A System-C based Microarchitectural Exploration Framework for Latency, Power and Performance Trade-offs of On-Chip Interconnection Networks

Basavaraj Talwar and Bharadwaj Amrutur
Electrical and Communication Engineering Department, Indian Institute of Science, Bangalore.
Email: {bt,amrutur}@ece.iisc.ernet.in

***Abstract*— We describe a System-C based framework we are developing, to explore the impact of various architectural and microarchitectural level parameters of the on-chip interconnection network elements on its power and performance. The framework enables one to choose from a variety of architectural options like topology, routing policy, etc., as well as allows experimentation with various microarchitectural options for the individual links like length, wire width, pitch, pipelining, supply voltage and frequency. The framework also supports a flexible traffic generation and communication model. We provide preliminary results of using this framework to study the power, latency and throughput of a 4x4 multi-core processing array using mesh, torus and folded torus, for two different communication patterns of dense and sparse linear algebra. The traffic consists of both Request-Response messages (mimicing cache accesses) and One-Way messages. We find that the average latency can be reduced by increasing the pipeline depth, as it enables higher link frequencies. We also find that there exists an optimum degree of pipelining which minimizes energy-delay product.**

I. Introduction

On-chip interconnection networks (ICN) are critical elements of modern system-on-chip as well as multi-core designs. These chips have a multiplicity of communicating entities like programmable processing elements, hardware acceleration engines, memory blocks as well as off-chip interfaces. The communication patterns between these entities is very application dependent and diverse in terms of connectivity, burstiness, latency and bandwidth requirements. With power having become a serious design constraint, there is a great need for designing ICN which meets the target communication requirements, while minimizing power using all the tricks available at the architecture, microarchitecture and circuit levels of the design.

Many simulation tools have been developed to aid designers in ICN space exploration [1] [2]. These tools usually model the ICN elements at a higher level abstraction of switches, links and buffers and help in power/performance trade-off studies [3]. These are used to research the design of Router architectures [4] [5] and ICN topologies [6] with varying area/performance trade-offs for general purpose SoCs or to cater to specific applications. Kogel et. al. [1] present a modular exploration framework to capture performance of point-to-point, shared bus and crossbar topologies. Impacts of varying topologies, link and router parameters on the overall throughput, area and power consumption of the system (SoCs and Multicore chips) using relevant traffic models is discussed in [7]. Orion [2] is a power-performance interconnection network simulator that is capable of providing power and performance statistics. Orion model estimates power consumed by Router elements (crossbars, fifos and arbiters) by calculating switching capacitances of individual circuit elements. Most of these tools do not allow for exploration of the various link level options of wire width, pitch, serialization, repeater sizing, pipelining, supply voltage and operating frequency.

On the other hand, tools exist to separately explore these low level link options to various degrees as in [8], [9] and [10]. Work in [8] explores use of heterogeneous interconnects optimized for delay, bandwidth or power by varying design parameters such as a buffer sizes, wire width and number of repeaters on the interconnects. Courtay et. al [9] have developed a high-level delay and power estimation tool for link exploration that offers similar statistics as Intacte does. The tool allows changing architectural level parameters such as different signal coding techniques to analyze the effects on wire delay/power. Intacte [10] provides a similar capability to explore link level design options and is used in this research.

It is clear from works like [11] that there is a need for a co-design of interconnects, processing elements and memory blocks to fully optimize the overall system-on-chip performance. This necessitates a simulation framework which allows a co-simulation of the communicating entities along with ICN simulation. Additionally, to optimize power fully, one also needs to incorporate the link-level microarchitectural choices of pipelining etc. Hence we are developing a System-C framework which enables one to hook up actual communicating entities, along with the ICN and also allows for exploration of architectural and microarchitectural parameters of the ICN, in order to obtain the latency, throughput and power trade-offs. Results of trade-off studies in this paper consider Energy-Delay product (of the NoC) as the optimization parameter. Effects of wire density and area of NoC have not been taken into account in our experiments. We defer this study for future work.

We report on the design of this framework in System-C in Section II. We are using this framework to study the network design of a multi-core chip, supporting various communication patterns as in [12] for different classes of parallel computing benchmarks. We use a mix of Request-

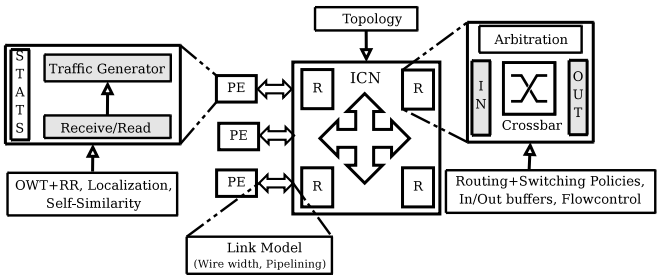

Fig. 1. Architecture of the SystemC framework.

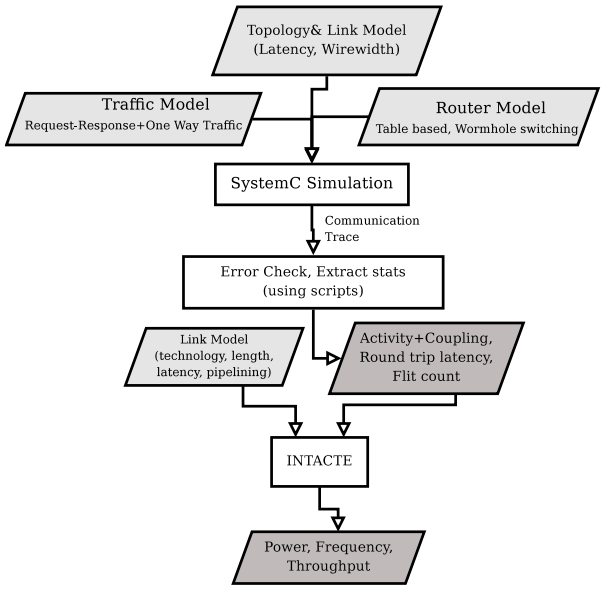

Fig. 2. Flowchart depicting simulation steps.

Response and One-way traffic generation model to mimic realistic traffic patterns generated by these benchmarks. We use two benchmarks, Dense Linear Algebra (DLA) and Sparse Linear Algebra (SLA) benchmark communication patterns on three NoC topologies (2D Mesh, 2D Torus and Folded 2D Torus) to determine the average latency, throughput and power under different amounts of link pipelining and present some preliminary results in Section III. We draw some conclusions and outline future work in Section IV.

II. NoC Exploration Framework

The NoC exploration framework (Figure 1) has been built upon Open Core Protocol-IP models [13] using OSCI SystemC 2.0.1 [14] on Linux (2.6.8-24.25-default). The framework contains Router, Link and Processing Element (PE) modules and each can be customized via various parameters and will be described in more detail next. The NoC modules can be interconnected to form a desired NoC. The PE module represents any communicating entity on the SoC and not just the processing element. We can either hookup an actual executable model of the entity or some abstract model representing its communication characteristics. For abstract models, we support many different traffic generation and communication patterns. The link module can be used to customize the bit-width of the links as well as the degree of pipelining in the link. A single run (Figure 2) uses these models to run a communication task and outputs data files of message transfer logs. From these log files, one-way and round trip flit latency, throughput and link capacitance activity factors are extracted. Intacte is then used to obtain the final power numbers for different operating frequency and supply voltage options. Table I summarizes the various parameters that can be varied in the framework.

TABLE I
ICN EXPLORATION FRAMEWORK PARAMETERS.

Parameter	*Description*
NoC Parameters	
Routing Algorithms	Source Routing and Table based routing
Switching Policy	Packet, Circuit, Wormhole, VC switching
Traffic Paradigm	Request-Response & One-Way Traffic
Traffic Generation Scheme	Deterministic, Self-Similar
Traffic Distribution Scheme	Deterministic, Uniformly random HotSpot, Localized, First Matrix Transpose
Router Microarchitecture	
No. of Input/Output Ports	2-8 (based on topology to be generated)
Input/Output buffer sizes	Flit-level buffers
Crossbar Switching capacity	In terms of flits (default=1)
Link Microarchitecture	
Length of interconnect	Longest link in mm
Bit width of the interconnect	
Circuit Parameters	
Frequency, Supply Voltage	

A. NoC Elements

1) Traffic Generation and Distribution Models (PE): To test NoCs on realistic multi-core applications we setup traffic generation and distribution to mimic various communication patterns. We support Request-Response (RR) and One-Way Traffic (OWT) generation. For example in multi-core chips, the former can correspond to activities like cache line loads and the latter can correspond cache line write backs.Traffic distribution input is given using two matrices of sizes N×N, where N is the number of communicating entities. Item (i, j) in a matrix gives the probability of communication of PE i with j in the current cycle. Two separate matrices correspond to Request-Response (RR) and One-Way traffic (OWT) generation. The probability of choosing among the two matrices depends on a global input to decide the percentage of RR traffic to be generated for the simulation run. This model can be further expanded to capture burst characterisitics as well as message size and is something we plan to add in the future. The communication packets are broken into a sequence of Flit transfers. The Flit header format is shown in Figure 3. The SQ field is used to identify in order arrival of all flits. Response flits have first 2 bits set to 11. SRCID, DSTID and FlitID fields are preserved in Response flit for the sake of error checking and latency calculations in the framework. The traffic receiver will read the header to determine if the flit type is RR or not (flag RQ). If RQ is set, then the Traffic

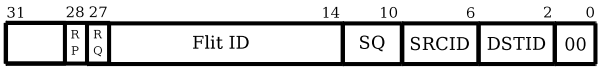

Fig. 3. Flit header format. DSTID/SRCID: Destination/Source ID, SQ:Sequence Number, RQ & RP: Request and Response Flags and a 13 bit flit id.

TABLE II

TRAFFIC GENERATION/DISTRIBUTION MODEL AND EXPERIMENT SETUP.

Parameter	Values	
Communication Patterns	DLA Traffic	SLA Traffic
NoCs Simulated	2D Mesh, Torus and Folded Torus	
Localization Factor	0.7	0.5
Traffic injection rate	20%	
RR Factor	0.03	0.1
Size of phit (Wire width)	32 bits	
OW and RR Request Flit	1 flit (2 phits)	
RR Response Flit	3 flits (6 phits)	
Simulation Time	40000 cycles	
Process	45nm	
Environment	Linux (2.6.8-24.25-default)+ OSCI SystemC 2.0.1 + Matlab 7.4	

generator is notified and the flit header is sent to the Traffic generator. RR traffic has priority over OW traffic and hence the request will be immediately serviced (without breaking an OW flit). Response flit to the request flit has RP set and RQ reset. In a received flit, if RQ is not set, then no action is taken. Table II lists out parameters used in our traffic model and in experiments. The framework is also capable of generating Deterministic, Uniformly Random, Hotspot and First Matrix Transpose traffic distributions.

2) Router Model: The router model is a parameterized, scalable module of a generic router [7]. *Router microarchitecture* parameters include number of Input/Output ports, sizes of input/output buffers, switching capacity of the crossbar (no. of bits that can be transfered from input to output buffers in a cycle) etc (Table I). Flow control is implemented through sideband signals [13].

B. Power Model

Intacte [10] is used for interconnect delay and power estimates. Design variables for Intacte's interconnect optimization are wire width, wire spacing, repeater size and spacing, degree of pipelining, supply (V_{dd}) and threshold voltage (V_{th}). Activity and coupling factors are input to Intacte from the System-C simulation results. Intacte arrives at a power optimal number of repeater, sizes and spacing for a given wire to achieve a desired frequency. The tool also includes flop and driver overheads for power and delay calculations. Intacte outputs total power dissipated including short circuit and leakage power values. We arrive at approximate wire lengths using floorplans. Other physical parameters are obtained from Predictive Technology Models [15] models for $45nm$.

Power consumed by routers have not been included in the results presented in the paper and will be added in the future.

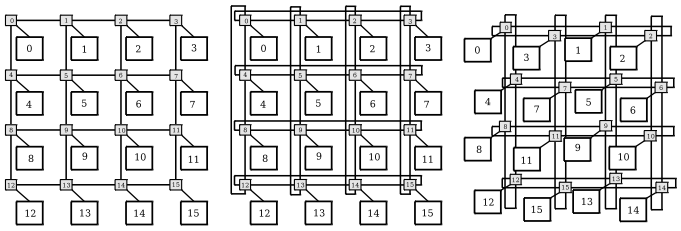

Fig. 4. Schematic of 3 compared topologies (L to R: Mesh, Torus, Folded Torus). Routers are shaded and Processing Elements(PE) are not.

However we can still draw some useful conclusions about those aspects of the ICN design which relate to the links like the degree of pipelining and optimal toplogy.

III. SIMULATION AND RESULTS

We study a 4x4 multi-core platform for three different network topologies of Mesh, Torus and Folded-torus. We use two communication patterns from [12] of Dense Linear Algebra (DLA) and Sparse Linear Algebra (SLA) benchmarks. DLA applications exhibit highly localized communication. The traffic model for DLA generates 70% traffic to immediate neighbors and remaining traffic is distributed uniformly to other nodes. SLA communication is reproduced using 50% localized traffic and rest of the traffic is destined to half of the remaining nodes. Further we assume all RR traffic to be localized. For eg. 10% of generated traffic over the simulation per PE will be of Request type if RR=0.1. All Request flits are destined to immediate neighbors. 70% of flits generated by any PE over the simulation time are destined to immediate neighbors if localization factor is 0.7(as in case of DLA).

Experiments are designed to calculate latency (clock cycles), throughput (Gigabits/sec) and power (milliWatts) of various topologies. Table II lists some of the simulation setup parameters used in the following experiments.

A. NoC Topologies

In this work we consider three similar topologies for tradeoff studies. Router and processing elements are identical in all three topologies. In fact the same communication trace is played out for all the different ICN parameter explorations. The schematic of the three NoCs is shown in Figure 4 with the Floorplans largely following the schematics. The floorplans are used to estimate the wire lengths which are then input to Intacte. Processing elements sizes are estimated by scaling down the processor in [16] to $45nm$to be of size 2.25×1.75mm. The routers are of size 0.3×0.3mm. The length of the longest links in the Mesh, Torus and Folded Torus are estimated as $2.5mm$, $8.15mm$ and $5.5mm$ respectively. The longest link in the torus connect the routers at the opposite sides. The routing policy for all topologies is table based. Routing tables are populated such that longer links have minimum activity. Lengths of links in each of the topologies and pipelining factors is illustrated in Table III. Pipelining factor corresponds to the longest link in the NoC. Pipelining factor of 1 means the longest link is

unpipelined, P=2 indicates it has a two cycle latency and so on.

TABLE III
LINKS AND PIPELINING DETAILS OF NOCS

Topology	Length in mm (no. of links)	Pipelining							
2D Mesh	2.5 (24)	1	2	3	4	5	6	7	8
	2.0 (56)	1	2	3	4	4	5	6	7
2D Torus	8.15(8)	1	2	3	4	5	6	7	8
	6.65(8)	1	2	3	4	4	5	6	7
	2.5 (24)	1	1	1	2	2	2	3	3
	2.0 (56)	1	1	1	1	2	2	2	2
Folded 2D Torus	5.5 (16)	1	2	3	4	5	6	7	8
	4.5 (16)	1	2	3	4	5	5	6	7
	2.75(16)	1	1	2	2	3	3	4	4
	2.25(16)	1	1	2	2	3	3	3	4
	2.0 (32)	1	1	2	2	2	3	3	3

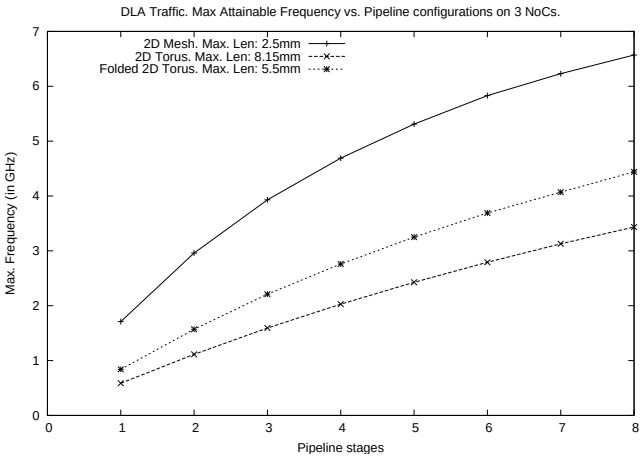

Fig. 5. Max. frequency of links in 3 topologies. Lengths of longest links in Mesh, Torus and Folded 2D Torus are $2.5mm$, $8.15mm$ and $5.5mm$.

B. Round Trip Flit Latency & NoC Throughput

Round trip flit latency is calculated starting from injection of the first phit (physical transfer unit in an NoC) to the reception of the last phit. In the case of OW traffic latency is one way. In case of RR traffic it is the delay in clock cycles of beginning of request injection to completion of response arrival. Communication traces are analysed using error checking (for phit loss, out-of-order reception, erroneous transit etc.) and latency calculation scripts to ensure functional correctness of the system.

Total throughput of the NoC (in $bits/sec$) is calculated as total number of bits received ($(flit_r * bits_{flit})$) at sink nodes divided by total (real) time ($(\frac{1}{f} * sim_{cycles})$) spent (Eqn 1).

$$Th_{total} = \frac{flit_r * bits_{phit}}{\frac{1}{f} * sim_{cycles}} \quad (1)$$

Max achievable frequency of a wire of given length is obtained using Intacte(Figure 5). Max throughput of each NoC running DLA traffic at P=1 is shown in Figure 6.2D Mesh has the shortest links and highest achievable frequency and hence the highest throughput.

Average round trip latencies in nano-seconds over various pipeline configurations in all 3 NoCs is shown in Fig. 7. Results show overall latency of flits actually decrease to a certain point by pipelining. Avg. latencies are larger for RR type of traffic and it also has a larger number of flits involved (1 Req + 3 Response). Clearly, there is a latency advantage by pipelining links in NoCs upto a point. This is because as the number of pipe stages increase, the operation frequency can also be increased as the length of wire segment in each pipe stage decreases. Real time latencies do not vary much after pipelining configuration P=5, as delay of flops start to dominate and there is not much marginal increase in frequency. Throughput and Latency behaviour for SLA traffic are identical (not shown here).

C. NoC Power/Performance/Latency Tradeoffs

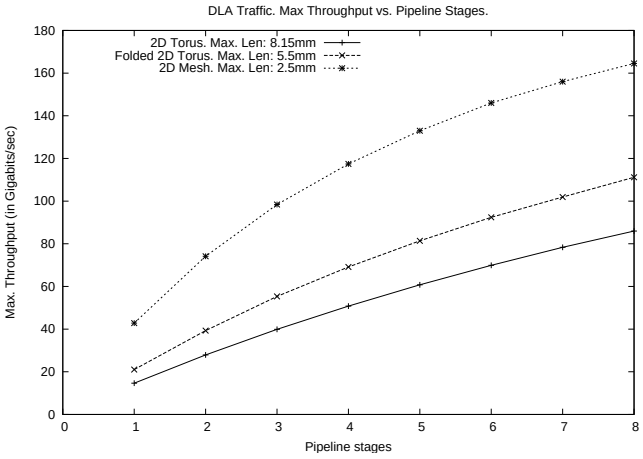

Fig. 6. Total NoC throughput in 3 topologies, DLA traffic.

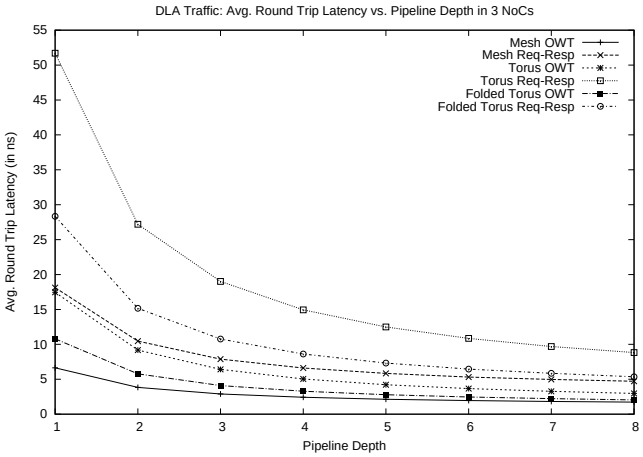

Fig. 7. Avg. round trip flit latency in 3 NoCs, DLA traffic.

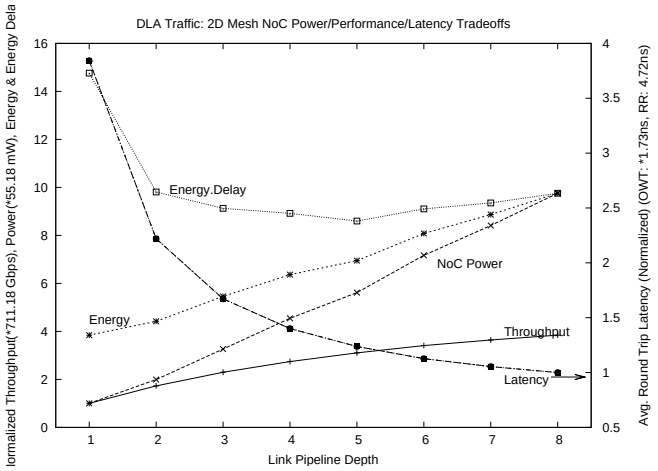

Fig. 8. 2D Mesh Power/Throughput/Latency tradeoffs for DLA traffic. Normalized results are shown.

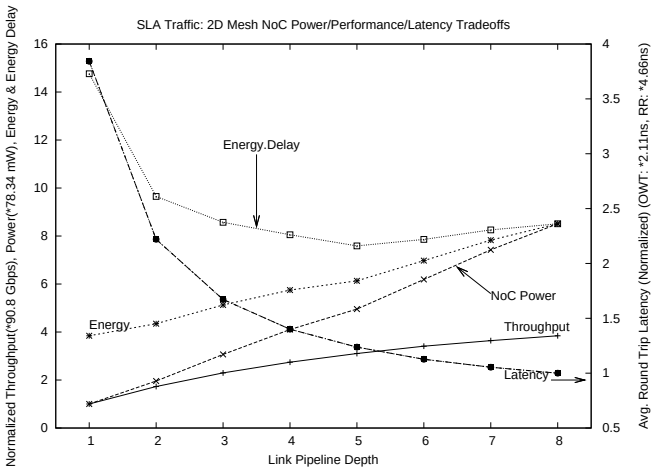

Fig. 9. 2D Mesh Power/Throughput/Latency tradeoffs for SLA traffic.

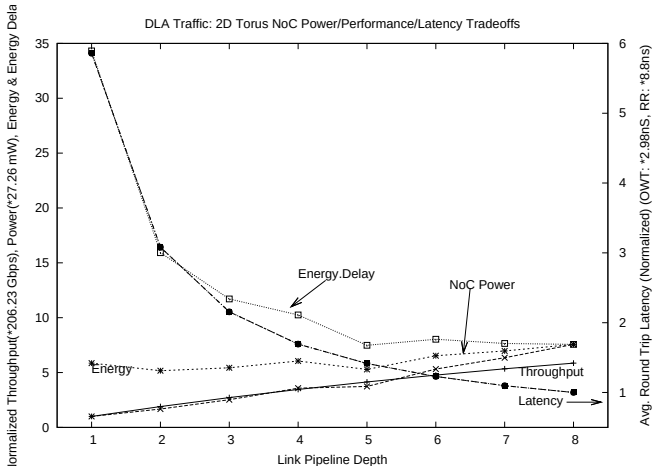

Fig. 10. DLA Traffic, 2D Torus Power/Throughput/Latency tradeoffs. Normalized results are shown.

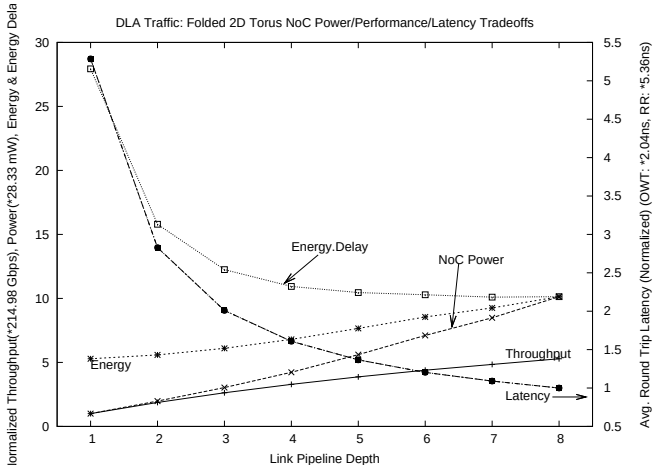

Fig. 11. DLA Traffic, Folded 2D Torus Power/Throughput/Latency tradeoffs. Normalized results are shown.

1) 2D Mesh: Figure 8 and 9 shows the combined normalized results of NoC power, throughput and latency experiments on a 2D Mesh for DLA and SLA traffic. Throughput and power consumption are lowest at P=1 and highest at P=8. Normalized avg. round trip flit latency for both OW and RR traffic is shown (the curves overlap). From the graph it is seen that growth in power makes configurations more than P=5 less desirable. Link pipelines with P=1,2 and 3 are also not optimal with respect to latency in both these benchmarks. Rise in throughput also starts to fade as configuration of more than P=6 are used. The optimal point of operation indicated by the results from both communication patterns is P=5. Energy curve is obtained as the product of normalized Latency and Power values. Energy for communication increases with pipeline depth. Energy Latency (Energy.Delay) is the product of Energy and Latency values. Quantitatively the optimal point for operation is when the longest link has pipeline segments (P=5). In DLA traffic, Avg. round trip flit latency of flits in the NoC is 1.23 times minimum and 32% of maximum possible. NoC power consumed is 57% of max and throughput 80.5% of max possible value.

2) 2D Torus and Folded 2D Torus: Similar power, throughput and latency tradeoff studies are done on both communication patterns on 2D Torus (Fig. 10) and Folded 2D Torus (Fig. 11) NoCs. Results obtained in 2D Torus experiments indicate that growth in power makes configurations more than P=5 is not desirable. Latencies of flits in pipeline configurations P=1-4 are large. Rise in throughput also starts to fade as configurations after P=5 are used. The optimal point of operation indicated by the Energy Delay curves in both DLA and SLA traffic (not shown here) for 2D Torus is P=5. In DLA traffic, this configuration shows power consumed by the NoC is 50% of the value consumed at P=8 and throughput is 70.5% the max value. Avg. Round Trip latency of flits for both OW & RR traffic is 1.4 times minimum and 24% of the maximum (when P=1).

Tradeoff curves for the Folded 2D Torus show similar trends as in the 2D Torus. Avg. round trip flit latency reduction and throughput gain after P=6 is not considerable. There is no single optimum obtained from the Energy Delay curve.

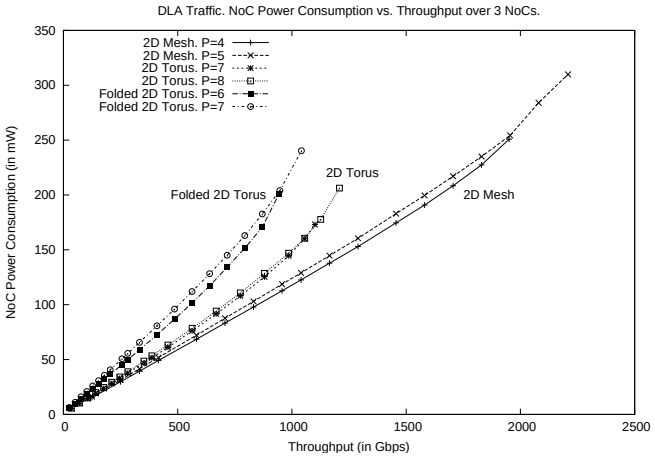

Fig. 12. Frequency scaling on 3 topologies, DLA Traffic.

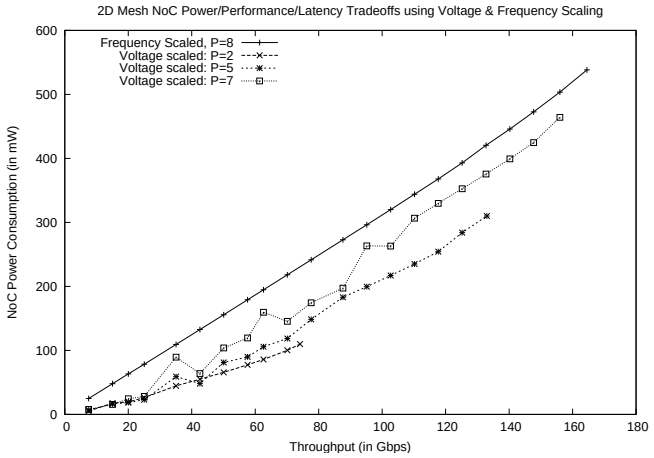

Fig. 13. Dynamic voltage scaling on 2D Mesh, DLA Traffic. Frequency scaled curve for P=8 is also shown.

Pipeline configurations from P=5 to P=7 present various throughput and energy configurations for approximately same Energy Delay product.

D. Power-Performance Tradeoff With Frequency Scaling

We discuss the combined effects of pipelining links and frequency scaling on power consumption and throughput of the 3 topologies (Figure 4) running DLA traffic. Maximum possible frequency of operation at full supply voltage $(1.0V)$ is determined using Intacte.

Figure 12 shows NoC power consumption for 3 example topologies over a pair of pipelining configurations along with frequency scaling (at V_{dd}). As observed from the graph, power consumption of a lower pipeline configuration exceeds the power consumed by a higher configuration after a certain frequency. Larger buffers (repeaters) are added to push frequencies to the maximum possible value. Power dissipated by these circuit element start to outweigh the speed advantage after a certain frequency. We call this the "crossover" frequency. The graph shows 3 example pairs from each NoC from each of the topologies to illustrate this fact.

Maximum frequency of operation of an unpipelined longest link in a 2D Mesh $(2.5mm)$ is determined to be 1.71GHz. This maximum throughput point is determined in each pipeline configuration in each topology. Frequency is scaled down from this point and power measurements are made for NoC activity obtained using the SystemC framework for DLA traffic.At crossover frequencies it is advantageous to switch to higher pipelining configurations to save power and increase throughput. For example in a 2D Mesh, link frequency of 3.5GHz can be achieved by pipelining configuration of 3 and above. NoC power consumption can be reduced by 54% by switching to a 3 stage pipeline configuration from 8 stage pipeline configuration.In other words, a desired frequency can be acheived by more than one pipeline configuration. For example, in a 2D Torus frequency (throughput) of 2.0GHz can be achieved by using pipeline configurations from 4 to 8. NoC power consumption can be reduced by 13.8% by switching from P=8 to P=4 and still achieve similar throughput.

TABLE IV
DLA TRAFFIC, FREQUENCY CROSSOVER POINTS IN 2D MESH

Pipe Stages	Trip Frequency (in GHz)		
	Mesh	Torus	Folded Torus
1-2	1.7	0.25	0.45
2-3	2.96	0.7	1.5
3-4	3.93	1.1	2.0
4-5	4.69	2.0	2.76
5-6	5.31	2.2	3.2
6-7	5.83	2.8	3.69
7-8	6.23	3.0	4.07

E. Power-Performance Tradeoff With Voltage and Frequency Scaling

In each topology, frequency is scaled down from the maximum and the least voltage required to meet the scaled frequency is estimated using Intacte and power consumption and throughput results are presented. Voltages are scaled from $1.0V$ till 0.1GHz is met for each pipelining configuration in each NoC.Similar to the frequency scaling results there exists a crossover frequency in a pipelining configuration after which it is power and throughput optimal to switch to a higher pipelining stage (Table IV).Figure 13 compares Power and Throughput values obtained by voltage and frequency scaling with a frequency scaled P=8 curve for 2D Mesh with DLA traffic. Scaling voltage along with frequency compared to scaling frequency alone can result in power savings of upto 14%, 27% and 51% in cases of P=7, P=5 and P=2 respectively.

Comparison of all 3 NoCs is presented in Table V.

IV. CONCLUSION

Consideration of low level link parameters like pipelining, bit widths, wire pitch, supply voltage, operating frequency etc, along with the usual architectural level parameters like router type, topology etc., of an ICN enables better optimization of the SOC. We are developing such a framework in System-C

TABLE V
COMPARISON OF 3 TOPOLOGIES FOR DLA TRAFFIC.

Topology	Pipe Stages	Power (mW)	Performance (Gbps)
Mesh	1	55.18	42.82
	2	109.87	74.12
	4	250.83	117.44
	7	464.16	156.00
Torus	1	27.26	14.67
	2	45.71	27.89
	4	97.48	50.78
	7	206.22	78.33
Folded Torus	1	28.32	21.03
	2	55.95	39.31
	4	119.75	69.11
	7	287.18	101.91

since it can allow co-simulation with models for the communicating entities along with the ICN.

Preliminary studies on a small 4x4 multi-core ICN for three different topologies and two different communication patterns indicate that there is an optimum degree of pipelining of the links which minimizes the average communication latency. There is also an optimum degree of pipelining which minimizes the energy-delay product. Such an optimum exists because increasing pipelining allows for shorter length wire segments which can be operated either faster or with lower power at the same speed.

We also find that the overall performance of the ICNs is determined by the lengths of the links needed to support the communication patterns. Thus the mesh seems to perform the best amongst the three topologies we have considered in this study. This opens up interesting research opportunities for reconfigurable ICNs with heterogenous links which can support different patterns efficiently.

It also points to an overall optimization problem that exists in the architecture of the individual PEs versus the overall SOC, since smaller PEs lead to shorter links between PEs, but more traffic, thus pointing to the existence of a sweet spot in terms of the PE size.

ACKNOWLEDGMENT

We thank Shailesh Kulkarni for help with the initial development of this framework. We acknowledge funding support from Texas Instruments, India.

REFERENCES

[1] T. Kogel et. al., "A modular simulation framework for architectural exploration of on-chip interconnection networks," in *Proc. of, Hardware/Software Codesign and System Synthesis, 2003 (CODES+ISSS'03). Intl. Conf. on*, pp. 338–351, Oct. 2003.

[2] H.-S. Wang, X. Zhu, L.-S. Peh, and S. Malik, "Orion: A power-performance simulator for interconnection networks," in *Proc. of, MICRO 35*, 2002.

[3] P. Gupta, L. Zhong, and N. K. Jha, "A high-level interconnect power model for design space exploration," in *Proc. of, Computer Aided Design (ICCAD '03). Intl. Conf. on*, pp. 551–558, 2003.

[4] K. Lee, S.-J. Lee, and H.-J. Yoo, "Low-power network-on-chip for high-perforamance soc design," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 14, pp. 148–160, Feb. 2006.

[5] S. E. Lee, J. H. Bahn, and N. Bagherzadeh, "Design of a feasible on-chip interconnection network for a chip multiprocessor (cmp)," in *Proc. of, Computer Architecture and High Performance Computing. Intl. Symp. on*, pp. 211–218, 2007.

[6] F. Karim et. al., "An interconnect architecture for networking systems on chips," *IEEE Micro*, vol. 22, pp. 36–45, Oct. 2002.

[7] P. P. Pande, C. Grecu, M. Jones, A. Ivanov, and R. Saleh, "Performance evaluation and design trade-offs for network-on-chip interconnect architectures," *IEEE Transactions on Computers*, vol. 54, pp. 1025–1040, Aug. 2005.

[8] R. Balasubramonian, N. Muralimanohar, K. Ramani, and V. Venkatachalapthy, "Microarchitectural wire management for performance and power in partitioned architectures," in *Proc. of, High-Performance Computer Architecture. HPCA-11. 11th International Symposium on*, pp. 28–39, Feb. 2005.

[9] A. Courtey, O. Sentieys, J. Laurent, and N. Julien, "High-level interconnect delay and power estimation," *Journal of Low Power Electronics*, vol. 4, pp. 1–13, 2008.

[10] R. Nagpal, M. Arvind, Y. N. Srikanth, and B. Amrutur, "Intacte: Tool for interconnect modelling," in *Proc. of 2007 Intl Conf. on Compilers, Architecture and Synthesis for Embedded Systems(CASES 2007)*, pp. 238–247, 2007.

[11] R. Kumar, V. Zyuban, and D. M. Tullsen, "Interconnections in multi-core architectures: Understanding mechanisms, overheads and scaling," in *Proc. of, Computer Architecture. ISCA '05. 32nd International Symposium on*, pp. 408–418, 2005.

[12] K. Asanovic, R. Bodik, B. C. Catanzaro, J. J. Gebis, P. Husbands, K. Keutzer, D. A. Patterson, W. L. Plishker, J. Shalf, S. W. Williams, and K. A. Yelick, "The landscape of parallel computing research: A view from berkeley," Tech. Rep. UCB/EECS-2006-183, EECS Department, University of California, Berkeley, Dec 2006.

[13] http://www.ocpip.org/socket/systemc/, "Ocp-ip, systemc ocp models."

[14] http://www.systemc.org/, "Open systemc initiative."

[15] http://www.eas.asu.edu/∼ptm/, "Predictive technology models."

[16] G. Konstadinidis et. al., "Implementation of a third generation 16-core, 32-thread, cmt sparc processor," in *ISSCC '08: Processor of the International Solid-State Circuits Conference*, pp. 84–85, IEEE, 2008.