**R1. List five nonproprietary Internet applications and the application-layer protocols that they use.**

1. File Transfer: FTP
2. Remote Login: Telnet
3. The Web: HTTP
4. BitTorrent file sharing: BitTorrent protocol
5. E-mail: SMTP

**R2. What is the difference between network architecture and application architecture?**

Network architecture refers to the organization of the communication process into layers (e.g., the five-layer Internet architecture). Application architecture, on the other hand, is designed by an application developer and dictates the broad structure of the application (e.g., client-server or P2P).

**R3. For a communication session between a pair of processes, which process is the client and which is the server?**

The process which initiates the communication is the client; the process that waits to be contacted is the server.

**R4. For a P2P file-sharing application, do you agree with the statement, “There is no notion of client and server sides of a communication session”? Why or why not?**

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.

**R5. What information is used by a process running on one host to identify a process running on another host?**

The IP address of the destination host and the port number of the socket in the destination process.

**R6. Suppose you wanted to do a transaction from a remote client to a server as fast as possible. Would you use UDP or TCP? Why?**

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.

**R7. Referring to Figure 2.4 , we see that none of the applications listed in Figure 2.4 requires both no data loss and timing. Can you conceive of an application that requires no data loss and that is also highly time-sensitive?**

One such example is remote word processing, for example, with Google docs. However, because Google docs runs over the Internet (using TCP), timing guarantees are not provided.

**R8. List the four broad classes of services that a transport protocol can provide. For each of the service classes, indicate if either UDP or TCP (or both) provides such a service.**

a) Reliable data transfer TCP provides a reliable byte-stream between client and server but UDP does not.

b) A guarantee that a certain value for throughput will be maintained Neither

c) A guarantee that data will be delivered within a specified amount of time Neither

d) Confidentiality (via encryption) Neither

**R9. Recall that TCP can be enhanced with SSL to provide process-to-process security services, including encryption. Does SSL operate at the transport layer or the application layer? If the application developer wants TCP to be enhanced with SSL, what does the developer have to do?**

SSL operates at the application layer. The SSL socket takes unencrypted data from the application layer, encrypts it and then passes it to the TCP socket. If the application developer wants TCP to be enhanced with SSL, she has to include the SSL code in the application.

**R10. What is meant by a handshaking protocol?**

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. Why do HTTP, SMTP, and POP3 run on top of TCP rather than on UDP?**

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.

**R12. Consider an e-commerce site that wants to keep a purchase record for each of its customers. Describe how this can be done with cookies.**

When the user first visits the site, the server creates a unique identification number, creates an entry in its back-end database, and returns this identification number as a cookie number. This cookie number is stored on the user’s host and is managed by the browser. During each subsequent visit (and purchase), the browser sends the cookie number back to the site. Thus the site knows when this user (more precisely, this browser) is visiting the site.

**R13. Describe how Web caching can reduce the delay in receiving a requested object. Will Web caching reduce the delay for all objects requested by a user or for only some of the objects? Why?**

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.

**R14. Telnet into a Web server and send a multiline request message. Include in the request message the If-modified-since: header line to force a response message with the 304 Not Modified status code.**

Telnet is not available in Windows 7 by default. to make it available, go to Control Panel, Programs and Features, Turn Windows Features On or Off, Check Telnet client. To start Telnet, in Windows command prompt, issue the following command

*> telnet webserverver 80*

where "webserver" is some webserver. After issuing the command, you have established a TCP connection between your client telnet program and the web server. Then type in an HTTP GET message.

**R15. List several popular messaging apps. Do they use the same protocols as SMS?**

A list of several popular messaging apps: WhatsApp, Facebook Messenger, WeChat, and Snapchat. These apps use the different protocols than SMS.

**R16. Suppose Alice, with a Web-based e-mail account (such as Hotmail or Gmail), sends a message to Bob, who accesses his mail from his mail server using POP3. Discuss how the message gets from Alice’s host to Bob’s host. Be sure to list the series of application-layer protocols that are used to move the message between the two hosts.**

The message is first sent from Alice’s host to her mail server over HTTP. Alice’s mail server then sends the message to Bob’s mail server over SMTP. Bob then transfers the message from his mail server to his host over POP3.

**R17. Print out the header of an e-mail message you have recently received. How many Received: header lines are there? Analyze each of the header lines in the message.**

**Received:** from 65.54.246.203 (EHLO bay0-omc3-s3.bay0.hotmail.com) (65.54.246.203) by mta419.mail.mud.yahoo.com with SMTP; Sat, 19 May 2007 16:53:51 -0700

**Received:** from hotmail.com ([65.55.135.106]) by bay0-omc3-s3.bay0.hotmail.com with Microsoft SMTPSVC(6.0.3790.2668); Sat, 19 May 2007 16:52:42 - 0700

**Received:** from mail pickup service by hotmail.com with Microsoft SMTPSVC; Sat, 19 May 2007 16:52:41 -0700

**Message-ID**: <BAY130-F26D9E35BF59E0D18A819AFB9310@phx.gbl>

**Received:** from 65.55.135.123 by by130fd.bay130.hotmail.msn.com with HTTP; Sat, 19 May 2007 23:52:36 GMT

**From:** "prithula dhungel"

**To:** [prithula@yahoo.com](mailto:prithula@yahoo.com)

**Bcc:**

**Subject:** Test mail

**Date:** Sat, 19 May 2007 23:52:36 +0000

**Mime-Version:**1.0

**Content-Type:** Text/html; format=flowed

**Return-Path:** [prithuladhungel@hotmail.com](mailto:prithuladhungel@hotmail.com)

**Figure: A sample mail message header**

Received: This header field indicates the sequence in which the SMTP servers send and receive the mail message including the respective timestamps. In this example there are 4 “Received:” header lines. This means the mail message passed through 5 different SMTP servers before being delivered to the receiver’s mail box. The last (forth) “Received:” header indicates the mail message flow from the SMTP server of the sender to the second SMTP server in the chain of servers. The sender’s SMTP server is at address 65.55.135.123 and the second SMTP server in the chain is by130fd.bay130.hotmail.msn.com. The third “Received:” header indicates the mail message flow from the second SMTP server in the chain to the third server, and so on. Finally, the first “Received:” header indicates the flow of the mail messages from the forth SMTP server to the last SMTP server (i.e. the receiver’s mail server) in the chain.

Message-id: The message has been given this number BAY130- F26D9E35BF59E0D18A819AFB9310@phx.gbl (by bay0-omc3- s3.bay0.hotmail.com. Message-id is a unique string assigned by the mail system when the message is first created.

From: This indicates the email address of the sender of the mail. In the given example, the sender is [prithuladhungel@hotmail.com](mailto:prithuladhungel@hotmail.com)

To: This field indicates the email address of the receiver of the mail. In the example, the receiver is “prithula@yahoo.com”

Subject: This gives the subject of the mail (if any specified by the sender). In the example, the subject specified by the sender is “Test mail”

Date: The date and time when the mail was sent by the sender. In the example, the sender sent the mail on 19th May 2007, at time 23:52:36 GMT.

Mime-version: MIME version used for the mail. In the example, it is 1.0.

Content-type: The type of content in the body of the mail message. In the example, it is “text/html”. Return-Path: This specifies the email address to which the mail will be sent if the receiver of this mail wants to reply to the sender. This is also used by the sender’s mail server for bouncing back undeliverable mail messages of mailer-daemon error messages. In the example, the return path is “prithuladhungel@hotmail.com”.

**R18. From a user’s perspective, what is the difference between the download-and-delete mode and the download-and-keep mode in POP3?**

With download and delete, after a user retrieves its messages from a POP server, the messages are deleted. This poses a problem for the nomadic user, who may want to access the messages from many different machines (office PC, home PC, etc.). In the download and keep configuration, messages are not deleted after the user retrieves the messages. This can also be inconvenient, as each time the user retrieves the stored messages from a new machine, all of non-deleted messages will be transferred to the new machine (including very old messages).

**R19. Is it possible for an organization’s Web server and mail server to have exactly the same alias for a hostname (for example, foo.com )? What would be the type for the RR that contains the hostname of the mail server?**

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.

**R20. Look over your received e-mails, and examine the header of a message sent from a user with a .edu e-mail address. Is it possible to determine from the header the IP address of the host from which the message was sent? Do the same for a message sent from a Gmail account.**

You should be able to see the sender's IP address for a user with an .edu email address. But you will not be able to see the sender's IP address if the user uses a gmail account.

**R21. In BitTorrent, suppose Alice provides chunks to Bob throughout a 30-second interval. Will Bob necessarily return the favor and provide chunks to Alice in this same interval? Why or why not?**

It is not necessary that Bob will also provide chunks to Alice. Alice has to be in the top 4 neighbors of Bob for Bob to send out chunks to her; this might not occur even if Alice provides chunks to Bob throughout a 30-second interval.

**R22. 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 other peers, since she has nothing to upload. How then will Alice get her first chunk?**

Recall that in BitTorrent, a peer picks a random peer and optimistically unchokes the peer for a short period of time. Therefore, Alice will eventually be optimistically unchoked by one of her neighbors, during which time she will receive chunks from that neighbor.

**R23. What is an overlay network? Does it include routers? What are the edges in the overlay network?**

The overlay network in a P2P file sharing system consists of the nodes participating in the file sharing system and the logical links between the nodes. There is a logical link (an “edge” in graph theory terms) from node A to node B if there is a semipermanent TCP connection between A and B. An overlay network does not include routers.

**R24. CDNs typically adopt one of two different server placement philosophies. Name and briefly describe them.**

One server placement philosophy is called Enter Deep, which enter deep into the access networks of Internet Service Providers, by deploying server clusters in access ISPs all over the world. The goal is to reduce delays and increase throughput between end users and the CDN servers. Another philosophy is Bring Home, which bring the ISPs home by building large CDN server clusters at a smaller number of sites and typically placing these server clusters in IXPs (Internet Exchange Points). This Bring Home design typically results in lower maintenance and management cost, compared with the enter-deep design philosophy.

**R25. Besides network-related considerations such as delay, loss, and bandwidth performance, there are other important factors that go into designing a CDN server selection strategy. What are they?**

Other than network-related factors, there are some important factors to consider, such as load-balancing (clients should not be directed to overload clusters), diurnal effects, variations across DNS servers within a network, limited availability of rarely accessed video, and the need to alleviate hot-spots that may arise due to popular video content.

Reference paper: Torres, Ruben, et al. "Dissecting video server selection strategies in the YouTube CDN." The 31st IEEE International Conference on. Distributed Computing Systems (ICDCS), 2011.

Another factor to consider is ISP delivery cost – the clusters may be chosen so that specific ISPs are used to carry CDN-to-client traffic, taking into account the different cost structures in the contractual relationships between ISPs and cluster operators.

**R26. In Section 2.7, the UDP server described needed only one socket, whereas the TCP server needed two sockets. Why? If the TCP server were to support n simultaneous connections, each from a different client host, how many sockets would the TCP server need?**

With the UDP server, there is no welcoming socket, and all data from different clients enters the server through this one socket. With the TCP server, there is a welcoming socket, and each time a client initiates a connection to the server, a new socket is created. Thus, to support n simultaneous connections, the server would need n+1 sockets.

**R27. For the client-server application over TCP described in Section 2.7 , why must the server program be executed before the client program? For the client-server application over UDP, why may the client program be executed before the server program?**

. For the TCP application, as soon as the client is executed, it attempts to initiate a TCP connection with the server. If the TCP server is not running, then the client will fail to make a connection. For the UDP application, the client does not initiate connections (or attempt to communicate with the UDP server) immediately upon execution.