Solutions to the Sample Questions on Application Layer

1) Why is SMTP not used for transferring e-mail messages from the recipient's mail server to the recipient's personal computer?

Answer: Whereas SMTP is a push protocol; the task of transferring e-mail messages from the recipient's mail server to the recipient's personal computer is a pull operation.

2) THIS WILL BE SOLVED IN THE FIRST TUTORIAL.

3) Why do you think DNS uses UDP, instead of TCP, for its query and response messages?

Answer: TCP involves a connection establishment phase while UDP does not. Using TCP for DNS may end up involving several TCP connections to be established since several name servers may have to be contacted to translate a name into an IP address. This imposes a high overhead in delay that is acceptable for larger transfers but not acceptable for very short messages such as DNS queries and responses. In addition, UDP affords a smaller packet size and also imposes a smaller load on name servers due to its simplicity in comparison to TCP.

4) Suppose you are sending an email from your Hotmail account to your friend, who reads his/her e-mail from his/her mail server using IMAP. Briefly describe how your email travels from your host to your friend's host. Also, what are the application-layer protocols involved?

Answer: Message is sent from your host to your mail server over HTTP. Your mail server then sends the message to your friend's mail server over SMTP. Your friend then transfers the message from his/her mail server to his/her host over IMAP.

5) In what way is instant messaging a hybrid of client-server and P2P architecture?

Answer: Instant Messaging involves the initiator to contact a centralized server to locate the address (IP address.) of the receiver: client server model. After this, the instant messaging can be peer to peer – message between the two communicating parties are sent directly between them.

6) Multiple Choice Questions – Choose one from the possible choices:

A) Suppose a client sends an HTTP request message with the *If-modified-since:* header. Suppose the object in the server has not changed since the last time that client retrieved the object. Then the server will send a response message with the status code:

- (i) 304 Not Modified
- (ii) 404 Not Found
- (iii) 200 OK
- (iv) 403 Permission Denied
- (v) None of the above

B) Which of the following is used to contain an Internet standard?

(i) RFC
(ii) IETF
(iii) DNS
(iv) PPP
(v) None of the above

C) HTTP request and response messages are not humanly readable

(i) True
(ii) False

D) All SMTP e-mail messages must be in 8-bit ASCII

(i) True
(ii) False

E) Which of the following protocols uses out-of-band signalling?

(i) HTTP
(ii) SMTP
(iii) FTP
(iv) All of the above
(v) None of the above

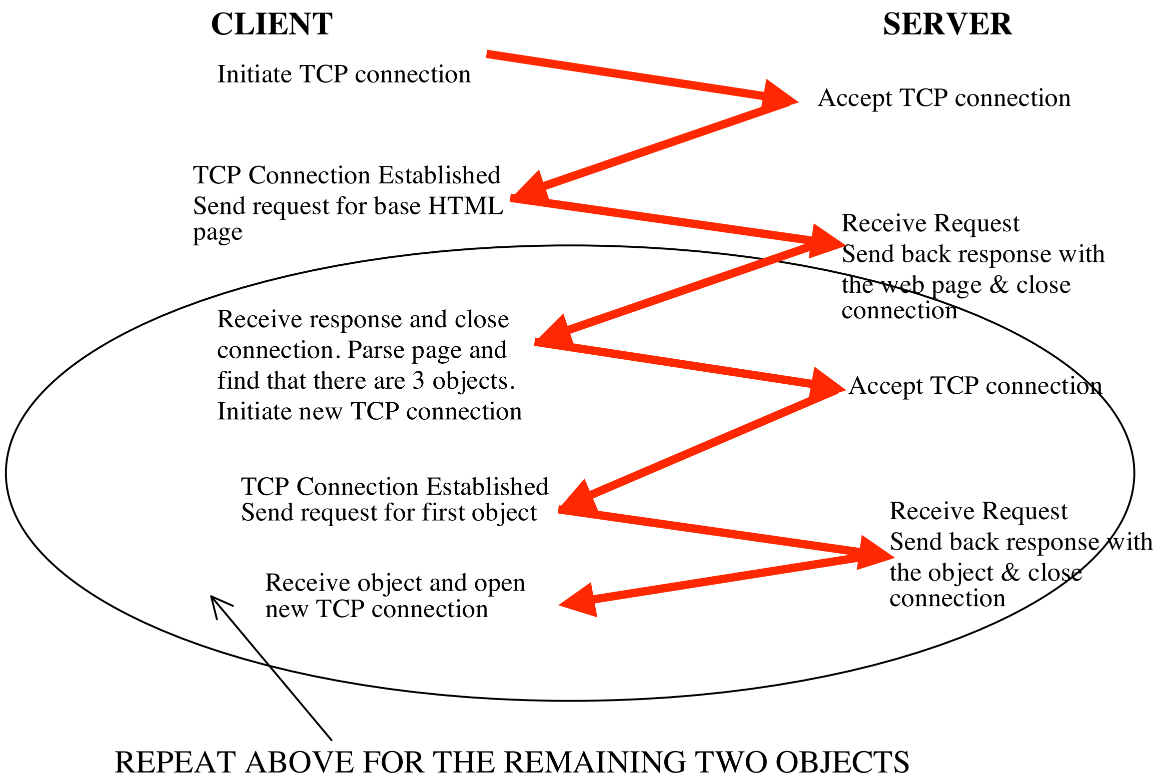
7) How can iterated DNS queries improve the overall performance?

Answer: Iterated request can improve overall performance by offloading the processing of requests from root and TLD servers to local servers. In recursive queries, root servers can be tied up ensuring the completion of numerous requests, which can results in a substantial decrease in performance. Iterated requests move that burden to local servers, and distributed the load more evenly throughout the Internet. With less work at the root servers, they can perform much faster.

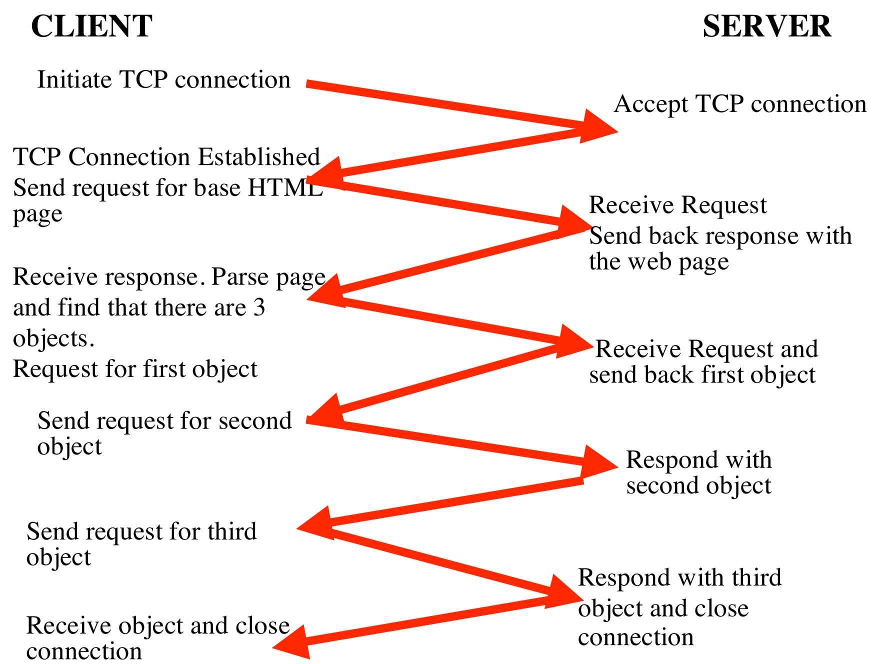
8) Suppose you needed to use HTTP to download a web page with three embedded images. Draw diagrams, similar to those from class, depicting the main interactions between the client and server when using non-persistent HTTP, persistent HTTP without pipelining, and persistent HTTP with pipelining.

Answer:

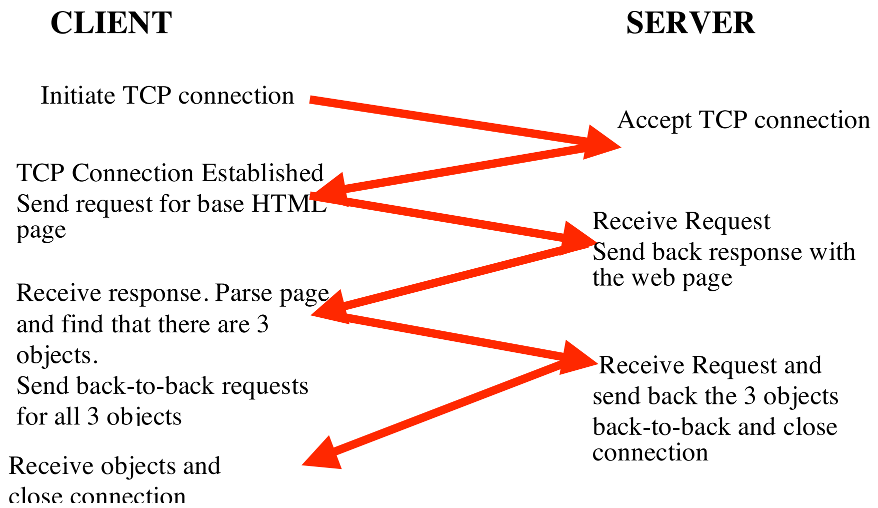
(a) Non-persistent HTTP



(b) Persistent HTTP without pipelining



(c) Persistent HTTP with pipelining



9) 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 look-up is necessary to obtain the IP address. Suppose that *n* DNS servers are visited before your host receives the IP address from DNS; the successive visits incur an RTT of RTT_1, ….., RTT_n. Further suppose that the web page associated with the link contains exactly one object, consisting of a small amount of a small amount of HTML text. Let RTT_0 denote the RTT between the local host and the server containing the object. Assuming zero transmission time of the object, how much time elapses from when the client clicks on the link until the client receives the *object*? (EXAM PROBLEM FROM PAST SESSIONS)

Answer: The total amount of time to get the IP address is

$$RTT_1 + RTT_2 + ... + RTT_n.$$

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

$$2RTT_0 + RTT_1 + RTT_2 + ... + RTT_n.$$

10) In the lecture we solved a problem, which involved finding the total delay incurred in downloading objects assuming that the web cache hit ratio was 40%. Now assume that the cache hit rate is 50% (i.e. 50% of the requests are satisfied locally from the web cache). The rest of the parameters are the same as in that problem. The average delay to download the objects is:

(a) between 1.2 and 1.3 seconds
(b) greater than two seconds
(c) between 1.0 and 1.1 seconds
(d) none of the above

Answer: In this case, 50% of the requests will be satisfied by the orign servers. Hence the utilization of the access link reduces to 50%, resulting in negligible delays (say < 10 msec).
Total average delay for unsatisfied requests = Internet delay + access delay + LAN delay < 2.01 sec

Overall Delay = 0.5*(2.01) + 0.5*(0.01) <1.01 seconds
Hence the correct answer is (c).

11) and 12) Please refer to Page 7.

13) Consider a new peer Alice that joins BitTorrent without possessing any chunks. Without any chunks, she cannot become a top-four uploader for any of the peers, since she has nothing to upload. How then will Alice get her first chunk?

Answer: Alice will get her first chunk as a result of she being selected by one of her neighbors as a result of an "optimistic unchoke," for sending out chunks to her. Recall that a peer periodically selects one of its neighbours at random as a peer for uploading irrespective of whether this neighbour is uploading data to it or not.

14) Consider distributing a file of $F = 10$ *Gbits* to N peers. The server has an upload rate of $u_s = 20Mbps$, and each peer has a download rate of $d_i = 1$ *Mbps* and an upload rate of u. For $N=10$, *100* and *1000* and $u = 200Kbps$, *600 Kbps* and *1 Mbps*, prepare a chart giving the minimum distribution time for each of the combinations of N and u for both client-server distribution and P2P distribution.

Answer: For calculating the minimum distribution time for client-server distribution, we use the following formula:

$$D_{cs} = max\ \{NF/u_s,\ F/d_{min}\}$$

Similarly, for calculating the minimum distribution time for P2P distribution, we use the following formula:

$$D_{P2P} = max\{F/u_s,\ F/d_{min},\ NF/(u_s + \sum_{i=1}^{N} u_i)\}$$

where, F = 10 Gbits = 10 * 1024 Mbits
u_s = 20 Mbps
$d_{min} = d_i$ = 1 Mbps

Client Server

		N		
		10	**100**	**1000**
	200 Kbps	10240	51200	512000
u	**600 Kbps**	10240	51200	512000
	1 Mbps	10240	51200	512000

Peer to Peer

		N		
		10	**100**	**1000**
	200 Kbps	10240	25904.3	47559.33
u	**600 Kbps**	10240	13029.6	16899.64
	1 Mbps	10240	10240	10240

11) A DHT (as we studied it) is a pure overlay network. Consequently, the assignment of keys to the peers does not take into account the underlying network topology. Do you think this may have an impact of the search performance?

Answer: Yes. The delay between successive nodes in DHT can be high because they may be separated by a long distance. It is also possible that the throughput between two successive nodes is low because the underlying network has a low bandwidth.

12) Consider the circular DHT example that we discussed in the lecture (see the relevant slides for the discussion on DHT). Explain how peer 6 would join the DHT. You may assume that peer 15 is the designated contact peer for the DHT.

Answer: The sequence of actions is:

1. Peer 6 will contact Peer 15 with a join request.
2. Peer 15, whose successor is peer 1, knows that Peer 6 should not be its successor. Peer 15 will forward the join request from Peer 6 to Peer 1.
3. Peer 1, whose successor is peer 3, knows that Peer 6 should not be its successor. Peer 1 will forward the join request from Peer 6 to Peer 3. The actions of peers 3 and 4 are identical to those of peers 15 and 1.
4. The join request will finally arrive at peer 5. Peer 5 knows that its current successor is peer 8, therefore peer 6 should become its new successor. Peer 5 will let peer 6 knows that its successor is peer 8. At the same time, peer 5 updates its successor to be peer 6.