**Assignment 2**

Question 1

Imagine two processes communicating across a network. One process running in your local system is a web browser and the other process running in the remote system is the web server. Once a connection is established between the web browser and web server, the server’s current date and time have to be displayed in the web browser. Write a suitable program for this scenario.

import java.io.\*;

import java.net.\*;

class DateClient

{

public static void main(String args[]) throws Exception

{

Socket soc=new Socket(InetAddress.getLocalHost(),5217);

BufferedReader in=new BufferedReader(new InputStreamReader(soc.getInputStream() ));

System.out.println(in.readLine());

}

}

import java.net.\*;

import java.io.\*;

import java.util.\*;

class DateServer

{

public static void main(String args[]) throws Exception

{

ServerSocket s=new ServerSocket(5217);

while(true)

{

System.out.println("Waiting For Connection ...");

Socket soc=s.accept();

DataOutputStream out=new DataOutputStream(soc.getOutputStream());

out.writeBytes("Server Date: " + (new Date()).toString() + "\n");

out.close();

soc.close();

}

}

}

Question 2

A) Stop and Wait

#include <iostream>

#include <time.h>

#include <cstdlib>

#include <ctime>

#include <unistd.h>

using namespace std;

class timer

{

private:

unsigned long begTime;

public:

void start()

{

begTime = clock();

}

unsigned long elapsedTime()

{

return ((unsigned long)clock() - begTime) / CLOCKS\_PER\_SEC;

}

bool isTimeout(unsigned long seconds)

{

return seconds >= elapsedTime();

}

};

int main()

{

int frames[] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};

unsigned long seconds = 5;

srand(time(NULL));

timer t;

cout << "Sender has to send frames : ";

for (int i = 0; i < 10; i++)

cout << frames[i] << " ";

cout << endl;

int count = 0;

bool delay = false;

cout << endl

<< "Sender\t\t\t\t\tReceiver" << endl;

do

{

bool timeout = false;

cout << "Sending Frame : " << frames[count];

cout.flush();

cout << "\t\t";

t.start();

if (rand() % 2)

{

int to = 24600 + rand() % (64000 - 24600) + 1;

for (int i = 0; i < 64000; i++)

for (int j = 0; j < to; j++)

{

}

}

if (t.elapsedTime() <= seconds)

{

cout << "Received Frame : " << frames[count] << " ";

if (delay)

{

cout << "Duplicate";

delay = false;

}

cout << endl;

count++;

}

else

{

cout << "---" << endl;

cout << "Timeout" << endl;

timeout = true;

}

t.start();

if (rand() % 2 || !timeout)

{

int to = 24600 + rand() % (64000 - 24600) + 1;

for (int i = 0; i < 64000; i++)

for (int j = 0; j < to; j++)

{

}

if (t.elapsedTime() > seconds)

{

cout << "Delayed Ack" << endl;

count--;

delay = true;

}

else if (!timeout)

cout << "Acknowledgement : " << frames[count] - 1 << endl;

}

} while (count != 10);

return 0;

}

B) Go Back N

#include <iostream>

#include <ctime>

#include <cstdlib>

using namespace std;

int main()

{

int nf, N;

int no\_tr = 0;

srand(time(NULL));

cout << "Enter the number of frames : ";

cin >> nf;

cout << "Enter the Window Size : ";

cin >> N;

int i = 1;

while (i <= nf)

{

int x = 0;

for (int j = i; j < i + N && j <= nf; j++)

{

cout << "Sent Frame " << j << endl;

no\_tr++;

}

for (int j = i; j < i + N && j <= nf; j++)

{

int flag = rand() % 2;

if (!flag)

{

cout << "Acknowledgment for Frame " << j << endl;

x++;

}

else

{

cout << "Frame " << j << " Not Received" << endl;

cout << "Retransmitting Window" << endl;

break;

}

}

cout << endl;

i += x;

}

cout << "Total number of transmissions : " << no\_tr << endl;

return 0;

}