTTP and HTML.
Electronic Mail (SMTP, MIME, IMAP)

- **Message Format**
  - RFC 822 defines messages to have two parts: a *header* and *a body*. *Both parts are* represented in ASCII text.
  - Originally, the body was assumed to be simple text. This is still the case, although RFC 822 has been augmented by MIME to allow the message body to carry all sorts of data.
  - This data is still represented as ASCII text, but because it may be an encoded version of, say, a JPEG image, it’s not necessarily readable by human users.
  - The message header is a series of `<CRLF>`-terminated lines. `<CRLF>` stands for carriage-return+ line-feed, which are a pair of ASCII control characters often used to indicate the end of a line of text.)
Electronic Mail (SMTP, MIME, IMAP)

Message Format

- The header is separated from the message body by a blank line. Each header line contains a type and value separated by a colon.
- Many of these header lines are familiar to users since they are asked to fill them out when they compose an email message.
- RFC 822 was extended in 1993 (and updated quite a few times since then) to allow email messages to carry many different types of data: audio, video, images, PDF documents, and so on.
MIME-enabled Email

- MIME consists of three basic pieces:
  1. a collection of header lines that augment the original set defined by RFC 822.

These header lines describe the data being carried in the message body

MIME-Version: (the version of MIME being used)
Content-Description: (a human-readable description of what’s in the message, analogous to the Subject: line)
Content-Type: (the type of data contained in the message)
Content-Transfer-Encoding (how the data in the message body is encoded).
MIME-enabled Email

- MIME consists of three basic pieces:
  1. a collection of header lines that augment the original set defined by RFC 822.
  2. The second piece is definitions for a set of content types (and subtypes). For example, MIME defines two different still image types, denoted image/gif and image/jpeg, each with the obvious meaning.
  3. The third piece is a way to encode the various data types so they can be shipped in an ASCII email message.
Message Transfer

- For many years, the majority of email was moved from host to host using only SMTP.
- While SMTP continues to play a central role, it is now just one email protocol of several,
- IMAP and POP being two other important protocols for retrieving mail messages.
Terminology

- **User Agent**: end-user mail program
- **Message Transfer Agent**: responsible for communicating with remote hosts and transmitting/receiving email
  - Both client and server
- **Mail Exchanger**: host that takes care of email for a domain
SMTP

Used to exchange mail messages between mail servers (Message Transfer Agents)

MTA

SMTP

MTA

SMTP

MTA

File System

UA

UA
SMTP Protocol

- SMTP sender is the client
- SMTP receiver is the server
- Alternating dialogue:
  - Client sends command and server responds with command status message
  - Order of the commands is important
  - Status messages include ASCII encoded numeric status code (like HTTP, FTP) and text string
SMTP Commands

- **HELO** - identifies sender

- **MAIL FROM:** - starts a mail transaction and identifies the mail originator

- **RCPT TO:** - identifies individual recipient. There may be multiple **RCPT TO:** commands

- **DATA** - sender ready to transmit a series of lines of text, each ends with \r\n. A line containing only a period ‘.’ indicates the end of the data
Data Format

- ASCII only - must convert binary to an ASCII representation to send via email
- What if we want to send a line containing only a period?
  - Sender prepends a period to any line staring with a period (in the message)
  - Receiver strips the leading period in any line that starts with a period and has more stuff
Typical Exchange

[gmporter@ramp gmporter]$ telnet fast.ucsd.edu 25
Trying 132.239.15.4...
Connected to fast.ucsd.edu.
Escape character is '^]'.
220 fast.ucsd.edu ESMTP
HELO ramp.ucsd.edu
250 fast.ucsd.edu Hello ramp.ucsd.edu [137.110.222.239], pleased to meet you
MAIL FROM: <gmporter@cs.ucsd.edu>
250 2.1.0 <gmporter@cs.ucsd.edu>... Sender ok
RCPT TO: gmporter
250 2.1.5 gmporter... Recipient ok
DATA
354 Enter mail, end with "." on a line by itself
This is a test message.
More text
.
250 2.0.0 j3QKGnsR029250 Message accepted for delivery
DATA
354 Enter mail, end with "." on a line by itself
Hi dave - this message is a test of SMTP
...
..foo
...
.
250 VAA0771 Message accepted for delivery
Resulting Message:

Hi dave - this message is a test of SMTP
.
.foo
.
.
Other SMTP Commands

**VRFY** - confirm that a name is a valid recipient

*Why might this be a problem?*

**EXPN** - expand an alias (group email address)

**TURN** - switch roles (sender <=> receiver)
**more Commands**

**SOML** - Send Or Mail

if recipient is logged in, display message on terminal, otherwise email.

**SAML** - Send and Mail

**NOOP** - send back a positive reply code

**RSET** - abort current transaction
Email messages contain many headers
- Some headers are created by UA
- Some automatically added by MTA

Every MTA adds (at least) a “Received:” header
- Spamassassin?

Some of the headers are read by (parsed) intermediate MTAs, but the content is ignored and passed on transparently
Email demo
POP – *Post Office Protocol*

- Used to transfer mail from a mail server to a UA.
POP (version 3)

- Similar to SMTP command/reply lockstep protocol
- Used to retrieve mail for a single user
  - Requires authentication
- Commands and replies are ASCII lines
  - Replies start with “+OK” or “-ERR”.
  - Replies may contain multiple lines
POP-3 Commands

- **USER** - specify username
- **PASS** - specify password
- **STAT** - get mailbox status
  - number of messages in the mailbox
- **LIST** - get a list of messages and sizes
  - One per line, termination line contains ‘.’ only
- **RETR** - retrieve a message
More POP-3 Commands

- **DELE** - mark a message for deletion from the mailbox
- **NOOP** - send back positive reply
- **RSET** - reset. All deletion marks are unmarked
- **QUIT** - remove marked messages and close the (TCP) connection
Optional Commands

- **TOP** - send header lines from messages

- **APOP** - alternative authentication
  - Message digest based on opening greeting sent from POP server
  - Requires shared secret
  - No cleartext password on network
  - Does not authenticate the server
A Pop3 Exchange

[gmporter@ramp gmporter]$ telnet fast pop3
Trying 132.239.15.4...
Connected to fast.
Escape character is '^]'.
+OK QPOP (version 3.1.2) at fast.ucsd.edu starting.
<263.1114547001@fast.ucsd.edu>
user gmporter
+OK Password required for gmporter.
pass mypasswd
+OK gmporter has 1 visible message (0 hidden) in 1761 octets.
Stat
+OK 1 1761
list
+OK 1 visible messages (1761 octets)
1 1761
Pop3 Example continued

- retr 1
  +OK 1761 octets
  Received: from mailbox7.ucsd.edu (mailbox7.ucsd.edu [132.239.1.59])
  Message-Id: <200504262023.j3QKNATg015405@smtp.ucsd.edu>
  Reply-To: <gmporter@cs.ucsd.edu>
  From: "George Porter" <gmporter@cs.ucsd.edu>
  To: <gmporter@cs.ucsd.edu>
  Subject: test
  Date: Tue, 26 Apr 2005 13:23:08 -0700
  MIME-Version: 1.0
  Content-Type: text/plain; charset="US-ASCII"
  X-Greylisting: NO DELAY (Trusted relay host);
  processed by UCSD_GL-v1.2 on mailbox7.ucsd.edu;
  X-MailScanner: PASSED (v1.2.8 34463 j3QKNATg8038256 mailbox7.ucsd.edu)
  X-Spam-Flag: Spam NO
  X-Scanned-By: milter-spamc/0.15.245 (fast.ucsd.edu [132.239.15.4]); pass=YES;
  Tue, 26 Apr 2005 13:23:12 -0700
  X-Spam-Status: NO, hits=-4.90 required=5.00

  test
  .
From: alice@crepes.fr
To: bob@hamburger.edu
Subject: Picture of yummy crepe.
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary=98766789
--98766789
Content-Transfer-Encoding: quoted-printable
Content-Type: text/plain
Dear Bob,
Please find a picture of a crepe.
--98766789
Content-Transfer-Encoding: base64
Content-Type: image/jpeg
base64 encoded data ..... 
...........................
......base64 encoded data
--98766789--

(Kevin Jeffay)