2

Page
Date 13/07/24

Aim:

Study of various network commands used in linux and windows

Basic Network commands:

arp: Interface: 172.16.75.54 --- 0x12

Internet address	Physical address	type
172.16.72.16	7C-5A-1C-CF-be-41	dynamic
172.16.72.133	AC-9C-a3-b5-97-f3	dynamic

hostname:

DESKTOP-C01BHTD

nbstat-a:

NBSTAT [[-a RemoteName] [-A IP address] [-c] [-n] [-N] [-R] [-RR] [-s] [-S] [interaval]]

-a (adapter strains) List the remote machine's name table given its name

nbstat:

Active connection

Photo	Local address	Foreign Address	status
TCP	127.16.72.1	DESKTOP-C01BH7D:49679	ESTABLISHED

nslookup:

Default server: unknown

Address: 172.16.72.1

pathping: usage: pathping [-g host-list] [-h maximum hops] [-i address] [-n] [-Period] [-q num-queries]

e. [root @ server N] # ip link set etho down

To alter the status of the interface by turning the interface enp35d

f. [root @ server N] # ip link set etho promise on

To alter the status of inter by enabling mode

g. [root @ server N] # ip route add default via 192.16.1.254 dev etho

To add a default route (for all addresses) via the local gateway 192.168.254 that can be reached on device enp350

h. [root @ server N] # ip route add 192.168.1.0/24 via 192.168.1.254

To add a route to 192.168.1.0/24 to via the gateway at 192.168.1.254

i. ~~route~~ [root @ server ~] # ip route add 192.168.1.0/24 dev etho

To add a route to 192.168.1.0/24 that can be searched on device enp350

j [root @ server N] # ip route delete 192.168.1.0/24 via 192.168.1.254

To delete the route for 192.168.1.0/24 via the gateway at 192.168.1.254

k. Display :→ [root @ server N] # ip route get 10.10.1.4

10.10.1.4 dev ens160 are 192.168.1.254 uid 1000 cache

2. IPconfig :

ens160 : flags = 4163<UP, BROADCAST, RUNNING, MULTICAST>
mtu 1550

inet 192.168.22.128 netmask 255.255.255.0 broadcast 192.168.22.255

3. mtr :

mtr <options> hostname / IP

a. [root @ server ~] # mtr google.com

keys : Help Display mode Restart statistics order of fields quit

	Packets			Pings			
1. Host	Loss%	Snt	Last	Avg	Best	Wrst	Stdev
gateway	0.0%	168	1.2	1.0	0.5	6.1	0.6

b. [root @ server N] # mtr -g google.com

-F, --filename FILE read hostname(s) from a file
-4 use IPV4 only
-6 use IPV6 only
-u --udp use UDP instead of ICMP echo

c. [root @ server N] # mtr -b google.com

keys : help display mode Restart statistics order of fields quit

	Packets			Ping			
Host	Loss%	Snt	Last	Avg	Best	Wrst	Stdev

d) [root@server~]# mtr -c to google.com
keys: Help Displaymode Restartstatistic ordering fields quit
packets

Host	Loss%	Snt	Last	Avg	Best	Wrst	stop
-gateway	0.0%	4	0.7	7	0.4	1.5	0.3

4) tcpdump
[root@server~]#dnf install -Y tcpdump

to mistake install -Y tcpdump
[root@server~]# tcpdump -D
1. ens10 [up, Running, connected]
2. any (pseudo-device that captures on all interfaces) [up, Running]

[root@server~]# tcpdump -i etho
dropped privs to tcpdump
TCP dump: Virbose output suppressed, use -v[v]... for full protocol decode

[root@server~]# tcpdump -i etho -c 10
10 packets captured
20 packets received by filter
0 packets dropped by kernel

[root@server~]# tcpdump -i etho -c 10 host 8.8.8.8
dropped privis to tcpdump
tcpdump: Verbose output suppressed, use -v[v]
for full protocol decode listening on 10, link-type

ENIOMB (Ethernet), snapshot length 262144 by
[root@server~]# tcpdump -i etho src host 8.8.8

dropped privis to tcpdump
tcpdump: verbose output suppressed, use -v[v]
for full protocol decode listening on 10,

[root@server~]# tcpdump -i etho dst host 8.8.8

dropped privis to tcpdump
tcpdump: verbose output suppressed, use -v[v]
for full protocol decode listening on 10.

[root@server~]# tcpdump -i etho net 10.1.0.0
mask 255.255.255.0

To capture traffic and to from a specific network using the command

[root@server~]# tcpdump -i etho net 10.1.0.012

To capture traffic and to form specific networ
using the command

[root@server~]# tcpdump -i etho host 8.8.8.8
and port 53 traffic

[root@server~]# tcpdump -i etho host 8.8.8.8 and port
53

ping:

usage: ping [-t][-a][-n count][-l size][-f][-i TTL][-v TOS][-r count][-s count][[-j host-list] | [-k host-list][-w timeout][-R][-S srcaddr][-c compartment][-p][-4][-6] target_name

[root@-server ~]# ping google.com
ping google.com (216.58.200.142) 56(84) bytes of data 64 bytes from maa05s10-in-f14.1e100.net (216.58.200.14.2): icmp-seq=1 ttl

[root@server ~]# ping -c 10 google.com
PING google.com (142.250.193.100) 56(84) bytes of data -google.com ping statistics
10 packets transmitted, 0 received, +1 errors, 100% packet loss, time 9197 ms pipe 3

Result:

Thus the study of various Network commands used in linux and windows is done and executed successfully.

nmcli:

Procedure to list the network manager connection profiles

#nmcli connection show

o/p: NAME UUID
wired connection 10 95a9408 2bbc395 lic 84f54
e22eba-8faa-4761341-b12942

#nmcli connection add con-name <connection_name> ifname <name> skip this step to modify as existing profile

Display the current setting of connection profile.

#nmcli connection show
connection.interface show
connection.autoconnect: yes
ipv4.method: auto
ipv6.method: auto

Page
Date

Study of Different Kinds of Network cable

Exp: 2 Date 27/1/24

Aim:

Study of different types of network cables

a) understand different type of cables used in networking are: 1. unshielded twisted pair (UTP) cable
2. Shielded twisted pair (STP) cable
3. coaxial cable
4. Fibre optic cable.

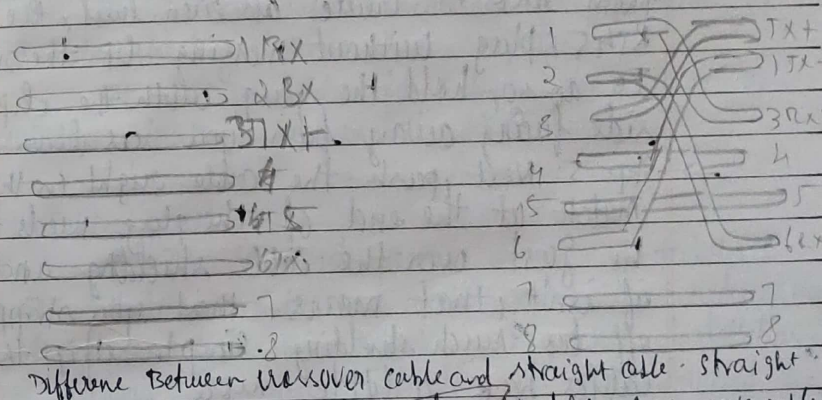
cable type	category	Maximum Data transmission	Advantages/ disadvantages	Apply/ use	Image
UTP	category 3 category 5	10 bps up to 100mbps	Advantages: *cheaper in cost *easy to install as they have a smaller overall diameter	10Base ethernet Fast ethernet	
	category 5e	1Gbps	Disadvantages: *More prone to emi? electro magnetic interference and noise	Gigabyte Ethernet Fastethernet gigabyte ethernet	
STP	category b,ba	10Gbps	Advantages Shielded, faster than UTP less susceptable to noise and interference	Gigabit ethernet 10GEthernet (ssm) widely used in data centres	

STP	category	10Gbps	disadvantages: expensive greater installation effort.	Gigabit Ethernet 10Gbethernet (100m)
coaxial cable	RG-6 RG-59 RG-11	10-100 mbps	Advantages: *High bandwidth *immune to insufficient *low cost band width Disadvantages: *Limited distance * cost * size is bulky	speed of signal is room Television network high speed internet connection
fibre optic cable	single mode multi mode	100gbps	Advantages: *High speed *high band width *long distance Disadvantages Expensive Requires skilled installers	maximum distance of fibre optics cable is around 100 metres

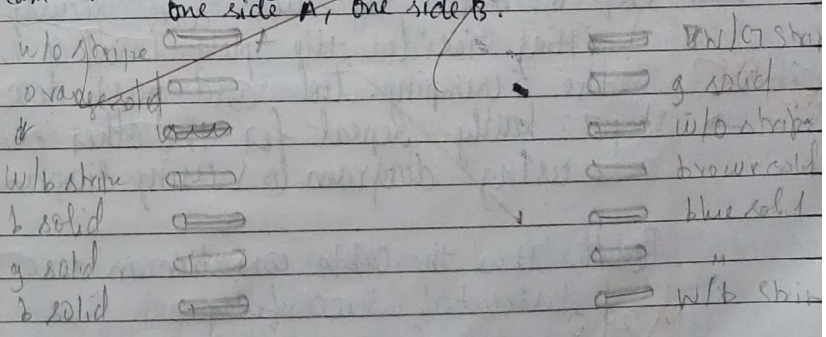
b) Make your own ethernet Cross-over cable/ straight cable.

Tools and parts needed

- ethernet cabling CAT5e is certified for gigabit support, but CAT5 cabling works as well just over shorter distance
- A crimping tool this is an all in one networking tool shaped to push down the pins in the plug and strip and ~~cut~~ cut the shielding off the cables.
  - * Two RJ45
  - * optional two plug shields



Difference Between crossover cable and straight cable straight cable through network cable: both sides should be A crossove cable one side A, one side B.



→ Step-1: To start construction of the cable, begin by threading shields onto the cable

→ Step-2: Next strip approximately 1.5cm of cable shielding from both the ends. the crimping tool both the ends.

→ step3: After you will need to ist angle the wire; there should be four "twisted pair" referencing back to the sheet, arrangement and other in B be in arrangement.

→ Step-4: Once the order is correct, bunch them together in a wire, and if there are any that stick out further than other, clip them back to create an even level, the RJ45, plug without messing up the order To do so, hold the plug with the clip side facing away from you as shown.

→ Step-5: Next, push the cable right in the notch at the end of the plug needs to be just over the cable shielding and if isn't, that means that you stripped off too much shielding. Simple strip the cables back a little more

→ Step-6: After the wire are securely sitting inside the plug insert it into the crimping tool and push down

→ Step7: Lastly, Repeat for the other end using diagram B using diagram A

Result: Thus the cable connection in done & executed successfully.

29

Study of packet interface Tool & user Interface overview

EXP-3 Date: 30/12/

Aim:

To study the packet tracer tool installation and user interface

To understand environment of CISCO packet tracker to design simple network

Introduction:

A simulation as the name suggests simulates network devices and ios environment

It allows you to model complex systems without the need for dedicated equipment. It helps you to practice your network configuration & trouble shooting skills via computer.

It is available for both the linux & window desktop environment.

Installing packet:

To download packet tracer go to https://www.netad.com & log in with your cisco networking academy credentials. then check the packet installation in windows is pretty simple and straightforward. The setup comes in a single file named as setup6.0.1.exe. Open. This file to begin the setup wizard, accept the license agreement, choose a location and start the installation.

Linux:

Linux users with an Libuntu/Debain distribution should download the file for ubuntu and those using fedoral (Redhat/must download the file for fedora.

chmod+x packet Tracer 601-i386-installer -vpn
bin :/ packet Tracer 601-i386-installer rpm-bi

Member: This is a common menu found in all software applications, it is used to open, save, print.

This bar provides shortcut to menu options that are commonly accessed, such as open, save, zoom undo & redo and on the right.

Logical / Physical Workspace Tabs: These tabs allow you to toggle between the logical physical workspace.

Workspace: This is the area where topologies are created and simultaneously displayed

Common tool bar: This tool bar provides controls the manipulating topologies.

Real time: These tabs are used to toggle between the real & simulation modes. Buttons are provided

Network computer interface: This component contains all at the network end end devices available with packet tracer and is further divided into two areas.

User related packet tracer: User can create highly customized packets to test their topology from this area, and the results are displayed on a list.

Analyse the behaviour of network devices using CISCO packet tracer simulation.

From the network component or click and drag and drop. The below components.

a) A generic PCs & One HUB
b) A generic Pcs & one switch

click on connections:

click on copper straight through cable select one of the pc and connect it similarly connect 4 PCs to the switch using copper straight-through case. click on the PC connected to go to desktop, click IP configuration. The default gateway and DNS server information is not needed as there are only two end devices in map.

click on the PDU from the one of PC's then drop it on another PC connected to HUB.

observe the flow at PDU from source PC to destination PC. Repeat Step #3 to step #5 connected switch. observe how hub & switch forwarding the PDU & write your observation.

20/

Result: The study of packet tracer tool user interface installation has been done successfully.

Setup and configure a LAN using a switch and ethernet cables in your lab

Date: 9/8/24

EXP-4

Aim: Set up configure a LAN (Local area network) using switch and ethernet cables.

LAN: It refers a network connected devices within a limited area such as an office boarding, school etc. It enables users to share resources including data printers and internet access. LAN connect devices to promote collaboration and information between users. It serves as primary connecting device, managing & directing communications within a local network. Each connected device on a LAN switch can communicate with each other, allowing for fast, secure data transfer.

Set up LAN:

1) plan an design an appropriate network topology taking into account network requirements and equipment location.

2) You can take 4 computers, a switch unit 8, 16, 24 ports which is sufficient for network of these extra sizes, 4 ethernet cable.

3) connect your computers to network switch via ethernet cable, which is as simple as plugging one end of the ethernet cable into your computer and the other end into your computer network switch

Step-4: Assign IP address to your PCs

1) log on to the client computer as Administrator as owner

2) click network in network connections

3) Right click Local area / Ethernet → Go to properties → select internet protocol (TCP/IPV4) → click on properties → select use the following ip address option and assign ip address.

PC1 - IP address: 10.1.1.1, subnetmask 255.0.0.0
PC2 - IP address: 10.1.1.2, subnetmask 255.0.0.0
PC3 - IP address: 10.1.1.3, subnetmask 255.0.0.0
PC4 - IP address: 10.1.1.4, subnetmask 255.0.0.0

Step-5: 1) connect your computer to switch: To access the switch's web interface, you will need to connect your computer to switch using ethernet cable

2) log into the web interface: open a web browser and enter the IP address of the switch into address bar. This should bring up to the login page for the switch's web interface enter username, password.

3) configure basic settings: once you're logged in, you will able to configure basic settings for switch.

4) Assign IP address as 10.1.1.5 subnet 255.0.0.0

Step-6: check the connectivity switch and other machine by using ping command in command prompt device.

Step-7 Select a folder → go to properties → click sharing tab → share it everyone in the same LAN

Step-8: Try to access the shared folder from other computers of the network.

Result:

IP: 192.168.0.1
Subnet: 255.255.255.0
IP: 192.168.0.2
subnet: 255.255.255.0
~~IP~~:
~~subnet~~:

Result:

Thus the setup & configure of LAN using switch & ethernet is successfully executed.

2018

Practical-4

* Aim: Experiment on packet capture tool whireshark

* packet sniffer sniff message send receive
  ① Store and display the content various practical.

* Wireshark: A network analysis tool formerly known as Ethereal capture packet in real time and display the human readable format used to inspect suspicious program network. capture network decode protect, protocol analyse problem

* Capturing and analysing packets using wireshark tool:
  To filter, capture view packet in wireshark to capture 100 packets from the ethernet: IEEE 802.3 LAN Interface and save it.

①

procedure: → Select local area connection wireshar
→ Go to capture → option
→ Select stop capture automatically after 100 packets
→ Then click start capture
→ save packets.

② Create a filter to display only ARP packets and inspect the packet.

Procedure: → Go to capture → o
→ Select stop capture automatically after 100 packet
→ search ARP packet search search bar
→ square packet

⑤ Create a filter to display only DNS packets & provide the follow graph

procedure: → Goto capture → option
→ Select stop capture automatically after 100
→ click start capture
→ Search DNS packet
→ Too see flow graph click → flow graph
→ save packet.

④ create a filter to display only HTTP packet
Procedur → Select local area connection wireshark
→ Go to capture → option
~~→ select stop Go to capture~~
→ select stop capture automatically after 100.
→ click start capture
→ search HTTP packet search bar
→ Save packet

⑤ Create filter to display only IP/ICMP packet and display inspect the packet:
→ Select local area network connection wireshark.
→ Go to capture → option
→ select stop capture automatically after 100
→ Then click start capture
→ Search ICMP/IP packet search bar
→ Save packet

⑥ Create a filter display only DHCP packet and inspect packet

Procedure: → select local area connection wireshark
→ Go to capture → option
→ Select stop capture automatically aft 100
→ click start capture
→ Search DHCP packet search bar
→ Save packets.

Students observation:

① What is promisous mode?
A: capture all network packet, not just those addressed to NIC.
② Does ARP Packet has transport layer header? explai
A: No, Transport layer header, operates at the data link layer
③ What transport layer protocol used by DNS
A: UDP (primarily used) & TCP (Large data)
④ What is broadcast IP address?
A: Address to send data to all device on local network.

1 Experiment: Study Of Various Network Commands.

1.1) which command is used to find the reachability of host machine from your device.

(A) ping a hostname or IP address

Exp-5

Aim:
Experiment on packet capture tool: wireshark

Packet sniffer:
- Sniff Message sent/receive
- Store and display the content various protocol

Structure diagnostic tools: TCP dump
Wireshark

capturing & Analysing packets using wireshark tool:
procedure:
- Select local area connection wireshark
- Go to capture → option
- Select stop capture automatically after 100 packets
- Then click start capture
- Save the packets.

(1) create a filter to display only TCP/UDP packet, inspect the packet & provide flow graph

(2) create a filter to display only ARP packets and inspect the packets

(3) create filter to display only DNS packet & provide the flow graph

(4) create a filter to display only HTTP packet & inspect the packet

(5) create filter display only ICMP packet & inspect the packet.

Q create and filter display only DHCP packet & inspect packet

Students Observation

Q What is promiscous mode?
A capture all network packet, not just those addressed to the NIC

Q Does ARP packet has transport header?
A No, Transport layer, operates at the data link layer

Q What is the port number used by HTTP protocol?
A 80

Q What is broadcast IP address
A Address to send data to all device on local network

23/4

Result:
Thus the experiment on packet capture & inspected tool wireshark

Exp-6 Hamming Code

Aim:
Write a program to implement error detection & correction using hamming code concept. Make a test run to input data stream & verify error correction feature.

Error correction at Datalink layer:
Hamming code is a set of error correction code that can be used to detect and correct the errors that can occur when the data is transmitted from the sender to the receiver

create sender program with below features:

Receiver program should read the input from channel file

If there is no error, display the position of the error

else remove the redant bits & convert the binary data to ascii & display the output.

Students observation:
code:

```
def parityBits(ist):
    for pos in pos:p
        i = pos
    temp = []
    while (i<m+p):
        temp.extent(ist[i:pos+i+1])
        i=(pos+i+1)+pos+1
    count_ones = "".join(temp).count("1")
    if(count_ones % 2)==0
```

```
        lst[pos] = '0'
    else
        lst[pos] = '1'
return list
def togglebits(lst, ep):
    if (lst[ep] == '1'):
        lst[ep] = '0'
    else
        lst[ep] = '1'
    return ~~list~~ lst
def Printparity(lst)
    for i in pos-p
    print('p', p+1, ':', lst[i], sep=' ')
s= input("enter the message to encode:")
l = [bin(ord(c)) for c in s]
k = []
for i in range(len(l))
    l[i] = l[i].replace('0b', '')
    l[i] = l[i].zfill
    k.extend(list l[i])
    l = k
Print("senter message in Binary:", "".join(l))
m= len(l)
l = parityBits(l)
hamming code = "".join(l)
Print(hamming code)
receiver_code = toggleBits(receivercode, errorpos)
receivercode = list(hamming_code)
Printparity(Mreceiver code)
error_index = ""
pos_p.reverse()
for in in pos_p
```

```
    receiver_code POP(i)
print("Received Message in Binary:", receiver code)
for i in decodelist
    decodemsg = decoded_msg + chr(int(i,2))
print("Decoded Message at Receiver side:" decode
                                    - msg)
```

o/p

enter the message to encode : ~~Abhilash~~

Sender message in Binary : 01000001 01100010
01101001

No. of parity bits : 6

Parity bits for sent message

$P_1 = 1$

$P_2 = 1$

$P_4 = 0$

$P_8 = 0$

$P_{16} = 1$

$P_{32} = 0$

Enter the position to change the bit : 6

Parity bits for received message :

$P_1 = 0$

$P_2 = 1$

$P_3 = 1$

$P_4 = 0$

$P_5 = 0$

$P_6 = 0$

error detected and corrected at position 6

Decoded message at receiver side : AB!

Result:

Thus the program to implement error detection & connection using Hamming code was executed and the output was verified.

EXP-7 Practical-7

Write a program to implement flow control as datalink layer using sliding Window protocol Simulate the flow of frames from one node to another.

create a sender program with following features:
1. Input window size from the user
2. Input a text message from the user
3. consider one character per frame
4. create a frame with following fields [frame no, data]
5. Send the frames
6. Wait for the acknowledgement from the receiver
7. Reader a file called receiver buffer
8. check ACK filled for acknowledgment number
9. If the acknowledgment no. is expected send new set of frames accordingly

create a receive file with following features
1. reader a file called sender - buffer
2. check the frame no
3. If the frame no. are as expected write the appropriate ack no. in the receiver - buffer file

Program code:

```
import time
import random
class frame:
    def __init__(self, frame_no, data):
        self.data = data
```

```
        self.acknowledged = False
def send_frames(frames, window_size):
    print("\n--- sending frames ---")
    for i in range(window_size):
        if i < len(frames) and not frames[i].
            acknowledgment
            Print(f"sent frame {frames[i] frames[i]
                frame_no}: {frames[i].data}")
    Print("frames sent, waiting for acknowledgments
                \n")

def receive_frames(frames, window_size):
    Print("\n--receiving frames --")
    for i in range(window_size):
        if i<len(frames) and not frames[i].
            acknowledgment
            if random.random()<0.2:
                print(f"received frame {frames[i].
frame_no}: {frame[i] data} [Error]")
                frames[i].acknowledgment = False
        else
            print(f"Received frame {frames[i].frame_
                no}: {frames[i].data} [OK]")
            frames[i].acknowledged = True

def sliding_window_protocol():
window_size = int(input("Enter without
    Size:"))
message = input("Enter a message to send:")
frames = [Frame(i, message[i] for i in
range(len(message))
    base = 0
while base < len(frames):
```

```
    send_frame(frame[base]: window_size)
    time.sleep(2)
    receive_frames(frame) and frames[base].
                                acknowledged

        based t=1
        if based <len(frames):
        Print("\n Resending unacknowledged frame,
            time.sleep(2)
    print("\n All frames sent and acknowledged:")
        if __name__ == "__main__":
            sliding_window_protocol()
```

o/p:

```
Enter the window size: 3
Enter a message to send: Protocol
---Sending frames---
sent Frame 0: P
sent Frame 1: r
Sent Frame 2: o

frames sent, waiting for acknowledgment...
---Receive frame---
Received 0: P [OK]
Received 1: r [Error]
Received 2: o [OK]

Resending unacknowledged frame...
---Sending Frame---
Sent Frame 1: r
Sent Frame 3: t
Sent Frame sent, waiting for acknowledgment
```

--- Receiving frames ---
Received frames 1: r [Error]
Received Frames 3: t [Error]

Resending unacknowledged frames...
--- Sending Frames ---
Sent Frame 1: r
Sent Frame 3: t
Frames sent, waiting for acknowledged
--- Receiving Frames ---
Received Frame 1: r [OK]
Received Frame 3: t [OK]
Resending unacknowledged frames
--- Sending frames ---
sent frame 4: 0
sent frame 5: c
sent frame 6: 0
Frames sent, waiting for acknowledgment
--- Receiving frames ---
Resending unacknowledged frame ---
--- Sending Frames ---
Sent Frame 7: 1
Frames sent, waited for acknowledged !
--- Received frames ---
received frame 7: 1 [OK]
All frame sent and acknowledged !

Result:

Thus, the program to implement sliding window protocol is executed successfully.

Practical 8

Aim:
a) simulate Virtual LAN configuration using cisco packet tracer

Objectives:
Part-1: Build the network and configure Basic device settings
Part-2: Create VLAN and assign and assign switch ports
Part-3: Maintain VLAN Port assignments and the VLAN Database
Part-4: Configure and 802.1Q Trunk between switches

Instructions:
Part-1: Build the network and configure basic device settings
Part-2: Build the network as shown in the topology
a. click and drag both switch S1 and S2 to the back.
b. click and drag both PC-A and PC-B to the table and use the power button to turn them on
c. Provide network connectivity
d. Connect console cables to the device.

Step-2: Configure basic settings for each switch
a. From the desktop tab on each PC, use the terminal to console into each switch
b. Enter configuration mode
c. Assign a device name to each switch
d. Assign class as the privileged EXEC encrypted password
e. Assign a cisco as console
f) Assign cisco as vty

g) encrypt the plain text words.
h) create a banner that warn anyone accessing the device
i) configure the IP address listed in the address table for VLAN on the switch
j) shutdown all the interfaces that will not be used
k) Set the clock on each switch
l) Close configuration window

Step-3: configure PC cost
Step-4: Test connectivity

Part-2: Create VLAN and assign switch ports
Step-1: create VLAN on switches
a) create the VLAN on S1
b) create the same VLAN on S2
c) Issue the show VLAN brief command to view the list of VLANs on S2

Step-2: Assign VLANs to the correct switch interfaces. Assign VLANs to the interfaces on S1
1) Assign PC-A to operation VLAN
2) From VLAN1, remove the management IP address and configure it on VLAN99
→ Issue the show VLAN brief command and verify that VLANs are designed to correct interface
→ Issue the show IP interface brief command
→ Assign PC-B to the operations VLAN on S2
→ From VLAN1, remove the management IP address and configure it on VLAN99

Part-3: Maintain VLAN PORT assignments and the VLAN database
Step-1: Assign a VLAN to multiple interfaces. From the desktop tab on each PC, use terminal to continue configuring both network switches
step-2: Remove a VLAN ID from the VLAN database
a. Add VLAN 30 to interface F0/24 without issuing the global VLAN command
b. Verify that the new VLAN is display in the VLAN table
c. Use the no VLAN 30 command to remove VLAN from the VLAN database

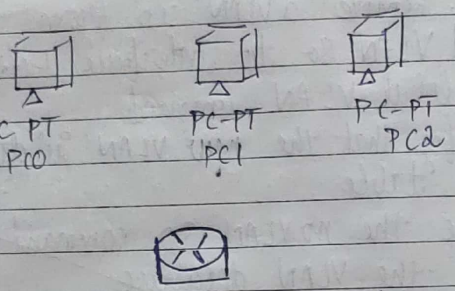
Part-4: Configure an 802.1Q trunk between the switches
Step-1: use DTP to initiat Trunking on F0/1
Step-2: Manually configure trunk interface

23/1/24

Result:
Simulating Virtual LAN configuration using cisco packet tracer executed successfully.

Practical-9

Aim:
configuration of wireless LAN using cisco packet tracer



TO complement these tasks follow these step by step instructions

ST-1: click wireless router
- select administration tab from top now set username and password to admin and click on save settings
- Next click on wireless tab and set default SSID to mother network
- Now select wireless security and change set default SSID to mother network
- Now select wireless security and change security mode to WEP
- Again go in the end of page and click on save setting

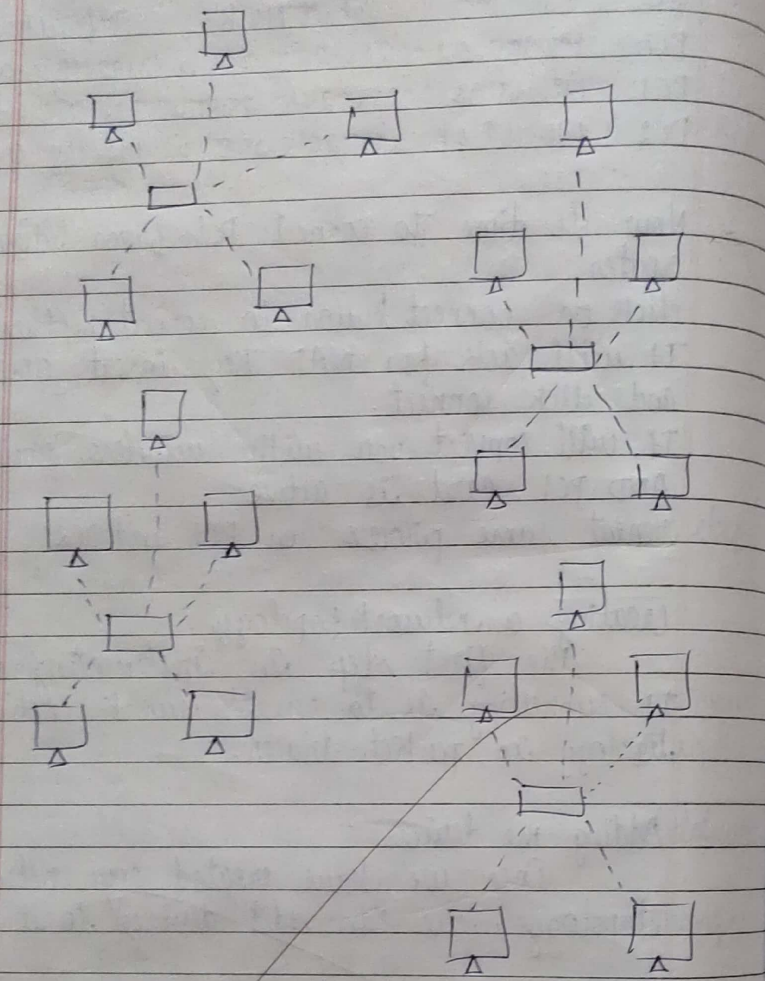
PC	IP	Subnet Mask	Default
PC0	192.168.0.2	255.255.255.0	192.168.0.1
PC1	192.168.03	255.255.255.0	192.168.0.1
PC2	192.168.0.4	255.25.5.255.0	192.168.0.1

Now its time to connect PC's from wireless router
click on connect button to connect mother network
It will ask for WAP key insert 0123456789 and click connect.
It will convert you with wireless router AND PC1 card is active
repeat same process on PC1 and PC2

creating a network topology:
The first step in implementing classes IP subnetting is to create work network topology in packet tracer.

Adding the devices:
Once we have created our network to topology, we can add devices to it

Subnetting:
TO subnet the network address of 192.168.1.0/24 to provide enough space for atleast 5 addresses for end devices, the switch, and the router, we can use a /27 subnet mask.



Configuring the devices:
Now that we have added our devices and connected them, we can start configuring them. This will open the command line interface (CLI) for the router. In the CLI enter the following commands:-

#enable
configure
interface faster o/o
ip address {IP address} {subnet mask}
exit.
interface fast ethernet 0/1
ip address {IP address} {subnet mask}
no shutdown
exit

Replace "{IP address}" and "{subnet mask}" with your desired IP addresses.
We will configure the switch. Right-click on the switch and select "CLI", enter the following commands

enable
configure terminal
interface Fast ethernet 0/1
switchport mode access
exit
interface Fast ethernet 0/2
switchport mode access
exit

TO configure the gigabyte ethernet faces on the router:
1) Right-click on router & select all
2) enter the following commands.

Testing the network: open a command prompt on each PC and try to ping it successfully.

then the network is functional properly. You can also use the "ping" command to test connectivity between the router and the PCs.

Result:

configuring a wireless LAN using CISCO is executed successfully.

Practical - 10

Aim:

a) Internetworking with routers in CISCO Packet Tracer simulator

In this network, a router and 2 PCs are used computer are connected with router using a copper network to check network connectivity a simple PDU transfered from PC0 to PC1

Procedure:

Step-1 (Configuring Router 1):

1. Select the router and CLI
2. Press ENTER to start configuration
3. Type enable to activate the privilaged mode

Step-2:

1. Assign IP addresses to every PC in the network
2. Select the PC, Go to the desktop and move
3. Assign the default gateway of PC0 as 192.168.10.1
4. Assign the default gateway of PC1 as 192.168.10.1

Step-3 (Connecting 0/0 port of router)

1. Fastethernet 0/0 port of router1
2. Fasternet 0/1 port of router1

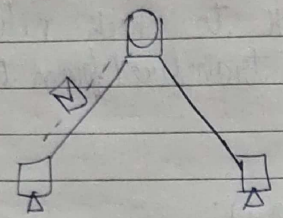
Router configuration Table

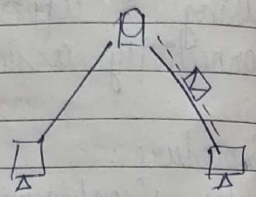
Device name	IP addresses Fastetherent	Subnet Mask	IP Address Fastether	Subnet Mask
Router1	192.168.10.1	255.255.255.0	-net	255.255.255.0

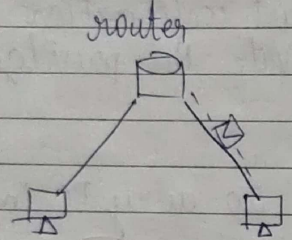
PC configuration Table:

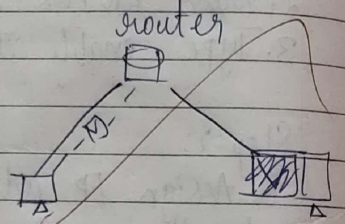
Device Name	IP address	subnet Mask	Gateway
PC1	192.168.10.2	255.255.255.0	192.168.10.1
PC01	192.168.10.2	255.255.255.0	192.168.20.1

PC1



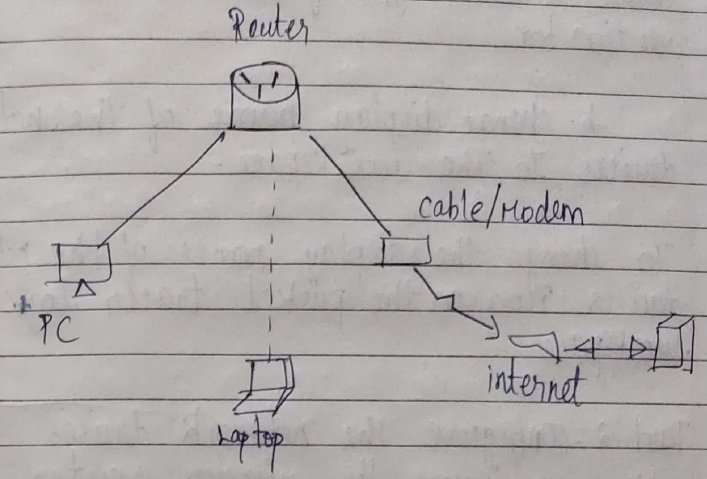






Aim:

b) Design and configure an interwork using wireless router, DHCP server and internet cloud



Addressing Table:

Device	Interface	IP address	Subnet mask	Default
PC	ethernet0	DHCP		192.168.0.1
wireless router	LAN	192.168.0.1	255.255.255.0	
Wireless router	Internet	DHCP		
CISCO.com	Ethernet0	208.67.220.220	255.255.255.0	
server		270		
laptop	wireless	DHCP		

Objectives:-

part 1: Building a simple network in the logical topology workspace

Step-2: Building the topology

a. Add network devices to the workspace. To place a device onto the workspace, first choose a device type from the device-type selection box

b. change display name of the network devices to the workspace.

To change the display names of the network devices icon on the packet tracer logical workspace

Part-2: configure the network devices

Step-1:- configure the wireless router

a. create the wireless network

b. click on the save setting tab

Step-2: configure the laptop

a. configure the laptop to access the wireless network.

Step-3: configure the PC

a) configure the PC

Step-4: a) install network modules if necessary

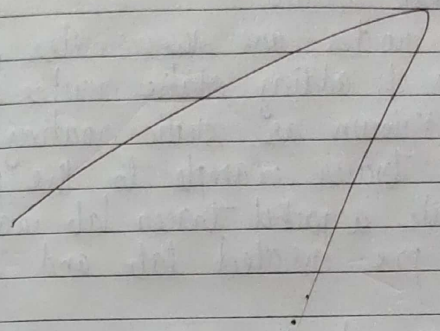
b) Identify the type of provider

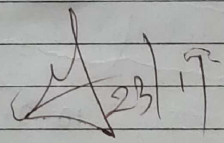
c) Identify the type of provider

Step-5: a) configure the CISCO.com server

b) configure the CISCO.com server to provide domain

c) configure the cisco.com server global settings.





Result:

Internetworking with routers in CISCO simulator is executed successfully

Practical 11

Aim:
a) Simulate static routing configuration using cisco packet tracer

Static routes are the routes, routing table the process of adding static routes to the table is known as static routing. Let's take a packet tracer example to the practical lab.

Create a packet tracer lab image or download pre-created lab and load it packet tracer.

IP route 30.0.0.0 255.0.0.0 20.0.0.2
ip 30.0.0. 255.0.0.0 40.0.0.20.20.

If 1st route fails the router automatically all the second route to the routaing table.

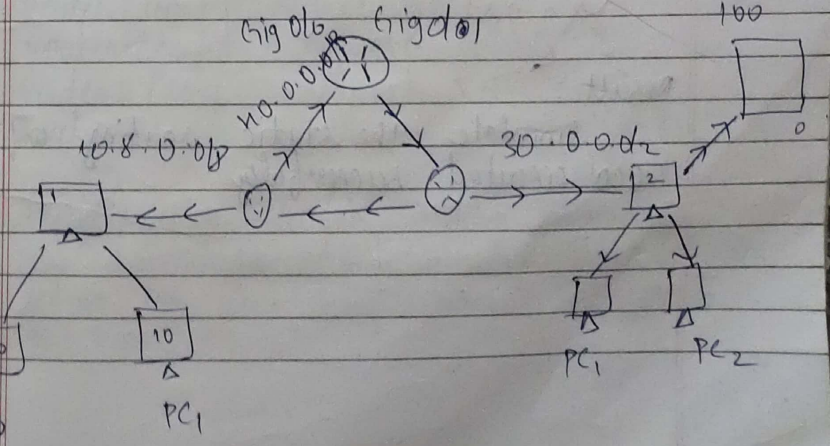
Router	Available networks on local interface	Network available on outer routers
0	10.0.0.0/8 20.0.0.0/8 40.0.0.0/8	30.0.0.0/8 50.0.0.0/8
1	20.0.0.0/8 30.0.0.0/8 50.0.0.0/8	10.0.0.0/8 40.0.0.0/8
2	40.0.0.0/8 50.0.0.0/8	10.0.0.0/8 20.0.0.0/8

Router 0 requirments:
create three routers for network 30.0.0.0/8 and config the first route as the main and configuration.

Router configuration:
router > enable
router# configure terminal
enter configuration command, one per line end with CNTL/Z

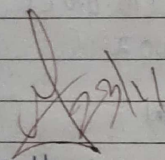
Router (config)# ip route 30.0.00.285.00.0.20.0.0.10
Router (config)# ip route 30.0.0.1.100 255.255.255.255.110.0.021
Router config # ip route 50.0.0.0 255.255.0.0.0 40.0.2.0
Router config # ip route 30.0.0.0 20.0.0.200 router (config) exit

Router # show ip route static
S 30.0.0/8 [10/0] via 20.0.02
30.0.0.100/32 [10/0] via (10.0.02)
80.0.0.0/8 [10/0] via 40.0.2



To delete a static route, as the following

* Use the show IP route static steps command to print all static routes
* use no ip routes
* If you have backup route, the backup becomes the main route when delete.



Result:

Simulate the static routing configuring cisco executed successfully

Practical - 11

Aim:

b) Simulate RIP using ISO packet traces

Device	Interface	IP configuration	connected with
PC0	Fast ethernet	10.0.0.218	Routers Fa0/1
R0	Fa 0/1	10.0.0.1/8	PC0's fastethernet
R0	S0/0/1	172.16.8.1.256/30	Router 2's S0/0/1
R0	S0/0/0	192.168.1256/30	Router 3 0/0/0
R1	S0/0/0	192.168.1256/30	Router 3 0/0/1
R1	S0/0/1	192.168.1256/30	Router 3 0/0/1
R2	S0/0/0	192.168.1 258/30	Router 3 0/0/1
R2	S0/0/1	192.168.1 243/30	Router 3 0/0/1
R2	Fa 0/1/1	20.0.0.1/30	PC1's Fast ethernet
PC1	Fast ethernet	20.0.0.2/30	router Fa0/1

We need to configure IP address and other parameters on the interface before we could actualy use node can be accessed from global configuration.

Router1 (R1):

Router1 (config) # route RIP
Router1 (config) # network 192.168.1.2.2HH
Router1 (config) # network 192.68.1.248
Router2 (config) # network RIP
Router2 (config) # network 192.16K.1.252.

Thanks our network ready to ready to take the advantages of RIP routing. To Verify the set up we use ping commands used to test connectivity.

Access command prompt of PC1, Ping RIP protocol automatically manage all routes. If one route goes down it automatically switches.
Now suppose route 1 is done, we can simulate the simulation by removing the cable attaches using trace command again to see the magic of dynamic routing.

Result:
Simulate RIP using 130 Packet Tracer executed successfully.

Practical - 12

Aim:
To implement each client saves TCP/int.

Algorithm:
* create TCP socket
* Built socket to local address port.
* listen for incoming client connections
* Loop
* receive data from client information
* If data is required received, conn is, base clien

TCP client algorithm:
* create TCP socket
* connect to server using specified addres

Port.
* Send message to serve
* receive the existing message from server
* Display received message
* close the socket

TCP client.py
```
import socket
def TCP-client ():
client-socket = socket.(Socket.AF-Inet,
                        socket, sock-stream)
client-socket = connect (enter message to send:)
client-socket.send all (Onemsage, encoded)
client = client_socket, rec (1024)
print (f"Received from server: (data, decode)
finally
    client.socket.close ()
if name == "main"
```

TCP, server.py

```
import socket
def TCP_server():
    server_socket = socket.socket(socket.socket(
        family, socket.SOCK_STREAM)
    server_socket.bind('localHost', 12345).
    server_socket.listen(2)
    print("TCP server is waiting for connection
    connection, client_address = server_socket.
    accepts
    print("connected to client_address))

try:
    while True:
        data = connection.recv(102 4)
    if data:
        print("received = {data.encode('Y')
connection.send all(data)" echo because
        received data
    else
        break

finally: connection.close() if __name__ == "__main__": tcpserver()
```

O/P: enter message to echo : letter
enter message to letter : letter.

Result:
implementing client services using UDP/TCP is executed successfully.

Practical - 12b

Aim:
To implement the chatclient serves using TCP/UDP serves.

Algorithm :-
chat server:

1. start the server by creating a secret bind to a specific address and past listen for connection
2. When a new chat connects add client a list of connected clients start a new process to talk to the clients
3. for each connected client keep shocking a new messages
4. If a client disconnects remove that client from the list & stop sharing to client
5. keep running to the process till the server stop

chat-client:

1) connect to the server by creating a secret and connect it to server address & port
2) start a process to listen to messages from server.
3) keep asking for new message from the server.
4) keep running till the user decides to quit

chat client.py

```
import secret
import threading
```

```
def receive_message (client socket)
    while true
try:
    message_client_socked_recv(1024)
- decode if message
        print (if "server = {message,}");
except exception are
    print(f"an error occured: {e}")
def start client()
    client_socket = socket, socket (secret
        netserver lhost = 127.0.0.1'
        port = 12345
client_socket.connects (host, port)
print ("connected to chat server")
while trunk
    message_input ("you")
client_socket_send (message, encode diff).
if name_:="__main__";
    start_client()
```

o/p: python chat > server.py
chat server stabled on 127.0.0:2348
new connection from ('127.0.0.1,52226)
received from message ~~to client~~
: Abhilash
Type from message to client: Hello

Python chat client
connected to chat server
you: Abhilash
you: server hello.

Result: Implementation of chat client using TCP/UDP socket has executed successfully

Practical - 13

Aim:
Implemented you run ping program.

Algorithm:
Ping - client.py
1) To set creation
2) The set a timeout of 2 second to ensure and if no suspons is received it will stop waiting and peal request timeout
3) Sends a ping msg do spatied hose 2 part.
4) It listens for a response a calculates the time difference b/w sending and recursing the packet

ping - server.py
1. Intialize ISDP sonel
2. Blind to IP address and proof
3. Listens for incoming messages
4. Receive data
5. Send response

Program:
Ping - server.py
import socket
diff start. service (nast 127.0.0.1 post 12345)
with socat. soant (AF that save Dev)
skind (host, pant)
print (f' top source running one {port})
While time :-
data add - receiver ()

If - name 1 main - :
start service
Ping client py
import time
def ping sour ce(host. 127.2.0.1 print 12345)
which secret select(socket AE thes, Socket
sece cofd en 1

-thing :-
set timeout (R)
start - time ture ()
4 bend to (b'ping', (host, port)
date, add - 5 her from (102H)
end = tune tune ()
print (t' Received 'data decide's from path's
Infuel - ford . 2 ()24
except send timeout
print ("Request find out")
if name 1 - main' :-
ping several ()

output:

enter the host to ping google.com pinging google.com
[142.250.42.46] with 32 bytes of date
reply from google.com (142:256.188.46):
bytes - 32 time =

Result:
The program to create and implement Ping message has been successfully created

Practical 14

Aim:
write a code doing section to implement packet sniffing

Algorithm:
1) Install python and scapy
2) create a program upen text editor and create a title in notepad collect packet snefferpy & code to capture & Analyse the netmic packet.
3) setup packet tracer by it the packet has ip layer identify packets protocol such as TCP, UDP, KMP
4) Run the packet snippen byprogram.

Program:
from scapy all import staff
from scapy . taylor . net import IP, TCP, UDP
def packet - call back (Packet):-
IP - layer - Pocket [IP]
protocol = IP - layer - proto
Src - IP = IP2 - layer - SVC
ds . P = IP layer - dst

Protocol name:
If protocol = 1:
protocol = 'Imp'
if protocol = 6
protocol = 6
protocol = name = 'TCP'
elif Protocol = 17
Protocol - name = 'UDP'

Use.
Protocol_name = "unknown protocol"
print(f" protocol: {protocol_name}")
print(f"screens IP scan = phy")
print(f" destination ip {dst_ip}")
print("-" 30)
def main()
Sniff (Pn=Packet_case back, filter="IP
store if __name__ "main=="
main()

O/P: python packet_sniffer.py
Protocol: TCP
Source IP: 23215.215.227
Destination IP 292.168.101.75
The program to implement packet sniffing done successfully

Result:
Thus Implementation of packet sniffing done successfully executed and verified.

Practical - 15

Aim:
To Analyse the different types of weblog using web-alizer tool.

Algorithm:-
Step-1: Run webalizer window version
Step-2: Input weblog file (download from web)
Step-3: press run webalizer

Result:
These the experiment to using webalizer for weblog analysis is executed and verified.