

EE673 – Laboratory

Node connectivity: Configuration and performance Testing

Objective:

- Network Configuration and testing on linux system

After you have booted your system and logged in as guest, open a terminal window.

NOTE:

1. The commands to be typed in the terminal window are in this font

2. Your reference document for the testbed ip and topology diagram are on brihaspati

ifconfig – understand ifconfig and configure a network interface

PART A

% **ifconfig / netstat -ie** (shows output similar to)

```
eth0   Link encap:Ethernet HWaddr 70:71:bc:9a:78:05
inet addr:172.26.82.77 Bcast:172.26.83.255 Mask:255.255.252.0
inet6 addr: 2001:e30:1401:1:7271:bcff:fe9a:7805/64 Scope:Global
inet6 addr: fe80::7271:bcff:fe9a:7805/64 Scope:Link
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:17072 errors:0 dropped:171 overruns:0 frame:0
TX packets:4450 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:4790614 (4.7 MB) TX bytes:1551500 (1.5 MB)
Interrupt:20 Memory:fb100000-fb120000
```

| |
|--|
| Link encap: Link encapsulation eth0: Interface Ethernet 0 Hwaddr: Hardware (MAC) Address inet addr: Internet Address (IPv4) Bcast: Broadcast Address Mask: mask |
|--|

```
lo     Link encap:Local Loopback
inet addr:127.0.0.1 Mask:255.0.0.0
inet6 addr: ::1/128 Scope:Host
UP LOOPBACK RUNNING MTU:16436 Metric:1
RX packets:28 errors:0 dropped:0 overruns:0 frame:0
TX packets:28 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:0
RX bytes:1680 (1.6 KB) TX bytes:1680 (1.6 KB)
```

| |
|---|
| Will also show unconfigured eth1,eth2,eth3,eth4,eth5 |
|---|

| |
|------------------------|
| lo: Loopback interface |
|------------------------|

netstat - Print network connections, routing tables, interface statistics, masquerade connections, and multicast memberships

% **netstat -r / netstat -rn** (shows output similar to)

Kernel IP routing table

| Destination | Gateway | Genmask | Flags | MSS | Window | irtt | Iface |
|-------------|---------------|---------------|-------|-----|--------|------|-------|
| 172.26.80.0 | * | 255.255.252.0 | U | 0 | 0 | 0 | eth0 |
| link-local | * | 255.255.0.0 | U | 0 | 0 | 0 | eth0 |
| default | 172.26.83.254 | 0.0.0.0 | UG | 0 | 0 | 0 | eth0 |

| | |
|---|---|
| Destination: Destination network / host | Gateway: router address |
| Genmask: netmask for destination net | MSS: Default TCP maximum segment size |
| window: TCP Window size | irtt: initial round trip time |
| Iface: Interface to which packets for this | Link-local: address block 169.254.0.0/16 |

| | |
|--|--|
| route will be sent | |
| Default: If destination address is not on this network (172.26.80.0)/ link-local, use default gateway | |

Flags :

Possible flags include

U (route is **up**)

H (target is a **host**)

G (use **gateway**)

R (**reinstall** route for dynamic routing)

D (**dynamically** installed by daemon or redirect)

M (**modified** from routing daemon or redirect)

A (installed by **addrconf**)

C (**cache** entry)

! (**reject** route)

ping: send ICMP ECHO_REQUEST to network hosts

`% ping <default gateway>` (If default gateway is reachable, shows output similar to)

```
PING 172.26.83.254 (172.26.83.254) 56(84) bytes of data.
64 bytes from 172.26.83.254: icmp_req=1 ttl=255 time=1.32 ms
64 bytes from 172.26.83.254: icmp_req=2 ttl=255 time=1.19 ms
64 bytes from 172.26.83.254: icmp_req=3 ttl=255 time=0.554 ms
64 bytes from 172.26.83.254: icmp_req=4 ttl=255 time=2.07 ms
```

arp: address resolution display and control

`% arp -an` (shows all the IP and hardware addresses the interface has learnt through some broadcast)

Traceroute: Print the route packets trace to network host

```
% traceroute 172.26.83.251
% traceroute 172.26.82.159
% traceroute 172.31.1.1
% traceroute www.iitk.ac.in
```

host: Name resolution

Use any of the commands learnt in lab-1 to see contents of /etc/resolv.conf.

`% host -v FQDN`

e.g. www.iitk.ac.in

understand the output of above command and contents of resolv.conf

route: show / manipulate the IP routing table

`% route -n`

PART B

Logging into the testbed:

`% ssh root@<ipaddress eth0 on testbed>`

You would log into the testbed.

% screen

Press Ctrl+a+c to get a new window on the testbed. You can switch between the windows using ctrl+a <screen no>

These commands are to be run on the testbed.

Your reference documents are Testbed-ip.pdf and topology-1.pdf uploaded on Brihaspati.

Objective: Create a network with point to point links using appropriate ip addr and subnet

% ifconfig -a

Note the output.

New command from iproute2 Older Linux command

ip addr ifconfig

ip neighbor arp

ip route route

See, which interfaces have been assigned IP, and which do not have an IP.

Use the above document to assign ip address to eth1, eth2 for the node you are logged into

Note- If you change ip for eth0, you will loose connectivity to the testbed.

ip addr add <ipaddr/msize> dev <ethno>

alternative to ##### ifconfig <interface x> <ip>/<mask>

Check if IP address is configured correctly

ip addr show

Delete the configured IP address on eth1 or eth2.

ip addr del <ipaddr/msize> dev <ethno>

Objective: Check the routing table

ip route show alternative to ##### route -n

Objective: Check reachability

Check that neighbors are reachable by ping < ip addr of neighbor >

% ping <ip>

ip neigh show alternative to ##### arp -an

Objective: DNS resolver configuration

Look at /etc/resolv.conf

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

Make following entries in /etc/resolv.conf

```
nameserver 192.168.3.128
search netlab.ee.iitk.ac.in
```

Check using host or nslookup

Objective: Setup routing

Set up static routing in forward and reverse directions

Enable intermediate nodes to act as router by setting ip_forward
(Path name is /proc/sys/net/ipv4/ip_forward)

```
# ip route add <network/msize> via <nh-ipaddr> dev <ethno>
```

alternative to

```
##### route add -net <network address> netmask <netmask> gw  
<gateway address>
```

Check the routing table and connectivity

```
# ip route show
```

Objective: To bring an ethernet interface up/down

```
# ip link set dev <ethno> down % ifconfig <ethx> down
```

Check the interface status

bring the interface up and see the interface status

Please note:

Understand the topology diagram, connectivity and the routing in the testbed. You will need to use this for the routing experiment.

PART C

Module 1:

(Refer to the lab sheet to work on this module. You will need topology.pdf and testbed-ip.pdf as reference documents. The command set to be used in this module has been given in lab sheet for lab-2)

1. Create a network with point to point links using appropriate ip addr and subnet mask.
2. Check that neighbors are reachable by ping < ip addr of neighbor >
3. Set up static routing in forward and reverse directions to set up a topology
4. Enable intermediate nodes to act as router by setting ip_forward
(Path name is /proc/sys/net/ipv4/ip_forward)

Module 2: (iperf) performance measurement

Generate Traffic using iperf . Typical performance parameters are throughput, delay, packet loss rate

Run iperf as a server on one node

```
% iperf -s
```

Run iperf as client

```
% iperf -c <server ip>
```

Note down the throughput achieved.

Data formatting: (-f argument)

The -f argument can display the results in the desired format: bits(b), bytes(B), kilobits(k), kilobytes(K), megabits(m), megabytes(M), gigabits(g) or gigabytes(G).

Generally the bandwidth measures are displayed in bits (or Kilobits, etc ...) and an amount of data is displayed in bytes (or Kilobytes, etc ...).

```
% iperf -c <server ip> -f -b
```

use the command netstat an | more to see the ports being used by iperf. By default server runs on port 5001

Stop the server and the client and test the following for udp traffic

```
% iperf -s -u -i 1
```

```
% iperf -c <serverrip> -u -b 10m
```

Observer the output.

Module 3 : Wireshark packet sniffing

wireshark: Application to Interactively dump and analyze network traffic .

To be executed on ***front end system terminal*** AND NOT on **testbed node** .

```
% ssh -C root@<eth0 ip> tcpdump -i <ethno> -U -w - " not port 22 " |wireshark -k -n -i -
```

Enter the root password on testbed when asked.