COMP3331/9331 Computer Networks and Applications

Assignment Mark Request for Term 2, 2023

Language:	Python 3

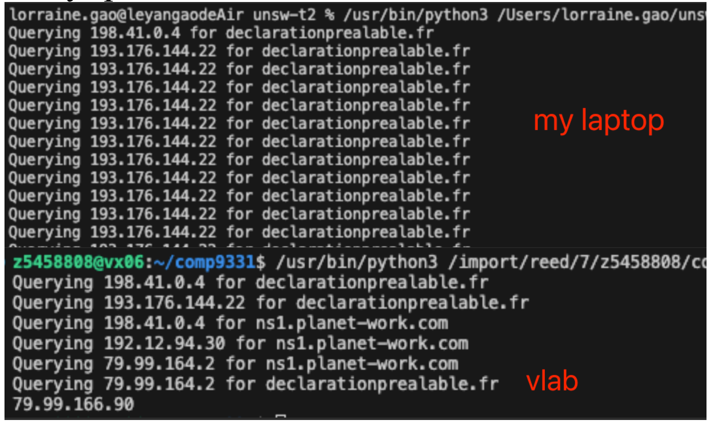
Basic Version

Limitations/Issues	**Possible Solutions**
The first issue is related to parse the DNS response, with respect to the answer section and authoritative nameservers section I've made an incomplete parse and only extracted the first one answer/authoritative nameserver, which is not perfect. Secondly, when parsing, my idea is to skip the content that occupies a fixed number of bytes such as TTL, length and so on. when I searched the information I want, I use the way which marking the position to extract the information one by one until I encountered a b’\X00\X00’ and stop. But throughout the parsing I think I was repeatedly writing more similar code, such as slicing the response, moving the index position, etc., making the code less readable and not elegant. The second problem is that when dealing with compressed pointers, my code is also complicated and not easy to read. I take a more rigid approach, parsing the offset bytes and then looking for them from the beginning of the responding message based on its number. Lastly, when implementing the iterative query for DNS, even though my code was able to successfully iterate the query, I don't think my judgement conditions were written	For these problems, I think there are kinds of solutions: 1) reasonable and appropriate integrated , such as in the parsing function, it can be split into parsing header, parsing queries, parsing records, including the processing of the compression function can also be integrated, so that i can use it better , and make the code more readable (2) I can use import BytesIO to use, so that I can then parse the byte data to better deal with, do not have to frequently use the list of slices. Including to parse all the records can write a good sub-function, use the form of for loop. (3) I can use the class, set some header class, flag class, so that it can be more logical, easy to call and pass parameters. In terms of in object-oriented programming I’m not skillful still now and I will exercise more. (4) About the concepts with byte data, binary to hexadecimal, DNS iterative query and other basic networking concepts need to consolidate memory and deeper understanding, some concepts are not mastered thoroughly enough.

adequately, and there were some edge cases that weren't considered.

Also as far as I known, some domain names have multiple Ip address but my resolver just return one Ip address.

By the way I'd to ask why there is a difference result between at my laptop and at cse'server with the same code, just like this. I don't know the reason until now but it indeed spends me lotssss of time to debug. I'm appreciate it if my question can be answered.



```
lorraine.gao@leyangaodeAir unsw-t2 % /usr/bin/python3 /Users/lorraine.gao/uns
Querying 198.41.0.4 for declarationprealable.fr
Querying 193.176.144.22 for declarationprealable.fr
Querying 193.176.144.22 for declarationprealable.fr
Querying 193.176.144.22 for declarationprealable.fr
Querying 193.176.144.22 for declarationprealable.fr
Querying 193.176.144.22 for declarationprealable.fr
Querying 193.176.144.22 for declarationprealable.fr
Querying 193.176.144.22 for declarationprealable.fr
Querying 193.176.144.22 for declarationprealable.fr
Querying 193.176.144.22 for declarationprealable.fr
Querying 193.176.144.22 for declarationprealable.fr
Querying 193.176.144.22 for declarationprealable.fr
```

```
z5458808@vx06:~/comp9331$ /usr/bin/python3 /import/reed/7/z5458808/co
Querying 198.41.0.4 for declarationprealable.fr
Querying 193.176.144.22 for declarationprealable.fr
Querying 198.41.0.4 for ns1.planet-work.com
Querying 192.12.94.30 for ns1.planet-work.com
Querying 79.99.164.2 for ns1.planet-work.com
Querying 79.99.164.2 for declarationprealable.fr
79.99.166.90
```

Enhancement 1:	Error Handling

Limitations/Issues	**Possible Solutions**
On the programming user's interface, I can only set the fourth parameter to be timeout value and the fifth parameter to be the query type. I think there should be a form so that no matter user input the fourth argv or not, the default value is '5' and 'a', and if one of these or both of them are entered it can be accurately determined whether it is timeout or type. Now my code is if u want you input the query type, you must print timeout value firstly.	For this, I think we can put the all kinds of query types into a list, and if one or two or more additional conditions are entered, we can check if they are positive numbers and if they match any of the elements in the list, and then we can determine exactly what the two parameters are.

Enhancement 2:	Advanced Records and Reverse DNS

Limitations/Issues	**Possible Solutions**
For some queries, it is necessary to identify the type of the response message in particular the answer section, and then take the next step based on its type. For example, when querying for mx messages, If the result returned does not have an answer for MX, we need to be restarting the query with the canonical name. And secondly for query type of ptr I did not succeed in build query and send.	When parsing the response message, it is important to extract all the information as comprehensively as possible, storing it in different variables to make it easier to call when enhancing. Also, in doing iterative query is to add more judgement conditions and if types. I need to be more familiar with the concept of ptr-type queries, and at the same time, I need to give myself enough time for assignment work and arrange the study pace reasonably.

Enhancement 3:	Performance Reporting
Limitations/Issues	**Possible Solutions**
I didn't categorise the different types of query failures and used time out for all of them, but it's likely that some of them are due to other reasons such as domain failure or decoding failures At the same time, the overall logic is pool. It's all about trying to automate the process, finding out the problem and being forced to pause, then looking at the error message to add conditions to debug that. Otherwise if one error is reported, it didn’t jump to next dns query instead of pausing.	Use more try-except ways to capture the type of exception, but also easy to manage what message to return, reduce the use of simple if judgement statement. And I’d learn to understand the dnspython library defined in some common error types, understand their respective meanings, this can be better used in the future, if there are similar scenarios can also be imitated