RESEARCH STATEMENT

Sundar Iyer (sundaes@cs.stanford.edu)

My research interests span the areas of network algorithms, system architecture and component design. A common thread in my research is in understanding the theory and design of scalable architectures and parallel systems. I have resorted to mathematical methods of proof which are borrowed from the areas of Algorithms, Architecture, Combinatorics, Probability, & Queueing Theory. Broadly speaking, my research belongs to the area of *Network Architecture* (which deals with the fundamental principles of network design), an upcoming field which is still in its infancy and whose theoretical foundations are just being laid.

Background and Current Work

It is natural that most large and complex systems are created from a set of smaller components, each of which is simpler than the system as a whole. The functioning of a complex system depends on the interaction between its constituent components. Parallelism is a standard technique used to scale the performance of a system. In a parallel system each individual component is identical in nature and can perform all the tasks of the larger system, albeit usually at a slower rate. Of course it is naive to expect that we could simply bunch together a number of parallel components and get high performance. In particular, the performance of any parallel system is governed by — 1) The architecture used to connect the many components and 2) the resource management (load balancing) algorithm used to allocate tasks amongst the parallel components.

There exists a rich body of previous work in the field of Computer Architecture, Distributed Systems & Theoretical Computer Science, which answer a number of fundamental questions regarding the performance of load balancing algorithms and the design of parallel architectures. However this has not happened in the field of networking. Academia and Industry in particular (rushed by the exponential growth in networking) have compromised by using ad hoc parallel solutions. This has resulted in systems which provide little or no performance guarantees. For example — most Internet core routers in use today do not give any guarantees on bounding packet delay. It is easy to see how this unhappy state of affairs can adversely affect future advances in the Internet.

We need to understand the general principles of designing large systems, which give performance guarantees. These guarantees could be statistical, say, guaranteeing less than 0.001% packet loss on a network, or deterministic, say, never more than 1 μs of packet delay across a network. Only recently has there been growing interest in the theory and design of scalable networking systems. In my Ph.D thesis, I attempt to lay a theoretical framework for this field. Most of my work has greatly benefited from interaction with a number of colleagues and my advisor. I describe my work below.

An Analytical Framework to Analyze Router Architectures

In the past decade, router design has enjoyed both widespread academic interest and commercial success. I ask the following question — *Is there a common technique, which allows us to analyze router architectures that give deterministic guarantees?* I observed the existence of such a technique called constraint sets, in the course of solving two open problems about scaling router capacity —

1. *Is it possible to emulate a fast ideal router, using only slower speed routers?*
2. *Is it possible to emulate an ideal centralized shared memory router using only distributed memories?*

Constraint sets are a generalization of the Pigeon-hole principle applied to routers. I showed that router design can be considered as a game where arriving pigeons (packets) are load balanced amongst pigeon-holes (memories). It is surprising that the above problems can be solved [**?**, **?**, **?**] in a simple manner using the Pigeon-hole principle because they refute many commonly accepted myths about router design. Also the method of analysis is eye-opening because it captures the structural requirements of any router. I came up with a generic model for a class of routers called Single Buffered Routers. I showed how the Pigeon-hole principle can be applied in the analysis of Single Buffered Routers that give deterministic guarantees. Later, I extended the analysis to routers with two stages of buffering [**?**]. Thus our model and analysis technique presently incorporates almost all the router architectures in use in the core of the Internet today and shows how router capacity can be scaled in an efficient manner.

Deterministic Architectures for Packet Processing

All network equipment perform packet processing. However it is still not well understood, primarily due to the variety of different processing tasks. These tasks place a heavy demand on instruction and memory bandwidth, which prevents them from being implemented on general-purpose network processors. While specific solutions exist, in most cases it is not known whether they are optimal, whether they are complete i.e. support all necessary packet processing features and whether they give any performance guarantees. I look at three different aspects of this problem —

1. *Optimal and flexible packet buffers, which eliminate cache misses*

   Packet buffers are built using cache hierarchies. As is well known, caching can only give statistical guarantees on packet access time, resulting in unpredictable packet latency. In contrast, I proposed deterministic algorithms, which exploit the characteristics of memory requirements for networking to design a memory hierarchy, which eliminates cache misses. I showed how the optimal buffer caching algorithm can be modeled using difference equations and used adversarial traffic patterns to show that it is optimal [**?**]. This resulting memory architecture supports the high access speeds of the cache while having the large storage capacity of main memory, obviating the need for any special purpose memory for networking. Later, I showed how the cache hierarchy could be designed to allow flexibility in choosing any buffer access latency. A number of router companies such as Cisco, Juniper as well as main memory manufacturers like Infineon, Rambus and Micron have shown interest in this technique.

2. *Optimal and deterministic architectures for statistics and state maintenance*

   I (along with a colleague) showed using potential functions how there is a direct relation between the optimal architectures for buffering and maintaining statistics counters [**?**]. Similarly I showed analytically how an algorithm, which gives deterministic delay bounds could be designed for maintaining connection state.

3. *A complete and flexible architecture for packet classification*

   Packet classification requirements vary widely. For example, firewalls need classification on packet headers, while an intrusion detection device requires classification of the packet content. Previous research has focused on being able to classify at very high rates. In contrast, I (along with a number of colleagues) focused on developing a classifier, which is flexible and complete i.e. it could be programmed to perform a number of classification tasks and give deterministic performance guarantees. As a first step, we identified the elementary building blocks for packet classification

in terms of an abstract language. We then designed a parallel hardware architecture to implement this language. This resulted in a commercial implementation of a chip set (presently marketed by PMC-Sierra) called ClassiPI [**?**]. Among others, the ClassiPI chip set is currently in use in Cisco's Content Services Switches.

Distributed and Greedy Algorithms for Packet Switching

Switching theory is replete with the analysis of optimal algorithms, which can give ideal performance, but have large complexity. What are of interest are practical algorithms that can be easily implemented. I answer the following open questions, which throw light on two classes of practical algorithms.

1. *Is there a distributed switching algorithm, which gives performance guarantees?*

   The crossbar is the most common switching fabric in the core of the Internet. However, the known switching algorithms required to give deterministic performance guarantees are centralized and hence have a high communication overhead. I (along with a colleague) analyzed the feasibility of distributed algorithms for a modification of the crossbar fabric called the buffered crossbar. We derive analytically using combinatorial arguments and counting techniques the conditions under which a suite of distributed algorithms can give both statistical and deterministic guarantees respectively. Since our algorithms need only local state, do not require communication with each other, and can operate independently on each input and output port, they are readily implementable. Our results show that Internet routers built using crossbars, such as Cisco routers, can be upgraded in a practical manner using our distributed algorithms on buffered crossbars and give ideal performance [**?**].

2. *When can greedy algorithms give optimal switching performance?*

   Contrary to intuition, it is known in queueing theory that a greedy switching algorithm such as the maximum size matching which maximizes the instantaneous throughput of the switch may not maximize the long-term switch throughput. Hence, greedy algorithms are not in use in practice. However greedy algorithms are of practical interest due to their low implementation complexity. I show using Lyapunov functions the conditions under which such algorithms give 100% throughput [**?**].

Network Architecture — A Research Agenda

In the course of my research, I have noticed that the overhead (in terms of size, power and cost) of designing networking components, which give performance guarantees is small. This is mainly due to two reasons. First, the inherent nature of networking makes many of these problems tractable. Second, a number of hardware advances in Architecture, insights in Algorithms *&* Combinatorics, as well as analysis techniques from Probability, *&* Queueing Theory aid in the design of elegant and simple solutions. I envisage the field of *Network Architecture* created from the ground up, building upon the foundations of a number of fields including those mentioned above.

In the near future, I am interested in the principles involved in the design of basic networking components. These include hardware components (e.g. scalable memories, network processor and co-processor architectures) and software techniques (e.g. network algorithms, packet processing techniques). Simultaneously, I intend to understand how large components, which use the above building blocks can be architected. My research will focus on how these basic and large components can be built in a scalable manner while maintaining performance guarantees. In particular, examples of large

components that I have a keen interest in are switches (e.g. packet and circuit switches, multi-service routers etc.), security devices (e.g. firewalls and intrusion detection systems), network maintenance devices (e.g. measurement, management infrastructure) and application aware devices (e.g. web server load balancers, proxies) etc.

In the future, though performance and scalability will remain key, I also intend to look at issues such as *fault tolerance, graceful degradation, reliability and uptime* of networking systems, which will become more relevant. I also believe that as systems become increasingly large and inter-dependent, *simplicity in design and component re-use* will be major factors. Parallelism can play a key role here. Indeed, many of our proposed solutions, involve component re-use and parallelism, which can aid and abet the above.

My research will involve a good mix of futuristic and present day research. One part of my work will focus on fundamentally different proposals and radical solutions. As an example — can we finally achieve real-time streaming over the Internet, assuming that the various network components give performance guarantees? In contrast, I intend to devote the other part of my work on practical systems, which have immediate relevance and impact in Industry. I intend to work closely with a number of researchers in related fields. Similarly, I intend to collaborate with Industry in understanding and developing solutions for practical problems. I believe my past experience of research work done jointly with a number of colleagues as well as my prior record of participation with Industry will help me achieve this. I am excited at the prospect of learning, contributing, giving shape and making an impact in this upcoming and challenging field.

References

[1] S. Iyer, N. McKeown, "Analysis of the Parallel Packet Switch Architecture", *IEEE/ACM Transactions on Networking*, Apr. 2003.

[2] S. Iyer, N. McKeown, "On the Speedup Required for a Multicast Parallel Packet Switch", *IEEE Communication Letters*, June 2001, vol. 5, no. 6, pp. 269-271.

[3] S. Iyer, R. Zhang, N. McKeown, "Routers with a Single Stage of Buffering", *Proceedings of ACM SIGCOMM*, Pittsburgh, Pennsylvania, Sep 2002. Also in *Computer Communication Review*, vol. 32, no. 4, Oct 2002.

[4] S. Iyer, N. McKeown, "Using Constraint Sets to Achieve Delay Bounds in CIOQ Switches", to appear in *IEEE Communication Letters*, 2003.

[5] S. Iyer, R. R. Kompella, N. McKeown, "Designing Packet Buffers for Router Line Cards". Submitted to *IEEE/ACM Transactions on Networking*. Also available as *HPNG Technical Report - TR02-HPNG-031001*, Stanford University, Mar. 2002.

[6] D. Shah, S. Iyer, B. Prabhakar, N. McKeown, "Maintaining Statistics Counters in Router Line Cards", *IEEE Micro,* Jan-Feb, 2002, pp. 76-81. Also appeared as "Analysis of a Statistics Counter Architecture" in *IEEE Hot Interconnects*, Stanford University, Aug. 2001.

[7] S. Iyer, R. R. Kompella, A. Shelat, "ClassiPI: An Architecture for Fast and Flexible Packet Classification", *IEEE NETWORK, Special Issue on Fast IP Packet Forwarding and Classification for Next Generation Internet Services*, Mar-Apr. 2001.

[8] "Attaining Statistical and Deterministic Switching Guarantees using Buffered Crossbars", with S. Chuang, N. McKeown. In preparation for *IEEE/ACM Transactions on Networking*.

[9] S. Iyer, N. McKeown, "Maximum Size Matchings and Input Queued Switches", *Proceedings of the 40th Annual Allerton Conference on Communication, Control, and Computing*, Monticello, Illinois, Oct 2002.