CHAPTER 6

Future Directions

As we saw in Section 1 of Chapter 1, the packet processing capacity of IP routers needs to keep up with the exponential increase in data rates of physical links. This chapter sets directions for future work by proposing the characteristics of what we believe would be 'ideal' solutions to the routing lookup and packet classification problems.

1 Ideal routing lookup solution

We believe that an ideal routing lookup engine (we restrict our scope to IPv4 unicast forwarding) has *all* of the following characteristics:

- **Speed:** An ideal solution achieves one routing lookup in the time it takes to complete one access in a (random-access) memory in the worst-case. This characteristic implies that an ideal solution lends itself to pipelining in hardware.
- **Storage:** The data structure has little or no overhead in storing prefixes. In other words, the storage requirements are nearly $32N$ bits, or better, for N prefixes in the worst-case. A less stringent, though acceptable, characteristic could be that the storage requirements scale no worse than linearly with the size of the forwarding

table. If the backbone forwarding tables continue to grow as rapidly as we saw in Section 1.2.1 of Chapter 1, exponentially decreasing transistor feature sizes will enable implementations of an ideal routing lookup solution to continue to gracefully meet the demands posed by future routing table growth.

- **Update rate:** Based on current BGP update rates, an ideal solution supports at least 20,000 updates per second[1] in the worst case. Furthermore, updates are atomic in that they do not cause interleaved search operations to give incorrect results.
- **Feasibility of implementation:** Implementations of an ideal lookup solution should be feasible with current technology, e.g., should not consume an unreasonable number of chips, or dissipate unreasonable amount of power, or be too expensive.

Note that amongst the solutions known at the time of writing, ternary CAMs have desirable storage (and possibly update rate) characteristics, but do not have the speed of one RAM access, and do not admit feasible implementations supporting large routing tables. Even though the algorithm proposed in Chapter 2 seems to satisfy all but the update rate requirements, the large storage requirements of this algorithm dictate that the fastest memory technology, i.e., SRAM, cannot be used. This imposes an inherent limitation on the routing lookup rates achievable using this algorithm.

If an ideal solution did exist today, it would consume $32 \times 256K = 8$ Mb of memory for 256K prefixes.[2] Now, 8 Mb of fast SRAM (with 3 ns cycle time) can be easily put on a reasonable sized chip in current 0.18 micron technology. Hence, an ideal solution would be able to lookup 333 million packets per second, enough to process 40 byte minimum-sized TCP/IP packets at line rates of 100 Gbps.[3] In contrast, only 66 million packets per

1. This is two orders of magnitude greater than the peak of a few hundred reported by Labovitz [47].
2. 256,000 is more than double the number of prefixes (98,000) at the time of writing (see Section 1.2 of Chapter 1).
3. Again, this ignores the packet-over-SONET overhead bytes.

second can be looked up by solutions available today — the algorithm proposed in Chapter 2 (using embedded DRAM), and ternary CAMs (using 2-4 chips).

2 Ideal packet classification solution

An ideal packet classification solution not only has all the characteristics we saw above of an ideal lookup solution — that of high speed (classification at line-rate), low storage (to support thousands of classification rules), fast incremental updates, and feasible implementation (cost effective) — but also the characteristics of flexibility in the number and specification syntax of packet header fields supported. In contrast with routing lookups, it is harder to quantify the desirable values of these parameters for packet classification because of the lack of a sufficient amount of statistical data about real-life classifiers. However, it is not unrealistic to imagine a carrier's edge router supporting 1000 ISP subscribers, each with at least 256 five-field classification rules, for a total of 256,000 128-bit rules required to be supported by the classification engine of a router.

In light of the worst-case lower bounds on multi-dimensional classification algorithms mentioned in Chapter 4, an ideal classification solution is unlikely to be able to support all possible worst case combinations of classification rules, and yet satisfy all the other characteristics mentioned above. We believe that intelligent heuristic solutions should be acceptable.

We now see how close the solutions known at the time of the writing of this thesis approach that of an ideal solution. A total of sixteen 2 Mb ternary CAM chips are required to support 256,000 128-bit classification rules. The resulting power dissipation and cost of the system would be clearly excessive. Similarly, the Recursive Flow Classification algorithm of Chapter 4 would require approximately 9 DRAM chips.[1] Though not in terms of

1. Based on experiments shown in Section 4.3 of Chapter 4, we assume that 4K rules occupy a maximum of approximately 4.5 Mbytes of memory, and that each DRAM chip has a density of 256 Mbits.

power dissipation, this solution is still expensive in terms of board real-estate. The HiCuts algorithm of Chapter 5 would require 4 DRAM chips but would be two to four times slower than a ternary CAM or recursive flow classification solution.

3 Final words

The above mentioned ideal lookup and classification solutions appear challenging to obtain, and will probably require not only improved circuit technologies, but also new data structures and algorithms. Hopefully, this dissertation will serve as a useful foundation for future research in this exciting field in general, and in attempts to obtain these (or similar) ideal solutions in particular.