MAWLANA BHASHANI SCIENCE AND TECHNOLOGY UNIVERSITY

Santosh,Tangail – 1902

Course Title: Computer Networks Lab

Lab Report Name: Networking in Linux

Lab Report no.: 01

Submitted by,

Name : ANIKA JAHIN

ID: IT-17056

Session: 2016-17

Dept. of ICT,MBSTU.

Submitted to,

NAZRUL ISLAM

Assistant Professor

Dept. of ICT,MBSTU.

Theory:

Computers are connected in a network to exchange information or resources each other.

Two or more computer connected through network media called computer network. There are number of network devices or media are involved to form computer network. Computer loaded with Linux Operating System can also be a part of network whether it is small or large network by its multitasking and multiuser natures. Maintaining of system and network up and running is a task of System / Network Administrator's job. In this article we are going to review frequently used network configuration and troubleshoot commands in Linux.

1) Introduction

If you have a network that ranges from 192.168.1.0 to 192.168.1.255 explain why Individual devices in the network can only be assigned IP addresses in the range of 192.168.1.1 to 192.168.1.254.

Answer: If your IP address ranges from 192.168.1.0 to 192.168.1.255 then you are connected to a private network. Having a router is having a private network. Home routers have their local address set to a default, private IP address number. It's usually the same address for the other models from that manufacturer, and it can be seen in the manufacturer's documentation.

Here's a look at the default private (also called "local") IP addresses for popular brands of routers:

- **Linksys** routers use 192.168.1.1
- **D-Link** and **NETGEAR** routers are set to 192.168.0.1
- **Cisco** routers use either 192.168.10.2, 192.168.1.254 or 192.168.1.1
- **Belkin** and **SMC** routers often use 192.168.2.1

192.168.1.254 is a Private IP address, one of the addresses for private networks. This means that a

device in this private network cannot be accessed directly from the internet using a Private IP, but by
any other device on the local network.

While the router itself has a Private IP of 192.168.1.254, it assigns any device in the network a different private IP address. All IP addresses on the network must have a unique address on that network to avoid IP address conflicts. Ipv4-addresses are internally 32 bits, they're often divided into 4 groups of 8 bits.
An octet can only be variety from 0 – 255, so as that leaves 256 possibilities for that last number. All addresses within the range of 192.168.1.0 to 192.168.1.255 are within an equivalent network. There are only 254 possibilities for variety. The addresses 192.168.1.0 and 192.168.1.255 are reserved for the network. 192.168.1.0, is reserved for the “network address.”

192.168.1.255, is that the “broadcast” address.

In an IP address, you've some dedicated to the network and a few of the address dedicated to the hosts. During a /24 network, meaning the first 3 octets are for the network.

192.168.1.0 is the subsequent in binary:

11000000.10101000.00000001.00000000

A /24 subnet mask in binary looks like this:

11111111.11111111.11111111.00000000

In decimal, this is: 255.255.255.0, so the first usable address is 192.168.1.1 and thus the last is 192.168.1.254. Since all devices within the network need to have unique addresses meaning that you simply can have 254 devices therein network. Other Private IP addresses used by the modem and router are 192.168.1.100 and 192.168.1.101.

2) **Find IP & MAC**

Find out about network and hardware information for the computer you are currently using.

Answer: The hardware and network information of my computer is,

```
anika@anika-VirtualBox:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.2.15  netmask 255.255.255.0  broadcast 10.0.2.255
        inet6 fe80::d04d:cb3e:2101:b7ed  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:aa:2c:bc  txqueuelen 1000  (Ethernet)
        RX packets 17350  bytes 24759623 (24.7 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 4629  bytes 361638 (361.6 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 288  bytes 24302 (24.3 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 288  bytes 24302 (24.3 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

anika@anika-VirtualBox:~$ hostname -i
127.0.1.1
```

3) Routing Table basics

Answer: The $ netstat –r command,

```
anika@anika-VirtualBox:~$ netstat -r
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
default         _gateway        0.0.0.0         UG        0 0          0 enp0s3
10.0.2.0        0.0.0.0         255.255.255.0   U         0 0          0 enp0s3
link-local      0.0.0.0         255.255.0.0     U         0 0          0 enp0s3
```

The Destination column identifies the destination network. The Gateway column identifies the defined gateway for the specified network. An asterisk (*) appears in this column if no forwarding gateway is needed for the network. The Genmask column shows the *netmask* for the network; in this case, it is 255.255.255.0. The Iface column shows the network interface. If you had more than one interface, you would see *lo* (for loopback), *eth0* (first Ethernet device), and *eth1* (for the second Ethernet device), and so on for the number of interfaces you have installed. Under the Flags section, the *U* flag means the route is up, and the *G* flag means that specified gateway should be used for this route. There are other flags that you may see, which include: *D* for dynamically installed, *M* for modified, and *R* for reinstated. These three flags indicate that the route was created or modified by a routing daemon after encountering an ICMP Redirect message. (Usually, you won't see

these flags unless you use dynamic routing.) Finally, you may see a *!* flag, which indicates a rejecting route.

The MSS column indicates the default Maximum Segment Size for TCP connections over this route. The Window column indicates the default window size for TCP connections over this route, and the Irtt column indicates the Initial Round Trip Time for this route. The kernel uses this to select values for certain TCP parameters without having to wait for potentially slow answers from remote hosts. These three columns you will normally not need to worry about unless performance is suffering and you're trying to find a way to tweak it. Under most circumstances, you will not redefine the defaults here.

4) Virtual Interfaces

a) Create a new virtual interface with the following IP address 192.168.2.32 and netmask 255.255.255.0 then check to see if the interface was created successfully?

Answer:

```
anika@anika-VirtualBox:~$ sudo ifconfig enp0s3 192.168.2.32 netmask 255.255.255
.0
anika@anika-VirtualBox:~$
anika@anika-VirtualBox:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.2.32  netmask 255.255.255.0  broadcast 192.168.2.255
        inet6 fe80::d04d:cb3e:2101:b7ed  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:aa:2c:bc  txqueuelen 1000  (Ethernet)
        RX packets 26814  bytes 38418935 (38.4 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 6499  bytes 487184 (487.1 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 365  bytes 31191 (31.1 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 365  bytes 31191 (31.1 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

b) **You need to set up a route for this interface so that your computer can see it. Issue the needed command, then issue the "$ netstat -r" command and check if the route to your added interface is visible?**

Answer:

```
anika@anika-VirtualBox:~$ sudo ip route add default via 192.168.2.32 dev enp0s3
RTNETLINK answers: File exists
anika@anika-VirtualBox:~$ ip route show
default via 192.168.2.32 dev enp0s3
192.168.2.0/24 dev enp0s3 proto kernel scope link src 192.168.2.32 metric 100
anika@anika-VirtualBox:~$ route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
default         anika-VirtualBo 0.0.0.0         UG    0      0        0 enp0s3
192.168.2.0     0.0.0.0         255.255.255.0   U     100    0        0 enp0s3
anika@anika-VirtualBox:~$ netstat -r
Kernel IP routing table
```

```
anika@anika-VirtualBox:~$ ip route show
default via 192.168.2.32 dev enp0s3
192.168.2.0/24 dev enp0s3 proto kernel scope link src 192.168.2.32 metric 100
anika@anika-VirtualBox:~$ route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
default         anika-VirtualBo 0.0.0.0         UG    0      0        0 enp0s3
192.168.2.0     0.0.0.0         255.255.255.0   U     100    0        0 enp0s3
anika@anika-VirtualBox:~$ netstat -r
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
default         anika-VirtualBo 0.0.0.0         UG        0 0          0 enp0s3
192.168.2.0     0.0.0.0         255.255.255.0   U         0 0          0 enp0s3
anika@anika-VirtualBox:~$
anika@anika-VirtualBox:~$ netstat r
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
udp        0      0 anika-VirtualBox:38478  8.8.8.8:domain          ESTABLISHED
udp        0      0 anika-VirtualBox:33908  8.8.8.8:domain          ESTABLISHED
udp        0      0 anika-VirtualBox:56027  8.8.8.8:domain          ESTABLISHED
Active UNIX domain sockets (w/o servers)
Proto RefCnt Flags       Type       State         I-Node   Path
unix  17     [ ]         DGRAM                    12687    /run/systemd/journal
/dev-log
unix  2      [ ]         DGRAM                    21518    /run/user/1000/syste
md/notify
unix  2      [ ]         DGRAM                    12795    /run/systemd/journal
```

c)

Next remove the route for this interface?

Answer:

```
anika@anika-VirtualBox:~$ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         192.168.2.32    0.0.0.0         UG    0      0        0 enp0s3
192.168.2.0     0.0.0.0         255.255.255.0   U     100    0        0 enp0s3
anika@anika-VirtualBox:~$ sudo route del -0.0.0.0 gw 192.168.2.32 netmask 0.0.0
.0 dev enp0s3
route: invalid option -- '0'
route: invalid option -- '.'
route: invalid option -- '0'
route: invalid option -- '.'
route: invalid option -- '0'
route: invalid option -- '.'
route: invalid option -- '0'
Usage: route [-nNvee] [-FC] [<AF>]           List kernel routing tables
       route [-v] [-FC] {add|del|flush} ...  Modify routing table for AF.

       route {-h|--help} [<AF>]              Detailed usage syntax for specifie
d AF.
       route {-V|--version}                  Display version/author and exit.

        -v, --verbose            be verbose
        -n, --numeric            don't resolve names
        -e, --extend             display other/more information
        -F, --fib                display Forwarding Information Base (default)
        -C, --cache              display routing cache instead of FIB
```

d**) Then remove the interface completely. Answer:** Command for removing the interface completely,

Sudo ifconfig enp0s3 down

```
anika@anika-VirtualBox:~$ sudo ifconfig enp0s3 down
anika@anika-VirtualBox:~$ ifconfig
lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 948  bytes 95773 (95.7 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 948  bytes 95773 (95.7 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

5) Add a New Network

a) Enter the command needed to add another network with the same values as your primary network meaning.

b) Assign the default gateway for newly added network (Your default Gateway Address):

c) Look for your newly added network in your routing table by issuing the "$ netstat -r "command.

d) Now remove your chang2s meaning the double routing table setup for your primary network. First issue the command needed to delete your newly added route then issue the command to delete you newly added default gateway.

6) Multi network scenario configuration:

You should now set up a working routing table for a multi-network scenario. Assume that you have two network cards available connected to two different LANs. The destination of the first network is, 10.0.2.0 with netmask 255.0.0.0 and the second, 192.168.1.0 with netmask 255.255.255.0 Furthermore, a firewall is assumed to exist between the two networks, where network card eth0 is attached to the 10.0.2.0 network and eth1 is attached to the 192.168.1.0 network. To forward packets on the internet the firewall needs to route packets from the 10.0.2.0 network through the 192.168.1.0 network. The firewall system must be set up with two IP addresses, 10.0.2.1 on eth0 and 192.168.1.25 on eth1. The gateway to the internet on the 192.168.1.0 network should be 192.168.1.1.

Provide the necessary commands to route on the firewall/router system.

a) Assign the firewall IP addresses to eth1 and eth2.

b) Add the routes for the networks, i.e. 192.168.1.0 on eth1 and 10.0.2.0 on eth0

c) Assign the internet gateway (meaning: 192.168.1.1) as the default gateway.

d) Enter the necessary commands in order for packets belonging to computers in the 10.0.2.0 network to be routed to the 192.168.1.0 network and the internet. In other words this should tell each computer on the 10.0.2.0, which the default gateway is, i.e.., your firewall/router. You do not need to be worry about the route back configuration it is enough to assign the proper default gateway for the 10.0.2.0 network.

Conclusion: Linux easily manages multiple network interface adapters. Laptops typically include both wired and wireless interfaces, and may also support WiMax interfaces for cellular networks. Linux desktop computers also support multiple network interfaces, and you can use your Linux computer as a multi-network client, or as a router for internal networks; such is the case with a couple of my own systems. Every network interface has its own configuration file in the */etc/sysconfig/network-scripts* directory. Each interface has a configuration file named *ifcfg-<interface-name>X*, where X is the number of the interface, starting with zero or 1 depending upon the naming convention in use; for example */etc/sysconfig/network-scripts/ifcfg-eth0* for the first Ethernet interface.

Most of the other files in the */etc/sysconfig/network-scripts* directory are scripts used to start, stop and perform various network configuration activities.

Each interface configuration file is bound to a specific physical network interface by the MAC address of the interface.

There are many configuration options for the interface configuration files. These are some of the more common options:

- DEVICE: The logical name of the device, such as eth0 or enp0s2.
- HWADDR: The MAC address of the NIC that is bound to the file, such as 00:16:76:02:BA:DB
- ONBOOT: Start the network on this device when the host boots. Options are yes/no. This is typically set to "no" and the network does not start until a user logs in to the desktop. If you need the network to start when no one is logged in, set this to "yes".
- IPADDR: The IP Address assigned to this NIC such as 192.168.0.10
- BROADCAST: The broadcast address for this network such as 192.168.0.255
- NETMASK: The netmask for this subnet such as the class C mask 255.255.255.0
- NETWORK: The network ID for this subnet such as the class C ID 192.168.0.0
- SEARCH: The DNS domain name to search when doing lookups on unqualified hostnames such as "example.com"
- BOOTPROTO: The boot protocol for this interface. Options are static, DHCP, bootp, none. The "none" option defaults to static.

- GATEWAY: The network router or default gateway for this subnet, such as 192.168.0.254
- ETHTOOL_OPTS: This option is used to set specific interface configuration items for the network interface, such as speed, duplex state, and autonegotiation state. Because this option has several independent values, the values should be enclosed in a single set of quotes, such as: "autoneg off speed 100 duplex full".
- DNS1: The primary DNS server, such as 192.168.0.254, which is a server on the local network. The DNS servers specified here are added to the */etc/resolv.conf* file when using NetworkManager, or when the peerdns directive is set to yes, otherwise the DNS servers must be added to */etc/resolv.conf* manually and are ignored here.
- DNS2: The secondary DNS server, for example 8.8.8.8, which is one of the free Google DNS servers. Note that a tertiary DNS server is not supported in the interface configuration files, although a third may be configured in a non-volatile *resolv.conf* file.
- TYPE: Type of network, usually Ethernet. The only other value I have ever seen here was Token Ring but that is now mostly irrelevant.
- PEERDNS: The yes option indicates that */etc/resolv.conf* is to be modified by inserting the DNS server entries specified by DNS1 and DNS2 options in this file. "No" means do not alter the *resolv.conf* file. "Yes" is the default when DHCP is specified in the BOOTPROTO line.
- USERCTL: Specifies whether non-privileged users may start and stop this interface. Options are yes/no.
- IPV6INIT: Specifies whether IPV6 protocols are applied to this interface. Options are yes/no.