

EXPERIMENT 2

NAME: Vishal Shashikant Salvi

UID: 2019230069

CLASS: TE COMPS

BATCH: C

CEL 51, DCCN, Monsoon 2020

Lab 2: Basic Network Utilities

Aim: To study Basic Network Utilities

This lab introduces some basic network monitoring/analysis tools. There are a few exercises along the way. You should write up answers to the ping and traceroute exercises and turn them in next lab. (You should try out each tool, whether it is needed for an exercise or not!).

Prerequisite: Basic understanding of command line utilities of Linux Operating system.

Some Basic command line Networking utilities

Start with a few of the most basic command line tools. These commands are available on Unix, including Linux (and the first two, at least, are also for Windows). Some parameters or options might differ on different operating systems. Remember that you can use `man <command>` to get information about a command and its options.

Ping^[1]:

The command `ping <host>` sends a series of packets and expects to receive a response to each packet. When a return packet is received, ping reports the round trip time (the time between sending the packet and receiving the response). Some routers and firewalls block ping requests, so you might get no response at all. Ping can be used to check whether a computer is up and running, to measure network delay time, and to check for dropped packets indicating network congestion. Note that `<host>` can be either a domain name or an IP address. By default, ping will send a packet every second indefinitely; stop it with Control-C

Network latency, specifically round-trip time (RTT), can be measured using ping, which sends ICMP packets. The syntax for the command in Linux or Mac OS is:

```
ping [-c <count>] [-s <packetsize>] <hostname>
```

The syntax in Windows is:

```
ping [-n <count>] [-l <packetsize>] <hostname>
```

The default number of ICMP packets to send is either infinite (in Linux and Mac OS) or 4 (in Windows). The default packet size is either 64 bytes (in Linux) or 32 bytes (in Windows). You can specify either a hostname (e.g., `spit.ac.in`) or an IP address.

To save the output from ping to a file, include a greater than symbol and a file name at the end of the command. For example:

```
ping -c 10 google.com > ping_c10_s64_google.log
```

EXPERIMENTS WITH PING

1. Ping the any hosts 10 times (i.e., packet count is 10) with a packet size of 64 bytes, 100 bytes, 500 bytes, 1000 bytes, 1400 bytes

```
C:\Users\Vishal>ping -n 10 -l 64 www.google.com
```

```
Pinging www.google.com [142.250.67.228] with 64 bytes of data:
```

```
Reply from 142.250.67.228: bytes=64 time=70ms TTL=118
```

```
Request timed out.
```

```
Reply from 142.250.67.228: bytes=64 time=4ms TTL=118
```

```
Reply from 142.250.67.228: bytes=64 time=3ms TTL=118
```

```
Reply from 142.250.67.228: bytes=64 time=3ms TTL=118
```

```
Reply from 142.250.67.228: bytes=64 time=6ms TTL=118
```

```
Reply from 142.250.67.228: bytes=64 time=4ms TTL=118
```

```
Reply from 142.250.67.228: bytes=64 time=27ms TTL=118
```

```
Request timed out.
```

```
Reply from 142.250.67.228: bytes=64 time=5ms TTL=118
```

```
Ping statistics for 142.250.67.228:
```

```
    Packets: Sent = 10, Received = 8, Lost = 2 (20% loss),
```

```
Approximate round trip times in milli-seconds:
```

```
    Minimum = 3ms, Maximum = 70ms, Average = 15ms
```

```
C:\Users\Vishal>ping -n 10 -l 100 www.google.com
```

```
Pinging www.google.com [142.250.67.228] with 100 bytes of data:
```

```
Reply from 142.250.67.228: bytes=68 (sent 100) time=9ms TTL=118
```

```
Reply from 142.250.67.228: bytes=68 (sent 100) time=7ms TTL=118
```

```
Reply from 142.250.67.228: bytes=68 (sent 100) time=6ms TTL=118
```

```
Reply from 142.250.67.228: bytes=68 (sent 100) time=4ms TTL=118
```

```
Reply from 142.250.67.228: bytes=68 (sent 100) time=4ms TTL=118
```

```
Reply from 142.250.67.228: bytes=68 (sent 100) time=3ms TTL=118
```

```
Reply from 142.250.67.228: bytes=68 (sent 100) time=10ms TTL=118
```

```
Reply from 142.250.67.228: bytes=68 (sent 100) time=4ms TTL=118
```

```
Reply from 142.250.67.228: bytes=68 (sent 100) time=3ms TTL=118
```

```
Reply from 142.250.67.228: bytes=68 (sent 100) time=4ms TTL=118
```

```
Ping statistics for 142.250.67.228:
```

```
    Packets: Sent = 10, Received = 10, Lost = 0 (0% loss),
```

```
Approximate round trip times in milli-seconds:
```

```
    Minimum = 3ms, Maximum = 10ms, Average = 5ms
```

```
C:\Users\Vishal>ping -n 10 -l 500 www.google.com

Pinging www.google.com [142.250.67.228] with 500 bytes of data:
Reply from 142.250.67.228: bytes=68 (sent 500) time=3ms TTL=118
Reply from 142.250.67.228: bytes=68 (sent 500) time=3ms TTL=118
Reply from 142.250.67.228: bytes=68 (sent 500) time=4ms TTL=118
Reply from 142.250.67.228: bytes=68 (sent 500) time=3ms TTL=118
Reply from 142.250.67.228: bytes=68 (sent 500) time=26ms TTL=118
Reply from 142.250.67.228: bytes=68 (sent 500) time=3ms TTL=118
Reply from 142.250.67.228: bytes=68 (sent 500) time=9ms TTL=118
Reply from 142.250.67.228: bytes=68 (sent 500) time=4ms TTL=118
Reply from 142.250.67.228: bytes=68 (sent 500) time=4ms TTL=118
Reply from 142.250.67.228: bytes=68 (sent 500) time=4ms TTL=118

Ping statistics for 142.250.67.228:
    Packets: Sent = 10, Received = 10, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 3ms, Maximum = 26ms, Average = 6ms

C:\Users\Vishal>ping -n 10 -l 1000 www.google.com

Pinging www.google.com [142.250.67.228] with 1000 bytes of data:
Reply from 142.250.67.228: bytes=68 (sent 1000) time=4ms TTL=118
Reply from 142.250.67.228: bytes=68 (sent 1000) time=5ms TTL=118
Reply from 142.250.67.228: bytes=68 (sent 1000) time=5ms TTL=118
Reply from 142.250.67.228: bytes=68 (sent 1000) time=4ms TTL=118
Reply from 142.250.67.228: bytes=68 (sent 1000) time=4ms TTL=118
Reply from 142.250.67.228: bytes=68 (sent 1000) time=4ms TTL=118
Reply from 142.250.67.228: bytes=68 (sent 1000) time=4ms TTL=118
Reply from 142.250.67.228: bytes=68 (sent 1000) time=5ms TTL=118
Reply from 142.250.67.228: bytes=68 (sent 1000) time=3ms TTL=118
Reply from 142.250.67.228: bytes=68 (sent 1000) time=3ms TTL=118

Ping statistics for 142.250.67.228:
    Packets: Sent = 10, Received = 10, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 3ms, Maximum = 5ms, Average = 4ms
```

```
C:\Users\Vishal>ping -n 10 -l 1400 www.google.com

Pinging www.google.com [142.250.67.228] with 1400 bytes of data:
Reply from 142.250.67.228: bytes=68 (sent 1400) time=4ms TTL=118
Reply from 142.250.67.228: bytes=68 (sent 1400) time=4ms TTL=118
Reply from 142.250.67.228: bytes=68 (sent 1400) time=3ms TTL=118
Reply from 142.250.67.228: bytes=68 (sent 1400) time=3ms TTL=118
Reply from 142.250.67.228: bytes=68 (sent 1400) time=4ms TTL=118
Reply from 142.250.67.228: bytes=68 (sent 1400) time=4ms TTL=118
Reply from 142.250.67.228: bytes=68 (sent 1400) time=5ms TTL=118
Reply from 142.250.67.228: bytes=68 (sent 1400) time=4ms TTL=118
Reply from 142.250.67.228: bytes=68 (sent 1400) time=4ms TTL=118
Reply from 142.250.67.228: bytes=68 (sent 1400) time=4ms TTL=118

Ping statistics for 142.250.67.228:
    Packets: Sent = 10, Received = 10, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 3ms, Maximum = 5ms, Average = 3ms

C:\Users\Vishal>
```

QUESTIONS ABOUT LATENCY

Now look at the results you gathered and answer the following questions about latency. Store your answers in a file named ping.txt.

1. Does the average RTT vary between different hosts? What aspects of latency (transmit, propagation, and queueing delay) might impact this and why?

Yes.

Transmission Delay:

Time taken to put a packet onto link. In other words, it is simply time required to put data bits on the wire/communication medium. It depends on length of packet and bandwidth of network.

Transmission Delay = Data size / bandwidth = (L/B) second

Propagation delay:

Time taken by the first bit to travel from sender to receiver end of the link. In other words, it is simply the time required for bits to reach the destination from the start point. Factors on which Propagation delay depends are Distance and propagation speed.

Propagation delay = distance/transmission speed = d/s

Queueing Delay :

Queueing delay is the time a job waits in a queue until it can be executed. It depends on congestion. It is the time difference between when the packet arrived Destination and when the packet data was processed or executed. It may be caused by mainly three reasons i.e. originating switches, intermediate switches or call receiver servicing switches.

- Distance – The length a signal has to travel correlates with the time taken for a request to reach a server and a response to reach a browser.
- Transmission medium – The medium used to route a signal (e.g., copper wire, fiber optic cables) can impact how quickly a request is received by a server and routed back to a user.
- Number of network hops – Intermediate routers or servers take time to process a signal, increasing RTT. The more hops a signal has to travel through, the higher the RTT.
- Traffic levels – RTT typically increases when a network is congested with high levels of traffic. Conversely, low traffic times can result in decreased RTT.

Server response time – The time taken for a target server to respond to a request depends on its processing capacity, the number of requests being handled and the

nature of the request (i.e., how much server-side work is required). A longer server response time increases RTT.

2. Does the average RTT vary with different packet sizes? What aspects of latency (transmit, propagation, and queueing delay) might impact this and why?

Yes, we can say that the Round Trip Time is impacted due to the difference in the size of the packets. This is because of the Transmission delay and the Queueing delay which depend on the size of the packets.

```
C:\Users\Vishal>ping

Usage: ping [-t] [-a] [-n count] [-l size] [-f] [-i TTL] [-v TOS]
           [-r count] [-s count] [[-j host-list] | [-k host-list]]
           [-w timeout] [-R] [-S srcaddr] [-c compartment] [-p]
           [-4] [-6] target_name

Options:
  -t             Ping the specified host until stopped.
                  To see statistics and continue - type Control-Break;
                  To stop - type Control-C.
  -a             Resolve addresses to hostnames.
  -n count       Number of echo requests to send.
  -l size        Send buffer size.
  -f            Set Don't Fragment flag in packet (IPv4-only).
  -i TTL         Time To Live.
  -v TOS         Type Of Service (IPv4-only. This setting has been deprecated
                  and has no effect on the type of service field in the IP
                  Header).
  -r count       Record route for count hops (IPv4-only).
  -s count       Timestamp for count hops (IPv4-only).
  -j host-list   Loose source route along host-list (IPv4-only).
  -k host-list   Strict source route along host-list (IPv4-only).
  -w timeout     Timeout in milliseconds to wait for each reply.
  -R            Use routing header to test reverse route also (IPv6-only).
                  Per RFC 5095 the use of this routing header has been
                  deprecated. Some systems may drop echo requests if
                  this header is used.
  -S srcaddr     Source address to use.
  -c compartment Routing compartment identifier.
  -p            Ping a Hyper-V Network Virtualization provider address.
  -4            Force using IPv4.
  -6            Force using IPv6.
```

```
C:\Users\Vishal>ping -n 10 -l 32 www.google.com
```

```
Pinging www.google.com [142.250.67.228] with 32 bytes of data:
```

```
Reply from 142.250.67.228: bytes=32 time=233ms TTL=118
```

```
Reply from 142.250.67.228: bytes=32 time=4ms TTL=118
```

```
Reply from 142.250.67.228: bytes=32 time=6ms TTL=118
```

```
Reply from 142.250.67.228: bytes=32 time=5ms TTL=118
```

```
Reply from 142.250.67.228: bytes=32 time=9ms TTL=118
```

```
Reply from 142.250.67.228: bytes=32 time=6ms TTL=118
```

```
Reply from 142.250.67.228: bytes=32 time=4ms TTL=118
```

```
Reply from 142.250.67.228: bytes=32 time=5ms TTL=118
```

```
Reply from 142.250.67.228: bytes=32 time=17ms TTL=118
```

```
Reply from 142.250.67.228: bytes=32 time=4ms TTL=118
```

```
Ping statistics for 142.250.67.228:
```

```
    Packets: Sent = 10, Received = 10, Lost = 0 (0% loss),
```

```
Approximate round trip times in milli-seconds:
```

```
    Minimum = 4ms, Maximum = 233ms, Average = 29ms
```

```
C:\Users\Vishal>ping -l 10 google.com
```

```
Pinging google.com [142.250.67.174] with 10 bytes of data:
```

```
Reply from 142.250.67.174: bytes=10 time=198ms TTL=117
```

```
Reply from 142.250.67.174: bytes=10 time=12ms TTL=117
```

```
Reply from 142.250.67.174: bytes=10 time=10ms TTL=117
```

```
Reply from 142.250.67.174: bytes=10 time=2646ms TTL=117
```

```
Ping statistics for 142.250.67.174:
```

```
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
```

```
Approximate round trip times in milli-seconds:
```

```
    Minimum = 10ms, Maximum = 2646ms, Average = 716ms
```

```
C:\Users\Vishal>ping www.google.com
```

```
Pinging www.google.com [142.250.67.228] with 32 bytes of data:
```

```
Reply from 142.250.67.228: bytes=32 time=7ms TTL=118
```

```
Reply from 142.250.67.228: bytes=32 time=5ms TTL=118
```

```
Reply from 142.250.67.228: bytes=32 time=4ms TTL=118
```

```
Reply from 142.250.67.228: bytes=32 time=7ms TTL=118
```

```
Ping statistics for 142.250.67.228:
```

```
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
```

```
Approximate round trip times in milli-seconds:
```

```
    Minimum = 4ms, Maximum = 7ms, Average = 5ms
```


Exercise 1: Experiment with ping to find the round trip times to a variety of destinations. Write up any interesting observations, including in particular how the round trip time compares to the physical distance. Here are few places from who to get replies: www.uw.edu, www.cornell.edu, berkeley.edu, www.uchicago.edu, www.ox.ac.uk (England), www.u-tokyo.ac.jp (Japan).

```
C:\Users\Vishal>ping www.uw.edu

Pinging www.washington.edu [128.95.155.135] with 32 bytes of data:
Reply from 128.95.155.135: bytes=32 time=1415ms TTL=47
Reply from 128.95.155.135: bytes=32 time=289ms TTL=47
Reply from 128.95.155.135: bytes=32 time=502ms TTL=47
Reply from 128.95.155.135: bytes=32 time=511ms TTL=47

Ping statistics for 128.95.155.135:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 289ms, Maximum = 1415ms, Average = 679ms

C:\Users\Vishal>ping www.ox.ac.uk

Pinging www.ox.ac.uk [151.101.130.133] with 32 bytes of data:
Reply from 151.101.130.133: bytes=32 time=193ms TTL=59
Reply from 151.101.130.133: bytes=32 time=8ms TTL=59
Reply from 151.101.130.133: bytes=32 time=9ms TTL=59
Reply from 151.101.130.133: bytes=32 time=11ms TTL=59

Ping statistics for 151.101.130.133:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 8ms, Maximum = 193ms, Average = 55ms

C:\Users\Vishal>www.cornell.edu
'www.cornell.edu' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\Vishal>ping www.cornell.edu

Pinging ucomm-gw1.cornell.media3.us [20.42.25.107] with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 20.42.25.107:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```



```
C:\Users\Vishal>ping berkeley.edu

Pinging berkeley.edu [35.163.72.93] with 32 bytes of data:
Reply from 35.163.72.93: bytes=32 time=502ms TTL=37
Reply from 35.163.72.93: bytes=32 time=302ms TTL=37
Reply from 35.163.72.93: bytes=32 time=322ms TTL=37
Reply from 35.163.72.93: bytes=32 time=529ms TTL=37

Ping statistics for 35.163.72.93:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 302ms, Maximum = 529ms, Average = 413ms

C:\Users\Vishal>ping www.uchicago.edu

Pinging wsee2.elb.uchicago.edu [54.89.29.50] with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 54.89.29.50:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

Round-trip time (RTT) is the duration, measured in milliseconds, from when a browser sends a request to when it receives a response from a server. It's a key performance metric for web applications and one of the main factors, along with Time to First Byte (TTFB), when measuring page load time and network latency.

RTT is typically measured using a ping a command-line tool that bounces a request off a server and calculates the time taken to reach a user device. Actual RTT may be higher than that measured by the ping due to server throttling and network congestion.

From the above ping command on various site, it is clear that the sites in USA take more Round Trip Time than the ones in UK. So, it is clear that RTT increases with distance.

Nslookup

The command nslookup <host> will do a DNS query to find and report the IP address (or addresses) for a domain name or the domain name corresponding to an IP address. To do this, it contacts a "DNS server." Default DNS servers are part of a computer's network configuration. (For a static IP address in Linux, they are configured in the file /etc/network/interfaces that you encountered in the last lab.) You can specify a different DNS server to be used by nslookup by adding the server name or IP address to the command: nslookup <host> <server>

```
C:\Users\Vishal>nslookup www.google.com
Server:   UnKnown
Address:  192.168.0.1

Non-authoritative answer:
Name:     www.google.com
Addresses: 2404:6800:4009:814::2004
          142.250.67.228
```

Ifconfig

You used ifconfig in the previous lab. When used with no parameters, ifconfig reports some information about the computer's network interfaces. This usually includes lo which stands for localhost; it can be used for communication between programs running on the same computer. Linux often has an interface named eth0, which is the first ethernet card. The information is different on Mac OS and Linux, but includes the IP or "inet" address and ethernet or "hardware" address for an ethernet card. On Linux, you get the number of packets received (RX) and sent (TX), as well as the number of bytes transmitted and received. (A better place to monitor network bytes on our Linux computers is in the GUI program System Monitor, if it is installed!!!.)

```
C:\Users\Vishal>ipconfig

Windows IP Configuration

Ethernet adapter Ethernet:

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

Ethernet adapter VirtualBox Host-Only Network:

    Connection-specific DNS Suffix  . :
    Link-local IPv6 Address . . . . . : fe80::b5a3:faa2:bd70:c712%5
    IPv4 Address. . . . . : 192.168.56.1
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . :

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  . :

Wireless LAN adapter Wi-Fi:

    Connection-specific DNS Suffix  . :
    Link-local IPv6 Address . . . . . : fe80::95b2:e12d:c238:3861%11
    IPv4 Address. . . . . : 192.168.0.103
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 192.168.0.1

Ethernet adapter Bluetooth Network Connection:

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

Netstat

The netstat command gives information about network connections. I often use netstat -t -n which lists currently open TCP connections (that's the "-t" option) by IP address rather than domain name (that's the "-n" option). Add the option "-l" (lower case ell) to list listening sockets, that is sockets that have been opened by server programs to wait for connection requests from clients: netstat -t -n -l. (On Mac, use netstat -p tcp to list tcp connections, and add "-a" to include listening sockets in the list.)

```
C:\Users\Vishal>netstat -t tcp
```

Displays protocol statistics and current TCP/IP network connections.

```
NETSTAT [-a] [-b] [-e] [-f] [-n] [-o] [-p proto] [-r] [-s] [-x] [-t] [interval]
```

- a Displays all connections and listening ports.
- b Displays the executable involved in creating each connection or listening port. In some cases well-known executables host multiple independent components, and in these cases the sequence of components involved in creating the connection or listening port is displayed. In this case the executable name is in [] at the bottom, on top is the component it called, and so forth until TCP/IP was reached. Note that this option can be time-consuming and will fail unless you have sufficient permissions.
- e Displays Ethernet statistics. This may be combined with the -s option.
- f Displays Fully Qualified Domain Names (FQDN) for foreign addresses.
- n Displays addresses and port numbers in numerical form.
- o Displays the owning process ID associated with each connection.
- p proto Shows connections for the protocol specified by proto; proto may be any of: TCP, UDP, TCPv6, or UDPv6. If used with the -s option to display per-protocol statistics, proto may be any of: IP, IPv6, ICMP, ICMPv6, TCP, TCPv6, UDP, or UDPv6.
- q Displays all connections, listening ports, and bound nonlistening TCP ports. Bound nonlistening ports may or may not be associated with an active connection.
- r Displays the routing table.
- s Displays per-protocol statistics. By default, statistics are shown for IP, IPv6, ICMP, ICMPv6, TCP, TCPv6, UDP, and UDPv6; the -p option may be used to specify a subset of the default.
- t Displays the current connection offload state.
- x Displays NetworkDirect connections, listeners, and shared endpoints.
- y Displays the TCP connection template for all connections. Cannot be combined with the other options.
- interval Redisplays selected statistics, pausing interval seconds between each display. Press CTRL+C to stop redisplaying statistics. If omitted, netstat will print the current configuration information once.

```
C:\Users\Vishal>netstat -p tcp
```

Active Connections

Proto	Local Address	Foreign Address	State
TCP	127.0.0.1:49670	LAPTOP-9VH1A37S:49671	ESTABLISHED
TCP	127.0.0.1:49671	LAPTOP-9VH1A37S:49670	ESTABLISHED
TCP	127.0.0.1:49674	LAPTOP-9VH1A37S:49675	ESTABLISHED
TCP	127.0.0.1:49675	LAPTOP-9VH1A37S:49674	ESTABLISHED
TCP	127.0.0.1:49676	LAPTOP-9VH1A37S:61900	ESTABLISHED
TCP	127.0.0.1:49677	LAPTOP-9VH1A37S:49678	ESTABLISHED
TCP	127.0.0.1:49678	LAPTOP-9VH1A37S:49677	ESTABLISHED
TCP	127.0.0.1:49683	LAPTOP-9VH1A37S:49794	ESTABLISHED
TCP	127.0.0.1:49683	LAPTOP-9VH1A37S:49818	ESTABLISHED
TCP	127.0.0.1:49684	LAPTOP-9VH1A37S:49685	ESTABLISHED
TCP	127.0.0.1:49685	LAPTOP-9VH1A37S:49684	ESTABLISHED
TCP	127.0.0.1:49686	LAPTOP-9VH1A37S:49687	ESTABLISHED
TCP	127.0.0.1:49687	LAPTOP-9VH1A37S:49686	ESTABLISHED
TCP	127.0.0.1:49688	LAPTOP-9VH1A37S:61900	ESTABLISHED
TCP	127.0.0.1:49689	LAPTOP-9VH1A37S:49690	ESTABLISHED
TCP	127.0.0.1:49690	LAPTOP-9VH1A37S:49689	ESTABLISHED
TCP	127.0.0.1:49705	LAPTOP-9VH1A37S:49912	ESTABLISHED
TCP	127.0.0.1:49715	LAPTOP-9VH1A37S:49716	ESTABLISHED
TCP	127.0.0.1:49716	LAPTOP-9VH1A37S:49715	ESTABLISHED
TCP	127.0.0.1:49717	LAPTOP-9VH1A37S:61900	ESTABLISHED
TCP	127.0.0.1:49718	LAPTOP-9VH1A37S:49719	ESTABLISHED
TCP	127.0.0.1:49719	LAPTOP-9VH1A37S:49718	ESTABLISHED
TCP	127.0.0.1:49720	LAPTOP-9VH1A37S:49727	ESTABLISHED
TCP	127.0.0.1:49720	LAPTOP-9VH1A37S:49730	ESTABLISHED
TCP	127.0.0.1:49720	LAPTOP-9VH1A37S:49733	ESTABLISHED
TCP	127.0.0.1:49720	LAPTOP-9VH1A37S:49734	ESTABLISHED
TCP	127.0.0.1:49720	LAPTOP-9VH1A37S:49737	ESTABLISHED
TCP	127.0.0.1:49720	LAPTOP-9VH1A37S:49738	ESTABLISHED
TCP	127.0.0.1:49720	LAPTOP-9VH1A37S:49749	ESTABLISHED
TCP	127.0.0.1:49720	LAPTOP-9VH1A37S:49770	ESTABLISHED
TCP	127.0.0.1:49727	LAPTOP-9VH1A37S:49720	ESTABLISHED
TCP	127.0.0.1:49730	LAPTOP-9VH1A37S:49720	ESTABLISHED
TCP	127.0.0.1:49733	LAPTOP-9VH1A37S:49720	ESTABLISHED
TCP	127.0.0.1:49734	LAPTOP-9VH1A37S:49720	ESTABLISHED
TCP	127.0.0.1:49737	LAPTOP-9VH1A37S:49720	ESTABLISHED
TCP	127.0.0.1:49738	LAPTOP-9VH1A37S:49720	ESTABLISHED
TCP	127.0.0.1:49742	LAPTOP-9VH1A37S:49743	ESTABLISHED
TCP	127.0.0.1:49743	LAPTOP-9VH1A37S:49742	ESTABLISHED
TCP	127.0.0.1:49744	LAPTOP-9VH1A37S:61900	ESTABLISHED
TCP	127.0.0.1:49745	LAPTOP-9VH1A37S:49746	ESTABLISHED
TCP	127.0.0.1:49746	LAPTOP-9VH1A37S:49745	ESTABLISHED

```
C:\Users\Vishal>netstat -a
```

Active Connections

Proto	Local Address	Foreign Address	State
TCP	0.0.0.0:135	LAPTOP-9VH1A37S:0	LISTENING
TCP	0.0.0.0:445	LAPTOP-9VH1A37S:0	LISTENING
TCP	0.0.0.0:2343	LAPTOP-9VH1A37S:0	LISTENING
TCP	0.0.0.0:3306	LAPTOP-9VH1A37S:0	LISTENING
TCP	0.0.0.0:3580	LAPTOP-9VH1A37S:0	LISTENING
TCP	0.0.0.0:3582	LAPTOP-9VH1A37S:0	LISTENING
TCP	0.0.0.0:5040	LAPTOP-9VH1A37S:0	LISTENING
TCP	0.0.0.0:6646	LAPTOP-9VH1A37S:0	LISTENING
TCP	0.0.0.0:8080	LAPTOP-9VH1A37S:0	LISTENING
TCP	0.0.0.0:49664	LAPTOP-9VH1A37S:0	LISTENING
TCP	0.0.0.0:49665	LAPTOP-9VH1A37S:0	LISTENING
TCP	0.0.0.0:49666	LAPTOP-9VH1A37S:0	LISTENING
TCP	0.0.0.0:49667	LAPTOP-9VH1A37S:0	LISTENING
TCP	0.0.0.0:49668	LAPTOP-9VH1A37S:0	LISTENING
TCP	0.0.0.0:49723	LAPTOP-9VH1A37S:0	LISTENING
TCP	0.0.0.0:49740	LAPTOP-9VH1A37S:0	LISTENING
TCP	0.0.0.0:59110	LAPTOP-9VH1A37S:0	LISTENING
TCP	0.0.0.0:59111	LAPTOP-9VH1A37S:0	LISTENING
TCP	0.0.0.0:59112	LAPTOP-9VH1A37S:0	LISTENING
TCP	127.0.0.1:15292	LAPTOP-9VH1A37S:0	LISTENING
TCP	127.0.0.1:15393	LAPTOP-9VH1A37S:0	LISTENING
TCP	127.0.0.1:16494	LAPTOP-9VH1A37S:0	LISTENING
TCP	127.0.0.1:27017	LAPTOP-9VH1A37S:0	LISTENING
TCP	127.0.0.1:45623	LAPTOP-9VH1A37S:0	LISTENING
TCP	127.0.0.1:49670	LAPTOP-9VH1A37S:49671	ESTABLISHED
TCP	127.0.0.1:49671	LAPTOP-9VH1A37S:49670	ESTABLISHED
TCP	127.0.0.1:49674	LAPTOP-9VH1A37S:49675	ESTABLISHED
TCP	127.0.0.1:49675	LAPTOP-9VH1A37S:49674	ESTABLISHED
TCP	127.0.0.1:49676	LAPTOP-9VH1A37S:61900	ESTABLISHED
TCP	127.0.0.1:49677	LAPTOP-9VH1A37S:49678	ESTABLISHED
TCP	127.0.0.1:49678	LAPTOP-9VH1A37S:49677	ESTABLISHED
TCP	127.0.0.1:49683	LAPTOP-9VH1A37S:0	LISTENING
TCP	127.0.0.1:49683	LAPTOP-9VH1A37S:49794	ESTABLISHED
TCP	127.0.0.1:49683	LAPTOP-9VH1A37S:49818	ESTABLISHED
TCP	127.0.0.1:49684	LAPTOP-9VH1A37S:49685	ESTABLISHED
TCP	127.0.0.1:49685	LAPTOP-9VH1A37S:49684	ESTABLISHED
TCP	127.0.0.1:49686	LAPTOP-9VH1A37S:49687	ESTABLISHED
TCP	127.0.0.1:49687	LAPTOP-9VH1A37S:49686	ESTABLISHED
TCP	127.0.0.1:49688	LAPTOP-9VH1A37S:61900	ESTABLISHED
TCP	127.0.0.1:49689	LAPTOP-9VH1A37S:49690	ESTABLISHED

telnet

Telnet is an old program for remote login. It's not used so much for that any more, since it has no security features. But basically, all it does is open a connection to a server and allow server and client to send lines of plain text to each other. It can be used to check that it's possible to connect to a server and, if the server communicates in plain text, even to interact with the server by hand. Since the Web uses a plain text protocol, you can use telnet to connect to a web client and play the part of the web browser. I will suggest that you to do this with your own web server when you write it, but you might want to try it now. When you use telnet in this way, you need to specify both the host and the port number to which you want to connect: telnet <host> <port>. For example, to connect to the web server on www.spit.ac.in: telnet spit.ac.in 80.

Tracert

Traceroute is discussed in man utility. The command traceroute <host> will show routers encountered by packets on their way from your computer to a specified <host>. For each $n = 1, 2, 3, \dots$, traceroute sends a packet with "time-to-live" (ttl) equal to n . Every time a router forwards a packet, it decreases the ttl of the packet by one. If the ttl drops to zero, the router discards the packet and sends an error message back to the sender of the packet. (Again, as with ping, the packets might be blocked or might not even be sent, so that the error messages will never be received.) The sender gets the identity of the router from the source of the error message. Traceroute will send packets until n reaches some set upper bound or until a packet actually gets through to the destination. It actually does this three times for each n . In this way, it identifies routers that are one step, two steps, three steps, ... away from the source computer. A packet for which no response is received is indicated in the output as a *.

Traceroute is installed on the computers. If was not installed in your virtual server last week, but you can install it with the command `sudo apt-get install traceroute`

The path taken through a network, can be measured using traceroute. The syntax for the command in Linux is:

```
traceroute <hostname>
```

The syntax in Windows is:

```
tracert <hostname>
```

You can specify either a hostname (e.g., cs.iitb.ac.in) or an IP address (e.g., 128.105.2.6)

EXPERIMENTS WITH TRACEROUTE

From your machine traceroute to the following hosts:

1. ee.iitb.ac.in
2. mscs.mu.edu
3. www.cs.grinnell.edu
4. csail.mit.edu
5. cs.stanford.edu
6. cs.manchester.ac.uk

Store the output of each traceroute command in a separate file named `traceroute_HOSTNAME.log`, replacing `HOSTNAME` with the hostname for end-host you pinged

(e.g., `traceroute_ee.iitb.ac.in.log`).

mscs.mu.edu

```
C:\Users\Vishal>tracert mscs.mu.edu

Tracing route to mscs.mu.edu [134.48.4.5]
over a maximum of 30 hops:

  1  302 ms    1 ms      1 ms    192.168.0.1
  2   2 ms     2 ms      3 ms    45.112.56.246
  3   8 ms     *          7 ms    45.112.56.245
  4   4 ms     3 ms      2 ms    172.16.2.101
  5   5 ms     3 ms      4 ms    121.241.42.57.static-mumbai.vsnl.net.in [121.241.43.57]
  6   5 ms     3 ms      5 ms    172.23.78.237
  7   4 ms     7 ms      4 ms    ix-ae-0-100.tcore1.mlv-mumbai.as6453.net [180.87.38.5]
  8   *        *          *        Request timed out.
  9  120 ms    *          114 ms   if-ae-8-1600.tcore1.pye-paris.as6453.net [80.231.217.6]
 10  111 ms   115 ms    123 ms   if-ae-11-2.tcore1.pvu-paris.as6453.net [80.231.153.49]
 11   *        *          *        Request timed out.
 12   *        *          *        Request timed out.
 13  282 ms   268 ms    269 ms   MARQUETTE-U.ear3.Chicago2.Level3.net [4.16.38.70]
 14  275 ms   261 ms    261 ms   134.48.10.27
 15   *        *          *        Request timed out.
 16   *        *          *        Request timed out.
 17   *        *          *        Request timed out.
 18   *        *          *        Request timed out.
 19   *        *          *        Request timed out.
 20   *        *          *        Request timed out.
 21   *        *          *        Request timed out.
 22   *        *          *        Request timed out.
 23   *        *          *        Request timed out.
 24   *        *          *        Request timed out.
 25   *        *          *        Request timed out.
 26   *        *          *        Request timed out.
 27   *        *          *        Request timed out.
 28   *        *          *        Request timed out.
 29   *        *          *        Request timed out.
 30   *        *          *        Request timed out.

Trace complete.
```


www.cs.grinnell.edu

```
C:\Users\Vishal>tracert www.cs.grinnell.edu

Tracing route to www.cs.grinnell.edu [132.161.132.159]
over a maximum of 30 hops:

  1  1631 ms    1 ms    1 ms  192.168.0.1
  2    4 ms    2 ms    1 ms  45.112.56.246
  3    4 ms    4 ms    6 ms  45.112.56.245
  4   18 ms    6 ms    4 ms  172.16.2.101
  5    5 ms    5 ms    4 ms  121.241.42.57.static-mumbai.vsnl.net.in [121.241.43.57]
  6    *        *        *    Request timed out.
  7   25 ms   26 ms   44 ms  172.31.244.45
  8   97 ms   44 ms   59 ms  ix-ae-4-2.tcore2.cxr-chennai.as6453.net [180.87.37.1]
  9  297 ms  243 ms  265 ms  if-ae-9-2.tcore2.mlv-mumbai.as6453.net [180.87.37.10]
 10  245 ms  247 ms  242 ms  if-ae-2-2.tcore1.mlv-mumbai.as6453.net [180.87.38.1]
 11    *    240 ms    *    if-ae-5-6.tcore1.wyn-marseille.as6453.net [180.87.38.126]
 12  259 ms  256 ms  247 ms  if-ae-2-2.tcore2.wyn-marseille.as6453.net [80.231.217.2]
 13    *    250 ms    *    if-ae-9-2.tcore2.l78-london.as6453.net [80.231.200.14]
 14  262 ms  246 ms  260 ms  if-ae-15-2.tcore2.ldn-london.as6453.net [80.231.131.118]
 15  257 ms  258 ms  245 ms  if-ae-32-2.tcore2.nto-newyork.as6453.net [63.243.216.22]
 16  253 ms  299 ms  245 ms  if-ae-26-2.tcore1.ct8-chicago.as6453.net [216.6.81.29]
 17    *    251 ms    *    63.243.129.121
 18  250 ms  255 ms  256 ms  gi0-0-0-3.agr02.mtld01-fl.us.windstream.net [169.130.82.82]
 19  266 ms  251 ms  251 ms  et3-1-0-0.agr03.desm01-ia.us.windstream.net [40.128.250.43]
 20  267 ms  255 ms  264 ms  ae4-0.pe04.grnl01-ia.us.windstream.net [40.128.248.35]
 21  271 ms  251 ms  252 ms  ae7-0.pe05.grnl01-ia.us.windstream.net [40.138.127.29]
 22    *        *        *    Request timed out.
 23    *        *        *    Request timed out.
 24    *        *        *    Request timed out.
 25    *        *        *    Request timed out.
 26    *        *        *    Request timed out.
 27    *        *        *    Request timed out.
 28    *        *        *    Request timed out.
 29    *        *        *    Request timed out.
 30    *        *        *    Request timed out.

Trace complete.
```

csail.mit.edu

```
C:\Users\Vishal>tracert csail.mit.edu

Tracing route to csail.mit.edu [128.30.2.109]
over a maximum of 30 hops:

  1   304 ms    1 ms    1 ms  192.168.0.1
  2    3 ms    2 ms    2 ms  45.112.56.246
  3   11 ms   10 ms    *    45.112.56.245
  4    3 ms    3 ms    4 ms  172.16.2.101
  5    3 ms    6 ms    3 ms  182.73.109.41
  6  240 ms  262 ms  245 ms  182.79.245.69
  7  224 ms  233 ms  232 ms  xe-5-1-0.edge1.LosAngeles6.Level3.net [4.26.0.89]
  8    *        *        *    Request timed out.
  9  284 ms  293 ms  284 ms  MASSACHUSET.bear1.Boston1.Level3.net [4.53.48.98]
 10  307 ms  307 ms  304 ms  dmz-rtr-1-external-rtr-1.mit.edu [18.0.161.17]
 11  297 ms  313 ms  298 ms  dmz-rtr-2-dmz-rtr-1-1.mit.edu [18.0.161.6]
 12  296 ms  343 ms  292 ms  mitnet.core-1-ext.csail.mit.edu [18.4.7.65]
 13  334 ms  325 ms    *    core-1-ext.bdr.csail.mit.edu [128.30.13.26]
 14  320 ms  320 ms  335 ms  bdr.core-1.csail.mit.edu [128.30.0.246]
 15  313 ms  327 ms  309 ms  inquir-3ld.csail.mit.edu [128.30.2.109]

Trace complete.
```

cs.stanford.edu

```
C:\Users\Vishal>tracert cs.stanford.edu

Tracing route to cs.stanford.edu [171.64.64.64]
over a maximum of 30 hops:

  1    1 ms    1 ms    1 ms  192.168.0.1
  2    5 ms    4 ms    3 ms  45.112.56.246
  3    7 ms    *        4 ms  45.112.56.245
  4    5 ms    3 ms    2 ms  172.16.2.101
  5    8 ms    19 ms   6 ms  182.73.109.41
  6   198 ms   199 ms  203 ms aes-static-150.36.144.59.airtel.in [59.144.36.150]
  7    *        *        *    Request timed out.
  8   271 ms   261 ms  294 ms 100ge8-1.core1.sjc2.he.net [184.105.81.218]
  9   269 ms   267 ms  268 ms 10ge4-5.core1.pao1.he.net [72.52.92.69]
 10   276 ms   283 ms  278 ms stanford-university.100gigabitethernet5-1.core1.pao1.he.net [184.105.177.238]
 11   267 ms   274 ms  273 ms csee-west-rtr-v13.SUNet [171.66.255.140]
 12   282 ms   261 ms  260 ms CS.stanford.edu [171.64.64.64]

Trace complete.
```

cs.manchester.ac.uk

```
C:\Users\Vishal>tracert cs.manchester.ac.uk

Tracing route to cs.manchester.ac.uk [130.88.101.49]
over a maximum of 30 hops:

  1  2427 ms    1 ms    1 ms  192.168.0.1
  2    3 ms    1 ms    1 ms  45.112.56.246
  3    *        *        *    Request timed out.
  4    6 ms    17 ms   6 ms  172.16.2.101
  5   15 ms    9 ms    5 ms  182.73.109.41
  6   152 ms   143 ms  136 ms 182.79.154.0
  7   176 ms    *       314 ms ldn-b4-link.telia.net [62.115.162.232]
  8   137 ms   155 ms  148 ms jisc-ic-345131-ldn-b4.c.telia.net [62.115.175.131]
  9   130 ms   129 ms  126 ms ae24.londhx-sbr1.ja.net [146.97.35.197]
 10   133 ms   133 ms  164 ms ae29.londpg-sbr2.ja.net [146.97.33.2]
 11   151 ms   144 ms  143 ms ae31.erdiss-sbr2.ja.net [146.97.33.22]
 12   140 ms   131 ms  173 ms ae29.manckh-sbr2.ja.net [146.97.33.42]
 13   159 ms   163 ms  154 ms ae23.mancrh-rbr1.ja.net [146.97.38.42]
 14    *        *        *    Request timed out.
 15   137 ms   153 ms  139 ms 130.88.249.194
 16    *        *        *    Request timed out.
 17   158 ms   168 ms  143 ms gw-jh.its.manchester.ac.uk [130.88.250.32]
 18   155 ms   147 ms  141 ms eps.its.man.ac.uk [130.88.101.49]

Trace complete.
```

```
C:\Users\Vishal>tracert

Usage: tracert [-d] [-h maximum_hops] [-j host-list] [-w timeout]
              [-R] [-S srcaddr] [-4] [-6] target_name

Options:
  -d                Do not resolve addresses to hostnames.
  -h maximum_hops   Maximum number of hops to search for target.
  -j host-list       Loose source route along host-list (IPv4-only).
  -w timeout         Wait timeout milliseconds for each reply.
  -R                Trace round-trip path (IPv6-only).
  -S srcaddr         Source address to use (IPv6-only).
  -4                Force using IPv4.
  -6                Force using IPv6.
```

```
C:\Users\Vishal>tracert 192.168.10.10
```

```
Tracing route to 192.168.10.10 over a maximum of 30 hops
```

1	11 ms	4 ms	123 ms	192.168.0.1
2	5 ms	114 ms	3 ms	45.112.56.246
3	134 ms	*	*	45.112.56.245
4	12 ms	77 ms	4 ms	172.16.2.101
5	*	*	*	Request timed out.
6	6 ms	6 ms	38 ms	172.16.2.253
7	*	*	*	Request timed out.
8	5 ms	5 ms	4 ms	172.16.2.253
9	*	*	*	Request timed out.
10	4 ms	3 ms	4 ms	172.16.2.253
11	*	*	*	Request timed out.
12	16 ms	6 ms	7 ms	172.16.2.253
13	*	*	*	Request timed out.
14	23 ms	19 ms	4 ms	172.16.2.253
15	*	*	*	Request timed out.
16	49 ms	1590 ms	5 ms	172.16.2.253
17	*	*	*	Request timed out.
18	663 ms	4 ms	5 ms	172.16.2.253
19	*	*	*	Request timed out.
20	6 ms	4 ms	6 ms	172.16.2.253
21	*	*	*	Request timed out.
22	1209 ms	6 ms	4 ms	172.16.2.253
23	*	*	*	Request timed out.
24	7 ms	6 ms	6 ms	172.16.2.253
25	*	*	*	Request timed out.
26	9 ms	5 ms	108 ms	172.16.2.253
27	*	*	*	Request timed out.
28	7 ms	6 ms	121 ms	172.16.2.253
29	*	*	*	Request timed out.
30	9 ms	7 ms	7 ms	172.16.2.253

```
Trace complete.
```

```
C:\Users\Vishal>tracert mit.edu.in
```

```
Tracing route to mit.edu.in [198.71.205.226]  
over a maximum of 30 hops:
```

1	285 ms	4 ms	2 ms	192.168.0.1
2	4 ms	89 ms	212 ms	45.112.56.246
3	*	*	6 ms	45.112.56.245
4	7 ms	3 ms	4 ms	172.16.2.101
5	10 ms	8 ms	369 ms	182.73.109.41
6	306 ms	180 ms	332 ms	182.79.146.216
7	*	529 ms	*	ldn-b4-link.telial.net [62.115.162.232]
8	296 ms	360 ms	345 ms	ldn-bb3-link.telial.net [62.115.122.188]
9	*	*	*	Request timed out.
10	553 ms	2666 ms	308 ms	ash-bb2-link.telial.net [62.115.136.201]
11	353 ms	297 ms	309 ms	las-b24-link.telial.net [62.115.121.220]
12	365 ms	299 ms	298 ms	ae9.ibrsa0107-01.lax1.bb.godaddy.com [62.115.171.243]
13	352 ms	401 ms	400 ms	148.72.34.34
14	328 ms	311 ms	598 ms	be39.trmc0215-01.ars.mgmt.phx3.gdg [184.168.0.73]
15	*	*	*	Request timed out.
16	*	*	*	Request timed out.
17	*	*	*	Request timed out.
18	*	*	*	Request timed out.
19	*	*	*	Request timed out.
20	*	*	*	Request timed out.
21	*	*	*	Request timed out.
22	*	*	*	Request timed out.
23	*	*	*	Request timed out.
24	*	*	*	Request timed out.
25	*	*	*	Request timed out.
26	*	*	*	Request timed out.
27	*	*	*	Request timed out.
28	*	*	*	Request timed out.
29	*	*	*	Request timed out.
30	*	*	*	Request timed out.

```
Trace complete.
```

Exercise 2: Use traceroute to trace the route from your computer to math.hws.edu and to www.hws.edu. Explain the difference in the results.

```
C:\Users\Vishal>tracert math.hws.edu

Tracing route to math.hws.edu [64.89.144.237]
over a maximum of 30 hops:

  1  1183 ms    5 ms    3 ms  192.168.0.1
  2    4 ms    3 ms   312 ms  45.112.56.246
  3   25 ms   32 ms    *    45.112.56.245
  4    7 ms    3 ms    5 ms  172.16.2.101
  5   73 ms  102 ms   87 ms  182.73.109.41
  6  333 ms  306 ms  498 ms  182.79.243.29
  7  324 ms  297 ms  251 ms  xe-9-1-0.edge1.LosAngeles6.Level3.net [4.26.0.61]
  8    *    *    *    Request timed out.
  9    *    *    *    Request timed out.
 10  307 ms  310 ms  305 ms  roc1-ar5-xe-0-0-0-0.us.twtelecom.net [35.248.1.158]
 11  349 ms  404 ms  506 ms  66-195-65-170.static.ct1.one [66.195.65.170]
 12  497 ms  318 ms  496 ms  64.89.144.100
 13    *    *    *    Request timed out.
 14    *    *    *    Request timed out.
 15    *    *    *    Request timed out.
 16    *    *    *    Request timed out.
 17    *    *    *    Request timed out.
 18    *    *    *    Request timed out.
 19    *    *    *    Request timed out.
 20    *    *    *    Request timed out.
 21    *    *    *    Request timed out.
 22    *    *    *    Request timed out.
 23    *    *    *    Request timed out.
 24    *    *    *    Request timed out.
 25    *    *    *    Request timed out.
 26    *    *    *    Request timed out.
 27    *    *    *    Request timed out.
 28    *    *    *    Request timed out.
 29    *    *    *    Request timed out.
 30    *    *    *    Request timed out.

Trace complete.
```

```
C:\Users\Vishal>tracert www.hws.edu

Tracing route to www.hws.edu [64.89.145.159]
over a maximum of 30 hops:

  1   382 ms    3 ms    3 ms  192.168.0.1
  2    5 ms    4 ms    3 ms  45.112.56.246
  3    6 ms    4 ms    *    45.112.56.245
  4   12 ms   10 ms    4 ms  172.16.2.101
  5   13 ms  154 ms    7 ms  182.73.109.41
  6  314 ms  303 ms  307 ms  182.79.243.29
  7  335 ms  398 ms  712 ms  ae58.edge1.LosAngeles6.Level3.net [4.26.0.17]
  8    *    *    *    Request timed out.
  9    *    *    *    Request timed out.
 10  402 ms  514 ms  417 ms  roc1-ar5-xe-0-0-0-0.us.twtelecom.net [35.248.1.158]
 11  384 ms  609 ms  534 ms  66-195-65-170.static.ct1.one [66.195.65.170]
 12  514 ms  604 ms  323 ms  64.89.144.100
 13    *    *    *    Request timed out.
 14    *    *    *    Request timed out.
 15    *    *    *    Request timed out.
 16    *    *    *    Request timed out.
 17    *    *    *    Request timed out.
 18    *    *    *    Request timed out.
 19    *    *    *    Request timed out.
 20    *    *    *    Request timed out.
 21    *    *    *    Request timed out.
 22    *    *    *    Request timed out.
 23    *    *    *    Request timed out.
 24    *    *    *    Request timed out.
 25    *    *    *    Request timed out.
 26    *    *    *    Request timed out.
 27    *    *    *    Request timed out.
 28    *    *    *    Request timed out.
 29    *    *    *    Request timed out.
 30    *    *    *    Request timed out.

Trace complete.
```

The only difference is that from 1st hop time taken by the packet is different but the end result is the same.

Exercise 3: Two packets sent from the same source to the same destination do not necessarily follow the same path through the net. Experiment with some sources that are fairly far away. Can you find cases where packets sent to the same destination follow different paths? How likely does it seem to be? What about when the packets are sent at very different times? Save some of the outputs from traceroute. (You can copy them from the Terminal window by highlighting and right-clicking, then paste into a text editor.) Come back sometime next week, try the same destinations again, and compare the results with the results from today. Report your observations.

```
C:\Users\Vishal>tracert www.google.com

Tracing route to www.google.com [142.250.67.228]
over a maximum of 30 hops:

  1    4 ms    3 ms    4 ms  192.168.0.1
  2    5 ms    6 ms    3 ms  45.112.56.246
  3    *      6 ms    5 ms  45.112.56.245
  4    8 ms    4 ms   140 ms 172.16.2.202
  5    8 ms    5 ms   236 ms 175.100.188.26
  6    5 ms    4 ms    4 ms 108.170.248.193
  7    8 ms   11 ms    6 ms 142.250.228.47
  8    7 ms    7 ms   118 ms bom07s24-in-f4.1e100.net [142.250.67.228]

Trace complete.

C:\Users\Vishal>tracert www.google.com

Tracing route to www.google.com [142.250.67.228]
over a maximum of 30 hops:

  1    2 ms    3 ms    2 ms  192.168.0.1
  2    5 ms    3 ms    3 ms  45.112.56.246
  3    *      *      *      Request timed out.
  4    4 ms    8 ms    7 ms  172.16.2.202
  5    7 ms    4 ms    5 ms  175.100.188.26
  6  2182 ms   87 ms    5 ms  108.170.248.193
  7   11 ms  209 ms   132 ms 142.250.228.47
  8    6 ms    5 ms   200 ms bom07s24-in-f4.1e100.net [142.250.67.228]

Trace complete.
```

It is true that 2 packets sent from the same source to the same destination do not necessarily follow the same path through the net. As seen in the above image the first one takes 8 hops and the second one takes 8 hops to reach the destination but at 3rd hop there is request time out otherwise result will be same.

```
C:\Users\Vishal>tracert mit.edu.in
```

```
Tracing route to mit.edu.in [198.71.205.226]  
over a maximum of 30 hops:
```

1	2 ms	1 ms	1 ms	192.168.0.1
2	5 ms	3 ms	4 ms	45.112.56.246
3	*	*	2207 ms	45.112.56.245
4	93 ms	4 ms	4 ms	172.16.2.101
5	211 ms	95 ms	9 ms	182.73.109.41
6	226 ms	197 ms	207 ms	182.79.146.216
7	*	326 ms	313 ms	ldn-b4-link.telial.net [62.115.162.232]
8	419 ms	329 ms	384 ms	ldn-bb3-link.telial.net [62.115.122.188]
9	*	*	*	Request timed out.
10	574 ms	414 ms	301 ms	ash-bb2-link.telial.net [62.115.136.201]
11	298 ms	308 ms	315 ms	las-b24-link.telial.net [62.115.121.220]
12	510 ms	509 ms	313 ms	ae9.ibrsa0107-01.lax1.bb.godaddy.com [62.115.171.243]
13	325 ms	400 ms	402 ms	148.72.34.34
14	426 ms	536 ms	575 ms	be39.trmc0215-01.ars.mgmt.phx3.gdg [184.168.0.73]
15	*	*	*	Request timed out.
16	*	*	*	Request timed out.
17	*	*	*	Request timed out.
18	*	*	*	Request timed out.
19	*	*	*	Request timed out.
20	*	*	*	Request timed out.
21	*	*	*	Request timed out.
22	*	*	*	Request timed out.
23	*	*	*	Request timed out.
24	*	*	*	Request timed out.
25	*	*	*	Request timed out.
26	*	*	*	Request timed out.
27	*	*	*	Request timed out.
28	*	*	*	Request timed out.
29	*	*	*	Request timed out.
30	*	*	*	Request timed out.

```
Trace complete.
```

```
C:\Users\Vishal>tracert mit.edu.in
```

```
Tracing route to mit.edu.in [198.71.205.226]  
over a maximum of 30 hops:
```

1	3 ms	2 ms	2 ms	192.168.0.1
2	4 ms	5 ms	3 ms	45.112.56.246
3	*	*	*	Request timed out.
4	245 ms	5 ms	5 ms	172.16.2.101
5	210 ms	7 ms	295 ms	182.73.109.41
6	213 ms	304 ms	398 ms	182.79.146.216
7	209 ms	202 ms	*	ldn-b4-link.telial.net [62.115.162.232]
8	341 ms	311 ms	300 ms	ldn-bb3-link.telial.net [62.115.122.188]
9	*	*	*	Request timed out.
10	583 ms	298 ms	301 ms	ash-bb2-link.telial.net [62.115.136.201]
11	918 ms	696 ms	314 ms	las-b24-link.telial.net [62.115.121.220]
12	1347 ms	499 ms	918 ms	ae9.ibrsa0107-01.lax1.bb.godaddy.com [62.115.171.243]
13	314 ms	405 ms	303 ms	148.72.34.34
14	1097 ms	622 ms	306 ms	be39.trmc0215-01.ars.mgmt.phx3.gdg [184.168.0.73]
15	608 ms	515 ms	409 ms	ip-97-74-255-129.ip.secureserver.net [97.74.255.129]
16	*	*	*	Request timed out.
17	*	*	*	Request timed out.
18	*	*	*	Request timed out.
19	*	*	*	Request timed out.
20	*	*	*	Request timed out.
21	*	*	*	Request timed out.
22	*	*	*	Request timed out.
23	*	*	*	Request timed out.
24	*	*	*	Request timed out.
25	*	*	*	Request timed out.
26	*	*	*	Request timed out.
27	*	*	*	Request timed out.
28	*	*	*	Request timed out.
29	*	*	*	Request timed out.
30	*	*	*	Request timed out.

```
Trace complete.
```


QUESTIONS ABOUT PATHS

Now look at the results you gathered and answer the following questions about the paths taken by your packets. Store your answers in a file named `traceroute.txt`.

1. Is any part of the path common for all hosts you tracerouted?

Yes, it is true that some part of path common for host that I traceroute. When packet sent from the same source to the same destination do not necessarily follow the same path through the net. Some of take more hope than other one. As I see in `mit.edu.in` first packet takes 14 hopes to reach where as other one take 15.

2. Is there a relationship between the number of nodes that show up in the traceroute and the location of the host? If so, what is this relationship?

If the distance between source and destination is more, then more hops will be required in order to reach the destination as more number of access points will be used for routing and the greater the number of access points involved, the greater are the chances of access points failing to respond and similarly for searching the alternative optimal path towards the destination.

3. Is there a relationship between the number of nodes that show up in the traceroute and latency of the host (from your ping results above)? Does the same relationship hold for all hosts?

If the latency of the host causes the traceroute request to get timed out even after the conventional three tries then it keeps on sending the data packets until the host responds or upto certain hops. The same relationship may not hold for all hosts.

Whois — The whois command can give detailed information about domain names and IP addresses. If it is not installed on the computers then install it with command `sudo apt-get install whois`. Whois can tell you what organization owns or is responsible for the name or address and where to contact them. It often includes a list of domain name servers for the organization.

When using whois to look up a domain name, use the simple two-part network name, not an individual computer name (for example, `whois spit.ac.in`).

Exercise 4: Use whois to investigate a well-known web site such as `google.com` or `amazon.com`, and write a couple of sentences about what you find out.

```
ubuntu@ubuntu:~$ whois spit.ac.in
Domain Name: spit.ac.in
Registry Domain ID: D2241401-IN
Registrar WHOIS Server:
Registrar URL: http://www.ernet.in
Updated Date: 2020-05-18T09:51:15Z
Creation Date: 2006-05-22T04:58:23Z
Registry Expiry Date: 2025-05-22T04:58:23Z
Registrar: ERNET India
Registrar IANA ID: 800068
Registrar Abuse Contact Email:
Registrar Abuse Contact Phone:
Domain Status: ok http://www.icann.org/epp#OK
Registry Registrant ID:
Registrant Name:
Registrant Organization: Bharatiya Vidya Bhavans Sardar Patel Institute of Technology Mumbai
Registrant Street:
Registrant Street:
Registrant Street:
Registrant City:
Registrant State/Province:
Registrant Postal Code:
Registrant Country: IN
Registrant Phone:
Registrant Phone Ext:
Registrant Fax:
```

```
Registrant Fax Ext:
Registrant Email: Please contact the Registrar listed above
Registry Admin ID:
Admin Name:
Admin Organization:
Admin Street:
Admin Street:
Admin Street:
Admin City:
Admin State/Province:
Admin Postal Code:
Admin Country:
Admin Phone:
Admin Phone Ext:
Admin Fax:
Admin Fax Ext:
Admin Email: Please contact the Registrar listed above
Registry Tech ID:
Tech Name:
Tech Organization:
Tech Street:
Tech Street:
Tech Street:
Tech City:
Tech State/Province:
Tech Postal Code:
Tech Country:
Tech Phone:
```

```
Tech Email: Please contact the Registrar listed above
Name Server: ns2.spit.ac.in
Name Server: ns1.spit.ac.in
DNSSEC: unsigned
URL of the ICANN Whois Inaccuracy Complaint Form: https://www.icann.org/wicf/
>>> Last update of WHOIS database: 2020-08-21T09:12:26Z <<<
```

```
ubuntu@ubuntu:~$ whois mit.edu.in
Domain Name: mit.edu.in
Registry Domain ID: D9581510-IN
Registrar WHOIS Server:
Registrar URL: http://www.ernet.in
Updated Date: 2017-12-12T06:28:29Z
Creation Date: 2015-06-22T07:54:24Z
Registry Expiry Date: 2026-06-22T07:54:24Z
Registrar: ERNET India
Registrar IANA ID: 800068
Registrar Abuse Contact Email:
Registrar Abuse Contact Phone:
Domain Status: ok http://www.icann.org/epp#OK
Registry Registrant ID:
Registrant Name:
Registrant Organization: Marathwada Institute of Technology
Registrant Street:
Registrant Street:
Registrant Street:
Registrant City:
Registrant State/Province:
Registrant Postal Code:
Registrant Country: IN
Registrant Phone:
```

Domain information

This type of information contains the general details about the domain. It will consist of the following fields.

- **Domain:** This field will give you the domain name which we are querying the WHOIS details.
- **Registrar:** This is the details of the registrar with whom the domain name is registered.
- **Registration Date:** This is the date when the domain name was first registered. With some whois lookup tools, it will be displayed as “Creation Date”.
- **Register Organization:** This is the name of the organisation.

Exercise 5: Because of NAT, the domain name spit.ac.in has a different IP address outside of SPIT than it does on campus. Using information in this lab and working on a home computer, find the outside IP address for spit.ac.in. Explain how you did it.

```
C:\Users\Vishal>nslookup spit.ac.in
Server: UnKnown
Address: 192.168.0.1

Non-authoritative answer:
Name: spit.ac.in
Address: 43.252.193.19

C:\Users\Vishal>tracert www.spit.ac.in

Tracing route to www.spit.ac.in [43.252.193.19]
over a maximum of 30 hops:

 1  1603 ms    2 ms    3 ms    192.168.0.1
 2    4 ms    3 ms    4 ms    45.112.56.246
 3    *        5 ms    7 ms    45.112.56.245
 4   216 ms    4 ms    5 ms    172.16.2.202
 5    8 ms    6 ms   84 ms   103.27.170.50
 6    7 ms    5 ms   94 ms   27.109.1.150
 7   10 ms   21 ms  171 ms   103.205.124.82
 8   494 ms    5 ms    9 ms   43.252.192.230
 9    *        *        *    Request timed out.
10    *        *        *    Request timed out.
11    *        *        *    Request timed out.
12    *        *        *    Request timed out.
13    *        *        *    Request timed out.
14    *        *        *    Request timed out.
15    *        *        *    Request timed out.
16    *        *        *    Request timed out.
17    *        *        *    Request timed out.
18    *        *        *    Request timed out.
19    *        *        *    Request timed out.
20    *        *        *    Request timed out.
21    *        *        *    Request timed out.
22    *        *        *    Request timed out.
23    *        *        *    Request timed out.
24    *        *        *    Request timed out.
25    *        *        *    Request timed out.
26    *        *        *    Request timed out.
27    *        *        *    Request timed out.
28    *        *        *    Request timed out.
29    *        *        *    Request timed out.
30    *        *        *    Request timed out.

Trace complete.

C:\Users\Vishal>
```

NAT is short for Network Address Translation. NAT is an Internet standard that enables a local-area network (LAN) to use one set of IP addresses for internal traffic and a second set of addresses for external traffic. Hence, we can see that the domain name spit.ac.in has a different IP address outside of SPIT than it does on campus.

Geolocation — A geolocation service tries to tell, approximately, where a given IP address is located physically. They can't be completely accurate—but they probably get at least the country right most of the time.

This geolocation program is not installed on our computers, but you can access one on the command line using the curl command, which can send HTTP requests and display the

response. The following command uses curl to contact a public web service that will look up an IP address for you: curl ipinfo.io/<IP-address>. For a specific example:

```
curl ipinfo.io/129.64.99.200
```

(As you can see, you get back more than just the location.)

Exercise 6: Find a few IP addresses that are connected to the web server on spit.ac.in right now, and determine where those IP addresses are located. (I'm expecting that there will be several; if not, try again in a few minutes or sometime later.) Find one that is far from Geneva, NY. Explain how you did it.

```
C:\Users\Vishal>curl ipinfo.io/43.252.192.230
{
  "ip": "43.252.192.230",
  "city": "Mumbai",
  "region": "Maharashtra",
  "country": "IN",
  "loc": "19.0728,72.8826",
  "org": "AS17625 BlazeNet's Network",
  "postal": "400070",
  "timezone": "Asia/Kolkata",
  "readme": "https://ipinfo.io/missingauth"
}
C:\Users\Vishal>
```

```
C:\Users\Vishal>curl ipinfo.io/129.64.99.200
{
  "ip": "129.64.99.200",
  "hostname": "websrv-prod.unet.brandeis.edu",
  "city": "Waltham",
  "region": "Massachusetts",
  "country": "US",
  "loc": "42.3765,-71.2356",
  "org": "AS10561 Brandeis University",
  "postal": "02453",
  "timezone": "America/New_York",
  "readme": "https://ipinfo.io/missingauth"
}
C:\Users\Vishal>curl ipinfo.io/192.168.0.1
{
  "ip": "192.168.0.1",
  "bogon": true
}
C:\Users\Vishal>curl ipinfo.io/172.16.2.202
{
  "ip": "172.16.2.202",
  "bogon": true
}
```

You can access location of IP address on the command line using the curl command, which can send HTTP requests and display the response. The following command uses curl to contact a public web service that will look up an IP address for you:

curl ipinfo.io/<IP-address>.

Reference:

1]Ping:

<https://www.imperva.com/learn/performance/round-trip-time-rtt/>

Conclusion:

Thus, I studied as well as implemented basic networking commands and utilities like ping, nslookup, ifconfig, netstat, traceroute, whois in detail.