Simulation of Ping and Traceroute commands using Twisted Python

Ex. No. 2
Date:

PROBLEM STATEMENT:

To simulate PING and TRACEROUTE commands using Twisted Python.

PROBLEM DESCRIPTION:

a) PING

- Ping stands for **Packet Internet** or **Inter-Network Groper**.
- Relies on the Internet Control Message Protocol (ICMP) at the internet layer of TCP/IP.
- It’s most basic use is to confirm network connectivity between two hosts.
- The Ping application should send ICMP Echo Request packetsto a specified target host or IP address and measure the round-trip time (RTT) for each packet. The application should display the RTT for each packet received, as well as the overall statistics, includingpacket loss percentage and average RTT.

b) TRACEROUTE

- A Traceroute command is a command line tool that is generallyused to locate the destination path from the host in the network.
- This command is useful when you want to know about the route and about all the hops that a packet takes.
- It is used in tracing and troubleshooting network problems.
- The Traceroute application should send UDP packets with increasing TTL values to a specified target host or IP address and print out the intermediate hops along the path. The application should display the IP address or hostname of each intermediate hop and the round-trip time (RTT) for each hop.

CODE:

a) PING – TWISTED PYTHON

```
import subprocess
from twisted.internet import reactor, defer

class PingProtocol:
    def_init_(self):
        self.deferred = defer.Deferred()

    def ping(self, host):
        process = subprocess.Popen(['ping', '-c', '4', host], stdout=subprocess.PIPE)
        output, error = process.communicate()
        if error:
            self.deferred.errback(error)
        else:
            self.deferred.callback(output)

def print_result(result):
    print(result.decode())

def print_error(failure):
    print(failure)

protocol = PingProtocol()
protocol.ping('google.com')
protocol.deferred.addCallbacks(print_result, print_error)
reactor.run()
```

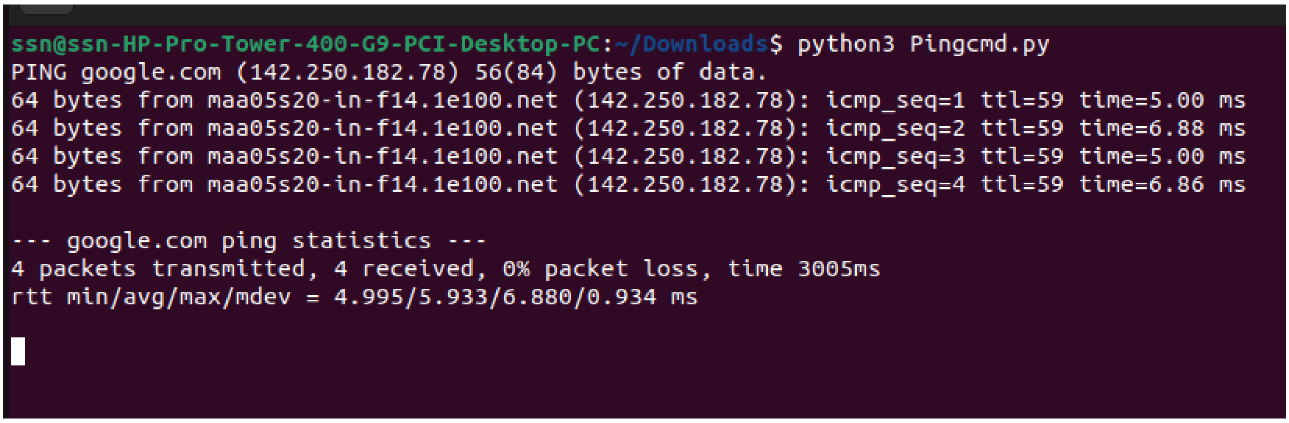
b) TRACE ROUTE - TWISTED PYTHON

```
import subprocess
from twisted.internet import reactor, defer

class TracerouteCmd:
    def_init_(self):
        self.deferred = defer.Deferred()

    def traceroute(self, host):
        process = subprocess.Popen(['traceroute', '-m', '10', host], stdout=subprocess.PIPE)
        output, error = process.communicate()
        if error:
            self.deferred.errback(error)
        else:
            self.deferred.callback(output)

def print_result(result):
    print(result.decode())

def print_error(failure):
    print(failure)

protocol = TracerouteCmd()
protocol.traceroute('google.com')
protocol.deferred.addCallbacks(print_result, print_error)
reactor.run()
```

SAMPLE INPUT AND OUTPUT:

a) **INPUT :** 'google.com' (with packet limit as 4 using -c)

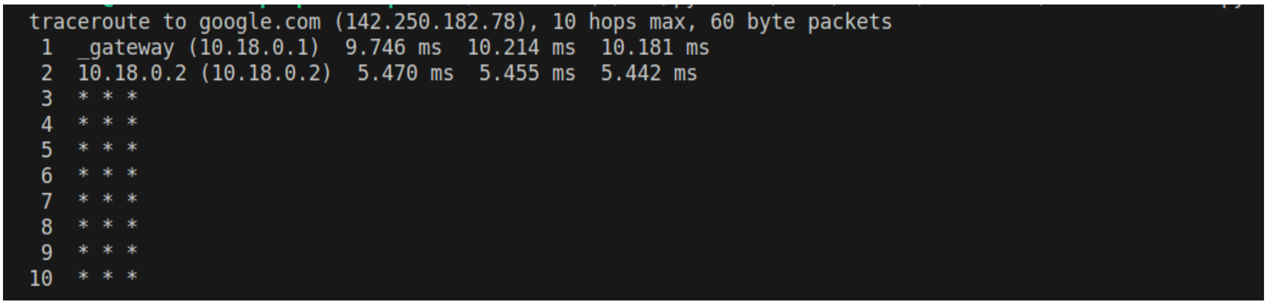
OUTPUT :



```
ssn@ssn-HP-Pro-Tower-400-G9-PCI-Desktop-PC:~/Downloads$ python3 Pingcmd.py
PING google.com (142.250.182.78) 56(84) bytes of data.
64 bytes from maa05s20-in-f14.1e100.net (142.250.182.78): icmp_seq=1 ttl=59 time=5.00 ms
64 bytes from maa05s20-in-f14.1e100.net (142.250.182.78): icmp_seq=2 ttl=59 time=6.88 ms
64 bytes from maa05s20-in-f14.1e100.net (142.250.182.78): icmp_seq=3 ttl=59 time=5.00 ms
64 bytes from maa05s20-in-f14.1e100.net (142.250.182.78): icmp_seq=4 ttl=59 time=6.86 ms

--- google.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 4.995/5.933/6.880/0.934 ms
```

b) **INPUT:** 'google.com' (with packet limit as 10)

OUTPUT :



```
traceroute to google.com (142.250.182.78), 10 hops max, 60 byte packets
 1  _gateway (10.18.0.1)  9.746 ms  10.214 ms  10.181 ms
 2  10.18.0.2 (10.18.0.2)  5.470 ms  5.455 ms  5.442 ms
 3  * * *
 4  * * *
 5  * * *
 6  * * *
 7  * * *
 8  * * *
 9  * * *
10  * * *
```

RESULT

Hence, the ping and traceroute commands are implemented using twisted python and verified the output successfully.