**Overview of Application**

This application enables communication between a client and a server through a VPN. The client sends requests to the VPN, and VPN forwards these requests to the target server based on the client's message format. The server processes the requests and sends the responses back through the VPN to the client. The application supports two operations: concatenation and reversal of messages.

**Client->VPN Server Message Format**

Messages sent from the client to the VPN server are structured as follows:

<SERVER\_IP>|<SERVER\_PORT>|<OPERATION>|<MESSAGE>

* SERVER\_IP: The IP address of the target server.
* SERVER\_PORT: The port number on which the target server is listening.
* OPERATION: The operation to be performed on the message (either "concat" or "reverse").
* MESSAGE: The actual message content to be processed, with words separated by spaces.

For a message intending to concatenate words at server IP 127.0.0.1 and port 65432, it would look like this:

127.0.0.1|65432|concat|hi cs

**VPN Server->Client Message Format**

The responses sent from the VPN back to the client contain the processed result from the server. The format is simply the raw response from the server, as follows:

<RESPONSE>

Where <RESPONSE> is the result of the requested operation (e.g., the concatenated or reversed message).

If the server concatenates the words "hi" and "cs," the response would be:

hics

But in the given echo-server.py, the returned response would be: concat hi cs

**Example Output**

The following is an example of the output in the terminal when running the client, VPN and echo-server:

1. Client Output:

client starting - connecting to VPN at IP 127.0.0.1 and port 12345

Enter the operation (concat/reverse): concat

Enter words separated by space: hi cs

connection established, sending message127.0.0.1|65432|concat|hi cs

message sent, waiting for reply

Received response: 'concat hi cs' [12 bytes]

client is done!

1. VPN Output:

VPN is done!

server starting - listening for connections at IP 127.0.0.1 and port 12345

Connected established with ('127.0.0.1', 52693)

Received client message: 'b'127.0.0.1|65432|concat|hi cs'' [28 bytes]

Received response from server: 'concat hi cs'

1. Server Output:

server starting - listening for connections at IP 127.0.0.1 and port 65432

Connected established with ('127.0.0.1', 52701)

Received client message: 'b'concat hi cs'' [12 bytes]

echoing 'b'concat hi cs'' back to client

server is done!

**Network Layers Interaction**

When the application is executed, the following interactions occur between the network layers:

1. Client Initialization:
   * The client establishes a TCP connection with the VPN using the specified IP and port.
   * It encodes the message with the server's IP and port, operation type, and message, and sends this to the VPN.
2. VPN Processing:
   * The VPN parses the message to extract the server's IP and port, the operation, and the message.
   * The VPN establishes a new TCP connection with the target server using the parsed IP and port.
   * It forwards the client's operation request to the server and waits for a response.
3. Server Response:
   * The server processes the received request and sends the result back to the VPN over the established connection.
   * The VPN receives the server's response and forwards it back to the client.
4. Client Finalization:
   * The client receives the response from the VPN and prints it to the terminal, concluding the interaction.

**Command-line trace** **client and server in operation**

% python echo-client.py --server\_IP 127.0.0.1 --server\_port 65432 --VPN\_IP 127.0.0.1 --VPN\_port 12345

client starting - connecting to VPN at IP 127.0.0.1 and port 12345

Enter the operation (concat/reverse): concat

Enter words separated by space: hi cs

connection established, sending message127.0.0.1|65432|concat|hi cs

message sent, waiting for reply

Received response: hics

client is done!

% python echo-server.py

server starting - listening for connections at IP 127.0.0.1 and port 65432

Connected established with ('127.0.0.1', 53740)

Received client message: 'b'concat hi cs'' [17 bytes]

echoing 'b'concat hi cs' back to client

% python echo-client.py --server\_IP 127.0.0.1 --server\_port 65432 --VPN\_IP 127.0.0.1 --VPN\_port 12345

client starting - connecting to VPN at IP 127.0.0.1 and port 12345

Enter the operation (concat/reverse): reverse

Enter words separated by space: hi cs

connection established, sending message127.0.0.1|65432|reverse|hi cs

message sent, waiting for reply

Received response: scih

client is done!

% python echo-server.py

Connected established with ('127.0.0.1', 53741)

Received client message: 'b'reverse hi cs'' [19 bytes]

echoing 'b'reverse hi cs' back to client

% python echo-client.py --server\_IP 127.0.0.1 --server\_port 65432 --VPN\_IP 127.0.0.1 --VPN\_port 12345

client starting - connecting to VPN at IP 127.0.0.1 and port 12345

Enter the operation (concat/reverse): add

Enter words separated by space: hi cs

connection established, sending message127.0.0.1|65432|add|hi cs

message sent, waiting for reply

Received response: Invalid operation

client is done!

% python echo-server.py

Connected established with ('127.0.0.1', 53573)

Received client message: 'b'addition hi cs'' [20 bytes]

echoing 'b'addition hi cs'' back to client

**Acknowledgments**

Zhirou and TA Ashley helped to clarify the requirements and debug with the code.