**07/11/2021 CN SPOT 9 2019103573**

**IMPLEMENTING THE NAGGLE’S ALGORITHM**

**SERVER**

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <unistd.h>

#include <sys/types.h>

#include <sys/socket.h>

#include <netinet/in.h>

#include <arpa/inet.h>

#define PORT 4040

int main()

{

    int socketfd = 0, clientfd = 0;

    struct sockaddr\_in host\_addr, client\_addr;

    socklen\_t length = sizeof(struct sockaddr\_in);

    char buffer[128], ackdata[20];

    socketfd = socket(AF\_INET, SOCK\_STREAM, 0);

    if (socketfd < 0)

    {

        fprintf(stderr, "Error in creating socket.\n");

        return -1;

    }

    host\_addr.sin\_family = AF\_INET;

    host\_addr.sin\_port = htons(PORT);

    inet\_pton(AF\_INET, "127.0.0.1", &host\_addr.sin\_addr);

    if (bind(socketfd, (struct sockaddr \*)&host\_addr, length) < 0)

    {

        fprintf(stderr, "Error in binding socket to port.\n");

        return -1;

    }

    if (listen(socketfd, 5) < 0)

    {

        fprintf(stderr, "Error in listening on %s:%d.\n", inet\_ntoa(host\_addr.sin\_addr),

                ntohs(host\_addr.sin\_port));

        return -1;

    }

    fprintf(stdout, "Listening on %s:%d.\n", inet\_ntoa(host\_addr.sin\_addr), ntohs(host\_addr.sin\_port));

    while (1)

    {

        int clientfd = accept(socketfd, (struct sockaddr \*)&host\_addr, &length);

        if (clientfd < 0)

        {

            fprintf(stderr, "Error in accepting connection.\n");

            continue;

        }

        int limit;

        fprintf(stdout, "Accepted connection.\n");

        recv(clientfd, &limit, sizeof(int), 0);

        fprintf(stdout, "%d\n", limit);

        fflush(stdout);

        int packet = 1;

        char recvbuff[2];

        while (1)

        {

            int j, temp;

            fprintf(stdout, "Receiving : ");

            for (j = 0; j < packet; j++)

            {

                recv(clientfd, recvbuff, sizeof(recvbuff), 0);

                if (strncmp(recvbuff, "$", sizeof("$")) == 0)

                {

                    break;

                }

                fprintf(stdout, "%s", recvbuff);

            }

            fprintf(stdout, "\nEnd of stream. %d packets.\n\n", j);

            char ackdata[10];

            temp = sprintf(ackdata, "ACK %d", j);

            ackdata[temp] = '\0';

            fprintf(stdout, "Sending acknowledgment for %d packets.\n", j);

            send(clientfd, ackdata, strlen(ackdata) + 1, 0);

            if (j < packet)

            {

                close(clientfd);

                break;

            }

            packet = limit;

        }

    }

    close(socketfd);

    return 0;

}

**CLIENT**

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <unistd.h>

#include <sys/types.h>

#include <sys/socket.h>

#include <netinet/in.h>

#include <arpa/inet.h>

#define PORT 4040

int main()

{

    int socketfd = 0;

    struct sockaddr\_in host\_addr;

    socklen\_t length = sizeof(struct sockaddr\_in);

    socketfd = socket(AF\_INET, SOCK\_STREAM, 0);

    if (socketfd < 0)

    {

        fprintf(stderr, "Error in creating socket.\n");

        return -1;

    }

    host\_addr.sin\_family = AF\_INET;

    host\_addr.sin\_port = htons(PORT);

    inet\_pton(AF\_INET, "127.0.0.1", &host\_addr.sin\_addr);

    if (connect(socketfd, (struct sockaddr \*)&host\_addr, length) < 0)

    {

        fprintf(stderr, "Error in connecting to server.\n");

        return -1;

    }

    fprintf(stdout, "Connection established.\n");

    float rtt, delay;

    char input[128];

    fprintf(stdout, "Enter Round Trip Time : ");

    scanf("%f", &rtt);

    fprintf(stdout, "Enter Uniform Delay : ");

    scanf("%f", &delay);

    fprintf(stdout, "Enter Message : ");

    scanf("%s", input);

    int value = (int)(rtt / delay), trips = 0;

    float totalRTT = 0;

    send(socketfd, &value, sizeof(int), 0);

    int i = 0, packet = 1;

    while (i < strlen(input))

    {

        int j, temp;

        char msg[2], recvbuff[10];

fprintf(stdout, "Sending : ");

        for (j = 0; j < packet && i < strlen(input); j++, i++)

        {

            msg[0] = input[i];

            msg[1] = '\0';

            fprintf(stdout, "%s", msg);

            send(socketfd, msg, sizeof(msg), 0);

        }

        if (i >= strlen(input))

            send(socketfd, "$", strlen("$") + 1, 0);

        fprintf(stdout, "\nStream sent. %d packets.\n\n", j);

        packet = value;

        char ackdata[10];

        temp = sprintf(ackdata, "ACK %d", j);

        ackdata[temp] = '\0';

        recv(socketfd, recvbuff, sizeof(recvbuff), 0);

        if (strncmp(recvbuff, ackdata, strlen(ackdata)) == 0)

        {

            fprintf(stdout, "Acknowledgement Received for %d packets.\n\n", atoi(&recvbuff[4]));

            totalRTT += rtt;

            trips += 1;

        }

    }

    fprintf(stdout, "Total Round Trips : %d\n Total Time : %f.\n", trips, totalRTT);

    return 0;

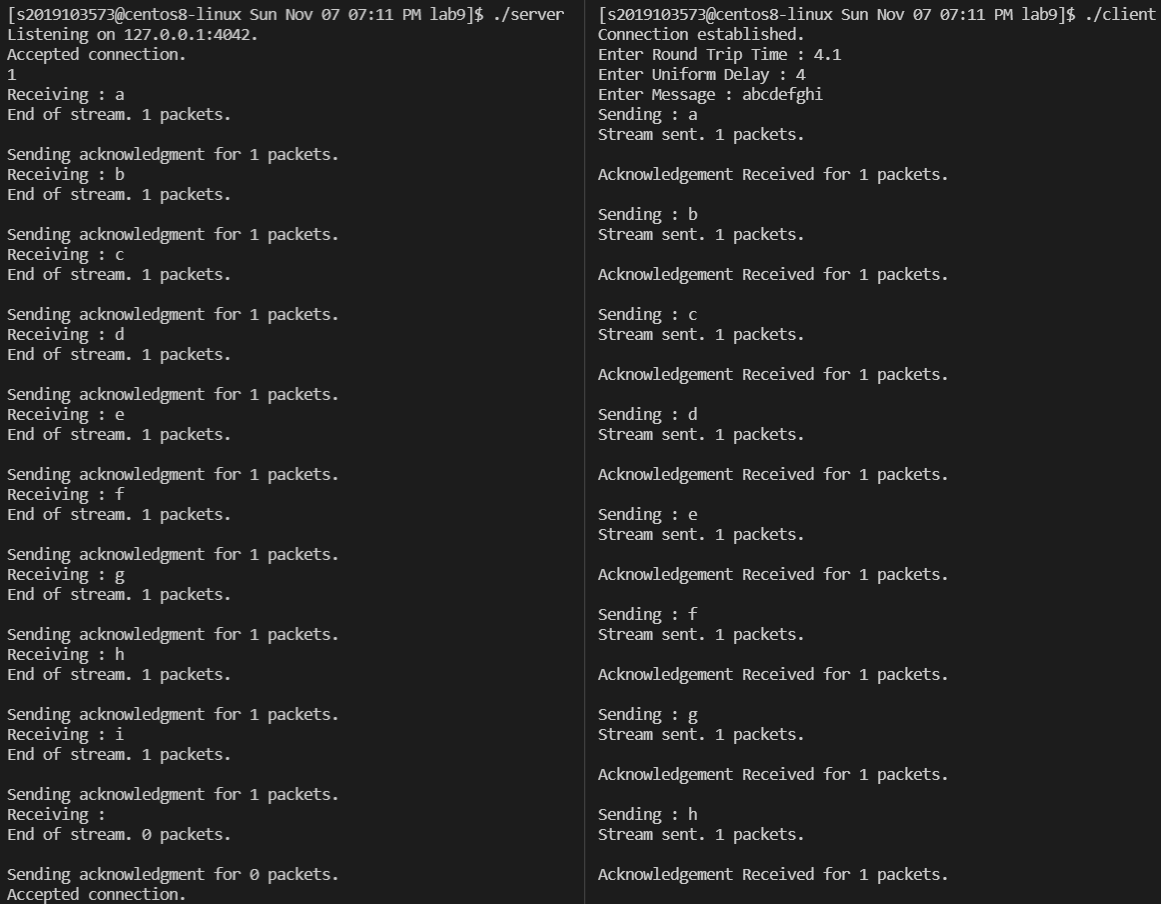
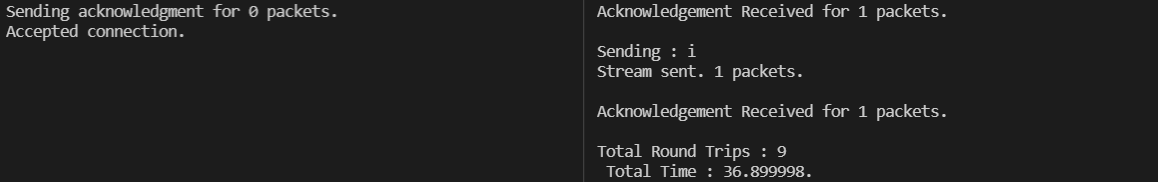
}

**OUTPUT :-**

**Message** : abcdefghi

**Round Trip Time** : 4.1

**Uniform Delay** : 4



**Total Round Trips** : 9

**Total Time** : 36.8998 ms