ELEC 5220/6220 Solution 2

Problem 1

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

Problem 4

a) The document request was http://gaia.cs.umass.edu/cs453/index.html. The Host : field indicates the server's name and /cs453/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.

Problem 5

a) The status code of 200 and the phrase OK indicate that the server was able to locate the document successfully. The reply was provided on Tuesday, 07 Mar 2008 12:39:45 Greenwich Mean Time.
b) The document index.html was last modified on Saturday 10 Dec 2005 18:27:46 GMT.
c) There are 3874 bytes in the document being returned.
d) The first five bytes of the returned document are : <!doc. The server agreed to a persistent connection, as indicated by the Connection: Keep-Alive field

Problem 9

a) The time to transmit an object of size *L* over a link or rate *R* is *L/R*. The average time is the average size of the object divided by *R*:

Δ = (850,000 bits)/(15,000,000 bits/sec) = .0567 sec

The traffic intensity on the link is given by $\beta\Delta$=(16 requests/sec)(.0567 sec/request) = 0.907. Thus, the average access delay is (.0567 sec)/(1 - .907) ≈ .6 seconds. The total average response time is therefore .6 sec + 3 sec = 3.6 sec.

b) The traffic intensity on the access link is reduced by 60% since the 60% of the requests are satisfied within the institutional network. Thus the average access delay is (.0567 sec)/[1 – (.4)(.907)] = .089 seconds. The response time is approximately zero if the request is satisfied by the cache (which happens with probability .6); the average response time is .089 sec + 3 sec = 3.089 sec for cache misses (which happens 40% of the time). So the average response time is (.6)(0 sec) + (.4)(3.089 sec) = 1.24 seconds. Thus the average response time is reduced from 3.6 sec to 1.24 sec.

Problem 14

SMTP uses a line containing only a period to mark the end of a message body.

HTTP uses “Content-Length header field” to indicate the length of a message body.

No, HTTP cannot use the method used by SMTP, because HTTP message could be binary data, whereas in SMTP, the message body must be in 7-bit ASCII format.

Problem 17

a) **C:** dele 1
C: retr 2
S: (blah blah …

S: ………..blah)
S: .
C: dele 2
C: quit
S: +OK POP3 server signing off

b) **C:** retr 2
S: blah blah …
S: ………..blah
S: .
C: quit
S: +OK POP3 server signing off

Problem 21

Yes, we can use dig to query that Web site in the local DNS server.
For example, "dig cnn.com" will return the query time for finding cnn.com. If cnn.com was just accessed a couple of seconds ago, an entry for cnn.com is cached in the local DNS cache, so the query time is 0 msec. Otherwise, the query time is large.

Problem 26

Yes. His first claim is possible, as long as there are enough peers staying in the swarm for a long enough time. Bob can always receive data through optimistic unchoking by other peers.
His second claim is also true. He can run a client on each host, let each client "free-ride," and combine the collected chunks from the different hosts into a single file. He can even write a small scheduling program to make the different hosts ask for different chunks of the file. This is actually a kind of Sybil attack in P2P networks.

Problem 27

Peer 3 learns that peer 5 has just left the system, so Peer 3 asks its first successor (Peer 4) for the identifier of its immediate successor (peer 8). Peer 3 will then make peer 8 its second successor.

Problem 28

Peer 6 would first send peer 15 a message, saying "what will be peer 6's predecessor and successor?" This message gets forwarded through the DHT until it reaches peer 5, who realizes that it will be 6's predecessor and that its current successor, peer 8, will become 6's successor. Next, peer 5 sends this predecessor and successor information back to 6. Peer 6 can now join the DHT by making peer 8 its successor and by notifying peer 5 that it should change its immediate successor to 6.

Problem 30

Yes, randomly assigning keys to peers does not consider the underlying network at all, so it very likely causes mismatches.

Such mismatches may degrade the search performance. For example, consider a logical path p1 (consisting of only two logical links): A→B→C, where A and B are neighboring peers, and B and C are neighboring peers. Suppose that there is another logical path p2 from A to C (consisting of 3 logical links): A→D→E→C.

It might be the case that A and B are very far away physically (and separated by many routers), and B and C are very far away physically (and separated by many routers). But it may be the case that A, D, E, and C are all very close physically (and all separated by few routers). In other words, a shorter logical path may correspond to a much longer physical path.