# 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:

1. **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 packets to 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, including packet loss percentage and average RTT.

# TRACEROUTE

* + A Traceroute command is a command line tool that is generally used 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:

1. **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()

# 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())

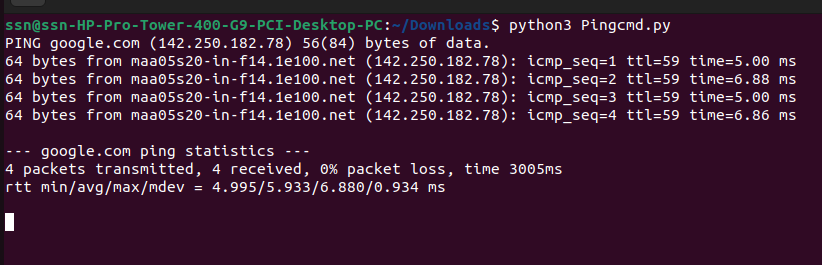
def print\_error(failure): print(failure)

protocol = TracerouteCmd() protocol.traceroute('google.com') protocol.deferred.addCallbacks(print\_result, print\_error) reactor.run()

# SAMPLE INPUT AND OUTPUT:

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

## OUTPUT :



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

## OUTPUT :

**RESULT**

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