**COMPUTER NETWORKS**

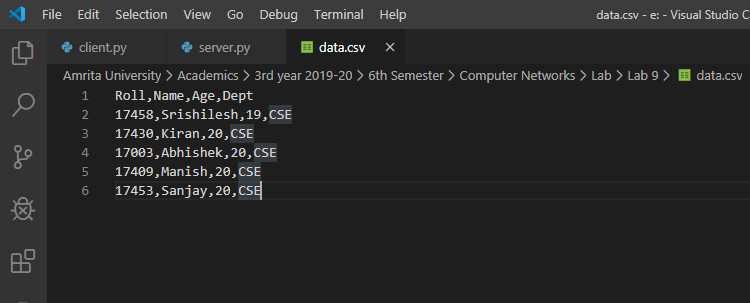
EVALUATION LAB 9

CONGESTION CONTROL

Name: Srishilesh P S

Roll No.: CB.EN.U4CSE17458

Data.csv



Assumptions:

# Timing for each operation

open\_time = 3

close\_time = 3

req\_time = 1

flag\_time = 1

rtt = open\_time + close\_time

Flags:

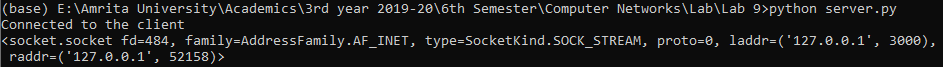
ACK – 0

SYN – 0

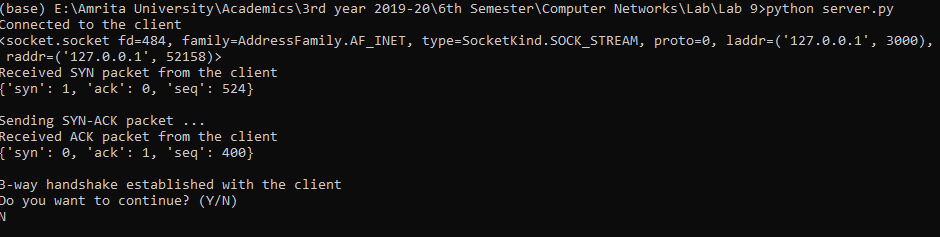
Window size – Accepted from user

Checksum for each data – 0

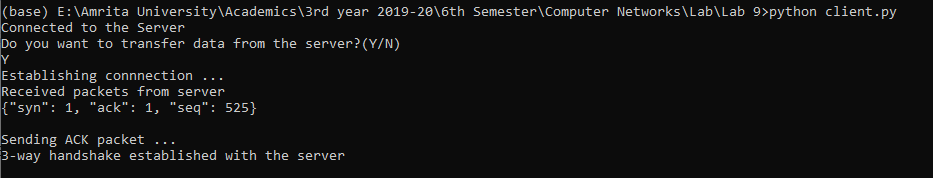
1. Server program: Connected with client



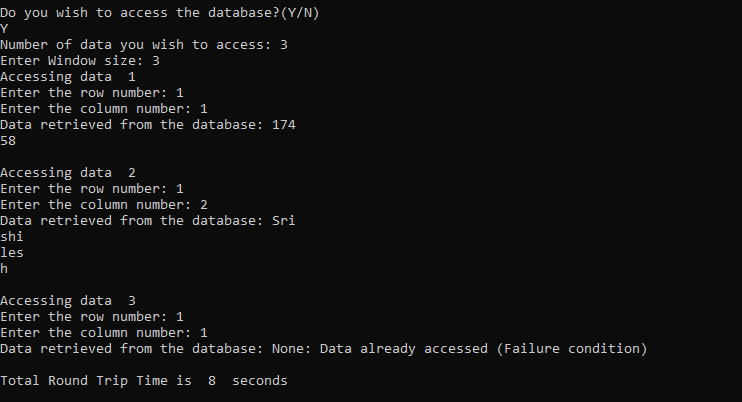
1. Server program: 3 way handshake



1. Client program: 3 way handshake

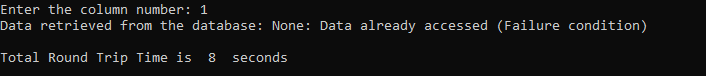


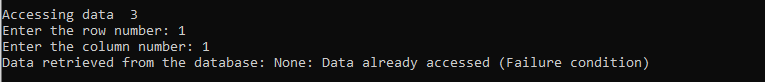
1. Client user requesting data
2. Request made by client (last condition)

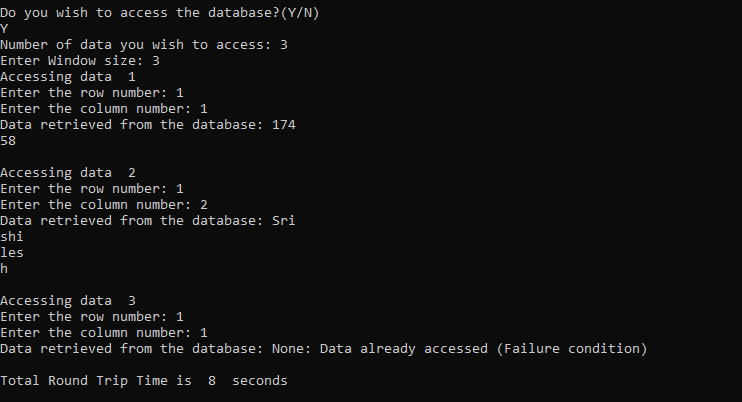


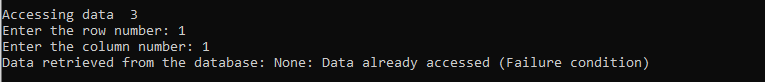
Inputs:  
Number of requests to be made  
Window size  
Row and column number

1. Round Trip Time



1. Accessing same data by client:   
     
   
2. Display according to Window size: (Accessing data 2)



1. Failure condition  
     
   

Server.py

import socket

import json

import pandas as pd

import numpy as np

data = pd.read\_csv("E:/Amrita University/Academics/3rd year 2019-20/6th Semester/Computer Networks/Lab/Lab 9/data.csv") # Read data from the file

data\_flag = np.zeros((5,4))     # Checksum flags

# Timing for each operation

open\_time = 3

close\_time = 3

req\_time = 1

flag\_time = 1

rtt = open\_time + close\_time

s = socket.socket() # Create socket

s.bind(('',3000))   # Bind socket

s.listen(5)         # Listen socket connection

while True:

    c, addr = s.accept()        # Accept the connection

    print("Connected to the client")

    print(c)

    c.send(bytes("200","utf-8"))

    msg = c.recv(1024).decode("utf-8")

    tcp = json.loads(msg)

    print("Received SYN packet from the client")

    print(tcp)

    print()

    tcp['ack']+=1

    tcp['seq']+=1

    rtt+=flag\_time

    print("Sending SYN-ACK packet ...")

    c.send(bytes(json.dumps(tcp),"utf-8"))

    msg = c.recv(1024).decode("utf-8")

    tcp = json.loads(msg)

    print("Received ACK packet from the client")

    print(tcp)

    print()

    rtt+=flag\_time

    print("3-way handshake established with the client")

    no = int(c.recv(1024).decode("utf-8"))

    window = int(c.recv(1024).decode("utf-8"))

    temp\_rtt = rtt

    for i in range(no):

        row = int(c.recv(1024).decode("utf-8"))

        col = int(c.recv(1024).decode("utf-8"))

        if(data\_flag[row][col]==0):

            if(row>0 and row<6 and col>=0 and col<=3):

                info = data.iloc[row-1][col-1]

                c.send(bytes(str(info),"utf-8"))

                data\_flag[row][col] = 1

            else:

                c.send(bytes("Data not found","utf-8"))

        else:

            msg = "None: Data already accessed (Failure condition)"

            c.send(bytes(msg,"utf-8"))

        temp\_rtt+=req\_time

    c.send(bytes(str(rtt),"utf-8"))

    print("Do you want to continue? (Y/N)")

    x = input()

    if(x == 'y' or x == 'Y'):

        continue

    else:

        break

Client.py

import socket

import json

import random

s = socket.socket()

s.connect(('127.0.0.1',3000))

tcp = {'syn':0, 'ack':0, 'seq':0 }

while True:

    conf = s.recv(1024).decode("utf-8")

    if(conf == "200"):

        print("Connected to the Server")

        print("Do you want to transfer data from the server?(Y/N)")

        x = input()

        print("Establishing connnection ... ")

        if(x == 'Y' or x == 'y'):

            r1 = random.randint(0,1000)

            tcp['seq'] = r1

            tcp['syn'] = 1

            msg = json.dumps(tcp)

            s.send(bytes(msg,"utf-8"))

            msg = s.recv(1024).decode("utf-8")

            tcp\_ = json.loads(msg)

            if(tcp\_['syn'] == 1 and tcp\_['ack'] == 1 and tcp\_['seq'] == tcp['seq']+1):

                print("Received packets from server")

                print(msg)

                print()

                tcp\_['syn'] = 0

                tcp\_['seq'] = random.randint(0,1000)

                print("Sending ACK packet ...")

                s.send(bytes(json.dumps(tcp\_),"utf-8"))

                print("3-way handshake established with the server")

                print()

                print("Do you wish to access the database?(Y/N)")

                resp = input()

                if(resp=='N' or resp=='n'):

                    break

                else:

                    no = int(input("Number of data you wish to access: "))

                    s.send(bytes(str(no),"utf-8"))

                    window = int(input("Enter Window size: "))

                    s.send(bytes(str(window),"utf-8"))

                    for i in range(no):

                        print("Accessing data ",(i+1))

                        row = input("Enter the row number: ")

                        col = input("Enter the column number: ")

                        s.send(bytes(row,"utf-8"))

                        s.send(bytes(col,"utf-8"))

                        print("Data retrieved from the database: ",end="")

                        recv\_data = str(s.recv(1024).decode("utf-8"))

                        if("None" in recv\_data):

                            print(recv\_data)

                            print()

                            continue

                        recv\_data\_len = len(str(recv\_data))

                        i = 0

                        while i<recv\_data\_len:

                            print(recv\_data[i:i+window])

                            i+=window

                        print()

                print("Total Round Trip Time is ",s.recv(1024).decode("utf-8")," seconds")

            else:

                print("Error!")

    break