Energy and Reliability Oriented Mapping for Regular Networks-on-Chip

Cristinel Ababei
Electrical and Computer Eng.
North Dakota State University
Fargo, ND 58102, USA
cristinel.ababei@ndsu.edu

Hamed Sajjadi Kia
Electrical and Computer Eng.
North Dakota State University
Fargo, ND 58102, USA
hamed.sajjadikia@ndsu.edu

Om Prakash Yadav
Industrial and Manuf. Eng.
North Dakota State University
Fargo, ND 58102, USA
om.yadav@ndsu.edu

Jingcao Hu
Tabula, Inc.
3250 Olcott St.
Santa Clara, CA 95054, USA
jhu@tabula.com

ABSTRACT

We formulate the problem of energy consumption and reliability oriented application mapping on regular Network-on-Chip topologies. We propose a novel branch-and-bound based algorithm to solve this problem. Reliability is estimated by an efficient Monte Carlo algorithm based on the destruction spectrum of the network. Simulation results demonstrate that reliability can be improved without sacrificing much of energy consumption.

Categories and Subject Descriptors

B.8 [**Performance and Reliability**]: Reliability, Testing, and Fault-Tolerance

General Terms

Algorithms, Reliability

Keywords

Regular Network-on-Chip, mapping, energy, reliability

1. INTRODUCTION

Network-on-Chip (NoC) has emerged as a new communication infrastructure for multiprocessor Systems-on-Chip (MPSoCs) [1, 2]. Because increasingly adverse process variations and wearout mechanisms result in an increased number of transient, intermittent, and permanent errors, NoC reliability is a growing challenge in the design of MPSoCs [3].

Fault tolerance is one of the oldest resilience areas [4, 5] and relies on redundancy as a technique to compensate for the random failure of components, consequently improving reliability [6]. Previous work has focused mainly on processing elements as the computation units of multiprocessor SoCs. They addressed reliability by employing fault tolerant techniques based on error detection [7], failure prediction [8], and error masking [9]. While in the field of computer networks there has been a lot of work done on reliability, there have been only few recent attempts to estimate or indirectly optimize NoC reliability. The organization in layers of NoCs resembles the open system interconnection (OSI) protocol stack. Thus, it is convenient to discuss resilience techniques for NoCs in association with this organization. At the physical layer, wires may be subject to delay variations [10], while routers may be impacted by single event upsets (SEUs) [11]. The data-link layer can provide the functional and procedural means to detect and possibly correct errors that may occur in the physical layer, by employing error correcting codes (ECCs) [12, 13, 14], data encoding [15, 16], and redundancy based reconfiguration [17]. At the network layer, reliability of the routing algorithms can be enhanced by routing multiple copies of the same packet via multiple paths [18, 19, 20, 21, 22] or by adaptive re-routing [23]. The system software provides an abstraction of the underlying hardware platform, which can be leveraged by the application developer to effectively exploit the hardware's capabilities via reconfiguration [24]. Modeling of link failures due to process variations is studied in [25]. The reliability of custom switch architectures is analyzed in [26] while of various NoC topologies is analyzed in [27]. Several recent studies have investigated NoC fault tolerant architectures and techniques [28, 29, 30, 31, 32, 33, 34].

For a given set of resilience techniques, its effectiveness in achieving the desired system-level reliability must be evaluated and its associated costs such as system-level energy and performance costs must be quantified. However, evaluation of the reliability of NoC based SoCs is a challenging task because reliability is affected by numerous factors including aging/wearout mechanisms, process variations, dynamic power and thermal management, workload conditions, and system architecture and configuration. In the context of reliable computing systems, high-level reliability metrics have been proposed including reliability function $R(t)$, mean time to failure (MTTF), mean time to repair (MTTR), architectural vulnerability factor (AVF), availability, data integrity, etc. The reliability of NoCs has been evaluated using various metrics including fraction or probability of correctly delivered packets [21, 17, 32], percentage of lost packets or undetected errors [22], reliability factor [26], number of corrected errors [28, 29], minimum edge cutset [32], probability of correct operation [34], and path reliability [35].

NOCS'11, May 1-4, 2011 Pittsburgh, PA, USA

Previous work does not evaluate and address reliability of NoCs directly during the optimization process of various design steps[1]. Instead, reliability is achieved indirectly via fault tolerance and redundancy based techniques. Moreover, the majority of current resilience techniques are concentrated on individual levels of the system stack. Because such single-level approaches are based on redundancy, they impose high overheads in power, area, and performance. A different approach is to design solutions that span multiple abstraction layers. This is the key idea of our reliability optimization solution, which is a collaborative approach of multiobjective mapping (design tool level) and adaptive routing (network layer). To this end, our contribution lies in (1) We formulate for the first time the problem of mapping for regular NoCs with energy consumption and reliability as direct objectives. We propose a simple reliability model, which relates the path diversity determined by the shape of the bounding box of a given $s-t$ communication pair to reliability (as $s-t$ connectivity). (2) We develop an efficient branch-and-bound algorithm to solve the mapping problem. This algorithm provides a tunable tradeoff between the objectives of energy consumption and reliability via the parameter α. This can be utilized to generate Pareto optimal curves. (3) We validate the proposed algorithm via simulation based on an efficient Monte Carlo reliability estimation technique.

2. BACKGROUND ON NETWORK RELIABILITY

In our discussion, for a given network $\mathbf{N}$, edges represent the unreliable components of the network. We assume that each edge can be in two mutually exclusive states, *up* and *down*: *down* means failure and *up* means normal functioning. An important network reliability parameter is the so called probability of terminal connectivity. If we refer to only a single source-destination pair $s-t$, then, we are interested in the probability of so called $s-t$ *connectivity*. In general, for an arbitrary number k of source-destination pairs, we will use the term probability of *source-terminal connectivity*.

Let us assume that the probability of each edge being *up* is p and of being *down* is $q = 1-p$. When each of the source-destination pairs is connected through at least a path of edges which are *up*, we say that the network $\mathbf{N}$ is *UP*; otherwise we say that the network is *DOWN*. The probability of the network being in the *UP* state is utilized as a direct measure for reliability. In other words, the static network reliability can be computed straightforwardly by the expression [36]:

$$R(\mathbf{N}) = P(\mathbf{N}\ is\ UP) = \sum_{i=1}^{r} P(S_i) \qquad (1)$$

where $S_i, i = 1, ..., r$ is a particular set of edges which are *up*, such that the network is *UP*.

In the context of NoCs, we are concerned with the connectivity between the sources and destinations of the communication pairs of the application. This connectivity must be realized via paths formed by edges in the *up* state. For simplicity, in this paper we consider routing paths which are *monotonic XY-YX* (i.e., paths are formed by traversing links only in the same direction along x and y for a given source-destination pair). For example, in Fig.1.a, there is a single source-destination pair $s-t$. Hence, we are interested in the probability of $s-t$ connectivity. The *UP* states of the network with four edges from Fig.1.a are $S_1 = \{e_1, e_2\}$, $S_2 = \{e_1, e_2, e_3\}$, $S_3 = \{e_1, e_2, e_4\}$, $S_4 = \{e_1, e_2, e_3, e_4\}$, $S_5 = \{e_3, e_4\}$, $S_6 = \{e_1, e_3, e_4\}$, and $S_7 = \{e_2, e_3, e_4\}$. These states have probabilities $P(S_1) = P(S_5) = p^2q^2$, $P(S_2) = P(S_3) = P(S_6) = P(S_7) = p^3q$, and $P(S_4) = p^4$. Therefore the static network reliability can be computed by equation 1:

$$R(\mathbf{N}) = \sum_{i=1}^{7} P(S_i) = 4p^3q + 2p^2q^2 + p^4 \qquad (2)$$

which, for let us say $p = 0.95$, equals 0.9904. This is higher than the reliability of the network in the upper part of Fig.1.a $R(\mathbf{N}) = p^2 = 0.9025$. The difference is due to the two paths, which connect s and t in the network from the bottom of Fig.1.a as opposed to only one path in the network from the top of Fig.1.a. It is important to note that in the context of NoCs, the consideration of the two existing monotonic XY-YX paths is possible under the assumption that the NoC architecture is equipped with minimal adaptivity. This adaptivity means the capability to detect and locate link failures and to update the routing tables such that the remaining healthy paths are utilized for routing packets. The simplicity of the adaptive routing will result in minimal area overhead. This will be discussed in more details in a later section.

The above discussion also applies to the networks in Fig.1.b and Fig.1.c. In general, it can be proven that the reliability of the networks at the bottom of these figures is higher than that of the networks at the top of these figures. Moreover, the reliability increases as the topology of the network is closer to a square shape. For example, the reliability of the network at the bottom left of Fig.1.c is higher than the reliability of the network at the bottom right of Fig.1.c, which in turn is higher than that of the network at the top of Fig.1.c. Note that the Manhattan distance (as number of

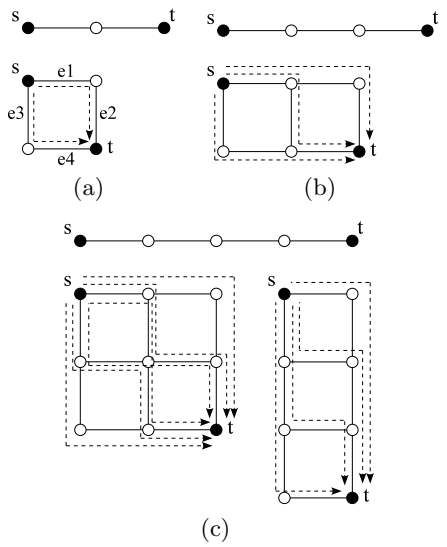

Figure 1: (a) Source destination pair with path length 2. (b) Path length 3. (c) Path length 4.

[1]The study in [32] only does topology selection and long-range link insertion to improve network reliability.

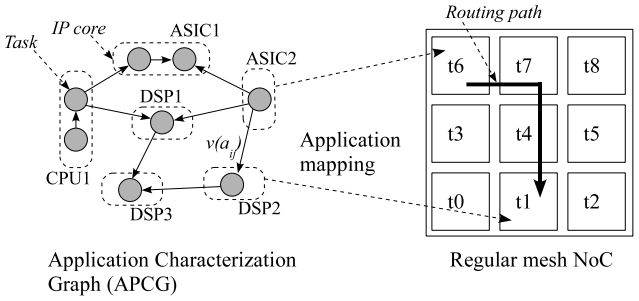

Figure 2: Illustration of the problem of mapping for regular mesh NoCs.

hops) between s and t is the same in each case.

It is precisely this relation between network reliability and the shape of the bounding box for a given source-destination pair, which we exploit to include reliability as an optimization goal in the mapping algorithm described in the next section.

3. ENERGY AND RELIABILITY ORIENTED MAPPING FOR REGULAR NOCS

3.1 The Problem of Mapping for Regular NoCs

In this paper, we assume a regular mesh NoC topology composed of a 2D regular array of identical tiles. Each tile contains a processing element (PE) and a router.

We utilize the terminology introduced in [37]. An application is given as an application characterization graph (APCG) $G(C, A)$, which is a directed graph, where each vertex $c_i \in C$ represents an IP core, and each directed arc $a_{ij} \in A$ represents the communication between source core c_i and destination core c_j. Each a_{ij} can be tagged with application-specific information (such as communication volume in bits $v(a_{ij})$ or communication rate) and specific design constraints (such as communication bandwidth, latency requirements, etc.). An APCG is derived from an application communication task graph (CTG) whose concurrent tasks have been already assigned and scheduled onto a list of selected IP cores. The *mapping problem then is to decide how to topologically place the selected set of IP cores onto the PEs (or tiles) of the NoC array such that the metrics of interest are optimized* (see Fig. 2).

3.2 Reliability Modeling

As we discussed in the previous section, for a given path length or Manhattan distance d between tiles t_i and t_j, the reliability is higher when the two tiles define a bounding box closer to a square shape. Therefore, one can use as a reliability cost:

$$r_{cost}(d) = r_{cost}^{max}(d-1) + (Const - w_{bbox} \cdot h_{bbox}) \qquad (3)$$

where w_{bbox} and h_{bbox} represent the width and height of the bounding box defined by the source and destination tiles. The first term $r_{cost}^{max}(d-1)$ is a biasing term that equals the maximum value of the reliability cost for a smaller distance $d-1$. This term is needed to keep $r_{cost}(d)$ proportional to $d = w_{bbox} + h_{bbox}$. If it is not used, then, longer Manhattan distances will be associated incorrectly with smaller reliability costs. In addition, the overall energy cost would increase too much. The constant $Const$ is specific to each given NoC mesh size and is selected as the minimum possible value such that the reliability cost $r_{cost}(d)$ remains positive for any possible bounding box within the given NoC mesh size. The overall network reliability cost can then be written as:

$$R_{cost} = \sum_{\forall\ pair\ (t_i,t_j)} r_{cost}(d_{ij}) \qquad (4)$$

The proposed reliability model basically relates the path diversity determined by the shape of the bounding box of a given $s-t$ communication pair to reliability (as $s-t$ connectivity). As mentioned earlier, this relationship is possible due to the fact that the NoC architecture is equipped with hardware support for minimal routing adaptivity. This is the key idea of the proposed reliability optimization, which is done collaboratively: during mapping (design tool level) and with adaptive routing (network layer) to directly address link failures.

3.3 Energy Modeling

To model the total communication energy consumption of the network, we adopt the bit energy metric proposed in [38]. Using this metric, the energy cost can be written as:

$$\begin{aligned} E_{cost} &= E_{Lbit} \sum_{a_{ij} \in A} v(a_{ij}) dist(t_i, t_j) + \\ &\quad E_{Rbit} \sum_{a_{ij} \in A} v(a_{ij})[dist(t_i, t_j) + 1] \end{aligned} \qquad (5)$$

where E_{Rbit} and E_{Lbit} represent the energy consumed when one bit of data is transported through one router and one physical link between two neighboring routers of the network. $v(a_{ij})$ is the communication volume between two cores of the APCG. $dist(t_i, t_j)$ represents the Manhattan distance between tiles t_i and t_j. Equation 5 states that the energy consumption is determined by the Manhattan distance between source and destination tiles and is proportional to the total communication volume.

3.4 The Problem of Energy and Reliability Oriented Mapping

By using the models introduced in the previous subsections, the problem of energy and reliability oriented mapping can be formulated as follows.

Prob. statement – Energy and Reliability Oriented Mapping
Given an APCG $G(C, A)$ and the regular mesh NoC topology formed by the routers R;
Find a mapping function $\Omega : C \rightarrow R$ which maps each IP core $c_i \in C$ in the APCG to a router (hence tile) $r_k \in R$ and a deterministic, deadlock-free, minimal routing function $\mathcal{R}()$;
Such that the following objective function is minimized:

$$Min: \quad \alpha R_{cost}^{'} + (1-\alpha) E_{cost}^{'} \qquad (6)$$

$$S.t. \quad \Omega(c_i) \neq \Omega(c_j), \forall c_i \neq c_j \in C \qquad (7)$$

$$B(l_k) \geq \sum_{a_{ij} \in A} v(a_{ij}) f(l_k, \mathcal{R}(\Omega(c_i), \Omega(c_j))), \forall l_k \qquad (8)$$

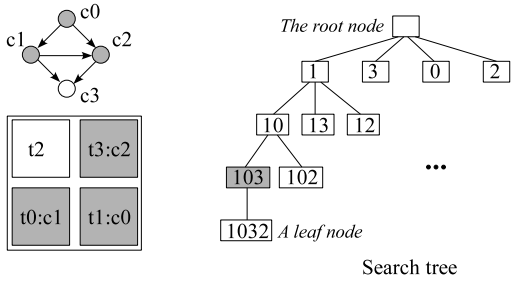

Figure 3: Illustration of the search tree. The internal node 103 **represents the partial mapping of cores** $c_0c_1c_2$ **to tiles** $t_1t_0t_3$**.**

where R'_{cost} and E'_{cost} are the normalized reliability and energy costs given by the equations 4 and 5. The normalization is necessary due to the potential large difference between the absolute values of the two cost components. It is done with respect with the worst case mapping, which one can achieve by placing individually the source and destination of communication pairs in opposite corners of the NoC. The parameter $\alpha \in [0, 1]$ (which can be specified by the user) is an objective weight, which is utilized to put more emphasis on reliability or energy objectives. $B(l_k)$ is the bandwidth of link l_k and the binary function f records if the link is utilized or not by the routing strategy $\mathcal{R}$:

$$f(l_k, path_{i,j}) = \begin{cases} 0, & \text{if } l_k \notin path_{i,j} \\ 1, & \text{if } l_k \in path_{i,j} \end{cases} \quad (9)$$

3.5 Branch-and-bound based Mapping

To solve the energy and reliability oriented mapping problem, we propose a branch-and-bound (BB) based algorithm [39]. In our implementation, we extend the mapping algorithm studied in [40]. The BB algorithm enumerates systematically all candidate solutions out of which many are discarded by using upper and lower estimated bounds of the cost function being optimized. The enumeration process utilizes a search tree data structure. The root node represents the state when no IP core is mapped yet. The tree is constructed during *branching* by creating new child nodes as IP cores are mapped sequentially on the remaining available tiles of the NoC architecture (see Fig. 3). Each internal node represents a partial solution while leaf nodes correspond to complete solutions. For each partial solution associated with an internal node, the *bounding* procedure computes upper and lower bounds for the minimum value of the cost function. The key idea of the BB algorithm is that if the lower bound for a currently created node is greater than the upper bound for some other node, then, the current node can be discarded from the search (its child nodes will not be created and explored). This step is called *pruning*.

The calculation of the upper bound cost (UBC) and lower bound cost (LBC) from [40] is extended to also include the reliability cost component of equation 4. The UBC of an internal node (which is a value that is no less than the minimum cost of its legal descendant leaf nodes) is computed by greedily mapping *temporarily* the remaining (not mapped yet) cores. During this greedy mapping, each of the remaining cores is assigned to a tile whose location is computed as the *weighted* center of gravity with respect to the already mapped cores. The weighting is done by the corresponding communication volumes. The LBC of an internal node (which is a value that is the lowest cost that its descendant leaf nodes can possibly have) is decomposed into three components.

$$LBC = LBC_{m,m} + LBC_{u,u} + LBC_{m,u} \quad (10)$$

where $LBC_{m,m}$ is the partial cost calculated exactly (by equation 6) for the cores which are already mapped. $LBC_{u,u}$ is the partial cost due to the communication among the unmapped cores. It is calculated by assuming the closest possible locations between unmapped cores (on available unoccupied tiles). $LBC_{m,u}$ represents the cost due to the communications between mapped and unmapped cores. It is calculated also by assuming the best possible mapping of the unmapped cores on unoccupied tiles with respect to the already mapped cores. The pseudocode of the energy and reliability oriented mapping algorithm is as follows.

Algorithm − Energy and Reliability oriented Mapping
1. Set α to desired value
2. Set $MinUBC = \infty$, $BestCost = \infty$
3. Create *root node*; **Insert** *root node* into priority queue PQ
4. Pop *node* from PQ
5. For each unoccupied tile t_i **do**
6. Generate child node n_c
7. Compute LBC and UBC as combined energy and reliability costs for node n_c
8. If $n_c.LBC > MinUBC$ **then** Go to *Step 10*
9.1. If n_c is leaf node **then**
  If $n_c.cost < BestCost$ **then**
    Record $BestCost = n_c.cost$
    Record n_c as *best mapping* so far
9.2. Else
  If $n_c.UBC < MinUBC$ **then**
    $MinUBC = n_c.UBC$
  Insert n_c into PQ
10. Repeat Steps 4-9 until PQ becomes empty
11. Report *best mapping*

---

4. RELIABILITY ESTIMATION

The reliability of each mapping solution must be evaluated. Note that while equation 4 provides an easy mechanism to incorporate reliability into the cost function of mapping, it does not represent a means to estimate the reliability per se. While for small networks direct reliability calculation by equation 1 is simple, for large networks with multiple source-destination pairs, reliability estimation becomes challenging. Therefore, we propose a novel technique to estimate network reliability based on the network *destruction spectrum*, which in turn is estimated by an efficient Monte Carlo algorithm [36]. This algorithm must be modified and adapted in the context of NoCs, where the network **N** represents the NoC architecture with the application cores mapped to it. In this representation, nodes of network **N** correspond to NoC routers while edges of **N** correspond to NoC links. The network status is *UP* when each communication pair $s_k - t_k, k = 1, ..., n$ can be connected via a valid routing path formed by edges (or links in the context of NoCs) which are *up*.

To describe the Monte Carlo algorithm we adopt the terminology introduced in [36]. For simplicity, we assume that all edges have the same lifetime $\tau_e \sim F_0(t)$. Note that the authors of [36] present also an algorithm for estimating the network lifetime when edges are characterized by arbitrary distributed lifetimes [41]. One can employ such an algorithm also in the NoC context as long as the individual edge lifetime distributions are known.

Then, the proposed technique to estimate NoC reliability for any $F_0(t)$ and t is as follows.

Algorithm – NetlifeSpectrum NoC

1. Set $N_1^k = N_2^k = ... = N_m^k = 0$ for each source destination pair $s_k - t_k, k = 1, ..., n$

2. Simulate random permutation π for the numbers $1, 2, ..., m$, $\pi = (i_1, i_2, ..., i_m)$

3. Assign weight r to the r-th edge $i_r, r = 1, ..., m$

4. Find the maximal-weight spanning tree for each source destination pair $s_k - t_k, k = 1, ..., n$

5. Find the minimal-weight edge in each of the spanning trees. Let these weights be $w_k^*, k = 1, ..., n$

6. Let $q_k = w_k^*$; **Put** $N_{q_k}^k = N_{q_k}^k + 1$, $k = 1, ..., n$

7. Repeat Steps 2-6 M times

8. For $r = 1, ..., m$ **put** $\hat{f}_r^k = N_r^k / M$, $k = 1, ..., n$

9. Estimate $F_{\mathbf{N}}^k(t)$ as $\hat{F}_{\mathbf{N}}^k(t) = \sum_{r=1}^{m} \hat{f}_r^k F_{(r)}(t)$, $k = 1, ..., n$

10. Let the maximum among these be $\hat{F}_{\mathbf{N}}^*(t) = \max_{k=1,...,n} \hat{F}_{\mathbf{N}}^k(t)$

11. Estimate $R_{\mathbf{N}}(t)$ as $\hat{R}_{\mathbf{N}}(t) = 1 - \hat{F}_{\mathbf{N}}^*(t)$

---

where m is the number of edges in the network (for example in a 4×4 regular mesh NoC architecture $m = 24$) and n is the number of communication pairs. $\hat{f}_r, r = 1, ..., m$ is the estimated D-spectrum. $F_{(r)}(t)$ is the cumulative distribution function (CDF) of the r-th order statistic of the edge lifetimes and is given by the following expression [36]:

$$F_{(r)}(t) = \sum_{j=r}^{m} \frac{m!}{j!(m-j)!} [F_0(t)]^j [1 - F_0(t)]^{m-j} \qquad (11)$$

In our implementation, the above algorithm is used to compute the system static reliability for different mapping solutions. The default values of M and of the probability of link failure are $M = 10,000$ and $F_0(t) = q = 0.01$ (hence reliability is $static - t$ is fixed or not present). This algorithm represents a callable routine inside the program which implements the proposed mapping algorithm.

Note that one could use this routine to replace equation 4 and thus estimate directly the reliability of a partial mapping solution. In this way the reliability cost component from equation 6 will be more accurate. However, the computational runtime of the mapping algorithm will increase because equation 6 must be evaluated thousands of times inside the branch-and-bound algorithm.

5. EXPERIMENTAL RESULTS

The proposed mapping algorithm was implemented as a computer program in C++. The program and testcases can be downloaded from [42]. Simulations were done on a Linux machine running on a 2.8 GHz Intel Quad processor with 2 GB memory. We investigate the performance of the proposed algorithm on the testcases shown in Table 1. The first testcase *mpeg4* is from [43] and the second testcase *telecom* is from [40]. The third and fourth testcases *ami25*, *ami49* were constructed from the classic MCNC benchmarks [44]. The initial connectivity between the modules was used to compute the communication volume in the communication task graph associated with each benchmark.

Table 1: Testcases and their characteristics.

Testcase	Num. of cores	APCG connectivity	Min/max comm. vol.
mpeg4	9	low	1/942
telecom	16	medium	11/71
ami25	25	high	1/4
ami49	49	high	1/14

5.1 Energy Only versus Combined Energy and Reliability Oriented Mapping

In the first set of experiments, for the BB based algorithm, we compare the case when both energy and reliability are included as objectives into the cost function against the case when only energy is optimized. The results are shown in Table 2. Energy is calculated by equation 5 and reliability is estimated using the *NetlifeSpectrum NoC* algorithm from the previous section. For example, the mapping solutions for testcase *telecom* are shown in Fig. 4.

Table 2: Branch-and-bound based algorithm.

Testcase	CPU runtime (s)	Objective: energy Energy (J)	Objective: energy Reliability (%)	Objective: energy & reliab. Energy (J)	Objective: energy & reliab. Reliability (%)
mpeg4	0.23	5.73	97.96	5.74	98.953
telecom	2.1	13.969	95.163	14.794	98.064
ami25	7.25	6.907	96.91	7.185	97.932
ami49	124.1	15.402	91.378	15.942	94.111

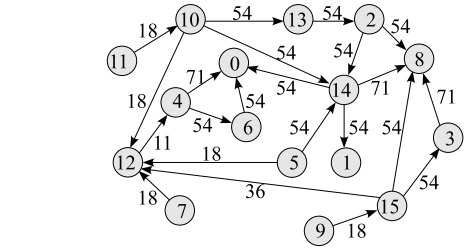

Figure 4: (a) Application characterization graph (APCG) of testcase *telecom*. (b) Mapping solutions with energy only as objective and with combined energy and reliability objective.

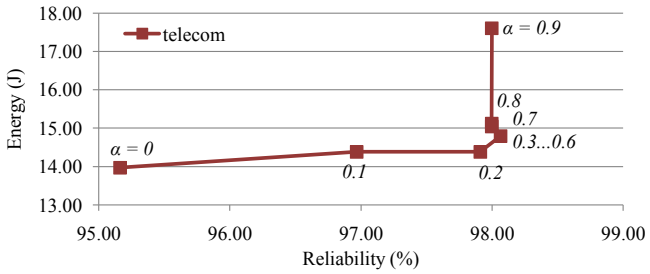

Figure 5: Pareto frontier for testcase *telecom*. Data points are displayed from left to right for α varied between 0 and 0.9 in increments of 0.1.

It can be observed that when reliability is also included into the cost function, the overall reliability of the mapped application is improved. However, the improvement in reliability is at the expense of increasing the energy consumption. Because, here we refer to the energy consumption of the NoC (which accounts for only up to 28% of the overall energy consumption of a multiprocessor SoC [45]), the energy consuption overhead is still within reasonable limits. The tradeoff between reliability and energy consumption can be controlled by α. The closer α is to 1, the higher the reliability improvement. For example, the Pareto frontier when α is varied between 0 and 0.9 in increments of 0.1 is shown in Fig. 5 for testcase *telecom*. The results from Table 2 are reported for $\alpha = 0.6$, which offers a good balance between reliability improvement and energy increase.

In our experiments, we noticed that reliability decreases as the connectivity of the APCG increases (for example testcases *ami25* and *ami49* have almost all cores communicating with each other). This can be explained in part by the fact that each link is utilized by more routing paths. Thus, a link failure has a bigger impact on the state of the network. Also, in highly connected APCGs there are more straight routing paths (as in the upper parts of Fig. 1.a-c), which increase the chances of the network to go *DOWN*. Another factor which affects overall reliability is the probability of link failure q. The results from Tables 2 were achieved for $q = 0.01$. We also investigated how q impacts the network reliability – the variation of reliability as a function of the link failure probability $q \in [0.0001, 0.5]$ is shown in Fig. 6.

Even though it may seem that reliability improvement of

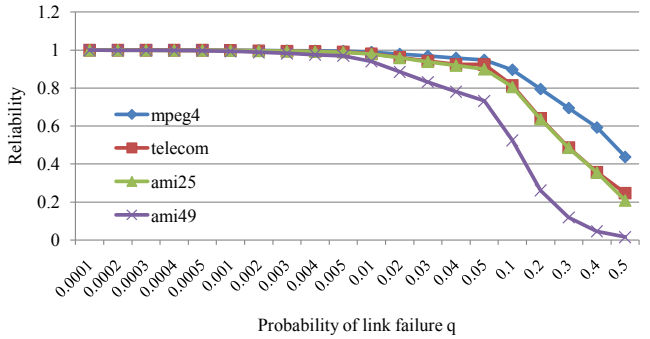

Figure 6: Variation of the static network reliability as a function of the probability of link failure.

only a few percentages is not much, this improvement cannot be directly compared as a percentage against the percentage of energy or performance overhead. Instead, the significance of such an improvement of reliability lies in its economic potential (what is the cost savings when for example 3% out of the total population of systems do not fail – i.e., when yield is improved by 3%), consequences (which can be severe in critical applications), and user satisfaction.

5.2 Comparison with Simulated Annealing based Mapping

In this section we compare the proposed branch-and-bound algorithm against simulated annealing (SA). The SA based algorithm has been fine-tuned by adjusting the cooling schedule and the number of *moves* per temperature. The basic move within the annealing process consists of swapping two cores. The results are shown in Table 3. It can be observed that while the energy and reliability results are similar to those in Table 2 the computational runtime of the SA algorithm is much longer.

Table 3: Simulated Annealing based algorithm.

		Objective: energy		Objective: energy & reliab.	
Testcase	CPU runtime (s)	Energy (J)	Reliability (%)	Energy (J)	Reliability (%)
mpeg4	8.17	5.865	96.951	6.321	98.953
telecom	21.47	13.722	95.208	14.877	98.927
ami25	306.2	6.861	96.054	6.815	97.929
ami49	7089.9	13.888	90.121	14.104	94.988

5.3 Discussion of Hardware Implementation

The collaborative nature of the proposed reliability optimization is based on NoC architectural support for cost-effective adaptive routing. That is, the NoC architecture integrates support to detect link failures and to update appropriately the routing tables such that all $s-t$ communication pairs of the APCG are connected via healthy monotonic XY-YX routing paths. The routing tables update is done periodically by rerouting controllers located in each router (Fig. 7.a), when the NoC is forced into *testing mode (TM)*. This is similar to the method proposed in [46]. The testing mode has two phases.

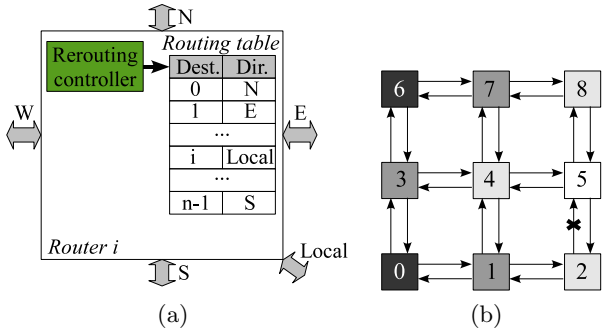

Figure 7: (a) The modified router integrates a rerouting controller, which updates the routing tables when link failures are detected. (b) Illustration of the broadcasting process: router 5 broadcasts, the information is received after one cycle by routers $2, 4, 8$, after two cycles by routers $1, 3, 7$, and after three cycles by routers $0, 6$.

In the first phase, each router with failed adjacent links broadcasts the state of its adjacent links[2]. When, for example router i broadcasts, all other routers will receive the information from router i after a number of cycles which depends on the distance from router i (Fig. 7.b). This process is done sequentially in that the router j which also has failed adjacent links will start broadcasting the state of its adjacent links only after the information from router i has reached all other routers.

In the second phase of the testing mode, the routing tables are updated by rerouting controllers. To minimize the hardware overhead, the routing tables update is done such that the next path closest to the previous monotonic XY-YX routing path is selected. For example, in Fig. 8 for the $s-t$ communication pair, path 2 is selected after failure 1 occurs and path 3 is selected after failure 2 occurs.

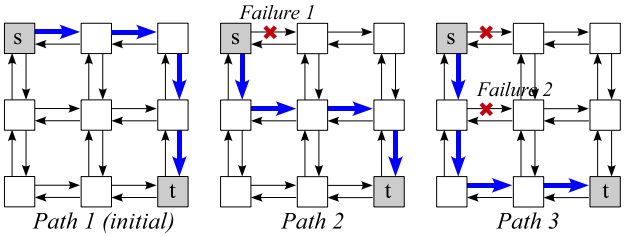

Figure 8: Illustration of path selection via routing tables update when link failures occur.

If at the end of the testing mode period all routing tables are updated successfully, the normal operation of the NoC is resumed. If there are too many link failures and at least one routing table cannot be updated successfully, then, an error signal is asserted to notify the software layer that the NoC cannot be utilized with the current application mapping.

The support for minimal adaptivity will incur hardware overhead, which translates in turn into area overhead. To estimate the area overhead due to routing controllers, we have implemented in Verilog and synthesized using Xilinx ISE [47] routers for NoC topologies with sizes 2×2, 3×3, and 4×4. The router architecture is similar to that in [1]. It has five input and five output ports, two virtual channels, round-robin based arbiters, input buffer size of ten, and packets with five flits for links with 32 bits wide. In all cases, the area overhead is less than 5%.

5.4 Discussion of Limitation

5.4.1 Deadlock and livelock

Routing algorithms for NoCs are desired to be free from deadlock and livelock [48]. The initial routing paths of the mapping solution achieved by the proposed algorithm are computed by XY or west-first and odd-even routing algorithms [40], which guarantee the deadlock-free property. However, after the first link failure occurs and paths are updated as monotonic XY-YX routing paths, the deadlock-free property cannot be guaranteed. On the other hand, monotonic XY-YX routing has the advantage of simple hardware implementation.

[2]The actual failure and diagnosis techniques are outside the scope of this paper. Such topics have been extensively studied in other works.

Deadlock could be guaranteed if the routing tables update were done by specialized adaptive routing algorithms [49]. Alternatively, the probability of deadlock could be minimized by employing techniques for routing paths recalculation [23] or partial deadlock removal [50]. However, the implementation of such adaptive routing algorithms and techniques is more challenging and will require higher area penalty. If such specialized routing algorithms are employed, the proposed mapping algorithm can still be utilized − only the reliability estimation procedure must be changed (i.e., computation of spanning trees) to account for the particular type of routing paths.

5.4.2 Bandwidth constraints

The initial mapping as found by the proposed mapping algorithm is under bandwidth constraints (see equation 8). Once link failures start to occur and routing tables get updated, the bandwidth constraints may be violated. We assume that in such situations, the whole NoC based SoC will be clocked at a lower frequency, thereby decreasing network performance. Operation with lower performance may still be desirable than no operation at all. Alternatively, this issue could be addressed by re-running the mapping algorithm or only its routing paths calculation as a software executable on the processing element of a central manager tile. In this way, the new routing paths could be found such that all bandwidth constraints are satisfied. However, there is no guarantee for that (failed links will ultimately demand decrease in performance) and the complexity of such an approach is higher.

5.5 Future Work

It would be interesting to study the combined effect of the proposed mapping algorithm and repairable links on reliability. If links are designed with redundant spare wires, then, several initial link failures could be addressed by link self-repair. After that, additional link failures could be addressed by the proposed routing tables update.

Moreover, one could formulate other interesting reliability problems. For example, one can ask for mapping with minimization of the power consumption under reliability constraints (e.g., reliability higher than 0.99). Also, one can ask to find the most reliable mapping within certain power consumption envelope.

As already discussed, one could replace equation 4 with calls to the *NetlifeSpectrum NoC* routine from Section 4. However, the computational runtime of the mapping algorithm will increase because equation 6 must be evaluated many times inside the branch-and-bound algorithm. An investigation of the tradeoff between the increase in runtime and the improvement in reliability is left for future work.

6. CONCLUSION

We proposed a multi-objective branch-and-bound based mapping algorithm for regular NoCs. The main advantages of the proposed algorithm are: (i) it provides a tunable tradeoff between the objectives of energy consumption and reliability via the parameter α; it was demonstrated that reliability can be improved without sacrificing much of energy consumption and (ii) it is very efficient compared to a simulated annealing based algorithm. Based on Verilog implementations of routers for several NoC sizes, it has been

found that the hardware overhead to support minimal adaptivity for routing tables update is less than 5%.

7. REFERENCES

[1] W.J. Dally and B.P. Towles, Principles and Practices of Interconnection Networks, Morgan Kaufmann, 2004.

[2] G. De Micheli and L. Benini, Networks on Chip, Morgan Kaufmann, 2006.

[3] J.D. Owens, W.J. Dally, R. Ho, D.N. Jayasimha, S.W. Keckler, and L.-S. Peh, "Research challenges for on-chip interconnection networks," *IEEE Micro*, 2007.

[4] J. von Neumann, "Probabilistic logic and the synthesis of reliable organizms from unreliable components," *Automata Studies*, C.E. Shannon and J. McCarthy Eds., Princeton Univ. Press, 1956.

[5] E. Moore and C.E. Shannon, "Reliable circuits using less reliable relays," *Journal of the Franklin Institute*, 1956.

[6] J.P.G. Sterbenz, D. Hutchison, E. Cetinkaya, A. Jabbar, J.P. Rohrer, M. Scholler, and P. Smith, "Resilience and survivability in communication networks: strategies, principles, and survey of disciplines," *Computer Networks*, 2010.

[7] S. Feng, S. Gupta, A. Ansari, and S. Mahlke, "Shoestring: probabilistic soft error reliability on the cheap," *ASPLOS*, 2010.

[8] Y. Li, Y.M. Kim, E. Mintarno, D.S. Gardner, and S. Mitra, "Overcoming early-life failure and aging for robust systems," *IEEE D&T of Computers*, 2009.

[9] M. Choudhury, V. Chandra, K. Mohanram, and R. Aitken, "TIMBER: time borrowing and error relaying for online timing error resilience," *ACM/IEEE DATE*, 2010.

[10] C. Hernandez, F. Silla, and J. Duato, "A methodology for the characterization of process variation in NoC links," *ACM/IEEE DATE*, 2010.

[11] A. Ejlali, B.M. Al-Hashimi, P. Rosinger, and S.G. Miremadi, "Joint consideration of fault-tolerance, energy-efficiency and performance in on-chip network," *ACM/IEEE DATE*, 2007.

[12] H. Zimmer and A. Jantsch, "A fault model notation and error-control scheme for switch-to-switch buses in a Network-on-Chip," *ISSS/CODES*, 2003.

[13] S.R. Sridhara and N.R. Shanbhag, "Coding for system-on-chip networks: a unified framework," *IEEE TVLSI*, 2005.

[14] S. Murali, T. Theocharides, N. Vijaykrishnan, M.J. Irwin, L. Benini, and G. De Micheli, "Analysis of error recovery schemes for Networks on Chips," *IEEE D&T of Computers*, 2005.

[15] D. Bertozzi, L. Benini, and G. De Micheli, "Error control schemes for on-chip communication links: the energy-reliability trade-off," *IEEE TCAD*, 2005.

[16] R. Singhal, G. Choi, and R. Mahapatra, "Information theoretic approach to address delay and reliability in long on-chip interconnects," *ACM/IEEE ICCAD*, 2006.

[17] T. Lehtonen, D. Wolpert, P. Liljeberg, J. Plosila, and P. Ampadu, "Self-adaptive system for addressing permanent errors in on-chip interconnects," *IEEE TVLSI*, 2010.

[18] M. Pirretti, G.M. Link, R.R. Brooks, N. Vijaykrishnan, M.T. Kandemir, M.J. Irwin "Fault tolerant algorithms for Network-On-Chip interconnect," *IEEE ISVLSI*, 2004.

[19] S. Manolache, P. Eles, Z. Peng, "Fault and energy-aware communication mapping with guaranteed latency for applications implemented on NoC," *ACM/IEEE DAC*, 2005.

[20] S. Murali, D. Atienza, L. Benini, G. De Micheli, "A multi-path routing strategy with guaranteed in-order packet delivery and fault-tolerance for networks on chip," *ACM/IEEE DATE*, 2006.

[21] A. Patooghy and S.G. Miremadi, "Complement routing: a methodology to design reliable routing algorithm for network on chips," *Microprocessors and Microsystems*, 2010.

[22] O. Ozturk, M.T. Kandemir, M.J. Irwin, S.H.K. Narayanan, "Compiler directed network-on-chip reliability enhancement for chip multiprocessors," *ACM LCTES*, 2010.

[23] A.D. Choudhury, G. Palermo, C. Silvano, and V. Zaccaria, "Yield enhancement by robust application-specific mapping on Network-on-Chips," *NocArc*, 2009.

[24] L. Zhang, Y. Yu, J. Dong, Y. Han, S. Ren, and X. Li, "Performance-asymmetry-aware topology virtualization for defect-tolerant NoC-based many-core processors," *ACM/IEEE DATE*, 2010.

[25] M. Mondal, X. Wu, A. Aziz, and Y. Massoud, "Reliability analysis for on-chip networks under RC interconnect delay variation," *Int. Conf. Nano-Networks and Workshops*, 2006.

[26] A. Dalirsani, M. Hosseinabady, and Z. Navabi, "An analytical model for reliability evaluation of NoC architectures," *IEEE IOLTS*, 2007.

[27] T. Lehtonen, P. Liljeberg, and J. Plosila, "Fault tolerance analysis of NoC architectures," *IEEE ISCAS*, 2007.

[28] J. Kim, D. Park, C. Nicopoulos, N. Vijaykrishnan, and C.R. Das, "Design and analysis of an NoC architecture from performance, reliability and energy perspective," *ACM ANCS*, 2005.

[29] D. Park, C.A. Nicopoulos, J. Kim, N. Vijaykrishnan, and C.R. Das, "Exploring fault-tolerant Network-on-Chip architectures," *IEEE/IFIP DSN*, 2006.

[30] F. Worm, P. Thiran, G. de Micheli, and P. Ienne, "Self-calibrating Networks-On-Chip," *IEEE ISCAS*, 2005.

[31] S. Shamshiri and K.-T. Cheng, "Yield and cost analysis of a reliable NoC," *IEEE VLSI Test Symp.*, 2009.

[32] H. Elmiligi, A.A. Morgan, M.W. El-Kharashi, and F. Gebali, "A reliability-aware design methodology for Networks-on-Chip applications," *IEEE Int. Conf. on Design and Technology of Integrated Systems in Nanoscale Era*, 2009.

[33] A. Patooghy, S.G. Miremadi, and M. Fazeli, "A low-overhead and reliable switch architecture for Network-on-Chips," *Integration, the VLSI Journal*, 2010.

[34] D. Fick, A. DeOrio, J. Hu, V. Bertacco, D. Blaauw, and D. Sylvester, "Vicis: a reliable network for unreliable silicon," *ACM/IEEE DAC*, 2009.

[35] M. Valinataj, S. Mohammadi, and S. Safari, "Reliability assessment of networks-on-chip based on analytical models," *Journal of Zhejiang University SCIENCE A*, 2009.

[36] Ilya B. Gertsbakh and Yoseph Shpungin, Models of Network Reliability: Analysis, Combinatorics, and Monte Carlo, CRC Press, 2009.

[37] R. Marculescu, U.Y. Ogras, L.-S. Peh, N.E. Jerger, and Y. Hoskote, "Outstanding research problems in NoC design: system, microarchitecture, and circuit perspectives," *IEEE TCAD*, 2009.

[38] T.T. Ye, L. Benini, and G. De Micheli, "Analysis of power consumption on switch fabrics in network routers," *ACM/IEEE DAC*, 2002.

[39] Sabih H. Gerez, Algorithms for VLSI Design Automation, Wiley, 1998.

[40] J. Hu and R. Marculescu, "Energy- and performance-aware mapping for regular NoC architectures," *IEEE TCAD*, 2005.

[41] I. Gertsbakh and Y. Shpungin, "Combinatorial approaches to Monte Carlo estimation of network lifetime distribution," *Applied Stochastic Models in Business and Industry*, vol. 20, no. 1, pp. 49-57, 2004.

[42] Reliable-NoC tool, 2010, `http://venus.ece.ndsu.nodak.edu/~cris/software.html`

[43] E.B. van der Tol and E.G.T. Jaspers, "Mapping of MPEG-4 decoding on a flexible architecture platform," *SPIE Media Processors*, 2002.

[44] MCNC Benchmarks, `http://vlsicad.eecs.umich.edu/BK/MCNCbench`

[45] Y. Hoskote, S. Vangal, A. Singh, N. Borkar, and S. Borkar, "A 5-GHz mesh interconnect for a teraflops processor," *IEEE Micro*, 2007.

[46] D. Fick, A. DeOrio, G. Chen, V. Bertacco, D. Sylvester, and D. Blaauw, "A highly resilient routing algorithm for fault-tolerant NoCs" *ACM/IEEE DATE*, 2009.

[47] ISE Design Suite, Xilinx, 2010, `http://www.xilinx.com/tools/webpack.htm`

[48] J. Wu, "A fault-tolerant and deadlock-free routing protocol in 2D meshes based on odd-even turn model," *IEEE Trans. on Computers*, 2003.

[49] R. Holsmark, M. Palesi, and S. Kumar, "Deadlock free routing algorithms for irregular mesh topology NoC systems with rectangular regions," *Journal of Systems Architecture - Embedded Systems Design*, 2008.

[50] C. Seiculescu, S. Murali, L. Benini, G. De Micheli, "A method to remove deadlocks in Networks-on-Chips with wormhole flow control," *ACM/IEEE DATE*, 2010.