Hybrid Electric/Photonic Networks for Scientific Applications on Tiled CMPs

Ankit Jain, Shoaib Kamil, Marghoob Mohiyuddin
CS258 Spring 2008 Final Project

***Abstract*— As multiprocessors scale to unprecedented numbers of cores in order to sustain performance growth, it is vital that the gains in speed not come with increasingly high energy consumption. Recent advances in 3D Integration (3DI) CMOS technology have made possible hybrid photonic networks-on-chip (NoC), which have the potential to result in high performance while consuming much less power than an equivalent electrical network. However, it remains to be seen whether the benefits of hybrid NoCs will carry over for real applications. Our work is the first attempt at a comparison of hybrid NoCs with electrical networks using both synthetic benchmarks as well as real scientific applications. We describe analytical models for the two networks as well as insights from simulation studies. Results show that the hybrid NoCs outperform electrical NoCs both in terms of performance and energy consumption, as long as the communications are sufficiently large to amortize the increased latency costs. Lastly, this work demonstrates the importance of finding good process-to-processor mappings in order to obtain high performance while reducing energy consumption. Overall, results illustrate the potential benefits of hybrid photonic networks for future manycore chips.**

I. Introduction

Due to the slowing ability of chip designers to scale processors to faster speeds, future trends point towards larger-scale chip multiprocessors (CMPs) in order to utilize the available transistors in a performance and power-efficient manner. As the number of processors in a CMP increases, the interconnect architecture will become more important. Major processor-manufacturing roadmaps point to simple mesh or torus networks-on-chip (NoC) as the medium-term solution, but previous work [3] has shown that such architectures may not be best-suited for balancing performance and energy usage. In this work, we explore two possible directions for future on-chip networks: one based entirely on electrical routers, and another using a hybrid approach, combining a limited electrical network with an on-chip photonic network made possible by recent advances in 3DI CMOS technology [2]. By stacking memory and interconnect resources on CMOS layers above the processors, it is possible to integrate larger memories and faster interconnects with future CMPs.

We present simple performance and energy consumption models that allow interconnect designers to quickly test a larger parameter space without resorting to slow cycle-accurate simulators. Next, we present two cycle accurate simulators—one for a purely electrical network and another for a hybrid network; we then compare best-of-breed electronic NoC performance with a hybrid photonic implementation using both models and simulators. Results show that hybrid networks can obtain higher performance at lower energy cost when compared to an electrical-only network.

This paper makes the following novel contributions:

- Unlike previous examinations of photonic network technology, we utilize traces of actual parallel applications to determine whether the potential benefits of the hybrid network are realizable for SPMD-style scientific codes.
- We show that a simple analytical model can accurately predict performance and energy consumption for the two interconnection networks studied.
- We explore and show the importance of good process-to-processor mappings to obtain optimal interconnection perfomance.

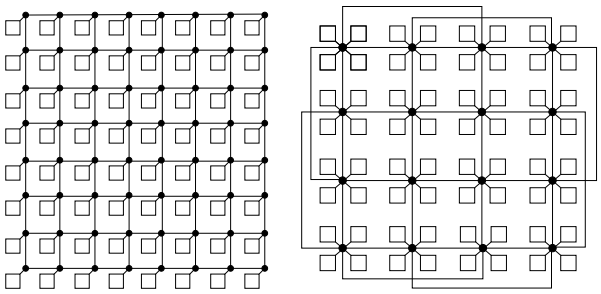
After surveying previous work in Section II, we discuss the architectures we study in Section III and benchmarks we run in Section IV. Next, we present an analytic model in Section V that provides a justification for the cycle-accurate simulators discussed in Section VI. We compare the results of running the studied applications on the electrical and hybrid interconnection network simulators in Section VII. Finally, we conclude and discuss future directions in Section VIII.

II. Previous Work

Schacham, *et. al.* [1] present the building blocks of a photonic NoC and describe an electronic control network augmented with a photonic network made up of light paths and *Photonic Switching Elements* (PSEs). Each PSE, shown in Figure 2, is composed of two silicon micro-ring resonators that deflect light when polarized, using just 1 pJ of energy to switch and 0.5 mW when active (power usage when inactive is negligible). In later work [5], these building blocks are extended to create a *non-blocking mesh network*, each using 8 PSEs to form a 4x4 switch. However, in this work we do not consider a non-blocking network; we leave that for future studies.

For electronic CMPs, Dally *et. al.* [3] compared several possible NoC topologies using detailed timing, area, and energy models for the network components. Of the explored networks, the best in terms of energy and communication time was a *Concentrated Mesh* (CMesh), a type of mesh topology that uses larger-radix routers to cluster four processors at each mesh node.

Previous work proposing a hybrid interconnection network for MPPs [4] characterized the communication requirements for full scientific applications using similar measurement tools. In that work, the hybrid networks were inter-chip; for most



(a) Mesh (b) CMesh

Fig. 1. Mesh and CMesh topologies. The CMesh requires a larger-radix switch, but reduces the average hop count.

applications, the interconnection network is overprovisioned, pointing to the feasibility of a less than fully-connected network being sufficient. Because the paper was concerned with off-chip communication topologies rather than performance, no timing models were used in the study. In addition, no timing information was considered; this work uses traces (as opposed to profiles) and thus considers the ordering of messages.

III. Studied Architecture

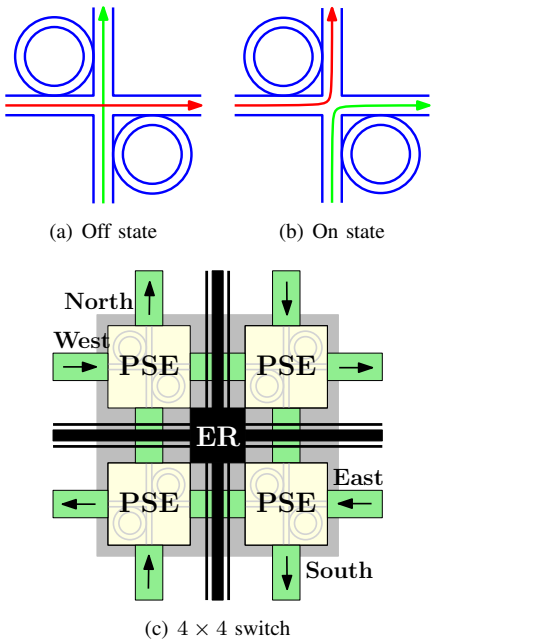
This work explores NoCs for a 64-processor CMP, with processors arranged in a 2D planar fashion. Although we do not simulate the processors, we assume they are simple in-order cores with some amount of local store memory. The tiles are nominally $1.5mm$ on each side, and are on the lowest layer of the 3DI CMOS die along with the electrical routers. Above this layer, we assume a layer strictly devoted to the local store, allowing our cores to each contain 0.5GB of memory. The latency and energy figures are based on a $22nm$ process, with a $24mm \times 24mm$ die. The overall structure of the layers is shown in Figure 3.

A. Electrical NoC Architecture

We model a concentrated mesh topology for our electrical network, as shown in Figure 1. While the mesh network has the advantage that each router is relatively simple compared to those in the CMesh due to the latter's need for a larger radix switch (which can potentially consume more energy), the average number of links traversed in the CMesh is lower, leading to significantly better performance. Each router is wormhole routed and the network supports virtual channels to eliminate deadlock and improve performance. For the implementation of the electrical NoC, there is no optical layer above the local memory layers on-chip. The work in [3] also explored multiple electrical networks. For this paper, however, we assume that a single electrical network is available.

B. Hybrid Architectures

We model a hybrid mesh network in which a simple electrical network is used to set up dedicated paths through multiple photonic networks, which are used only for highly efficient transfers of data. Having multiple photonic networks increases the amount of bookkeeping that the single electrical network must perform while potentially reducing contention. The topology of both the control and optical networks is a 2D mesh (Figure 1). However, the optical mesh is *blocking*, in that only one path may be allocated through a particular optical switch at a time. As a result, we must be careful to avoid deadlock in the optical network during path setup.



(a) Off state (b) On state

(c) 4×4 switch

Fig. 2. Photonic Switching Element. When off, the device consumes no power and allows light to travel straight (a). Switching the PSE on causes the messages to turn (b). Using 4 PSEs allows the construction of a 4×4 blocking optical switch (c).

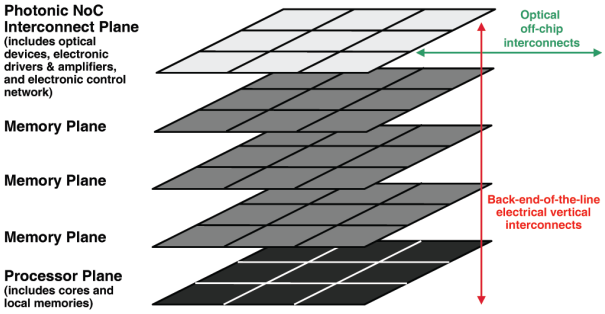
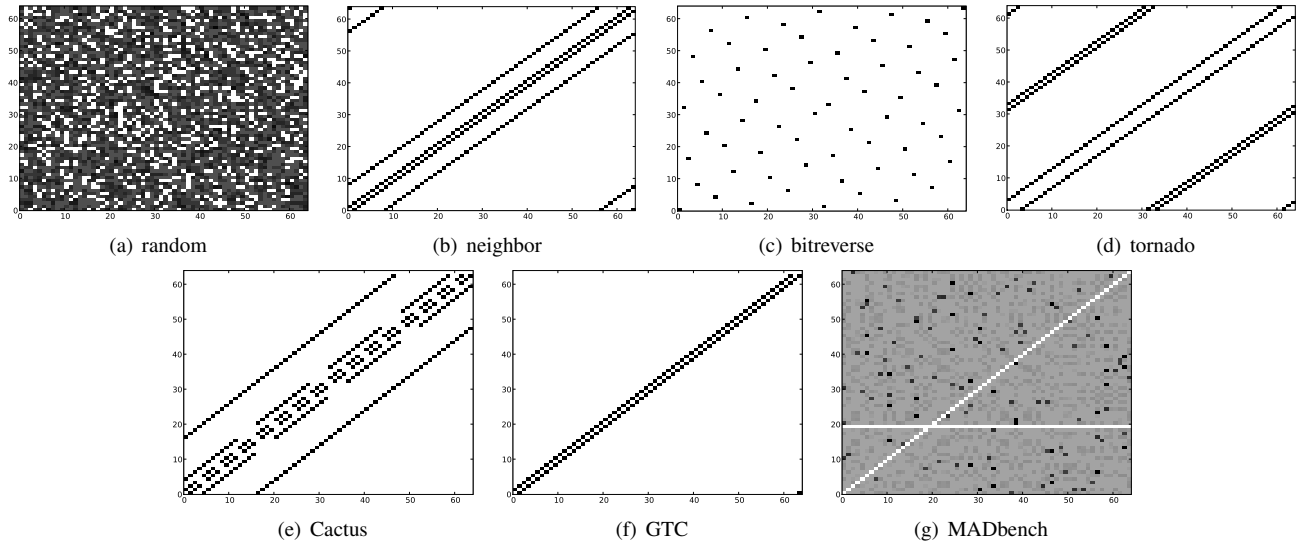


Fig. 3. Multiple layers of integration for the proposed future hybrid NoC in [5].

Each blocking optical switch (Figure 2(c)) is capable of routing a single path from any source to any destination using four *Photonic Switching Elements* (PSEs, Figure 2). Each PSE is a simple structures that, when inactive(Figure 2(a)), consumes little power and simply passes optical data through. Switching a PSE uses a tiny amount of power, and the element consumes a small active power while switched to bend the beam of light 90 degrees, causing the message to turn (Figure 2(b)).



(a) random (b) neighbor (c) bitreverse (d) tornado

(e) Cactus (f) GTC (g) MADbench

Fig. 4. Spyplots for the synthetic traces (top) and studied applications (bottom).

IV. Studied Benchmarks

Previous work has focused on artificially generated network traffic in order to demonstrate the viability of a hybrid network. We take this a step further, by two sets of benchmarks: synthetic and actual application-based. While the synthetic benchmarks help us identify the kinds of traffic that are best suited for each architecture, the application-based communication traces put real workloads (which may or may not resemble the synthetic benchmarks studied) on the networks and test different parameters. Figure 4 shows the spy plots of the seven benchmarks we use. These plots illustrate the communication volume between each set of processors: A white square at the coordinate (p_i, p_j) in the plot represents no communication while darker shades of gray represent increasing volumes of communication between the two processors.

A. Synthetic Benchmarks

We compare the two NoCs using the following standard synthetic benchmarks [13]:

- random: each processor sends messages to random destinations.
- neighbor: each processor sends a message to its neighbor in a 2D mesh.
- bitreverse: each processor sends to the partner corresponding to its bitreversed address.
- tornado: a benchmark designed to stress low-connectivity networks.

B. Application-Based Benchmarks

One of the novel contributions of this research is the use of actual application communication information for simulating network performance. We profile and study three different SPMD-style scientific applications, with traces obtained using a custom framework to measure MPI communication. SPMD-style applications are an ideal starting point for such a study because of their easily understandable synchronous communication model and because they are used widely in the scientific community.

We use the MPI profiling interface along with Linux's library preloading feature to overload the communication functions, keeping track of all function calls in an efficient, fixed-size array. When `MPI_Finalize` is called by the application, we output our trace data to a separate file for each process; the files are later combined. In order to accurately approximate communication behavior without including computation time, the trace tools order the communication into "phases" that are composed of sets of communications that must complete before further communication; essentially, we use the point-to-point synchronizations inherent in message passing to build an ordering of the communication.

The first application in this study is Cactus [6], an astrophysics computational toolkit designed to solve coupled nonlinear hyperbolic and elliptic equations that arise from Einstein's Theory of General Relativity. Consisting of thousands of terms when fully expanded, these partial differential equations (PDEs) are solved using finite differences on a block domain-decomposed regular grid distributed over the processors. The Cactus communication characteristics reflect the requirements of a broad variety of PDE solvers on non-adaptive block-structured grids.

The Gyrokinetic Toroidal Code (GTC) is a 3D particle-in-cell (PIC) application developed to study turbulent transport in magnetic confinement fusion [7]. GTC solves the non-linear gyrophase-averaged Vlasov-Poisson equations in a geometry characteristic of toroidal fusion devices. By using the particle-in- cell (PIC) method, the non-linear PDE describing particle motion becomes a simple set of ordinary differential equations (ODEs) that can be easily solved in the Lagrangian coordinates. GTC's Poisson solver is localized to individual processors, so the communication traces only reflect the needs of the PIC core.

A benchmark based on the MADspec cosmology code that calculates the maximum likelihood angular power spectrum of the cosmic microwave background (CMB), MADbench [8] inherits the characteristics of the application without requiring massive input data files. MADbench tests the overall performance of the subsystems of real massively-parallel architectures by retaining the communication and computational complexity of MADspec and integrating a dataset generator that ensures realistic input data. Much of the computational load of this application is due to its use of dense linear algebra, which is reflective of the requirements of a broader array of dense linear algebra codes in scientific workloads.

Together, these three applications represent a broad subset of scientific codes with particular communication requirements both in terms of communication topology and volume of communication. For example, the stencil in Cactus represents a communication component from a number of applications that utilize stencil-type applications. Thus, the results of this study are applicable to more than just the studied scientific codes.

V. Analytic Model

Before building a cycle-accurate full-system simulator, it is prudent to model the networks using reasonable simplifications to see whether the full simulator is worth the effort. Thus, we construct a simple model that uses approximations of the studied networks, providing an analytic upper-bound on performance and lower-bound on energy consumption.

Given a particular application trace, we study only the communication, and, furthermore, assume that all communication in each phase occurs at the same time; the models attempt to characterize the energy consumed as well as the overall time. Our models attempt to consider latency and bandwidth as well as possible contention. Additionally, the model does not consider the performance and energy impacts of deadlock avoidance techniques. The parameters used are described in this section; their specific values are shown in Section VII.

A. Electrical NoC

For each message in the communication trace, we use dimension-order routing to determine which links $L_{msg} = \{l_0, l_1, ...l_k\}$ are traversed by the message. Then, for each link l_i, the model serializes the volume of data traversing it; the overall time is at least the time to route the messages through the most congested link $l_{bottleneck}$. For computing the time to route a message, we use a bandwidth-only model— a reasonable assumption due to our use of virtual channels. So, the time to route a message msg is:

$$T_{msg} = \frac{size_{msg}}{bandwidth}$$

To determine the total time for routing all messages, we route the messages on the network and determine the most-used link, and use the time to route the total volume of communication across this link as the bottleneck point, assuming that the overall communication time will be at least this time.

For energy usage, we model the energy using the energy consumed per hop, E_{hop}, which represents the energy consumption of a router to route a message as well as the energy to travel along a link to the next router or processor. Thus, the overall energy usage is

$$E_{total} = \sum (|L_{msg}| \times E_{hop})$$

B. Hybrid NoC

The hybrid network uses a similar model as the electrical NoC, but there are two networks to account for. We determine the bottleneck link in a similar manner as above, but serialize the messages themselves instead of the bytes through each link, since each link can only be used in a single path at a time. The time for a message transmission in the hybrid network is

$$T_{msg} = |L_{msg}| \times 2 \times latency_{electrical} + \frac{size_{msg}}{bandwidth_{optical}}$$

with the latency term accounting for the time to transmit setup and teardown messages (which are small and therefore incur only a latency cost).

For the photonic network, there are three additional costs associated with the energy consumption: a cost for switching each PSE, an active cost while the PSE is on, and a cost for the Electro-Optical and Optical-Electrical conversions (see Section VII). Since we assume an XY or YX dimension-ordered routing, there is only one turn in each route, with a single PSE activated. Thus, the overall energy used is

$$\begin{aligned} E_{total} &= \sum (|L_{msg}| \times E_{electrical} + E_{PSEswitching} \\ &\quad + T_{msg} \times E_{PSEactive} + E_{EOE} \times size_{msg}) \end{aligned}$$

These simplified models enable us to estimate and understand the potential performance of the various networks without implementing full cycle-accurate simulators; the results of the models are presented in Section VII. In the next section, we describe a cycle-accurate simulation methodology for the two NoC schemes.

VI. Simulation Methodology

We implemented cycle-accurate event-driven simulators for both the electrical and the hybrid NoCs in Python. The simulators take as input the communication traces (containing the phase information, the communication topology and the volume of communication). Computation is not modeled in the simulators. An implicit barrier synchronization is assumed once all the communication for a phase has finished.

Cycle-accurate simulators are useful because they provide a realistic simulation of network performance, but such event-driven simulation is complex and time-consuming. Most of our simulation runs took on the order of tens of minutes to simulate communication that occurs over time periods of less than a second. By implementing cycle-accurate simulators, we are able to best model the contention that will occur in real systems and thus model (1) deadlock scenarios so we can avoid them, (2) energy consumption and (3) system performance.

Tables I and II present the parameters we use in our simulators. These are based on detailed models developed by Bergman, *et. al.* in [5].

A. Electrical Simulator

The simulator implements the CMesh NoC described in Section III and [3]. The key features of our simulator are summarized below:

- Processor: The destination processors take flits out of the network as soon as they arrive, under the assumption that there is no delay due to the processor being busy with other computation/communication.
- Router: The routers implement XY dimension order routing. If possible, the routers use express links when routing on the periphery [3]. Virtual channel wormhole routing was implemented to avoid any deadlock issues. The routers implement credit-based flow control, keeping track of buffer space available at the downstream routers to avoid overrunning buffers. Each router has 8 input ports (4 for attached processors and 4 for neighboring routers) and 8 output ports, which necessitates an 8×8 switch.
- Channels: Buffering is assumed at both the ends of the channels. Furthermore, we assume that the maximum wire length equals a side of the processor core. Therefore, sequencing elements need to be inserted in router-to-router links (consuming some energy). The links can accomodate one flit per cycle between neighboring processors.

B. Hybrid Simulator

Our simulator for the hybrid network accounts for path setup through an electronic control network followed by the actual message transmission over the optical network; lastly, a path takedown message is sent over the electronic network to free the links for subsequent messages. Each electrical router buffers up to 8 path setup messages from its corresponding processor (there is a 1:1 processor to electrical router mapping in the 2D Mesh topology) and attempts to route them forward on each cycle. These path setup messages are minimally-sized and therefore take just one cycle to traverse between two routers. The optical routers transmit full messages once a dedicated path has been set up on the electrical layer.

Since deadlock is a well-known problem in circuit-switched networks, much of the complexity in the network logic is in deadlock avoidance. We use the following techniques within the router of our simulator:

- Exponential Backoff: After a failed path setup packet returns to the source router, there is a exponentially increasing waiting time before attempting setup of the same path. This prevents livelock situations where processors repeatedly attempt to setup paths but are rebuffed due to circular dependencies.
- Dimension Order Routing: Before a path is set up, we randomly choose whether to use XY or YX dimension order routing. If the setup for a path fails, a new ordering decision is made for each attempt. This allows us to fully utilize the available paths through each optical switch—using only one order, half the paths would be wasted.
- Optical Network Choice: When there are multiple optical networks (on the same optical plane), we randomly choose which network for a given path before attempting to set up that path. If the setup for a path fails, a new optical network choice is made for each attempt. This allows us to coarsely load-balance the paths across the networks without too much added complexity.

In the next section, we present results from the two simulators as well as the analytical model.

VII. Results

After presenting the values of various parameters that we fix as well as parameters whose values that we vary in our two simulators, we explain the results of our experiments in four subsections. First, we explore the effect of varying various parameters in the electrical and hybrid simulators for each application. We do this by looking separately at performance, as measured by total cycles to completion, and energy consumed. The next subsection compares the best numbers for performance as well as energy for the analytical model (Section V) and simulators. Finally, we outline the effects of varying process to processor mappings for the three applications, as well as the energy and time spent in communication versus computation to place NoC performance in a full system context.

In all results, performance (in cycles) is the time to completion for all phases of communications for each application. Modeling the processor is beyond the scope of this interconnect exploration.

A. Parameters

1) Electrical: Table I lists the parameters for our experiments, which are based on values from [5]. The energy consumed contains the following components: energy consumed by links, reading/writing buffers in a router, and the switching energy in a router. Note that we have three different link lengths because we assumed Concentrated Mesh as the NoC.

For experiments on the electrical NoC simulator, we vary the number of virtual channels and the buffer size per virtual channel. A larger number of virtual channels is desirable for better network performance. Similarly, larger buffer sizes generally lead to better performance since the routers and processors stall less frequently. The total buffer space in a router is the product of the number of virtual channels and the buffer size per virtual channel. Since the area occupied by the buffers in a router is proportional to the total buffer size, it is desirable to keep this space small.

2) Hybrid: Table II summarizes the parameters fixed for the hybrid NoC experiments. We use the same parameter values as mentioned in [5].

For experiments on the hybrid simulator, we vary two parameters: Path Multiplicity (PM) and Time to Timeout (TTT). The Path Multiplicity represents the number of optical

TABLE I

ELECTRICAL SIMULATOR PARAMETERS

Model Parameter	Sim Parameter	Value
$latency_{electrical}$	Router Latency	2 cycles
	Router-Router Link Latency	2 cycles
	Virtual channels	1,2,4,6
	Buffer size in flits	1,2,3,4
	Frequency	5 GHz
$bandwidth_{electrical}$	Electrical Bandwidth	640 Gb/sec
$E_{electrical}$	Joules Per Electrical Hop	0.82e-12

TABLE II

HYBRID SIMULATOR PARAMETERS

Model Parameter	Sim Parameter	Value
$latency_{electrical}$	Router Latency	2 cycles
	Router-Router Link Latency	1 cycle
	Path Multiplicity	1,2,4
	Time To Timeout	2,10,20
	Frequency	5 GHz
$bandwidth_{optical}$	Optical Bandwidth	960 Gb/sec
$E_{PSESwitching}$	Joules Per PSE Switching	1.0e-12
$E_{PSEActive}$	Joules Per PSE on Per Second	1.0e-6
$E_{electrical}$	Joules Per Electrical Hop	0.82e-12
E_{EOE}	Joules Per Bit EOE[1]	0.4e-12

networks available and the Time to Timeout represents the number of cycles for which a router tries to propagate a message forward before sending a failure packet backwards along the path set up to the point. We expect a higher PM to increase performance while using more energy. A higher TTT should yield higher performance while using less energy. However, too high of a TTT can block a partial path for too long; therefore, a search is justified in order to find the ideal TTT.

B. Performance

1) Electrical: Figure 5 shows the results for parameter explorations on the electrical simulator. Since the y-axis shows normalized number of cycles and energy consumed, a smaller value corresponds to better performance. The performance of the electrical network generally improves with the number of virtual channels and the buffer size. For each total buffer size, we chose the number of virtual channels and buffer size as the one which yielded best performance.

Since the performance numbers are normalized with respect to the single buffer, single virtual channel case, Figure 5 shows the slowdown compared to the base case. There is a performance variation of 7x over the search space, with the applications showing similar behavior.

For choosing the optimal buffer size and the number of virtual channels, Figure 5 shows that there is marginal improvement in increasing the total buffer size beyond 8 flits, which means a relatively small area penalty. Therefore, a good

[1]EOE: Electrical to Optical conversion at source router and Optical to Electrical conversion at destination router. Approximately 0.2pJ/bit is used for the modulating of photons and another 0.2pJ/bit is used to receive the bit [5]

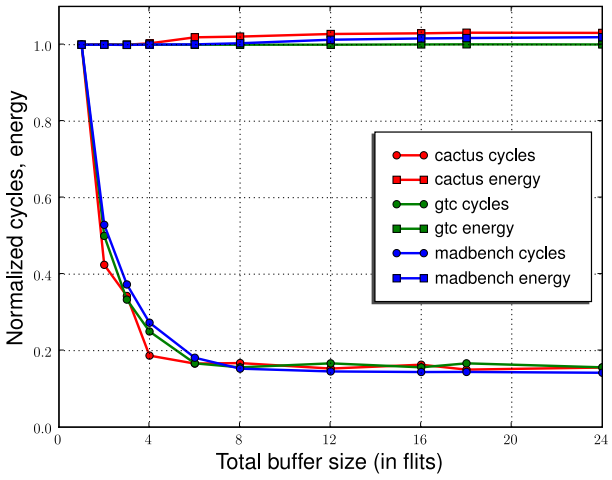


Fig. 5. Parameter exploration results for electrical NoC. Results are for communication phases only. The energy and cycle numbers are normalized w.r.t. total buffer size of 1 case (lower is better).

choice of the parameters is to use a buffer size of 2 and 2 virtual channels.

2) Hybrid: Figure 6 shows the performance for Cactus, GTC and Madbench as the path multiplicity (PM) and time to timeout (TTT) are varied. It can be seen that the network performance is poor for small TTT, as a large number of retransmissions are done because the timeouts occur early. Therefore, for small TTT, much time is spent in the electrical network trying to setup the paths. We expect to coarsely load balance the message across the multiple optical networks in an effort to decrease overall system contention and thus increase performance, i.e., multiple messages that share the same links can now progress in parallel. This is exactly what we see in Figure 6. We can see that the network performance is strongly dependent on path multiplicity— a higher value always yields better performance. With respect to the time to timeout, such monotonic behavior is not seen, especially for GTC(Figure 6(b)).

C. Energy

1) Electrical: Figure 5 shows that the energy expended increases with additional hardware on the chip (more virtual channels, more buffer size) as expected. However, since most of the energy is expended in the wires, this change is marginal, as the same volume of data must traverse the same number of links. So, the optimal values of buffer size and the number of virtual channels is chosen dependent on the number of cycles taken for the communication to finish.

2) Hybrid: We expected that adding addition hardware to our on chip network will significantly increase the amount of energy consumed. However, experiments show that this is not the case. We chose not to plot our energy numbers for our parameter-search experiment because there is little difference in energy usage as we vary the number of optical networks and the time to timeout. This can be attributed to the fact that the electrical setup network consumes at most

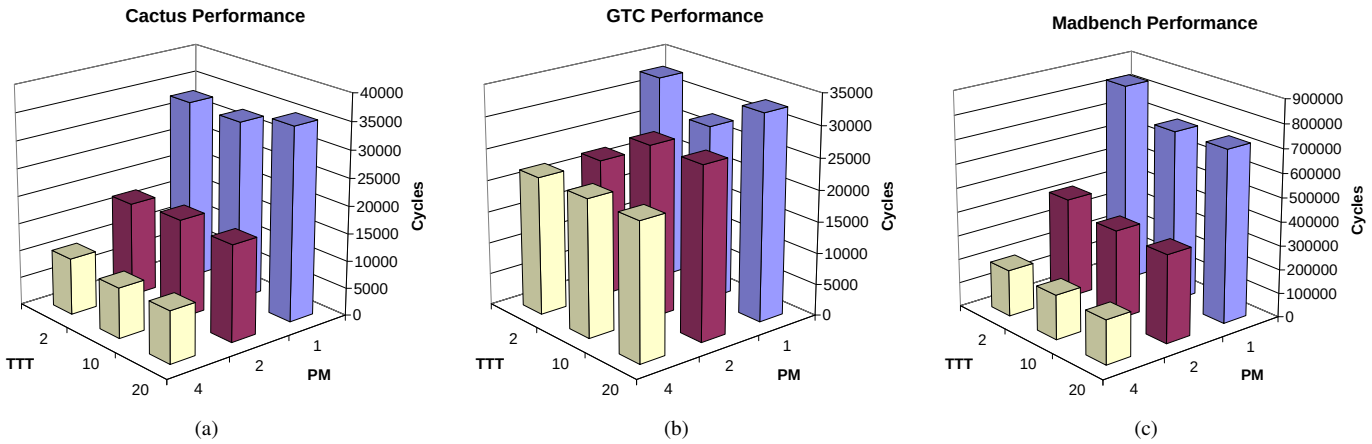


Fig. 6. Parameter exploration results for Hybrid NoC. Results are for communication phases only.

only 5.1% of the overall interconnection energy of the entire chip and the optical transfer uses at most 2% of the overall interconnection energy; the rest of the energy is spent on unavoidable EOE conversions. Given the fact that the number of bits that are converted from electrical to optical and then back to electrical is constant across our parameter search, and the energy used in the EOE conversions is purely a function of bits converted, it is not a surprise that energy use does not vary much at all. Although adding additional optical networks significantly reduces contention on the electrical layer, because this accounts for at most 7.1% of the interconnection energy, the savings do not show in the overall energy use.

D. Electrical vs. Hybrid NoC

For the synthetic benchmarks, we used two different runs, one with small messages, and another with larger messages. Figure 7 shows these results. As we expect, the hybrid interconnect performs slightly worse with smaller, latency-bound messages, but far outperforms the electrical network with larger messages such as those used in most of the applications in this study. Subfigures (c) and (d) show this clearly with the blue bar for small messages being larger than the red bar. However, this is never true in the large message communication case. In addition, in both cases for all four benchmarks, the hybrid interconnection network uses much less power than the purely-electrical network.

Figure 8 compares the performance of the two networks for the 3 applications, using both simulation and modeling. For the simulation of the electrical network, we used an optimal mapping whenever possible. For GTC and Cactus, an optimal process-to-processor mapping was easy to find because of the simple communication toplogy (3D mesh for Cactus and a ring for GTC). For MADbench, we used the default process to processor mapping since its communication topology lacked a regular structure.

For the electrical model, we see inaccuracies of as much as 60% w.r.t. simulation results— this is moreso true for MADbench because of the large number of small messages involved. There are two reasons for this: the model is bandwidth-only and ignores the effect of backpressure due to queues getting full. However, the energy model for the electrical network is closer to the simulation results; the only inaccuracies arise due to ignoring the energy consumed in reading/writing to router buffers. The models for the hybrid network, however, are quite accurate as most of the delays are deterministic because of the lack of queues in this network.

Comparing the hybrid and electrical networks, we see that for two of the applications, the hybrid network outperforms the electrical network in terms of time. For all applications, the energy savings by using the hybrid network are substantial, up to two orders of magnitude. In order to reap the bandwidth benefits of a hybrid network, the message sizes must be large enough to amortize the extra latency costs of the setup and takedown messages over the electrical network, as well as the potential latency caused by cases where multiple paths must be serialized due to link contention. The only application where this criteria does not occur is GTC, resulting in similar performance between the two NoC implementations.

E. Processor Mapping

Although we do not here attempt to find optimal mappings for our codes, we randomly generate 100 mappings and present the minimum, average and maximum performance and energy consumed. The results for this experiment are in Figure 9.

The difference in performance and energy consumed between the minimum and maximum can be as much as two orders of magnitude, with the average performance and energy consumed being close to the maximum. Thus, more important than finding the optimal mapping is making sure not to use a 'bad' mapping. The results are more pronounced on the hybrid NoC because bad mappings can cause longer average path lengths, resulting in more paths sharing links, and therefore, more transmissions on the electrical network due to failed path setups.

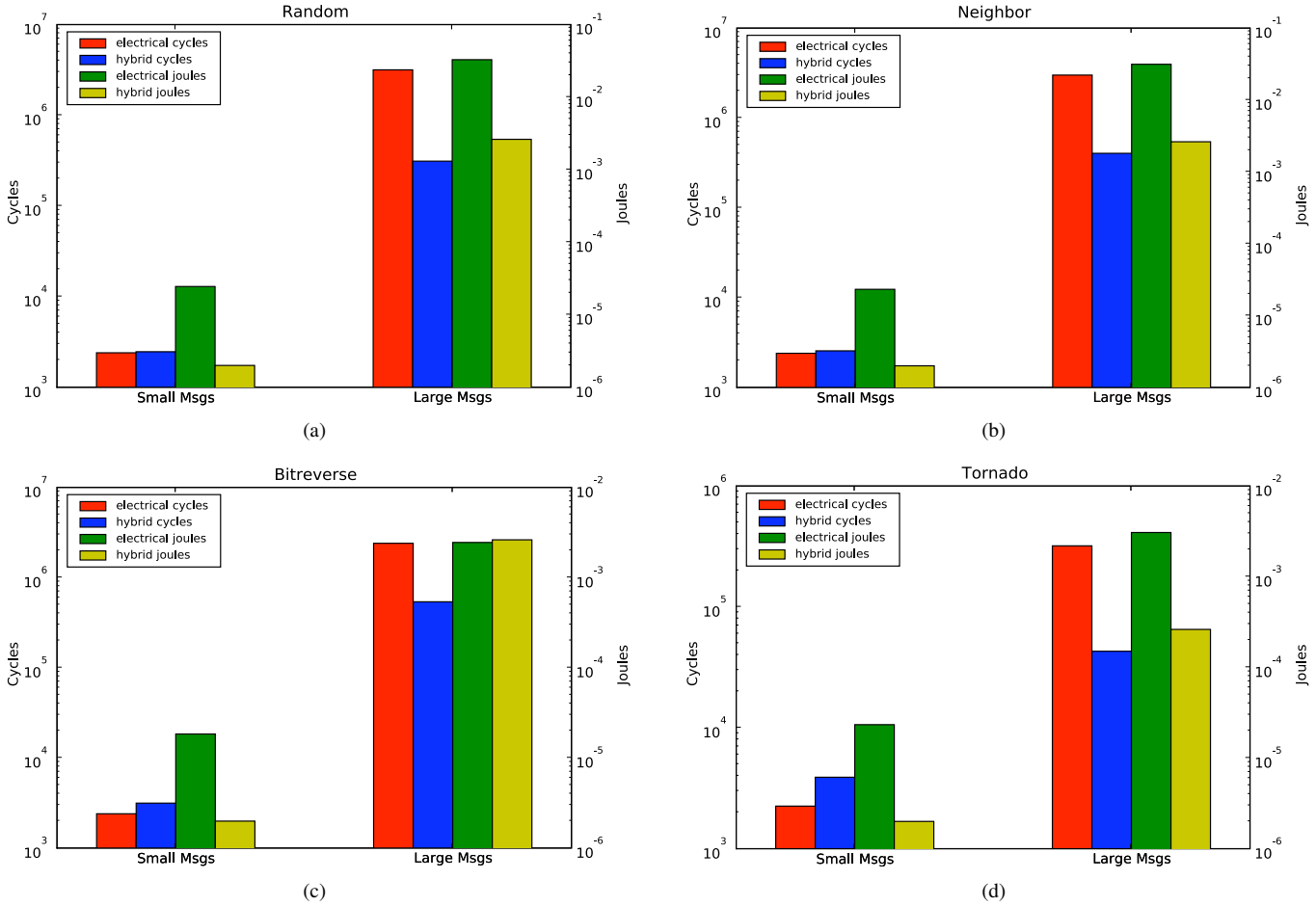


Fig. 7. Performance and energy consumption for synthetic benchmarks using models of the two networks. Each plot has two sets of bars corresponding to small message communication and large message communication.

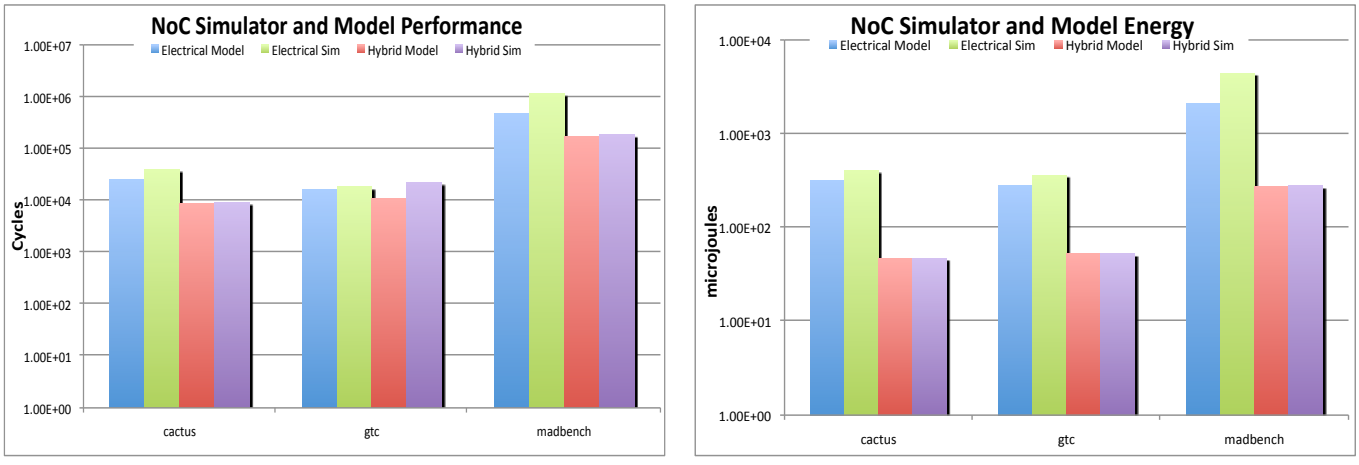
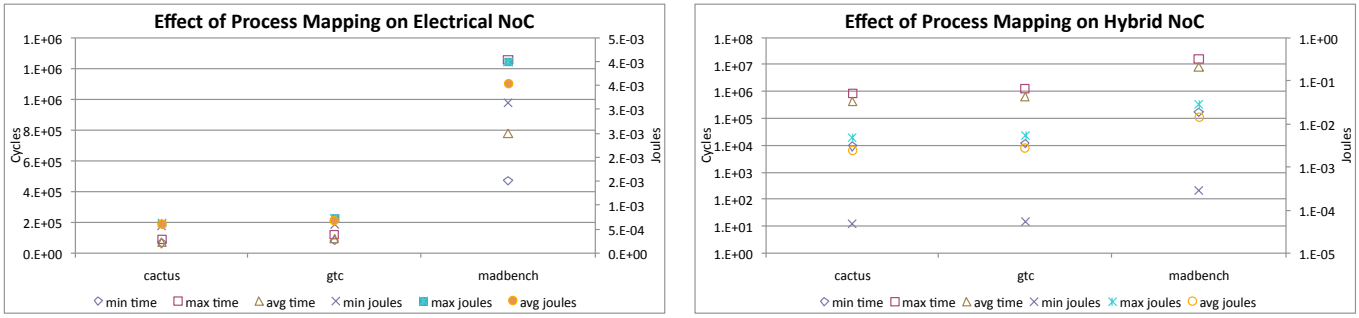


Fig. 8. Performance and energy consumption for modelling and simulation of the studied applications. Results are for communication phases only. For the Cactus and GTC applications, we used optimal mappings instead of random ones, because of the regular structure in the communication topology.

F. NoC Subsystem Energy

Primarily, the concern here has been the NoC energy and performance independent of the rest of chip. In this section, we examine the NoC as a subsystem in a full CMP by comparing the energy and time of the NoC to the energy and time of just performing the floating-point operations in the CMP.

The time for the floating point operations is derived by assuming that the cores will be able to retire a single double-



(a) (b)

Fig. 9. Effects on process-to-processor mapping for runtime and energy consumption for our four applications on Electrical and Hybrid NoCs. Results are from 100 runs with randomized mapping and for communication phases only. Note: The electrical results are on a linear y-scale while the hybrid results are on a logarithmic y-scale in order to better show the results.

TABLE III
NOC AS A SUBSYSTEM OF A CMP.

App	% Time due to Electrical NoC	% Time due to Optical NoC	% Energy due to Electrical NoC	% Energy due to Optical NoC
cactus	0.079	0.019	12.5	1.91
gtc	7.89E-5	9.46E-5	0.03	4.6E-3
madbench	0.013	1.9eE-3	6.12E-4	7.86E-5

precision multiply-add each cycle; this is reasonable, given that they will be relatively simple cores but that floating point will still be a priority. For energy, using models from the Merrimac project [14], a processor that featured an energy-efficient floating point unit, we scale using the ITRS roadmap to the target feature size of 22nm to determine the joules per flop. These two numbers are then used, along with FLOP information from the trace infrastructure, to determine the energy and time of the FLOPs.

The results of this calculation are shown in Table III. It is apparent from these numbers that for our test applications, which are SPMD bulk-synchronous style, the interconnect is not a large portion of the overall energy consumption or time, using at most 12.5% of the energy and at most 0.08% of the time. However, one must note that the time metric is misleading. This calculation does not take into account how the computation time may depend on communication. In particular, after a phase of computation, a further phase may not start until the necessary communication completes. Thus, although the NoC numbers reflect that most of the time is spent on computation, they may underemphasize the importance of the interconnect.

Furthermore, the dependence is more important for tightly-synchronized applications, since there are many more points where computation can potentially be held up by the interconnect. Lastly, it is important to consider that, given the power trends of CMOS technology, any potential savings in energy will be important. Reducing the portion of energy consumed by the network by an order of magnitude (and in some cases, two orders) is important even at the levels shown in this section.

VIII. CONCLUSIONS & FUTURE WORK

This work compared the performance and energy used for a suite of synthetic communication benchmarks as well as traces from SPMD-style scientific applications on simulators and simple models for an electrical NoC as well as a hybrid NoC. The models accurately predict both performance and energy consumption for the three applications on the two interconnection networks in this study.

We showed that a hybrid NoC has the potential to outperform electrical NoCs in terms of performance as well as mitigating the power/energy issues that plague electrical NoCs when the communications are sufficiently large to amortize the increased latency costs. From our results, the following points can be made for the energy demands of these networks:

- The majority of hybrid network energy is due to Optical-to-Electrical and Electrical-to-Optical conversions (>94%).
- Adding additional hardware to the hybrid network decreases energy use while the same is not true of the electrical network.

These observations will be essential to guide future CMP designers in choosing an interconnect that does not become the bottleneck for performance or energy. As future architectures scale to even higher concurrencies, the power requirements and performance benefits of photonic interconnects will become more and more attractive.

We also showed that considering process-to-processor mappings can significantly impact performance as well as energy consumption. However, finding the optimal mapping is not always of utmost importance— making sure not to use a 'bad' mapping is.

Bergman, *et. al.* [5] explored using a non-blocking mesh hybrid interconnection network and have found that such a network provides higher throughput than the blocking counterparts. This is primarily due to the decrease in contention during path setup: the only contention remains at the destination router which can accept only one connection per optical network at a given time. The disadvantage of such a non-blocking network is that it uses significantly more switches and scales (in terms of the number of switches) like a crossbar. We intend to explore such topologies in the future, comparing them to those studied in this paper.

Although this work has addressed a few questions about how different applications would behave on different networks on chip, it also raises a number of questions that will lead to interesting future studies. This work focuses completely on the interconnection network and does not account for data transfer onto the chip. Furthermore, it not clear how the performance and energy consumption of the networks fits into overall system performance and energy. To fully understand the impact of an interconnection network, it is necessary to model the processors— this takes care of the computation part of the application, as well as the memory. Our work assumed no overlap of communication and computation. This means that performance improvements in the network always translate to improvements to the overall system (even though large performance improvements in the network might make a small impact in the overall system performance). Modeling the processors will enable us to explore algorithms which overlap communication with computation. The presence of external memory (DRAM) means that the interconnection network will also encounter processor-to-memory traffic in addition to interprocessor communication. While this paper focused on SPMD style applications found in the scientific community, future studies could also explore applications with less synchronous communication models. All of these refinements are potential subjects for future work, using the foundation presented in this paper.

Acknowledgements: We would like to thank Dr. Keren Bergman (Columbia University) for her guidance regarding photonic networks. Dr. John Shalf at LBNL provided excellent feedback and help throughout the duration of this research. We would also like to thank the BeBOP research group at UC Berkeley for their insightful comments.

REFERENCES

[1] Assaf Shacham, Keren Bergman, and Luca Carloni. On the Design of a Photonic Network-on-Chip. In *Proceedings of the First International Symposium on Networks-on-Chip*, 2007.

[2] K. Bernstein, P. Andry, J. Cann, P. G. Emma, D. Greenberg, W. Haensch, M. Ignatowski, S. Koester, J. Magerlein, R. Puri, and A. Young. Interconnects in the Third Dimension: Design Challenges for 3D ICs. In *Proceedings of the Design Automation Conference*, 2007.

[3] James Balfour, and William Dally. Design Tradeoffs for Tiled CMP On-Chip Networks. In *Proceedings of the International Conference on Supercomputing*, 2006.

[4] Shoaib Kamil, Ali Pinar, Daniel Gunter, Michael Lijewski, Leonid Oliker, and John Shalf. Reconfigurable Hybrid Interconnection for Static and Dynamic Applications. In *Proceedings of the ACM International Conference on Computing Frontiers*, 2007.

[5] Bergman *et. al.*. Topology Exploration for Photonic NoCs for Chip Multiprocessors. *Unpublished to date*.

[6] Cactus Homepage. `http://www.cactuscode.org`, 2004.

[7] Z. Lin, S. Ethier, T.S. Hahm, and W.M. Tang. Size Scaling of Turbulent Transport in Magnetically Confined Plasmas. Phys. Rev. Lett., 88, 2002.

[8] Julian Borrill, Jonathan Carter, Leonid Oliker, David Skinner, and R. Biswas. Integrated performance monitoring of a cosmology appli- cation on leading hec platforms. In Proceedings of the International Conference on Parallel Processing (ICPP), 2005.

[9] A. Canning, L.W. Wang, A. Williamson, and A. Zunger. Parallel Empirical Pseudopotential Electronic Structure Calculations for Million Atom Systems. *J. Comput. Phys.*, 160:29, 2000.

[10] Xiaoye S. Li and James W. Demmel. SuperLU-dist: A Scalable Distributed-Memory Sparse Direct Solver for Unsymmetric Linear Systems. *ACM Trans. Mathematical Software*, 29(2):110140, June 2003.

[11] J. Qiang, M. Furman, and R. Ryne. A Parallel Particle-in-Cell Model for Beam-Beam Interactions in High Energy Ring Colliders. *J. Comp. Phys.*, 198, 2004.

[12] IPM Homepage. `http://www.nersc.gov/projects/ipm`, 2005.

[13] W. Dally and B. Towles. Principles and Practices of Interconnection Networks. Morgan Kaufmann Publishers, San Francisco.

[14] Mattan Erez. "Merrimac – High-Performance, Highly-Efficient Scientific Computing with Streams". Ph.D. dissertation, November 2006, Stanford University, Stanford, California.