The server program, implemented in `server.py`, is designed to handle UDP requests and respond to client pings. It incorporates a straightforward mechanism to simulate occasional packet loss by when random integers less than 4 out of 10, enhancing the realism of network communication. The server has no external dependencies, making the setup straightforward. In the event of issues, such as port binding problems, users are advised to ensure that the specified port (in this case, 12000) is available and not in use by other applications.

On the client side, the `client.py` program initiates ten ping requests to the server. It measures the round-trip time and prints the server's response, showcasing a basic network interaction scenario. To run the client program, similarly, navigate to its directory and execute `python3 client.py` in the terminal. The client, too, has no external dependencies beyond the Python standard library. Troubleshooting tips include checking for timeouts, verifying the server's responsiveness, and understanding that occasional packet loss is an intentional part of the simulation. Overall, this pair of programs offers a practical illustration of UDP communication and network interactions.