Virtual Channels and Multiple Physical Networks: Two Alternatives to Improve NoC Performance

Young Jin Yoon, Nicola Concer, Michele Petracca, and Luca P. Carloni

***Abstract*—Virtual channels (VC) and multiple physical (MP) networks are two alternative methods to provide better performance, support quality-of-service, and avoid protocol deadlocks in packet-switched network-on-chip design. Since contention can be dynamically resolved, VCs give lower zero-load packet latency than MPs; however, MPs can be built with simpler routers and narrower channels, which improves the target clock frequency, power dissipation, and area occupation. In this paper, we present a comprehensive comparative analysis of these two design approaches, including an analytical model, synthesis-based designs with both FPGAs and standard-cell libraries, and system-level simulations. The result of our analysis shows that one solution does not outperform the other in all the tested scenarios. Instead, each approach has its own specific strengths and weaknesses. Hence, we identify the scenarios where each method is best suited to achieve high performance, very low power dissipation, and increased design flexibility.**

***Index Terms*—multiplane, multiple physical networks, network-on-chip (NoC), virtual channel.**

I. Introduction

The increasing number of heterogeneous cores for general-purpose chip multiprocessors (CMP) [1] and systems-on-chip (SoCs) [2], [3] leads to a complex variety of on-chip communication scenarios where multiple applications running simultaneously, trigger the exchange of various messages across processors, accelerators, cache memories, and memory controllers. Consequently, the next generation of networks-on-chip (NoC) must not only provide high-performance and energy-efficient data delivery but also cooperate with the network interfaces of the embedded cores to meet special requirements such as message-class isolation and real-time data delivery. In order to design NoCs that provide both correct (e.g., deadlock free) communications and high-performance data delivery, the literature offers two main approaches: virtual channel (VC) flow control [4], and multiple physical networks or multiplanes (MP) [5]–[8].

Combined with virtual circuit switching or any packet switching technique such as wormhole and virtual cut-through [9], a router that supports VC flow control has multiple buffers per input port and a logical channel assigned to each buffer. Flits from the upstream router are delivered with a logical channel identifier. Based on the identifier value, the downstream router can separately store the flits that use the same physical channels but come from different packets. VC flow control was initially designed to avoid routing deadlock up to the number of provided logical channels. But it can also be used to improve the maximum sustained throughput, to manage quality-of-service (QoS) [10], and to avoid protocol deadlock caused by message-dependency chains [11]. However, the more virtual channels in VC flow control, the more complex the router logic. This typically increases the delay of critical path, dissipates more power, and occupies more area.

Instead of having a single network with the complex allocation logic necessary to support VC flow control on large channels, it is possible to use simpler flow control mechanisms and partition the channel widths across multiple independent and parallel networks. This leads to MP NoCs, which can be designed to have smaller power dissipation and area occupation by leveraging the fact that they consist of many simpler networks operating independently.

Fig. 1(a) and(b) illustrate the main differences between a VC NoC and an equivalent MP NoC, particularly with respect to handling possible congestion when multiple packets simultaneously traverse the network. This example assumes that the two NoCs have the same aggregate channel width, that is, the sum of the widths of the two MP channels equals the VC's one. The three orange rectangles (the left-most and the two top ones) illustrate network interfaces (NI) as traffic sources, and the two light-blue rectangles (the bottom and right-most ones) show the NIs as traffic sinks. The two and four rounded squares in the middle (purple) represent routers using VCs [Fig. 1(a)] and MPs [Fig. 1(b)], respectively. When a packet remains blocked because of back-pressure, VC routers can exploit their additional buffers to improve the channel utilization. For instance, in Fig. 1(a) if the packet of message α is locked by contention on channel C_{v_2} while also occupying channel C_{v_1}, then the packets of message β can still advance by exploiting the second VC supported by the router. In the equivalent MP NoC of Fig. 1(b), instead, the channels are partitioned into two subnetworks (e.g., C_{v_1} into $C_{p_{11}}$ and $C_{p_{12}}$). Although the width of each channel is reduced by half, the number of bits transmitted per clock cycle remains the same. Differently from the VC case, here the packets of messages

Manuscript received August 14, 2012; revised June 19, 2013; accepted July 13, 2013. Date of current version November 18, 2013. This work was supported in part by the National Science Council, R.O.C., under contract NSC 97-2221-E-008-095-MY3 and contract NSC 97-2221-E-008-094-MY3, in part by the Gigascale Systems Research Center (GSRC) and the Center for Future Architectures Research (C-FAR), in part by the STARnet phase of the Focus Center Research Program (FCRP), and in part by a Semiconductor Research Corporation program sponsored by MARCO and DARPA. This paper was recommended by Associate Editor R. Marculescu.

A portion of this work was published in the IEEE/ACM Design Automation Conference (DAC), 2010.

Y. Yoon and L. P. Carloni are with the Department of Computer Science, Columbia University, New York, NY 10027 USA (e-mail: youngjin@cs.columbia.edu; luca@cs.columbia.edu).

N. Concer is with NXP Semiconductors, 5656 AE Eindhoven, The Netherlands (e-mail:nicola.concer@nxp.com).

M. Petracca is with Cadence Design Systems, San Jose, CA 95134 USA (e-mail:petracca@cadence.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCAD.2013.2276399

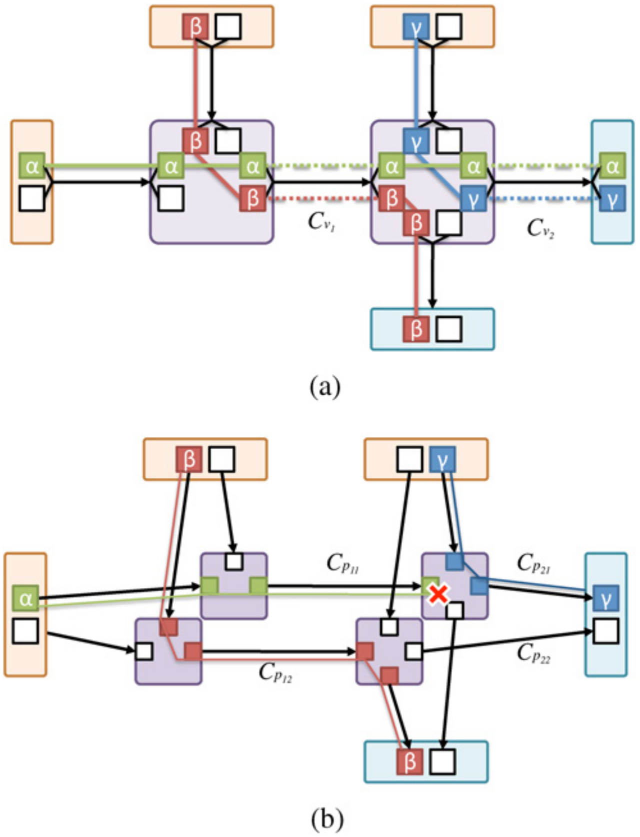


Fig. 1. (a) Example of contention resolution in MP and VC NoCs. (b) Multiple physical NoC (MP).

α and β can be routed on two different partitions and processed in parallel by two different sets of routers.

Thanks to the narrower flit-width, which impacts each router crossbar size in a quadratic way, the total area occupied by MP routers is smaller than the total area of the equivalent set of VC routers. Moreover, routers with VC flow control integrate additional logic modules to maintain and arbitrate the VC status and scheduling of the input and output ports. This additional logic increases the complexity of their design and, in turn, limits the frequency at which they can operate. Indeed, typically the logic implementing the packet-header processing is on the router's critical path and it determines the clock frequency of the overall NoC. Instead, an MP NoC exploits the parallel processing of the packet headers and the limited complexity of the routers logic to improve the overall performance by enabling operations at higher clock frequencies. Each plane of an MP NoC, however, has a reduced flit-width that increases the serialization latency because of the larger number of frames composing a packet.

Contributions. In this paper, we present a comprehensive comparative analysis of VC and MP NoCs and illustrate their strength and weaknesses for different application scenarios. We consider power dissipation, area occupation, and performance evaluation across multiple technology libraries, traffic patterns, and applications. In doing so, we extend and complete our preliminary results described in a short paper presented at DAC'10 [12]. In Section II, we offer a more complete discussion of the related work and highlight the novelty of our contributions with respect to the existing literature. In Section III, we present our comparison methodology along with the details of both VC and MP router architectures. In Section IV, we compare the two NoC design methods by using an analytical model that allows us to estimate the area occupancy and the minimum communication latency of the packets traversing the two networks. In Section V, we extend our previous results by completing RTL logic syntheses and technology mapping with three different standard-cell libraries (for 90 nm, 65 nm, and 45 nm processes) and one FPGA platform to compare critical path delay, area occupation, and power consumption of VCs and MPs. Differently from our previous work [12], we analyze multiple RTL designs across a range of possible target clock periods. Additionally, the power analysis is based on netlist simulations that can achieve a much more accurate power estimation. In Section VI, we enrich the analysis of the network performance with open-loop simulations considering multiple synthetic traffic patterns and network topologies such as 8 × 8 Mesh and 16-node Spidergon, in addition to the analysis of the 4 × 4 Mesh which we had previously presented. Finally, in Section VII, we discuss extensively two case studies: a 16-core CMP and a 64-core CMP running the SPLASH-2 [13] and PARSEC [14] benchmark suites on top of the Linux operating system. These case studies confirm how to consider the options offered by VC and MP for NoC design provides a richer design space in terms of area-power-performance trade offs and increased design flexibility.

II. Related Work

Virtual channels and multiplanes have been extensively used to design and optimize system-area networks and NoCs. The Alpha 21364 processor uses VCs with virtual cut-through flow control to avoid both message-dependent and routing deadlocks in system-area networks [15]. Examples of MP NoCs include the RAW processor that contains two static- and two dynamic-routing NoCs [16], and the Tilera Tile64 processor that has five parallel mesh networks for NoCs [8]. The reason for implementing physically-separated networks and using different routing schemes is to accommodate different types of traffic in general-purpose systems.

The *Æthereal* NoC uses virtual channels to support best-effort service with wormhole flow control and guaranteed service with circuit-switching [10]. Some automated NoC design frameworks such as ×*pipesCompiler* [17] include VC flow control to support quality of service or better performance. Nicopoulos *et al.* [18] and Lai *et al.* [19] propose dynamic virtual channel architectures to improve network performance by adjusting the number of virtual channels in the routers based on the degree of congestion. Both papers show that while reducing the number of VCs improves buffer utilization, for a given buffer storage to increase the total number of VCs is an efficient performance optimization to handle heavy congestion.

The use of various routing scenarios affect power and performance of VCs and MPs. Shim *et al.* [20] explore the performance trade-offs of static and dynamic VC allocation for various oblivious routing methods, including DOR, ROMM, Valiant and a novel bandwidth-sensitive oblivious routing scheme (BSORM).

Balfour and Dally present a comprehensive comparative analysis of NoC topologies and architectures where they discuss the idea of duplicating certain NoC topologies, such as Mesh and *CMesh*, to improve the system performance [21]. Carara *et al.* [5] propose a router architecture to replicate the physical networks by taking advantage of the abundance of wires between routers and compare this solution to the VC approach. Our work differs from these analyses because, instead of duplicating the NoC, we actually partition it into a number of subnetworks while keeping the overall amount of wire and buffering resources constant.

Grot *et al.* [6] propose the multidrop express channels (MECS) topology and discuss an implementation based on two parallel and partitioned networks (MECS-X2). Their work, however, focuses on this specific implementation and does not consider multiplane as an alternative design point to VCs.

Kumar *et al.* [7] show how the poor channel utilization introduced by concentration can be mitigated by channel slicing in NoC but do not include a comparison with corresponding VC implementations.

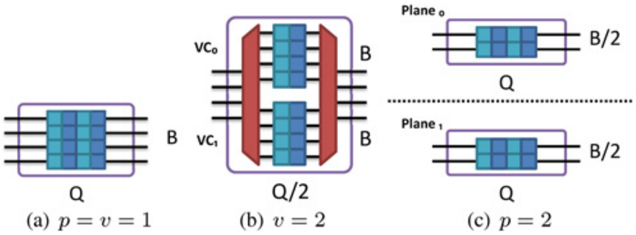
Similar to the multiplane NoC, Teimouri *et al.* [22] divide the n-bit wide network resources in a router, such as links, buffers, and a crossbar, into two parallel $n/2$-bit sub-networks to support reconfigurable shortcut paths. Gomez *et al.* [23] divide the wires into several parallel links connecting to the same two routers to improve the network throughput while improving area occupation and power dissipation. Volos *et al.* [24] present cache-coherence network-on-chip (CCNoC), a specialized architecture that combines asymmetric multiplane and virtual channels to provide efficient support for cache coherence communication. Differently from these works, we do not focus on the analysis of optimized architectures. In order to provide a fair comparison between MPs and VCs, in our analysis the subnetworks (planes) in MPs do not use virtual channels, and they are completely isolated from each other.

Noh *et al.* [25] propose a multiplane-based design for a VC-enabled router where the internal crossbar switch is replaced with a number of parallel crossbars (planes) that increase the flit transfer rate between input and output ports. The resulting router has a simpler design that performs better than a single-plane router with a larger number of VCs. However, Noh *et al.* maintain the flit-width constant as they scale the number of additional lanes, which is different from our analyses.

Gilabert *et al.* [26] propose a new VC implementation, called multiswitch, and compare it to a multiplane-based NoC and a traditional VC implementation called multistage VCs. They argue that the multiswitch approach provides better performance than an equivalent multinetwork only with small area overhead. Their experiments show the power analysis based on two switching activity profiles: 50% and idle. Instead, we include a detailed power analysis with simulation-based switching activity across multiple target clock periods and three different technology generations. We also present a scenario where MPs achieve a better performance than VCs (Section VI). Finally, we present the experimental results of full-system closed-loop simulations for two case studies with heterogeneous partitioning to demonstrate why multiple physical networks can be an efficient solution in terms of area-power-performance trade-offs to build a system (Section VII).

III. Comparison Methodology

While both VC and MP approaches can be combined with any type of buffered flow control, in our paper we focus on wormhole flow control, a very common protocol for NoC implementations. Table I summarizes the key parameters used in our comparison. For a fair comparison between MP and VC NoCs, we keep the aggregated channel width B (a.k.a. flit width) constant, thereby providing the same bisection bandwidth. For example, the VC router with two virtual channels in Fig. 2(b) has channel width $B = 4$, while the corresponding two routers in the two separate planes in Fig. 2(c) have total aggregated channel width $B_{MP} = 2 + 2 = 4$.



(a) $p = v = 1$ (b) $v = 2$ (c) $p = 2$

Fig. 2. Storage and channel allocations with the parameters under competitive sizing. (a) Reference NoC. (b) Virtual-channel NoC with $v = 2$. (c) Multiplane NoC with $p = 2$.

TABLE I

NoC Parameters Used in Our Study

Terms	Definitions
B	channel width per port
Q	buffer depth in number of flits
S	input storage per port
p	number of physical channels
v	number of virtual channels

Besides keeping B constant, we also maintain the aggregated input storage S constant for the comparison. For example, starting from a reference wormhole flow-control router with channel width B and input-buffer depth Q [Fig. 2(a)], the corresponding VC router with two virtual channels $v = 2$ has buffer depth $Q/2$ and channel width B [Fig. 2(b)]. Instead, the corresponding set of $p = 2$ parallel MP routers for MP NoCs have buffer depth Q and channel width $B/2$ [Fig. 2(c)]. This comparison approach, which maintains both B and the total storage S constant, is called competitive sizing.

Competitive sizing is the fair way to compare MPs and VCs as alternative solutions that use the same amount of resources to build a NoC. We found, however, some interesting design points where given a small amount of storage resources a NoC can be designed based on MPs, but not on VCs. Specifically, since the buffer depth of the VC router with $v = n$ is Q/n, there are some configurations where MP NoCs are still feasible while VC NoCs are not due to the insufficient buffer depth. Thus, we also consider a configuration where MP and VC NoCs are configured with the minimum possible storage. This configuration is important for very low-power SoCs and highlights the capabilities of the two alternatives. We call this analysis minimum sizing comparison.

Competitive Sizing. For competitive sizing, we first define a reference NoC architecture based on a wormhole router where the input storage at each port is $S = B \times Q$ bits. This can be seen either as a VC router with one virtual channel ($v = 1$) or an MP router for a single-plane NoC ($p = 1$). Then, we vary the number v of virtual channels [Fig. 2(b)] and number of planes p [Fig. 2(c)] by partitioning the available storage S according the following rules: 1) the buffer size of the i^{th} virtual channel in a VC router is $Q_i = Q/v$ and 2) the aggregated channel width of an MP router is $\sum_{i=1}^{p} B_i = B$. Notice that these rules force S and B to remain constant across all possible MP and VC NoC configurations. Additionally, to set $B_i = B/p$ makes the channels of the WH router be homogeneously partitioned in an MP NoC. With these rules,

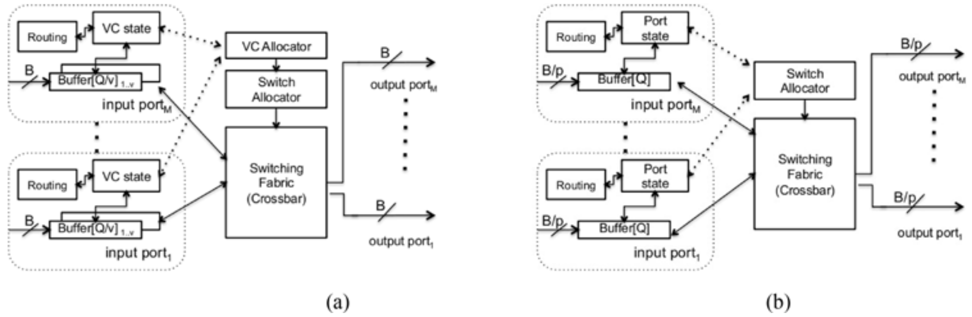


Fig. 3. Block diagrams of a VC router and a MP router inspired by [27]. (a) VC router. (b) MP router.

each input port of the VC router contains v buffers, each of which has buffer depth Q/v. Thus, the total amount of storage used to implement a VC NoC is $S = B \times v \times Q/v = B \times Q$. For homogeneously-partitioned MP NoCs, each MP router has channel width B/p and one buffer per input port with buffer depth Q. Thus, the total amount of storage used to implement MP NoCs is $S = B/p \times p \times Q = B \times Q$.

Since VCs and MPs with competitive sizing can be constructed with the same amount of resources to build a NoC, all comparisons starting from Section IV will present the results with competitive sizing unless otherwise specified.

Minimum sizing. The most common flow control protocol on router-to-router links in an NoC is credit-based flow control, which uses credits to allow the upstream router to keep track of the storage availability in the input buffer of the downstream router. In order to guarantee minimal zero-load latency, this protocol imposes a constraint on the minimum size of the router input buffer, which should be at least equal to the round-trip-time (RTT) of one credit on the link. In the best case, the RTT is equal to two clock cycles, that is, $Q_{min} = 2$ (minimum sizing constraint) [4]. If we build the least expensive p-plane MP NoC that satisfies such requirement, the aggregated storage is $S = 2 \times \sum_{i=1}^{p} B_i = 2 \times B$. Instead, for the corresponding VC NoCs the minimum aggregated storage becomes $S = 2 \times v \times B$ because each virtual channel at each input port must satisfy the minimum sizing constraint.

While longer buffers generally increase the maximum sustained throughput of an NoC, an application often need much less throughput. In these cases, longer buffers cause area and power overheads without providing a comparable improvement in the overall system performance. Under these conditions, an MP NoC with minimum sizing could potentially deliver enough performance while saving power and area with respect to an equivalent VC-based NoC.

IV. Analytical Model

We analyze the overhead of MPs and VCs with respect to the reference wormhole architecture under competitive sizing, in terms of area and performance. First, we derive the relative area of each component at the microarchitecture level. Then, we analyze the impact on packet-transmission latency of MP routers, which is critical to the performance of MP NoCs due to the narrower channel width.

A. Microarchitectural Analysis

Fig. 3(a) shows the block diagram of a classic M-port VC router for a 2-D Mesh network. Each input/output port is connected to a physical channel that has a data parallelism of B bits, which matches the flit size. In a VC router with v virtual channels each input port is equipped with: 1) a routing-logic block that determines the destination port for each packet based on the information contained in the head flit and the specific routing algorithm (e.g., XY routing); 2) a set of v buffers, each dedicated to one virtual channel; and 3) a VC control block that holds the state needed to coordinate the handling of the flits of the various packets. When a header flit arrives to an input port, a VC allocator arbitrates the matching between input and output virtual channels. After the arbitration, a switch allocator controls the matching between multiple input ports to one output port through the switching fabric. In a VC router with v virtual channels under competitive sizing, a VC allocator with the size of $(M \times v)^2$ is required to allocate an output virtual channel per packet. Also, additional control wires need to be placed to send and receive the virtual channel identifier, whose size is proportional to $\log_2 v$.

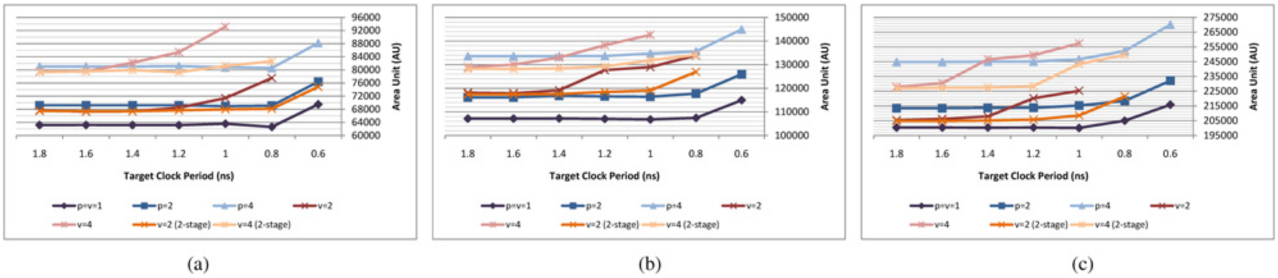
TABLE II
Area Comparison of Wormhole Router, MPs and VCs

Component	wormhole	MPs	VCs
Switching Fabric	$(B \times M)^2$	$(B \times M)^2/p$	$(B \times M)^2$
Switch Allocator	M^2	$M^2 \times p$	M^2
VC Allocator	–	–	$(M \times v)^2$
Control Channel	$M \times x$	$M \times x \times p$	$M \times (x + 2 \times \log_2 v)$

This is a comparison among the reference wormhole router, MPs with p, and VCs with v, each of which contains M input and output ports. A channel in the wormhole router contains a x-bit control channel.

Fig. 3(b) shows the block diagram of an MP router that can be used on each plane of a multiplane 2-D Mesh NoC. The structure of this router is simpler than an equivalent VC router because it implements the basic wormhole flow control with a single queue per each input port and, therefore, does not need a VC allocator. In an MP NoC with p planes each router contains a switching fabric of size of $(B \times M/p)^2$ and a local copy of a switch allocator equal to the one in the reference wormhole router. Further, a control channel is also required per plane to maintain flow control. Thus, the aggregated size of the switching fabrics is $(B \times M/p)^2 \times p = (B \times M)^2/p$. The aggregated sizes of the control channel and switch allocator are $M \times p \times x$ and $M^2 \times p$, respectively.

Table II summarizes the area comparison between the MP and VC routers with respect to the reference wormhole router from the viewpoint of this analytical model. Note that the actual area of both MPs and VCs depends on the implementation details of each module and on the effectiveness of CAD tool optimization. For example, one can implement an area-efficient VC allocator by sacrificing the input-output virtual channel matching ratio as discussed in [28]. The analytical



(a)

(b)

(c)

Fig. 4. Area occupation as function of target clock period ($B = 128$, 45 nm). (a) $S = 1024$ ($Q_{WH} = 8$). (b) $S = 2048$ ($Q_{WH} = 16$). (c) $S = 4096$ ($Q_{WH} = 32$).

model, however, illustrates some basic differences between the two micro-architectures: MPs benefit from smaller switching fabrics, while requiring additional switch allocators and control channels. VCs do not save any area with respect to the wormhole router, and need additional logic to implement the VC allocator. Note that the area of the network interfaces (NI) for both VCs and MPs is identical. If the NIs contain a separate buffer per virtual channel or plane with the same allocation scheme, that is, round-robin arbiters, the only differences between the VC and MP NIs consist in the logic that serializes a packet into multiple flits. However, we found that if the storage capacity in both NIs is the same then there is no area difference between these two interfaces.

Another important metric of comparison is the internal critical path, which may constrain the maximum clock frequency at which the router logic can operate, as well as the maximum transmission rate on a router-to-router link and the latency of the packets flowing through the router. In VC routers, the critical path always traverses the VC allocator because its logic must interact with all input and output virtual channels. Instead, an MP router does not have VC allocation logic, and therefore has a shorter critical path. As a result, MP routers can be clocked at higher frequencies than the corresponding VC routers.

B. Latency Analysis

Latency is a key performance metric for NoCs. Latency T is the time required for a packet to traverse the network, and can be divided into head latency T_h, serialization latency T_s, time of flight on wires T_w, and contention latency T_c [4].[1]

In order to achieve a minimal latency, all components of the equation must be well balanced. For example, for a given bisection bandwidth b, a NoC with a Flatten Butterfly topology [29] uses more channels compared to a traditional 2-D-Mesh NoC. With more channels, a Flatten Butterfly reduces the head latency T_h with low average hop count but increases serialization latency T_s due to the limited bisection bandwidth: by rebalancing T_h and T_s, it achieves low latency without sacrificing the network throughput. Similarly, for MP routers with $p = n$ the serialization latency T_s becomes n times bigger than the latencies for the reference wormhole and VC routers where $p = 1$. As discussed in Section V, however, MP routers can achieve a higher clock frequency, which will result to a shorter T_h.

For small or less-congested networks, T_s strongly contributes to the total latency T due to relatively small T_h or T_c. However, when the congestion of the networks increases or the networks becomes large, T_h and T_c become dominant in the total latency (e.g., more than T_s). In this case, using higher clock frequency reduces not only T_h but also T_c by helping to reduced congestion in the routers. Thus, MPs allow NoC designers to rebalance T_h, T_s, and T_c to optimize latency.

V. Synthesis-Based Analysis

In order to compare the area occupation and the power dissipation of the two NoC architectures, we first synthesized routers for MPs and VCs starting from the NETMAKER library of router implementations [30] and using standard cells from an industrial library while scaling the technology process from 90 to 65 and 45 nm. To derive an optimal design of the wormhole router we augmented the NETMAKER library with the options of disabling the generation of the logic necessary to support virtual channels and of placing additional flip-flops on the switch allocator to keep track of input-output allocations. We verified the correctness of this design by running RTL and post-synthesis netlist simulations with the test-bench provided in the library.

We used a Synopsys design compiler for logic synthesis, and Cadence incisive simulation suite to capture the switching activity of the input ports of a router. Our simulation setup features a 2 × 2 2-D-Mesh topology with a 0.4 offered traffic load (roughly the saturation throughput of 2-D-Mesh) under Uniform-Random Traffic (URT).[2] Differently from our previous work [12], we back-annotated the activity extracted from simulation into Synopsys Primetime PX to estimate the power dissipation.

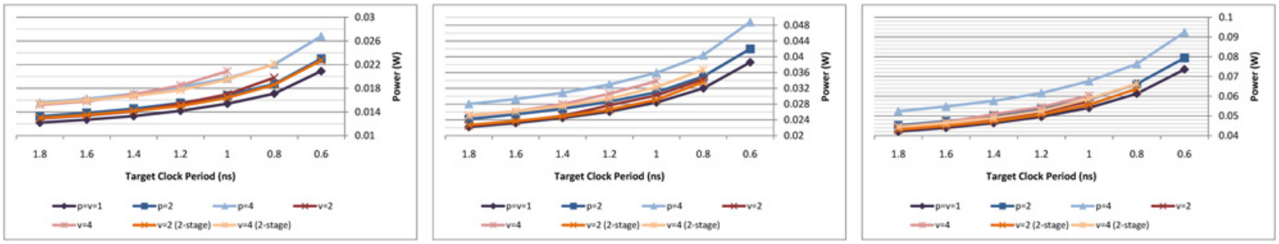
For this analysis, we varied the number of virtual channels v and physical planes p in $\{1, 2, 4\}$, while keeping the same total amount storage S. To analyze the behavior of both MP and VC routers under different storage and channel-width constraints, we also varied Q in $\{2, 4, 8, 16, 32\}$, and the channel width B in $\{64, 128, 256\}$. Moreover, we synthesized each router with different target clock periods T_{CLK}. Specifically, starting from $2.8ns$ we decreased T_{CLK} by a $0.2ns$ step until neither the MP nor the VC router could be successfully synthesized.

A. Area Analysis

Fig. 4 reports the area as function of T_{CLK} with 45 nm technology node. For brevity, we do not report the results for

[1] Contention latency T_c depends on traffic injection behavior of the network and is hard to generalize without any traffic assumptions. Thus, instead of analyzing T_c, we present simulation results with various traffic patterns in Section VI

[2] Notice that the results presented in this section can be applied with different sizes of 2-D-Mesh when normalized. As discussed in [12], the normalized results are sufficient to illustrate the behavior for larger $n \times n$ 2-D-Mesh NoCs. Differently from the results in [12], here we present absolute numbers to show clear trends with respect to different target clock periods.



(a) (b) (c)

Fig. 5. Power dissipation as function of target clock period (B = 128, 45 nm). (a) S = 1024 (Q_{WH} = 8). (b) S = 2048 (Q_{WH} = 16). (c) S = 4096 (Q_{WH} = 32).

65 and 90 nm technologies because they follow a similar trend as 45 nm. Along with the total amount of storage S in the caption of each sub-figure, we also provide the input-buffer depth of the baseline wormhole router. Q_{WH}[3]

Independently from the technology node, as we lower T_{CLK}, the area of VC routers rapidly increases with respect to the corresponding MPs. This difference is due to the higher complexity of the logic implementing the scheduling and arbitration of the different queues inside the VC routers. To meet very low T_{CLK} value the synthesis tool uses bigger gates that can switch quickly but also occupy more area.

Furthermore, after a rapid increment in the area occupation, VC routers become no longer synthesizable, while the equivalent MP routers still are. Since VC routers are more complex than MP routers, their internal critical path are typically longer, thus preventing implementations that can match the same low T_{CLK} requirements as MP routers. Only after T_{CLK} reaches a limit of $T_{CLK} = 0.8ns$ (at 45 nm), MP routers experience a rapid increment in their area occupation as well, but this happens for T_{CLK} values that are 25% smaller for $v = p = 2$ and 40% smaller when $v = p = 4$ [Fig. 4(a)].

When the total amount of storage is small, $Q_{WH} \leq 8$, the area occupation of MP routers is always lower than the equivalent VC routers at low T_{CLK}. For some configurations with $Q_{WH} \leq 8$, MP routers start from having bigger area than the equivalent VC routers for high T_{CLK} but they eventually occupy less area as we decrease T_{CLK}. For longer queues *i.e.* $Q_{WH} \geq 16$, the area of MP routers is always worse than that of the equivalent VCs routers under high T_{CLK}. As we decrease T_{CLK}, however, VC routers start occupying more area than MP routers and, eventually, they cannot even be synthesized for very low T_{CLK}.

In summary, when routers use short queues, MP routers are always smaller than VC routers and their area increases more linearly when varying T_{CLK}. This property is particularly interesting in the context of dynamic voltage-frequency scaling (DVFS) techniques. In fact, to fully support DVFS, every router should be designed for the lowest possible T_{CLK} of the system. MP routers not only have a larger span of possible T_{CLK} values, but also achieve low T_{CLK} with smaller area penalty.

B. Power Analysis

Fig. 5 shows power consumption versus T_{CLK} of both VC and MP implementations at 45 nm. Again, for brevity, we do not report results for 65 and 90 nm technology nodes because the normalized power of MP and VC routers with respect to the reference wormhole routers is almost invariant across various T_{CLK}.

TABLE III

NORMALIZED AREA AND POWER WITH MINIMUM SIZING OF Q = 2

(a) Normalized area

		p=2	p=4	v=2	v=4
$45nm$	B=64	1.38	1.83	2.21	5.22
	B=128	0.98	1.35	1.79	3.52
	B=256	1.06	1.04	1.70	3.06
$65nm$	B=64	1.20	1.55	1.78	3.92
	B=128	1.11	1.33	1.62	3.23
	B=256	1.05	1.17	1.56	2.79
$90nm$	B=64	1.19	1.60	1.79	3.94
	B=128	1.13	1.35	1.75	3.42
	B=256	1.06	1.18	1.56	2.79

(b) Normalized power

		p=2	p=4	v=2	v=4
$45nm$	B=64	1.20	1.62	1.86	4.07
	B=128	1.12	1.35	1.77	3.45
	B=256	1.04	1.17	1.70	3.11
$65nm$	B=64	1.20	1.65	1.83	3.86
	B=128	1.13	1.35	1.76	3.48
	B=256	1.05	1.18	1.70	3.19
$90nm$	B=64	1.20	1.66	1.84	3.86
	B=128	1.13	1.35	1.75	3.42
	B=256	1.07	1.21	1.68	3.16

Differently from the area analysis, the total power of all routers tend to increase as we lower T_{CLK} due to the power dissipation of the clock-tree, which accounts for the 62.2% of the total dissipated power at 45 nm. Similarly to the analysis of Section V-A, MP routers dissipate less power than the equivalent VC routers for low values of T_{CLK} when $Q_{WH} \leq 8$. Instead, with deeper buffer depth, such as Q_{WH} = 32, MP routers dissipate more power with high T_{CLK}.

Although we did not show the results with 90 nm and 65 nm, the power differences between MP and VC routers are more significant in 45 nm than 90 nm and 65 nm due to the large leakage power that characterizes this technology. Furthermore, having large leakage power in 45 nm is also highly related to the different amount of logic gates in MP and VC routers. From our analyses, at 90 nm and 65 nm, the power dissipated by the clock-tree represents up to 85.6% of the total power under those technology nodes. Instead, as we scale the technology down to 45 nm, the contribution of leakage to the total power dissipation becomes more noticeable while, correspondingly, the portion of clock-tree power decreases to 62.2%.

In summary, the difference of power dissipation as function of T_{CLK} between MP and VC NoCs is negligible with 65 and 90 nm technologies but becomes more relevant with the scaling of technology processes due to the impact of leakage power. Increasing the number of planes p or virtual channel v comes with a power overhead. With $Q_{WH} \leq 8$, MP routers dissipate equal, or less, power than the equivalent VC routers. Instead, when the amount of storage is higher, then VC routers generally have a lower power overhead. In that case, however, VC routers cannot be synthesized for low values of T_{CLK}.

C. Analysis with Architectural Optimizations of VC Routers

In order to achieve higher clock frequencies, the logic controlling VC routers can be pipelined into multiple stages.

[3] Recall that the buffer depth of a VC router with v virtual channels is Q_{WH}/v, while MP routers have buffer depth Q_{WH} for each plane

Figs. 4 and 5 also report results for 2-stage implementations of VC routers that allow us to compare them with the equivalent MP and 1-stage VC routers in terms of area occupation and power dissipation, respectively. In both analysis we scaled T_{CLK} while using a 45 nm technology and setting $B = 128$. Since pipelining divides the internal critical path into multiple stages, the synthesis tool has more options to optimize the router for area and power. In fact, we can observe that 2-stage VC routers can be synthesized for lower T_{CLK} than 1-stage VC routers and that in general they occupy less area. On the other hand, they do not necessarily outperform the equivalent MP routers. MP routers can still: 1) be synthesized at T_{CLK} at which 2-stage VC routers cannot be synthesized, and 2) save area and power at low T_{CLK} for small values of Q_{WH}. Furthermore, having an extra pipeline stage increases the head latency H_s of the entire packet because it takes two clock cycles to process a flit in a 2-stage pipelined VC router. Hence, the equivalent MP routers perform better than the pipelined VC routers in terms of average latency. Another possible micro-architecture optimization proposed for VC routers is speculation. The purpose of pipeline speculation is to reduce the latency under low injection rate by reducing the effective number of pipeline stages [27]. Hence, it is not surprising that the synthesis results for speculative VC routers in terms of area, power and T_{CLK} are very similar to those presented for the nonspeculative VC routers (and are not included here for the sake of brevity).

D. Effect of Channel Width

To analyze the effect of changing the aggregated channel width B, we extended the studies of $B = 128$ to the cases of $B = 64$ and $B = 256$.[4]

For $B = 256$, the area and power overheads for MP routers are lower than those for $B = 64$ or $B = 128$. Since each plane contains a control channel implemented as a bundle of control wires, its associated overhead increases as the channel width narrows. Still, the main observations derived from the comparison between VC and MP routers in terms of both area occupation and power dissipation remain valid for the new values of B. Specifically, when targeting a low clock period T_{CLK} across all three values of B, MP routers occupy less area if $Q_{WH} \leq 16$ and dissipate less power if $Q_{WH} \leq 8$.

E. Analysis with Minimum Sizing

For those systems where the connectivity is more critical than the throughput, designers might want to place the minimum amount of buffers per input port in order to save area and power. In such systems, having VC routers may not be as area- and power-efficient as having MP routers with minimum buffer size; as anticipated in Section III, this motivates us to compare MPs to VCs under minimum sizing.

Table III shows the area and power normalized to the reference wormhole router under minimum sizing ($Q = 2$). T_{CLK} is set to the lowest possible value at which all VCs and MPs are synthesizable, i.e. $1ns$ for 45 nm, and $1.6ns$ for 65 and 90 nm. From Table III(a), the area overhead introduced by MP routers with $p = 2$ is smaller than 38%, while the overhead for an equivalent VC router varies between 56 and 121%. When we increase the MPs and VCs to four the overhead is 83% and $179 - 422\%$ respectively. From Table III(b), the power overhead of MP routers with $p = 2$ and $p = 4$ is less than 20% and varies between 17% and 66%, respectively. Instead, the equivalent VC routers dissipate $58 - 86\%$ and $211 - 306\%$ more than the reference wormhole router.

TABLE IV
DIFFERENCES BETWEEN NETMAKER AND NOCEM

	NETMAKER	NOCEM
HDL	SystemVerilog	VHDL
low-level flow control	credit-based	on-off
router pipelining support	2-stage	N
router speculation support	Y	N
routing-lookahead	default	N
target platform	ASIC	FPGA

F. Synthesis Analysis with FPGA

We also conducted experiments that compare VCs and MPs under competitive sizing with the NOCEM toolkit [31] and an FPGA synthesis tool chain. The differences between NETMAKER and NOCEM are summarized in Table IV. Note that we used NETMAKER for the synthesis with ASIC design flows because it provides more flexibility in configuring the NoC parameters. However, for the synthesis with FPGA design tools, using NOCEM is more appropriate in terms of RTL design and synthesis tool support. We use the Xilinx ISE framework to analyze FPGA-based implementations targeting the Xilinx XC6VLX75T FPGA. To measure the router area we counted the number of used LUTs.

Fig. 6 shows the normalized area and delay of the critical path with respect to the reference wormhole router. Fig. 6(a) shows that MP routers occupy less area than the equivalent VC routers, when the total amount of storage S is small. Finally, Fig. 6(b), shows that an MP router with $p = 4$ planes can run at a clock frequency that is 18 to 35% higher than a VC router with $v = 4$. These results are consistent with the results based on standard-cell libraries discussed in previous subsections: they confirm that across different platforms such as FPGA and ASIC there exist interesting design trade-offs between MP and VC as we vary the aggregated amount of storage S.

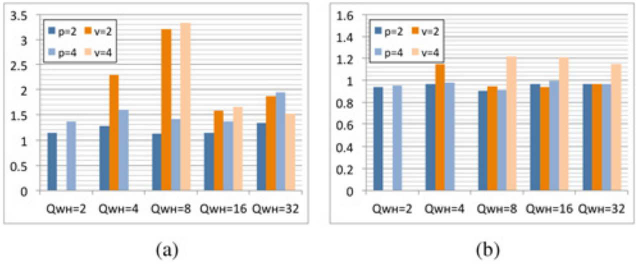
G. Summary

Based on competitive sizing, we find that MP networks scale better than the equivalent VC networks in terms of power dissipations and area occupations, with different technologies, target platforms, and microarchitectural optimizations. When routers use short queues, MP routers are smaller and dissipate less power than VC routers. Furthermore, MPs scale more linearly than VCs in terms of area and power when varying the target clock period, which is interesting in the context of DVFS techniques.

MPs are a power-efficient solution whenever providing basic connectivity among components is more important than optimizing the average latency and effective throughput. Under minimum sizing, MPs have 18∼83% less area overhead, and 38∼66% less power overhead with respect to the baseline wormhole router.

Using MP networks, however, is not always the best solution to design power-efficient NoCs. VC networks occupy less area and dissipate less power when the target clock period is high, the technology is less advanced, and/or when routers

[4] Since results with different channel widths show similar behaviors to Fig. 4 and 5, here we only summarize the results for the sake of brevity



(a)

(b)

Fig. 6. Normalized area and delay of the FPGA synthesis with B = 128. (a) Normalized area. (b) Normalized delay.

use long queues. Furthermore, as discussed in Sections VI and VII, VC networks sometimes perform better than MP in terms of average latency and effective bandwidth. Therefore, NoC designers should choose carefully between MPs and VCs based on the given system requirements.

VI. System-Level Performance Analysis

We developed an event-driven simulator that includes a detailed model of NoC components, such as routers and network interfaces, using the Omnet++ framework [32]. We ran open-loop simulations with synthetic traffic patterns considering only the steady-state data collected over multiple runs[5] We considered 4 × 4 and 8 × 8 2-D meshes and 16-node Spidergon, a well-known NoC composed by a ring enriched by cross-links that connect facing cores [33]. For both MP and VC NoCs, we used the well-known XY routing in the 2-D-Mesh, while for the Spidergon NoC we used the Across-First routing algorithm, which first forwards the flits along the cross links and then along the channels of the ring. Since VCs may be pipelined to support a low T_{CLK}, all routers in both MP and VC NoCs are implemented with a 3-stage pipeline.

We performed a system-level comparative analysis of VC and MP NoCs using four synthetic traffic patterns which allow us to study the network behavior under specific stress conditions: uniform random traffic (URT), Tornado, Transpose, and 4-HotSpot. While the traffic under URT is uniformly distributed among all routers of an NoC, the traffic under Transpose stresses a limited number of hotspot channels with 2-D-Mesh and XY Routing.[6] Moreover, we also include 4-Hotspot and Tornado because they are median traffic patterns between URT and Transpose. Specifically, 4-Hotspot is similar to URT but the randomness is limited to four nodes placed in the center of the Mesh, while in Tornado destinations are predefined for each source but, differently from Transpose, it generates less contention on the NoC channels.

We set the channel width of the reference wormhole router to B = 256, and partition B equally among the number of planes of the MP NoCs (e.g., 64 bits per plane with p = 4). The size of a packet is fixed to 1024 bits.[7] Thus, a total of four flits is sent to the reference wormhole router whenever a packet is generated based on the offered traffic load.[8]

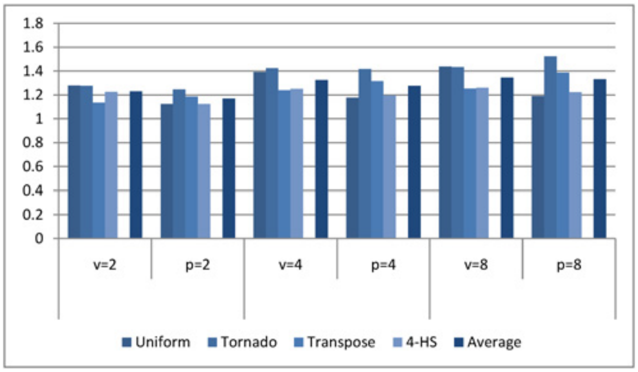


Fig. 7. Normalized maximum sustained throughput.

To compare MPs and VCs in terms of average latency and maximum sustained throughput, we varied the offered load from 0.1 to 1.0 flit/cycle/node. We ran each configuration multiple times with different random seeds and averaged the results before reporting the final value.

A. Throughput

Fig. 7 shows the maximum sustained throughput computed as the minimum load that causes the packet-delivery time to become unstable and increase toward infinity. The values are normalized with respect to the performance of the reference wormhole NoC on a 4×4 2-D-Mesh for the case of competitive sizing of the router queues. In order to obtain a unique reference value that summarizes the performance of the system, we averaged the normalized throughput across different input-buffer sizes. Clearly both VCs and MPs improve the maximum sustained throughput from 17 to 45%, depending on the traffic pattern. VCs improve the performance of wormhole because they reduce the negative impact of head-of-line (HoL) blocking through the multiple queues used on each input port. HoL happens when a header flit gets blocked in a router after losing a channel contention with another packet. VCs enable the interleaving of flits from multiple packets to access the shared physical channel [34]. Hence, when a packet is blocked and cannot advance, the flits of another packet, potentially waiting on a second VC, can be forwarded along the channel that otherwise would have remained idle.

MPs parallelize the forwarding of the packets on multiple physical networks and reduce the negative effects of the head latency T_h (Section IV-B). When T_h is large, it becomes dominant in the delivery of the packet. By using multiple planes, multiple headers can be processed in parallel, thereby speeding up the forwarding of packets and, in turn, the system throughput. Hence, MPs are particularly suited for traffic patterns that do not show much contention but rather a few hot-spot channels.

To highlight the differences in the maximum sustained throughput between the MP and VC NoCs, we define the throughput improvement ratio (TIR) as

$$TIR = 1 - Th(MP_p)/Th(VC_v),$$

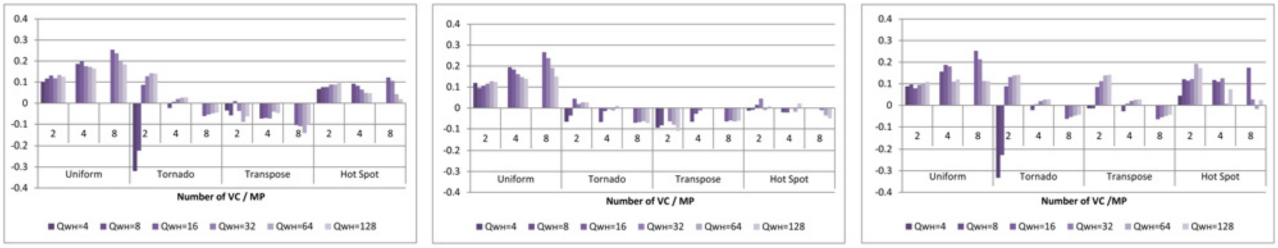
where $Th(VC_v)$ is the maximum sustained throughput of a VC NoC with v virtual channels, and $Th(MP_p)$ the throughput of

[5] Note that the experiments in this section are designed to measure theoretical limits, not to reflect the behavior of MPs and VCs in many real systems. As a more practical example, however, in Section VII we present experimental results for the case study of a Chip Multiprocessor.

[6] With 2-D-Mesh and XY routing, the channels from $\langle 0, 1\rangle$ to $\langle 0, 0\rangle$ and from $\langle 3, 2\rangle$ to $\langle 3, 3\rangle$ are the most contended channels of the network.

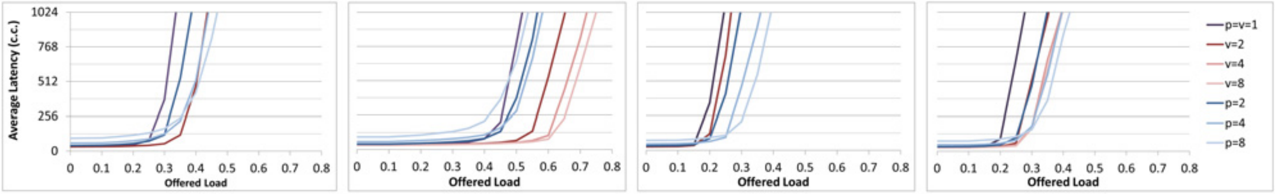
[7] Note that performance is not related to the channel width B but the number of flits per packet. Therefore, results with $B = 256$ are essentially equivalent to the results with $B = 32$, if they have the same number of flits per packet.

[8] Results with packets of 8 and 16 flits show similar behaviors and therefore are omitted.



(a) (b) (c)

Fig. 8. Throughput improvement ratio (TIR). (a) 4 × 4 2-D-Mesh. (b) 8 × 8 2-D-Mesh. (c) 16-node Spidergon.



(a) (b) (c) (d)

Fig. 9. Latency in the 4 × 4 2-D-Mesh with various traffic patterns. (a) URT, Q_{WH} = 4. (b) URT, Q_{WH} = 32. (c) Transpose, Q_{WH} = 4. (d) Transpose, Q_{WH} = 32.

MP NoC with p planes. When $TIR > 0$, a VC NoC provides better maximum sustained throughput than the equivalent MP NoC, while the opposite is true for $TIR < 0$.

Fig. 8(a) compares the values of TIR for different traffic patterns and NoC configurations. Under URT, VC NoCs outperform MP NoCs up to 20% for all configurations. URT does not generate hot-spot channels but it stresses the interconnect in a fairly uniform way. Thus, URT favors VCs over MPs because the packets traversing the interconnect can often avoid the contention penalties by exploiting the multiple independent queues that are dynamically assigned by the routers each time they process a head-flit. On the other hand, MPs are not suited for URT traffic because packets are forwarded along planes which are selected by the source NI and do not change until final delivery. MPs outperform VCs by up to the 30% for Transpose and Tornado traffic patterns, The reason behind this gain is the way these two traffic patterns stress the interconnect. Considering a 4 × 4 Mesh under Transpose, all cores of the Row 0 send data to specific cores located on Column 0. Hence, the input channel of the router $r_{0,0}$ (located in the top left corner of the mesh), contended by three flows of data. By splitting the flows across multiple planes, the headers of the flits can be processed in parallel by $r_{0,0}$, thus improving the system throughput. With 4-Hotspot, contention occurs only in the rows and columns located in the middle of the mesh where all flows of data are directed. Thus, the average number of contention is lower than in the case of URT but still higher than in the cases of Tornado and Transpose. As a consequence the TIR for this traffic pattern is higher for the VCs but lower than the one of URT.

Fig. 8(a) also shows the effect of changing queue sizes on the performance of MP and VC NoCs. In particular, by increasing the total storage S the performance of the two architectures improves in different ways. In the case of URT and 4-Hot-Spot traffic patterns, increasing S for the VC NoC has diminishing returns. That is, MPs are better handling much contention generated by these traffic patterns because they have longer queues than VCs. On the other hand, in Transpose and Tornado, the performance improvement of MPs is not strictly bounded to the queue size but to the number of planes in NoCs, thus increasing S has less impact on TIR than the other traffic patterns.

Fig. 8(b) shows the TIR under 8 × 8 2-D-Mesh. The main difference from 4×4 NoC discussed above is the performance of the MP NoC under the hot-spot traffic pattern. Here, by scaling the system from 16 to 64 nodes, more source cores send packets to the same number of destinations. This generates regular flows of data directed to the center of the mesh whose central channels become very contended. This scenario is favorable to MP NoCs, which present improved performance and TIR.

Fig. 8(c) reports the TIR for the 16-nodes Spidergon. Due to the different topology, the destination of each source in the Transpose traffic is very similar to that in Tornado traffic under the 4 × 4 mesh; thus, the TIR results of both Tornado and Transpose for the 16-nodes Spidergon are similar to the ones presented in Fig. 8(a).

In summary, the input buffer depth and the contention patterns determine the differences of maximum sustained throughput in VC and MP NoCs. For traffic patterns with some specific contention spots, such as Tornado and Transpose, MPs provide better maximum sustained throughput than VCs. Instead, VCs perform better than MPs in URT and 4-Hotspot, but this performance gap can be reduced by placing more buffers in the MP NoCs.

B. Latency

Fig. 9 reports NoC latency as function of the offered load for different amounts of total storage S (specifically, S = 1024 with Q_{WH} = 4 and S = 32768 with Q_{WH} = 32) and traffic patterns. We chose URT and Transpose for the comparison because they stress the network in opposite ways and, therefore, show different behaviors when using MPs and VCs. The differences in the maximum sustained throughput between Q_{WH} = 4 in Fig. 9(a) and Q_{WH} = 32 in Fig. 9(b) are much more significant than the corresponding throughput differences between Fig. 9(c) and (d). This confirms that a NoC with

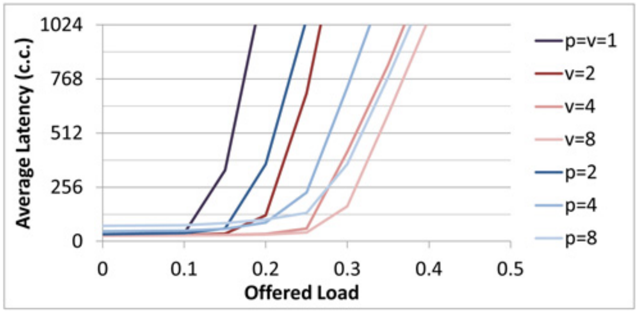


Fig. 10. Transpose latency for 4 × 4 2-D-Mesh with minimum sizing.

large amount of storage S gives a better maximum sustained throughput for URT, but has little effect for Transpose.

C. Impact of Minimum Sizing

Fig. 10 shows the latency graph with the Transpose traffic under minimum sizing with $Q = 2$: here, VC NoCs perform better than the equivalent MP NoCs in terms of both maximum sustained throughput and average latency of all offered loads. However, notice that the total amount of storage to build a VC NoC with v virtual channels under minimum sizing is exactly v times more than that of the equivalent MP NoC.

If we consider not only the performance but also the amount of total storage, building VC NoCs under minimum sizing becomes very inefficient in terms of performance improvement per unit of storage. Although VC NoCs with large Q, such as $Q = 8$ and $Q = 32$, use larger amount of storage, they provide a worse maximum sustained throughput than the equivalent MP NoCs with Tornado and Transpose.

In summary, using MPs instead of VCs gives the opportunity to save the total amount of storage of the network and to increase the maximum sustained throughput for the traffic patterns that do not present excessive random behaviors.

VII. Case Study: Shared-Memory CMPs

We completed full-system simulations for two case studies: 16-core and 64-core CMPs running with the Linux operating system. The two CMPs feature a complex distributed shared memory hierarchy with L1 and L2 private caches and four/eight on-chip memory controllers. We implemented a directory-based cache coherence protocol similar to the one used in the work of Peh *et al.* [35]. To avoid message-dependent deadlock issues (also called protocol deadlock), a typical directory-based cache coherence protocol defines a set of message classes used to manage the distributed state of the memory hierarchy. To ensure the correctness of the system, the delivery of any message belonging to a specific class must be orthogonal to the status of the network regarding the other message classes [11]. In such a scenario, VCs and MPs are two alternative techniques to provide the message-class isolation.

A. Experimental Methodology

We used Virtutech Simics [36] with the GEMS toolset [37], augmented with GARNET, a detailed cycle-accurate NoC model that provides support for modeling packet-switched NoC pipelined routers with either wormhole or virtual channel flow controls [38]. We extended GARNET to accommodate the modeling of heterogeneous multiplane NoCs with different flit sizes per plane and to support on-chip directory caches.

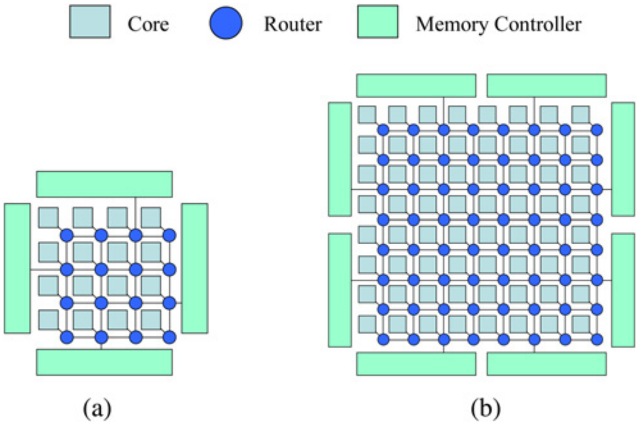


Fig. 11. Target system: the logical design of a node and the topology of the 16- and 64-core CMP systems. (a) 16-core CMP. (b) 64-core CMP.

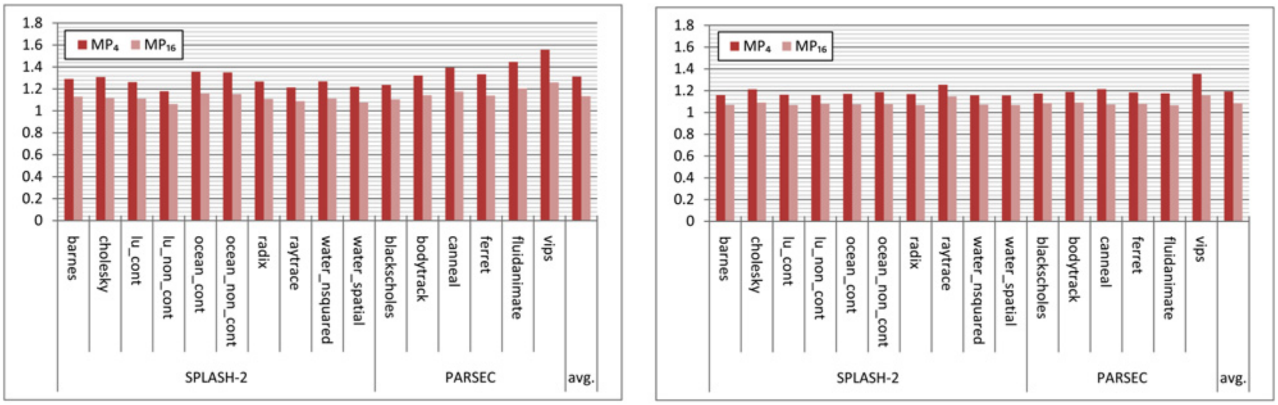
TABLE V
Main Simulation Parameters of the Target CMP Systems

Processors	16 or 64 in-order 1-way 64-bit SPARC cores, with the SPARC V9 instruction set
L1 Caches	Split IL1 and DL1 with 16KB per core, 4-way set associative, 64B line size, and 1 cycle access time.
L2 Caches	1 MB per core with 4-way set associative, 64B line size, and 3 cycle access time.
Directory Caches	256kB per memory controller with 4-way set associative, 4 cycle access time.
Memory	4 or 8 memory controllers on chip, 275-cycle DRAM access + on-chip delay

Workloads. We run simulations with eight benchmarks from the SPLASH-2 suite [13] and six benchmarks from the PARSEC suite [14]. For the benchmarks from the PARSEC suite, we use the *simmedium* input dataset and collect statistics from Region Of Interest (ROI) provided by the benchmarks. We measured the performance of the parallelized portion of each workload. To avoid cold-start effects, all caches were warmed up before running the benchmarks.

Target System. We assume that the 16- and 64-core CMPs are designed with a 45 nm technology and run at $2Ghz$. Each core is a single-issue in-order SPARC processor with 16KB of instruction and data split L1-caches and a private 1MB unified L2-cache. Cache access latency was characterized using CACTI [39]. Each core is connected to a node of a 2-D-Mesh NoC through a network interface. The NoC provides support for communication with the off-chip DRAM memory through multiple memory controllers as illustrated in Fig. 11(a) and (b). Cache coherence between the L2-caches and DRAM memory is based on the MOESI directory protocol [40], whose model is provided in GEMS. Each memory controller is equipped with a 256kB directory cache, where each block consists of a 16-bit vector matching the number of private L2-caches in the CMP. The bandwidth of DRAMs, off-chip links, and memory controllers is assumed to be ideal, i.e. high enough to support all outstanding requests. The basic simulation parameters are summarized in Table V.

Network-on-Chip Configurations. Cache coherence protocols are generally characterized by a number of functionally dependent data and control messages. In the MOESI cache-coherence protocol, there are four classes of messages exchanged among the private L2-caches and the memory controllers: data request (REQ), request forward (FWD), data



(a) (b)

Fig. 12. Normalized average latency results for 16- and 64-core CMP systems. (a) 16-core. (b) 64-core.

TABLE VI
CLASSES OF MESSAGE AND PLANE ASSIGNMENTS FOR MP_4 AND MP_{16}

Message Class	From → To	Size (bits)	MP assignment Plane ID	(flits)
REQ	Cache →Mem	64	0	8
FWD	Mem → Cache	64	1	8
DATA	Mem → Cache	576	2	18
DATA	Cache → Cache	576	2	18
WB	Cache → Mem	{64,576}	3	{8,18}

transfer (DATA), and write back (WB). Causality dependencies across messages of different classes can be expressed by message-dependency chains [11]. These dependencies may cause message-dependent deadlock. A common way to guarantee the absence of message-dependent deadlock is to introduce an ordering in the use of the network resources. In particular, causality relations among pairs of message types can be modeled as partial-order relations over the set of all possible message classes in the network. From an NoC design viewpoint this translates into assigning a separate set of channels and queues to each message type.

Since the system requires four separate virtual (or physical) networks, we cannot use the reference wormhole router as we did in the two previous analyses of Section V and VI. Thus, a total of $v = 4$ virtual channels, where each message class has a distinct virtual channel, is used as the baseline VC NoC for our comparison. The flit width, which also corresponds to the channel parallelism, is $B_{VC} = 64$ bits. For each virtual channel the router has an input queue of size $Q_{VC} = 4$ and, therefore, the total amount of storage per input port is $S_{VC} = 64 \times 4 \times 4 = 1024$ bits.

As possible MP implementations that correspond to the baseline VC NoC, we consider two MP NoC configurations with $p = 4$ planes based on competitive and minimum sizing, named MP_{16} and MP_4 for the MP NoCs with $Q = 16$ and $Q = 4$, respectively. For both multiplane configurations we partitioned the 64 bits channel parallelism of the baseline VC NoC as follows: $B_0 = B_1 = 8$ bits for *Plane 0* and *1*, $B_2 = 32$ bits for *Plane 2*, and $B_3 = 16$ bits for *Plane 3*. Since the goal is to assign a distinct plane to each of the four possible message classes to avoid message-dependent deadlock without introducing a severe serialization latency penalty, the values of this partitioning are chosen based on our knowledge of the size and the total number of injected packets per message class[9] Notice that this heterogeneous partitioning does not change the total amount of storage S; total amount of storage for both MP_4 and MP_{16} is equal to the their homogeneous counterparts.

Table VI reports the plane assignment for each message class together with the message size expressed both in bits and in the number of flits that are necessary when this message is transferred on a plane of the MP NoCs. For example, a DATA message, which consists of a cache line of 512 bits and an address of 64 bits, is transmitted as a worm of 18 flits on Plane 2, whose flit width is $B_2 = 32$. Notice that the same message incurs a much smaller serialization latency when transmitted as a sequence of 9 flits on the baseline VC NoC, whose flit width is $B_{VC} = 64$ bits[10]. Similarly, a REQ message, which consists of 64 bits, requires 8 flits to be transmitted on Plane 0 of either MP_4 or MP_{16}, but only one flit on the baseline VC NoC. Both the baseline VC and the two MP NoCs use 5-stage pipelined routers with credit-based flow control.

B. Experimental Results

The bar diagrams in Fig. 12 reports the average flit latency that was measured on the 16- and 64-core CMPs for the two MP NoC configurations. The values are normalized with respect to the corresponding values for the baseline VC NoC configuration. The latency is measured from the time the head flit departs from the source to the time the tail of the packet arrives at the destination and includes the serialization latency (the flits are queued into the network interface right after identifying the coherent status of L2-cache block).

Fig. 12 shows that both MP_4 and MP_{16} achieve worse performance in terms of average latency than the baseline VC NoC. This additional delay is mainly due to the higher serialization latency T_s, which dominates the contention latency T_c. In fact, the analysis of the traffic load shows that on average less than 5% of the channels are used on each clock cycle during the simulations of all benchmarks. This low load is due to the limited miss rate of the L2 caches that in the experiments remains always below 1%. In this scenario MPs cannot exploit their parallel packet-processing capabilities while suffering

[9] To find the best partitioning, we first run simulations with the baseline VC NoCs to retrieve the total number of bits transferred per each message class per second. Based on the simulations, the data transfer ratio of different message classes is approximately 1:1:3:2, and we use this information to partition B heterogeneously.

[10] Notice that Garnet does not model the head/tail flits of a worm

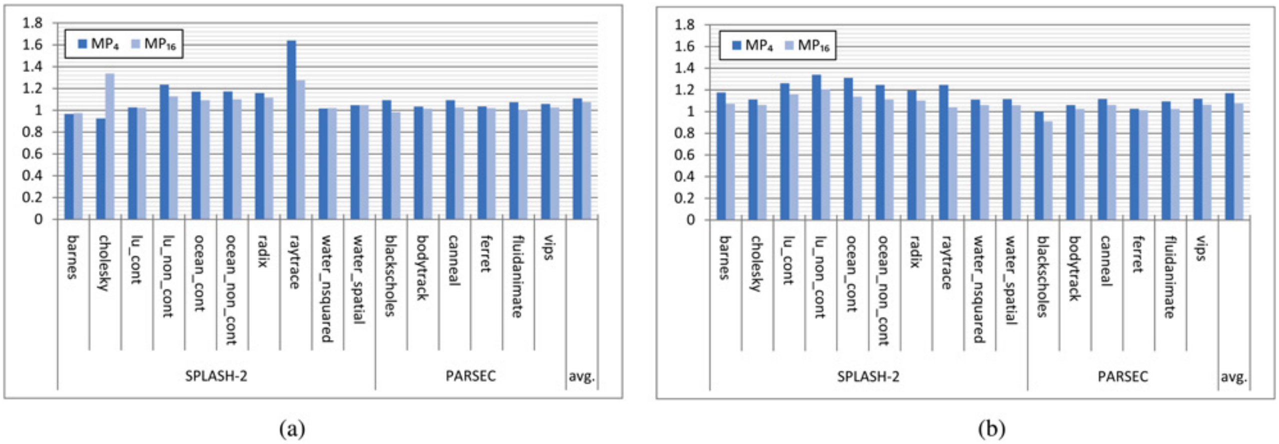


Fig. 13. Normalized execution time results for 16- and 64-core CMP systems. (a) 16-core. (b) 64-core.

from the increased serialization latency: hence, they have a worse performance than the baseline VC NoC.

On the other hand, the analysis of the average latency shows that the heterogeneous channel partitioning is a powerful technique to optimize the performance of the MP NoCs. By prioritizing more important message classes (e.g. by assigning wider planes) we measured that the average packet serialization overhead introduced by heterogeneous MPs is 15% to 30% shorter than the overhead of homogeneous MP NoCs.

As we scale from 16-core to 64-core systems, the performance differences in terms of average latency become smaller. If we compare the normalized result of MP_4 in Fig. 12(a) to that in Fig. 12(b), the performance degradation of MP_4 with respect to the baseline VC NoC is reduced from 32.17% to 19.26%. For MP_{16}, it is also reduced from 13.37% to 8.55%.

Fig. 13 shows the normalized execution time of two MP NoC configurations with respect to the baseline VC NoC on the 16- and 64-core CMPs. The results confirm that the average communication latency of a NoC does not fully characterize the real performance of a complex CMP. In fact, the system performance depends also on all the other system components such as cores, caches, and off-chip memory, and the performance degradation caused by MPs contributes only to a small portion of the entire application execution time (i.e., the NoC is not a bottleneck of the system). This fact is clearly visible in the figure where for some benchmarks the two MP NoC configurations show a smaller execution time than the reference VC NoC. This is due to the fact that the synchronization overhead among multiple cores can vary when using different networks. In particular, because conventional locks and semaphores synchronize by polling on specific memory locations, the number of instructions executed by a program can change when altering its NoC architecture.

For the configurations where MP NoCs outperform the baseline interconnect (i.e., `cholesky`), we find that the total number of instructions executed by the simulated processors is also reduced. Moreover, in the configurations where the baseline VC significantly outperforms the MP NoCs (i.e., `raytrace`), the number of instructions executed by the simulated cores has also a significant impact on the total application execution time.

On the other hand, the MP NoCs can offer some interesting design points if we combine these results with the area and power results in Section V. Under competitive sizing, MP NoCs dissipate 8% more power and present 13.37 and 8.55% of performance degradation for 16- and 64- core respectively, with 13% of area saving, compared to the baseline VC NoC. Under minimum sizing, the performance degradation grows to 31.27 and 19.26%, respectively, but MP NoCs save over 70% in both area occupation and power dissipation. Furthermore, if one considers the total execution time instead of the average communication latency as the main performance metric, the benefits of MP NoCs become even more significant: MP NoC with competitive sizing present only 7.42 and 7.37% slow-down with 16- and 64-core CMPs, respectively, while 10.90% and 17.02% performance degradations are obtained with minimum sizing.

VIII. Future Work

We sketch here some important topics of future research that go beyond the scope of this paper.

Path Adaptivity. This property refers to the ability of a router to dynamically adapt the path of a packet according to the current status of the network. When the router detects that an output port is congested, it dynamically changes the routing policy so that following packets can avoid the blocked channel [41]. Path adaptivity relies on the extensive use of virtual channels because they can be used to implement refined solutions for deadlock avoidance or recovery. Instead, a possible option to investigate for MP NoCs is the use of adaptive techniques such as the turn model, which introduce a certain degree of routing flexibility without using virtual channels [4].

Channel/Plane Allocation Adaptivity. In VC NoCs, when a packet arrives at a router, the output virtual channel of the packet is selected among multiple possible ones to maximize the utilization of the output ports. In MP NoCs, instead, a packet cannot change its plane once allocated by the network interface. Hence, to dynamically distribute the traffic across the planes of the MP NoC, it is necessary to study more sophisticated interfaces which account for the current status of the network or use scheduling algorithms like *iSlip* [4]

Traffic Adaptivity. The number of VCs can be adjusted based on contention if their buffers can be shared [18], [19]. For light traffic, having many VCs with shorter queues is more efficient than having a few VCs with deeper queues, while the opposite is true for heavy traffic.

The buffers in each input port of a router can be different in size to optimize performance [42]. Furthermore, if buffers can be shared across all input ports in a router, the buffer depth can be dynamically adjusted for both VCs and MPs [43].

Since most systems are known to exploit non-random behaviors [44], comparing VC and MP routers implemented with these techniques with bursty traffic patterns can give an interesting perspective on traffic adaptivity.

Fault Tolerance. The tolerance to temporary and permanent faults in the NoC components is an issue of growing importance for complex SoCs [45].

VC renaming is a technique to support fault tolerance across different virtual channels [46]. With VC renaming, the number of VCs recognized by a network (e.g. the routers and NIs) can be larger than the number of physical queues used to implement them. This solution is particularly effective when VCs are used to partition incompatible traffic patterns (e.g. for deadlock avoidance reasons) but it is limited to handle faults that occur in VC components.

Instead, MP NoCs offer additional fault-tolerance options: e.g. the network interfaces can avoid routing packets to a plane with faulty routers and/or links by choosing a different plane. Since all planes are completely separated from one another, MP NoCs can tolerate faults caused by links, crossbars, and switch allocators.

IX. Conclusion

We presented a comparative analysis of NoC implementations based on virtual channels (VC) versus multiple physical (MP) networks. Our analysis included an exploration of the design space considering area occupation, target clock period, and power dissipation, as well as system-level performance metrics such as average latency and maximum sustained throughput.

We found many interesting design points by comparing VC to MP routers. When the total amount of storage is limited such as $Q_{WH} \leq 8$, we showed that MP routers save more area and dissipate less power than VC routers. MP routers manage to meet a very low target clock period, and can be instantiated with a minimal amount of storage. Although VC routers dissipate less power than MP routers when the total amount of storage is large, MP routers may save more power than VC routers with small buffer sizes. Further, the benefits given by MP routers increase with technology scaling.

We showed that both VCs and MPs improve the performance of a reference wormhole router and that the benefits given by VC and MP NoCs depend on the traffic pattern generated by the system. Under a high offered load, when the traffic introduces well-distributed contention, VC NoCs yield better maximum sustained throughput and average latency than MP NoCs. Instead, when the traffic pattern generates hotspots in the NoC channels MP NoCs provide area-efficient solutions with better maximum sustained throughput.

We also compared MP and VC NoCs by simulating 16- and 64-core CMPs with various application benchmarks. Overall VC NoCs perform better than MPs in terms of both average latency and execution time. However, with the heterogeneous partitioning based on the frequency of each message class, we demonstrated that the serialization penalty introduced by MP NoC may not be as significant as expected.

In terms of power-performance efficiency, we showed that MP NoCs under competitive sizing provide reasonable trade-offs between performance and power. However, since the minimum possible amount of storage in MP NoCs is much less than that of VC NoCs, MP NoCs under minimum sizing can provide performance-per-watt efficient solutions for low-throughput applications. As the total number of nodes in a system increases, we also demonstrated that the effect of packet serialization on the average latency becomes less significant due to the large average hop count.

If implemented with shared storage, VC NoCs can dynamically adjust the number of virtual channels in the input port of a router. This dynamic adjustment may provide good opportunity for performance optimization based on traffic behavior. Instead, this feature is hard to implement in a MP NoC without introducing a complex physical adjustment of the MP routers.

Since the contention rate in MP NoCs is determined not only by the traffic patterns but also by the plane allocation-policy in the network interfaces, more work is needed to design intelligent plane allocation algorithms. Moreover, by exploiting the redundancy introduced by having multiple parallel networks, MP NoCs may provide a robust network infrastructure when controlled by an intelligent fault-tolerance policy.

Finally, the potential benefits of using MPs includes heterogeneous partitioning, where some planes can be dedicated to efficient data transfers while others can be dedicated to control tasks, such as to dynamically manage computation and storage resources in heterogeneous multicore SoCs, for example, to implement integrated fine-grain power-management policies.

References

[1] M. Yuffe, E. Knoll, M. Mehalel, J. Shor, and T. Kurts, "A fully integrated multi-CPU, GPU and memory controller 32nm processor," in *Proc. Int. Solid State Circuits Conf.*, Feb. 2011, pp. 264–266.

[2] C. H. K. van Berkel, "Multi-core for mobile phones," in *Proc. Conf. DATE*, Apr. 2009, pp. 1260–1265.

[3] P. Kollig, C. Osborne, and T. Henriksson, "Heterogeneous multi-core platform for consumer multimedia applications," in *Proc. Conf. DATE*, Apr. 2009, pp. 1254–1259.

[4] W. J. Dally and B. Towles, *Principles and Practices of Interconnection Networks*. San Mateo, CA, USA: Morgan Kaufmann, 2004.

[5] E. Carara, N. Calazans, and F. Moraes, "Router architecture for high-performance NoCs," in *Proc. Conf. SBCCI*, Jan. 2007, pp. 111–116.

[6] B. Grot, J. Hestness, S.W. Keckler, and O. Mutlu, "Express cube topologies for On-Chip interconnects," in *Proc. Int. Symp. HPCA*, Feb. 2009, pp. 163–174.

[7] P. Kumar, Y. Pan, J. Kim, G. Memik, and A. Choudhary, "Exploring concentration and channel slicing in on-chip network router," in *Proc. Int. Symp. NOCS*, May 2009, pp. 276–285.

[8] D. Wentzlaff, P. Griffin, H. Hoffman, L. Bao, B. Edwards, C. Ramey, M. Mattina, C.-C. Miao, J. F. Brown, and A. Agarwal, "On-Chip interconnection architecture of the tile processor," *IEEE Micro.*, vol. 27, no. 5, pp. 15–31, Oct. 2007.

[9] G. D. Micheli and L. Benini, *Networks on Chips: Technology and Tools (Systems on Silicon)*. San Mateo, CA, USA: Morgan Kaufmann, 2006.

[10] K. Goossens, J. Dielissen, and A. Radulescu, "Æthereal network on chip: Concepts, architectures, and implementations," *IEEE Des. Test Comput.*, vol. 22, no. 5, pp. 414–421, Sep.–Oct. 2005.

[11] Y. H. Song and T. M. Pinkston, "A progressive approach to handling message-dependent deadlock in parallel computer systems," *IEEE Trans. Parallel Distrib. Syst.*, vol. 14, no. 3, pp. 259–275, Mar. 2003.

[12] Y. Yoon, N. Concer, M. Petracca, and L. P. Carloni, "Virtual channels vs. multiple physical networks: A comparative analysis," in *Proc. DAC*, Jun. 2010, pp. 162–165.

[13] S. C. Woo, M. Ohara, E. Torrie, J. P. Singh, and A. Gupta, "The SPLASH-2 programs: Characterization and methodological considerations," in *Proc. ISCA*, Jun. 1995, pp. 24–36.

[14] C. Bienia, S. Kumar, J. P. Singh, and K. Li, "The PARSEC benchmark suite: Characterization and architectural implications," in *Proc. Int. Conf. PACT*, Oct. 2008, pp. 72–81.

[15] S. S. Mukherjee, P. Bannon, S. Lang, A. Spink, and D. Webb, "The alpha 21364 network architecture," *IEEE Micro.*, vol. 22, no. 1, pp. 26–35, Aug. 2002.

[16] M. B. Taylor, J. Kim, J. Miller, and D. Wentzlaff, "The Raw microprocessor: A computational fabric for software circuits and general purpose programs," *IEEE Micro.*, vol. 22, no. 2, pp. 25–35, Mar.–Apr. 2002.

[17] A. Jalabert, S. Murali, L. Benini, and G. De Micheli, "xpipesCompiler: A tool for instantiating application specific networks on chip," in *Proc. Conf. DATE*, Feb. 2004, pp. 884–889.

[18] C. A. Nicopoulos, D. Park, J. Kim, N. Vijaykrishnan, M. S. Yousif, and C. R. Das, "ViChaR: A dynamic virtual channel regulator for network-on-chip routers," in *Proc. IEEE/ACM Int. Symp. MICRO*, Dec. 2006, pp. 333–346.

[19] M. Lai, Z. Wang, L. Gao, H. Lu, and K. Dai, "A dynamically-allocated virtual channel architecture with congestion awareness for on-chip routers," in *Proc. DAC*, Jun. 2008, pp. 630–633.

[20] K. S. Shim, M. H. Cho, M. Kinsy, T. Wen, M. Lis, G. E. Suh, and S. Davadas, "Static virtual channel allocation in oblivious routing," in *Proc. Int. Symp. NOCS*, May 2009, pp. 38–43.

[21] J. Balfour and W. J. Dally, "Design tradeoffs for tiled CMP on-chip networks," in *Proc. Int. Conf. Supercomput.*, Nov. 2006, pp. 187–198.

[22] N. Teimouri, M. Modarressi, A. Tavakkol, and H. Sarbazi-Azad, "Energy-optimized on-chip networks using reconfigurable shortcut paths," in *Proc. Conf. Arch. Comput. Syst.*, Feb. 2011, pp. 231–242.

[23] C. Gomez, M. E. Gomez, P. Lopez, and J. Duato, "Exploiting wiring resources on interconnection network: Increasing path diversity," in *Proc. Workshop Parallel Distrib. Network-Based Process.*, Feb. 2008, pp. 20–29.

[24] S. Volos, C. Seiculescu, B. Grot, N. K. Pour, B. Falsafi, and G. De Micheli, "CCNoC: Specializing on-chip interconnects for energy efficiency in cache-coherent servers," in *Proc. Int. Symp. NOCS*, May 2012, pp. 67–74.

[25] S. Noh, V.-D. Ngo, H. Jao, and H.-W. Choi, "Multiplane virtual channel router for network-on-chip design," in *Proc. ICCE*, Oct. 2006, pp. 348–351.

[26] F. Gilabert, M. E. Gomez, S. Medaroni, and D. Bertozzi, "Improved utilization of NoC channel bandwidth by switch replication for cost-effective multi-processor Systems-on-Chip," in *Proc. Int. Symp. NOCS*, May 2010, pp. 165–172.

[27] L.-S. Peh, "Flow control and micro-architectural mechanisms for extending the performance of interconnection networks." Ph.D. dissertation, Stanford Univ., 2001.

[28] D. U. Becker and W. J. Dally, "Allocator implementations for network-on-chip routers," in *Proc. Conf. High Perf. Comput Network. Storage Anal.*, Nov. 2009, pp. 1–12.

[29] J. Kim, J. Balfour, and W. Dally, "Flattened butterfly topology for on-chip networks," in *Proc. IEEE Micro.*, Dec. 2007, pp. 172–182.

[30] "Netmaker." [Online]. Available: http://www-dyn.cl.cam.ac.uk/~rdm34/wiki/index.php

[31] Opencores. "Nocem." [Online]. Available: http://opencores.org/project,nocem

[32] Omnetpp. "OMNeT++ discrete event simulation system." [Online]. Available: http://www.omnetpp.org

[33] M. Coppola, M. D. Grammatikakis, R. Locatelli, G. Maruccia, and L. Pieralisi, *Design of Cost-Efficient Interconnect Processing Units: Spidergon STNoC*. Boca Raton, FL, USA: CRC Press, 2008.

[34] T. C. Huang, U. Y. Ogras, and R. Marculescu, "Virtual channels planning for networks-on-chip," in *Proc. ISQED*, Mar. 2007, pp. 879–884.

[35] L.-S. Peh, N. Agarwal, and N. Jha, "In-network snoop ordering (INSO): Snoopy coherence on unordered interconnects," in *Proc. Int. Sym. HPCA*, Feb. 2009, pp. 67–78.

[36] P. S. Magnusson, M. Christensson, J. Eskilson, D. Forsgren, G. Hallberg, J. Hogberg, F. Larsson, A. Moestedt, and B. Werner, "Simics: A full system simulation platform," *IEEE Comput.*, vol. 35, no. 2, pp. 50–58, Feb. 2002.

[37] M. M. K. Martin, D. J. Sorin, B. M. Beckmann, M. R. Marty, M. Xu, A. R. Alameldeen, K. E. Moore, M. D. Hill, and D. A. Wood, "Multifacet's general execution-driven multiprocessor simulator (GEMS) toolset," *ACM SIGARCH Comp. Arch. News (CAN)*, vol. 33, no. 4, pp. 92–99, Nov. 2005.

[38] L.-S. Peh, N. Agarwal, N. Jha, and T. Krishna, "GARNET: A detailed on-chip network model inside a full-system simulator," in *Proc. ISPASS*, Apr. 2009, pp. 33–42.

[39] S. Thoziyoor, N. Muralimanohar, J. H. Ahn, and N. P. Jouppi, "CACTI 5.1," Hewlett Packard, Tech. Rep., 2008.

[40] B. Jacob, S. W. Ng, and D. Wang, *Memory Systems: Cache, DRAM, Disk*. San Mateo, CA, USA: Morgan Kaufmann, 2007.

[41] J. Duato, S. Yalamanchili, and L. Ni, *Interconnection Networks: An Engineering Approach*. San Mateo, CA, USA: Morgan Kaufmann, 2003.

[42] J. Hu, U. Ogras, and R. Marculescu, "System-level buffer allocation for application-specific networks-on-chip router design," *IEEE Trans. Computer-Aided Design Integr. Circuits Syst.*, vol. 25, no. 12, pp. 2919–2933, Dec. 2006.

[43] D. Matos, C. Concatto, A. Kologeski, L. Carro, F. Kastensmidt, A. Susin, and M. Kreutz, "Adaptive router architecture based on traffic behavior observability," in *Proc. Int. Workshop NoCARC*, Dec. 2009, pp. 17–22.

[44] V. Soteriou, H. Wang, and L. S. Peh, "A statistical traffic model for on-chip interconnection networks," in *Proc. IEEE Int. Symp. MASCOTS*, Sep. 2006, pp. 104–116.

[45] M. R. Kakoee, V. Bertacco, and L. Benini, "ReliNoC: A reliable network for priority-based on-chip communication," in *Proc. Conf. DATE*, Mar. 2011, pp. 1–6.

[46] M. Evripidou, C. Nicopoulos, V. Soteriou, and J. Kim, "Virtualizing virtual channels for increased network-on-chip robustness and upgradeability," in *Proc. ISVLSI*, Aug. 2012, pp. 21–26.

Young Jin Yoon received the B.S. degree in computer science from Korea University, Seoul, Korea, in 2006, and the M.S. degree in computer science from Columbia University, New York, NY, USA, in 2008. He is currently pursuing the Ph.D. degree in computer science at Columbia University.

His current research interests include network-on-chip design and its automation, computer architecture, parallel simulation, and statistical traffic modeling.

Nicola Concer received the Laurea (*Summa Cum Laude*) and Ph.D. degrees in computer science from the University of Bologna, Bologna, Italy, in 2005 and 2009, respectively.

From 2009 until 2012, he was a Post-Doctoral Researcher with the Department of Computer Science, Columbia University, New York, NY, USA. Since 2013, he has been a Senior Researcher at NXP Semiconductors, Eindhoven, The Netherlands. His current research interest includes the design of power-efficient interconnection systems for heterogeneous multicore Systems-on-Chip.

Michele Petracca received the M.Sc. degree in computer and communication networks engineering, and the Ph.D. degree in electronic engineering in 2005 and 2009, respectively, from the Politecnico di Torino, Turin, Italy.

From 2007 to 2011, he was first a visiting Ph.D. student and then a Post-Doctorate Research Scientist in the System-Level Design (SLD) Group in the Computer Science Department, Columbia University, New York, NY, USA. Since 2011, he is with Cadence Design Systems, Berkshire, U.K., where he is involved in technology and methodology development for the top-down design of electronic systems. His current research interests include focus on the design methodologies for complex systems-on-Chip, with particular attention on the power management, the architectural aspects, the on-chip communication infrastructure, and the interaction between hardware and software enhancements of latency-insensitive systems.

Luca P. Carloni (S'95-M'04-SM'09) received the Laurea degree (*Summa Cum Laude*) in electrical engineering from the Universit di Bologna, Bologna, Italy, in 1995, and the M.S. and Ph.D. degrees in electrical engineering and computer sciences from the University of California, Berkeley, USA, in 1997 and 2004, respectively.

He is currently an Associate Professor with the Department of Computer Science, Columbia University, New York, NY, USA. He has authored over 100 publications and holds one patent. His current research interests include design methodologies and tools for system-on-chip platforms, distributed embedded systems design, and design of high-performance computer systems.

Dr. Carloni was a recipient of the 2002 Demetri Angelakos Memorial Achievement Award for recognition of altruistic attitude toward fellow graduate students, Faculty Early Career Development (CAREER) Award from the National Science Foundation in 2006, and the ONR Young Investigator Award in 2010. He was also selected as an Alfred P. Sloan Research fellow in 2008. In 2002, one of his papers was selected for the Best of ICCAD, a collection of the best IEEE International Conference on Computer-Aided Design papers of the past 20 years. He is a Senior Member of the Association for Computing Machinery.