Homework 2 Problems
(Due: 10/23 Wednesday classroom)

P1. Consider an HTTP GET message below. Answer the following questions, indicating where in the HTTP GET message below you find the answer.

```
GET /index.html HTTP/1.1
Host: www.hanyang.ac.kr
User-Agent: Mozilla/5.0 (Window;U; Windows NT 5.1; en-US)
Accept: ext/xml, application/xml, application/xhtml, text/html
Accept-Language: en-us,en;q=0.5
Accept-Encoding: zip,deflate
Keep-Alive: 300
Connection: Keep-alive
```

(a) What is the URL of the document requested by the browser?

(b) What version of HTTP is the browser running?

(c) Does the browser request a non-persistent or a persistent connection?

(d) What is the IP address of the host on which the browser is running? (Trick question)

(e) What type of browser initiates this message? Why is the browser type needed in an HTTP request message?

P2. Consider a short, 10-meter link, over which a sender can transmit at a rate of 100 bits/sec in both directions. Suppose that packets containing data are 100,000 bits long, and packets containing only control (e.g., ACK or handshaking) are 300 bits long. Assume that N parallel connections each get $1/N$ of the link bandwidth. Now consider the HTTP protocol, and suppose that each downloaded object is 100 Kbits long, and that the initial downloaded object contains 20 referenced objects from the same sender. Would parallel downloads via parallel instances of non-persistent HTTP make sense in this case? Now consider persistent HTTP. Do you expect significant gains over the non-persistent case? Justify and explain your answer.

P3. Consider the scenario introduced in the previous problem. Now suppose that the link is shared by Bob with four other users. Bob uses parallel instances of non-persistent HTTP, and the other four users use non-persistent HTTP without parallel downloads.

(a) Do Bob's parallel connections help him get Web pages more quickly?

(b) If all five users open five parallel instances of non-persistent HTTP, then would Bob's parallel connections still be beneficial? Why or why not?

P4. In this problem, we use the useful *dig* tool available on Unix and Linux hosts to explore the hierarchy of DNS servers. Recall that Figure 2.21, a DNS server higher in the DNS hierarchy delegates a DNS query to a DNS server lower in the hierarchy, by sending back to the DNS client the name of that lower-level DNS server. First read the man page for *dig*, and then answer the following questions.

(a) Starting with a root DNS server (from one of the root server [a-m].root-servers.net), initiate a sequence of queries for the IP address for our department's Web server (cse.hanyang.ac.kr) by using *dig*. Show the list of the names of DNS servers in the delegation chain in answering your query.

(b) Repeat part (a) for any one popular Web sites, such as www.google.com.