High Level Languages for Low Level Networking

Gordon Adam - 1107425a

February 9, 2015

1 Project Description

The aim of this project is to develop a DNS Resolver. The resolver should be able to take a recursive query from a client, process the request in a suitable manner and return the result. This is to be implemented using rust, a systems programming language which is still under development.

2 Progress

Currently the resolver works in its most basic form i.e. it is capable of taking a request from a client and resolving that request by performing iterative queries. An iterative query is one where the answer is not always returned, if the answer is not returned an alternative server with which to query is provided. The resolver so far can take requests, from programs such as a browser or nslookup, and process records of three different types. That is of NS, A and AAAA; which stands for a name server, an IPv4 address and an IPv6 address respectively. These types have been the minimum required to get the resolver working.

3 Plan

The plan of work to be carried out for the rest of this project is to implement support for other query and record types that are recieved, there are too many unfortunately to feasibly cover them all in the scope of this project. For example there are sixty types that I am aware of and at this point, as mentioned earlier, there are only three of which I have actually implemented support for. However the more common types such as the CNAME, SOA, PTR and TXT types should be included, and if I recieve a request for a type that is not implemented I can return the request with the error code `0100` inside it which simply stands for not implemented. A caching service should also be included in order to increase the performance of the resolver.

4 Problems

Problems that I have experienced so far have been the compression of text within a dns message, this was difficult to resolve as I didn't properly research the rules surrounding the compression and decompression of dns messages in the appropriate rfc documents. Instead I took it upon myself to analyse the raw data in the packets, giving myself needless hardship. After having done the proper research however the whole problem was made clearer and simpler. Some issues that I foresee are the complexity of approaching the problems of caching and handling many different record types. This however, just as was demonstrated with the problem encountered earlier, can be resolved by researching the problem more fully before stepping in to solve it.