

EXPERIMENT – 1

AIM: - Study of various Network commands used in Linux and Windows:

BASIC NETWORKING COMMANDS:

arp –a:- ARP is short form of address resolution protocol, It will show the IP address of your computer along with the IP address and MAC address of your router.

hostname: This is the simplest of all TCP/IP commands. It simply displays the name of your computer.

ipconfig /all: This command displays detailed configuration information about your TCP/IP connection including Router, Gateway, DNS, DHCP, and type of Ethernet adapter in your system

nbtstat –a: This command helps solve problems with NetBIOS name resolution. (Nbt stands for NetBIOS over TCP/IP)

netstat: (network statistics) netstat displays a variety of statistics about a computers active TCP/IP connections. It is a command line tool for monitoring network connections both incoming and outgoing as well as viewing routing tables, interface statistics etc.

e.g.:– netstat –r

nslookup: (name server lookup) is a tool used to perform DNS lookups in Linux. It is used to display DNS details, such as the IP address of a particular computer, the MX records for a domain or the NS servers of a domain. nslookup can operate in two modes: interactive and non-interactive.

e.g.:– nslookup www.google.com

pathping: Pathping is unique to Window's, and is basically a combination of the Ping and Tracert commands. Pathping traces the route to the destination address then launches a 25 second test of each router along the way, gathering statistics on the rate of data loss along each hop.

ping: (Packet INternet Groper) command is the best way to test connectivity between two nodes. Ping use ICMP (Internet Control Message Protocol) to communicate to other devices.

1. #ping hostname(ping localhost)
2. #ping ip address (ping 4.2.2.2)
3. #ping fully qualified domain name(ping www.facebook.com)

Route: route command is used to show/manipulate the IP routing table. It is primarily used to setup static routes to specific host or networks via an interface.

Some important Linux networking commands

1. ip

The ip command is one of the basic commands every administrator will need in daily work, from setting up new systems and assigning IPsto troubleshooting existing systems. The ip command can show addressinformation, manipulate routing, plus display network various devices, interfaces, and tunnels.

ip <OPTIONS> <OBJECT> <COMMAND>

Here are some common use cases for the ip command.

a. To show the IP addresses assigned to an interface on your server:

[root@server ~]# ip address show

b. To assign an IP to an interface, for example, enps03:

[root@server ~]# ip address add 192.168.1.254/24 dev enps03

c. To delete an IP on an interface:

[root@server ~]# ip address del 192.168.1.254/24 dev enps03

d. Alter the status of the interface by bringing the interface eth0 online:

[root@server ~]# ip link set eth0 up

e. Alter the status of the interface by bringing the interface eth0 offline:

[root@server ~]# ip link set eth0 down

f. Alter the status of the interface by enabling promiscuous mode for eth0:

[root@server ~]# ip link set eth0 promisc on

g. Add a default route (for all addresses) via the local gateway 192.168.1.254 that can be reached on device eth0:

[root@server ~]# ip route add default via 192.168.1.254 dev eth0

h. Add a route to 192.168.1.0/24 via the gateway at 192.168.1.254:

[root@server ~]# ip route add 192.168.1.0/24 via 192.168.1.254

i. Add a route to 192.168.1.0/24 that can be reached on device eth0:

[root@server ~]# ip route add 192.168.1.0/24 dev eth0

j. Delete the route for 192.168.1.0/24 via the gateway at 192.168.1.254:

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

k. Display the route taken for IP 10.10.1.4:

[root@server ~]# ip route get 10.10.1.4

2. ifconfig

The ifconfig command was/is a staple in many sysadmin's tool belt for configuring and troubleshooting networks. It hassince been replaced by the ip command discussed above.

3. mtr

MTR (Matt's traceroute) is a program with a command-line interface that serves as a network diagnostic and troubleshooting tool. This command combines the functionality of the ping and traceroute commands. Just like a traceroute, the mtr command will show the route from a computer to a specified host. mtr provides a lot of statistics about each hop, such as response time and percentage. With the mtr command, you will get more information about the route and be able to see problematic devices along the way. If you see a sudden increase in response time or packet loss, then obviously, there is a bad link somewhere.

The syntax of the command is as follows:

```
mtr <options> hostname/IP
```

Let's look at some common use cases.

- a. The basic mtr command shows you the statistics, including each hop (hostnames) with time and loss%:

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

- b. Show numeric IP addresses (if you use -g, you will get IP addresses (numbers) instead of hostnames):

```
[root@server ~]# mtr -g google.com
```

- c. Show the numeric IP addresses and hostnames, too:

```
[root@server ~]# mtr -b google.com
```

- d. Set the number of pings that you want to send:

```
[root@server ~]# mtr -c 10 google.com
```

4. tcpdump

The tcpdump command is designed for capturing and displaying packets.

You can install tcpdump with the command below:

```
[root@server ~]# dnf install -y tcpdump
```

Before starting any capture, you need to know which interfaces tcpdump can use. You will need to use sudo or have root access in this case.

```
[root@server ~]# tcpdump -D
```

If you want to capture traffic on eth0, you can initiate that with tcpdump -i eth0 sample output:

```
[root@server ~]# tcpdump -i eth0
```

5. ping

Ping is a tool that verifies IP-level connectivity to another TCP/IP computer by sending Internet Control Message Protocol (ICMP) Echo Request messages. The receipt of corresponding Echo Reply messages is displayed, along with round-trip times. Ping is the primary TCP/IP command used to troubleshoot connectivity, reachability, and name resolution.

```
[root@server ~]# ping google.com
```

Configuring an Ethernet connection by using nmcli

If you connect a host to the network over Ethernet, you can manage the connection's settings on the command line by using the nmcli utility.

Procedure

1. List the NetworkManager connection profiles:

```
# nmcli connection show
```

```
NAME UUID TYPE DEVICE
```

```
Wired connection 1 a5eb6490-cc20-3668-81f8-0314a27f3f75 ethernet enp1s0
```

2. # nmcli connection add con-name <connection-name> ifname <device-name>

```
type ethernet
```

Skip this step to modify an existing profile.

3. Optional: Rename the connection profile:

```
# nmcli connection modify "Wired connection 1"
```

Here, “Wired connection 1” is the name of the connection

4. Display the current settings of the connection profile:

```
# nmcli connection show
```

```
connection.interface-name: enp1s0
```

```
connection.autoconnect: yes
```

```
ipv4.method: auto
```

```
ipv6.method: auto
```

```
...
```

5. Configure the IPv4 settings:

To use DHCP, enter:

```
# nmcli connection modify "Wired connection 1" ipv4.method auto
```

Skip this step if ipv4.method is already set to auto (default).

To set a static IPv4 address, network mask, default gateway, DNS servers, and search domain, enter:

```
# nmcli connection modify "Wired connection 1" ipv4.method manual
```

```
ipv4.addresses 192.0.2.1/24 ipv4.gateway 192.0.2.254 ipv4.dns 192.0.2.200
```

```
ipv4.dns-search example.com
```

6. Configure the IPv6 settings:

To use stateless address autoconfiguration (SLAAC), enter:

```
# nmcli connection modify "Wired connection 1" ipv6.method auto
```

Skip this step if ipv6.method is already set to auto (default).

7. Activate the profile:

```
# nmcli connection up Internal-LAN
```

Verification

1. Display the IP settings of the NIC:

```
# ip address show enp1s0
```

```
enp1s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel
```

```
state UP group default qlen 1000
```

```
link/ether 52:54:00:17:b8:b6 brd ff:ff:ff:ff:ff:ff
```

```
inet 192.0.2.1/24 brd 192.0.2.255 scope global noprefixroute enp1s0
```

```
valid_lft forever preferred_lft forever
```

```
inet6 2001:db8:1::fffe/64 scope global noprefixroute
```

```
valid_lft forever preferred_lft forever
```

2. Display the IPv4 default gateway:

```
# ip route show default
```

```
default via 192.0.2.254 dev enp1s0 proto static metric 102
```

3. Display the IPv6 default gateway:

```
# ip -6 route show default
```

```
default via 2001:db8:1::ffee dev enp1s0 proto static metric 102 pref medium
```

4. Display the DNS settings:

```
# cat /etc/resolv.conf
```

```
search example.com
```

```
nameserver 192.0.2.200
```

```
nameserver 2001:db8:1::ffbb
```

If multiple connection profiles are active at the same time, the order of nameserver entries depend on the DNS priority values in these profile and the connection types.

5. Use the ping utility to verify that this host can send packets to other hosts:

```
# ping <host-name-or-IP-address>
```

Troubleshooting

Verify that the network cable is plugged-in to the host and a switch.

Check whether the link failure exists only on this host or also on other hosts connected to the same switch.

Verify that the network cable and the network interface are working as expected.

Perform hardware diagnosis steps and replace defect cables and network interface cards.

If the configuration on the disk does not match the configuration on the device, starting or restarting NetworkManager creates

OUTPUT:-

```
Command Prompt + ^

Microsoft Windows [Version 10.0.26200.7019]
(c) Microsoft Corporation. All rights reserved.
The system cannot find the path specified.

C:\Users\mdars>ping www.google.com

Pinging www.google.com [2404:6800:4007:835::2004] with 32 bytes of data:
Reply from 2404:6800:4007:835::2004: time=6ms
Reply from 2404:6800:4007:835::2004: time=9ms
Reply from 2404:6800:4007:835::2004: time=5ms
Reply from 2404:6800:4007:835::2004: time=6ms

Ping statistics for 2404:6800:4007:835::2004:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 5ms, Maximum = 9ms, Average = 6ms

C:\Users\mdars>ipconfig

Windows IP Configuration

Unknown adapter Local Area Connection:
    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix . :

Wireless LAN adapter Local Area Connection* 1:
    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix . :

Wireless LAN adapter Local Area Connection* 2:
    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix . :

Ethernet adapter VMware Network Adapter VMnet1:
    Connection-specific DNS Suffix . :
    Link-local IPv6 Address . . . . . : fe80::6f80:9364:79f2:b663%5
    IPv4 Address. . . . . : 192.168.244.1
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . :
```

```
Command Prompt + X
C:\Users\mdars>nslookup www.google.com
Server: Unknown
Address: fe80::1

Non-authoritative answer:
Name: www.google.com
Addresses: 2404:6800:4007:835::2004
          142.251.222.196

C:\Users\mdars>arp

Displays and modifies the IP-to-Physical address translation tables used by
address resolution protocol (ARP).

ARP -s inet_addr eth_addr [if_addr]
ARP -d inet_addr [if_addr]
ARP -a [inet_addr] [-N if_addr] [-v]

-a           Displays current ARP entries by interrogating the current
            protocol data. If inet_addr is specified, the IP and Physical
            addresses for only the specified computer are displayed. If
            more than one network interface uses ARP, entries for each ARP
            table are displayed.
-g           Same as -a.
-v           Displays current ARP entries in verbose mode. All invalid
            entries and entries on the loop-back interface will be shown.
inet_addr   Specifies an Internet address.
-N if_addr   Displays the ARP entries for the network interface specified
            by if_addr.
-d           Deletes the host specified by inet_addr. inet_addr may be
            wildcarded with * to delete all hosts.
-s           Adds the host and associates the Internet address inet_addr
            with the Physical address eth_addr. The Physical address is
            given as 6 hexadecimal bytes separated by hyphens. The entry
            is permanent.
eth_addr    Specifies a physical address.
if_addr     If present, this specifies the Internet address of the
            interface whose address translation table should be modified.
            If not present, the first applicable interface will be used.

Example:
> arp -s 157.55.85.212 00-aa-00-62-c6-09 .... Adds a static entry.
> arp -a                   .... Displays the arp table.

C:\Users\mdars>
```

```
C:\Users\mdars>hostname
DarshanM

C:\Users\mdars>getmac

Physical Address      Transport Name
=====
00-FF-FA-39-11-A2    Media disconnected
C4-3D-1A-21-C9-A1    \Device\Tcpip_{BE2EFAF0-2D78-4A0B-A09D-2C97479F013E}
00-50-56-C0-00-01    \Device\Tcpip_{2EC0D958-8684-4DD3-BB94-DCED67E010C5}
00-50-56-C0-00-08    \Device\Tcpip_{8C5770D9-87EE-455A-A8D8-91CB47B1BE6F}

C:\Users\mdars>
```

RESULT: -

Studied of various Network commands used in Linux and Windows

