# Solution: Assignment 2 – Chapter 2

## Review Questions

**R4**. No. In a P2P file-sharing application, the peer that is receiving a file is typically the client and the peer that is sending the file is typically the server.

**R6**. You would use UDP. With UDP, the transaction can be completed in one roundtrip time (RTT) - the client sends the transaction request into a UDP socket, and the server sends the reply back to the client's UDP socket. With TCP, a minimum of two RTTs are needed - one to set-up the TCP connection, and another for the client to send the request, and for the server to send back the reply.

**R10**. A protocol uses handshaking if the two communicating entities first exchange control packets before sending data to each other. SMTP uses handshaking at the application layer whereas HTTP does not.

**R11**. The applications associated with those protocols require that all application data be received in the correct order and without gaps. TCP provides this service whereas UDP does not.

**R13**. Web caching can bring the desired content “closer” to the user, possibly to the same LAN to which the user’s host is connected. Web caching can reduce the delay for all objects, even objects that are not cached, since caching reduces the traffic on links. However, it adds a cache checking penalty (which can increase the delay but insignificantly as the cache is on the same LAN as the client.)

**R19**. Yes an organization’s mail server and Web server can have the same alias for a host name. The MX record is used to map the mail server’s host name to its IP address.

## Problems

**Problem 1**

a) F
b) T
c) F
d) F
e) F

**Problem 7**

The total amount of time to get the IP address is
$RTT_1 + RTT_2 + \boxed{?} + RTT_n$.

Once the IP address is known, $RTT_O$ elapses to set up the TCP connection and another $RTT_O$ elapses to request and receive the small object. The total response time is
$2RTT_o + RTT_1 + RTT_2 + \boxed{?} + RTT_n$

## Problem 8

a)

$$RTT_1 + \boxed{?} + RTT_n + 2RTT_o + 8 \cdot 2RTT_o$$
$$= 18RTT_o + RTT_1 + \boxed{?} + RTT_n .$$

b)

$$RTT_1 + \boxed{?} + RTT_n + 2RTT_o + 2 \cdot 2RTT_o$$
$$= 6RTT_o + RTT_1 + \boxed{?} + RTT_n$$

c) Persistent connection with pipelining. This is the default mode of HTTP.

$$RTT_1 + \boxed{?} + RTT_n + 2RTT_o + RTT_o$$
$$= 3RTT_o + RTT_1 + \boxed{?} + RTT_n .$$

Persistent connection without pipelining, without parallel connections.

$$RTT_1 + \boxed{?} + RTT_n + 2RTT_o + 8RTT_o$$
$$= 10RTT_o + RTT_1 + \boxed{?} + RTT_n .$$

## Problem 10

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 using non-persistent connections. Parallel downloads would allow 10 connections to share the 150 bits/sec bandwidth, giving each just 15 bits/sec. Thus, the total time needed to receive all objects is given by:

(200/150+*T*p + 200/150 +*T*p + 200/150+*T*p + 100,000/150+ *T*p )
+ (200/(150/10)+*T*p + 200/(150/10) +*T*p + 200/(150/10)+*T*p + 100,000/(150/10)+ *T*p )
= 7377 + 8**T*p (seconds)

Now consider a persistent HTTP connection. The total time needed is given by:

(200/150+*T*p + 200/150 +*T*p + 200/150+*T*p + 100,000/150+ *T*p )
\+ 10*(200/150+*T*p + 100,000/150+ *T*p )
=7351 + 24**T*p (seconds)

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

Thus, we see that persistent HTTP is not significantly faster (less than 1 percent) than the non-persistent case with parallel download.

## Problem 11

a) Yes, because Bob has more connections, he can get a larger share of the link bandwidth.
b) Yes, Bob still needs to perform parallel downloads; otherwise he will get less bandwidth than the other four users.

## Problem 20

We can periodically take a snapshot of the DNS caches in the local DNS servers. The Web server that appears most frequently in the DNS caches is the most popular server. This is because if more users are interested in a Web server, then DNS requests for that server are more frequently sent by users. Thus, that Web server will appear in the DNS caches more frequently.

For a complete measurement study, see:
Craig E. Wills, Mikhail Mikhailov, Hao Shang
"Inferring Relative Popularity of Internet Applications by Actively Querying DNS Caches", in IMC'03, October 2729, 2003, Miami Beach, Florida, USA