ASSIGNMENT - 2

(Computer Networks Laboratory)

[Usage of Command Line Commands]

Name: Kinjal Raykarmakar

Enrolment No.: 12018009019439

Section: 3H

Roll No.: 29

---

1. netstat

Description:

netstat displays active TCP connections, ports on which the computer is listening, Ethernet statistics, the IP routing table, IPv4 statistics (for the IP, ICMP, TCP, and UDP protocols), and IPv6 statistics (for the IPv6, ICMPv6, TCP over IPv6, and UDP over IPv6 protocols). Used without parameters, this command displays active TCP connections.

Help:

```
Displays protocol statistics and current TCP/IP network connections.

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

Output:

```
Active Connections

  Proto  Local Address          Foreign Address        State
  TCP    127.0.0.1:1521         kubernetes:49873       ESTABLISHED
  TCP    127.0.0.1:49873        kubernetes:1521        ESTABLISHED
  TCP    127.0.0.1:52182        kubernetes:52183       ESTABLISHED
  TCP    127.0.0.1:52183        kubernetes:52182       ESTABLISHED
  TCP    127.0.0.1:52185        kubernetes:52186       ESTABLISHED
  TCP    127.0.0.1:52186        kubernetes:52185       ESTABLISHED
  TCP    127.0.0.1:52190        kubernetes:52191       ESTABLISHED
  TCP    127.0.0.1:52191        kubernetes:52190       ESTABLISHED
  TCP    127.0.0.1:52192        kubernetes:52193       ESTABLISHED
  TCP    127.0.0.1:52193        kubernetes:52192       ESTABLISHED
  TCP    192.168.0.101:51171    74.125.24.188:5228     ESTABLISHED
  TCP    192.168.0.101:51480    117.18.237.29:http     CLOSE_WAIT
  TCP    192.168.0.101:51482    117.18.232.200:https   CLOSE_WAIT
  TCP    192.168.0.101:51486    a23-221-53-10:https    CLOSE_WAIT
  TCP    192.168.0.101:51581    bom07s18-in-f5:https   ESTABLISHED
  TCP    192.168.0.101:52189    ec2-3-235-82-221:https  CLOSE_WAIT
  TCP    192.168.0.101:52196    ec2-3-235-69-46:https  CLOSE_WAIT
  TCP    192.168.0.101:52202    134.224.133.126:https  ESTABLISHED
  TCP    192.168.0.101:52205    ec2-3-235-72-242:https  CLOSE_WAIT
  TCP    192.168.0.101:52220    ec2-18-205-93-223:https  ESTABLISHED
  TCP    192.168.0.101:52221    ec2-3-80-20-236:https  CLOSE_WAIT
  TCP    192.168.0.101:52224    ec2-3-235-69-7:https   CLOSE_WAIT
  TCP    192.168.0.101:52230    ec2-3-80-20-236:https  CLOSE_WAIT
  TCP    192.168.0.101:52231    ec2-3-235-82-188:https  ESTABLISHED
  TCP    192.168.0.101:52234    ec2-3-235-83-2:https   CLOSE_WAIT
  TCP    192.168.0.101:52235    ec2-3-235-83-2:https   CLOSE_WAIT
  TCP    192.168.0.101:52238    ec2-3-235-83-2:https   CLOSE_WAIT
  TCP    192.168.0.101:52239    ec2-3-80-20-173:https  CLOSE_WAIT
  TCP    192.168.0.101:52263    cdn-185-199-108-133:https  ESTABLISHED
  TCP    192.168.0.101:52273    dns:https              ESTABLISHED
  TCP    192.168.0.101:52287    bom12s18-in-f14:https  TIME_WAIT
  TCP    192.168.0.101:52288    bom07s24-in-f14:https  TIME_WAIT
  TCP    192.168.0.101:52289    bom12s11-in-f14:https  TIME_WAIT
  TCP    192.168.0.101:52291    117.18.232.200:https   ESTABLISHED
```

2.netsh

Description:

netsh is a command-line scripting utility that allows you to display or modify the network configuration of a computer that is currently running. Netsh commands can be run by typing commands at the netsh prompt and they can be used in batch files or scripts. Remote computers and the local computer can be configured by using netsh commands.

Help:

```
Usage: netsh [-a AliasFile] [-c Context] [-r RemoteMachine] [-u [DomainName\]UserName] [-p Password | *]
             [Command | -f ScriptFile]

The following commands are available:

Commands in this context:
?               - Displays a list of commands.
add             - Adds a configuration entry to a list of entries.
advfirewall     - Changes to the `netsh advfirewall' context.
branchcache     - Changes to the `netsh branchcache' context.
bridge          - Changes to the `netsh bridge' context.
delete          - Deletes a configuration entry from a list of entries.
dhcpclient      - Changes to the `netsh dhcpclient' context.
dnsclient       - Changes to the `netsh dnsclient' context.
dump            - Displays a configuration script.
exec            - Runs a script file.
firewall        - Changes to the `netsh firewall' context.
help            - Displays a list of commands.
http            - Changes to the `netsh http' context.
interface       - Changes to the `netsh interface' context.
ipsec           - Changes to the `netsh ipsec' context.
lan             - Changes to the `netsh lan' context.
mbn             - Changes to the `netsh mbn' context.
namespace       - Changes to the `netsh namespace' context.
netio           - Changes to the `netsh netio' context.
p2p             - Changes to the `netsh p2p' context.
ras             - Changes to the `netsh ras' context.
rpc             - Changes to the `netsh rpc' context.
set             - Updates configuration settings.
show            - Displays information.
trace           - Changes to the `netsh trace' context.
wcn             - Changes to the `netsh wcn' context.
wfp             - Changes to the `netsh wfp' context.
winhttp         - Changes to the `netsh winhttp' context.
winsock         - Changes to the `netsh winsock' context.
wlan            - Changes to the `netsh wlan' context.
```

```
The following sub-contexts are available:
 advfirewall branchcache bridge dhcpclient dnsclient firewall http interface ipsec lan mbn namespace netio p2p ras rpc trace wcn wfp winhttp winsock
 wlan

To view help for a command, type the command, followed by a space, and then
 type ?.
```

Output:

```
C:\Users\kinja>netsh
netsh>wlan show profiles

Profiles on interface Wi-Fi:

Group policy profiles (read only)
---------------------------------
    <None>

User profiles
-------------
    All User Profile     : GamerKRK2k
    All User Profile     : realme 5
    All User Profile     : hyperHAWK
    All User Profile     : SRK WiFi 2
    All User Profile     : Nokia 6.1 Plus
    All User Profile     : Galaxy M30sF056
    All User Profile     : Nokia 5.1 Plus
    All User Profile     : Redmi note 7 pro
    All User Profile     : sysnet_network
    All User Profile     : SRK WiFi
```

3.arp

Description:

arp displays and modifies entries in the Address Resolution Protocol (ARP) cache. The ARP cache contains one or more tables that are used to store IP addresses and their resolved Ethernet or Token Ring physical addresses. There is a separate table for each Ethernet or Token Ring network adapter installed on your computer. Used without parameters, arp displays help information.

Help:

```
Displays and modifies the IP-to-Physical address translation tables used by
address resolution protocol (ARP).

ARP -s inet_addr eth_addr [if_addr]
ARP -d inet_addr [if_addr]
ARP -a [inet_addr] [-N if_addr] [-v]

  -a            Displays current ARP entries by interrogating the current
                protocol data.  If inet_addr is specified, the IP and Physical
```

```
               addresses for only the specified computer are displayed.  If
               more than one network interface uses ARP, entries for each ARP
               table are displayed.
  -g           Same as -a.
  -v           Displays current ARP entries in verbose mode.  All invalid
               entries and entries on the loop-back interface will be shown.
  inet_addr    Specifies an internet address.
  -N if_addr   Displays the ARP entries for the network interface specified
               by if_addr.
  -d           Deletes the host specified by inet_addr. inet_addr may be
               wildcarded with * to delete all hosts.
  -s           Adds the host and associates the Internet address inet_addr
               with the Physical address eth_addr.  The Physical address is
               given as 6 hexadecimal bytes separated by hyphens. The entry
               is permanent.
  eth_addr     Specifies a physical address.
  if_addr      If present, this specifies the Internet address of the
               interface whose address translation table should be modified.
               If not present, the first applicable interface will be used.
Example:
  > arp -s 157.55.85.212   00-aa-00-62-c6-09  .... Adds a static entry.
  > arp -a                                    .... Displays the arp table.
```

Output:

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

Interface: 192.168.40.1 --- 0x3
  Internet Address      Physical Address      Type
  192.168.40.255        ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  224.0.0.252           01-00-5e-00-00-fc     static
  239.255.255.250       01-00-5e-7f-ff-fa     static

Interface: 192.168.0.101 --- 0x4
  Internet Address      Physical Address      Type
  192.168.0.1           84-16-f9-3e-30-0a     dynamic
  192.168.0.102         e4-5d-75-a0-4e-8b     dynamic
  192.168.0.255         ff-ff-ff-ff-ff-ff     static
  224.0.0.2             01-00-5e-00-00-02     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  224.0.0.252           01-00-5e-00-00-fc     static
  224.0.0.253           01-00-5e-00-00-fd     static
  239.255.255.250       01-00-5e-7f-ff-fa     static
  255.255.255.255       ff-ff-ff-ff-ff-ff     static
```

```
Interface: 192.168.63.1 --- 0x12
  Internet Address      Physical Address      Type
  192.168.63.255        ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  224.0.0.252           01-00-5e-00-00-fc     static
  239.255.255.250       01-00-5e-7f-ff-fa     static
```

4.finger

Description:

Displays information about users on a specified remote computer (typically a computer running UNIX) that is running the finger service or daemon. The remote computer specifies the format and output of the user information display. Used without parameters, finger displays help.

Help:

```
Displays information about a user on a specified system running the
Finger service. Output varies based on the remote system.

FINGER [-l] [user]@host [...]

  -l        Displays information in long list format.
  user      Specifies the user you want information about. Omit the user
            parameter to display information about all users on the
            specifed host.
  @host     Specifies the server on the remote system whose users you
            want information about.
```

Output:

```
C:\Users\kinja>finger kinja@Kinjal-Lappy

[Kinjal-Lappy]
> Finger: connect::Connection refused
```

5.net view

Description:

net view displays a list of domains, computers, or resources that are being shared by the specified computer. Used without parameters, net view displays a list of computers in your current domain.

Help:

```
C:\Users\kinja>net view /?
The syntax of this command is:

NET VIEW
[\\computername [/CACHE] | [/ALL] | /DOMAIN[:domainname]]
```

Output:

```
C:\Users\kinja>net view
System error 6118 has occurred.

The list of servers for this workgroup is not currently available
```

Conclusion:

All the network commands namely: `netstat`, `netsh`, `arp`, `finger` and `net view` were executed successfully using the windows command prompt.

ASSIGNMENT - 3

(Computer Networks Laboratory)

[Client-Server Connection – Packet Tracer]

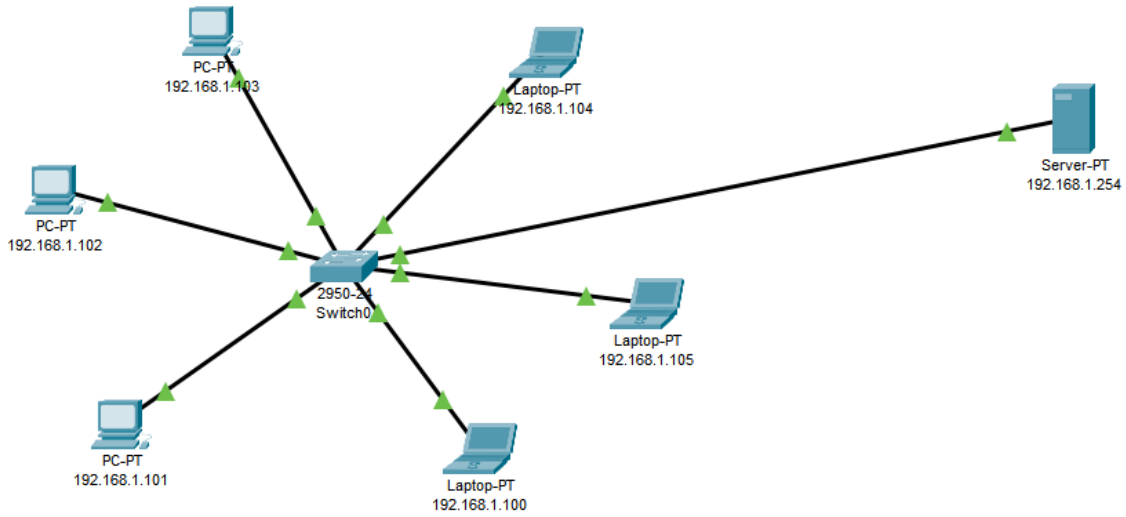
Name: Kinjal Raykarmakar

Enrolment No.: 12018009019439

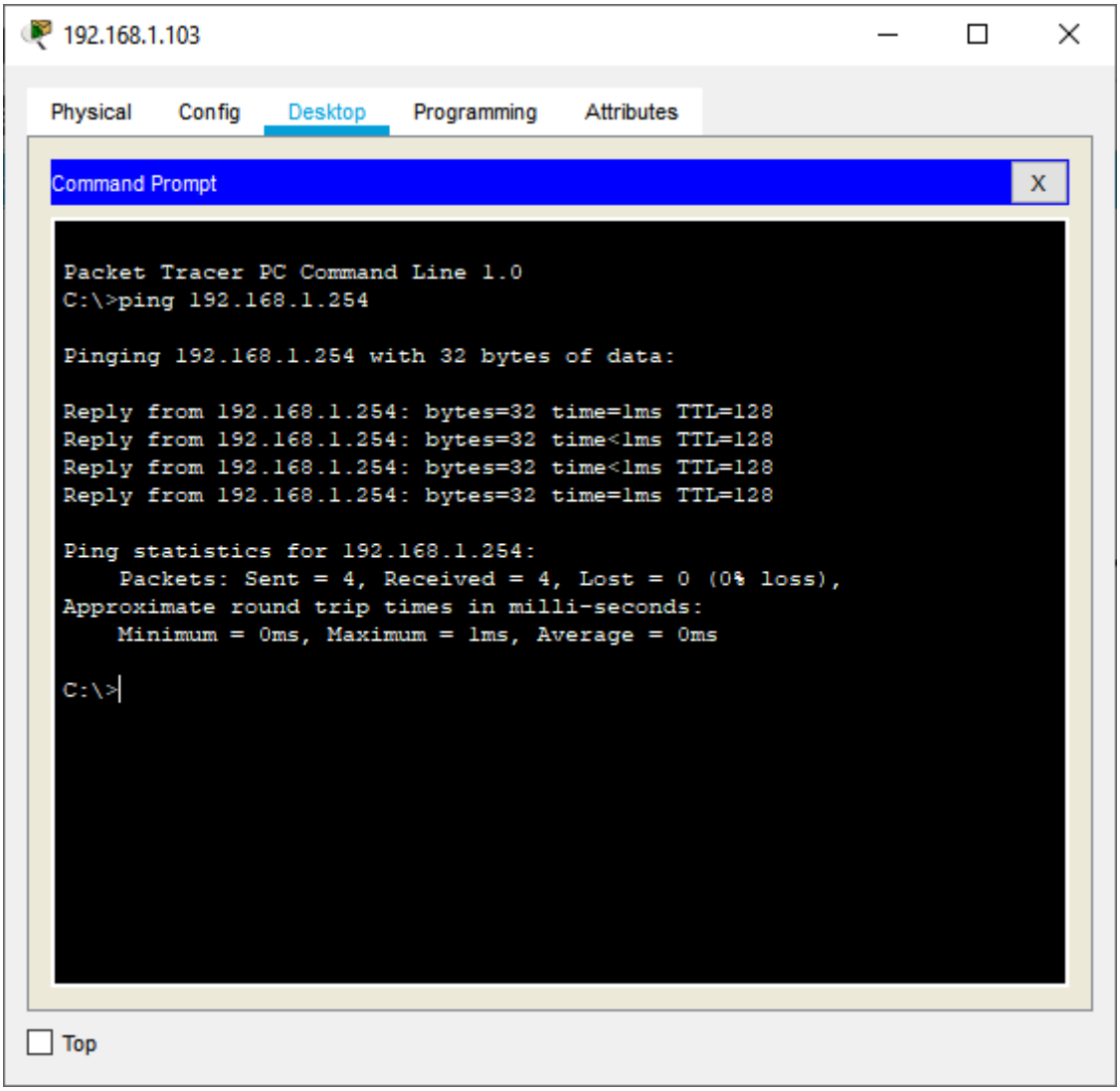
Section: 3H

Roll No.: 29

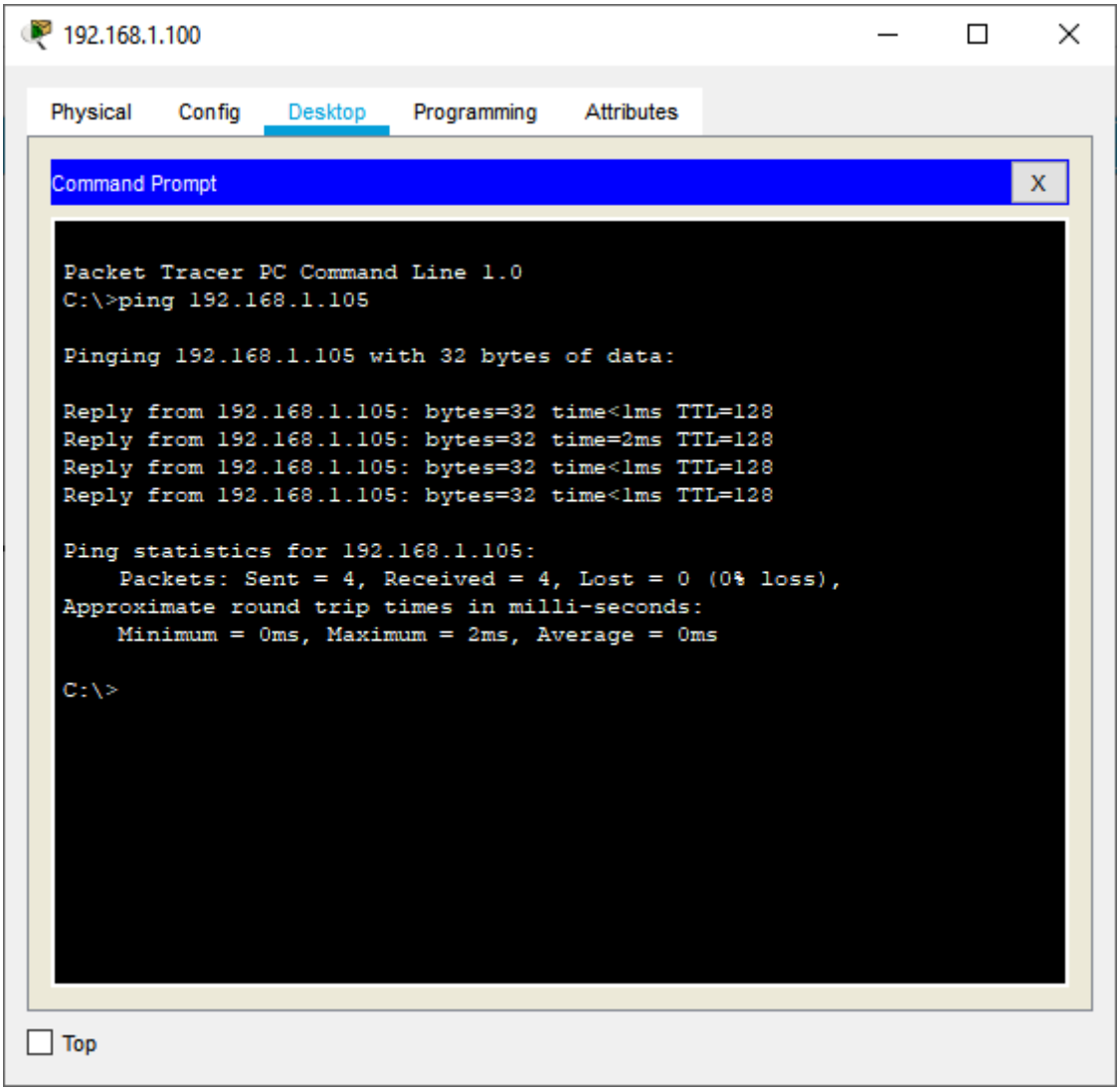
Connections



Pinging the server from a client



Pinging from one client to another



WEEK - 3 ASSIGNMENTS

Computer Networks Laboratory

Name: Kinjal Raykarmakar

Section: CSE 3H

Roll No.: 29

Enrollment No.: 12018009019439

Amplitude Modulation

In [1]:
```
import numpy as np
import matplotlib.pyplot as plt
```

In [6]:
```
Ac = float(input("Enter the amplitute of Carrier"))
Fc = float(input("Enter the frequency of Carrier"))

Am = float(input("Enter the amplitute of Message"))
Fm = float(input("Enter the frequency of Message"))
```

In [7]:
```
t = np.linspace(0, 1, 1000)
```

In [8]:
```
m = Am * np.cos(2 * np.pi *Fm * t)

c = Ac * np.cos(2 * np.pi *Fc * t)

AM = (Ac + m) * np.cos(2 * np.pi * Fc * t)
```

In [11]:
```
plt.title("Amplitude Modulation")

plt.subplot(3, 1, 1)
plt.plot(m, "g")
plt.ylabel("Amplitude")
plt.xlabel("Message")

plt.subplot(3, 1, 2)
plt.plot(c, "r")
plt.ylabel("Amplitude")
plt.xlabel("Carrier")

plt.subplot(3, 1, 3)
plt.plot(AM, "b")
plt.ylabel("Amplitude")
plt.xlabel("AM")

# Adjustments
plt.subplots_adjust(hspace=1)
plt.rc("font", size=15)

fig = plt.gcf()
fig.set_size_inches(16, 9)

fig.show()

fig.savefig("AM-ipynb.png", dpi=100)
```