

COMPUTER NETWORKS (CS F303)  
SECOND SEMESTER 2021-22  
Assignment #1  
(Wireshark and Network Programming)

Date: 06-03-2022

Submission Deadline: 5:00 PM

MM: 25

Attempt the Assignment with full integrity. Plagiarized submissions will not be evaluated and will attract NC grade in the course. Follow the submission instructions written at the end.

Q.1 In this question, you have to build a simple client-server database application using TCP sockets. Write and submit the code for the client and server as **client.c** and **server.c** respectively. Your code must be in C language.

**[18 Marks]**

The application which you are developing is a simple **key-value (<int, char\*>)** store.

- The client takes user input to get/put/delete keys and passes these requests over the server.
- The server maintains a data structure of key-value pairs in a file (**database.txt**) as the persistent database and returns a suitable reply to the server.
- When the request from the client is of the form "**put key value**", the server must store the key-value pair in its data structures, provided the key does not already exist. Put requests on an existing key must return an error message.
- When the client sends a request of the form "**get key**", the value stored against the key must be returned to the client. If the key is not found, a suitable error message must be returned by the server.
- The client should be able to issue a delete request (on an existing key) as follows: "**del key**".
- For all requests that succeed, the server must either return the value (in case of **get**), or the string "**OK**" (in case of **put/del**).
- You may assume that all keys are integers and the corresponding values are strings. When the user inputs "**Bye**", the server should reply "**Goodbye**" and close the client connection.
- Upon receiving this message from the server, your client program terminates. However, the server must keep the database in a file for subsequent run of client program.
- **Note that your server must be able to communicate with multiple active clients at a time, as we will test your code with multiple clients concurrently. It may be better for you to write and debug code for a server that handles one client at a time, and then move on to the version with multiple concurrent clients. When multiple clients talk to the server, one client should be able to see the data stored by the other clients.**
- **Submit the error free code which can be compiled using gcc. Non compiled code will not be evaluated. Add appropriate comments in your code.**

Below is a sample execution of the client, when the server is running on the local host.

```
Enter request: put 1 Shreeram
Server replied: OK
Enter request: put 1 Rubal
Server replied: Key already exists
Enter request put 2 Laxmi
Server replied: OK
Enter request: get 1
Server replied: Shreeram
Enter request: del 1
Server replied: OK
Enter request: get 1
Server replied: Key not found
Enter request: del 5
Server replied: Key not found
Enter request: Bye
Server replied: Goodbye
```

## Q.2 Traffic capturing and analysis using Wireshark.

[07 Marks]

Use Wireshark to capture packets from your Network Card. Here is what you need to do:

- i. Close all opened running windows and applications (web browser, email program and other applications that typically send data to the Internet). This will minimize the amount of traffic you are going to capture.
- ii. Open Wireshark program, then open web browser and start capturing packets on your active network card.
- iii. Type [www.google.com](http://www.google.com) In the address bar (URL bar) of your web browser and search for anything you want. Browse top two-2-3 results given by google search. Further, click on URLs and images in the web pages. In summary, you are supposed browse the web for about a minute and capture network traffic to be analyzed in in this assignment. Stop Wireshark capture and save it as **your\_BITS\_id.pcap** file (*you have to upload this file. Your answers will not be evaluated without pcap file.*).
- iv. Close the browser completely.
- v. Open the pcap file in the Wireshark and start analyzing packets and protocols.

Answer following questions. Provide the details of steps used by you to get the answer and also the screen shots of the outcome of these results to justify your answer in a single file named as **your\_BITS\_id.pdf**.

- 1) What is the duration of your packet capture in seconds? What about the start and end time of the capture expressed in hh:mm:ss?
- 2) How long did it take from when the HTTP GET message was sent until the HTTP OK reply was received for the webpages (at least 3) you visited in your web browser?
- 3) What is the Internet (IP) address of the URLs you visited and what is the Internet address of your computer?
- 4) What is the IP address of the DNS server you are connecting to?
- 5) List the application layer protocols that you see in protocols field that are using UDP and TCP respectively.
- 6) Locate TCP handshake segments and find the sequence number of SYN, SYN+ACK and ACK messages of all the TCP connections made by your computer.
- 7) Find out all incoming (received by your machine) http traffic.
- 8) Find out the list of all TCP connections which have been reset. Provide appropriate reason for connection reset.
- 9) List all TCP segments which are send and received by your machine having header length more than 20 bytes. Give the appropriate reason for header length larger than the default size.
- 10) List all the duplicate ACK TCP segments.
- 11) Provide the sequence number of any one out-of-order TCP segment captured in your trace file.
- 12) How many number of HTTP request (i.e., GET and POST) messages did your browser send?
- 13) Find out all the traffic between your machine and a particular (of your choice) web site (IP address).
- 14) Calculate the throughput of all the TCP connection involved in question 13.

### Submission Instructions:

Create a single zip file (**assignment.zip**) of **client.c**, **server.c**, **your\_BITS\_id.pcap**, **your\_BITS\_id.pdf** and upload on NALANDA-AWS using the specified link before 5:00 PM.

Note: Double check before uploading your files so that you do not miss out any file. No other mode of submission will be considered.

XX-OO-XX