

# CS304– Computer Network

Instructor : Shashi Shekhar Jha (shashi@iitrpr.ac.in)

---

## Lab Assignment - 2 | Due on 16/03/2020 2400 Hrs | 50 Marks

### Submission Instructions:

All submission is through google classroom in one zip file. In case you face any trouble with the submission, either contact the TAs or you can post your queries in the classroom.

The programming language to be used to code the assignment must be **Python**.

Your submission must be your original work. Do not indulge in any kind of plagiarism or copying. Abide by the honour and integrity code to do your assignment. It is okay to discuss the problems with your peers but all implementations should be your own work.

As mentioned in the class, late submissions will attract penalties.

**Penalty Policy:** There will be a penalty of **20%** for every 24 Hr delay in the submission. E.g. for first 24 Hr delay the penalty will be 20%, for submission with a delay of >24 Hr and < 48 Hr, the penalty will be 40% and so on.

### You submission must include:

- A legible PDF document with all your answers to the assignment problems, stating the reasoning and output.
- A folder named as 'code' containing the scripts for the assignment along with the other necessary files to run yourcode.
- Your code must be properly commented.
- A README file explaining how to execute your code.

### Naming Convention:

Name the ZIP file submission as follows:

YourName\_rollnumber\_Assignmentnumber.zip

E.g. if your name is ABC, roll number is 2017csx1234 and submission is for lab1 then you should name the zip file as: ABC\_2017csx1234\_lab1.zip

## Framing, Flow and Errors

=====

In the last assignment, you created a peer-to-peer network of multiple nodes which were able to communicate with one-another by exchanging messages (there were a total of 6 Nodes A, B, C and D, a client node and a server node).

Let us extend the capabilities of that peer-to-peer network to transfer a file of moderate size from one node to another. You need to perform an application level framing of the file in order to transfer it from one node to another. Fix the size of the frames to be 100 KB or you can take frame size as input from the user (in KBs).

Let us say when the authentication is successful, the client needs to upload a file (like assignment submission) to the server. The server however directs all the frames from a user (authenticated client) to the node which maintains the username and password for that particular user. Moreover, for the purpose of more system reliability, we use two servers nodes in the network. Both the servers are all the same except for their own IP addresses and port numbers. In addition, these servers also cause random delays (the minimum delay is the RTT between client and server nodes) in the forwarding of frames to the destination nodes.

The client knows about both the servers and for each message/frame, it chooses one of the servers alternatively.

For the file transfer, here is a sample text file: Sakoontala or The Lost Ring: <http://www.gutenberg.org/cache/epub/12169/pg12169.txt> However, you are free choose any other of your favourite books' text file from the [Gutenberg project](#) as samples for this assignment. You must test your implementation with 2-3 sample files of different sizes.

Further, to ensure an error free delivery of the frames, you should implement a **checksum** based error detection method for the frames. The checksum should be transmitted with each frame. In case, the received frames get corrupted during the transmission, such frames must be retransmitted. Basically, you have to implement Stop-and-Wait ARQ protocol.

As you would not note any error in the transmission in the LAN of your Lab, create a mechanism that corrupts the frames with some probability  $p$ . Take the value of  $p$  as an input from the user. You can set  $p = 0.1$  as default value.

At the ultimate destination of each file, the data in the frames must be combined in a particular order so that the original text file can be recovered. Only when the complete file is recovered, the destination node should send a message of successful data transfer to the client node.

Note: You must display all the intermediate message exchanges on the consoles of all the nodes of your peer-to-peer network.

Topic (in that order)	Marks
Correct transfer of file with framing	20
Server replica with random delays	10
Checksum and Stop-and-Wait ARQ	20