

A two-bit scheme for routing lookup

Balaji Prabhakar Pankaj Gupta Stephen Boyd
Stanford University, Stanford, CA 94305, USA

I Summary

Performing address lookup at a router has become a severe bottleneck. This is because increasing link speeds leave very little time to lookup an address, and an explosion in the number of hosts has led to a drastic increase in router table sizes. As a result, this problem has received widespread attention both in academia and in the industry.

For destination-based routing, work on routing lookup has led to matching based on longest address prefixes, and several fast algorithms have recently been proposed to do address lookups. Important attributes that a look-up engine should possess are hardware-implementability (for speed), dynamic-reconfigurability (because address tables are frequently updated).

We consider the longest-prefix matching problem in which the end points of the interval corresponding to a prefix are stored at the leaves of a binary search tree. The novelty of our approach is that it considers the probabilities with which a leaf is accessed. Then the restriction to binary search procedures naturally confine us to alphabetic trees. Further, in order to bound the worst-case lookup time, the alphabetic trees need to be depth constrained.

We present a fast algorithm for determining depth constrained alphabetic trees, and discuss its complexity and optimality properties. We will also describe its performance on actual address tables taken from routers in the backbone of the Internet.