**Routing with Desktop Computers**

# Exercises description:

In this exercise we will see **how to configure Linux based routers and then experience the forwarding of packets by these routers when data is exchanged between machines connected via these routers.** The basic network setup connectivity will look as given below



Figure 1‑2

We treat all the three networks as IPv4 networks and do the exercises using IPv4 address assignment and routing and seeing packet flow in IPv4. The logical view of IPv4 network will appear as follows:



Figure 1‑3

# Pre-requisite

## Pre-requisite

Ensure that systems are connected as shown in physical connectivity (). For machines shown as R1 and R2, please note down the proper interface names. To identify the interfaces, use the command

‘ip addr show | grep eth’

Note 1: Please ensure that motherboard interface is connected to the network corresponding to hosts Ha and Hb. If the name of this interface is different e.g. eth2 in this case, then use that interface name while doing configuration for networks connecting the hosts.

## Network Nomenclature

We will refer to Network between **Ha** and **R1** as **N1**, between **R1** and **R2** as **N2** and between **R2** and **Hb** as **N3**. Similarly, for the IP addresses of these interfaces, we will use the following nomenclature

A11 IP address of **Ha**

A12 IP address of **R1** interface connecting to **Ha**

A21 IP address of **R1** interface connecting to **R2**

A22 IP address of **R2** interface connecting to **R1**

A31 IP address of **R2** interface connecting to **Hb**

A32 IP address of **Hb**

# Setup-1 Configuring IPv4 Network

## Diagram for 1st setup with two routers all using IPv4

The addresses used in this diagram are as follows. You have to use your assigned IP addresses.

N1 – 172.16.1.0/24

N2 – 172.16.2.0/24

N3 – 172.16.3.0/24

A11 172.16.1.1/24

A12 172.16.1.201/24

A21 172.16.2.1/24

A22 172.16.2.201/24

A31 172.16.3.1/24

A32 172.16.3.201/24

**Your network interface has been defined using DHCP, so please change it to use as manual IP address and assign the respective address.**

The linux system requires that all the network configuration commands can be used with only super user (i.e. root) privileges. Thus, **all the commands should be used with sudo prefix.**

The command to remove an existing IP address is

sudo ip addr del **a.b.c.d/n** dev eth1

## Setup 1 configuration

### Ha Configuration

Add the IP address for the interface

$ sudo ip addr add 172.16.1.1/24 dev eth1

Define the routing table for network N2 and N3.

$ sudo ip route add 172.16.3.0/24 via 172.16.1.201 dev eth1

$ sudo ip route add 172.16.2.0/24 via 172.16.1.201 dev eth1

### Setup 1 – R1 Configuration

Add the IP addresses of the 2 interfaces

$ sudo ip addr add 172.16.1.201/24 dev eth1

$ sudo ip addr add 172.16.2.1/24 dev eth2

**Convert the multi-homed machine into the router.**

By default, a linux m/c does not forward packets from one interface to another. To enable forwarding, i.e. to convert it into a router for IPv4 networks, issue the following command. Ensure there is no space before and after the ‘=’

$ sudo sysctl –w net.ipv4.ip\_forward=1

Define the routing table for network N3 as below (It is directly connected to network N1 and N2 and thus no routing entries need to be added for these networks).

$ sudo ip route add 172.16.3.0/24 via 172.16.2.201

### Setup 1 – R2 Configuration

Add the IP addresses of the 2 interfaces

$ sudo ip addr add 172.16.2.201/24 dev eth2

$ sudo ip addr add 172.16.3.1/24 dev eth1

As in R1, convert the multi-homed machine into the router.

$ sudo sysctl –w net.ipv4.ip\_forward=1

Define the routing table for network N1.

$ sudo ip route add 172.16.1.0/24 via 172.16.2.1 dev eth2

### Setup 1 - Hb Configuration

Configure the network address

$ sudo ip addr add 172.16.3.201/24 dev eth1

Define the routing table for network N1, N2

$ sudo ip route add 172.16.1.0/24 via 172.16.3.1 dev eth1

$ sudo ip route add 172.16.2.0/24 via 172.16.3.1 dev eth1

## Verification of configuration

Issue the sudo ip addr show command to verify the network addresses and sudo ip route show to verify the routing entries.

## Steps to see the exercise in working

### Using ping

First check local reachability within the network.

From Host **Ha**, issue the following command to check router **R1** is reachable

ping –c2 172.16.1.201

then ping R1, R2 and Hb

### Using Netcat

Use netcat (nc) to exchange data between **Ha** and **Hb**.

Using nc:

From Host **Ha**, issue the command to start a tcp server listening on port 22222

nc –l 22222

From Host **Hb**, issue command to start the tcp client connecting to tcp server on **Hb**

nc <A11> 22222,

i.e. the actual command will be

nc 172.16.1.1 22222

Now type anything at Host **Ha**, it will be sent across thro the network to Host **Hb** and visible on its terminal.

### Using Browser and Server Communication

Ensure that Apache web server is running on **Hb.** Check using the following command

sudo service apache2 status

If web server is running it will display the following output.

Apache2 is running (pid nnnn).

If web server is not running it will display the following output.

Apache2 is NOT running.

In case it provides the output as “apache2: unrecognized service” you can install the same by issuing

sudo apt-get install apache2

Restart the apache by

sudo service apache2 restart

and verify that it is running

Now open the browser in **Ha**, and type the following URL.

http://[ <A32>], for the address <A32>, the actual URL will be

<http://172.16.3.201>.

## Post Lab activity:

## Verification exercise correctness

Run wireshark on **R1** and capture ICMP packets on both eth1 and eth2 interfaces. When it receives a ping request (ICMP ECHO Request), it will have a TTL value of 64. When the same packet is sent on other interface, the TTL value should decrease. Wireshark capture on **Hb.** should show TTL value as 62. Similarly, wireshark capture on eth1 interface of **R2** will show TTL value of 62, but wireshark capture on eth2 interface will show TTL value of 63. Since TTL is changing, you can also notice that checksum field also gets recalculated when a router sends out packet.

## Learning Summary:

By this exercise, we have learnt the following

1. Configuration of network interfaces
2. Configuration of routers
3. Understanding how TTL decreases with each router hop

## Additional Exercises:

1. Use a single /24 subnet mask and divide it into 3 networks (e.g. /26) and then configure the three networks with N1, N2 and N3
2. Use a single /24 subnet mask and use variable subnet mask for three networks. For example, for N1, use /25, for N2 use /27, and for N3 use /26.