Tutorial 2 Solutions

Instructions

1. Form ad-hoc groups of 2 to 3 students to solve this week’s exercise.
2. Each group must answer the following review Q’s and problems
3. Each group will use shared google docs to work with all group members and tutor. The document must include the group member’s names and the tutorial sheet number.

Review Questions

1. Q2-1. Assume we add a new protocol to the application layer. What changes do we need to make to other layers?

   Solutions: No changes are needed. The new protocol needs to use the services provided by one of the transport-layer protocols.

2. Q2-3. In the client-server paradigm, explain why a server(process) should be running all the time; While a client can initiate its request process when it is needed.

   Solution: A server should always be on because a client may need to access it at any time. A client is normally the initializer of the connection; it can be run when it is needed.

3. Q2-7. A new application is to be designed using the client-server paradigm. If only small messages need to be exchanged between the client and the server without the concern for message being lost or corrupted, what transport-layer protocol do you recommend?

   Solution: In this case, UDP is more appropriate because it does not have the overhead of TCP in connection establishment and teardown.

4. Q2-9. A source socket address is a combination of an IP address and a port number. Explain what each section identifies.

   *Solution: The **IP address** identifies the **source computer**; the **port number** identifies the **source process**.*

5. Q2-10. Explain how a client process finds the IP address and the port number to be inserted in a remote socket address.

   Solution: The client should either know the IP address of the server being communicated with or should know the name of the server (URL) and use the DNS to map the name to the IP address. The client should also know the standard well-known (TCP/UDP) port number of the corresponding server process.

6. Q2-13. Alice has a video clip that Bob is interested in getting it; Bob has another video clip that Alice is interested in getting it. Bob creates a web page and runs an HTTP server. How can Alice get Bob's clip? How can Bob get Alice's clip?

   *Solution: If Bob posts his clip on his website, Alice can, get it by running an HTTP client (a browser) using a **GET** message. Since Alice is not running an HTTP server, she needs to use the **PUT command and post her clip on Bob’s web site**.*

7. Q2-17. FTP uses two separate well-known port numbers for control and data connection. Does this mean that two separate TCP connections are created for exchanging control information and data?

   *Solution: The answer is yes. Two separate connections are needed for setup and teardown to use FTP. FTP uses **two TCP connections** for communication.*

   - *Executing an FTP port connection through a client is a two-stage process requiring the use of two different ports. Once the user enters the name of the server and the login credentials in the authorization fields of the FTP client, the FTP connection is established and the FTP control port of the FTP server (**the default tcp port for sending commands is 21**) is opened.*

   - *Then a second connection to the server is made by the client, followed by a response of the FTP server from the port for sending data (**the default data sending tcp port is 20**), when the real file transfer actually begins.*

8. Q2-18. FTP uses the services of TCP for exchanging control information and data transfer. Could FTP have used the services of UDP for either of these two connections used in previous question? Explain.

   Solution: Simple Answer is NO!

   *FTP definitely **cannot** use the services of UDP for control connection because the client and the server need to be connected during the whole session using a three-way TCP handshake.*

   *Since UDP is **not** a connection-oriented protocol, it cannot do this task. FTP does not use the services of UDP during data transmission for another reason. A file to be transferred may be too large to fit in a single user datagram (UDP packet).*

   In addition, UDP is not a reliable transport-layer protocol; for file transfer, reliability is an important issue. TCP is more appropriate for this purpose.

9. Q2-21. In FTP, if the client needs to retrieve one file from the server site and store one file on the server site, how many control connections and how many data transfer connections are needed?

   ***Solution**: (**YES,** i.e. upload and download a file from the same server). The task can be done using only **one control connection**, but **two data-transfer connections are needed**, **one for retrieving** and **one for storing**. Although the data-transfer connection is a two-way connection, one is used for data transfer, the other for acknowledging.*

10. Q2-23. In FTP, can a server get the list of the files or directories from the client?

    Solution: The answer is no. Only the client can get the list of files from the server. In the client-server paradigm, the request is from the client; the server only can respond.

11. Q2-33. Alice has been on a long trip without checking her e-mail. She then finds out that she has lost some e-mails or attachments her friends claim they have sent to her. What can be the problem?

    Solution: Mail servers normally allocate a limited amount of storage for each mailbox. If the e-mails or attachments are not retrieved, the mailbox may become full and some old e-mails may be discarded.

12. Q2-35. The TELNET application has no commands such as those found in FTP or HTTP to allow the user to do something such as transfer a file or access a web page. In what way can this application be useful?

    Solution: TELNET allows a ***host to log into a remote computer*** *that can offer application programs. After the user logs in, she can use any services provided by the remote computer. For example, the user can create a program in any computer language supported by the remote computer, compile it, run it, and see the result.*

13. (P2-5). Draw a diagram to show the use of a proxy server that is part of the client network:

    a. Show the transactions between the client, proxy server, and the target server when the **<u>response is stored</u>** in the proxy server.

    b. Show the transactions between the client, proxy server, and the target server when the **<u>response is not stored</u>** in the proxy server.

    Solution: The following shows a simple example.

    - *In part-a, the request can be responded to by the proxy server.*
    - *In part-b, the proxy needs to send the request to the true server. When the response is received, the proxy server saves it in the* ***cache*** *for future use, and then sends it to the client.*

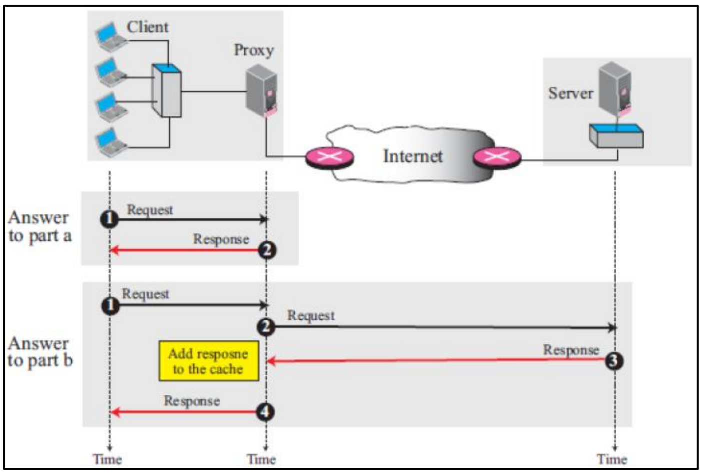

Figure shows the solution for a. & b.

14. (P2-6.) In Chapter 1, we mentioned that the TCP/IP suite, unlike the OSI model, has no presentation layer. But an application-layer protocol can include some of the features defined in this layer if needed. Does HTTP have any presentation layer features?

    Solution: HTTP provides some presentation features, using the request and response headers. For example, HTTP messages can define the format and the language of the messages exchanged, which is a kind of presentation.

15. (P2.8) In Chapter 1, we mentioned that the TCP/IP suite, unlike the OSI model, has no session layer. However, an application-layer protocol can include some of the features defined in this layer if needed. Does HTTP have any session-layer features?

    Solution: When HTTP operates in ***persistent-connection*** *mode, it is actually implementing a* ***session*** *that can last for a period of time. This is simulating the* ***session layer*** *defined in the* ***OSI*** *model.*