National University of Singapore

School of Computing

CS2105  **Tutorial 3** Question paper

***To students:***

1. Launch your browser and open its network diagnostic tool (e.g. press F12 if you use Chrome on Windows, or Cmd + Opt + I for Mac). Then click the “Network” tab to observe network communication.

Copy-and-paste the following URL in the address bar of your browser:

<http://tiny.cc/atupaz>

Enter your choice and press the “Submit” button.

1. Look at the entry named “formResponse”. What is the HTTP request method issued?

Answer: POST

1. Briefly explain when HTTP POST and GET methods are used.

Answer: The GET request is used when the client is sending a message to the server to provide it an information it is looking for. The POST request is used when the server has received the GET request and sending back the information to the client the relevant data required.

1. **[KR, Chapter 2, P21]** 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 determine if an external Web site was likely accessed from a computer in your department a couple of seconds ago? Explain.

Anwer: Yes we can. By using the dig command, if the query time to find the website is 0ms, the website is stored in the DNS cache and has been queried before, else if the query time is large, it has not been visited

1. **[Modified from KR, Chapter 2, P31]** You are given 4 programs: **TCPEchoServer.py**, **TCPEchoClient.py**, **UDPEchoServer.py** and **UDPEchoClient.py**.
2. Suppose you run **TCPEchoClient** before you run **TCPEchoServer**. What happens? Why?

Answer: If you run TCPEchoClient first, it will try to establish a connection to the server, however since there is no server available, there will be no connection made.

1. Suppose you run **UDPEchoClient** before you run **UDPEchoServer**. What happens? Why?

Answer: Since UDP does not requre the establish a TCP connection with the server, everything will work fine.

1. **[KR, Chapter 3, R7]** Suppose a process in Host C has a UDP socket with port number 6,789. Suppose both Host A and Host B each sends a UDP segment to Host C with destination port number 6,789. Will both of these segments be directed to the same socket at Host C? If so, how will the process at Host C know that these two segments originated from two different hosts?

Answer: Yes both segments will be directed to the same socket at Host C. For each segment received, the IP addresses of each Host A and B are provided by the OS, allowing Host C to identify the sockets which contains values non-identical source addresses.

1. **[Modified from KR, Chapter 3, P4]**
2. Suppose you have the following 2 bytes: **01011100** and **01100101**. What is the 1s complement of the sum of these 2 bytes?

Answer: sum = 11000001

Complement = 00111110

1. Suppose you have the following 2 bytes: **11011010** and **01100101**. What is the 1s complement of the sum of these 2 bytes?

Answer: Sum = 100111111 = 01000000

Complement = 10111111

(Note: UDP and TCP use 16-bit words in computing their checksums. For simplicity you are asked to consider 8-bit checksums in this problem).

1. **[Modified from KR, Chapter 3, P5]** Suppose that UDP receiver computes the checksum for the received UDP segment and finds that it matches the value carried in the checksum field. Can the receiver be absolutely certain that no bit errors have occurred? You may use Q5 as an example to explain.

Answer: No we cannot do so, even tho the checksum is correct, there is a possibility that the receiver cannot detect the error present within the packet. Sometimes, even though the checksum is correct, some of the value of the original checksum might have flipped but still result in the same answer to occur. Since the checksum is not able to detect this, they cannot acertain if there is no bit error.

1. **[KR, Chapter 3, R9]** In our rdt protocols, why did we need to introduce sequence numbers?

Answer: This is to allow the server to identify the packet which it is sending, allowing to know which packet to send in case there is no ACK signal or NAK signal, requiring the sender to resend the packet to the receiver.