Parameters of a On-Chip Network for Multi-Processor System-on-Chip Environments

Sang Kyun Kim
skkim38@stanford.edu

Wook-Jin Chung
wookc@stanford.edu

1. INTRODUCTION

Network-on-chips (NoC) are replacing the conventional way of on-chip communication via buses (global wires) as we head into the billion transistor era. Such a transition is inevitable as the relative speed of wires do not improve in length alongside the technology scaling of transistors [4]. In addition, inserting regular switching fabrics is relatively straightforward within a modular system-on-chip (SoC) paradigm. This permits systematic network scaling and alleviates the VLSI design complexity problem [5][8].

As single processor performance approaches an inevitable wall, imposed by a multitude of factors, the exploitation of thread level parallelism (TLP) using chip multi-processors (CMP) is an attractive option [6]. In conjunction with the billion transistor era, this motivates multi-processor system-on-chips (MPSoC) where computational cores, memory and I/O all can reside on the same silicon. The network to connect these components thus becomes a critical factor in building these chips.

Multi-processor interconnect architecture has been well studied and key design parameters have been identified (for example, Cray supercomputers' interconnection networks) [7]. However, those results cannot be directly applied to MPSoC environments as it presents additional constraints as well as granting new resources. More importantly, because network architectures are highly sensitive in regards to being optimal for a particular task, it is difficult to generate a good solution via variable tweaking.

With this paper, we attempt to identify the important parameters involved in constructing on-chip networks for MPSoCs. We have chosen three previous works [1], [2], [3] where each captures a portion of the answer to our question. By analytically merging the boundaries of the papers, we hope to develop a higher level view that can be helpful in gaining a more complete picture of the design space.

The remainder of this paper is organized as follows. In Section 2, we describe in detail the difficulties of formulating the design space. Sections 3, 4 and 5 are dedicated to each of the papers where we discuss their contributions in solving the problem. We combine and analyze the knowledge gained from the papers in Section 6 and then follow up with future research directions (Section 7). Section 8 concludes the paper.

2. THE PROBLEM

There seems to be a virtual consensus in that NoCs should resemble interconnect architectures of high-performance parallel computing systems [2]. Thus the conventional performance metrics of throughput and latency are still applicable. However, due to the implications from the on-chip multi-processor environment (detailed in this section), the optimization goals are unclear alongside the set of metrics and parameters being incomplete.

2.1 Network on Silicon

In today's integrated circuit (IC) design, virtually all engineers must consider the power budget whereas transistors are practically free. This constraint is especially prominent in architecting processors [10]. Adding an entire network to a chip can only worsen the already tight limitation in power and thus energy efficient NoCs are highly in favor. Similarly, the additional silicon area required for implementing the network also should not be overlooked as it relates strongly to manufacturability and packaging.

On the other hand, there also are benefits when conceiving a network entirely on silicon. The on-chip wires are much shorter (compared to wires connecting neighboring chips), drastically reducing the propagation latency. Moreover, no longer is the number of pins on a node a concern which then allows vastly wide channels.

2.2 Multi-Processor Implications

As we restrict our NoC scope to MPSoC interconnects, this gives us a general direction to focus on because homogenous tiled processors should exhibit a particular traffic distribution, both in space and time. However, having processing cores as the network's client also complicates the problem as now the performance of the processors running applications must be considered when evaluating that of the network. It is not just that one should not be a bottleneck of the other but rather the two should be designed in concert to attain optimal performance for a finite set of resources.

2.3 Interdependencies

As seen with the relationship between the processors and the on-chip network, they are not independent because they contend for the same resource (silicon area) and share common or similar parameters (packet size). It is these interdependencies within the design spectrum that makes the problem of identifying key parameters hard. Due to this complex nature, all the papers of our selection ([1], [2], [3]) had to cull the exploration space in order to extract meaningful results or insights. In the following sections, our critique touches on this aspect as to whether such narrowing of scope is justifiable and how the results should be interpreted.

3. BALFOUR AND DALLY [1]

3.1 Overview

In this paper, the authors develop comprehensive analytical area and energy models specifically for MPSoC interconnection networks. With these, they then explore how parameters such as topology, routing algorithm, and router buffer size affect throughput, area and energy. Using the insights acquired from the experimental results, a new NoC architecture is proposed.

A 64 tile CMP is considered for the on-chip target environment. Each tile contains a processing core as well as a portion of the memory (256KB). The communication protocol between nodes is abstracted to request and replies where packets are of two lengths: 64 and 576 bits.

Balfour models the area and energy of the network components into a set of equations, parameterized in terms of the configuration that defines the architecture being evaluated. This configuration includes channel width, router buffer depth, and also technology dependent parameters such as metal pitch and transistor capacitance. The optimal set of values is found by sweeping these variables and measuring the performance of the network. For a systematic comparison, performance is defined with area efficiency (work completion time × chip area) and energy efficiency (work completion time × energy dissipated).

The paper evaluates the following networks: Mesh/MeshX2, Torus, Concentrated Mesh (CMesh)/CMeshX2, Fat-Tree (FTree), and Tapered Fat-Tree (TTree) [11]. CMesh is the new architecture proposed by the authors. As illustrated in Figure 1, it is a radix-4 mesh where each router serves 4 tiles. MeshX2 and CMeshX2 contain 2 complete set of networks in parallel. This design is reasonable as both Mesh and CMesh have enough unused silicon area for the second network.

The experimental results strongly favor the CMeshX2 architecture as it has both the best area and energy efficiency. Compared to all other topologies, CMeshX2 exhibited the fastest completion time with the smallest chip area, as well as having relatively low power dissipation.

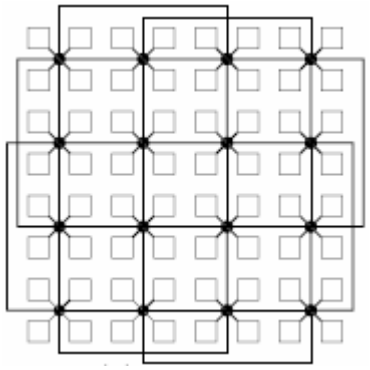

Figure 1. CMesh topology on 64 nodes. The express channels at the boundary routers permit equal degrees.

3.2 Main Contributions

Architecting the CMeshX2 is a large contribution of this paper. However it is equally as important to know what makes the design so efficient. Indeed the authors identify two critical characteristics that allow CMeshX2 to excel: concentration and having an independent second network. Concentration of nodes provides a more compact layout and reduces wire, allowing wider channel widths. Also fewer routers permit a lower hop count without increasing wiring complexity (as opposed to tree-based designs). Combined, the overall latency is drastically reduced. The added second network, not possible with all topologies, further reduces completion time via doubling peak throughput. Cost of the second network is higher power (but not area).

Knowing why CMeshX2 shines, this in turn reveals the weakness of tree based networks. Trees too are effective in reducing the average hop count and results indicate that they have better efficiencies than MeshX2. However, this gain pays the high price in wiring complexity. Not only does complex wiring inhibit wider channels (which may be acceptable because on-chip environment provides wider channels) but lengthens wires on average increasing energy dissipation. In addition, the irregularity of the wiring layout makes the insertion of the second network more difficult.

3.3 Critique

The paper compares NoC architectures with two primary performance metrics, area and energy efficiency. Since the workload completion time is a factor in both, it indeed is the most pivotal measure in the evaluation methodology. The workload used for this purpose was a mixture of common synthetic traffic patterns and CMeshX2 performed relatively well on all (to varying degrees). In spite of this, we are concerned with the fact that all synthetic traffic

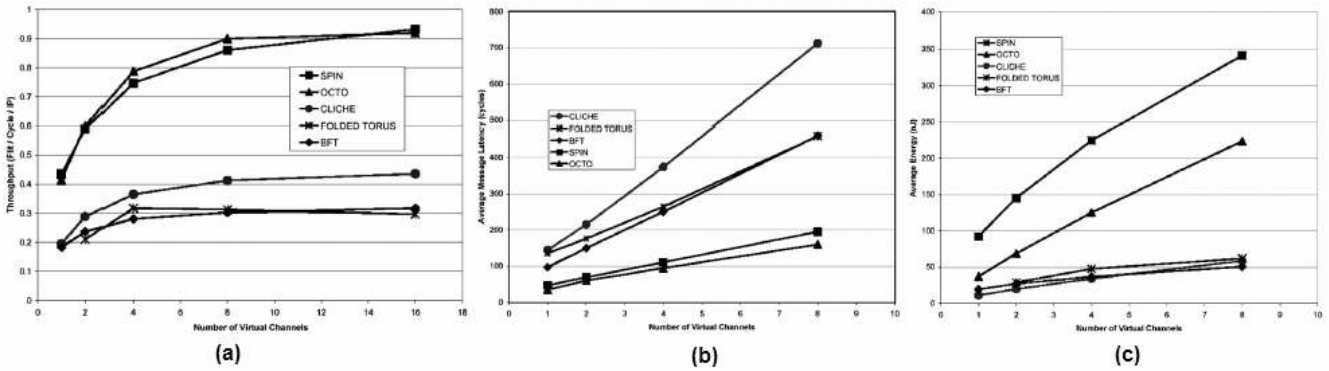

Figure 2. Number of virtual channels swept on all topologies. (a) shows the throughput, (b) the average message latency (cycles), and (c) graphs the average energy dissipation per packet (nJ).

patterns were weighted equally. Clearly uniform and taper traffic are more relevant to a CMP environment than others. However, large time savings from less relevant patterns also increased the relative difference in completion time with the same degree of force.

Another component that may have helped CMeshX2 seem superior to other designs is the unrealistic input traffic generation. According to [12], the experiments were run via simulating the worst case traffic generating application: a program solely composed of memory or cache coherence instructions. When purely comparing in terms of efficiency, this indeed proves that CMeshX2 outperforms its competitors. However, this raises the question as to whether with practical applications the architecture would demonstrate the same degree of relative efficiency gains.

CMesh uses concentration to reduce the number of hops and wires. It then comes to question why this technique was not attempted on the FTree. Currently, the FTree has a zero-load latency of 13 cycles. Even assuming no contention in the network, each request/reply transaction pair would take at least 26 cycles to complete. Given that a processor tile may have at most 4 pending transactions and assuming that an average packet takes around 3 cycles to send (short 1 cycle, long 4 cycles), the node network interface will be busy at most 12 cycles (out of the 26). If we use 2:1 concentrators, we can cut the number of switches and channels in half, greatly reducing the both area and energy. If we attempt to use 4:1 concentrators, although there may be contention at the injection point of the network, we can reduce the hop count by 2 as we rid of an entire level.

4. PANDE et al. [2]

4.1 Overview

The main task this paper takes upon is characterizing various NoC interconnect architectures with respect to their performance and design tradeoffs. With numerous proposed on-chip network topologies of the time (2005), the authors objectively sweep multiple parameters to determine their effect on performance. The findings of this paper are valuable not only because they paint a significant portion of the NoC design space but also due to its attempt in developing a meaningful comparison methodology.

The 5 architectures compared are mesh, torus, SPIN [13], Butterfly Fat-Tree (BFT) [14], and the Octagon [15]. SPIN essentially is a fat-tree where each level has both an uplink and a downlink degree of 4. BFT is a variant of the tapered tree [11] with a tapering factor of 2. Octagon, however, is unique in a sense that it is based off a tree or a mesh, but rather a ring topology. It groups 8 nodes into a ring with an addition of a bidirectional connection to the furthest node (diameter of the ring).

The comparison methodology uses throughput, latency, energy and area overhead as performance metrics. For all candidates, the following design parameters are swept: number of virtual channels (#VC), packet injection rate, localization factor, and input packet distribution. Localization factor represents the spatial distribution of the traffic where the number itself indicates the percentage of packets destined to its neighbors. Orthogonally, input packet distribution is related to the chronological distribution of the input. The authors consider a Poisson distributed injection as well as self-similar distribution [16] to model the bursty character of multi-processor environments.

The results show that it is possible to trade throughput and latency for energy and/or area by choosing the appropriate architecture. SPIN and Octagon differentiate themselves from the other three due to their exceptional throughput and low latency even under heavy loads. However, they also consume the most power and/or area. Such positioning on the design tradeoff space is mainly attributed to having more links between source and destination pairs.

4.2 Main Contributions

The paper does a good job mapping the relations between design parameters and performance. An excellent example of this is the #VC sweep against throughput, latency, and energy as depicted in Figure 2. From the three graphs, it is clear that using 4 virtual channels is the most attractive choice as it well balances all three performance measures. Of course, if one metric was more important than the other, the designer can now easily determine the cost of prioritizing that. Using the same methodology, as the entire set of results for all sweeps are presented in the paper, any particular tradeoff within the design space explored can be visualized.

Another important merit of the authors' study is the variation given to input traffic distribution. In terms of space, instead of just running the nearest-neighbor traffic, it was mixed with uniform traffic in various ratios. This is a more realistic way of modeling future MPSoC traffic patterns as there will be closely coupled[1] processor nodes that are mapped physically near each other. Regarding traffic distribution in time, using self-similar distribution to imitate the bursty character of memory requests once again is more representative of actual application behavior. Experimental results show that bursty traffic degrades throughput and average latency as well as consuming more energy.

The authors claim that the superior throughput and latency exhibited by SPIN and Octagon result from the high number of channels per node (compared to mesh, torus, and BFT). However Figure 3 shows that as the localization factor goes up, other topologies are on par. This is somewhat expected as fewer average hops would less pressure the network, nullifying the advantage of SPIN and Octagon. The critical finding here is that when localization is around 0.8 or higher, there is absolutely no reason to waste area and energy (using SPIN or Octagon).

4.3 Critique

The most questionable assumption made by the authors throughout the entirety of the paper is constant channel width. Given chip area constraints, it is logical that only a finite amount of wire can be placed. This would then signify that in order to increase the number of channels, the width must be reduced. If the channel width of SPIN and Octagon were reduced, this will increase serialization latency worsening both throughput and latency. (The authors avoid this accusation by calculating percentage area

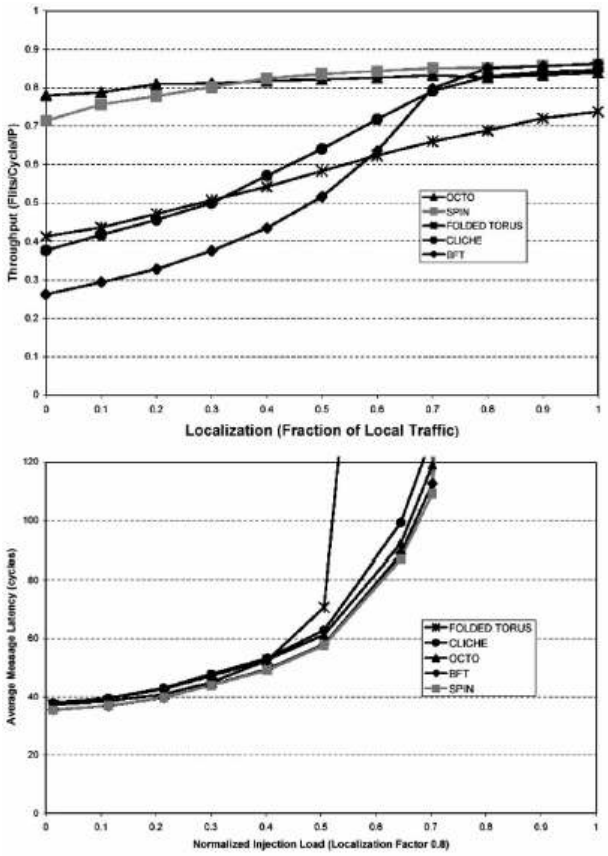

Figure 3. Throughput versus localization factor (up) and average message latency with a localization factor of 0.8 (down).

overhead for each architecture rather than keeping an upper bound on area.)

Relating to the constant channel width assumption, Pande et al. also never mention the actual channel width itself. Since their experimental framework uses flits as the base unit, it is presumed that a channel is wide enough to transfer one flit per cycle. Yet also in their experiment, they only use one message type which is 64 flits long. Considering that these messages were generated by processors requesting cache lines, which are typically 16, 32 or 64 bytes, it then makes each flit carry only 2, 4 or 8 bits of payload. Obviously this is too low even for an off-chip network and especially when we are expecting longer flits in on-chip environments. It is evident that the authors meant a larger packet and also their work is strictly not restricted to MPSoC purposes (cache line requests, coherence protocols). But it is hard to overlook the inconsistency where some parts of the paper assume processor nodes while others do not.

Lastly, a breakdown of energy and area between routers, wires, repeaters, and processors would have given precious insight for determining the appropriate tradeoff parameters.

[1] Closely coupled processors are a pair or group of processor nodes that communicate with one another extensively. Close coupling may occur between processors that are running threads of the same application.

For example, two architectures with similar throughput, latency and area overhead are indistinguishable in their characteristics in this paper. However, it may be the case that their area constitution was drastically different.

5. YE et al. [3]

5.1 Overview

This paper describes the impact of packetization on on-chip networks. Ye et al. tackle the question of how different packet sizes affect the performance of not just the interconnection network but rather the multi-processor system as a whole. For investigation, switch buffer depth and packet size are swept with the critical presumption of tying the packet payload to the cache line size.

The system considered throughout the paper is an on-chip multi-processor with 16 tiles, interconnected with a mesh network. Each tile has a processing core, 16KB L1 and 64KB L2 cache. The channel width is considered to be 64 bits wide. A packet is defined to include at least a head and a tail flit. Read replies and write requests contain payload data which is the size of a single L2 cache line. Real application benchmarks (*quicksort*, *sor*, *water*, *lu*, and *mp3d*) are run for generating network traffic.

The switch buffer depth sweep shows that both dimension order and adaptive routing with flit-based flow control benefit from larger buffer depths in respect to average packet delay. However, only dimension order routing's execution time is reduced considerably with growing buffer sizes. This is evidence that adaptive routing better load balances the traffic as there are few pile-ups. Experimenting with various packet sizes (or namely L2 cache line sizes), there exists a tradeoff relationship between cache miss rate and miss penalty. Longer packets induce a lower miss rate, but in turn increases the miss penalty. The optimal packet payload size was determined to be 64 bytes.

The authors also demonstrate a balance between network energy and processor node energy depending on the packet size. As the payload of the packets increases, fewer misses at the processor reduce both cache and memory energy. Conversely, the network energy increases with the heavier load. The optimal payload size of 128 bytes gave minimum system energy.

5.2 Main Contributions

This paper points out the critical interdependency between packet sizes and cache line sizes for an interconnect network serving an MPSoC client. As mentioned by the authors, due to cache coherence protocols and implementation of conventional memory operations, it is only sensible that the two are equal to one another. With this new restriction, it can be claimed that for optimal

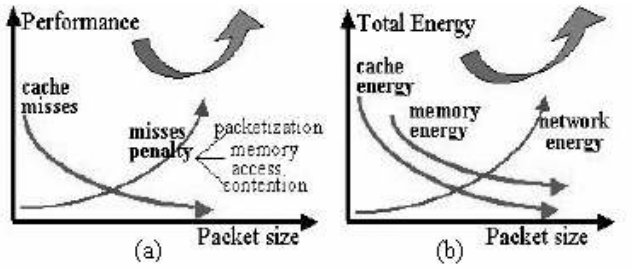

Figure 4. Qualitative analysis on the impact of packet size. For both (a) and (b), lower is better.

performance, CMP design decisions must entail simultaneous considerations of both the processors and the network. Moreover, the percentage of total chip energy that the network consumes will only grow as technology scales [17].

Figure 4 summarizes the impact of packet sizes. The effect on completion time is illustrated in (a) and system energy in (b). Miss penalty increases super-linearly as larger packets not only increase packetization and memory access delay but also create higher contention (for virtual channels). This in turn causes more misroutes (with adaptive routing) and increases the average number of hops. This same justification can be used to explain super-linear growth of network energy as energy is proportional to the time a packet spends on the network. The key insight to be gained from the two diagrams is that there exists an optimal packet size which provides best performance regarding execution time and energy.

5.3 Critique

Looking back at Figure 4, it is noted that both findings are derived from the presumption that cache miss rate decreases with increasing line size. Surely this is true for many applications as they possess some degree of spatial locality. Putting aside applications that do not have spatial locality, the authors never mention the possibility of cache miss rates increasing along with line size. If the total cache size is fixed, having larger cache lines would translate into fewer lines of cache. This would then raise the conflict miss rate as more memory would map to the same line of cache. Furthermore, with multi-processor cache coherence, having larger lines increases the chances of being invalidated and therefore results in more coherence misses. We recognize that even with rising cache miss rates, there still exists an optimal packet size. But just simply presuming that miss rates will decrease is unacceptable especially within multi-processor design space.

Another shortcoming of the paper is with the description of the contention-look-ahead routing algorithm. Essentially, this is a non-minimal adaptive routing scheme which is then prone to both deadlock and livelock. As it demonstrates the

best performance, the authors use this algorithm for their experiments. However, never has it been mentioned how both deadlock and livelock are avoided.

One way to avoid deadlocks is to use virtual channels. Yet the authors were reluctant in implementing the idea with the rationale that buffers are relatively expensive resources in an on-chip environment. On the other hand, they could have left the buffer size constant and just varied the number of virtual channels instead of a buffer size sweep. We predict that with virtual channels, the optimal packet size will become larger as both cache miss penalty and network energy would not increase as fast due to increased throughput. But because virtual channels increase latency, it may worsen overall performance with short packet sizes.

6. ANALYSIS

The three papers discussed in the previous sections suggest a multitude of design parameters. However, before we fudge boundaries to construct a high level picture, we would like to resolve what seems to be a major discrepancy between two papers. While [1] suggests that CMeshX2, a mesh based design, has the best throughput than any other competitors, results from [2] illustrate that SPIN (fat-tree based) and Octagon (ring based) have far better throughput than both the mesh and the torus. This may raise eyebrows at first sight but careful scrutiny assures us that both results are valid. The experimental space of [1] allowed variable channel widths according to topologies and CMesh, due to its simple wiring, was permitted double the width. [2] on the other hand fixed the channel width taking away the advantage of mesh's simplicity. Moreover, judging from the fact that MeshX2 performs the worst and CMeshX2 the best, we can assume that the "concentration" factor opens a whole new dimension that [2] does not encompass. Thus all direct comparisons to CMesh can be nullified.

Figure 5 is the comprehensive parameter map that we have constructed. It is our solution for visualizing the complex design space. The white boxes indicate design specifications and constraints which the network designer has little control over. The grey boxes are parameters where there is at least some degree of freedom. Large and bold grey boxes are the most flexible and critical parameters. The arrows denote the influence (or interdependence) of one parameter on another. Each section of the maps is detailed below:

(1) In an MPSoC environment, packets are usually memory requests and replies. Thus packet payload size must equal the cache block size. (Coherence protocols bolster this constraint.) Depending on the implementation, pre-fetching may require packets to carry multiples of the block size. Technology scaling may further limit packet sizes as network energy will eventually dominate total chip energy [17].

(2) The physical layout of the topology will be influenced by the area constraint and the number of nodes. With a coarse layout, we can then determine the available channel width via fitting wires atop. Concentration also needs to be considered as it reduces the number of channels, enabling the increase in width. (The upper limit on channel width is packet size.) Concentration is usually desirable if the shared channel has enough bandwidth. Knowing the traffic pattern in advance drastically helps decision making. If the localization factor is high, topologies that exploit locality (mesh, BFT) are preferred. For uniform traffic, those with low hop counts and high bisection bandwidth (CMesh, FTree, Octagon) are appropriate. Remember that the 2nd network can be inserted for any topology as long as there exists enough unused space.

(3) If routing complexity must be avoided, deterministic routing with large buffer size is suggested. Else adaptive routing is preferred. Flit-based flow control will typically be used in on-chip networks since packet-based flow control cannot take advantage of the abundant bandwidth and requires more buffers.

(4) Buffers may be partitioned into multiple virtual channels to increase throughput. Generally, this results as longer packet latency. Regardless of virtual channels, adaptive routing algorithms may benefit with smaller buffer depths for building up fast back pressure. For faster traffic information propagation, dedicated control lines may be used.

(5) The architecture of the router depends on many network parameters: topology, routing algorithm, flow control and virtual channels. For example, topology determines the radix and the routing algorithm must be built into the router. If area permits, separate input buffers for different packet sizes reduce energy consumption.

By no means do we consider our map to be final. However we do hope that it is easy to understand and inserting new dependencies, constraints or parameters is straightforward. This visualization tool is not a formula or methodology for designing NoCs but rather a reminder for the designer of existing degrees of freedom and other considerations to be made.

7. FUTURE RESEARCH

In constructing an experimental target platform for on-chip networks, all previous works that we reviewed assume a homogeneous node containing a processor and some

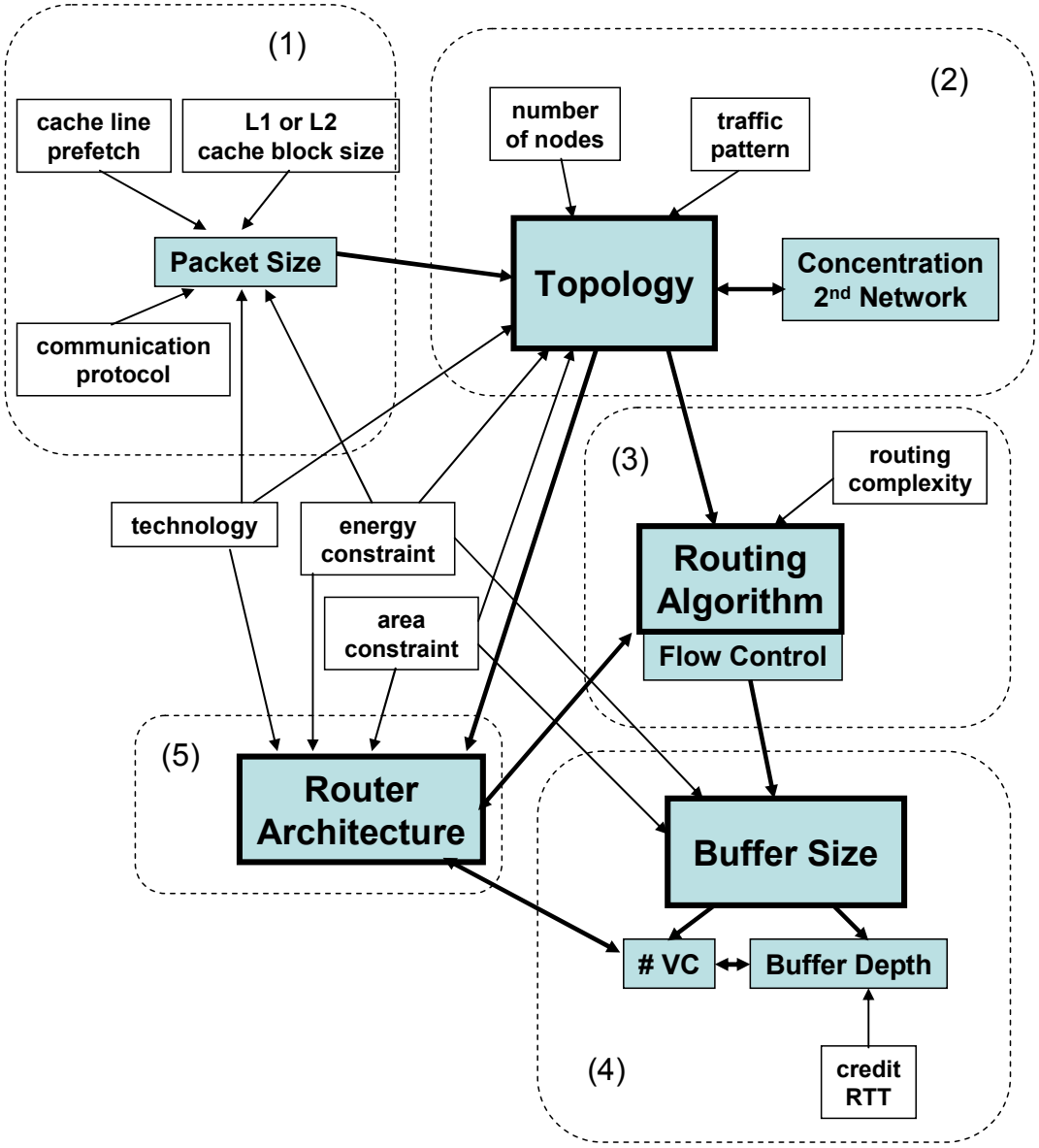

Figure 5. Parameter map of MPSoC network design space.

amount of memory. This simplifies the problem of designing the interconnection fabric due to the symmetric nature. In contrast, it is highly probably that SoCs require heterogeneous nodes. For example, some nodes may have application specific hardware (soundcard, graphics chip, etc.) This additional degree of freedom (heterogeneity) then brings upon a whole new set of parameters, one of them being the mapping of nodes onto the provided network [9]. From the MPSoC perspective, it would be interesting to design topologies for processors mixed with memory nodes as commercial CMPs yearn for abundant on-chip memory.

The primary performance metrics of on-chip networks seems to be the following: throughput, latency, chip area and energy. However, as it will become possible to place 1000s of nodes on a chip, we think that metrics such as testability and reliability should also be included in evaluating architectures. Even putting these new measures aside, we believe that the difficulty of designing on-chip networks will increase exponentially as the number of nodes scale due to the wire scaling problem and energy constraints. Only future research will tell if such networks are realizable.

8. CONCLUSION

NoCs cannot directly inherit interconnection architectures of existing parallel computers as the MPSoC environment

provides a different set of constraints and resources. To make matters worse, due to the complex interdependencies between architectural parameters, the complete design space is hard to visualize. By merging the exploration boundaries and insights of three previous works, we have constructed a comprehensive parameter map for multi-processor on-chip networks. In addition, we identify the more critical parameters that exercise much influence to others. We recognize that our map is not absolute and that it may be expanded with future research. However, we do believe that we have covered a significant portion of the design space.

9. REFERENCES

[1] Balfour, J., and Dally, W.J. Design Tradeoffs for Tiled CMP On-Chip Networks. In *Proceedings of the International Conference on Supercomputing,* 2006. 187-198.

[2] Pande, P.P. et al. Performance Evaluation and Design Trade-Offs for Network-on-Chip Interconnect Architectures. *IEEE Trans. Computers, vol. 54, no. 8,* Aug. 2005, 1025-1040.

[3] Ye, T.T., Benini, L., and De Micheli, G. Packetization and Routing Analysis of On-Chip Multiprocessor Networks. *Journal of Systems Architecture, vol. 50, no. 2-3,* Feb. 2004. 81-104.

[4] Ho, R., Mai, K.W., and Horowitz, M.A. The Future of Wires. In *Proceedings of the IEEE, vol. 89, no. 4,* Apr. 2001. 490-504.

[5] Hemani, A. et al. Network on a Chip: An Architecture for Billion Transistor Era. In *Proceedings of the IEEE NorChip Conference.* Nov. 2000.

[6] Olukotun, K. et al. The Case for a Single-Chip Multiprocessor. In *Proceedings of the International Conference on Architectural Support for Programming Languages and Operating Systems*, 1996. 2-11.

[7] Dally, W.J. and Towles. B. *Principles and Practices of Interconnection Networks.* Morgan Kaufmann Publishers, 2004.

[8] Benini, L. and De Micheli, G. Networks on Chips: A New SoC Paradigm. *Computer, vol. 35, no. 1.* Jan. 2002. 70-78.

[9] Hu, J. and Marculescu, R. Energy-Aware Mapping for Tile-Based NoC Architectures Under Performance Constraints. In *Proceedings of the Conference on Asia South Pacific Design Automation*, 2003. 233-239.

[10] Hennessy, J. and Patterson, D. *Computer Architecture, Fourth Edition: A Quantitative Approach.* Morgan Kaufmann Publishers, 2006.

[11] Ohring, S.R. et al. On Generalized Fat Trees. In *Proceedings of the International Symposium on Parallel Processing*, Apr. 1995, 37-44.

[12] Balfour, J. Interview. May 11th, 2007.

[13] Guerrier, P. and Greiner, A. A Generic Architecture for On-Chip Packet-Switched Interconnections. In *Proceedings of the Design and Test in Europe,* Mar. 2002. 250-256.

[14] Pande. P.P. et al. Design of a Switch for Network on Chip Applications. In *Proceedings of the International Symposium on Circuits and Systems, vol. 5,* May 2003. 217-220.

[15] Karim, F., Nguyen, A., and Dey, S. An Interconnect Architecture for Networking Systems on Chips. *IEEE Micro, vol. 22, no. 5,* Sep./Oct. 2002. 36-45.

[16] Park, K. and Willinger, W. *Self-Similar Network Traffic and Performance Evaluation.* John Wiley & Sons, 2000.

[17] Kapur, P. et al. Technology and Reliability Constrained Future Copper Interconnects – Part II: Performance Implications. *IEEE Transactions on Electron Devices, vol. 49, no. 4,* Apr. 2002. 598-604.