EECS 122: Introduction to Communication Networks
Homework 2

(15 points)

Due: 1999-Sep-13-Mon (start of class)

Problem 1. Suppose one of your instructors received the following spam:

```
From dogboyseven@aol.com Sat Sep 4 16:55:41 1999
Received: from cs2.CS.Berkeley.EDU (cs2.CS.Berkeley.EDU [169.229.60.56])
  by mnemosyne.CS.Berkeley.EDU (8.9.1a/) with ESMTP id QAA20836
  for <amc@mailspool.CS.Berkeley.EDU>;
  Sat, 4 Sep 1999 16:55:38 -0700 (PDT)
Received: from mail.everfaster.com (mail.everfaster.com [197.46.220.4])
  by cs2.CS.Berkeley.EDU (8.9.1a/8.6.6.Beta11) with ESMTP id LAA18735
  for <amc@cs.berkeley.edu>; Sat, 4 Sep 1999 16:55:04 -0700 (PDT)
Received: from gate.hypermoon.com (pool37.qs4w.longlink.net [217.6.1.7])
  by mail.everfaster.com (8.8.7/8.8.7) with SMTP id PAA20074;
  Sat, 4 Sep 1999 19:54:21 -0400 (EDT)
Received: from fritz.hotdogcity.com (fritz.hotdogcity.com [221.88.9.16])
  by server.big-hello.com (8.8.8/8.8.8) with SMTP id RAA04617;
  Sat, 4 Sep 1999 19:53:33 -0400 (EDT)
Received: by fritz.hotdogcity.com with Internet Mail Service (5.5.248.0)
  id Q19G494F; Sat, 4 Sep 1999 19:53:25 -0400 (EDT)
Date: Sat, 4 Sep 1999 19:53:23 -0400 (EDT)
From: Charles Lewis <clewis@hotmail.com>
To: amc@cs.berkeley.edu
Subject: You’ll never believe this!
Message-ID: <19990904195323.H8159@fritz.hotdogcity.com>
Mime-Version: 1.0
Content-Type: text/plain; charset=us-ascii

You won’t believe this, but some company just paid me to surf the web!
Check out...
```

Refer to RFC 822 for the meaning of the header fields. The first “From” line (without the colon) is created by the mail delivery agent on most Unix systems and contains the return-path from the SMTP `MAIL` command (see RFC 821). Assume that the machine that originally sent the message is dishonest and never puts its own name in the header, and all other machines through which the message passed are honest.

a) (2 points) Which machine did the message originate from? How can you tell? (Don’t bother trying to look up the names from this fictional message in the real DNS. You can do this problem offline.)

- **b) (2 points)** Which machine is unfortunately configured to relay messages that are neither coming from nor going to its domain, and is thus being used as an unwitting accomplice in propagating the spam?

- **c) (0 points)** Where should complaints be sent about each of these two machines? (See RFC 2142.)

Problem 2. (hands-on) Play the part of a mail transfer agent: Determine the mail exchanger for the domain of your own email address (see RFC 974) using the Unix command `nslookup`. Then use `telnet` to connect to the SMTP port (port 25) of the mail exchanger (enter "`telnet` *host* `25`"), and type SMTP commands to send yourself a message. [The SMTP server does *not* use the telnet protocol—the `telnet` command functions as a simple transparent pipe when connecting to ports other than the telnet port (port 23).]

Problem 3. (2 points) Suppose there is an HTTP server running on foo.bar.org, listening on port 12345, serving files from some directory (which we call the *server root*). Inside the server root is a subdirectory called "`networking`", and inside that is a file called "`What is TCP/IP?`". Give a URL for this file. (Assume the hierarchy of the directory tree is reflected in the URL. Consult RFC 2396 regarding the escaping of reserved characters. Refer to the `ascii` man page or the class web page for an ASCII code chart.)

Problem 4. (2 points) Write the HTML code for a paragraph element whose content is "I am doing EECS 122 homework." where "EECS 122" is a hyperlink to the class web page. (The latest version of the HTML specification is 4.0, but version 2.0, in RFC 1866, is simpler and sufficient for this.)

Problem 5. In SMTP, the end of a message body is indicated by a line containing only a period. In HTTP, message bodies can be binary data, so a different method is used to indicate where the body ends: the `Content-Length` header field gives the length of the body in bytes.

- **a) (1 point)** Under what circumstances might it be difficult for the server to supply this header field?

- **b) (1 point)** What provision does HTTP 1.1 (RFC 2068) make for this situation?

Problem 6. (hands-on) Play the part of a web browser: Fetch the class web page using `telnet` to connect to the HTTP port (port 80) of the server. (In order to see the header of the response, you'll need a window with a large scrollback buffer, or you can use the Unix `script` command.)

Problem 7. (2 points) Usually, if email is sent from one machine to another, the receiving mail transfer agent records the address of the sending machine in the message header (as we saw in problem 1). Suppose there exists a public FTP server that allows anyone to store and retrieve files. How can someone use this FTP server to send an arbitrary email message from their own machine without having its address recorded in the header? (Hint: look at the `PORT` command in RFC 959.) (People have used this technique in the past, but most modern FTP servers place restrictions on the `PORT` command to make it impossible.)

Problem 8. While there do exist many newsgroups that are *not* propagated to all news servers, there are many thousands of newsgroups that do indeed go to almost all servers, so for this problem let's keep the model simple and assume that every article goes to every server. Suppose there are u users, each of whom, on average, reads articles (from a server) at a rate r and posts articles (to a server) at a rate p, where $r > p$. Suppose there are n news servers, each of which, on average, can send articles at a maximum rate m, and can also receive articles at a maximum rate m. Remember that news servers are responsible for sending articles both to users and to each other. Assume that the servers successfully avoid sending duplicate articles.

a) **(2 points)** How many users can be supported? That is, what is the maximum value of u in terms of the other parameters?

b) **(0 points)** If there is only one news server, how many users can be supported?

c) **(0 points)** As the number of servers increases to infinity, the maximum number of users that can be supported approaches what upper bound?

d) **(1 point)** How many news servers get us halfway to the upper bound?