

IP address lookup is an important processing function of Internet routers. The challenge lies in finding the longest prefix that matches the packet's destination address. One of the issues concerning IP address lookups is the average lookup time. In previous works, caching was shown to be an effective method to minimize the average lookup time. Caching involves storing information on recent IP lookup results in order to decrease average lookup times.

In this thesis, we present two architectures that contain a prefix cache and a dynamic substride cache. The dynamic substride cache stores longest possible substrides from previous lookups, and is used in conjunction with a prefix cache. Successful hits in both the caches help reduce the number of worst-case lookups in the low level memory containing the IP routing table in a trie data structure. From simulations, we show that the two architectures show up to 99.9% global hit rate. Furthermore we present analytical models to find optimal designs for the two architectures. We also show that the architectures can support incremental updates once appropriate modifications are made to the trie data structure.