

10/10/22

~~QUESTION~~

EXP-1

Aim:- Implementing star topology using hubs and switches.

Procedure :- i) End devices are connected through the hub.

ii) Hubs are connected by using switch.

iii) IP address of end devices are configured.

iv) A simple PDU file connection between all devices are checked.

v) A simple PDU file is transmitted between a source and destination.

Result:- Message transmission between any devices is successful.

Observation :- i) PDU is first sent to the hub  
ii) Hub will broadcast to all the device connected to it, if any of the receiving device is destination it will spread message otherwise discard it.  
iii) Initially switch will broadcast to all the ports. Later fill the detail of ip address & ports in a table. And later on this table is used to broadcast a message to particular port.

→ Hub :- When source sends a packet in network the hub source the packet and ends broadcast over the network, i.e., it sends data to all the end devices in network and stops when it matches.

Result :-

PC > ping 10.0.0.1

Pinging 10.0.0.1 with 32 byte of data

Reply from 10.0.0.1 : byte=32 time=0ms

Packet sent = 4 , received = 4 , lost = 0

→ Switches :- When source device send a message to the switch once a connection is established which takes some time called learning time the switch received the packet.

Result :-

PC > Ping 10.0.0.1

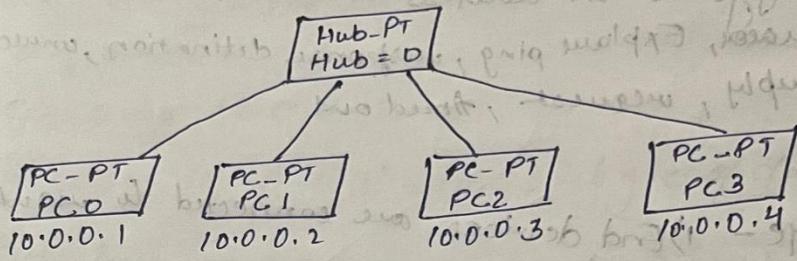
Pinging 10.0.0.1 with 32 byte of data

Reply from 10.0.0.1 : byte=32 time>0ms

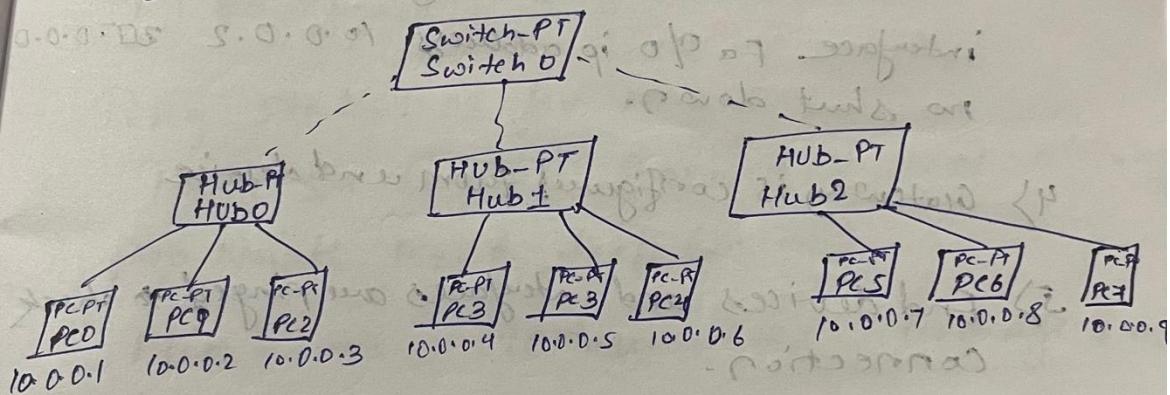
Ping statistics for 10.0.0.3

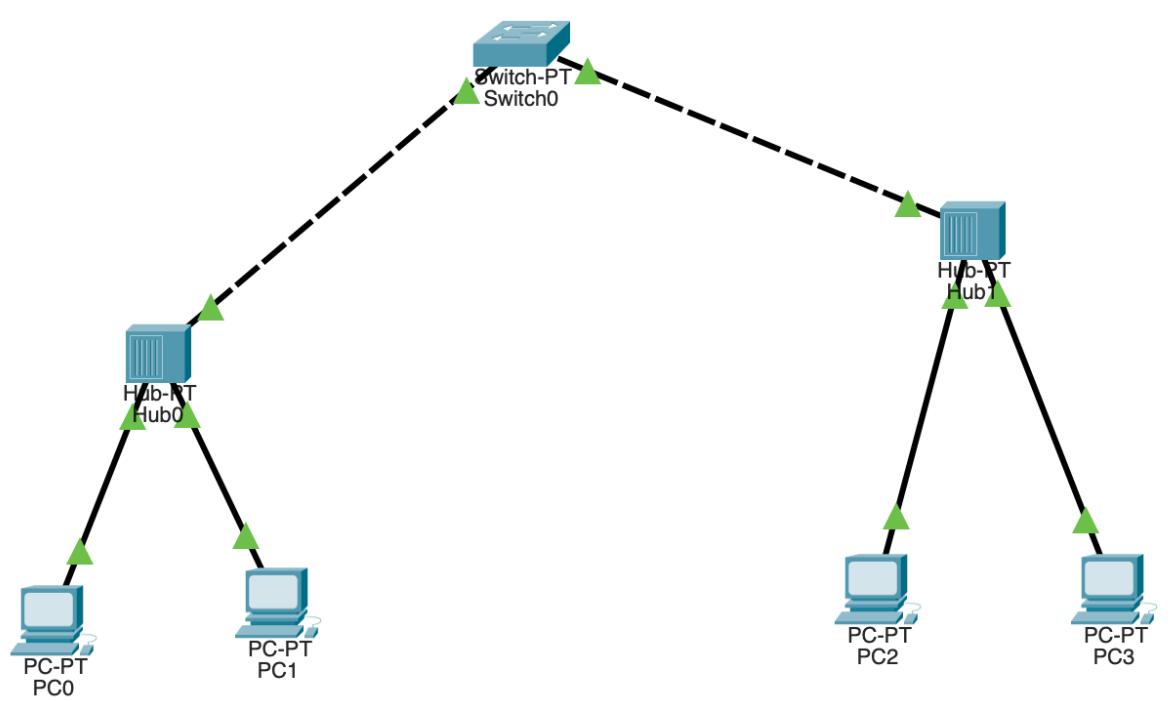
Packet sent = 4 , Received = 4 , Lost = 0 .

using Hub :-



using Switch :-





Exp-2

17/11/22

Aim — Configure IP address to router in packet tracer, Explain ping, response destination, unreachable reply, request, timed out.

- Procedure —
- 1) End devices are connected to router
  - 2) IP address is configured to end devices
  - 3) Config IP Address & subnet mask using commands enable, config terminal, interface Fa 0/0 ip address 10.0.0.2 255.0.0.0 no shut down.
  - 4) Gateway is configured from end device
  - 5) End devices and interfaces are pinged to check connection.

Topology — Star Topology.

Result — Successfully pinged end devices.

Observation — When we configure both end devices and router with appropriate ip address and by configuring subnet mask of interface of router as 255.0.0.0 and gateway of PC0 is set as 10.0.0.2 which is of fa0/0 interface followed by same for PC1.

Then we could successfully Ping.

when gateway of end devices is not configured  
then we get request time out.  
~~after configuration~~

PC > Ping 10.0.0.1

Pinging 10.0.0.1 with 32 byte of data

Reply from 10.0.0.1 byte=32

time=6ms TTL=128

Reply from 10.0.0.1 byte=32

time=2ms TTL=128

Reply from 10.0.0.1 byte=32

time=5ms TTL=128

Reply from 10.0.0.1 byte=32 time=3ms TTL=128

Ping statistics for 10.0.0.1

Packets: Sent=4, Received=4, Lost=0 (0% loss)

Approximate round trip (in ms)

Min=2ms Maximum=6ms Average=4ms

before configuration:-

PC > Ping 10.0.0.1

~~Pinging 10.0.0.1 with 32 byte of data~~

Request timed out

Request timed out

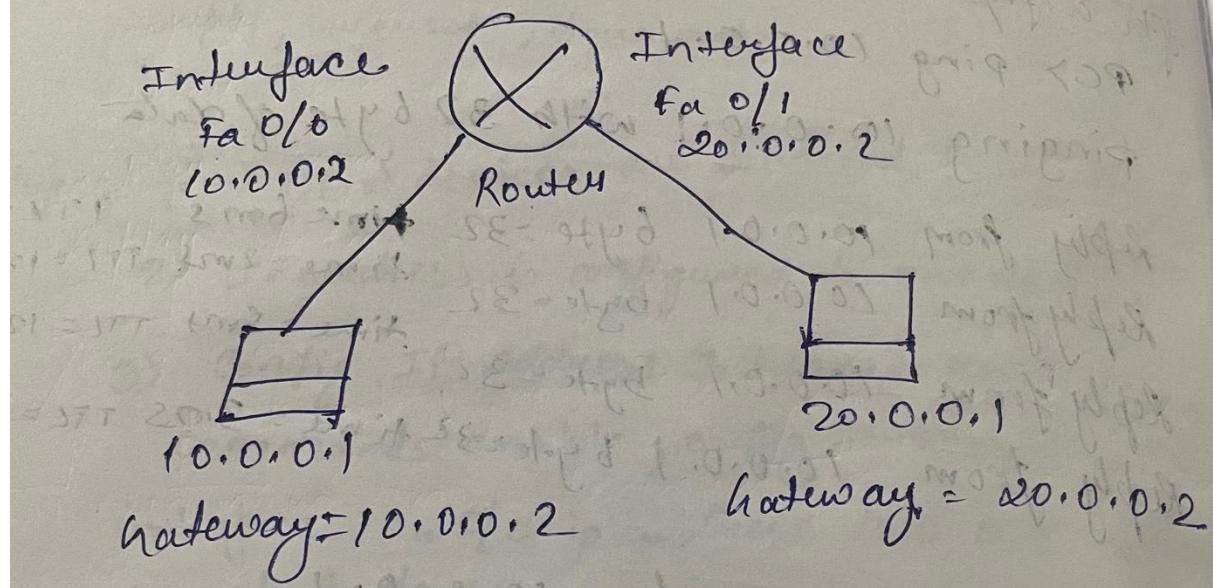
Request timed out

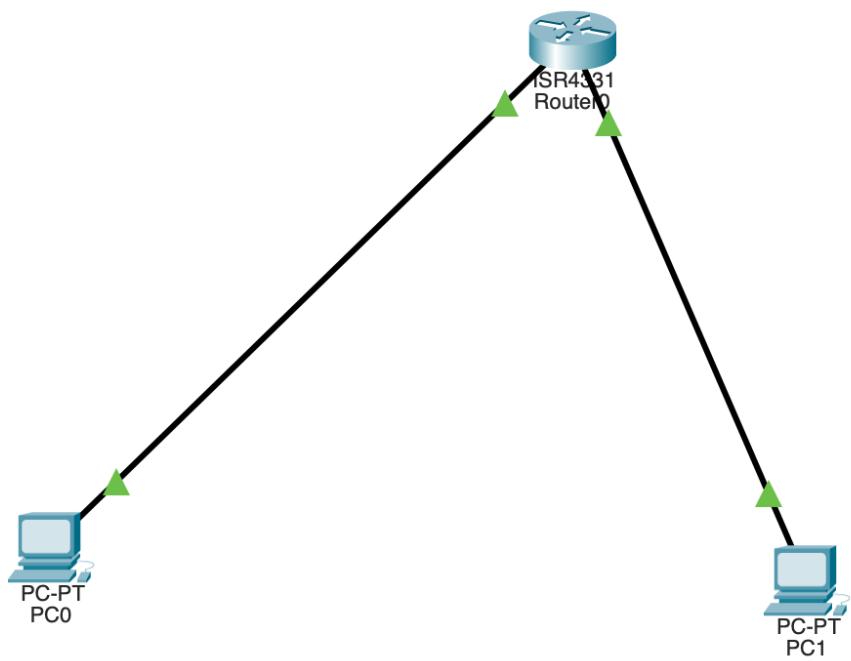
Request timed out

Ping statistics for 10.0.0.1

Packets: Sent=4, Received=0, Lost=4 (100% loss)

## Topology:-

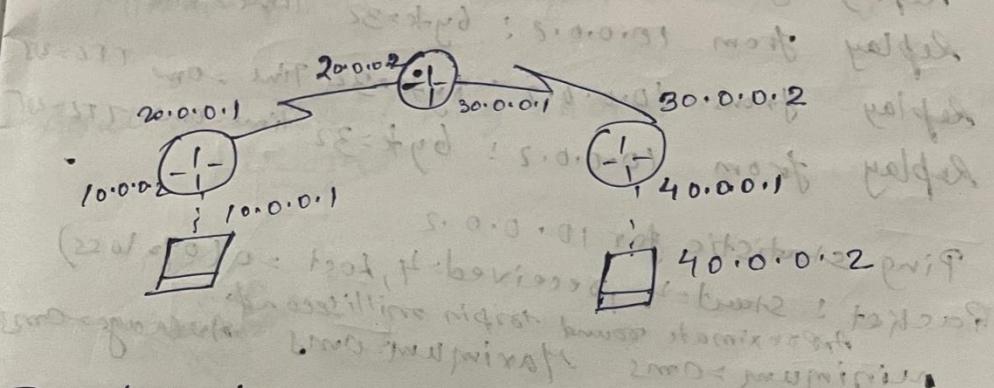




24/11/22 2nd day session Ex-3 from notes - (Prakash 2019)

Aim - Configure default gateway to router

Topology - Star Topology.



Procedure :-

- 1) Select two end devices, configure their fast ethernet ip address to 10.0.0.1 & 40.0.0.1
- 2) Select configure the three routers in their CLI
- 3) Started pinging from end devices to their nearest router, it shows request timed out
- 4) Therefore we need to set up path for end devices from one router to other router end.
- 5) For each router in privilege mode given command show ip route. It shows networks that are directly connected to that router.
- 6) There must be two router path for each router.
- 7) Then start ping each devices, it shows successful display.

Observation! - Before configuring router paths.  
From one device PC-PT(10.0.0.1) ping next router  
(10.0.0.2)

Replay from 10.0.0.2 : byte=32 Time=20ms TTL=255

Replay from 10.0.0.2 : byte=32 Time=0ms TTL=255

Replay from 10.0.0.2 : byte=32 Time=0ms TTL=255

Replay from 10.0.0.2 : byte=32 Time=0ms TTL=255

Ping statistic for 10.0.0.2

Packet's sent=4, received=4, Lost=0 (0% loss)  
Approximate round trip milliseconds:  
minimum=0ms Maximum=0ms Average=0ms

- \* Then from the same device PC-PT (10.0.0.1) ping 20.0.0.1, 20.0.0.2
- It shows request timeout.

Then set gateway for end devices.

Then pinged 20.0.0.1 it shows successful reply.

Then pinged 30.0.0.1 it shows destination host unreachable.

After configuring router paths.

- \* Ping end device. PC-PF(10.0.0.1) to PC-PT2(30.0.0.1)

Pinging from 40.0.0.1 with 32 byte of data

Replay from 40.0.0.1 : byte=32 time=16ms TTL=105

Replay from 40.0.0.1 : byte=32 time=21ms TTL=105

Replay from 40.0.0.1 : byte=32 time=21ms TTL=105

Replay from 40.0.0.1 : byte=32 time=14ms TTL=105

Ping statistic for 40.0.0.1

• Packets : Sent = 4, Received = 4, Lost = 0 (0% loss)

Approximate round trip in millisecond.

Minimum = 2ms, Maximum = 21ms, Average = 13ms.

Result :- Successful ping only on manually connecting networks.



• Network provides link layer as well as physical layer. It maintains and provides services like connection between LAN & LAN or between LAN & WAN. It also provides security, reliability, and QoS.

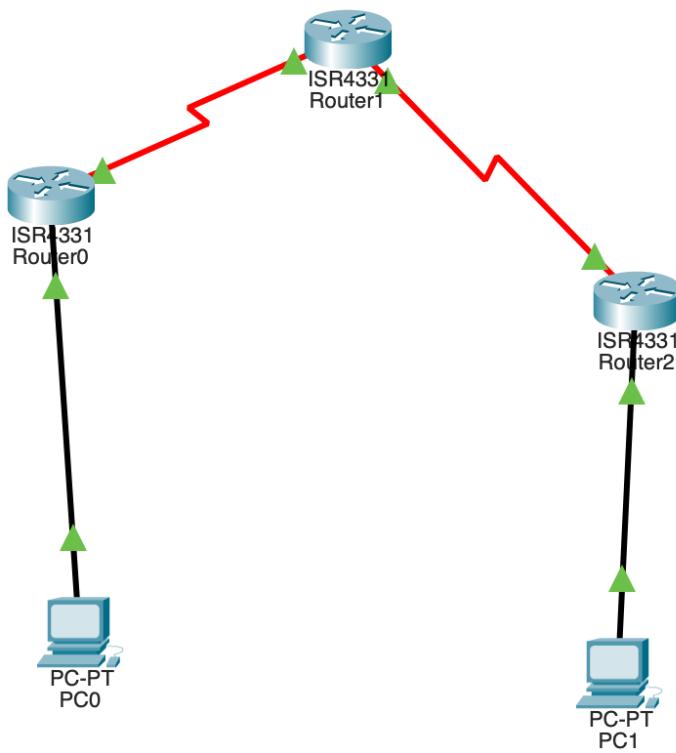
• Network layer is responsible for end-to-end delivery of data.

• Data transfer: Data flow between hosts is described in terms of packets.

• Data transfer: Data flow between hosts is described in terms of packets.

• Data transfer: Data flow between hosts is described in terms of packets.

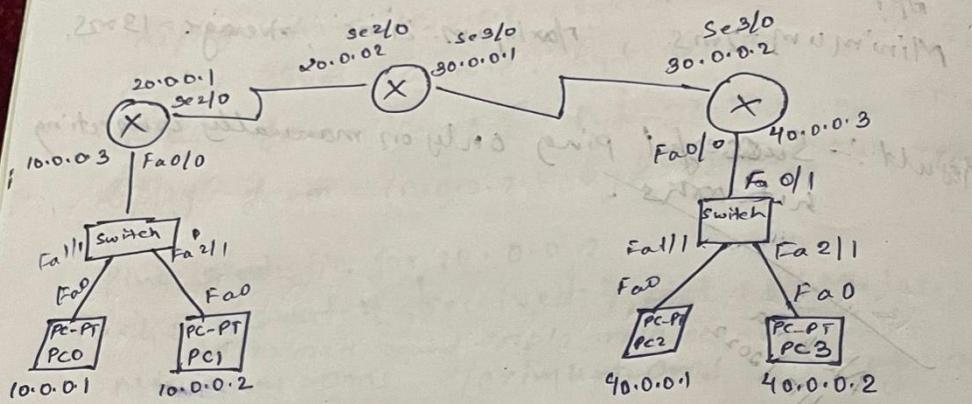
• Data transfer: Data flow between hosts is described in terms of packets.



1/12/22

Ex-4

Aim - Configuring default route to the routers.



Procedure:- A topology was created using 3 routers, 2 switch and 2 PC-PT connected to each switch using copper straight and serial DCE connection.

- \* Default gateway and unique IP address were configured for each PC.
- \* IP address were configured for each interface using CLI.

Observation!- \* Pinging PC 2 from PC 0 gave destination host unreachable.

- \* IP routes for each router was viewed using ! show ip route .
- \* Static ip route was configured for router 1 by ip route > destination network > subnet mask > next hop address.

\* Default ip configuration was for interface 0 and interface 2 by 0.0.0.0 0.0.0.0 next-hop address.

\* Pinging PC2 from PC0 gave required reply.

Result! -

PC> ping 10.0.0.3

Pinging 10.0.0.3 with 32 bytes of data

Reply from 10.0.0.3: byte = 32 time = 0ms TTL = 255

Reply from 10.0.0.3: byte = 32 time = 1ms TTL = 255

Reply from 10.0.0.3: byte = 32 time = ~0ms TTL = 255

Reply from 10.0.0.3: byte = 32 time = ~0ms TTL = 255

PC> ping 20.0.0.1

Pinging 20.0.0.1 with 32 bytes of data

Reply from 20.0.0.1: byte = 32 time = 0ms TTL = 255

Reply from 20.0.0.1: byte = 32 time = 0ms TTL = 255

Reply from 20.0.0.1: byte = 32 time = 5ms TTL = 255

Reply from 20.0.0.1: byte = 32 time = 0ms TTL = 255

Ping statistics for 20.0.0.1

Package sent = 4, Received = 4, Loss = 0 (0% loss)

Approximate round trip in ms:

Maximum = 5ms Minimum = 0ms Average = 1ms

PC> ping 30.0.0.2

Pinging 30.0.0.2 with 32 byte of data:

Request timed out

Reply from 30.0.0.2: byte=32 time=12ms TTL=253

Request timed out

Reply from 30.0.0.2: byte=32 time=2ms TTL=253

Ping statistics for 30.0.0.2

Packet sent = 4, Received = 2, Lost = 2 (50% loss)

Approx round trip in ms:

Minimum = 2ms Maximum = 12ms Average = 7ms.

PC> ping 50.0.0.1

Reply: Pinging 50.0.0.1 with 32 byte of data?

Reply from 50.0.0.1: byte=32 time=2ms TTL=125

Reply from 50.0.0.1: byte=32 time=3ms TTL=125

Reply from 50.0.0.1: byte=32 time=7ms TTL=125

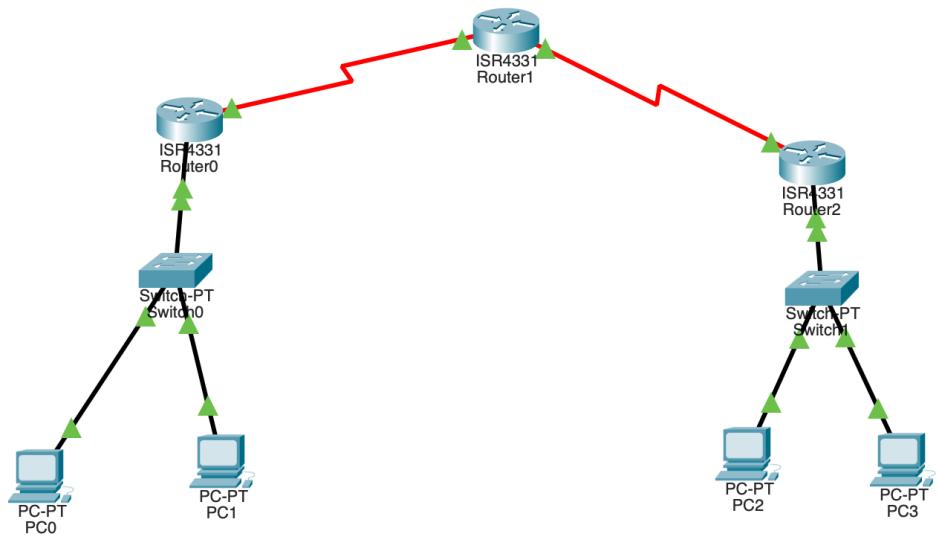
Reply from 50.0.0.1: byte=22 time=10ms TTL=125

Ping statistic for 50.0.0.1

Packet sent = 4, Received = 4, Lost = 0 (0% loss)

Approx round trip in ms:

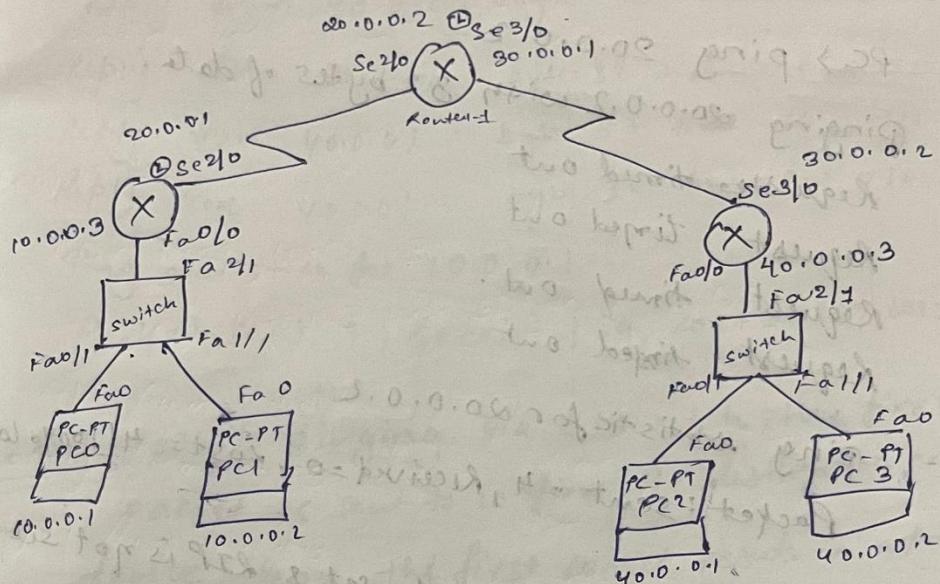
Minimum = 2ms Maximum = 10ms Average = 5ms.



*8/Sept/22*

Ex-5

Aim - Configure RIP Routing Protocol in Routers.



Procedure :- Created a topology using 3 routers, 2 switches, 2 PC-PT connected to each switch using copper straight and DCE cable.

- \* Default gateway and unique IP address was configured for each PC.
- \* IP address were configured for each interface using CLI.

#### Observations

- \* Configure router ip address using ip address <address> <subnet-mask>

- \* Encapsulation ppp and clock-rate 64000 used to specify ppp protocol and clock rate in router 0 and 1 for the serial port.
- \* Configure RIP routing using rip network <address> command, where network refers .

### Output:-

(i) When no gateway or RIP is set

PC > ping 20.0.0.2

Pinging 20.0.0.2 with 32 bytes of data!

Request timed out

Request timed out

Request timed out

Request timed out

Ping statistic for 20.0.0.2

Packet: Sent = 4, Received = 0, Lost = 4 (100% loss)

(ii) When only gateway is set & RIP is not set.

PC > ping 30.0.0.1

Pinging 30.0.0.1 with 32 byte of data:

Reply from 10.0.0.3: Destination host unreachable

Ping statistics for 30.0.0.1

Packet: Sent = 4, Received = 0, Lost = 4 (100% loss).



(iii) When both gateway & RIP is set.

PC> Ping 40.0.0.1

Reply from 40.0.0.1 : byte = 32 time = 9ms TTL=125

Reply from 40.0.0.1 : byte = 32 time = 2ms TTL=125

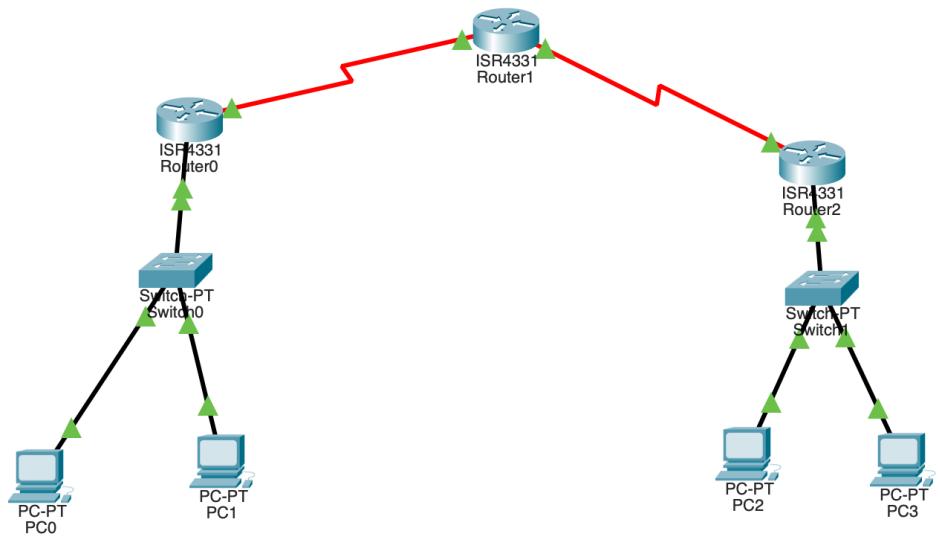
Reply from 40.0.0.1 : byte = 32 time = 16ms TTL=125

Reply from 40.0.0.1 : byte = 32 time = 9ms TTL=125

Ping statistics for 40.0.0.1

Packet sent 4, Received = 4, Lost = 0 (0% loss)

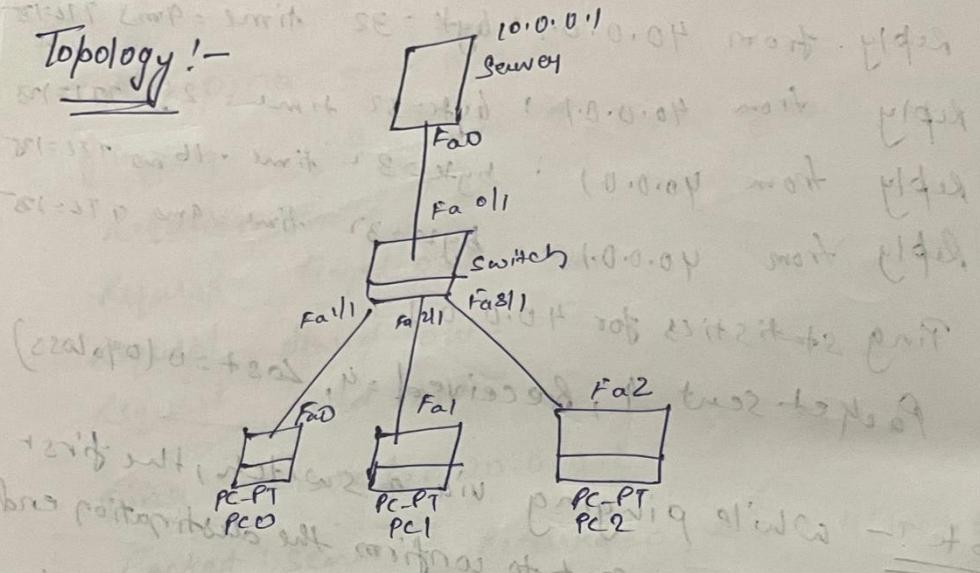
Note! - While pinging via a switch, the first packet is sent to confirm the destination end device. Hence, the first packet results in armed out situation. 4 packets are sent & 3 packets are received.



15/1/22

## Aim:- Configure DHCP Server

### Topology :-



Procedure:- 1) Configure ip address for the server 10.0.0.1

2) Select server and configure DHCP in server and turn on service.

3) Assign any pool name and give start ip address of the server & the subnet mask is also added.

4) We can ~~either~~ add DNS server ~~of the same~~ <sup>same</sup> ~~as~~ <sup>as</sup> that of server's ip address ~~as well~~.

5) Once we click save the pool is created

6) When we click on the end devices in order to config the ip address and select DHCP in the ip configuration the ip address is automatically configured.

### Observation:-

DHCP (Dynamic host configuration Protocol)  
 we don't have to give the ip for end devices manually  
 it dynamically configures.

### Result:-

PC > ping 10.0.0.3  
 Pinging 10.0.0.3 with 32 bytes of data.

Reply from 10.0.0.3 : byte = 32 time = 8ms TTL = 128

Reply from 10.0.0.3 : byte = 32 time = 0ms TTL = 128

Reply from 10.0.0.3 : byte = 32 time = 0ms TTL = 128

Reply from 10.0.0.3 : byte = 32 time = 0ms TTL = 128

Reply from 10.0.0.3 : byte = 32 time = 0ms TTL = 128

Ping statistics for 10.0.0.3

Packets: Sent = 4, Received = 4 Lost = 0 (0% Loss)

Approximate round trip times in ms:

PC > ping 10.0.0.1

Pinging 10.0.0.1 with 32 bytes of data

Reply from 10.0.0.1 : byte = 32 time = 0ms TTL = 128

Reply from 10.0.0.1 : byte = 32 time = 0ms TTL = 128

Reply from 10.0.0.1 : byte = 32 time = 0ms TTL = 128

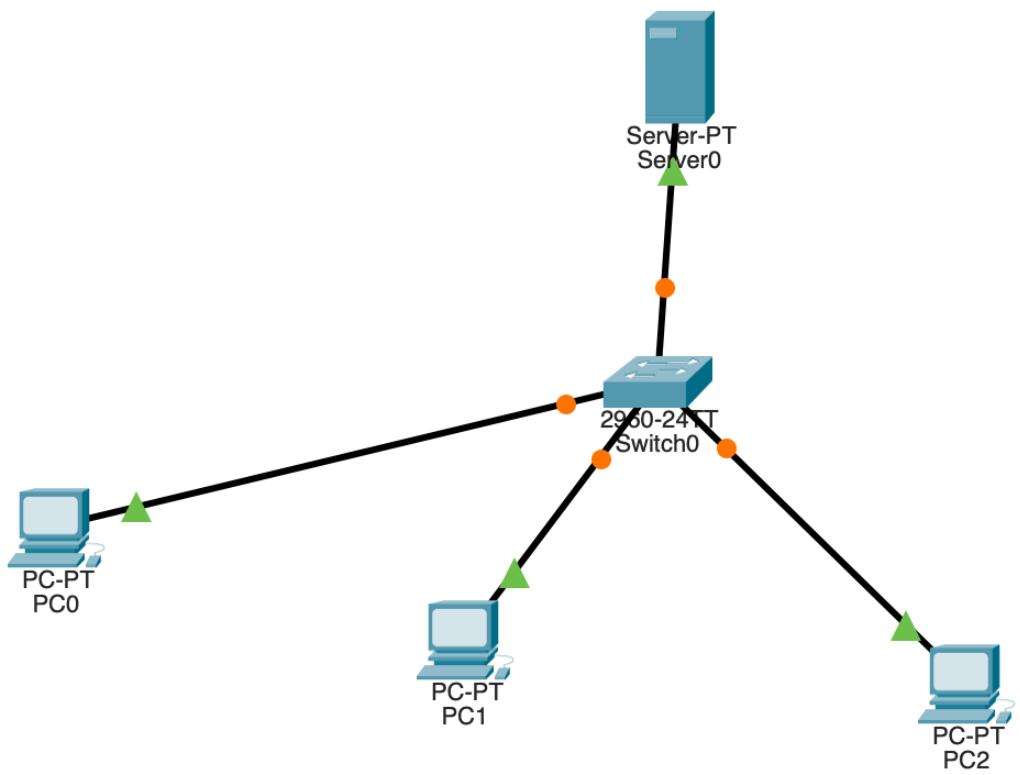
Reply from 10.0.0.1 : byte = 32 time = 0ms TTL = 128

Reply from 10.0.0.1 : byte = 32 time = 0ms TTL = 128

Ping statistics for 10.0.0.1

Packets: Sent = 4, Received = 4, Lost = 0 (0% Loss)

Approximate round trip times in ms:

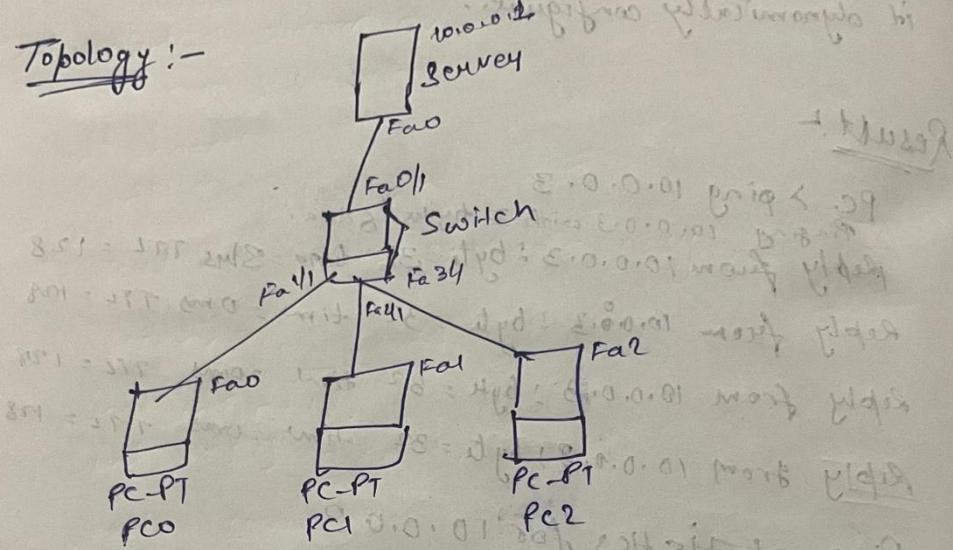


1st year

EX-7

Aim :- Configure WebServer and DNS service.

Topology :-



Procedure :- ① Configure ip address of server 10.0.0.1

② Select server and config DHCP server and turn on service

③ Assign any pool name and give start ip address of the server

④ We can add DNS server same that of server's IP address

⑤ We click on end devices to config IP address select DHCP in the IP and the IP address is config automatically.

⑥ Click on server and services and HTTP make it on.

⑦ Then on DNS make it on and put name : www.queerty.com  
type : A Record Address : 10.0.0.1 → same as server IP.

⑧ After click on add and the goto end web browser in end devices and type www.queerty.com.

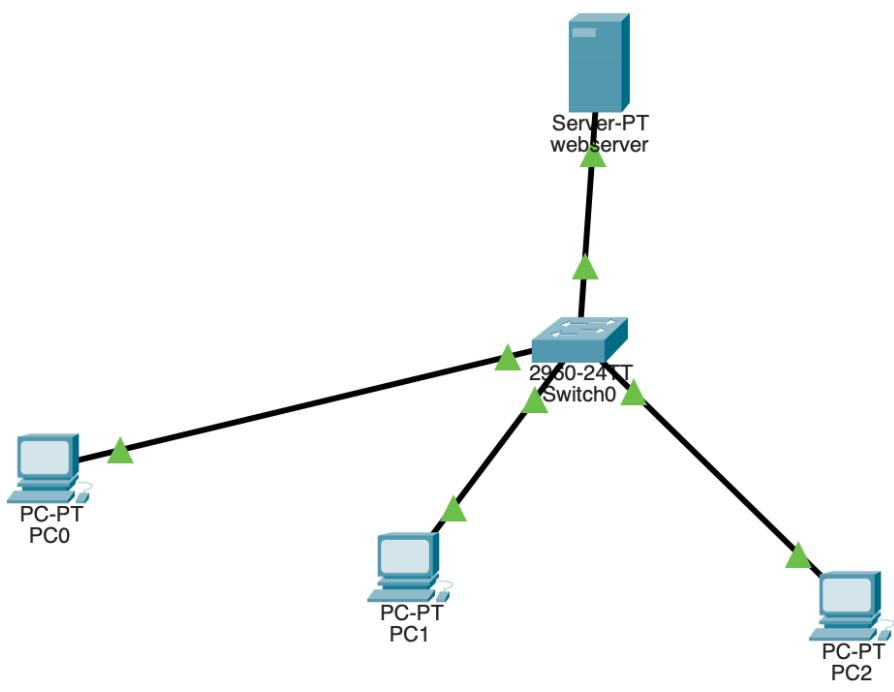
### Observation:-

DNS server maps domain name with IP address of the server.

Result:- ① If configured properly the page of Cisco packet tracker is opened.

② If not configured properly i.e. DNS server and default gateway, 'Host Unresolved' is shown.

Neelima  
15/12/2022



29/12/22 Write a program for error detection using <sup>LAB-8</sup> CRC - 16 bit.

```
#include <iostream.h>
using namespace std;
int main()
{ int i=0, j=0, n;
cout << "Enter no. of data entry: ";
cin >> n;
int a[n+1];
int g[16] = {1, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0};
int rem[n];
int m= n+1;
for(i=0; i<m; i++)
{ if (i>n)
    a[i]=0;
else
    cin >> a[i];
    if (i<n)
        rem[i]=a[i];
}
for(i=0; i<n; i++)
{ if (a[i] != 1)
    continue;
for(j=0; j<17; j++)
{ a[i+j]=a[i+j]^g[j];
}}
```

```

for (i=0; i < n; i++)
{
    a[i] = rem[i];
}
for (i=0; i < m; i++)
{
    cout << a[i];
}

```

Output :-

2

```

1
1
1
111 0 11 00 000 11 0 00 11
001 00 0 0 0 1 000 000 0 0 1

```

N  
3/1/2023

Note :- (one of 2 output) sum of two nos. =

```
Enter no of data
2
1
1

.....
11000000000000000000
.....
110011000001100011
.....
0000000000000000Program ended with exit code: 0
```



All Output ▾

Filter



## Leaky Bucket

8/1/22

```
#include <iostream>
using namespace std;

int main()
{
    cout << "Enter bucket size" << endl;
    uint bucket size;
    int filled = 0;
    int outputrate, inputrate, choice;
    cin >> bucket size;
    cout << "Enter outputrate" << endl;
    cin >> outputrate;
    do {
        cout << "Enter packet size" << endl;
        cin >> inputrate;
        if (inputrate <= bucket size) {
            if (filled + inputrate > bucket size)
                cout << "Packet too big" << endl;
            else {
                filled = filled + inputrate;
            }
        } else {
            cout << "Packet is too big" << endl;
        }
        if (filled <= outputrate)
            if (filled == 0)
                cout << "Amount of bucket filled" << filled;
                cout << "Do you want to continue (1 to yes 2 to no)" << endl;
                cin >> choice;
    } while (choice == 1);
}
```

Output:- Enter Bucket size : 500

Enter output mate : 50

Enter packet size : 700

Packet too big.

Do you want to continue (1 to yes 2 to no) : 1

Enter packet size : 250

Bucket filled 150

Do you want to continue (1 to yes 2 to no) : 1

Enter packet size : 250

Bucket filled 350.

Do you want to continue (1 to yes 2 to no) : 2

✓ 12/1/2023

```
Enter the bucket capacity: 300
Enter the outflow rate: 30
1.Insert
2.ExitEnter choice: 1

Enter the packet size: 250
250
After outflow: 2201.Insert
2.ExitEnter choice: 1

Enter the packet size: 200
Bucket overflow1.Insert
2.ExitEnter choice: 1

Enter the packet size: 100
Bucket overflow1.Insert
2.ExitEnter choice: 1

Enter the packet size: 30
250
After outflow: 2201.Insert
2.ExitEnter choice: 2
No more inputs. Program exited1.Insert
2.ExitEnter choice: |
```

Bellman Ford.

12/1/23

```

→ #include <stdio.h>
#include <stdlib.h>
int Bellman_Ford (int G[20][20], int V, int E,
                   int edge[20][20])

```

→ #include <iostream>

```

#define MAX_ID
using namespace std;
typedef struct edge
{
    int src;
    int dest;
    int wt;
} edge;
void Bellman_Ford (int nv, edge e[], int src_graph,
                    int ne)
{
    int u, v, weight, i, j = 0;
    int dis[MAX];
    for (i = 0; i < nv; i++)
        dis[i] = 999;
    dis[src - graph] = 0;
    for (i = 0; i < nv - 1; i++)
        for (j = 0; j < ne; j++)
            {
                u = e[j].src;
                v = e[j].dest;
                weight = e[j].wt;
                if (dis[u] != 999)
                    dis[v] = dis[u] + weight;
            }
    for (j = 0; j < ne; j++)
        {
            u = e[j].src;
            v = e[j].dest;
            weight = e[j].wt;
        }
}

```

```

if (dis[u] + weight < dis[v])
    cout << "Negative cycle present";
}
cout << "vertex " << "Distance from source";
for (i=1; i<=nv; i++)
    cout << "\n" << i << ":" << dis[i];
}

int main()
{
int nv, ne, src_graph;
edge e[NMAX];
cout << "Enter the no. of vertices!";
cin >> nv;
cout << "Enter the source vertex of graph";
cin >> src_graph;
cout << "Enter no. of edges";
cin >> ne;
cout << "Enter no. of edges";
cin >> ne;
for (i=0; i<ne; i++)
{
    cout << "for edge " << i+1 << " ";
    cout << "Enter source vertex ";
    cin >> e[i].src;
    cout << "Enter destination vertex ";
    cin >> e[i].dest;
    cout << "Enter weight ";
    cin >> e[i].wt;
}
}

```

*N P  
2/2/2022 return 0*  
**Bellman-Ford (nv, e, src-graph, ne)**  
 Output :-  
 Enter no. of vertices: 4  
 Enter graph in matrix form  
 0 5 4 9 9 9  
 5 0 6 3  
 9 9 3 1 6  
 2 0 1 4

Enter source vertex: 1.

Vertex 1 → cost = 0 parent = 0  
 Vertex 2 → cost = 5 parent = 1  
 Vertex 3 → cost = 4 parent = 1  
 Vertex 4 → cost = 8 parent = 2

No negative weight cycle.

---

```
Enter the number of vertices: 4
Enter the source vertex of the graph: 1

Enter no. of edges: 5

For edge 1=>
Enter source vertex :1
Enter destination vertex :2
Enter weight :4

For edge 2=>
Enter source vertex :1
Enter destination vertex :3
Enter weight :5

For edge 3=>
Enter source vertex :3
Enter destination vertex :2
Enter weight :7

For edge 4=>
Enter source vertex :2
Enter destination vertex :4
Enter weight :7

For edge 5=>
Enter source vertex :4
Enter destination vertex :3
Enter weight :10

Vertex Distance from source
1   0
2   4
3   5
4   10Program ended with exit code: 0
```

12/1/23

### Dijkstra

```
#include <stdio.h>
#define INFINITY 9999
#define MAX 10
void dijkstra(int G[MAX][MAX], int n, startnode),
int main()
{
    int G[MAX][MAX], i, j, n, u;
    printf("Enter no. of vertices:");
    scanf("%d", &n);
    printf("Enter adjacency matrix:");
    for (i=0; i<n; i++)
        for (j=0; j<n; j++)
            scanf("%d", &G[i][j]);
    printf("Enter starting node:");
    scanf("%d", &u);
    dijkstra(G, n, u);
    return 0;
}
void dijkstra(int G[MAX][MAX], int n, int startnode)
{
    int cost[MAX][MAX], distance[MAX], pred[MAX];
    int visited[MAX], count, mindistance, nextnode, i, j;
    for (i=0; i<n; i++)
        for (j=0; j<n; j++)
            if (G[i][j] == 0)
                cost[i][j] = infinity;
            else
                cost[i][j] = G[i][j];
    for (i=0; i<n; i++)
        distance[i] = cost[startnode][i];
    pred[i] = startnode;
    visited[i] = 0;
    while (count < n - 1)
    {
        mindistance = INFINITY;
        for (i=0; i<n; i++)
            if (visited[i] == 0)
                if (distance[i] < mindistance)
                    mindistance = distance[i];
                    nextnode = i;
        if (mindistance == INFINITY)
            break;
        visited[nextnode] = 1;
        for (j=0; j<n; j++)
            if (visited[j] == 0)
                if (cost[nextnode][j] + distance[nextnode] < distance[j])
                    distance[j] = cost[nextnode][j] + distance[nextnode];
                    pred[j] = nextnode;
        count++;
    }
    distance[startnode] = 0;
    visited[startnode] = 1;
}
```

```

for (i=0; i<n; i++)
    if (i == startnode)
        printf("Distance of node %d = %d, i, distance[i].D");
        printf("Path = %d", i);
    else {
        j = pred[i];
        printf(" < %d, j");
        while (j != startnode)
            j = pred[j];
        printf(" before %d", j);
    }
}

```

**Output:** Enter no. of vertices : 4

Enter the adjacency matrix :

0	5	4	999
5	0	6	3
999	3	1	6
2	0	1	4

Enter the starting node : 2

Distance of node 0 = 5

Path = 0 → 1

Distance of node 2 = 4

Path = 2 → 3 → 1

Distance of node 3 = 3

Path = 3 → 1

N  
2/2/2022

---

```
Enter the graph
0 9 2 5
9 0 6 8
2 6 0 0
5 8 0 0
Vertex      Distance from Source
0            0
1            8
2            2
3            5
Program ended with exit code: 0
```

## TCP/IP Socket Write client-Server programs

⇒ Client TCP

```
from socket import *
serverName = '127.0.0.1'
serverPort = 62000
```

```
ClientSocket = socket (AF_INET, SOCK_STREAM)
ClientSocket . connect ((serverName, serverPort))
Sentence = input ("Enter filename").
```

```
ClientSocket . send (Sentence . encode ())
```

```
fileContents = ClientSocket . recv (1024) . decode ()
```

```
print ("from Server")
```

```
print (fileContents)
```

```
ClientSocket . close ()
```

⇒ Server TCP.

```
from socket import *
```

```
serverName = '127.0.0.1'
```

```
serverPort = 12000
```

```
serverSocket = socket (AF_INET, SOCK_STREAM)
```

```
serverSocket . bind ((serverName, serverPort))
```

```
serverSocket . listen (1)
```

```
while True:
```

```
    print ("The Server is ready to receive")
```

```
connectionSocket, addr = serverSocket . accept ()
```

```
Sentence = connectionSocket . recv (1024) . decode ()
```

```
file = open (Sentence, "r")
```

```
d = file . read (1024)
```

```
connectionSocket . send (d . encode ())
```

```
print ("Sent contents of " + Sentence)
```

```
file . close ()
```

```
connectionSocket . close ()
```

```
D:\5th sem\Computer-Network-Lab\Lab10\TCP>python -u client.py
Enter file name: file.txt
From Server:
this is a file
```

```
D:\5th sem\Computer-Network-Lab\Lab10\TCP>python -u server.py
The server is ready to receive
Sent contents of file.txt
The server is ready to receive
```

## UDP Socket      write client - Server program

⇒ Client UDP

```
from socket import *
serverName = '127.0.0.1'
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_DGRAM)
Sentence = input("Enter filename:")
clientSocket.sendto(bytes(Sentence, "utf-8"), (serverName, serverPort))
fileContents, serverAddress = clientSocket.recvfrom(2048)
print("Reply from Server")
print(fileContents.decode("utf-8"))
clientSocket.close()
```

⇒ Server UDP

```
from socket import *
serverPort = 12000
serverSocket = socket(AF_INET, SOCK_DGRAM)
serverSocket.bind(("127.0.0.1", serverPort))
print("The server is ready")
while True:
    sentence, clientAddress = serverSocket.recvfrom(2048)
    sentence = sentence.decode("utf-8")
    file = open(sentence, "r")
    d = file.read(2048)
    serverSocket.sendto(bytes(d, "utf-8"), clientAddress)
    print(f"Sent content of {d}, end={'}")
    print(sentence)
    file.close()
```

```
The server is ready to receive
Sent contents of  serverUDP.py
```

```
Enter file name: serverUDP.py

Reply from Server:

from socket import *
serverPort = 12000
serverSocket = socket(AF_INET, SOCK_DGRAM)
serverSocket.bind(("127.0.0.1", serverPort))
print ("The server is ready to receive")
while 1:
    sentence, clientAddress = serverSocket.recvfrom(2048)
    sentence = sentence.decode("utf-8")
    file=open(sentence,"r")
    l=file.read(2048)
    serverSocket.sendto(bytes(l,"utf-8"),clientAddress)
    print ('\nSent contents of ', end = ' ')
    print (sentence)
# for i in sentence:
# print (str(i), end = '')
    file.close()
```