

CR:基于一类新型结构的可扩展路由器*

乐祖晖⁺, 吴建平, 赵有健

(清华大学 计算机科学与技术系,北京 100084)

CR: Scalable Routers Based on a New Architecture

YUE Zu-Hui⁺, WU Jian-Ping, ZHAO You-Jian

(Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China)

+ Corresponding author: Phn: +86-10-62795818 ext 6854, E-mail: yuezuhui@csnet1.cs.tsinghua.edu.cn

Yue ZH, Wu JP, Zhao YJ. CR: Scalable routers based on a new architecture. *Journal of Software*, 2007, 18(10):2624-2634. <http://www.jos.org.cn/1000-9825/18/2624.htm>

Abstract: The switching fabric in the Avici TSR uses a 3-D torus topology with each line card carrying one node of the torus. It requires no centralized switching fabric, whereas its scalability is limited by the bisection bandwidth. This paper proposes a novel architecture called Cellular Router (CR). There exist some problems with the basic CR architecture. They are solved by introducing Mirror Points (MPs). This paper also gives the design of line cards in this architecture. In the end, the design of routing algorithms is introduced on this architecture. The CR architecture shows excellent scalability and fault tolerance. It is a promising choice for the design of scalable routers.

Key words: scalable routers; cellular router; cobweb; switch fabric; interconnection network

摘要: 互连网络,例如 3-D torus 拓扑结构,已成功应用于可扩展路由器的设计中.但是,3-D torus 结构在实际应用时存在设计缺陷,扩展规模受到了限制.提出了一类新型的交换架构,称为蜂巢式路由器.基本蜂巢结构存在一些问题,通过引入镜像点可以有效解决,还给出线卡的具体设计方案.最后介绍了该架构下几类路由算法的设计.蜂巢结构表现出优秀的扩展能力和容错性,非常适合可扩展路由器的设计.

关键词: 可扩展路由器;蜂巢式路由器;蛛网;交换网络;互连网络

中图法分类号: TP393 文献标识码: A

1 Introduction

Routers play very important roles in the Internet. Nowadays most routers consist of line cards and centralized switching fabrics. The continual growth of user traffic requires a corresponding increase in the router capacity. To meet this demand, switching fabrics are required to support both faster ports and have a very large number of ports. However there exist a lot of problems^[1] that limit the increase of port-rate. So focus has been moved to support more ports.

* Supported by the National Natural Science Foundation of China under Grant No.90604029 (国家自然科学基金); the National Basic Research Program of China under Grant No.2003CB314801 (国家重点基础研究发展计划(973))

Received 2005-12-13; Accepted 2006-04-10

Historically, switching fabrics based on *buses* and *crossbar* switches were widely used in routers. It is well known that buses cannot be scaled to support high bit rates for the limited bandwidth. For low numbers of ports, crossbar is often selected as the switch topology, owing to the simplicity and non-blocking properties. However, its cost grows as the square of the number of ports, and cannot be economically scaled to a large number of ports. Multistage switching fabric architectures can handle modest or large numbers of ports. Researchers have been looking at such topologies since the days of electromechanical telephony^[2]. The *Banyan* network^[3] has a low cost of $N \cdot \log N$ (N is the number of ports) and a lot of paths. But it suffers from internal blocking. The *Benes* network^[4] features a low cost of $N \cdot 2 \log N$ and is free of internal blocking. The Benes network is rearrangeably non-blocking, and setting up new connections may destroy the existing ones. The *Clos* network^[5] is useful in practice and stimulating in research. There are still some unsolved theoretical problems with Clos network. The *Load-Balanced* switch architecture^[6] is considered promising for this approach eliminates the centralized scheduler and can be realized with optics^[7]. However the router built with this switch architecture cannot be built with technology available today.

There exists a centralized switching fabric in the router implemented with the switches listed above. This greatly limits the scalability of the router and the centralized switch becomes the Single Point of Failure (SPF). Most of them need complicated schedulers. For example, Maximum Weight Matching (MWM) schedulers, such as LQF and OCF^[8], can achieve 100% throughput asymptotically under any admissible workload, uniform or not, with no speedup needed. However, they have a high computational complexity of $O(N^3)$, hence are infeasible in practice. Although some other algorithms have less computational complexity, they are related to the number of ports. That is, as the number of ports increases the time slots used increase nonlinearly.

Interconnection networks are originally used as switch for processor-memory interconnect and for I/O interconnect. Afterwards interconnection networks based on 3-D torus topology^[9] are used as router fabrics in the *Avici TSR*^[10]. In the Avici TSR switching fabric, each line card carries one node of the torus. There are many optional paths between source node and destination node. This design offers some good properties^[10]: Economical scalability, incremental extensibility, load balance, fault tolerance, non-blocking and low bounded delay for CBR traffic. Although 3-D torus topology shows good scalability, its implementation in TSR has limited the number of line cards, which can be only added to 560^[10]. More line cards cannot improve the bisection width of TSR.

The Cellular Router (CR) introduced in this paper is a new architecture that can be used to scale routers. The number of line cards is not limited. There are more optional paths between source-destination pair. With some modifications, this architecture provides better fault tolerance than 3-D torus topology.

The rest of this paper is organized as follows. Section 2 presents the basic CR architecture. In Section 3, we give a brief introduction of the line card design. The basic CR architecture shows poor fault tolerance and some improvements will be described in Section 4. Section 5 introduces several routing algorithms that can be applied to the CR architecture. Section 6 presents a simulation system and some results can be obtained from this system. Section 7 provides a summary of our work and talks about the future work.

2 Introduction of the Basic CR Architecture

In this section, we will introduce the basic CR architecture. This architecture is motivated by some observations and shows good scalability. When we build scalable routers with this architecture, all the active components of the fabric are carried on the line cards and the line cards are connected only with data channels.

2.1 Motivations

The basic CR architecture is motivated by the following three observations.

Observation 1. There are only three types of regular polygons that can cover the whole plane seamlessly as shown in Fig.1: *Equilateral triangles, squares and regular hexagons.*

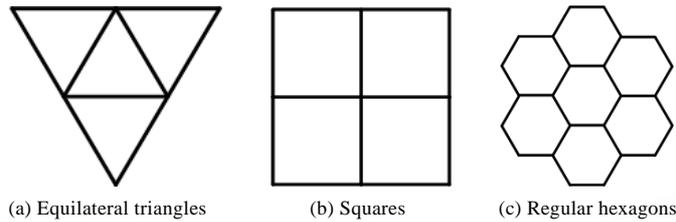


Fig.1 Three types of regular polygons

When we want to build a router with interconnection network, we must first choose the topology. During the deployment of router racks, it is economically meaningful to make full use of the limited space. It can be easily proven that there are only three types of regular polygons that can cover the whole place seamlessly: Equilateral triangles, squares and regular hexagons. In the 3-D torus topology, square is adopted as the atomic structure. The equilateral triangles case shown in Fig.1(a) is a good choice for its connectivity. However, triangles are not well organized and become much more disorder as the number of nodes increases. The regular hexagons shown in Fig.1(c) are well organized and show good scalability, which we will show soon.

Observation 2. If we introduce a central point in each regular hexagon as showed in Fig.2(a) and connect it with the other vertexes in the hexagon, the modified regular hexagons can cover the other two equilateral cases.

Figure 2(a) shows the modified regular hexagons. In fact, Fig.2(b) shows just another organization of Fig.1(a). However, these equilateral triangles are well organized. If we only make use of some links as shown in Fig.2(c), the modified hexagons can cover the squares case.

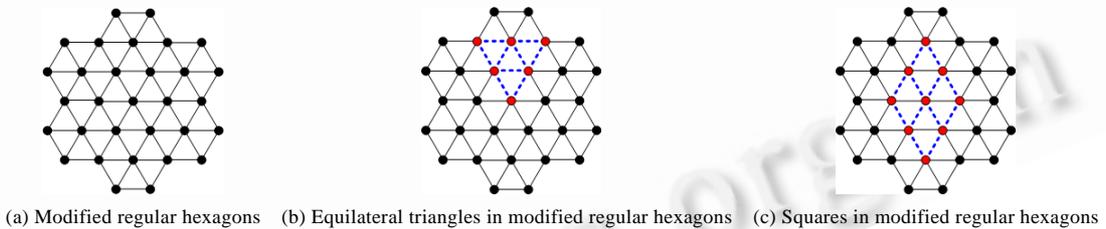


Fig.2 The modified regular hexagons can cover the other two cases

Observation 3. The cost to build a line card is much more expensive than that of a data link, especially when the links are realized with backplane wiring.

In an interconnection network, if we increase the number of links, the degree of each node will increase accordingly. This means that there will be more channels between nodes. This can improve the fault tolerance of the network. In the mean time, more paths can also be used to realize good load balance. If we only consider the 2-D case, the degree of a fully connected node in Fig.1(a), Fig.1(b) and Fig.2(a) is 6, 4 and 6 respectively. The CR architecture provides more links between nodes (line cards) than that of the 2-D torus architecture.

2.2 The basic CR architecture

Now we introduce the basic CR architecture. As line cards are plugged in, they will appear in only one layer in a single rack, several layers in a single rack or several different racks. Here we will show all these three cases.

2.2.1 One layer in a single rack

We first consider the case in which there is only one layer in a single rack. According to the modified hexagon,

we can place 7 line cards in one layer (in this layer we can also place some control cards or some redundant line cards, so it is reasonable to place a hexagon in one layer). As shown in Fig.3(a), we number these line cards from 0 to 6. These numbers represent the sequence in which the line cards are plugged into the rack. When a new line card is added, we will also add the related link(s) with the existing node(s). Each link is unidirectional, so there are two separate links between any adjacent nodes—One for each direction. We define such a hexagon as a *Basic Element* (BE). The numbers shown in Fig.3(a) are defined as *BE Numbers* (BENs).

In practice, the line cards are always placed parallel. So we can reorganize the line cards in one BE as shown in Fig.3(b), in which arrowed solid lines represent the data channels implemented on the backplane with equal length.

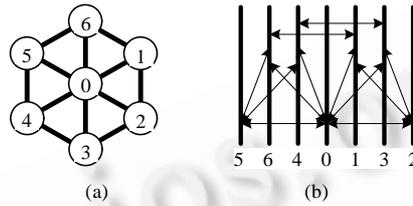


Fig.3 Line cards placed parallel in one BE

2.2.2 Multi-Layer in a single rack

We can place one BE in each layer and connect the corresponding line cards as shown in Fig.4(a).

The maximum degree of each line card is 8 (here we consider the 3-D case). We reserve two of them to connect line cards of adjacent layers. In Fig.4(a), to increase the connectivity of the nodes in the topmost layer and the bottommost one, we can connect the nodes in the topmost layer with the corresponding ones in the bottommost layer (these links are not shown in Fig.4(a)). We can find that in the 3-D case there exist two types of polygons: Squares and regular hexagons.

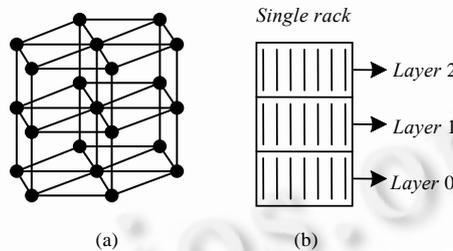


Fig.4 Multilayer in a single rack

2.2.3 Multi-Rack system

Each generation of routers consumes more power than the last, and it is now difficult to package a router in only one rack of equipment. There has therefore been a move towards multi-rack systems. We can see that CR shows good scalability in a multi-rack system.

First, we show how to place the racks. In Fig.5, each node represents a rack with one layer or multiple layers. We define the central point as *Polar Point* (PP). There are totally six axes that start from the PP. They are defined as 0-axis, $-\pi/3$ -axis, $-2\pi/3$ -axis, π -axis, $2\pi/3$ -axis and $\pi/3$ -axis respectively. These six axes divide the whole plane into six equal-area regions, and we define these regions as *region0*, *region1*, *region2*, *region3*, *region4* and *region5*. We first place one rack on the PP. Then we place the second rack on the $\pi/3$ -axis. After we have finished placing six racks around the PP clockwise, we place other racks around these six racks. In Fig.5, the arrowed solid line starting from the PP shows the order in which the racks are deployed in such a multi-rack system. We define the PP as

circle0 and the six racks around it as *circle1*, etc. We call this architecture as *cobweb* architecture. The sequence number the rack appears in Cobweb is defined as the Cobweb Number (CN). The cobweb architecture can effectively minimize the diameter of the multi-rack system. The central office where routers are placed is very expensive. So this architecture can make full use of the limited place. At the same time, shorter diameter can also reduce the latency of cells in such an interconnection network.

In the Avici TSR^[10], each line card is assigned a three-coordinate address (x,y,z) . The five line cards in the z -dimension are organized into a group named quadrant. The line cards in the y -dimension are organized into two rows of racks. When line cards are added in the x -dimension, the network bisection bandwidth in such a layout remains constant. To keep up the speedup of the network, the scale cannot be increased when the number of line cards reaches 560. While in a cobweb architecture, even when the number of line cards in the z -dimension remains constant; we can increase the network bisection bandwidth by place rack in the order shown in Fig.5.

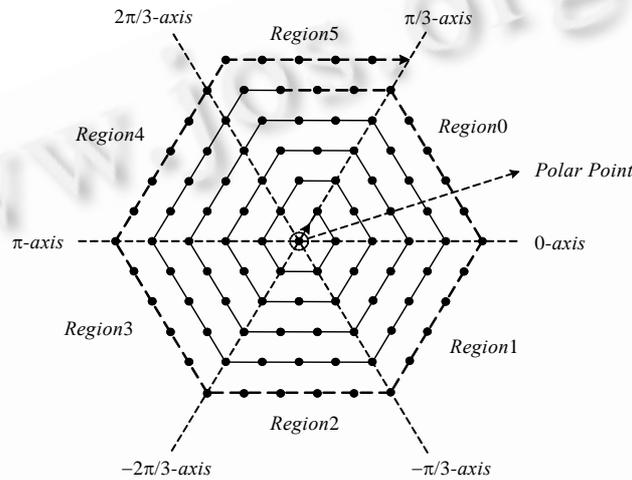


Fig.5 Cobweb architecture

3 Design of Line Cards

Unlike the architectures of most other routers, where dedicated switching fabrics are need, CR architecture distributes the switching task into each line card. This indicates that each line card should handle not only its own traffic but also the traffic from the neighbor line cards. As shown in Fig.6, the input traffic could arrive from the eight ingress channels and the ingress link of its own. This input traffic is then distributed to the different output queues. The inner switching fabric can be normal crossbar. We can introduce separated queues, named Virtual Output Queue (VOQ), for each output. Unlike the traditional switching fabrics, the inner switching fabric need not be changed as the number of line cards increases. That is, the computational complexity of the scheduling algorithm remains constant. Line cards in a same rack can be connected with backplane wires. Line cards in different racks can be connected with either extended backplane wires or optical fibers.

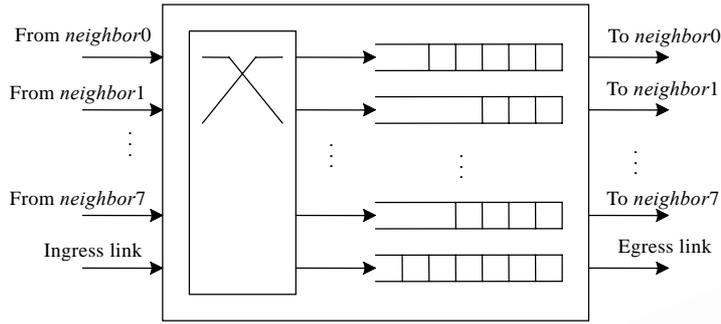


Fig.6 The line card structure in CR architecture

4 Improvements of the Basic CR Architecture

4.1 Problems with the basic CR architecture

Here we first discuss the BE again. As shown in Fig.7(b), it can be found that if the central point (line card) is down, the BE will change to a dual-ring (for there are two unidirectional links between any adjacent nodes). If one node in the remainders is down again, the dual-ring will turn to a chain. If another node breaks down, the left nodes become unconnected as shown in Fig.7(d). So the fault tolerance is poor in the BE. It is known that the maximum degree of each line card is 6 excluding the two reserved links used for adjacent layers. However, in the BE, the degree of each node, except for the central point, is only 3. When some nodes fail, the left nodes are affected greatly.

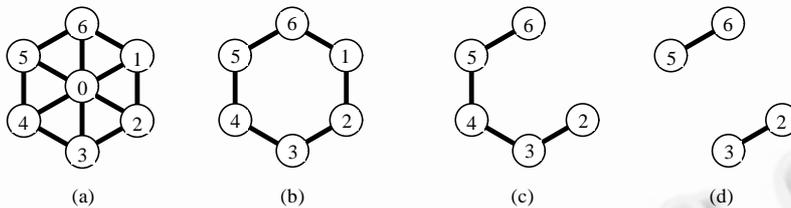


Fig.7 Node failure in the BE

4.2 Improvements of the basic CR architecture

To improve the basic CR architecture, we design two schemes: Largest Number First and Mirror Points. They can improve the connectivity of edge nodes and reduce the diameter.

4.2.1 Largest number first

We define the set Γ , which consists of nodes with Linked Degree (LD) of less than 6. We also define the set Λ , which consists of nodes with LD of 6. In the BE, from Γ we choose the node with the largest number, node6. The Unlinked Degree (UD) of node6 is 3. This means that we can connect node6 with 3 nodes other than node0, node1 and node5. We choose from Γ the nodes that have not been connected with node6. In this case, we connect node6 with node2, node3 and node4. Now LD of node6 is 6, so we move it from Γ to Λ . Then we connect node5 with node1, node2 and node3. Next time we connect node4 with node1 and node2. Now Γ is empty and all the nodes have been moved to Λ .

In general, we can design an algorithm—Largest Number First. With this algorithm, we choose the largest line card in Γ . Then we choose from Γ the nodes, which evenly distribute on the boundary. If the LD of any node reaches 6, we move it to Λ . We do the same procedure until Γ is empty or the graph is already fully connected. This algorithm also applies to irregular CR topology. In a fully-connected graph with n nodes, there are $n(n-1)/2$ links. In

a modified CR architecture with n nodes, there are $3n$ links. When n is less than 8, the modified CR architecture is fully connected.

4.2.2 Mirror points

First we introduce the concept of *Envelope*. Envelope is the regular hexagon that can encapsulate the CR architecture. If no nodes appear on the envelope, we define it as *External Envelope*. If some nodes appear on the envelope, we define it as *Internal Envelope*. As shown in Fig.8(a), the dashed regular hexagon is the External Envelope of the BE. In Fig.8(a), *node4'* on the External Envelope is defined as the Mirror Point (MP) of node4 related to node0. Node4' can be connected to node6, node1 and node2. That is, in a practical system we will connect *node4* with *node6*, *node1* and *node2*. We can also connect *node1* to *node3*, *node4* and *node5* with the help of MP, etc., as shown in Fig.8(b). In the end we get a fully-connected system as shown in Fig.8(c).

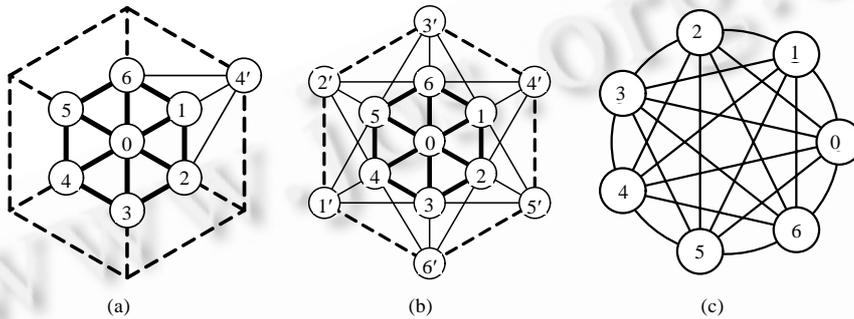


Fig.8 Mirror points

5 Minimal Routing Algorithms

Before discussing the minimal algorithms, we first introduce some basic concepts.

5.1 Preliminaries

The following discussions describe the routing algorithms that are *minimal*. That is, they select the shortest path among all the optional paths. We restrict our discussion to 2-D CR architecture and ignore the impact of *mirror points*. Each link is unidirectional, and we define the length of each link as 1.

Suppose the source node is s , and the destination node is d . When cells are forwarded from s to d , they pass a series of intermediate nodes: $J=(j_1, j_2, \dots, j_n)$. We define D_{sd} , the distance between s and d , as the length of the shortest path between them. For the shortest paths, we get $D_{sd} > D_{j_1d} > D_{j_2d} > \dots > D_{j_nd} = 1$.

We denote the six neighbors of s as n_0, n_1, n_2, n_3, n_4 and n_5 . If we connect s and d , there exist two cases: d appears on the extended line connecting s and one of its neighbors n_i ; d appears between two extended lines. When we choose the next-hop node on the shortest path, n_2 is selected in Fig.9(a) while in Fig.9(b) we can choose one between n_2 and n_3 . We call n_2 as the Right Neighbor (RN) of s and n_3 as the Left Neighbor (LN) of s for source destination pair (s, d) . With the minimal routing algorithms, we can always choose the RN of the current node where the cell locates. We call this algorithm as Right Neighbor First (RNF) algorithm. In the same way, we can design Left Neighbor First (LNF) algorithm. Although these two algorithms only provide a single path between source node and destination node, they are very simple and inexpensive to implement in hardware.

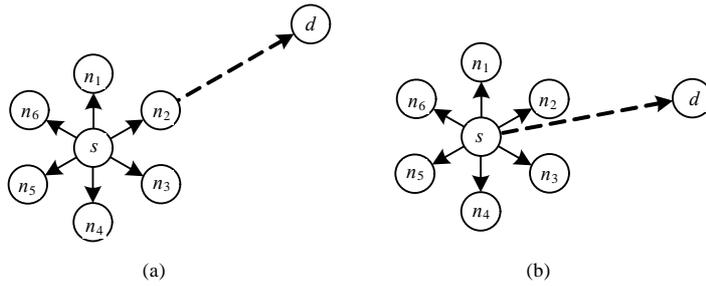


Fig.9 Relationship between node s and d

5.2 Identification of each node

To find the next-hop node on the shortest path, we should first identify each node in the architecture. There are two methods to achieve this: method based on the PP and method based on the source node.

5.2.1 Method based on the PP

According to the characteristic of the CR architecture, it is proper to identify each node under polar coordinate. We choose the 0-axis as polar axis OX . So node P can be identified as (ρ, θ) . Here ρ represents the length of OP , and θ is the value of $\angle POX$. In Fig.10, $tg\theta = |PN|/|ON| = (PM \cdot \sin\pi/3) / (OM - PM \cdot \cos\pi/3) = (3 \cdot \sin\pi/3) / (5 - 3 \cdot \cos\pi/3)$. In this way, we can easily calculate the values of ρ and θ for each node.

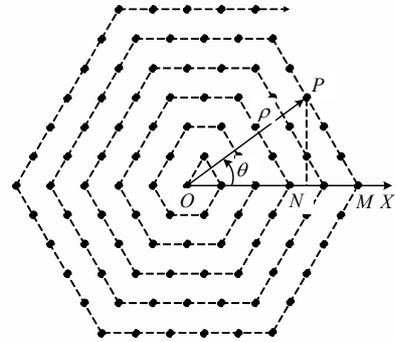


Fig.10 Polar coordinate

To find the next-hop node on the shortest path, we should find the association between $s \rightarrow d$ and $s \rightarrow n_0, \dots, s \rightarrow n_5$. In Fig.11, we can calculate the values of ρ_s, θ_s, ρ_d and θ_d . Then we can calculate the value of θ'_d according to these values. In the end we can choose the next-hop node in the way discussed in the above section.

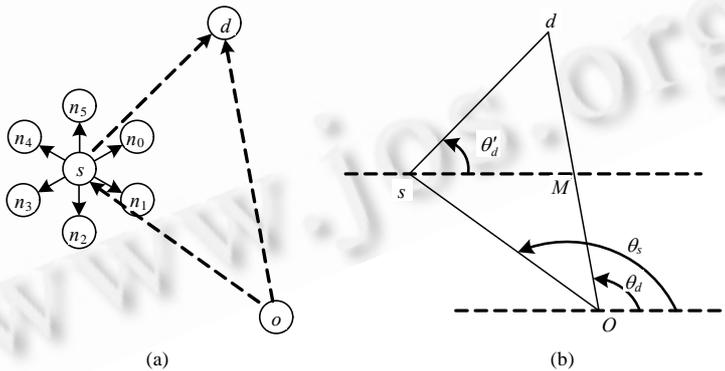


Fig.11 Calculation of θ'_d

5.2.2 Method based on the source node

It is easy to calculate the coordinate of each node with the method based on the PP. However, the relationship between s and d should be calculated again to determine the next-hop node. If we set up the coordinate of each node on the source node, the speed of switching will be improved greatly. All the calculation can be carried out offline. That is, we can get these values before switching.

6 Performance Analysis

We develop a simulation system to evaluate the performance of the CR architecture. There are two common approaches for the design of the simulation system: *cycle-based* and *event-driven*. We choose the former. In each time slot, one or zero cell arrives at the node and is put into one of the VOQs according to its source node and destination node (the source and destination addresses of each cell are translated to the node identifiers in the CR).

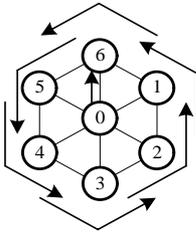


Fig.12 Tornado traffic in the BE

We choose the BE as our target and *RNF* as the routing algorithm. We choose two traffic pattern, *uniform random traffic* and *tornado traffic*. With *uniform random traffic*, each node in the BE randomly chooses one destination node for each incoming cell. With *tornado traffic*, *node0* sends cells to *node6*, *node1* sends cells to *node5*, *node6* sends cells to *node4*, etc., as shown in Fig.12. We assume that at each time slot a cell arrives at each node.

6.1 Simulator warm-up

Our simulator is initialized with empty queues before any cells are injected. This will introduce a systematic error into our measurement. For cells that are injected early in the nodes, we can see a relatively empty network. These cells have less contention and therefore traverse the network more quickly. However, as buffers begin to fill up later cells meet more contention, increasing their latencies. Over time the influence of the initialization becomes minimal, and at this point the simulation is *warmed up*. By ignoring all the events that happen before the warm-up point, the impact of systematic error on measurements can be minimized.

Here the length of each queue is set as 100. The random traffic is chosen. We run our simulator for 10000 time slots. We calculate the ratio of the dropped cells to input cells. Figure 13(a) shows the statistics in 1000 time slots and Fig.13(b) shows the result in 10 000 time slots. We can find that Fig.13(a) is only a snippet of Fig.13(b), and Fig.13(b) can give us more comprehensive scene. There is no a universal method for determining the length of the warm-up period. In the following parts, we choose 10 000 time slots to analyze the BE architecture.

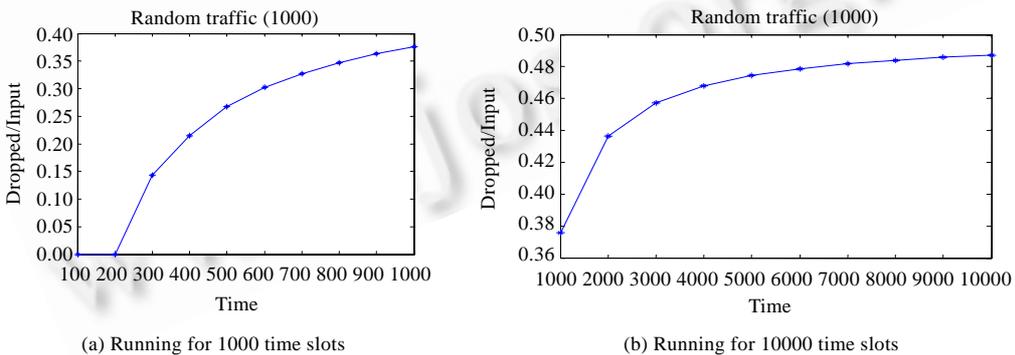


Fig.13 Simulator runs for different time slots

6.2 Throughput

Here we set the length of each queue as 100, we run our simulator for 10 000 time slots. We compare the throughput of tornado traffic with that of random traffic. As shown in Fig.14(a), RNF algorithm performs well with random traffic, while it performs poorly with tornado traffic. With tornado traffic pattern, RNF algorithm results in

a considerable load imbalance. The counterclockwise links are fully loaded while the clockwise links are idle. So the throughput remains about 0.5 when the system is steady. If we distribute some load to the clockwise links, we can get a better result. The random traffic pattern is benign for RNF algorithm since the load is more balanceable.

When designing a routing algorithm, we always assume that each node has buffers of infinite length. However in practice, this can't be true. We choose tornado traffic pattern, and set the length of each queue as 20, 40, 60, 80 and 100 respectively. We run our program for 10 000 time slots.

Figure 14(b) shows the throughput with different queue lengths. The influence of queue length is obvious in early time, and becomes more and more inconspicuous over time. For cells injected early in the nodes, we can see the relatively empty queues. These cells can be stored in the queues waiting for their turn. However, as queues begin to fill up later cells might be dropped for there is no room for them. Over time the influence of the queue length becomes minimal. So it is not a good idea to reduce the dropped cells only by increasing the capacity of buffers. It is more important to design a good routing algorithm.

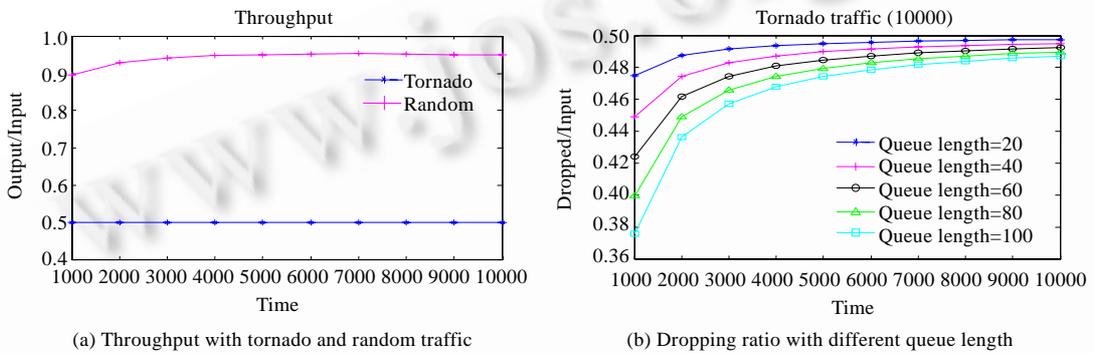


Fig.14 Performance analysis on BE

6.3 Speedup

This time we choose tornado traffic pattern again, and set the length of each queue as 100. We set the speedup of the switching fabrics as 1.0, 1.2, 1.5, 1.8 and 2.0 respectively. As the speedup increases, the number of the dropped cells is greatly reduced. A speedup of 2.0 is enough for the RNF routing algorithm with tornado traffic pattern as shown in Fig.15.

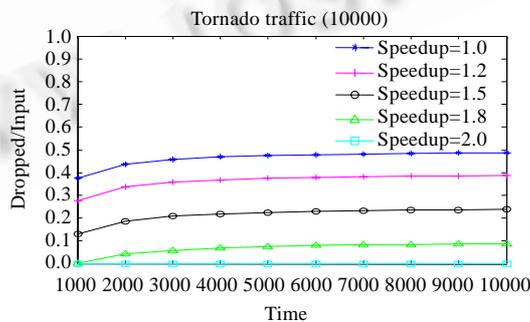


Fig.15 Dropping ratio with different speedup

7 Conclusion

As we mentioned at the very beginning of this paper, the driving forces for the evolution of router design is the

stupendous growth of user traffic. It becomes more and more difficult to increase the speed of ports because we are encountering not only some intrinsic limitations of silicon technology but also a whole set of physical, electrical and mechanical issues. To support a very large number of ports, traditional switching fabrics are not suitable for complicated schedulers. 3-D torus topology used in Avici TSR gives us a pleasant surprise for its simplicity and good scalability. However, TSR can only support up to 560 line cards for its physical limitation.

The CR architecture is a new architecture. It is promising and suitable for scalable routers. In theory, this architecture can be scaled from a system with only one line card to a system with an infinite number of line cards. With some improvements, this architecture works well even when some nodes or links are down.

In future, we plan to go deep into the properties of the regular and irregular CR architectures. We plan to develop routing algorithms, which will be applied to normal operations, and some other algorithms that can be used in case of faults. With this architecture, we also can make some research on multicast or QoS switching.

Acknowledgement YUE Zu-Hui would like to thank Drs. Zhang Xiao-Ping, XU Ke and CUI Yong for their good ideas.

References:

- [1] Chiussi FM, Francini A. Scalable electronic packet switches. *IEEE Journal on Selected Areas in Communications*, 2003,21(4): 486–500.
- [2] Marcus MJ. The theory of connecting networks and their complexity: A review. *IEEE Proc.*, 1977,65(9):1263–1271.
- [3] Narasimha MJ. The batcher-banyan self-routing network: Universality and simplification. *IEEE Trans. on Communications*, 1988, 36(10):1175–1178.
- [4] Sapountzis G, Katevenis M. Benes switching fabrics with $O(N)$ -complexity internal backpressure. *IEEE Communications Magazine*, 2005,43(1):88–94.
- [5] Jajszczyk A. Nonblocking, repackable, and rearrangeable clos networks: Fifty years of the theory evolution. *IEEE Communications Magazine*, 2003,41(10):28–33.
- [6] Chang CS, Lee DS, Jou YS. Load balanced Birkhoff–von Neumann switches, part I: One-Stage buffering. *Computer Communications*, 2002,25(6):611–622.
- [7] Keslassy I, Chuang ST, Yu K, Miller D, Horowitz M, Solgaard O, McKeown N. Scaling Internet routers using optics. In: *Proc. of the Special Interest Group on Data Communication (SIGCOMM)*. ACM Press, 2003. 189–200.
- [8] McKeown N, Mekkittikul A, Anantharam V, Walrand J. Achieving 100% throughput in an input-queued switch. *IEEE Trans. on Communications*, 1999,47(8):1260–1267.
- [9] Dally WJ. Performance analysis of k -ary n -cube interconnection networks. *IEEE Trans. on Computers*, 1990,39(6):775–785.
- [10] Dally WJ. Scalable switching fabrics for Internet routers. <http://www.avici.com/technology/whitepapers/TSRfabric-WhitePaper.pdf>



YUE Zu-Hui was born in 1978, Ph.D. candidate. His current research areas are computer networks, hardware architecture of high-speed scalable routers, switching fabrics, scheduling algorithms, routing algorithms.



ZHAO You-Jian was born in 1969, Associate professor, a CCF senior member. His research areas are computer networks, hardware architecture of high-speed scalable routers, switching fabrics, scheduling algorithms.



WU Jian-Ping was born in 1953, Professor, a CCF senior member. His research areas are the next generation of computer networks, formal methods and protocol engineering on computer networks.