Homework1 Solution

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

S1.
a) The document request was http://www.hanyang.ac.kr/index.html. The Host : field indicates the server's name and /index.html indicates the file name.

b) The browser is running HTTP version 1.1, as indicated just before the first <cr><lf> pair.

c) The browser is requesting a persistent connection, as indicated by the Connection: keep-alive.

d) This is a trick question. This information is not contained in an HTTP message anywhere. So there is no way to tell this from looking at the exchange of HTTP messages alone. One would need information from the IP datagrams (that carried the TCP segment that carried the HTTP GET request) to answer this question.

e) Mozilla/5.0. The browser type information is needed by the server to send different versions of the same object to different types of browsers.

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.

S2. Note that each downloaded object can be completely put into one data packet. Let *Tp* denote the one-way propagation delay between the client and the server.

First consider parallel downloads via non-persistent connections. Parallel download would allow 20 connections share the 100 bits/sec bandwidth, thus each gets just 5 bits/sec. Thus, the total time needed to receive all objects is given by:
(300/100+Tp + 300/100 +Tp + 300/100+Tp + 100,000/100+ Tp)
+ (300/(100/20)+Tp + 300/(100/20) +Tp + 300/(100/20)+Tp + 100,000/(100/20)+ Tp)
= 7377 + 8*Tp (seconds)

Then consider persistent HTTP connection. The total time needed is give by:
(300/100+Tp + 300/100 +Tp + 300/100+Tp + 100,000/100+ Tp)
+ 20*300/100+Tp + 20*100,000/100+ Tp
=7351 + 6*Tp (seconds)

Assume the speed of light is $300*10^6$ m/sec, then Tp=10/($300*10^6$)=0.03 microsec. Tp is negligible compared with transmission delay.

Thus, we see that the persistent HTTP does not have significant gain (less than 1 percent) over the non-persistent case with parallel download.

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?

S3. a). Yes, because Bob has more connections, so he can proportionally get more aggregate bandwidth share out of the total link bandwidth.
b) Yes, Bob still needs to perform parallel download, otherwise he will get less bandwidth share than other four users. In fact, all users might tend to open more connections in order to gain more bandwidth share.

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.

S4.

a) The following delegation chain is used for cse.hanyang.ac.kr
a.root-servers.net

First command:
dig +norecurse @a.root-servers.net any cse.hanyang.ac.kr

;; AUTHORITY SECTION:

kr.	172800	IN	NS	c.dns.kr.
kr.	172800	IN	NS	e.dns.kr.
kr.	172800	IN	NS	g.dns.kr.
kr.	172800	IN	NS	d.dns.kr.
kr.	172800	IN	NS	b.dns.kr.
kr.	172800	IN	NS	f.dns.kr.

Among all returned kr DNS servers, we send a query to the first one.
dig +norecurse @ c.dns.kr any cse.hanyang.ac.kr

hanyang.ac.kr.	86400	IN	NS	hynetm.hanyang.ac.kr.
hanyang.ac.kr.	86400	IN	NS	ansan-d.hanyang.ac.kr.

Among all two returned authoritative DNS servers, we send a query to the 2nd one.
dig +norecurse @ ansan-d.hanyang.ac.kr any cse.hanyang.ac.kr

cse.hanyang.ac.kr.	86400	IN	A	166.104.239.93

b) The answer for google.com could be:
a.root-servers.net
E.GTLD-SERVERS.NET
ns1.google.com(authoritative)