Q2) (a) F (b) T (c) F (d) F (e) F

Q3) Application Layer Protocols: DNS and HTTP
Transport Layer Protocols: UDP for DNS, TCP for HTTP

Q6) (a) Either the client or the server can indicate to the other that it is going to close the persistant connection. It does so by including the connection token "close" in the connection-header field of the Http request/reply.
(b) HTTP does not provide any encryption services.
(c) "Clients that use persistent connections should limit the number of simultaneous connections that they maintain to a given server. A single-user client should not maintain more than 2 connections with any server or proxy.
(d) Yes, "A client might have started to send a new request at the same time that the server has decided to close the idle connection. From the server's point of view, the connection is being closed while it was idle, but from the client's point of view, a request is in progress

Q7) The total amount of time to get the IP address is:
$RTT_1 + RTT_2 + \cdots + RTT_n$
Once IP address is known, RTT_0 elapses to set up the TCP connection and another RTT_0 elapses to request and receive the small object. The total response time is:
$2RTT_0 + RTT_1 + RTT_2 + \cdots + RTT_n$

Q8) (a) $RTT_1 + \cdots + RTT_n + 2RTT_0 + 8 \cdot 2RTT_0$
$= 18RTT_0 + RTT_1 + \cdots + RTT_n$
(b) $RTT_1 + \cdots + RTT_n + 2RTT_0 + 2 \cdot 2RTT_0$
$= 6RTT_0 + RTT_1 + \cdots + RTT_n$
(c) $RTT_1 + \cdots + RTT_n + 2RTT_0 + RTT_0$
$= 3RTT_0 + RTT_1 + \cdots + RTT_n$

Q9) (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 object divided by R: $\Delta = (850{,}000 \text{ bits})/(15{,}000{,}000 \text{ bits/sec}) = 0.0567 \text{ sec}$
The traffic intensity on the link is given by $\beta\Delta = (16 \text{ request/sec}) \cdot (0.0567 \text{ sec/request}) = 0.907$. Thus, the average access delay is $(0.0567 \text{ sec})/(1 - 0.907) \approx 0.6 \text{ sec}$. The total average response time is therefore $0.6 \text{ sec} + 3 \text{ sec} = 3.6 \text{ sec}$
(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 $(0.0567)/[1 - (0.4)(0.907)] = 0.089 \text{ sec}$
The response time is approximately zero if the request is

satisfied by the cache, the average response time is 0.08 sec + 3 sec = 3.089 sec for cache misses (which happens 40% of time). So the average response time is (0.6)(0) + (0.4)(3.089) = 1.24 sec. Thus the average response time is reduced from 3.6 sec to 1.24 sec.

(Q10) Note that each downloaded object can be completely put into one data packet. Let T_p denote the one-way propagation delay between the client and 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 recieve all objects is given by:

$= (200/150 + T_p + 200/150 + T_p + 200/150 + T_p + 100000/150 + T_p) + (200/(150/10) + T_p + 200/(150/10) + T_p + 200/(150/10) + T_p + 100000/(150/10) + T_p)$

$= 7377 + 8 * T_p$ sec

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

$= (200/150 + T_p + 200/150 + T_p + 200/150 + T_p + 100000/150 + T_p) + 10 * (200/150 + T_p + 100000/150 + T_p)$

$= 7351 + 24 * T_p$ sec

Assuming the speed of light is 300×10^6 m/s, then $T_p = 10/3 \times 10^8 \Rightarrow 0.03$ ms

T_p is therefore negligible compared with transmission delay. Thus, we see that persistant HTTP is not significantly faster (less than 1 percent) than the non-persistent case with parallel download.

Q11) (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 other four users.

Q14) SMTP uses a line containing only a period to mark the end of a message body. HTTP uses "Content-Length field" to indicate the length of a message body. No, HTTP can not 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.

Part-II

(a) Handshaking protocol is a process when one device sends a message to another device by initiating the communications with the use of communication channels. It begin to establish a connection between a computer and a device.

(b) TCP provide all application data be received in the correct data and without gaps, but UDP does not. So, HTTP, FTP, SMTP, & POP3 run on the top of TCP rather than on UDP.