Name: Neha Kale
Batch: B
UID: 2018130018

CEL 51, DCCN, Monsoon 2020
Lab 2: 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 the 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 the 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 — 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 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\kalen>ping -n 10 -l 64 www.princeton.edu

Pinging www.princeton.edu.cdn.cloudflare.net [104.18.4.101] with 64 bytes of data:
Reply from 104.18.4.101: bytes=64 time=7ms TTL=60
Reply from 104.18.4.101: bytes=64 time=10ms TTL=60
Reply from 104.18.4.101: bytes=64 time=10ms TTL=60
Reply from 104.18.4.101: bytes=64 time=7ms TTL=60
Reply from 104.18.4.101: bytes=64 time=8ms TTL=60
Reply from 104.18.4.101: bytes=64 time=9ms TTL=60
Reply from 104.18.4.101: bytes=64 time=24ms TTL=60
Reply from 104.18.4.101: bytes=64 time=8ms TTL=60
Reply from 104.18.4.101: bytes=64 time=9ms TTL=60
Reply from 104.18.4.101: bytes=64 time=8ms TTL=60

Ping statistics for 104.18.4.101:
    Packets: Sent = 10, Received = 10, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 7ms, Maximum = 24ms, Average = 10ms
```

```
C:\Users\kalen>ping -n 10 -l 100 www.princeton.edu

Pinging www.princeton.edu.cdn.cloudflare.net [104.18.4.101] with 100 bytes of data:
Reply from 104.18.4.101: bytes=100 time=36ms TTL=60
Reply from 104.18.4.101: bytes=100 time=11ms TTL=60
Reply from 104.18.4.101: bytes=100 time=8ms TTL=60
Reply from 104.18.4.101: bytes=100 time=8ms TTL=60
Reply from 104.18.4.101: bytes=100 time=7ms TTL=60
Reply from 104.18.4.101: bytes=100 time=7ms TTL=60
Reply from 104.18.4.101: bytes=100 time=14ms TTL=60
Reply from 104.18.4.101: bytes=100 time=13ms TTL=60
Reply from 104.18.4.101: bytes=100 time=12ms TTL=60
Reply from 104.18.4.101: bytes=100 time=11ms TTL=60

Ping statistics for 104.18.4.101:
    Packets: Sent = 10, Received = 10, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 7ms, Maximum = 36ms, Average = 12ms
```

```
C:\Users\kalen>ping -n 10 -l 500 www.princeton.edu

Pinging www.princeton.edu.cdn.cloudflare.net [104.18.4.101] with 500 bytes of data:
Reply from 104.18.4.101: bytes=500 time=25ms TTL=60
Reply from 104.18.4.101: bytes=500 time=16ms TTL=60
Reply from 104.18.4.101: bytes=500 time=12ms TTL=60
Reply from 104.18.4.101: bytes=500 time=8ms TTL=60
Reply from 104.18.4.101: bytes=500 time=8ms TTL=60
Reply from 104.18.4.101: bytes=500 time=6ms TTL=60
Reply from 104.18.4.101: bytes=500 time=6ms TTL=60
Reply from 104.18.4.101: bytes=500 time=14ms TTL=60
Reply from 104.18.4.101: bytes=500 time=28ms TTL=60
Reply from 104.18.4.101: bytes=500 time=16ms TTL=60

Ping statistics for 104.18.4.101:
    Packets: Sent = 10, Received = 10, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 6ms, Maximum = 28ms, Average = 13ms
```

```
C:\Users\kalen>ping -n 10 -l 1000 www.princeton.edu

Pinging www.princeton.edu.cdn.cloudflare.net [104.18.5.101] with 1000 bytes of data:
Reply from 104.18.5.101: bytes=1000 time=20ms TTL=60
Reply from 104.18.5.101: bytes=1000 time=50ms TTL=60
Reply from 104.18.5.101: bytes=1000 time=9ms TTL=60
Reply from 104.18.5.101: bytes=1000 time=10ms TTL=60
Reply from 104.18.5.101: bytes=1000 time=11ms TTL=60
Reply from 104.18.5.101: bytes=1000 time=28ms TTL=60
Reply from 104.18.5.101: bytes=1000 time=16ms TTL=60
Reply from 104.18.5.101: bytes=1000 time=18ms TTL=60
Reply from 104.18.5.101: bytes=1000 time=32ms TTL=60
Reply from 104.18.5.101: bytes=1000 time=6ms TTL=60

Ping statistics for 104.18.5.101:
    Packets: Sent = 10, Received = 10, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 6ms, Maximum = 50ms, Average = 20ms

C:\Users\kalen>ping -n 10 -l 1400 www.princeton.edu

Pinging www.princeton.edu.cdn.cloudflare.net [104.18.4.101] with 1400 bytes of data:
Reply from 104.18.4.101: bytes=1400 time=13ms TTL=60
Reply from 104.18.4.101: bytes=1400 time=12ms TTL=60
Reply from 104.18.4.101: bytes=1400 time=15ms TTL=60
Reply from 104.18.4.101: bytes=1400 time=9ms TTL=60
Reply from 104.18.4.101: bytes=1400 time=26ms TTL=60
Reply from 104.18.4.101: bytes=1400 time=21ms TTL=60
Reply from 104.18.4.101: bytes=1400 time=13ms TTL=60
Reply from 104.18.4.101: bytes=1400 time=14ms TTL=60
Reply from 104.18.4.101: bytes=1400 time=18ms TTL=60
Reply from 104.18.4.101: bytes=1400 time=95ms TTL=60

Ping statistics for 104.18.4.101:
    Packets: Sent = 10, Received = 10, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 9ms, Maximum = 95ms, Average = 23ms
```

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?

```
C:\Users\kalen>ping www.princeton.edu

Pinging www.princeton.edu.cdn.cloudflare.net [104.18.4.101] with 32 bytes of data:
Reply from 104.18.4.101: bytes=32 time=7ms TTL=60
Reply from 104.18.4.101: bytes=32 time=9ms TTL=60
Reply from 104.18.4.101: bytes=32 time=7ms TTL=60
Reply from 104.18.4.101: bytes=32 time=12ms TTL=60

Ping statistics for 104.18.4.101:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 7ms, Maximum = 12ms, Average = 8ms
```

```
C:\Users\kalen>ping www.facebook.com

Pinging star-mini.c10r.facebook.com [31.13.79.35] with 32 bytes of data:
Reply from 31.13.79.35: bytes=32 time=48ms TTL=58
Reply from 31.13.79.35: bytes=32 time=24ms TTL=58
Reply from 31.13.79.35: bytes=32 time=11ms TTL=58
Reply from 31.13.79.35: bytes=32 time=16ms TTL=58

Ping statistics for 31.13.79.35:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 11ms, Maximum = 48ms, Average = 24ms

C:\Users\kalen>ping www.youtube.com

Pinging youtube-ui.l.google.com [172.217.174.78] with 32 bytes of data:
Reply from 172.217.174.78: bytes=32 time=1173ms TTL=118
Reply from 172.217.174.78: bytes=32 time=54ms TTL=118
Reply from 172.217.174.78: bytes=32 time=11ms TTL=118
Reply from 172.217.174.78: bytes=32 time=23ms TTL=118

Ping statistics for 172.217.174.78:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 11ms, Maximum = 1173ms, Average = 315ms
```

Yes, the average RTT varies between different hosts.

Propagation delay is usually the dominant component in RTT. It ranges from a few milliseconds to hundreds of milliseconds depending on whether the endpoints are separated by a few kilometers or by an entire ocean.

The round trip time(RTT) can also be influenced by:

- 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?

RTT increases with increase in packet size. The observational results match with theoretical results. There is an increase in latency with increase in packet size due to transmission delay and propagation delay.

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 a 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).

The website based in India has an average RTT of 5ms while the one in the UK has an average RTT of 8ms. This demonstrates that with increase in physical distance the RTT increases.

```
C:\Users\kalen>ping www.india.gov.in

Pinging a1822.dscd.akamai.net [1.186.190.114] with 32 bytes of data:
Reply from 1.186.190.114: bytes=32 time=6ms TTL=61
Reply from 1.186.190.114: bytes=32 time=7ms TTL=61
Reply from 1.186.190.114: bytes=32 time=4ms TTL=61
Reply from 1.186.190.114: bytes=32 time=5ms TTL=61

Ping statistics for 1.186.190.114:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 4ms, Maximum = 7ms, Average = 5ms
```

```
C:\Users\kalen>ping www.ox.ac.uk

Pinging www.ox.ac.uk [151.101.2.133] with 32 bytes of data:
Reply from 151.101.2.133: bytes=32 time=12ms TTL=60
Reply from 151.101.2.133: bytes=32 time=7ms TTL=60
Reply from 151.101.2.133: bytes=32 time=8ms TTL=60
Reply from 151.101.2.133: bytes=32 time=8ms TTL=60

Ping statistics for 151.101.2.133:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 7ms, Maximum = 12ms, Average = 8ms
```

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 nslokup by adding the server name or IP address to the command: nslookup <host> <server>

```
C:\Users\kalen>nslookup www.india.gov.in
Server:  114.79.130.66.dvois.com
Address:  114.79.130.66

Non-authoritative answer:
Name:    in.domain.name
Address:  78.47.226.171
Aliases:  www.india.gov.in.domain.name
```

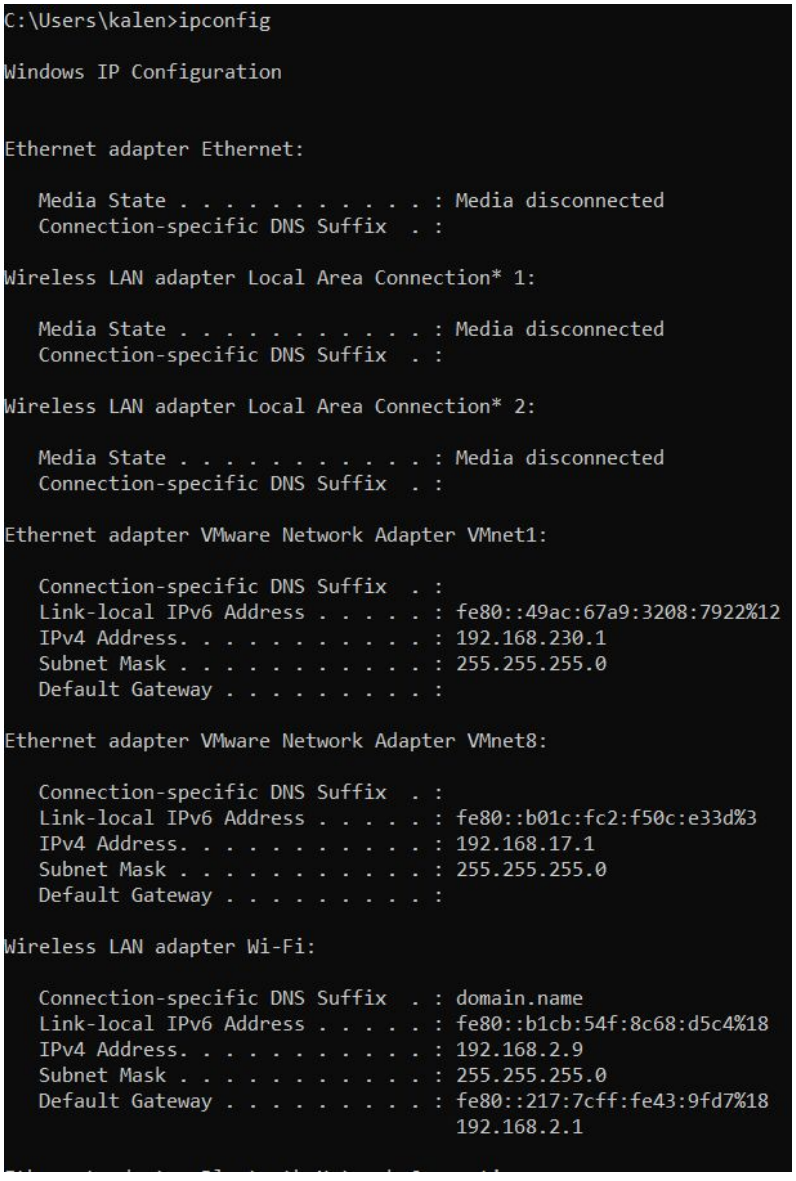
```
C:\Users\kalen>nslookup www.ox.ac.uk
Server:  114.79.130.66.dvois.com
Address:  114.79.130.66

Non-authoritative answer:
Name:    www.ox.ac.uk
Addresses:  151.101.2.133
          151.101.194.133
          151.101.66.133
          151.101.130.133
```

```
C:\Users\kalen>nslookup www.princeton.edu
Server:  114.79.130.66.dvois.com
Address:  114.79.130.66

Non-authoritative answer:
Name:    www.princeton.edu.cdn.cloudflare.net
Addresses:  2606:4700::6812:565
          2606:4700::6812:465
          104.18.4.101
          104.18.5.101
Aliases:  www.princeton.edu
```

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\kalen>ipconfig

Windows IP Configuration


Ethernet adapter Ethernet:

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

Wireless LAN adapter Local Area Connection* 1:

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

Wireless LAN adapter Local Area Connection* 2:

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

Ethernet adapter VMware Network Adapter VMnet1:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::49ac:67a9:3208:7922%12
   IPv4 Address. . . . . . . . . . . : 192.168.230.1
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . :

Ethernet adapter VMware Network Adapter VMnet8:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::b01c:fc2:f50c:e33d%3
   IPv4 Address. . . . . . . . . . . : 192.168.17.1
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . :

Wireless LAN adapter Wi-Fi:

   Connection-specific DNS Suffix  . : domain.name
   Link-local IPv6 Address . . . . . : fe80::b1cb:54f:8c68:d5c4%18
   IPv4 Address. . . . . . . . . . . : 192.168.2.9
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : fe80::217:7cff:fe43:9fd7%18
                                       192.168.2.1
```

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\kalen>netstat

Active Connections

  Proto  Local Address          Foreign Address        State
  TCP    127.0.0.1:9012         LAPTOP-6I3L5S6B:49935  ESTABLISHED
  TCP    127.0.0.1:9487         LAPTOP-6I3L5S6B:49934  ESTABLISHED
  TCP    127.0.0.1:49673        LAPTOP-6I3L5S6B:49922  ESTABLISHED
  TCP    127.0.0.1:49674        LAPTOP-6I3L5S6B:49675  ESTABLISHED
  TCP    127.0.0.1:49675        LAPTOP-6I3L5S6B:49674  ESTABLISHED
  TCP    127.0.0.1:49690        LAPTOP-6I3L5S6B:49813  ESTABLISHED
  TCP    127.0.0.1:49690        LAPTOP-6I3L5S6B:49855  ESTABLISHED
  TCP    127.0.0.1:49691        LAPTOP-6I3L5S6B:49692  ESTABLISHED
  TCP    127.0.0.1:49692        LAPTOP-6I3L5S6B:49691  ESTABLISHED
  TCP    127.0.0.1:49706        LAPTOP-6I3L5S6B:49707  ESTABLISHED
  TCP    127.0.0.1:49707        LAPTOP-6I3L5S6B:49706  ESTABLISHED
  TCP    127.0.0.1:49708        LAPTOP-6I3L5S6B:61900  ESTABLISHED
  TCP    127.0.0.1:49709        LAPTOP-6I3L5S6B:49710  ESTABLISHED
  TCP    127.0.0.1:49710        LAPTOP-6I3L5S6B:49709  ESTABLISHED
  TCP    127.0.0.1:49711        LAPTOP-6I3L5S6B:49712  ESTABLISHED
  TCP    127.0.0.1:49712        LAPTOP-6I3L5S6B:49711  ESTABLISHED
  TCP    127.0.0.1:49713        LAPTOP-6I3L5S6B:61900  ESTABLISHED
  TCP    127.0.0.1:49714        LAPTOP-6I3L5S6B:49715  ESTABLISHED
  TCP    127.0.0.1:49715        LAPTOP-6I3L5S6B:49714  ESTABLISHED
  TCP    127.0.0.1:49716        LAPTOP-6I3L5S6B:49717  ESTABLISHED
  TCP    127.0.0.1:49717        LAPTOP-6I3L5S6B:49716  ESTABLISHED
  TCP    127.0.0.1:49718        LAPTOP-6I3L5S6B:61900  ESTABLISHED
  TCP    127.0.0.1:49719        LAPTOP-6I3L5S6B:49720  ESTABLISHED
  TCP    127.0.0.1:49720        LAPTOP-6I3L5S6B:49719  ESTABLISHED
  TCP    127.0.0.1:49725        LAPTOP-6I3L5S6B:49730  ESTABLISHED
  TCP    127.0.0.1:49725        LAPTOP-6I3L5S6B:49735  ESTABLISHED
  TCP    127.0.0.1:49725        LAPTOP-6I3L5S6B:49737  ESTABLISHED
  TCP    127.0.0.1:49725        LAPTOP-6I3L5S6B:49741  ESTABLISHED
  TCP    127.0.0.1:49725        LAPTOP-6I3L5S6B:49742  ESTABLISHED
  TCP    127.0.0.1:49725        LAPTOP-6I3L5S6B:49743  ESTABLISHED
  TCP    127.0.0.1:49725        LAPTOP-6I3L5S6B:49755  ESTABLISHED
  TCP    127.0.0.1:49725        LAPTOP-6I3L5S6B:49777  ESTABLISHED
  TCP    127.0.0.1:49730        LAPTOP-6I3L5S6B:49725  ESTABLISHED
  TCP    127.0.0.1:49735        LAPTOP-6I3L5S6B:49725  ESTABLISHED
  TCP    127.0.0.1:49737        LAPTOP-6I3L5S6B:49725  ESTABLISHED
  TCP    127.0.0.1:49741        LAPTOP-6I3L5S6B:49725  ESTABLISHED
  TCP    127.0.0.1:49742        LAPTOP-6I3L5S6B:49725  ESTABLISHED
  TCP    127.0.0.1:49743        LAPTOP-6I3L5S6B:49725  ESTABLISHED
  TCP    127.0.0.1:49746        LAPTOP-6I3L5S6B:49747  ESTABLISHED
  TCP    127.0.0.1:49747        LAPTOP-6I3L5S6B:49746  ESTABLISHED
  TCP    127.0.0.1:49748        LAPTOP-6I3L5S6B:61900  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 the 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 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: telent <host> <port>. For example, to connect to the web server on www.spit.ac.in: telnet spit.ac.in 80

traceroute — 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,..., 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`).

1.2.1 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

```
C:\Users\kalen>tracert www.ox.ac.uk

Tracing route to www.ox.ac.uk [151.101.2.133]
over a maximum of 30 hops:

  1     7 ms     5 ms     2 ms  192.168.2.1
  2    15 ms    15 ms    13 ms  1.186.179.1.dvois.com [1.186.179.1]
  3     7 ms    13 ms     6 ms  114.79.129.97.dvois.com [114.79.129.97]
  4     *        *        *     Request timed out.
  5     *        *        *     Request timed out.
  6     *        *        *     Request timed out.
  7     *        *        *     Request timed out.
  8     9 ms     6 ms     8 ms  151.101.2.133

Trace complete.
```

```
C:\Users\kalen>tracert www.princeton.edu

Tracing route to www.princeton.edu.cdn.cloudflare.net [104.18.4.101]
over a maximum of 30 hops:

  1     7 ms     3 ms     6 ms  192.168.2.1
  2    13 ms     7 ms     6 ms  1.186.179.1.dvois.com [1.186.179.1]
  3    20 ms    84 ms    15 ms  114.79.129.97.dvois.com [114.79.129.97]
  4   146 ms    53 ms    75 ms  103.27.170.48
  5     7 ms     9 ms     7 ms  104.18.4.101

Trace complete.
```

```
C:\Users\kalen>tracert timesofindia.indiatimes.com

Tracing route to e12582.dsce12.akamaiedge.net [104.120.58.167]
over a maximum of 30 hops:

  1     2 ms     2 ms     4 ms  192.168.2.1
  2    14 ms     6 ms     6 ms  1.186.179.1.dvois.com [1.186.179.1]
  3     4 ms     5 ms    23 ms  114.79.129.97.dvois.com [114.79.129.97]
  4    85 ms    62 ms    84 ms  103.27.170.100
  5     9 ms     5 ms     9 ms  a104-120-58-167.deploy.static.akamaitechnologies.com [104.120.58.167]

Trace complete.
```

```
C:\Users\kalen>tracert www.india.gov.in

Tracing route to a1822.dscd.akamai.net [1.186.190.114]
over a maximum of 30 hops:

  1     4 ms    13 ms     9 ms  192.168.2.1
  2    37 ms     7 ms     5 ms  1.186.179.1.dvois.com [1.186.179.1]
  3     7 ms    10 ms     4 ms  114.79.129.97.dvois.com [114.79.129.97]
  4     8 ms    10 ms     9 ms  1.186.190.114.dvois.com [1.186.190.114]

Trace complete.
```

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

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

  1     2 ms     2 ms     2 ms  192.168.2.1
  2     8 ms     7 ms     9 ms  1.186.179.1.dvois.com [1.186.179.1]
  3    18 ms     6 ms     7 ms  114.79.129.97.dvois.com [114.79.129.97]
  4     *        *        *     Request timed out.
  5     *        *        *     Request timed out.
  6     *        *        *     Request timed out.
  7     *        *        *     Request timed out.
  8     *        *        *     Request timed out.
  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.
```

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`).

Exercise 2: (Very short.) 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\kalen>tracert www.princeton.edu

Tracing route to www.princeton.edu.cdn.cloudflare.net [104.18.5.101]
over a maximum of 30 hops:

  1     9 ms    11 ms    11 ms  192.168.2.1
  2    15 ms    11 ms    12 ms  1.186.179.1.dvois.com [1.186.179.1]
  3    43 ms     8 ms   117 ms  114.79.129.97.dvois.com [114.79.129.97]
  4     8 ms    47 ms     6 ms  103.27.170.48
  5     9 ms     9 ms     7 ms  104.18.5.101

Trace complete.
```

```
C:\Users\kalen>tracert www.alumni.princeton.edu

Tracing route to acquia-psb.princeton.edu [34.228.150.183]
over a maximum of 30 hops:

  1     6 ms     4 ms     6 ms  192.168.2.1
  2    12 ms    42 ms     4 ms  1.186.179.1.dvois.com [1.186.179.1]
  3    19 ms    13 ms     7 ms  114.79.129.97.dvois.com [114.79.129.97]
  4     9 ms     6 ms     7 ms  1.7.160.224
  5     *        *        *     Request timed out.
  6     *        *        *     Request timed out.
  7     *        *        *     Request timed out.
  8     *        *        *     Request timed out.
  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.
```

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.

This was done on 25th August 2020.

```
C:\Users\kalen>tracert www.csail.mit.edu

Tracing route to fe3.edge.pantheon.io [23.185.0.3]
over a maximum of 30 hops:

  1    17 ms     2 ms     8 ms  192.168.2.1
  2    11 ms     5 ms     4 ms  1.186.179.1.dvois.com [1.186.179.1]
  3     5 ms     7 ms     4 ms  114.79.129.97.dvois.com [114.79.129.97]
  4     *        *        *     Request timed out.
  5     *        *        *     Request timed out.
  6     *        *        *     Request timed out.
  7     *        *        *     Request timed out.
  8     9 ms    10 ms     9 ms  23.185.0.3

Trace complete.
```

This was done on 31st August 2020.

```
C:\Users\kalen>tracert www.csail.mit.edu

Tracing route to fe3.edge.pantheon.io [23.185.0.3]
over a maximum of 30 hops:

  1     2 ms     3 ms     2 ms  192.168.2.1
  2     7 ms     6 ms     5 ms  1.186.179.1.dvois.com [1.186.179.1]
  3     5 ms     5 ms     5 ms  114.79.129.97.dvois.com [114.79.129.97]
  4     *        *        *     Request timed out.
  5     *        *        *     Request timed out.
  6     *        *        *     Request timed out.
  7     *        *        *     Request timed out.
  8     5 ms     6 ms     6 ms  23.185.0.3

Trace complete.
```

From the above experiments i can conclude that for the same website, when the packets are sent at different times, the RTT taken is different and the path they take is different as well.

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, the tracerouting follows a particular path from the user's IP address through the IP addresses of the ISP and then the path really depends on which access point is ready to respond and which access points or routers have firewalls configured for blocking the requests and accordingly, the destination can be reached through different paths at different times.

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?
   A hop depends on the location of the host. If the distance between the location of the user and that of the destination url 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 a certain maximum hops. The same relationship may not hold for each host as it really depends on the time which the host takes to respond. If the host responds in the first request itself, the tracerouting stops with a success message.

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 in. *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: (Short.) 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.

The whois command gives information about the domain name, the Registry Domain ID and some other details such as the details of the Registrar and the Registrant.

Exercise 5: (Should be short.) 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.

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.

CONCLUSION
Learnt about some basic command line network utilities.
Learnt about Network Latency, RTT and the factors impacting RTT.
Learnt that network depends a lot on the time when the experiment is performed and on the host.