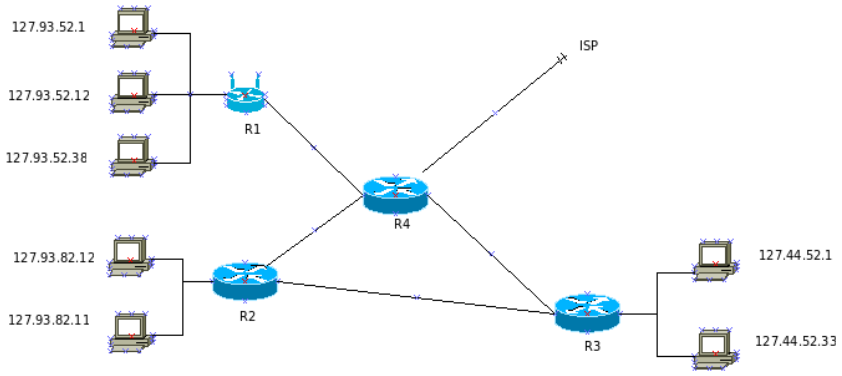
Operating Systems	Homework 5
CS 4410	Due Tuesday, August 15 (before lecture)

1. **IP** Consider the following network:



   (a) For each of the routers (R1 through R4), write an appropriate routing table.

   (b) Suppose the MTU of the wireless link between hosts and R1 is 7981 bytes, and the MTU of all other links is 1500 bytes. Suppose a UDP packet containing 12,000 bytes is sent from host 127.95.52.1 to host 127.44.52.33. Describe the IP packets that are sent along each link to handle this datagram (don't forget to account for the space taken by headers!).

2. **Application protocol** In the files `motd_server.py` and `motd_client.py`, you will implement a simple application layer protocol.

   The server maintains a single string containing the "message of the day" (MOTD). It listens for requests sent on port 44100, and should recognize three kinds of requests: `"GET"`, `"SET <new message>"`, and `"REPLACE <old message> WITH <new message>"`

   - If the server receives a `"GET"` message it should respond with the current message of the day and close the connection.
   - If it receives a `"SET"` message it should update the MOTD and respond with the string `"OK"` and then close the connection.
   - If it receives a `"REPLACE"` message, it should compare the current MOTD with the received "old message";
     - if they match, it should replace the MOTD with the new version, reply with `"OK"` and then close the connection.
     - if they do not match, it should not change the MOTD, and should reply with `"ERROR: <current message>"` and close the connection.

   If it receives an invalid message it should respond with `"RETRY"` and try to receive another request. If it receives 3 invalid messages, it should respond `"ERROR"` and close the connection.

   The client should allow the user to connect to an IP address specified on the command line, and should allow them to perform both operations (get and set).

   Your server should be able to handle multiple concurrent requests.

   Note: there is no skeleton code for this assignment; you should write and submit the entire program.

3. **TCP** Suppose that a client and server are using the protocol from the previous question; they have the following conversation:

client→server: `hello`
server→client: `RETRY`
client→server: `GET`
server→client: `I can haz cheezburger?`

(a) Describe the set of messages sent by the TCP protocol during this session.

(b) Choose two packets to drop, and describe how the session proceeds without them.