WINTER – SEMESTER

Course Code: MCSE505P **Name:** Nidhi Singh

Course-Title: – Computer Network Lab **Reg. No:** 22MAI0015

DIGITAL ASSIGNMENT - 3 (LAB)

Slot- L35+L36

Faculty: - SRIMATHI C (SCOPE)

Socket Programming:

1. **Message passing single client server :-**

Server Code:

```
#include <stdio.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <string.h>
#include<arpa/inet.h>

int main(){
int welcomeSocket, newSocket;
char buffer[1024];
struct sockaddr_in serverAddr;
struct sockaddr_storage serverStorage;
socklen_t addr_size;


welcomeSocket = socket(PF_INET, SOCK_STREAM, 0);

serverAddr.sin_family = AF_INET;

serverAddr.sin_port = htons(7891);

serverAddr.sin_addr.s_addr = inet_addr("10.30.154.76");

memset(serverAddr.sin_zero, '\0', sizeof serverAddr.sin_zero);


bind(welcomeSocket, (struct sockaddr *) &serverAddr, sizeof(serverAddr));


if(listen(welcomeSocket,5)==0)
printf("Listening\n");
else
  printf("Error\n");
```

```
addr_size = sizeof serverStorage;
newSocket = accept(welcomeSocket, (struct sockaddr *) &serverStorage,
&addr_size);


strcpy(buffer,"Hello World\n");
send(newSocket,buffer,13,0);

return 0;
}
```

Client :-

```
#include <stdio.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <string.h>
#include<arpa/inet.h>

int main(){
int welcomeSocket, newSocket;
char buffer[1024];
struct sockaddr_in serverAddr;
struct sockaddr_storage serverStorage;
socklen_t addr_size;


welcomeSocket = socket(PF_INET, SOCK_STREAM, 0);

serverAddr.sin_family = AF_INET;

serverAddr.sin_port = htons(7891);

serverAddr.sin_addr.s_addr = inet_addr("10.30.154.76");

memset(serverAddr.sin_zero, '\0', sizeof serverAddr.sin_zero);


bind(welcomeSocket, (struct sockaddr *) &serverAddr, sizeof(serverAddr));


if(listen(welcomeSocket,5)==0)
printf("Listening\n");
else
  printf("Error\n");

addr_size = sizeof serverStorage;
newSocket = accept(welcomeSocket, (struct sockaddr *) &serverStorage,
&addr_size);


strcpy(buffer,"Hello World\n");
send(newSocket,buffer,13,0);

return 0;
}
```

Screenshot:

```
matlab@sjt319scope066:~/22MAI0066$ touch Server.c
matlab@sjt319scope066:~/22MAI0066$ gedit Server.c
matlab@sjt319scope066:~/22MAI0066$ gcc Server.c -o Server.out
matlab@sjt319scope066:~/22MAI0066$ ./Server.out
Listening
matlab@sjt319scope066:~/22MAI0066$
```

2. Multiple client message :-

Client :-

```
#include <stdio.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include<arpa/inet.h>
#include <string.h>

int main(){
int clientSocket;
char buffer[1024];
struct sockaddr_in serverAddr;
socklen_t addr_size;
clientSocket = socket(PF_INET, SOCK_STREAM, 0);
serverAddr.sin_family = AF_INET;
serverAddr.sin_port = htons(7891);
serverAddr.sin_addr.s_addr = inet_addr("10.30.154.76");
memset(serverAddr.sin_zero, '\0', sizeof serverAddr.sin_zero);
addr_size = sizeof serverAddr;
connect(clientSocket, (struct sockaddr *) &serverAddr, addr_size);
recv(clientSocket, buffer, 1024, 0);
printf("Data received: %s",buffer);
return 0;
}
```

Server :-

```
#include <stdio.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <string.h>

int main(){
  int welcomeSocket, newSocket;
  char buffer[1024];
  struct sockaddr_in serverAddr;
  struct sockaddr_storage serverStorage;
  socklen_t addr_size;
  serverAddr.sin_family = AF_INET;
```

```
  serverAddr.sin_port = htons(7891);
  serverAddr.sin_addr.s_addr = inet_addr("10.30.154.75");
  memset(serverAddr.sin_zero, '\0', sizeof serverAddr.sin_zero);
  bind(welcomeSocket, (struct sockaddr *) &serverAddr,
sizeof(serverAddr));
 if(listen(welcomeSocket,5)==0)
    printf("Listening\n");
  else
  printf("Error\n");
  addr_size = sizeof serverStorage;
  newSocket = accept(welcomeSocket, (struct sockaddr *) &serverStorage,
&addr_size);

   strcpy(buffer,"Hello World\n");
  send(newSocket,buffer,13,0);

  return 0;
}
```

Output :-

```
matlab@sjt319scope065:~$ gcc Client.c -o Client.out
matlab@sjt319scope065:~$ ./Client.out
Data received: Hello World
matlab@sjt319scope065:~$
```

TCP Chat Application

Sever.c

```
#include<sys/socket.h>
#include<sys/types.h>
#include<stdio.h>
#include<arpa/inet.h>
#include<netinet/in.h>
#include<string.h>
#include<unistd.h>
#define SER_PORT 1200
int main()
{
int a,sersock,newsock,n;
char str[25],str2[25];
struct sockaddr_in seraddr;
struct sockaddr_in cliinfo;
socklen_t csize=sizeof(cliinfo);
seraddr.sin_family=AF_INET;
seraddr.sin_port=htons(SER_PORT);
seraddr.sin_addr.s_addr=htonl(INADDR_ANY);
if((sersock=socket(AF_INET,SOCK_STREAM,0))<0)
{
error("\n socket");
exit(0);
}
```

```
if(bind(sersock,(struct sockaddr *)&seraddr,sizeof(seraddr))<0)
{
error("\nBIND");
exit(0);
}
if(listen(sersock,1)<0)
{
error("\n LISTEN");
}
if((newsock=accept(sersock,(struct sockaddr *)&cliinfo,&csize))<0)
{
error("\n ACCEPT");
exit(0);
}s
else
printf("\n now connected to %s\n",inet_ntoa(cliinfo.sin_addr));
read(newsock,str,sizeof(str));
do
{
printf("\n client msg:%s",str);
printf("\n server msg:");
scanf("%s",str2);
write(newsock,str2,sizeof(str2));
listen(newsock,1);
read(newsock,str,sizeof(str));
n=strcmp(str,"BYE");
a=strcmp(str2,"BYE");
}
while(n!=0||a!=0);
close(newsock);
close(sersock);
return 0;
}
```

Client.c

```
#include<stdio.h>
#include<sys/socket.h>
#include<sys/types.h>
#include<arpa/inet.h>
#include<netinet/in.h>
#include<unistd.h>
#define SER_PORT 1200
int main(int count,char*arg[])
{
int a,clisock;
char str[20],str2[20];
struct sockaddr_in cliaddr;
cliaddr.sin_port=htons(SER_PORT);
cliaddr.sin_family=AF_INET;
cliaddr.sin_addr.s_addr=inet_addr("10.30.154.76");
clisock=socket(AF_INET,SOCK_STREAM,0);
if(clisock<0)
{
perror("\n SOCKET");
exit(0);
```

```
}
if(connect(clisock,(struct sockaddr*)&cliaddr,sizeof(cliaddr))<0)
{
perror("\n CONNECT");
exit(0);
}
printf("\nclient connected to %s",arg[1]);
printf("\nCLIENT");
scanf("%s",&str);
if(write(clisock,str,sizeof(str))<0)
{
printf("\n data could not be sent");
}
do
{
listen(clisock,1);
read(clisock,str2,sizeof(str2));
printf("\nserver msg:%s",str2);
printf("\nclient msg:");
scanf("%s",&str);
a=strcmp(str2,"BYE");
write(clisock,str2,sizeof(str2));
}
while(a!=0);
close(clisock);
return 0;
}
```

Output :-

```
matlab@sjt319scope065:~$ ./a.out

 now connected to 10.30.154.76

 client msg:hiinidhi
 server msg:hi_shivani

 client msg:hi_shivani
 server msg:how_are_you
```

UDP Chat Application

Server

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sys/socket.h>
#include <sys/types.h>
#include <netinet/in.h>
#include <arpa/inet.h>
int main(int argc, char **argv){
 if (argc != 2) {
   printf("Usage: %s <port>\n", argv[0]);
```

```
    exit(0);
  }
  char *ip = "127.0.0.1";
  int port = atoi(argv[1]);
  int sockfd;
  struct sockaddr_in server_addr, client_addr;
  char buffer[1024];
  socklen_t addr_size;
  int n;
  sockfd = socket(AF_INET, SOCK_DGRAM, 0);
  if (sockfd < 0) {
    perror("[-]socket error");
    exit(1);
  }
  memset(&server_addr, '\0', sizeof(server_addr));
  server_addr.sin_family = AF_INET;
  server_addr.sin_port = htons(port);
  server_addr.sin_addr.s_addr = inet_addr(ip);
  n = bind(sockfd, (struct sockaddr*)&server_addr, sizeof(server_addr));
  if (n < 0){
    perror("[-]bind error");
    exit(1);
  }
  bzero(buffer, 1024);
  addr_size = sizeof(client_addr);
  recvfrom(sockfd, buffer, 1024, 0, (struct sockaddr*)&client_addr,
&addr_size);
  printf("[+]Data recv: %s\n", buffer);
  bzero(buffer, 1024);
  strcpy(buffer, "Welcome to the UDP Server.");
  sendto(sockfd, buffer, 1024, 0, (struct sockaddr*)&client_addr,
sizeof(client_addr));
  printf("[+]Data send: %s\n", buffer);
  return 0;
}
```

Output :-

```
File  Edit  View  Search  Terminal  Help
matlab@sjt319scope065:~$ cc -o bb bb.c
matlab@sjt319scope065:~$ ./bb 4455
^C
matlab@sjt319scope065:~$ cc -o bb bb.c
matlab@sjt319scope065:~$ ./bb 4455
[+]Data recv: Hello World!
[+]Data send: Welcome to the UDP Server.
matlab@sjt319scope065:~$
```

Client

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sys/socket.h>
#include <sys/types.h>
#include <netinet/in.h>
#include <arpa/inet.h>
int main(int argc, char **argv){
 if (argc != 2) {
   printf("Usage: %s <port>\n", argv[0]);
   exit(0);
 }
 char *ip = "127.0.0.1";
 int port = atoi(argv[1]);
 int sockfd;
 struct sockaddr_in addr;
 char buffer[1024];
 socklen_t addr_size;
 sockfd = socket(AF_INET, SOCK_DGRAM, 0);
 memset(&addr, '\0', sizeof(addr));
 addr.sin_family = AF_INET;
 addr.sin_port = htons(port);
 addr.sin_addr.s_addr = inet_addr(ip);
 bzero(buffer, 1024);
 strcpy(buffer, "Hello World!");
 sendto(sockfd, buffer, 1024, 0, (struct sockaddr*)&addr,
sizeof(addr));
 printf("[+]Data send: %s\n", buffer);
 bzero(buffer, 1024);
 addr_size = sizeof(addr);
 recvfrom(sockfd, buffer, 1024, 0, (struct sockaddr*)&addr,
&addr_size);
 printf("[+]Data recv: %s\n", buffer);
 return 0;
}
```

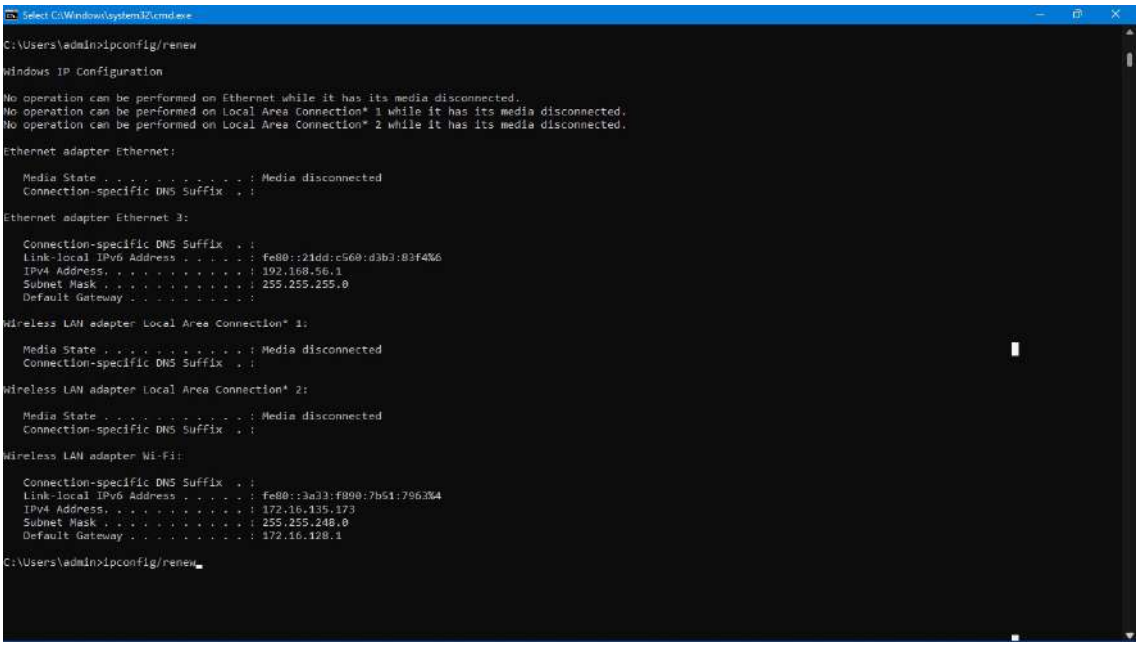
Output :-

```
matlab@sjt319scope065: ~
File  Edit  View  Search  Terminal  Help
matlab@sjt319scope065:~$ cc -o nidhi_c nidhi_c.c
matlab@sjt319scope065:~$ ./nidhi_c
Usage: ./nidhi_c <port>
matlab@sjt319scope065:~$ ./nidhi_c 4455
[+]Data send: Hello World!
[+]Data recv: Welcome to the UDP Server.
matlab@sjt319scope065:~$
```

DHCP

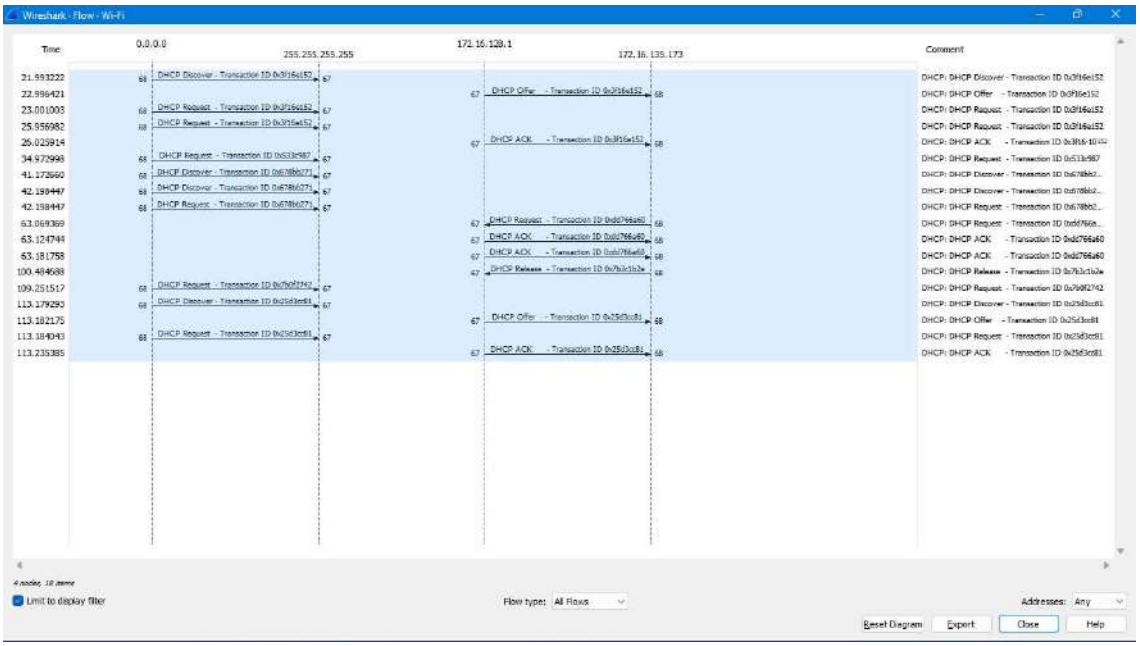
```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 10.0.22000.1817]
(c) Microsoft Corporation. All rights reserved.

C:\Users\admin>ipconfig/release

Windows IP Configuration

No operation can be performed on Ethernet while it has its media disconnected.
No operation can be performed on Local Area Connection* 1 while it has its media disconnected.

Ethernet adapter Ethernet:

   Media State . . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Ethernet adapter Ethernet 3:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::21dd:c560:d3b3:83f4%6
   IPv4 Address. . . . . . . . . . . : 192.168.56.1
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . :

Wireless LAN adapter Local Area Connection* 1:

   Media State . . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Wireless LAN adapter Local Area Connection* 2:

   Media State . . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Wireless LAN adapter Wi-Fi:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::3a33:f890:7b51:7963%4
   Default Gateway . . . . . . . . . :

C:\Users\admin>
```

```
Select C:\Windows\system32\cmd.exe
   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::3a33:f890:7b51:7963%4
   Default Gateway . . . . . . . . . :

C:\Users\admin>ipconfig/renew

Windows IP Configuration

No operation can be performed on Ethernet while it has its media disconnected.
No operation can be performed on Local Area Connection* 1 while it has its media disconnected.
No operation can be performed on Local Area Connection* 2 while it has its media disconnected.

Ethernet adapter Ethernet:

   Media State . . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Ethernet adapter Ethernet 3:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::21dd:c560:d3b3:83f4%6
   IPv4 Address. . . . . . . . . . . : 192.168.56.1
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . :

Wireless LAN adapter Local Area Connection* 1:

   Media State . . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Wireless LAN adapter Local Area Connection* 2:

   Media State . . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Wireless LAN adapter Wi-Fi:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::3a33:f890:7b51:7963%4
   IPv4 Address. . . . . . . . . . . : 172.16.135.173
   Subnet Mask . . . . . . . . . . . : 255.255.248.0
   Default Gateway . . . . . . . . . : 172.16.128.1

C:\Users\admin>ipconfig/renew

Windows IP Configuration
```



1. **Are DHCP messages sent over UDP or TCP?**
   Answer: DHCP messages are sent over UDP (User Datagram Protocol).

2. **Draw a timing datagram illustrating the sequence of the first four-packet Discover/Offer/Request/ACK DHCP exchange between the client and server. For each packet, indicated the source and destination port numbers. Are the port numbers the same as in the example given in this lab assignment?**
   Answer: The port numbers are the same as the example given in this lab assignment.



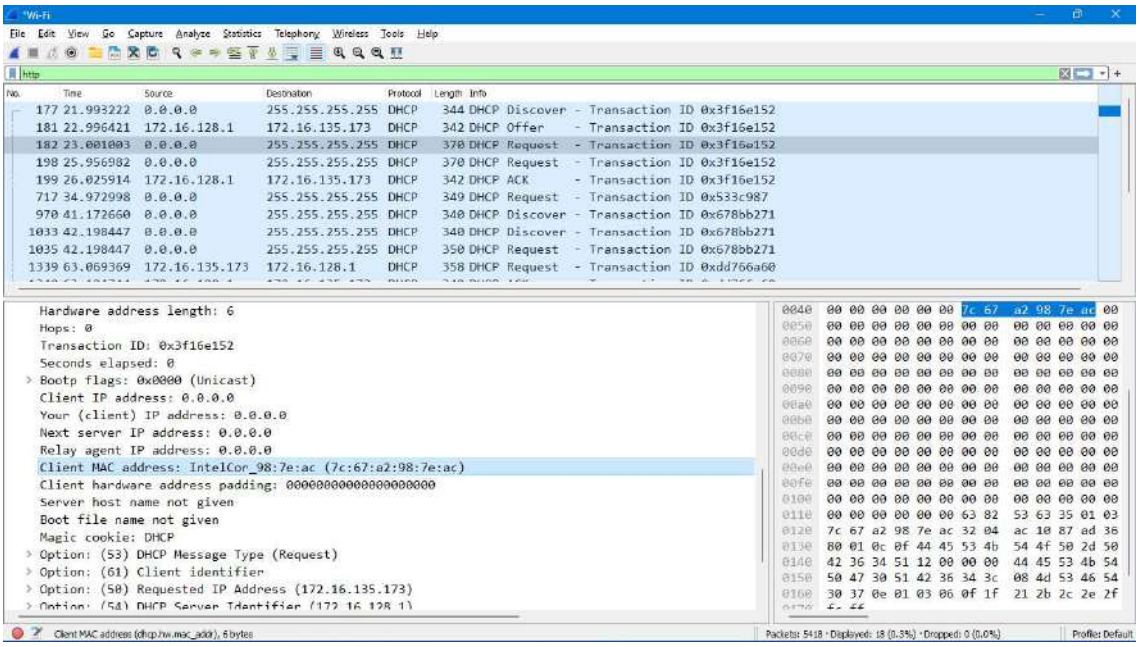
1) Src – 68, Dst – 67

2) Dst – 67, Src – 68

3) Src – 68, Dst – 67

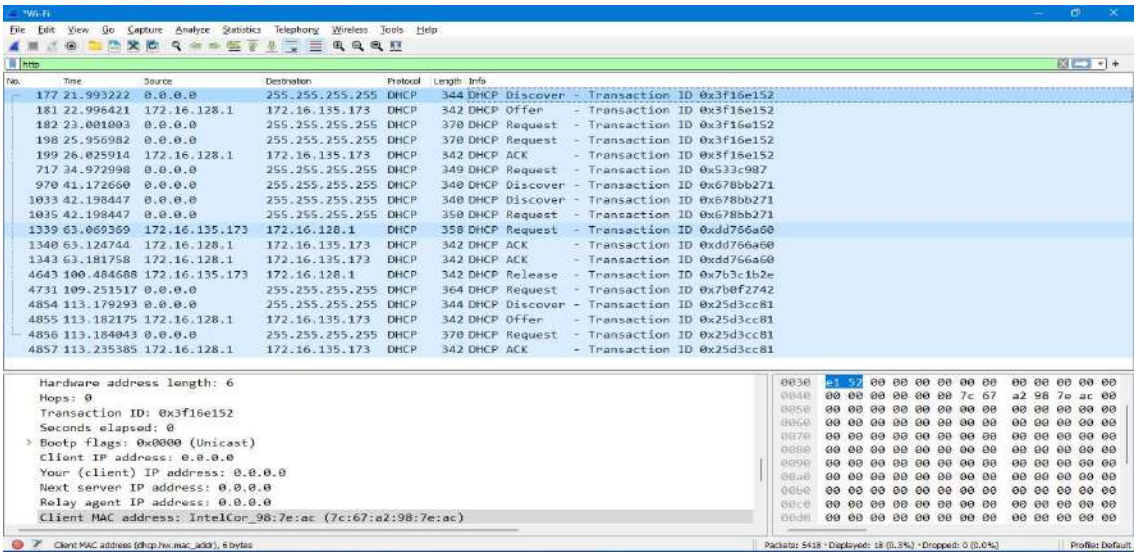
4) Dst – 67, Src – 68

3. What is the link-layer (e.g., Ethernet) address of your host?



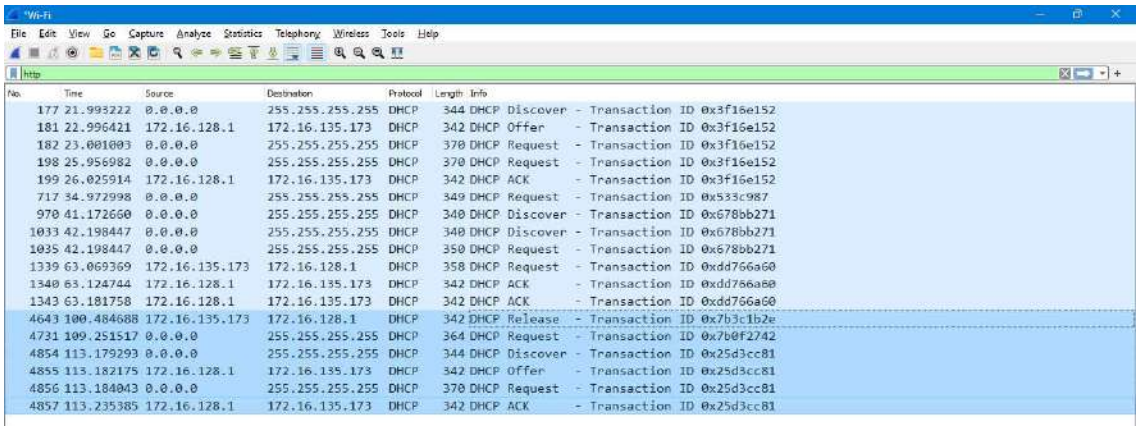
3. **What values in the DHCP discover message differentiate this message from the DHCP request message?**
   The message type value differentiates from the discover and request message. 1 is for discover and 3 is for request.

4. **What is the value of the Transaction-ID in each of the first four (Discover/Offer/Request/ACK) DHCP messages? What are the values of the Transaction-ID in the second set (Request/ACK) set of DHCP messages? What is the purpose of the Transaction-ID field?**



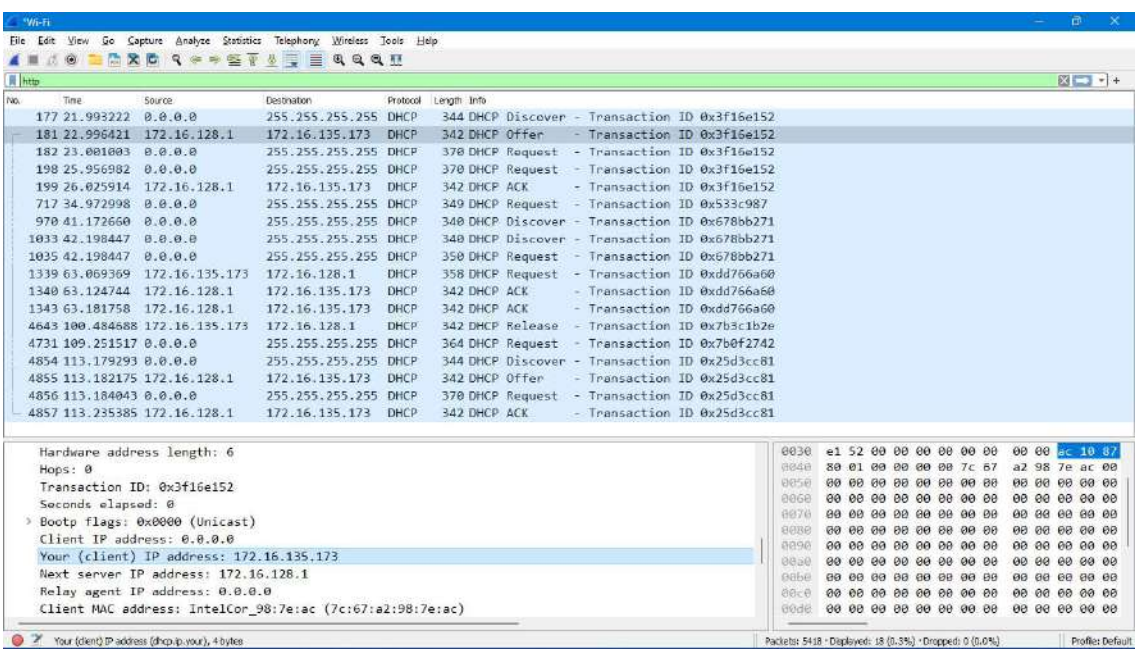
The purpose is to differentiate between the groups of messages.

5. **A host uses DHCP to obtain an IP address, among other things. But a host's IP address is not confirmed until the end of the four-message exchange! If the IP address is not set until the end of the four-message exchange, then what values are used in the IP datagrams in the four-message exchange? For each of the four DHCP messages (Discover/Offer/Request/ACK DHCP), indicate the source and destination IP addresses that are carried in the encapsulating IP datagram.**



No.	Time	Source	Destination	Protocol	Length	Info
177	21.993222	0.0.0.0	255.255.255.255	DHCP	344	DHCP Discover - Transaction ID 0x3f16e152
181	22.996421	172.16.128.1	172.16.135.173	DHCP	342	DHCP Offer - Transaction ID 0x3f16e152
182	23.001003	0.0.0.0	255.255.255.255	DHCP	370	DHCP Request - Transaction ID 0x3f16e152
198	25.956982	0.0.0.0	255.255.255.255	DHCP	370	DHCP Request - Transaction ID 0x3f16e152
199	26.025914	172.16.128.1	172.16.135.173	DHCP	342	DHCP ACK - Transaction ID 0x3f16e152
717	34.972998	0.0.0.0	255.255.255.255	DHCP	349	DHCP Request - Transaction ID 0x533c987
970	41.172660	0.0.0.0	255.255.255.255	DHCP	340	DHCP Discover - Transaction ID 0x678bb271
1033	42.198447	0.0.0.0	255.255.255.255	DHCP	340	DHCP Discover - Transaction ID 0x678bb271
1035	42.198447	0.0.0.0	255.255.255.255	DHCP	350	DHCP Request - Transaction ID 0x678bb271
1339	63.069369	172.16.135.173	172.16.128.1	DHCP	358	DHCP Request - Transaction ID 0xdd766a60
1340	63.124744	172.16.128.1	172.16.135.173	DHCP	342	DHCP ACK - Transaction ID 0xdd766a60
1343	63.181758	172.16.128.1	172.16.135.173	DHCP	342	DHCP ACK - Transaction ID 0xdd766a60
4643	100.484688	172.16.135.173	172.16.128.1	DHCP	342	DHCP Release - Transaction ID 0x7b3c1b2e
4731	109.251517	0.0.0.0	255.255.255.255	DHCP	364	DHCP Request - Transaction ID 0x7b0f2742
4854	113.179293	0.0.0.0	255.255.255.255	DHCP	344	DHCP Discover - Transaction ID 0x25d3cc81
4855	113.182175	172.16.128.1	172.16.135.173	DHCP	342	DHCP Offer - Transaction ID 0x25d3cc81
4856	113.184043	0.0.0.0	255.255.255.255	DHCP	370	DHCP Request - Transaction ID 0x25d3cc81
4857	113.235385	172.16.128.1	172.16.135.173	DHCP	342	DHCP ACK - Transaction ID 0x25d3cc81

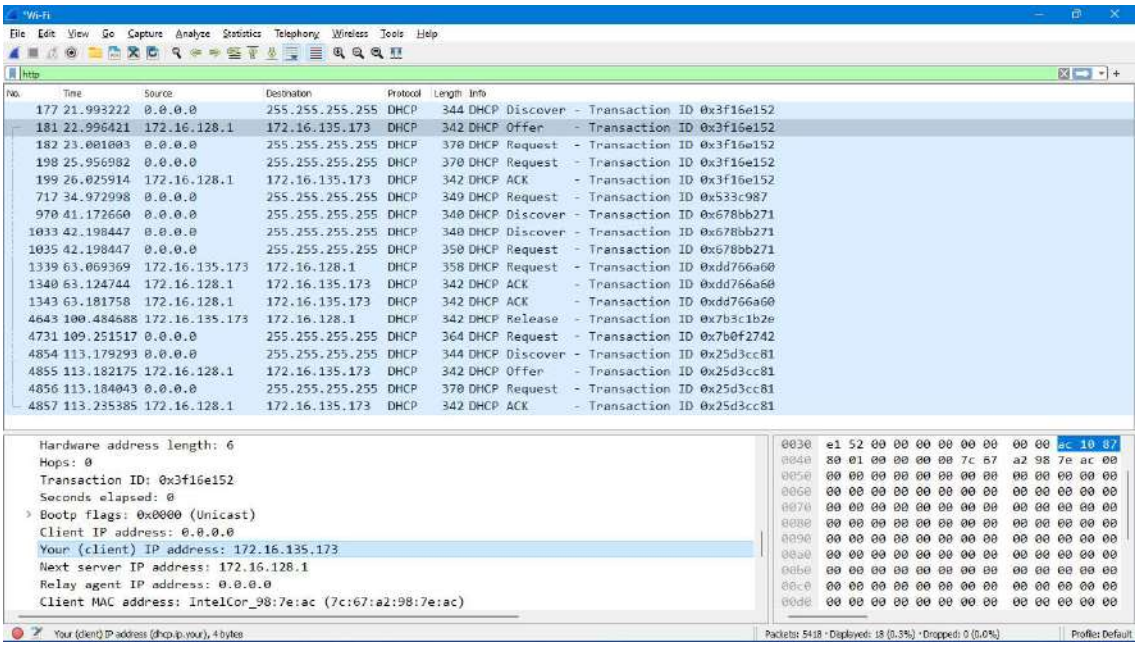
6. **What is the IP address of your DHCP server?**



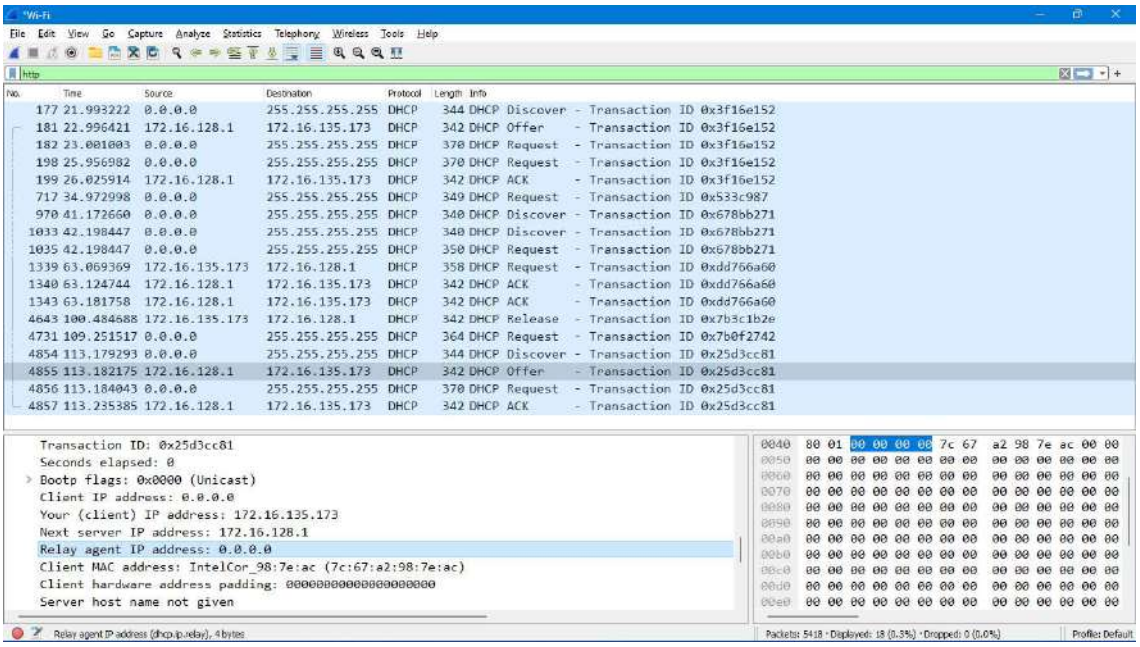
No.	Time	Source	Destination	Protocol	Length	Info
177	21.993222	0.0.0.0	255.255.255.255	DHCP	344	DHCP Discover - Transaction ID 0x3f16e152
181	22.996421	172.16.128.1	172.16.135.173	DHCP	342	DHCP Offer - Transaction ID 0x3f16e152
182	23.001003	0.0.0.0	255.255.255.255	DHCP	370	DHCP Request - Transaction ID 0x3f16e152
198	25.956982	0.0.0.0	255.255.255.255	DHCP	370	DHCP Request - Transaction ID 0x3f16e152
199	26.025914	172.16.128.1	172.16.135.173	DHCP	342	DHCP ACK - Transaction ID 0x3f16e152
717	34.972998	0.0.0.0	255.255.255.255	DHCP	349	DHCP Request - Transaction ID 0x533c987
970	41.172660	0.0.0.0	255.255.255.255	DHCP	340	DHCP Discover - Transaction ID 0x678bb271
1033	42.198447	0.0.0.0	255.255.255.255	DHCP	340	DHCP Discover - Transaction ID 0x678bb271
1035	42.198447	0.0.0.0	255.255.255.255	DHCP	350	DHCP Request - Transaction ID 0x678bb271
1339	63.069369	172.16.135.173	172.16.128.1	DHCP	358	DHCP Request - Transaction ID 0xdd766a60
1340	63.124744	172.16.128.1	172.16.135.173	DHCP	342	DHCP ACK - Transaction ID 0xdd766a60
1343	63.181758	172.16.128.1	172.16.135.173	DHCP	342	DHCP ACK - Transaction ID 0xdd766a60
4643	100.484688	172.16.135.173	172.16.128.1	DHCP	342	DHCP Release - Transaction ID 0x7b3c1b2e
4731	109.251517	0.0.0.0	255.255.255.255	DHCP	364	DHCP Request - Transaction ID 0x7b0f2742
4854	113.179293	0.0.0.0	255.255.255.255	DHCP	344	DHCP Discover - Transaction ID 0x25d3cc81
4855	113.182175	172.16.128.1	172.16.135.173	DHCP	342	DHCP Offer - Transaction ID 0x25d3cc81
4856	113.184043	0.0.0.0	255.255.255.255	DHCP	370	DHCP Request - Transaction ID 0x25d3cc81
4857	113.235385	172.16.128.1	172.16.135.173	DHCP	342	DHCP ACK - Transaction ID 0x25d3cc81

```
Hardware address length: 6
Hops: 0
Transaction ID: 0x3f16e152
Seconds elapsed: 0
> Bootp flags: 0x0000 (Unicast)
Client IP address: 0.0.0.0
Your (client) IP address: 172.16.135.173
Next server IP address: 172.16.128.1
Relay agent IP address: 0.0.0.0
Client MAC address: IntelCor_98:7e:ac (7c:67:a2:98:7e:ac)
```

7. **What IP address is the DHCP server offering to your host in the DHCP Offer message? Indicate which DHCP message contains the offered DHCP address.**

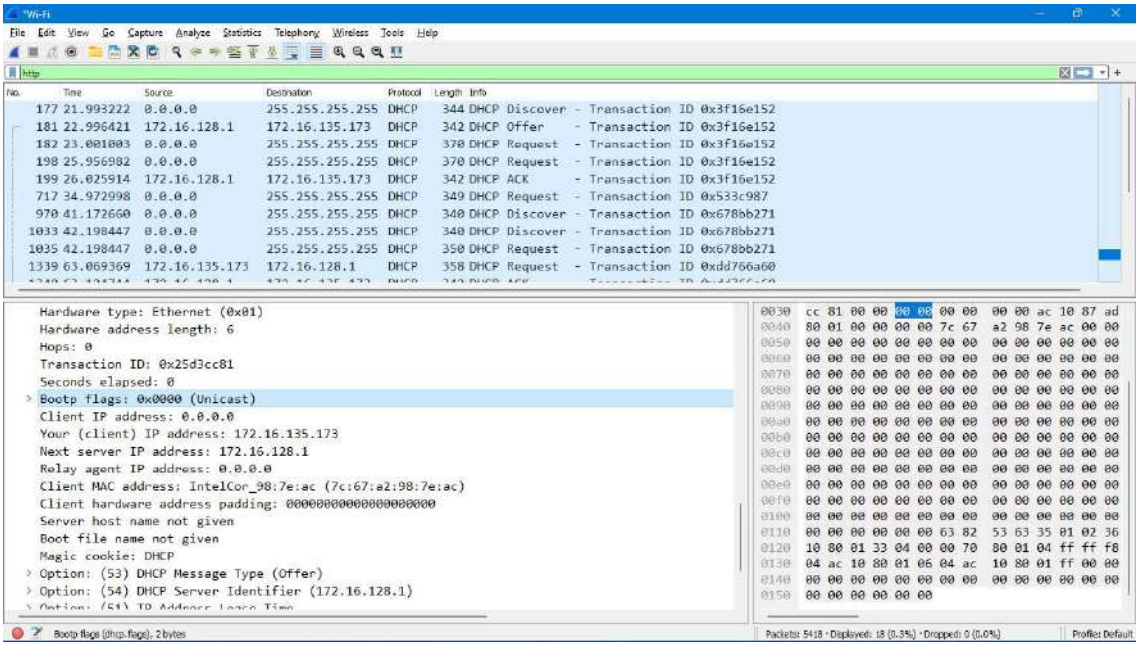


8. **In the example screenshot in this assignment, there is no relay agent between the host and the DHCP server. What values in the trace indicate the absence of a relay agent? Is there a relay agent in your experiment? If so what is the IP address of the agent?**



9. **Explain the purpose of the router and subnet mask lines in the DHCP offer message.**
   To show us the default gateway.

10. **In the DHCP trace file noted in footnote 2, the DHCP server offers a specific IP address to the client (see also question 8. above). In the client's response to the first server OFFER message, does the client accept this IP address? Where in the client's RESPONSE is the client's requested address?**



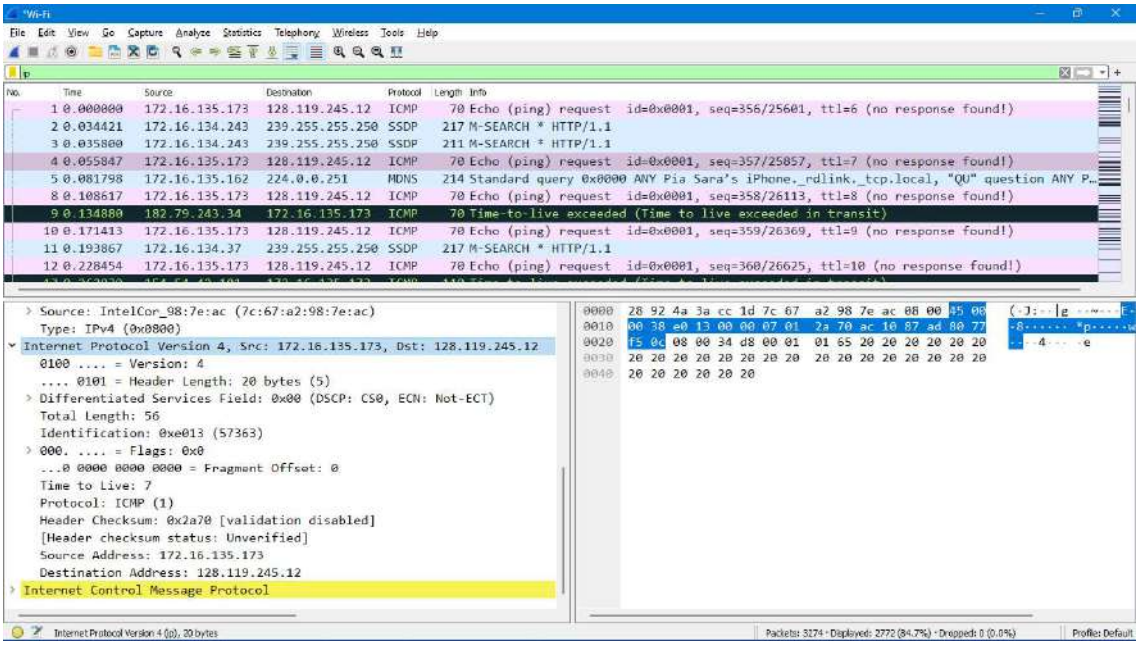
11. **Explain the purpose of the lease time. How long is the lease time in your experiment?**
    It is the amount of time the user is allowed to use the connection.
12. **What is the purpose of the DHCP release message? Does the DHCP server issue an acknowledgment of receipt of the client's DHCP request? What would happen if the client's DHCP release message is lost?**
    The DHCP release message ends the users lease. Yes it does issue an acknowledgment of receipt and if its lost it will just continue to run until the lease expires.

13. **Clear the bootp filter from your Wireshark window. Were any ARP packets sent or received during the DHCP packet-exchange period? If so, explain the purpose of those ARP packets.**

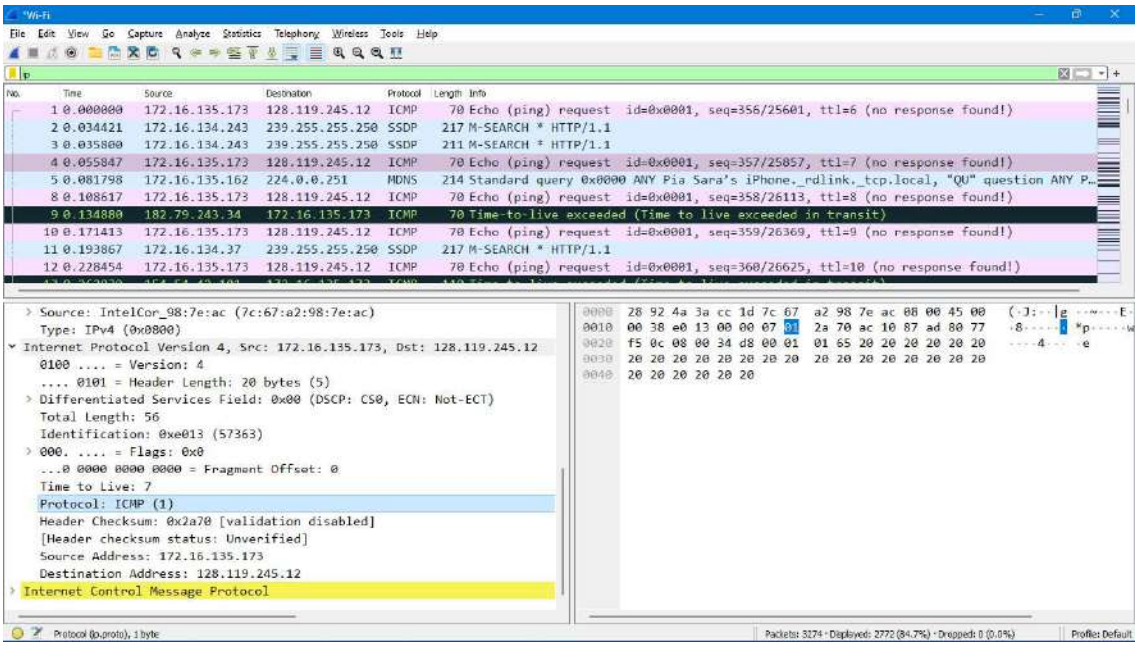
The ARP packets that show up are there in order to help sort out the MAC and IP addresses.

IP :-

1. **Select the first ICMP Echo Request message sent by your computer, and expand the Internet Protocol part of the packet in the packet details window.What is the IP address of your computer?**



2. **Within the IP packet header, what is the value in the upper layer protocol field?**

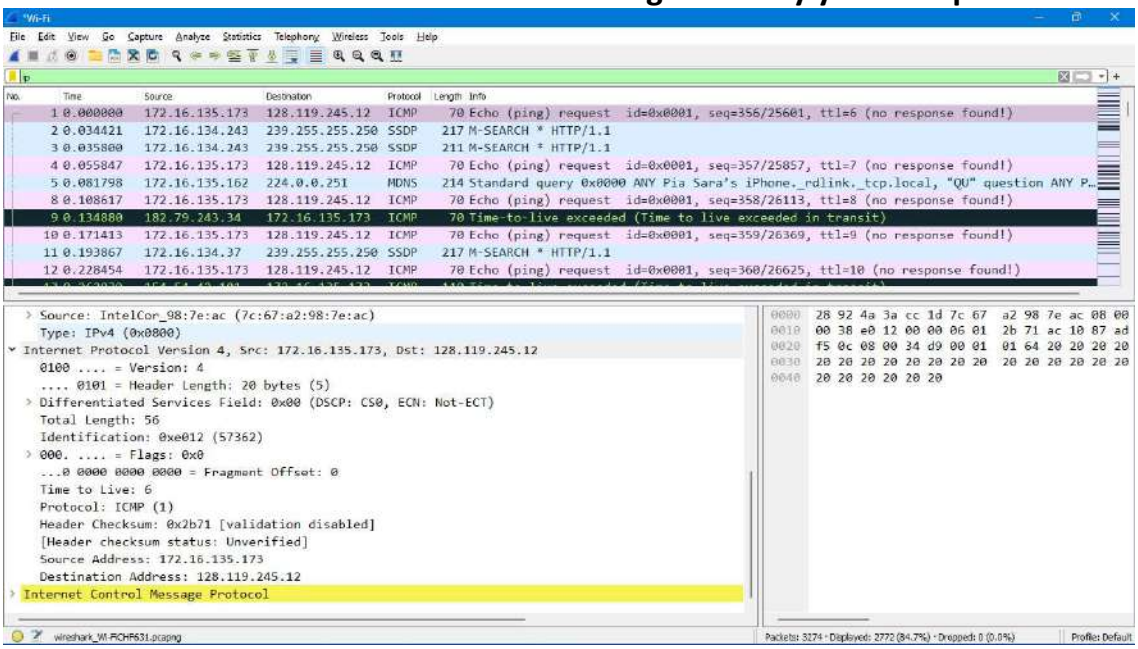


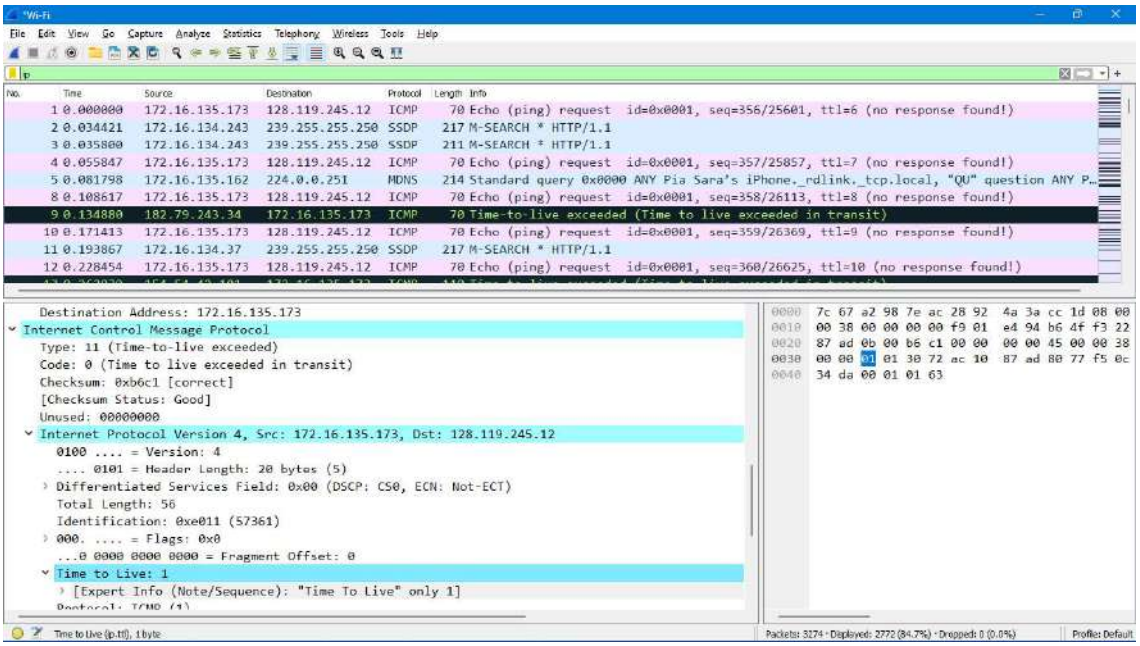
3. **How many bytes are in the IP header? How many bytes are in the payload of the IP datagram? Explain how you determined the number of payload bytes.**
   According to the figure 1, the header length is 20 bytes and the total length is 56 bytes. Therefore, the payload of the IP datagram should be 36 bytes (56 bytes – 20 bytes).

4. **Has this IP datagram been fragmented? Explain how you determined whether or not the datagram has been fragmented. Next, sort the traced packets according to IP source address by clicking on the Source column header; a small downward pointing arrow should appear next to the word Source. If the arrow points up, click on the Source column header again. Select the first ICMP Echo Request message sent by your computer, and expand the Internet Protocol portion in the "details of selected packet header" window. In the "listing of captured packets" window, you should see all of the subsequent ICMP messages (perhaps with additional interspersed packets sent by other protocols running on your computer) below this first ICMP. Use the down arrow to move through the ICMP messages sent by your computer.**

According to the figure, under flags section, the more fragments bit = 0, so the data is not fragmented.

5. **Which fields in the IP datagram always change from one datagram to the next within this series of ICMP messages sent by your computer?**



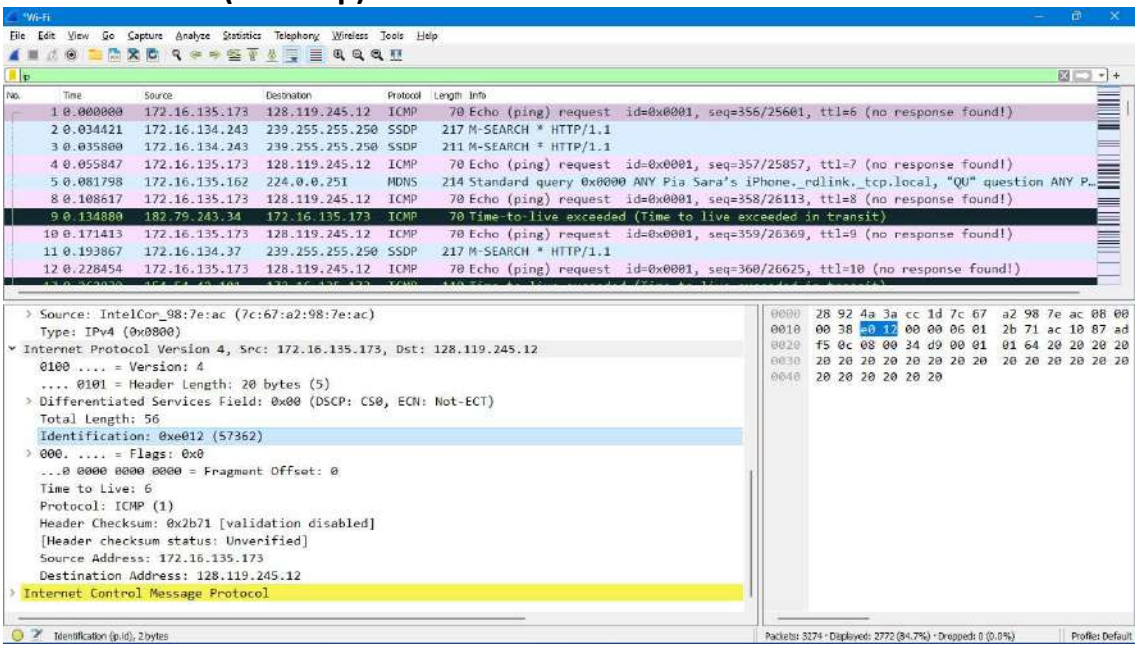


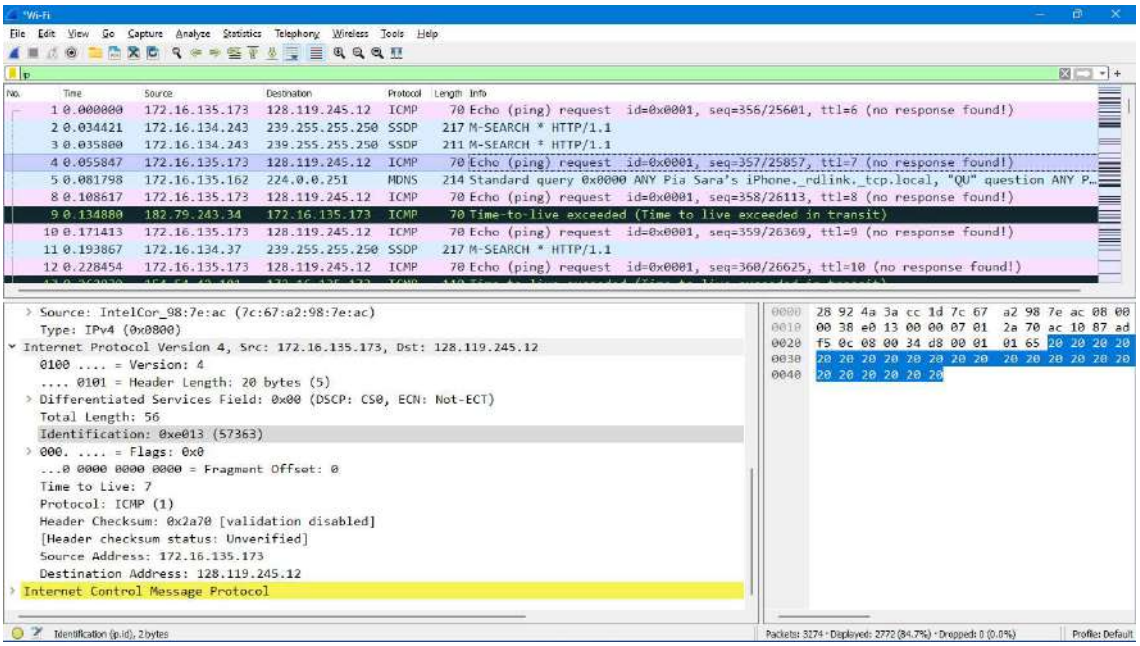
According to above two screenshots, identification, Time to live and Header checksum always change.

6. **Which fields stay constant? Which of the fields must stay constant? Which fields must change? Why?**
   The fields that stay constant are:
   Version (since we are using IPv4),
   header length (since these are UDP packets),
   source IP (since all packets are sent from my computer),
   destination IP (since we are sending to the same host),
   Differentiated Services (since all packets are UDP),
   Upper Layer Protocol (since these are UDP packets)

   The fields that must stay constant are:
   Version (since we are using IPv4),
   header length (since these are UDP packets),
   source IP (since all packets are sent from my computer),
   destination IP (since we are sending to the same host),
   Differentiated Services (since all packets are UDP),
   Upper Layer Protocol (since these are UDP packets)

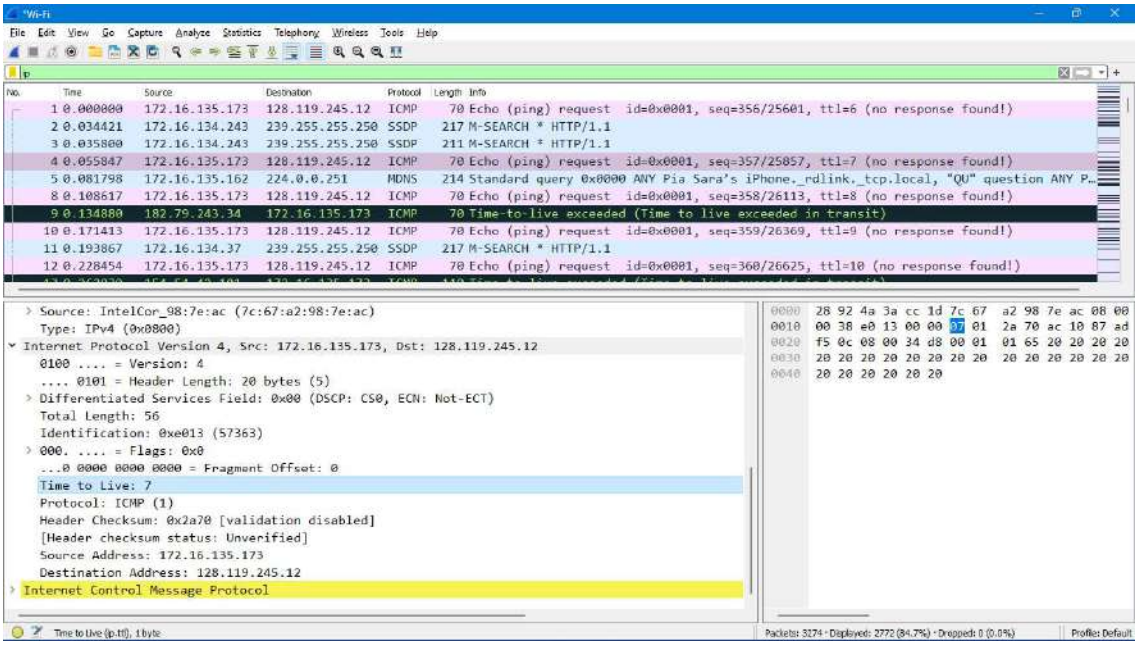
   The fields that must change are:
   Identification (IP packets have different ids),
   Time to live (traceroute increments each packet),
   Header checksum (since header changes)
7. **Describe the pattern you see in the values in the Identification field of the IP datagram Next (with the packets still sorted by source address) find the series of ICMP TTLexceeded replies sent to your computer by the nearest (first hop) router.**





According to above two screenshots, the pattern is the IP header Identification field increment with each UDP request.

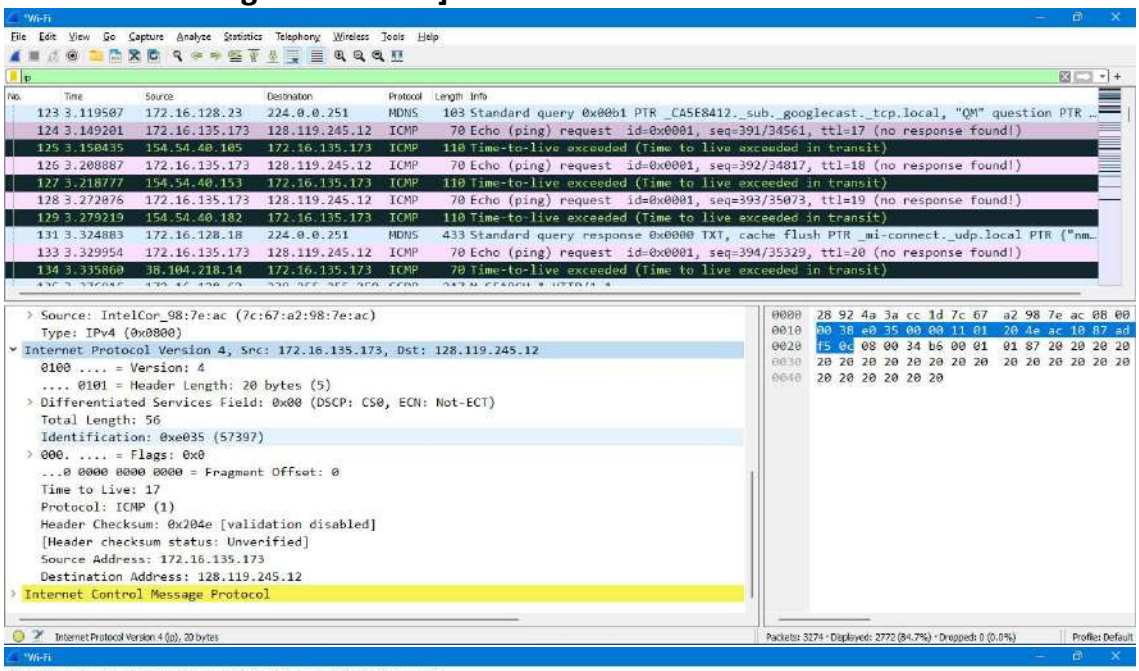
8. What is the value in the Identification field and the TTL field?

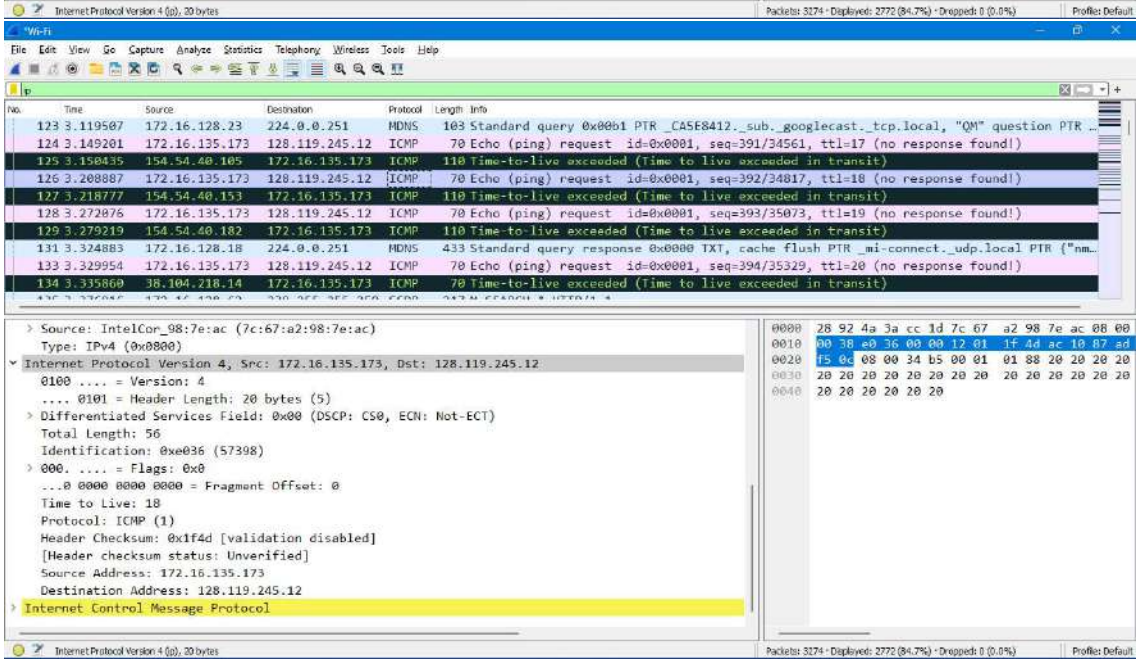


9. Do these values remain unchanged for all of the ICMP TTL-exceeded replies sent to your computer by the nearest (first hop) router? Why?

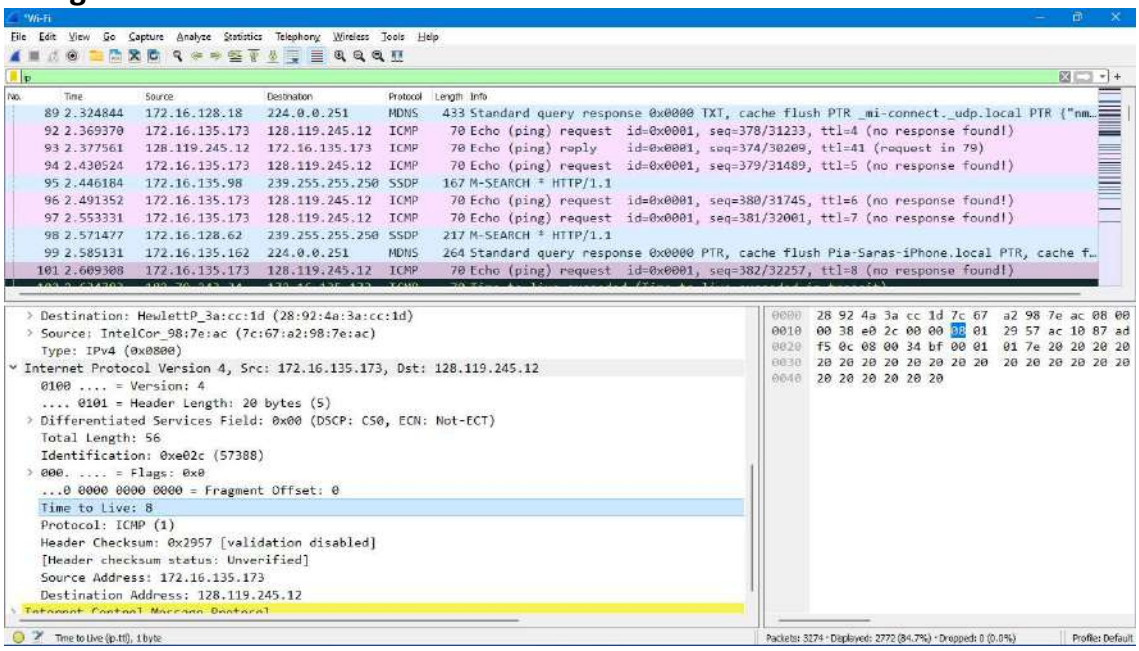
The values of identification field changes for all the ICMP TTL-exceeded replies since the identification field is a unique value. If two or more IP datagrams have the same identification value, then it means that these IP datagrams are fragments of a single large IP datagram. The TTL field was unchanged since the TTL for the nearest router is always the same (Linux, TTL 7).

10. Find the first ICMP Echo Request message that was sent by your computer after you changed the Packet Size in pingplotter to be 2000. Has that message been fragmented across more than one IP datagram? [Note: if you find your packet has not been fragmented, you should download the zip file http://gaia.cs.umass.edu/wireshark-labs/wireshark-traces.zip and extract the ipethereal-trace-1packet trace. If your computer has an Ethernet interface, a packet size of 2000 should cause fragmentation.3]

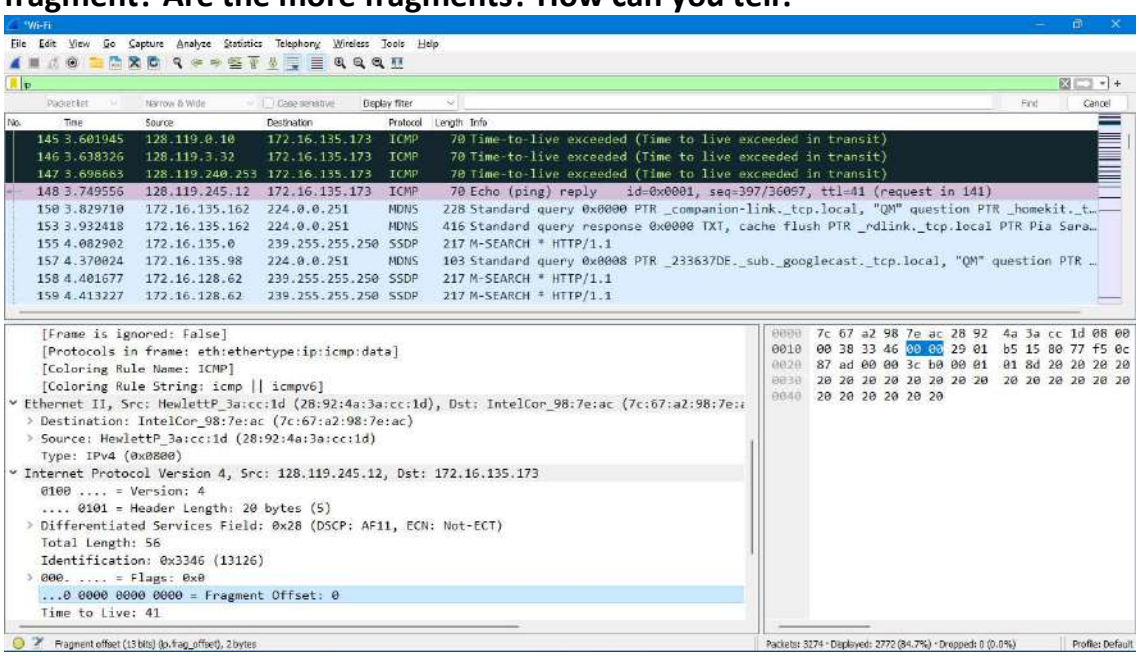




11. Print out the first fragment of the fragmented IP datagram. What information in the IP header indicates that the datagram been fragmented? What information in the IP header indicates whether this is the first fragment versus a latter fragment? How long is this IP datagram?



12. Print out the second fragment of the fragmented IP datagram. What information in the IP header indicates that this is not the first datagram fragment? Are the more fragments? How can you tell?



13. What fields change in the IP header between the first and second fragment? Now find the first ICMP Echo Request message that was sent by your computer after you changed the Packet Size in pingplotter to be 3500.

Total length, flags, fragment offset, and checksum.

14. How many fragments were created from the original datagram?

According to above screenshot, 0 packets created from the original datagram.

15. What fields change in the IP header among the fragments?

Fragment offset, checksum. Moreover, for the first two packets, the total length is 56 with the more fragments flag set to 0, and the third packet's total length is 56 with the more fragments flag set to 0.

DNS

1. **Run nslookup to obtain the IP address of a Web server in Asia. What is the IP address of that server?**

```
C:\Users\admin>nslookup asdu.ait.ac.th
Server:  UnKnown
Address:  172.16.128.1

Name:    asdu.ait.ac.th
```

2. **Run nslookup to determine the authoritative DNS servers for a university in Europe.**

```
C:\Users\admin>nslookup -type=NS www.cam.ac.uk
Server:  UnKnown
Address:  172.16.128.1

cam.ac.uk
        primary name server = primary.dns.cam.ac.uk
        responsible mail addr = hostmaster.cam.ac.uk
        serial  = 1682357798
        refresh = 1800 (30 mins)
        retry   = 900 (15 mins)
        expire  = 604800 (7 days)
        default TTL = 3600 (1 hour)

C:\Users\admin>
```

3. **Run nslookup so that one of the DNS servers obtained in Question 2 is queried for the mail servers for Yahoo! mail. that is its IP address?**

```
C:\Users\admin>nslookup www.cam.ac.uk mail.yahoo.com
Server:  e1-ha.ycpi.inb.yahoo.com
Address:  27.123.43.204

Non-authoritative answer:
Name:    www.cam.ac.uk
Addresses:  2a05:b400:5:270::80e8:8408
          128.232.132.8
```

```
C:\Windows\system32\cmd.exe

C:\Users\admin>ipconfig /all

Windows IP Configuration

   Host Name . . . . . . . . . . . . : DESKTOP-PG0Q864
   Primary Dns Suffix  . . . . . . . :
   Node Type . . . . . . . . . . . . : Hybrid
   IP Routing Enabled. . . . . . . . : No
   WINS Proxy Enabled. . . . . . . . : No

Ethernet adapter Ethernet:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : Realtek PCIe GbE Family Controller
   Physical Address. . . . . . . . . : 54-EE-75-D4-A8-13
   DHCP Enabled. . . . . . . . . . . : Yes
   Autoconfiguration Enabled . . . . : Yes

Ethernet adapter Ethernet 3:

   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : VirtualBox Host-Only Ethernet Adapter
   Physical Address. . . . . . . . . : 0A-00-27-00-00-06
   DHCP Enabled. . . . . . . . . . . : No
   Autoconfiguration Enabled . . . . : Yes
   Link-local IPv6 Address . . . . . : fe80::21dd:c560:d3b3:83f4%6(Preferred)
   IPv4 Address. . . . . . . . . . . : 192.168.56.1(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . :
   DHCPv6 IAID . . . . . . . . . . . : 722075607
   DHCPv6 Client DUID. . . . . . . . : 00-01-00-01-2A-30-37-39-54-EE-75-D4-A8-13
   NetBIOS over Tcpip. . . . . . . . : Enabled

Wireless LAN adapter Local Area Connection* 1:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : Microsoft Wi-Fi Direct Virtual Adapter
   Physical Address. . . . . . . . . : 7C-67-A2-9B-7E-AD
   DHCP Enabled. . . . . . . . . . . : Yes
   Autoconfiguration Enabled . . . . : Yes

Wireless LAN adapter Local Area Connection* 2:
```

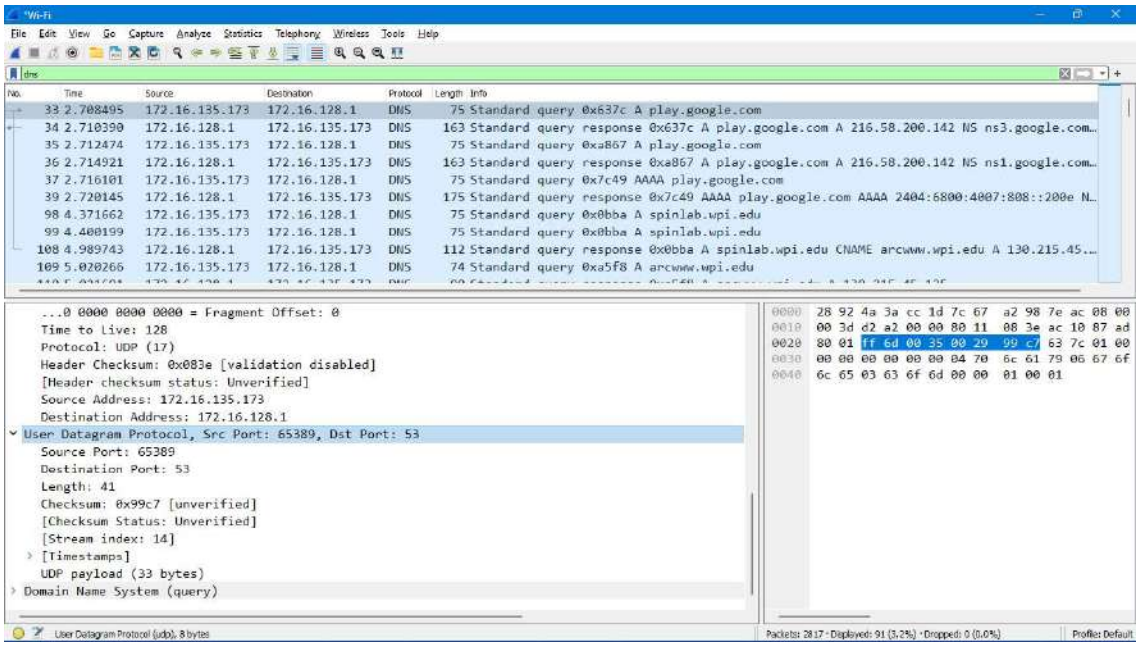
```
C:\Windows\system32\cmd.exe

Wireless LAN adapter Local Area Connection* 2:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : Microsoft Wi-Fi Direct Virtual Adapter #2
   Physical Address. . . . . . . . . : 7E-67-A2-9B-7E-AC
   DHCP Enabled. . . . . . . . . . . : No
   Autoconfiguration Enabled . . . . : Yes

Wireless LAN adapter Wi-Fi:

   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : Intel(R) Dual Band Wireless-AC 3165
   Physical Address. . . . . . . . . : 7C-67-A2-9B-7E-AC
   DHCP Enabled. . . . . . . . . . . : Yes
   Autoconfiguration Enabled . . . . : Yes
   Link-local IPv6 Address . . . . . : fe80::3a33:f890:7b51:7963%4(Preferred)
   IPv4 Address. . . . . . . . . . . : 172.16.135.173(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.248.0
   Lease Obtained. . . . . . . . . . : 24 April 2023 19:45:38
   Lease Expires . . . . . . . . . . : 25 April 2023 04:05:16
   Default Gateway . . . . . . . . . : 172.16.128.1
   DHCP Server . . . . . . . . . . . : 172.16.128.1
   DHCPv6 IAID . . . . . . . . . . . : 58484642
   DHCPv6 Client DUID. . . . . . . . : 00-01-00-01-2A-30-37-39-54-EE-75-D4-A8-13
   DNS Servers . . . . . . . . . . . : 172.16.128.1
   NetBIOS over Tcpip. . . . . . . . : Enabled

C:\Users\admin>ipconfig /displaydns

Windows IP Configuration

    218.5.54.154.in-addr.arpa
    ----------------------------------------
    Record Name . . . . . : 218.5.54.154.in-addr.arpa
    Record Type . . . . . : 12
    Time To Live  . . . . : 5414
    Data Length . . . . . : 8
    Section . . . . . . . : Answer
    PTR Record  . . . . . : be2979.ccr21.elp02.atlas.cogentco.com


    arcwww.wpi.edu
    ----------------------------------------
    Record Name . . . . . : arcwww.wpi.edu
```

```
C:\Windows\system32\cmd.exe

    arcwww.wpi.edu
    ----------------------------------------
    Record Name . . . . . : arcwww.wpi.edu
    Record Type . . . . . : 1
    Time To Live  . . . . : 531
    Data Length . . . . . : 4
    Section . . . . . . . : Answer
    A (Host) Record . . . : 130.215.45.125


    i.ytimg.com
    ----------------------------------------
    Record Name . . . . . : i.ytimg.com
    Record Type . . . . . : 28
    Time To Live  . . . . : 28
    Data Length . . . . . : 16
    Section . . . . . . . : Answer
    AAAA Record . . . . . : 2404:6800:4007:81b::2016


    Record Name . . . . . : i.ytimg.com
    Record Type . . . . . : 28
    Time To Live  . . . . : 28
    Data Length . . . . . : 16
    Section . . . . . . . : Answer
    AAAA Record . . . . . : 2404:6800:4007:81c::2016


    Record Name . . . . . : i.ytimg.com
    Record Type . . . . . : 28
    Time To Live  . . . . : 28
    Data Length . . . . . : 16
    Section . . . . . . . : Answer
    AAAA Record . . . . . : 2404:6800:4007:81e::2016


    Record Name . . . . . : i.ytimg.com
    Record Type . . . . . : 28
    Time To Live  . . . . : 28
    Data Length . . . . . : 16
    Section . . . . . . . : Answer
    AAAA Record . . . . . : 2404:6800:4007:81a::2016


    i.ytimg.com
```

```
C:\Windows\system32\cmd.exe
    Record Name . . . . . : i.ytimg.com
    Record Type . . . . . : 1
    Time To Live  . . . . : 44
    Data Length . . . . . : 4
    Section . . . . . . . : Answer
    A (Host) Record . . . : 142.250.183.246


    Record Name . . . . . : i.ytimg.com
    Record Type . . . . . : 1
    Time To Live  . . . . : 44
    Data Length . . . . . : 4
    Section . . . . . . . : Answer
    A (Host) Record . . . : 142.250.193.118


    Record Name . . . . . : i.ytimg.com
    Record Type . . . . . : 1
    Time To Live  . . . . : 44
    Data Length . . . . . : 4
    Section . . . . . . . : Answer
    A (Host) Record . . . : 142.250.193.150


    Record Name . . . . . : i.ytimg.com
    Record Type . . . . . : 1
    Time To Live  . . . . : 44
    Data Length . . . . . : 4
    Section . . . . . . . : Answer
    A (Host) Record . . . : 142.250.196.182


    Record Name . . . . . : i.ytimg.com
    Record Type . . . . . : 1
    Time To Live  . . . . : 44
    Data Length . . . . . : 4
    Section . . . . . . . : Answer
    A (Host) Record . . . : 172.217.163.182


    Record Name . . . . . : i.ytimg.com
    Record Type . . . . . : 1
    Time To Live  . . . . : 44
    Data Length . . . . . : 4
    Section . . . . . . . : Answer
    A (Host) Record . . . : 172.217.163.214
```

```
C:\Users\admin>ipconfig /flushdns

Windows IP Configuration

Successfully flushed the DNS Resolver Cache.
```

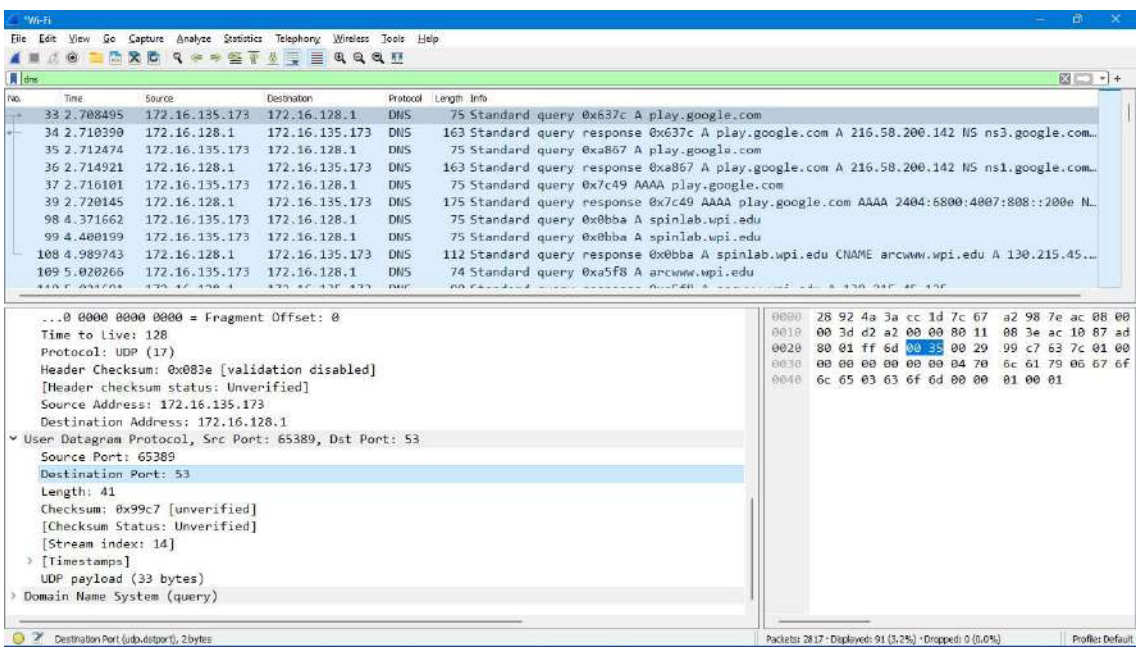
4. **Locate the DNS query and response messages. Are then sent over UDP or TCP?**



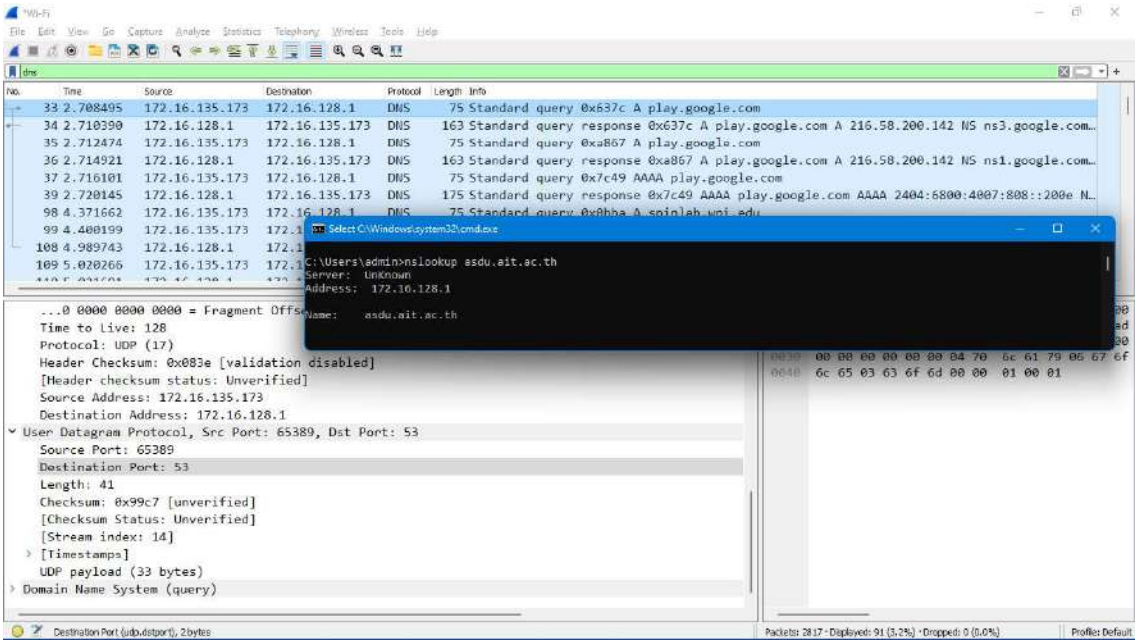
5. **What is the destination port for the DNS query message? What is the source port of DNS response message?**

The destination port is 53

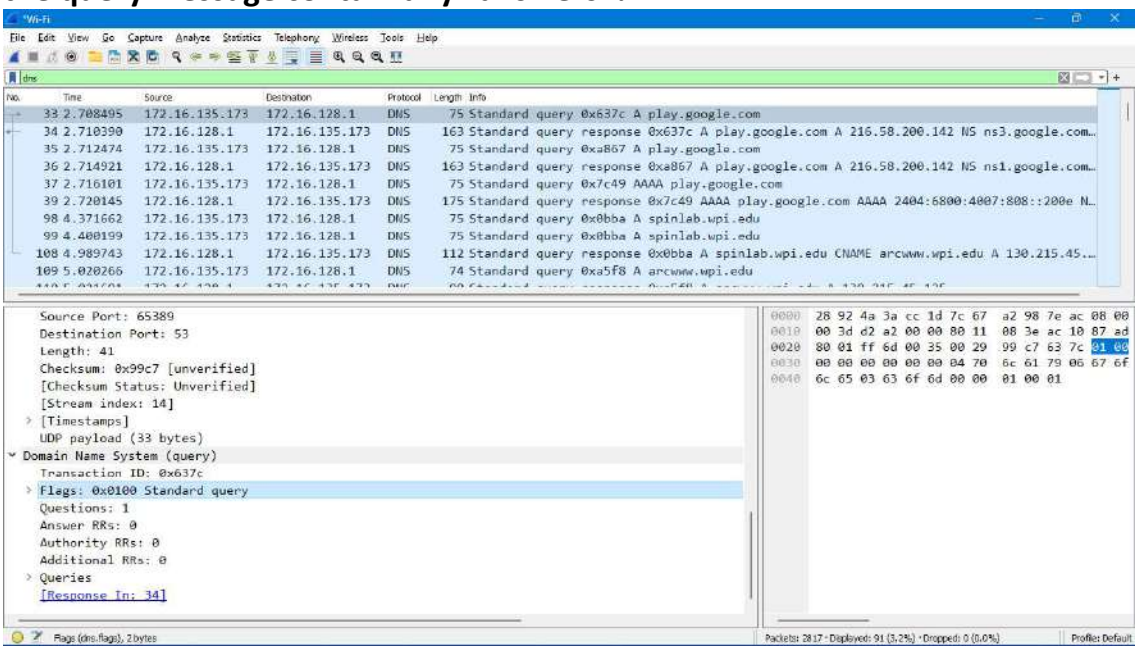
The source port is 65389

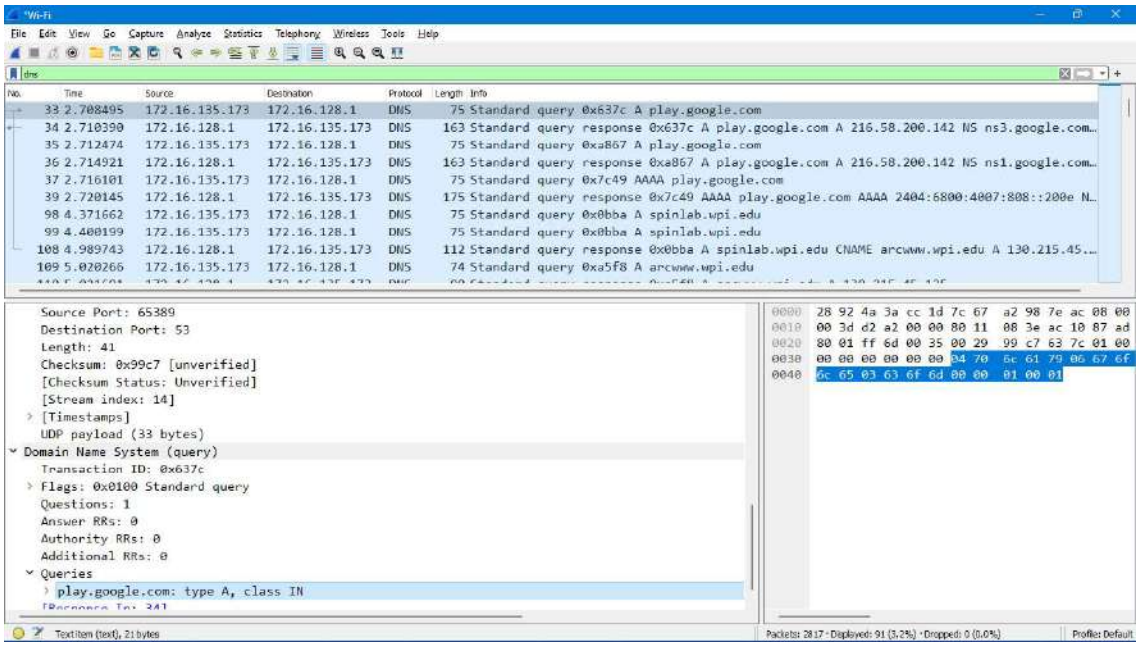


6. **To what IP address is the DNS query message sent? Use ipconfig to determine the IP address of your local DNS server. Are these two IP addresses the same?**

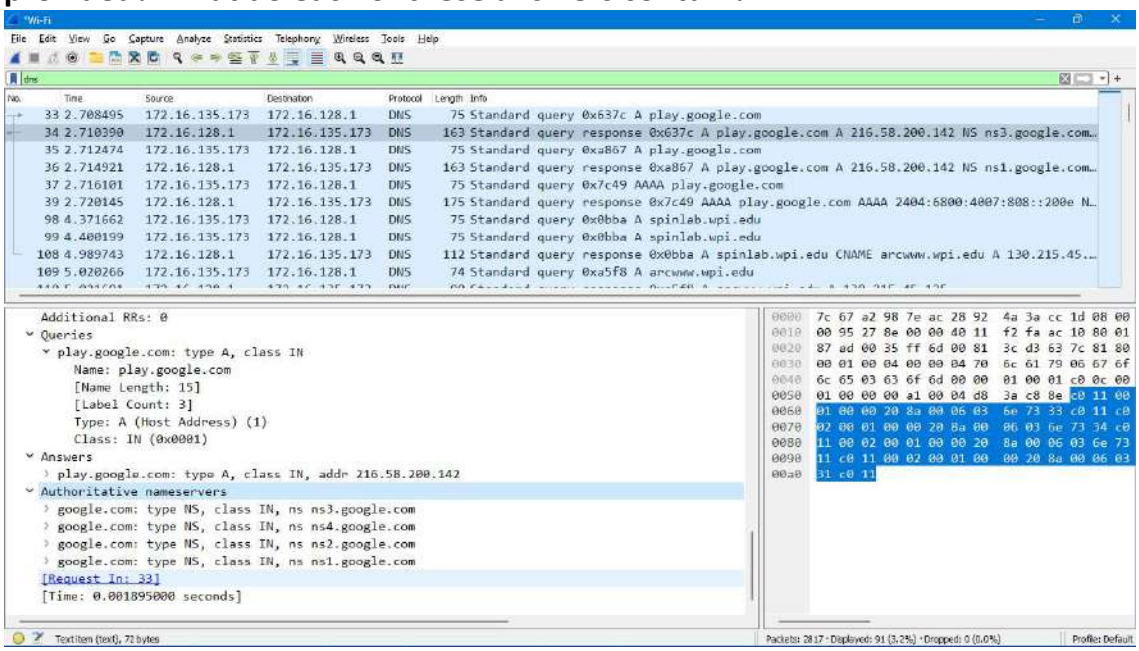


7. **Examine the DNS query message. What "Type" of DNS query is it? Does the query message contain any "answers"?**

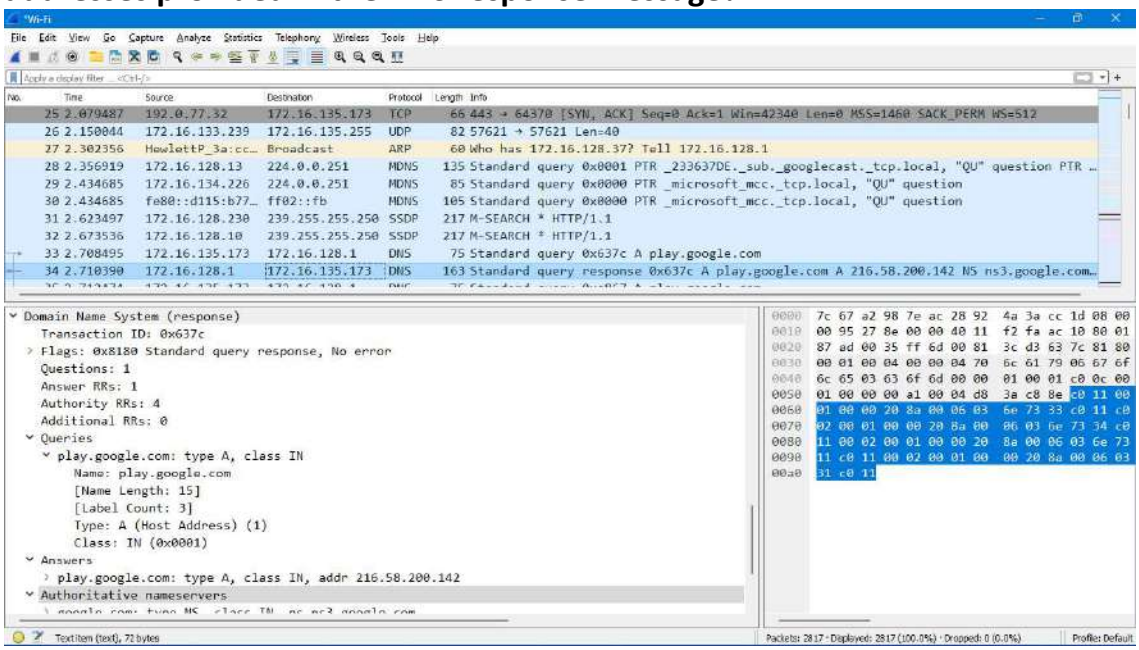




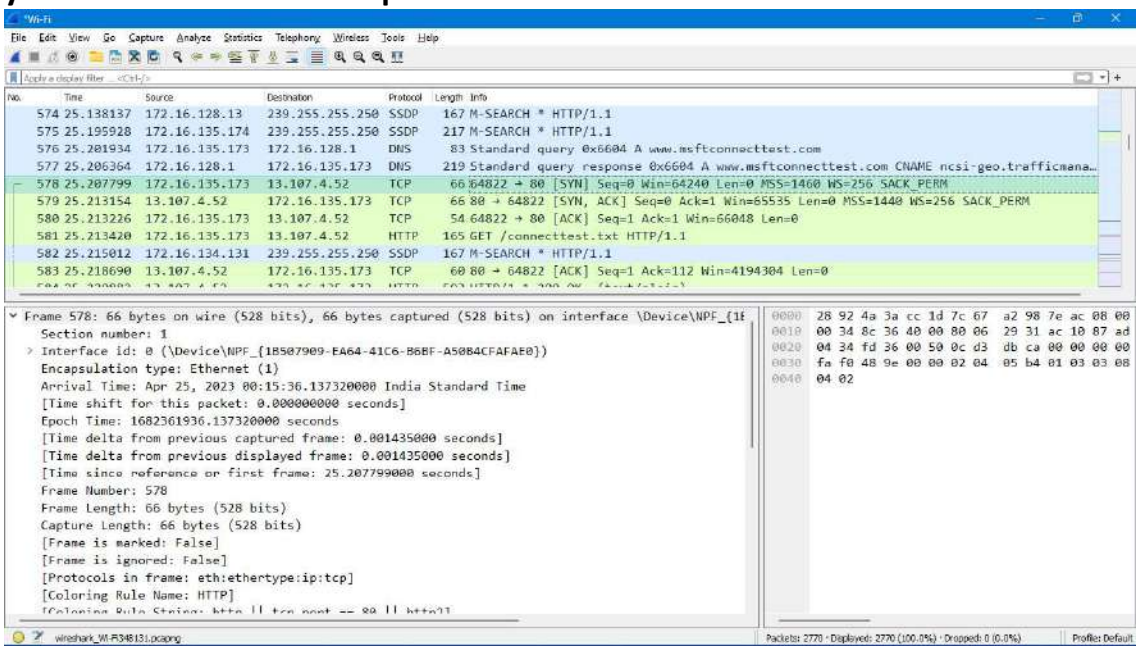
8. **Examine the DNS response message. How many "answers" are provided? What do each of these answers contain?**



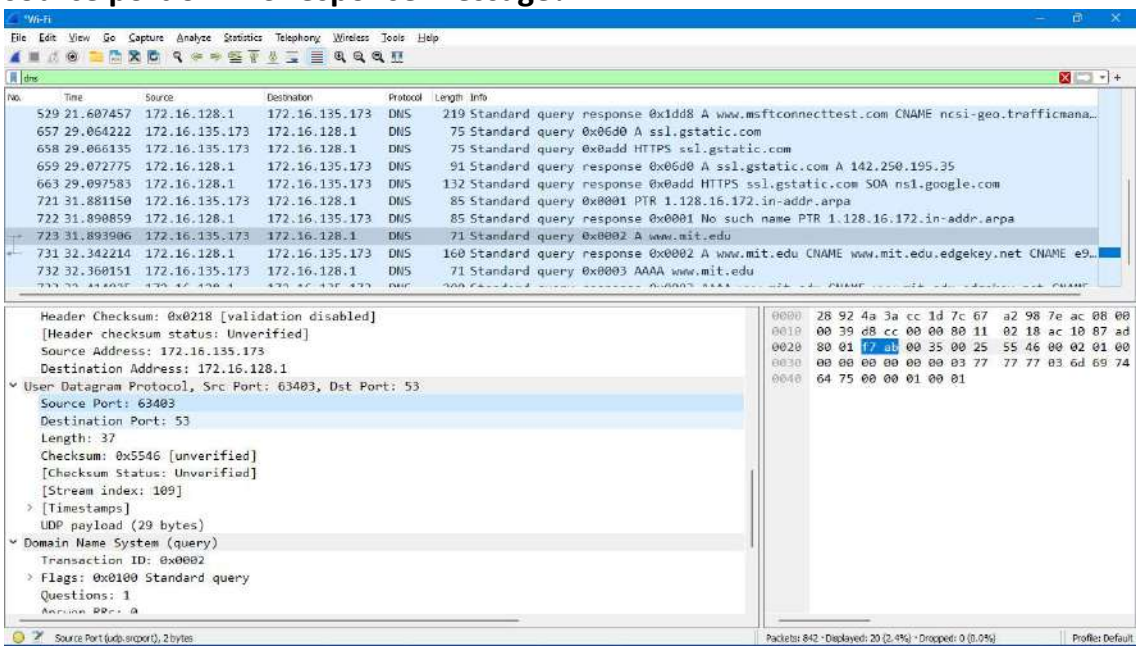
9. **Consider the subsequent TCP SYN packet sent by your host. Does the destination IP address of the SYN packet correspond to any of the IP addresses provided in the DNS response message?**

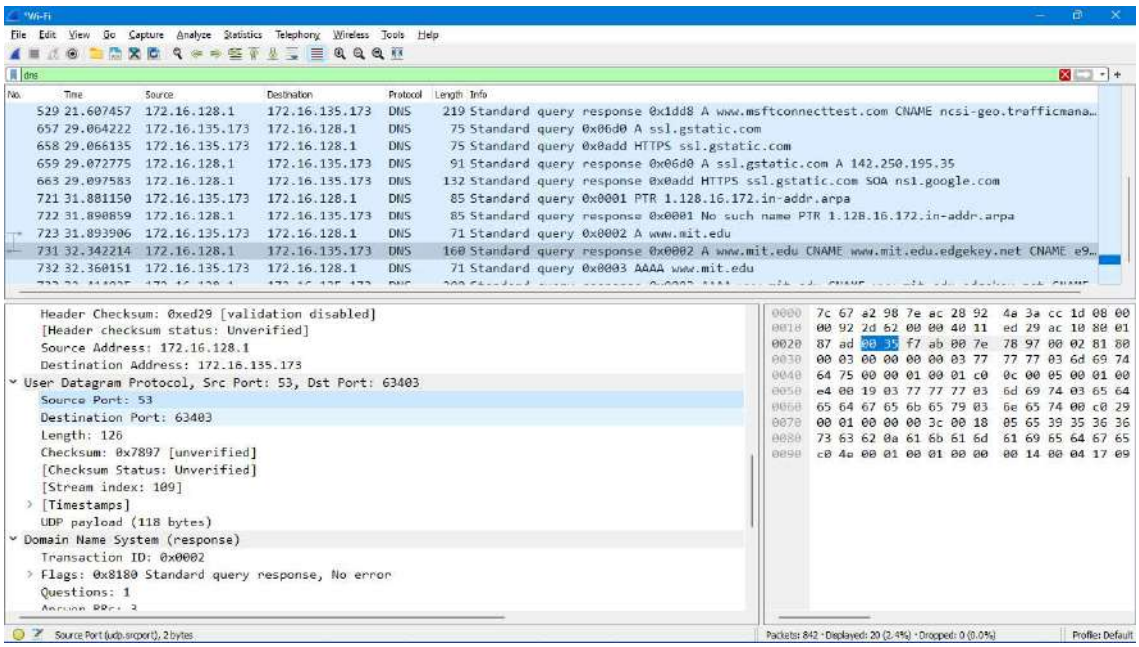


10. **This web page contains images. Before retrieving each image, does your host issue new DNS queries?**

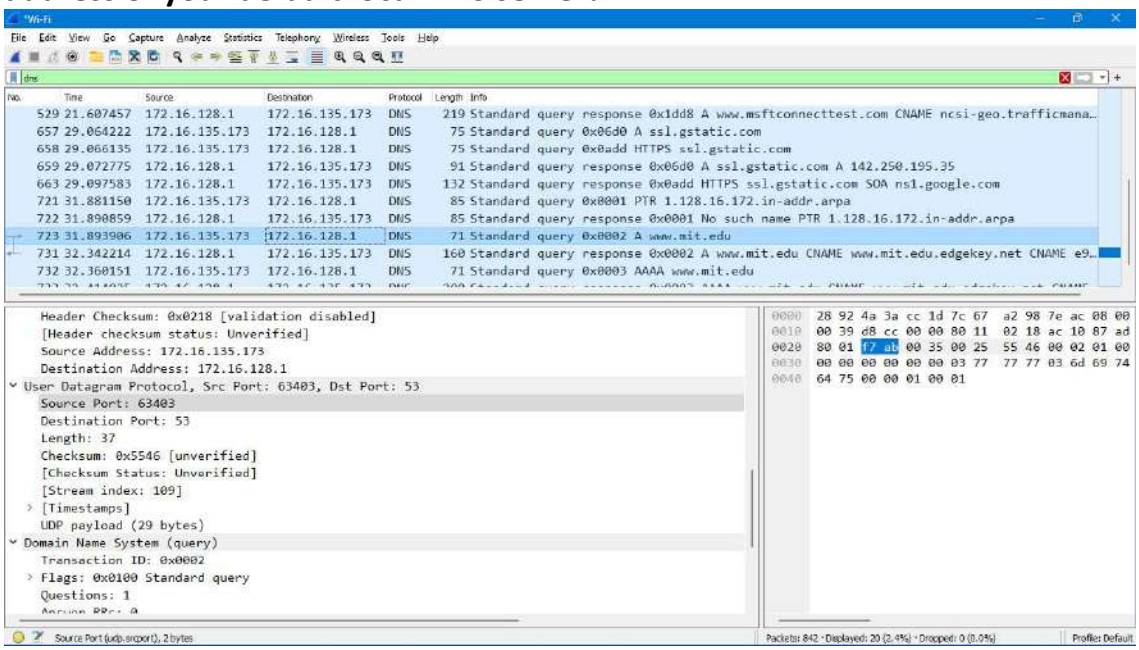


11. What is the destination port for the DNS query message? What is the source port of DNS response message?

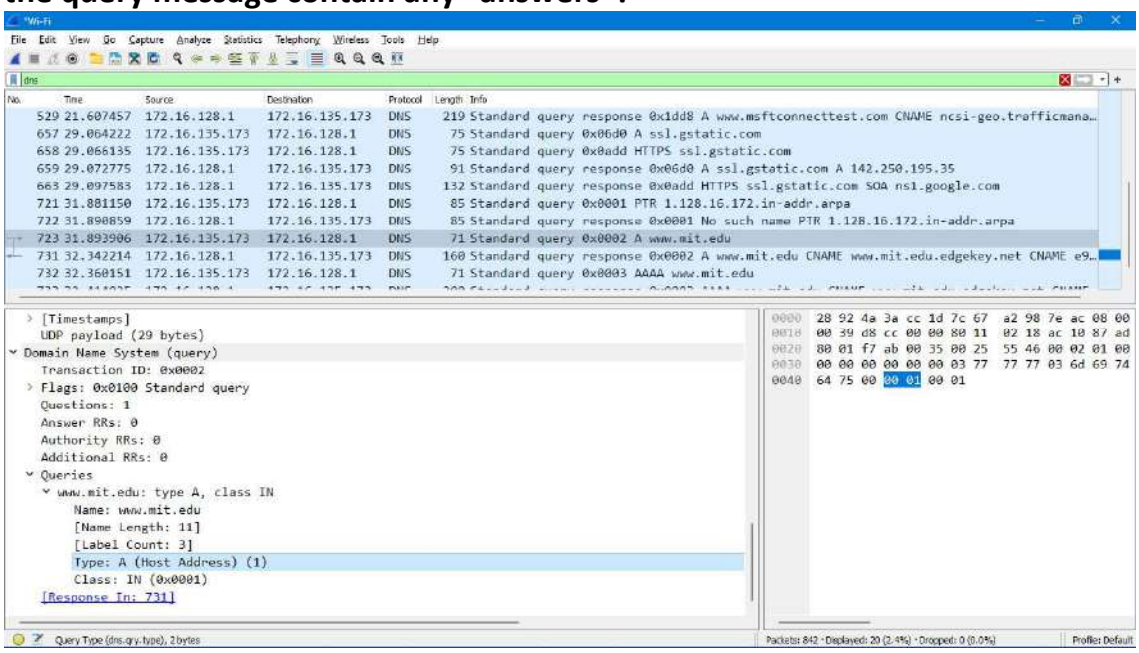


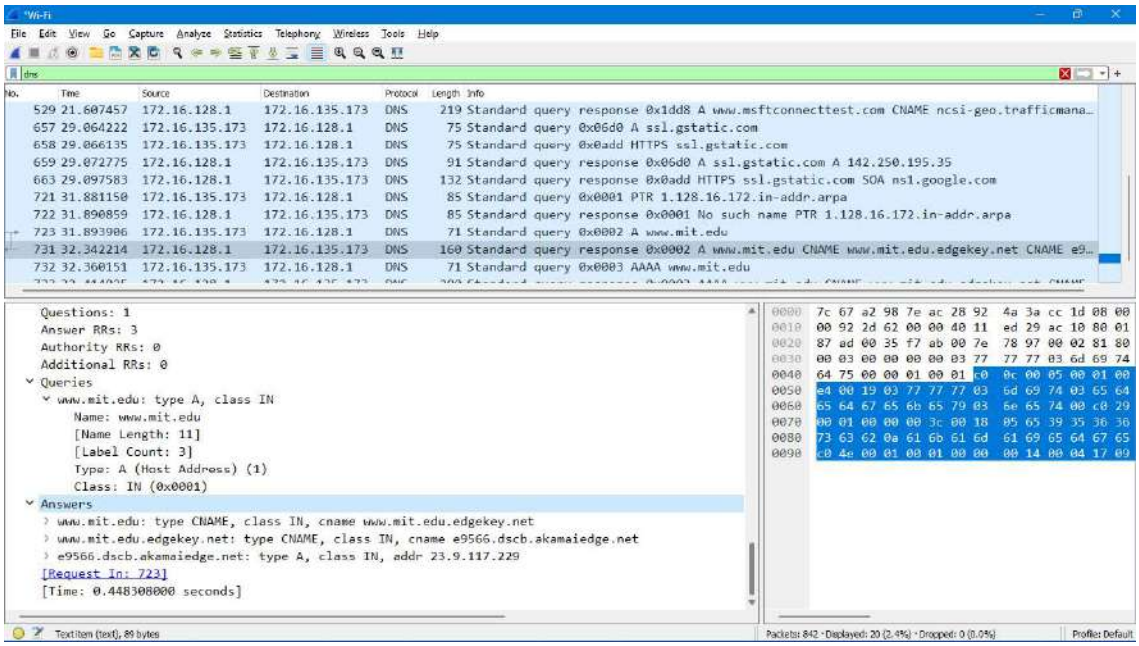


12.To what IP address is the DNS query message sent? Is this the IP address of your default local DNS server?

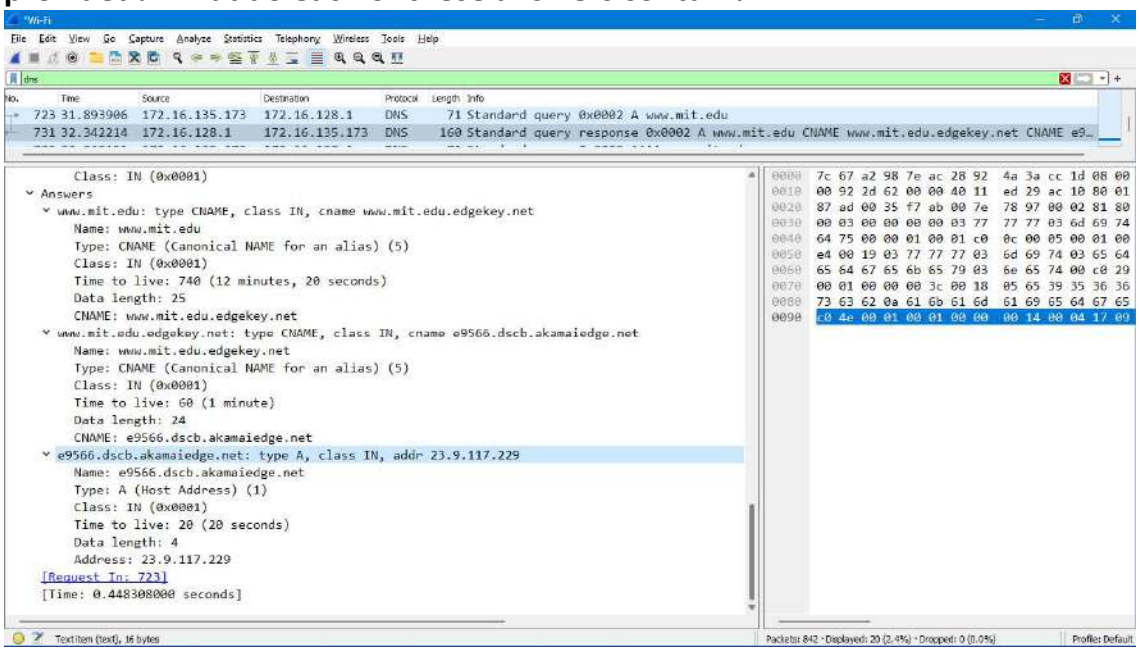


13.Examine the DNS query message. What "Type" of DNS query is it? Does the query message contain any "answers"?

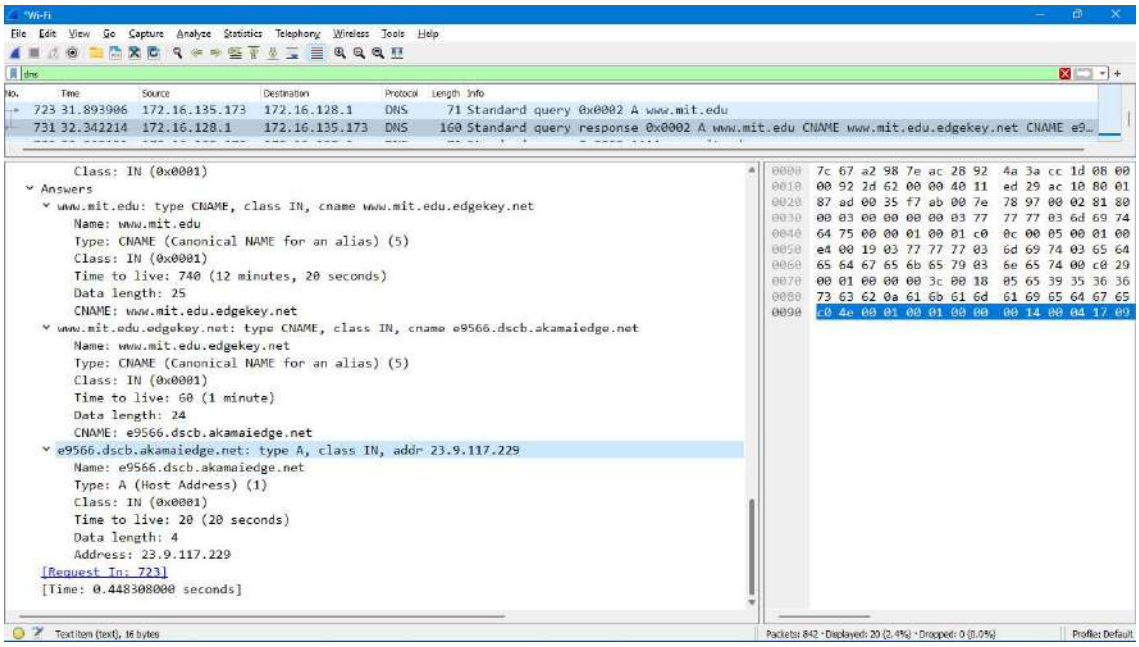




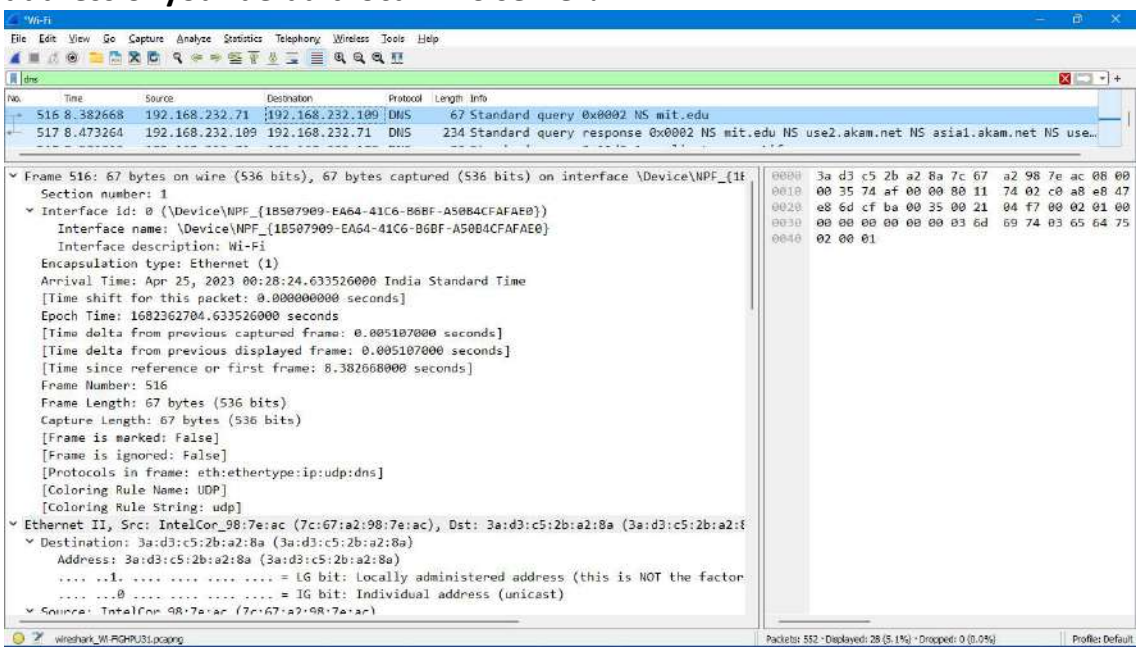
14. Examine the DNS response message. How many "answers" are provided? What do each of these answers contain?



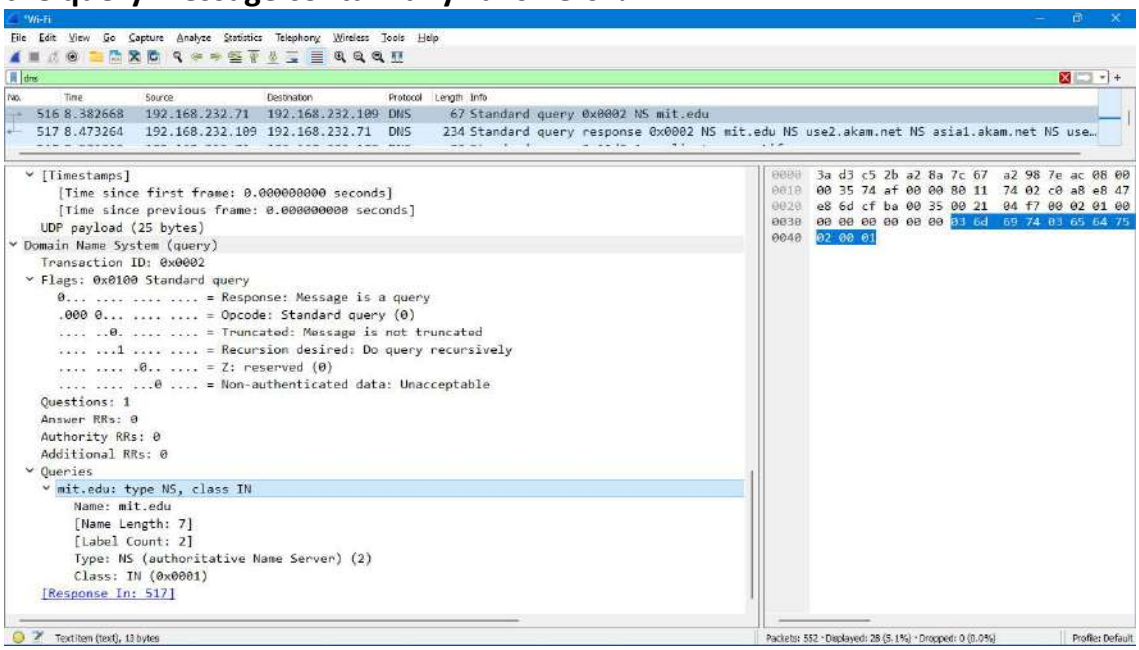
15. Provide a screenshot.



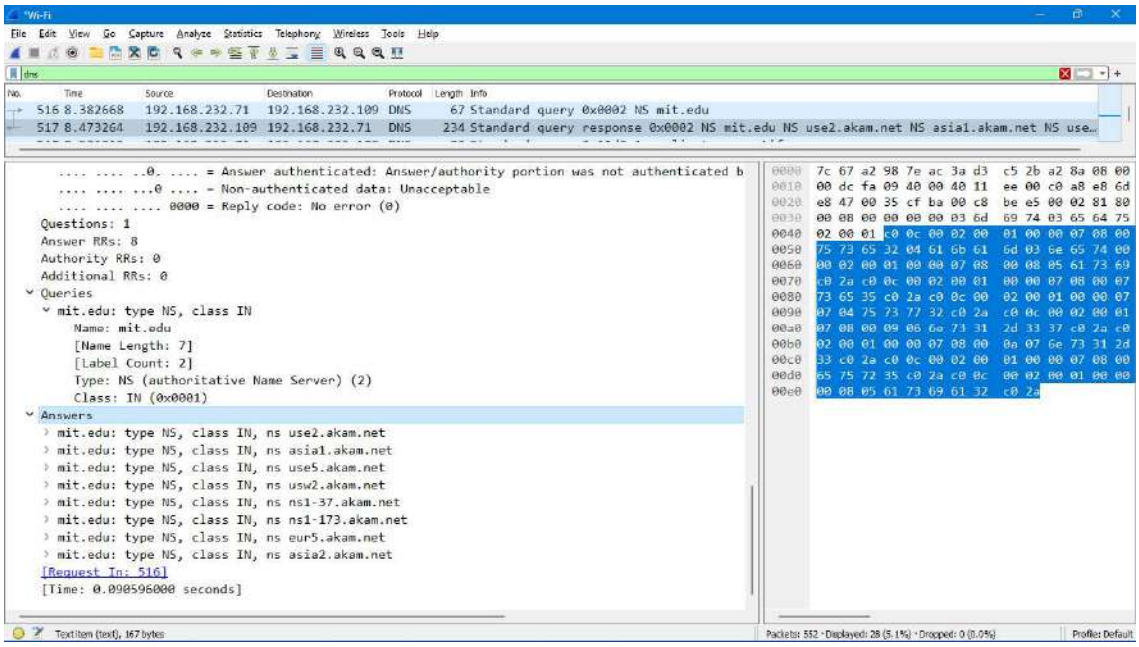
16. To what IP address is the DNS query message sent? Is this the IP address of your default local DNS server?



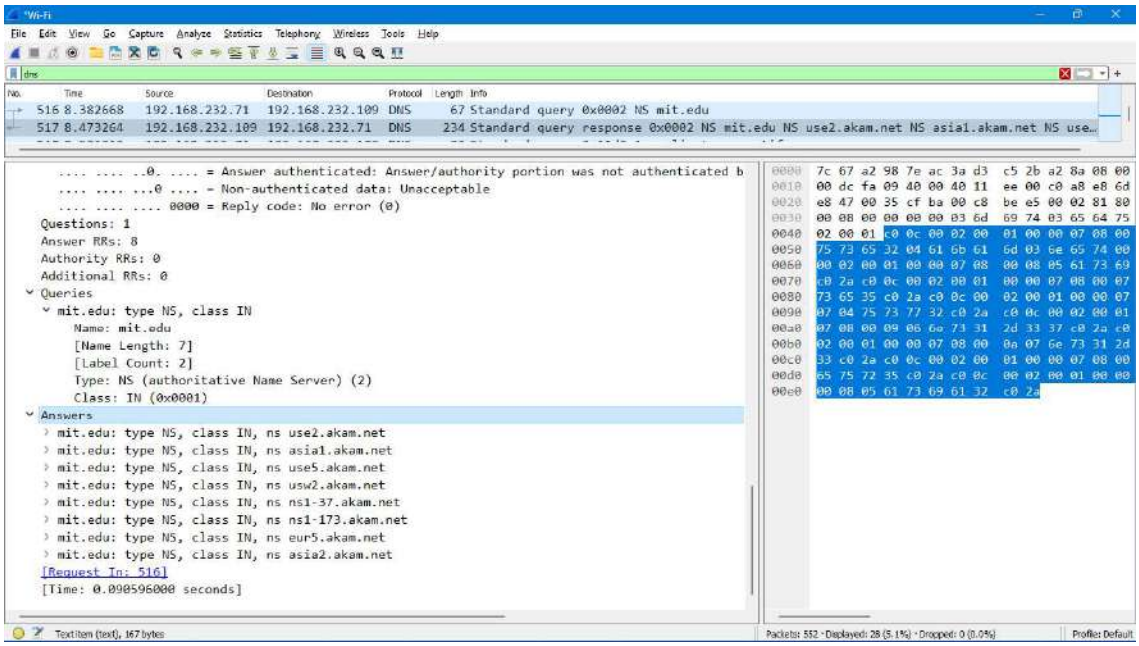
17. Examine the DNS query message. What "Type" of DNS query is it? Does the query message contain any "answers"?



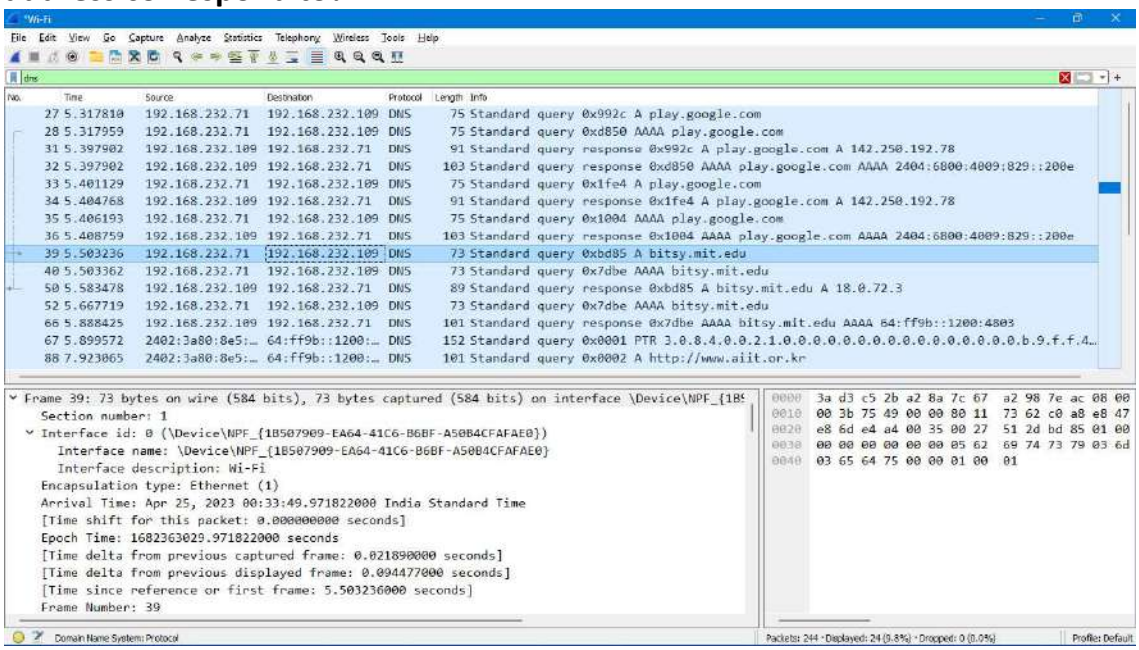
18. Examine the DNS response message. What MIT nameservers does the response message provide? Does this response message also provide the IP addresses of the MIT namesers?

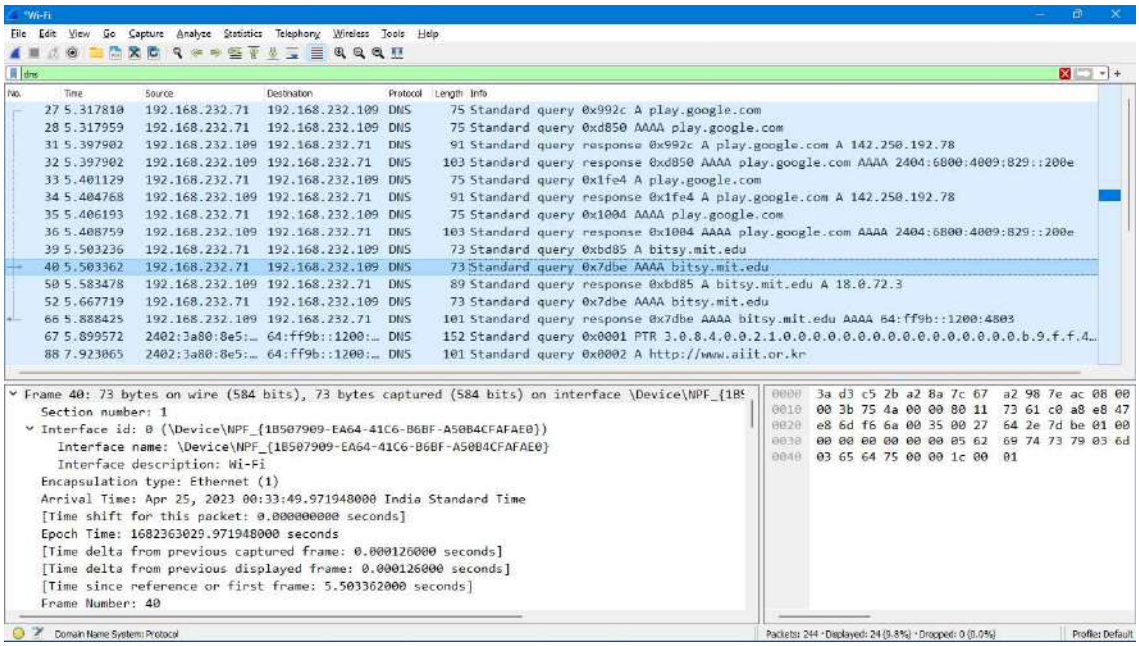


19. Provide a screenshot.

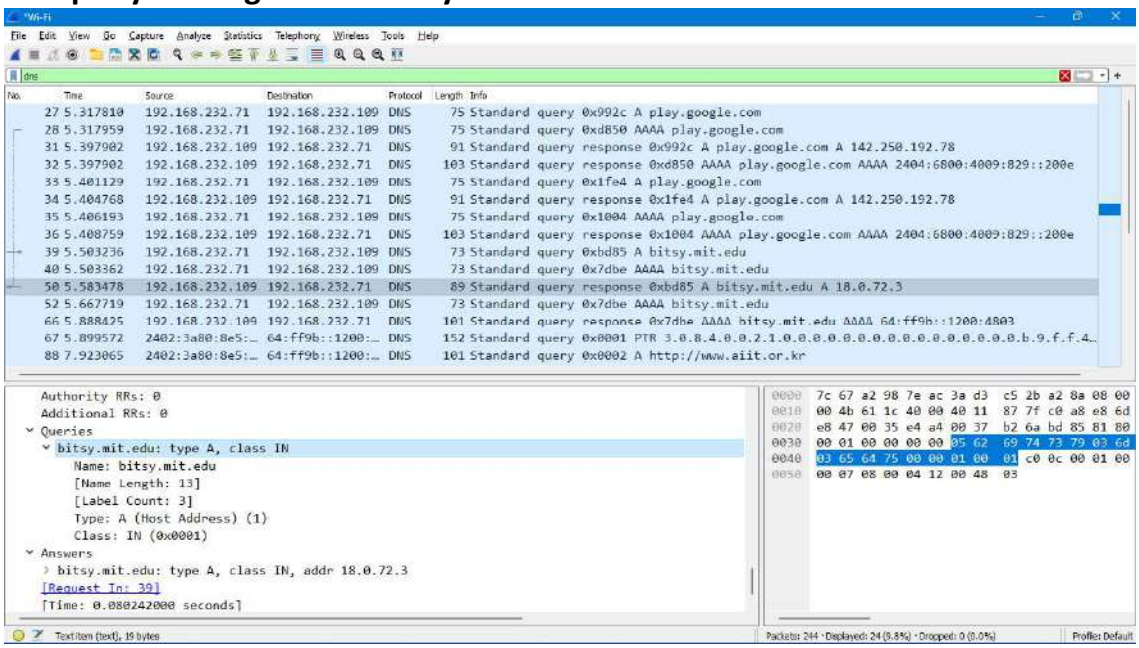


20. To what IP address is the DNS query message sent? Is this the IP address of your default local DNS server? If not, what does the IP address correspond to?

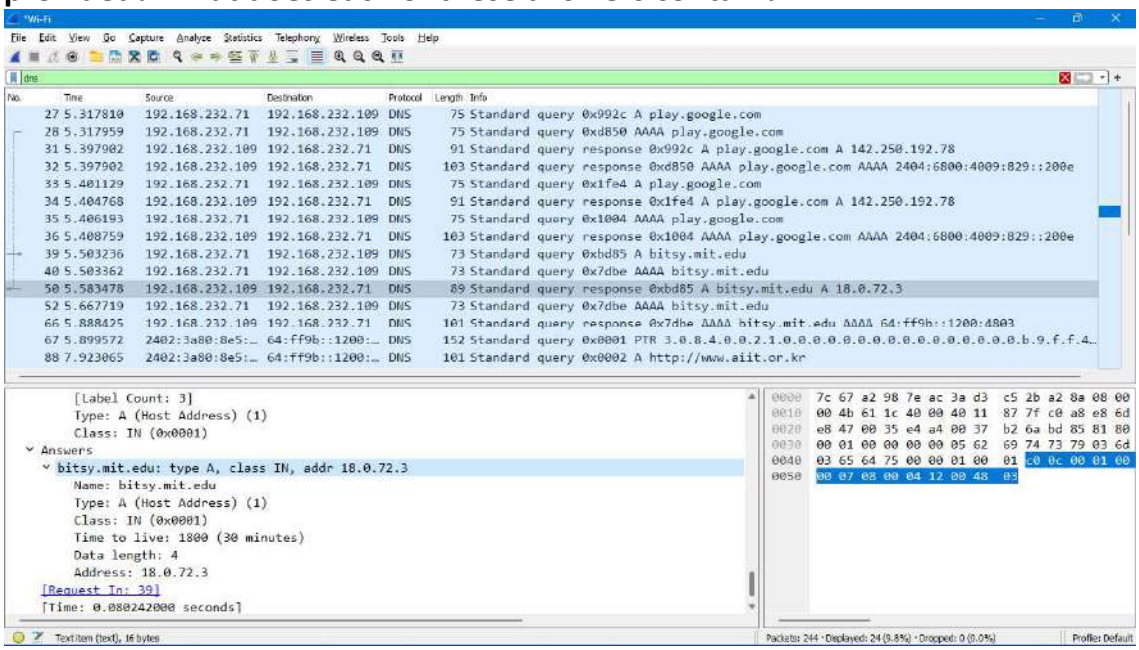




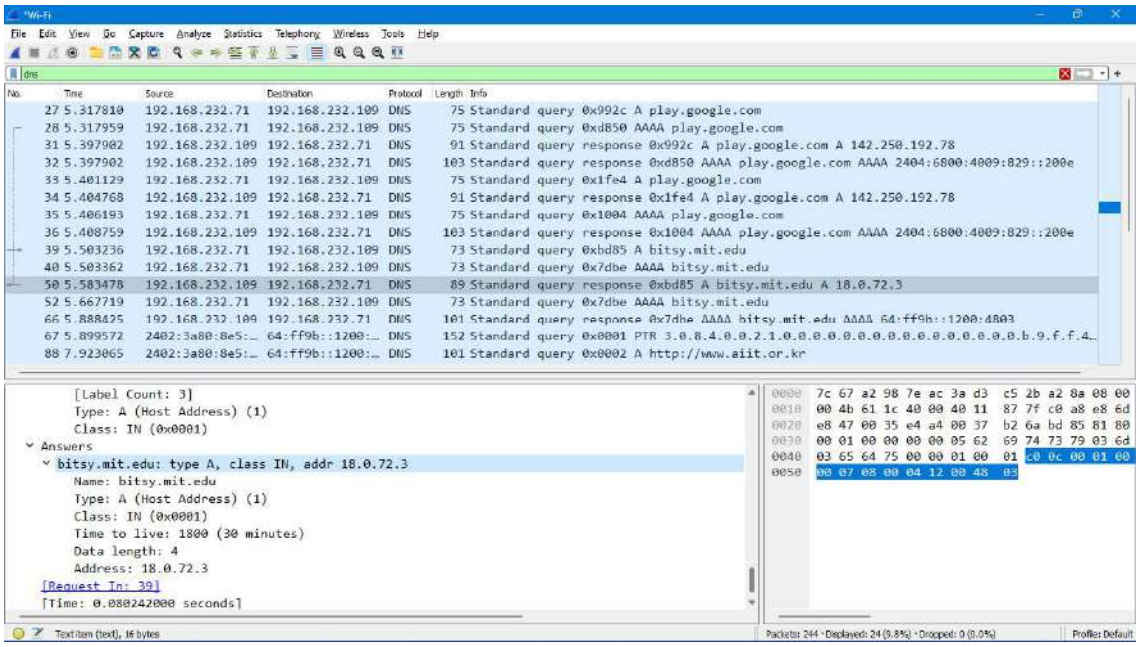
21. Examine the DNS query message. What "Type" of DNS query is it? Does the query message contain any "answers"?



22. Examine the DNS response message. How many “answers” are provided? What does each of these answers contain?

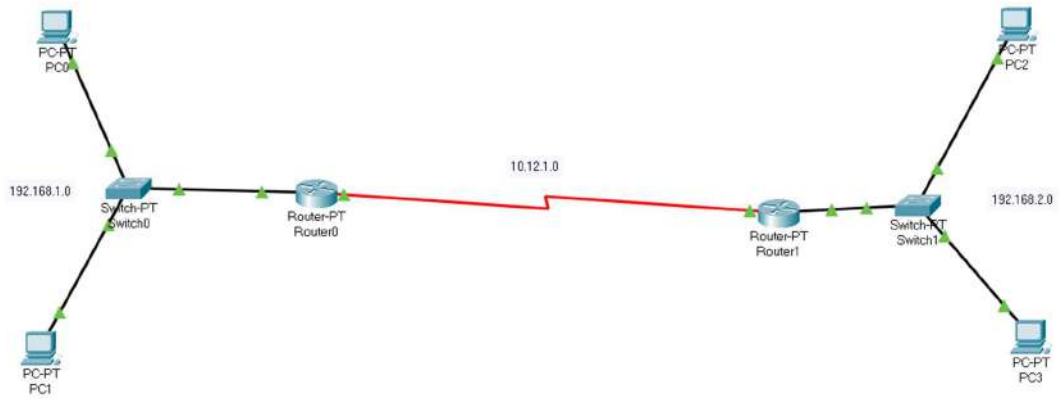


23. Provide a screenshot.



Interconnecting routers in Cisco packet tracer :-

Topology Diagram :-

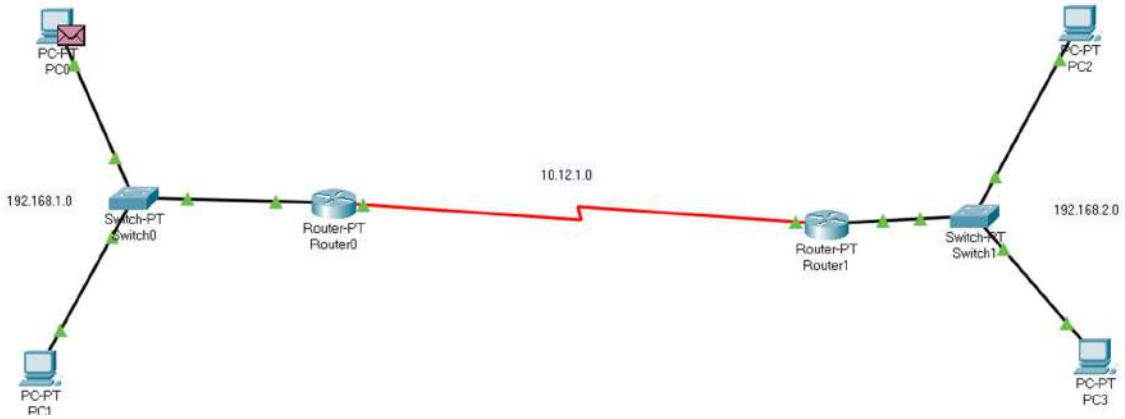


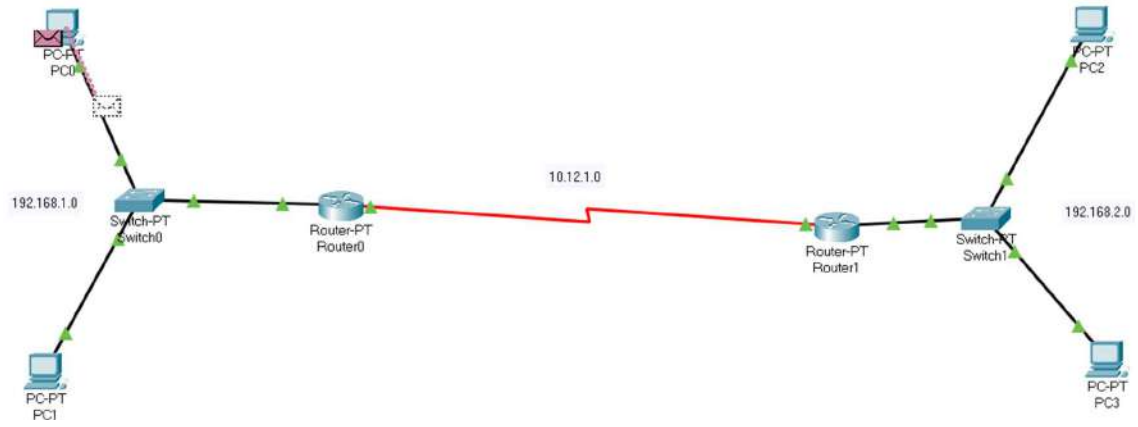
Check connection establishment :-

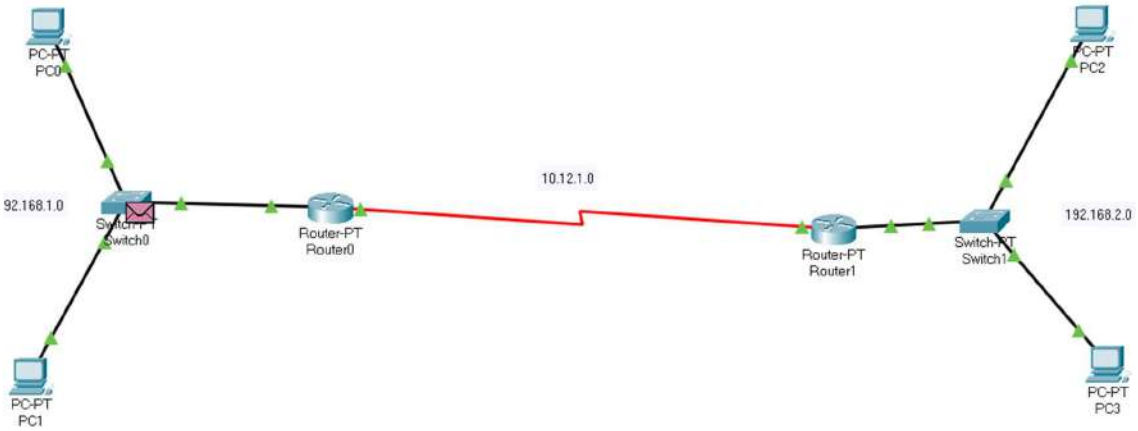
Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
	Successful	PC1	PC3	ICMP		0.000	N	0	(edit)	
	Successful	PC0	PC1	ICMP		0.000	N	1	(edit)	
	Successful	PC2	PC3	ICMP		0.000	N	2	(edit)	
	Successful	PC0	PC2	ICMP		0.000	N	3	(edit)	
	Successful	PC3	PC0	ICMP		0.000	N	4	(edit)	

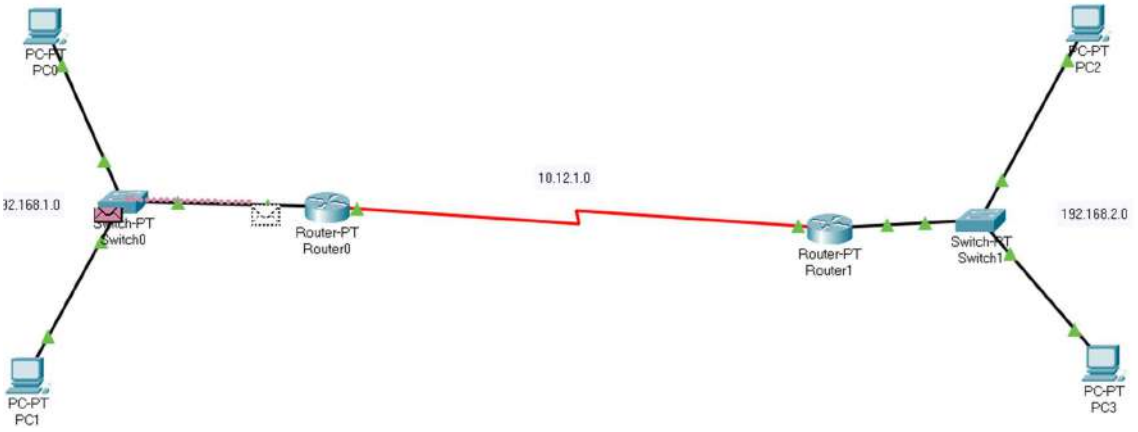
Realtime Simulation :-

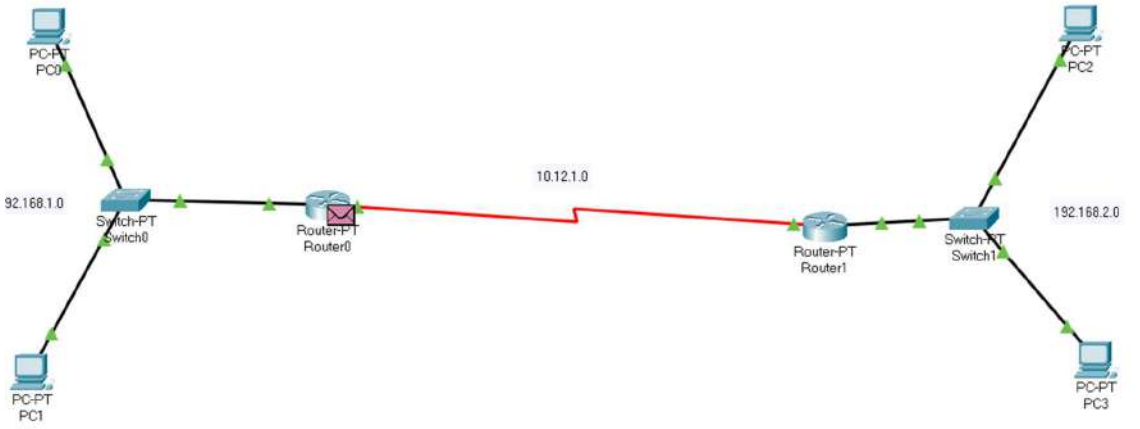
Send packet from PC0 to PC2 :-



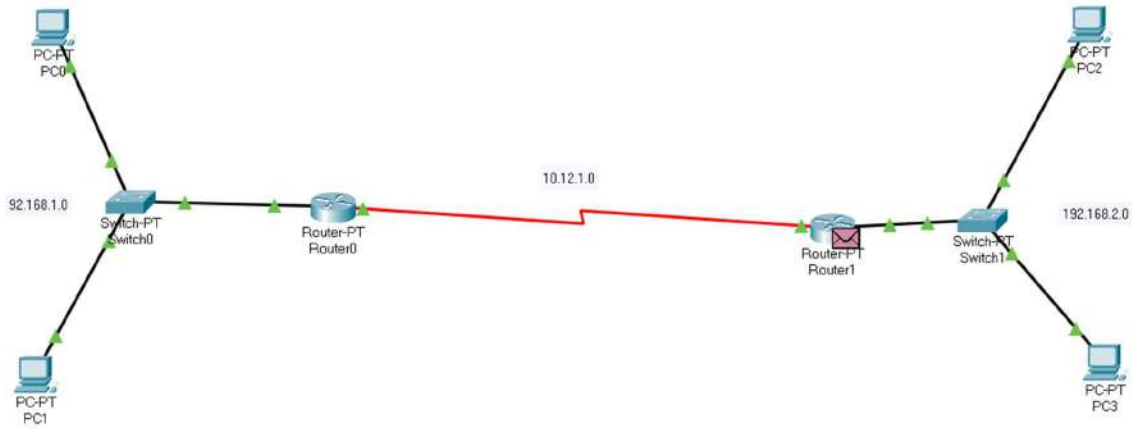

PC-PT
PC0
PC-PT
PC2
10.12.1.0
192.168.1.0
Switch-PT
Switch0
Router-PT
Router0
Router-PT
Router1
Switch-PT
Switch1
192.168.2.0
PC-PT
PC1
PC-PT
PC3

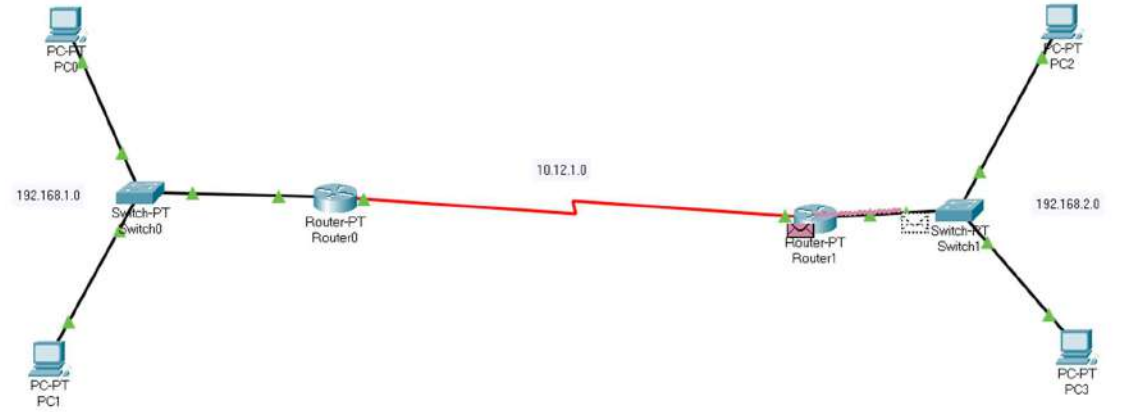

PC-PT
PC0
PC-PT
PC2
10.12.1.0
92.168.1.0
Switch-PT
Switch0
Router-PT
Router0
Router-PT
Router1
Switch-PT
Switch1
192.168.2.0
PC-PT
PC1
PC-PT
PC3

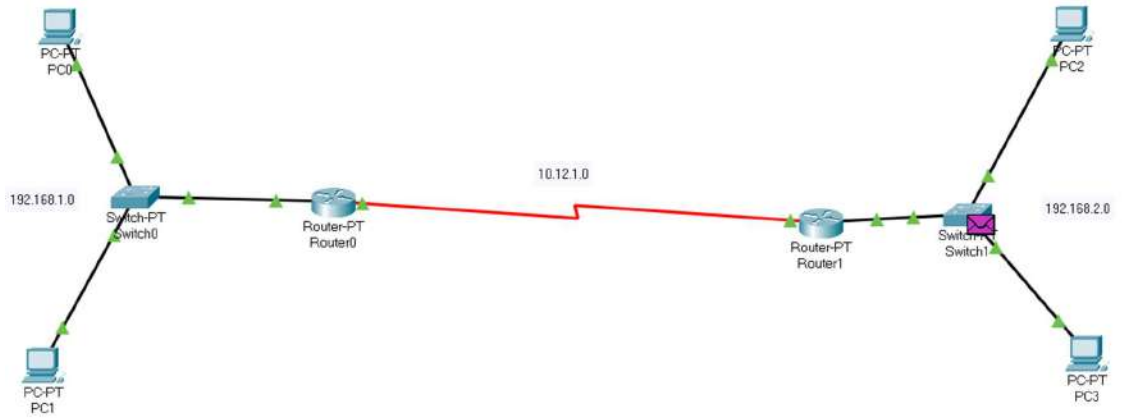

PC-PT
PC0
PC-PT
PC2
10.12.1.0
92.168.1.0
Switch-PT
Switch0
Router-PT
Router0
Router-PT
Router1
Switch-PT
Switch1
192.168.2.0
PC-PT
PC1
PC-PT
PC3

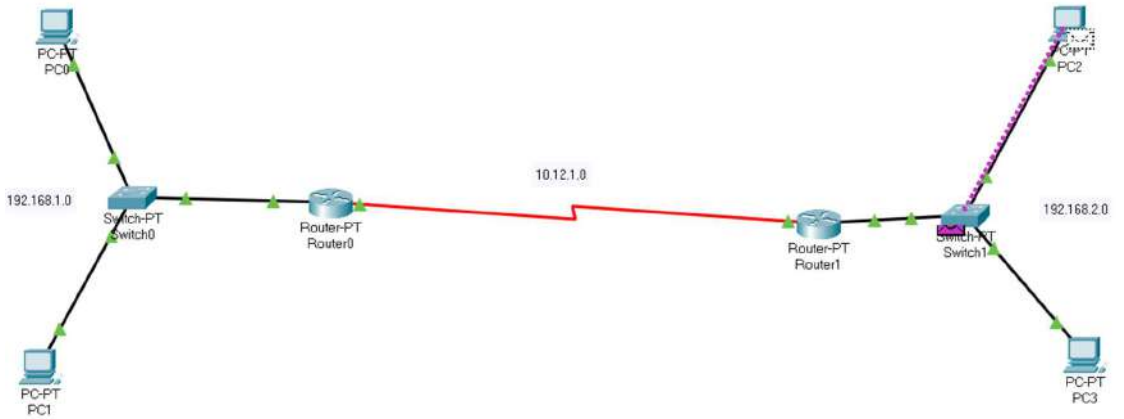

PC-PT
PC0
PC-PT
PC2
10.12.1.0
92.168.1.0
Switch-PT
Switch0
Router-PT
Router0
Router-PT
Router1
Switch-PT
Switch1
192.168.2.0
PC-PT
PC1
PC-PT
PC3

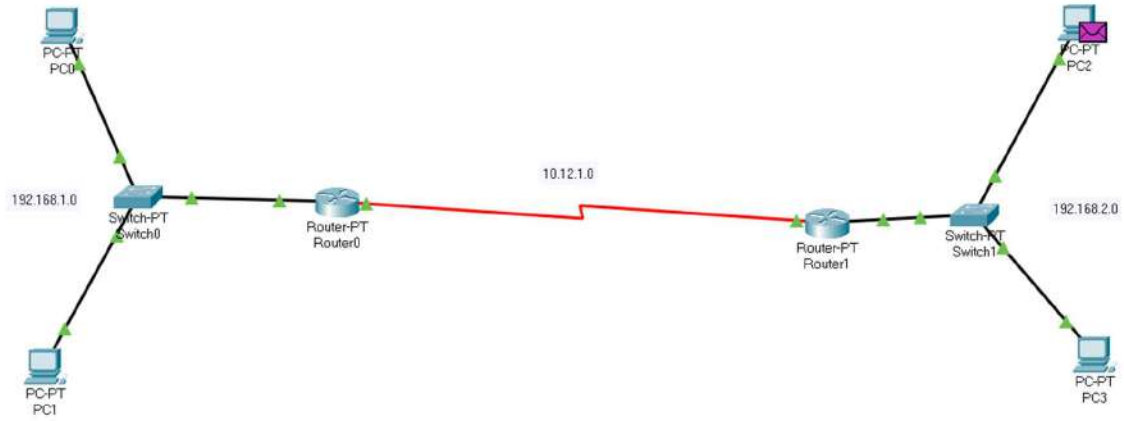

PC-PT
PC0
PC-PT
PC2
10.12.1.0
.168.1.0
Switch-PT
Switch0
Router-PT
Router0
Router-PT
Router1
Switch-PT
Switch1
192.168.2.0
PC-PT
PC1
PC-PT
PC3

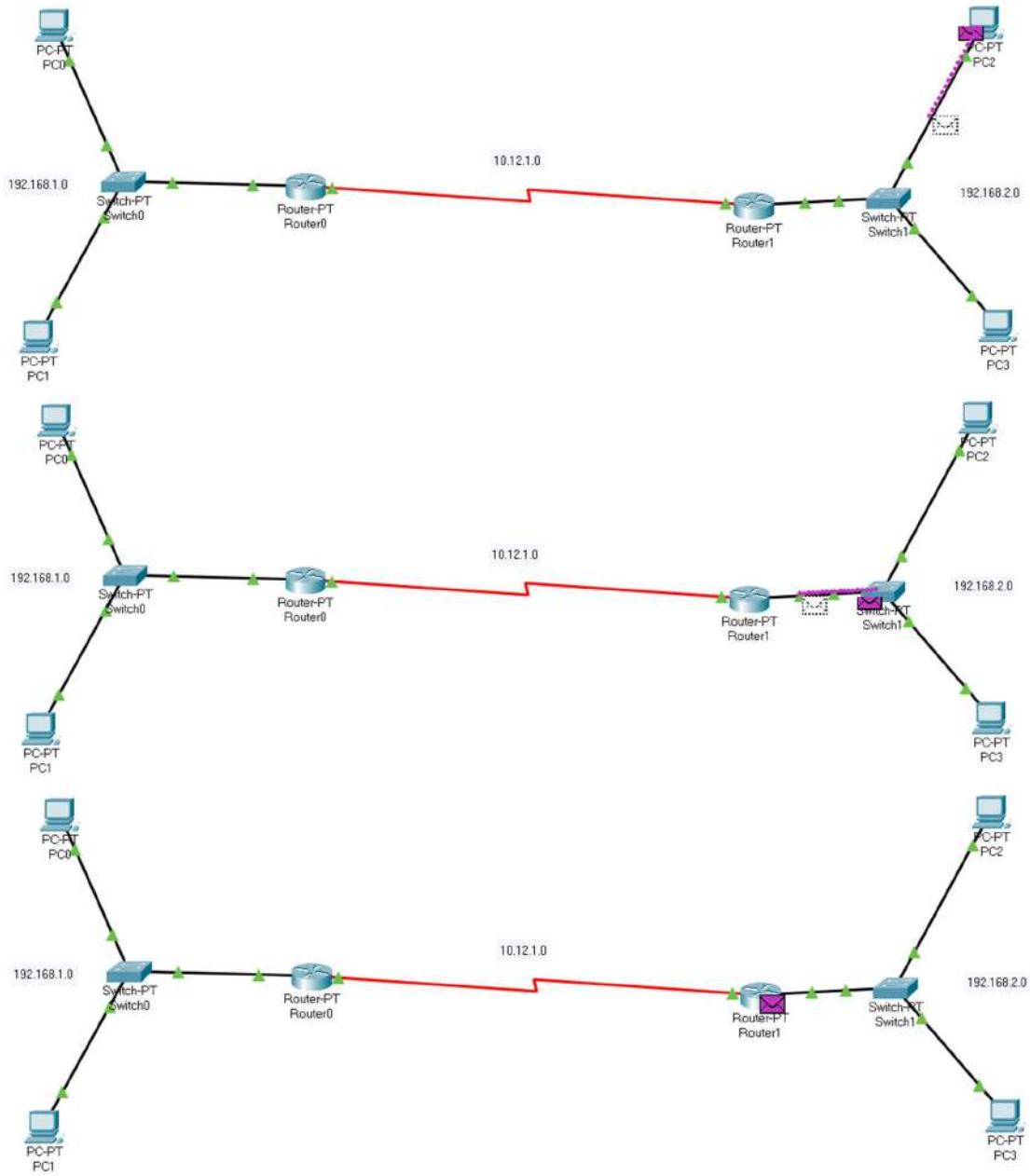

PC-PT
PC0
PC-PT
PC2
10.12.1.0
92.168.1.0
Switch-PT
Switch0
Router-PT
Router0
Router-PT
Router1
Switch-PT
Switch1
192.168.2.0
PC-PT
PC1
PC-PT
PC3

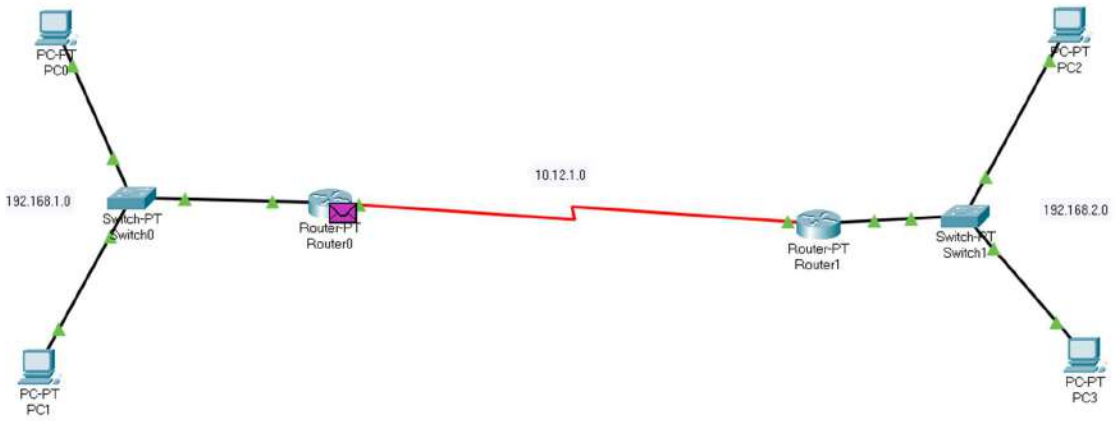

PC-PT
PC0
PC-PT
PC2
10.12.1.0
192.168.1.0
Switch-PT
Switch0
Router-PT
Router0
Router-PT
Router1
Switch-PT
Switch1
192.168.2.0
PC-PT
PC1
PC-PT
PC3

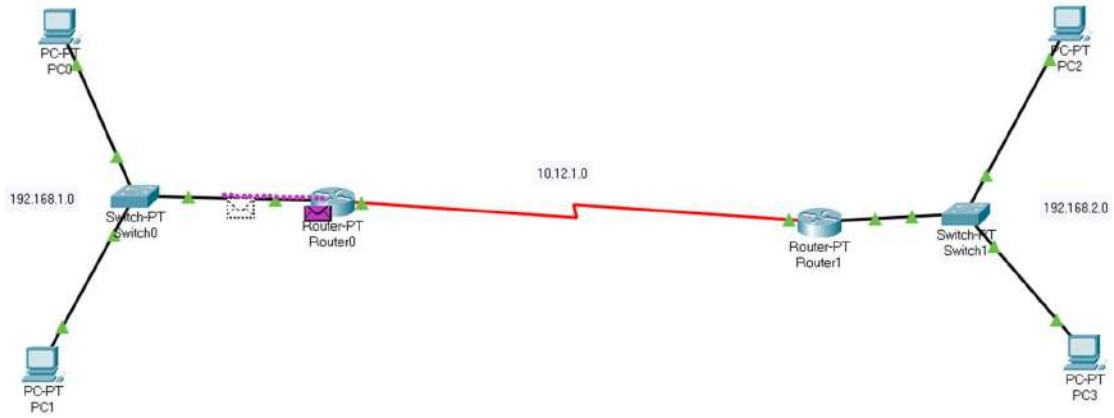

PC-PT
PC0
PC-PT
PC2
10.12.1.0
192.168.1.0
Switch-PT
Switch0
Router-PT
Router0
Router-PT
Router1
Switch-PT
Switch1
192.168.2.0
PC-PT
PC1
PC-PT
PC3

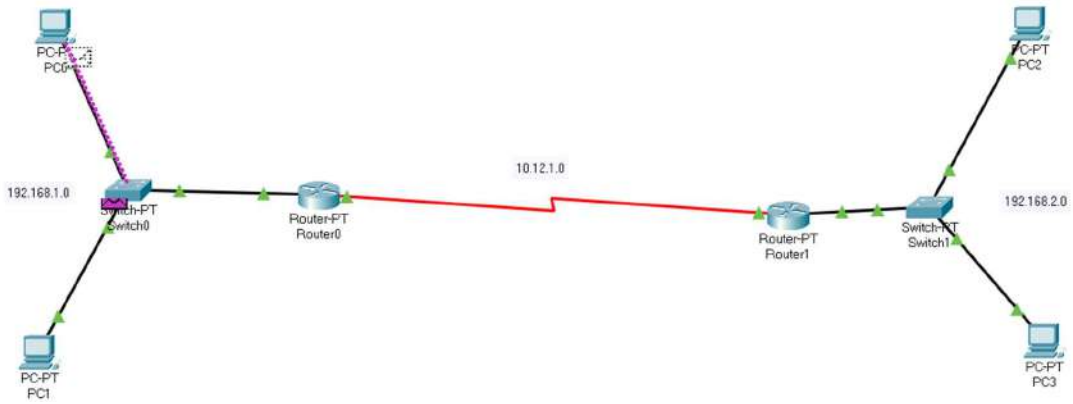

PC-PT
PC0
PC2
10.12.1.0
192.168.1.0
Switch-PT
Switch0
Router-PT
Router0
Router-PT
Router1
Switch-PT
Switch1
192.168.2.0
PC-PT
PC1
PC-PT
PC3

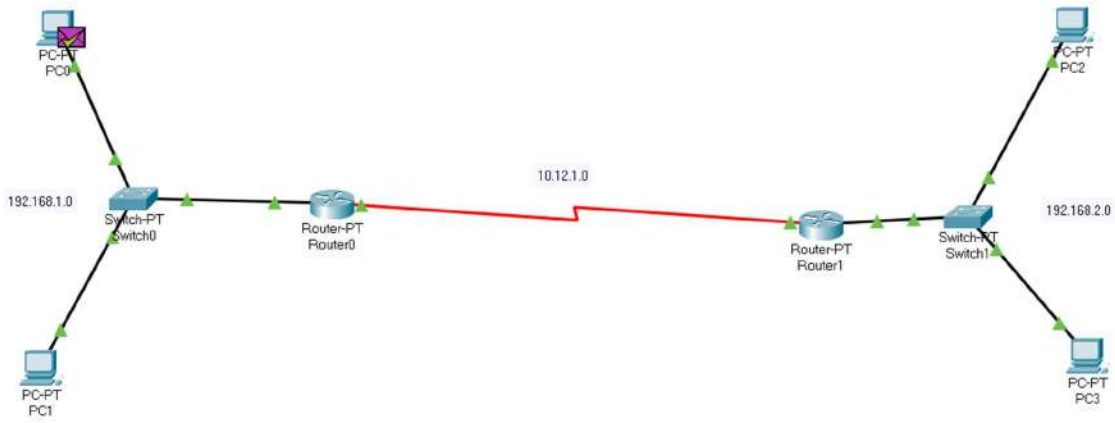

PC-PT
PC0
PC-PT
PC2
10.12.1.0
192.168.1.0
Switch-PT
Switch0
Router-PT
Router0
Router-PT
Router1
Switch-PT
Switch1
192.168.2.0
PC-PT
PC1
PC-PT
PC3


PC-PT
PC0
PC-PT
PC1
192.168.1.0
Switch-PT
Switch0
Router-PT
Router0
10.12.1.0
Router-PT
Router1
Switch-PT
Switch1
192.168.2.0
PC-PT
PC2
PC-PT
PC3

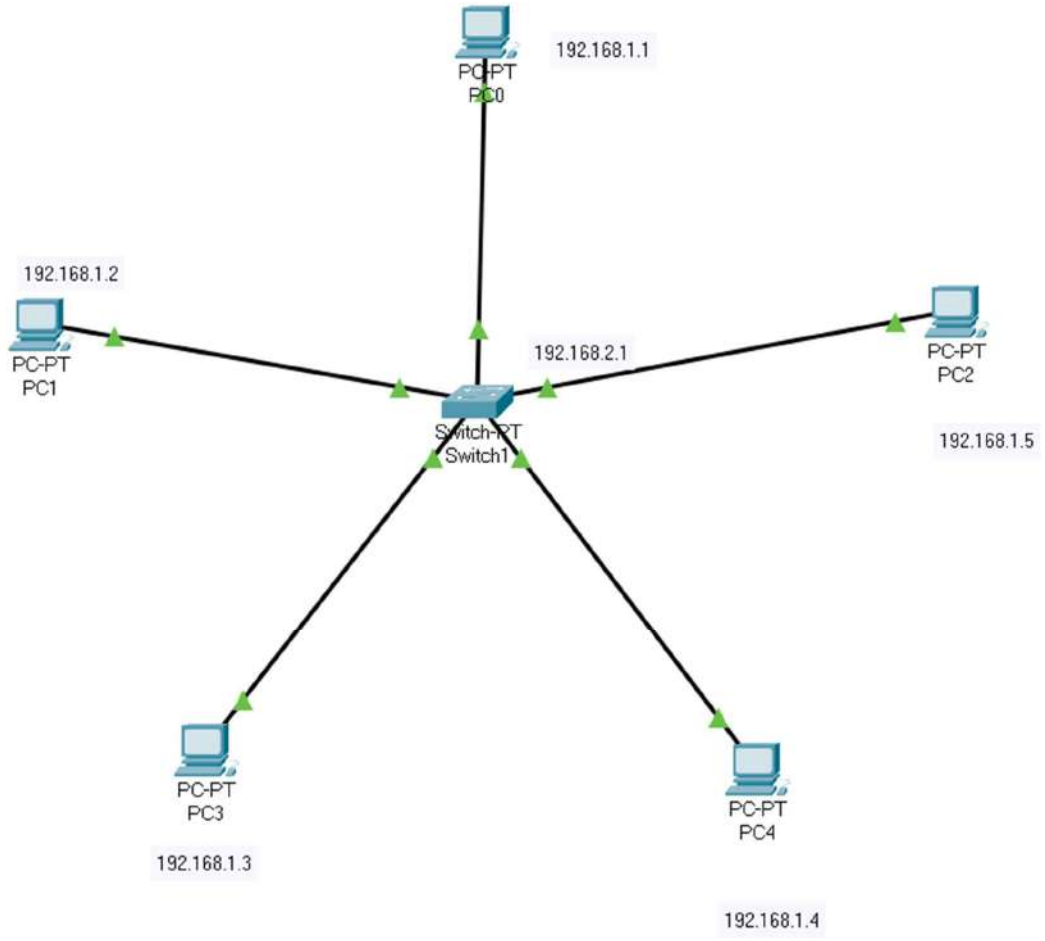

PC-PT
PC0
PC-PT
PC1
192.168.1.0
Switch-PT
Switch0
Router-PT
Router0
10.12.1.0
Router-PT
Router1
Switch-PT
Switch1
192.168.2.0
PC-PT
PC2
PC-PT
PC3


PC-PT
PC0
PC-PT
PC1
192.168.1.0
Switch-PT
Switch0
Router-PT
Router0
10.12.1.0
Router-PT
Router1
Switch-PT
Switch1
192.168.2.0
PC-PT
PC2
PC-PT
PC3


PC-PT
PC0
PC-PT
PC1
192.168.1.0
Switch-PT
Switch0
Router-PT
Router0
10.12.1.0
Router-PT
Router1
Switch-PT
Switch1
192.168.2.0
PC-PT
PC2
PC-PT
PC3



Star Topology using switch:-

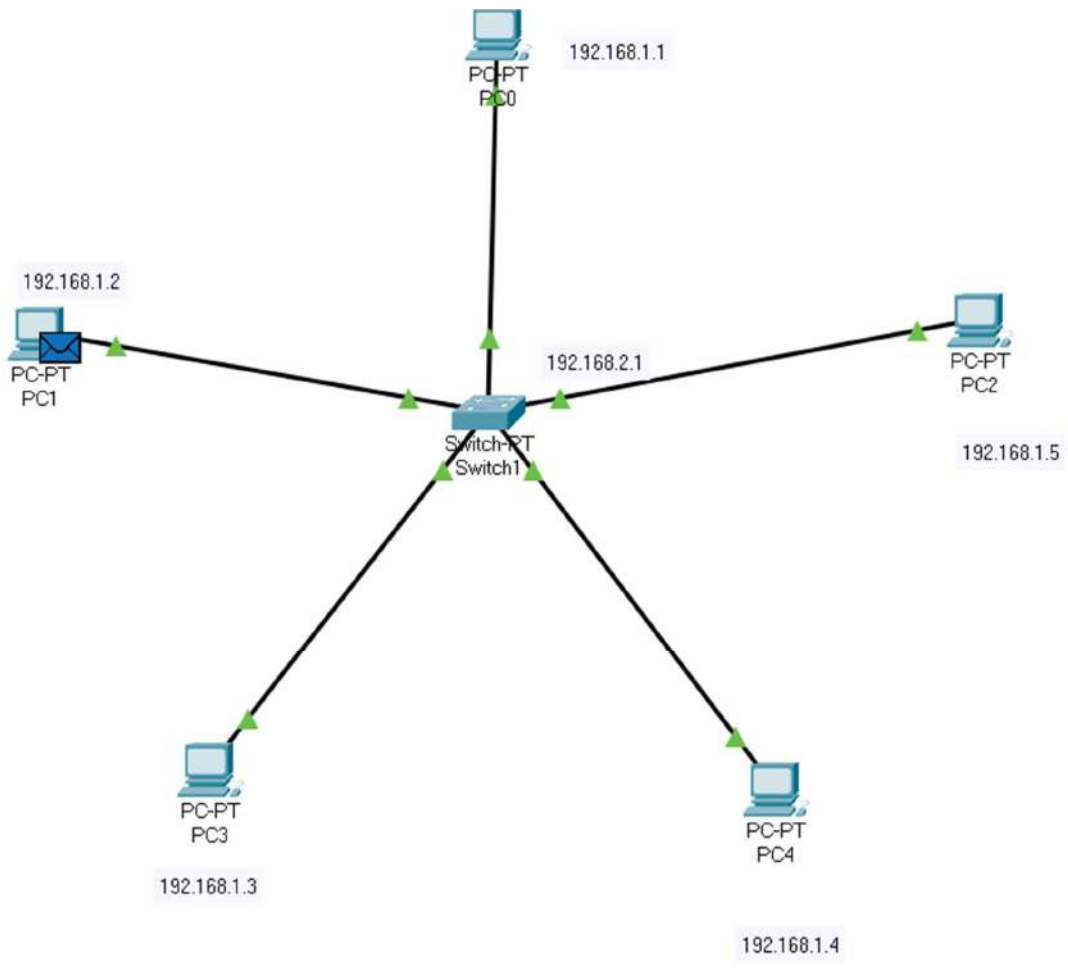


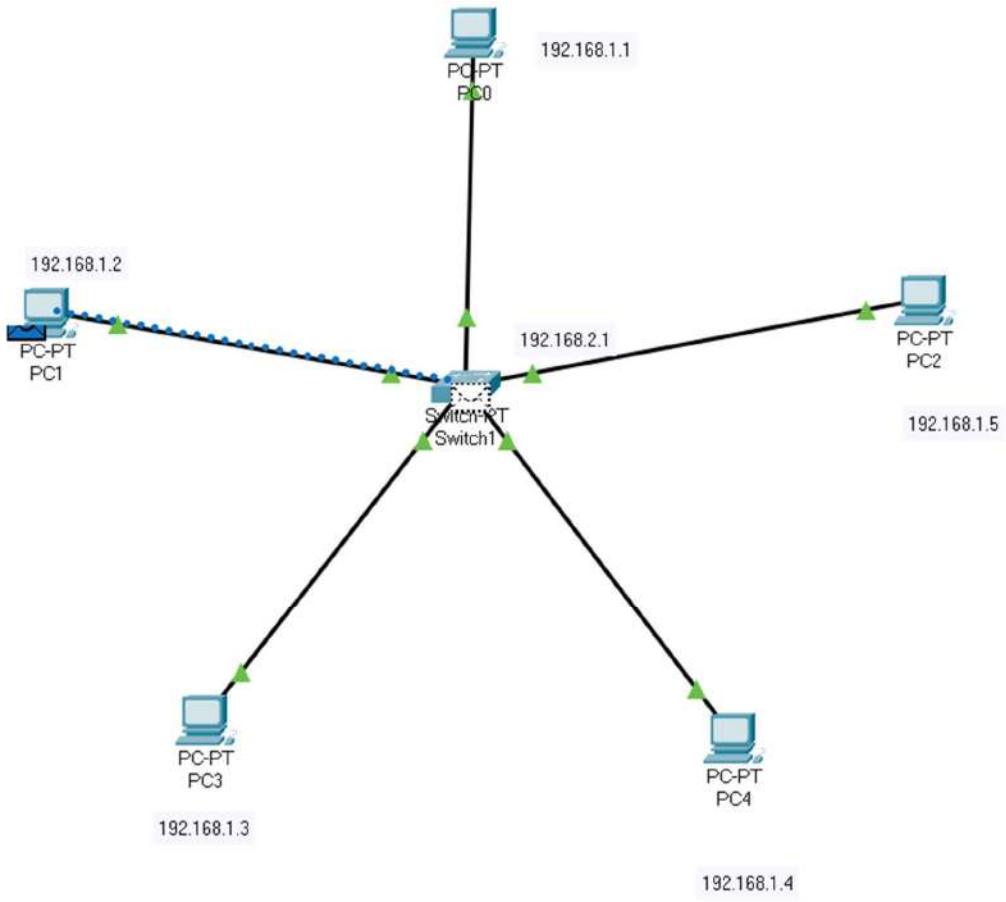
Check connection establishment :-

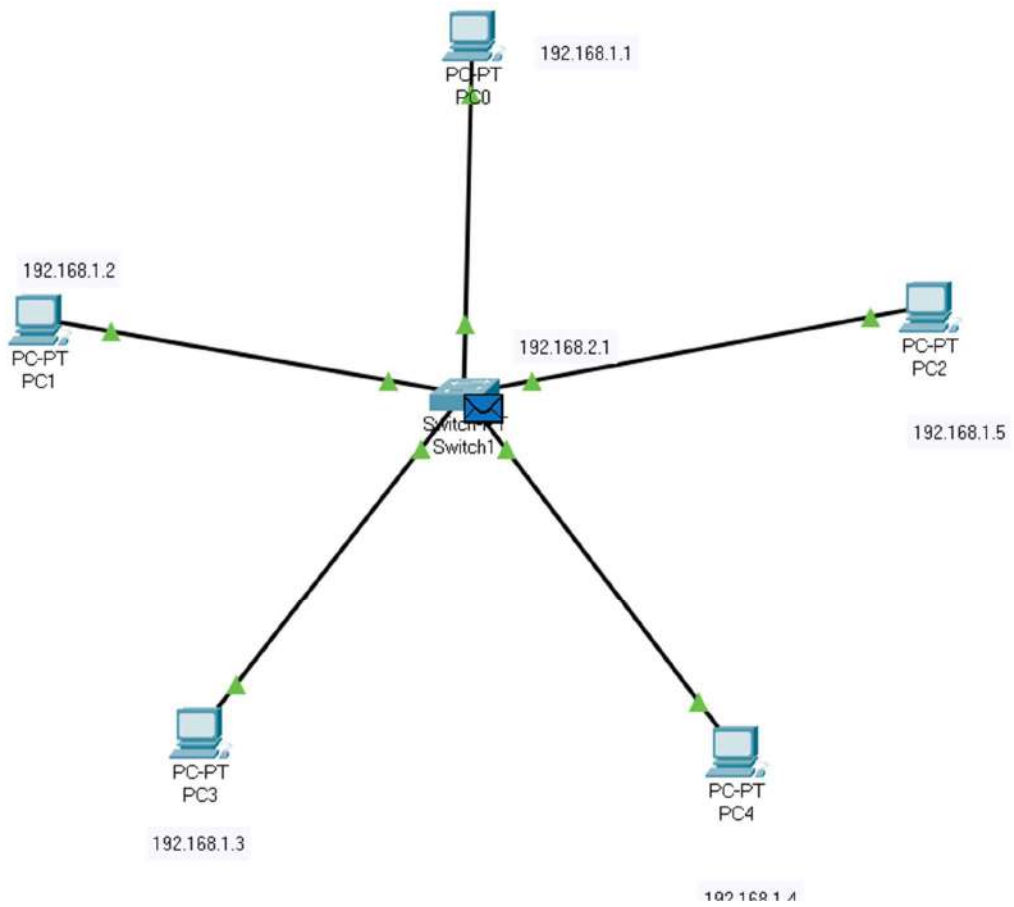
Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
	Successful	PC1	PC0	ICMP		0.000	N	0	(edit)	
	Successful	PC0	PC2	ICMP		0.000	N	1	(edit)	
	Successful	PC4	PC3	ICMP		0.000	N	2	(edit)	
	Successful	PC1	PC4	ICMP		0.000	N	3	(edit)	
	Successful	PC0	PC3	ICMP		0.000	N	4	(edit)	

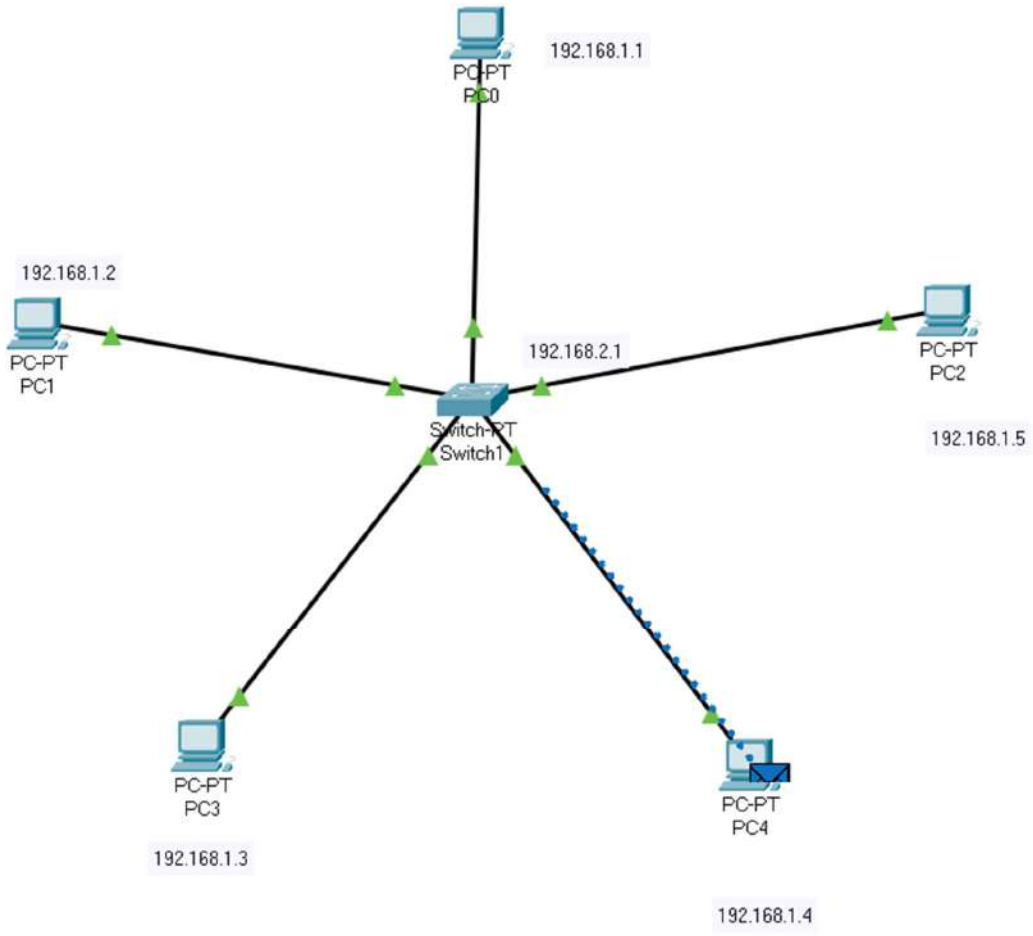
Real time simulation :-

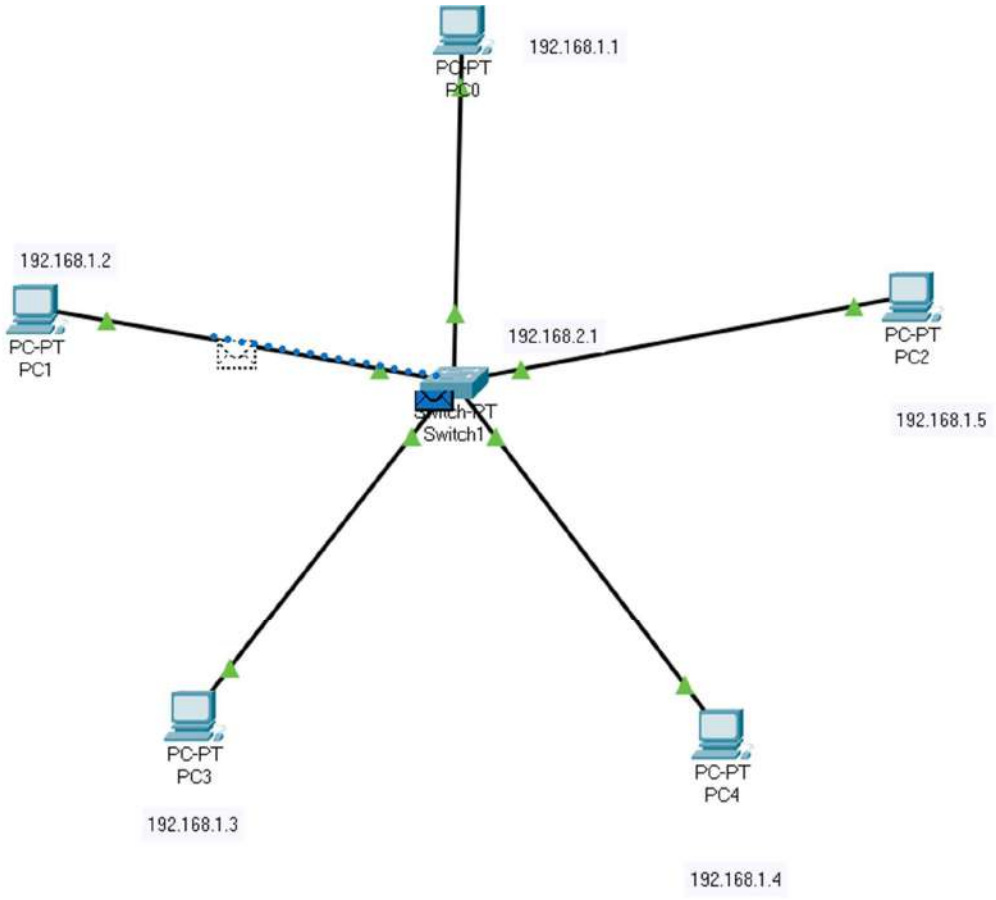
Send packet from PC1 to PC4 :-

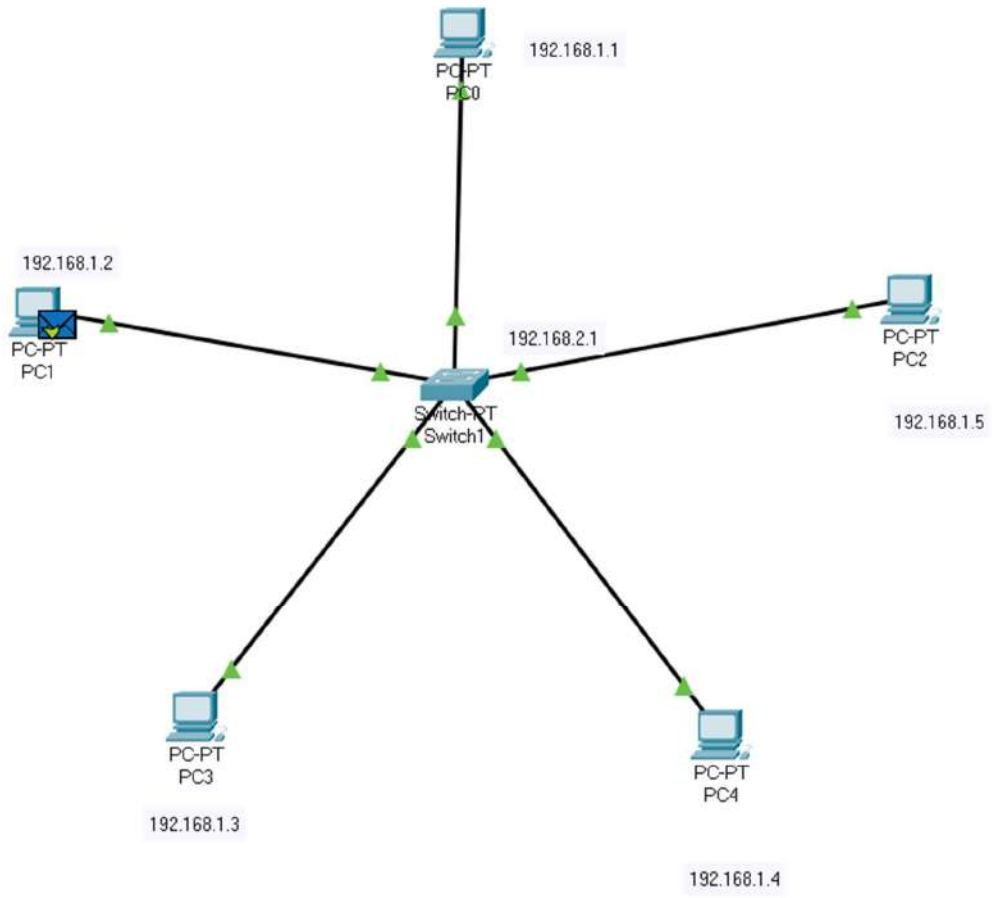



192.168.1.1
PC-PT
PC0
192.168.1.2
PC-PT
PC1
192.168.2.1
PC-PT
PC2
192.168.1.5
Switch-PT
Switch1
PC-PT
PC3
192.168.1.3
PC-PT
PC4
192.168.1.4

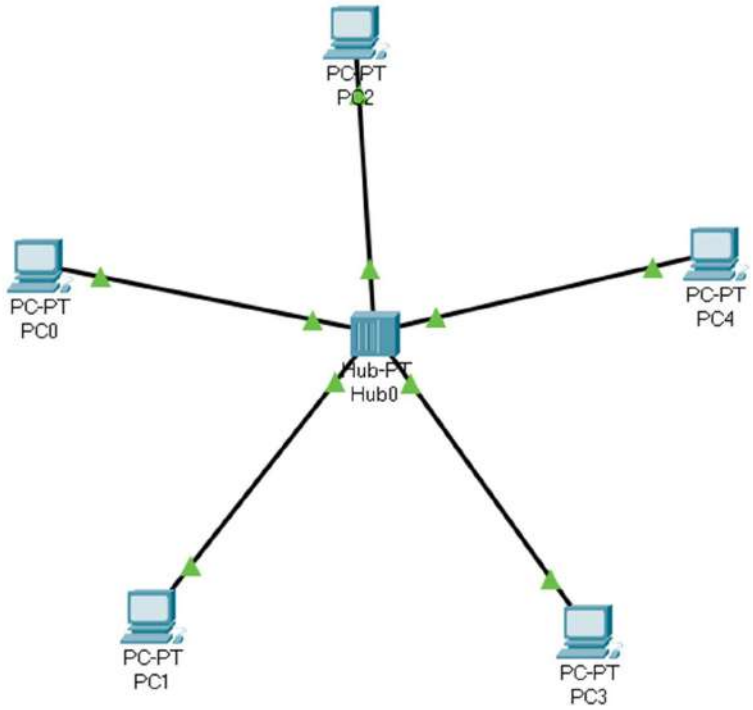

192.168.1.1
PC-PT
PC0
192.168.1.2
PC-PT
PC1
192.168.2.1
PC-PT
PC2
192.168.1.5
Switch1
PC-PT
PC3
192.168.1.3
PC-PT
PC4


192.168.1.1
PC-PT
PC0
192.168.1.2
PC-PT
PC1
192.168.2.1
PC-PT
PC2
Switch-PT
Switch1
192.168.1.5
PC-PT
PC3
192.168.1.3
PC-PT
PC4
192.168.1.4


192.168.1.1
PC-PT
PC0
192.168.1.2
PC-PT
PC1
192.168.2.1
PC-PT
PC2
Switch-PT
Switch1
192.168.1.5
PC-PT
PC3
192.168.1.3
PC-PT
PC4
192.168.1.4

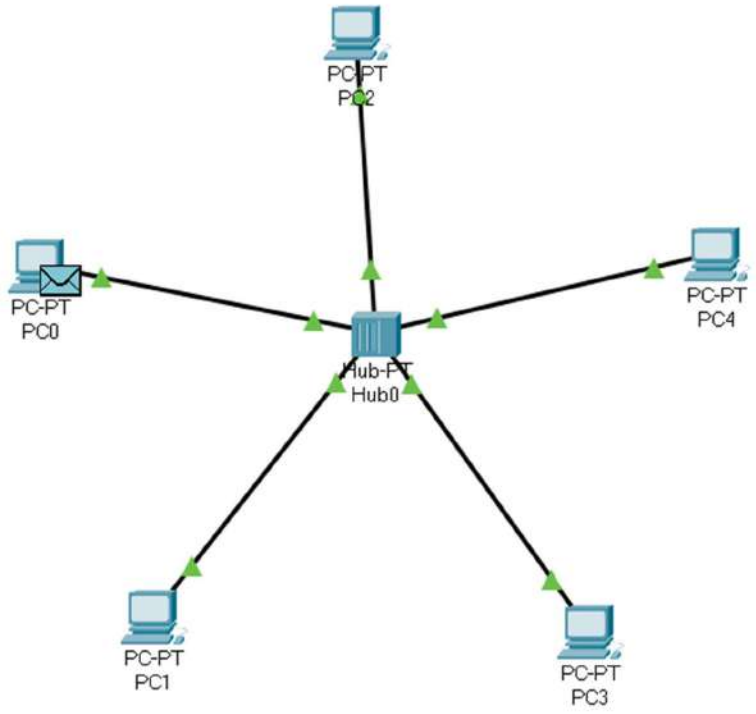


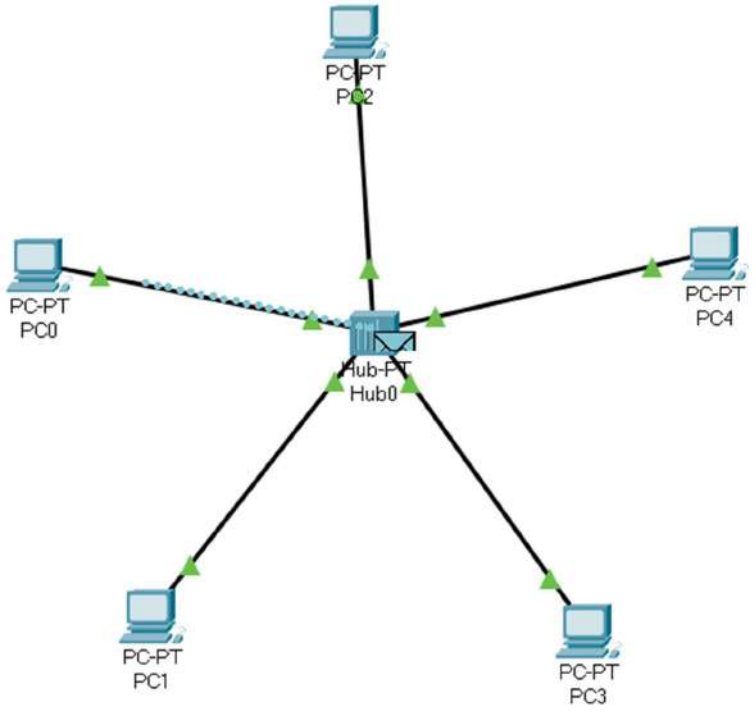
Star topology using HUB :-

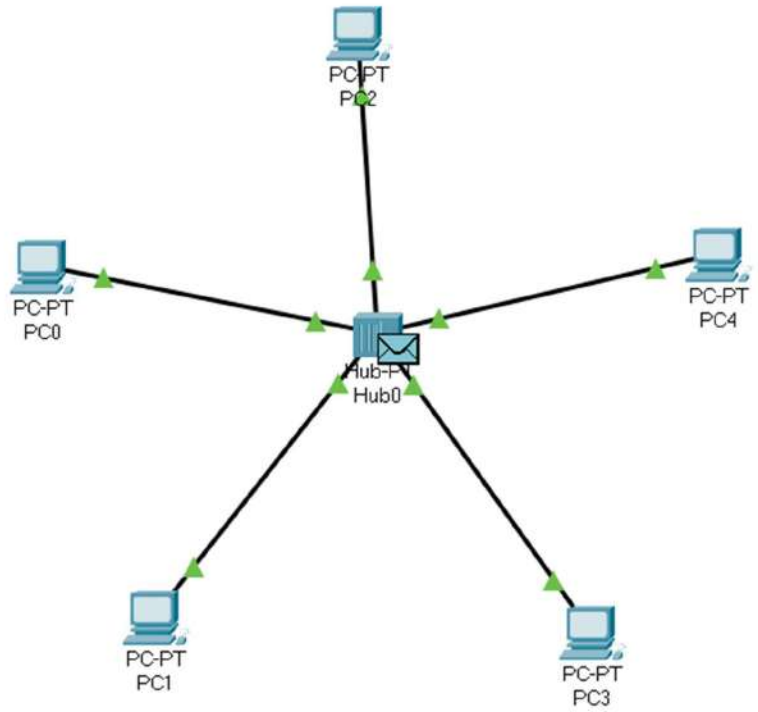


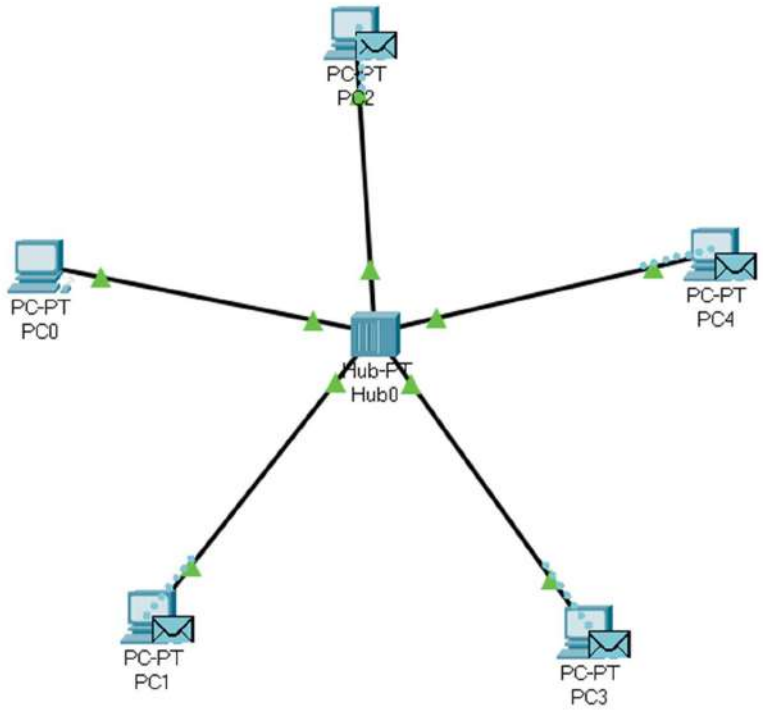
Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
	Successful	PC1	PC4	ICMP		0.000	N	0	(edit)	
	Successful	PC0	PC3	ICMP		157.254	N	1	(edit)	
	Successful	PC1	PC3	ICMP		391.643	N	2	(edit)	

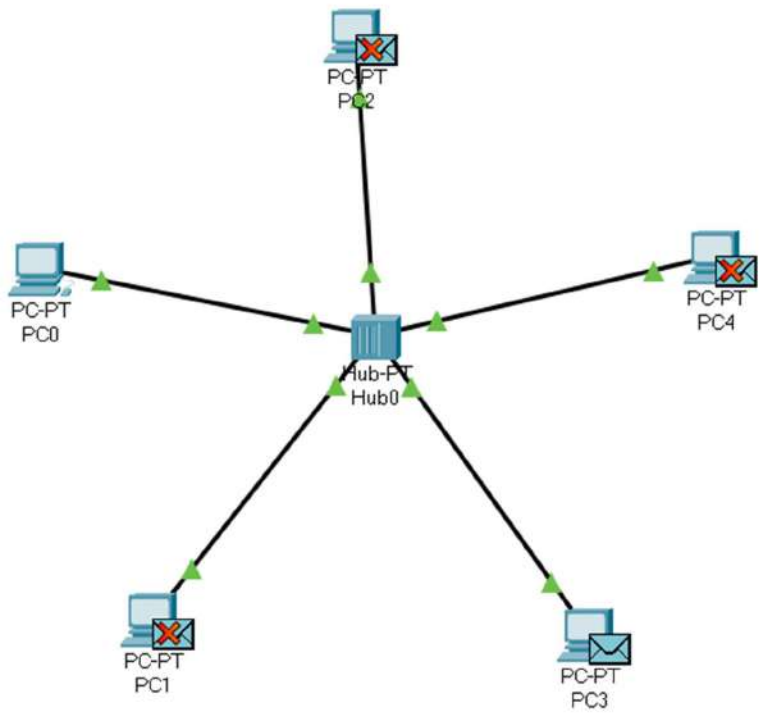
SEND PACKET FROM PC0 TO PC 1 :-

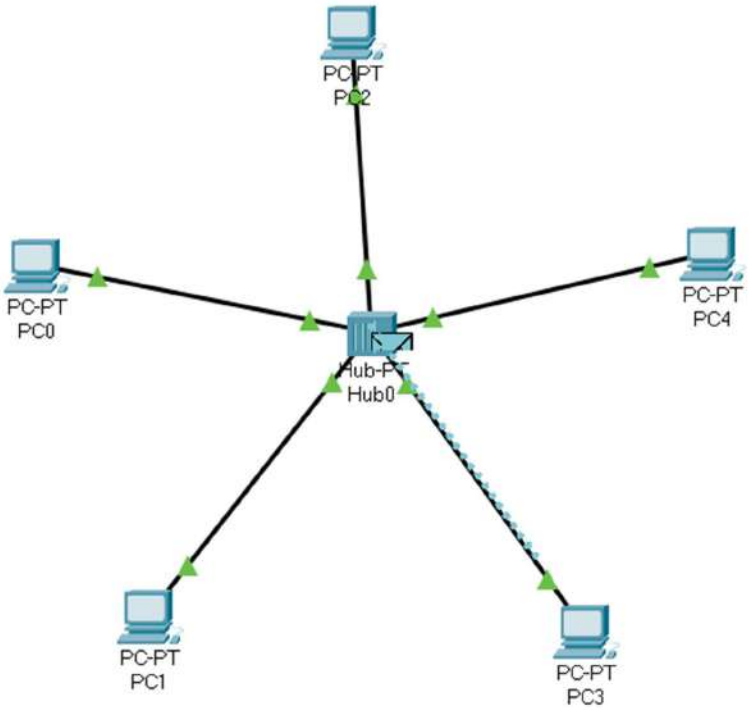

PC-PT
PC2
PC-PT
PC0
PC-PT
PC4
Hub-PT
Hub0
PC-PT
PC1
PC-PT
PC3

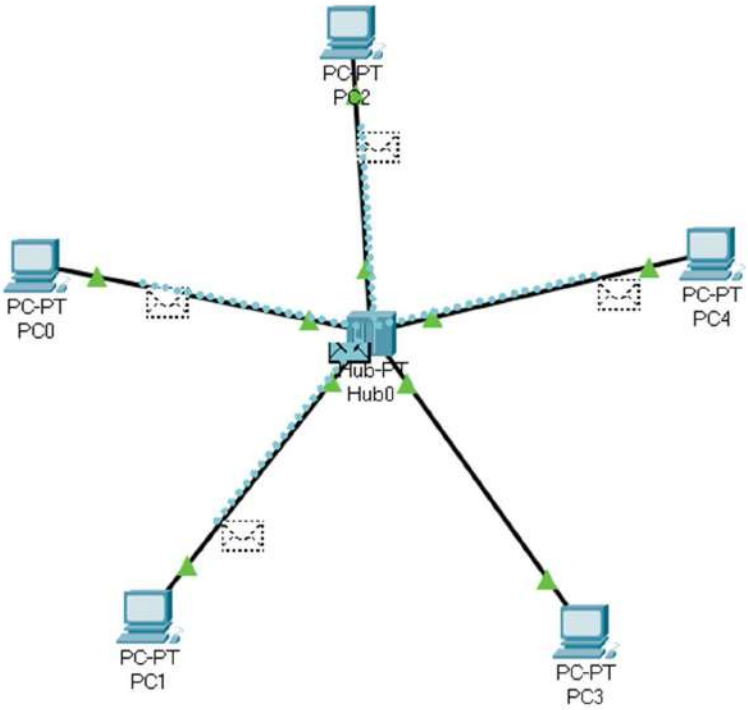

PC-PT
PC2
PC-PT
PC0
PC-PT
PC4
Hub-PT
Hub0
PC-PT
PC1
PC-PT
PC3

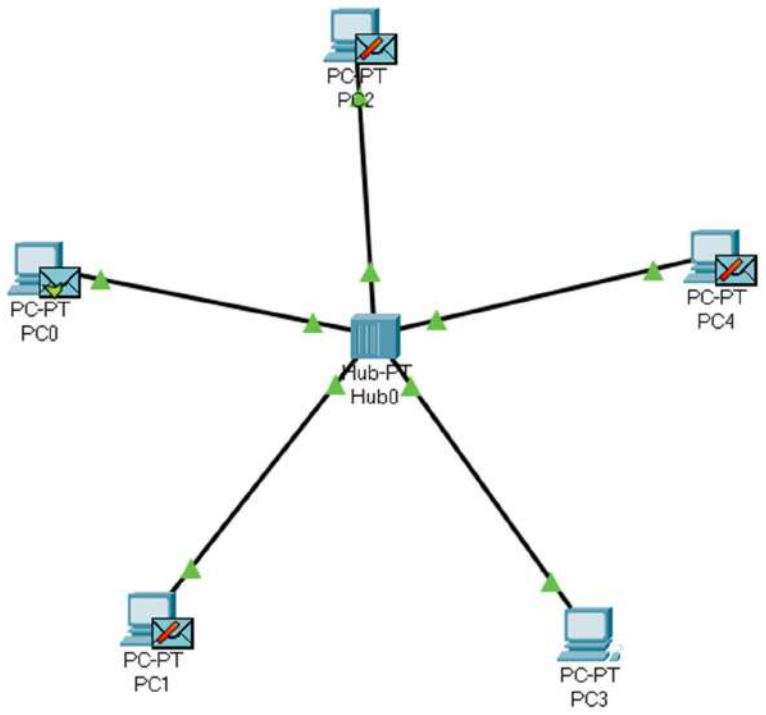

PC-PT
PC2
PC-PT
PC0
PC-PT
PC4
Hub-PT
Hub0
PC-PT
PC1
PC-PT
PC3

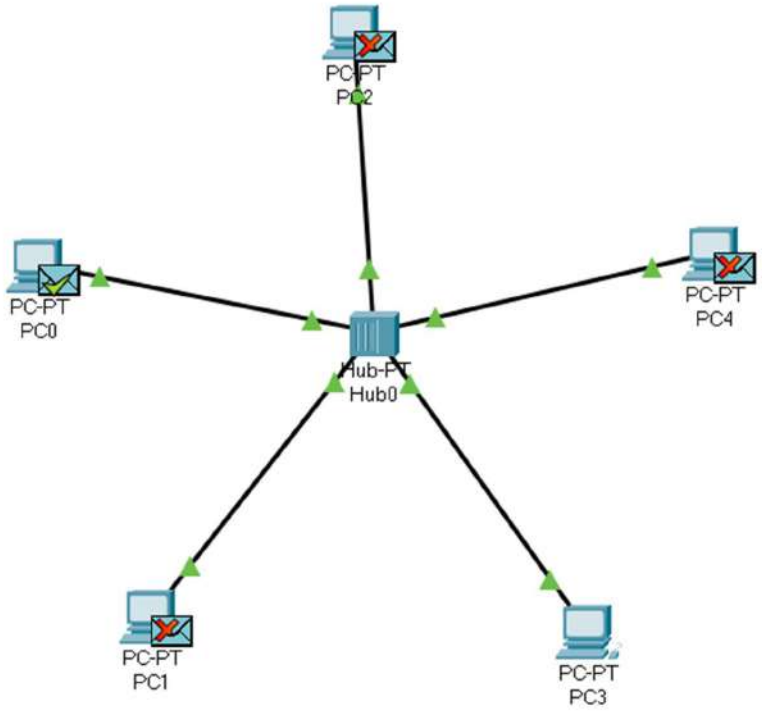

PC-PT
PC2
PC-PT
PC0
PC-PT
PC4
Hub-PT
Hub0
PC-PT
PC1
PC-PT
PC3


PC-PT
PC2
PC-PT
PC0
PC-PT
PC4
Hub-PT
Hub0
PC-PT
PC1
PC-PT
PC3

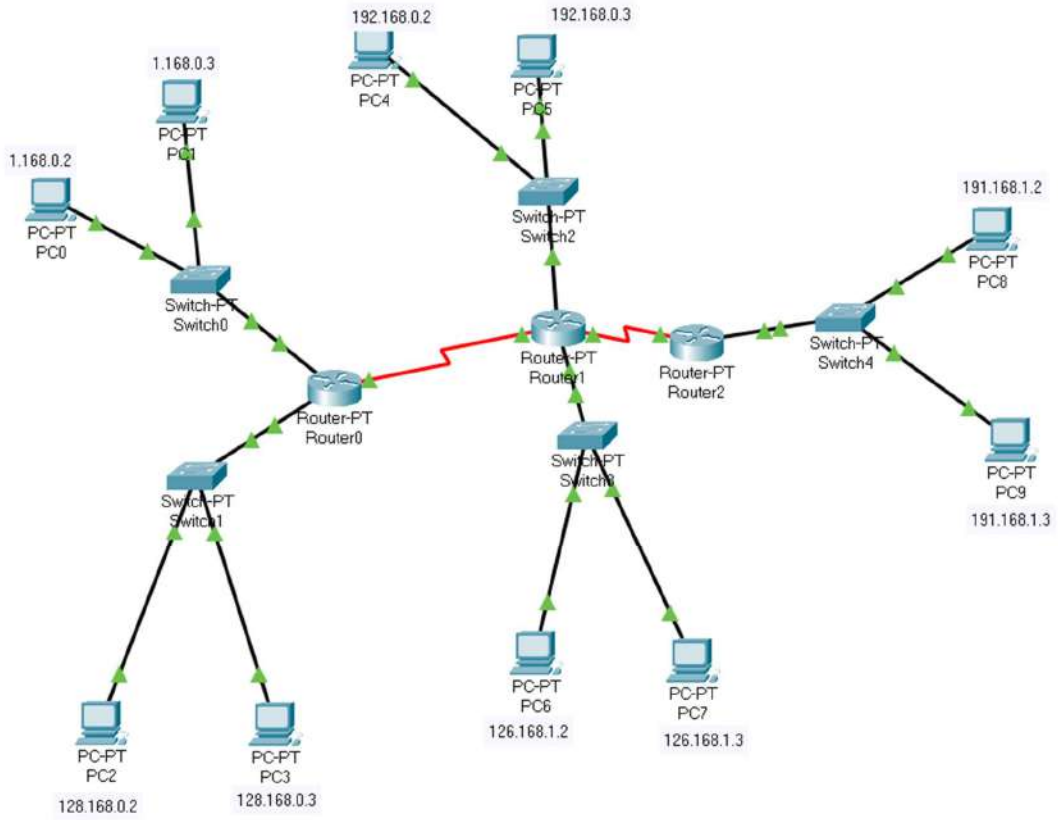

PC-PT
PC2
PC-PT
PC0
PC-PT
PC4
Hub-PT
Hub0
PC-PT
PC1
PC-PT
PC3


PC-PT
PC2
PC-PT
PC0
PC-PT
PC4
Hub-PT
Hub0
PC-PT
PC1
PC-PT
PC3


PC-PT
PC2
PC-PT
PC0
PC-PT
PC4
Hub-PT
Hub0
PC-PT
PC1
PC-PT
PC3


PC-PT
PC2
PC-PT
PC0
PC-PT
PC4
Hub-PT
Hub0
PC-PT
PC1
PC-PT
PC3

LAN using CISCO PACKET TRACER :-

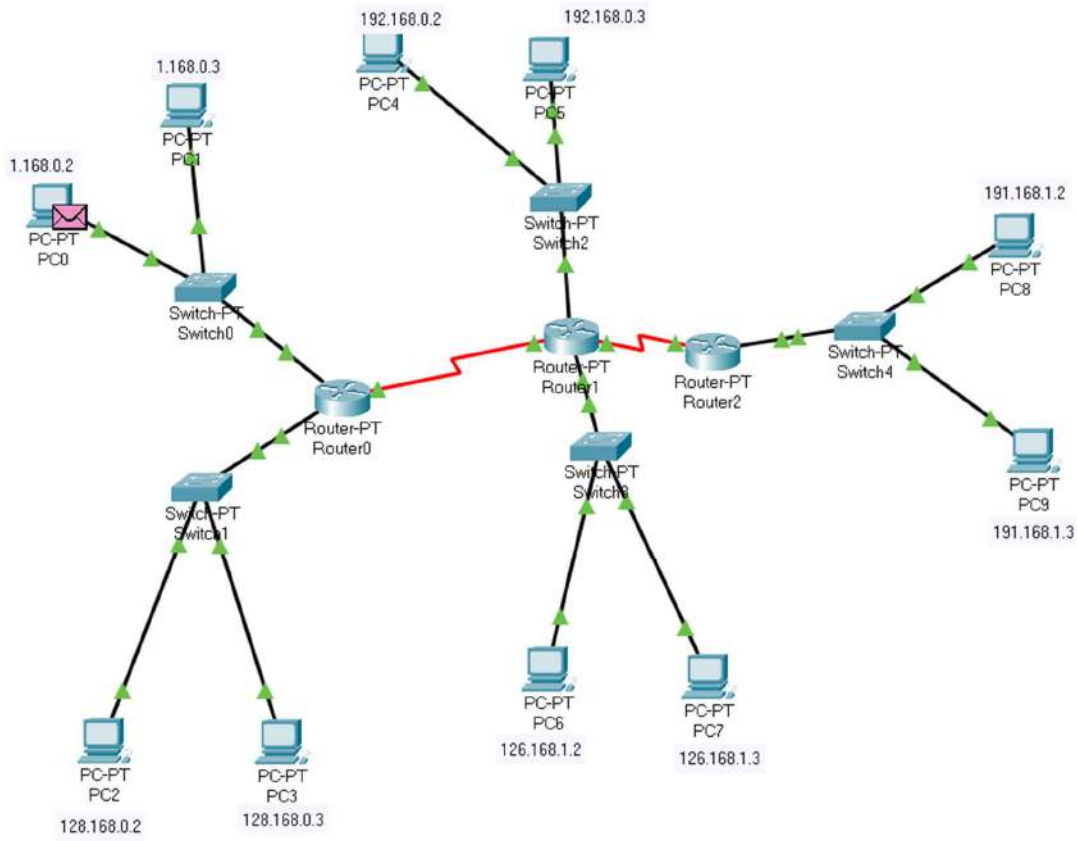


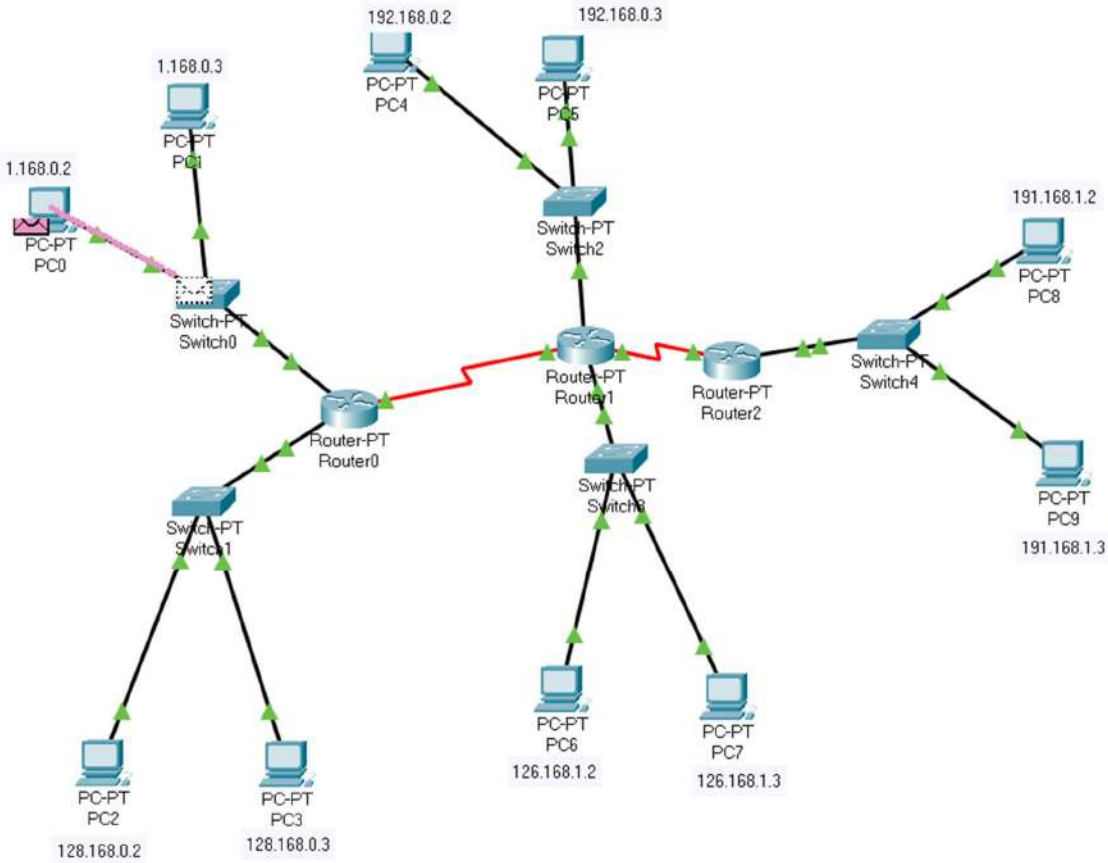
CHECK CONNECTION ESTABLISHMENT :-

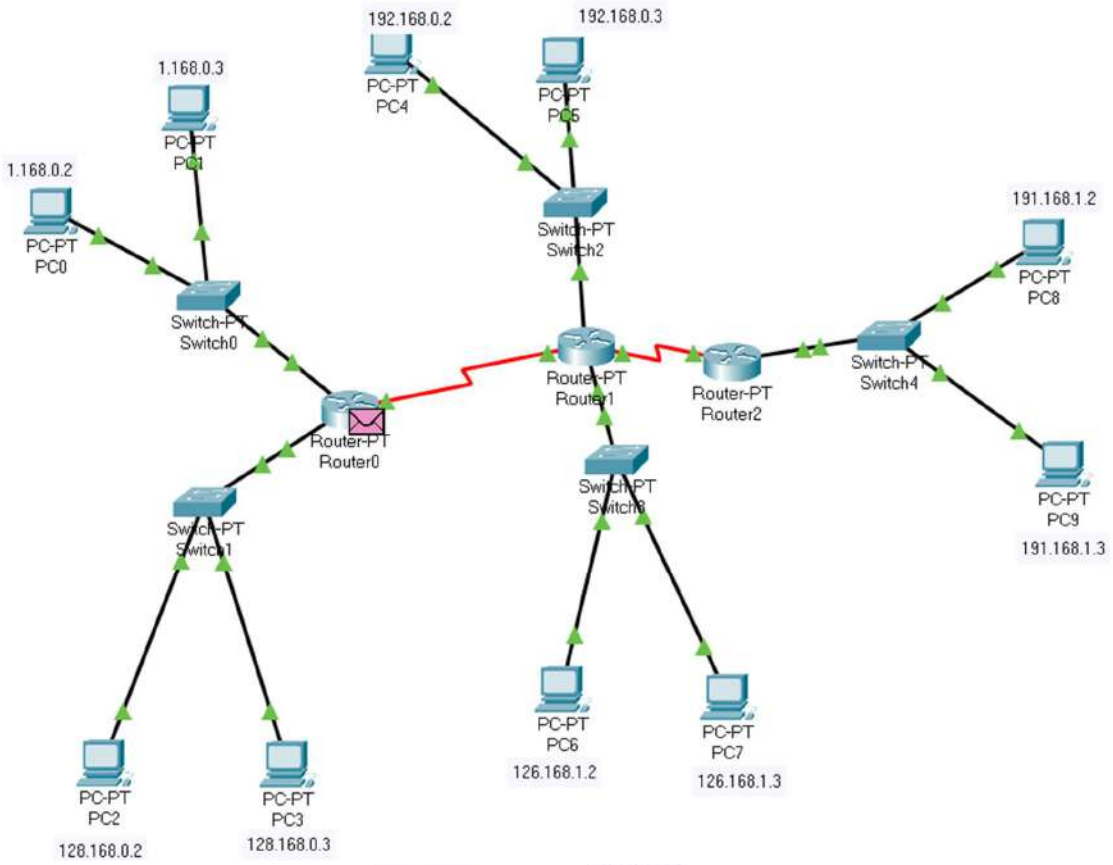
Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
	Successful	PC4	PC1	ICMP		0.000	N	0	(edit)	
	Successful	PC1	PC2	ICMP		0.000	N	1	(edit)	
	Successful	PC0	PC6	ICMP		0.000	N	2	(edit)	
	Successful	PC0	PC8	ICMP		0.000	N	3	(edit)	
	Successful	PC7	PC9	ICMP		0.000	N	4	(edit)	
	Successful	PC8	PC5	ICMP		0.000	N	5	(edit)	
	Successful	PC9	PC3	ICMP		0.000	N	6	(edit)	
	Successful	PC2	PC4	ICMP		0.000	N	7	(edit)	
	Successful	PC5	PC6	ICMP		0.000	N	8	(edit)	
	Successful	PC8	PC6	ICMP		0.000	N	9	(edit)	

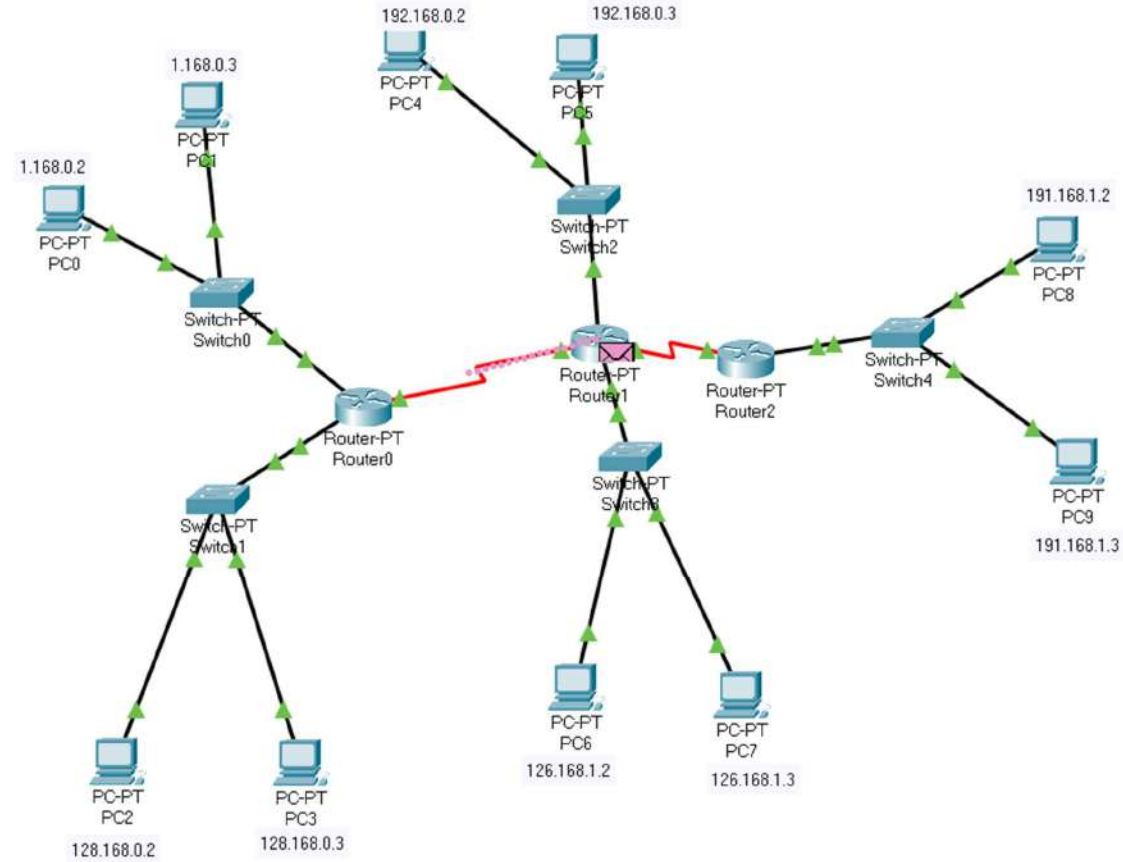
Real time simulation :-

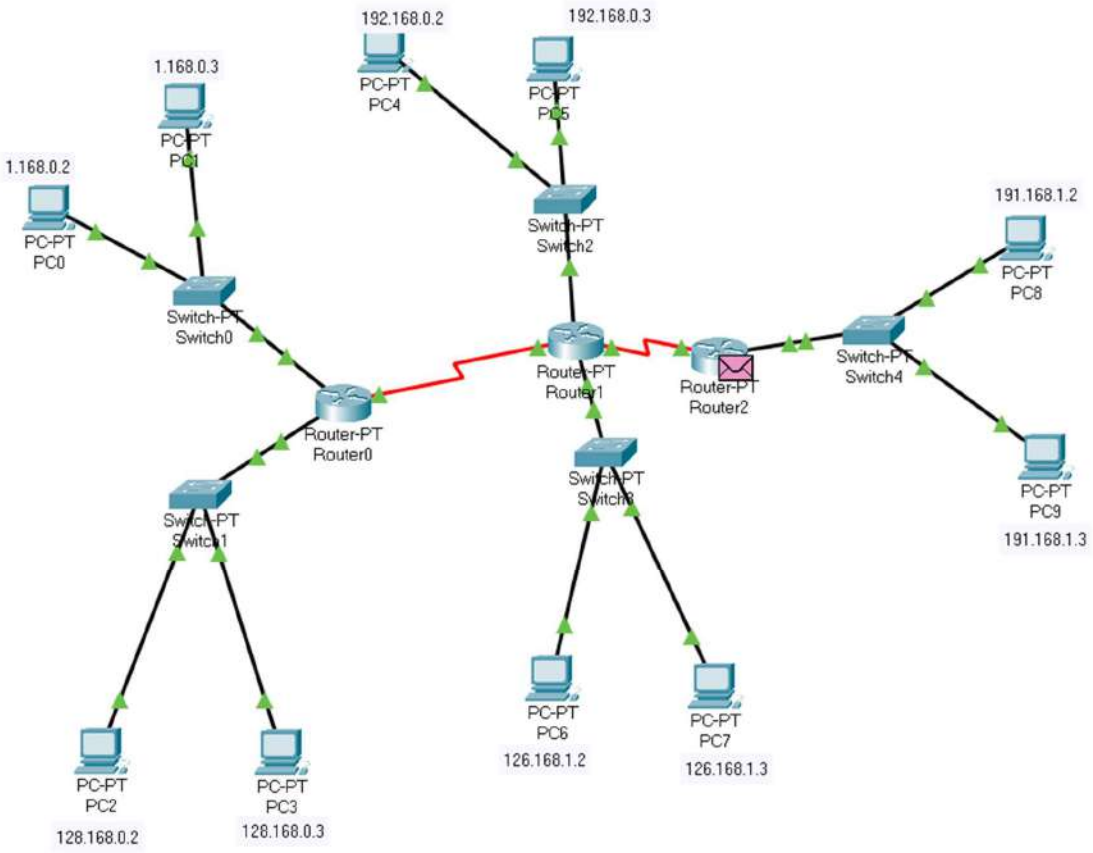
Send packet from PC0 to PC9

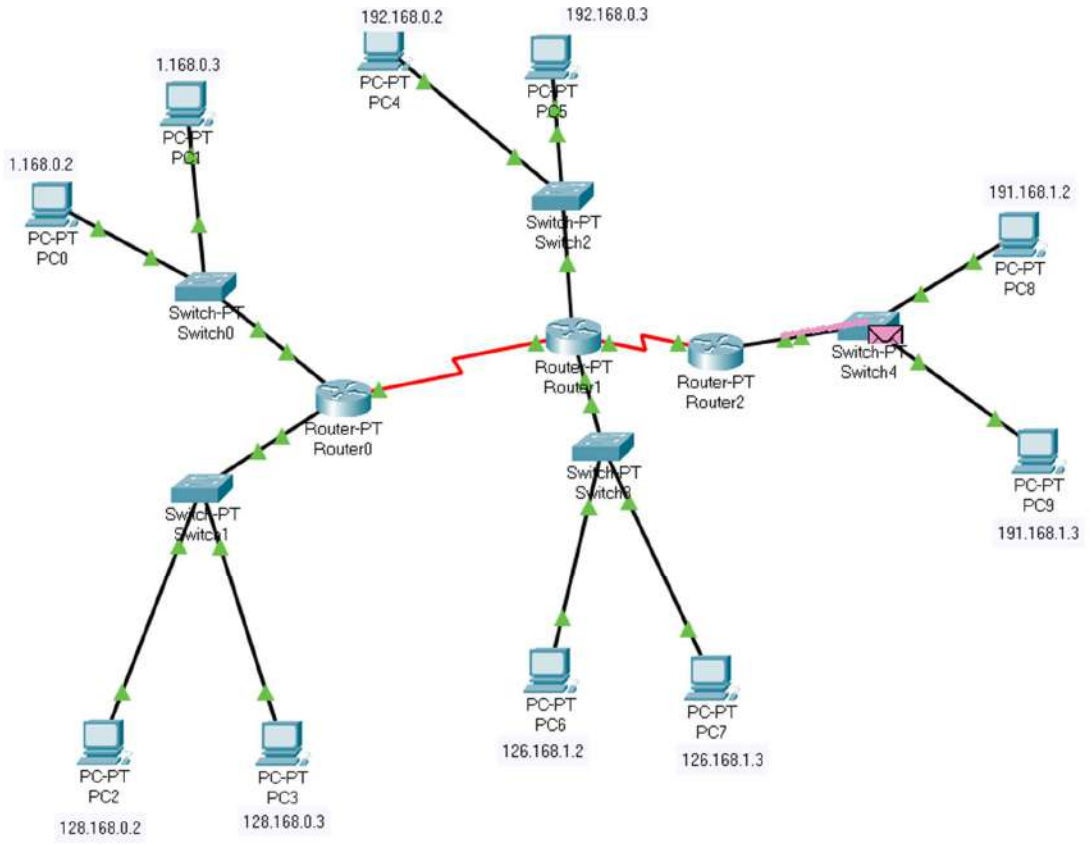

192.168.0.2
192.168.0.3
PC-PT
PC4
PC-PT
PC5
1.168.0.3
PC-PT
PC1
1.168.0.2
PC-PT
PC0
Switch-PT
Switch2
191.168.1.2
PC-PT
PC8
Switch-PT
Switch0
Router-PT
Router1
Router-PT
Router2
Switch-PT
Switch4
Router-PT
Router0
Switch-PT
Switch3
PC-PT
PC9
191.168.1.3
Switch-PT
Switch1
PC-PT
PC6
126.168.1.2
PC-PT
PC7
126.168.1.3
PC-PT
PC2
128.168.0.2
PC-PT
PC3
128.168.0.3

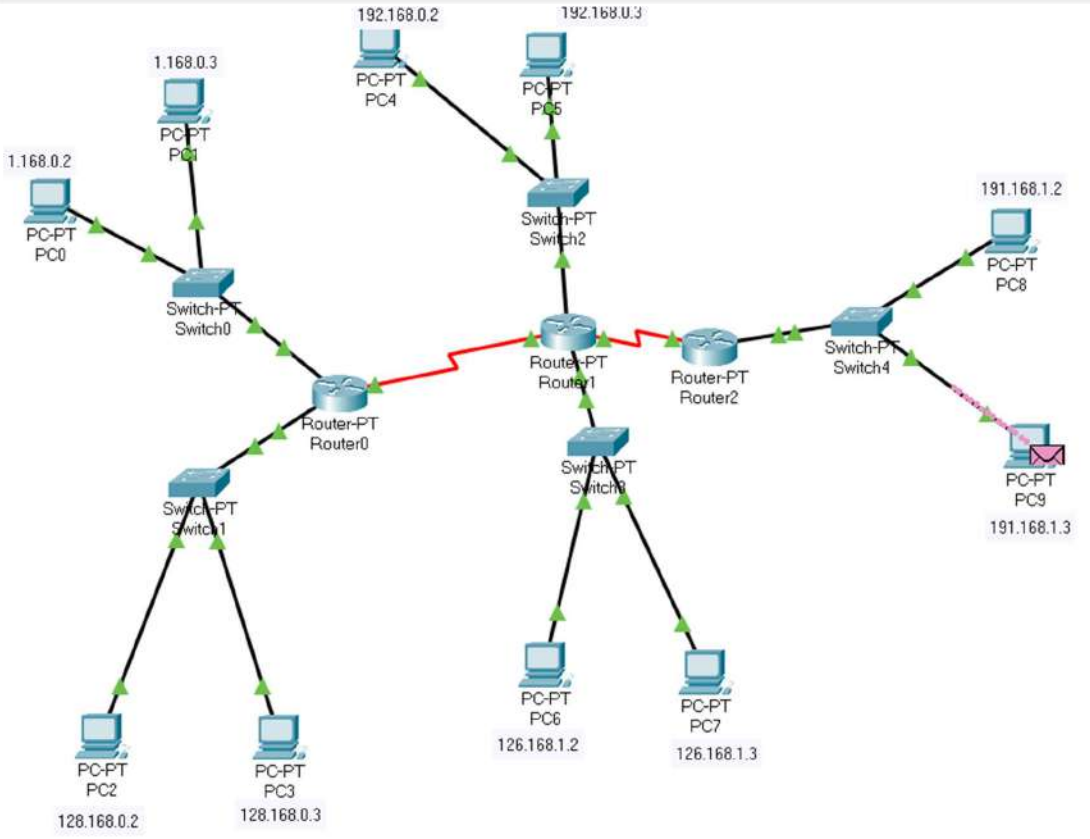

192.168.0.2
192.168.0.3
PC-PT
PC4
PC-PT
PC5
1.168.0.3
PC-PT
PC1
1.168.0.2
PC-PT
PC0
Switch-PT
Switch2
191.168.1.2
PC-PT
PC8
Switch-PT
Switch0
Router-PT
Router1
Router-PT
Router2
Switch-PT
Switch4
Router-PT
Router0
Switch-PT
Switch3
PC-PT
PC9
191.168.1.3
Switch-PT
Switch1
PC-PT
PC6
126.168.1.2
PC-PT
PC7
126.168.1.3
PC-PT
PC2
128.168.0.2
PC-PT
PC3
128.168.0.3

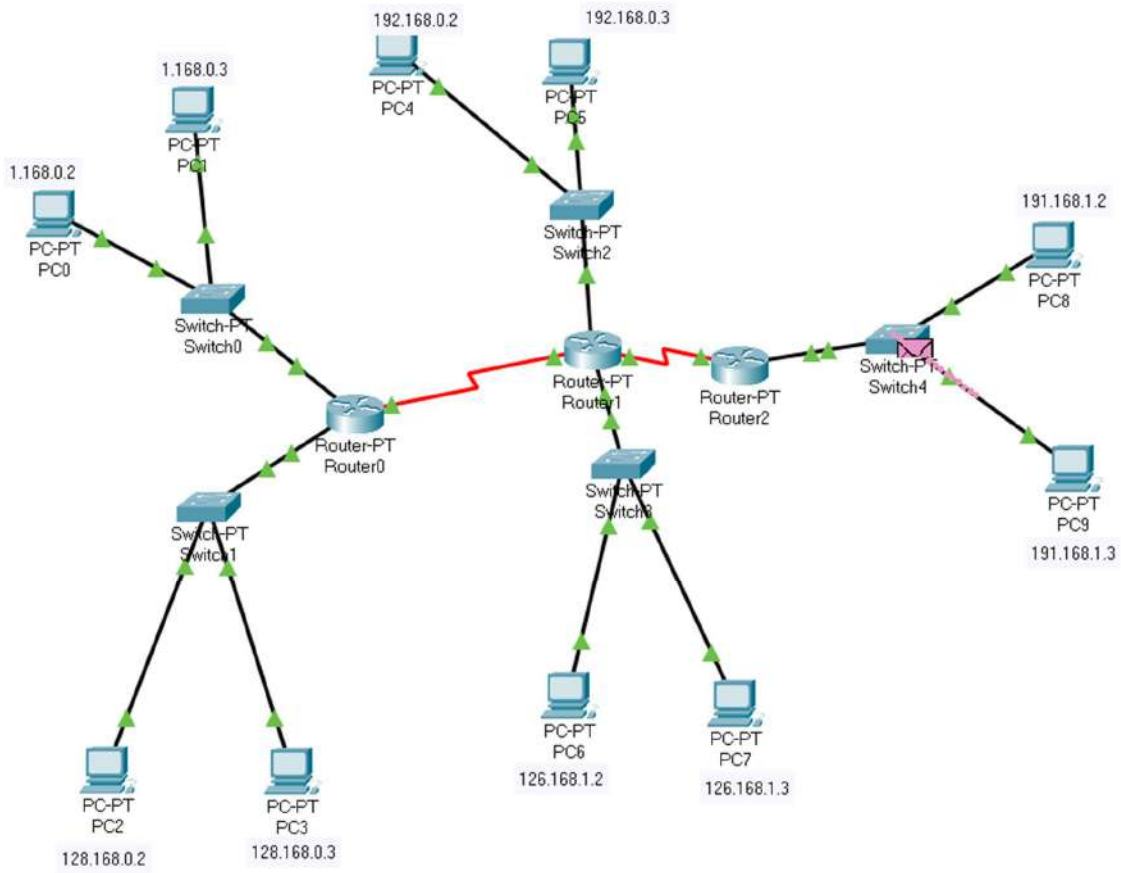

192.168.0.2
192.168.0.3
PC-PT
PC4
PC-PT
PC5
1.168.0.3
PC-PT
PC1
1.168.0.2
PC-PT
PC0
Switch-PT
Switch2
191.168.1.2
PC-PT
PC8
Switch-PT
Switch0
Router-PT
Router1
Router-PT
Router2
Switch-PT
Switch4
Router-PT
Router0
Switch-PT
Switch3
PC-PT
PC9
191.168.1.3
Switch-PT
Switch1
PC-PT
PC6
126.168.1.2
PC-PT
PC7
126.168.1.3
PC-PT
PC2
128.168.0.2
PC-PT
PC3
128.168.0.3

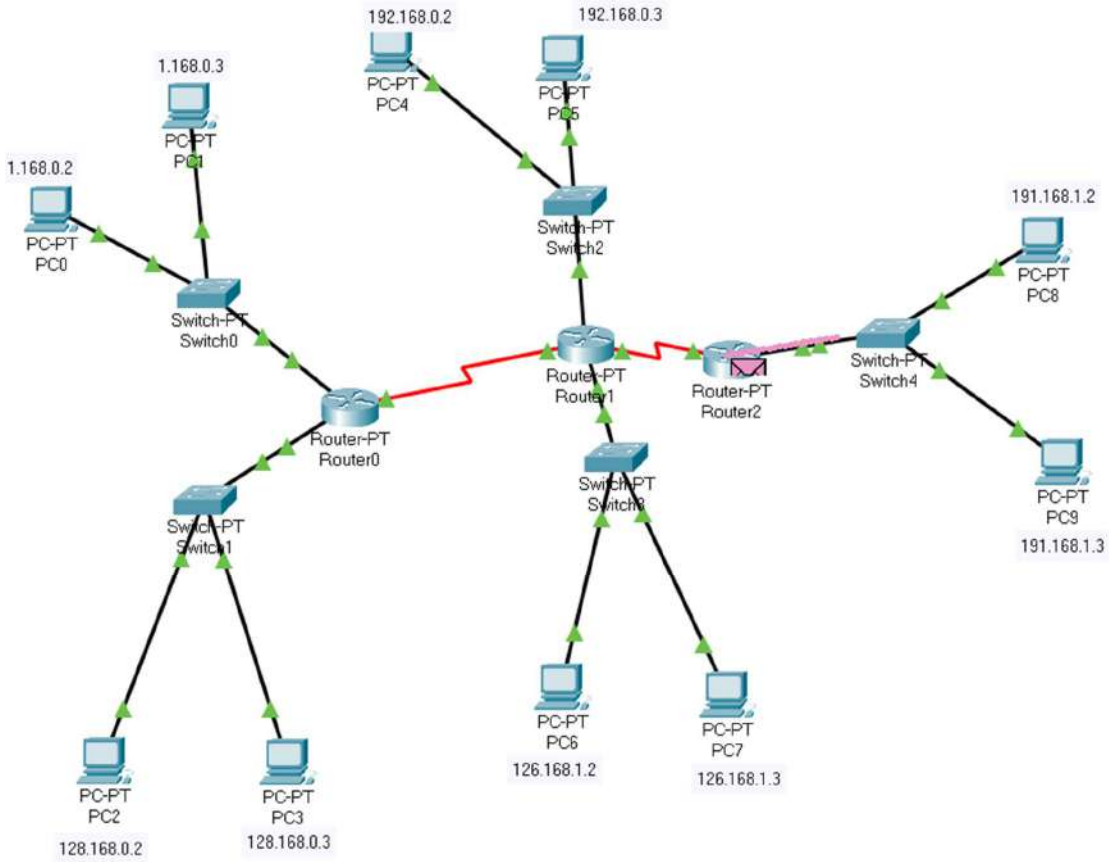

192.168.0.2
192.168.0.3
PC-PT
PC4
PC-PT
PC5
1.168.0.3
PC-PT
PC1
1.168.0.2
PC-PT
PC0
Switch-PT
Switch2
191.168.1.2
PC-PT
PC8
Switch-PT
Switch0
Router-PT
Router1
Router-PT
Router2
Switch-PT
Switch4
Router-PT
Router0
Switch-PT
Switch3
PC-PT
PC9
191.168.1.3
Switch-PT
Switch1
PC-PT
PC6
126.168.1.2
PC-PT
PC7
126.168.1.3
PC-PT
PC2
128.168.0.2
PC-PT
PC3
128.168.0.3

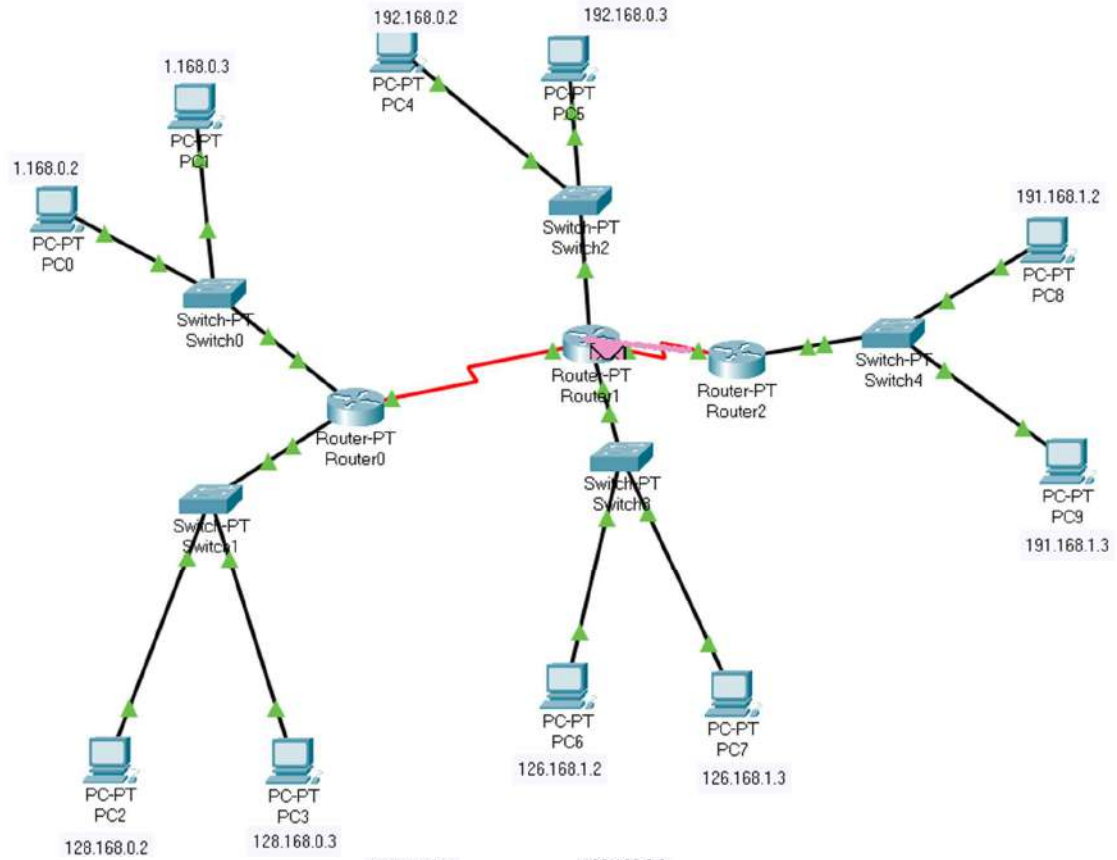

192.168.0.2
192.168.0.3
PC-PT
PC4
PC-PT
PC5
1.168.0.3
PC-PT
PC1
1.168.0.2
PC-PT
PC0
Switch-PT
Switch2
191.168.1.2
PC-PT
PC8
Switch-PT
Switch0
Router-PT
Router1
Router-PT
Router2
Switch-PT
Switch4
Router-PT
Router0
Switch-PT
Switch3
PC-PT
PC9
191.168.1.3
Switch-PT
Switch1
PC-PT
PC6
126.168.1.2
PC-PT
PC7
126.168.1.3
PC-PT
PC2
128.168.0.2
PC-PT
PC3
128.168.0.3

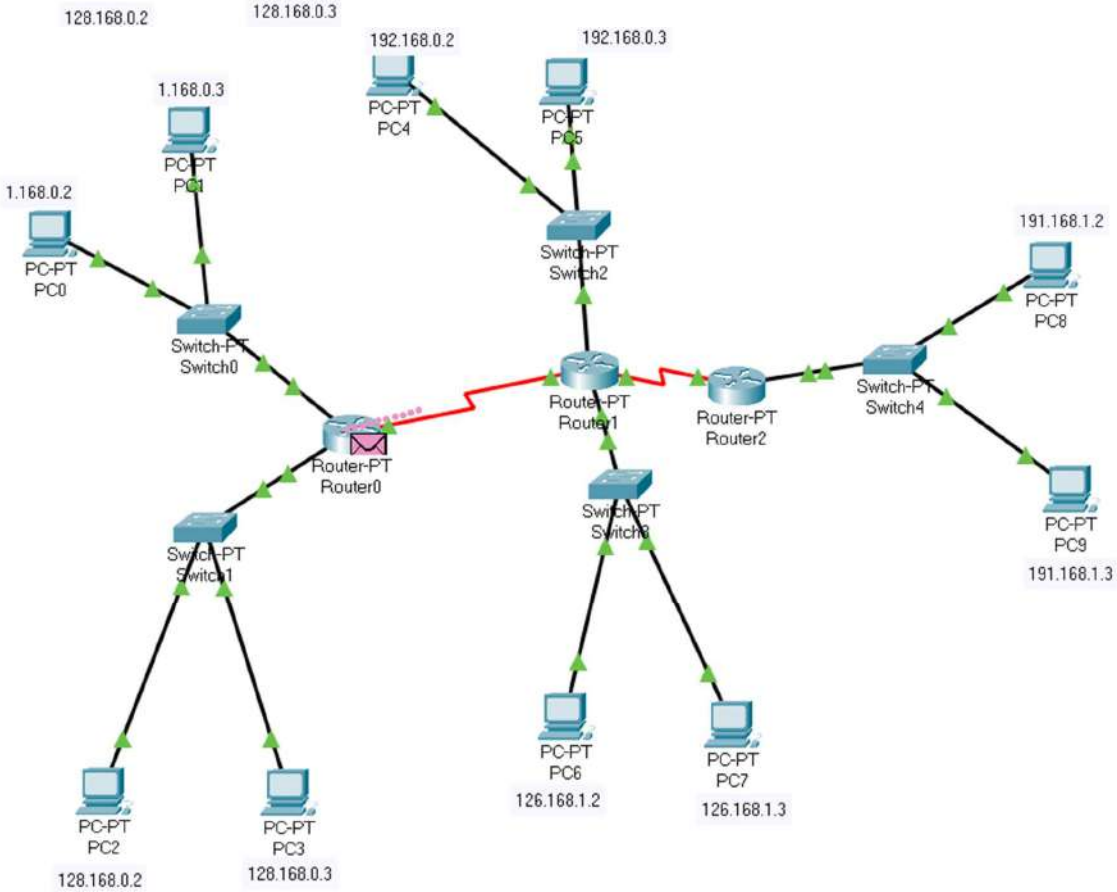

192.168.0.2
192.168.0.3
1.168.0.3
PC-PT
PC4
PC-PT
PC5
PC-PT
PC1
1.168.0.2
PC-PT
PC0
Switch-PT
Switch2
191.168.1.2
PC-PT
PC8
Switch-PT
Switch0
Router-PT
Router1
Router-PT
Router2
Switch-PT
Switch4
Router-PT
Router0
Switch-PT
Switch3
PC-PT
PC9
191.168.1.3
Switch-PT
Switch1
PC-PT
PC6
126.168.1.2
PC-PT
PC7
126.168.1.3
PC-PT
PC2
128.168.0.2
PC-PT
PC3
128.168.0.3

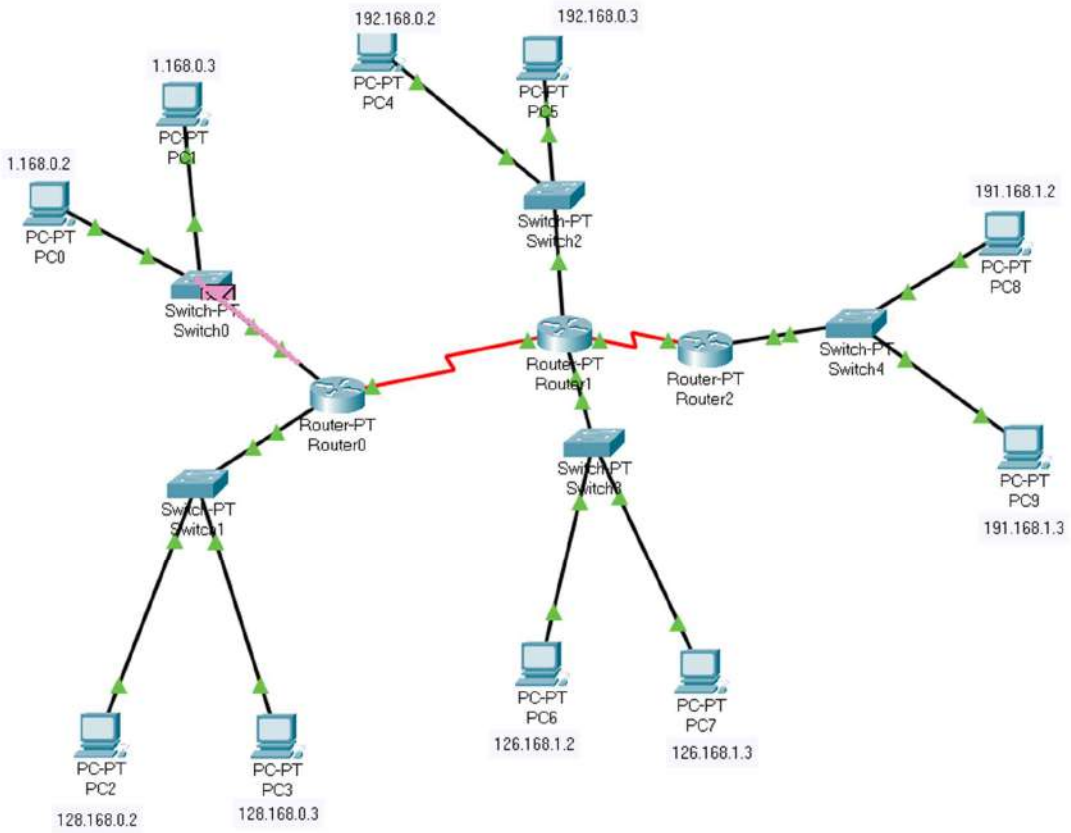

192.168.0.2
192.168.0.3
1.168.0.3
PC-PT
PC4
PC-PT
PC5
PC-PT
PC1
1.168.0.2
PC-PT
PC0
Switch-PT
Switch2
191.168.1.2
PC-PT
PC8
Switch-PT
Switch0
Router-PT
Router1
Router-PT
Router2
Switch-PT
Switch4
Router-PT
Router0
Switch-PT
Switch3
PC-PT
PC9
191.168.1.3
Switch-PT
Switch1
PC-PT
PC6
126.168.1.2
PC-PT
PC7
126.168.1.3
PC-PT
PC2
128.168.0.2
PC-PT
PC3
128.168.0.3

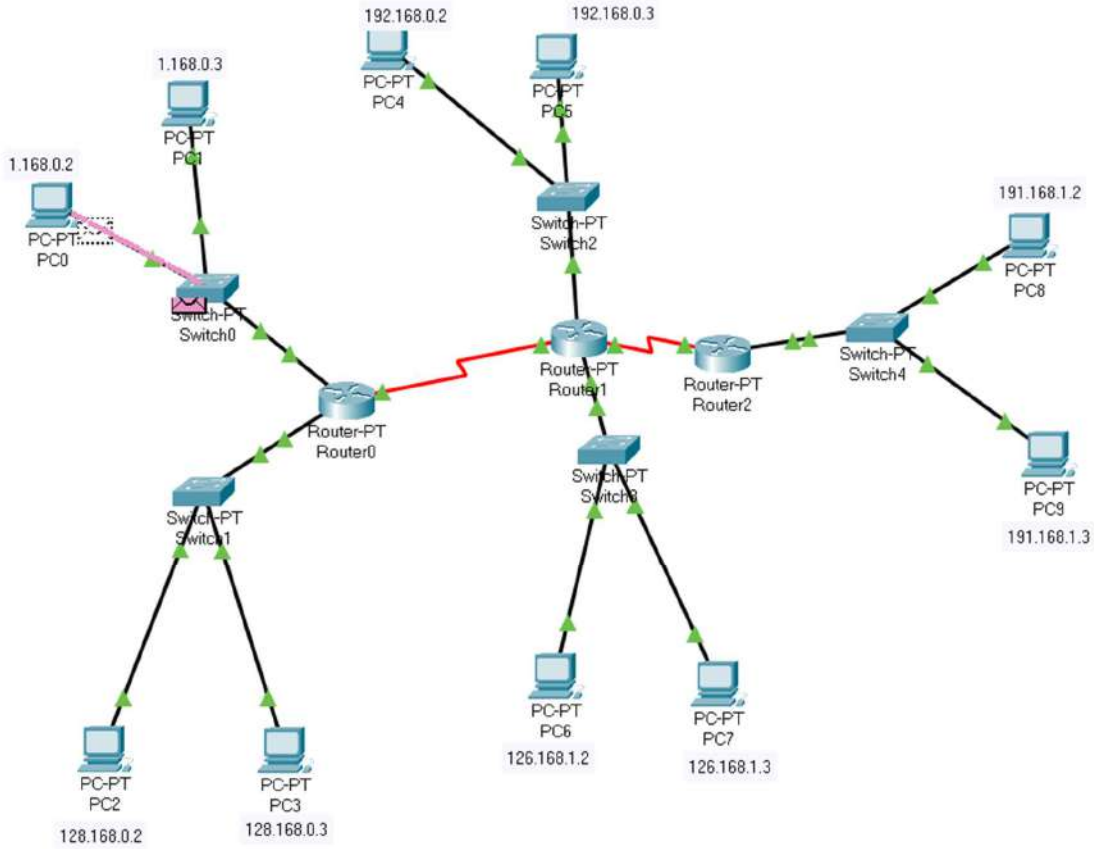

192.168.0.2
192.168.0.3
PC-PT
PC4
PC-PT
PC5
1.168.0.3
PC-PT
PC1
1.168.0.2
PC-PT
PC0
Switch-PT
Switch2
191.168.1.2
PC-PT
PC8
Switch-PT
Switch0
Router-PT
Router1
Router-PT
Router2
Switch-PT
Switch4
Router-PT
Router0
Switch-PT
Switch3
PC-PT
PC9
191.168.1.3
Switch-PT
Switch1
PC-PT
PC6
126.168.1.2
PC-PT
PC7
126.168.1.3
PC-PT
PC2
128.168.0.2
PC-PT
PC3
128.168.0.3

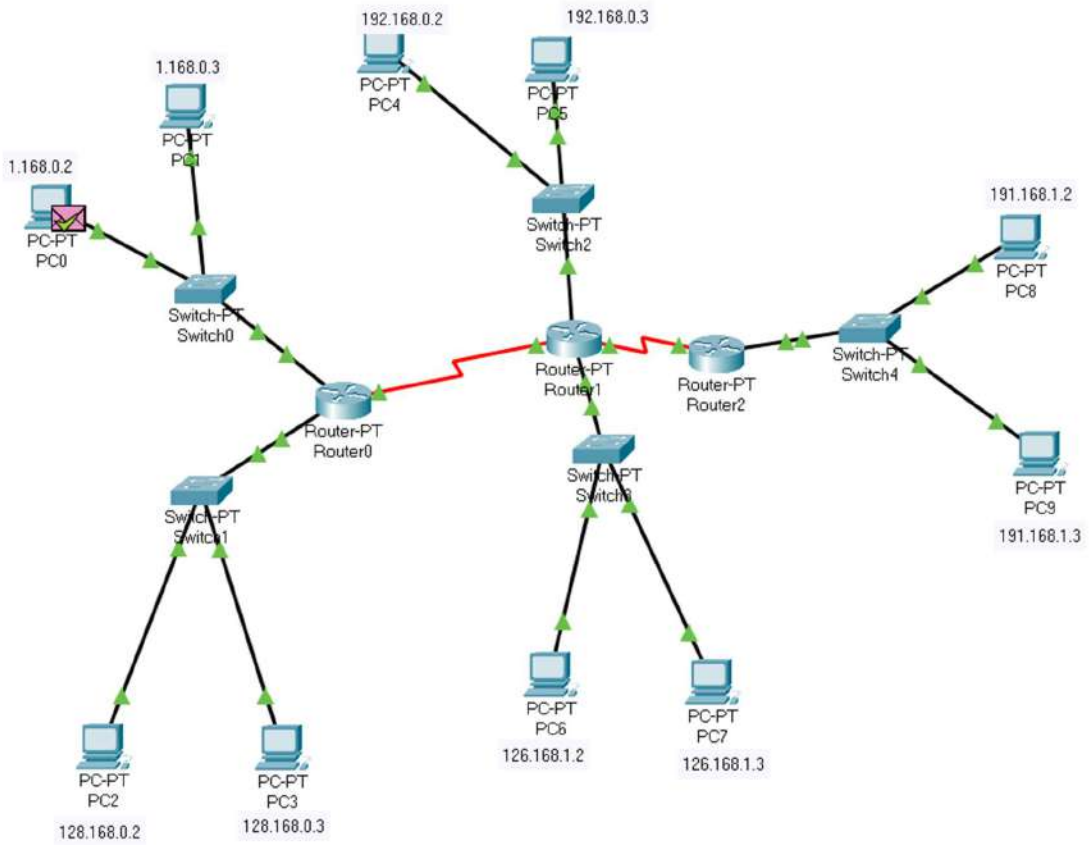

192.168.0.2
192.168.0.3
PC-PT
PC4
PC-PT
PC5
1.168.0.3
PC-PT
PC1
1.168.0.2
PC-PT
PC0
Switch-PT
Switch2
191.168.1.2
PC-PT
PC8
Switch-PT
Switch0
Router-PT
Router1
Router-PT
Router2
Switch-PT
Switch4
Router-PT
Router0
Switch-PT
Switch3
PC-PT
PC9
191.168.1.3
Switch-PT
Switch1
PC-PT
PC6
126.168.1.2
PC-PT
PC7
126.168.1.3
PC-PT
PC2
128.168.0.2
PC-PT
PC3
128.168.0.3


192.168.0.2
192.168.0.3
PC-PT
PC4
PC-PT
PC5
1.168.0.3
PC-PT
PC1
1.168.0.2
PC-PT
PC0
Switch-PT
Switch2
191.168.1.2
PC-PT
PC8
Switch-PT
Switch0
Router-PT
Router1
Router-PT
Router2
Switch-PT
Switch4
Router-PT
Router0
Switch-PT
Switch3
PC-PT
PC9
191.168.1.3
Switch-PT
Switch1
PC-PT
PC6
126.168.1.2
PC-PT
PC7
126.168.1.3
PC-PT
PC2
128.168.0.2
PC-PT
PC3
128.168.0.3


192.168.0.2
192.168.0.3
PC-PT
PC4
PC-PT
PC5
1.168.0.3
PC-PT
PC1
1.168.0.2
PC-PT
PC0
Switch-PT
Switch2
191.168.1.2
PC-PT
PC8
Switch-PT
Switch0
Router-PT
Router1
Router-PT
Router2
Switch-PT
Switch4
Router-PT
Router0
Switch-PT
Switch3
PC-PT
PC9
191.168.1.3
Switch-PT
Switch1
PC-PT
PC6
126.168.1.2
PC-PT
PC7
126.168.1.3
PC-PT
PC2
128.168.0.2
PC-PT
PC3
128.168.0.3


192.168.0.2
192.168.0.3
PC-PT
PC4
PC-PT
PC5
1.168.0.3
PC-PT
PC1
1.168.0.2
PC-PT
PC0
Switch-PT
Switch2
191.168.1.2
PC-PT
PC8
Switch-PT
Switch0
Router-PT
Router1
Router-PT
Router2
Switch-PT
Switch4
Router-PT
Router0
Switch-PT
Switch3
PC-PT
PC9
191.168.1.3
Switch-PT
Switch1
PC-PT
PC6
126.168.1.2
PC-PT
PC7
126.168.1.3
PC-PT
PC2
128.168.0.2
PC-PT
PC3
128.168.0.3


192.168.0.2
192.168.0.3
PC-PT
PC4
PC-PT
PC5
1.168.0.3
PC-PT
PC1
1.168.0.2
PC-PT
PC0
Switch-PT
Switch2
191.168.1.2
PC-PT
PC8
Switch-PT
Switch0
Router-PT
Router1
Router-PT
Router2
Switch-PT
Switch4
Router-PT
Router0
Switch-PT
Switch3
PC-PT
PC9
191.168.1.3
Switch-PT
Switch1
PC-PT
PC6
126.168.1.2
PC-PT
PC7
126.168.1.3
PC-PT
PC2
128.168.0.2
PC-PT
PC3
128.168.0.3


192.168.0.2
192.168.0.3
PC-PT
PC4
PC-PT
PC5
1.168.0.3
PC-PT
PC1
1.168.0.2
PC-PT
PC0
Switch-PT
Switch2
191.168.1.2
PC-PT
PC8
Switch-PT
Switch0
Router-PT
Router1
Router-PT
Router2
Switch-PT
Switch4
Router-PT
Router0
Switch-PT
Switch3
PC-PT
PC9
191.168.1.3
Switch-PT
Switch1
PC-PT
PC6
126.168.1.2
PC-PT
PC7
126.168.1.3
PC-PT
PC2
128.168.0.2
PC-PT
PC3
128.168.0.3