1. Consider an HTTP client that wants to retrieve a Web document at a given URL. The IP address of the HTTP server is initially unknown. What transport and application-layer protocols besides HTTP are needed in this scenario?
   Application layer protocols: DNS and HTTP
   Transport layer protocols: UDP for DNS; TCP for HTTP

2. Obtain the HTTP/1.1 specification (RFC 2616). Answer the following questions:
   (a) What encryption services are provided by HTTP?
   HTTP does not provide any encryption services.
   (b) Explain the mechanism used for signalling between the client and server to indicate that a persistent connection is being closed.
   Persistent connections are discussed in section 8 of RFC 2616 (the real goal of this question was to get you to retrieve and read an RFC). A client or server indicates the connection is beng closed by including the connection-token "close" in the Connection-header field of the http request/reply.
   (c) Can the client, the server, or both signal the close of a connection?
   Sections 8.1.2 and 8.1.2.1 of the RFC indicate that either the client or the server can indicate to the other that it is going to close the persistent connection.
   (d) Can a client open three or more simultaneous connections with a given server?
   Yes. Parallel connections are allowed

3. Suppose within your Web browser you click on a link to obtain a Web page. The IP address for the associated URL is not cached in your local host, so a DNS lookup is necessary. Suppose that n DNS servers are visited before your host receives the IP address from DNS; the successive visits incur a RTT of $\mathrm{RTT}_1, \ldots, \mathrm{RTT}_n$. Further suppose the web page contains exactly two objects. Let RTT_0 denote the RTT between the local host and the server containing the object. Assume transmission time for each object i is T_i.
   (a) How much time elapses from when the client clicks on the link until the client receives the objects using persistent HTTP?
   $$T_p = RTT_1 + \ldots + RTT_n + 2RTT_o + RTT_o + T_1 + T_2 = 3RTT_o + RTT_1 + \ldots + RTT_n + T_1 + T_2$$
   (b) How much time elapses from when the client clicks on the link until the client receives the objects using non-persistent HTTP?
   $$T_p = RTT_1 + \ldots + RTT_n + 2RTT_o + 2RTT_o + T_1 + T_2 = 4RTT_o + RTT_1 + \ldots + RTT_n + T_1 + T_2$$
   (c) How much time elapses from when the client clicks on the link until the client receives the objects using non-persistent HTTP with parallel connections?
   $$T_p = RTT_1 + \ldots + RTT_n + 2RTT_o + 2RTT_o + T_1 + T_2 = 4RTT_o + RTT_1 + \ldots + RTT_n + T_1 + T_2$$

4. Consider a short 10 meter link, over which a sender can transmit at a rate of 128 bits/sec in both directions. Suppose that packets containing the data are 96,000 bits long, and packets containing only control (ACK, handshaking) are 240 bits long. Assume that N parallel connections get $1/N$ of the link bandwidth. Now consider HTTP where each object is 96,000 bits long, and the initial downloaded object references 10 additional objects from the same sender.

(a) Would parallel downloads via parallel instances of non-persistent HTTP make sense in this case?
(240/128+Tp + 240/128 +Tp + 240/128+Tp + 96,000/128+ Tp) + (240/(128/10)+Tp + 240/(128/10) +Tp + 240/(128/10)+Tp + 96,000/(128/10)+ Tp) = 8311.875 + 8*Tp (seconds)

(b) Now consider persistent HTTP. Are there significant gains over the non-persistent case?
(240/128+Tp + 240/128 +Tp + 240/128+Tp + 96,000/128+ Tp) + 10*(240/128+Tp + 96,000/128+ Tp) =8274.375 + 24*Tp (seconds)
Since the propagation speed is about 3×10^8m/s, there is no appreciable difference.

5. What is the difference from `MAIL FROM:` in SMTP and `From:` in the mail message itself?
The MAIL FROM: in SMTP is a message from the SMTP client that identifies the sender of the mail message to the SMTP server. The From: on the mail message itself is NOT an SMTP message, but rather is just a line in the body of the mail message.

6. (a) How does SMTP mark the end of a message body? Explain.
SMTP uses a line containing only a period to mark the end of a message body.

(b) How about HTTP? Explain.
HTTP uses "Content-Length header field" to indicate the length of a message body.

(c) Can HTTP use the same method as SMTP to mark the end of a message body? Explain.
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.

7. Describe at least 3 differences between the POP3 and IMAP protocols.

- POP3 downloads emails to the user device and deletes the email on the server, whereas IMAP keeps a copy of the email at the server.
- IMAP remembers state between sessions, POP3 does not.
- A user can manage folders in IMAP, this cannot be done with POP3
- POP3 requires less memory at the server.
- Many more differences.

8. Suppose that your department has a local DNS server for all computers in the department. You are an ordinary user (i.e., not a network/system administrator). Can you come up with a way to determine if an external Web site was very likely accessed from a computer in your department a couple of seconds ago? Explain

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 is 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.

9. Read RFC 959 for FTP. List all of the client commands that are supported by the RFC.
Access control commands:
USER, PASS, ACT, CWD, CDUP, SMNT, REIN, QUIT.
Transfer parameter commands:
PORT, PASV, TYPE STRU, MODE.
Service commands:
RETR, STOR, STOU, APPE, ALLO, REST, RNFR, RNTO, ABOR, DELE, RMD, MRD, PWD, LIST, NLST, SITE, SYST, STAT, HELP, NOOP.