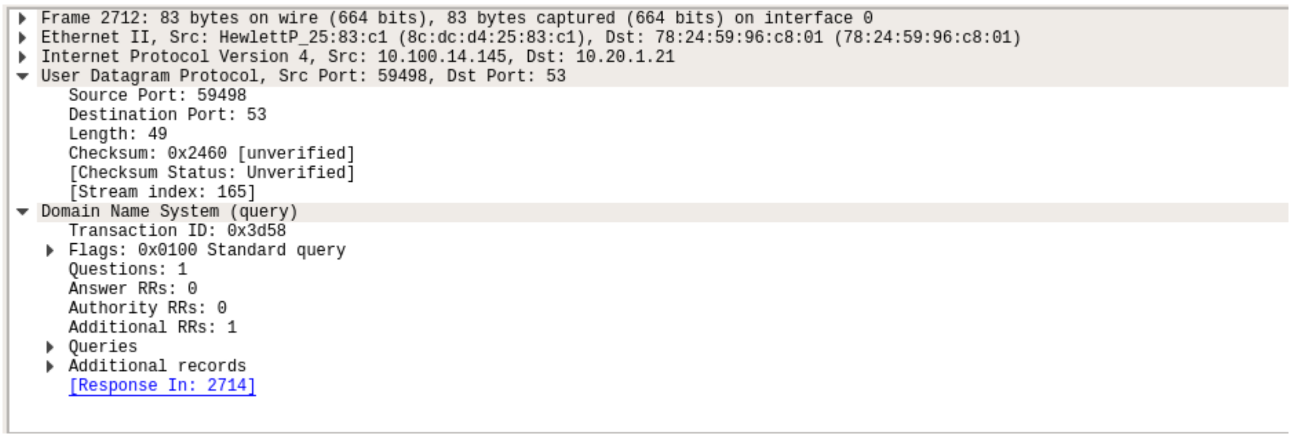
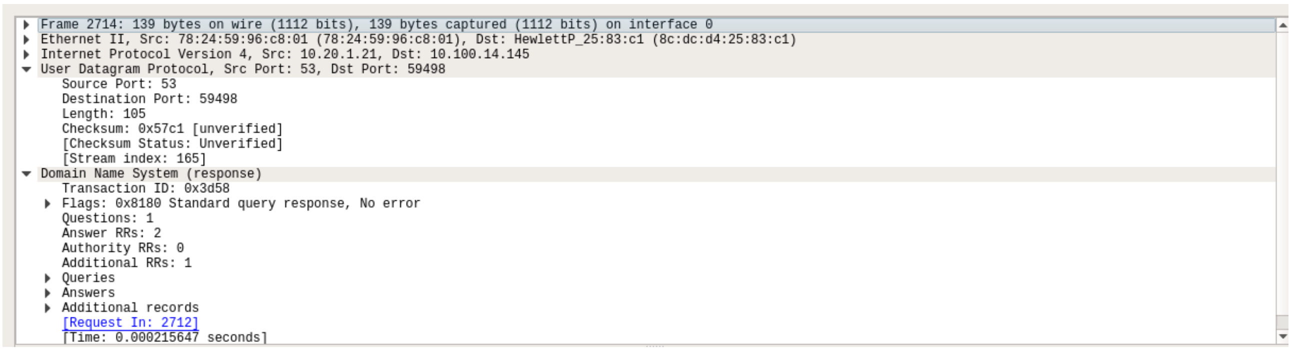


```
▸ Frame 2712: 83 bytes on wire (664 bits), 83 bytes captured (664 bits) on interface 0
▸ Ethernet II, Src: HewlettP_25:83:c1 (8c:dc:d4:25:83:c1), Dst: 78:24:59:96:c8:01 (78:24:59:96:c8:01)
▸ Internet Protocol Version 4, Src: 10.100.14.145, Dst: 10.20.1.21
▾ User Datagram Protocol, Src Port: 59498, Dst Port: 53
    Source Port: 59498
    Destination Port: 53
    Length: 49
    Checksum: 0x2460 [unverified]
    [Checksum Status: Unverified]
    [Stream index: 165]
▾ Domain Name System (query)
    Transaction ID: 0x3d58
  ▸ Flags: 0x0100 Standard query
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 1
  ▸ Queries
  ▸ Additional records
    [Response In: 2714]
```



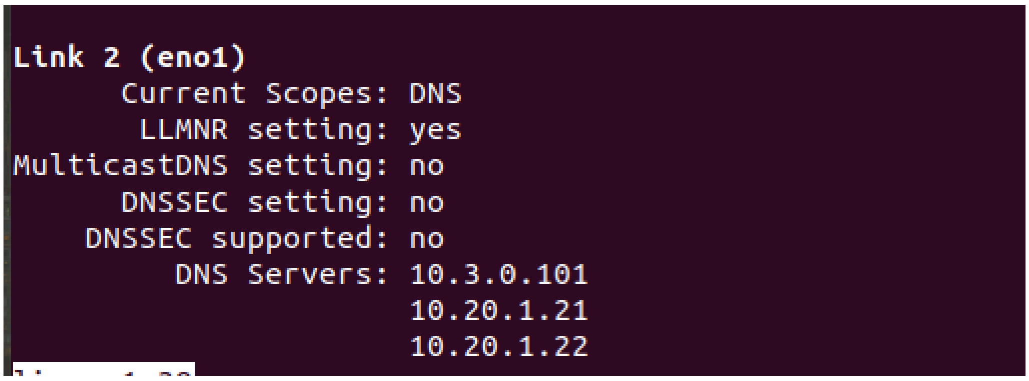
```
▸ Frame 2714: 139 bytes on wire (1112 bits), 139 bytes captured (1112 bits) on interface 0
▸ Ethernet II, Src: 78:24:59:96:c8:01 (78:24:59:96:c8:01), Dst: HewlettP_25:83:c1 (8c:dc:d4:25:83:c1)
▸ Internet Protocol Version 4, Src: 10.20.1.21, Dst: 10.100.14.145
▾ User Datagram Protocol, Src Port: 53, Dst Port: 59498
    Source Port: 53
    Destination Port: 59498
    Length: 105
    Checksum: 0x57c1 [unverified]
    [Checksum Status: Unverified]
    [Stream index: 165]
▾ Domain Name System (response)
    Transaction ID: 0x3d58
  ▸ Flags: 0x8180 Standard query response, No error
    Questions: 1
    Answer RRs: 2
    Authority RRs: 0
    Additional RRs: 1
  ▸ Queries
  ▸ Answers
  ▸ Additional records
    [Request In: 2712]
    [Time: 0.000215647 seconds]
```

1.The DNS query uses UDP

2. client->server: src is ephimeral and dest is 53

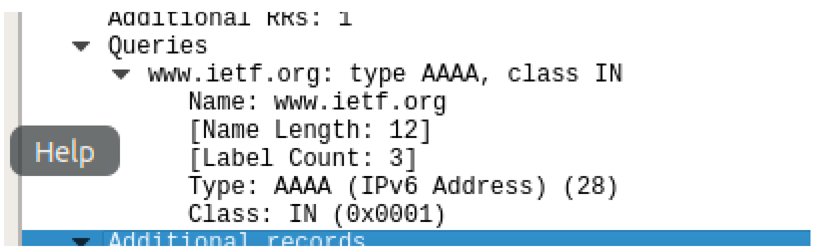
   server->client : src port is 53 and dest port is an ephimeral port

3. it is sent to 10.20.1.21



```
Link 2 (eno1)
      Current Scopes: DNS
       LLMNR setting: yes
MulticastDNS setting: no
      DNSSEC setting: no
    DNSSEC supported: no
         DNS Servers: 10.3.0.101
                      10.20.1.21
                      10.20.1.22
```

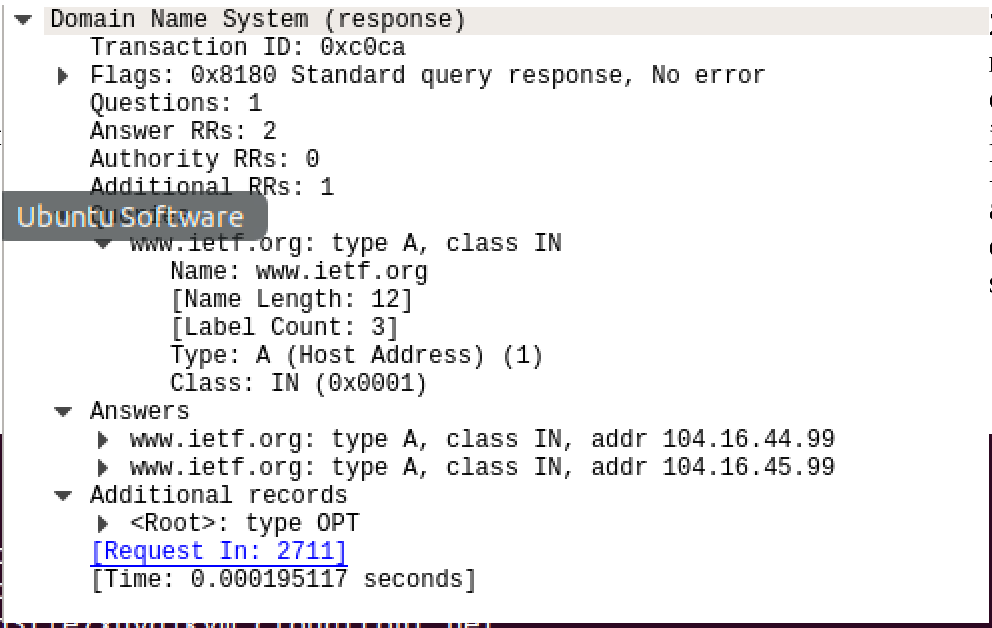
4. It queries for AAAA ip address. The quert message has no answers



```
    Additional RRs: 1
  ▾ Queries
    ▾ www.ietf.org: type AAAA, class IN
        Name: www.ietf.org
        [Name Length: 12]
        [Label Count: 3]
        Type: AAAA (IPv6 Address) (28)
        Class: IN (0x0001)
  ▾ Additional records
```

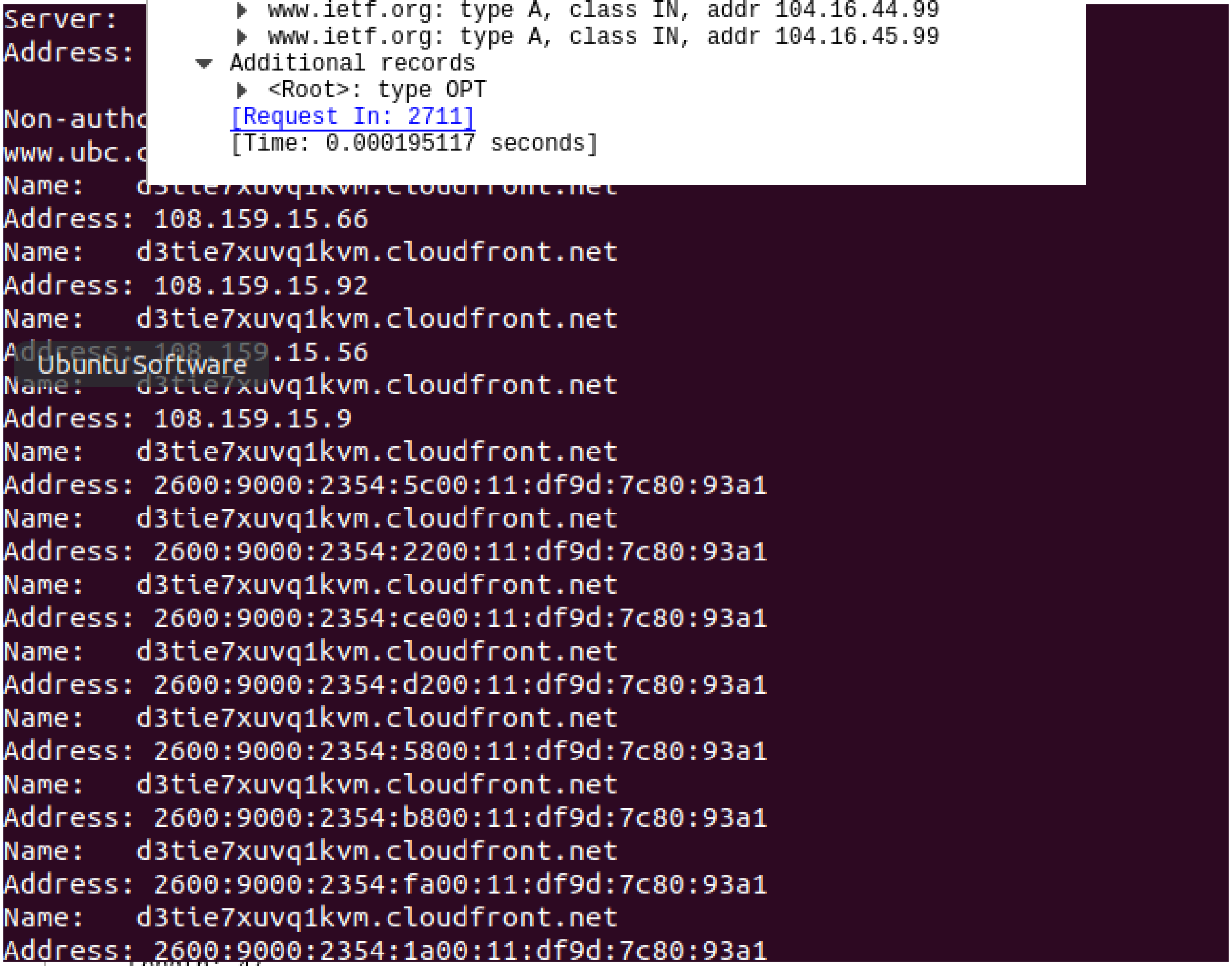
5. Contains
both IP
6. No as
get the IP it
the local
FILE to
subsequent
DNS

2 answers,
mapping
once we
is stored in
HOST
avoid
queries to
server



```
▼ Domain Name System (response)
    Transaction ID: 0xc0ca
  ▶ Flags: 0x8180 Standard query response, No error
    Questions: 1
    Answer RRs: 2
    Authority RRs: 0
    Additional RRs: 1
      www.ietf.org: type A, class IN
        Name: www.ietf.org
        [Name Length: 12]
        [Label Count: 3]
        Type: A (Host Address) (1)
        Class: IN (0x0001)
  ▼ Answers
    ▶ www.ietf.org: type A, class IN, addr 104.16.44.99
    ▶ www.ietf.org: type A, class IN, addr 104.16.45.99
  ▼ Additional records
    ▶ <Root>: type OPT
    [Request In: 2711]
    [Time: 0.000195117 seconds]
```

Q2.

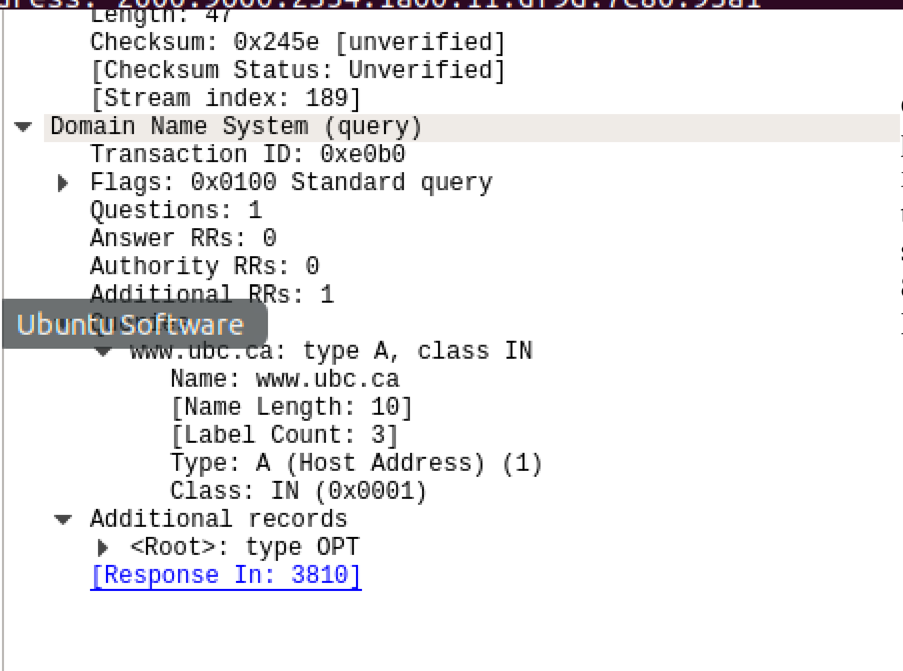


```
Server:
Address:

Non-autho
www.ubc.c
Name:
Address: 108.159.15.66
Name:    d3tie7xuvq1kvm.cloudfront.net
Address: 108.159.15.92
Name:    d3tie7xuvq1kvm.cloudfront.net
Address: 108.159.15.56
Name:    d3tie7xuvq1kvm.cloudfront.net
Address: 108.159.15.9
Name:    d3tie7xuvq1kvm.cloudfront.net
Address: 2600:9000:2354:5c00:11:df9d:7c80:93a1
Name:    d3tie7xuvq1kvm.cloudfront.net
Address: 2600:9000:2354:2200:11:df9d:7c80:93a1
Name:    d3tie7xuvq1kvm.cloudfront.net
Address: 2600:9000:2354:ce00:11:df9d:7c80:93a1
Name:    d3tie7xuvq1kvm.cloudfront.net
Address: 2600:9000:2354:d200:11:df9d:7c80:93a1
Name:    d3tie7xuvq1kvm.cloudfront.net
Address: 2600:9000:2354:5800:11:df9d:7c80:93a1
Name:    d3tie7xuvq1kvm.cloudfront.net
Address: 2600:9000:2354:b800:11:df9d:7c80:93a1
Name:    d3tie7xuvq1kvm.cloudfront.net
Address: 2600:9000:2354:fa00:11:df9d:7c80:93a1
Name:    d3tie7xuvq1kvm.cloudfront.net
Address: 2600:9000:2354:1a00:11:df9d:7c80:93a1
```

1.
src
2.
is

3.

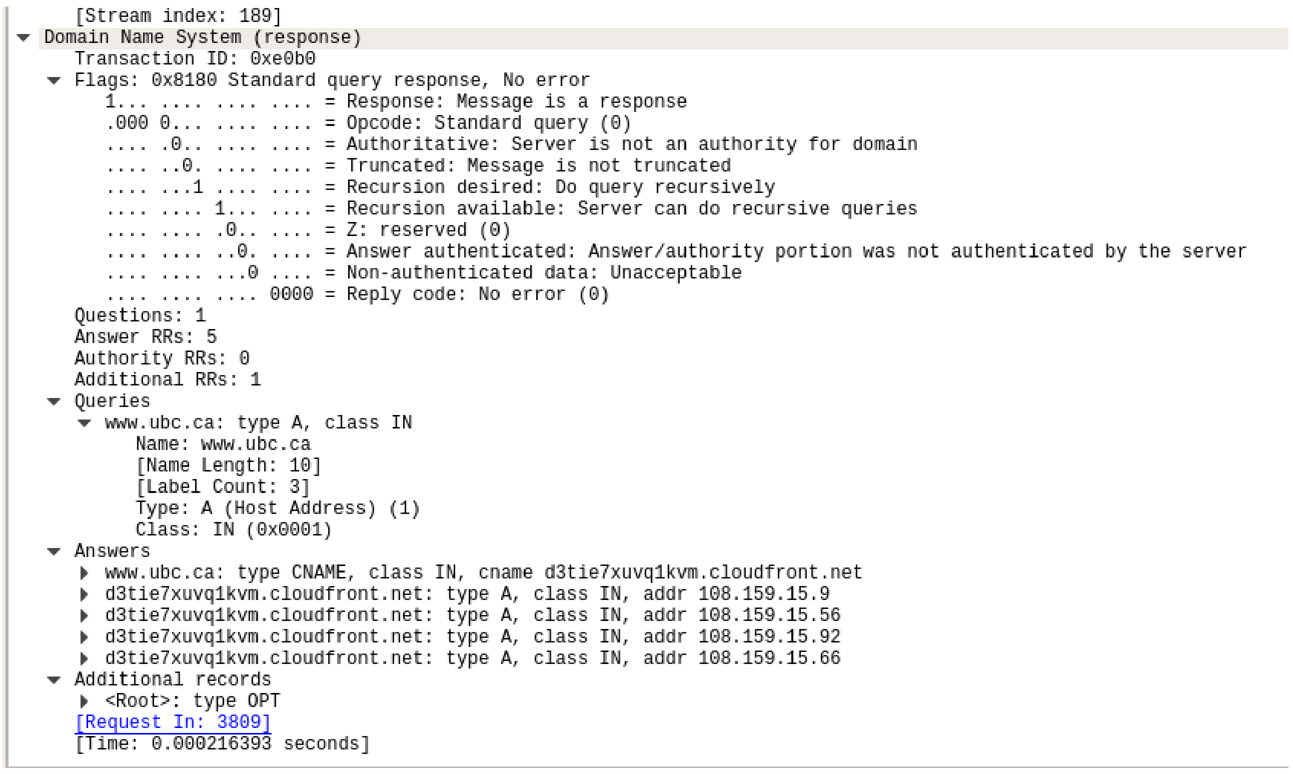


```
    Length: 47
    Checksum: 0x245e [unverified]
    [Checksum Status: Unverified]
    [Stream index: 189]
▼ Domain Name System (query)
    Transaction ID: 0xe0b0
  ▶ Flags: 0x0100 Standard query
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 1
      www.ubc.ca: type A, class IN
        Name: www.ubc.ca
        [Name Length: 10]
        [Label Count: 3]
        Type: A (Host Address) (1)
        Class: IN (0x0001)
  ▼ Additional records
    ▶ <Root>: type OPT
    [Response In: 3810]
```

destination port is 53 and
port is an ephimeral port
it is sent to 10.20.1.21. It
the ip address of my DNS
server (not something like
8.8.8.8 which is google’s
DNS server)

type-A query. Recursion desired query.

4.



```
    [Stream index: 189]
▾ Domain Name System (response)
    Transaction ID: 0xe0b0
  ▾ Flags: 0x8180 Standard query response, No error
      1... .... .... .... = Response: Message is a response
      .000 0... .... .... = Opcode: Standard query (0)
      .... .0.. .... .... = Authoritative: Server is not an authority for domain
      .... ..0. .... .... = Truncated: Message is not truncated
      .... ...1 .... .... = Recursion desired: Do query recursively
      .... .... 1... .... = Recursion available: Server can do recursive queries
      .... .... .0.. .... = Z: reserved (0)
      .... .... ..0. .... = Answer authenticated: Answer/authority portion was not authenticated by the server
      .... .... ...0 .... = Non-authenticated data: Unacceptable
      .... .... .... 0000 = Reply code: No error (0)
    Questions: 1
    Answer RRs: 5
    Authority RRs: 0
    Additional RRs: 1
  ▾ Queries
    ▾ www.ubc.ca: type A, class IN
        Name: www.ubc.ca
        [Name Length: 10]
        [Label Count: 3]
        Type: A (Host Address) (1)
        Class: IN (0x0001)
  ▾ Answers
    ▸ www.ubc.ca: type CNAME, class IN, cname d3tie7xuvq1kvm.cloudfront.net
    ▸ d3tie7xuvq1kvm.cloudfront.net: type A, class IN, addr 108.159.15.9
    ▸ d3tie7xuvq1kvm.cloudfront.net: type A, class IN, addr 108.159.15.56
    ▸ d3tie7xuvq1kvm.cloudfront.net: type A, class IN, addr 108.159.15.92
    ▸ d3tie7xuvq1kvm.cloudfront.net: type A, class IN, addr 108.159.15.66
  ▾ Additional records
    ▸ <Root>: type OPT
    [Request In: 3809]
    [Time: 0.000216393 seconds]
```

5 answers are provided. Each of them give a IP to the domain. Load balencing in action.

Server

```
import socket
import threading
import sys
# Server setup
HOST = '127.0.0.1'
PORT = 5000
clients = []
def broadcast(message, sender_socket):
    for client in clients:
        if client != sender_socket:
            try:
                client.send(message)
            except:
                # If a client fails, remove it
                client.close()
                clients.remove(client)
def handle_client(client_socket, address):
    print("New connection from %s:%s" % (address[0], address[1]))
    clients.append(client_socket)

    while True:
        try:
            message = client_socket.recv(1024)
            if message:
                print("Received from client %s:%s: %s" % (address[0], address[1], message.strip()))
                broadcast(message, client_socket)
            else:
                # Client disconnected
                client_socket.close()
                clients.remove(client_socket)
                print("Client %s:%s disconnected." % (address[0], address[1]))
                break
        except:
            client_socket.close()
            clients.remove(client_socket)
            break
def main():
    server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    server.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
    server.bind((HOST, PORT))
    server.listen(5)
    print("Server started on %s:%s" % (HOST, PORT))

    while True:
        client_socket, address = server.accept()
        thread = threading.Thread(target=handle_client, args=(client_socket, address))
        thread.daemon = True
        thread.start()
if __name__ == "__main__":
    main()
```

Client

```
import socket
import threading
import sys
# Client setup
HOST = '127.0.0.1'
PORT = 5000
def receive_messages(sock):
    while True:
        try:
            data = sock.recv(1024)
            if data:
                sys.stdout.write("\n" + data.decode('utf-8') + "\n>> ")
                sys.stdout.flush()
            else:
                break
        except:
            print("Disconnected from server.")
            break
def send_messages(sock):
    while True:
        # Change this line
        message = input(">> ")  # Use input() instead of raw_input()
        try:
            sock.send(message.encode('utf-8'))
        except:
            print("Failed to send message.")
            break
def main():
    try:
        sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        sock.connect((HOST, PORT))
        print("Connected to server on %s:%s" % (HOST, PORT))

        # Threads for receiving and sending
        receive_thread = threading.Thread(target=receive_messages, args=(sock,))
        receive_thread.daemon = True
        receive_thread.start()

        send_thread = threading.Thread(target=send_messages, args=(sock,))
        send_thread.daemon = True
        send_thread.start()

        # Keep the main thread alive
        while True:
            pass

    except Exception as e:
        print("Could not connect to server:", e)
        sys.exit(1)
if __name__ == "__main__":
    main()
```