Report on DNS query server design using UDP

In this project, I use Python to implement the DNS query server under UDP.

The detail of the comment is in the code files, the UDP construction is based on the socket [1]package and the threading are based on the threading package[2]. I would like to introduce my design in the following paragraph

1. I initialize the server in a class object such that we can store some information inside the object, for example the cache loaded from the master file.
2. I choose the dictionary to store the information as cache, as dictionary is easy for the server to look up the information response to the client query. During the query process, we need to look up the domain and query type, then response with values, where the values could be multiple. Therefore, the design of the data structure to store the information is a dictionary of dictionary where the nested value is an array.
3. Then I go to the Client side, to make sure the consistency of the code style, I also initialize the client in a class object.
4. I integrated the input into formatted message as described in the comments
5. Then I go back to the Server side to parse the message sent by the client to obtain the useful information like qid, qname qtype.
6. Follow the instructions, I set a random delay to process the query.
7. I use the information obtained to obtain the value the client wants.
8. During the query processing, I use the cache to look up the domain and the record_type, if the domain is matched, I recursively trace the CNAME and then look up the record_type. If the domain is not matched, I use string split function split by the '.' to find all ancestors, and then find the first closest ancestors of the NS value, then find the A value of the corresponding NS.
9. Then pack all the information described in the comment send back to the client.
10. Then I go back to the Client side to handle the message sent from the server.
11. At the end I set up the logs for the server where I record the time received the message from the client and sent to the client.

Limitation:

1. Reboot time for the server is not set
2. The information stored inside the cache of the server must be manually updated
3. Size of the message is still limited.
4. A single client must wait until the last query from itself was finalised.
5. Cannot find some missing record, limited to the cache.

[1] https://docs.python.org/3/library/socket.html
[2] https://docs.python.org/3/library/threading.html