# Network within a Network Approach to Create a Scalable High-Radix Router Microarchitecture

Jung Ho Ahn
*Dept. of Intelligent Convergence Systems*
*Seoul National University*
gajh@snu.ac.kr

Sungwoo Choo
*Dept. of Intelligent Convergence Systems*
*Seoul National University*
choos@snu.ac.kr

John Kim
*Dept. of Computer Science & Web Science Technology Division*
*KAIST*
jjk12@kaist.edu

## Abstract

*Cost-efficient networks are critical in creating scalable large-scale systems, including those found in supercomputers and datacenters.* High-*radix routers reduce network cost by lowering the network diameter while providing a high bisection bandwidth and path diversity. However, as the port count increases, the high-radix router microarchitecture needs to scale efficiently. Hierarchical crossbar organization has been proposed where a single large crossbar is partitioned into many small crossbars and overcomes the limitations of conventional switch microarchitecture. Although the organization provides high performance, its scalability is limited due to power and area overheads by the wires and intermediate buffers.*

*We propose alternative scalable router microarchitectures that leverage a network within the switch design of the high-radix routers themselves. These designs lower the wiring complexity and buffer requirements. For example, when a folded-Clos switch is used instead of the hierarchical crossbar switch for a radix-64 router, it provides up to 73%, 58%, and 87% reduction in area, energy-delay product, and energy-delay-area product, respectively. We also explore more efficient switch designs by exploiting the traffic-pattern characteristics of the* global *network and its impact on the* local *network design within the switch. In particular, we propose a* bilateral butterfly *switch organization that has fewer crossbars and half the number of global wires compared to the topology-agnostic folded-Clos switch while achieving better low-load latency and equivalent saturation throughput.*

## 1 Introduction

As the size of large-scale systems increases, the interconnection network that connects all the components together becomes increasingly important and can determine the overall performance and cost of the system. These large-scale networks have been traditionally found in supercomputers, but with the recent emergence of *warehouse*-computing [14] and datacenters that can have up to millions of servers, cost-efficient large-scale networks are also needed in datacenters to provide high performance and minimize overall energy consumption [1]. Previously, large-scale networks were built with *low*-radix routers where the number of ports was relatively small [2, 8]. However, with the exponentially increasing pin bandwidth, recent work [22] has shown how it is more cost-effective to partition the increasing pin bandwidth into a large number of ports and create *high*-radix networks. The Cray BlackWidow [30] is one of the first systems that used high-radix networks as it leveraged radix-64 routers.

One of the challenges with high-radix routers is creating a scalable router microarchitecture. The cost of these large-scale networks is dominated by the cables [21] and the performance is often limited by the off-chip channel bandwidth. Thus, the router microarchitecture needs to be scaled properly to ensure that the off-chip channel bandwidth is properly saturated and the internal router microarchitecture does not become the bottleneck. Unfortunately, the canonical router microarchitecture that consists of a single crossbar switch [10] scales poorly with the router radix ($k$) as the complexity of the allocation is proportional to $k^2$ and the fanout of the crossbar wires increases linearly with $k$. To overcome these limitations, a hierarchical crossbar switch organization [22] was proposed and adapted in the YARC router [30]. It partitions a single large crossbar into many small crossbars or subswitches and places intermediate buffers at the input and the output of the subswitches. This lowers the allocation complexity and also reduces the number of output ports driven by each input port, compared with a conventional crossbar. However, the YARC router design requires an excessive number of wires as described in Section 3.1 and does not scale.

In this paper, we propose alternative scalable router microarchitectures using local network and leverage high-radix topologies, such as folded-Clos [10] and HyperX [3],

978-1-4673-0826-7/12/$26.00 ©2011 IEEE

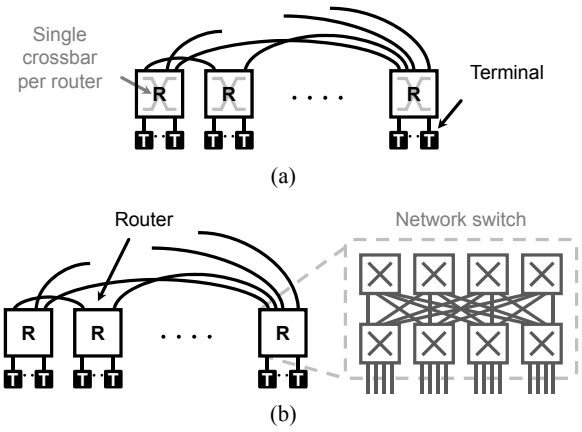



**Figure 1: (a) Conventional crossbar switch based network and (b) network within a network approach.**

in the switch design of the high-radix routers. We create an on-chip network [9] within a router to create a *local* network that connects the switch ports of the router, which is used to construct a *global* network by interconnecting the routers together – creating a *network within a network* approach to building a scalable switch microarchitecture (Figure 1(b)). Compared to the hierarchical crossbar design, these network switch organizations require fewer global wires, crossbars, and intermediate buffers and results in improvement of both area and energy efficiency. In addition, since the local network is used within a global network, we leverage the traffic-pattern characteristics of the global network projected on individual routers to simplify the switch router microarchitecture. For example, load-balancing is often necessary for high performance on any network topology. We show how load-balancing within the local network is no longer needed, and thus, the complexity of the local network within the switch can be further optimized.

Our experimental results show that the switch design adopting the folded-Clos topology is more area and energy efficient than alternatives, including the hierarchical crossbar organization. The design provides up to 73% area reduction compared to the hierarchical crossbar switch while achieving up to 58% and 87% reduction in energy-delay and energy-delay-area products for radix-64 routers. We also propose a new topology-aware switch design called *bilateral butterfly* for the local networks in a folded-Clos global network, which consists of fewer crossbars and half the number of global wires compared to the topology-agnostic folded-Clos switch, while achieving better low-load latency and equivalent saturation throughput.

The contributions of this paper include the following:

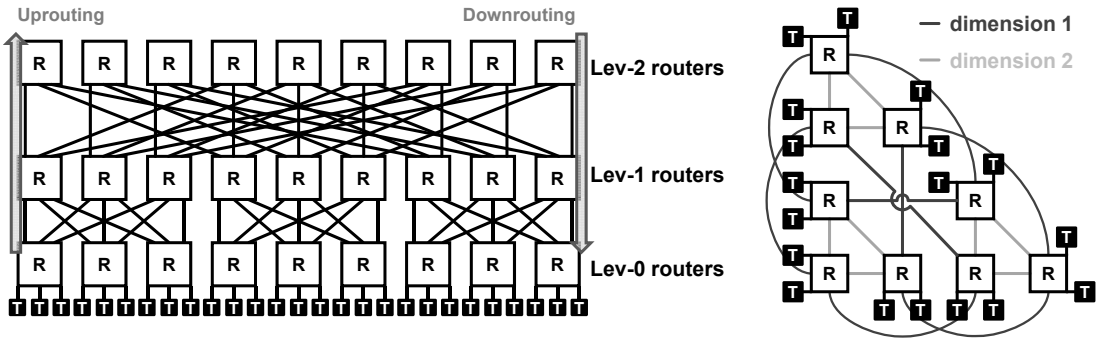
- We show that prior approaches to building a high-radix router using hierarchical organization provide high performance, but suffer from high area and low energy-efficiency with limited scalability.
- We leverage the concept of *network within a network* – using a local network within a router to create the internal switch microarchitecture and using it within a large-scale, global network.
- We show how the local network can exploit the characteristics of the global network to simplify the router design and propose an area- and energy-efficient bilateral butterfly switch design for global networks employing the folded-Clos topology.
- We provide a detailed analysis of alternative approaches and show that the folded-Clos switch designs provide the best area and energy efficiency and are capable of providing fine-grain trade-offs between switch performance and area/energy efficiency.

## 2 Background

A router microarchitecture consists of datapath and control structures, such as I/O ports, buffers, crossbars, route computation units, and allocators. In designing the high-radix routers, reducing the die area and improving the performance and energy efficiency are all important. Performance is the primary design goal of any VLSI circuits, and a highly energy efficient design enables the router to achieve better performance within the power budget of the package. Reduction in the area enables the additional area to be used for more I/Os and other functionalities. For example, in the Cray YARC router [30], the SerDes (serializer/deserializer) I/Os occupy approximately 20% of the total area. In the Cray Gemini Router [29], the network interface (NIC) was incorporated into the same router chip. In this section, we review the technology trends for critical components of high-radix routers and popular high-radix topologies.

### 2.1 Technology Trends

Prior work [22] showed the exponential growth in pin bandwidth over the past 30 years. The Cray YARC router published in 2006 has an aggregate off-chip bandwidth of 2.4Tb/s, Rambus demonstrated a 8Tb/s memory system in 2008 [6], and a hub chip in the PERCS interconnect published in 2010 has a $56 \times 56$ crossbar with 9Tb/s of raw off-chip bandwidth [4]. A high-speed serial link design that achieves sub 1pJ/b of energy efficiency over 10Gbps of transfer rate was reported in 2010 [25]. Area efficiency of high-speed links starts to gain interest as well [28]. Considering the FO4 delay of a future 22nm process projected by a PTM model [38] and a 4 FO4 design rule [36], we expect that the off-chip I/O bandwidth continues to increase over the next 5 to 10 years. In addition, the recent advances in nanophotonic interconnects are expected to not only provide an increasing bandwidth, but also improve the




(a) Folded-Clos (b) HyperX

**Figure 2: (a) An exemplar 3-level folded-Clos network connecting 27 terminal nodes using radix-6 routers. (b) An exemplar HyperX topology with** $(L = 2, S = 3, K = 1, T = 2)$**.**

*efficiency* of signaling so that the impact of I/O signaling on the overall power consumption will be reduced [6].

The performance of on-chip global wires rather than transistor speed has become a major limiting factor on IC design [13] in deep submicron technology. There is a huge design space on semi-global and global wires for different design goals, such as delay, throughput density, and energy efficiency [37]. We assume that the wires used for datapath within a crossbar and between crossbars are RC repeated and have the same pitch. Since throughput density and energy efficiency are important for these wires in high-radix routers, we assume that these are minimum pitch (in the range of dozens of the minimal wire pitch of metal layer 1), similar to the assumptions made by Balfour and Dally [5] and by Joshi et al. [16] for their on-chip network designs.

### 2.2 Topology

In order to fully exploit the benefits of high-radix routers, conventional low-radix topologies, such as 2D/3D mesh or torus topologies, are not appropriate, and topologies tailored to high-radix routers, such as folded-Clos [10] or flattened butterfly [20], are necessary. A folded-Clos network is a multi-level network made by folding a Clos network that consists of an odd number of stages and then by fusing input and output switch pairs that collocate. A symmetric 3-stage Clos network can be represented by three tuples ($m$, $n$, $r$), where $m$, $n$, and $r$ are the number of middle-stage router, input ports per input router, and the input routers, respectively. The Clos network has a path diversity $m$ and is non-interfering [10] if $m \geq n$. By folding, the 3-stage Clos network becomes a 2-level folded-Clos network where each port becomes bidirectional by having an input and an output channel, a middle-stage router becomes a top-level router with radix $r$, and a pair of an input and an output router becomes a bottom-level router with radix $(n + m)$. Throughout the paper, we assume that each router or switch has the same number of input and output channels and we call it a radix of the router or switch. A folded-Clos network with more than 2 levels can be constructed by composing each top-level router using a folded-Clos network recursively. Figure 2(a) shows an exemplar 3-level folded-Clos network connecting 27 terminal nodes using radix-6 routers. Routing in a folded-Clos consists of two phases – uprouting, and then, downrouting. In uprouting, the packets are routed to the nearest common ancestor between the source and destination. Once the packet reaches this intermediate router, the packet is downrouted to its destination.

The flattened butterfly [20] is another topology that can exploit high-radix routers. An $m$-ary $n$-flat network is derived from an $m$-ary $n$-fly butterfly network where all the routers in each row of the butterfly network are *flattened* into a single router and the same connections are maintained. HyperX [3] extends the flattened butterfly topology and we use the notation provided by the HyperX framework in this paper. A regular HyperX is an $L$-dimensional direct network where the size of each dimension is $S$ and the routers in each dimension are fully connected. Each router in the HyperX is connected to $T$ terminals and the bandwidth of links between the routers is $K$ times the bandwidth of links to terminals. Using the $(L, S, K, T)$ notation, an $m$-ary $n$-flat network can be represented as a $(L = n - 1, S = m, K = 1, T = m)$ HyperX network. It connects $TS^L$ terminals with $\frac{KS^{L+1}}{2}$ channels crossing its bisection. Therefore, the ratio of bisection bandwidth to aggregate terminal bandwidth $\beta = \frac{KS}{2T}$ is 0.5 for the $m$-ary $n$-flat network. Figure 2(b) shows an exemplar 2-dimensional HyperX network composed of radix-6 routers with $(L = 2, S = 3, K = 1, T = 2)$. In this paper, HyperX and flattened butterfly terms are used interchangeably.

## 3 High-Radix Router Microarchitecture

In this section, we first describe the previously proposed hierarchical switch organization for high-radix routers [22, 30] and its limitations. Then, we describe an alternative switch design that uses a network, and thus, creates a net-

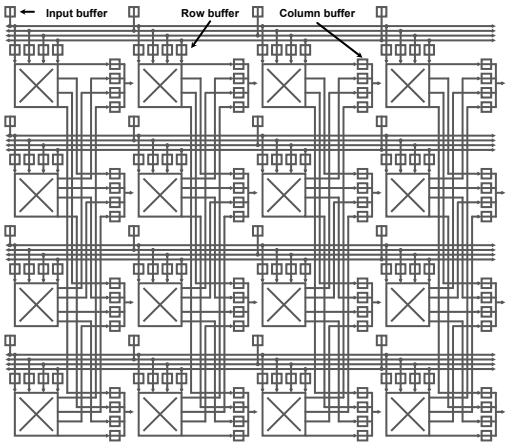



**Figure 3: A layout of a radix-16 hierarchical crossbar switch organization. Subswitch size $p$ is 4.**

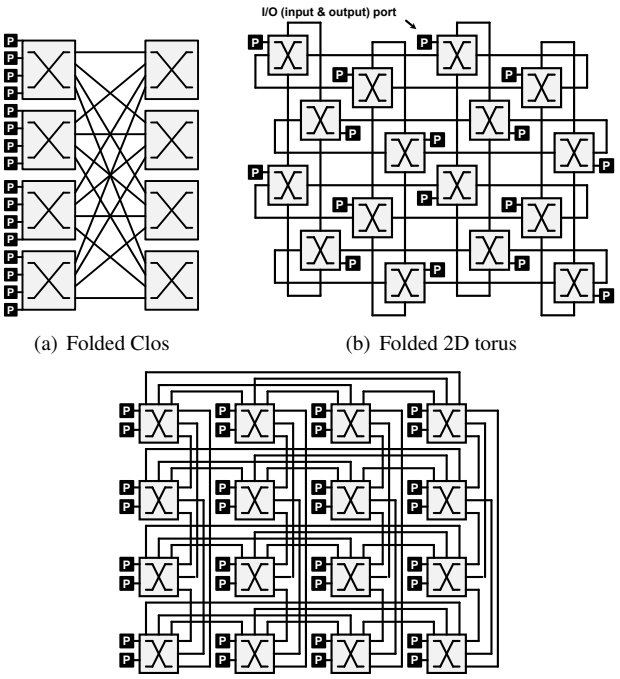
work within a network organization. Throughout this work, we define a *local* network as the network used for a switch within a single router and a *global* network as the system-level network that interconnects the routers together.

### 3.1 Hierarchical Switch Organization

The hierarchical router organization [22] partitions a single $k \times k$ switch into $(k/p)^2$ $p \times p$ subswitches where $k$ is the router radix and $p$ is the subswitch crossbar radix. The hierarchical router organization introduces $2\frac{k^2}{p}$ intermediate buffers (or subswitch buffers) at both the input and the output of the subswitches. The intermediate buffers decouple the input and the output arbitration and avoid global arbitration by localizing the arbitration. Figure 3 shows a layout of a hierarchical crossbar switch organization with $k = 16$ and $p = 4$.

The hierarchical crossbar switch consists of $k$ row buses or horizontal broadcast channels and $\frac{k}{2} \times \frac{k}{p}$ vertical, column channels since all output ports in a column are fully connected to all column buffers in the same column. These horizontal and vertical channels require global wires, whose area scales at the rate of $k \times \frac{k}{2} \times \frac{k}{p} = \frac{k^3}{2p}$. Note that the total area of all the subswitch crossbars scales at the rate of $k^2$ as there are $(k/p)^2$ subswitches and each $p \times p$ subswitch area is proportional to $p^2$. We set $p = \sqrt{k}$ to minimize the aggregate fanout [1] and minimize the dynamic energy of a packet to traverse the switch [33]. As a result, the area of horizontal and vertical wires dominates the switch area with a large $k$, which scales at the rate of $k^{2.5}$ The number of subswitch buffers also scales super-linearly at the rate of $\frac{k^2}{p} = k^{1.5}$. These all lead to an inefficient area and power scaling of the hierarchical switches. Thus, we explore an alternative switch organization using different multi-stage networks as the switch organization to provide a scalable switch design.

[1] In this paper, we define fanout as the number of output ports that an input port has to drive in a component, such as a crossbar, a mux, or a bus. When a switch consists of multiple components, we add the fanout values of all the components that a packet passes in the switch and call it an aggregate fanout. In the hierarchical switch, the broadcast bus delivers the packet to $\frac{k}{p}$ row buffers, a $p \times p$ subswitch, and a column multiplexer so that the aggregate fanout is $\frac{k}{p} + p + 1$.




(a) Folded Clos (b) Folded 2D torus

(c) 2D HyperX

**Figure 4: (a) A folded 2D torus, (b) a 2D HyperX, and (c) a folded-Clos switch organization.**

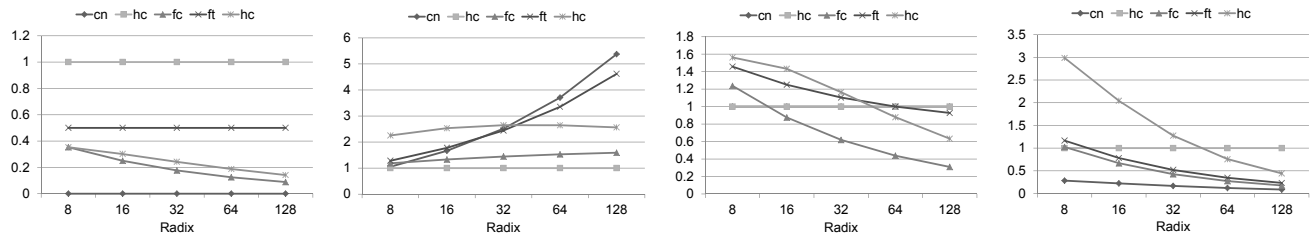
### 3.2 Network within a Network Switch Architecture

We explore the switch microarchitecture designs based on network topologies and create a network within a network switch architecture, or a network switch architecture, where a switch is composed of multiple subswitches connected through internal point-to-point channels. Each subswitch is input buffered so that arbitration can be localized to a subswitch, similar to the hierarchical switch organization. For the local network, we focus on three different topologies: folded-Clos [10], 2D torus [10], and 2D HyperX [18, 3]. Other topologies can be used such as a 2D mesh topology or a tree-based topology for the local network. However, these either introduce non-uniformity (i.e., 2D mesh) or a single-point bandwidth bottleneck (i.e., tree) and are not suitable as the local network. The off-chip bandwidth is an expensive resource in large-scale systems and they need to be fully utilized. Thus, the local network cannot become the bottleneck.

We set all switch organizations to support the same aggregate off-chip bandwidth at a given router radix $k$ and use the following metrics: the number of buffers and cross-

| | Canonical Crossbar | Hierarchical Crossbar | Folded-Clos | Folded 2D Torus | 2D HyperX |
|---|---|---|---|---|---|
| Subswitch buffers | N/A | $2\frac{k^2}{p}$ | $2k$ | $k^{\frac{3}{2}}$ | $4k^{\frac{2}{3}}(k^{\frac{1}{3}}-1)$ |
| Aggregate fanout | $k-1$ | $\frac{k}{p}+p+1$ | $\frac{3k}{r}+r-2$ | $\frac{3}{4}k+\frac{5}{4}k^{\frac{1}{2}}-1$ | $25(k^{\frac{1}{3}}-1)$ |
| Total crosspoints | $k^2$ | $k^2$ | $(rk+\frac{3k^2}{r})$ | $k(2k^{\frac{1}{2}}+\frac{3}{4}k)$ | $(5(k^{\frac{1}{3}}-1)k^{\frac{1}{3}})^2$ |
| Switch area | $k^2$ | $\frac{k}{p}(k^2+2p^2)$ | $\frac{3}{2}k(\frac{4k}{r}+k+r)$ | $\frac{9}{4}k(k^{\frac{1}{2}}+1)^2$ | $(\frac{3}{2}(5k^{\frac{1}{3}}-4)k^{\frac{1}{3}}+k)^2$ |

**Table 1: Router microarchitecture complexity analysis. $k$ is the radix of the switches, $p$ is the radix of the subswitch crossbars in a hierarchical crossbar switch, and $r$ is the radix of the top-level subswitches in a folded-Clos switch.**




(a) Subswitch buffers (b) Aggregate fanout (c) Total crosspoints (d) Switch area

**Figure 5: The relative (a) switch buffers, (b) aggregate fanout, (c) total crosspoints, and (d) switch area of the five switch organizations in Table 1. cn, hc, fc, ft, and hx stand for canonical crossbar, hierarchical crossbar, folded-Clos, folded 2D torus, and 2D HyperX switches. The hierarchical crossbar switch design is used as a baseline.**

points in a switch, the aggregate fanout, and the total switch area. Each crosspoint in a crossbar and each buffer are implemented by transistors that consume leakage power regardless of switch activity, so that more crosspoints and buffers result in higher static power. The aggregate fanout impacts dynamic energy as larger transistors are required to drive a higher fanout, consuming more dynamic energy.

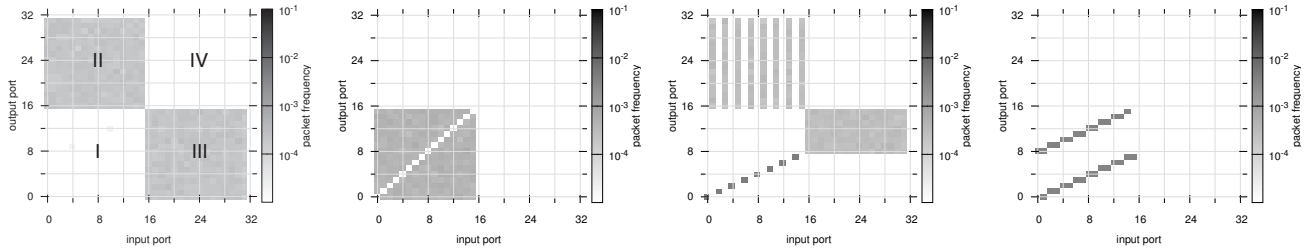
The block diagrams of the alternative local networks are shown in Figure 4. For the folded 2D torus organization, we assume a $\sqrt{k}$-ary 2-cube network with $\sqrt{k} \times \sqrt{k} = k$ subswitches. The network has $4\sqrt{k}$ channels crossing its bisection, so the bandwidth of each internal channel must be $\frac{k}{4\sqrt{k}} = \frac{\sqrt{k}}{4}$ times higher than the bandwidth of each external channel connected to an I/O port. We assume that each subswitch dedicates a port to an I/O port and a single logical internal channel consists of $\frac{\sqrt{k}}{4}$ physical channels, so the radix of a subswitch becomes $\frac{\sqrt{k}}{4} \times 4 + 1$. We choose Valiant's routing algorithm [34], which gives $\sqrt{k}$ as the average hop count. As a result, the folded 2D torus switch has $k\sqrt{k}$ subswitches, the aggregate fanout is proportional to $k$, and both switch area and the number of crosspoints are proportional to $k^2$.

For the folded-Clos switch, we assume that it is a ($m$, $\frac{k}{r}$, $r$) Clos network and $m = \frac{k}{r}$ (Section 5.3 discusses the folded-Clos switches with a higher $m$). This 2-level local network consists of $r$ radix-$\frac{2k}{r}$ bottom-level subswitches (which are connected to the I/O ports) and $\frac{k}{r}$ radix-$r$ top-level subswitches, so there are $rk + \frac{3k^2}{r}$ crosspoints and the aggregate fanout is up to $\frac{3k}{r} + r - 2$. We place the top-level subswitches in the middle and half of the bottom-level subswitches on each side to provide a better aspect ratio, as shown in Figure 7(b). The switch area is dominated by global wires that are used for channels connecting subswitches, which scales at the rate of $k^2$.

As for the 2D HyperX, we assume $K = 2$ and $S = T$, so that a channel between subswitches has twice the bandwidth of a channel to an I/O port. This switch is a ($L = 2$, $S = k^{\frac{1}{3}}$, $K = 2$, $T = k^{\frac{1}{3}}$) HyperX network. The subswitch radix is $5k^{\frac{1}{3}} - 4$ and there are $k^{\frac{2}{3}}$ subswitches, so the local network has about $4k$ subswitch buffers and $25k^{\frac{4}{3}}$ crosspoints. We assume a minimal or Valiant's routing algorithm for the 2D HyperX switch so that the worst case aggregate fanout becomes $25(k^{\frac{1}{3}} - 1)$. The switch area is dominated by global wires as well so that it scales at the rate of $k^2$.

We summarizes the complexity of the canonical crossbar, hierarchical crossbar, folded-Clos, folded 2D torus, and 2D HyperX switch organizations in Table 1. The switch area is normalized to the square of a channel pitch, and includes the area of buffers, crossbars, and the global wires used for the channels between subswitches. We ignore the area from the control logic as they represent a very small fraction of the total area [35]. Figure 5 shows the relative subswitch buffers, aggregate fanout, total crosspoints, and switch area of the switch organizations as the switch radix is increased from 8 to 128. The results are normalized to the values of the hierarchical crossbar. We set $p = \sqrt{k}$ and $r = 2\sqrt{k}$ to minimize aggregate fanout and switch area.

The network switch architectures, especially the folded-Clos switch design, scales better than the canonical crossbar




(a) Router 0 – bottom-level - UR (b) Router 512 – top-level - UR (c) Router 0 – bottom-level - BR (d) Router 512 – top-level - BR

**Figure 6: 2D Heatmaps showing the frequencies of packets moving from the input ports (in the $x$ axis) to the output ports (in the $y$ axis) at (a,c) a bottom-level and (b,d) a top-level radix-32 router of a 4096 node folded-Clos network. (a,b) and (c,d) are on the UR and BR traffic, respectively.**

and the hierarchical crossbar switches in terms of power and energy efficiency as the switch radix increases. All the network switch organizations require fewer subswitch buffers than the hierarchical crossbar switch, and also have fewer crosspoints as the switch radix is 64 or higher. The results show that the switches adopting torus, folded-Clos, or HyperX topologies consume less static power than the hierarchical crossbar organization on high-radix designs. These network switches require no broadcast channels through global wires and results in a decrease of relative switch area as the switch radix is increased. These switches have higher aggregate fanout values than the hierarchical crossbar switch, consuming more dynamic power. However, static power consumption is more important for high-radix routers since most of the crosspoint transistors do not drive wires. For example, in a canonical crossbar, only $k$ out of $k^2$ crosspoints are active and the remaining $k^2 - k$ ones consume static power only, evidenced by the YARC design where static power dominates dynamic power [30]. The torus switch scales worse than the folded-Clos switch in all aspects because it has higher hop count and requires wider channels between subswitches to provide the same bisection bandwidth. Table 1 shows that the HyperX switch has a lower exponent value than the folded-Clos switch on the aggregate fanout and the number of crosspoints. Therefore, the 2D HyperX switch has better scalability, but its large coefficients make the folded-Clos switch become more area and power efficient on the radices we consider.

# 4 Exploiting Global Network Traffic Characteristics

In the previous section, we explored alternative router switch microarchitectures built using a network, instead of a single-stage crossbar. Our analysis showed that a switch design based on the folded-Clos network scales better in terms of energy efficiency and die area, compared to other alternatives, as the router radix increases. In this section, we further reduce the cost of a high-radix switch microarchitecture by exploiting the traffic characteristics of the *global* network and its impact on the *local* network. We make the observation that *the traffic of the global network, based on its topology and routing, can be exploited to further reduce the cost of the switch microarchitecture*. We first describe the impact of the global network traffic on the local network and then describe *bilateral butterfly* – a novel switch organization that exploits the characteristics of the global network traffic for high-radix routers.

## 4.1 Global Network's Impact on Local Traffic

In Figure 6, we show the heatmap of traffic on a non-interfering folded-Clos network. The results are shown for a 3-level 4096-node network with radix-32 routers ($k = 32$), but the observations made in this section can be generalized to other folded-Clos configurations. The heatmap shows the frequencies of packets moving from input ports (x-axis) to output ports (y-axis) at routers in different levels of a folded-Clos network on the UR (uniform-random) and BR (bit-rotation) traffic patterns. On each router, the ports 0 to $\frac{k}{2} - 1$ are connected to lower level routers (or terminal nodes from level 0 routers) while ports $\frac{k}{2}$ to $k - 1$ are connected to higher level routers. The communication characteristics can be partitioned into four quadrants, as shown in Figure 6(a). Each quadrant represents different types of traffic. Quadrant II represents uprouting traffic while quadrant III is downrouting traffic in the folded-Clos network. Quadrant I represents traffic 'turning' in the network – i.e., a packet arrives at the nearest common ancestor between the source and the destination and switches from uprouting to downrouting. However, quadrant IV corresponds to the packets that would switch from downrouting to uprouting. This is not possible in the folded-Clos network because a packet that starts downrouting will continue to be downrouted until it reaches its destination. The heatmaps reflect this by showing that quadrant IV is not used in any routers, regardless of the traffic pattern.

Folded-Clos networks use random routing or its variants [19] as a packet chooses one of the nearest common ancestors between its source and destination with equal proba-

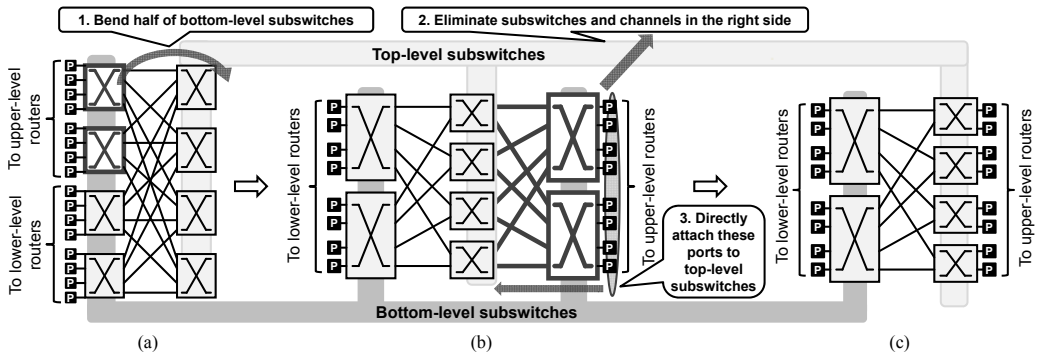



**Figure 7: Bottom-level subswitches connected to upper-level routers and global wires connected to these subswitches (highlighted with thick lines) in a folded-Clos switch are eliminated forming a bilateral butterfly network. All channels are bidirectional.**

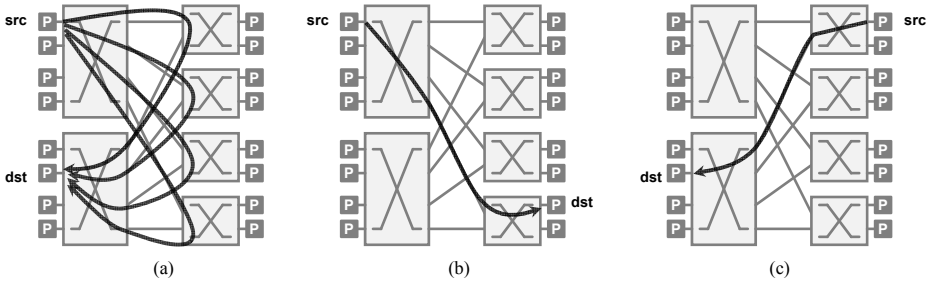



**Figure 8: Routing example of the bilateral butterfly for (a) packets "turning", (b) uprouting, and (c) downrouting.**

bility. Because of this load-balancing in the global network, the uprouting within a switch is load-balanced and results in uniformly utilizing the output ports shown in quadrant II. Similarly, regardless of the traffic pattern, the randomization in the uprouting leads to the input ports used in downrouting to be used with equal probability (quadrant III). However, the characteristics of the quadrant I traffic depend on the traffic patterns of the network. On UR, most packets reach top-level routers and fewer packets 'turn' in middle- and bottom-level routers. Thus, quadrant I is more heavily utilized in upper-level routers. On BR, the bit-rotating traffic pattern of the network is reflected only in quadrant I, as shown in Figure 6(c,d).

The router switch needs to provide high performance for traffic 'turning' (quadrant I traffic) while others (quadrant II and quadrant III traffic) are load-balanced by the global network. The *local* network also does not need to load-balance the traffic. In comparison, based on the global network, the quadrant IV traffic is non-existent. We take advantage of these observations to further optimize the switch microarchitecture and propose the bilateral butterfly switch microarchitecture in Section 4.2.

### 4.2 Bilateral Butterfly Switch Architecture

The observations described in Section 4.1 enables further optimization for the folded-Clos switch design used in the folded-Clos global network. Figure 7 illustrates how some of the subswitches in the folded-Clos switch architecture can be removed. By performing a simple transformation from Figure 7(a) to Figure 7(b) by moving the port location, the subswitches in the rightmost column can be eliminated and the ports on the right side are connected directly to the middle-stage subswitches. This forms a *bilateral butterfly* switch (Figure 7(c)) where the switch resembles a butterfly switch but all channels are bidirectional. The middle stages of a folded-Clos are needed for load-balancing, but with *global* load-balancing achieved from the global network, the load-balancing within the *local* network is no longer needed. A routing example in the bilateral bufferly is shown in Figure 8. For both uprouting or downrouting (Figure 8(b,c)), no path diversity is provided within the switch as it becomes deterministic routing. However, when the packet is turning (Figure 8(a)), the same path diversity as the folded-Clos is still available.

Depending on the router radix, one fourth or one third of crossbar crosspoints are saved on each router. But more importantly, half of global wires are eliminated, providing significant energy and static power savings compared to the already efficient folded-Clos switch. The bilateral butterfly switch provides sufficient path diversity for packets that 'turn' in the router, (Figure 8(a)), but provides no path diversity to packets passing through the router. However, based on the traffic characteristics described in the previous sec-

tion, no further load-balancing is needed for packets continuing to move upwards or downwards. Thus, deterministic routing is sufficient (Figure 8(b,c)). Packets entering the bilateral butterfly switch experiences fewer aggregate fanout on average and results in a lower dynamic power. The bilateral butterfly switches can also be used as top-level routers by not utilizing ports dedicated to upper-level routers or by pairing an input port at one side with an output port on the other side.

A similar approach described in this section can also be applied to further optimize the HyperX switches within a HyperX network. When the Valiant routing or other adaptive routing algorithms are used for adversarial traffic, the output ports of the packets traversing the global HyperX network are uniformly distributed unless they turn. As a result, a HyperX switch with half the bisection bandwidth of the HyperX switch design in Section 3.2 ($\beta = 0.5$) is enough to be a local network within the global HyperX switch with $\beta = 0.5$.

# 5 Evaluation

We evaluate the performance and efficiency of the proposed switch organizations using various traffic patterns on single router and global network configurations. Single router results show that the folded-Clos switch design performs better than other network switch designs, but still performs worse than the hierarchical crossbar design. The performance of the folded-Clos switch is further improved by adding more top-level subswitches, providing input speedup on bottom-level subswitches, or both. This fine-tuning makes the folded-Clos switch perform comparable to the hierarchical crossbar design and more energy efficient. The bilateral butterfly switch organization further improves the performance and energy-efficiency of the global networks.

## 5.1 Experimental Setup

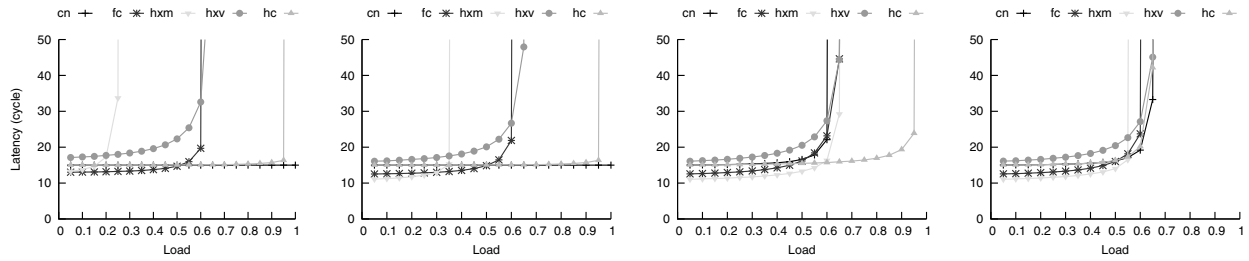
We developed an event-driven cycle accurate simulator which supports single-stage crossbars and hierarchical crossbars for local networks and folded-Clos and HyperX for both local and global networks. The folded-Clos networks use an adaptive routing algorithm that randomly picks the two nearest common ancestors between the source and destination of a packet and chooses one with less congestion as an intermediate node, which is the same as the *greedy_r*(2) algorithm in [19]. The HyperX networks use Valiant's routing [34] or minimal adaptive routing algorithm [3]. A packet that arrives at a router proceeds through the steps of route computation, virtual-channel allocation, switch allocation, and switch traversal. If the router has a network switch, it performs both the routing of the global and the local network during the route computation step so that the subswitches within the router only have subswitch allocation and traversal steps. Routers and subswitches are input buffered and each buffer has multiple virtual channels (VCs). iSLIP [24] separable allocation method is used for VC and switch allocation. We assume virtual cut-through flow control, which was also used in the YARC router [30]. Unless mentioned otherwise, switch speedup is not used. We warm up the network before measuring the performance of the networks. Experiments with single-flit packets are presented. Simulations with multiple-flit packets show similar trends as single-flit experiments and are omitted due to limited space.

We apply the 22 nm predictive technology models [38] and interconnect projections from ITRS [31] to a modified McPAT [23] for SRAM and wire modeling. We target 5 GHz operating frequency for buffers, crossbars, and global wires. Propagation delay is assumed to be 2 cycles for internal channels between subswitches, except for hc, where it becomes 4 cycles due to its relatively large area, as shown in Table 2. Even though it is infeasible to operate the crossbar in cn at 5 GHz, we assume that it can be pipelined and has the same zero-load latency as hc. We estimate 40fJ for sending a bit over a 1-cycle distance, 40fJ and 10fJ for reading and writing a bit on a 128kb input port buffer and a 8kb subswitch buffer, and 40fJ per radix to send a bit through a crossbar. As for static power, we assume 20fJ for a 1-cycle distance global wire per cycle and 8fJ per radix for each crosspoint per cycle, 10mW and 2mW on an input port buffer and a subswitch buffer, and 1pJ for transferring a bit off-chip. These values are similar to the ones Joshi et al. [16] estimated on a 22nm process.

## 5.2 Comparing the Performance of High-Radix Router Microarchitectures

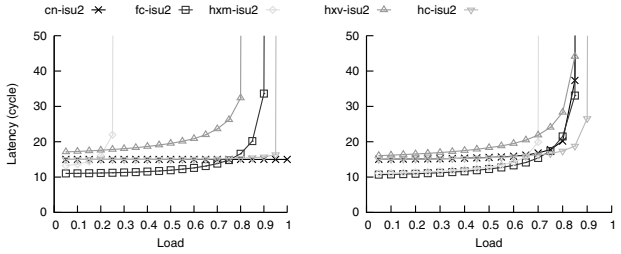
We evaluate the following switch microarchitecture organizations: canonical crossbar (cn), folded-Clos (fc), HyperX with Valiant's routing (hxv), HyperX with minimal-adaptive routing (hxm), and hierarchical crossbar (YARC) switches (hc). We use different traffic patterns, including two bit permutation patterns (BC and BR) and two random traffic patterns (UR and TR). TR is a transpose-random traffic, where a node in a row $A$ of a square matrix is equally likely to send packets to nodes in a column $A$ of the square matrix. Note that TR is one of the worst-case traffic patterns for the hierarchical crossbar [22] [2]. Figure 9 shows the average latency of injected packets over offered loads of radix-64 switches on BC, BR, UR, and TR patterns. The subswitch radix in hc ($p$) is 8 and the top-level subswitch radix in fc ($r$) is 16. Crossbars have no input speedup. All switch designs use 4 VCs, where hxv uses half of them to traffic

[2] We also tested other traffics, but they showed similar performance trends to BC, BR, UR, and TR patterns. Their results were not included due to the space limitation, except in Section 5.3.



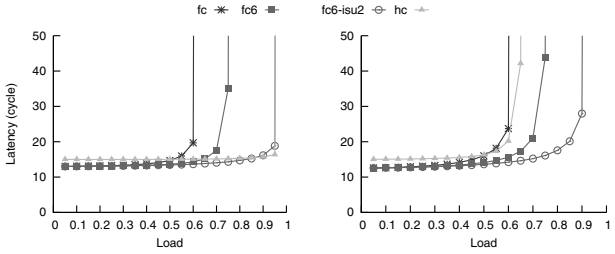

(a) BC (bit complement) (b) BR (bit rotation) (c) UR (uniform random) (d) TR (transpose random)

**Figure 9: Load-latency graphs of radix-64 switches on (a) BC, (b) BR, (c) UR, and (d) TR traffic patterns. cn, fc, hxv, hxm, and hc stand for canonical crossbar, folded-Clos, HyperX with Valiant's routing, HyperX with minimal-adaptive routing, and hierarchical crossbar switches. No input speedup on crossbars.**




(a) BC (bit complement) (b) TR (transpose random)

**Figure 10: Load-latency graphs of radix-64 switches on (a) BC and (b) TR traffic patterns. -isu2 stands for the input speedup of 2 for each crossbar.**




(a) BC (bit complement) (b) TR (transpose random)

**Figure 11: Load-latency graphs of radix-64 switches on (a) BC and (b) TR traffic patterns.**

random intermediate subswitches and the others to traffic to destinations ports. hxm chooses the VC channel based on the number of hops until the destination.

The infeasible canonical crossbar design performs ideally on permutation patterns such as BC and BR, but it experiences the head-of-line (HoL) blocking [17] on random traffics similar to other designs and its achieved throughput saturates around $66\%$. fc, hxv, and hxm all suffer from the HoL blocking problem. Under low loads, fc has a lower average latency than hc because fc is smaller so it has lower global wire propagation delays. The average latency of hxm is the lowest, except for the BC traffic pattern, because there are packets with minimum paths whose hop counts are lower than 3. However, it saturates quickly in adversarial traffic compared to hxv, so adaptive routing algorithms combining the benefits of both can be beneficial, such as UGAL [32], Adaptive Clos [20], and DAL [3]. hc performs almost ideally on BC, BR, and UR, even without the crossbar input speedup owing to its property of distributing loads from a port to $8$ subswitches on random traffic [22]. As a result, the average load in a subswitch is $0.125$, even with full loads from the ports due to this effective output speedup through broadcast. This load distribution does not happen at the TR traffic pattern and hc suffers from the same HoL blocking problem (Figure 9(d)).

Through speeding up the input ports of the crossbars, we can improve the saturation throughput of the switch designs. Load-latency graphs in Figure 10 show that the performance of folded-Clos and 2D HyperX switches improves noticeably but is still outperformed by the hierarchical crossbar switches. Considering the high area and energy-efficiency overhead imposed by speeding up the crossbars, crossbar speedup should be applied carefully, which is further explored in the following subsection.

### 5.3 Exploring the Design Space of the Folded-Clos Switch Architecture

Section 5.2 shows that the folded-Clos switch design performs better than the 2D HyperX design, but the performance does not match that of the hierarchical crossbar design. Since fc results in lower area and dissipates lower static power than hc with comparable dynamic energy consumption, additional performance-cost trade-off can be made with fc to increase its performance.

These tradeoffs are possible with the folded-Clos topology by putting more top-level switches (increasing $m$) and speeding-up bottom-level switches. We have assumed $n = m$ but with $m > n$, an output speedup of $\frac{m}{n}$ in the input-stage subswitches is provided such that $\frac{n}{m}$ becomes the maximum load of a top-level subswitch. Since top-level subswitches are chosen randomly, fc with a higher $m$ experiences adversarial traffic less frequently than hc. Fur-

| | Canonical Crossbar | Hierarchical Crossbar | Folded-Clos | | | Folded 2D Torus | 2D HyperX |
|---|---|---|---|---|---|---|---|
| | | | fc | fc6 | fc6-isu2 | | |
| Subswitch buffers | N/A | 1,024 | 128 | 192 | 192 | 512 | 192 |
| Aggregate fanout | 63 | 17 | 26 | 28 | 57 | $45 \sim 75$ | |
| Total crosspoints | 4,096 | 4,096 | 1,664 | 2,400 | 3,872 | 4,096 | 3,600 |
| Switch area | 4,096 | 33,792 | 9,216 | 16,896 | 24,640 | 11,664 | 25,600 |

**Table 2: Router design complexity analysis on a radix-64 router. $p = 8$, $r = 16$, while $m$ is varied.**

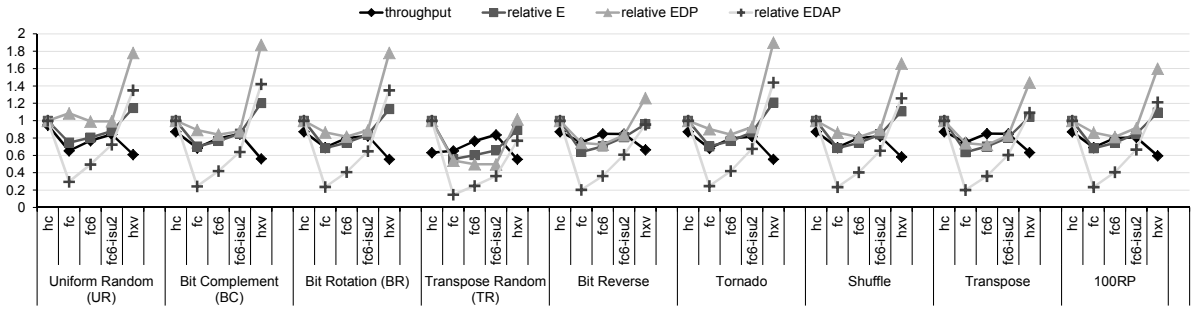



**Figure 12: Achieved throughput, relative energy, relative energy-delay product (EDP), and relative energy-delay-area product (EDAP) of 5 switch designs on UR, BC, BR, TR, bit-reverse, tornado, shuffle, transpose, and 100RP patterns. All relative values are normalized to those of hc.**

ther performance improvement is available by speeding-up bottom-level subswitches, whose speedup overhead is lower than top-level ones since bottom-level subswitches typically have a lower radix. Figure 11 shows the performance improvement on both BC and TR patterns (other traffic patterns show similar trends but are omitted due to space limitation), where fc6-isu2 performs comparable to hc on BC and even fc6 outperforms hc on TR. fc6 stands for a folded-Clos switch with $m = 6$ and fc6-isu2 is fc6 with an input speedup of 2. Table 2 compares their design complexity. The complexity of the folded-Clos switch increases as more speedup techniques are applied, but even with fc6-isu2, the complexity is still lower than hc.

In Figure 12, we compare the energy consumption, energy-delay product (EDP), and energy-delay-area product (EDAP) of hc, fc, fc6, fc6-isu2, and hxv. The energy consumption, EDP, and EDAP values are normalized to those of hc. Results show that fc consumes the lowest energy, but also performs the worst. It has the best EDAP as well (lower is better for the energy consumption, EDP, and EDAP metrics). For example, on TR, fc consumes 2.1× lower energy and shows 7.8× better EDAP over hc. fc6 has the best energy-delay product and it has 2.4× lower EDP than hc on TR and has 1.2× lower EDP than fc on BR. Folded-Clos switches are better than the hierarchical crossbar switch, while the merit of 2D HyperX switch is limited to the TR traffic. We also evaluated other traffic patterns, such as bit reverse, tornado, shuffle, transpose, and 100RP [10]. 100RP is the average of 100 random permutation traffic patterns. These traffic patterns show similar trends to UR, BC, BR, and TR.

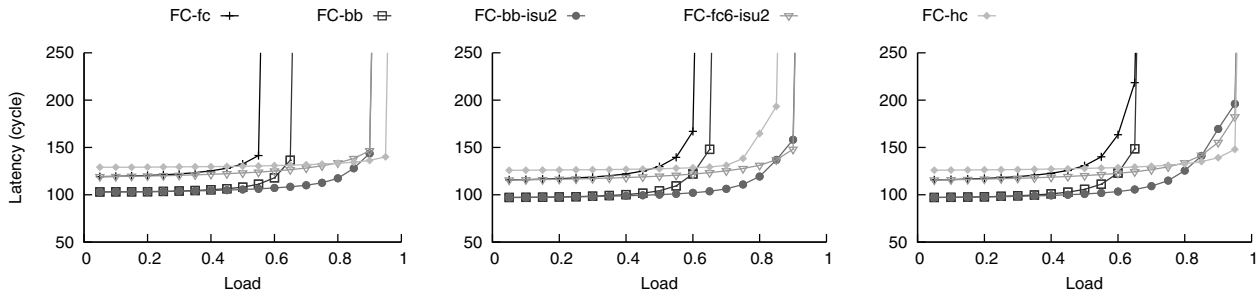
### 5.4 Comparing the Performance of Network Switch Designs in Global Networks

In this section, we evaluate the performance of the network-within-network switch designs within a global network. The folded-Clos topology is used as the global network and is simulated with the following switch designs – fc, fc6-isu2, hc and the bilateral butterfly (bb). 4 VCs are used for all the configurations and the port to port latency between two routers is assumed to be 10 cycles.

Figure 13 shows that hc performs well, except on BR, where the network traffic pattern projected on each router causes some subswitches in the hierarchical crossbar switches to be utilized more heavily. [3] fc6-isu2 performs comparable to hc on BC and UR and it outperforms hc on BR. These results show that the cost- and energy-efficient folded-Clos switch designs perform well on the global network traffic patterns as well as the single router traffic patterns. Considering that they scale better than the hierarchical crossbar switches and perform comparable, or even better, by adding more top-level subswitches and speeding-up bottom-level subswitches, the folded-Clos switch deigns are competitive router microarchitecture options.

Without speedup, bb performs better than fc. Even with an input speedup of 2 for crossbars, bb performs better on most cases, especially when loads are low, since a packet traverses one fewer internal channel and subswitch per router in a bilateral butterfly router. Compared to the

[3] Note that the bit-rotation (BR) traffic pattern on a single hierarchical router is not adversarial, as shown earlier in Figure 9(b). This is an example showing the impact of high-radix network traffic on high-radix routers.




(a) BC (bit complement) (b) BR (bit rotation) (c) UR (uniform random)

**Figure 13: Load-latency graph of folded-Clos networks with 4096 nodes connected by radix-32 switches on (a) BC, (b) BR, and (c) UR traffic patterns. bb stands for a bilateral butterfly switch.**

folded-Clos switch design, bb reduces the switch area by 25% and the switch power by 10% on UR when neither switch has input speedup. These results support that further cost- and energy-efficiency optimizations are possible for the routers composing high-radix folded-Clos networks.

## 6 Related Work

Kim et al. [22] showed that increasing the router radix and creating a high-radix router is more cost-effective than increasing the bandwidth per port. They proposed a hierarchical router organization for high-radix routers and it was implemented in the Cray YARC router [30]. However, the hierarchical organization did not consider how the microarchitecture can exploit the global network and it has scalability limitations. In addition to electrical signaling, different technologies have been recently proposed for high-radix switches, including ones using proximity communication [12] and optics [7]. Mora et al. [26] also proposed a high-radix switch organization to reduce the impact of head-of-line blocking. These studies are orthogonal to this work as we explore alternative switch microarchitectures. To properly exploit high-radix routers, an appropriate topology is needed. The folded-Clos [10] is one such topology and the Cray BlackWidow system employed a variant of the folded-Clos topology [30]. Cost-efficient high-radix topologies [3, 20, 21] have been recently proposed to further reduce network cost. These topologies require a load-balancing routing algorithm [32, 3, 15] to properly exploit the path diversity. In this work, we exploited the load-balancing of these high-radix topologies to optimize the switch design of a high-radix router.

Many different hierarchical networks [11, 10] have been proposed in the literature and our work can be viewed as a hierarchical approach as well. For example, the recently proposed Dragonfly topology [21] also provides a hierarchical network. However, our work differs from prior approaches as we used a local network to create a scalable high-radix switch and the local network is created within the constraint of being placed inside a single chip. The techniques presented in Section 4.2 are not feasible for conventional, hierarchical networks. The Cray T3D router [10] used a similar approach of creating a network-of-switches in the router design. Because of the technological constraints, a single switch was partitioned into three subswitches – one for each $x$, $y$, and $z$ dimension. However, this organization can severely limit the network throughput because of the limited connection between subswitches. A significant amount of research has been done recently in on-chip networks [9, 27]. We also leverage on-chip networks to build a switch. However, the architecture for a scalable high-radix switch differs from on-chip networks for multi/many-core processors because of the different constraints on the type of terminal nodes, bandwidth requirement, traffic patterns, and switch area.

## 7 Conclusion

This paper approached the problem of building a high-radix router by treating it as a network design. We explored alternative topologies, such as folded-Clos, 2D torus, and HyperX, for router switch microarchitecture and compared them with the state-of-the-art hierarchical crossbar architecture. We showed that our network within a network design approach provides better scalability than the hierarchical crossbar architecture on switch area and power consumption. We also showed that the folded-Clos switch design has the smallest area and dissipates the lowest power. The folded-Clos switch also enables fine-grain trade-offs between switch performance and cost/energy efficiency by adjusting the number of top-level subswitches and speeding up bottom-level subswitches. Compared to the hierarchical crossbar design, a radix-64 folded-Clos switch without speedup consumes $35\%$ and $53\%$ less energy on the uniform random and transpose random traffic. A folded-Clos switch with two more top-level subswitches is 1.2 and 2.4 $\times$ better in energy-delay product on the uniform random and transpose random traffic compared with the hierarchical crossbar. To further optimize the local network, we exploit the traffic pattern of the global network and propose a

*bilateral bufferfly* organization that removes up to 33% of crosspoints in a folded-Clos – and thus, achieving a more area and energy efficient switch design.

The network within a network approach to building high-radix switch microarchitectures opens opportunities for other future work. Although we focused on the folded-Clos and the HyperX topology, our approach can be extended to other networks. For example, for the recently proposed Dragonfly topology [21], its hierarchical organization can be further exploited to optimize the local network of the switch microarchitecture. In this work, we focused mainly on optimizing the local network organization based on the global network characteristics, but it remains to be seen if both the local and global network can be jointly optimized to reduce overall network cost.

## Acknowledgements

Jung Ho Ahn was supported by the IT R&D program of MKE/KEIT [10038768, The Development of Supercomputing System for the Genome Analysis] and by the Smart IT Convergence System Research Center funded by the Ministry of Education, Science and Technology (MEST) as Global Frontier Project. John Kim was supported by WCU (World Class University) program under the National Research Foundation of Korea and funded by the MEST of Korea (Project No: R31-30007) and by Microsoft Research Asia.

## References

[1] D. Abts, *et al.*, "Energy Proportional Datacenter Networks," in *ISCA*, Jun 2010.

[2] A. Agarwal, "Limits on Interconnection Network Performance," in *IEEE TPDS*, vol. 2, no. 4, Oct 1991.

[3] J. Ahn, *et al.*, "HyperX: Topology, Routing, and Packaging of Efficient Large-Scale Networks," in *SC*, Nov 2009.

[4] B. Arimilli, *et al.*, "The PERCS High-Performance Interconnect," in *HOTI*, Jun 2010.

[5] J. Balfour and W. J. Dally, "Design Tradeoffs for Tiled CMP On-Chip Networks," in *ICS*, 2006.

[6] W. Beyene, *et al.*, "Design and analysis of a TB/sec memory system," in *Electrical Performance of Electronic Packaging*, 2008.

[7] N. Binkert, *et al.*, "The Role of Optics in Future High Radix Switch Design ," in *ISCA*, Jun 2011.

[8] W. J. Dally, "Performance Analysis of k-ary n-cube Interconnection Networks," in *IEEE Transactions on Computers*, Jun 1990.

[9] W. J. Dally and B. Towles, "Route Packets, Not Wires: On-Chip Interconnection Networks," in *DAC*, 2001.

[10] ——, *Principles and Practices of Interconnection Networks*. Morgan Kaufmann Publishers Inc., 2003.

[11] J. Duato, S. Yalamanchili, and N. Lionel, *Interconnection Networks: An Engineering Approach*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2002.

[12] H. Eberle, *et al.*, "High-Radix Crossbar Switches Enabled by Proximity Communication," in *SC*, 2008.

[13] R. Ho, K. W. Mai, and M. A. Horowitz, "The Future of Wires," in *Proceedings of the IEEE*, Apr 2001.

[14] U. Hoelzle and L. A. Barroso, *The Datacenter as a Computer: An Introduction to the Design of Warehouse-Scale Machines*, 1st ed. Morgan and Claypool Publishers, 2009.

[15] N. Jiang, J. Kim, and W. J. Dally, "Indirect Adaptive Routing on Large Scale Interconnection Networks," in *ISCA*, Jun 2009.

[16] A. Joshi, *et al.*, "Silicon-Photonic Clos Networks for Global On-chip Communication," in *NOCS*, Jun 2009.

[17] M. Karol, M. Hluchyj, and S. Morgan, "Input Versus Output Queueing on a Space-Division Packet Switch," in *Communications, IEEE Transactions on*, vol. 35, no. 12, Dec 1987.

[18] J. Kim, J. Balfour, and D. Abts, "Flattened Butterfly Topology for On-Chip Networks," in *MICRO*, Dec 2007.

[19] J. Kim, W. J. Dally, and D. Abts, "Adaptive routing in high-radix clos network," in *SC*, Nov 2006.

[20] ——, "Flattened butterfly: A cost-efficient topology for high-radix networks," in *ISCA*, Jun 2007.

[21] J. Kim, *et al.*, "Technology-Driven, Highly-Scalable Dragonfly Topology," in *ISCA*, Jun 2008.

[22] ——, "Microarchitecture of a High-Radix Router," in *ISCA*, Jun 2005.

[23] S. Li, *et al.*, "McPAT: An Integrated Power, Area, and Timing Modeling Framework for Multicore and Manycore Architectures," in *MICRO*, Dec 2009.

[24] N. McKeown, "The iSLIP scheduling algorithm for input-queued switches," in *IEEE/ACM Transactions on Networking*, Apr 1999.

[25] N. Minura, *et al.*, "An 8Tb/s 1pJ/b 0.8mm$^2$/Tb/s QDR Inductive-coupling Interface between 65nm CMOS GPU and 0.1um DRAM," in *ISSCC*, Feb 2010.

[26] G. Mora, *et al.*, "Towards an Efficient Switch Architecture for High-Radix Switches," in *ACM/IEEE Symposium on Architecture for Networking and Communications Systems*, 2006.

[27] C. Nicopoulos, V. Narayanan, and C. R. Das, *Network-on-Chip Architectures*. Springer, 2010.

[28] F. O'Mahony, *et al.*, "A 47x10Gb/s 1.4mW/(Gb/s) parallel interface in 45nm CMOS," in *ISSCC*, Feb 2010.

[29] D. Roweth and T. Jones, "QsNetIII an Adaptively Routed Network for High Performance Computing," in *HOTI*, Aug 2008.

[30] S. Scott, *et al.*, "The BlackWidow High-Radix Clos Network," in *ISCA*, Jun 2006.

[31] Semiconductor Industries Association, "International Technology Roadmap for Semiconductors," http:// www.itrs.net, 2009 Edition.

[32] A. Singh, "Load-Balanced Routing in Interconnection Networks," Ph.D. dissertation, Stanford Univ., 2005.

[33] I. E. Sutherland, R. F. Sproull, and D. F. Harris, *Logical Effort: Designing Fast CMOS Circuits*. Morgan Kaufmann, 1999.

[34] L. G. Valiant, "A Scheme for Fast Parallel Communication," *SIAM Journal on Computing*, vol. 11, no. 2, 1982.

[35] H.-S. Wang, L.-S. Peh, and S. Malik, "A Power Model for Routers: Modeling Alpha 21364 and InfiniBand Routers," in *HOTI*, Aug 2002.

[36] C.-K. K. Yang, "Design of High-speed Serial Links in CMOS," Ph.D. dissertation, Stanford Univ., 1998.

[37] Y. Zhang, *et al.*, "Prediction of High-Performance On-Chip Global Interconnection," in *SLiP*, 2009.

[38] W. Zhao and Y. Cao, "New Generation of Predictive Technology Model for Sub-45nm Design Exploration," in *International Symposium on Quality Electronic Design*, Mar 2006.