

Institute of Computer Technology  
B. Tech Computer Science and Engineering  
Subject: Basics Of Communication Systems (2CSE202)

**PRACTICAL-2**

**AIM: - Networking Commands**

**1. What are networking commands?**

- The **commands** (such as tracert, traceroute, ping, arp, netstat, nbstat, NetBIOS, ipconfig, winipcfg and nslookup) and their arguments, options and parameters used to troubleshoot the computer **network**.

**2. Why we require networking commands?**

- For trouble shooting and reassuring network activity

**3. Where to perform those commands?**

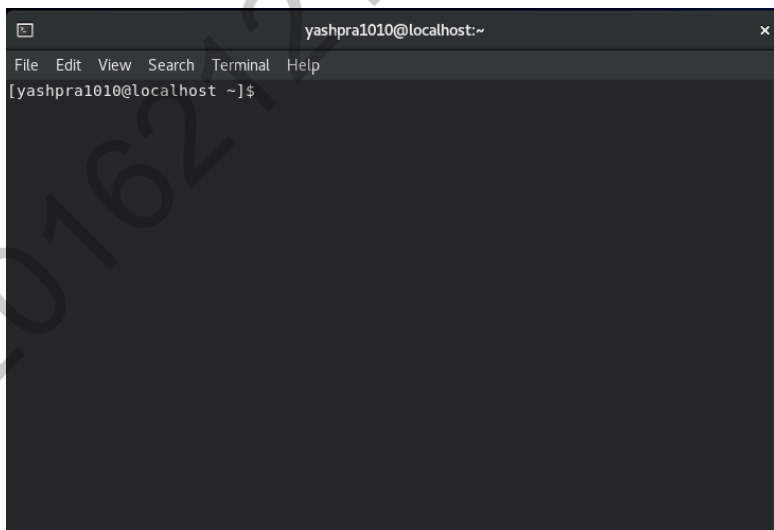
- On compiler for windows cmd and for linux, ubuntu or cent-os terminal

**Below is CMD in Windows**

A screenshot of the Windows Command Prompt window. The title bar reads "Command Prompt". The window content shows: "Microsoft Windows [Version 10.0.19042.906]", "(c) Microsoft Corporation. All rights reserved.", and the command prompt "C:\Users\admin>".

```
Microsoft Windows [Version 10.0.19042.906]
(c) Microsoft Corporation. All rights reserved.
C:\Users\admin>
```

**Below is CMD in Linux - CentOS**

A screenshot of a Linux terminal window on CentOS. The title bar shows the user "yashpra1010@localhost" and the terminal icon. The menu bar includes "File", "Edit", "View", "Search", "Terminal", and "Help". The terminal content shows the prompt "[yashpra1010@localhost ~]\$".

```
yashpra1010@localhost:~
File Edit View Search Terminal Help
[yashpra1010@localhost ~]$
```

#### 4. What is default gateway and subnet mask?

- Default Gateway is IP of your Router. In simple words, 192.168.0.1 and Subnet mask will be automatically deduced by operating system. 255.255.255.0.

#### 5. Why we need default gateway?

- A default gateway makes it possible for devices in one network to communicate with devices in another network. If a computer, for example, requests a web page, the request goes through the default gateway before exiting the local network to reach the internet. Think of a default gateway as an intermediate device between the local network and the internet. The default gateway transfers internal data to the internet and back again.

#### 6. Types of default gateways

- Broadband-Routers
- Dial-up
- Network-adaptors

### SCENARIOS AND LIST OF COMMANDS

#### CASE 1:

Consider a situation, where you need to set a default gateway then how you will find your default gateway ip address?

For windows: ipconfig

For linux : netstat and iproute

#### CMD OUTPUT OF :ipconfig

```

C:\Users\admin>ipconfig

Windows IP Configuration

Unknown adapter Local Area Connection:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . :

Wireless LAN adapter Wi-Fi:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . :

Wireless LAN adapter Local Area Connection* 10:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . :

Wireless LAN adapter Local Area Connection* 12:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . :

Ethernet adapter Ethernet:

    Connection-specific DNS Suffix  . :
    Link-local IPv6 Address . . . . . : fe80::e4c1:c36d:c78e:3fb8%4
    IPv4 Address. . . . . : 192.168.0.105
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 192.168.0.1

Ethernet adapter VMware Network Adapter VMnet1:

    Connection-specific DNS Suffix  . :
    Link-local IPv6 Address . . . . . : fe80::848a:1a48:dddd:dd6a%21
    IPv4 Address. . . . . : 192.168.15.1
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . :

Ethernet adapter VMware Network Adapter VMnet8:

    Connection-specific DNS Suffix  . :
    Link-local IPv6 Address . . . . . : fe80::6d15:38ab:ba56:3772%5
    IPv4 Address. . . . . : 192.168.77.1
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . :

C:\Users\admin>

```

**DOMAIN NAME SERVER.**

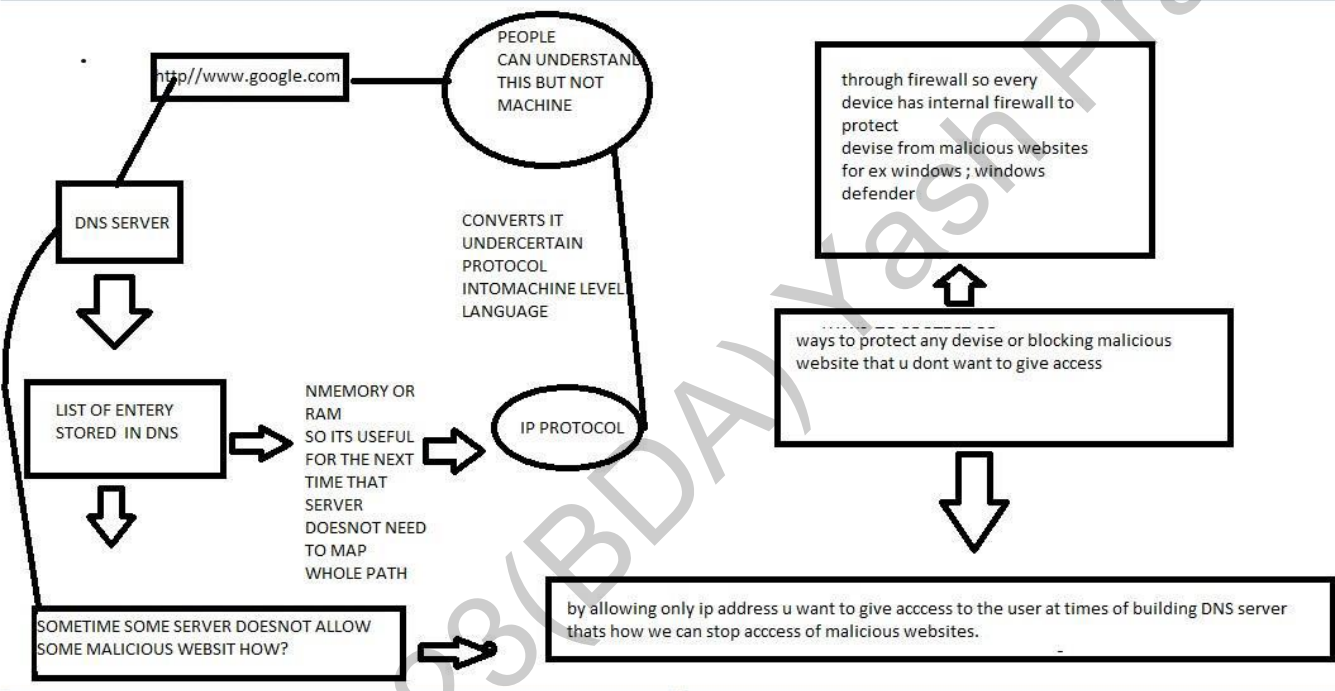
For-example <https://www.google.com>

When a user request it this URL is: A directory (list)of domain names and translate them to Internet Protocol (IP) addresses. This is necessary because, although domain names are easy for people to remember, computers or machines, access websites based on IP addresses.

Scenario1: you don't want to give access to students particular to some websites but how to do?

Scenario 2: How DNS server exactly works?

DNS SERVER:



4.Wifi for ipconfig that wifi adapter currently wifi is on so, Here it indicates details of wifi currently XYZ phone is having this ip and ipv4 and subnet mask it is used to get default gateway information in this case xyz is default gateway.

```
Ethernet adapter Ethernet:

Connection-specific DNS Suffix . : 
Link-local IPv6 Address . . . . . : fe80::e4c1:c36d:c78e:3fb8%4
IPv4 Address. . . . . : 192.168.0.105
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . : 192.168.0.1
```

5. ipconfig all: to get all details of ipv4 and ipv6

```

Command Prompt

USAGE:
ipconfig [/allcompartments] [/? | /all |
        /renew [adapter] | /release [adapter] |
        /renew6 [adapter] | /release6 [adapter] |
        /flushdns | /displaydns | /registerdns |
        /showclassid adapter |
        /setclassid adapter [classid] |
        /showclassid6 adapter |
        /setclassid6 adapter [classid] ]

where
adapter          Connection name
                  (wildcard characters * and ? allowed, see examples)

Options:
/?              Display this help message
/all           Display full configuration information.
/release       Release the IPv4 address for the specified adapter.
/release6      Release the IPv6 address for the specified adapter.
/renew         Renew the IPv4 address for the specified adapter.
/renew6        Renew the IPv6 address for the specified adapter.
/flushdns      Purges the DNS Resolver cache.
/registerdns   Refreshes all DHCP leases and re-registers DNS names
/displaydns    Display the contents of the DNS Resolver cache.
/showclassid   Displays all the dhcp class IDs allowed for adapter.
/setclassid    Modifies the dhcp class id.
/showclassid6  Displays all the IPv6 DHCP class IDs allowed for adapter.
/setclassid6   Modifies the IPv6 DHCP class id.

The default is to display only the IP address, subnet mask and
default gateway for each adapter bound to TCP/IP.

For Release and Renew, if no adapter name is specified, then the IP address
leases for all adapters bound to TCP/IP will be released or renewed.

For Setclassid and Setclassid6, if no ClassId is specified, then the ClassId is removed.

Examples:
> ipconfig           ... Show information
> ipconfig /all       ... Show detailed information
> ipconfig /renew     ... renew all adapters
> ipconfig /renew EL* ... renew any connection that has its
                        name starting with EL
> ipconfig /release *Con* ... release all matching connections,
                        eg. "Wired Ethernet Connection 1" or
                        "Wired Ethernet Connection 2"
> ipconfig /allcompartments ... Show information about all
                        compartments
> ipconfig /allcompartments /all ... Show detailed information about all
                        compartments

C:\Users\admin>

```

Scenario 3: consider a situation where I want to change or flush my old DHCP ip so for ipv4:  
 release  
 Ipv6: release6  
 And then renew

```

C:\Users\admin>ipconfig/release

Windows IP Configuration

No operation can be performed on Local Area Connection while it has its media disconnected.
No operation can be performed on Wi-Fi while it has its media disconnected.
No operation can be performed on Local Area Connection* 10 while it has its media disconnected.
No operation can be performed on Local Area Connection* 12 while it has its media disconnected.

Unknown adapter Local Area Connection:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . :

Wireless LAN adapter Wi-Fi:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . :

Wireless LAN adapter Local Area Connection* 10:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . :

Wireless LAN adapter Local Area Connection* 12:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . :

Ethernet adapter Ethernet:

    Connection-specific DNS Suffix  . :
    Link-local IPv6 Address . . . . . : fe80::e4c1:c36d:c78e:3fb8%4
    Default Gateway . . . . . :

Ethernet adapter VMware Network Adapter VMnet1:

    Connection-specific DNS Suffix  . :
    Link-local IPv6 Address . . . . . : fe80::848a:1a48:dddd:dd6a%21
    Default Gateway . . . . . :

Ethernet adapter VMware Network Adapter VMnet8:

    Connection-specific DNS Suffix  . :
    Link-local IPv6 Address . . . . . : fe80::6d15:38ab:ba56:3772%5
    Default Gateway . . . . . :

C:\Users\admin>

```

```

C:\Users\admin>ipconfig/renew

Windows IP Configuration

No operation can be performed on Local Area Connection while it has its media disconnected.
No operation can be performed on Wi-Fi while it has its media disconnected.
No operation can be performed on Local Area Connection* 10 while it has its media disconnected.
No operation can be performed on Local Area Connection* 12 while it has its media disconnected.

Unknown adapter Local Area Connection:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . :

Wireless LAN adapter Wi-Fi:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . :

Wireless LAN adapter Local Area Connection* 10:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . :

Wireless LAN adapter Local Area Connection* 12:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . :

Ethernet adapter Ethernet:

    Connection-specific DNS Suffix  . :
    Link-local IPv6 Address . . . . . : fe80::e4c1:c36d:c78e:3fb8%4
    IPv4 Address. . . . . : 192.168.0.105
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 192.168.0.1

Ethernet adapter VMware Network Adapter VMnet1:

    Connection-specific DNS Suffix  . :
    Link-local IPv6 Address . . . . . : fe80::848a:1a48:dddd:dd6a%21
    IPv4 Address. . . . . : 192.168.15.1
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . :

Ethernet adapter VMware Network Adapter VMnet8:

    Connection-specific DNS Suffix  . :
    Link-local IPv6 Address . . . . . : fe80::6d15:38ab:ba56:3772%5
    IPv4 Address. . . . . : 192.168.77.1
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . :

C:\Users\admin>

```

6. Consider a scenario when you want to cross check whether your request is being sent properly or not whether anybody is not accessing your data.

### **Ping:**

To check whether I am connected to any website or not? Tracert

For ex: google.com

Ping google.com

Ping 8.8.8.8

Ping 4.4.4.4

```

C:\Users\admin>ping 8.8.8.8

Pinging 8.8.8.8 with 32 bytes of data:
Reply from 8.8.8.8: bytes=32 time=9ms TTL=116
Reply from 8.8.8.8: bytes=32 time=10ms TTL=116
Reply from 8.8.8.8: bytes=32 time=10ms TTL=116
Reply from 8.8.8.8: bytes=32 time=10ms TTL=116

Ping statistics for 8.8.8.8:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 9ms, Maximum = 10ms, Average = 9ms

C:\Users\admin>

```

Yes, all packets are sent perfectly to transmitter and I am receiving all

Tracert command

for example : a client comes to me that my internet connectivity is not up to the mark how would I find?

Fire this command check the particular route and find out that whether a request is received by server, which route and at which point it has been showing dilemmas or whether switch/router/firewall/ISP/devices. where is the problem occurring?

```
C:\Users\admin>tracert 8.8.8.8

Tracing route to dns.google [8.8.8.8]
over a maximum of 30 hops:

  1    <1 ms    <1 ms    <1 ms    192.168.0.1
  2     1 ms     1 ms     <1 ms    10.230.192.1
  3     *        *        *        Request timed out.
  4     *        *        *        Request timed out.
  5    10 ms     9 ms    10 ms    103.241.47.142
  6    10 ms    10 ms     9 ms    103.241.47.206
  7    18 ms    10 ms    10 ms    142.250.47.236
  8    11 ms    11 ms    10 ms    209.85.245.247
  9    11 ms    10 ms    10 ms    209.85.242.111
 10    10 ms     9 ms     9 ms    dns.google [8.8.8.8]

Trace complete.

C:\Users\admin>
```

It shows that request is going out from phone (wifi) to switch then lost somewhere and finally to the dns google server

Netstat 8.8.8.8: it is showing that these many active connections are in between to show connections are established there so

```
C:\Users\admin>netstat 8.8.8.8

Active Connections

Proto Local Address           Foreign Address         State
TCP    127.0.0.1:443           na1r:58845             ESTABLISHED
TCP    127.0.0.1:49681         na1r:49694             ESTABLISHED
TCP    127.0.0.1:49681         na1r:49702             ESTABLISHED
TCP    127.0.0.1:49681         na1r:49703             ESTABLISHED
TCP    127.0.0.1:49681         na1r:49704             ESTABLISHED
TCP    127.0.0.1:49681         na1r:49705             ESTABLISHED
TCP    127.0.0.1:49694         na1r:49681             ESTABLISHED
TCP    127.0.0.1:49702         na1r:49681             ESTABLISHED
TCP    127.0.0.1:49703         na1r:49681             ESTABLISHED
TCP    127.0.0.1:49704         na1r:49681             ESTABLISHED
TCP    127.0.0.1:49705         na1r:49681             ESTABLISHED
TCP    127.0.0.1:49758         na1r:65001             ESTABLISHED
TCP    127.0.0.1:49761         na1r:49780             ESTABLISHED
TCP    127.0.0.1:49780         na1r:49761             ESTABLISHED
TCP    127.0.0.1:52106         na1r:52107             ESTABLISHED
TCP    127.0.0.1:52107         na1r:52106             ESTABLISHED
TCP    127.0.0.1:52108         na1r:52109             ESTABLISHED
TCP    127.0.0.1:52109         na1r:52108             ESTABLISHED
TCP    127.0.0.1:52110         na1r:52111             ESTABLISHED
TCP    127.0.0.1:52111         na1r:52110             ESTABLISHED
TCP    127.0.0.1:52130         na1r:52131             ESTABLISHED
TCP    127.0.0.1:52131         na1r:52130             ESTABLISHED
TCP    127.0.0.1:58839         na1r:https             TIME_WAIT
TCP    127.0.0.1:58845         na1r:https             ESTABLISHED
TCP    127.0.0.1:65001         na1r:49758             ESTABLISHED
TCP    192.168.0.105:58678     40.90.189.152:https    ESTABLISHED
TCP    192.168.0.105:58683     192.168.0.103:8009     ESTABLISHED
TCP    192.168.0.105:58684     40.90.189.152:https    ESTABLISHED
TCP    192.168.0.105:58700     ed1sgcb15002:https     ESTABLISHED
TCP    192.168.0.105:58735     m26symcs112:https      ESTABLISHED
TCP    192.168.0.105:58739     47:https               ESTABLISHED
TCP    192.168.0.105:58740     m26symcs112:https      ESTABLISHED
TCP    192.168.0.105:58741     bom05s15-in-f10:https  ESTABLISHED
TCP    192.168.0.105:58745     unn-138-199-14-88:http ESTABLISHED
TCP    192.168.0.105:58747     162.159.135.234:https  ESTABLISHED
TCP    192.168.0.105:58818     mjp26mcs304:https      ESTABLISHED
TCP    192.168.0.105:58820     192.168.0.103:8008     ESTABLISHED
TCP    192.168.0.105:58833     sc-in-f188:5228        ESTABLISHED
TCP    192.168.0.105:58834     162.159.135.232:https  ESTABLISHED
TCP    192.168.0.105:58835     162.159.135.232:https  ESTABLISHED
TCP    192.168.0.105:58836     162.159.138.234:https  ESTABLISHED
TCP    192.168.0.105:58837     sj3-tsa:https           ESTABLISHED
TCP    192.168.0.105:58841     52.139.250.253:https    ESTABLISHED
TCP    192.168.0.105:58844     168.62.57.154:https     ESTABLISHED
TCP    192.168.0.105:58847     52.109.56.20:https      TIME_WAIT
TCP    192.168.0.105:58848     52.114.75.79:https      ESTABLISHED
TCP    [::1]:8307             TOGO:58846             CLOSE_WAIT
```



ARP command

ARP - Address Resolution Protocol

Short for Address Resolution Protocol, a network layer protocol **used to** convert an IP address into a physical address (called a DLC address), such as an Ethernet address. A host wishing to obtain a physical address broadcasts an **ARP** request onto the TCP/IP network. Consider a situation where ARP address that how many entries has been saved at ARP address

```
C:\Users\admin>arp -a

Interface: 192.168.0.105 --- 0x4
  Internet Address      Physical Address      Type
  192.168.0.1           3c-84-6a-25-88-bc    dynamic
  192.168.0.103         ec-fa-5c-bf-cc-c9    dynamic
  192.168.0.106         00-0c-29-d9-35-db    dynamic
  192.168.0.255         ff-ff-ff-ff-ff-ff    static
  224.0.0.22            01-00-5e-00-00-16    static
  224.0.0.251           01-00-5e-00-00-fb    static
  224.0.0.252           01-00-5e-00-00-fc    static
  239.255.102.18        01-00-5e-7f-66-12    static
  239.255.255.250       01-00-5e-7f-ff-fa    static
  255.255.255.255       ff-ff-ff-ff-ff-ff    static

Interface: 192.168.77.1 --- 0x5
  Internet Address      Physical Address      Type
  192.168.77.255        ff-ff-ff-ff-ff-ff    static
  224.0.0.22            01-00-5e-00-00-16    static
  224.0.0.251           01-00-5e-00-00-fb    static
  224.0.0.252           01-00-5e-00-00-fc    static
  239.255.255.250       01-00-5e-7f-ff-fa    static
  255.255.255.255       ff-ff-ff-ff-ff-ff    static

Interface: 192.168.15.1 --- 0x15
  Internet Address      Physical Address      Type
  192.168.15.255        ff-ff-ff-ff-ff-ff    static
  224.0.0.22            01-00-5e-00-00-16    static
  224.0.0.251           01-00-5e-00-00-fb    static
  224.0.0.252           01-00-5e-00-00-fc    static
  239.255.255.250       01-00-5e-7f-ff-fa    static
  255.255.255.255       ff-ff-ff-ff-ff-ff    static

C:\Users\admin>
```

arp -a so these many entries are there save already if I will remove this than it will require more time as no catch memory is stored so it need to map ip address again Looking it we come to know one is dynamic ip address that is of wifi being connected to the device

```
[yashpra1010@localhost Desktop]$ ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.0.106 netmask 255.255.255.0 broadcast 192.168.0.255
    inet6 fe80::d80:8067:12c6:ad0a prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:d9:35:db txqueuelen 1000 (Ethernet)
    RX packets 2703 bytes 625569 (610.9 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 1142 bytes 141211 (137.9 KiB)
    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 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

virbr0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
    inet 192.168.122.1 netmask 255.255.255.0 broadcast 192.168.122.255
    ether 52:54:00:36:f2:5a txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

[yashpra1010@localhost Desktop]$
```

It shows that how many packets are being sent and how many packets are being received:

TX: TRANSMITTED: 1777 bytes

RX: RECEIVED: 1777 bytes Interface configuration:

Consider a situation where your internet connectivity is having problem and you want to see whether all transmitted packets have been received or lost somewhere else.

Whether my ISP is better or on upto which standard so this command will help that how efficient INTERNET SERVICE PROVIDER

### traceroute:

```
[yashpra1010@localhost ~]$ traceroute 8.8.8.8
traceroute to 8.8.8.8 (8.8.8.8), 30 hops max, 60 byte packets
 1 _gateway (192.168.0.1) 1.115 ms 0.969 ms 0.872 ms
 2 10.230.192.1 (10.230.192.1) 6.103 ms 6.061 ms 6.002 ms
 3 * * *
 4 * * *
 5 103.241.47.142 (103.241.47.142) 14.099 ms 14.067 ms 14.010 ms
 6 103.241.47.206 (103.241.47.206) 13.944 ms 17.715 ms 17.631 ms
 7 142.250.47.236 (142.250.47.236) 16.244 ms 14.479 ms 14.416 ms
 8 10.252.183.94 (10.252.183.94) 14.359 ms 10.252.253.126 (10.252.253.126) 22.637 ms 1
0.252.212.126 (10.252.212.126) 22.621 ms
 9 dns.google (8.8.8.8) 13.978 ms 13.813 ms 17.591 ms
[yashpra1010@localhost ~]$
```



**ss command:**

It is similar to netstat utility used to display **network** connections for the TCP/UDP, **network** protocol statistics, interface statistics, routing tables, masquerade connections, multicast memberships etc. netstat program is obsolete now and its replacement is **ss**.

TCP: consider a situation where we want to acknowledge a connection-oriented request

UDP: without acknowledgement

```
[yashpra1010@localhost ~]$ ss
Netid  State  Recv-Q  Send-Q                               Local Address:
Port                                     Peer Address:Port
u_str  ESTAB   0        0                               *
49358                                     * 49359
u_str  ESTAB   0        0                               /run/user/1000/bus
40850                                     * 43885
u_str  ESTAB   0        0                               *
40656                                     * 44250
u_str  ESTAB   0        0                               /run/systemd/journal/stdout
28319                                     * 26900
u_str  ESTAB   0        0                               *
45631                                     * 45635
u_str  ESTAB   0        0                               *
45437                                     * 45438
u_str  ESTAB   0        0                               *
35647                                     * 37601
u_str  ESTAB   0        0                               /run/systemd/journal/stdout
44890                                     * 45688
u_str  ESTAB   0        0                               /run/systemd/journal/stdout
46398                                     * 46396
u_str  ESTAB   0        0                               /run/user/1000/bus
42836                                     * 44123
u_str  ESTAB   0        0                               /run/dbus/system_bus_socket
40130                                     * 41091
u_str  ESTAB   0        0                               @/tmp/dbus-HJNqGt0uLB
41988                                     * 41987
u_str  ESTAB   0        0                               /run/dbus/system_bus_socket
```

**ss -ta**

```
[yashpra1010@localhost ~]$ ss -ta
State  Recv-Q  Send-Q  Local Address:Port  Peer Address:Port
LISTEN  0        128      0.0.0.0:sunrpc      0.0.0.0:*
LISTEN  0        32      192.168.122.1:domain 0.0.0.0:*
LISTEN  0        128      0.0.0.0:ssh         0.0.0.0:*
LISTEN  0        5       127.0.0.1:ipp       0.0.0.0:*
LISTEN  0        128      [::]:sunrpc        [::]:*
LISTEN  0        128      [::]:ssh           [::]:*
LISTEN  0        5       [::]:ipp           [::]:*
[yashpra1010@localhost ~]$
```

**ss -ua**

```
[yashpra1010@localhost ~]$ ss -ua
State  Recv-Q  Send-Q  Local Address:Port  Peer Address:Port
UNCONN 0        0       0.0.0.0:mdns        0.0.0.0:*
UNCONN 0        0       0.0.0.0:56841       0.0.0.0:*
UNCONN 0        0       192.168.122.1:domain 0.0.0.0:*
UNCONN 0        0       0.0.0.0%virbr0:bootps 0.0.0.0:*
ESTAB  0        0       192.168.0.106%ens33:bootpc 192.168.0.1:bootps
UNCONN 0        0       0.0.0.0:sunrpc      0.0.0.0:*
UNCONN 0        0       127.0.0.1:323       0.0.0.0:*
UNCONN 0        0       [::]:mdns          [::]:*
UNCONN 0        0       [::]:60954         [::]:*
UNCONN 0        0       [::]:sunrpc        [::]:*
UNCONN 0        0       [::]:323           [::]:*
UNCONN 0        0       [fe80::d80:8067:12c6:ad0a]%ens33:dhcpv6-client [::]:*
```

**SS -xa**

```
[yashpra1010@localhost ~]$ ss -xa
Netid State  Recv-Q Send-Q                               Peer Address:Port              Local Address:Port
u_str LISTEN 0      5                                * 0                             /var/run/lsm/ipc/simc 26111
u_str LISTEN 0      5                                * 0                             /var/run/lsm/ipc/sim 26113
u_str LISTEN 0     10                                * 0                             @/tmp/dbus-pjbBtIGR 33559
u_str LISTEN 0      5                                * 0                             /run/user/1000/pipewire-0 44037
u_str LISTEN 0     128                                * 0                             /run/rpcbind.sock 19719
u_str LISTEN 0      5                                * 0                             /run/user/1000/pulse/native 44039
u_str LISTEN 0     128                                * 0                             /run/user/1000/bus 44042
u_dgr UNCONN 0      0                                * 0                             /var/run/chrony/chronyd.sock 28940
u_str LISTEN 0     10                                * 0                             @/tmp/dbus-QMtwgDp8 33560
u_seq LISTEN 0     128                                * 0                             /run/systemd/coredump 19727
u_str LISTEN 0     128 @/org/kernel/linux/storage/multipathd 19726
u_str LISTEN 0     128                                * 0                             @/tmp/.ICE-unix/2251 42837
u_str LISTEN 0     128                                * 0                             /var/run/cups/cups.sock 25879
u_str LISTEN 0     128                                * 0                             /run/user/1000/keyring/control 44057
```

**nslookup:**

```
[yashpra1010@localhost ~]$ nslookup google.com
Server:          192.168.0.1
Address:         192.168.0.1#53

Non-authoritative answer:
Name:   google.com
Address: 172.217.160.206
Name:   google.com
Address: 2404:6800:4009:80b::200e

[yashpra1010@localhost ~]$
```

**dig command in linux:**

With the dig command, you can query information about various DNS records, including host addresses, mail exchanges, and name servers. It is the most commonly used tool among system administrators for troubleshooting DNS problems because of its flexibility and ease of use.

```
[yashpra1010@localhost ~]$ dig google.com

;<<>> DiG 9.11.20-RedHat-9.11.20-5.el8_3.1 <<>> google.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53561
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:;, udp: 4096
;; QUESTION SECTION:
;google.com.                IN      A
;; ANSWER SECTION:
google.com.                102     IN      A      172.217.160.206

;; Query time: 1 msec
;; SERVER: 192.168.0.1#53(192.168.0.1)
;; WHEN: Wed Apr 07 11:03:52 IST 2021
;; MSG SIZE rcvd: 55

[yashpra1010@localhost ~]$
```

We can also use dig command in another form: dig google and host id:

```
[yashpra1010@localhost ~]$ dig 127.0.0.53

; <<>> DiG 9.11.20-RedHat-9.11.20-5.el8_3.1 <<>> 127.0.0.53
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NXDOMAIN, id: 7038
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;127.0.0.53.                IN      A
;; AUTHORITY SECTION:
.                3600    IN      SOA      a.root-servers.net. nstld.verisign-grs.co
m. 2021040601 1800 900 604800 86400

;; Query time: 140 msec
;; SERVER: 192.168.0.1#53(192.168.0.1)
;; WHEN: Wed Apr 07 11:05:05 IST 2021
;; MSG SIZE rcvd: 114

[yashpra1010@localhost ~]$
```