Review of Small Forwarding Tables for Fast Routing Lookups

Swaminathan Sundararaman

February 04, 2008

This paper demonstrates that IP routing lookups can be done in software at gigabits speeds. The authors present a new forwarding table data structure that enabled general purpose processors to perform full routing lookups at gigabit speeds. The forwarding table data structure compressed the complete prefix tree that stores the entire IPv4 address space using bitmaps, code words, and base index arrays to locate the pointer entry in their maptable data structure. The maptable data structure stores the next hop addresses on the path towards the destination.

The advantages of the authors approach are: (1) space efficiency through compact representation to reduce memory requirements, (2) faster lookup speeds by minimizing the number of memory accesses and size of the in-memory data structures, (3) inexpensive hardware, and (4) scalability to support increase in the routing table sizes. The main drawback I see in this paper is that their evaluation methodology is very weak. The authors performed each look up in the forwarding table twice to eliminate cache pollution. The cache pollution was caused by their statistics collection code, and scheduling mechanisms in the operating system used for benchmarking. This is not correct as the processors would still have to bring the packet headers inside its cache, and run code to read packet headers along with lookup in the forwarding table. Hence, the overheads shown in the paper does not truly represent the actual overheads. In my opinion, simulation of the lookups in forwarding table would have been more meaningful.

The main contribution of the paper is that the authors disproved the popular belief of the networking community that fast routing lookup is not possible in software at gigabits speeds. They used a compact representation of the forwarding tables that scaled with increase in routing table sizes for IPv4 addresses. Though their solution scaled well for IPv4 based addresses, it cannot be directly applied to routing on IPv6 addresses for gigabit speeds. This is because IPv6 has an exponentially large address space and compacting and placing most of the forwarding table in the processors cache is not possible.