

A Performance Analysis Framework for Routing Lookup in Scalable Routers

Zhenhua Liu¹, Xiaoping Zhang², Youjian Zhao³, Ruisheng Wang³

*Department of Computer Science and Technology, Tsinghua University
Beijing, P. R. China, 100084*

¹liu-zh02@mails.tsinghua.edu.cn

²zhxp@tsinghua.edu.cn

³{zhaoyj, wangrs06}@csnet1.cs.tsinghua.edu.cn

Abstract— Scalable router, as an effective way to rapidly improve the performance of routers, has received increasingly attention from both academe and industry. However, the routing lookup in scalable routers is different from that in traditional routers and lack of research. In this paper, we present the first systematic study on this topic. We first discuss routing lookup in scalable routers and provide a performance analysis framework for routing lookup delay based on queueing theory. Furthermore, we provide four different routing lookup schemes (Source Oblivious Routing, Hop-by-hop Oblivious Routing, Source Adaptive Routing and Hop-by-hop Adaptive Routing) and analyze their average delays and overheads under the framework. Finally we provide simulation results.

I. INTRODUCTION

There is an urgent demand to build high-capacity routers with high scalability, throughput guarantees, and no packet reordering. However, traditional routers based on crossbar suffer from the centralized scheduling and centralized structure, which prevent them from scaling to fast line rates and high port numbers. Scalable router seems a promising way to rapidly improve the performance, especially the capacity, of routers. In scalable routers, there are many routing nodes (RNs). Packets are injected into one of these RNs, then forwarded to the correct destination RN and finally sent to the correct output port on that RN. In the paper we use the term scalable router although many other terms are also used in the literature, such as distributed router, cluster-based router or distributed routing fabric.

There are already plenty of studies on scalable router, which can be further divided into the following areas: model and architecture [1]-[6], underlying topology [7]-[11], forwarding [12]-[14] and operating system [15]-[16]. There are also paradigms in the industry, for instance, CISCO's CRS-1 [17] and AVICI TSR [18]. Readers of interests can refer to [19] for a comprehensive survey.

However, routing lookup in scalable routers is still an open problem and its characteristics still remain unclear. For tradition crossbar routers, we only need to look up the output port for the destination IP-address and then put the packet into the corresponding VOQ. However, in scalable routers, after obtaining the output port, we still need to forward the packet to the RN with the correct output port through inner routing. How to do the lookup of inner routing and what is the corresponding performance has not been studied before.

In this paper, we present the first systematic study on routing lookup in scalable routers. We first provide an overview of routing lookup in scalable routers. Based on queueing theory, we then present a performance analysis framework. Using this framework, we can do performance analysis for different lookup schemes on different topologies. In order to reveal the power of the framework, we provide four different lookup schemes and then analyze their performances with this framework.

Our contribution in this paper is two fold: First, we present the framework for performance analysis of routing lookup schemes in scalable routers. To the best of our knowledge, this is the first systematic study on this issue. Second, we provide four different routing lookup schemes and analyze their performance. These four schemes are suitable for different situations and we provide a thorough study in Section V.

The rest of the paper is organized as follows. We briefly discuss the related work in the next section. In Section III, we provide an overview of routing lookup in scalable routers. Then we present the performance analysis framework in Section IV. In Section V we propose four different lookup schemes and analyze their performance. Section VI provides simulation results and Section VII concludes the paper.

II. RELATED WORK

W. J. Dally *et. al.* [12] study the throughput and average delay performance of k-ary n-cube interconnection network. Our work focuses on routing lookup schemes in scalable routers constructed with arbitrary topology. D. Liu *et. al.* [20] provide a simple performance study of three lookup schemes in scalable routers. However, they only consider the lookup of the output port and we consider the lookup of inner routing, which is essential for scalable routers and has not been studied before. There are also a lot of studies done on scalable routers. We omit them due to space limitation.

III. ROUTING LOOKUP IN SCALABLE ROUTER

Routing lookup in traditional crossbar-based router is done as follows: the destination IP address of the incoming packet is looked up in TCAM [21]-[24] to obtain the corresponding SRAM address for the correct output port. Then the packet is placed into the corresponding VOQ and finally sent to the right output port through switching. This lookup is single-step:

finding the correct output port. However, the situation is different in scalable routers. In this section, we will first describe scalable router and then focus on the routing lookup in scalable routers.

A. Scalable Router

Scalable router is the single-image router constructed by the connection of dependent routing nodes through a certain interconnection network [19]. The topology properties of the interconnection network are essential for the performance of the corresponding scalable router. Traditional topology includes Torus, k-ary n-cube [9], H-torus [8], fully-connected mesh. Recently, we propose a novel unidirectional direct interconnection topology, named P2i, for extra-high capacity routers. We further prove in [7] that the P2i topology is suitable for load-balanced architecture [25]-[30], which is a promising way to scale core routers to extra high capacities. We now use the depiction of a scalable router based on the P2i topology to provide a picturesque view and scalable routers based on other topologies are similar.

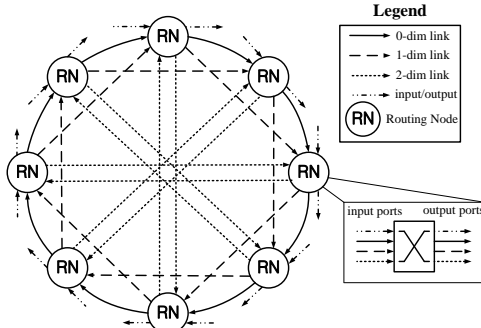


Fig. 1 A 8-node scalable router based on P2i

As illustrated in Fig. 1, in a scalable router constructed on P2i, the interconnection of routing nodes is as follows: for a P2i with N ($2^{n-1} + 1 \leq N \leq 2^n$, where n is a positive integer) RNs, for each node i , it is connected by n unidirectional links to $j = (i + 2^k) \bmod N, k = 0, 1, \dots, n-1$. We denote the link connecting node i to $i + 2^k$ by k -dim (the k -th dimension) link. Readers can refer to [7] for details. For each RN, there are inner links as well as input and output links. In order to perform switching at each RN, there exists a switching fabric, crossbar for instance. The data flow is shown in Fig. 2.

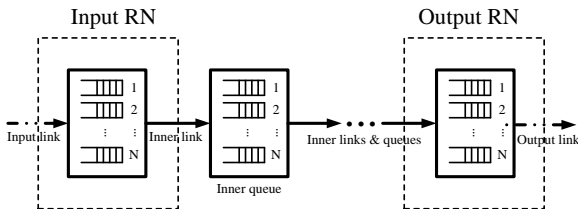


Fig. 2 Data flow in scalable routers

B. Routing Lookup in Scalable Routers

Now we focus on the routing lookup in scalable routers. In scalable router, since there are many RNs connected by an interconnection network, routing lookup is two-step: first, we

should obtain the output port and the RN on which the output port is; second, we should obtain the inner route to the destination RN. The first steps are similar to traditional switching and the second one is similar to routing.

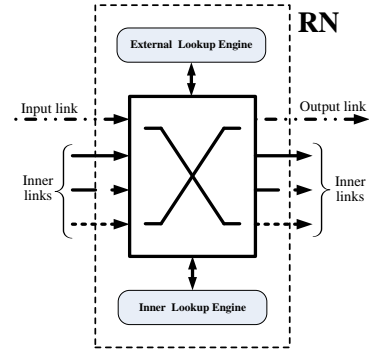


Fig. 3 Routing Lookup in Scalable Routers

The basic structure of routing lookup is shown in Fig. 3. There are two different routing lookup engines: External Lookup Engine (ELE) for the destination RN and Port by looking up in the traditional route table; and Inner Lookup Engine (INE) for the inner route to the destination RN. Since ELE is similar to the lookup engine in current routers, in this paper we focus on INE.

IV. PERFORMANCE ANALYSIS FRAMEWORK

Up to this point, we have discussed routing lookup in scalable routers. In this section, we provide the formal performance analysis framework based on queueing theory [31] and discuss the solutions based on classic theory and recent research [32]-[35].

There are four major sources contributing to packet delay: external routing lookup, inner routing lookup, queueing and other necessary operations. External routing lookup has been studied in former studies including three schemes in [20]. In this paper we focus on the delay incurred by inner routing lookup, which has not been studied ever before.

The performance analysis framework for routing lookup delay is illustrated in Fig. 4. This framework is based on queueing theory, especially queueing network. The incoming packets will have been queued in several stages. It is worth noting that there exist fake queues with infinite service rate, which means the packet jumps over this stage of queues. In addition, this framework is also suitable for the analysis of external routing lookup.

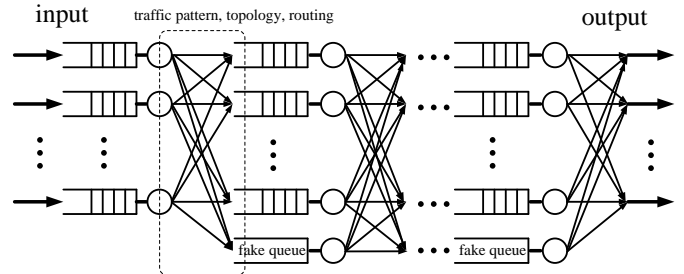


Fig. 4 Performance Analysis Framework

This framework can be used in many perspectives. Here we list some most important ones:

- (1) Different arrival processes and serving processes are expressed in the nature of corresponding queue, e.g. M/M/1, M/D/1, M/G/1 and G/D/1.
- (2) The interconnection of queues between different queues stands for the traffic load, which is co-decided by traffic pattern, topology and routing algorithm.
- (3) The routing lookup scheme decides the corresponding service time distribution of each queue.

Now we discuss the solution under such a framework. It is obviously that product-form queueing network, e.g. M/M/1, can be easily solved. However, under other service process the traffic processes will not retain their statistical properties while traversing the queueing network. Although there are many sophisticated techniques developed for this obstacle, recent researches [32]-[35] show the possibility to greatly simplify the analysis. The main idea is that if internal nodes are serving many flows, we can remove these nodes from consideration and the queueing behavior of other network nodes remains largely the same. The work in [36] further validates this approximate method.

V. LOOKUP SCHEMES AND PERFORMANCE

As stated in Section III-B, inner routing lookup in scalable routers should obtain the route to the destination RN. This can be done at either the source RN or every RN on the route. The inner routing from the source RN to the destination RN can be done by oblivious routing, which is suitable for TCAM implementation, and adaptive routing, which has better global load balancing. According to these considerations, we provide four different lookup schemes and analyze their performances under the framework of Section IV.

A. Source Oblivious Routing Lookup

In Source Oblivious Routing (SOR) Lookup, the route to the destination RN is obtained at the source RN through the lookup in TCAM. It is well-known that the delay incurred by TCAM lookup is deterministic and independent of the entries number [21], so we denote this delay by μ . Since the queues are only at the source RN, scalable routers with SOR Lookup can be considered as a single queue¹, as shown in Fig. 5.



Fig. 5 Queueing Model for SOR Lookup

Then we can employ the results of queueing theory to obtain the average delay. For Poisson arrival process with average arrival rate λ ($\rho = \lambda/\mu$), which is the M/D/1 queueing model, the average delay is

$$T_{SOR} = \frac{(2 - \rho)}{2\mu(1 - \rho)} \quad (1)$$

¹ Here we only consider *symmetric* situations where the traffic pattern and service process is identical for every RN.

If there are N RNs in the scalable router, then every RN should keep N route entries in the TCAM. The lookup result is the complete route to the destination RN, which we call label. If the average length of the route between two RNs is L and there are M ports on each RN, then the average length of a label is $L \lceil \log_2 N \rceil$ because we need $\lceil \log_2 N \rceil$ bits to denote a RN, where the $\lceil \cdot \rceil$ is the ceiling function. We also need $\lceil \log_2 M \rceil$ bits to denote the correct port on the destination RN. So the total storage is $NL \lceil \log_2 N \rceil + \lceil \log_2 M \rceil$.

Comments: The most important advantage of SOR Lookup is low average delay. However, it needs hardware support (TCAM) and label management. Because the routing algorithm is oblivious, the performance under adversary traffic pattern can be fairly poor. So this scheme is suitable for the situation where delay is the main concern.

B. Hop-by-hop Oblivious Routing Lookup

Like the Internet, in Hop-by-hop Oblivious Routing (HOR) Lookup, every RN on the route only obtains the next-hop through lookup. So there are multi-stage queues, as shown in Fig. 6.



Fig. 6 Queueing Model for THN lookup

Using the approximate method developed in [32]-[36], we can obtain the average delay is

$$T_{HOR} = \frac{L(2 - \rho)}{2\mu(1 - \rho)} \quad (2)$$

, which is L times that of SOR Lookup.

Also, if there are N RNs in the scalable router, then every RN should keep N route entries in the TCAM. The lookup result is the next-hop, instead of the whole route to the destination RN. If the average node degree D the average length of a route is $\lceil \log_2(D-1) \rceil$ because we need $\lceil \log_2(D-1) \rceil$ bits to denote the next-hop port. We also need $\lceil \log_2 M \rceil$ bits to denote the correct port on the destination RN. So the total storage is $N \lceil \log_2(D-1) \rceil + \lceil \log_2 M \rceil$, which is much smaller than that of SOR Lookup. For instance, in 2D-Torus with 256 RNs with 16 ports on each RN (4096 ports totally), the average length of the route between two RNs is $L=16$ and the node degree $D=4$, so for SOR Lookup, the storage overhead is $N \lceil \log_2 N \rceil + \lceil \log_2 M \rceil = 16388$ bits, but for HOR Lookup, the storage overhead is $N \lceil \log_2(D-1) \rceil + \lceil \log_2 M \rceil = 516$ bits, which is only 3% of that in SOR Lookup.

Comments: The most important advantage of HOR Lookup is low storage and low management overhead. Its average delay is larger than that of SOR and much smaller than that of adaptive routing implemented by software. Similar to SOR Lookup, the performance under adversary traffic pattern can be fairly poor. So this scheme is suitable for the situation where delay is the main concern and we need low storage and management overhead.

C. Source Adaptive Routing Lookup

Up to this point, we have considered the lookup schemes based on oblivious routing, which can be easily implemented by hardware. However, when traffic patterns are adversary, the performance under oblivious routing can be fairly poor. In order to achieve global load balancing under arbitrary traffic pattern, we should deploy adaptive routing, which considers the load situation when routing. This is very difficult, if possible, to be implemented by hardware. So we have to do adaptive routing by software, which will considerably increase average delay². In this part we will consider Source Adaptive Routing (SAR) Lookup and discuss the Hop-by-hop Adaptive Routing (HAR) Lookup in the next part.

In SAR Lookup, the route to the destination RN is calculated at the source RN through routing algorithm. The queueing model is similar to Fig. 5. For generalized servicing distribution with mean τ ($\rho' = \lambda\tau$) and variance σ^2 , from classic queueing theory, we can obtain the average delay for M/G/1, M/D/1 and M/M/1 queues:

$$T_{SAR} = \frac{\tau}{(1-\rho')} \left[1 - \frac{\rho'}{2} \left(1 - \frac{\sigma^2}{\tau^2} \right) \right] \quad (\text{M/G/1}) \quad (3)$$

$$T_{SAR} = \frac{\tau}{(1-\rho')} \left(1 - \frac{\rho'}{2} \right) \quad (\text{M/D/1}) \quad (4)$$

$$T_{SAR} = \frac{\tau}{(1-\rho')} \quad (\text{M/M/1}) \quad (5)$$

Since the route is obtained through calculation instead of TCAM lookup, there is no storage overhead incurred by TCAM lookup. Nevertheless, we still need labels. If the average length of the route between two RNs is L and there are M ports on each RN, then the average length of a label is $L \lceil \log_2 N \rceil + \lceil \log_2 M \rceil$.

Comments: The most important advantage of SAR Lookup, including HAR Lookup in the next part, is better global load-balancing and no hardware support. However, software-based lookup may incur much larger average delay. So this scheme is suitable for the situation where global load-balancing is demanded and large delay is acceptable.

D. Hop-by-hop Adaptive Routing Lookup

Similar to the difference between HOR Lookup and SOR Lookup, in HAR Lookup, the route is calculated per hop. The model is similar to Fig. 6. Using the approximate method developed in [32]-[36], we can obtain the average delays for exponential service time and constant service time both with mean ζ ($\rho'' = \lambda\zeta$) are

$$T_{HAR} = \frac{L\zeta}{(1-\rho'')} \left(1 - \frac{\rho''}{2} \right) \quad (\text{M/D/1}) \quad (6)$$

$$T_{HAR} = \frac{L\zeta}{(1-\rho'')} \quad (\text{M/M/1}) \quad (7)$$

Comments: HAR Lookup needs least storage overhead and no label management. Also, it can achieve best global load-balancing. But its average delay is the highest.

² High-speed hardware implementation can do one lookup in several nanoseconds, software may incur several milliseconds.

E. Comparison

The comparison of different schemes is in Table I.

TABLE I
SUMMARY OF FOUR LOOKUP SCHEMES

Scheme	Storage Overhead	Average Delay	Label
SOR	$NL \lceil \log_2 N \rceil + \lceil \log_2 M \rceil$	$\frac{(2-\rho)}{2\mu(1-\rho)}$ (M/D/1)	Yes
HOR	$N \lceil \log_2(D-1) \rceil + \lceil \log_2 M \rceil$	$\approx \frac{L(2-\rho)}{2\mu(1-\rho)}$ (M/D/1) ³	No
SAR	$L \lceil \log_2 N \rceil + \lceil \log_2 M \rceil$	$\frac{\tau}{(1-\rho')} \left[1 - \frac{\rho'}{2} \left(1 - \frac{\sigma^2}{\tau^2} \right) \right]$ (M/G/1)	Yes
		$\frac{\tau}{(1-\rho')} \left(1 - \frac{\rho'}{2} \right)$ (M/D/1)	
		$\frac{\tau}{(1-\rho')}$ (M/M/1)	
HAR	$\lceil \log_2 M \rceil$	$\approx \frac{L\zeta}{(1-\rho'')} \left(1 - \frac{\rho''}{2} \right)$ (M/D/1)	No
		$\frac{L\zeta}{(1-\rho'')}$ (M/M/1)	

VI. SIMULATION RESULTS

In this section, we provide two kinds of results. Fig. 7 shows queue depth at each hop for P2i topology, from which we can see that the depth distributions trend the same. This validates the approximate method. Fig. 8 illustrates the storage overhead of all the four schemes on three different topologies: P2i, 2D-Torus and 3D-Torus. From the results we can see: first, storage overheads from largest to the least are SOR, HOR, SAR and HAR; second, under most circumstance, P2i needs less storage.

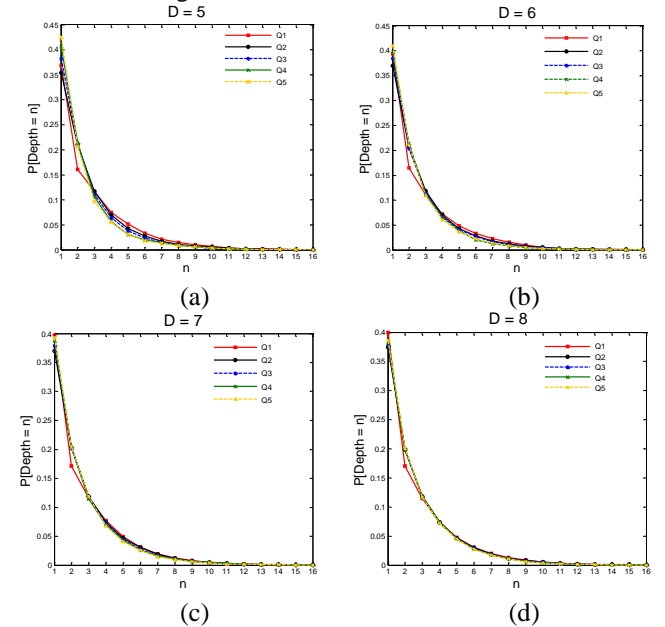


Fig. 7 Queue depth at each hop of P2i with degree D (a) $D=5$; (b) $D=6$; (c) $D=7$; (d) $D=8$

³ \approx means we use the approximate method.

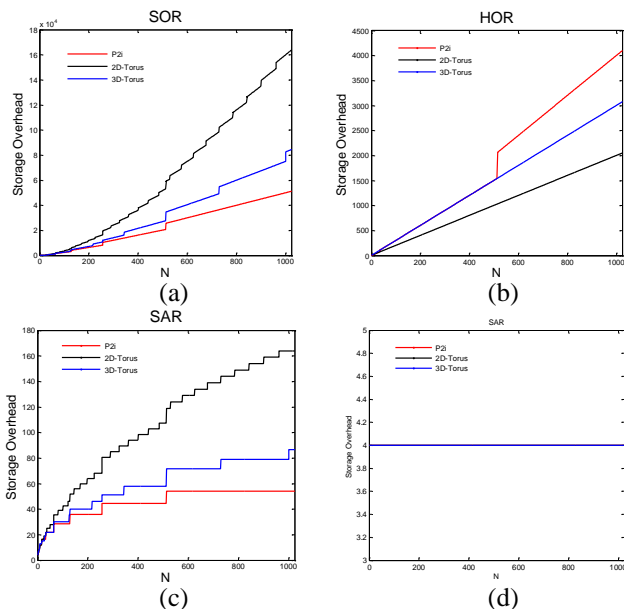


Fig. 8 Storage Overhead of (a) SOR; (b) HOR; (c) SAR; (d) HAR when $M=16$

VII. CONCLUSION

In this paper, we present the first systematic study on this topic. We provide the performance analysis framework for routing lookup delay based on queueing theory and designed four different routing lookup schemes with performance analysis under the framework. We show that different scheme suits for different situation and because in scalable routers, delay is still the main concern, we favour the SOR Lookup.

ACKNOWLEDGMENT

This research is supported by National Natural Science Foundation of China under Grant No. 90604029, 60773150 and National High Technology Research and Development Program of China under Grant No. 2007AA01Z219.

REFERENCES

- [1] M. Handley, O. Hodson, and E. Kohler. "Xorp: an open platform for network research," *ACM SIGCOMM Computer Communication Review*, vol. 33(1), pp. 53-57, Jan. 2003.
- [2] G. Welling, M. Ott, and S. Mathur. "A cluster-based active router architecture," *IEEE Micro*, vol. 21(1), pp. 16-25, Jan. 2001.
- [3] M. Ott, G. Welling, S. Mathur, D. Reininger, and R. Izmailov. "The JOURNEY active network model," *IEEE Journal on Selected Areas in Communications*, vol. 19(3), pp. 527-537, Mar. 2001.
- [4] S. Karlin, and L. Peterson. "VERA: an extensible router architecture," *Computer Networks*, vol. 38(3), pp. 277-293, Feb. 2002.
- [5] J. Biswas, A. Lazar, J. Huard, and K. Lim. "The IEEE P1520 standards initiative for programmable network interfaces," *IEEE Communications Special Issue on Programmable Networks*, vol. 36(10), pp. 64-70, Oct. 1998.
- [6] A. Doria. *ForCES Protocol Specification*, IETF Draft, 2007.
- [7] Z. Liu, X. Zhang, Y. Zhao, and H. Guan. "An asymptotically Minimal Node-degree Topology for Load-Balanced Architectures," to appear in *IEEE GLOBECOM 2008*.
- [8] Y. Zhao, Z. Yue, J. Wu, and X. Zhang. "Topological Properties and Routing Algorithms in Cellular Router," in *Proc. ICNS'06*, 2006, p. 101.

- [9] W. Dally, and B. Towles. *Principles and Practices of Interconnection Networks*. Morgan Kaufmann, 2004.
- [10] A. Jajszczyk. "Nonblocking, repackable, and rearrangeable Clos networks: fifty years of the theory evolution," *IEEE Communications Magazine*, vol. 41(10), pp. 28-33, Oct. 2003.
- [11] G. Sapountzis, and M. Katevenis. "Benes switching fabrics with $O(N)$ -complexity internal backpressure," *IEEE Communications Magazine*, vol. 43(1), pp. 88-94, 2005.
- [12] W. Dally. "Performance analysis of k-ary n-cube interconnection networks," *IEEE Transactions on Computers*, vol. 39(6), pp. 775-785, Jun. 1990.
- [13] K. Zheng, C. Hu, H. Lu, and B. Liu. "A TCAM-based distributed parallel IP lookup scheme and performance analysis," *IEEE/ACM Transactions on Networking*, vol. 14(4), pp. 863-875, Apr. 2006.
- [14] X. Zhang, N. Zhang, J. Wu, and Y. Zhao. "BPA - a parallel shortest path algorithm for cluster-router," in *Proc. PDCS'06*, 2006.
- [15] K. Xu, J. Wu, Z. Yu, and M. Xu. "HEROS: router-oriented distributed real-time operating system," *Journal of Software*, vol. 42(1), pp. 52-55, Jan. 2002.
- [16] H. Chan, H. Alnuweiri, and V. Leung. "A Framework for Optimizing the Cost and Performance of Next-Generation IP Routers," *IEEE Journal on Selected Areas in Communications*, vol. 17(6), pp. 1013-1029, Jun. 1999.
- [17] (2006) Cisco CRS-1 Series Carrier Routing System Getting Started Guide. [Online]. Available: <http://www.cisco.com>.
- [18] W. Dally, P. Carvey, and L. Dennison. "The Avici terabit switch/router," in *Proc. Hot Interconnects*, 1998, p. 41-50.
- [19] X. Zhang, Z. Liu, Y. Zhao, and H. Guan. "Scalable router," *Journal of Software*, vol. 19(6), pp. 1452-1464, Jun. 2008.
- [20] D. Liu, Y. Zhao, and X. Zhang. "Performance analysis and comparison of lookup architecture in cluster-based scalable router," in *Proc. PDPTA'06*, 2006.
- [21] T. Pei, and C. Zukowski. "Putting routing tables in silicon," *IEEE Network Magazine*, vol. 6(1), pp. 42-50, Jan. 1992.
- [22] A. McAuley, and P. Francis. "Fast routing table lookup using CAMs," in *Proc. IEEE INFOCOM '93*, 1993, p. 1382-1391.
- [23] P. Lekkass. *Network Processors - Architectures, Protocols, and Platforms*. McGraw Hill, 2004.
- [24] V. Srinivasan, B. Nataraj, and S. Khanna. "Methods for longest prefix matching in a content addressable memory," US Patent 6 237 061, May 1, 1999.
- [25] L. Valiant, and G. Brebner. "Universal schemes for parallel communication," in *Proc. SOTC*, 1981, p. 263-277.
- [26] C. Chang, D. Lee, and Y. Jou. "Load balanced Birkhoff-von Neumann switches, Part I: one-stage buffering," *Computer Communications*, vol. 25(6), pp. 611-622, Apr. 2002.
- [27] C. Chang, D. Lee, and C. Lien. "Load balanced Birkhoff-von Neumann switches, Part II: multi-stage buffering," *Computer Communications*, vol. 25(6), pp. 623-634, Apr. 2002.
- [28] I. Keslassy, S. Chuang, K. Yu, D. Miller, M. Horowitz, O. Solgaard, and N. McKeown. "Scaling Internet routers using optics," in *Proc. ACM SIGCOMM '03*, 2003.
- [29] B. Lin, and I. Keslassy. "The concurrent matching switch architecture," in *Proc. IEEE INFOCOM '06*, 2006.
- [30] C. Yu, C. Chang, and D. Lee. "CR switch: a load-balanced switch with contention and reservation," in *Proc. IEEE INFOCOM '07*, 2007.
- [31] L. Kleinrock. *Queueing Systems: Volume II*. John Wiley, 1975.
- [32] D. Wischik. "The output of a switch, or, effective bandwidths for networks," *Queueing Systems - Theory and Applications*, vol. 32(4), pp. 383-396, Nov. 1999.
- [33] D. Wischik. "Sample path large deviations for queues with many inputs," *Annals of Applied Probability*, vol. 11(2), pp. 379-404, May 2001.
- [34] D. Eun, and N. Shroff. "Simplification of network analysis in large bandwidth systems," in *Proc. IEEE INFOCOM '03*, 2003.
- [35] D. Eun, and N. Shroff. "Network decomposition in the many-sources regime," *Advances in Applied Probability*, vol. 36(3), pp. 893-918, September 2004.
- [36] A. Singh, "Load-balanced routing in interconnection networks," *Ph.D. thesis*, Stanford University, Stanford, CA, Mar. 2005.