Daniel Garcia

111157599

README

PART A

The mydig.py program works almost similar to the dig command in UNIX. To start, write mydig.py in the command line and provide it with domains to be queried. The program will accept as many queries as the user wants. (i.e. If the user writes mydig.py facebook.com google.com, it will do a DNS lookup for both facebook.com and google.com.) The program currently uses some of the external libraries in DNSPython, such as dns.name, dns.message, and dns.query.

To start, the program currently sets the output of the script to a new file called “mydig_output.txt”. Afterwards, it puts all domain names entered from the command line in a list. If no domains are supplied, the list will only contain an empty string. For each domain, a dns.name.Name object is created from the given domain string, and this .Name object is used to make a query of type A. Afterwards, an empty list where all the answers of the query will be contained is created, a timer is started, and the myDig method is called to perform UDP requests.

Inside this method, a UDP request is created, and if it yields IP addresses in the “answer” section, it is appended to the list. Next, the “additional” section is checked for A type records, and if they are found, myDig is again called recursively until answers are yielded to find the IP of the domain.

If the answer list is still empty after checking both sections, the “authority” section will then be called to look for name servers, and the myDig function will be called recursively with a new query. The myDig function will then keep getting called until the “answer” section becomes filled.

Afterwards, the timer is ended and query time is calculated and outputted. The date is also outputted, as well as the total message size of the answer received.

One thing to note is that the UDP request has a timeout of 60 seconds. If nothing is received for this amount of time, the method gives a timeout error, and the program exits.

PART B

The data from experiment one is the myDig program called with the root server of “a.root-servers.net”, while experiment 2 and 3 uses the UNIX dig command to the local DNS server and the google DNS server. For experiment one, the average query time and percentiles are high due to the fact that there is no cache, and because the program is called recursively while not being optimized for recursion. However, for experiment two, the query time is high for the first time, while the other 9 times are super low since the DNS request will look at the cache and see that the answer already exists in the cache. For experiment three, the results are between the root server and the local DNS server since there is still cache in the google domain, but there is still a lot of distance between the host machine and the google servers.