To establish a client-server communication channel where the client sends a "ping" message, and the server responds with a "pong" message while accurately measuring and reporting the Round-Trip Time (RTT).
Implementation:

Server (server.py):

- The server listens on a specified IP address and port.
- Upon accepting a connection from the client, it receives the "ping" message and responds with a "pong" message.
- The server socket is closed when the communication is finished.

Client (client.py):

- The client connects to the server using the specified IP address and port.
- In each interaction with the server, it records the start time before sending the "ping" message.
- After receiving the "pong" response, it records the end time.
- The RTT (Round-Trip Time) is calculated as the difference between end time and start time, and it's reported in milliseconds.
- The client socket is closed when the communication is finished.

Execution:

- Run the server code (server.py) in one terminal to listen for connections.
- Run the client code (client.py) in another terminal to initiate communication.
- The client sends a "ping" message to the server, records the RTT, and receives a "pong" response.
- The client prints the server's response along with the calculated RTT for each interaction.
- The client continues this process until it receives the "pong" response and then exits.

This setup ensures that the client accurately measures the RTT for each message sent to the server and reports it. This information can be used to monitor the performance of the communication channel. The client will send "ping" messages to the server, and the server will respond with "pong" messages while incorporating sensor data from the Excel file. The round-trip time (RTT) is printed for each interaction.