Experiment - 12

Aim:
To implement echo client server using TCP/UDP Search

Algorithm:
Server - py:
-> Create a UDP socket
-> Bind the socket to specific IP address (127.0.0.1) & port (12345)
-> Continuous section for running message
-> when message received - decode it
-> Repeat infinitely.
Client - py
-> Create UDP socket
-> Set a timeout for socket to avoid waiting
-> If no response received in period, print timeout message
-> close socket after sending message.

Code:
server - py
import socket
def start_server(host='127.0.0.1', port=12345):
with socket.socket(socket.AF_INET, socket.SOCK_DGRAM) as s:
s.bind((host, port))
print(f"UDP Server running on {host}:{port}")

while True:
data, addr = s.recvfrom(1024)
print(f"Received message from {addr}: {data.decode()}")

if __name__ == "__main__":
start_server()

client.py
def ping_server(host='127.0.0.1', port=12345):
with socket.socket(socket.AF_INET, socket.SOCK_DGRAM) as s:

s.settimeout(5)
try:
s.sendto(b"Hello", (host, port))
print("message sent to server")
except socket.timeout:
print("Request timed out")

if __name__ == "__main__":
ping_server()

O/P:
Server.py

Terminal
> Python Server.py
>>

UDP server running on 127.0.0.1:12369

client.py

Terminal

> python client.py

>>

message sent to server

Server terminal:

Recieved message from ('127.0.0.1', 56003):
Hello

Result:

Thus the program of echo client server using UDP sockets has been implemented & executed successfully.