Dynamic Programming-Based Lifetime Aware Adaptive Routing Algorithm for Network-on-Chip

Liang Wang*, Xiaohang Wang† and Terrence Mak*
*Department of Computer Science and Engineering
The Chinese University of Hong Kong, Shatin, N.T., Hong Kong
Email: {lwang, stmak}@cse.cuhk.edu.hk
†Guangzhou Institute of Advanced Technology, CAS, China
Email: xh.wang@gat.ac.cn

Abstract—Technology scaling leads to the reliability issue as a primary concern in Network-on-Chip (NoC) design. Due to the routing algorithms, some router may age much faster than others, which becomes a bottleneck for system lifetime. In this paper, a metric lifetime budget is associated with each router, indicating the maximum allowed workload for current period. Since the heterogeneity in router lifetime reliability has strong correlation to the routing algorithm, we define a problem to optimize the lifetime by routing flits along the path with maximum lifetime budgets. A dynamic programming-based lifetime aware routing algorithm is proposed based on the lifetime budget metric. The dynamic programming network approach is employed to solve this problem with linear complexity and low cost. Furthermore, we implement a low cost hardware unit to accelerate lifetime budget computation. The experimental results show that the lifetime aware routing has around 20%, 45%, 55% minimal MTTF improvement than XY routing, NoP routing, odd/even routing, respectively.

I. INTRODUCTION

Network-on-Chip (NoC) is emerging as an efficient communication infrastructure for connecting resources in many core system. However, with shrinking feature size and increasing transistor density, reliability issue becomes a primary concern for NoC design. The failure rate of electronic components increases 316% as the features size decreases 64% [1]. Along with shrinking feature size and integrated density, power density of chips increases exponentially, leading to overheat. High temperature also greatly shortens the lifetime of a device.

The reliability of NoC depends on the routers and links. For the sake of simplicity, it is assumed links is a part of routers in this paper. Router reliability has strong correlation with the routing algorithm. Fig. I shows a case study, which presents the distribution of router reliability of two different routing algorithms, XY and odd/even. The case study is evaluated in 8 × 8 2D mesh NoC. The traffic pattern is random and injection rate is 0.005 flits/cycle. The lifetime, measured in MTTF metric (mean time to failure), is normalized to the maximum one. For both routing algorithms, there is a heterogeneity observed among the routers. Especially for odd/even routing, the minimum MTTF of router is only less than 20% of the maximum one. It means that the minimum MTTF router is aging more than 5 times faster than the maximum MTTF router. The unbalanced lifetime distribution would become a bottleneck for the lifetime of system. Furthermore, the two distribution function differs in slope for XY and odd/even, indicating the correlation of router reliability and routing algorithms.

The above example indicates routing paths can be a control knob to optimize the router reliability. In this paper, we apply dynamic reliability management (DRM) to NoC and propose a lifetime aware routing to optimize the lifetime reliability of NoC routers. A metric lifetime budget is associated with each router, indicating the maximum allowed workload for current time. We define a problem to optimize the router lifetime by routing flits along the path with maximum lifetime budgets. The problem is represented with Bellman equation and solved by dynamic programming with linear time complexity. The key idea is to use lifetime budget as the cost for the dynamic programming. Moreover, a low cost hardware unit is also implemented to accelerate the lifetime budget computation at runtime.

The main contributions of this paper include:

1. We define a lifetime budget for each router, indicating the maximum allowed workload for current period.
2. We define a problem to optimize the lifetime by routing flits along the path with maximum lifetime budgets.
3. We propose a lifetime aware adaptive routing algorithm, which solves the problem through dynamic programming network approach with linear time complexity.

The remainder of the paper is organized as follows. Section 2 presents the related work. Section 3 discusses the DRM and defines the lifetime budget for a router. Section 4 presents the adaptive routing, including problem formulation and routing.
algorithms. Section 5 analyzes the experimental results and section 6 concludes this paper.

II. RELATED WORK

As the reliability is highly related to temperature, most prior works consider thermal issues, with the objectives to balance the temperature or to take temperature as a constraint. However, the thermal techniques neglect other factors on reliability, such as switch activity, operating frequency, etc. The lifetime could not be efficiently balanced. Dynamic reliability management (DRM), proposed in [2], [3], regards the lifetime as a source that could be consumed. Reliability management is mainly studied for single-core processor or multi-core processors through various solutions, such as task scheduling [4], frequency control [5], reliability monitoring and adaptation [6], etc.

Since NoC is becoming more important for many core system interconnection, reliability in NoC domain is attracting increasing attentions. Some works make attempt to improve the NoC reliability through microarchitecture design. A wear-resistant router microarchitecture is designed in [7] to improve reliability of routers. However, they did not consider the routing algorithm impacts on the router lifetime. Task mapping is another solution to improve NoC reliability. A compile-time task mapping algorithm is proposed in [8] to balance the MTTF of NoC. However, at runtime the task mapping only adjusts according to applications without considering the runtime operating conditions variation. The reliability of NoC can also be improved through routing algorithms. Bhardwaj et al. proposed an aging-aware adaptive routing algorithm for NoC [9], [10]. They introduced an aging model that defines stressed links and routers, in which the traffic of a router or link exceeds the upper limit called Traffic Threshold per Epoch (TTPe). However, the routing algorithm actually reduces the workloads of routers with high utilization, which may not exhibit the most aging effects. Different from their works, we directly apply reliability management to NoC, and define a lifetime budget for each router. A lifetime aware routing is proposed to balance the lifetime budgets of NoC routers at runtime.

III. LIFETIME BUDGET DEFINITION

For the long term reliability management of routers, we only consider wear-out related faults. The failure rate, a metric for lifetime reliability, keeps almost constant if the operating condition (e.g. constant current, temperature, frequency and voltage) keeps unchanged. The mean time to failure (MTTF) is inverse of failure rate when the operating conditions are constant. The MTTF under time-varying current density and temperature stresses is derived as [3]:

\[ T_f = A \left( \frac{\exp(-\frac{Q}{kT_f})}{kT_f} \right) \]  

(1)

where \( A \) is a constant related to the structure, \( j(t) \) is current, and \( T_f \) is temperature. The varying failure rate, also called lifetime consumption rate, is denoted as \( \lambda(t) = j(t) \left( \frac{\exp(-\frac{Q}{kT(t)})}{kT(t)} \right) \). The MTTF is the inverse of failure rate expectation value. The Eq. 1 is based on the assumption of Electromigration (EM) failure mechanism. We only consider EM because EM is the primary aging factor for interconnection, and the reliability of both routers and links are highly related to interconnection.

Based on Eq. 1, we also derive an equation as an approximation relationship between \( \lambda(t) \) and the routers workload. The current is \( j(t) = V_{dd} \times f \times p_i \), where \( p_i \) is switch activity. The voltage and frequency are assumed constant. In a router, switch activity is proportional to the incoming rate of a router because the incoming flits are assumed the only stimuli to the allocator. The failure rate can be expressed as follows:

\[ \lambda(t) \propto d(t) \left( \exp\left(\frac{-Q}{kT(t)}\right) \right) \]  

(2)

where \( d(t) \) is the flits incoming rate at time \( t \). The flits incoming rate is the number of flits passing through the router per unit time. The incoming rate also stands for the workload of a router. It is assumed that the ports of the incoming flits are not considered. The equation provides an approximated relationship between the lifetime and routers workloads.

Another equation derived in [3] is \( \int_0^t r(t)dt = C \), where \( C \) is a constant. In the equation, lifetime is modeled as a resource consumed over time. When a chip is designed, the expected lifetime is specified. \( r_{nominal} \) is derived from the specified expected lifetime, indicating the lifetime consumption rate under nominal conditions. \( r(t) \) is lifetime consumption rate under actual conditions. They define a lifetime deposit as \( \int_0^t (r_{nominal} - r(t))dt \). As suggested by [3], we define a lifetime budget for each router, denoted as

\[ LB(t) = \int_0^t (\lambda_{nominal} - \lambda(t))dt \]  

(3)

where \( \lambda_{nominal} \) is the inverse of expected MTTF, i.e., \( \lambda_{nominal} \cdot T_f = C \). If \( LB(t) > 0 \), the expected lifetime could satisfy the predefined constraint, and vice versa. The failure rate is related to operating conditions, i.e., temperature and workload, which are monitored periodically. Under discrete monitored conditions, the lifetime budget is expressed as

\[ LB(n) = \begin{cases} 0, & \text{if } n = 0, \\ LB(n-1) + \lambda_{nominal} - \lambda(n), & \text{Otherwise} \end{cases} \]  

(4)

where \( LB(n) \) is the lifetime budget at the \( n \)-th time interval. The lifetime budget indicates the maximum allowed failure rate for current the period.

From the perspective of flits, the selection paths determine to the workloads of the routers. Therefore the routing algorithm, closely related to the selection paths, plays an important role in the lifetime distribution of routers. We will propose a lifetime aware routing to balance the lifetime budgets of routers.

IV. LIFETIME AWARE ADAPTIVE ROUTING

A. Problem Definition

To balance the lifetime distribution, the lifetime aware adaptive routing aims to find a path with maximum lifetime
budget from designated path sets for each packet. The formulated problem is similar to the shortest path problem. Given a directed graph $G = (V, A)$ with $n = |V|$ nodes, $m = |A|$ edges, and a cost associated with each edge $u \rightarrow v \in A$, which is donated as $C_{u,v}$. There are two nodes $s, d \in V$, and $P_{s,d}$ is the set of minimal distance paths from $s$ to $d$. The cost of a path $p = \langle s = v_0, ..., d = v_k \rangle \in P_{s,d}$, from $s$ to $d$, is the sum of the costs of its constituent edges: $Cost(p) = \sum_{i=0}^{k-1} C_{v_i,v_{i+1}}$. We aim to find the path with the maximum cost, expressed in $V(s, d)$. The destination node $d$ initially receives a value $V(d, d) = 0$. Let $u$ be an intermediate node between $s$ and $d$, and $w$ is on the one of the minimal distance paths, i.e. $u \in p$. We have a constraint $V(s, d) \geq V(s, u) + C_{u,d}$. We can obtain the following linear programming:

$$
\text{maximize } \sum_{u \in V} V(s, d) \\
\text{subject to } V(s, d) \geq V(s, u) + C_{u,d} \\
V(d, d) = 0
$$

(5)

The above formulation yields the optimal path from any nodes $s$ in $V$ to the destination node $d$. With the nodes corresponding to the routers, the key idea of the adaptive routing is to use lifetime budget as the cost for the path, denoted as

$$
C_{r_i,r_{i+1}} = LB_i
$$

(6)

$LB_i$ is the lifetime budget of the $i$-th router $r_i$. The total lifetime budgets along the path $p = \langle s = v_0, ..., d = v_k \rangle$ is

$$
C_{s,d} = \sum_{i=0}^{k-1} LB_i
$$

(7)

With lifetime budget as the cost, the problem is to find a path with maximum lifetime budgets.

### B. Dynamic Programming-based Formulation

Mak et al. have proposed dynamic programming (DP) for adaptive routing, in which the shortest path problem is solved optimally [11]. The dynamic programming based adaptive routing has already been applied in congestion avoidance, fault tolerance [12], thermal management [13], etc.

The problem to find a path with maximum lifetime budgets can also be stated in the form of Bellman equations:

$$
V^{(k)}(s, d) = \max_{u \in V} \{ V^{(k-1)}(u, d) + C_{s,u} \}
$$

(8)

$V^{(k)}(s, d)$ is the cost from $s$ to $d$ at the $k$-th iteration. The cost $C_{s,u}$ is associated with lifetime budget. Initially, $u = d$ and $V^{(0)}(d, d) = 0$. Then Eq. 8 is solved recursively and the recursion is expanded from $s$ to $d$. After $k$ iterations, the optimal cost from $s$ to $d$ is $V^*(s, d)$, which is maximum among all minimal distance paths $P_{s,d}$. The optimal cost is represented as following equation:

$$
V^*(s, d) = \max_{\{v_0 = s, ..., v_k = d\} \in P_{s,d}} \left\{ \sum_{i=0}^{k-1} LB_i \right\}
$$

(9)

Compared with linear programming, the dynamic programming presents an opportunity for solving the problem using parallel architecture and can greatly improved the computation speed.

### C. Lifetime Aware Adaptive Routing Algorithm with Dynamic Programming Network

We propose a dynamic programming-based lifetime aware adaptive routing algorithm, which is outlined in Algorithm 1. This algorithm outputs the direction to be taken for current node $s$. First, according to the positions of local node and destination node, the available directions $D_s$ are restricted to the minimal distance paths to destination (line 1). If the local node is the destination, the optimal cost is 0 and the routing direction is local port. Given an available direction $j \in D_s$, the expected cost is computed by adding up the local cost $LB_j$ and the optimal cost from neighbor node $N(j)$ to $d$ (lines 6-8). The maximum cost is obtained by taking the maximum value from all $V_j(s, d)$, which are the costs taking direction $j$ for local node (line 9). Finally, the optimal direction $\mu(d)$ is obtained from the argument of the maximum operator (line 10). The dynamic programming-based adaptive algorithm outputs an optimal direction for each router. In the algorithm, the loop is realized in dynamic programming network. The optimal value for local node is also passed to the all neighbor nodes through dynamic programming network. The computational-delay complexity can be reduced to linear. The network converge time, i.e. the time to converge optimal solution, is proportional to the network diameter, which is the longest path in the network.

The dynamic programming network approach, introduced by Mak et al. [11], is composed of distributed computation units and links. The dynamic programming network is coupled with NoC. Each computation unit implements the DP unit equations e.g. shortest path calculations, and propagates the numerical solution to neighbor units. In addition, routing table is implemented in routers. Algorithm 1 presents the operations required for updating the routing directions using the DP unit. The routing table will be updated periodically by the DP unit. For each router, the temperature and flits incoming rate are

<table>
<thead>
<tr>
<th>Definitions</th>
<th>s: local node;</th>
</tr>
</thead>
<tbody>
<tr>
<td>$D_s$: set of directions to minimal distance paths;</td>
<td></td>
</tr>
<tr>
<td>$N(j)$: the neighbor node in the direction $j$;</td>
<td></td>
</tr>
<tr>
<td>$V_j(s, d)$: the cost of the path taking direction $j$ for $s$;</td>
<td></td>
</tr>
<tr>
<td>Input d: destination node;</td>
<td></td>
</tr>
<tr>
<td>$V^*(N(j), d)$: the optimal cost from $N(j)$ to $d$;</td>
<td></td>
</tr>
<tr>
<td>Output $\mu(d)$: the optimal routing direction to $d$;</td>
<td></td>
</tr>
<tr>
<td>$V^*(s, d)$: the optimal cost from $s$ to $d$.</td>
<td></td>
</tr>
</tbody>
</table>

Algorithm 1 Lifetime Aware Adaptive Routing

1. Calculate available direction $D_s$ according to positions of $s$ and $d$
2. if $s = d$ then
3. $V^*(s, d) = 0$
4. $\mu(d) = LOCAL$
5. else
6. for all directions $j \in D_s$ do
7. $V_j(s, d) = V^*(N(j), d) + LB_j$
8. end for
9. $V^*(s, d) = \max_{\{v_0 = s, ..., v_k = d\} \in P_{s,d}} \left\{ \sum_{i=0}^{k-1} LB_i \right\}$
10. $\mu(d) = \arg \max_{\{v_0 = s, ..., v_k = d\} \in P_{s,d}} V_j(s, d)$ → Update optimal routing directions
11. end if
also monitored periodically. Failure rate is computed through the lifetime budget computation unit, which is presented in the next part. According to the computed failure rate and nominal failure rate, the lifetime budget is updated. The lifetime budget values also propagated to the DP unit as the DP cost. The dynamic programming network quickly resolves the optimal solution and pass the control decisions to the router, then the routing table is updated.

D. Lifetime Budget Runtime Computation

The failure rate computation is an exponential function, not applicable for runtime computation. Similar to the methods proposed in [14], we use lookup tables that fit with Eq. 2 to pre-calculate failure rate. The runtime computation process is accelerated. To compute the lifetime budget at runtime, we design a hardware unit called lifetime budget computation unit (LBCU). The architecture of LBCU is presented in Fig. 2. The temperature rated part \( \frac{\exp(-\frac{t}{\tau})}{kT^1} \) is pre-computed and kept in a lookup table. Each entry is corresponding to a temperature range. Another potential problem is that it may require much area to multiply with the incoming flit rate. In stead of computing the multiplication at the end of each time interval, we compute the failure rate per cycle. As shown in Fig. 2, E, S, W, N, L are from 5 ports of a router, indicating whether there is a flit in current cycle. The failure rate per cycle is computed by multiplying the lookup table result with the sum of the ports. Because the maximum number to be multiplied is 5, the multiplication is achieved by shifting and addition in stead of multiplier. A counter is used to judge if it reaches the end of the time interval. At the end of the time interval, the lifetime budget is attained through addition. Despite of a little accuracy loss, the lifetime budget computation is accelerated at runtime while only some basic logic units are used. The time complexity for the computation decreases from \( O(\exp(n)/n) \) to \( O(n) \). The implementation of LBCU will be evaluated in terms of area in the section V-F.

![Fig. 2. Lifetime Budget Computation Unit](image)

TABLE I. BENCHMARKS

<table>
<thead>
<tr>
<th>PARSEC</th>
<th>streamcluster, swaptions, ferret, fluidanimate, blackscholes, freqmine, dedup, canneal, vips</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPLASH-2</td>
<td>barnes, raytrace</td>
</tr>
</tbody>
</table>

### V. EXPERIMENTAL RESULTS

A. Experimental Setup

Experiments are performed using Noxim simulator, which is an open source SystemC simulator for mesh-based NoC. In Noxim, the power of routers are evaluated using ORION 2.0 NoC power simulator [15]. To model the temperature, we adopt HotSpot thermal model [16]. The thermal configuration is the default configuration of HotSpot. The frequency of NoC is configured 1GHz. The time interval for temperature monitoring is 5096 clock cycles. The buffer depth is 10 flits and the packet size is 5 flits.

In the experiments, we compare the lifetime aware routing algorithm with XY routing, NoP routing and oddeven routing, respectively. The NoP routing algorithm, a congestion aware routing, is the oddeven turn model with neighbors-on-path (NoP) selection scheme; the oddeven routing is the oddeven turn model with random selection scheme. Besides, the evaluation is also performed over a suite of benchmarks: 9 benchmarks in PARSEC and 2 benchmarks in SPLASH-2. The benchmarks are listed in Tab. I. This experiment is performed in an in-house developed simulator [17]. The configuration for the simulator is listed in Tab. II.

B. MTTF Distribution

As a comparison with the case study mentioned in Section I, we evaluate the MTTF distribution of the lifetime aware routing and NoP routing. The injection rate is also 0.005 flits/cycle; the NoC size is \( 8 \times 8 \). The experimental results are presented in Fig. 3. For the lifetime aware routing, the

![Fig. 3. MTTF distribution of NoP routing and lifetime aware routing](image)
TABLE III. MINIMAL MTTF COMPARISON UNDER DIFFERENT ROUTING ALGORITHMS.

<table>
<thead>
<tr>
<th>NoC size</th>
<th>XY/hours</th>
<th>NoP/hours</th>
<th>oddeven/hours</th>
<th>lifetime/hours</th>
<th>Minimal MTTF improvement (lifetime vs.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>8x8</td>
<td>57490</td>
<td>453085</td>
<td>435682</td>
<td>683209</td>
<td>18.3%</td>
</tr>
<tr>
<td>10x10</td>
<td>321487</td>
<td>264756</td>
<td>253358</td>
<td>393711</td>
<td>22.4%</td>
</tr>
<tr>
<td>12x12</td>
<td>173618</td>
<td>144960</td>
<td>133637</td>
<td>203117</td>
<td>16.9%</td>
</tr>
</tbody>
</table>

MTTF distribution is more concentrated than others, namely, the MTTF is more evenly distributed.

C. Minimal MTTF Evaluation

We evaluate the minimal MTTF of all routers, expressed in \( \min \{MTTF_i\} \). The evaluation is under synthetic traffic. The traffic pattern is set random and the injection rate is set to 0.005 flits/cycle. The routing algorithms are also compared in different NoC sizes, \( 8 \times 8, 10 \times 10, 12 \times 12 \). The results are shown in Tab. III. In the table, the minimal MTTF value is evaluated. The evaluation metric is hour. It can be observed that the lifetime aware routing has around 20%, 45%, 55% minimal MTTF improvement than XY routing, NoP routing, oddeven routing, respectively. Additionally, the minimal MTTF also decreases dramatically with NoC size, because the workloads of routers increase with the area of NoC. The MTTF improvement against XY routing is relatively smaller as the XY routing also brings relatively less traffic for the routers in the central region.

We also evaluate the minimal MTTF with real benchmarks. The experimental results are demonstrated in Fig. 4. Here all the minimal MTTF values are normalized to the minimal MTTF under lifetime aware routing. The minimal MTTF under lifetime aware routing is highest, which is consistent with previous results. In addition, the minimal MTTF under different benchmarks varies a lot because the workloads are inherently unbalanced.

D. NoC MTTF Evaluation

We treat NoC as a whole and evaluate the MTTF of NoC. This is based on the assumption that NoC fails when a router fails. Therefore, the failure rate of NoC is the sum of all routers, denoted as \( \lambda_{NoC} = \sum_{i=1}^{N} \lambda_i \). The MTTF of NoC is calculated according to Eq. 1. We evaluate the MTTF of NoC under real benchmarks with 4 different routing algorithms. The results are presented in Fig. 5. From the results, we found that the lifetime aware routing leads to around 5% NoC MTTF improvement due to its better lifetime distribution. The NoC MTTF improvement is less because the NoC is treated as a whole and the total workloads of NoC almost keep the same despite different routing algorithms. However, the global MTTF could not efficiently reflect the reliability of routers. Unbalanced lifetime distribution would make some routers age much faster although the average lifetime still keeps the same. An example is illustrated in [18], showing that MTTF metric is not adequate for overall reliability specification. And the more balanced lifetime distribution has higher reliability for the whole system. Besides minimal MTTF, we also use the MTTF variance metric to show that lifetime more balanced. The results are shown in Fig. 6. In the figure, the MTTF variance is normalized for comparison. The lifetime aware routing algorithm exhibits the less variance, showing that the lifetime distribution is more balanced.

E. Flits Delay Comparison

To evaluate the impacts on the global average delay, the lifetime aware routing is also compared with the other three routing algorithms. The global average delay is evaluated with synthetic traffic. The buffer size is configured 10 flits. The comparison is under flits injection rate from 0.02 to 0.17 flits/cycle. The experimental results are shown in Fig. 7. The delay is measured in cycles. It can be observed that the saturated flit injection rate of the lifetime aware routing is 0.10.
flits/cycle, which is 0.02 less than odd/even routing, 0.03 less than NoP routing, and 0.04 less than XY routing. Therefore, when the injection rate is less than 0.10 flits/cycle, the lifetime aware routing has less impacts on the global average delay.

![Fig. 7. Global average delay comparison](image)

F. Evaluation of Lifetime Budget Computation Unit

We implement the LCBU in FPGA and compare the LCBU with router in terms of area. The lookup table of LCBU contains 64 entries to keep pre-computed values, which corresponds to different temperature ranges. It is assumed that temperature granularity is 1K. The size of each entry is 16 bits. The registers for lifetime budget and failure rate value are 32 bits. The buffer size is 4 flits; the flit size is 75 bits. The LBCU and router are implemented using a Xilinx Virtex-6 XC6VLX240T FPGA device and synthesized using the Xilinx ISE synthesis tools. The synthesis results of original router and the LBCU are 11782 slices and 283 slices. The LBCU leads to around 16% buffer less than NoP routing, which is 0.02 less than odd/even routing, 0.03 less than XY routing, when the injection rate is less than 0.10 flits/cycle, which is 0.02 less than odd/even routing, 0.03 less than NoP routing, and 0.04 less than XY routing. Therefore, when the injection rate is less than 0.10 flits/cycle, the lifetime aware routing has less impacts on the global average delay.

### VI. Conclusions

In this paper, we propose a dynamic programming-based lifetime aware routing algorithm for NoC reliability management. First, we define a lifetime budget metric for each router. With this metric, a problem is defined to optimize the lifetime by routing flits along the path with maximum lifetime budgets. We propose a lifetime aware routing algorithm using dynamic programming approach. Finally, the lifetime aware routing algorithms are evaluated in synthetic traffic and real benchmarks. The experimental results show that the lifetime aware routing can distribute the lifetime of router more evenly. The lifetime aware routing has around 20%, 45%, 55% minimal MTTF improvement than XY routing, NoP routing, odd/even routing, respectively.

### References