

# Chapter 11: Conclusions

*June 2008, San Francisco, CA*

## Contents

---

11.1 Thesis of Thesis . . . . .	309
11.2 Summary of Contributions . . . . .	309
11.3 Remarks Pertaining to Industry . . . . .	310
11.4 Remarks Pertaining to Academia . . . . .	312
11.4.1 Spawning New Research . . . . .	313
11.5 Closing Remarks . . . . .	314
11.5.1 Limitations and Open Problems . . . . .	314
11.5.2 Applications Beyond Networking . . . . .	315
11.5.3 Ending Remarks . . . . .	316

---

You cannot simply cross the Bay Area on a whim. San Francisco Bay, more than 1600 square miles in size, neatly bisects the area in two separate halves. A series of freeways with toll booths connect the Peninsula and East Bay. For a picturesque crossing, you can take the Golden Gate Bridge at the mouth of the Bay, one of the world's largest suspension bridges, considered a marvel of engineering when it was built in 1937.

However, if you want to avoid the toll (\$5 if you are well-behaved) and are in a mood to inspect more recent engineering wonders, take the turnoff to Highway 237 at the Bay's opposite end. The freeway begins in Mountain View, home of Google, a small company specializing in niche search engine technology (we suggest you look them up). If you're not in a rush, take the last exit. You'll soon find yourself in a large campus with buildings that all look alike, the home of a networking systems vendor. Bear right on Cisco Way, and you'll see an inconspicuous sign — "*Datacenter Networked Applications (DNA) lab, Cisco Systems*".

Unlike its large, frugal setting, the DNA lab is small and holds an expensive assemblage of the highest-speed Ethernet switches and Enterprise routers, including storage equipment, high-performance computing systems, disk arrays, and high-density server farms. These products are placed neatly side by side to showcase the next-generation, high-speed data center networking technologies. Glance around and you'll see a large array of overhead projectors, and walls adorned with ~20+ onscreen displays. Go ahead, look around and experience the tremendous speed and capabilities of these modern engineering marvels. Be sure to ask a network expert for a demo. Legend has it that if you look carefully, you'll find (tell us if you do) a nondescript sticker on each router: "High-speed router — fragile, handle with care. . ."



*“If you are afraid of change, leave it here”.*

— Notice on a Tip Box<sup>†</sup>

*“Make everything as simple as possible, but not simpler”.*

— Albert Einstein<sup>‡</sup>



## Conclusions

### 11.1 Thesis of Thesis

The thesis of this thesis is two-pronged, with one prong pertaining to industry and the other to academia —

1. We have *changed* the way high-speed routers are built in industry, and we have showed that (a) their performance can be scaled (even with slow memories), (b) their memory subsystems need not be fragile (even in presence of an adversary, either now or, provably, ever in future) and (c) they can give better deterministic memory performance guarantees.
2. We have contributed to, unified, and, more important, *simplified* our understanding of the theory of router architectures, by introducing (and later extending) a powerful technique, extremely simple in hindsight, called “constraint sets”, based on the well-known pigeonhole principle.

### 11.2 Summary of Contributions

↻**Note 11.1.** A summary of the 14 key results of this thesis is available in Section 1.11. For more details on (1) the specific problems solved in this thesis, refer to Section 1.7; and for (2) the industry impact and consequences of these ideas, see Section 1.9; and (3) for a list of academic contributions

---


<sup>†</sup>Unknown Cafe, Mountain View, CA, Oct 2004.

<sup>‡</sup>An obscure scientist of the early twentieth century.

that unify our understanding of the theory of router architectures, refer to Table 2.1 and Section 2.3.

The next two sections discuss the impact of this thesis on industry and academia respectively.

## 11.3 Remarks Pertaining to Industry

 **Observation 11.1.** Networking both reaps the boons and suffers the curses of massive current deployment. This means that new ideas can make a large impact, but that they have an extremely hard time gaining acceptance. Deploying a new idea in existing networks can be technically challenging, require cooperation among hosts in the network, and can sometimes be economically infeasible to put in practice. Consequently, we face the unfortunate reality that a large percentage of really good ideas, in academia and industry, are never deployed in networking. In that sense, we were fortunate, because our ideas pertain to the networking infrastructure layer, *i.e.*, routers, so their acceptance does not need cooperation among existing routers.


Writing a thesis several years after finishing the main body of the research<sup>1</sup> has at least the benefit that we don't need to predict the potential of our academic work. At the time of writing, we have managed to add to the repertoire several new research ideas relevant to this area, and most important we have brought these ideas to fruition. However, their deployment has been by no means easy. It has been a learning, challenging, and humbling experience.

Routers are complex devices — they must accommodate numerous features and system assumptions, require backward compatibility, and have ever-changing requirements due to the advent of new applications and network protocols. This has meant

---

<sup>1</sup>I wouldn't recommend this to new students — “*all but dissertation*” is not a pleasurable state of mind.

that, in order to realize our techniques in practice, we have had to modify them, devise and introduce new techniques, and fine-tune existing ones. Achieving these goals has taken nearly four years and significant resources. In hindsight, it was important to spend time in industry, understand the real problems, and solve them from an insider's framework. In the course of implementation, we have learned that designs are complex to implement, and their verification is challenging.

 **Example 11.1.** For example, the packet buffer cache (see Figure 7.3) has six independent data paths, each of which can access the same data structure within four clocks on 40 Gb/s line cards. This leads to data structure conflicts, and requires several micro-architectural solutions to implement the techniques correctly. Similarly, almost all our designs have had to be parameterized to cater to the widely varying router requirements of Ethernet, Enterprise, and Internet routers. The state space explosion, to deal with massive feature requirements from different routers, meant that these solutions take years and large collaborative efforts to build and deliver correctly.

It is a good time to ask — *Have we achieved the goals we set for ourselves at the onset of this thesis?* At the current phase of deployment, I believe that it is an ongoing but incomplete success. At the time of this writing, we have largely met our primary goals — to scale the memory performance of high-speed routers, enable them to give deterministic guarantees, and alleviate their susceptibility to adversaries. We have designed, evangelized, and helped deploy these techniques on a widespread scale, and it is expected that up to 80% of all high-speed Ethernet switches and Enterprise routers<sup>2</sup> will use one or more instances of these technologies. We are also currently deploying these techniques on the next generation of 100 Gb/s line cards (for high-volume Ethernet, storage, and data center applications). However, our work is still incomplete, because we have yet to cater to high-speed Internet core routers. At this time, we are in discussions for deploying these ideas in those market segments as well.

---

<sup>2</sup>Based on Cisco's current proliferation in the networking industry.

The secondary consequences of our work have resulted in routers becoming more affordable – by reducing memory cost, decreasing pins on ASICs, and reducing the physical board space. They have also enabled worst-case and average-case power savings. In summary, our work has brought tremendous engineering, economic, and environmental benefits.

☞**Note 11.2.** From a user’s perspective, a network is only good if every switch or router (potentially from several different router vendors) in a packet’s path can perform equally well, give deterministic guarantees, and be safe from adversarial attacks. To that end, most of the ideas described in this thesis were done at Stanford University, are open source, and available for use by the networking community at large.


## 11.4 Remarks Pertaining to Academia

I will now touch on the academic relevance of this work, and comment separately on load balancing and caching techniques. It is clear that the constraint set technique introduced in this thesis (to analyze load-balanced router memory subsystems) simplifies and unifies our understanding of router architectures. However, it also has an interesting parallel with the theory of circuit switching.

☞**Observation 11.2.** Note that router architecture theory deals with packets that can arrive and be destined to any output (there are a total of  $N^N$  combinations), while circuit switches deal with a circuit-switched, connection-based network that only routes one among  $N!$  permutations between inputs and outputs. In that sense, router architecture theory is a superset of circuit switching theory [110]. However, circuit switch theory and constraint sets both borrow from the same pigeonhole principle. This points to an underlying similarity between these two fields of networking. From an academic perspective, it is pleasing that the technique of analysis is accessible and understandable even by a high school student!

### 11.4.1 Spawning New Research


Our initial work has led to new research in both algorithms and architectural techniques pertaining to a broader area of *memory-aware algorithmic design*. Some significant contributions include — simplified caches using frame scheduling, VOQ buffering caches [181], packet caching [182], lightweight caching algorithms for counters [202, 203, 204, 205], caching techniques to manage page allocation, and increased memory reliability and redundancy [32].

 **Observation 11.3.** Our caches have had some interesting and unforeseen benefits. L2 caches<sup>3</sup> (which are built on top of the current algorithmic L1 caches) have been proposed and are currently being designed to decrease average case power [25]. Also, larger L1 caches have been implemented to hide memory latencies and allow the use of complementary high-speed interconnect and memory serialization technology [32]. The structured format in which our L1 caches access memory also allows for the creation of efficient memory protocols [214] that are devoid of the problems of variable-size memory accesses, and avoid the traditional “65-byte” problem and memory bank conflicts.

In the process of development, we have sometimes re-used well-known existing architectural techniques, and have in some cases invented new ones to deal with complex high-speed designs. For example, the architectural concepts of cache coherency, the use of semaphores, maintaining the ACID [215] properties of data structures (which are updated at extremely high rates), deep pipelining, and RAID [30] are all concepts that we have re-used (and tailored) for high-speed routers. Similarly, we have re-applied the ideas pertaining to adversary obfuscation (see Chapter 10) to many different applications. While these concepts are borrowed from well-known systems ideas in computer architecture [155], databases, and the like, they also have significant differences and peculiarities specific to networking.

---

<sup>3</sup>These terms are borrowed from well known-computer architecture terminology [155].

 **Observation 11.4.** From an academic perspective, it is heartening that there is a common framework of load balancing and caching techniques that are re-used for different purposes. Also, router data path applications, for the most part, have well-defined data structures, and so lend themselves to simple techniques and elegant results. I believe that routers have reached a semi-mature stage, and I hope that this thesis will convince the reader that we are converging toward an end goal of having a coherent theory and unified framework for router architectures.

## 11.5 Closing Remarks

### 11.5.1 Limitations and Open Problems

Based on my experience in the networking industry, I would like to divide the limitations of the techniques presented in this thesis into three broad categories. I hope that this will stimulate further research in these three areas.

1. **Unusual Demands From Current Applications:** There are additional features that data path applications are required to support on routers, outside their primary domain. In general, any features that break the assumptions made by the standard data structures of these applications can place unexpected demands on our memory management techniques, and make their implementation harder, and sometimes impractical. For example, unicast flooding requires packet order to be maintained among packets belonging to *different* unicast and unicast flood queues. This usually doubles the size of data structures for the queue caching algorithms. Similarly, dropping large numbers of packets back to back<sup>4</sup> requires over-design of the buffer and scheduler cache; buffering packets received in temporary non-FIFO order makes the implementation of the tail cache more complex, and handling extremely small packets at line rates can cause resource

---

<sup>4</sup>Packets can easily be dropped *faster* than the line rate, because in order to drop a packet of any size, only a constant-size descriptor needs to be dropped.




problems when accessing data structures. All of the above require special handling or more memory resources to support them at line rates.

2. **Scalability of Current Applications:** The complexity of implementation of our results (in terms of number of memories required, cache sizes, *etc.*) usually depends on the performance of the memory available. While our techniques are meant to alleviate the memory performance problem, and in theory, have no inherent limitations, there are practical limits in applying these approaches, especially when memories become extremely slow. Also, there are instances where the feature requirements are so large that our techniques are impractical to implement. As an example, some core routers require tens of thousands of queues. Multicast routers require  $\Theta(2^f)$  queues, where  $f \leq q$  is the maximum multicast fanout, and  $q$  is the number of unicast queues. The cache size needed to support this feature is  $\Theta(f * 2^f)$ , which can be very large. Similarly, load balancing solutions require a large ( $\equiv \Theta(\sqrt{f})$ ) speedup (refer to Appendix J) to achieve deterministic performance guarantees for multicast traffic. This is an area of ongoing concern that needs further improvement.
3. **Unpredictability of Future Applications:** Our caching and load balancing solutions are not a panacea. The router today is viewed as a platform, and is expected to support an ever-increasing number of applications in its data path. These new applications may access memories in completely different ways and place unforeseen demands on memory. We cannot predict these applications, and new techniques and continued innovation will be necessary to cater to and scale their performance.

### 11.5.2 Applications Beyond Networking

Our techniques exploit the fundamental nature of memory access. While the most useful applications that we have found are for high-speed routers, their applicability is not necessarily limited to networking. In particular, the constraint set technique could be used wherever there are two (or more) points of contention in any load balancing application, for example, in job scheduling applications. They can also

be used in applications that keep FIFO and PIFO<sup>5</sup> queues. Similarly, our caching techniques can be used in any application that uses queues, manipulates streams of data, walks linked lists (*e.g.*, data structures that traverse graphs or state tables, as used in deterministic finite automata (DFA)), aggregates or pre-fetches blocks of data, copies data, or measures events.

 **Observation 11.5.** Of course, the memory acceleration technique introduced in this thesis can be used to speed up (by trading off memory capacity) the random cycle time performance of *any* memory. The speedup is by a factor,  $\Theta(\sqrt{h})$ , where  $h$  is the number of memory banks. This points to an interesting and fundamental tradeoff. As memory capacity increases at the rate of Moore's law, we expect this to become a very useful and broadly applicable technique. In addition, this technique is orthogonal to the rest of the load balancing and caching techniques described in this thesis, and so can be used in combination with these various techniques to increase memory performance.

### 11.5.3 Ending Remarks

*"Sufficiency is the child of all discovery".*

— Quick Quotations Corporation

As system requirements increase, the capabilities of hardware may continue to lag. The underlying hardware can be slow, inefficient, unreliable, and perhaps even variable and probabilistic in its performance. Of course, some of the above are already true with regard to memory. And so, it will become necessary to find architectural and algorithmic techniques to solve these problems. If we can discover techniques that are sufficient to emulate the large performance requirements of the system, then it is possible to build solutions that can use such imperfect underlying hardware. I believe that such techniques will become more common in future, and are therefore a continuing area of interest for systems research.

---

<sup>5</sup>This is defined in Section 1.4.2.

High-speed routers are complex devices. They function at the heart of the tremendous growth and complexity of the Internet. While their inner workings can in some instances be complex (and, like musical notation, the underlying mathematics can at times be intimidating and can hide their inherent beauty), for the most part I hope to have conveyed in this thesis that they are, in fact, quite simple, have a common underlying framework, and lend themselves to elegant analysis. If a high school student can appreciate this fact, I would consider the thesis successful.