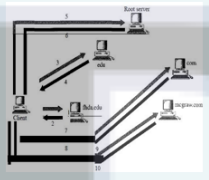
COMPUTER NETWORK LAB CAT

VIT LMS Home Dashboard My courses P2

Back

Question 1
Not yet answered
Marked out of 1.00
Flag question

Send Website name to the DNS server and get back IP address using TCP Sockets. Assume Iterative resolution for DNS protocol using socket programming. Consider the following design with 2 DNS servers.



Sample Input - Output:

Client 1	Server 1	Server 2
Input : VIT.COM Request Server 1 to send IP address of VIT.COM	Print: Not Available	
Request Server 2 to send IP address of VIT.COM		Print: Website : VIT.COM IP Address : 10.10.1.10

Answer:

Quiz navigation

P2

PURVA SHARMA 21BCE0169

1 2

Finish attempt ...

https://moovit.vit.ac.in/mv/

CODE:

21BCE0169_server1.py | 21BCE0169_server2.py | 21BCE0169_client1.py

21BCE0169_server1.py > ...

```python
import socket
import threading
import time

# Global variables
HOST = 'localhost'
PORT = 5000
EXPIRY_TIME = 200  # seconds

# Thread function to handle each client connection
def handle_client(client_socket, client_address):
    # Get the client's acknowledgement number
    ack_num = client_socket.recv(1024).decode()
    print(f"[NEW CONNECTION] Client {client_address} connected with ACK number: {ack_num}")

    # Receive and send messages until expiry time is reached
    start_time = time.time()
    while (time.time() - start_time) <= EXPIRY_TIME:
        # Receive message from client
        message = client_socket.recv(1024).decode()
        if not message:
            break

        # Print the received message
        print(f"[Not Available] Client {client_address}: {message}")

        # Send acknowledgement back to the client
        client_socket.send(ack_num.encode())

    # Close the connection
    client_socket.close()
    print(f"[CONNECTION CLOSED] Client {client_address} disconnected")

```

```python
# Main function to handle incoming client connections
def
    (variable) server_socket: socket
    server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

    # Bind the socket to the host and port
    server_socket.bind((HOST, PORT))

    # Listen for incoming connections
    server_socket.listen(5)
    print("[LISTENING] Server is listening for incoming connections...")

    # Accept incoming connections and create a new thread for each client
    while True:
        client_socket, client_address = server_socket.accept()
        client_thread = threading.Thread(target=handle_client, args=(client_socket, client_address))
        client_thread.start()

# Run the server
if __name__ == '__main__':
    main()
```

21BCE0169_server1.py × 21BCE0169_server2.py × 21BCE0169_client1.py

21BCE0169_server2.py > ...

```python
import socket
import threading
import time

# Global variables
HOST = 'localhost'
PORT = 5500
EXPIRY_TIME = 200  # seconds

# Thread function to handle each client connection
def handle_client(client_socket, client_address):
    # Get the client's acknowledgement number
    ack_num = client_socket.recv(1024).decode()
    print(f"[NEW CONNECTION] Client {client_address} connected with ACK number: {ack_num}")

    # Receive and send messages until expiry time is reached
    start_time = time.time()
    while (time.time() - start_time) <= EXPIRY_TIME:
        # Receive message from client
        message = client_socket.recv(1024).decode()
        if not message:
            break

        # Print the received message
        print(f"[Website : VIT.COM , IP Address : 10.10.1.10] Client {client_address}: {message}")

        # Send acknowledgement back to the client
        client_socket.send(ack_num.encode())

    # Close the connection
    client_socket.close()
    print(f"[CONNECTION CLOSED] Client {client_address} disconnected")
```

```python
# Main function to handle incoming client connections
def main():
    # Create a socket
    server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

    # Bind the socket to the host and port
    server_socket.bind((HOST, PORT))

    # Listen for incoming connections
    server_socket.listen(5)
    print("[LISTENING] Server is listening for incoming connections...")

    # Accept incoming connections and create a new thread for each client
    while True:
        client_socket, client_address = server_socket.accept()
        client_thread = threading.Thread(target=handle_client, args=(client_socket, client_address))
        client_thread.start()

# Run the server
if __name__ == '__main__':
    main()
```

21BCE0169_server1.py | 21BCE0169_server2.py | 21BCE0169_client1.py

21BCE0169_client1.py > ...

```python
import socket
import threading

# Global variables
HOST = 'localhost'
PORT = 5500


# Thread function to handle receiving messages from the server
def receive_messages(client_socket):
    while True:
        # Receive message from server
        message = client_socket.recv(1024).decode()
        if not message:
            break

        # Print the received message
        print(f"[MESSAGE RECEIVED] Server: {message}")



# Main function to handle client operations
def main():
    # Create a socket
    client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

    # Connect to the server
    client_socket.connect((HOST, PORT))



    # Send the acknowledgement number to the server
    ack_num = input("Enter your acknowledgement number (0 or 1): ")
    client_socket.send(ack_num.encode())

    # Create a separate thread to receive messages from the server
    receive_thread = threading.Thread(target=receive_messages, args=(client_socket,))
    receive_thread.start()

    # Send messages to the server
    while True:
        message = input()
        if message.lower() == 'exit':
            break

        # Send message to the server
        client_socket.send(message.encode())

    # Close the connection
    client_socket.close()

# Run the client
if __name__ == '__main__':
    main()
```

INPUT CLIENT 1 :

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

(base) matlab@sjt516scope029:~/Downloads$ /bin/python3.9 /home/matlab/Downloads/21BCE0169_client1.py
Enter your acknowledgement number (0 or 1): 0
VIT.COM Request Server 1 to send IP address of VIT.COM
[MESSAGE RECEIVED] Server: 0
Enter your acknowledgement number (0 or 1): 1
Request Server 2 to send IP address of VIT.COM
[MESSAGE RECEIVED] Server: 1
```


SERVER 1 OUTPUT:

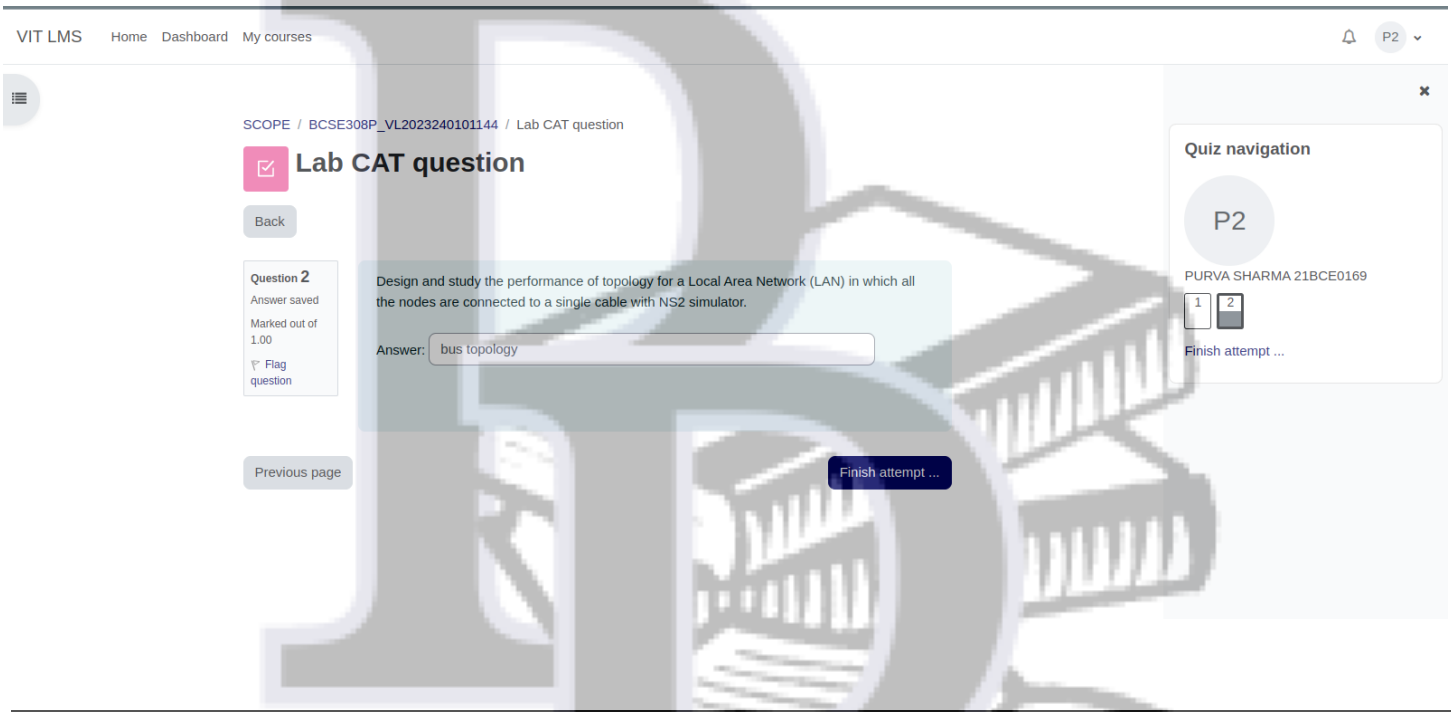
```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

/bin/python3.9 /home/matlab/Downloads/21BCE0169_server1.py
(base) matlab@sjt516scope029:~/Downloads$ /bin/python3.9 /home/matlab/Downloads/21BCE0169_server1.py
[LISTENING] Server is listening for incoming connections...
[NEW CONNECTION] Client ('127.0.0.1', 35018) connected with ACK number: 0
[Not Available] Client ('127.0.0.1', 35018): VIT.COM Request Server 1 to send IP address of VIT.COM
```

SERVER 2 OUTPUT:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

(base) matlab@sjt516scope029:~/Downloads$ /bin/python3.9 /home/matlab/Downloads/21BCE0169_server2.py
[LISTENING] Server is listening for incoming connections...
[NEW CONNECTION] Client ('127.0.0.1', 40434) connected with ACK number: 1
[Website : VIT.COM , IP Address : 10.10.1.10] Client ('127.0.0.1', 40434): Request Server 2 to send IP address of VIT.COM
```




```
matlab@sjt516scope029: ~
set ns [new Simulator]
set nr [open 21BCE0169.tr w]
$ns trace-all $nr
set nf [open 21BCE0169.nam w]
$ns namtrace-all $nf
proc finish { } {
global ns nr nf
$ns flush-trace
close $nf
close $nr
exec nam 21BCE0169.nam &
exit 0
}
# Creating nodes
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
set n4 [$ns node]

#Creating  Lan connection between the nodes
set lan0 [$ns newLan "$n0 $n1 $n2 $n3 $n4" 1 Mb 40 ms LL Queue/DropTail MAC/Csma/Cd Channel]

# Setting up traffic
set tcp [new Agent/TCP]
$tcp set class_ 2
$ns attach-agent $n0 $tcp
set sink [new Agent/TCPSink]
$ns attach-agent $n3 $sink
$ns connect $tcp $sink

set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ftp set type_ FTP
$ftp set packet_size_ 1000
```

```
# Setting simulation parameters
$ns at 1.0 "$ftp start"
$ns at 2.0 "$tcp send 1000"

# Running the simulation
$ns at 5.0 "finish"
$ns run
-- INSERT --                                                    44,8          Bot
```

```
matlab@sjt516scope029: ~
(base) matlab@sjt516scope029:~$ vi 21BCE0169_bus.tcl
(base) matlab@sjt516scope029:~$ ns 21BCE0169_bus.tcl
```