SYNTHESIS OF NOC INTERCONNECTS FOR CUSTOM MPSOC ARCHITECTURES

GUL N. KHAN AND ANITA TINO
Department of Electrical and Computer Engineering
RYERSON UNIVERSITY
Toronto, ON Canada

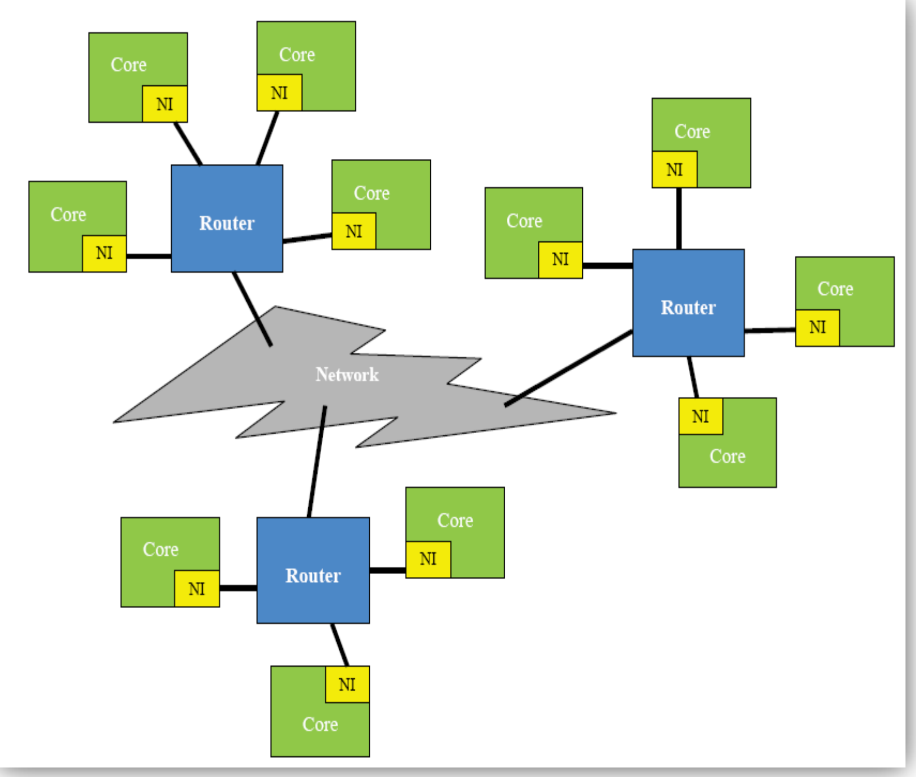
NOCS 2012

OVERVIEW

- Introduction to NoCs – Power and Performance
- NoC Topology Design Flow
- Concept Overviews
    - TABU Search
    - Layered Queuing Networks (LQN)
- Topology Generation Technique
    - Initial Solution
    - Successive Filter Strategy, neighborhood selection
    - Contention Analysis
- Experimental Results
- Final Remarks

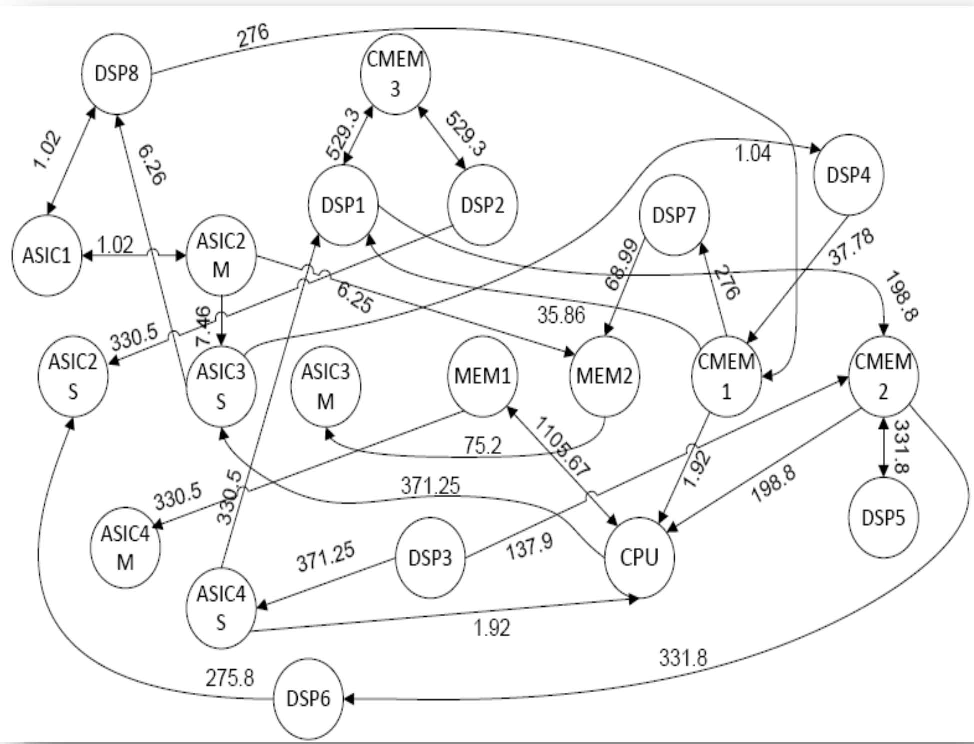
NOC INTRODUCTION

- Network-on-Chips (NoC)
  - Application Specific
- Power
  - Automated Technique
  - Models: Static, Dynamic, Leakage
- Performance
  - Contention Modeling
  - Deadlock Avoidance
- Trade-offs



NOC DESIGN FLOW

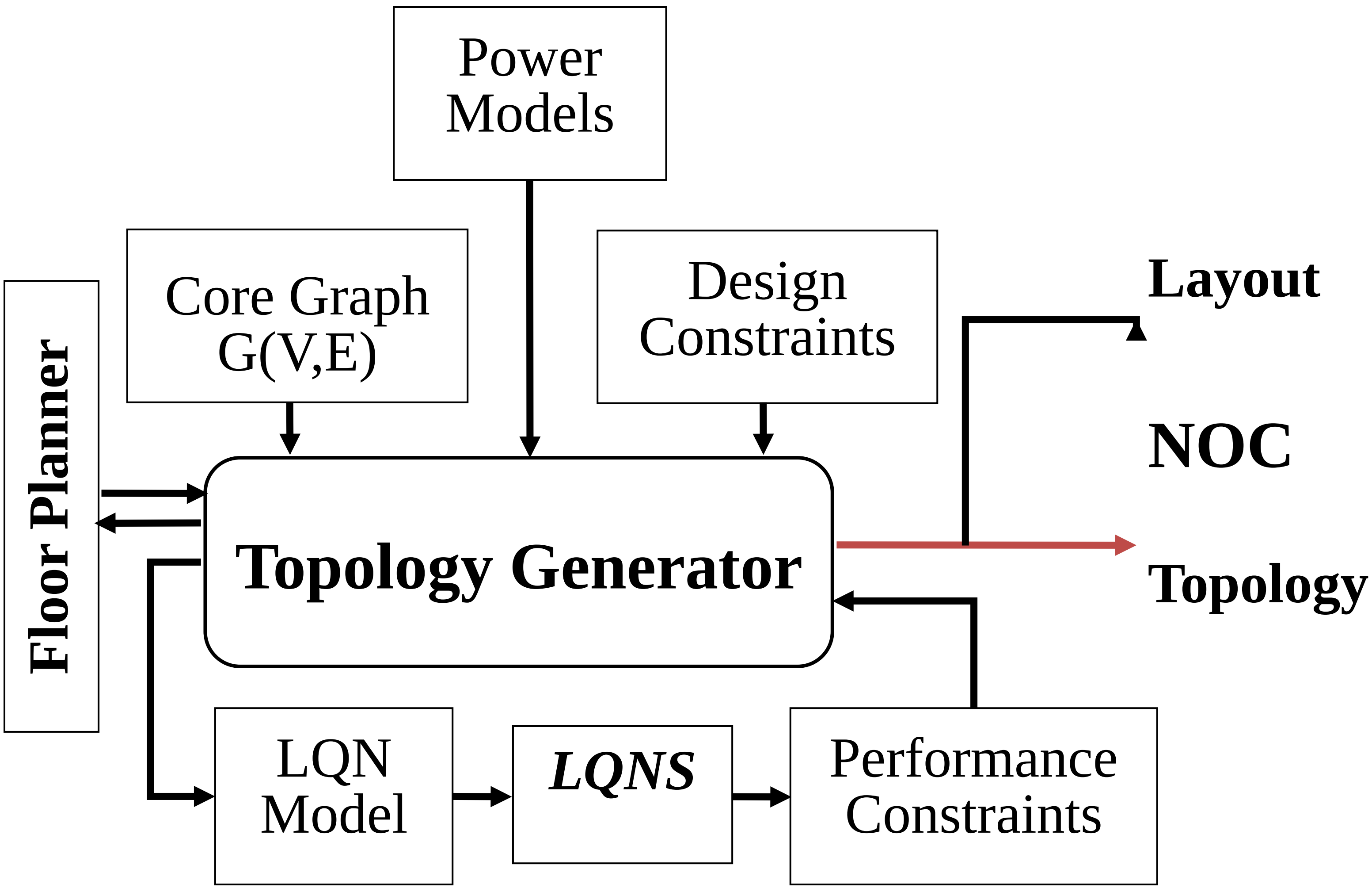
Input
Core Graph



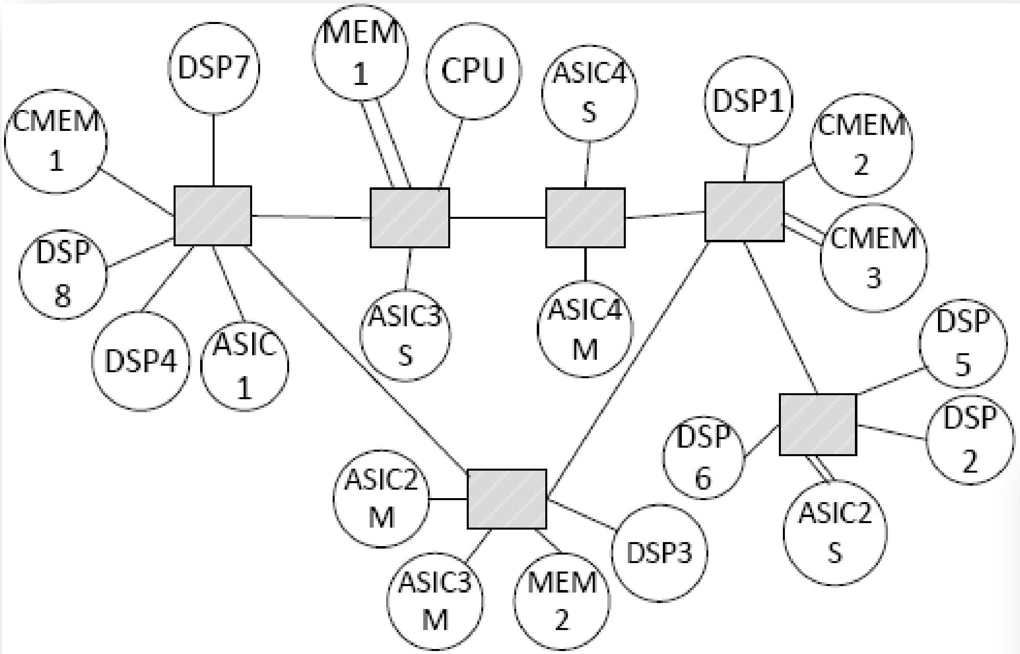
Input directed graph *G(V,E)*

- Each vertex $v_i \in V$ represents a core within the graph.
- The communication between vertex *i* and *j* represent a directed edge (v_i, v_j), expressed as $e_{i,j} \in E$.
- The weight found on an edge $e_{i,j}$ denoted by $b(e_{i,j})$ characterizes the bandwidth.
- A destination vertex (core) d_x, where $d_x \in V$ may have 1 to many sources cores s_x.
- The source vertex $s_x \in V$, and $x = \{1 \ldots N\}$.
- *N* represents the number of cores in the core graph.

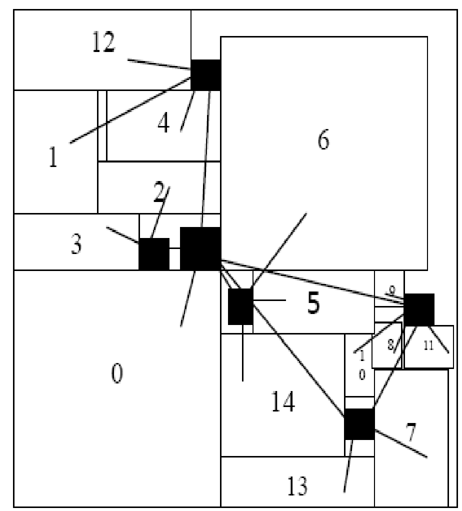
NOC DESIGN FLOW



NOC DESIGN FLOW

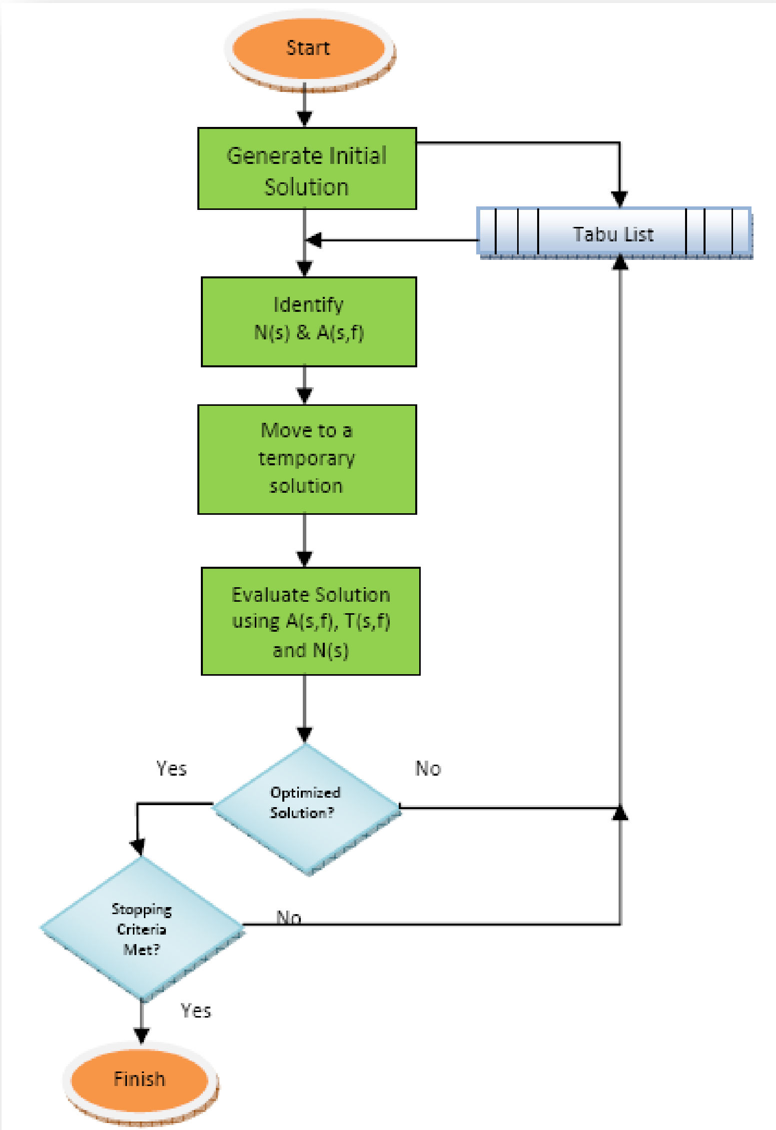


Output : Topology & Floorplan



Core Num	Core Name
0	ARM
1	ROM
2	SWITCH
3	RAM2
4	USB2.0
5	EXT-IF
6	ASIC1
7	SDRAM-IF
8	TIMER
9	UART
10	RTC
11	ITC
12	RAM1
13	DMA
14	RAM3

TS: TABU SEARCH



- Meta-heuristic optimization method designed to escape the trap of local optimums.
- Start with initial feasible solution
- Iterates through a neighborhood of possible solutions until optimal solution is found.
- Tabu List *TL(s)* – memory list of non-optimal/last optimal solution.
- Aspiration list *AL(s)*– list which will allow a Tabu move in list if results of a solution are better as compared to the current solution.

TS: TABU SEARCH

Two types of memory used to provide quality solution and multiple objectives:

- Explicit Memory – Directs search towards influential/ quality-based solution
  - *TL(s)* and *AL(s)*
  - *Candidate List CL(s)* – generates list of possible moves
- Attributive Memory – Long term memory (a.k.a FR-Memory)
  - Diversification
  - Recency of vertices, frequency of moves within each neighbourhood area
- Candidate List Strategies – Strategies which narrow the examination of solutions in the *CL(s)*

NOC Topology Generation

Simulated Annealing (SA): Min-cut Partitioning

- Limited to cost function
- Single objective
- Problems determining global optimums

Genetic Algorithms (GA): Chromosome Strings Generation

- Fitness function
- Random generations

Both GA and SA techniques invoke Pareto curve technique
i.e. almost SINGLE OBJECTIVE MAPPING

ILP: Single-objective Limitations

Experiments show it takes lot of time to converge

ANOC: Recursive Slicing ree

- Heuristic with low complexity
- Limited to small design spaces (not suitable for MPSoCs)

TS & Topology Generation

- Supports **multiple** objectives by memory functions
  Not limited to cost function
- Base solutions are **priori information**
- Able to keep track of **optimal/non-optimal** solutions
  - Less computation to find **global optimum solutions**
  - Shorter runtimes
- Candidate list strategies – narrow down the solution space
  - Search for solutions that fit various constraints

Limitation

Solved during topology synthesis

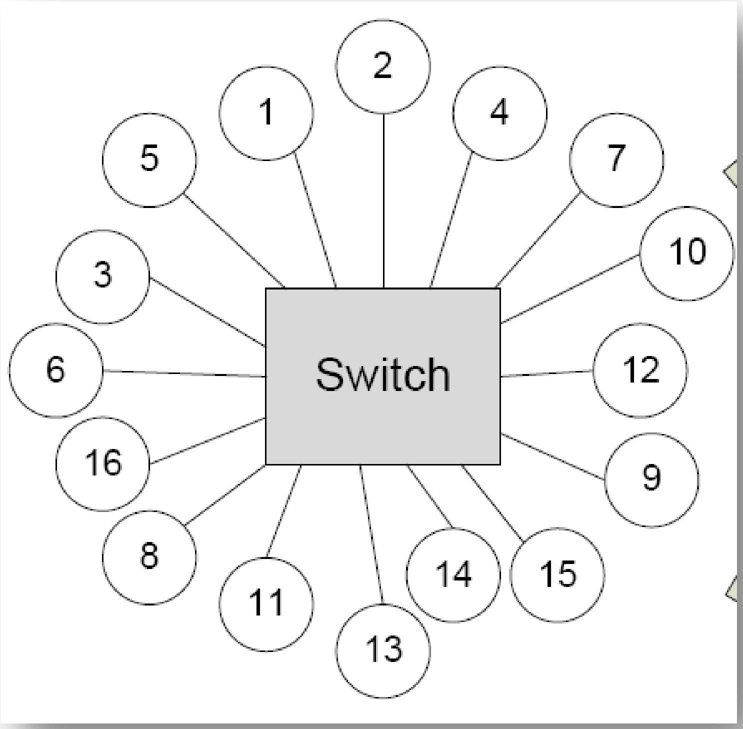
TS TOPOLOGY GENERATION

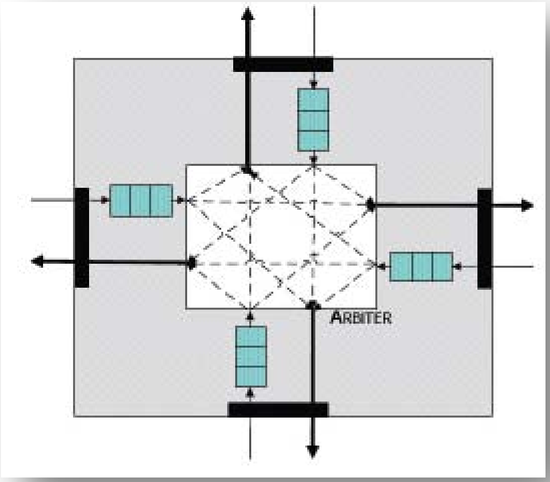
Algorithm 2 Tabu Search Topology Generation Algorithm

```
1:  Generate initial topology solution N(s,f,P)
2:  Evaluate N(s,f,P) for initial frequency f and power P
3:  TL(s) = {}  //ensure empty Tabu List
4:  WHILE performance and power constraints NOT met
    DO
5:     Identify s' = N(s)
6:     Move to the temporary solution s'
7:     Evaluate s' solution for f' and P' using Orion models
8:     ConstraintCheck(s', AL(s), TL(s))
9:     IF solution meets AL(s), TL(s), f', and P' constraints
10:         Create LQN models for sub-networks
11:         Invoke LQNS tool for performance analysis
12:         Place solution as last optimal TL(s) entry
13:         Update current solution, N(s,f,P) = s'
14:         Check for deadlock, contention, utilization
15:         VCInsertion(); (T_arb)
16:         Determine f' based on updated T_arb
17:         IF f' ≤ f_max THEN
18:             Run through system-level floorplanner
            ELSE
19:             Go to line 22
            END
20:         IF power/ perf constraints & router ports satisfied
21:             N(s,f,P) and EXIT
            END
       ELSE
22:         Place as a non-optimal TL(s) entry
23:         Refer to AL(s) to restore last optimal N(s)
       END
    END
```

- ***N(s, f, P)*** - current feasible NoC Topology solution *s* consuming power *P* at a frequency *f*
- ***N(s)*** - new possible solution *s'* within the neighbourhood set.
- ***TL(s)*** – Tabu List - contains non-optimal solutions
- ***AL(s)*** – *Aspiration List* - responsible for consulting *TL(s)* to ensure that *s'* is optimal and more desirable than the previous encountered solutions.

Initial NoC Topology

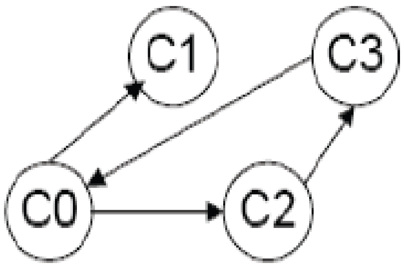




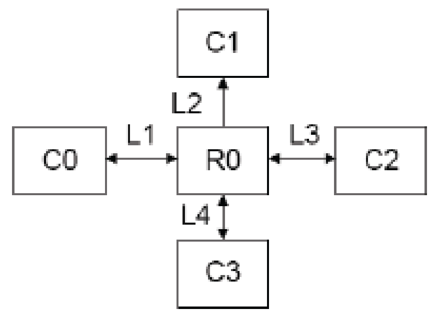
Initial topological solution

- Crossbar approach
- Poor solution
  - Power
  - Performance

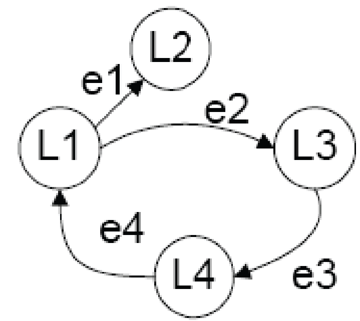
NoC Solution Evaluation



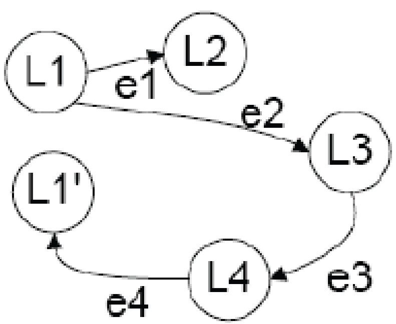
Core Graph

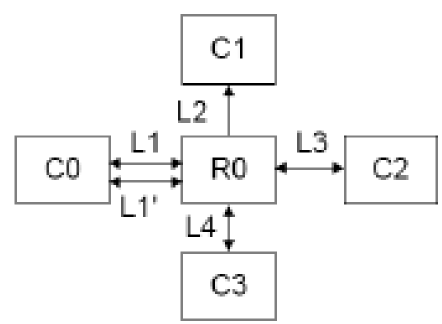


TSG



LRG





VC Insertion

NoC Solution Evaluation

$$T_{lat} = T_{pk} + \sum_{i=0}^{y} T_{arb} + \sum_{i=0}^{y} T_{blck} + T_{dpk}$$

$$\Gamma_{r,p} = \sum_{\forall s_x} \sum_{\forall d_x} \Lambda(sx, dx, r, p)$$

$$T_{arb_r} = \sum_{r=0}^{\forall R} \left[\sum_{p=0}^{\forall P,r} \Gamma_{r,p} \left(\frac{\theta(\Lambda(sx, dx, r, p), i, j)}{\sum_{p=0}^{\forall P,r} \Gamma_{r,p}} \right) T_{arb\_init} \right]$$

$$\theta(\Lambda(sx, dx, r, p), i, j) = \begin{cases} \mathrm{N}_{i \to j} & \text{if } \exists\, \Lambda \text{ from port } i \text{ to port } j \\ 0 & \text{otherwise} \end{cases}$$

Deadlock

$$E = \{ (l_i, l_j) \mid CNX(l_i, V) = l_j,\ v \in V \}$$

$$\sigma_{l_i} = \sum_{k=0}^{E_{l_i}} e_k \qquad Max^{\circ} = max(\sigma_{l_i}) \quad \forall\, i \in L$$

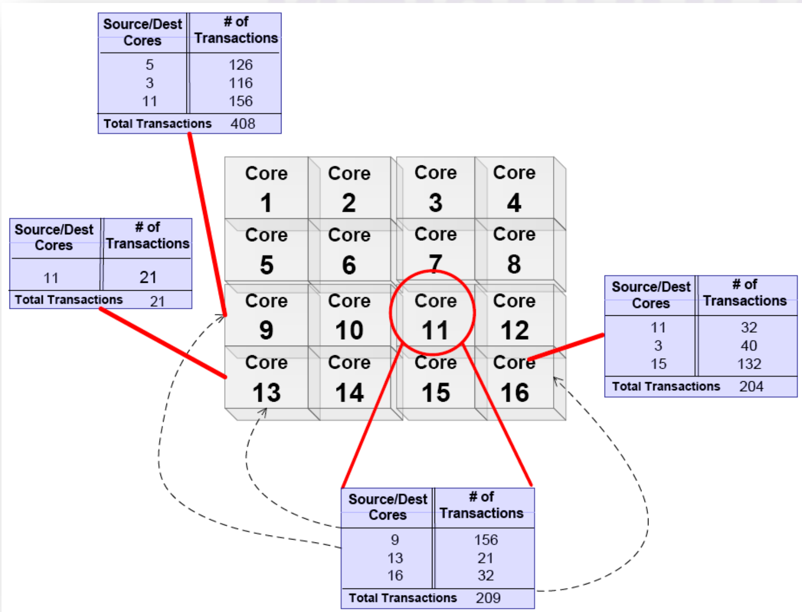
Contention

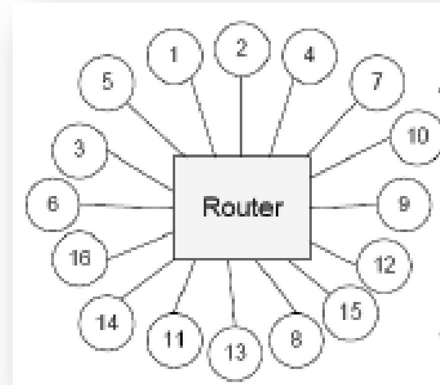
$$\mu_{r_{p_z}} = \max\big(\theta(\Lambda(sx, dx, r, p), i, j)\big)$$

$$\delta = \{ Max^{\circ} \cup \mu_{r_{p_z}} \cup l_i \}$$

Term	Definition
TDG	Topology Dot Graph is a directed graph G where *TDG* = *G(V,L)*. *V* is a set of vertices which represent the resources (cores and routers) and *L* is a set of edges denoting the links within the topology
CNX	Connection function, which connects the link l_i to the next communicating link l_j in order for s_x to reach d_x
LRG	Link Reliance Graph is a directed graph G where *LRG* = *G(L,E)*. *L* is now a set of vertices signifying links, and *E* is a set of edges denoting the pairs of links connected through *CNX*
s_x	A source core, or source vertex in the *TDG*
d_x	A destination core, which source s_x in the *TDG*
T_{arb}	Arbitration delay
T_{lat}	Overall latency delay
$T_{pk/dpk}$	Packing or de-packing delay of the NIs
$\Gamma_{r,p}$	Number of transactions expected in/out port p of router r
$\theta(\Lambda(sx, dx, r, p), i, j)$	Overall traffic flow from port i to j
ϕ	Deadlock communication link
σ_{l_i}	Degree of a vertex link l_i
Max°	Heavily utilized links of *LRG*
$Trans_{r_{y p_z}}$	Amount of transactions incurred by router r_y port p_z
$\mu_{r_{p_z}}$	Highly utilized ports of *TDG*
δ	An expected contention point within the topology
$\Omega[]$	1D array which holds all contention and deadlock points to be modeled with LQNs

Neighborhood Selection





$$MinTrans = min_n \left[V_n \left(\sum_{\forall s_x}^{N} \sum_{\forall d_x}^{N} N_{tr} \right) \right]$$

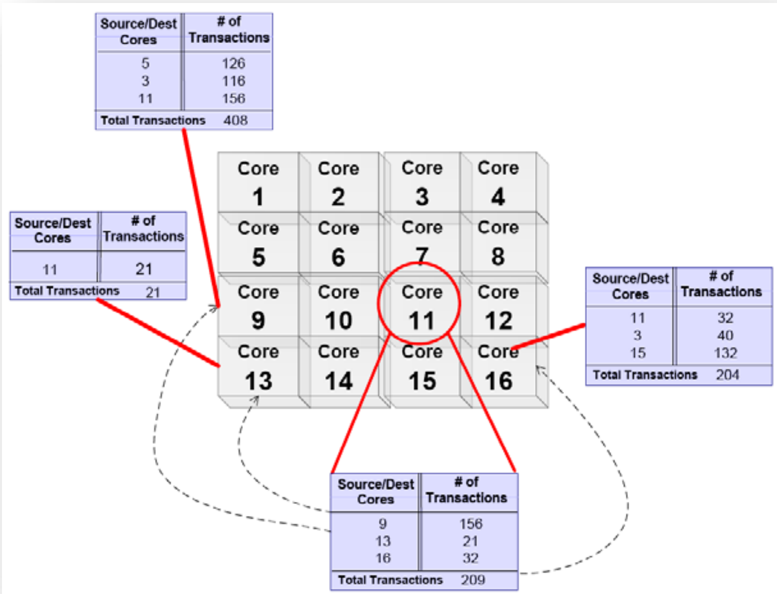
Successive Filter Strategy (SFS):

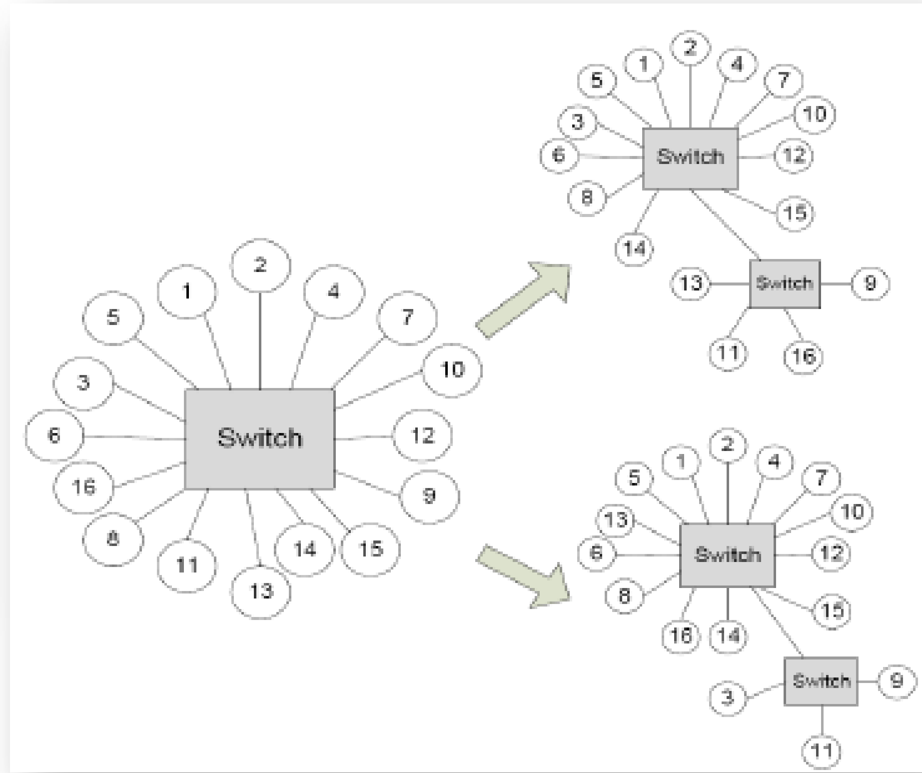
- *N* is the total number of vertices/cores in the core graph, where *n* = {1,2,…,*N*}
- $\pi \in \Pi$, where Π is a set of positions in the search space.
- $\pi(j)$ represent core *j* attaining the head candidate position.
- $\Omega(s)$ denote the set of possible moves that core *j* can have, when core *j* has occupied the position $\pi(j)$.
- *m* signify all possible combination of moves formed by the SFS.
- $\Omega(s)$ be divided into subsets $\Omega(1,s)$, $\Omega(2,s)$, … $\Omega(m,s)$, where $\Omega(1,s)$ denotes the 1st subset move in the possible set of total moves generated by the topology generator etc.

Given a vertex/core *n* :

- *N* is the total number of vertices/cores in the core graph, where *n* = {1,2,…,*N*}
- N_{tr} is the number of source, s_x, and/or destination, d_x, transactions that the vertex V_n is expected to incur.
- *X* is the total amount of sources or destinations for the respective core *n*, where *x* = {1,2,…,*X*}.
- $V_n(f)$ represents vertex *n* and its expected total number of transactions *f*.

Neighborhood Selection



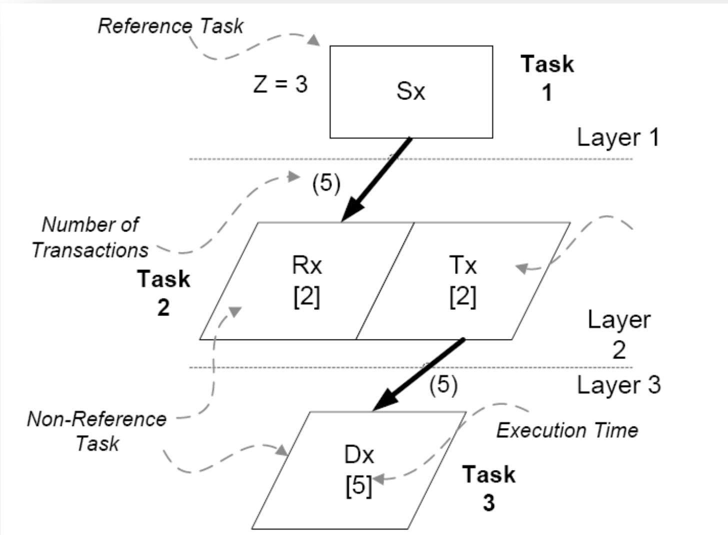


Successive Filter Strategy (SFS):

- *N* is the total number of vertices/cores in the core graph, where *n* = {1,2,…,*N*}
- $\pi \in \Pi$, where Π is a set of positions in the search space.
- $\pi(j)$ represent core *j* attaining the head candidate position.
- $\Omega(s)$ denote the set of possible moves that core *j* can have, when core *j* has occupied the position $\pi(j)$.
- *m* signify all possible combination of moves formed by the SFS.
- $\Omega(s)$ be divided into subsets $\Omega(1,s)$, $\Omega(2,s)$, … $\Omega(m,s)$, where $\Omega(1,s)$ denotes the 1st subset move in the possible set of total moves generated by the topology generator etc.

Layered Queuing Network

LQN and NoC Contetion in NoC



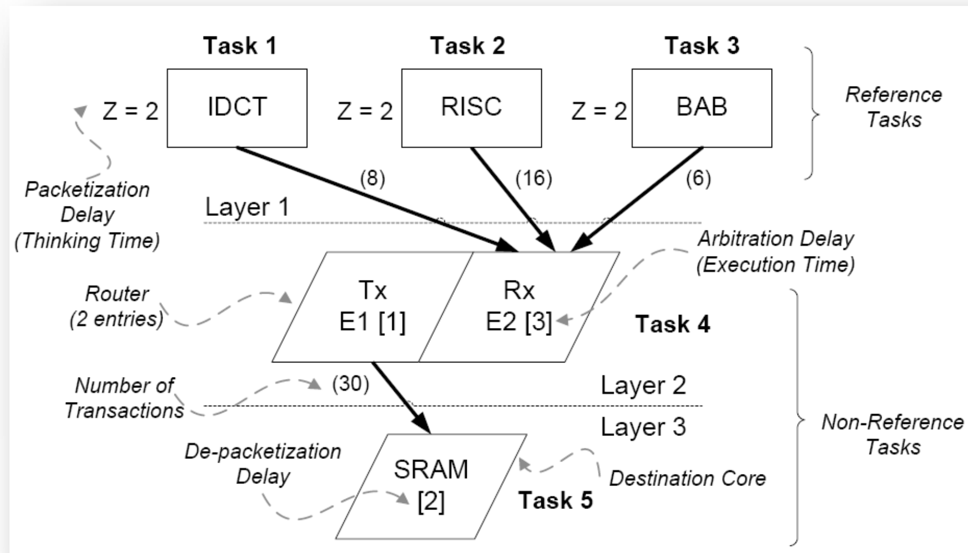


Table 1: LQN/Contention Model Conversion

LQN Element	Contention Model Element
Reference Task	Source Core
Non-Reference Task	Router/ Destination Core
Thinking Time (Z)	Packetization Delay
Execution Time of Task	Reference Task Packetization Delay Non-Reference Task De-packetization Delay Router Arbitration Delays
Number of Transactions	Number of packets sent from component *i* to *j*

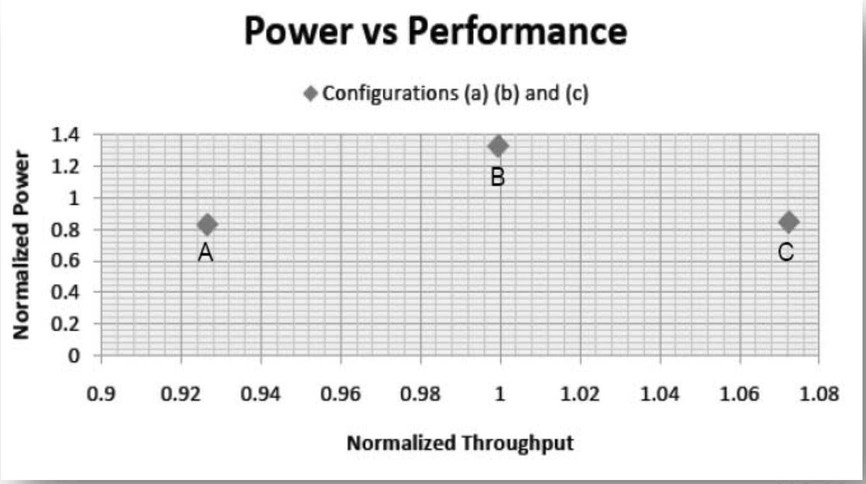
CONTENTION ALLEVIATION

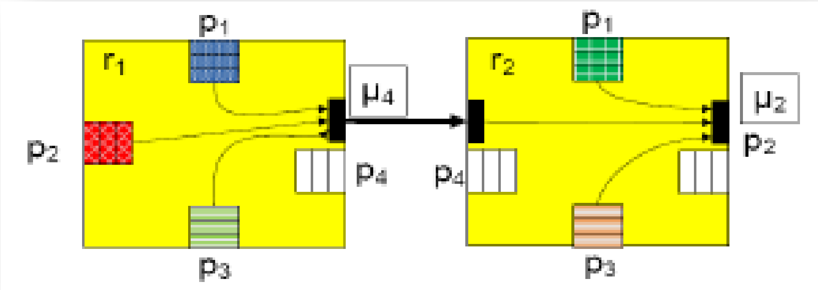
Different methods include:

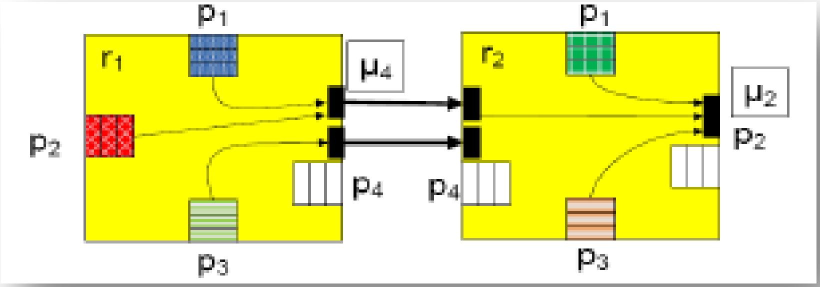
- Adaptive routing
- Task rescheduling
- Router buffer space allocation
- Virtual Channel (VC) insertion

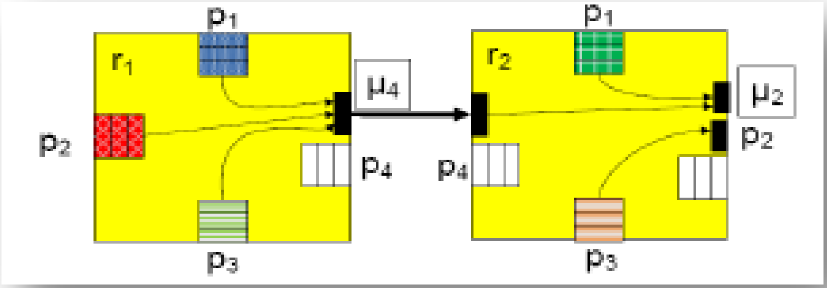
VC Insertion

- Power
- Performance

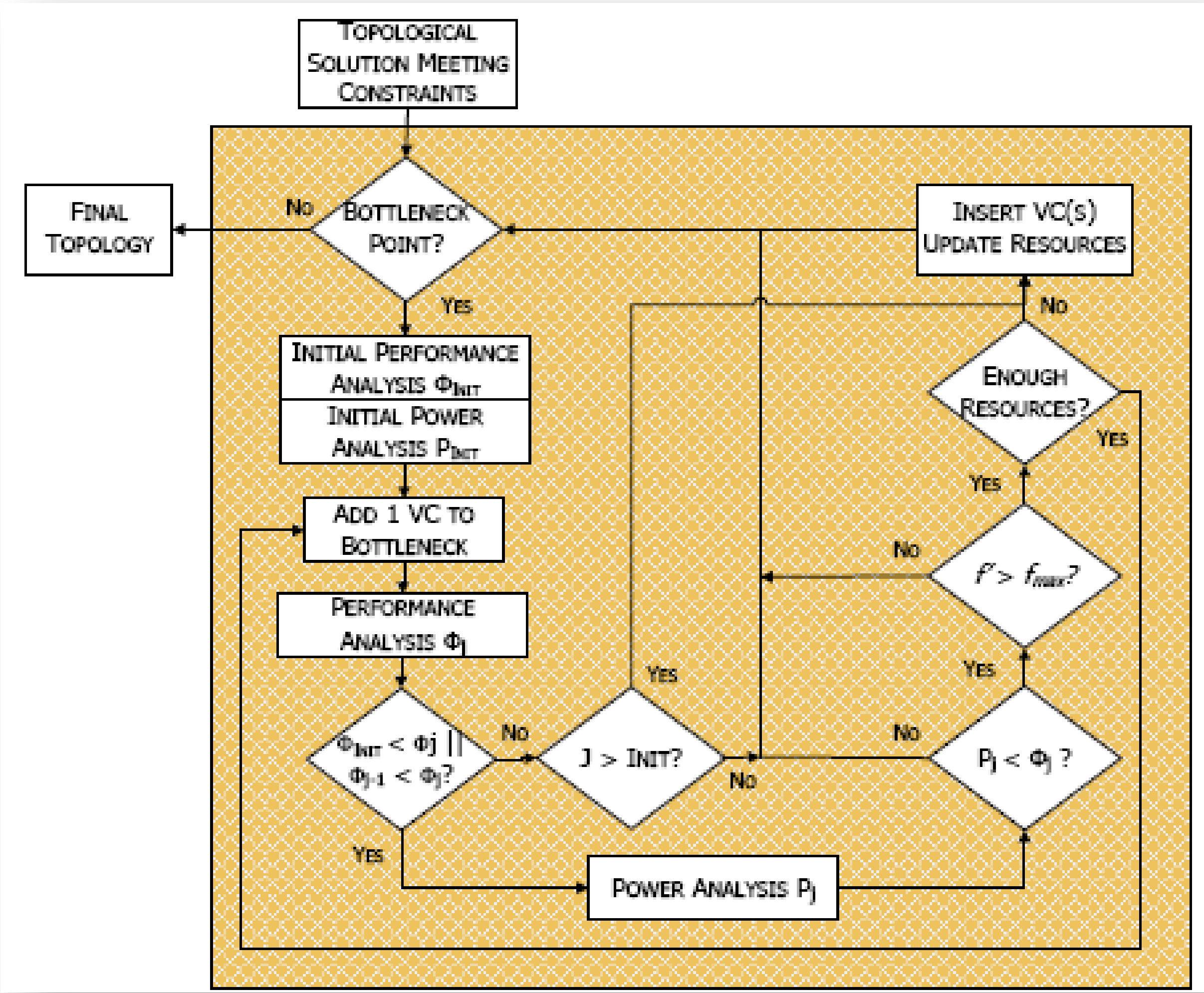








Contention Analysis



- The performance improvement $\boldsymbol{\Phi_j}$ is greater than the extra power dissipation $\boldsymbol{P_j}$ that the on-chip network will experience.
- There are enough VC resources for the insertion to take place.
- The new frequency of operation will not exceed the maximum allowable frequency.

Complexity Analysis

A Solution Space of ***N*** Cores

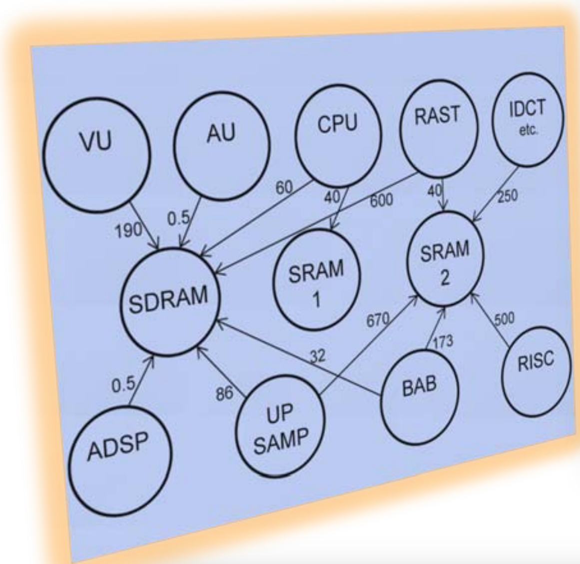
- For a move within the TS given the constraints imposed by the SFS yields ***N(N-1)/2*** moves = $\boldsymbol{O(N^2)}$
- Swaps needed to place the cores in the new topological arrangement results in a complexity of *O(1)*
- ***O(N)*** time is needed to evaluate the ***N*** cores.
- *For a total of* ***k*** iterations within the search
- Average *TL(s)* search time of ***i***
- Overall Complexity of the method can be expressed as:

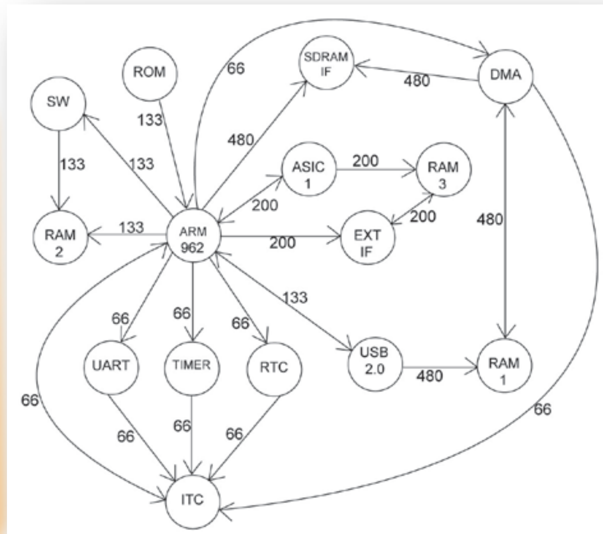
$$\boldsymbol{O(k*[N^2+N+ i])}$$

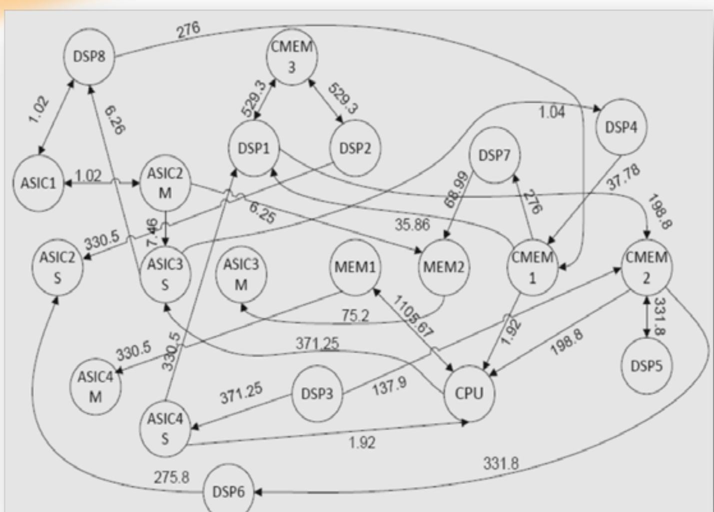
It can be further simplified as $O(N^2 + N)$, assuming ***k*** and ***i*** are much smaller than N

---- EXPERIMENTAL RESULTS ----

BENCHMARKS







BENCHMARKS

- MPEG4 Decoder – 12 cores
- Network Communication System (NCS) – 15 cores
- Multi-Window Display (MWD) – 15 cores
- Set Top Box – 25 cores
- Audio Video (A/V) – 21 cores
- D26_media – 26 cores
- Layer-3 Switch – 12 cores

TEST SET UP

- 1 GB RAM, 1.66 GHz Pentium based Linux System
- Network Interfaces = 0.2 mm^2
- Routers modeled as individual components in floorplanner (VCs also considered)
- Fully specified temporal pattern traffic generation
- Buffer sizes = 4 flits per port, max 6 ports/router and frequency of 2 GHz

Experimental Results

Method	Normalized Metric	Benchmark						
		B1	B2	B3	B4	B5	B6	B7
Tabu w/o VC	Power	0.7277	0.6999	0.7299	0.71822	0.6936	0.796	0.6895
	Throughput	0.612	1.120	0.871	1.453	0.393	0.967	1.113
Tabu w/ VC	Power	0.7813	0.7391	0.807	0.7827	0.7711	0.851	0.723
	Throughput	1.594	1.323	1.215	1.891	0.532	1.476	1.213
Application-Specific	Power	0.88737	0.84749	0.7299	0.77137	0.8292	--	0.9551
	Throughput	1.557	1.122	0.979	0.95	0.423	--	0.949
Mesh	Power	1.60356	1.71335	1.735	1.7276	1.6846	1.3527	1.708
	Throughput	0.0611	0.0921	0.1855	0.711	0.1457	0.557	0.721

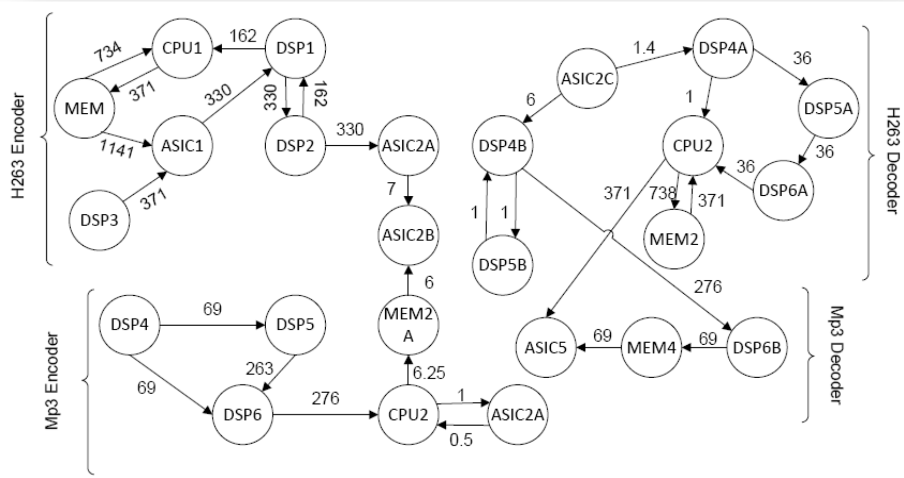
Benchmark	Graph ID	Cores	App-Specific
D26_media	B1	26	[53]
Set-Top Box	B2	25	[21]
MWD	B3	15	[6]
Audio/Video (A/V)	B4	21	[6]
Layer-3 Switch	B5	12	[48]
NCS1	B6	21	[21]
MPEG4 Decoder	B7	12	[6]

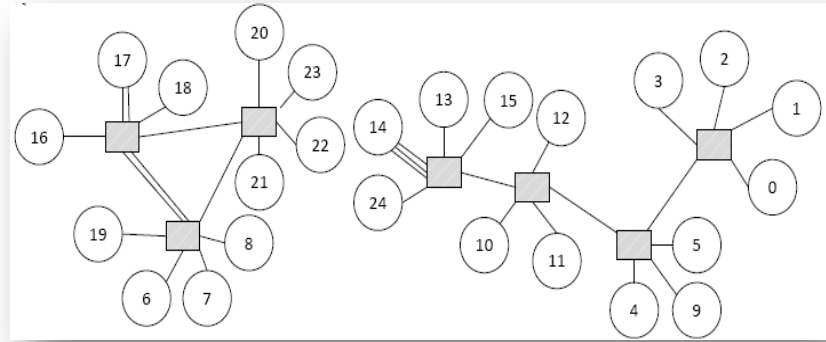
Benchmark	Area (mm^2)		
	Tabu	Mesh	App-Specific
B1	8.89	14.93	18.22
B2	6.41	13.4	6.54
B3	3.07	22.03	10.87
B4	11.19	32.35	22.92
B5	2.35	16.3	9.129
B6	4.00	32.35	--
B7	5.05	16.3	22.1

[6] Dimitriu V., and Khan G. N., "Throughput-Oriented NoC Topology Generation and Analysis for High Performance SoCs," *IEEE Trans. VLSI Sys.*, vol. 17, no. 10, pp. 1433-1446, 2009.
[21] Leary G., Chatha K., Srinivasa, Mehta, K., "Design of Network-on-Chip Architectures with a Genetic Algorithm-Based Technique," *IEEE Trans. VLSI Systems,* vol 17, no. 5, pp. 674-687, 2009.
[48] Dumitriu V., "Network-on-Chip Topology Generation and Analysis for Transaction-Based Systems-on-Chip", *MASc Thesis, Ryerson University*, 2008.
[53] Rahmati D., Murali S., Benini L., Angiolini F., De Micheli G., Sarbazi-Azad H., "A Method for Calculating Hard QoS Guarantees for Networks-on-Chip," *IEEE/ACM Int'l Conf. ICCAD*, pp.579-586, 2009.

- All topologies redesigned using 65 nm technology
- Comparable on-chip area overhead
- < 50 sec to generate and analyze topologies
- 33.1% less power with 33.6% increase in throughput (flits/sec)

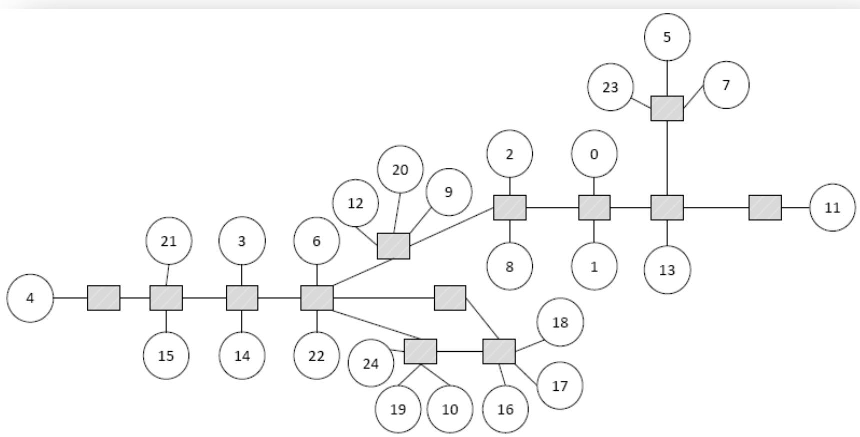
Experimental Results

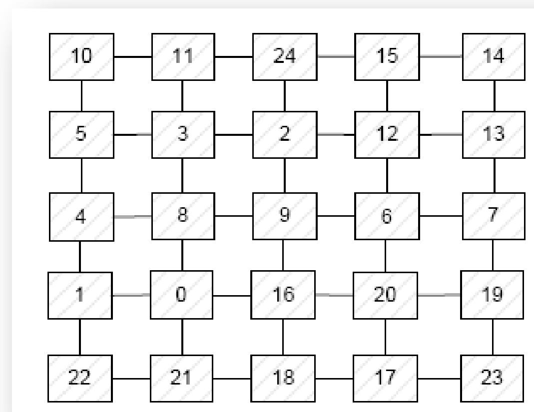




Tabu search based

Genetic Algorithm based
Topology [21]



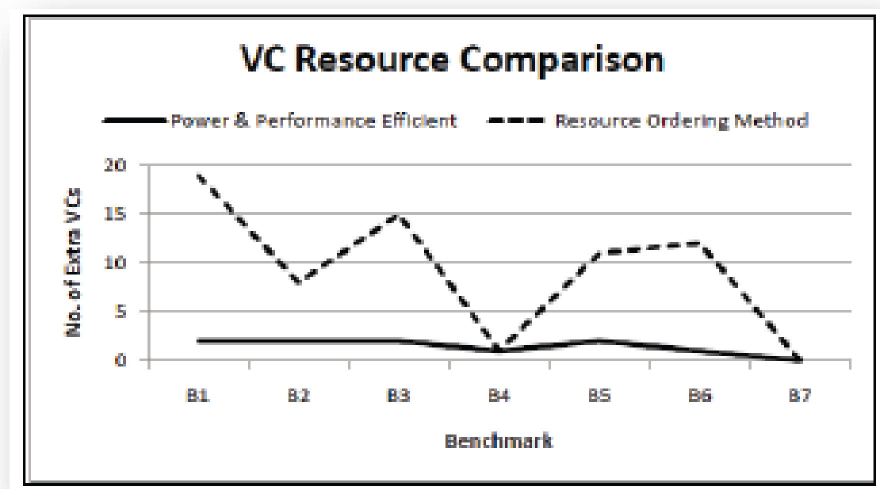


Mesh
based

Experimental Results

Topology	*Sub-Network*	*Performance (%)*	*Power (%)*	*Total VCs*
MPEG4	SDRAM2	0.3215	6.37	0
	SDRAM	11.5646 12.01	6.37 14.57	1
	RISC	0	7.79	0
	Actual Performance/Power Increase: 9.0 / 3.2 (%)			
NCS	ITC	12.24 19.10 20.89	5.87 10.46 20.79	2*
	ARM926	4.18	4.70	0*
	Router 0 to Router 3	5.877 9.04 12.65	4.71 8.47 16.54	2
	Actual Performance/Power Increase: 32.6 / 10.1 (%)			
Layer-3 Switch	DMA_1 Master	5.7 6.38	2.00 7.814	1*
	DMA_2 Master	36.157 36.851 37.023	6.390 18.269 37.54	2*
	MEM1	1.483	8.945	0
	Actual Performance/Power Increase: 24.8 / 7.8 (%)			
A/V	CPU	30.00 31.2	12.70 30.9	1
	CMEM2	0.6	12.7	0
	ASIC2 Slave	18.2 22.47 24.52	13.99 19.77 27.35	2
	CMEM3	19.48 20.95	12.70 20.78	1*
	Actual Performance/Power Increase: 30.14 / 10.01 (%)			
Set-Top Box	MEM1	18.9 32.7	12.1 59.1	1*
	ASIC1	49.0 52.3	26.16 59.1	1
	CPU2	21.99 26.98	12.1 26.16	1
	Actual Performance/Power Increase: 18.3 / 9.3 (%)			
D26_Media	DMA	23.4 25.1	13.22 24.9	1*
	ARM CPU	15.2 18.3	13.22 24.8	1
	Actual Performance/Power Increase: 16.7 / 13.68 (%)			

- Deadlock/ Contention Analyzer – accurate within a 19.8% error margin
- Many contention points also occurred at deadlock cycles
- Deadlock comparison to resource ordering technique of Dally: able to save 84.8% resource saving, in turn using 9.35 times less power with a slight performance improvement of 4%



Summary

- Proposed a methodology to produce efficient performance and power optimization in NoC design using Tabu Search optimization method
- New approach to contention relief using Layered Queuing Networks (LQN) and a power and performance tradeoff
- 33.1 % less power dissipation (on average) as compared to previous works
- Average performance improvement (including contention relief) of 33.6% (flits/cycle)
- Deadlock and Contention VC insertion technique allowing upto 84.8% resource savings with lower power


QUESTIONS