Analytical Evaluation and Automated Design of Networks-on-Chip
EE382C Research Paper

Zain Asgar, Vishal Parikh, JongSoo Park
Stanford University
{zasgar, vparikh1, jongsoo}@stanford.edu

1 Introduction

As *Systems-on-Chip (SoC)* become more prevalent, and as computer architecture as a whole makes a shift to more modular and parallel designs, on-chip interconnection networks are seen more and more frequently [2], [1]. As with all aspects of microprocessor design, as the problem size and design space grows, arriving at an efficient solution by hand quickly becomes intractable. It becomes necessary to automate the design process. This principle applies to interconnection networks as well. The interconnection network field is fairly developed; however, only in recent years has semiconductor technology provided enough transistors on a single chip to make on-chip networks a possibility.

Network-on-Chips (NoCs) face many different design constraints than traditional off-chip networks. Chip design is becoming more power constrained and on-chip networks can consume large portion of power. For example, MIT Raw on-chip network consumes 36% of total power [22]. There are also strict area and routing constraints. Additionally the layout of the interconnection network has a strong influence on both its latency and power [10].

NoCs typically have slightly different constraints than off-chip networks. Since the granularity of communication is much lower, it is necessary that network overhead be lower as well. Bandwidth requirements are typically higher, and there are often stricter latency requirements. NoCs may also require guaranteed service since much of the communication is critical [3].

A well designed NoC must fulfill the throughput, latency, and *Quality of Service (QoS)* requirements of the system, but it must also have a minimal impact on the rest of the system. The area and energy of the system must be minimized, and the wire routing should not impact the functional units of the chip. As previously mentioned, in order to design a reasonably complicated chip in a manageable time frame, the design process needs a certain degree of automation. This problem is difficult due to the large design space. A proper automated NoC design system must analyze the requirements of the system, and determine what the best topology is that satisfies not only the latency and throughput, but also the area and power requirements of the system. In addition to the topology, it should produce an appropriate routing algorithm, flow control scheme, and router architecture to support the topology and fulfill the aforementioned requirements. Finally, the tools must consider how the layout of the chip will affect its performance. Latency and power are strongly related to the link length and layout. When designing an interconnection network, it necessary to think about not only the logical connections between nodes, but also their physical locality and how that affects the communication cost. This simultaneous consideration of both high level and low level design also makes the problem difficult.

The papers studied in this review propose analytical and automated methods of analyzing and generating network topologies. These methods could be included in a tool chain for automatic design and synthesis of NoCs, or they could be used in performance analysis tools and techniques. There are many important issues that must be considered when designing such systems. Since the energy and efficiency of modern processors is critical, the energy requirements of the NoCs must be considered early in the design phase. Good energy models for the system are needed that accurately describe both link and routing energy. These depend heavily on the layout and the router architecture. The energy models should also take into account network congestion, routing inefficiencies, and flow control, as these things affect the latency and the energy efficiency of the network. A good analysis will also take into account the heterogeneous nature of most SoCs when producing an interconnection network. A typical SoC consists of processor cores, memories, IO, and custom functional units among other things. Each of these has different communication requirements. This should be taken into account in the network design. Lastly, a proper analysis

	100nm	70nm	50nm	35nm
Transistor Capacitance	1	0.78	0.65	0.66
Wire Capacitance	1	0.94	0.89	0.85
Static Power	1	2.01	3.35	4.30

TABLE 1
Scaling Factors Used in [23]

will compare an NoC solution to conventional point-to-point communication. It is not clear that an NoC is always necessary, especially when the number of elements is small. It is important that the NoC produced provides better or at least comparable efficiency to the alternative. Lastly, it is important than any design methodology scales well with technology. As technology scales, the relative cost of wires to gates grows, and the distance a global signal can travel in a cycle shrinks [4]. The synthesis tools need to account for such variations.

The remainder of this paper is organized as follows. Section 2 reviews Wang et al. [23]. This paper talks about technology scaling dependent energy consumption model and its impact on topology selection. Section 3 reviews Srinivasan et al. [19], which presents a method to generate an application specific NoC topology. Section 4 reviews Murali and De Micheli [10], which describes the automatic topology mapping algorithm used in their `NetChip` tool chain. Section 5 concludes the paper.

2 A Technology-aware and Energy-oriented Topology Exploration for On-chip Networks

In [23], Wang et al. present topology exploration based on underlying device technology for NoC. Their evaluation is conducted for technologies between $0.1\mu m$ and 35nm.

2.1 Motivations for Technology Aware Exploration

The authors describe that the underlying semiconductor technology has a significant impact on the final topology. They characterize their energy using the following equation:

$$E_{flit} = H_{avg} \cdot E_R + D_{avg} \cdot E_{L0} \quad (1)$$

Where E_{L0} is the energy for unit length, E_R is the router energy. H_{avg} and D_{avg} are the average hops and link distances, respectively, which are influenced by the topology.

The authors show the scaling factors for transistor and wire capacitance, and static power, as shown in Table 1. They explain that these subtle changes have a big impact on router architecture. According to the authors, shift registers built out of flops will consume less power than those built out of SRAM when the buffer utilization is low.

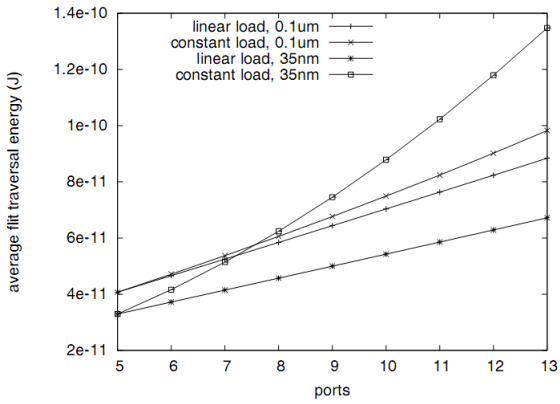

Fig. 1. Flit Traversal Energy vs. Ports

2.2 Methodology

The authors make several reasonable assumptions for their evaluation. They use Orions power model [24] to obtain the low level energy estimates and use the flit energy as the evaluation metric for energy efficiency. They assume that the network is a $N \times N$ matrix and use a port based workload model. Two models are provided one with constant load independent of the number of router ports and another model where the flit traffic grows with the number of ports.

To provide an example of this model, they use a 2GHz, 4-flit size buffer, 128-bit flit and input queueing router. The energy obtained for the two workloads, for this example system, is shown in Figure 1.

According to the authors the flit traversal energy for a large number of ports is actually higher at 35nm than 100nm. This is due to the static leakage current.

2.3 Topology Exploration

The authors consider high dimensional meshes/tori, hierarchical meshes/tori and express cube topologies in their evaluation. They discuss the tradeoff of adding express channels. Adding express channels decreases the average hop count at the expense of higher router energy. They also state that it increases the average link distances. For these network topologies, they derive the average flit traversal energy and the network edge size N_{min} at which it become more efficient than a 2-D torus network. They also derive the most energy efficient configuration for the topology.

2.3.1 2D Torus and Mesh Networks

The energy of a torus and mesh network can be easily derived as shown by the authors. For a $N \times N$ mesh we have:

$$E_{flit} = \frac{2N}{3}(E_R + E_L) \quad (2)$$

Similarly for a $N \times N$ torus we have:

$$E_{flit} = \frac{N}{2}(E_R + 2 \cdot E_L) \quad (3)$$

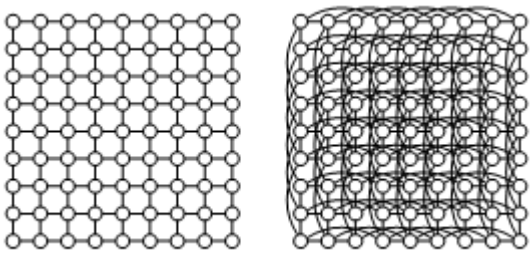

Fig. 2. A 9-ary 2-mesh (left) and high order 3-ary 4-mesh (right)

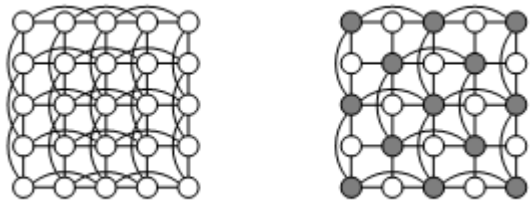

Fig. 3. 5x5 Hierarchical Tori (left) and 5x5 Express Cube (Right)

This torus network consumes less router router energy but more link energy than a mesh network. For the remainder of the paper the authors only discuss the torus topology because they are looking for relative trends.

2.3.2 Higher order networks

The authors present and derive the minimal network size of a higher order network, example in 2, by setting the derivative of the energy to zero. These minimal network size is shown in Table 2.

	Linear Load		Constant Load	
Buffer Size	4-flit	16-flit	4-flit	16-flit
100nm	53,8	44,7	61,8	51,8
70nm	49,7	42,7	64,8	55,8
50nm	46,7	39,7	82,10	73,9
35nm	46,7	36,6	165,13	140,12

TABLE 2
Minimal network size of $\sqrt[n]{M}$ -ary n-cube network, in the form of (M, N_{min})

They explain that, for a linear load, 35nm technology requires a smaller network size to benefit from higher order tori. At constant load, the opposite is true. They explain that this is because of the lower slope of E_R and static power respectively, as shown in Table 1. They argue that in future scaled technologies, more accurate workload prediction is required for making correct topology predictions.

2.3.3 Hierarchical Tori and Express Cubes

Hierarchical tori and express cubes provide express channels which give another variable that can be optimized, as shown in Figure 3. The authors derive an optimal express interval, v, which is the node separation for the express channel. They provide results for hierarchical tori and express cubes as shown in Tables 3 and 4, respectively.

	Linear Load		Constant Load	
Buffer Size	4-flit	16-flit	4-flit	16-flit
100nm	3,11	3,10	3,12	3,10
70nm	3,10	3,9	3,12	3,11
50nm	3,10	3,9	4,14	4,13
35nm	3,10	3,9	4,21	4,19

TABLE 3
Minimal Network Size and Corresponding Express Interval for Hierarchical Tori, in the form of (v, N_{min})

	Linear Load		Constant Load	
Buffer Size	4-flit	16-flit	4-flit	16-flit
100nm	3,13	3,13	3,14	3,14
70nm	3,13	3,12	3,14	3,14
50nm	3,13	3,12	4,15	4,15
35nm	3,13	3,12	4,19	4,18

TABLE 4
Minimal Network Size and Corresponding Express Interval for Express Cube, in the form of (v, N_{min})

According to the authors, a higher dimensional tori should never be chosen over a hierarchical tori or express cube network. This can easily be seen by comparing the flit traversal energy.

2.4 Case Study

They apply their analysis to predict the optimal network configuration for future technologies using TRIPS [17] as an example. TRIPS uses an interconnection network to connect ALUs to memory. They also assume that the chip size is constant while the transistor count doubles each node, implying that they have a 7×7, 10×10, 14×14 network in 70nm, 50nm and 35nm respectively. From there data earlier, they make the prediction shown in Table 5. They then use Orion to estimate the energy and obtain the results shown in Tables 6 and 7. According to these results, they correctly predicted the optimal topology.

They perform another study where they use SPLASH [18] with the 50nm network discussed above to show the flit traversal energy. These are shown in Table 8. Although H-3 was more optimal under random traffic H-2 is more optimal for SPLASH.

Technology	70nm	50nm	35nm
Optimal Topology	2-D Torus	H-3 Torus	H-4 Torus

TABLE 5
Predicted topologies from [23]

Torus Dimension	2	3	4	5	6
70nm	**159**	179	198	244	NA
50nm	198	214	221	267	283
35nm	353	388	340	403	427

TABLE 6
Simulated Flit Traversal Energy (pJ) of High-dimensional Tori

Topology	H-2	H-3	H-4	E-2	E-3	E-4
70nm	182	178	188	196	198	215
50nm	207	**194**	197	219	212	222
35nm	347	314	**301**	363	338	342

TABLE 7
Simulated Flit Traversal Energy (pJ) of Hierarchical Tori and Express Cubes

2.5 Critique

The main contribution made by this paper is demonstrating how technology scaling affects the topology. The authors also give a port based assumption which shows good relative power. However, we suspect that there are several things missing in this analysis. For example, they explain that shift registers are better implemented using flops not SRAM if utilization is low because it saves on global word and bit lines. However, there are several techniques such as hierarchical memories and low swing circuits which can be used to alleviate these problems [9].

They also show with their port based assumption that the energy for 35nm is actually higher than the energy for 100nm. A system where leakage current is higher than dynamic current, is not desireable and not likely to be adopted. It is not likely that processes dominated by leakage will be adopted. It seems that the implementation they choose is suboptimal for scaled technologies.

Their analysis of the various network topologies is very clear and easy to understand and the results presented are reasonable. They then use this information to predict the optimal structure for the TRIPS system. They show that hierarchical torus networks become better as technology scales and simple two dimensional networks are less attractive. Although this seems reasonable, their conclusions are based on a less than 10% difference in the energy values. It seems that this might be less than the margin of error of their simulation, but they do not discuss that.

Another strong point of the paper is that they use SPLASH benchmarks in their evaluation, even though they only show results for three benchmarks. However, these results seems to contradict their initial prediction. They did all their work with random traffic but SPLASH give a completely different optimal topology. It also seems that less scalable applications perform poorly, which is expected. However, it seems that again their conclusions are based on small differences in numbers which might be in their margin of error. It seems that, in most cases, even using a less optimal network is fine depending on the actual workload.

Benchmarks	LU	Water	MP3
2-D Torus	5.07, 201	4.93, 196	1.55, 61
H-2	3.05, 190	2.97, 185	1.55, 82
H-3	3.07, 191	3.00, 186	1.55, 82
H-4	3.26, 198	3.20, 194	1.55, 82
E-2	4.26, 212	4.09, 207	1.55, 65
E-3	4.81, 213	4.58, 207	1.55, 65
E-4	4.87, 211	4.67, 205	1.55, 65

TABLE 8
Average Hop Count and Flit Energy(pJ)

The authors also fail to derive the area overhead. This is usually an important consideration for chip design because it directly impacts yield and costs [25]. If the area overhead is also considered, it might significantly impact their results and hence conclusions.

3 An Automated Technique for Topology and Route Generation of Application Specific On-Chip Interconnection Networks

In [19], Srinivasan et al. present a method to generate an NoC optimized for a target application. The authors provide a detailed algorithm for layout, core-to-router mapping, and route generation in addition to scalable power estimates. The authors also provide experimental results for their algorithm on real applications.

Srinivasan's topology generator relies on the directed *communication trace graph (CTG)* of a target application. This graph is fed into a three-phase system that designs the interconnection network. The CTG is first partitioned and a preliminary layout is done to estimate communication cost. Then each node in the graph is mapped to a router. Lastly, routes are generated such that every communication trace in the CTG has a unique route in the network and every route matches the bandwidth and latency requirements of its corresponding trace in the CTG. The output of the system is a layout-aware topology with power and area models that meets the specified performance requirements. This topology is optimized primarily for power and then for area. The network is built out of pre-characterized building blocks. These include unit routers and the interconnection links among other things. The authors prevent network congestion by static routing of the communication traces based on router bandwidth. The authors claim that this allows them to model the network as uncongested, and model latency as simply the number of network hops.

3.1 Three phase algorithm

The authors define their CTG as follows. Each vertex is either a processing element or a memory (referred to as a node from now on). Each edge is a communication trace.

Each trace is weighed by its bandwidth requirement and its latency constraint. The router architecture is fixed, and it is assumed that the router can support equal bandwidth in the input and output ports. Then, power for the router is measured as power from inbound and outbound traffic as a function of bit rate. Lastly, the physical link power model is predetermined and is linearly dependent on bit rate and link length.

The output of the system is a set of routers and nodes, with two types of links: those between routers and those between routers and nodes. For every edge in the CTG, there is a route in the final design that satisfies its bandwidth and latency constraints. The bandwidth constraints on the router ports are also not exceeded. Lastly, the total power consumption for all routers, nodes, and links is minimized, and the number of routers is minimized. The authors state that power consumption can be minimized by minimizing traffic flow. They attempt to do this by placing communicating cores close to each other. However, this must be traded off with latency constraints.

3.1.1 Phase 1: Floorplanner

The first stage of the NoC synthesis is the initial floorplanner. The basic algorithm is a slicing tree described in [20]. The algorithm basically works by recursively dividing the layout into vertical and horizontal sections. They then use a graph equicut algorithm to partition the graph into sections with minimal total edge weight. This will minimize the bisection bandwidth requirements. Latency is given a higher priority than bandwidth because latency constraints cannot be alleviated by routing through a different path as bandwidth constraints can. The authors assume that routers are placed at the four corners of each core and use this assumption to estimate latency. Once the CTG and the layout are both partitioned, the CTG partitions are mapped to layout sections. After the initial floorplan is generated, it is compacted based on the physical size of the processing cores to generate the final layout. This is necessary because the slicing algorithm does not take into account the size and shape of the nodes.

3.1.2 Phase 2: Core to Router Mapping

The authors present a linear programming based technique called CMT (we guess author means this to stand for *Core Mapping Technique*.) that maps each node to one of the routers at its four corners. The choice of router impacts the overall communication cost since it is important for the routers for nodes with more communication between them be placed closer together. The purpose of the mapping phase is to provide a mapping from cores to routers that minimizes communication cost. This problem is a special case of the *Quadratic Assignment Problem (QAP)*. They approximate the problem as a *Linear Programming (LP)* with randomized rounding that can be solved in polynomial time.

Benchmark	Nodes	Edges	Power (μW)	Routers
dsp	6	5	1686	2
263 encoder	7	7	172.6	2
mp3 encoder	8	8	6.48	3
mpeg4	12	13	7392.18	3
mwd	12	13	993.6	3
vopd	12	13	2611.1	5
mp3 enc mp3 dec	13	12	9.15	4
263 dec mp3 dec	14	12	11.98	5
263 enc mp3 enc	15	17	181.6	4
263 enc 263 dec	16	17	159.5	5

TABLE 9
Results from [19]

3.1.3 Phase 3: Routing

The authors present a linear programming based algorithm for routing communication traces that minimizes the number of routers required. The algorithm assumes that all routing is minimal and deterministic. The algorithm minimizes the sum of all the routers required. Otherwise stated, the algorithm maximizes the sharing of routers between paths. However, the algorithm does not take into account maximum bandwidth constraints. After the initial mapping, those routes that violate bandwidth constraints are unmapped and rerouted using a shortest path algorithm.

3.2 Results

The authors tested their synthesis system on a variety of multimedia benchmarks including mp3 encoding and decoding, H.263 video, and MPEG4 among others. They estimated the input and output power for 100nm technology to be 328nW/Mbps and 65.5nW/Mbps, and the link power to be 80nW/Mbps/mm. The results for each application are shown in Table 9.

3.3 Critique

The main contributions made by Srinivasan et al. are the layout aware topology generation, and polynomial time analytical solutions to router mapping and trace routing. Unlike some other topology synthesis schemes, the authors estimated communication costs based on a preliminary layout of each core. This is a critical step that is overlooked by many papers [5], [15]. The latency and the energy cost of communication for NoCs is largely determined by link length. Link length also determines how many cycles a particular channel takes to traverse, as the authors point out. A signal can only travel a short distance in a single clock cycle, so it is often necessary to pipeline channels. This adds latency and power that is not visible unless the NoC is laid out before analysis. In addition, the nearly optimal solutions to minimize communication costs and routing resources are extensible. They can be applied to any communication trace graph and router architecture that is specified with the given constraints. The choice of constraints, such

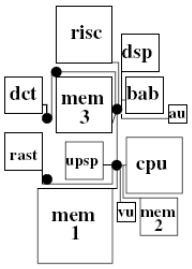

Fig. 4. Layout as shown in [19]

as port bandwidth, throughput, latency, is logical and relevant, as well as the choice to minimize system power and area. Moreover, both LP algorithms can be solved in polynomial time. This means that, although the authors used relatively small (10-16 nodes) networks in this paper, it is likely that these algorithms will scale to much larger networks.

Furthermore, the authors correctly observe that NoCs tend to have heterogeneous nodes and they account for such heterogeneity in the CTG by weighting the edges and in the layout. This allows the production of a network that is more suitable for the application. Those links that are more highly critical are shorter and shared less.

However, there are a few shortcomings in the analysis. The authors claim that by routing deterministically based on the CTG they can avoid network congestion altogether. It seems like this claim may not hold if the application has a traffic pattern that changes over time. A well formed synthesis tool should also account for flow control in the case of network congestion, and network congestion should also be taken into account when determining latency bounds and power estimation. In addition, the CTG is limited in that it cannot represent time-varying traffic. If the CTG provides peak bandwidth numbers, then it is likely that the network is overprovisioning bandwidth and is therefore inefficient. If the CTG provides average traffic, then applications with bursty traffic patterns may face unanticipated network congestion.

Moreover, the initial floorplan appears to be poor in terms of area utilization. The only constraint in the layout is minimizing communication cost. As a result, the layout looks quite suboptimal compared to conventional layout algorithms. The layout is shown in Figure 4.

In addition, it may be necessary to revise the layout once the router mapping and routing is completed, as this will likely change the latency and expected channel lengths. In fact, it may take many iterations through the three phases before a truly optimal solution is reached. Since the authors claim that their algorithm finishes quickly, it should not be prohibitive to iterate multiple times.

Another important issue is that the power estimations for router and link power provided by the authors are not substantiated. A reference is provided for power model of router architectures, but none in provided for link power. Even for the router architectures, no indication was given as to which specific router architecture is assumed. In addition to that, no performance numbers were provided. It is difficult to judge the produced network if the actual performance is not tested. If the power for these particular applications was measured, presumably they could have produced performance numbers as well. Performance numbers may also explain the huge discrepancy in some of the power statistics. MPEG4 consumed 7300μW with 12 nodes and 13 edges while mp3 encode and decode only consumed 9μW for 13 nodes and 12 edges. This discrepancy is not addressed in the paper.

There are a few key points that remain to be done in this study. Firstly, randomized rounding LP is a widely used approach, but they added a heuristic of using a shortest path when a constraint is violated. Therefore, the validity of that heuristic should be evaluated by comparing with an optimal or at least other related works. Secondly, it might be beneficial to try a wider variety of applications. All the applications evaluated were streaming multimedia applications. It would be interesting to see the results for something quite different. Thirdly, the authors should have considered technology scaling as well. As pointed by the previous paper [23], link power does not scale the same as router power, thus the tradeoffs may change with different technologies and it is important that their algorithm still produces optimal results. Lastly, the authors might also want to consider latency and bandwidth degradation from routing and flow control, in addition to evaluating the network for deadlock conditions.

4 Mapping and Physical Planning of Networks-on-Chip Architectures with Quality-of-Service Guarantees

Murali and De Micheli have published a series of papers about NoC design automation [10]. The authors describe the latest version of their mapping algorithm used in SUNMAP system. While Srinivasan et al. [19] focus on "generating" a customized irregular topology, this paper focuses on "mapping" an application to several regular topologies and then "selecting" among them. The authors consider almost every aspect of NoC design including topology, routing algorithm, flow control, QoS, router micro-architecture, link level error control and physical floorplan. They also use similar multimedia applications used by [19].

4.1 Solution

4.1.1 Mapping algorithm

Instead of approximating QAP with randomized rounding LP as [19], the authors use *robust tabu search* which is

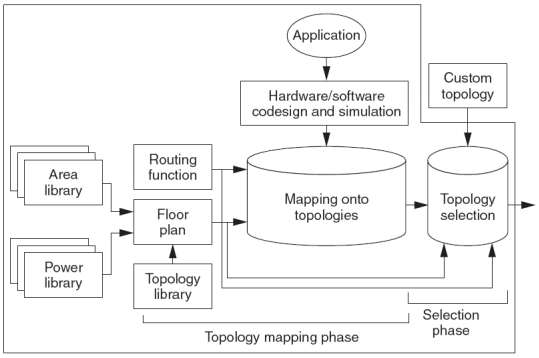

Fig. 5. SUNMAP design flow

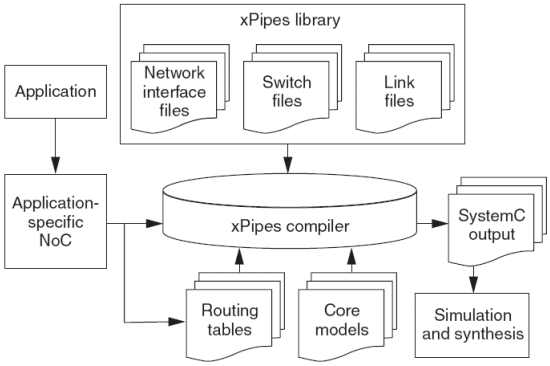

Fig. 6. xpipes synthesis flow

shown to be the most effective heuristic for many classes of QAP [21]. The authors also integrate physical floor planning algorithm into their search procedure for the best mapping. By this integration and using xpipes SystemC softcore, which is another part of NetChip framework, SUNMAP can accurately measure the cost of each mapping (Figure 5, 6). During the mapping procedure, SUNMAP also considers QoS guarantee.

The application is specified by two graphs: a *weighted core graph* and a *bandwidth constraint graph*. The *weighted core graph* is mostly equivalent to CTG in [19], except that each edge cost is the average bandwidth instead of the peak bandwidth and they are weighted by the criticality of the communication. For example, some control signals are more latency critical than data, thus should have higher priority. The bandwidth constraint graph specifies the minimum latency requirements of each bursty traffic.

Robust tabu search is based on local search, but provides more movements to escape from a local minima and uses a *tabu list* to escape from cyclic movements. SUNMAP starts from a greedy mapping where most communication demanding cores are mapped first. The *weighted core graph* is used at this greedy phase to measure which core is more communication demanding. The algorithm swaps the cores in each pair of positions as the movements for *robust tabu search*. For each movement, routing function, physical planning, buffer sizes are determined, and QoS constraints are checked against the *bandwidth constraint graph*. Then, design area, power and hop delay are estimated. After finishing the search, depending the object function, SUNMAP selects most area, power, or delay efficient mapping. SUNMAP repeats this procedure for every candidate topologies. It currently supports mesh, torus, hypercube, Clos and butterfly. Finally, xpipesCompiler synthesizes the best topology mapping using xpipes SystemC softcores.

[13] describes the details of routing function selection. They support the following routing functions: dimension ordered, minimum paths, traffic splitting across minimum paths, and traffic splitting across all paths. All of them are oblivious routing. For flow control, the algorithm can choose either wormhole or virtual channel flow control. xpipes softcores supports flexible number of pipeline stages to match the latency requirement.

4.2 Results

In the series of papers, the authors use multimedia applications similar to [19]. They also use network processors and digital filters. In [13], they present the first version of SUNMAP algorithm. They compare that with an optimal solution obtained using ILP and the previous work using branch-and-bound with heuristic [6]. They show that their algorithm achieves the average latency within 10% of an optimal and 32% less than the previous works. Some portion of the improvement is from their algorithm itself, and some other is from load balanced splitting routing algorithm. In [14], they extend the algorithm to multiple topologies and point out that an optimal topology varies on applications (Figure 7, 8). They also incorporate a simple initial floorplanning to have more accurate modeling. In this paper [10], the authors further integrate floorplanning with other parts of their algorithm. Instead of as an initial phase, the floorplanning is now executed in every step of *tabu search*. Moreover, the floorplanning algorithm is a more sophisticated one, which is based on *Mixed Integer Linear Program (MILP)* [8]. This optimization gives 1.4x area savings compare to the previous algorithm. They also extend the algorithm to support QoS. If the traffic is bursty, the average throughput and the peak throughput have a large discrepancy. A design for the average throughput results in high latency due to contention. On the other hand, a design for the peak throughput requires high bandwidth links. The authors demonstrate that QoS consideration provides a good tradeoff between these two approaches.

4.3 Critique

Murali and De Micheli made an important contribution by designing their algorithm within the framework which supports a broad aspect of NoC design automation. Many other papers evaluate their algorithm using inaccurate analytical models, which omit some

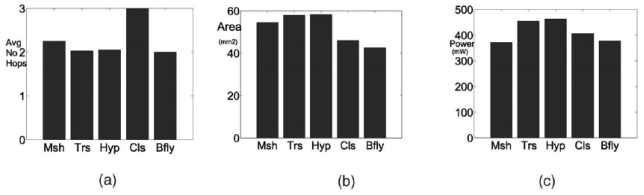

Fig. 7. Mapping results of VOPD on various topologies. (a) Avg. hop delay, (b) design area, and (c) design power.

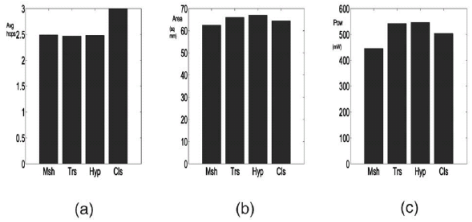

Fig. 8. Mapping results of MPEG4 on various topologies. (a) Avg. hop delay, (b) design area, and (c) design power.

aspects of NoC. For example, [6] models the energy consumption per flit transmission as the sum of switch energy (we assume this is the energy consumption in a router minus buffer energy consumption) and link energy. They neglect buffer energy by just saying that on-chip interconnection fabric should have small memories and that their algorithm guarantees small number of congestions thus reduces the number of buffer accesses. However, the energy consumption on each router should be very dependent on routing algorithm, flow control, and router micro-architecture. More significantly, we do not know what numbers are used as the constant factors in their analytical models. The problem of the previous paper [19] is stated above. In [15], Ogras et al. try to measure realistic energy consumption, but it is done for only one application (AES) using the Xpower utility of Xilinx. In contrast, Murali and De Micheli use the power mode from a credible source [4] for selecting an optimal mapping, and measure the energy consumption on the selected mapping by synthesizing `xpipes` softcores, which implements many aspects of NoC. Therefore, we can say that their measurement on delay, area, and power is much more reliable.

Another strong aspect of their papers is that they compared with other techniques. Many other papers about this topic provide no baseline [6], [19], [15], hence we cannot judge the usefulness of them. On the other hand, in their first paper [13], Murali and De Micheli compare their algorithm with an optimal solution and a previous work [6]. Subsequent papers [14], [10] compare the modified algorithms with previous ones. In spite of no direct comparison of their up-to-date algorithm with other work, at least we can do an indirect comparison.

In addition, despite a limited form, the authors consider the time variance of the traffic. They model the burstiness of a traffic by specifying the duration of the burst and the idle time between adjacent bursts (Figure 9). By using this model, they escape from the over-provisioning on the required bandwidth for each channel. Furthermore, in [11], they extend this work to support multiple use-cases.

Overall, the authors did thorough evaluations, but their contribution would have been more substantive if they had addressed following weaknesses. First, their consideration on time variation is somewhat limited.

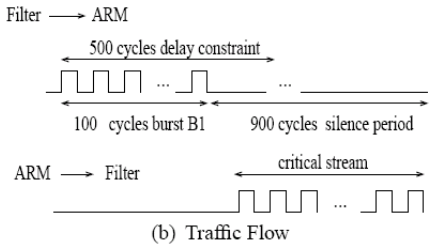

Fig. 9. Modeling of a burst traffic

They argue that the traffic patterns of many SoC applications do not vary a lot with input data. The authors refer two papers [16], [5] to support their argument, but the referenced papers actually provide no concrete result for the argument. It is also quite possible that traffics between different pairs of cores can have either statistical or deterministic correlations. Considering a large amount of cost reduction from burst traffic modeling shown in this paper, incorporating correlations between individual traffics also could result in another large cost reduction.

Secondly, the authors give no description on deadlock avoidance. In [12], the authors present a methodology to avoid higher level deadlocks in automated design tools for NoC. However, they mention nothing about more basic deadlock problems occurring from channel dependences. Since dimension order routing is not the only routing function they support, deadlock avoidance is necessary. In addition, deadlock avoidance can be the place where design automation plays its role most effectively: deadlock-free design is not trivial and can be error-prone, but a tool can efficiently perform that as shown in [7].

Lastly, there is no clear description on splitting traffic routing functions. Even for minimal load balanced oblivious routing, there are many possibilities such as Valiant within minimum quadrants, or selecting the path depending on even/odd position of source and/or target. The throughput requirement and design of deadlock avoidance should be dependent on the exact routing function.

5 Conclusion

Wang et al. [24] describes that an optimal topology changes over technology scaling and predicts that 2-D torus, H-3 torus and H-4 torus will be the most energy efficient topologies for 70nm, 50nm, and 35nm respectively. Srinivasan et al. [19] proposes an algorithm for automatic topology and route generation and argues that their low complexity algorithm achieves optimal results within their modeling. Murali and De Micheli [10] presents the mapping algorithm used in their `NetChip` framework. They demonstrates that the integration of physical planning into the mapping algorithm and burst traffic modeling has a significant impact on the quality of the results.

In general, the literature we looked at has the following issues.

Accuracy and Thoroughness of Models: Even if the algorithm is fast and achieves the near optimal of the formulated problem, it produces misleading results if the underlying modeling does not sufficiently reflect the real problem. All of the NoC design aspect, such as topology, routing strategy, flow control, micro-architecture, and physical planning, have some amount of impact on latency, throughput, area, and power consumption. A pertinent model should capture crucial parts of those aspects. It is especially true for the power consumption because it is strongly dependent on low level aspects: micro-architecture, physical planning, and circuit design. However, many papers neglect some crucial aspects. For example, [19] fails to consider the flow control which should affect the bandwidth constraint and deadlock problem. [10] use a reasonable simulation model based on softcore synthesis, but it might have been more through study if they had checked the validity of their results with a real implementation.

Time Variation of Traffic: Many papers [6], [7], [15], [19] optimizes network for peak bandwidth, which results in over-provisioning of required resources. [10] considers the burstiness of traffics, but does not capture dynamic variation of traffic characteristics. Embedded SoC systems are likely to have phases on their traffic pattern, and we can optimize for each phase. In addition, we can apply statistical analysis for the time variance of traffic patterns. By the same token, adaptive routing algorithm can be a good choice for some applications to deal with transient load imbalance. In [13], Murali and De Micheli already showed that non-minimal routing lessen the bandwidth requirement by splitting a have heavy traffic. [7] argues that deterministic routing algorithms should be used due to the resource limitation, stringent latency requirement, and traffic predictability requirement. However, there is no quantitative comparison about the tradeoffs between router complexity and channel bandwidth reduction.

Potential Problems of Customized Irregular Topologies: [19], [5], [15] argue that custom irregular topologies are more cost efficient than regular ones. However, those custom irregular topologies have two potential problems. The first one is sensitivity. Even an application specific SoC may change its configuration. For example, one if its core can be modified to support the next generation standard as a version-up. This kind of change can disturb highly tuned custom topologies and make the performance worse than regular topologies. Therefore, sensitivity on design changes should be evaluated. For instance, the performance of an irregular topologies tuned for a specific video standard can be also measured on similar but different video standards.

Scalability and Evaluation on Many Nodes: It is not clear if NoC is a good idea for small number of nodes. Someone should study where is NoC's reasonable break-even point compare with conventional techniques such as buses. If the break-even point is 16, then NoC synthesis algorithm should be evaluated on the networks larger than that. If we do not have a real application with that many nodes yet, the research community should develop a representative implementation, which can be used as a benchmark. If it is not possible, some synthetic traffic patterns should be established.

To make NoC technology widely acceptable, analytical evaluation and automated design has a critical role. The problem is that NoC design should consider various layers of chip design from device technology scaling to application characteristics. To address this issue, the research community should develop more through analytical models and design automation frameworks.

References

[1] Luca Benini and Giovanni De Micheli. Networks on chips: A new soc paradigm. *IEEE Transactions on Computer*, 35(1):70–78, 2002.

[2] William James Dally and Brian Towles. Route packets, not wires: On-chip interconnection networks. In *DAC '05: Proceedings of the 41st annual conference on Design automation*, pages 684–689, 2001.

[3] K. Goossens, J. van Meerbergen, A. Peeters, and P.Wielage. Networks on silicon: Combining best-effort and guaranteed services. In *DATE '02: Proceedings of the conference on Design, Automation and Test in Europe*, pages 423–425, 2002.

[4] Ron Ho, Kenneth W. Mai, and Mark A. Horowitz. The future of wires. *Proceedings of the IEEE*, 89(4):490–504, 2004.

[5] Wai Hong Ho and Timothy Mark Pinkston. A methodology for designing efficient on-chip interconnects on well-behaved communication patterns. In *HPCA '03: The Ninth International Symposium on High-Performance Computer Architecture*, pages 377–388, 2003.

[6] Jingcao Hu and Radu Marculescu. Energy-aware mapping for tiled-based noc architectures under performance constrains. In *ASP-DAC '03: Preceedings of the 2005 conference on Asia South Pacific design automation*, pages 233–239, 2003.

[7] Jingcao Hu and Radu Marculescu. Exploiting the routing flexibility for energy/performance aware mapping of regular noc architectures. In *DATE '03: Proceedings of the conference on Design, Automation and Test in Europe*, pages 688–693, 2003.

[8] Jae-Gon Kim and Yeong-Dae Kim. A linear programming-based algorithm for floorplanning in vlsi design. *IEEE Transactions on Compuiter-Aided Design of Integrated Circuits and Systems*, 22(5):584–592, 2003.

[9] K. Mai, T. Mori, B. Amrutur, R. Ho, B. Wilburn, M. Horowitz, I. Fukushi, T. Izawa, and S. Mitarai. Low-power sram design using half-swing pulse-mode techniques, 1998.

[10] Srinivasan Murali, Luca Benini, and Giovanni De Micheli. Mapping and physical planning of networks-on-chip architectures with quality-of-service guarantees. In *ASP-DAC '05: Proceedings of the 2005 conference on Asia South Pacific design automation*, pages 27–32, New York, NY, USA, 2005. ACM Press.

[11] Srinivasan Murali, Martjin Coenen, Andrei Radulescu, Kees Goossens, and Giovanni De Micheli. A methodology for mapping multiple use-cases onto networks on chips. In *DATE '06: Proceedings of the conference on Design, Automation and Test in Europe*, pages 118–123, 2006.

[12] Srinivasan Murali, Paolo Meloni, Federico Angiolini, David Atienza, Salvatore Carta, Luca Benini, Giovanni De Micheli, and Luigi Raffo. Designing message-dependent deadlock free networks on chips for application-specific systems on chips. In *VLSI-SoC '06: IFIP International Conference on Very Large Scale Integration*, 2006.

[13] Srinivasan Murali and Giovanni De Micheli. Bandwidth-constrained mapping of cores onto noc architectures. In *DATE '04: Proceedings of the conference on Design, Automation and Test in Europe*, page 20896, 2004.

[14] Srinivasan Murali and Giovanni De Micheli. Sunmap: A tool for automatic topology selection and generation for nocs. In *DAC '05: Proceedings of the 41st annual conference on Design automation*, pages 914–919, 2004.

[15] Umit Y. Ogras and Radu Marculescu. Energy- and performance-driven noc communication architecture synthesis using a decomposition approach. In *DATE '05: Proceedings of the conference on Design, Automation and Test in Europe*, pages 352–357, Washington, DC, USA, 2005. IEEE Computer Society.

[16] Alessandro Pinto, Luca P. Carloni, and Alberto L. Sangiovanni-Vincentelli. Efficient synthesis of networks on chip. In *ICCD '03: Proceedings of ths 21st International Conference on Computer Design*, page 146, 2003.

[17] Karthikeyan Sankaralingam, Ramadass Nagarajan, Haiming Liu, Changkyu Kim, Jaehyuk Huh, Doug Burger, Stephen W. Keckler, and Charles R. Moore. Exploiting ilp, tlp, and dlp with the polymorphous trips architecture. In *ISCA '03: 30th Annual International Symposium on Computer Architecture*, pages 422–433, 2003.

[18] Jaswinder Pal Singh, Wolf-Dietrich Weber, and Anoop Gupta. Splash: Stanford parallel applications for shared-memory. *ACM SIGARCH Computer Architecture News*, 20(1):5–44, 1992.

[19] K. Srinivasan, K. S. Chatha, and G. Konjevod. An automated technique for topology and route generation of application specific on-chip interconnection networks. In *ICCAD '05: Proceedings of the 2005 IEEE/ACM International conference on Computer-aided design*, pages 231–237, Washington, DC, USA, 2005. IEEE Computer Society.

[20] Krishnan Srinivasan and Karam S. Chatha. A technique for low energy mapping and routing in network-on-chip architectures. In *ISLPED '05: Proceedings of the 2005 international symposium on Low power electronics and design*, pages 387–392, New York, NY, USA, 2005. ACM Press.

[21] E. D. Taillard. Robust tabu search for the quadratic assignment problem. *Parallel Computing*, 16(4–5):443–455, 1991.

[22] Hangsheng Wang, Li-Shiuan Peh, and Sharad Malik. Power-driven design of router microarchitectures in on-chip networks. In *MICRO '03: Proceedings of the 36th annual IEEE/ACM International Symposium on Microarchitecture*, page 105, 2003.

[23] Hangsheng Wang, Li-Shiuan Peh, and Sharad Malik. A technology-aware and energy-oriented topology exploration for on-chip networks. In *DATE '05: Proceedings of the conference on Design, Automation and Test in Europe*, pages 1238–1243, Washington, DC, USA, 2005. IEEE Computer Society.

[24] Hansheng Wang, Xinping Zhu, Li-Shiuan Peh, and Sharad Malik. Orion: a power-performance simulator for interconnection networks. In *MICRO '02: Proceedings of the 35th annual IEEE/ACM International Symposium on Microarchitecture*, pages 294–305, 2002.

[25] Neil Weste and David Harris. *CMOS VLSI Design: A Circuits and Systems Perspective, 3rd edition*. Addison Wesley, 2004.