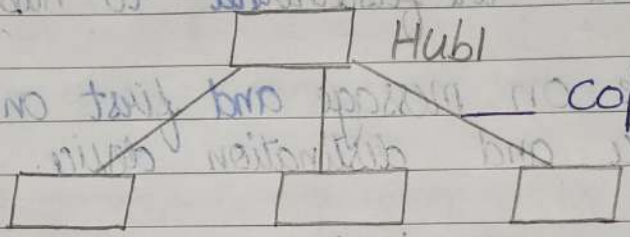
Experiment : 1

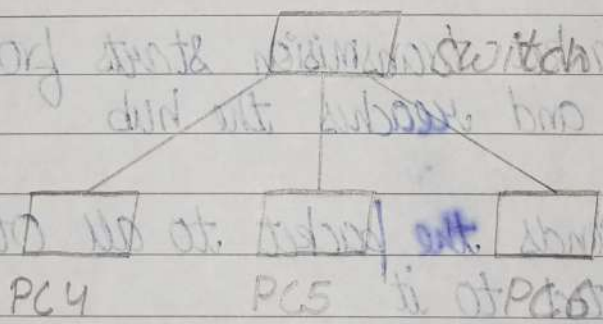
Aim: Create a topology and simulate sending a simple PDU from source to destination using hub and switch as connecting devices and demonstrate ping message

Topology:

Hub:

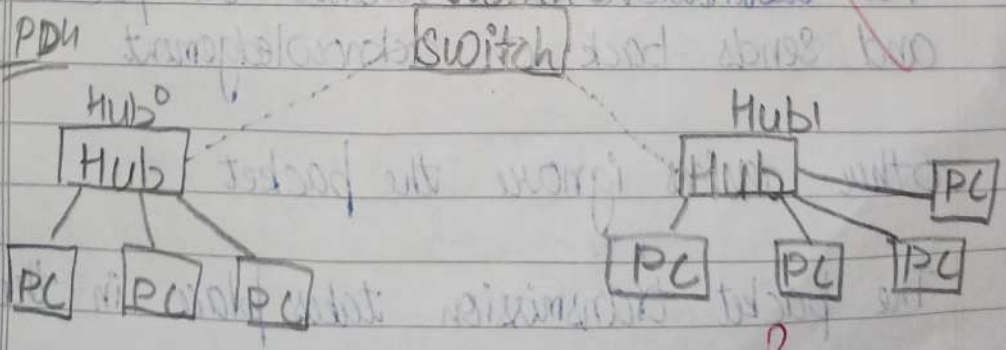


Switch:



combination Hub and Switch

*Simple PDU



Procedure: (Hub)

1. Select the end devices and change their IP addresses.
2. Select the hub as the connecting devices.
3. Select copper straight-through as the connecting wire between end devices and hub
4. Connect the fastethernet to hub ports.
5. Click on message and first on source device and distination device.
6. Observe the packet transmission and acknowledgement receiving procedure

Observation: (Hub)

1. The packet transmision starts from source device and reaches the hub
2. Hub sends the packet to all other devices connected to it
3. The distination device receives the packets and sends back on acknowledgement
4. other devices ignore the packet
5. The packet transmission takes place in the

above scen

Result
PC > ping
pinging
Reply fro
Reply fro
Reply fro
Reply fro

ping s

Packet

Approxi
minimu

above scenario everytime

Result

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=128

Reply from 10.0.0.2 bytes=32 times=0ms TTL=128

Reply from 10.0.0.2 bytes=32 times=0ms TTL=128

Reply from 10.0.0.2 bytes=32 times=0ms TTL=128

ping statistics for 10.0.0.2

Packets Sent=4, Recived:4, lost=0

Approximate round trip times in milli-seconds

Minimum=0 ms, Maximum: 0ms, Average: 2ms

Switch

Procedure:

* Select the end devices and changes their IP address
* Select switch as the connecting devices
* Select copper straight through as connection wire both the end devices and switches
* Connect fastethernet to switch ports
* Add simple PDU from source to destination device
* ping a PDU using command prompt in one device

Observation:

* The packet transmission starts from source device and reaches the switch
* Switch sends the packet to all devices connects to it
* The destinated devices recives the packet and sends an acknowledgment back to switch stating it has received the packet
* Switch remembers the device sending the acknowledgment and only communicates with that device

for further transmission.

* Other devices do not receive the packet from next transmission.

Result

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 = 2ms TTL = 128
Reply from 10.0.0.1 bytes = 32 time = 4ms TTL = 128
Reply from 10.0.0.1 bytes = 32 time = 8ms TTL = 128
Reply from 10.0.0.1 bytes = 32 time = 3ms TTL = 128

Ping statistics for 10.0.0.1:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss)
Approximate round trip time in Mill-seconds:
Minimum = 2ms, Maximum = 8ms, Average = 4ms.

Hub and Switch

Procedure:

- Select the end devices and change their IP address
- Select hubs for end devices and for connecting devices Select switch as connecting devices of hubs
- Select copper straight through as connection wire both the end devices and hubs
* Select copper as connection wire between fastethernets to ports
- connect fastethernet to ports.

- Add simple PDU form source to destination device
- Ping a PDU using command prompt in one device.

observation :

- The packet transmission start from source device and reaches the hub
- Hub sends packet to switch and from switch it sends to another hub.
- Hub sends the packet to all other devices connected to it.
- The destination device recives the packet and sends back an acknowledgement stating it has received the packet.
- other device ignore the packet
- The packet transmission takes place in the above scenario everytime

8/10

NL

22/6/23

Result:

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 = 9ms TTL = 128
Reply from 10.0.0.1 bytes = 32 time = 19ms TTL = 128
Reply from 10.0.0.1 bytes = 32 time = 4ms TTL = 128
Reply from 10.0.0.1 bytes = 32 time = 0ms TTL = 128

Ping statistics for 10.0.0.1:

Packet Sent = 4 Received = 4 Lost = 4 (0% loss)
Approximate round trip times in milli-second
Minimum = 0ms, Maximum = 19ms, Average = 8ms.

8/10

NL
22/6/23

PAGE NO.
DATE 22/06/23

Experiment - 2

Configure IP address to routers in packet tracer. Explore the following message: ping responses, destination unreachable, request time out. reply

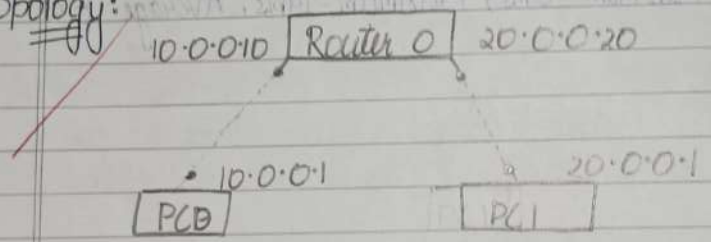
2a.

Aim:

To configure IP address to routers in packet tracer and get ping responses - request time out. reply

* Always in router put gateway

Topology:



Procedure:

- 2 PCs are connected to a router using copper cross over
- IP addresses are set for PCs and router
- IP address for Router is set by giving following commands -

router # enable
router # config

router (config)# interface fastethernet 0/0 ds.
router (config-if) # ip address 10.0.0.10 255.0.00
router (config-if) # no shut
router (config-if) # exit
router (config)# interface fasterethernet 1/0
router (config-if) # ip address 20.0.0.10 255.0.00
router (config-if) # no shut
router (config-if) # exit
after all IPs are set, ping message is sent.

→ observations:

PC 0 is in network 10.0.0.0 and PC1 is in 20.0.0.0 Hence we use router to connect them. when a ping message is sent from PC 10.0.0.1 to 20.0.0.1 the message reaches the destination through router when a message is sent the router captures it and sends to the distination PC which is in another network

Result:

PC > ping 20.0.0.1

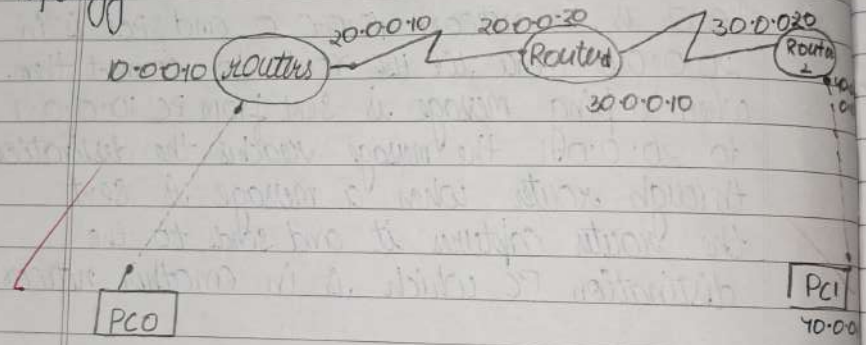
pinging 20.0.0.1 with 32 bytes of data:
Request time out
Reply from 200.0.1 bytes = 32 time = 0ms TTL = 127
Reply from 20.0.0.1 bytes = 32 time = 0ms TTL = 127
Reply from 20.0.0.1 bytes = 32 time = 0ms TTL 127

Ping statistic for 20.0.0.1
packets sent = 4 received = 3 lost = 1 (25% loss),
approximate round trip time in milli-second
minimum = 0ms Maximum = 0ms Average = 0ms

2b To configure IP address to routers in packets traces. explore the following message ping responses, distination unreachable, request timeout, reply.

Aim: To configure IP address to routers in packets trace and get ping responses - request time out, distination host unreached reply.

Topology:



Procedure:

- Connect PC to corresponding router using copper cross-over
- connect routers using serial - DCE
- Set IP address for all PCs
- Configure IP address to routers by giving commands in CLI

- After all IPs are set, ping PC to get destination host unreachable host message
- router the IPs to the adjacent IPs using following commands
for router0 - router(config)# ip route 30.0.0.0 255.0.0.0 20.00x0
router(config)# ip route 40.0.0.0 255.0.0.0 20.0.0.20

for router 1 -
router1:~ router(config)# ip router 10.0.0.0 255.0.0.0 20.0.0.10
. router(config)# ip route 40.0.0.0 255.0.0.0 30.0.0.20

for router2 - router(config)# ip route 10.0.0.0 255.0.0.0 30.0.0.10

router(config)# ip route 20.0.0.0 255.0.0.0 30.0.0.0

After this is done, ping PC to get reply messages.

Observations:

PC0 is in network 10.0.0.0 and PC1 is in network 40.0.0.0 There the 3 routers in between which initally directly connects 10.000, 20.000, 30.0.0.0 and 40.0.0.0 Hence when a ping message is sent from 10.0.0.1 to 40.0.0.1 it doesn't reach the destination Instead it only reaches the first router and gives destination host unreachable message

After letting the routers know about other adjacent network we send a ping message from 40.0.0.1 to 10.0.0.1 to get desired result

The message reaches the destination.

Result:

① PC > ping 40.0.0.1
pining 40.0.0.1 with 32 bytes of data

Reply from 10.0.0.10 destination host unreachable
Reply from 10.0.0.10 destination host unreachable
Reply from 10.0.0.10 destination host unreachable
Reply from 10.0.0.10 destination host unreachable

Ping statics for 40.0.0.1
packets sent = 4 received = 0 lost = 4 (100% loss)

② PC > ping
pining
Reply fr
Reply fr
Reply fr
Reply f
ping
Packet
Approx
minim
Show
in
Router (config)
IP

② PC > ping 10.0.0.1
pinging 10.0.0.1 with 32 bytes of data
Reply from 10.0.0.10 : bytes = 32 time = 16ms
TTL = 128

Reply from 10.0.0.10 bytes = 32 times 1ms TTL = 128
Reply from 10.0.0.10 bytes = 32 times 2ms TTL = 128
Reply from 10.0.0.10 bytes = 32 times 4ms TTL = 128

ping statics for 10.0.0.1:

Packets : sent = 4 Received = 4 loss = 0 (0% loss)
Approximate round trip times in mill-seconds
minimum = 1ms, max = 16ms, average = 6ms.

show ip route?

after ping it will will not readerable
in router 0 will go to show ip.

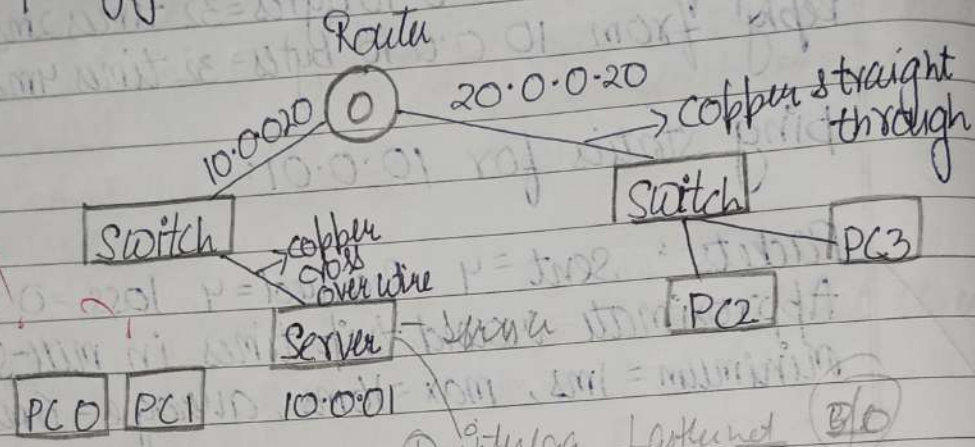
Router (config)#
ip route 30.0.0.0

14/7/23

DHCP

Aim: Connection of server LAN within and outside the network using switches and routers

Topology



Procedure:

1. Select two or more PC and a server connecting to switch and another network with only end devices and switch
2. Connect both switches to router
3. Set IP address of server as 10.0.0.1
4. Now, go to Services < Select DHCP < save the current IP address 20.0.0.2
5. Now, check the IP addresses of other devices in the network in IP configuration in desktop.

6. Now in the CLI of router enable follow steps -

```
>enable
# config t
# interface fastethernet 4/0
# ip address 10.0.0.10 255.0.0.0
# no shut
# exit
# interface fastethernet 0/0
# ip address 20.0.0.20 255.0.0.0
# no shut
# exit
```

7) go to server < config < gateway 10.0.0.20

8) Now in router, we need to set ip address of server

```
# config t
# fast ethernet 0/0
# ip address 10.0.0.1
# no shut
# exit
```

9) Now go to Server < Service < DHCP < add new IP 20.0.0.2

10) To check the connection go to the ip configuration of PC outside the network and click on DHCP and IP gateway will be visible

Result:

From server from PC2 to PC0 whose ip address is 10.0.0.2

PC > ping 10.0.0.2

ping 10.0.0.2 with 32 bytes of data

Request timed out

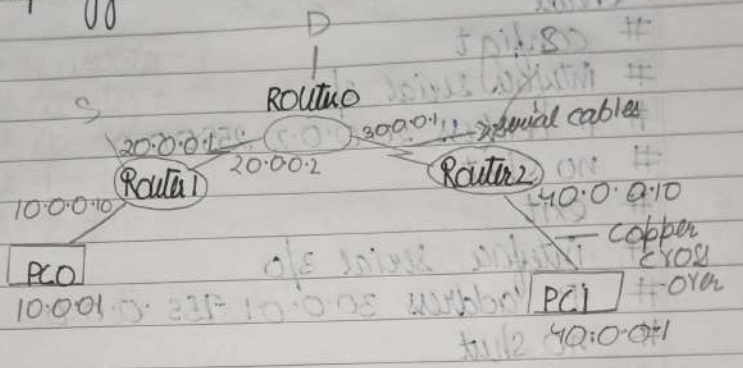
Reply from 10.0.0.2 bytes = 32 time = 6ms TTL = 125
Reply from 10.0.0.2 bytes = 32 time = 2 ms TTL = 125
Reply from 10.0.0.2 bytes = 32 time = 12 ms TTL = 125

Ping statistics for 10.0.0.2
Packets sent = 4 Received = 3 lost = 1
Approximate round trip time in milliseconds:
Minimum = 2ms Maximum = 12ms
Average = 6ms.

14/7/23

Aim Configure default and static route for a connection of routers

Topology



Procedure:

1. Select 3 generic routers and two PC's as end devices connect the PCs to different routers with copper over connect both the router
2. Set the IP addresses of PC and gateway
3. Set the gateway addresses in all the routers taking interface as fastethernet
4. Connect the PCs to the interface.
5) Config Step

> enable (router)
config t
interface fastethernet do

ip address 10.0.0.10 255.0.0.0
no shut
exit

Similarily for routu-0

> enable
config t
intufac serial 2/0
IP address 20.0.0.2 255.0.0.0
no shut
exit
intefac serial 3/0
IP address 30.0.01 255.0.0.0
no shut
exit

For routu-2
> enable
config-t
intufac fastethernet 0/0
IP address 40.0.0.10 255.0.0.0
no shut
exit
intufac serial 2/0
IP address 30.0.0.2 255.0.0.0
no shut
exit

6. We need to set IP route for all routes via routes

For router-1 & router-2 we do default routing and for router-0 static routing is done

For router-1
config t
ip route 0.0.0.0 0.0.0.0 20.0.0.2
no shut
exit

Show ip rout

C 10.0.0.0/8 is directly connected FastEthernet 0/0

C 20.0.0.0/8 is directly connected serial 1/0

S* 0.0.0.0/0 [1/0] via 20.0.0.2

Similarly for router 2

config t
exit
Show ip rout
For router-0 (static routing)
config t
ip rout 10.0.0.0 255.0.0.0 20.0.0.0
ip rout 40.0.0.0 255.0.0.0 30.0.0.0
exit

show ip route

S 10.0.0.0/8 [1/0] via 20.0.0.0
C 20.0.0.0/8 is directly connected
serial 2/0
C 30.0.0.0/8 is directly connected,
serial 3/0

S 40.0.0.0/8 [1/0] via 30.0.0.0

7 Now we ping 10.0.0.1 from the command prom of 40.0.0.1

Result:

~~7 ping 40.0.0.1~~
~~pinting 40.0.0.1 with 32 bytes of data~~

~~Reply from~~

PC > ping 10.0.0.1

pinging 10.0.0.1 with 32 bytes of data:

Request timed out

Reply from 10.0.0.1 bytes = 32 time = 6ms TTL = 125
Reply from 10.0.0.1 bytes = 32 time = 2ms TTL = 125
Reply from 10.0.0.1 bytes = 32 time = 12ms TTL = 125
~~Reply from 10.0.0.1 bytes = 32 time = 16ms TTL = 125~~

10/10

21/7/23

ping statistics for 10.0.0.1

packets sent = 4 Received = 3 lost = 1

Approximate round-trip time in milli-second

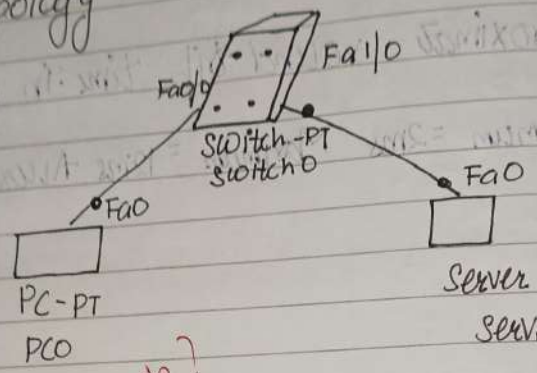
Minimum = 2ms Maximum = 12ms Average = 6ms.

19/10

21/7/23

Aim: configure web server DNS within a LAN

Topology



IP ?

Procedure:

→ Connect a switch, PC and a Server to form a LAN

→ Set PC's IP address by cheking on it and go to config, there in fastEthernet option set the address as 10.0.0.1 and Subnet mask

→ Set service IP address as 10.0.0.2 and subnet mask respectively.

→ go to PC's desktop and click on Web process in the URL Tab type 10.0.0.2 you will get a default display.

→ To make a CV here, we need to make changes in Server services

→ Go to Server → Services → HTTP → index.html here create the CV

→ Again go to pc → desktop → web browser and type 10.0.0.2 you will cv or content that is changed.

→ Next goto server → Services → DNS and switch on the services. Now add a domain name and type the IP address as 10.0.0.2 Press add & save it

→ Again go to PC · desktop Webbrowser and type the given domain name item we can see the cv which has been created earlier.

Observation:

* If you wanted to go to a certain website you open web browser and type domain name of website or use you can also type the IP address instead if you know that website IP address.

* & we can't remember IP address of all website. DNS server will search through it cache to find a matching IP address for that Domain name & when it finds it will resolve that domain name to IP address of website, and that is done that computer is able to communicate with a webserver & retrive the webpage.

~~configure for creating Protocol in Router~~

Aim:

output:

web browser

< > URL http://Akansha GO Stop

CV

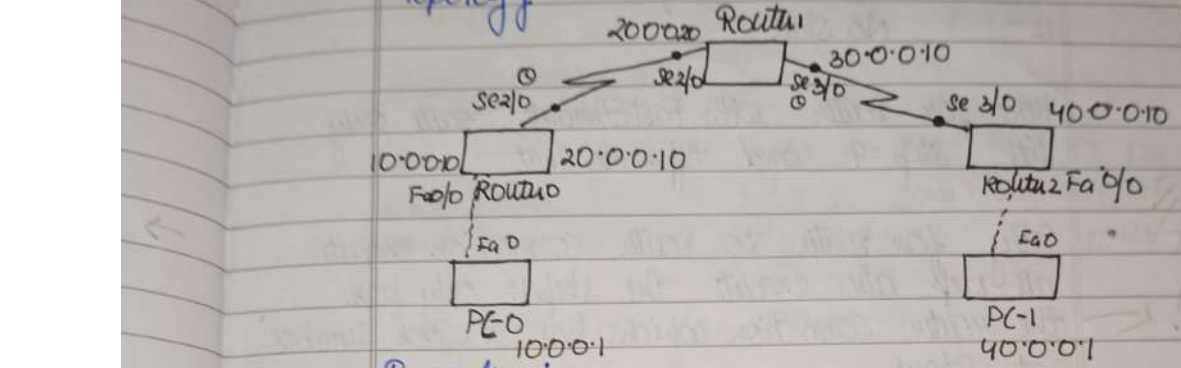
Akansha Jain
USN: 1BM21CS012
Mobileno: 9389840610

Image:

20/07/23

Aim: Configure RIP routing Protocol in Routers.

Topology



Procedure:

- Create a Network using 3 routers and 2PC's connect routers using Serial DCE cable and PC to router using Copper-Cross over cable.
- Set the IP address and gateway no for both PC's as

10.0.0.1 - IP 10.0.0.10 gateway - PC0
40.0.0.1 - IP 40.0.0.10 gateway - PC1

Go to Router CLI mode and execute the following commands

1- NO
2- Enable
3- Config T
4- Interface Fastethernet 0/0
5- IP address 10.0.0.10 255.0.0.0
6- NO Shut

7 - Exit
8 - Interface se2/0
9 - IP address 20.0.0.10 255.0.0.0
10 - Encapsulation ppp
11 - clock rate ?
12 - No shut

- Here for Router with FastEthernet execute only till step-9 and type No shut
- Only for Router to Router connection execute all step also execute the step 11 only for the router connection which has a clock symbol at start.
- Repeat these steps for all routers
- At last now go to each router and type show IP route
- Here IP address associated with that router will be labeled as C and other IP addresses are labbelled as R
- lastly go to PC0 and ping a message to PC1 using ping destination IP address command.

PING output

Packet tracer PC command line 10
PC > ping 40.0.0.1
pinging 40.0.0.1 with 32 bytes of data:

Request timed out.

Reply from 40.0.0.1 : bytes 32 times : 8ms TTL 125
Reply from 40.0.0.1 : bytes 32 times : 5ms TTL 125
Reply from 40.0.0.1 : bytes 32 times : 10ms TTL 125

Ping Statistics for 40.0.0.1

packets sent = 4 Recived = 3 lost 1 (25% loss)
Approximate round trip times in millisecond
Minimum = 5ms Maximum 10ms Average = 7ms

Observation :

- Routing information protocol (rip) is dynamic routing protocol that used hop count as a routing metric to find distance - vector routing protocol.
- Hop count is the no of router coming in b/t source and destination.
- Updates of the Network are exchanged periodically
- Update of routing information are always

- Full routing table are sent in updates
- Router always trust routing information received from neighbor router.

Example 6

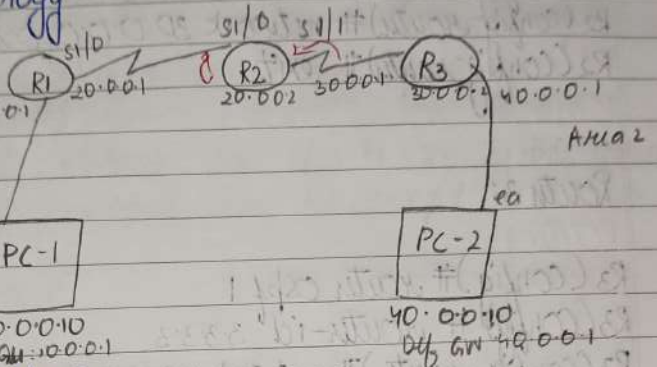
8/10

1/8/23

Experiment - 7

Aim: Configure OSPF routing Protocol. (area)

Topology:



Procedure:

1. Configure the IP address and gateway according to the topology seen above

2. Configure each of the routers according to the IP address given in topology.

3. Encapsulation ppp and clock rate need to be set as done in RIP protocol experiments

Router 1:

R1 (config)# router ospf 1
R1 (config-router)# router-id 1.1.1.1
R1 (config-router)# network 10.0.0.0 0.255.255.255 area 3

R1 (config-router)# network 20.0.0.0 0.255.255.255 area 1
R1 (config-router)# exit

Router 2:

```
R2(config)# router ospf 1
R2(config-router)# router-id 2.2.2.2
R2(config-router)# network 20.0.0.0 0.255.255.252 area 0
R2(config-router)# network 30.0.0.0 0.255.255.252 area 0
R2(config-router)# exit
```

Router 3:

```
R3(config)# router ospf 1
R3(config)# router-id 3.3.3.3
R3(config-router)# network 30.0.0.0 0.255.255.255
                    area 0
R3(config-router)# network 40.0.0.0 0.255.255.
                    255 area 2
R3(config-router)# exit
```

4 To keep the routers Active we have to configure the interface loopback.

Router 1:

```
R1(config-if)# interface loopback 0
R1(config-if)# ip address 172.16.1.252
                    255.255.0.0
R1(config-if)# no shut down.
```

Router 2:

```
R2(config-if)# interface loopback 0
R2(config-if)# ip address 172.16.1.253 255.255.0.0
```

R2 (config-if)# no shutdown.

Router 3:

R3 (config-if)# interface loopback 0
R3 (config-if)# ip address 172.16.1.254 255.255.0.0
R3 (config-if)# no shutdown

5. Create virtual link b/w R_1, R_2 by this we can create a virtual link to connect to Area 0

Router 1:

R1 (config)# router ospf 1
R1 (config-router)# area 1 virtual-link 2.2.2.2

Router 2:
R2 (config)# router ospf 1
R2 (config-router)# area 1 virtual-link 1.1.1.1
R2 (config-router)# exit

Finally, After creating virtual link, show ip route for all routers

Result:
PC > ping 40.0.0.10
pinging 40.0.0.10 with 32 bytes of data

Request timed out

Reply from 40.0.0.10 bytes=32 time 10ms TTL=125
Reply from 40.0.0.10 bytes=32 time 2ms TTL=125

Aim:

Reply from 40.0.0.10 bytes 32 time 9ms TTL=128

Packets: Sent = 4 Recived = 3 lost = 1

Approx round trip:

Minimum = 2ms Maxmum = 10ms Average 7ms

Router 1:

show ip route

O IA 10.0.0.0/8 [110/65] via 20.0.0.1 00:00:11
Serial 2/0

20.0.0.0/8 is vaviably subnetted, 2subnets, 2mask

C 20.0.0.0/8 is directly connected, Serial 2/0

C 20.0.0.0/32 is directly connected, Serial 2/0

30.0.0.0/8 is vaviably subnetted, 2 subnets, 2mask

8/10

C 30.0.0.0/8 is directly connected, Serial 3/0

C 30.0.0.0/32 is directly connected serial 3/6

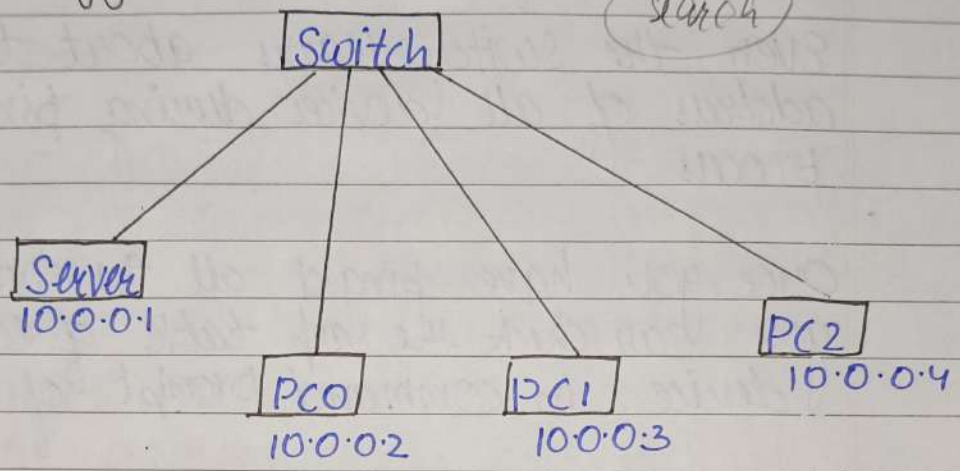
3/8/23

O IA 40.0.0.0/8 [110/65] via 30.0.0.2, 00:64:44,
Serial 3/0

C 172.16.0.0/16 is directly connected, Loopback

Aim: To construct simple LAN and understand the concept and operation of Address Resolution Protocol (ARP)

Topology:



Procedure:

1) Select a switch; server and 3PC's and connect them to the switch as shown in the topology above

2) Connect them with copper straight through wire

3) Set the IP addresses of switch and PCs as shown

4) Select the inspect tool from the tool bar and open the ARP table of all the devices.

5) Then, ping the devices from the command prompt of other devices and click on

capture in the simulation mode to know the packet routing

With every ping the arp table of devices get filled with MAC address of the corresponding devices

Even the switch learns about the MAC address of all devices during pinging process

Once you have pinged all the devices you can check the arp table of each device in command prompt of PC's

> arp -a

Internet address	Physical address	Type
10.0.0.1	0060.47a4.043	Dynamic
10.0.0.2	0060.47e5.1425	Dynamic
10.0.0.3	0005.5eal.0b46	Dynamic

9) In switch → CLI you can check the MAC address of the device as follow
switch > show mac address table

VLAN	MAC address	Type	Ports
1	0005.5eal.0b46	Dynamic	Fa3/1
1	0006.8546.6ac0	Dynamic	Fa3/1
1	0060.47a7.0032	Dynamic	Fa0/1
1	0060.47c5.1424	Dynamic	Fa1/1

Observation:

ARP protocol is communication protocol used for discovering the link layer address such as a MAC address. After pining, every device learn about like MAC address of the pinged devices and the switch stores like MAC address in the ARP table for fixtime pining. ARP learn about the MAC address by pining all the devices and the right IP address respond with the acknowledgement

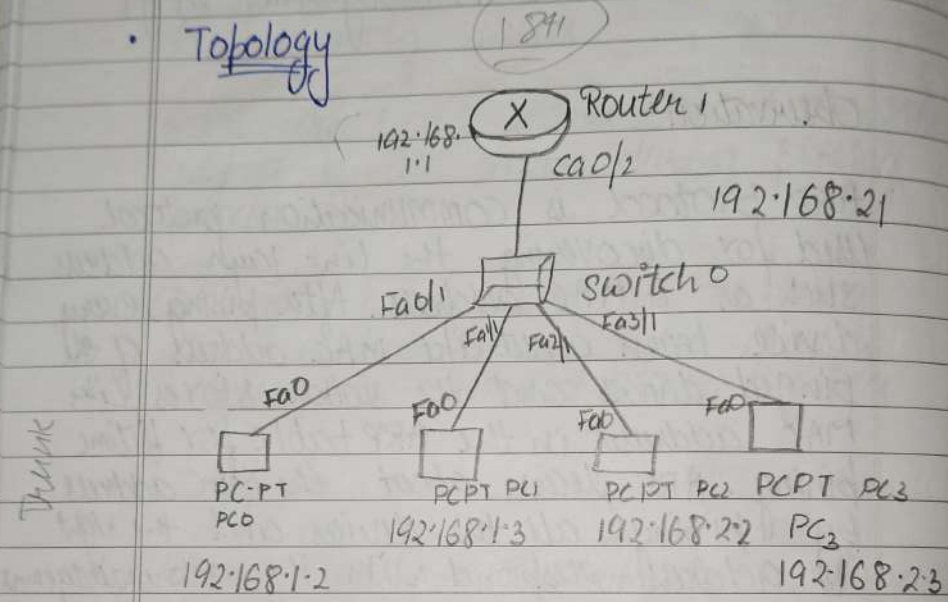
10

4/8/23

Q.7.

- Aim:
  To construct a VLAN and make the PC's communicate among a Vlan

- Topology



Procedure:

* take a router a switch four PC's and drop them in the workspace, choose 1841 router.

* use copper straight through connection to connect all four PCs to the switch and connect the switch to the router

* Set the IP address and the gateway for all PCs

* In the switch → Config → select VLAN

database, given the VLAN Number as 20 and VLAN Name as NewVLAN
Select add

* Select interface fastethernet 0/1 (Near the switch from router and make its trunk)
* Select router → Config → VLAN database → enter number of VLAN and name of the VLAN created.

Router (VLAN) # exit
Apply completed
Exiting
Router # config t
Router (config) # interface Fastethernet 0/0.1
Router (config-subif) # encapsulation dot1Q 20

Router (config-subif) # ip address 192.168.20.1 255.255.255.0

Router (config-subif) # no shut
Router (config-subif) # exit

observation

Virtual local area network broadcast domain that is portitioned and isolated in computer network at data link layer. The PCs in the experiment communicate through a nuclear LAN

Result in (PC 0)

pc > ping 192.168.20.3

pining 192.168.20.3 with 32 bytes of data

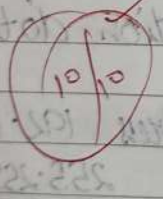
Reply from 192.168.20.3 bytes = 32 time = 1ms TTL = 128

Reply from 192.168.20.3 bytes = 32 time 0ms TTL = 128

Reply from 192.168.20.3 bytes = 32 time 0ms TTL = 128

Reply from 192.168.20.3 bytes = 32 time 0ms TTL = 128

packets : Sent = 4 received = 4 losses 0 (0% loss)



Aim

1.

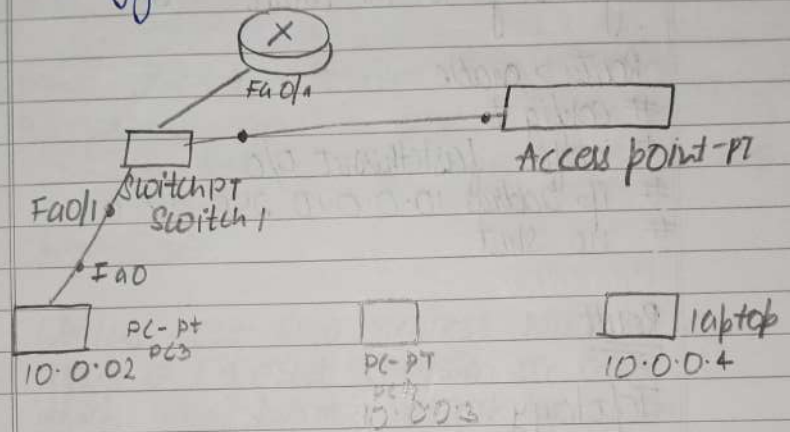
2

3)

Q? A?

Aim: To construct a WLAN and make the Node communicate wirelessly

Topology:



Procedure:

1. Construct the above topology use access point-pt connect it to router set the IP address of the PC connected with wire and configure router 1

2. Configure access-point1-port 1 - SSID name → WLAN
Select WEP and given any 10 digit key (1234567890)

3) To configure PC9 and Laptop wirelessly. Switch off device Drag the existing PT-HOST NM-1AM to the component listed in the LHS Drag WMP300N wirelessly

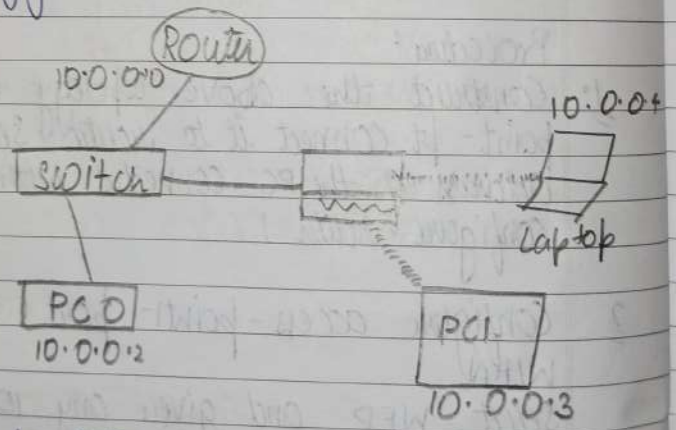
interface to the empty port and switch on the device

4) Now in the config tab, a new wireless interface would have been added config SSID, WEP, WEP key, IP address & gateway to the device

```
Router > enable
# config t
# interface fastethernet 0/0
# ip address 10.0.0.10 255.0.0.0
# no shut
```

Result

Topology.



Result in PC0

PC> ping 10.0.0.3

Pinging 10.0.0.3 with 32 bytes of data:

Reply from 10.0.0.3 bytes=32 time=21ms TTL=128
Reply from 10.0.0.3 bytes=32 times=12ms TTL=128
Reply from 10.0.0.3 bytes=32 times=6ms TTL=128
Reply from 10.0.0.3 bytes=32 times=0ms TTL=128

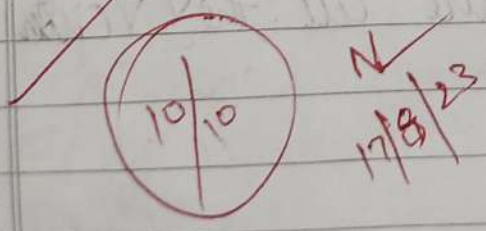
Ping statistics for 10.0.0.3

Packets Sent = 4 Received = 4 lost = 0
Approximate roundtrip time in milliseconds
Minimum = 6ms Maximum = 21ms Average = 12ms

Observation:

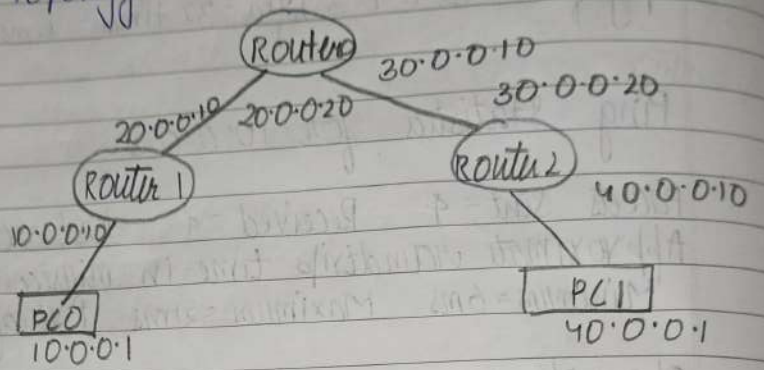
1) Wireless local area network WLAN is a group of allcated computer or other devices that form a network based on radio transmission rather than wired

2) After the WLAN is setup, the wired connection appers in the topology from the access point



Aim- To demonstrate the TTL/life of a packet

Topology -



Procedure

1) Create a 2PC and 3router configuration as shown.

2) Use serial DTE between routers and copper cross over between router and PC

3) Configure the IP address and gateway of PC and configure all the router

for router

> enable
config t
interface fastethernet 0/0
ip address 10.0.0.10 255.0.0.0
no shut
exit
ip route 30.0.0.0 255.0.0.0 20.0.0.2

ip route 40.0.0.0 255.0.0.0 20.0.0.20
exit

For router 1
> enable
config t
interface serial 2/0
ip address 20.0.0.20 255.0.0.0
no shut
exit
interface serial 3/0
ip address 30.0.0.10 255.0.0.0
no shut
exit
ip route 10.0.0.0 255.0.0.0 20.0.0.10
ip route 40.0.0.0 255.0.0.0 30.0.0.20

For router 2
> enable
interface serial 2/0
ip address 30.0.0.20 255.0.0.0
no shut
exit
interface fastethernet 0/0
ip address 40.0.0.10 255.0.0.0
no shut
exit
ip route 10.0.0.0 255.0.0.0 30.0.0.10
ip route 20.0.0.0 255.0.0.0 30.0.0.10

Select Simulation mode, select simple PDU and select source & distination PCs

5 Click on capture button to send PDU

6) Click on PDU during every transfer to see the inbound and outbound PDU details

Results:

PDU information at PC 0
Outbound PDU details:
TTL = 255

PDU information at Router 0
Inbound PDU details
TTL = 255

Outbounds PDU details
TTL = 254

PDU information at Router 1
Inbound PDU details
TTL = 254

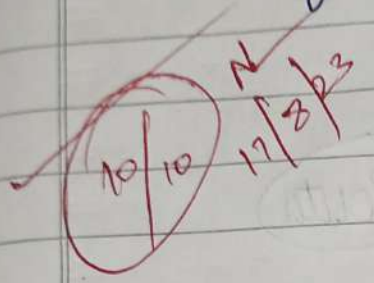
Outbound PDU details
TTL = 253

PDU information at Router 2
Inbound PDU details
TTL = 253

Outbound PDU details:
TTL = 252

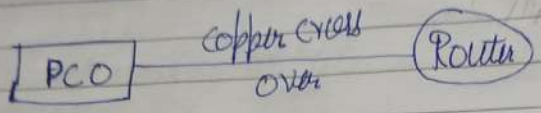
Observation:

The TTL is reduced by 1 in every routers
TTL is a mechanism which limits the number of hops between source and destination.



Aim: To understand the operation of TELNET by accessing the router in server room from a PC in IT office

Topology



Procedure

1) Configure Topology as above use copper cross over wire to connect both configure IP address and gateway and the router generally

2) In Router CLI

```
Router> enable
Router # Config t
Router (config) # hostname r1
r1 (config) # enable secret p1
r1 (config) # interface fastethernet 0/0
r1 (config) # 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.
```

% login disabled on line 132 until password is set

%. login disabled on line 133 until password is set

%. login disabled on line 134 until password is set

%. login disabled on line 135 until password is set

%. login disable on line 136 until password is set

%. login disable on line 137 until password is set

r1 (config-line)# password p0
r1 (config-line) # exit
r1 # exit
Building configuration...

Result
In PC0
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=21ms TTL=255
Reply from 10.0.0.1 bytes=32 time=13ms TTL=255
Reply from 10.0.0.1 bytes=32 time=6ms TTL=255
Reply from 10.0.0.1 bytes=32 time=0ms TTL=255

Pining Statistics from 10.0.0.1
packet Sent=4 Received=4 lost=0

PC> telnet 10.0.0.1
Trying 10.0.0.1 open

User access verfication

Password (Typed PO)
r1> enable
Password (Typed P1)
r1 # show ip rolte

Code:
gateway of last resort is not set
C 10.0.0.8 is directly connected
fastethernet 0/0

r1 #

observation

1) Telnet is used by terminal emulation programs that allows you to log into a remote host

2) We logged into 10.0.0.1 IP device through 10.0.0.2 IP device

3) The password typed is not visible

10/10

17/8/23

cycle-2

Experiment 1

Aim: To write program for error detecting code using CRC-CCITT (16-bits)

```
# include <stdio.h>
# define <string.h>
# define N strlen(poly)
 char data[20];
 char check value[30];
 char poly[10];
 int data length ij;

void XOR
{
 for (j=1; j<N; j++)
 check value[j] = ((check value[j] == poly[j] ? 0:1)
 ); }

void receiver()
{
printf("Enter the recived data:");
scanf("%s", data);

printf("Data recived: %s", data);

for (i=0; (i<N-1) && (check value[i] != i); i++)

 printf("\n Error detected \n"); }

void crd
 for (i=0; i<n; i++)
  check value[i] = data[i];
```

```
do {
  if (check value[0] == '1')
    XOR ();
  for (j=0; j<N-1; j++)
    check value[j] = check value[j+1];
  check value[j] = data[i++];
} while (i <= data length + N + 1);

int main ()
{
printf ("\n Enter data to be transmitted");
  scanf ("%s", data);

  data_length = strlen(data);
  for (i = data length; i <= data length + N - 1; i++)
    data[i] = 0;

printf ("\n data padded with n-1 zero bits");
  crc();
printf ("\n crc value is %s", check value);

  data );
  receiver ();
  return 0; }
```

10/10

q?

Q Write a program for congestion control using leaky bucket.

```
#include <stdio.h>

int main(){
int incoming, outgoing, buck_size, n, store = 0;
printf("Enter Bucket Size");
scanf("%d", &buck_size);
printf("Enter outgoing rate:");
scanf("%d", &outgoing);
printf("Enter no of input");
scanf("%d", &n);

while(n!=0){
printf("Enter the incoming packet size");
scanf("%d", &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 - (bucket_size - store));
printf("Dropped %d no of packets\n
incoming - (buck_size - store));
printf("Bucket buffer size %d out of %d\n",
store, buck_size);
}
store = store - outgoing;
printf("After outgoing %d packets left out of
%d in buffer\n", store, buck_size);
}
}
```

10/10

Output: CRC

Enter the data : 101100
Enter divisor polynomial : 1001
data padded : 10110000
Final code word sent : 10110001
NO error detected

output for leaky bucket

Enter bucket size : 5000
Enter outgoing rate : 2000
Enter number of input : 2
Enter the incoming packet

TCP

Q Using UDP sockets write client - sever program. to make client sending the file name and server to send back the contents of the requested file if present.

TCP

A Client UDP.py

```
from socket import *
ServerName = 12000
Server Port socket = socket(AF_INFT, SOCK_DGRAM)
Sentence = input ("\n Enter the filename :");
ClientSocket sendto (bytes(Sentence, utf-8);
                     (ServerName, ServerPort))

file contents = clientSocket.recv(1024).decode()
print("\n From Server \n")
print(filecontents)
clientSocket.close()
```

Server TCP.py

```
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()
Sentance = connectionSocket.recv(1024).decode()
file = open(Sentance, "r")

l = file.read(1024)

connectionSocket.send(l.encode())
print("\n Sent contents of", Sentance)
file.close()
connectionSocket.close()
```

output

Run ServerUDP.py

The server is ready to receive

Run clientUDP.py : Enter file name : serverUDP.py

Reply from Server

In Server UDP.py

The server is ready to receive

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.

Python program

```
Client UDP.py
from socket import*
ServerName = "127.0.0.1"
Serverport = 12000
clientSocket = socket(AF_INET, SOCK_DGRAM)
Sentence = input("\n Enter file name: ")
clientSocket.sendto(bytes(Sentence, "utf-8")(server
name, Server Port))

filecontents, serveraddress = clientSocket.recvfrom(2048)
print("\n Reply from server \n")
print(filecontents.decode("utf-8"))
# for i in filecontents:
  # print(str(i), end='')
    clientSocket.close()
```

server UDP.py

```
from socket import *
serverPort = 12000
serverSocket = socket(AF_INET, SOCK_DGRAM)
serverSocket.bind(("127.0.0.1", serverPort))
print("The server is receive")
while 1:

Sentence, clientAddress = serverSocket.recvfrom(2048)
Sentance = Sentance.decode("utf-8")

    file = open(Sentance, "8")
    Con= file.read(2048)

    serverSocket.sendto(bytes(Con,'utf-8'),
                        clientAddress)
    print("\n Sent contents of ", end=' ')
    print(Sentance)
    # for i in Sentance:
    #     print(str(i), end='')
        file.close()
```

Result -

Client window

Enter the file name: severTCP.py

contents of the file are displayed

PAGE NO :
DATE :

Server window

The Server is ready to receive
Sent contents of serverTCP.py
The server is ready to received

Wireshark

Wireshark is a network protocol analyzer, or an application that captures packets from a network connection, such as from your computer to your home office or the Internet

Packet is the name given to a discrete unit of data in a typical ethernet network. Wireshark is the most often-used packet sniffer in the world

Benefit of Wireshark

The key features of Wireshark are a live view of arriving packets. A color-coded identification of packet types, which can be customized. A protocol analyzer that identifies the protocol being used by dereferencing the port number shown in the packet header