CS305 Homework2

Content: Chapter 2 and Chapter 3 of *Computer Networking -- A Top Down Approach 7th*

1. What is the difference between *MAIL FROM* : in SMTP and *From* : in the mail message itself?

2. How does SMTP mark the end of a message body? How about HTTP? Can HTTP use the same method as SMTP to mark the end of a message body? Explain.

3. Suppose you can access the caches in the local DNS servers of your department. Can you propose a way to roughly determine the Web servers (outside your department) that are most popular among the users in your department? Explain.

4. Consider distributing a file of F = 15Gbits to *N* peers. The server has an upload rate of us = 30 Mbps, and each peer has a download rate of di = 2Mbps and an upload rate of *u*. For *N* = 10,100, and 1000 and *u* = 300Kbps, 700Kbps, and 2Mbps, 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.

5. Consider a DASH system for which there are *N* video versions (at *N* different rates and qualities) and *N* audio versions (at *N* different rates and qualities). Suppose we want to allow the player to choose at any time any of the *N* video versions and any of the *N* audio versions.

a. If we create files so that the audio is mixed in with the video, so server sends only one media stream at given time, how many files will the server need to store (each a different URL)?

b. If the server instead sends the audio and video streams separately and has the client synchronize the streams, how many files will the server need to store?

6. Install and compile the Python programs TCPClient and UDPClient on one host and TCPServer and UDPServer on another host.

a. Suppose you run TCPClient before you run TCPServer. What happens? Why?

b. Suppose you run UDPClient before you run UDPServer. What happens? Why?

c. What happens if you use different port numbers for the client and server sides?

Here are the codes for both sides of the applications:

UDPClient.py

```
from socket import *
serverName = 'hostname'
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_DGRAM)
message = raw_input('Input lowercase sentence:')
clientSocket.sendto(message.encode(),(serverName, serverPort))
modifiedMessage, serverAddress = clientSocket.recvfrom(2048)
print(modifiedMessage.decode())
clientSocket.close()
```

UDPServer.py

```
from socket import *
serverPort = 12000
serverSocket = socket(AF_INET, SOCK_DGRAM)
serverSocket.bind(('', serverPort))
print("The server is ready to receive")
while True:
    message, clientAddress = serverSocket.recvfrom(2048)
```

```
    modifiedMessage = message.decode().upper()
    serverSocket.sendto(modifiedMessage.encode(), clientAddress)
```

TCPClient.py

```
from socket import *
serverName = 'servername'
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_STREAM)
clientSocket.connect((serverName, serverPort))
sentence = raw_input('Input lowercase sentence:')
clientSocket.send(sentence.encode())
modifiedSentence = clientSocket.recv(1024)
print('From Server: ', modifiedSentence.decode())
clientSocket.close()
```

TCPServer.py

```
from socket import *
serverPort = 12000
serverSocket = socket(AF_INET, SOCK_STREAM)
serverSocket.bind(('', serverPort))
serverSocket.listen(1)
print('The server is ready to receive')
while True:
    connectionSocket, addr = serverSocket.accept()
    sentence = connectionSocket.recv(1024).decode()
    capitalizedSentence = sentence.upper()
    connectionSocket.send(capitalizedSentence.encode())
    connectionSocket.close()
```

7. Consider a TCP connection between Host A and Host B. Suppose that the TCP segments traveling from Host A to Host B have source port number *x* and destination port number *y*. What are the source and destination port numbers for the segments traveling from Host B to Host A?

8. Describe why an application developer might choose to run an application over UDP rather than TCP.

9. UDP and TCP use 1's complement for their checksums. Suppose you have the following three 8-bit bytes: 01010011, 01100110, 01110100. What is the 1's complement of the sum of these 8-bit bytes? (Note that although UDP and TCP use 16-bit words in computing the checksum, for this problem you are being asked to consider 8-bit sums.) Show all work. Why is it that UDP takes the 1's complement of the sum; that is, why not just use the sum? With the 1's complement scheme, how does the receiver detect errors? Is it possible that a 1-bit error will go undetected? How about a 2-bit error?