Sahil Khose

180953218

CCE-B

C4 batch

**LAB 5:**

**Q1:**

**Server:**

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <sys/socket.h>

#include <sys/types.h>

#include <netinet/in.h>

#include <string.h>

#include <arpa/inet.h>

int main(){

int sockfd, newsockfd, retval, recvbytes, sendbytes;

char buff[50];

struct sockaddr\_in server, client;

printf("Hi\n");

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

if(sockfd == -1){

printf("Creation Error\n");

exit(0);

}

server.sin\_family = AF\_INET;

server.sin\_port = htons(3212);

server.sin\_addr.s\_addr = htonl(INADDR\_ANY);

retval = bind(sockfd, (struct sockaddr\*) &server, sizeof(server));

if(retval == -1){

printf("Binding Error\n");

exit(0);

}

printf("Socket Binded\n");

retval = listen(sockfd, 5);

if(retval == -1){

printf("Listening Error\n");

exit(0);

}

printf("Socket Listening\n");

pid\_t child;

socklen\_t clilen;

int connection = 0;

char str[50];

str[0] = '\0';

while(1){

clilen = sizeof(client);

newsockfd = accept(sockfd, (struct sockaddr\*) &client, &clilen);

if(newsockfd == -1){

printf("Accepting Error\n");

close(sockfd);

exit(0);

}

printf("Socket Accepting\n");

connection++;

if(connection > 2){

printf("Connection exceeds Max Limit\n");

FILE \*fptr1, \*fptr2;

if((fptr1 = fopen("new.txt", "r")) == NULL){

close(newsockfd);

close(sockfd);

exit(0);

}

char temp[100];

char str[100];

while(fgets(temp, 100, fptr1) != NULL){

strcpy(str, temp);

printf("%s\n", str );

}

if(fptr1){

fclose(fptr1);

}

if((fptr2 = fopen("ip.txt", "r")) == NULL){

close(newsockfd);

close(sockfd);

exit(0);

}

char temp2[100];

char str2[100];

while(fgets(temp2, 100, fptr2) != NULL){

strcpy(str2, temp2);

printf("%s\n", str2 );

}

if(fptr2){

fclose(fptr2);

}

close(newsockfd);

close(sockfd);

exit(0);

}

printf("%s\n", str);

if((child = fork()) == 0){

close(sockfd);

recvbytes = recv(newsockfd, buff, sizeof(buff), 0);

if(recvbytes == -1){

printf("Receiving Error\n");

close(newsockfd);

exit(0);

}

puts(buff);

strcat(str, buff);

strcat(str, " ");

FILE \*fp;

fp = fopen("ip.txt", "a");

char ip[INET\_ADDRSTRLEN];

inet\_ntop(AF\_INET, &client.sin\_addr, ip, sizeof(ip));

fputs(ip, fp);

fputs(" ", fp);

fclose(fp);

FILE \*fp1;

fp1 = fopen("new.txt", "a");

fputs(buff, fp1);

fputs(" ", fp1);

fclose(fp1);

close(newsockfd);

}

}

close(sockfd);

exit(0);

}

**Client 1:**

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <sys/socket.h>

#include <sys/types.h>

#include <netinet/in.h>

#include <string.h>

#include <arpa/inet.h>

int main(){

int sockfd, recvbytes, sentbytes, retval;

char buff[50];

struct sockaddr\_in server;

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

if(sockfd == -1){

printf("Creation Error\n");

exit(0);

}

server.sin\_family = AF\_INET;

server.sin\_port = htons(3212);

server.sin\_addr.s\_addr = inet\_addr("127.0.0.1");

retval = connect(sockfd, (struct sockaddr\*)&server, sizeof(server));

if(retval == -1){

printf("Connection Error\n");

close(sockfd);

exit(0);

}

printf("Connection Estabilished\n");

strcpy(buff, "Institute of");

puts(buff);

sentbytes = send(sockfd, buff, sizeof(buff), 0);

if(sentbytes == -1 ){

printf("Send Error\n");

close(sockfd);

exit(0);

}

close(sockfd);

exit(0);

}

**Client 2:**#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <sys/socket.h>

#include <sys/types.h>

#include <netinet/in.h>

#include <string.h>

#include <arpa/inet.h>

int main(){

int sockfd, recvbytes, sentbytes, retval;

char buff[50];

struct sockaddr\_in server;

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

if(sockfd == -1){

printf("Creation Error\n");

exit(0);

}

server.sin\_family = AF\_INET;

server.sin\_port = htons(3212);

server.sin\_addr.s\_addr = inet\_addr("127.0.0.1");

retval = connect(sockfd, (struct sockaddr\*)&server, sizeof(server));

if(retval == -1){

printf("Connection Error\n");

close(sockfd);

exit(0);

}

printf("Connection Estabilished\n");

strcpy(buff, "Technology");

puts(buff);

sentbytes = send(sockfd, buff, sizeof(buff), 0);

if(sentbytes == -1 ){

printf("Send Error\n");

close(sockfd);

exit(0);

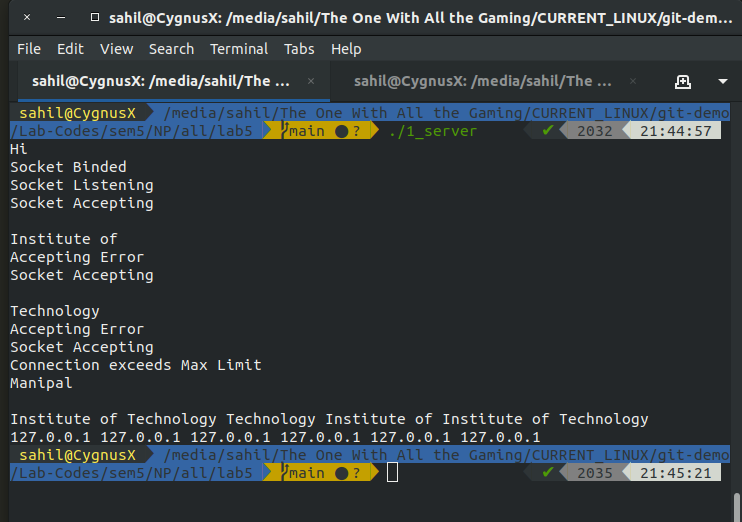
}

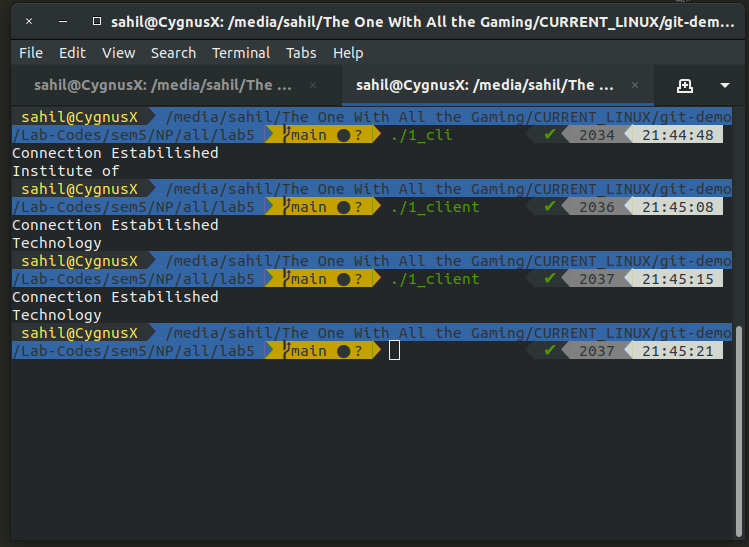
close(sockfd);

exit(0);

}

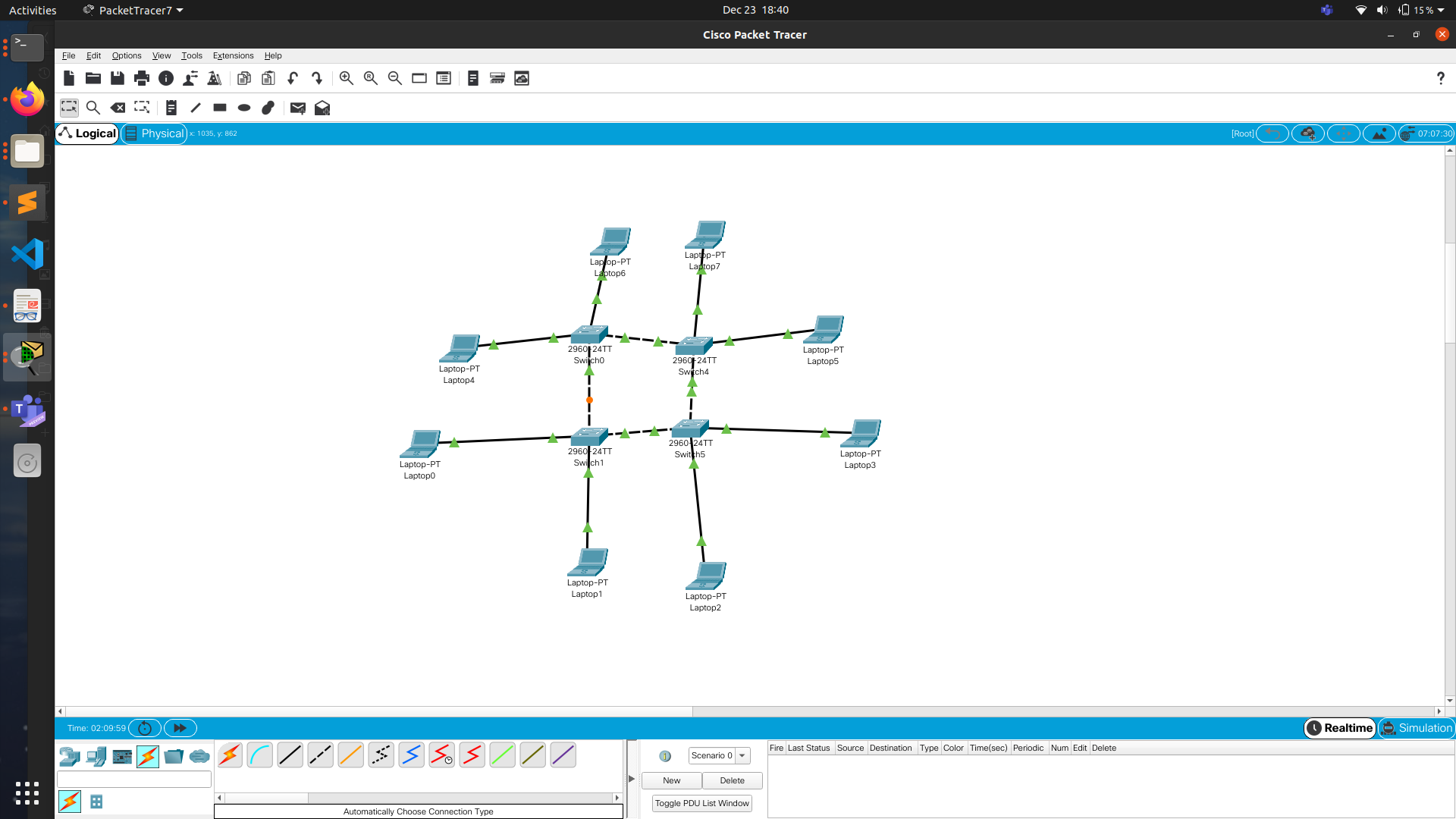
**Output:**

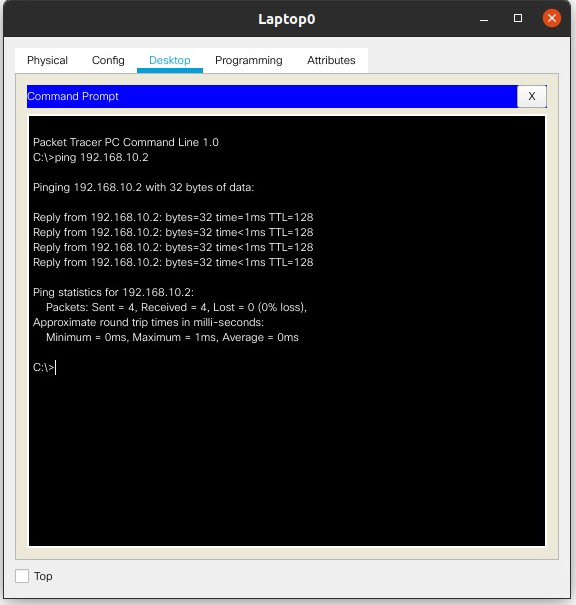
****

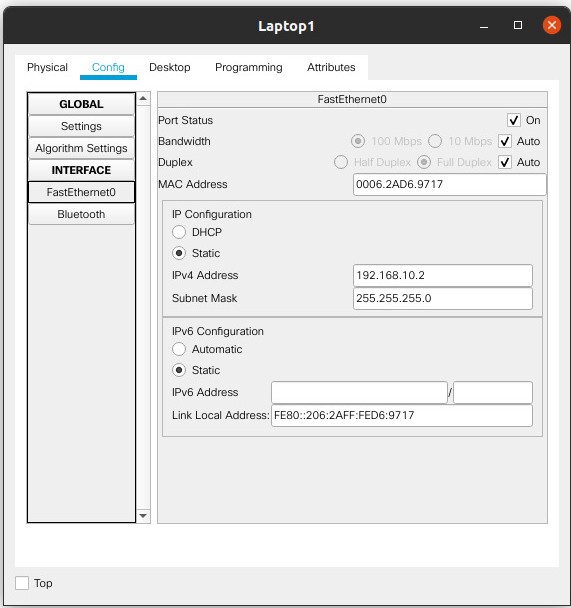
****

**LAB 8:**

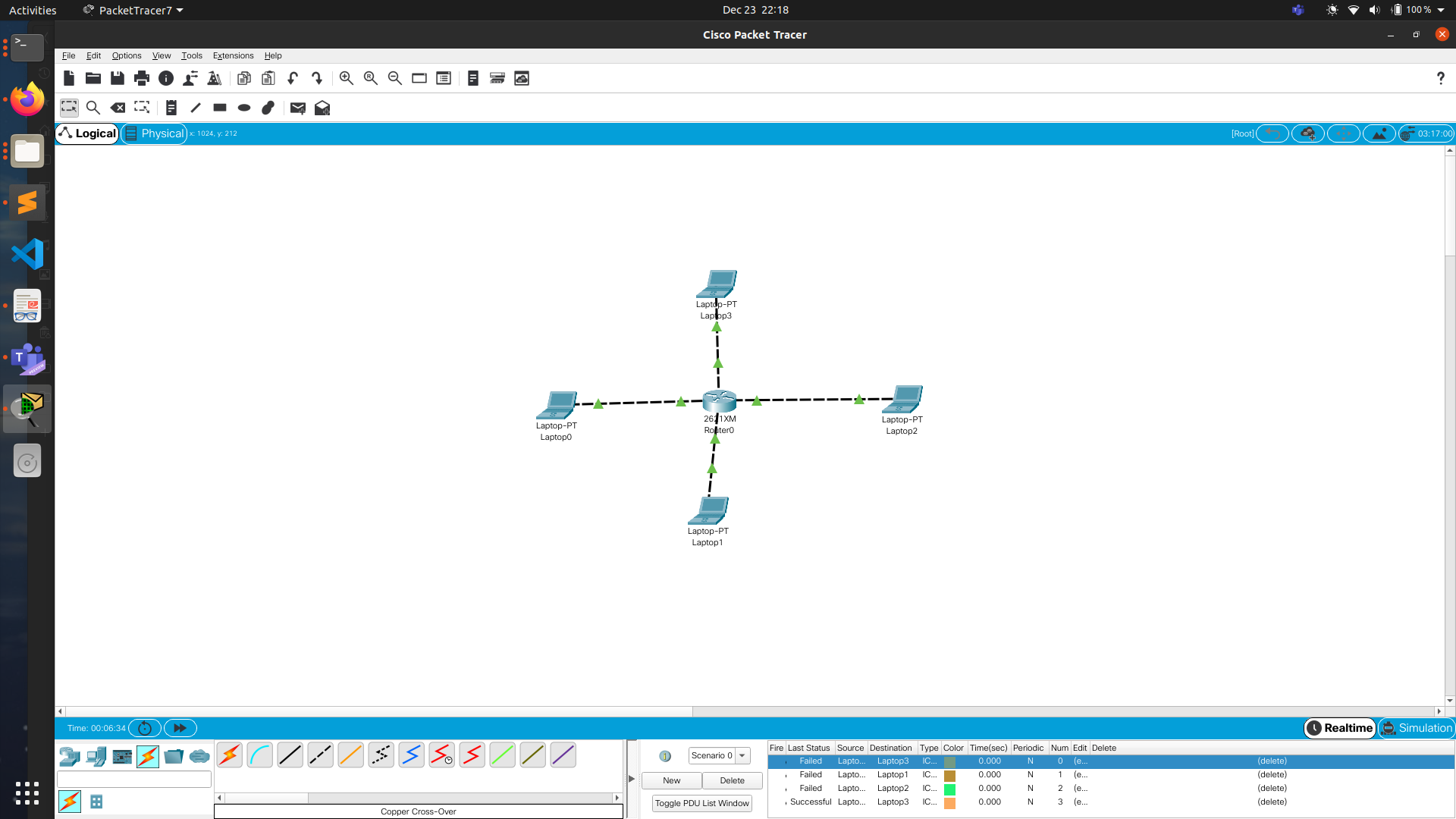
**Q1:**

****

****

****

**Q2:**

****

**CLI:**

en

conf t

int fa0/0

ip add 192.168.10.1 255.255.255.0

no shutdown

exit

int fa0/1

ip add 192.168.20.1 255.255.255.0

no shutdown

exit

int fa1/0

ip add 192.168.30.1 255.255.255.0

no shutdown

exit

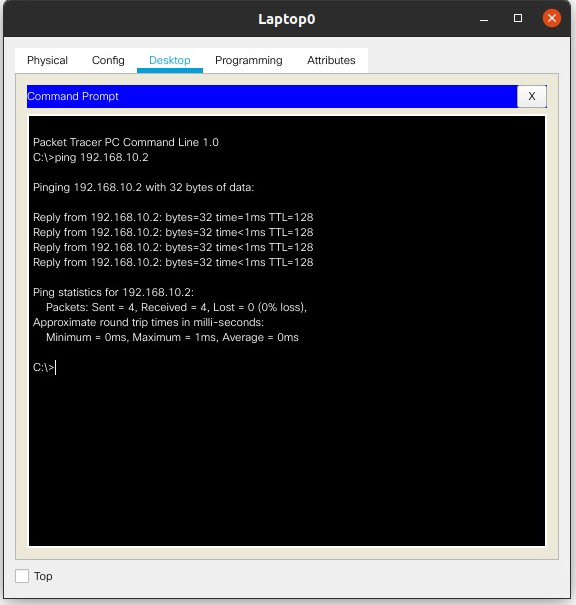
int fa1/1

ip add 192.168.140.1 255.255.255.0

no shutdown

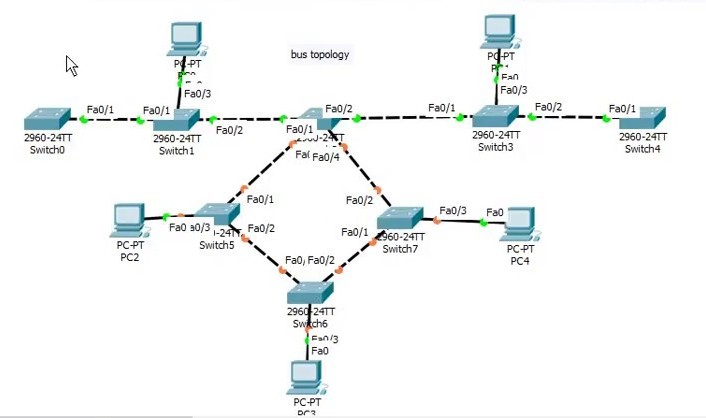
exit

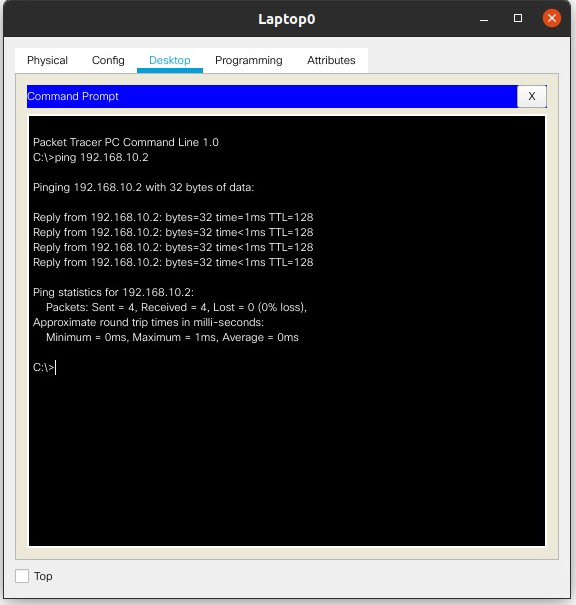
exit

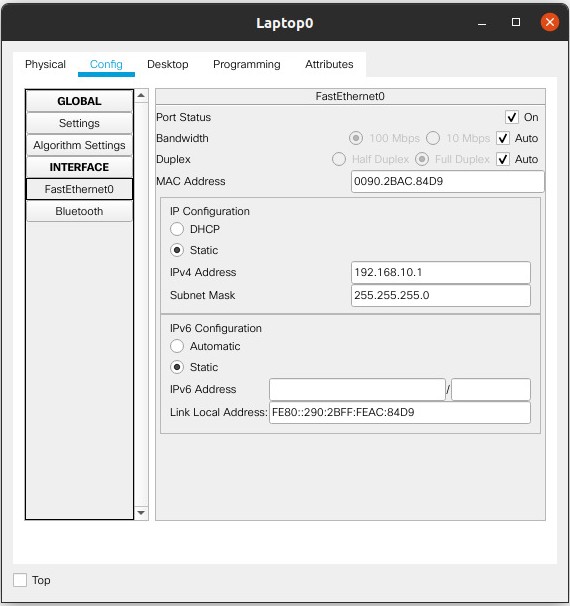


**LAB 9:**

**A1:**

****

****

****