

Name: Apurba Koirala

Reg no: 22BCE3799

Subject Code: BCSE308P

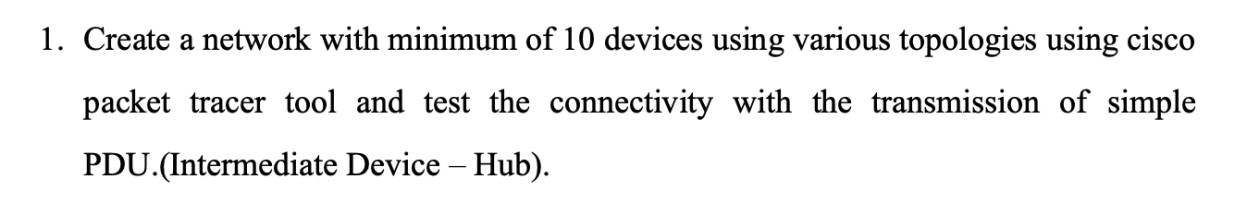
Course Title: Computer Networks Lab

Lab Slot: L31 + L32

Guided by: Dr. Arivoli A

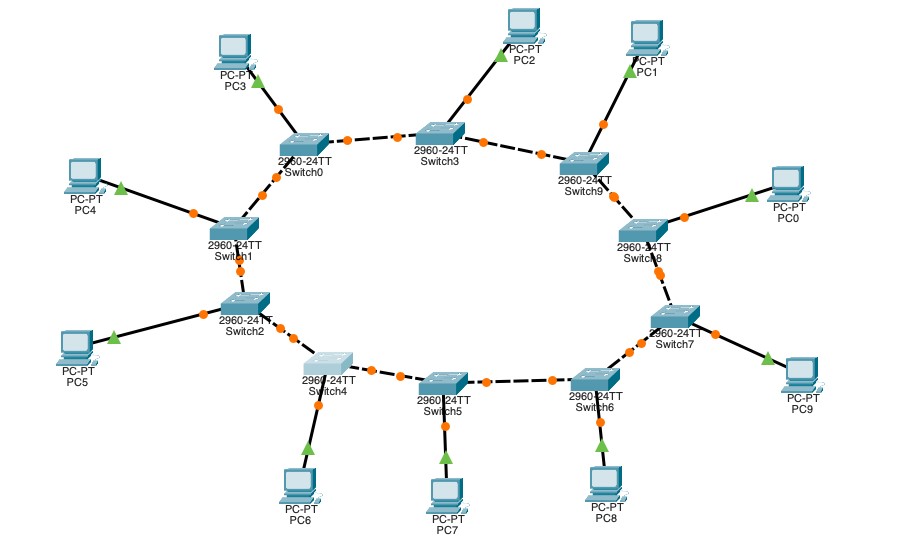
Lab Assessment 3

Question no. 1

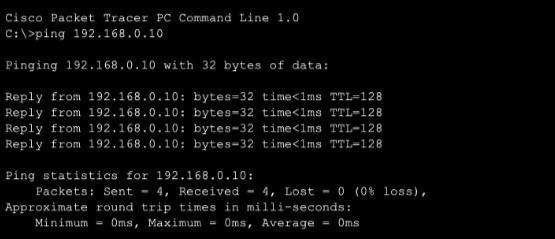


Solution;

RING TOPOLOGY:



PC0:

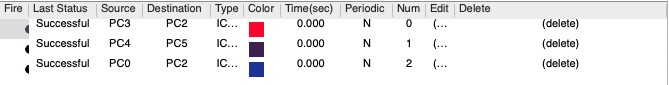
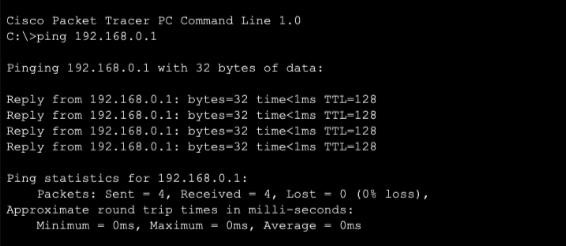


PC4:

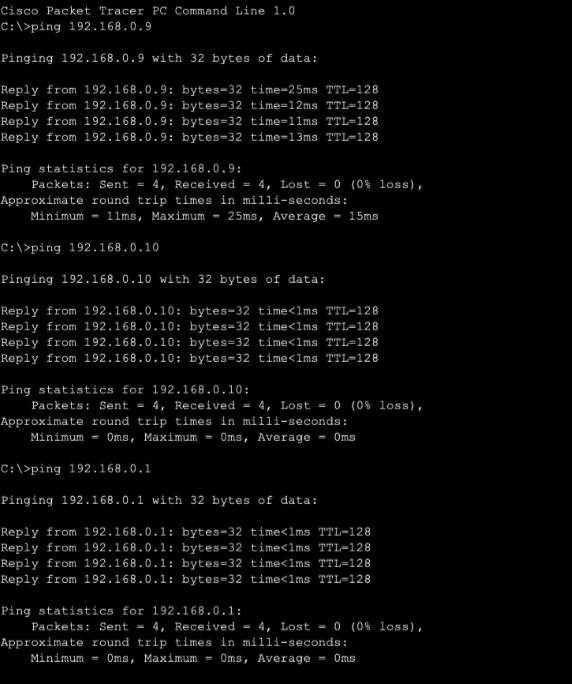
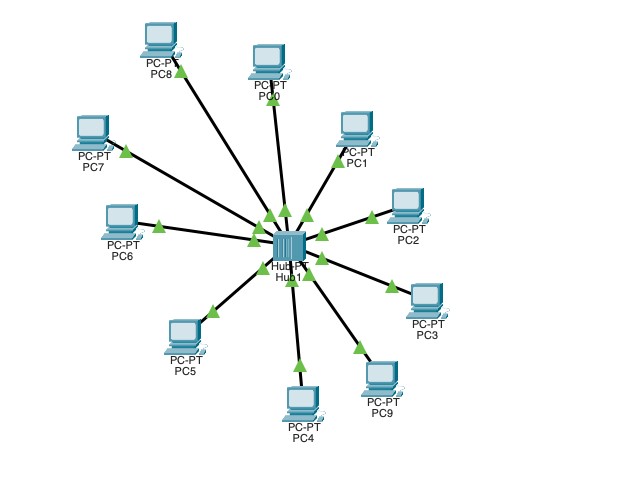
A computer screen with white text

Description automatically generated

PC6:



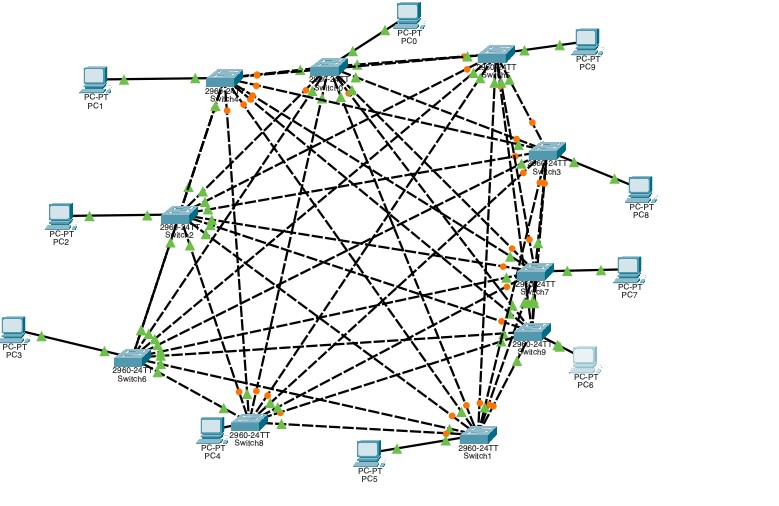
STAR TOPOLOGY:

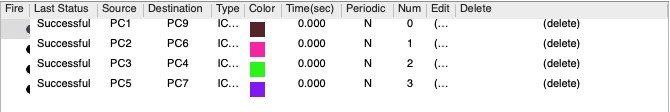
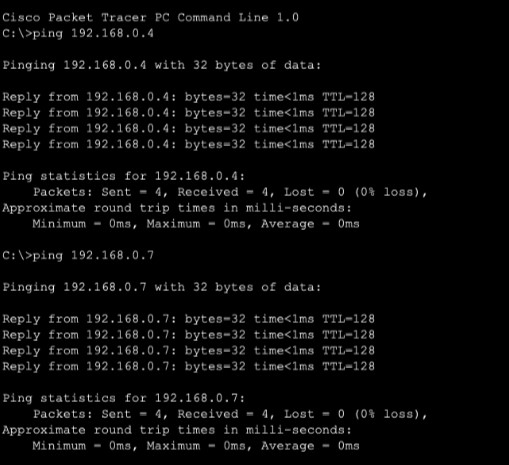


A screenshot of a computer

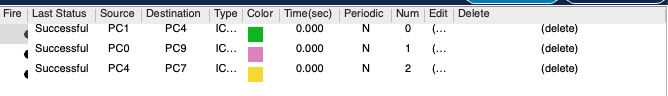
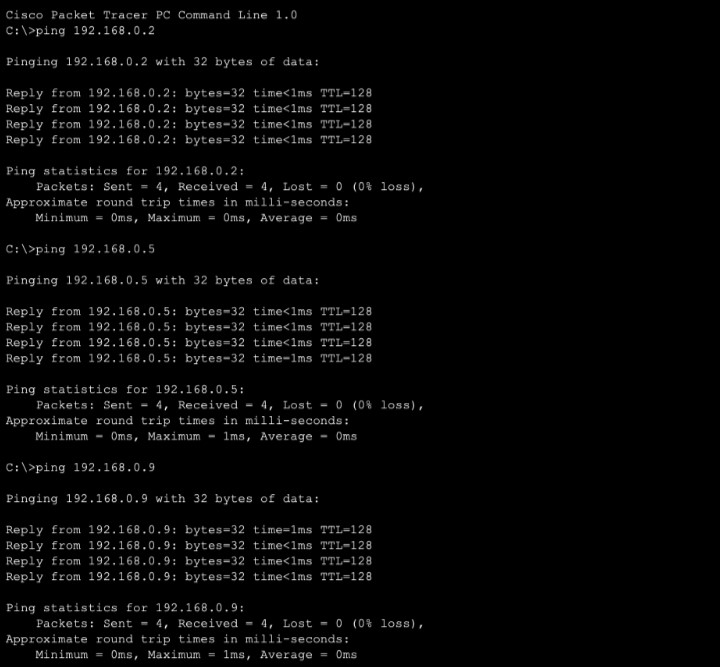
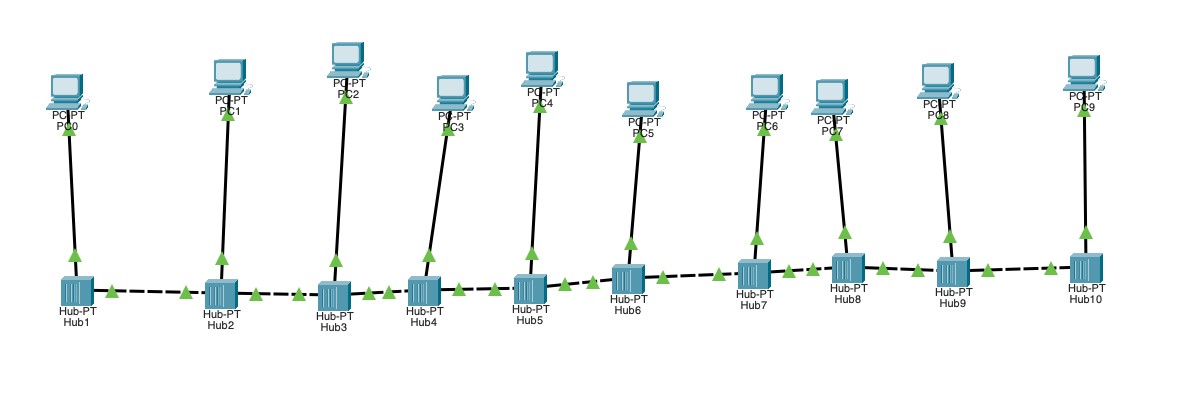
Description automatically generated

MESH TOPOLOGY:

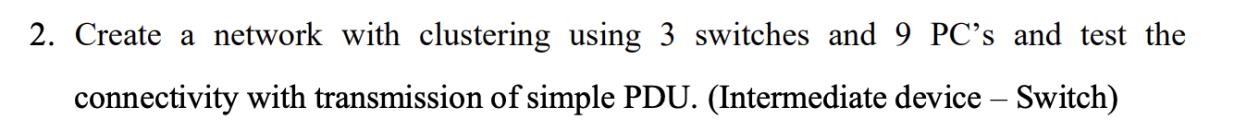




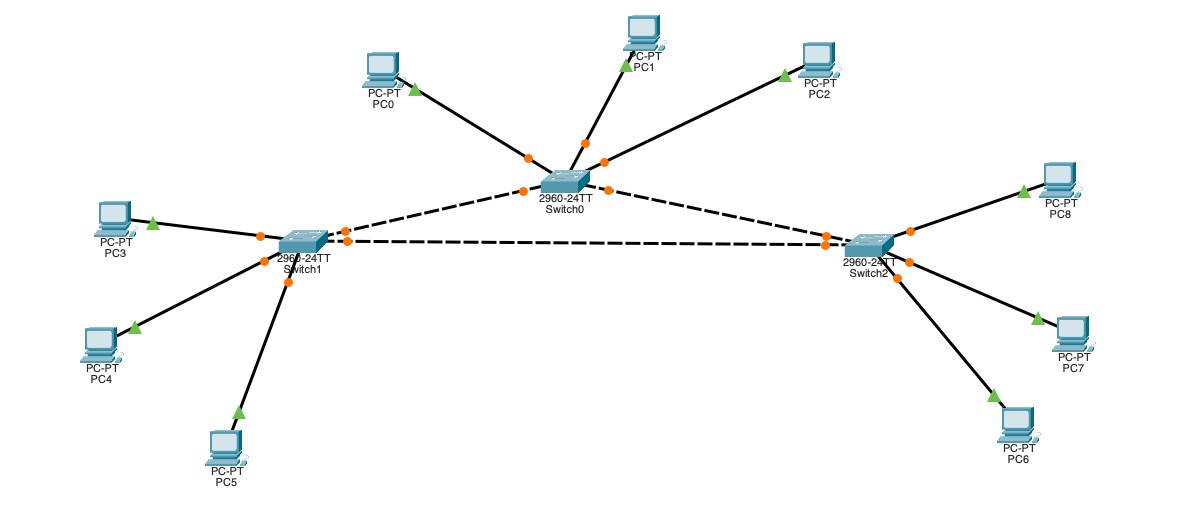
BUS TOPOLOGY:



Question no. 2:



Solution;

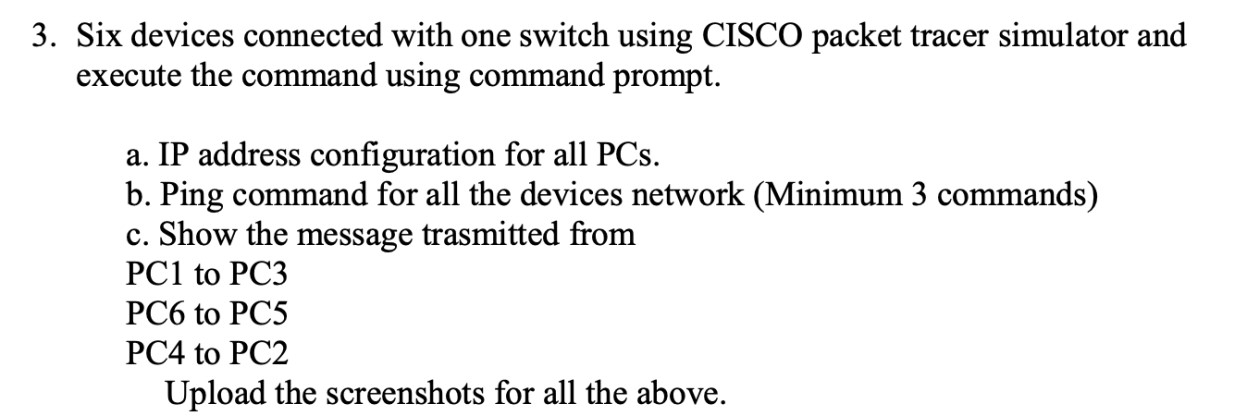


Transmission using simple PDU results:

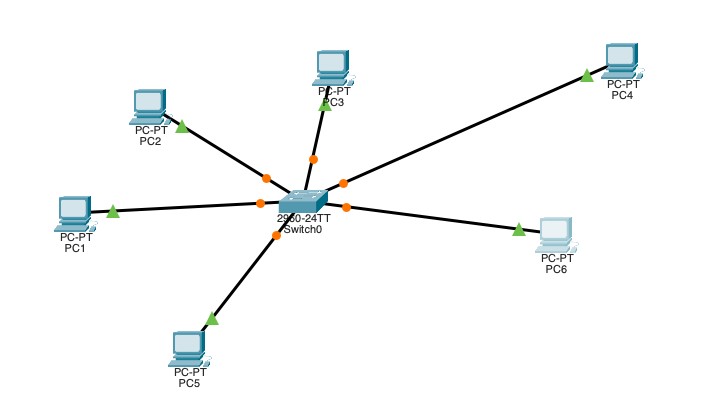
A screenshot of a computer

Description automatically generated

Question no. 3



Solution;

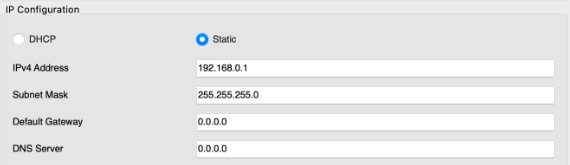
 a. IP address configuration for all PCs

PC1:

A screenshot of a computer

Description automatically generated

PC2:

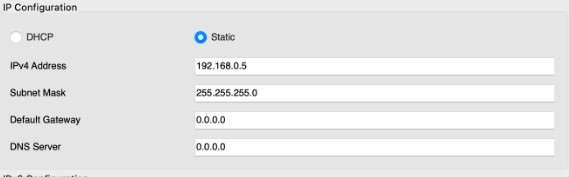


PC3:

A screenshot of a computer

Description automatically generated

PC4:



PC5:

A screenshot of a computer

Description automatically generated

PC6;

A screenshot of a computer

Description automatically generated

b. Ping commands for all devices network (Minimum 3 commands)







c. Show the message transmitted from

PC1 to PC3

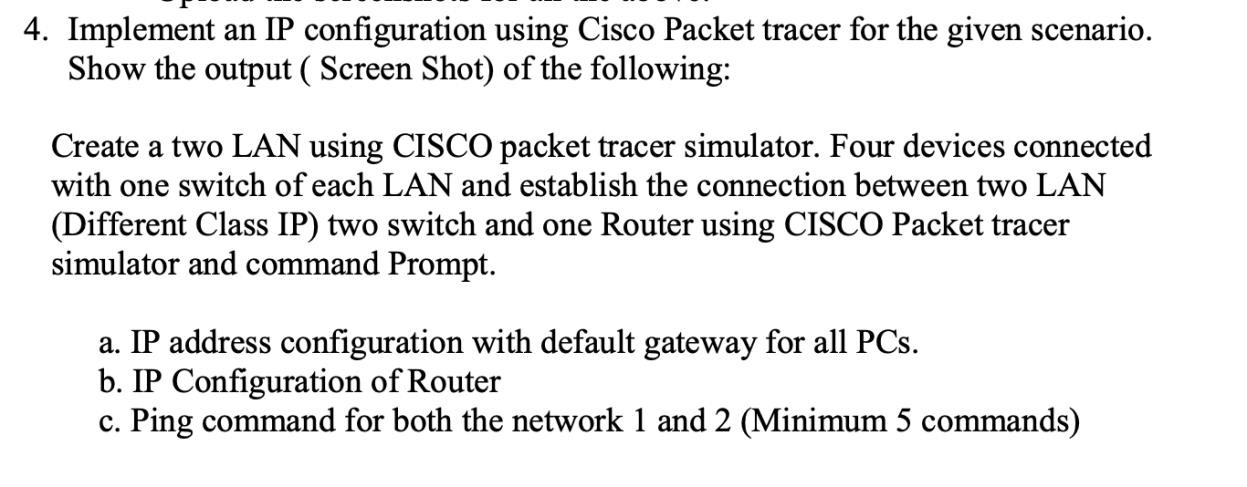
PC6 to PC5

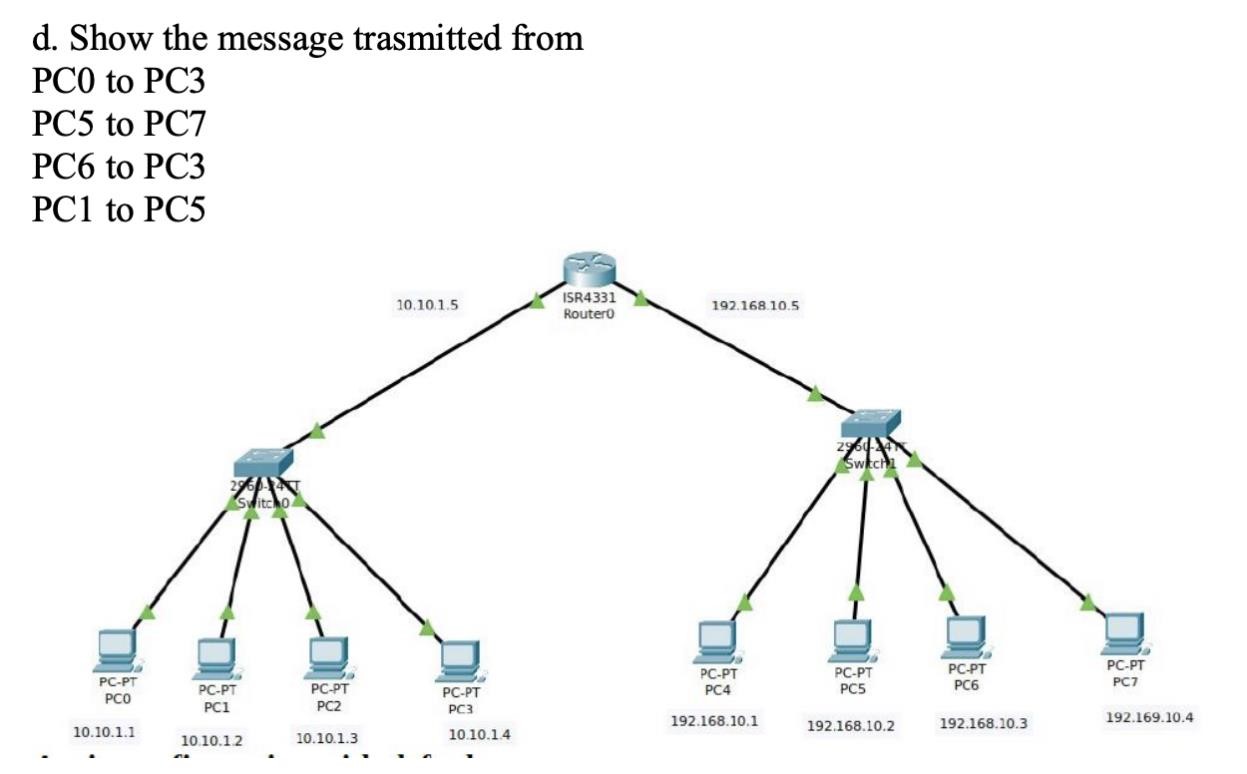
PC4 to PC2

A screenshot of a computer

Description automatically generated

Question no. 4





Solution;

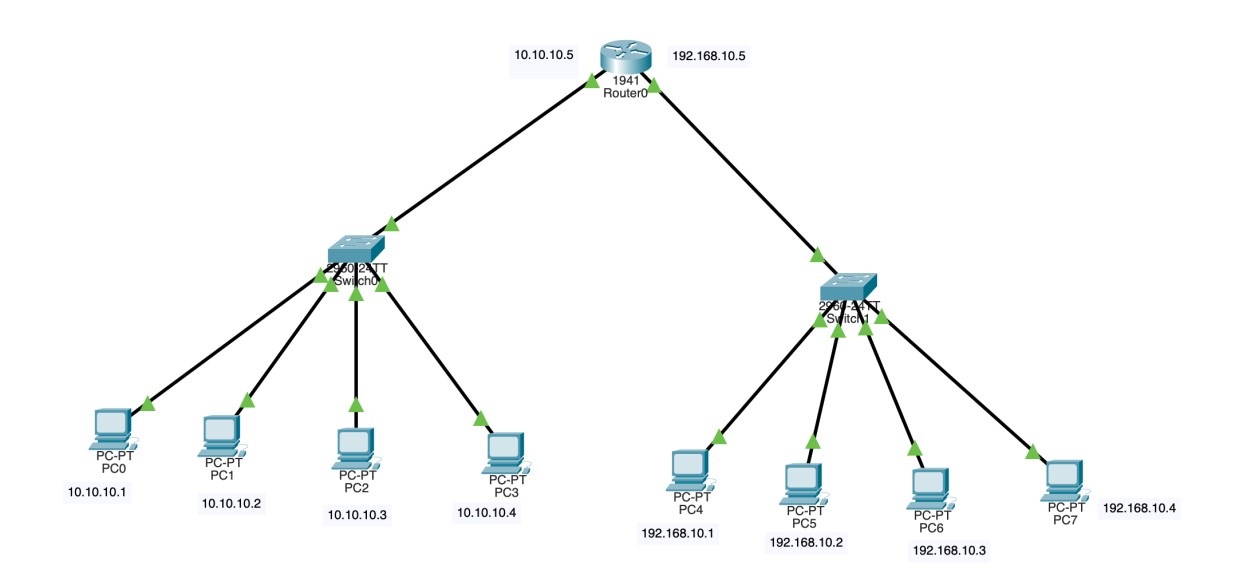


FIG 1

1. IP Configuration and Default Gateway for PCS:

Pc0:

A screenshot of a computer

Description automatically generated

Pc1:

A screenshot of a computer

Description automatically generated

Pc2:

A screenshot of a computer

Description automatically generated

Pc3:

A screenshot of a computer

Description automatically generated

Pc4:

A screenshot of a computer

Description automatically generated

Pc5:

A screenshot of a data

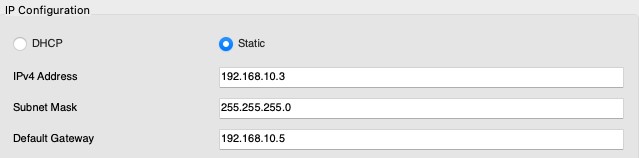
Description automatically generated

Pc6:

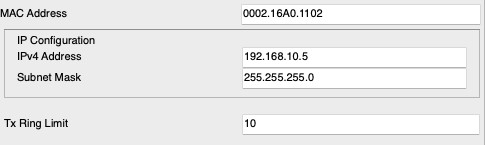
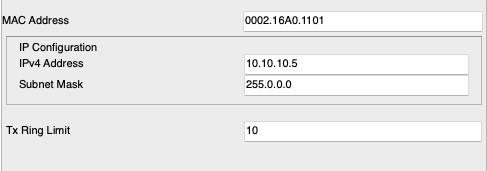
A screenshot of a data

Description automatically generated

Pc7:



1. IP Configuration of Router



1. Ping command for both network 1 and 2

Network 1

A computer screen with white text

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screen shot of a computer

Description automatically generated

A computer screen with white text

Description automatically generated

A computer screen with white text

Description automatically generated

Network 2

A computer screen with white text

Description automatically generated

A screenshot of a computer program

Description automatically generated

A computer screen with white text

Description automatically generated

A computer screen with white text

Description automatically generated

A computer screen with white text

Description automatically generated

1. . Show the message transmitted from

PC0 to PC3

PC5 to PC7

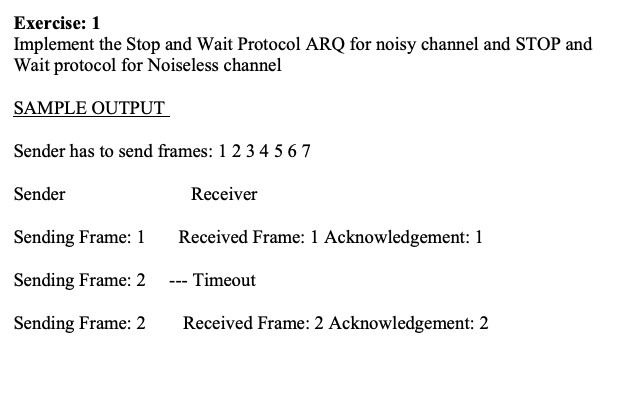
PC6 to PC3

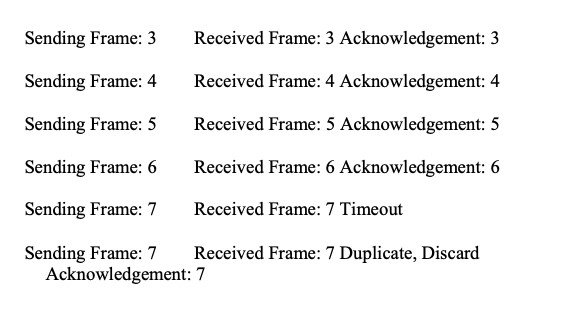
PC1 to PC5

A screenshot of a computer

Description automatically generated

FLOW CONTROL





Solution;

#include <stdio.h>

#include <unistd.h>

void send\_frame(int frame) {

printf("Sending Frame: %d\n", frame);

sleep(1);

}

void receive\_frame(int frame) {

printf("Received Frame: %d\n", frame);

sleep(1);

}

void send\_acknowledgement(int ack) {

printf("Acknowledgement: %d\n", ack);

sleep(1);

}

void stop\_and\_wait\_protocol(int frames[], int len) {

int i = 0;

while (i < len) {

int frame = frames[i];

send\_frame(frame);

receive\_frame(frame); send\_acknowledgement(frame);

i++;

if (i < len) {

send\_frame(frames[i]);

receive\_frame(frames[i]);

send\_acknowledgement(frames[i]);

i++; } else {

break;

}

}

}

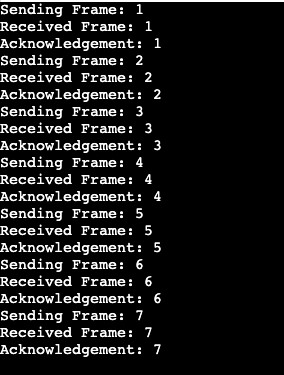
int main() {

int frames\_to\_send[] = {1, 2, 3, 4, 5, 6, 7};

int len = sizeof(frames\_to\_send) / sizeof(frames\_to\_send[0]); stop\_and\_wait\_protocol(frames\_to\_send, len);

return 0;

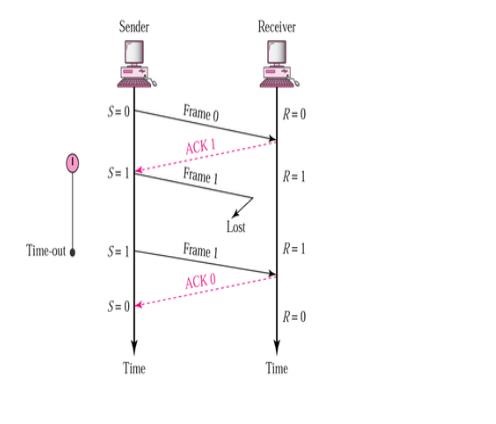
}



A white background with black text

Description automatically generated

Case 1: Frames lost:



#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <time.h>

void send\_frame(int frame\_number) {

printf("Sending Frame %d\n", frame\_number);

sleep(1);

}

int receive\_ack(int expected\_ack) {

int ack = rand() % 2; // Ensures random ack is 0 or 1

sleep(1);

if (ack == expected\_ack) {

printf("ACK %d received\n", ack);

return 1;

} else {

printf("ACK %d received, expected ACK %d\n", ack, expected\_ack);

return 0;

}

}

void stop\_and\_wait\_arq(int total\_frames) {

int frame\_number = 0;

int expected\_ack = 0;

int frames\_sent = 0;

while (frames\_sent < total\_frames) {

send\_frame(frame\_number);

if (receive\_ack(expected\_ack)) {

frame\_number = (frame\_number + 1) % 2;

expected\_ack = (expected\_ack + 1) % 2;

frames\_sent++;

} else {

printf("Frame %d lost. Retransmitting Frame %d\n", frame\_number, frame\_number);

}

}

}

int main() {

srand(time(0));

int total\_frames = 7; // Total number of frames to send

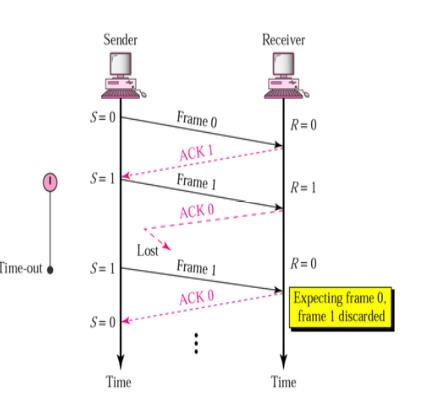
stop\_and\_wait\_arq(total\_frames);

return 0;

}



Case 2: ACK Lost



#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <time.h>

void send\_frame(int frame\_number) {

printf("Sending Frame %d\n", frame\_number);

sleep(1);

}

int receive\_ack(int expected\_ack) {

int ack = rand() % 2;

sleep(1);

if (ack == expected\_ack) {

printf("ACK %d received\n", ack);

return 1; } else {

printf("ACK %d lost\n", ack);

return 0;

}

}

void stop\_and\_wait\_arq(int total\_frames) {

int frame\_number = 0;

int expected\_ack = 0;

int frames\_sent = 0;

while (frames\_sent < total\_frames) {

send\_frame(frame\_number);

if (receive\_ack(expected\_ack)) {

frame\_number = (frame\_number + 1) % 2;

expected\_ack = (expected\_ack + 1) % 2;

frames\_sent++;

} else {

printf("ACK %d lost. Retransmitting Frame %d\n", expected\_ack, frame\_number);

}

}

}

int main() {

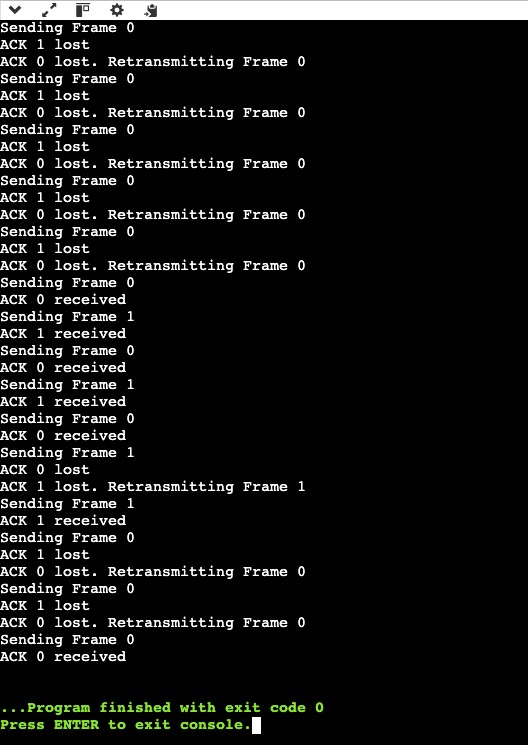
srand(time(0));

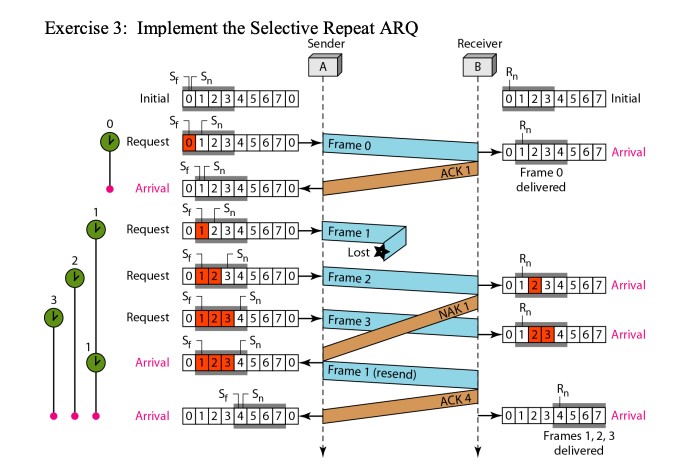
int total\_frames = 7; // Total number of frames to send

stop\_and\_wait\_arq(total\_frames);

return 0;

}





#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <time.h>

void send\_frame(int frame\_number) {

printf("Sending Frame %d\n", frame\_number);

sleep(1);

}

int receive\_ack(int expected\_ack) {

int ack = (rand() % 2 == 0) ? expected\_ack : -1; // Randomly generate ACK or NAK

sleep(1);

if (ack == expected\_ack) {

printf("ACK %d\n", ack);

return 1;

} else {

printf("NAK %d\n", expected\_ack);

return 0;

}

}

void selective\_repeat\_arq() {

int window\_size = 3;

int base = 0;

int next\_sequence\_number = 0;

int frames[] = {0, 1, 2, 3, 4, 5, 6, 7};

int total\_frames = sizeof(frames) / sizeof(frames[0]);

while (base < total\_frames) {

if (next\_sequence\_number < base + window\_size && next\_sequence\_number < total\_frames) {

int frame\_number = frames[next\_sequence\_number];

send\_frame(frame\_number);

next\_sequence\_number++;

}

if (receive\_ack(base)) {

base++;

} else {

printf("Resending Frame %d\n", frames[base]);

}

}

}

int main() {

srand(time(0)); // Seed the random number generator

selective\_repeat\_arq();

return 0;

}A screen shot of a computer

Description automatically generated