

server.py

on server.py

running on 127.0.0.1: 1234

client.py

client.py

to server.

ad:

message from ('127.0.0.1',
1110)

program of echo

using UDP

is implemented

successfully

Exp: 13

Ping Program

Date: 11/10/24

Aim:

To implement your own ping program.

Algorithm:

UDP server:

- create UDP socket & bind it to a specific address & port.
- wait for message.
- print message & client address.
- send back ping to client.

UDP client:

- create UDP socket & set a 2 Sec timeout.
- send 'ping' to server.
- If a response (ping received - print response & calculate RTT)
- If no response within 2 Sec print request time out.

Code : server.py

```
import socket
def start_server(host='127.0.0.1', port=8080):
    with socket.socket(socket.AF_INET,
        socket.SOCK_STREAM) 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()}")
            s.sendto(b'pong', addr)
if __name__ == "__main__":
    start_server()
```

client.py

```
import time
import socket
def ping_server(host='127.0.0.1', port=12345):
    with socket.socket(socket.AF_INET,
        socket.SOCK_DGRAM) as s:
        try:
            s.settimeout(2)
            start = time.time()
            s.sendto(b'ping', (host, port))
            data, addr = s.recvfrom(1024)
            end = time.time()
```


xy

```
(host='127.0.0.1', port=1024)
sock = socket.AF_INET,
) as s:
    s.bind((host, port))
    s.listen(1)
    print 'Server running'
    while True:
        conn, addr = s.accept()
        print 'Received connection from', addr
        data = conn.recv(1024)
        if not data:
            continue
        print 'Received message from', addr
        conn.send('pong')
```

```
= s.recv(1024)
received message from (127.0.0.1, 50001): ping
main--"
```

127.0.0.1',

```
[socket.AF_INET,
) as s:
```

```
(host, port)
from (1024)
```

```
print # If 'received {data-decode()}'
from {addr} in {end-start}.2f seconds"
except socket.timeout:
    print "Request timed out"
if __name__ == '__main__':
    ping_server()
```

output:

Terminal
 > python server.py
 UDP server running
 on 127.0.0.1 : 12345
 Received message from
 '127.0.0.1', 50001: ping

Terminal
 > python client.py
 Received ping
 from (127.0.0.1,
 12345) in 0.00
 seconds

Result:

Thus a ping program has been
 executed successfully.

[Signature]