Tue
1/10/2024

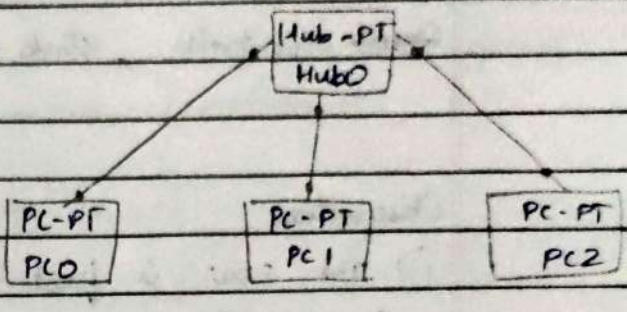
Page No.
Date

LAB 01

AIM : To demonstrate the transmission of a simple PDU between 2 devices connected using a hub and a switch

I. Hub

Topology : Star topology



1) From device-type selection box tab, choose end devices in this choose the PC-PT generic component. Take 3 and place them in the white workspace

2) In the same tab under hub, choose generic, Hub-PT and place it in the workspace.

3) Under connections, choose either automatic or copper cross-over and connect all devices to the hub in star topology.

Configuration :

1) Click on PC-PT PC0, a window/dialog box appears. Under config tab, in the left panel, choose/click FastEthernet0.

2) Repeat this for all 3 devices. In IP address under static, fill 10.0.0.1, 10.0.0.2, 10.0.0.3 and

click on subnet Mask to automatically fill. Close the window.

Procedure:

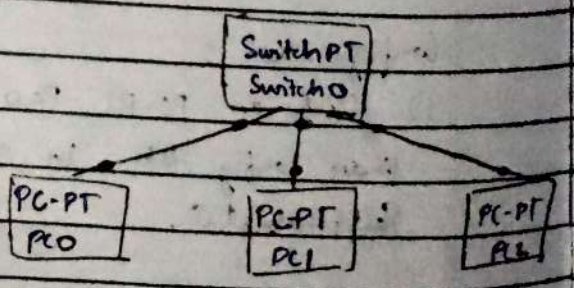
1) Under common tools bar, choose simple PDU option.
2) Click on source first, then destination device.
3) Click on simulation environment. Under play controls, click Auto Capture/Play.

Observation: (physical layer)

1) The PDU is first transmitted from source (PC0) to Hub (Hub0)
2) From Hub, the PDU is transmitted to all other devices connected to that Hub.
3) Other than destination device, all reject the message.
4) The destination device sends the acknowledgement back to the Hub
5) The Hub transmits the acknowledgement back to all the other devices.
6) Only the source device accepts it.

II Switch

Topology: Star topology



1) Instead of Hub, choose switches, under that generic. Switch-PT (switch0).

2) Connect it using connections > automatic, connect all devices to the switch

Configuration:

1) Click on each device, a dialog box appears. In Config tab, under fastEthernet0, fill static IP address.

2) Repeat for all devices. Give 10.0.0.1 ....

Procedure:

1) Under common tools, select simple PDU and choose source and destination.

2) Click on simulation environment and click auto capture play.

Observation: (data link layer)

1) The PDU is transmitted to switch from the source. ~~Once loaded~~

2) Once loaded it is sent to destination only.

3) Acknowledgement is sent back from destination to source. switch

4) This is passed on further from switch to source.

Note: Only ICMP is the message, once done, the STP protocol keeps running to look for shortest path

11/10/24

10/10

8/10/2024

Lab 02

Commands:

1. enable
2. config t
3. interface FastEthernet 0/0
4. ip address 10.0.0.2 255.0.0.0
5. no shutdown

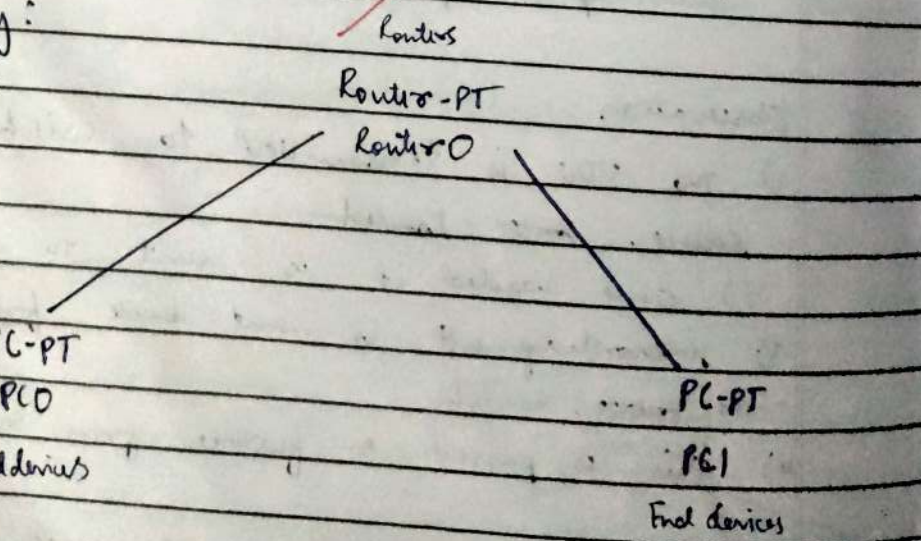
To send message

1) Click on PC-PT
2) Go to desktop tab
3) Click on command prompt
4) Command: ping 20.0.0.1

need to set gateway for the message to go.

Go to config tab, in gateway type 10.0.0.2

Topology:



OBSERVATION:

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 = 4ms TTL = 128

"

"

"

Ping statistics for 10.0.0.1 :

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

Approximate round trip times in milli-seconds:

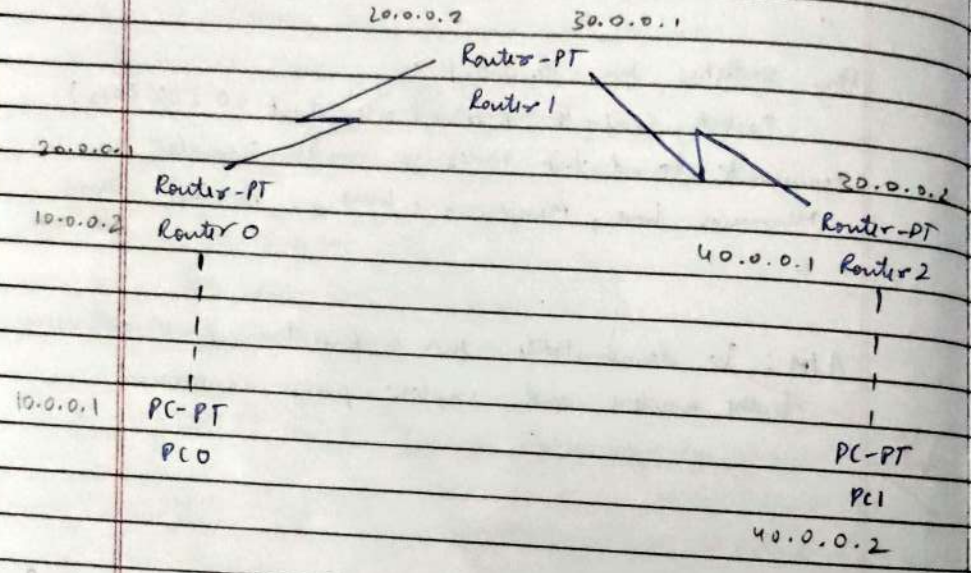
Minimum = 1ms, Maximum = 4ms, Average = 2ms

AIM: To demonstrate the configuration of IP addresses to the routers and explore ping command

Experiment 3

AIM: to demonstrate the configuration of default route to the router

Topology:



Router> show ip route

Router# ip route 30.0.0.0 destination n/w ip } general command
config destination subnet mask next hop

PC-PT
PC0 > Desktop > Command prompt > ping 40.0.0.2

16/10/24

Page No.
Date

Lab 03

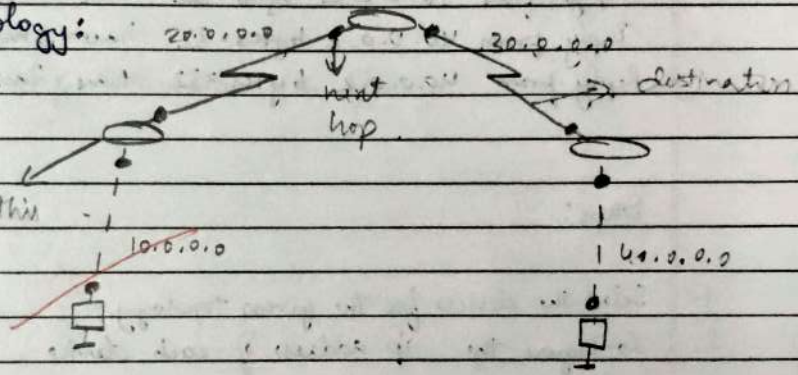
Router> show ip route

(This shows which networks the router has access to directly/indirectly).

Router# ip route space destination nw ip space
Router#(config): destination subnet mask space next hop

Ex:- ip route 30.0.0.0 255.0.0.0 20.0.0.0

Topology:...



Commands:

ip route 30.0.0.0 255.0.0.0 20.0.0.0 } Router0
ip route 40.0.0.0 255.0.0.0 20.0.0.0 }

ip route 20.0.0.0 255.0.0.0 30.0.0.0 } Router2
ip route 10.0.0.0 255.0.0.0 30.0.0.0 }

ip route 10.0.0.0 255.0.0.0 20.0.0.0 } Router1
ip route 40.0.0.0 255.0.0.0 30.0.0.0 }

Aim: OUTPUT:

PC> Ping 40.0.0.2

Ping statistics for 40.0.0.2:
Packets: Sent = 4 Received = 3, Lost = 1 (25% loss)
Approximate round trip times in milli-seconds:
Minimum = 6ms, Maximum = 8ms, Average = 7ms

Pinging 40.0.0.2 with 32 bytes of data:
Request timed out
Reply from 40.0.0.2 bytes = 32 time = 7ms TTL = 125
Reply from 40.0.0.2 bytes = 32 time = 6ms TTL = 125
Reply from 40.0.0.2 bytes = 32 time = 7ms TTL = 125

Steps:

1. Select the devices for the given topology.
2. Configure the IP address of each device

PC-PT PC0 - FastEthernet 0/0 - 10.0.0.1
Router-PT - FastEthernet 0/0 - 10.0.0.2
Router0 Serial 2/0 - 20.0.0.1
Router-PT - Serial 2/0 - 20.0.0.2
Router1 Serial 3/0 - 30.0.0.1
Router PT Serial 2/0 - 30.0.0.2
Router 2 FastEthernet 0/0 - 40.0.0.1
PC-PT PC1 - FastEthernet 0/0 - 40.0.0.2

3. Enable gateway connection for both PC's
PC0 - gateway - 10.0.0.2
PC1 - gateway - 40.0.0.1

4) For each router follow these steps:

i) enable

ii) config t

iii) interface fastethernet o/o or serial 2/o.

iv) ip address 10.0.0.2 255.0.0.0

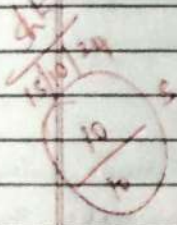
v) No shutdown

This makes all the connections green.

5). Connect the routers to all networks using.

ip route destination subnet hop(next).

6) ping destination PC i/p.



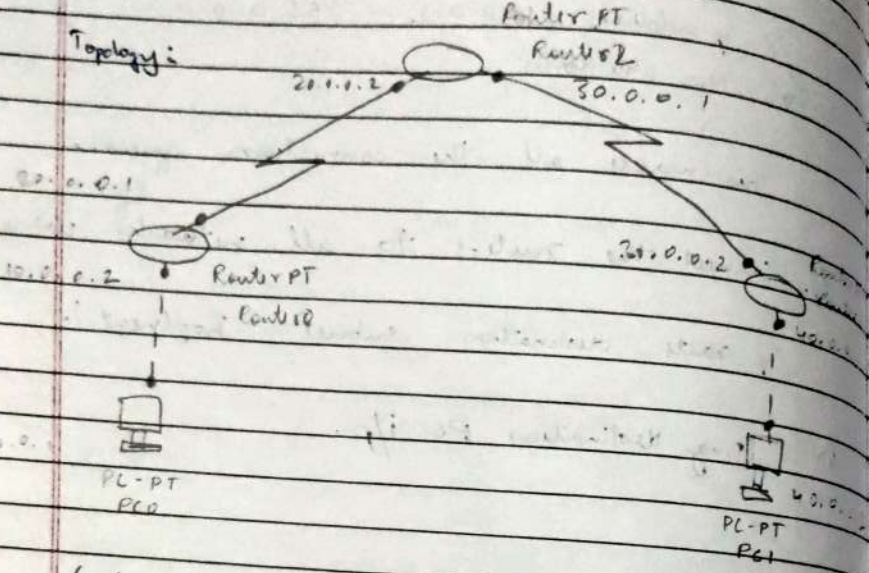
22/10/2024

LAB-04

Experiment 4

AIM: To configure default and static routes to a connection of Routers

Topology:



Configuration:

1) Set IP addresses for all end devices and routers.
2) Open CLI and run the commands to make the connections work, and running (green colour).
3) Set up the gateways for end devices.

Commands:

1) config t
2) # ip route 0.0.0.0 0.0.0.0 20.0.0.2 for R0
3) # ip route 0.0.0.0 0.0.0.0 30.0.0.1 for R1

for R2 4) # ip route 10.0.0.0 255.0.0.0 ~~30.0.0.0~~ 20.0.0.1 } static

5) # ip route 40.0.0.0 255.0.0.0 20.0.0.1 / 30.0.0.2 }

destination network — subnet mask — next hop

OUTPUT:

PC > ping 40.0.0.2

Pinging 40.0.0.2 with 32 bytes of data:

Reply from 40.0.0.2 : bytes = 32 time = 8ms TTL = 125
Reply from 40.0.0.2 : bytes = 32 time = 15ms TTL = 123
Reply from 40.0.0.2 : bytes = 32 time = 15ms TTL = 123
Reply from 40.0.0.2 : bytes = 32 time = 16ms TTL = 123

Ping statistics for 40.0.0.2:
Packets : Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 8ms, Maximum = 16ms, Average 13ms.

Configuration steps for Router 2

Router > enable
Router # config t
Router (config)# interface serial 2/0
Router (config-if)# ip address 20.0.0.2 255.0.0.0
Router (config-if)# no shutdown
Router (config-if)# exit

Router (config)# interface serial 3/0
Router (config-if)# ip address 30.0.0.1 255.0.0.0
Router (config-if)# no shutdown

Similarly configure for

Observation:

On Executing ping command in PC0, desired output is obtained.

Ping 40.0.0.2

22/10/24

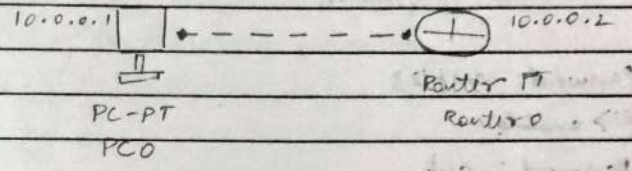
29/10/24

LAB: 05

Experiment 5

AIM: To understand the operation of TELNET by accessing the ~~router~~ server routers placed in the server room from a PC in IT Office.

Topology:



Configuration of PC & Router:

1) Set up the IP address of both PC & router (same network).
2) Set up the gateway for PC (as router IP address)
3) Do the CLI steps to get the queen dots.

Commands:

1) Be in config prompt

Router(config)# hostname R1 giving the host a name
R1(config)# enable secret 123 → (give any password)
R1(config)# line vty 0 5
R1(config-line)# login
% Login disabled on line 123, until 'password' is set

R1(config-line)# password ash123 set password.
exit
exit
R1# wr

Now open terminal in PC

1) ping 10.0.0.2
2) telnet 10.0.0.2

OUTPUT:

PC > telnet 10.0.0.2

Trying 10.0.0.2 ... Open

User Access Verification

Password: aish123

R1> enable

Password: aish

R1#

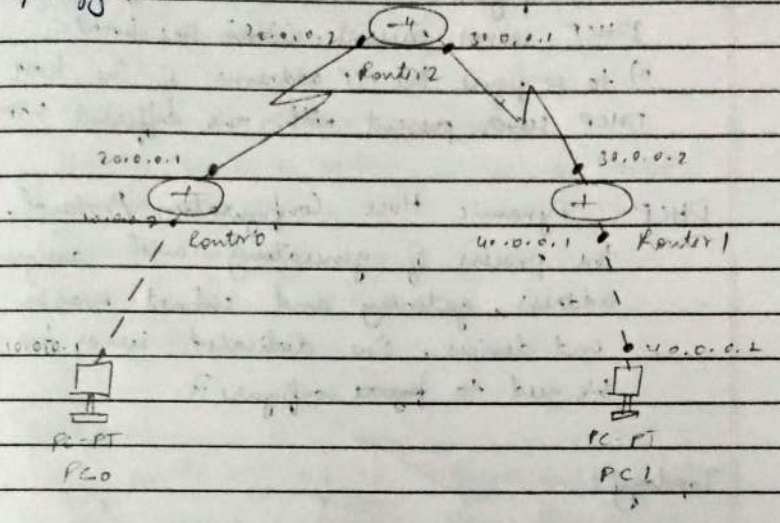
PC > ping 10.0.0.2

Pinging 10.0.0.2 with 32 bytes of data:

Reply from 10.0.0.2: bytes = 32 time = 0ms TTL = 255
Reply from 10.0.0.2: bytes = 32 time = 2ms TTL = 255
Reply from 10.0.0.2: bytes = 32 time = 0ms TTL = 255
Reply from 10.0.0.2: bytes = 32 time = 0ms TTL = 255

AIM: To understand how ~~TCP~~ TTL works.

Topology:



Configuration:

Same as previous lab.

Steps:

1) Choose simple PDU, send it from PC0 to PC1.
2) Go to simulation mode, click auto capture/play.
3) Click on the envelopes on PC and routers at each step.
4) Go to the Inbound PDU tab, notice the TTL as it keeps reducing in each hop.
5) By the time message is sent, TTL is reduced to 0.

12/11/24

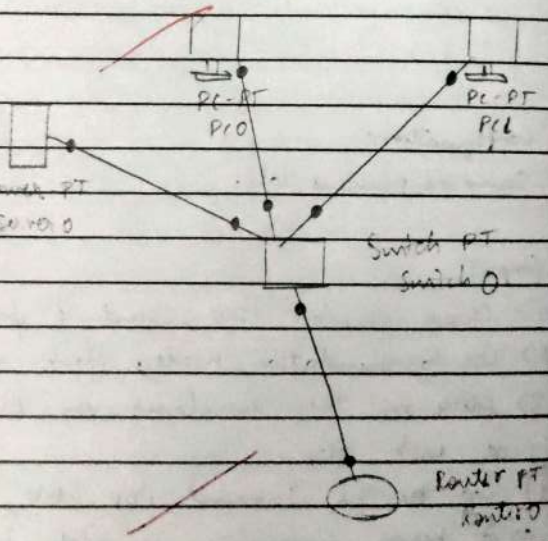
LAB-06

AIM: a) To configure the IP addresses of the host using DHCP server present within the LAN

b) to configure the IP addresses of the host using DHCP server present within the different LAN.

DHCP - Dynamic Host Configuration Protocol. Automates the process of generating and assigning IP address, gateway and subnet masks to the end devices. One dedicated server for this is ~~ignore~~ configured. We need to configure it.

Topology:



Configuration:

1) Assign the IP address (10.0.0.2) manually to the server
2) On Router we can't apply DHCP, apply IP address (10.0.0.2) manually
3) In Router CLI, do these steps

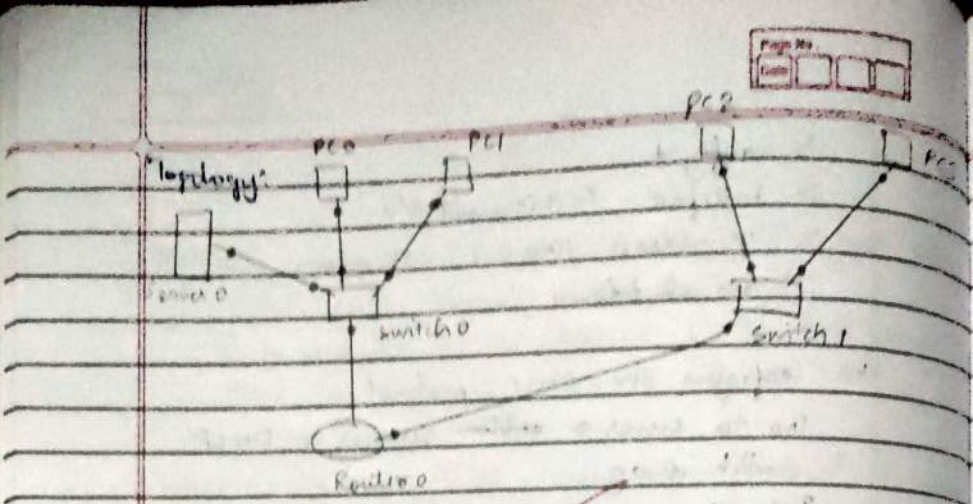
. > enable

> config t
> interface Fast Ethernet 0/0
> ip address 10.0.0.2 255.0.0.0
> no shutdown

4) Configure the DHCP protocol.
Go to server → ~~setting~~ services → DHCP.
Switch it on.
Default gateway : 10.0.0.2
DNS Server : 10.0.0.1
Click save.

PROCEDURE:
1) Click on PC-PT, go to config → Fast Ethernet 0.
Here instead of Static, choose DHCP. This automatically chooses one IP address from the pool and assigns it.
2) Do the same for PC PT 1.
3) Go to Command prompt in PC-PT 0.
> ping 10.0.0.4.

OUTPUT: PC> ping 10.0.0.4
Pinging 10.0.0.4 with 32 bytes of data:
Reply from 10.0.0.4 bytes=32 time=0ms TTL=128
Reply from 10.0.0.4 bytes=32 time=0ms TTL=128
Reply from 10.0.0.4 bytes=32 time=0ms TTL=128
Reply from 10.0.0.4 bytes=32 time=0ms TTL=128.
Ping statistics for 10.0.0.4:
Packets: Sent=4, Received=4, Lost=0 (0% loss)
Approximate round trip times in milli-seconds:
Minimum=0ms, Maximum=0ms; Average=0ms



Configuration:

1) For router, go to Fast Ethernet 1/0, give IP address 20.0.0.1. In CLI, do these:

> interface FastEthernet 1/0
> # ip address 20.0.0.1 255.0.0.0
> # no shut down.
> # ip helper 10.0.0.1
> # exit.
> (config)# interface FastEthernet 0/0
> # ip helper 10.0.0.1
> # exit.

2) Go to server → Services → DHCP.

Pool name : serverPool1
Default Gateway: 20.0.0.1
DNS Server : 20.0.0.1
Start IP address: 20 0 0 0
Click save.

Procedure:

1) Go to PC2 → Config → FastEthernet 0:
Here instead of static, click DHCP. This automatically assigns IP address.

2) Do the same for PC3

3) Go to PC2 → Desktop → command line. Type
> ping 20.0.0.5.

4) Next
> ping 10.0.0.4.

OUTPUT: PC> ping 20.0.0.3

Pinging 20.0.0.3 with 32 bytes of data:
Reply from 20.0.0.3 : bytes = 32 time = 0ms TTL = 128
Reply from 20.0.0.3 : bytes = 32 time = 0ms TTL = 128
Reply from 20.0.0.3 : bytes = 32 time = 0ms TTL = 128
Reply from 20.0.0.3 : bytes = 32 time = 0ms TTL = 128

Ping statics for 20.0.0.3:
Packets: Sent = 4, Received = 4, Lost = 0 (0% Loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 1ms, Average = 0ms

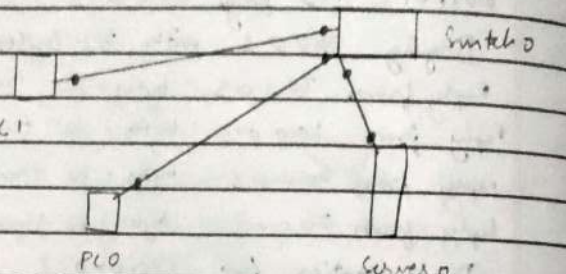
PC> ping 10.0.0.3

Pinging 10.0.0.3 with 32 bytes of data:
Reply from 10.0.0.3 : bytes = 32 time = 3ms TTL = 127
Reply from 10.0.0.3 : bytes = 32 time = 1ms TTL = 127
Reply from 10.0.0.3 : bytes = 32 time = 0ms TTL = 127
Reply from 10.0.0.3 : bytes = 32 time = 0ms TTL = 127

Ping statistics for 10.0.0.3:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 3ms, Average = 1ms.

AIM: To configure DNS Server to demonstrate the mapping of IP addresses and domain names

Topology:



Configuration:

1) Assign IP addresses to both PC0 & PC1, Assign one to server also
2) Go to server → Services → DNS.
   Enable on
   Name: website1
   Address: 10.0.0.1
   Click Add.
   Go to HTTP.
   Click edit for index.html.
   Click save.

Procedure:

1) Go to PC0 → Desktop → web browser.
2) Search website 1.
3) Search 10.0.0.1

OUTPUT:

URL : https://website1

Cisco Packet Tracer

Welcome to Cisco Packet Tracer

Quick links:

A page

Copyrights

Image

Image1

URL: https://10.0.0.1

Cisco Packet Tracer

Welcome to Cisco Packet Tracer

· Quick links:

A page

Copyrights

Image

Image1

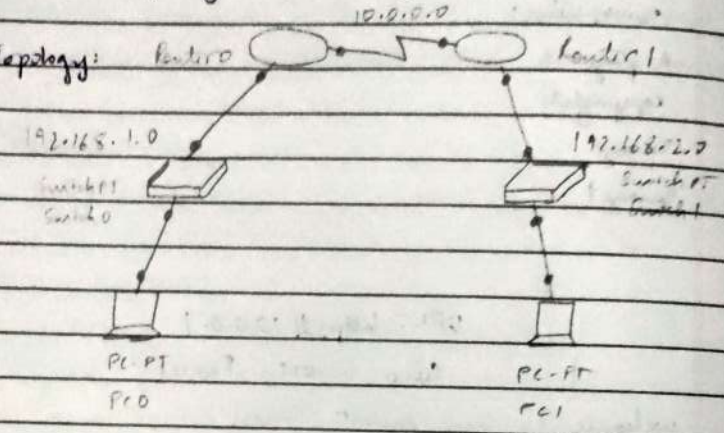
18/11/24

19/4/2024

LAB-07

AIM: Configuring RIP (Routing)

Topology:



Configuration:

1) Assign IP addresses and set gateway for both PC0 and PC1.
2) In both Router, set IP addresses for the Fast Ethernet 0/0 interface. Turn the Port status to on.
3) In Router 0, go to serial 2/0, assign IP address 10.10.0.2, change clock rate to 64000 and click Port status On. In settings, click save.
4) In Router 1, go to serial 2/0, clock rate to Not set, IP address 10.10.0.3 and Port status On. Click save in settings.

Procedure:

1) Send a simple PDU from PC0 to PC1.

~~Output~~

~~Successful~~

RIP Routing:

1) Go to Router0 → Routing → RIP.
2) Add 192.168.1.0 and 10.0.0.0

3) Go to Router1 → Routing → RIP
4) Add 192.168.2.0 and 10.0.0.0.
5) Go to settings → Save NVRAM.

OUTPUT:

In PC0 → Desktop → Command Prompt

PC> ping 192.168.2.2

Pinging 192.168.2.2 with 32 bytes of data:

Reply from 192.168.2.2: bytes=32 time=5ms TTL=126
Reply from 192.168.2.2: bytes=32 time=5ms TTL=126
Reply from 192.168.2.2: bytes=32 time=5ms TTL=126
Reply from 192.168.2.2: bytes=32 time=5ms TTL=126

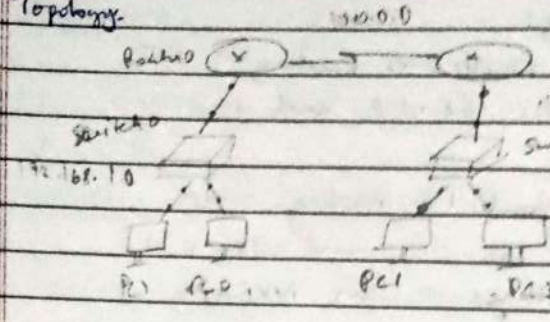
Ping statistics for 192.168.2.2:

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

Approximate round trip times in milli seconds:

Minimum = 1ms, Maximum = 5ms, Average = 2ms

Topology:



Add 2 more PC's, PC2 & PC3 in both networks. Configure the IP address 192.168.1.3 & 192.168.2.3. Add gateways. Increase clock rate to 72000.

Output:

> ping 192.168.2.3

Pinging 192.168.2.3 with 32 bytes of data:
Reply from 192.168.2.3: bytes=32 time=1ms TTL=126
Reply from 192.168.2.3: bytes=32 time=1ms TTL=126
Reply from 192.168.2.3: bytes=32 time=1ms TTL=126
Reply from 192.168.2.3: bytes=32 time=1ms TTL=126

Ping statistics for 192.168.2.3:
Packets: Sent=4, Received=4, Lost=0 (0% loss)
Approximate round trip time in milli-seconds:
Minimum=1ms, Maximum=7ms, Average=2ms

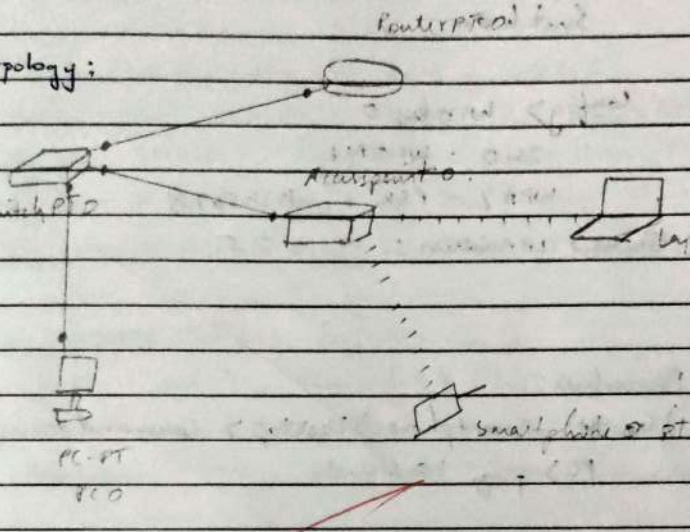
18/11/24

26/11/24.

LAB - 08

AIM: To demonstrate communication between 2 devices using a wireless LAN

Topology:



Configuration:

1) Give IP address to PCPT0 as 10.0.0.1 and Router 0 as 10.0.0.2. For laptop 0 given 10.0.0.3 and smartphone 10.0.0.4.

2) For PC0 & Laptop 0, give gateway as 10.0.0.2 as Router IP address

3) Click on Access point, in Config

PORT 0 : Click Auto.

PORT 1 : SSID : WLAN1

Authentication: WPA2-PSK .. 12345678

4) Click smartphone > config > wireless

SSID: WLAN1

WPA2-PSK 12345678

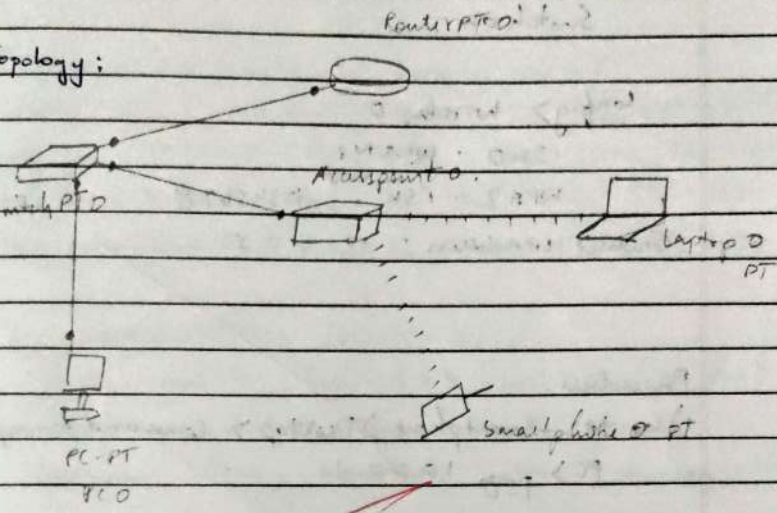
Static 10.0.0.4

26/11/24.

LAB-08

AIM: To demonstrate communication between 2 devices using a wireless LAN

Topology:



Configuration:

1) Give IP address to PC-PT-0 as 10.0.0.1 and Router-0 as 10.0.0.2. For laptop-0 given 10.0.0.3 and Smartphone 10.0.0.4.

2) For PC-0 & Laptop-0, give gateway as 10.0.0.2 as Router IP address

3) Click on Access point, in Config

PORT-0: Click auto.

PORT-1: SSID : WLAN1

Authentication: WPA2-PSK 12345678

4) Click smartphone > config > wireless

SSID: WLAN1

WPA2-PSK 12345678

Static 10.0.0.4

5) Laptop > physical
Switch off.
Drag WPC300N.
Replace it.
Switch on

Config > Wireless 0
SSID : WLAN1
WPA2 - PSK 12345678.
Static > IP Address : 10.0.0.5

Procedure:
1) Go to Smartphone > Desktop > Command prompt
PC > ping 10.0.0.1

2) Go to Laptop > Desktop > Command Prompt
PC > ping 10.0.0.1

OUTPUT:

1) 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 = 128
Reply from 10.0.0.1 : bytes = 32 time = 12ms TTL = 128
Reply from 10.0.0.1 : bytes = 32 time = 9ms TTL = 128
Reply from 10.0.0.1 : bytes = 32 time = 15ms TTL = 128
Ping statistics for 10.0.0.1 :
Packets : Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip time in milli-seconds :
Minimum = 9ms, Maximum = 21ms, Average = 14ms

2) Laptop.

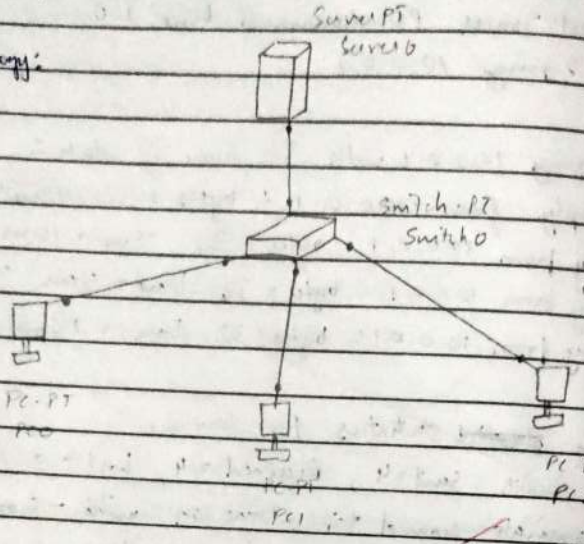
Packet Tracer PC Command Line 1.0
PC > ping 10.0.0.1

Pinging 10.0.0.1 with 32 bytes of data:
Reply from 10.0.0.1: bytes = 32 time = 24ms TTL = 12
Reply from 10.0.0.1: bytes = 32 time = 10ms TTL = 128
Reply from 10.0.0.1: bytes = 32 time = 5ms TTL = 128
Reply from 10.0.0.1: bytes = 32 time = 13ms TTL = 128

Ping statistics for 10.0.0.1:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip time in milli-seconds:
Minimum = 2ms, Maximum = 21ms, Average = 15ms

Aim: Demonstrate the working of Address Resolution Protocol for communication within a LAN

Topology:



Configuration:

1) Arrange 1 server, 1 switch & 3 PC in above configuration.
2) Set IP address for 3 PC & server
   - PC0 : 10.0.0.1
   - PC1 : 10.0.0.2
   - PC2 : 10.0.0.3
   - Server : 10.0.0.4
3) In PC0 > Desktop
   PC> arp -a
   No ARP found Entries
4) Click inspect & and select PC0 (ARP table).
   Empty
   Do same for Server 0.

Page No.
Date

Procedure:

1) Send a PDU from PC0 to Server0. On successful completion, ARP was called and respective MAC address entries are entered in ARP table.

2) Now send a PDU from PC1 to Server0 using capture / forward mode. 2 packets visible:
ICMP and ARP
ARP → outbound PDU details:
Destination MAC address : 0000.0000.0000
as MAC address is not Resolved.

OUTPUT:

After successfull completion and receiving response from Server0, Inbound PDU details will have resolved destination MAC address and server ARP table will have new entry for PC1

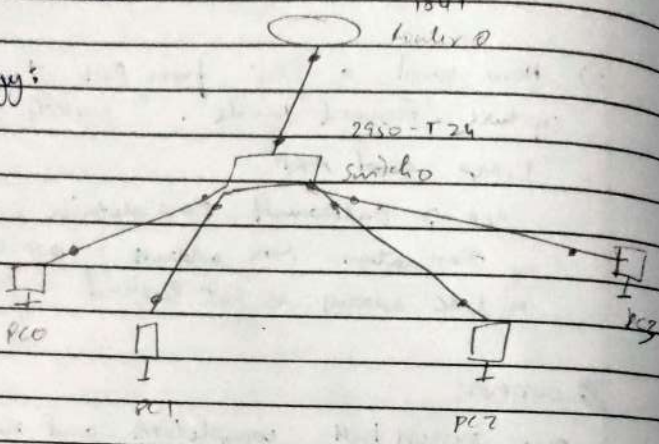
Inbound PDU details:
DST MAC: 00D0.D304.7191
SRC MAC: 0001.965C.1B77
SRC IP: 10.0.0.4
DST IP: 10.0.0.2

3/12/2024.

LAB-09

AIM: To create a virtual lan on top of the physical lan and enable communication between physical lan and virtual lan

Topology:



Configuration:

1) Connect the PC's to switch, the switch to router
2) Open CLI in Router
   - > enable
   - # config t
   - > interface FastEthernet 0/0
   - > ip address 192.168.10.1 255.255.255.0
   - > no shutdown.

3) Configure the 2 left most PC.

IP: 192.168.10.2	IP: 192.168.10.3
Gateway: 192.168.10.1	Gateway: 192.168.10.1
Subnet mask: 255.255.255.0	Subnet mask: 255.255.255.0

4) Go to switch -> FastEthernet 0/5, under the drop down choose Trunk

Trunk - interface

5) Go the VLAN database,
VLAN Number : 123
VLAN Name : vlan1
Click Add.

6) Go the Right most FastEthernet 0/9 ok,
in VLAN drop down, Choose 123 vlan1.

7) The 2 Right PC, configure with IP address
192.168.20.1 & 192.168.20.2 with gateway
192.168.20.3

8) Go to CLI of router -
enable
config t
Interface FastEthernet 0/0.1
ip address 192.168.20.3 255.255.255.0.
no shutdown.

9) Go to config > VLAN database -
add 123 & vlan1

10) Go back to CLI
enable
config t
> interface fastethernet 0/0.1
encapsulation dot1q 123 (LAN Number)
> ip address 192.168.20.3 255.255.255.0.
> no shutdown.

Procedure:

1) Send a PDU from router to both physical LAN and virtual LAN. Also send a PDU from PC0 to PC1

OUTPUT:

Last status	Source	Destination	Type	Time (sec)	Num
Successful	Router0	PC0	ICMP	0.000	0
Successful	Router0	PC1	ICMP	0.000	1
Successful	Router0	PC2	ICMP	0.000	2
Successful	Router0	PC3	ICMP	0.000	3
Successful	PC0	PC2	ICMP	0.000	4
Successful	PC0	PC3	ICMP	0.000	5
Successful	PC1	PC2	ICMP	0.000	6
Successful	PC1	PC3	ICMP	0.000	7
Successful	PC0	PC1	ICMP	0.000	8
Successful	PC2	PC3	ICMP	0.000	9

Transmission of PDU from physical LAN to virtual LAN is successful.

3/12/24

2/12/2024

CRC Checksum Code

```
#include <stdio.h>
#include <string.h>
#define N strlen(gen_poly)

char data[28], check_value[28], gen_poly[10];
int data_length, i, j;

void XOR() {
  for(j=1; j<N; j++)
  check_value[j] = ((check_value[j] == gen_poly[j])?'0':'1');
}

void receiver(){
  printf("Enter the received data: ");
  scanf("%s", data);
  printf("\n-----\n");
  printf("Data received: %s", data);
  crc();
  for(i=0; (i<N-1) && (check_value[i]!='1'); i++);
    if(i<N-1)
      printf("\nError detected\n\n");
    else
      printf("\nNo error detected\n\n");
}

void crc(){
  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("\nEnter data to be transmitted: ");
scanf("%s", data);
printf("\nEnter the generating polynomial: ");
scanf("%s", gen_poly);
data_length = strlen(data);
for(i= data_length; i< data_length +N-1; i++)
    data[i]='0';
printf("\nData padded with n-1 zeroes: %s",data);
crc();
printf("\nCRC or Check value is: %s", check_value);
for(i=data_length; i<data_length+N-1; i++)
    data[i] = check_value[i-data_length];
printf("\nFinal data to be sent: %s.\n", data);
receive();
return 0;
}
```

Lab - 10

Leaky Bucket

```
storage = 0
no_of_queries = 4
bucket_size = 10
input_pkt_size = 4
output_pkt_size = 1
for i in range (0, no_of_queries ):
    size_left = bucket_size - storage
    if input_pkt_size <= size_left :
        storage += input_pkt_size
    else :
        print ("Packet loss = ", input_pkt_size)
print ("Buffer size = {storage} out of bucket size
    = {bucket_size}")

storage -= output_pkt_size
```

24/12/24

LAB - 11

Using TCP/IP 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 found.

Client.py

```
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('From Server:', filecontents)
clientSocket.close()
```

Server.py

```
from socket import *
serverName = "127.0.0.1"
serverPort = 12000
serverSocket = socket(AF_INET, SOCK_STREAM)
serverSocket.bind((serverName, serverPort))
serverSocket.listen(1)
print("The server is ready to receive")
while 1:
    connectionSocket, addr = serverSocket.accept()
    sentence = connectionSocket.recv(1024).decode()
    file = open(sentence, "r")
    l = file.read(1024)
```

connectionSocket.send(l.encode())
file.close()
connectionSocket.close()

→ OUTPUT:

>python Server.py
The server is ready to receive

>python client.py
Enter file name: C:\Users\ADMIN\Downloads\task.txt
From Server: Hello world

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

ClientUDP.py

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

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)
    file = open(sentence, "r")
    l = file.read(2048)
    serverSocket.sendto(bytes(l, "utf-8"), clientAddress)
    print ("sent back to client", l)
file.close().
```