Experiment No.2

Aim: TO EXECUTE AND IMPLEMENT IP NETWORKING AND NETWORK COMMAND:PING.

SOFTWARE REQUIRED: LINUX, WINDOWS, MAC.

Procedure:

There are a variety of reasons for wanting to ping any given IP address. You may be a network administrator, attempting to gather information about what is happening on a particular computer or server, or you may be a gamer trying to determine what the latency times are for a locally-hosted game. Whatever the reason, pinging an IP address can come in handy in many circumstances and knowing how to do it from any platform is key. There are also automatic programs designed to do this for you and depending on your circumstance, you may find these easier to use. The following describes how to manually ping IP address on Windows, Mac and Linux-based systems as well as how to use automatic ping tools.

Manually Ping on Windows

In order to run these commands on Windows, you will first need to open your command prompt. Open the command window by clicking „Start", then select „Run". Type „cmd" into the run program box, and hit enter or press „OK". From here, type „ping host name" or „ping IP address". For example, if you want to ping yahoo.com (a host name), you would type „ping www.yahoo.com„. If you want to ping the specific IP address 22.87.47.67, you would type „ping 22.87.47.67. After typing either one of these in, press the „Enter" key on your keyboard. When the results appear, look at the section labeled „pingOutput" for your results.

Manually Ping on Mac

In order to begin the pinging process on systems running the Mac OS, you will need to navigate to the Applications folders, followed by Utilities and then Network Utility. From inside this interface, you can quickly check the status of an IP address or host name by inserting the appropriate address into the domain/IP Address box in the „Ping" section of the Network Utility. After inputting the needed information, click „Ping" and your results should appear rather instantaneously.

Manually Ping on Linux

Doing a manual ping on Linux is just as simple as doing it on Windows or Mac-based systems. Start off by opening a Telnet or TERMINAL window; this is where you will enter your lines and is similar to Window"s „CMD" executable. The process works very similar to Windows. "ping IP address" (just like in Windows) to properly direct the command system to check that IP address or host name. Once inputted, press „Enter" and view the „pingOutput" information that will appear upon completion of the pinging process

Name : Vivek Vishwakarma Class : TEET2 Roll No. : 36 Subject : CCN
Date of Performance : 19/01/2022 Date of Submission : 19/01/2022

Hostname

Automatically Pinging an IP Address or

Rather than worrying about built-in command prompts that can be frustrating and difficult to use for some, try using the Pingler Ping Domain/IP tool, which can automatically grab the results of a ping for you in the blink of the eye. Simply insert your IP or domain, click „Continue" and fetch the corresponding piece of information (IP if it"s a domain name and vice-versa). If you are constantly having to check and compare pieces of information about multiple websites, domains and IP addresses, this browser-based tool makes for a great solution to repetitive, complicating cross-checking.

Theory & Results:

Effective Ping Command Examples

As you already know, ping command is used to find out whether the peer host/gateway is reachable.

If you are thinking ping is such a simple command and why do I need 15 examples, you should read the rest of the article.

Ping command provides lot more options than what you might already know.

Ping Example 1. Increase or Decrease the Time Interval between Packets

By default ping waits for 1 second before sending the next packet. You can increase or decrease this using option -i as shown below.

Increase Ping Time Interval

Example: Wait for 5 seconds before sending the next packet.

```
$ ping -i 5 IP
```

```
C:\Users\Vivek>ping -i 5 192.168.0.104

Pinging 192.168.0.104 with 32 bytes of data:
Reply from 192.168.0.104: bytes=32 time<1ms TTL=128
Reply from 192.168.0.104: bytes=32 time<1ms TTL=128
Reply from 192.168.0.104: bytes=32 time<1ms TTL=128
Reply from 192.168.0.104: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.0.104:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

Decrease Ping Time Interval

Example: Wait 0.1 seconds before sending the next packet.

```
# ping -i 0.1 IP
```

```
C:\Users\Vivek>ping -i 0.1 192.168.0.104
Bad value for option -i, valid range is from 1 to 255.
```

Note: Only super user can specify interval less than 0.2 seconds. If not, you"ll get the following error message.

```
$ ping -i 0.1 127.0.0.1

PING 0 (127.0.0.1) 56(84) bytes of data.

ping: cannot flood; minimal interval, allowed for user, is 200ms
```

Ping Example 2. Check whether the local network interface is up and running

Before checking whether the peer machine is reachable, first check whether the local network network is up and running using any one of the following 3 methods.

Ping localhost using zero (0)

This is probably the easiest and simplest way to ping a local host

```
$ ping 0

PING 0 (127.0.0.1) 56(84) bytes of data.

64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.024 ms

^C
```

```
C:\Users\Vivek>ping 0

Pinging 0.0.0.0 with 32 bytes of data:
PING: transmit failed. General failure.
PING: transmit failed. General failure.
PING: transmit failed. General failure.
PING: transmit failed. General failure.

Ping statistics for 0.0.0.0:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

Ping localhost using name

```
$ ping localhost

PING localhost (127.0.0.1) 56(84) bytes of data.

64 bytes from localhost (127.0.0.1): icmp_seq=1 ttl=64 time=0.051 ms

64 bytes from localhost (127.0.0.1): icmp_seq=2 ttl=64 time=0.055 ms
```

```
^C

--- localhost ping statistics ---

2 packets transmitted, 2 received, 0% packet loss, time 999ms

rtt min/avg/max/mdev = 0.051/0.053/0.055/0.002 ms
```

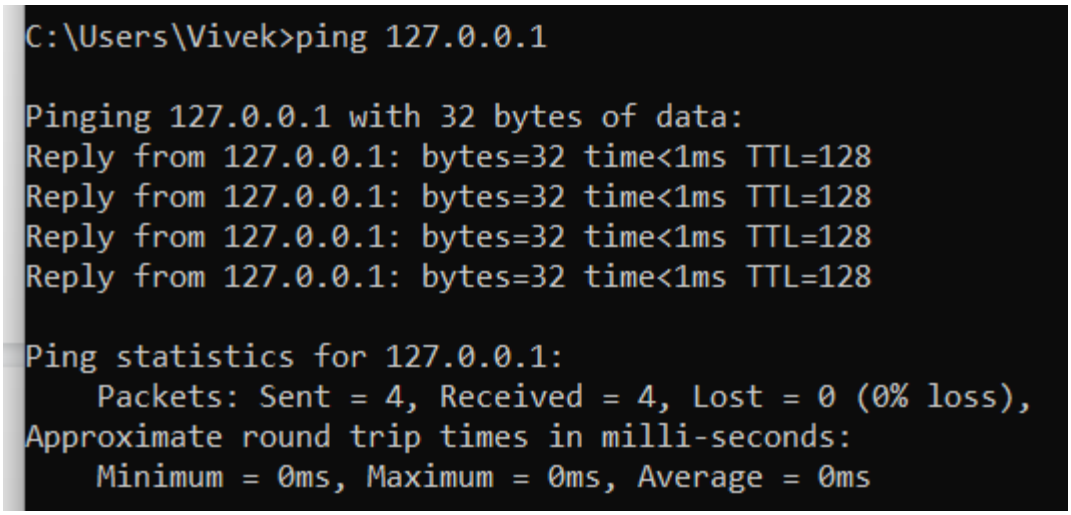
```
C:\Users\Vivek>ping localhost

Pinging VIVEK [::1] with 32 bytes of data:
Reply from ::1: time<1ms
Reply from ::1: time<1ms
Reply from ::1: time<1ms
Reply from ::1: time<1ms

Ping statistics for ::1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

Ping localhost using ip

```
$ ping 127.0.0.1
```

To quit the ping command, send SIGINT signal by pressing CTRL+C. If you have not specified any option to make the ping to exit automatically, then you will be terminating using CTRL+C (SIGINT) which will show the statistics and then terminate the ping process. When everything is working properly, it should say "0% packet loss"

2 packets transmitted, 2 received, **0% packet loss,** time 999ms
rtt min/ avg /max/mdev = 0.051/0.053/0.055/0.002 ms



Ping Example 3. Send N packets and stop

Send N packets specified with -c option and then stop. This way the ping command can exit automatically instead of pressing CTRL+C to exit.

In the following example, ping command sends 5 packets, and waits for response from the destination host. Ping will exit after receiving the response or error.

```
$ ping -c 5 google.com

PING google.com (74.125.45.100) 56(84) bytes of data.

64 bytes from yx-in-f100.google.com (74.125.45.100): icmp_seq=1 ttl=44 time=731 ms
```

```
64 bytes from yx-in-f100.google.com (74.125.45.100): icmp_seq=2 ttl=44 time=777

ms 64 bytes from yx-in-f100.google.com (74.125.45.100): icmp_seq=3 ttl=44

time=838 ms 64 bytes from yx-in-f100.google.com (74.125.45.100): icmp_seq=4

ttl=44 time=976 ms 64 bytes from yx-in-f100.google.com (74.125.45.100):

icmp_seq=5 ttl=44 time=1071 ms



--- google.com ping statistics ---

5 packets transmitted, 5 received, 0% packet loss, time 4216ms

rtt min/avg/max/mdev = 731.039/879.129/1071.050/126.625 ms
```

Ping Example 4. Show Version and Exit

Display the current version of ping program using -V option.

```
$ ping -V

ping utility, iputils-sss20071127
```

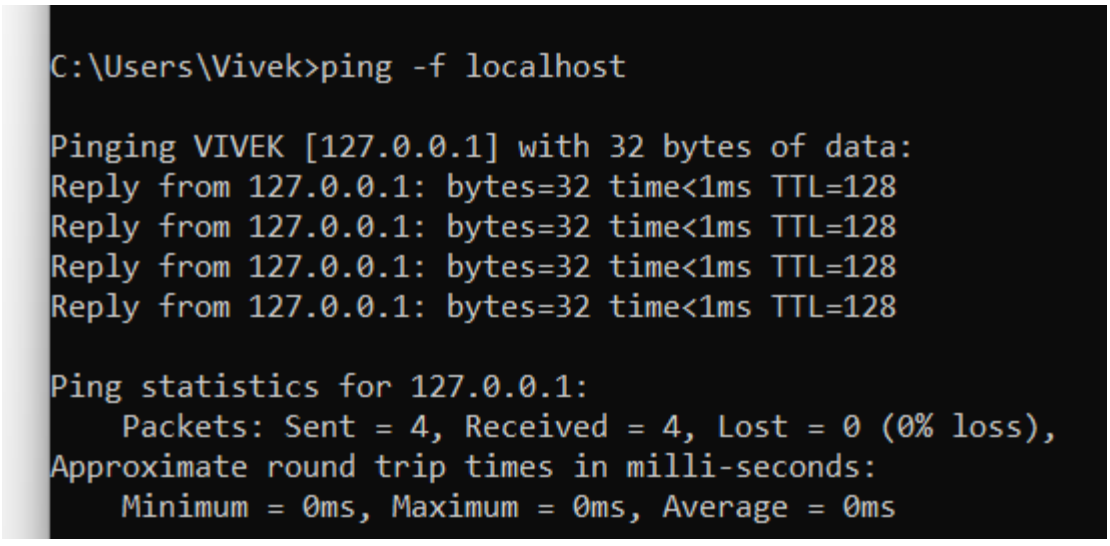
Ping Example 5. Flood the network

Super users can send hundred or more packets per second using -f option. It prints a „.“ when a packet is sent, and a backspace is printed when a packet is received.

As shown below, ping -f has sent more than 400,000 packets in few seconds.

```
# ping -f local host
```

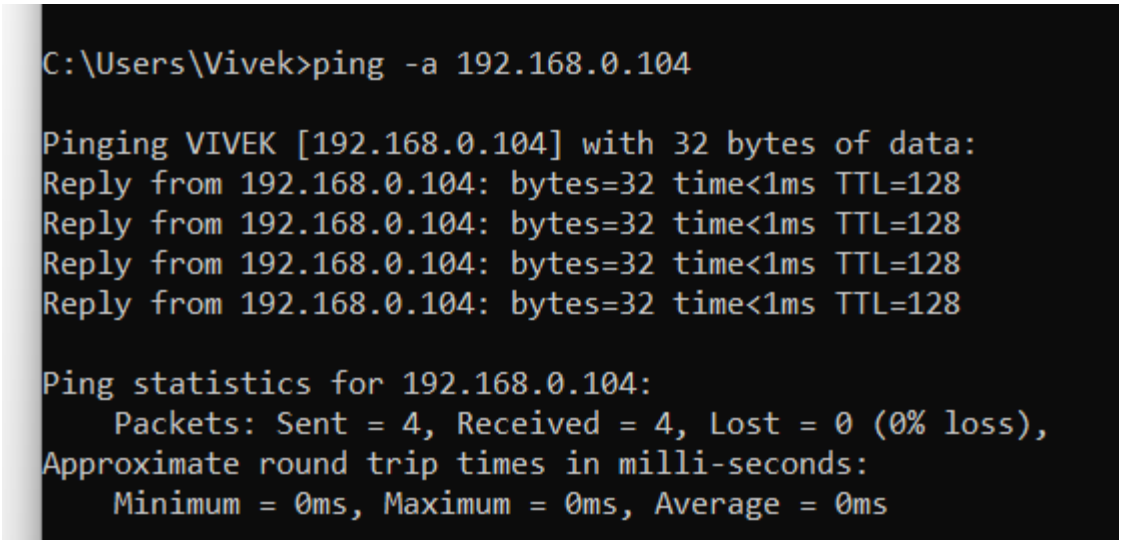
```
PING local host (127.0.0.1) 56(84) bytes of data.

.^C

--- local host ping statistics ---

427412 packets transmitted, 427412 received, 0% packet loss, time 10941ms
rtt min/avg/max/mdev = 0.003/0.004/1.004/0.002 ms, ipg/ewma 0.025/0.004 ms
```



Ping Example 6. Audible ping: Give beep when the peer is reachable

This option is useful for sysadmin during troubleshooting. There is no need for you to look at the ping output after each and every change. You can continue working with your changes, and when the remote machine become reachable you‟ll hear the beep automatically.

$ ping -a IP



Note: It can give beep only from terminal number 1 through 7 and gnome-terminal (It will not work in console).

Ping Example 7. Find out the IP address

You can identify the ip-address using the host name as shown below.

$ ping -c 1 google.com

PING google.com (**74.125.67.100**) 56(84) bytes of data.
64 bytes from gw-in-f100.google.com (74.125.67.100): icmp_seq=1 ttl=43 time=287 ms

--- google.com ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 287.903/287.903/287.903/0.000 ms

Ping Example 8. Print Only Ping Command Summary Statistics

Use option -q to view only the ping statistics summary as shown below.

```
$ ping -c 5 -q 127.0.0.1

PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.



--- 127.0.0.1 ping statistics ---

5 packets transmitted, 5 received, 0% packet loss, time 3998ms

rtt min/avg/max/mdev = 0.047/0.053/0.061/0.009 ms
```

Ping Example 9. Change Ping Packet Size

You can change the packet size of ping command using -s option.

Example: Change the default packet size from 56 to 100.

```
$ ping -s 100 localhost

PING localhost (127.0.0.1) 100(128) bytes of data.

108 bytes from localhost (127.0.0.1): icmp_seq=1 ttl=64 time=0.022 ms

108 bytes from localhost (127.0.0.1): icmp_seq=2 ttl=64 time=0.021 ms

108 bytes from localhost (127.0.0.1): icmp_seq=3 ttl=64 time=0.020 ms

^C

--- localhost ping statistics ---
```

```
3 packets transmitted, 3 received, 0% packet loss, time 1998ms

rtt min/avg/max/mdev = 0.020/0.021/0.022/0.000 ms
```

```
C:\Users\Vivek>ping -s 1 localhost

Pinging VIVEK [127.0.0.1] with 32 bytes of data:
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128

Ping statistics for 127.0.0.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

Ping Packet Size

In the above example, when we set the packet size to 100, it displays "128 bytes" in the output. This is because of the Ping packet header size, which is 28 bytes. So, if you specify the packet size as 100, 28 bytes for the header will be added to it and 128 bytes will be sent.

Ping Bytes Sent = Ping Packet Size + Ping Header Packet Size (28 bytes)

Ping Example 10. Timeout -w

Ping -w option specifies the deadline to terminate the ping output. This specifies the total number of seconds the ping command should send packets to the remote host.

The following example will ping for 5 seconds. i.e. ping command will exit after 5 seconds irrespective of how many packets are sent or received.

```
$ ping -w 5 localhost
```

```
C:\Users\Vivek>ping -w 5 localhost

Pinging VIVEK [::1] with 32 bytes of data:
Reply from ::1: time<1ms
Reply from ::1: time<1ms
Reply from ::1: time<1ms
Reply from ::1: time<1ms

Ping statistics for ::1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

Note: When you specify both -w, and -c, whichever comes first will terminate the ping command.

Ping Example 11. Online ping

Ping from different locations and check the reach ability (availability or time for reaching) of your server from different locations.

If you want to do an online ping, try just ping.

Ping Example 12. Option -w or -c Exits Ping

```
$ ping -c 4 0 -w 2
PING 0 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.064 ms
```

```
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.060 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.058 ms

--- 0 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 1998ms
rtt min/avg/max/mdev = 0.058/0.060/0.064/0.009 ms
$ ping -c 4 0 -w 10
PING 0 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.063 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.060 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.055 ms
64 bytes from 127.0.0.1: icmp_seq=4 ttl=64 time=0.061 ms

--- 0 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 2997ms
rtt min/avg/max/mdev = 0.055/0.059/0.063/0.009 ms
```

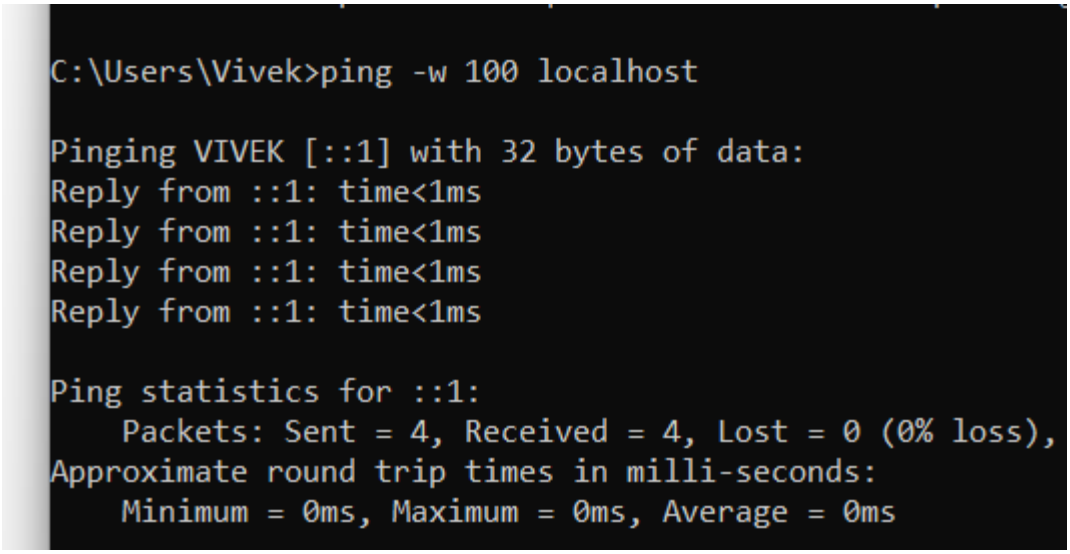
Ping Example 13. Shorter statistics with SIGQUIT

While ping is printing the individual packet status, when you want to view the shorter statistics you can use this technique.

Pressing **CTRL+|** (Control key followed by pipe symbol) for the shows the summary in between, and continues with it packet sending and receiving process.

```
$ ping -w 100 localhost

PING localhost (127.0.0.1) 56(84) bytes of data.

64 bytes from localhost (127.0.0.1): icmp_seq=10 ttl=64 time=0.021 ms

64 bytes from localhost (127.0.0.1): icmp_seq=11 ttl=64 time=0.022 ms

11/11 packets, 0% loss, min/avg/ewma/max = 0.020/0.022/0.022/0.024 ms
64 bytes from localhost (127.0.0.1): icmp_seq=12 ttl=64 time=0.021 ms
64 bytes from localhost (127.0.0.1): icmp_seq=13 ttl=64 time=0.022 ms
```

```
64 bytes from localhost (127.0.0.1): icmp_seq=14 ttl=64 time=0.021 ms
64 bytes from localhost (127.0.0.1): icmp_seq=15 ttl=64 time=0.021 ms
19/19 packets, 0% loss, min/avg/ewma/max = 0.020/0.022/0.022/0.024 ms
64 bytes from localhost (127.0.0.1): icmp_seq=31 ttl=64 time=0.022 ms
64 bytes from localhost (127.0.0.1): icmp_seq=32 ttl=64 time=0.022 ms
32/32 packets, 0% loss, min/avg/ewma/max = 0.020/0.022/0.022/0.027 ms
64 bytes from localhost (127.0.0.1): icmp_seq=33 ttl=64 time=0.023 ms
..
```



Ping Example 14. Specify path for ping to send the packet

You can also specify through which path the ping should send the packet to destination.

```
$ ping hop1 hop2 hop3 .. hopN destination

$ ping 192.168.3.33 192.168.7.1 192.168.4.45
```

Note: If one of the hop in the path is not reachable then you will have failure in pinging.

Ping Example 15. Record and print route of how ECHO_REQUEST sent and ECHO_REPLY received

It records, and prints the network route through which the packet is sent and received. This is useful for network engineers who wish to know how the packet is sent and received.

```
$ ping -R 192.168.1.63

PING 192.168.1.63 (192.168.1.63) 56(84) bytes of data.

64 bytes from 192.168.1.63: icmp_seq=1 ttl=61 time=2.05 ms

RR: 192.168.9.118

 192.168.3.25
```

```
 192.168.10.35

 192.168.1.26

 192.168.1.63

 192.168.1.63

 192.168.10.4

 192.168.3.10

 192.168.4.25

64 bytes from 192.168.1.63: icmp_seq=2 ttl=61 time=2.00 ms (same route)
```

Conclusion:

Hence, we executed and implemented IP networking and Network command : ping.