Exp 1 :- Netkit

* vstart pc1 --eth0=A
* vstart pc2 --eth1=B
* vstart r1 --eth0=A --eth1=B

i)pc1 commands:- 1)ifconfig eth0 192.168.0.2

2)route add default gw 192.168.0.1

3)ifconfig

4)route –e

ii)pc2 commands:- 1)ifconfig eth0 192.168.1.2

2)route add default gw 192.168.1.1

3)ifconfig

4)route –e

iii)r1 command:- 1)ifconfig eth0 192.168.0.1

2)ifconfig eth1 192.168.1.1

3)ping 192.168.1.2

EXP 2] AIM :  To explore static routing configuration through the Kathara emulator.

Steps for setting up kathara virtual lab.

* Download lab.conf, pc1.startup, pc2.startup, r1.startup and r2.startup
* Add Kathará public key to your keyring: sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys 21805A48E6CBBA6B991ABE76646193862B759810
* Add the Kathará repo to your repos: sudo add-apt-repository ppa:katharaframework/kathara
* Update your apt cache by running sudo apt update
* Install Kathará running sudo apt install kathara
* sudo apt install kathara
* Got to downloads
* kathara lstart

EXECUTION :

PC1 creation:

kathara vstart -n Pc1 --eth 0:A

ifconfig eth0 192.168.10.2

route add default gw 192.168.10.1

PC2 creation:

kathara vstart -n Pc2 –eth 0:B

ifconfig eth0 192.168.10.3

route add default gw 192.168.10.1

R1 creation:

kathara vstart -n R1 –eth 0:A

ifconfig eth0 192.168.10.1

EXP 3: Aim : -To study requirements and scope of Subnetting and Network Address translation by using Kathara Emulator

1)On terminal:

kathara vstart -n r1 --eth 0:A 1:B

On virtual machine:

ifconfig eth0 192.168.25.125/25

ifconfig eth1 192.168.25.253/25

2)On terminal:

kathara vstart -n pc1 --eth 0:A

On virtual machine:

ifconfig eth0 192.168.25.1/25

route add default gw 192.168.25.125

3)On terminal:

kathara vstart -n pc2 --eth 0:A

On virtual machine:

ifconfig eth0 192.168.25.126/25

route add default gw 192.168.25.125

4)On terminal:

kathara vstart -n pc3 --eth 0:B

On virtual machine:

ifconfig eth0 192.168.25.129/25

route add default gw 192.168.25.253

5)On terminal:

kathara vstart -n pc4 --eth 0:B

On virtual machine:

ifconfig eth0 192.168.25.254/25

route add default gw 192.168.25.253

Exp 4:-Wireshark

1. Sudo apt – get install vstpd
2. Sudo service vstpd restart
3. Ifconfig----🡪(idhr se jo ip address milega khudka voh ip address ftp pe assign krna hai)
4. ftp 192.168.0.1--->(upr se liya hua ip add)
5. open wireshark app
6. check traffic on ports by commands:-

* udp.port == 67🡪 for dhcp
* udp.port==53🡪for dns
* udp.port==21🡪for tcp
* ip.dst == 192.168.6.80🡪for specific ip add

EXP 5: AIM - To study three way handshaking process as well as working process for connection oriented Protocols like SSH, HTTP & analysing packets generated by using packet capturing tool like tcpdump

1. sudo tcpdump -D
2. sudo tcpdump --interface any
3. sudo tcpdump -i any -c5
4. sudo tcpdump -i any -c5 -nn
5. sudo tcpdump -i any -c5 --nn -v
6. sudo tcpdump -i any -c5 icmp
7. ping 192.168.92.10 [In Terminal 1]

sudo tcpdump -i any -c5 icmp [In Terminal 2]

1. sudo tcpdumb -I any port 23 > vedant.txt

**Exp 6** aim: **To study and implement network topologies that handle Connection Oriented (TCP) and Connection less (UDP) traffic by using NS2 Simulation Format.**

#Create a simulator object

set ns [new Simulator]

#Define different colors for data flows (for NAM)

$ns color 1 Blue

$ns color 2 Red

#Open the NAM trace file

set nf [open out.nam w]

$ns namtrace-all $nf

#Define a 'finish' procedure

proc finish {} {

global ns nf

$ns flush-trace

#Close the NAM trace file

close $nf

#Execute NAM on the trace file

exec nam out.nam &

exit 0

}

#Create four nodes

set n0 [$ns node]

set n1 [$ns node]

set n2 [$ns node]

set n3 [$ns node]

#Create links between the nodes

$ns duplex-link $n0 $n2 2Mb 10ms DropTail

$ns duplex-link $n1 $n2 2Mb 10ms DropTail

$ns duplex-link $n2 $n3 1.7Mb 20ms DropTail

#Set Queue Size of link (n2-n3) to 10

$ns queue-limit $n2 $n3 10

#Give node position (for NAM)

$ns duplex-link-op $n0 $n2 orient right-down

$ns duplex-link-op $n1 $n2 orient right-up

$ns duplex-link-op $n2 $n3 orient right

#Monitor the queue for link (n2-n3). (for NAM)

$ns duplex-link-op $n2 $n3 queuePos 0.5

#Setup a TCP connection

set tcp [new Agent/TCP]

$tcp set class\_ 2

$ns attach-agent $n0 $tcp

set sink [new Agent/TCPSink]

$ns attach-agent $n3 $sink

$ns connect $tcp $sink

$tcp set fid\_ 1

#Setup a FTP over TCP connection

set ftp [new Application/FTP]

$ftp attach-agent $tcp

$ftp set type\_ FTP

#Setup a UDP connection

set udp [new Agent/UDP]

$ns attach-agent $n1 $udp

set null [new Agent/Null]

$ns attach-agent $n3 $null

$ns connect $udp $null

$udp set fid\_ 2

#Setup a CBR over UDP connection

set cbr [new Application/Traffic/CBR]

$cbr attach-agent $udp

$cbr set type\_ CBR

$cbr set packet\_size\_ 1000

$cbr set rate\_ 1mb

$cbr set random\_ false

#Schedule events for the CBR and FTP agents

$ns at 0.1 "$cbr start"

$ns at 1.0 "$ftp start"

$ns at 4.0 "$ftp stop"

$ns at 4.5 "$cbr stop"

#Detach tcp and sink agents (not really necessary)

$ns at 4.5 "$ns detach-agent $n0 $tcp ; $ns detach-agent $n3 $sink"

#Call the finish procedure after 5 seconds of simulation time

$ns at 5.0 "finish"

#Print CBR packet size and interval

puts "CBR packet size = [$cbr set packet\_size\_]"

puts "CBR interval = [$cbr set interval\_]"

#Run the simulation

$ns run

**Steps1-->Save as name.tcl file**

**Step2🡪 gedit name.tcl**

**Step3🡪 ns name.tcl**

**Exp7 aim**: [**To study and analyze routing protocols with respect to QOS parameters Assignment**](http://moodle.apsit.org.in/moodle/mod/assign/view.php?id=170093) **OR** **To create simulation environment by using NS 2.35 that uses static or dynamic routing to route connection oriented and Connection less traffic generated over simulation Environment.**

Simulation program to genrate coneection oriented and connectionless trafic over given network which uses static and dynamic routing.

#Create a simulator object

set ns [new Simulator]

#Open the Trace file

set nr [open apsit.tr w]

$ns trace-all $nr

#Open the NAM trace file

set nf [open apsit.nam w]

$ns namtrace-all $nf

#Define a 'finish' procedure

proc finish { } {

global ns nr nf

$ns flush-trace

close $nf

close $nr

exec nam apsit.nam &

exit 0

}

#Create nodes byusing looping constructs.

for { set i 0 } { $i < 12} { incr i 1 } {

set n($i) [$ns node]}

#Create links between the nodes

for {set i 0} {$i < 8} {incr i} {

$ns duplex-link $n($i) $n([expr $i+1]) 2.0Mb 10ms DropTail }

$ns duplex-link $n(0) $n(8) 2.0Mb 10ms DropTail

$ns duplex-link $n(1) $n(10) 2.0Mb 10ms DropTail

$ns duplex-link $n(0) $n(9) 2.0Mb 10ms DropTail

$ns duplex-link $n(9) $n(11) 2.0Mb 10ms DropTail

$ns duplex-link $n(10) $n(11) 2.0Mb 10ms DropTail

$ns duplex-link $n(11) $n(5) 2.0Mb 10ms DropTail

#Setup a udp connection

set udp0 [new Agent/UDP]

$ns attach-agent $n(0) $udp0

#Setup a CBR over UDP connection

set cbr0 [new Application/Traffic/CBR]

$cbr0 set packetSize\_ 500

$cbr0 set interval\_ 0.005

$cbr0 attach-agent $udp0

set null0 [new Agent/Null]

$ns attach-agent $n(5) $null0

$ns connect $udp0 $null0

#Setup a TCP connection

set tcp [new Agent/TCP]

$ns attach-agent $n(1) $tcp

set sink [new Agent/TCPSink]

$ns attach-agent $n(5) $sink

$ns connect $tcp $sink

#Setup a FTP over TCP connection

set ftp [new Application/FTP]

$ftp attach-agent $tcp

#Dynamic Routing by using Distance Vector Routing Protocol

$ns rtproto DV

#Schedule events for the CBR and FTP agents

$ns rtmodel-at 3.0 down $n(11) $n(5)

$ns rtmodel-at 5.0 down $n(7) $n(6)

$ns rtmodel-at 7.0 up $n(11) $n(5)

$ns rtmodel-at 9.0 up $n(7) $n(6)

#Set Flow ID

$udp0 set fid\_ 1

$tcp set fid\_ 2

#Set Colours to traffic Flows

$ns color 1 Red

$ns color 2 Blue

$ns at 1.0 "$cbr0 start"

$ns at 2.0 "$ftp start"

#Call the finish procedure after 12 seconds of simulation time

$ns at 12 "finish"

#Run the simulation

$ns run

**Steps1-->Save as name.tcl file**

**Step2🡪 gedit name.tcl**

**Step3🡪 ns name.tcl**

**Exp 8 aim :** -**To study configuration of remote login services such as TELNET and SSH by using basics of networking.**

**Telnet Servive**

Step 1: Check the Availability of Telnet

$ apt show telnetd

Step 2: Install Telnet

$ sudo apt install telnetd -y

Step 3: Check the Telnet’s Status

$ sudo systemctl status inetd

Step 4: Configure Firewall for Telnet

$ sudo ufw enable

$ sudo ufw allow 23

Step 5: Test Telnet Server

$ telnet 192.168.1.25 or 96.20

It will ask for the login credentials, once provided, it will successfully access the other Linux machine:

You may come out of the Telnet’s session using the command:

$ exit

**SSH Service**

sudo apt update  
sudo apt upgrade

To install openssh-server package, run:

sudo apt-get install openssh-server

**enable the ssh server and start it as follows by typing the systemctl command:**

sudo systemctl enable ssh

sudo systemctl start ssh

### Configure firewall and open port 22

udo ufw allow ssh  
sudo ufw enable  
sudo ufw status

### Test it

Ssh localhost

ssh [apsit @192.168.1.105](mailto:apsit@192.168.1.105)

**Exp9Aim: To study use of CISCO packet tracer for designing organization network.**

1]To run cisco packet tracer in terminal:  **packettracer**

2]on router :

enable

config t

hostname router0

Interface GigabitEthernet0/0

ip address 192.168.0.1 255.255.255.0

no shutdown

Interface GigabitEthernet0/1

ip address 192.168.1.1 255.255.255.0

no shutdown

exit

**EXP 10:- UDP USING SOCKET API**

i)on server terminal

1)sudo su root

2)cd Downloads

3)its gcc –o user user.c OR cc -o user.c OR cc -o user user.c

4)./user

ii)On client terminal

1)sudo su root

2)cd Downloads

3)its gcc –o ucln ucln.c OR cc -o ucln.c OR cc-o ucln ucln.c

4)./ucln

iii) on 3rd terminal

1)sudo su root

2)cd Downloads

3)netstat –aunl-p-c |grep 8080

OR

Sudo tcpdump -vvv -n -e -I lo port 8080

Exp 11(A) To study and implement TCP client and TCP server.

RUN ON SERVER TERMINAL

1)Sudo su root

2)cd downloads

3)cc -o server server.c

4)./server

RUN ON CLIENT TERMINAL

1)sudo su root

2)cd downloads

3)cc -o client client.c

4)./client

RUN ON 3RD TERMINAL

1)sudo su root

2)netstat -atnl -c | grep 7891

Exp 11(B)

SERVER TERMINAL

1)sudo su root

2)cd downloads

3)cc -o forks forks.c

4)./forks

CLIENT TERMINAL

1)sudo su root

2)cd downloads

3)cc -o forkc forkc.c

4)./forkc

ON 3RD TERMINAL

1)sudo su root

2)cd downloads

3)netstat -atnl -c | grep 9734