

Exp No: 12a
22/9/23

Practical - 12.4

End-to-End communication at transport layer

Aim: a) Implement echo client server using TCP/UDP sockets

Algorithm:

import socket

import time

def ping_server(host='127.0.0.1', port=10345):
 with socket.socket(socket.AF_INET, socket.SOCK_DGRAM) as s:

try:

s.sendto(b'Hello', (host, port))

except S.timeout:

print("Request timed out")

ping_server()

The results show all the entries in the

practical. The message "Hello" was successfully sent to the client/queried server. Below are the results obtained.

UML

10/10/23

22/9/25

Implement chat client over using TCP/UDP

b) Aim: Implement chat client server using TCP/UDP
new sockets was very interesting

Algorithm:

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)) with message

(event): print(f"UDP server running on {host} {port}")

while True: time - sleep 0.1

data, addr = s.recvfrom(1024)

print(f"Perceived Message from {addr}: {data.decode('utf-8')})")

start_server() time.sleep 0.1

output: (socket) address: 127.0.0.1:12345

UDP server running on 127.0.0.1:12345

Received message from (127.0.0.1:12345): Hello

(broadcast - broadcast)

time, sleep 0.1

("no broadcast") time

time, sleep 0.1

print("Hello - " + str(addr[1]))

time, sleep 0.1

Result:

(socket, ('127.0.0.1', 12345))

This implementation of chat client server

using TCP/UDP socket performed successfully.

Using TCP/UDP socket performed successfully.

VirtualBox

Windows machine running ping and telnet