Lab 1

9/06/23

Basic terms definition:

1] LAN: It consists of a series of computers linked together to form a network in circumscribed location.

2] WAN: A computer network that connects smaller networks that is not tied to a single location.

3] Ethernet: A system for connecting a number of computers systems to form a LAN with protocols to control the passing of information b/w systems

4] IP address: A unique string of characters that identify each computer using internet protocol to communicate over a network.

5] Hub: It is a node that broadcasts data to every computer or ethernet based device that is connected to it.

6] Switch: It connects devices in a network to each other enabling them to talk by exchanging data packets.

7] Server: It is a computer program or device that provides a service to another computer program and its user known as client.

8] End device: They are either the source or destination of data transmitted over the network.

9] Node: The connection point among network devices such as routes, printers or switches that can recieve and send data from one end point to another.

steps involved:

Step 1: Drag and drop the PC and server to workspace.

Step 2: Firstly, copper straight-through cable was selected and connected the devices with it. If the red lights are shown remove that and copper cross-over cable is selected and green lights are shown

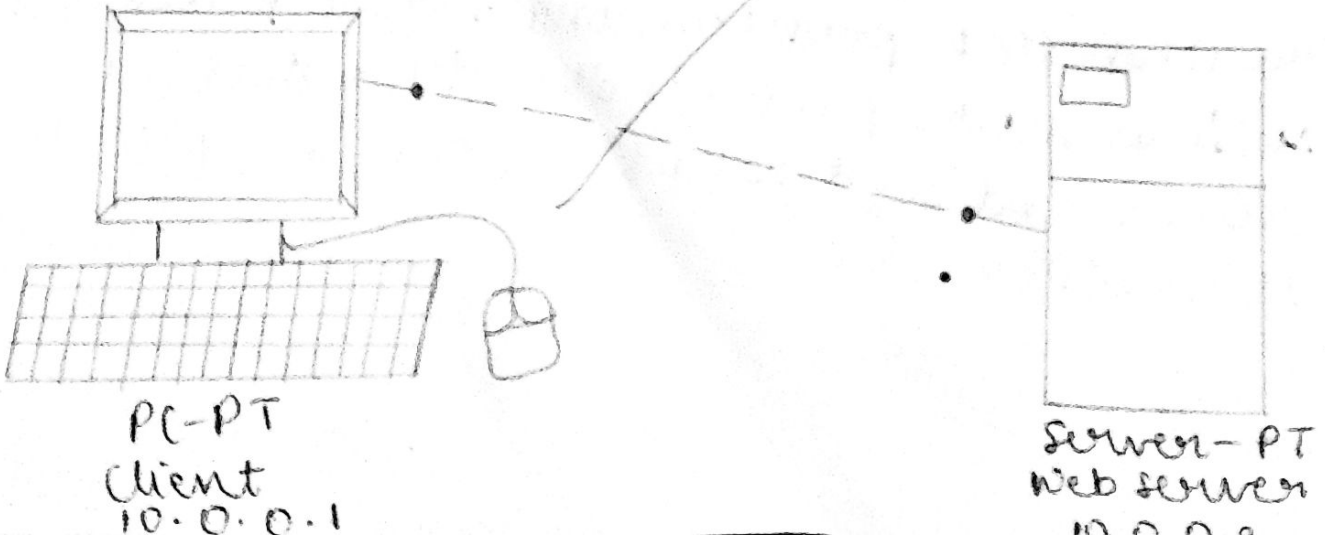
Step 3: Click on the PC. while paying attention to the link lights, turn the power on, off and on. Same step is followed for servers.

Step 4: Open PC configuration window, set the display name as client and DNS server to 10.0.0.2. Under Interface, click fast ethernet and set the IP address as 10.0.0.1

Step 5: Open server configuration window, change the display name to web server and set IP address as 10.0.0.2

Step 6: Open server service. Click DNS and set the Domain name as www.firstlab.com. Set the IP address as 10.0.0.2 and click add

Step 7: Save the work using file>save option.



Realtime [In command Prompt]

ping 10.0.0.1

Pinging 10.0.0.1 with 32 bytes of data:

Reply from 10.0.0.1 : bytes = 32 time=0ms TTL=128
Reply from 10.0.0.1 : bytes = 32 time=1ms TTL=128
Reply from 10.0.0.1 : bytes = 32 time=5ms TTL=128
Reply from 10.0.0.1 : bytes = 32 time=3ms TTL=128

Ping statistics for 10.0.0.1:
Packets: sent = 4, Recieved = 4, Lost = 0 (0% loss)
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 5ms, Average = 2ms

Ping 10.0.0.3

Pinging 10.0.0.3 with 32 bytes of data:

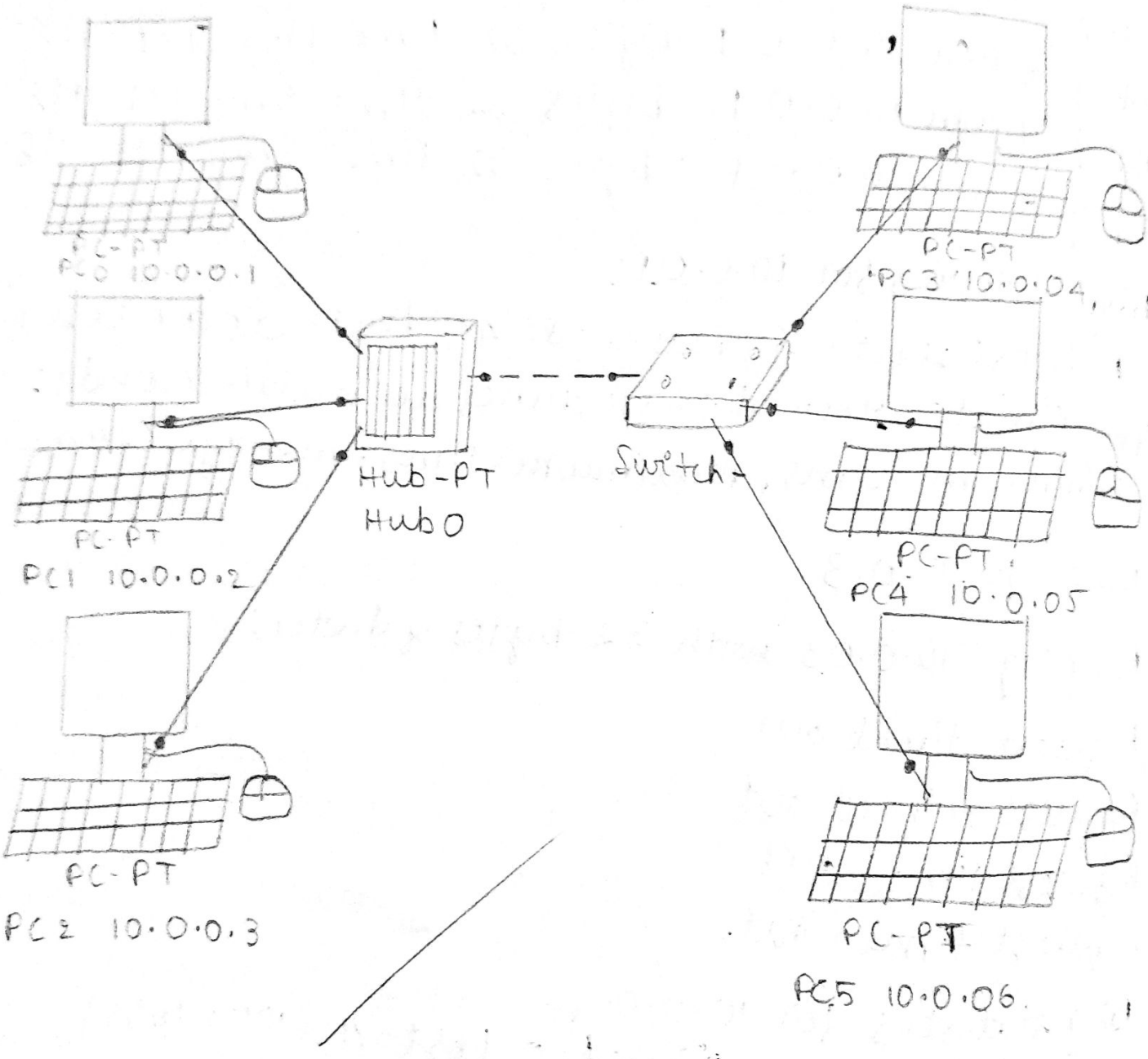
Request timed out
Request timed out
Request timed out
Request timed out.

Ping statistics for 10.0.0.3:
Packets: sent=4, Recieved=0, Lost=4 (100% loss)

4/6/6

16/06/20

① Create a topology and simulate sending a simple PDU and source to destination using simple hub and switch as connecting domains.



Step 1: Drag and drop 6 PCs, 1 Hub and 1 switch to workspace

Step 2: Set the IP address to all 6 PCs

Step 3: Take a packet and select the source and destination PCs. In simulation mode, select auto capture/play. If connection is proper, it will be successful.

Step 4: Take a packet, then select the source and destination PCs in network connected by only Hub. In simulation mode, select play. If connect-ion is proper event will be successful. In real mode, ping with IP address of other.

Step 5: Off the switch, then repeat Step 4. Since switch is off, the event will be failed.

Step 6: Do the above operations in real mode.

Output:

S1: when switch is off

Ping 10.0.0.1 (in 10.0.0.4)

Ringing 10.0.0.1 with 32 bytes of data:

Request timed out

Request timed out

Request timed out

Request timed out

Ping statistics for 10.0.0.1:

Packets: sent=4, Recieved=0, Lost=4 (100% loss)

S2: Packet transfer via Hub and switch:

Ping 10.0.0.5 (in 10.0.0.3)

Pinging 10.0.0.5 with 32 bytes of data:

Reply from 10.0.0.5: bytes=32 time=0ms TTL=128
Reply from 10.0.0.5: bytes=32 time=0ms TTL=128
Reply from 10.0.0.5: bytes=32 time=0ms TTL=128
Reply from 10.0.0.5: bytes=32 time=0ms TTL=128

Ping statistics for 10.0.0.5:
Packets: sent=4, Recieved=4, Lost=0(0% Loss)
Approximate round trip times in milli seconds:
Minimum=0ms, Maximum=0ms, Average=0ms

no

chat) Router> enable

Lab 2

23/06/2023

configure IP address to routers in packet tracer. Explore the following messages: ping responses, destination unreachable, request timed out, reply.

steps:

1) Select 2 PCs and a router. Set the IP address and gateway of the PC's and connect them to the router.

2) Configure the router using the following steps:

→ Router> enable
→ Router> config terminal
→ Router(config)# interface fastEthernet 0/0
→ Router(config-if)# ip address 10.0.0.2 255.0.0.0
→ Router(config-if)# no shutdown.
→ Router(config-if)# interface fastEthernet 1/0
→ Router(config-if)# ipaddress 20.0.0.2 255.0.0.0
→ Router(config-if)# no shutdown.

3) Now Use PDU's to tramit messages through router ~~for~~ from 1 PC to another.

4) Select another 2 PC's and router and repeat the procedure mentioned above with different IP addresses and gateway

5) Connect the router of these 2 networks to another router.

6) Now configure this router according to their interface i.e., serial 2/0 → IP address - 60.0.0.1
serial 3/0 → IP address - 50.0.0.1

7] After all these steps, go to command line Interface of all 3 routers and set ip routes.

Eg: Router# config t → Router 1

Router(config)# ip route 30.0.0.0 255.0.0.0 50.0.0.
Router(config)# ip route 10.0.0.0 255.0.0.0 50.0.0.
Router(config)# ip route 60.0.0.0 255.0.0.0 50.0.0.

similarly, do it for router 2 with IP's 10.0.0.0, 20.0.0.0, 50.0.0.0 and gateway 60.0.0.2

Router 3:

Router(config)# ip route 10.0.0.0 255.0.0.0 50.0.0.2
Router(config)# ip route 20.0.0.0 255.0.0.0 50.0.0.2
Router(config)# ip route 30.0.0.0 255.0.0.0 60.0.0.2
Router(config)# ip route 40.0.0.0 255.0.0.0 60.0.0.2

8] After these steps,

Router(config)# exit
Router# show ip route which shows the IP routes.

9] Then, using command prompt ping the ip address of PC's for different scenarios.

Output:

S1: Pinging PC3 on PC1

PC> ping 30.0.0.1

pinging -30.0.0.1 with 32 bytes of data:

Reply from 10.0.0.2: Destination host unreachable
Reply from 10.0.0.2: Destination host unreachable
Reply from 10.0.0.2: Destination host unreachable
Reply from 10.0.0.2: Destination host unreachable.

Ping statistics for 30.0.0.1
Packets: Sent = 4, Recieved = 0, Lost = 4 (100% loss)

S2: Pinging PC4 on PC1

PC> ping 40.0.0.1

pinging 40.0.0.1 with 32 bytes of data:

Request timed out
Request timed out
Request timed out
Request timed out

Ping statistics for 40.0.0.1:
Packets: Sent=4, Recieved=0, Lost=4 (100% loss).

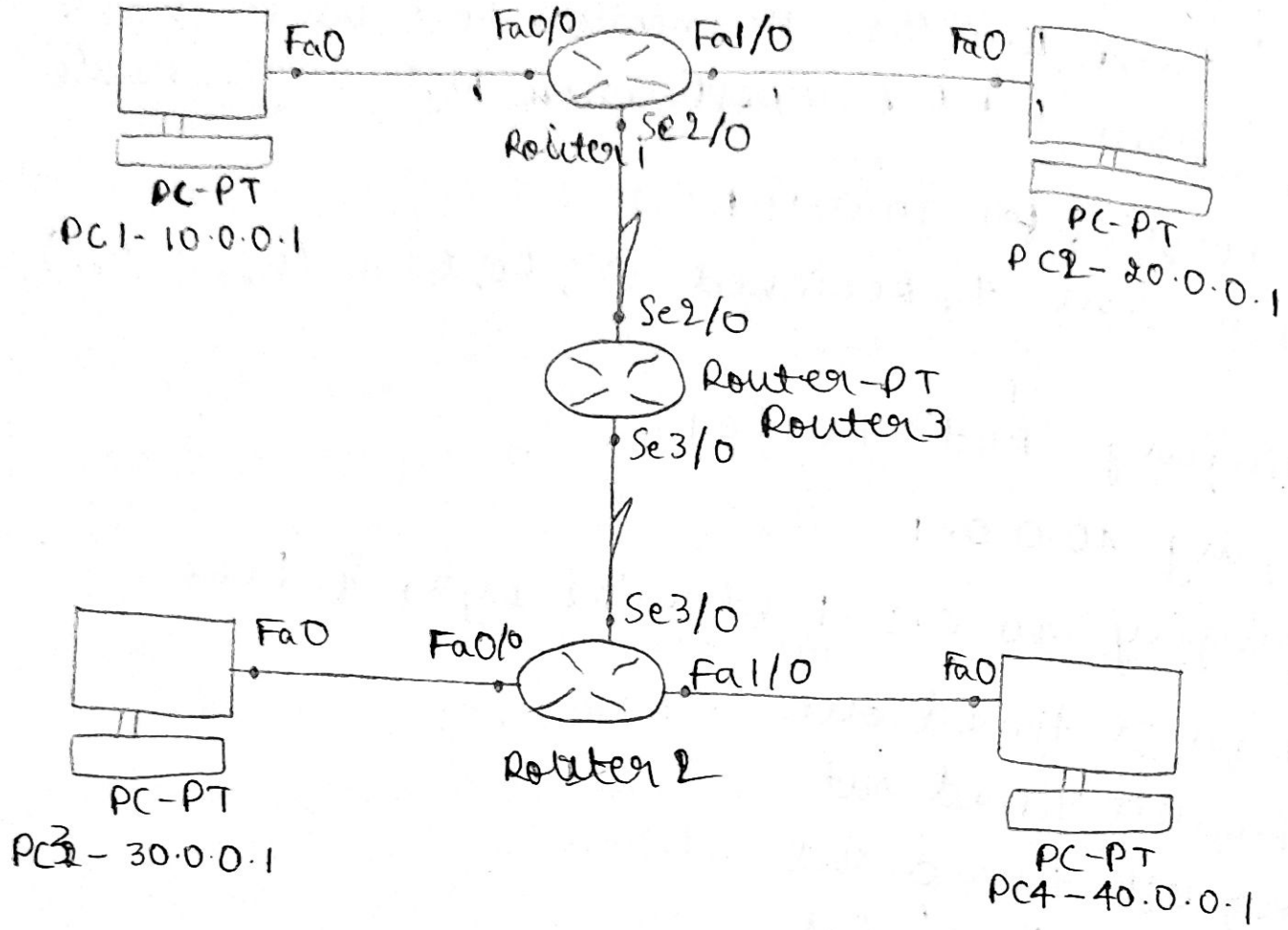
S3: Pinging PC2 on PC1 (same network)

PC> ping 20.0.0.1 with 32 bytes of data:

Reply from 20.0.0.1: bytes=32 time=0ms TTL=127
Reply from 20.0.0.1: bytes=32 time=0ms TTL=127
Reply from 20.0.0.1: bytes=32 time=0ms TTL=127
Reply from 20.0.0.1: bytes=32 time=0ms TTL=127

Ping statistics for 20.0.0.1:
Packets: Sent=4, Recieved=4, Lost=0 (0% loss)

Topology :



Aim
Topology
Procedure
Results : Ping O/P
Observation : learning
and screenshot.

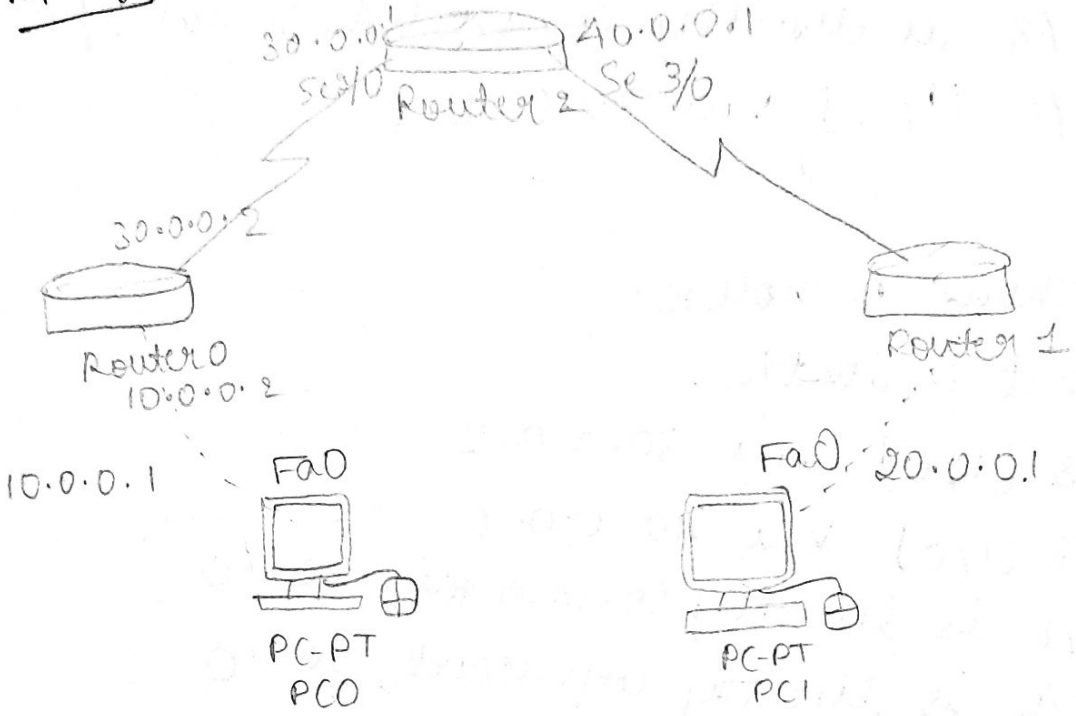
26/6/23

Lab3 30/6/23

Default Routing and static Routing:

Aim: To configure default router, static router to the router

Topology:



Procedure:

Step 1: Drag and drop 2 PC's and 3 router to the workspace and connect them as shown above.

Step 2: Set IP address of 1st PC as 10.0.0.1 and 2nd PC as 20.0.0.1. Also set gateway of 2 PC as 10.0.0.2 & 20.0.0.2 respectively.

Step 3: Place different n/w IP address as 30.0.0.1 and 40.0.0.1 to left and right of Router-2 and start configuring the router interface for Router 0.

Router 0

Router # show IP route.

C - connected S - static * - candidate default

Gateway of last route is 30.0.0.1 to network 0.0.0.0

C 10.0.0.0/8 is directly connected, Fast ethernet

C 30.0.0.0/8 is directly connected, serial 2/0

S* 0.0.0.0/0 [1/0] via 30.0.0.1

Router 2

Router # show ip route

C-connected S-static

S 10.0.0.0/8 [1/0] via 30.0.0.2

S 20.0.0.0/8 [1/0] via 40.0.0.2

C 30.0.0.0/8 is directly connected, se 2/0

C 40.0.0.0/8 is directly connected, se 2/0

Ping operation:

Ping 20.0.0.1

Pinging 20.0.0.1 with 32 bytes of data

Reply from 20.0.0.1 : bytes=32 : time=2ms TTL=125

Reply from 20.0.0.1 : bytes=32 time=4ms TTL=125

Reply from 20.0.0.1 : bytes=32 time=17ms TTL=125

Reply from 20.0.0.1 : bytes=32 time=25ms TTL=125

Ping statistics for 20.0.0.1

Packets: sent 4, Recieved 4, Lost = 0

Approx round trip time in ms

min = 2ms max = 25ms Avg = 12ms.

Observation:

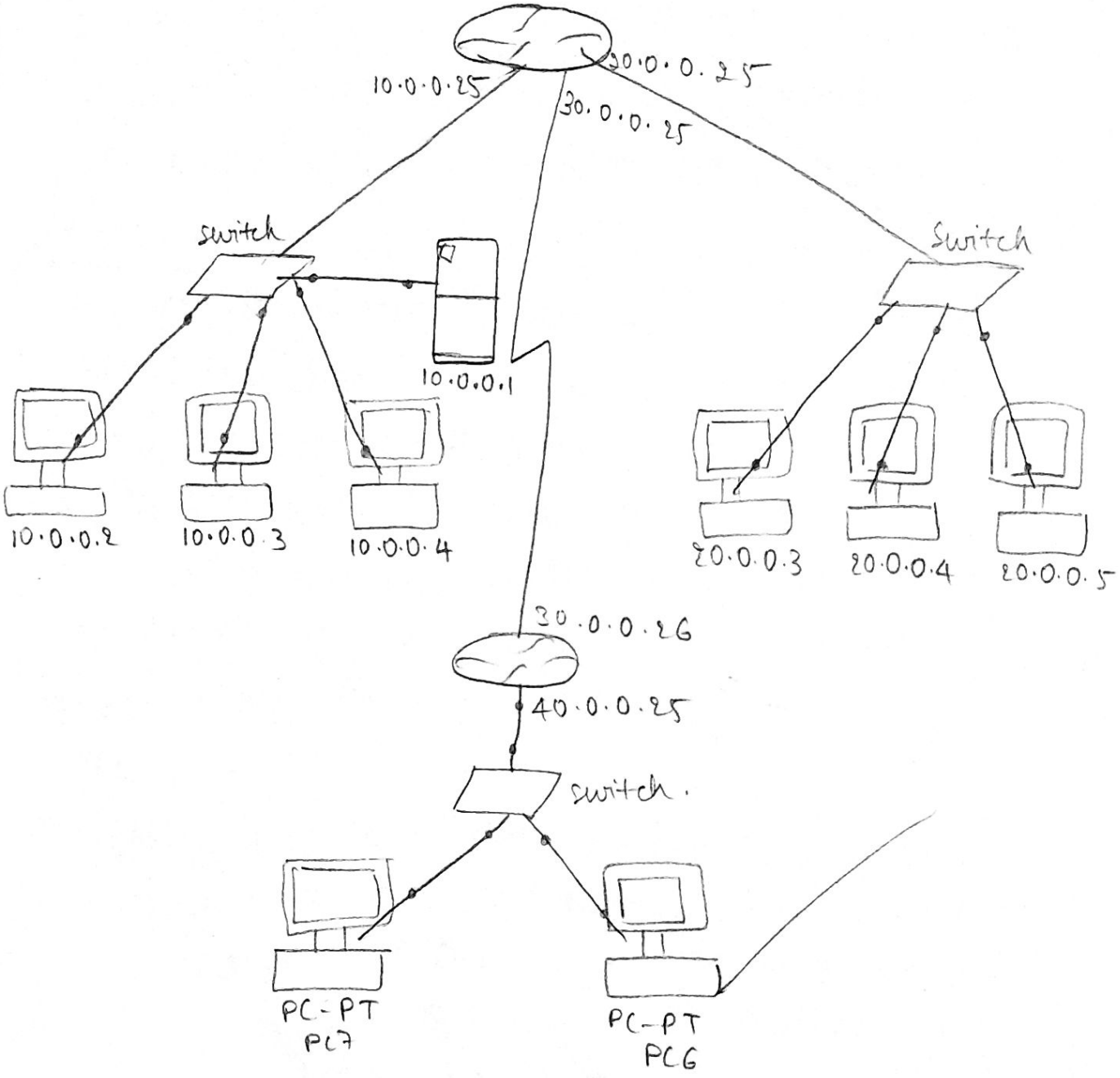
Default route is done in routers, since there is only 1 path available for it to transfer the packets

2017

Lab 4

Dynamic Host Configuration Protocol

Topology:



Procedure:

1] Connect 3 PCs and a server to a switch. Connect another 3 PCs to a switch. Connect the 2 switches to a router.

2] Connect this router to another router. Let that router be connected to a switch which is connected to 2 PCs.

3] Set server IP address as 10.0.0.1 & its gateway as 10.0.0.25.

4] Perform static routing on the routers.

5] Set ip helper address of Router 1 as 10.0.0.1

6] Set ip routes for router2.

commands:

Router 1 : Set IP address for respective interface.

```
Router(config)# interface serial 2/0
Router(config-if)# ip address 30.0.0.25 255.0.0.0
Router(config-if)# no shut.
Router(config-if)# exit
Router(config)# ip route 40.0.0.0 255.0.0.0 30.0.0.26
```

Router 2: Set IP address for respective interface.

```
Router#config t.
Router(config)# interface fastethernet 0/0
Router(config-if)# ip helper-address 10.0.0.1
Router(config-if)# no shut.
Router(config)# ip route 10.0.0.0 255.0.0.0 30.0.0.
Router(config)# ip route 20.0.0.0 255.0.0.0 30.0.0.
```

Observation: The Dynamic allocation of IP address can be done across 1 or more LAN using a server.

Lab 5

21/7/2023

1) DNS

step 1: Create a topology with 1 PC, 1 generic server and 1 switch.

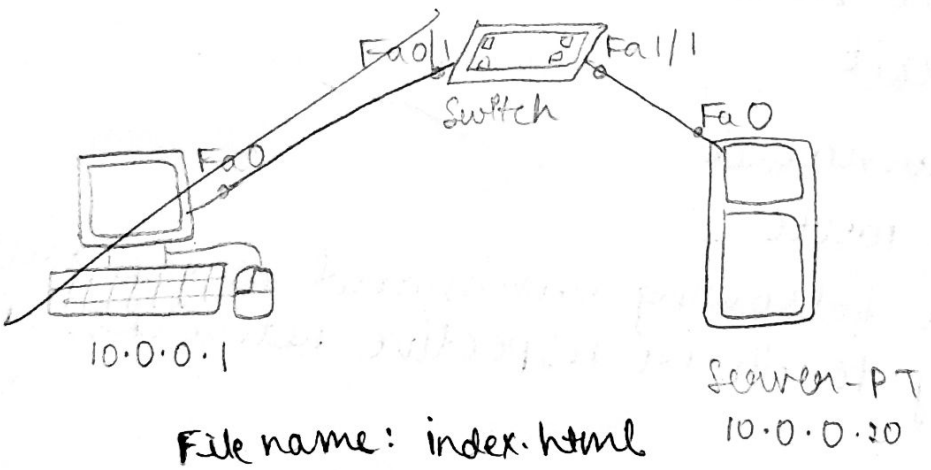
step 2: Configure the PC → IP address, gateway
Configure the server → IP address 10.0.0.20

step 3: Open web browser in PC and input 100.0.20.

step 4: Go to DNS in server. Give the name and URL as 10.0.0.20

step 5: Go to web browser in PC and put the name and see if the page is coming.

step 6: In server, go to <index.html> and input data/make changes.



File name: index.html

Output: URL: http://hello

Cisco Packet Tracer

Welcome to Cisco Packet Tracer

Student details:

Name: Pragnya B S

USN: 1BM21CS132.

2) RIP - Routing Information Protocol.

Step1: Create a network with 2 PCs and 3 routers.

Step2: Configure the PCs and Routers

Step3: Input the following commands in routers.

```
Router> enable
Router# configt
Router(config) interface FastEthernet 0/0
Router(config-if) ip address 10.0.0.10 255.0.0.0
Router(config-if) no shut.
```

At nodes 2, 3, 4, 5 put

```
Router(config-if) encapsulation ppp
```

At nodes with clock symbol i.e., 2, 4

```
Router(config-if) clock rate 64000
Router(config-if) no shut
Router(config-if) exit.
```

Step 4: Input command

```
Router# show ip route
```

Step 5: Input the following command in the routers according to their respective networks.

Eg: For R1.

```
Router(config)#router rip
Router(config-router)#network 10.0.0.0
Router(config-router)#network 20.0.0.0
Router(config-router)#exit.
```

Output:

Pinging PC2 on PC1

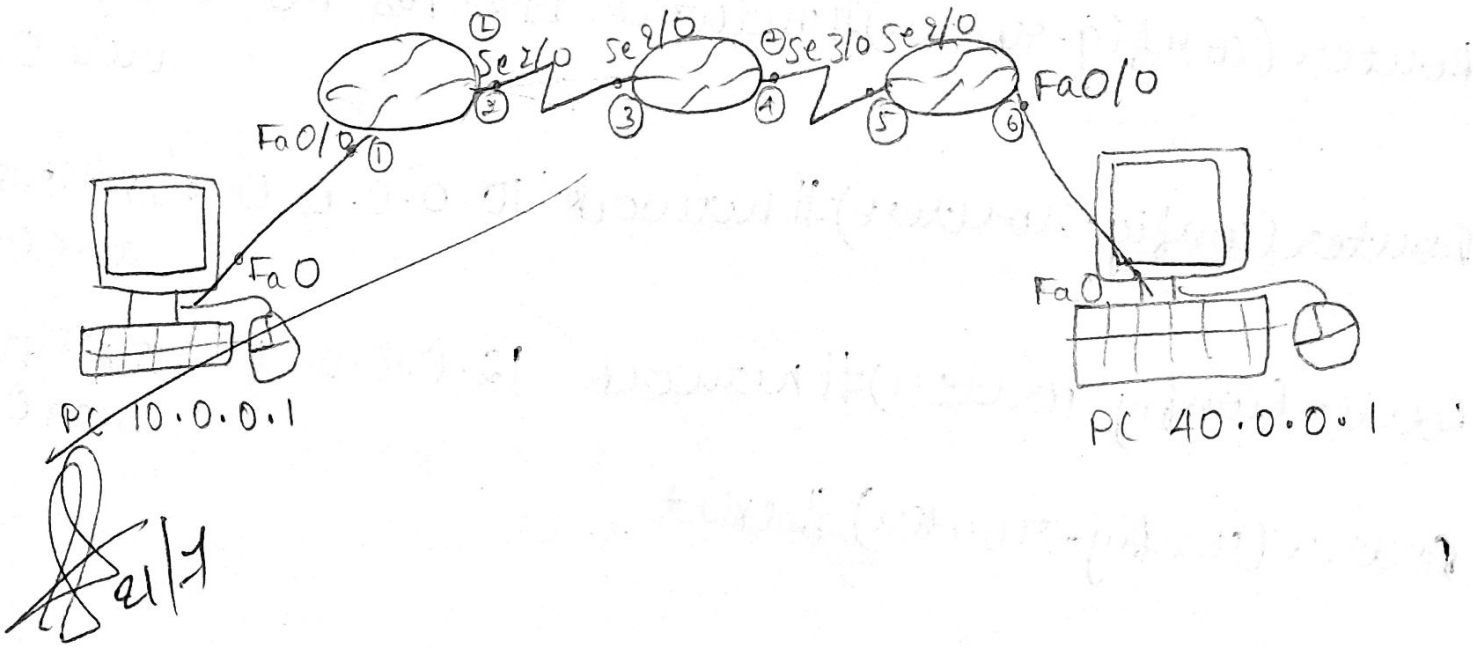
PC>ping 40.0.0.1

Pinging 40.0.0.1 with 32 bytes of data:

Reply from 40.0.0.1 : bytes=32 time=3ms TTL=125
Reply from 40.0.0.1 : bytes=32 time=5ms TTL=125
Reply from 40.0.0.1 : bytes=32 time=8ms TTL=125
Reply from 40.0.0.1 : bytes=32 time=12ms TTL=125

Ping statistics for 40.0.0.1:
Packets: Sent=4, Recieved=4, Lost=0 (0% Loss),
Approximate round trip times in milli-seconds:
Minimum=3ms, Maximum=12ms, Average=7ms

Topology:

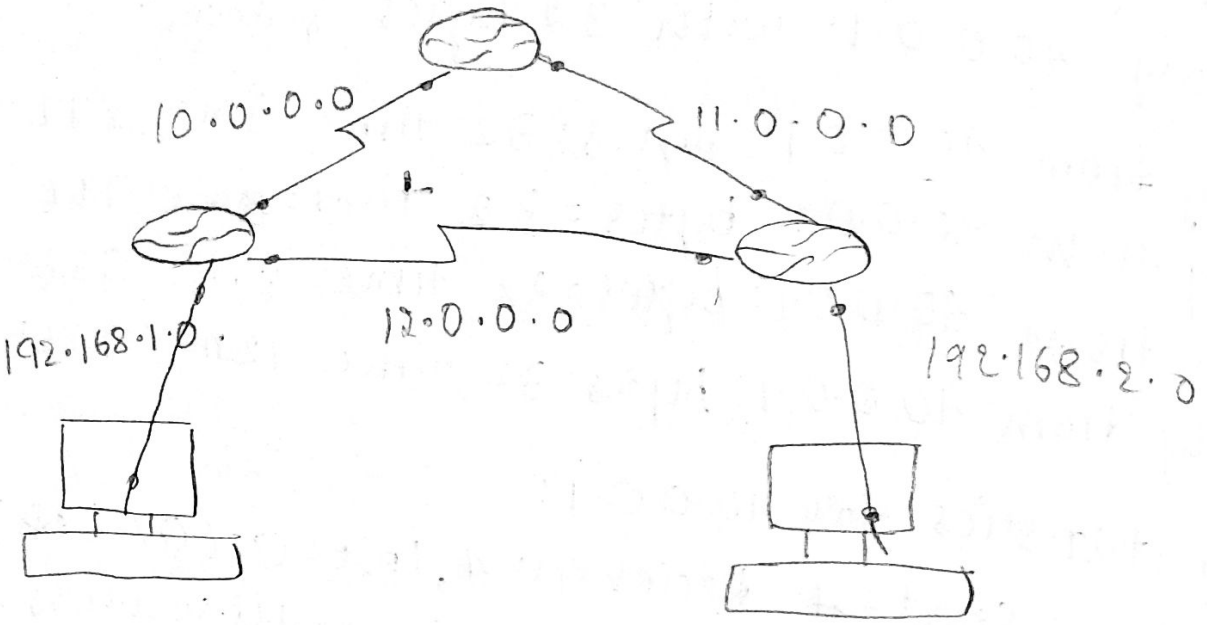


21/1

OSPF (Open Shortest Path First):

7/3/23

1. Topology:



Router0:

Router (config-if)#exit

Router (config)# router ospf 1

Router (config-router)#network 192.168.1.0 0.0.0.255 area 0

Router (config-router)#network 10.0.0.0 0.255.255.255 area 0

Router (config-router)#network 12.0.0.0 0.255.255.255 area 0

Router (config-router)#exit

Router1:

Router(config)#router ospf 1

Router(config-router)#network 10.0.0.0 0.255.255.255 area 0

Router(config-router)#network 11.0.0.0 0.255.255.255 area 0

exit.

Router 2:

Router(config)#router ospf 1

Router(config-router)#network 192.168.2.0 0.0.0.255 area 0

Router(config-router)#network 11.0.0.0 0.255.255.255 area 0

Router(config-router)#network 12.0.0.0 0.255.255.255 area 0.

Output:

PC> ping 192.168.2.1

Pinging 192.168.2.1 with 32 bytes of data:

Request timed out:

~~Reply from 192.168.2.1 with 32 bytes of data:~~

Reply from 192.168.2.1 : bytes=32 time=5ms TTL=126

Reply from 192.168.2.1 : bytes=32 time=7ms, TTL=126

Reply from 192.168.2.1 : bytes=32 time=6ms TTL=126

Ping statistics for 192.168.2.1:

Packets: sent=4, Recieved=3, Lost=1 (25% loss)

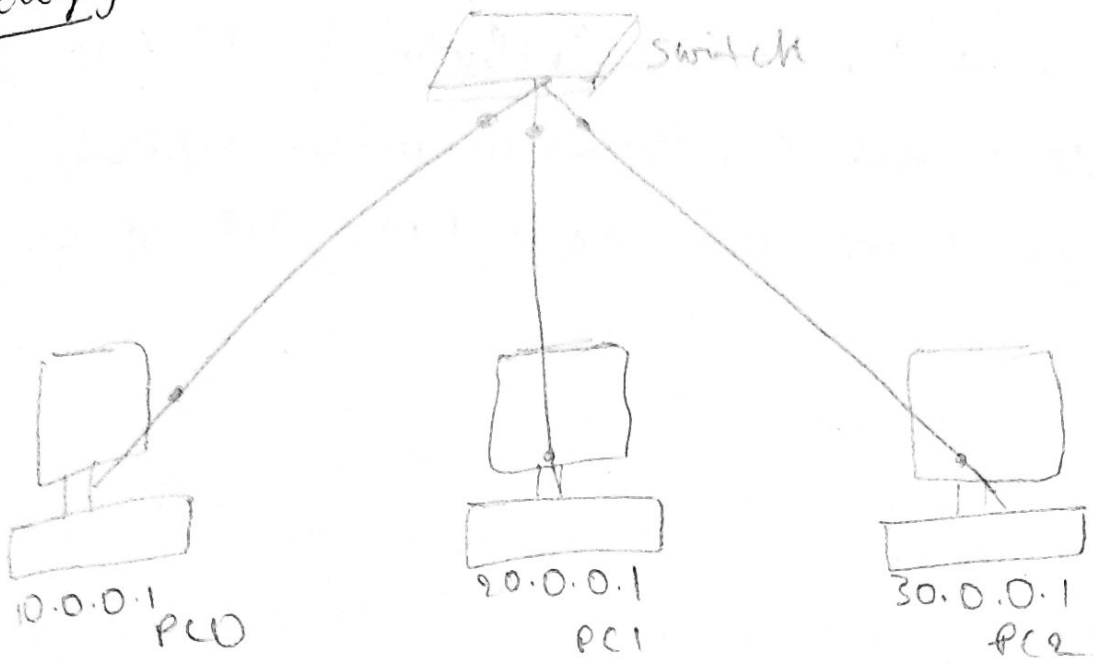
~~Approxi~~

Approximate round trip times in milli-seconds:

Minimum = 5ms, Maximum = 7ms, Average = 6ms.

ARP (Address Resolution Protocol):

Topology:



Steps:

* commands used:

1. arp -a → to check arp table (from PC command prompt)

2. arp -d → to clear arp.

click capture forward.

Output:

PC> arp -d

PC> ping 10.0.0.2

Pinging 10.0.0.2 with 32 bytes of data

Reply from 10.0.0.2: bytes=32 time=0ms TTL=120

Reply from 10.0.0.2: bytes=32 time=0ms TTL=120

Reply from 10.0.0.2: bytes=32 time=0ms TTL=120

Reply from 10.0.0.2. bytes=32 time=0ms TTL=128

Ping statistics from 10.0.0.2:

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

Approximate round trip times in milli-seconds:

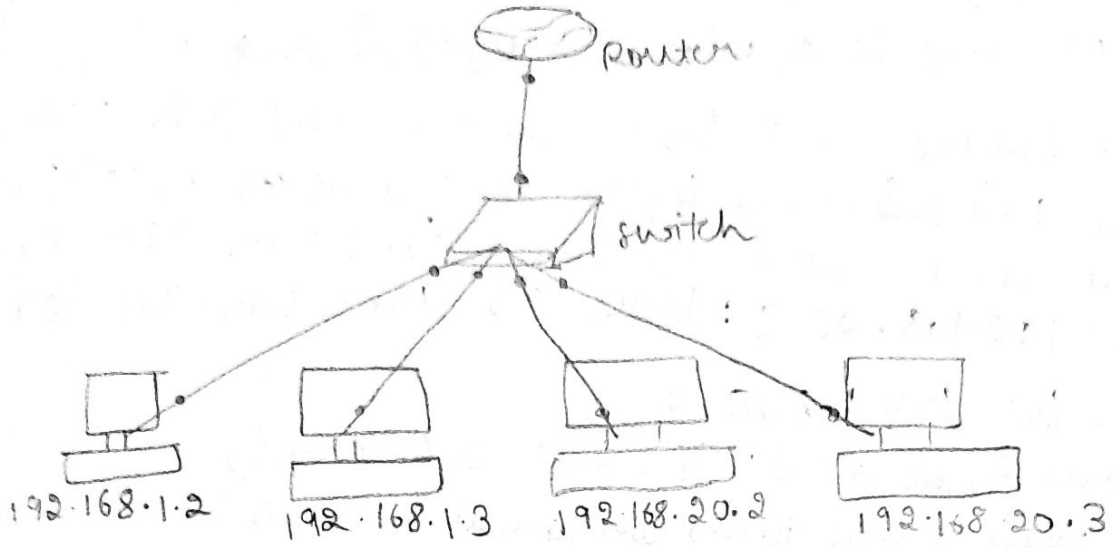
Minimum=0ms, Maximum=0ms, Average=0ms

S 1/8

VLAN:

11/8/23

Topology:



Procedure:

→ In switch ⇒ VLAN Database ⇒ configure VLAN number & VLAN name

FastEthernet 0/5 ⇒ In dropdown, select TRUNK

FastEthernet 0/4 ⇒ configure VLAN

FastEthernet 0/3 ⇒ configure VLAN

→ In Router

⇒ VLAN Database ⇒ configure VLAN number & VLAN name

CLI of router:

```
config t
interface fastEthernet 0/0.1
encapsulation dot1q 20
ip address 192.168.20.1 255.255.255.0
no shut
exit
```

→ Ping the device

Result:

In PC0,

ping 192.168.20.2

Pinging 192.168.20.2 with 32 bytes of data:

Reply from 192.168.20.2 bytes = 32 time=0ms TTL=127
Reply from 192.168.20.2: bytes = 32 time=0ms TTL=127
Reply from 192.168.20.2: bytes = 32 time=0ms TTL=127
Reply from 192.168.20.2: bytes = 32 time=0ms TTL=127

Ping statistics for 192.168.20.2:

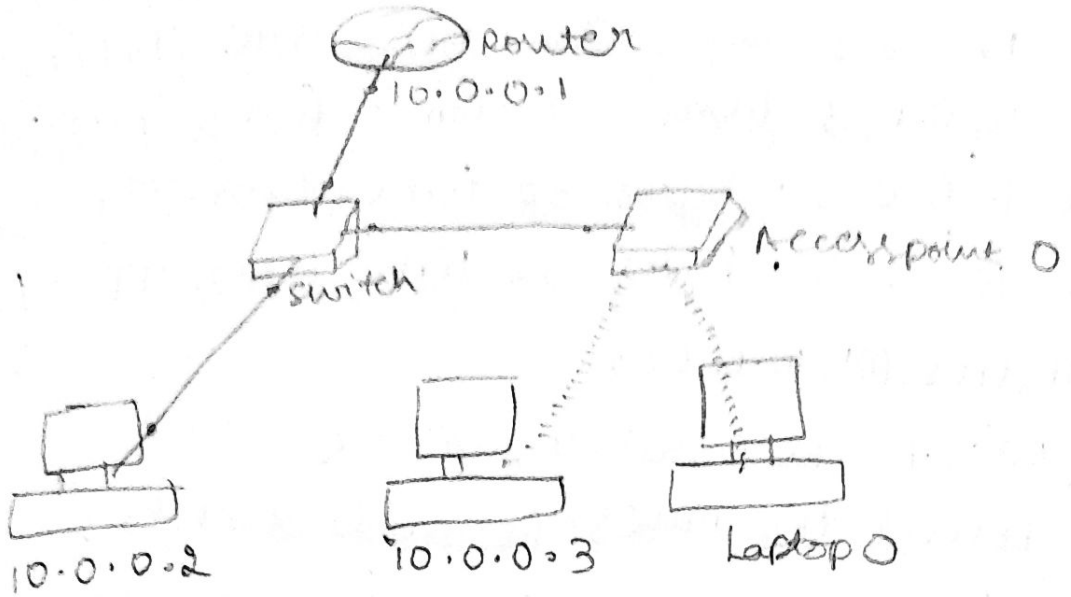
Packets: sent=4, Recieved = 4, Lost=0 (0% loss)

Approximate round trip times in milliseconds:

Minimum = 0ms, Maximum = 0ms, Average = 0ms

WLAN:

Topology:



Procedure:

→ Construct the topology

→ Set the ip address and gateway of PC's

→ for Accesspoint 0,

=> In port 1

SSID => WLAN

WEN key => 1234567890

→ for PC1

switch off. Drag the existing PT-HOST-NM-1AM to the component list and drag WMP300N to empty port. Then switch on the device.

In wireless 0,

SSID => WLAN

WEN key => 1234567890

IP address => 10.0.0.3

Gateway => 10.0.0.1

→ Repeat the same to laptop

→ Ping the device.

Result:

Ping 10.0.0.3

Pinging 10.0.0.3 with 32 bytes of data:

Reply from 10.0.0.3: bytes=32 time=15ms TTL=128

Reply from 10.0.0.3: bytes=32 time=6 ms TTL=128

Reply from 10.0.0.3: bytes=32 time=9 ms TTL=128

Reply from 10.0.0.3: bytes=32 time=11ms TTL=128

Ping statistics for 10.0.0.3:

Packets: sent=4, Recieved=4, Lost=0

Approximate round trip times in milli-seconds:

Minimum = 6ms, Maximum=15ms, Average=10ms

Telnet

18/8/23

Topology:

Procedure:

* Configure the ip address for PC and router.
* In Router,

```
Router(config)#hostname r1
r1(config)#~~interf~~ enable secret p1
r1(config)#interface fastEthernet 0/0
r1(config-if)#ip address 10.0.0.1 255.0.0.0
r1(config-if)#no shut
r1(config-if)#line vty 0 5
r1(config-line)#login
r1(config-line)#password p0
r1(config)#exit
r1#wr
r1#show ip route.
```

Output:

In PC, ping the router.

PC> ping 10.0.0.1

Pinging 10.0.0.1 with 32 bytes of data:

Reply from 10.0.0.1: bytes=32 time=1ms TTL=255
Reply from 10.0.0.1: bytes=32 time=0ms TTL=255
Reply from 10.0.0.1: bytes=32 time=0ms TTL=255
Reply from 10.0.0.1: bytes=32 time=0ms TTL=255

Ping statistics for 10.0.0.1:
Packets: Sent=4, Recieved=4, Lost=0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum=0ms, Maximum=1ms, Average=0ms

PC> telnet 10.0.0.1
Trying 10.0.0.1 .... Open
User Access Verification
Password: P0
R1> enable
Password: P1
R1#
[connection to 10.0.0.1 closed by foreign host]

18/8

R1# show ip route.
Codes: C-connected, S-static, I-IGRP, R-RIP, M-mobile, B-BGP
... P-periodic downloaded static route.
Gateway of last resort is not set
C 10.0.0.0/8 is directly connected, FastEthernet0/0
R1#

Write a program for error detecting code using CRC-CCITT(16-bits):

```
#include<stdio.h>
char m[50], g[50], r[50], q[50], temp[50];
void caltrans(int);
void crc(int);
void calxram();
void shiftl();
int main()
{ int n,i=0;
char ch, flag=0;
printf("Enter the frame bits:");
while((ch=getc(stdin))!='\n')
m[i++]=ch;
n=i;
for(i=0; i<16; i++)
m[n++]='0';
m[n]='\0';
printf("message after appending 16 zeros: %s",m);
for(i=0; i<=16; i++)
g[i]='0';
g[0]=g[4]=g[11]=g[16]='1'; g[17]='\0';
printf("\n generator: %s\n", g);
crc(n);
printf("\n quotient: %s", q);
caltrans(n);
printf("\n transmitted frame: %s", m);
```

```
printf("Enter transmitted frame:");
scanf("\n%s",m);
printf("CRC checking\n");
crc(n);
printf("\n\nlast remainder: %s",r);
for(i=0;i<16;i++)
if(r[i]!='0')
flag=1;
else
continue;
if (flag==1)
printf("error during transmission");
else
printf("\n\nReceived frame is correct");
}
void crc(int n)
{ int i,j;
for(i=0;i<n;i++)
temp[i]=m[i];
for(i=0; i<16;i++)
r[i]=m[i];
for(i=0;i<n-16;i++)
{ if(r[0]=='1')
```

```
{ q[i]='1';
  calcram();
}else
{ q[i]='0';
  shiftl();
}
r[16]=m[17+i];
r[17]='10';
for(j=0;j<=17;j++)
temp[j]=r[j];
}
q[n-16]='10';
}
void calcram()
{ int i,j;
for(i=1;i<=16;i++)
r[i-1]=((int)temp[i]-48)^((int)q[i]-48)+48;
}
void shiftl()
{ int i;
for(i=1;i<=16;i++)
r[i-1]=r[i];
}
```

```
void caltrans(int n)
{ int i,k=0;
  for(i=n-16; i<n; i++)
  m[i]=((int)m[i]-48)^((int)g[k++]-48)+48;
  m[i]='\0';
}
```

Output:

Enter the frame bits: 1011

Message after appending 16 zeroes: 1011000000000000

generator: 10001000000100001

intermediate remainder

remainder 1: 0111000000100010

remainder 2: 1110000001000100

remainder 3: 1101000010100101O

remainder 4: 1011000101101011

quotient: 1011

transmitted frame: 10111011000101101011

Enter transmitted frame: 10111011000101101011

last remainder: 0000000000000000

Received frame is correct.

Write the algorithm & Explain the working.

18/8

Write a program for congestion control using leaky bucket algorithm:

```
#include<stdio.h>
int main()
{ int incoming, outgoing, buck_size, n, store=0;
 printf("Enter bucket size, outgoing rate & no. of inputs: ");
 scanf("%d %d %d", &buck_size, &outgoing, &n);
 while (n!=0)
 { printf("Enter the incoming packet size:");
  scanf("%d", &incoming);
  printf("Incoming packet size %d\n", incoming);
 if (incoming <= buck_size-store))
 { store += incoming;
 printf("Bucket buffer size %d out of %d\n", store,
                                        buck_size);
 }else
 { printf("Dropped %d no of packets\n",
            incoming - (buck_size-store));
   printf("Bucket buffer size %d out of %d", store,
                                        buck_size);
   store = buck_size;
 }
 store = store - outgoing;
 printf("After outgoing %d packets left out of %d
        in buffer\n", store, buck_size);
   n--;
}
}
```

Output:

Enter bucket size, outing rate & no of inputs: 20 10 2
Enter the incoming packet size: 30
Incoming packet size: 30
Dropped 10 no of packets
Buffer size 0 out of 20
After outgoing 10 packets left out of 20 in buffer
Enter the incoming packet size: 10
Incoming packet size: 10
Buffer size 20 out of 20
After outgoing 10 packets out of 20 in buffer

Write the algorithm & explain its working

18/8

CRC-CCITT:

Algorithm:

1) Start
2) Enter the message to be transmitted
3) Append the message with 16 0's (i.e., if you input 5 digit message, the appended message should be 21-bits)
4) XOR appended message and transmit it (Here, you compare with an already existing string such as 10001000000100001 and replace the bits the same way XOR operation works)
5) Verify the message that is received is the same as the one sent.
6) End.

Leaky bucket algorithm:

1) Start
2) Set the bucket size or the buffer size
3) Set the output rate
4) Transmit the packets such that there is no overflow.
5) Repeat the process of transmission until all packets are transmitted
6) Stop.

1) Using TCP/IP sockets, write a client-server program 11/9/23
to make client sending the file name and the server
to send back the contents of the requested file
if present.

* Create a file server.py.

Code:
```
from socket import *
serverName = "127.0.0.1"
serverPort = 12000
serverSocket = socket(AF_INET, SOCK_STREAM)
serverSocket.bind((serverName, serverPort))
serverSocket.listen(1)
while 1:
    print("The server is ready to receive")
    connectionSocket, addr = serverSocket.accept()
    sentence = connectionSocket.recv(1024).decode()
    file = open(sentence, "r")
    l = file.read(1024)
    connectionSocket.send(l.encode())
    print('\n Sent contents of ' + sentence)
    file.close()
    connectionSocket.close()
```

Output: The server is ready to recieve.

* Create a file client.py

code:
```
from socket import *
serverName = '127.0.0.1'
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_STREAM)
clientSocket.connect((serverName, serverPort))
sentence = input("Enter file name:")
clientSocket.send(sentence.encode())
fileContents = clientSocket.recv(1024).decode()
print("\nFromServer:\n")
print(fileContents)
clientSocket.close()
```

Output: Enter filename: server.py

From server:

(The server code is recieved)

Output: The server is ready to receive

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

2) Using UDP sockets, write a client server program to make client sending the file name and the server to send back the contents of the requested file if present.

* Create a file ServerUDP.py

```
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")
 con = file.read(2048)
 serverSocket.sendto(bytes(con, "utf-8"), clientAddress)
 print("\nSent contents of', end = ' ')
 print(sentence)
 # for i in sentence:
    # print (str(i), end="")
    file.close()
```

Output:

The server is ready to receive.

* Create a file clientUDP.py

```
from socket import *
serverName = "127.0.0.1"
serverPort=12000
clientSocket=socket(AF_INET, SOCK_DGRAM)
sentence = input("\nEnter file name: ")
clientSocket.sendto(bytes(sentence,"utf-8"),
                    (serverName, serverPort))

fileContents, serverAddress = clientSocket.recvfrom
                                (2048)
print("\nReply from Server.\n")
print(fileContents.decode("utf-8"))
#for i in fileContents:
    #print(str(i), end="")

clientSocket.close()
clientSocket.close()
```

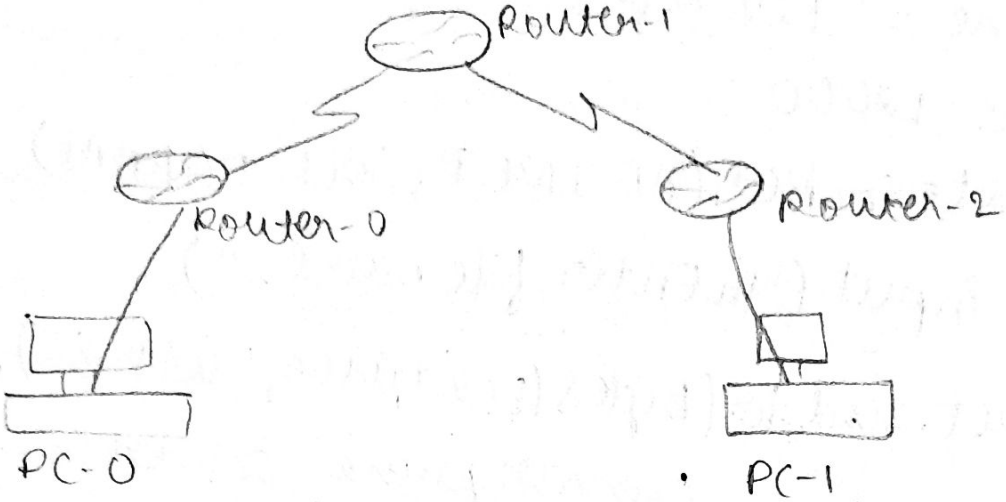
Output:

```
Enter file name: ServerUDP.py
Reply from server:
(The serverUDP code is received)
```

Output: The server is ready to receive
Sent contents of ServerUDP.py
The server is ready to receive.

Demonstration of TTL / Life of a Packet:

Topology:



Procedure:

* Create topology as shown above with 2 PCs & 3 Routers
* Configure the devices as per static/default/dynamic routing
* In simulation mode, send a simple PDU from 1 PC to another.
* Use button capture to capture every transfer.
* Click on PDU during every transfer to see ~~Inbound~~ Inbound & outbound PDU details.

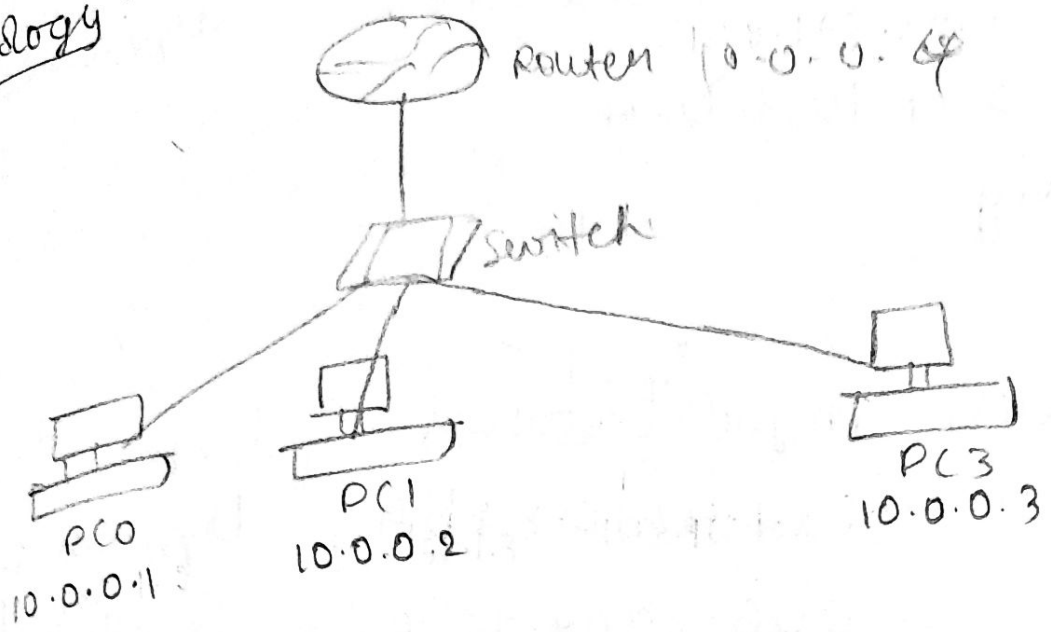
Observe that there is a difference of 1 in TTL when it crosses every router.

1/9

LAB

ARP (with Router)

Topology:



Procedure:

1) arp -a
2) arp -d

click capture forward.

Output:

In PC0

arp -a

No ARP ~~entities~~ entries found

Ping 10.0.0.2

Pinging 10.0.0.2 with 32 bytes of data

Reply from 10.0.0.2 bytes=32 time=12ms TTL=126

Reply from 10.0.0.2 bytes=32 time=12ms TTL=126

Reply from 10.0.0.2 bytes=32 time=12ms TTL=126

Reply from 10.0.0.2 bytes=32 time=12ms TTL=126

arp -a

Internet address	Physical address	type
10.0.0.2	001.432@.C7C9	dynamic

Ping 10.0.0.3 & 10.0.0.4

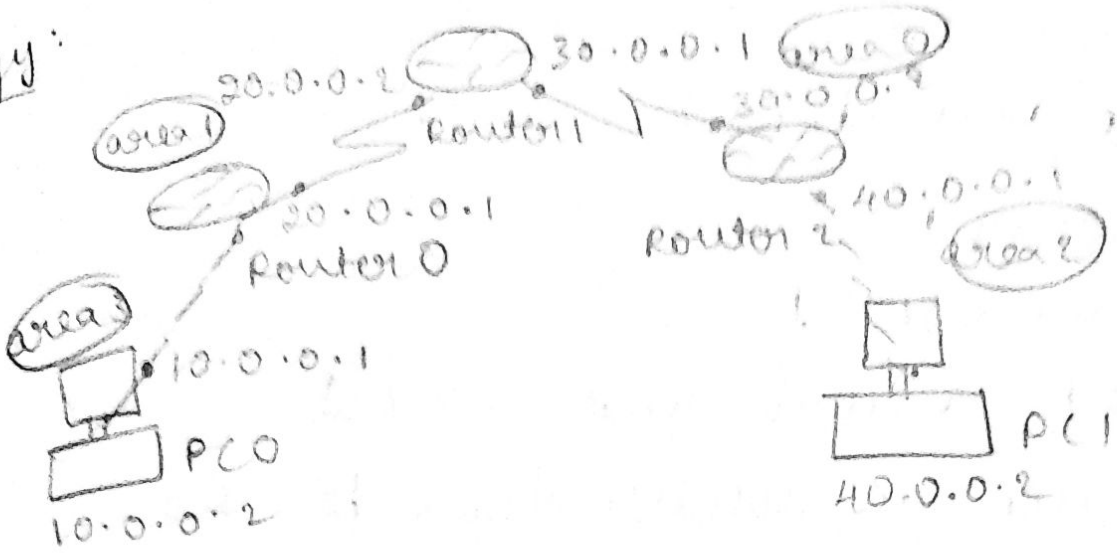
After Pinging

arp -a

Internet address	Physical address	type
10.0.0.2	001.429e.E2b4	Dynamic
10.0.0.3	006.2ab4.9e7e	Dynamic
10.0.0.4	006.2a67.4029	Dynamic

OSPF using Area:

Topology:



Procedure:

In Router R1

router OSPf 1

router -id 1.1.1.1

network 10.0.0.0 0.255.255.255 area 3

network 20.0.0.0 0.255.255.255 area 1

exit.

In router 2 & router 3

Repeat the same by using approachable areas.

Configure loopback address to routers.

R1 => enter loopback 0

ip add 172.16.1.253 255.255.255.0.0

no shut.

R2 => interface loopback 0

ip add 172.16.1.254 255.255.0.0

no shut.

R3 => interface loopback 0
ip add 172.16.1.254 255.255.0.0
no shut

→ Create virtual link

In router 1
router OSPF 1
area 1 virtual-link 2.2.2.2

→ Now check the router table for R3
show ip route

→ Now create virtual link between router 1 and router 2. By this we create virtual link to connect area 3 to area 0.

→ R2 & R3 get updated abt check routing table for R3

show ip route.
coder - C - connected O - OSPF
Gateway of last resort is not set.
OIA 20.0.0.0/8 via 30.0.0.1, 00.01:56,
C 40.0.0.0/8 is directly connected,
fastEthernet 0/0

→ Now ping 10.0.0.2 to 40.0.0.2

4/9