

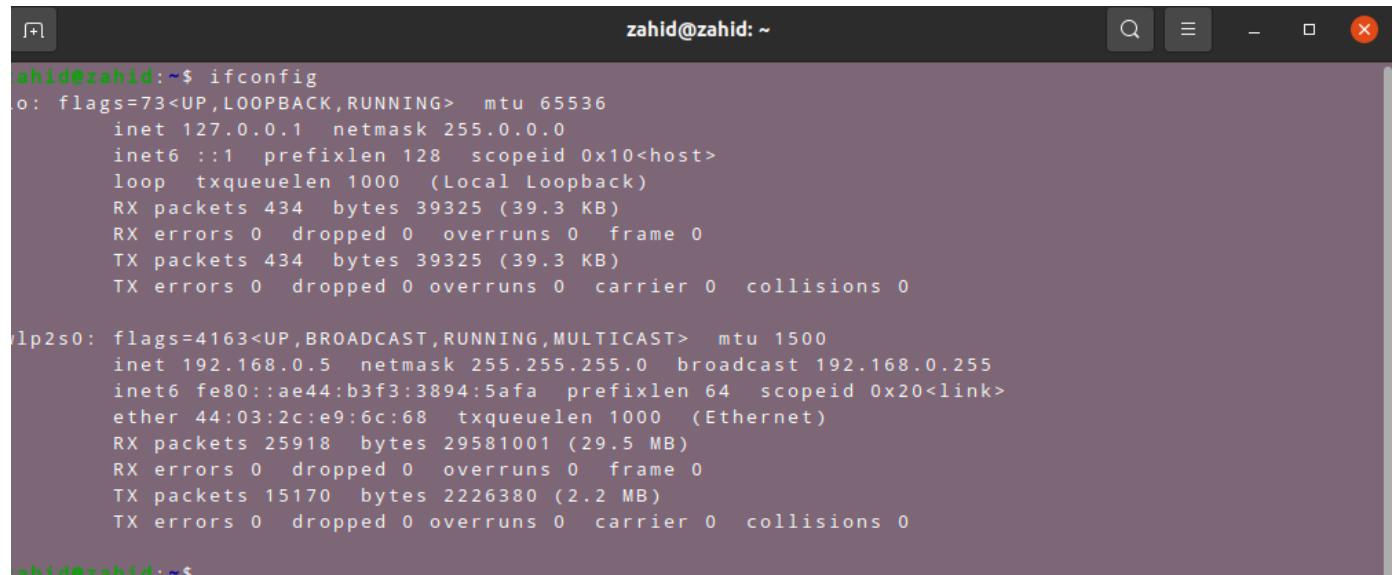
## Linux Network Tools

There are some common linux networking tools given bellow :

### Ifconfig:

The command ifconfig stands for interface configurator. This command enables us to initialize an interface, assign IP address, enable or disable an interface. It display route and network interface.

A newer version of ifconfig is ip command. ifconfig command works for all the versions.



The screenshot shows a terminal window with a dark background and light-colored text. The title bar says "zahid@zahid: ~". The command "ifconfig" is run, and its output is displayed. The output shows two network interfaces: "lo" (loopback) and "lp2s0" (Ethernet). For each interface, it lists flags, MTU, queueing discipline, and various statistics like RX/TX bytes, errors, and collisions. The "lp2s0" interface also shows an IPv4 and IPv6 address configuration.

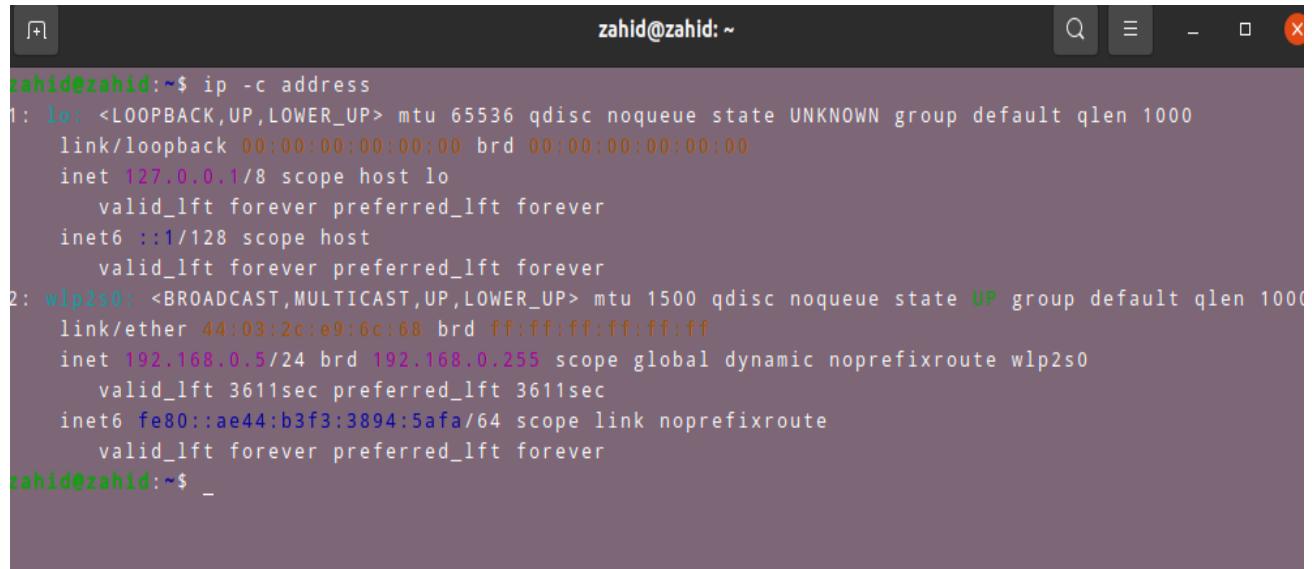
```
zahid@zahid:~$ ifconfig
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
      inet 127.0.0.1 netmask 255.0.0.0
      inet6 ::1 prefixlen 128 scopeid 0x10<host>
          loop txqueuelen 1000 (Local Loopback)
          RX packets 434 bytes 39325 (39.3 KB)
          RX errors 0 dropped 0 overruns 0 frame 0
          TX packets 434 bytes 39325 (39.3 KB)
          TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lp2s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
      inet 192.168.0.5 netmask 255.255.255.0 broadcast 192.168.0.255
      inet6 fe80::ae44:b3f3:3894:5afa prefixlen 64 scopeid 0x20<link>
          ether 44:03:2c:e9:6c:68 txqueuelen 1000 (Ethernet)
          RX packets 25918 bytes 29581001 (29.5 MB)
          RX errors 0 dropped 0 overruns 0 frame 0
          TX packets 15170 bytes 2226380 (2.2 MB)
          TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

zahid@zahid:~$
```

## IP:

Linux IP command is the newer version of the ifconfig command. It is a handy tool for configuring the network interfaces for Linux administrators. It can be used to assign and remove addresses, take the interfaces up or down, and much more useful tasks.



```
zahid@zahid:~$ ip -c address
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: wlp2s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default qlen 1000
    link/ether 44:03:2c:e9:6c:68 brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.5/24 brd 192.168.0.255 scope global dynamic noprefixroute wlp2s0
        valid_lft 3611sec preferred_lft 3611sec
    inet6 fe80::ae44:b3f3:3894:5afa/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
zahid@zahid:~$ _
```

## Ipcalc:

**Ipcalc** actually does a lot more – it takes an IP address and netmask and provides the resulting broadcast, network, Cisc

o wildcard mask, and host range. You can also use it as a teaching tool to present subnetting results in an easy to understand binary values. Some of the uses of **ipcalc** are:

- Validate IP address
- Show calculated broadcast address
- Display hostname determined via DNS
- Display network address or prefix

```
zahid@zahid:~$ ipcalc 192.168.0.1
address: 192.168.0.1          11000000.10101000.00000000. 00000001
netmask: 255.255.255.0 = 24  11111111.11111111.11111111. 00000000
wildcard: 0.0.0.255          00000000.00000000.00000000. 11111111
>
network: 192.168.0.0/24      11000000.10101000.00000000. 00000000
hostMin: 192.168.0.1         11000000.10101000.00000000. 00000001
hostMax: 192.168.0.254       11000000.10101000.00000000. 11111110
broadcast: 192.168.0.255     11000000.10101000.00000000. 11111111
hosts/Net: 254                Class C, Private Internet

zahid@zahid:~$ _
```

## Iwconfig:

**iwconfig** command in Linux is like **ifconfig** command, in the sense it works with kernel-resident network interface but it is dedicated to wireless networking interfaces only. It is used to set the parameters of the network interface that are particular to the wireless operation like SSID, frequency etc. *iwconfig* may also be used to display the parameters, and the wireless statistics which are extracted from */proc/net/wireles*

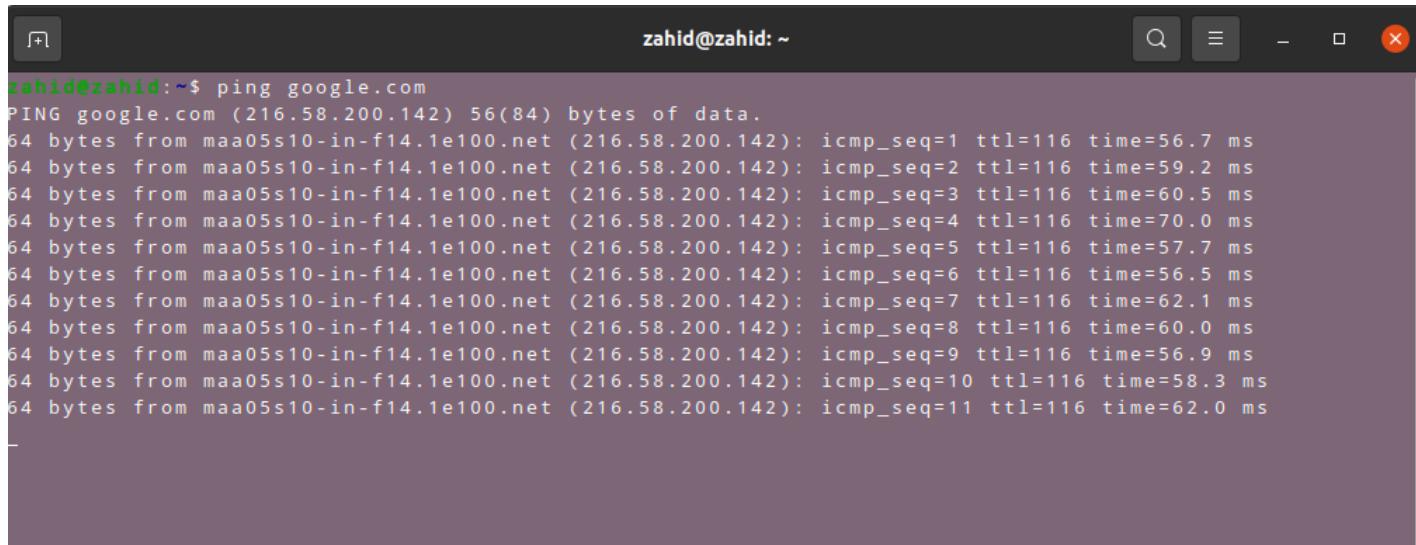
```
zahid@zahid:~$ iwconfig
wlp2s0    IEEE 802.11  ESSID:"Zoha Wi-Fi"
          Mode:Managed  Frequency:2.417 GHz  Access Point: 00:AD:24:F1:42:7D
          Bit Rate=300 Mb/s   Tx-Power=22 dBm
          Retry short limit:7   RTS thr:off   Fragment thr:off
          Power Management:on
          Link Quality=51/70  Signal level=-59 dBm
          Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
          Tx excessive retries:98  Invalid misc:1625    Missed beacon:0

lo        no wireless extensions.

zahid@zahid:~$ _
```

## Ping:

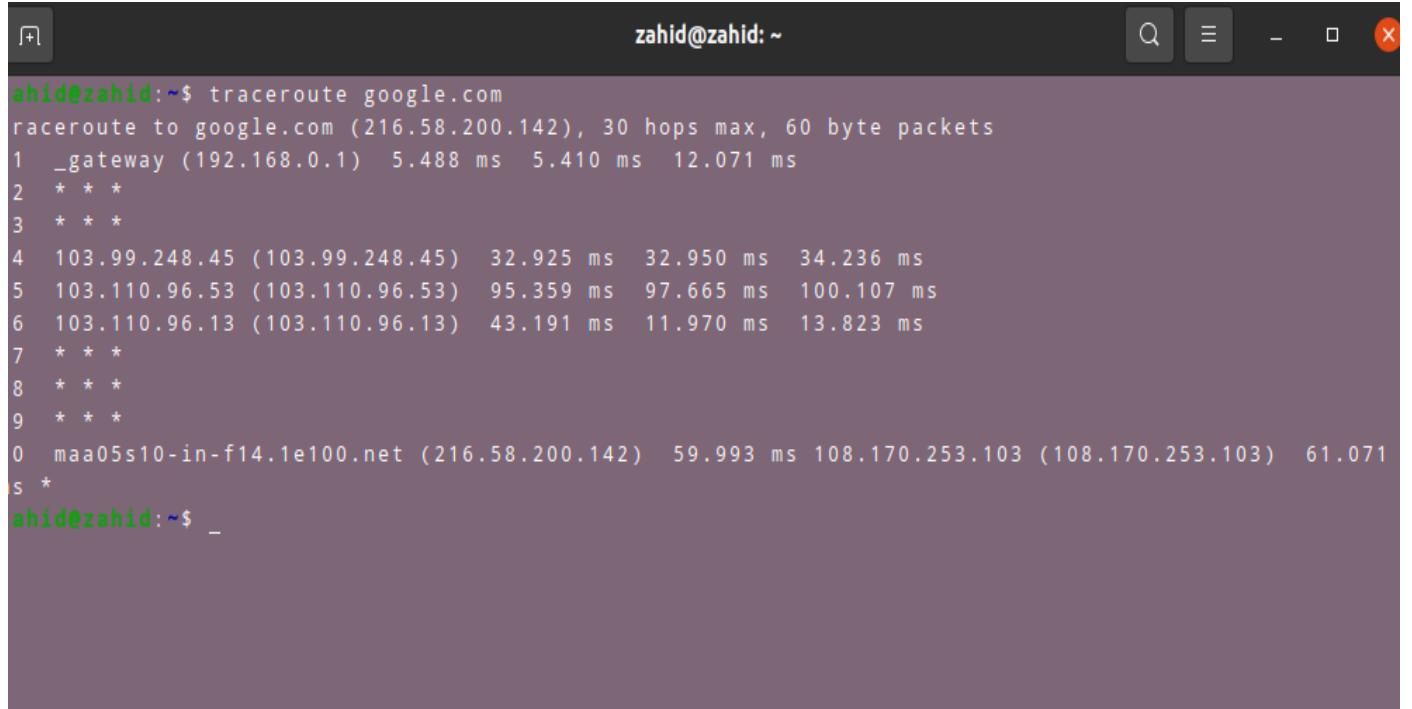
Ping command stands for (Packet Internet Groper). It checks connectivity between two nodes to see if a server is available. It sends ICMP ECHO\_REQUEST packets to network hosts and displays the data on the remote server's response. It checks if a remote host is up, or that network interfaces can be reached. Further, it is used to check if a network connection is available between two devices. It is also handy tool for checking your network connection and verifying network issues.



```
zahid@zahid:~$ ping google.com
PING google.com (216.58.200.142) 56(84) bytes of data.
64 bytes from maa05s10-in-f14.1e100.net (216.58.200.142): icmp_seq=1 ttl=116 time=56.7 ms
64 bytes from maa05s10-in-f14.1e100.net (216.58.200.142): icmp_seq=2 ttl=116 time=59.2 ms
64 bytes from maa05s10-in-f14.1e100.net (216.58.200.142): icmp_seq=3 ttl=116 time=60.5 ms
64 bytes from maa05s10-in-f14.1e100.net (216.58.200.142): icmp_seq=4 ttl=116 time=70.0 ms
64 bytes from maa05s10-in-f14.1e100.net (216.58.200.142): icmp_seq=5 ttl=116 time=57.7 ms
64 bytes from maa05s10-in-f14.1e100.net (216.58.200.142): icmp_seq=6 ttl=116 time=56.5 ms
64 bytes from maa05s10-in-f14.1e100.net (216.58.200.142): icmp_seq=7 ttl=116 time=62.1 ms
64 bytes from maa05s10-in-f14.1e100.net (216.58.200.142): icmp_seq=8 ttl=116 time=60.0 ms
64 bytes from maa05s10-in-f14.1e100.net (216.58.200.142): icmp_seq=9 ttl=116 time=56.9 ms
64 bytes from maa05s10-in-f14.1e100.net (216.58.200.142): icmp_seq=10 ttl=116 time=58.3 ms
64 bytes from maa05s10-in-f14.1e100.net (216.58.200.142): icmp_seq=11 ttl=116 time=62.0 ms
```

## Traceroute:

Traceroute command is a network troubleshooting utility that helps us determine the number of hops and packets traveling path required to reach a destination. It is used to display how the data transmitted from a local machine to a remote machine. Loading a web page is one of the common examples of the traceroute. A web page loading transfers data through a network and routers. The traceroute can display the routes, [IP](#) addresses, and hostnames of routers over a network. It can be useful for diagnosing network issues.



A screenshot of a terminal window titled "zahid@zahid: ~". The window shows the output of the "traceroute google.com" command. The output details the path from the user's machine to Google's IP address (216.58.200.142) through various routers and an ISP gateway. The traceroute shows 10 hops, with the last hop being the ISP's gateway. The command "traceroute google.com" is entered at the prompt, followed by the output results.

```
zahid@zahid:~$ traceroute google.com
traceroute to google.com (216.58.200.142), 30 hops max, 60 byte packets
1 _gateway (192.168.0.1)  5.488 ms  5.410 ms  12.071 ms
2 * * *
3 * * *
4 103.99.248.45 (103.99.248.45)  32.925 ms  32.950 ms  34.236 ms
5 103.110.96.53 (103.110.96.53)  95.359 ms  97.665 ms  100.107 ms
6 103.110.96.13 (103.110.96.13)  43.191 ms  11.970 ms  13.823 ms
7 * * *
8 * * *
9 * * *
0 maa05s10-in-f14.1e100.net (216.58.200.142)  59.993 ms  108.170.253.103 (108.170.253.103)  61.071
s *
zahid@zahid:~$ _
```

## Netstat:

Netstat command stands for Network statistics. It displays information about different interface statistics, including open sockets, routing tables, and connection information. Further, it can be used to displays all the socket connections (including TCP, UDP). Apart from connected sockets, it also displays the sockets that are pending for connections. It is a handy tool for network and system administrators.

```
zahid@zahid:~$ sudo netstat -aptu
active Internet connections (servers and established)
proto Recv-Q Send-Q Local Address          Foreign Address        State      PID/Program name
tcp      0      0 localhost:domain        0.0.0.0:*              LISTEN     846/systemd-resolve
tcp      0      0 localhost:ipp           0.0.0.0:*              LISTEN     962/cupsd
tcp      0      0 localhost:mysql         0.0.0.0:*              LISTEN     1696/mysql
tcp      0      0 zahid:53522            1b-140-82-113-25-:https ESTABLISHED 3639/chrome --type=maa03s31-in-f14.1:https ESTABLISHED 3639/chrome --type=maa03s21-in-f78.1:https ESTABLISHED 3639/chrome --type=maa05s04-in-f10.1:https ESTABLISHED 3639/chrome --type=maa03s23-in-f202.:https ESTABLISHED 3639/chrome --type=maa03s31-in-f3.1e:https ESTABLISHED 3639/chrome --type=maa05s13-in-f10.1:https ESTABLISHED 3639/chrome --type=maa05s04-in-f10.1:https ESTABLISHED 3639/chrome --type=s-a-in-f188.1e100.n:5228 ESTABLISHED 3639/chrome --type=74.125.24.189:https ESTABLISHED 3639/chrome --type=maa05s13-in-f10.1:https ESTABLISHED 3639/chrome --type=[::]:*                LISTEN     962/cupsd
tcp6     0      0 [::]:33060             [::]:*                  LISTEN     1696/mysql
dp      0      0 localhost:domain        0.0.0.0:*              LISTEN     846/systemd-resolve
dp      0      0 zahid:bootpc          _gateway:bootps        ESTABLISHED 878/NetworkManager
dp      0      0 0.0.0.0:631           0.0.0.0:*              LISTEN     996/cups-browsed
dp      0      0 0.0.0.0:33771          0.0.0.0:*              LISTEN     873/avahi-daemon: r
dp      0      0 224.0.0.251:mdns       0.0.0.0:*              LISTEN     3471/chrome --no-de
dp      0      0 224.0.0.251:mdns       0.0.0.0:*              LISTEN     3639/chrome --type=0.0.0.0:mdns          0.0.0.0:*              LISTEN     873/avahi-daemon: r
dp6     0      0 [::]:mdns             [::]:*                  LISTEN     873/avahi-daemon: r
dn6     0      0 [::]:42645             [::]:*                  LISTEN     873/avahi-daemon: r
```

## Curl:

Linux curl command is used to download or upload data to a server via supported protocols such as HTTP, FTP, IMAP, SFTP, TFTP, IMAP, POP3, SCP, etc. It is a remote utility, so it works without user interaction.

The data transfer from one place to another is one of the vital and most used tasks of a computer system. However, there are many [GUI](#) tools available for data transfer. But, when working on the command-line, it becomes a bit complicated. The curl utility allows us to transfer data via the command line

```
zahid@zahid:~$ curl mbstu.ac.bd
<!DOCTYPE html>
<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
        <title>MBSTU | Home</title>
        <link rel="stylesheet" href="nivo-slider/themes/default/default.css" type="text/css" media="screen" />
        <link rel="stylesheet" href="nivo-slider/nivo-slider.css" type="text/css" media="screen" />
        <link rel="stylesheet" href="nivo-slider/demo/style.css" type="text/css" media="screen" />
            <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/font-awesome/4.4.0/css/font-awesome.min.css">

            <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js"></script>

            <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.css" type="text/css" media="screen" />
            <link href="assets/css/countdown.css" rel="stylesheet" type="text/css" />
            <link href="style/main_layout.css" rel="stylesheet" type="text/css" />
            <link href="images/mbstu.ico" rel="shortcut icon" type="image/x-icon" />
            <link href="images/mbstu.ico" rel="icon" type="image/x-icon" />

<style>
.mid {
    float: left;
    width: 515px;
```

## Whois:

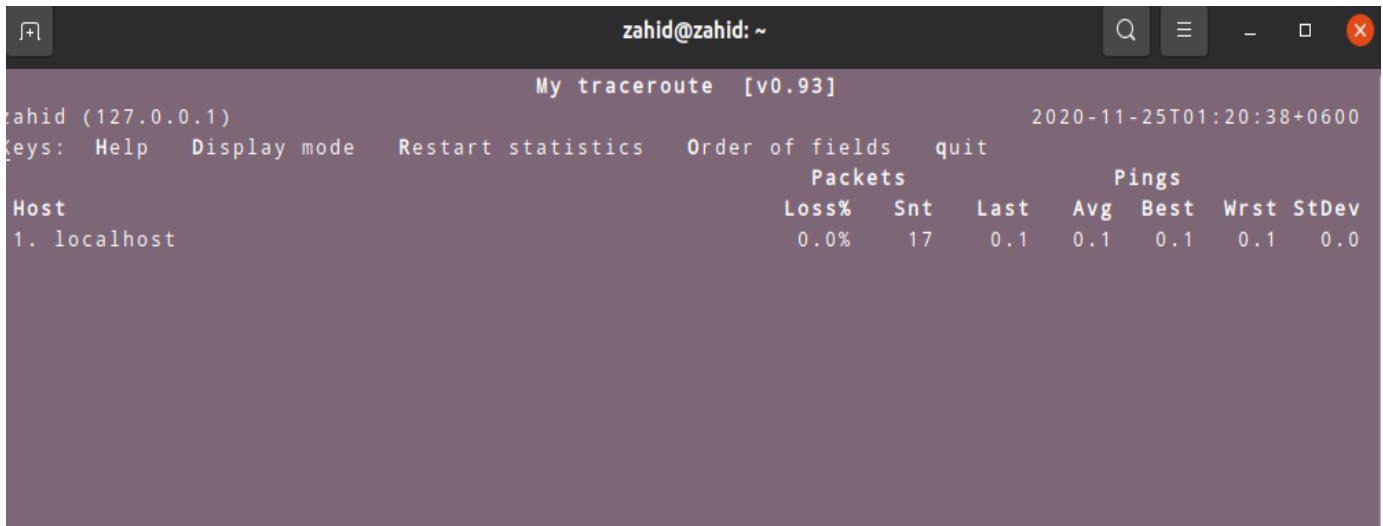
**WHOIS** (pronounced as the phrase "**who is**") is a query and response protocol that is widely used for querying databases that store the registered users or assignees of an Internet resource, such as a domain name, an IP address block or an autonomous system, but is also used for a

```
zahid@zahid:~$ whois google.com
Domain Name: GOOGLE.COM
Registry Domain ID: 2138514_DOMAIN_COM-VRSN
Registrar WHOIS Server: whois.markmonitor.com
Registrar URL: http://www.markmonitor.com
Updated Date: 2019-09-09T15:39:04Z
Creation Date: 1997-09-15T04:00:00Z
Registry Expiry Date: 2028-09-14T04:00:00Z
Registrar: MarkMonitor Inc.
Registrar IANA ID: 292
Registrar Abuse Contact Email: abusecomplaints@markmonitor.com
Registrar Abuse Contact Phone: +1.2083895740
Domain Status: clientDeleteProhibited https://icann.org/epp#clientDeleteProhibited
Domain Status: clientTransferProhibited https://icann.org/epp#clientTransferProhibited
Domain Status: clientUpdateProhibited https://icann.org/epp#clientUpdateProhibited
Domain Status: serverDeleteProhibited https://icann.org/epp#serverDeleteProhibited
Domain Status: serverTransferProhibited https://icann.org/epp#serverTransferProhibited
Domain Status: serverUpdateProhibited https://icann.org/epp#serverUpdateProhibited
Name Server: NS1.GOOGLE.COM
Name Server: NS2.GOOGLE.COM
Name Server: NS3.GOOGLE.COM
Name Server: NS4.GOOGLE.COM
DNSSEC: unsigned
URL of the ICANN Whois Inaccuracy Complaint Form: https://www.icann.org/wicf/
>>> Last update of whois database: 2020-11-24T19:19:19Z <<<
For more information on Whois status codes, please visit https://icann.org/epp
```

wider range of other information.

## Mtr:

The mtr command is a combination of ping and traceroute commands. It is a network diagnostic tool that continuously sends packets showing ping time for each hop. It also displays network problems of the entire route taken by the network packets.



A screenshot of a terminal window titled "My traceroute [v0.93]" running on a Linux system. The window shows the command-line interface for the Mtr tool. At the top, it displays the user "zahid@zahid: ~". Below the title, there is a help menu with keys: Help, Display mode, Restart statistics, Order of fields, quit. The main output shows a single host entry: "1. localhost". The table below the host entry has two columns: "Packets" and "Pings". The "Packets" column includes "Loss%" and other metrics like Snt, Last, Avg, Best, Wrst, and StDev. The "Pings" column shows values of 0.0% loss, 17 sent packets, 0.1 ms average, 0.1 ms best, 0.1 ms wrst, and 0.0 ms standard deviation. The timestamp at the top right of the window is 2020-11-25T01:20:38+0600.

	Packets	Pings
Host	Loss%	Snt Last Avg Best Wrst StDev
1. localhost	0.0%	17 0.1 0.1 0.1 0.1 0.0

## Host:

Linux host command displays domain name for given IP address or vice-versa. It also performs DNS lookups related to the DNS query. The host command's default behavior displays a summary of its command-line arguments and supported options.



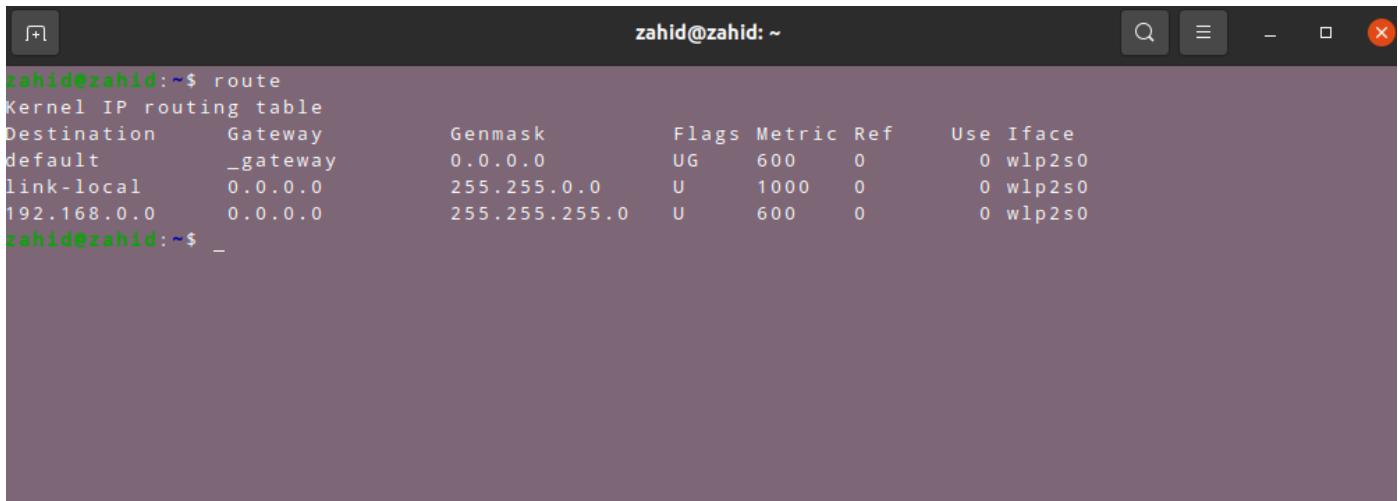
A screenshot of a terminal window titled "zahid@zahid: ~". The window contains the following text output from the "host" command:

```
zahid@zahid:~$ host google.com
google.com has address 216.58.200.142
google.com has IPv6 address 2404:6800:4007:808::200e
google.com mail is handled by 40 alt3.aspmx.l.google.com.
google.com mail is handled by 50 alt4.aspmx.l.google.com.
google.com mail is handled by 10 aspmx.l.google.com.
google.com mail is handled by 20 alt1.aspmx.l.google.com.
google.com mail is handled by 30 alt2.aspmx.l.google.com.
zahid@zahid:~$ _
```

# Route:

The route command displays and manipulate IP routing table for your system.

A router is a device which is basically used to determine the best way to route packets to a destination.



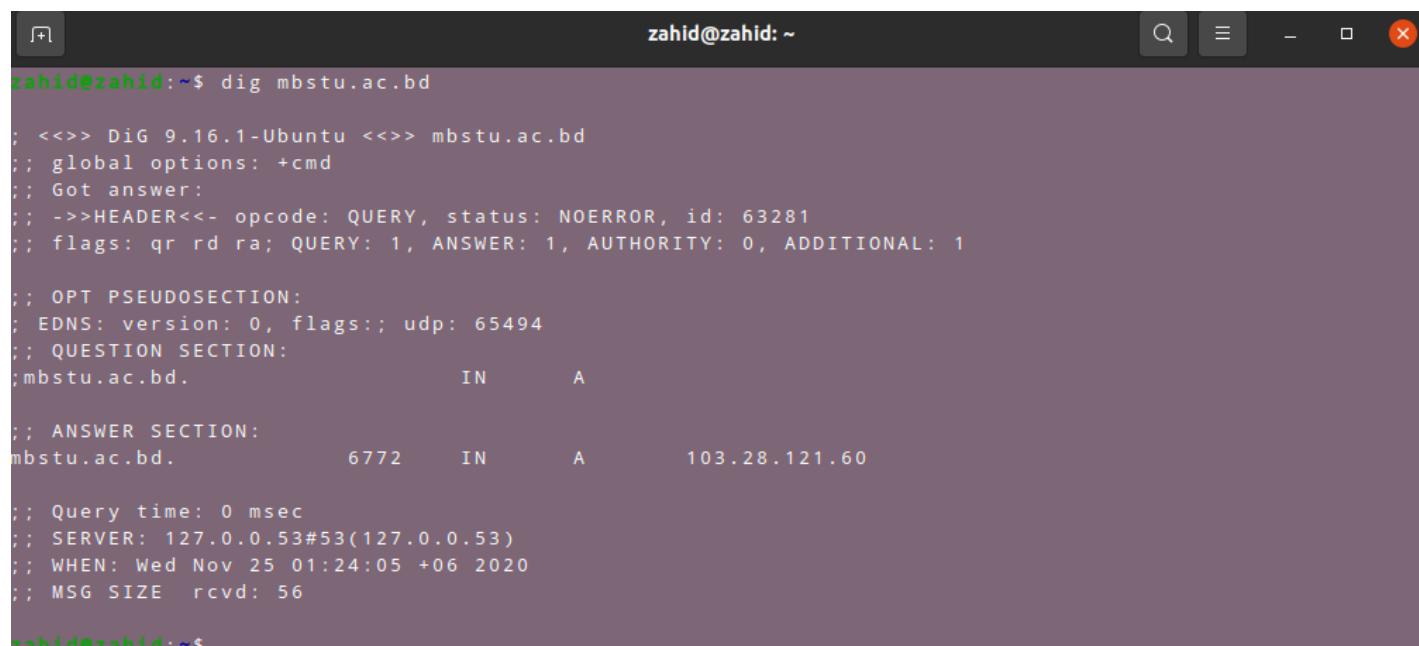
A screenshot of a terminal window titled "zahid@zahid: ~". The window shows the output of the "route" command. The output is as follows:

```
Kernel IP routing table
Destination      Gateway          Genmask        Flags Metric Ref    Use Iface
default         _gateway        0.0.0.0        UG     600    0        0 wlp2s0
link-local      0.0.0.0        255.255.0.0    U      1000   0        0 wlp2s0
192.168.0.0    0.0.0.0        255.255.255.0  U      600    0        0 wlp2s0
```

## Dig:

Linux dig command stands for Domain Information Groper. This command is used for tasks related to DNS lookup to query DNS name servers. It mainly deals with troubleshooting DNS related problems. It is a flexible utility for examining the DNS (Domain Name Servers). It is used to perform the DNS lookups and returns the queried answers from the name server. Usually, it is used by most DNS administrators to troubleshoot the DNS problems. It is a straightforward tool and provides a clear output. It is more functional than other lookups tools.

The dig command supports plenty of command-line options. Additionally, it facilitates batch mode, which is useful for accessing the lookup requests from a file. If it is not specified to the dig command to query a specific name server, it will access each of the servers from "/etc/resolv.conf." The dig without any command-line options will perform an NS query for "." (the root).



```
zahid@zahid:~$ dig mbstu.ac.bd

; <>> DiG 9.16.1-Ubuntu <>> mbstu.ac.bd
; global options: +cmd
; Got answer:
; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 63281
; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 65494
; QUESTION SECTION:
;mbstu.ac.bd.           IN      A

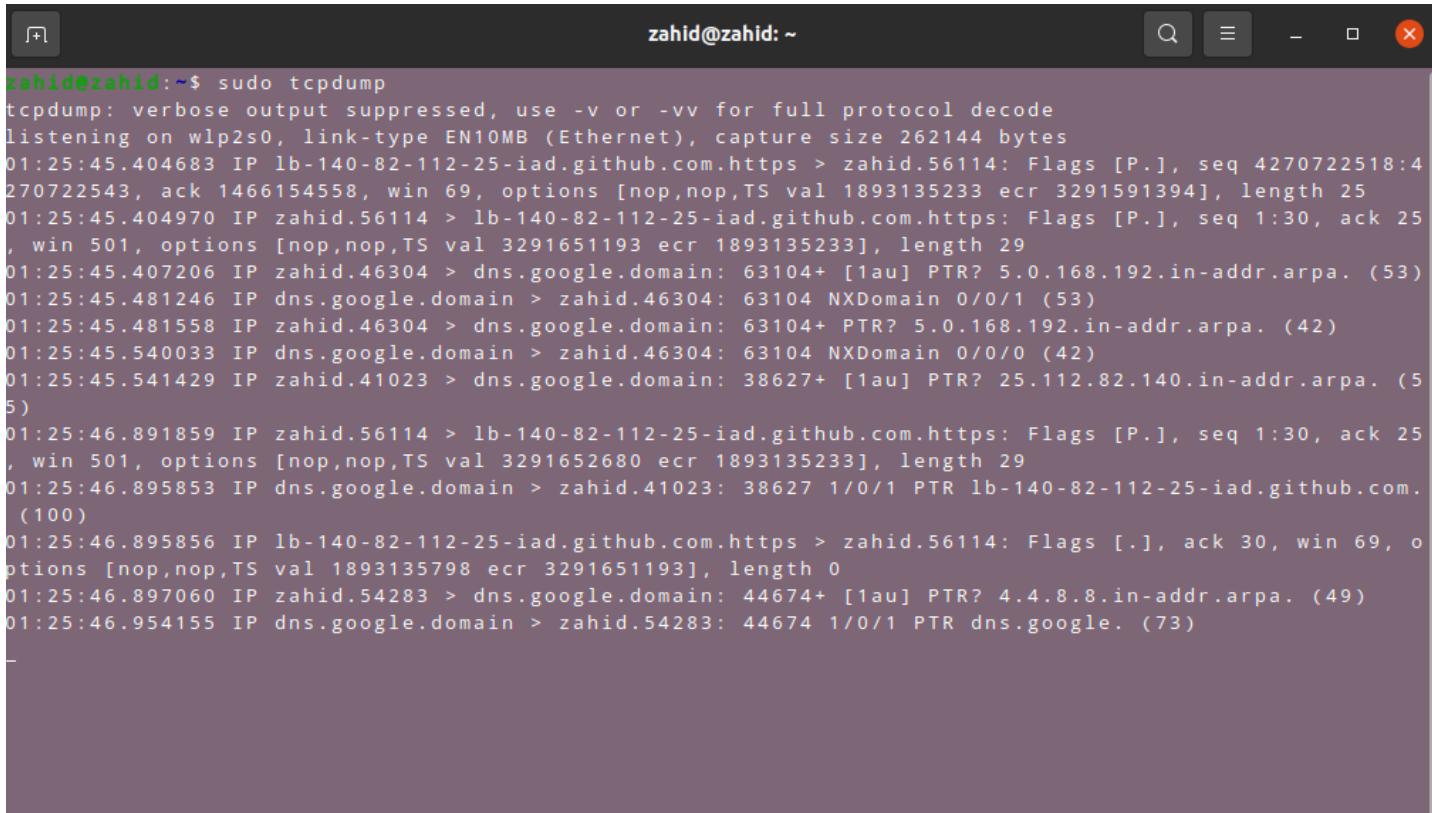
;ANSWER SECTION:
mbstu.ac.bd.       6772    IN      A      103.28.121.60

; Query time: 0 msec
; SERVER: 127.0.0.53#53(127.0.0.53)
; WHEN: Wed Nov 25 01:24:05 +06 2020
; MSG SIZE  rcvd: 56

zahid@zahid:~$
```

# Tcpdump:

Tcpdump is a command line utility that allows you to capture and analyze network traffic going through your system. It is often used to help troubleshoot network issues, as well as a security tool. A powerful and versatile tool that includes many options and filters, tcpdump can be used in a variety of cases. Since it's a command line tool, it is ideal to run in remote servers or devices for which a GUI is not available, to collect data that can be analyzed later. It can also be launched in the background or as a scheduled job using tools like cron.



A screenshot of a terminal window titled "zahid@zahid: ~". The window shows the output of a "tcpdump" command run with "sudo". The output displays network traffic captured on the "wlp2s0" interface, specifically showing DNS queries and responses between the user's machine and external hosts like "lb-140-82-112-25-iad.github.com" and "dns.google.domain". The traffic includes various TCP flags, sequence numbers, and timestamps.

```
zahid@zahid:~$ sudo tcpdump
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on wlp2s0, link-type EN10MB (Ethernet), capture size 262144 bytes
01:25:45.404683 IP lb-140-82-112-25-iad.github.com.https > zahid.56114: Flags [P.], seq 4270722518:4270722543, ack 1466154558, win 69, options [nop,nop,TS val 1893135233 ecr 3291591394], length 25
01:25:45.404970 IP zahid.56114 > lb-140-82-112-25-iad.github.com.https: Flags [P.], seq 1:30, ack 25, win 501, options [nop,nop,TS val 3291651193 ecr 1893135233], length 29
01:25:45.407206 IP zahid.46304 > dns.google.domain: 63104+ [1au] PTR? 5.0.168.192.in-addr.arpa. (53)
01:25:45.481246 IP dns.google.domain > zahid.46304: 63104 NXDomain 0/0/1 (53)
01:25:45.481558 IP zahid.46304 > dns.google.domain: 63104+ PTR? 5.0.168.192.in-addr.arpa. (42)
01:25:45.540033 IP dns.google.domain > zahid.46304: 63104 NXDomain 0/0/0 (42)
01:25:45.541429 IP zahid.41023 > dns.google.domain: 38627+ [1au] PTR? 25.112.82.140.in-addr.arpa. (55)
01:25:46.891859 IP zahid.56114 > lb-140-82-112-25-iad.github.com.https: Flags [P.], seq 1:30, ack 25, win 501, options [nop,nop,TS val 3291652680 ecr 1893135233], length 29
01:25:46.895853 IP dns.google.domain > zahid.41023: 38627 1/0/1 PTR lb-140-82-112-25-iad.github.com. (100)
01:25:46.895856 IP lb-140-82-112-25-iad.github.com.https > zahid.56114: Flags [.], ack 30, win 69, options [nop,nop,TS val 1893135798 ecr 3291651193], length 0
01:25:46.897060 IP zahid.54283 > dns.google.domain: 44674+ [1au] PTR? 4.4.8.8.in-addr.arpa. (49)
01:25:46.954155 IP dns.google.domain > zahid.54283: 44674 1/0/1 PTR dns.google. (73)
```