**PRACTICAL SUBMISSION RECORD- A.Y. 2024-25**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Class: FY MCA**  **Semester: II** | **Div: A** | **Course Code: MCA10552**  **Course Name: Laboratory Practice - II** | | **Batch: F2** |
| **Name: Pranav Raju Malwatkar** | | | **Roll No: 51037** |  |
| **CO No: CO517C.5** | | | **Assignment No: 9** |  |

**Title:** Study of any network simulation tools - To create a network with three nodes and establish a TCP connection between node 0 and node 1 such that node 0 will send TCP to node 2 via node 1 Guve simpler and shortest code possible.

**Code:**

import socket

import threading

# Node 0 sends a message to Node 1

def node\_0():

with socket.socket(socket.AF\_INET, socket.SOCK\_STREAM) as sock:

sock.connect(('localhost', 9999))

sock.sendall(b"Message from Node 0")

# Node 1 listens, then forwards to Node 2

def node\_1():

with socket.socket(socket.AF\_INET, socket.SOCK\_STREAM) as server\_sock:

server\_sock.bind(('localhost', 9999))

server\_sock.listen(1)

conn, \_ = server\_sock.accept()

message = conn.recv(1024)

conn.close()

with socket.socket(socket.AF\_INET, socket.SOCK\_STREAM) as sock:

sock.connect(('localhost', 8888))

sock.sendall(message)

# Node 2 listens for message

def node\_2():

with socket.socket(socket.AF\_INET, socket.SOCK\_STREAM) as server\_sock:

server\_sock.bind(('localhost', 8888))

server\_sock.listen(1)

conn, \_ = server\_sock.accept()

message = conn.recv(1024)

print(f"Node 2 received: {message.decode()}")

conn.close()

# Run nodes in parallel using threading

def run\_nodes():

threading.Thread(target=node\_2).start()

threading.Thread(target=node\_1).start()

threading.Thread(target=node\_0).start()

run\_nodes()

**Output:**

Node 1 listening...

Node 2 listening...

Node 1 received: Message from Node 0

Node 2 received: Message from Node 0