LIFE: a Limited Injection, Fully adaptivE, Recovery-Based Routing Algorithm

Fabrizio Petrini

International Computer Science Institute
1947 Center Street, Berkeley, CA 94704, USA
e-mail: petrini@icsi.berkeley.edu

José Duato, Pedro López and Juan-Miguel Martínez

DISCA, Universidad Politécnica de Valencia
Camino de Vera s/n, 46701 Valencia, Spain
e-mail: {jduato,plopez,jmmr}@gap.upv.es

Abstract

Networks using wormhole switching have traditionally relied upon deadlock avoidance strategies for the design of deadlock-free algorithms. The past few years have seen a rise in popularity of deadlock recovery strategies, that are based on the property that deadlocks are quite rare in practice and happen only at or beyond the network saturation point. In fact, recovery-based routing algorithms have a higher potential performance over the deadlock avoidance-based ones which allow less routing freedom.

In this paper we present a recovery-based fully adaptive routing algorithm, LIFE, which is based on an innovative injection policy that reduces the probability of deadlocks to negligible values, both with uniform and non-uniform traffic patterns.

The experimental results, conducted on a 8*-ary* 3*-cube with* 512 *nodes, show that it is possible to implement true fully adaptive routing using only two virtual channels. Also, LIFE outperforms state-of-the-art avoidance- and recovery-based algorithms of the same cost, both in terms of throughput and message latency under uniform traffic and provides stable throughput under non-uniform traffic patterns.*

1 Introduction

Deadlock recovery as a viable alternative to deadlock avoidance has recently gained consideration in the scientific community. It has been shown that, in a wormhole network using virtual channels, potential deadlock situations are quite rare except when the network is close to saturation [1]. More recent studies have also provided a theoretical background on this aspect and shown that also actual deadlocks are quite rare [2, 3]. Thus the hardware dedicated for deadlock prevention is not necessary most of the time. This consideration and the fact that wormhole routing on shallow networks (as tori) provides feedback at the source on the progress made by a given packet form the basis for the Compressionless Routing [1]. Deadlock is avoided by keeping track of whether the packet header has reached the destination. If the packet is blocked for more than a predetermined amount of time, the network interface of the source node tears down the partial packet and tries again later.

A more elegant deadlock recovery scheme which is based on progressive redirection of deadlocked packets rather than regressive abort-and-retry is Disha [4, 5]. Each router is provided with an additional special flit buffer to be used in case of deadlocks. These buffers form a separated logical network. On deadlock, detected with a local timeout, one of the packets in the cycle is switched to deadlock-free buffers and routed along this path until it reaches its destination. To ensure that the special buffer path is deadlock-free, only one packet is allowed to use it at any given time. Disha requires additional hardware in the form of a hardwired token path to guarantee mutual exclusion. To send a deadlocked packet on the escape buffers each router must first seize the circulating token. As recovery from cycles is sequential, the bandwidth available is small and can easily become a bottleneck for large networks or if deadlocks are temporarily clustered. The bandwidth limitation and the single point of failure have been addressed by a concurrent version that allows simultaneous recovery from multiple deadlock cycles [6]. This solution uses a mixed set of resources which includes both central buffers and virtual channels.

An in-depth formal characterization of deadlocks in wormhole interconnection networks has recently been proposed in [2] and [3]. These insightful works identify the network design parameters which significantly influence deadlock formation and model the frequency and the characteristics of deadlocks. They also suggest that routing algorithms designed primarily to maximize routing freedom tend to cause less message blocking than algorithms designed primarily to avoid deadlocks. Hence, if the network supports sufficient routing freedom to make deadlock highly improbable, deadlock recovery-based routing algorithms should outperform deadlock avoidance-based algorithms of the same cost, as resources are used more efficiently.

In this paper we present a fully adaptive recovery-based routing algorithm, called LIFE, that finalizes the theoretical results already obtained on deadlock recovery. Our algorithm is based on a selective injection policy that limits the access of new packets into the network by considering the status of the profitable directions. Irrespective previous studies in the literature, we allow the parallel injection of more messages at the same time. This is very important to achieve good throughput in the most common case, the uniform traffic. LIFE introduces an important innovation on the virtual channel allocation. While existing algorithm require the emptiness of both input and output buffers in a virtual channels, LIFE weakens this condition in the routing function by allowing the acquisition of a virtual channel even if it is partially occupied by another message. One of the main results of this paper is that it is possible to find an injection strategy that virtually eliminates the presence of deadlock with the most common communication benchmarks, even when the network is offered a saturation load. This strategy is very robust in practice, because it is neither sensitive to the size of the messages nor to the communication pattern.

The rest of this paper is organized as follows. Section 2 motivates our work, discussing some open problems in deadlock recovery and deadlock avoidance as well. Section 3 describes our algorithm, the directional injection function, the channel allocation strategy and some important aspects of the deadlock recovery. An overview of the simulation methodology and experimental results is provided in section 4 and some concluding remarks are given in section 5.

2 Motivation

Several aspects influence the presence of deadlocks in a wormhole network. In particular, as routing freedom is increased, the number of blocked messages and the likelihood of deadlock decrease exponentially [3]. The simplest manner of increasing the degree of freedom is to augment the number of virtual channels or the network dimension. In [2] it is shown that in a torus connected 16-ary 2-cube, using a fixed message size of 32 flits, a single injection channel and a *true fully adaptive routing* (TFAR) that allows unrestricted movement of packets in any profitable dimension using any available virtual channel, two virtual channels provide enough routing freedom to virtually eliminate all deadlock. With this configuration, no deadlocks occur with deeply saturated uniform traffic for a duration of $200,000$ cycles.

Unfortunately, this important result is weakened by some problems. First of all, it is achieved using a single injection channel. It has been shown that the presence of two or more injection channels has a strong effect on the network throughput [7]. For example, in Figure 1 we can see the impact of using more injection channels in a toroidal 8-ary 3-cube that uses an avoidance-based adaptive algorithm [8]. We can see that it is possible to get 82.2% of the network capacity, rather than 61%, using 3 injection channels (there is a diminishing return using more injection channels). But, if we increase the number of injection channels we loose the deadlock freedom of the network and most of the potential advantages of the recovery-based algorithm are lost.

The second problem is that the non-uniform traffic patterns are more deadlock-prone than the uniform one. For example, with the bit reversal or complement traffics, even using a single injection channel, it is possible to generate a deadlock in few hundred cycles, by properly choosing the packet and the network size. The situation is furtherly exacerbated if we use more injection channels. So avoidance-based adaptive algorithms can be more competitive if we succeed in using more than an injection channel and if face more efficiently the potential deadlock under non-uniform traffic.

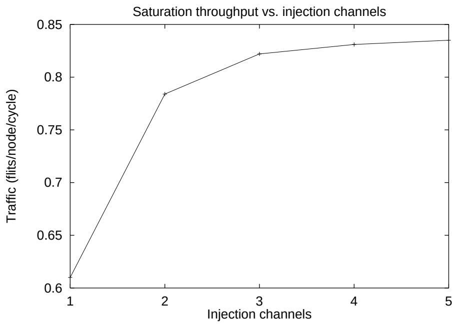

Figure 1: Saturation throughput vs number of injection channels in a 8-ary 3-cube under uniform traffic.

Vcs	Inj	Uni	Bit	Com	Shu
2	1	∞	1534	469	1220
2	2	519	472	228	560
3	1	∞	1820	1008	4600
3	2	2520	765	995	1730
3	3	771	760	990	949

Table 1: **Vcs** virtual channels, **Inj** injection or memory channels, **Uni**form, **Bit** Reversal, **Com**plement, **Shu**ffle. 8-ary 3-cube and 16-flit packets.

In Table 2 we try to quantify these problems by using several injection configurations and traffic patterns in a 8-ary 3-cube. We analyze a TFAR algorithm without any deadlock-avoidance or deadlock-recovery mechanism, varying the number of adaptive virtual channels per physical link and the number of injection channels. The results show the amount time, expressed in cycles, taken to get to a complete deadlock (if any) when the network is offered a saturation load. These results are the average of three runs with different initial seeds. An ∞ means that during each of these runs (the single run length is 100,000 cycles) the network did not deadlock.

We can see that, when we have a single injection channel, with either two or three adaptive virtual channels, there are no deadlocks with the uniform traffic. Things change in the presence of non-uniform patterns. The network usually blocks within 1000 cycles with either 2 or 3 adaptive channels. For example, with the complement traffic the network deadlocks very quickly (after few hundreds cycles), even if a complete halt is achieved after 468 and 1008 cycles.

When we increase the number of injection channels deadlocks become much more frequent, even under uniform traffic.

Another problem is the following. When the network is heavily loaded and is close to or beyond the saturation point (regardless of whether deadlock avoidance or recovery is used), messages block cyclically very quickly. As a consequence, those blocked messages will occupy channel bandwidth and decrease the network throughput considerably. Additionally, the latency of those messages will also increase considerably [9]. At least one of the following things must occur to mitigate this behavior: either escape resources must provide enough bandwidth to drain blocked messages or some mechanisms must prevent the build-up of cyclically blocked messages that would subsequently be drained. A solution to this problem could be to increase routing freedom by adding more virtual channels. However an excessive number of virtual channels could lead to a reduced clock frequency [10].

We will try to cope with all these problems in a unified framework that optimizes the resources and keeps the network below the saturation point under uniform and non-uniform traffic patterns.

3 The Routing Algorithm

The Limited Injection Fully adaptivE (LIFE for short) algorithm is a true fully adaptive routing algorithm based on deadlock recovery. It hinges on three main mechanisms: (1) a selective injection policy for those packets coming from the processing nodes, (2) an efficient channel allocation policy that provides an increased degree of freedom in the routing decision and (3) a deadlock detection mechanism.

These aspects will be discussed in more detail in the following sections.

3.1 Limited Injection Mechanisms

The fact that a wormhole network using adaptive routing is unstable at high traffic loads is not new. Dally and Aoki solved this problem in their dynamic adaptive routing algorithm by restricting entry packets to use only a subset of the available virtual channels in each physical link [11]. A similar technique has also been utilized in [12]. This solution has the problem that is not finely-tunable when we have few virtual channels. For example, in the original paper of Dally and Aoki the algorithms use 16 virtual channels.

Rather than considering the state of a single link, it is possible to gather information from all the available virtual channels. In [9] a message can be injected into the network only if the number of free virtual channels is more than a given threshold. This policy guarantees a better control on the network injection under uniform traffic, as shown in [13].

The main drawback of the previous policy is that different traffic patterns require different injection thresholds to achieve optimal performance. Usually, non-uniform traffic patterns need stricter injection configurations. This problem has been addressed in [14] taking into account the message destination distribution at any given time, using an history table that contains information for the last injected messages. The global number of available virtual channels available in the router is compared with a dynamically-updated threshold according to the information stored in the history table.

Based on these previous experiences, LIFE adopts a more selective injection policy that operates on two distinct levels, the injection threshold and the number of injection channels.

An array of injection thresholds is computed according to the number of directions available to the packet for minimal adaptive routing and the status of the virtual channels in the corresponding physical links. If $i \in \{1 \dots n\}$ is the number of dimensions where the source and destination are not aligned, the array of thresholds is defined as

$$Free(i) = \left\lceil \frac{Free}{n-i+1} \right\rceil, \ \ i \in \{1 \dots n\}, \tag{1}$$

where $Free$ is the basic threshold applied in case of maximum routing freedom. When we can use all directions in an n-dimensional cube we consider the threshold $Free(n) = Free$, and when source and destination are aligned in all but one dimension the threshold becomes $Free(1) = \lceil \frac{Free}{n} \rceil$. The appropriate threshold is compared with the number of available virtual channels in those physical channels and we can inject the packet only if there is enough room. We consider a virtual channel available if both the output and the corresponding input buffers are fully emptied, that is the virtual channel is treated as an atomic entity during the injection policy. For example, in a 3-dimensional cube equipped with 2 or more virtual channel per physical link with $Free = 3$ we require at least 3 empty channels to inject a message if there is no alignment, 2 free virtual channels if there is overlap in only one dimension and 1 free virtual channel otherwise. A larger threshold implies a stricter injection policy.

The second level of the LIFE injection policy operates on the number of injection channels. While parallel injection is necessary to achieve good throughput in the most common case, the uniform traffic, it overkills in the non-uniform ones. So we add this further limitation: we associate to each injection channel a register that contains the address of the packet currently injected, if any, and we allow the parallel injection of another packet only if the destination is different to the ones of the other packets in transmission in the router at the same time. This minor limitation has no effect on the uniform traffic when we use a reasonably large number of nodes, but it is able to control hot-spot and non-uniform traffic patterns, as will be shown in more detail in the experimental results.

This solution has a negligible complexity from the implementation point of view. For example, the Reliable Router associates to each virtual channel an independent routing unit [15]. Also, the computation of the injection function is not in the critical path, so it has no negative effect on the determination of the clock cycle of the routing chip.

3.2 True Fully Adaptive Routing

In true fully adaptive minimal routing a packet can take any of the available virtual channels in any of the profitable directions on a minimal path towards the destination. Deadlocks are possible since no other restrictions are enforced, and we rely on a deadlock detection mechanism and on spare resources to drain the deadlocked configurations.

In message routing, virtual channels along the path can be allocated according to one of the following approaches. With the *atomic allocation* a message can acquire a virtual channel only if both the output buffers in the current router and the corresponding input buffers in the neighboring router that logically compose the virtual channel are empty. With atomic allocation the header of message will always occupy the top of an input lane in case of blocking. The *non atomic allocation* allows the acquisition of a virtual channel even if it has not fully emptied, allowing the presence of more than a packet at a time in the virtual channel.

LIFE introduces an important innovation over existing examples of recovery-based algorithms: a virtual channel can be used by a packet even if it has not been fully emptied, that is LIFE adopts non atomic allocation. Irrespective Disha [4], Disha Concurrent [6] and the algorithms presented in [12, 13] and [14], we allow the presence of more than a packet in a virtual channel. Even if this represents a small increase in the degree of freedom (when virtual channels are allocated atomically we may delay the utilization of the buffers) we expect a sensible reduction of the probability of deadlock in particular when we use a reduced number of virtual channels.

3.3 Deadlock Detection Mechanism

The following lemma guarantees that at least one packet in each deadlocked configuration will be able to use the escape resources, even if the assumption on atomic channel allocation is removed.

Lemma 3.1 *In a deadlocked configuration that can be reached by routing packets starting from an empty network, there is at least one packet whose header flit is at the head of the queue containing it.*

Proof: Initially the network is empty, and nodes start injecting packets. When a deadlocked configuration is being formed, the first packet header that blocks must be at the head of a queue. Otherwise, it would have been blocked by the tail flits of a previously blocked packet. ■

Thus, in any deadlocked configuration at least a packet will occupy the top of an input lane, allowing the utilization of a standard deadlock detection mechanism based on timeouts. The recovery can be implemented using a set of central buffers, as in Disha [4]. LIFE utilizes a software-based technique that absorbs the potentially deadlocked message at the current node and later re-injects it for continued routing towards its destination.

A message is presumed to be deadlocked if all the alternative output channels that are requested by the message contain blocked messages. It is worth noting that LIFE uses all the virtual channels in each physical channel in the same way, and thus it is only necessary to monitor the activity in the physical channels. The mechanism is implemented as follows. A counter is associated with each output physical channel, it is incremented every clock cycle and it is reset when a flit is transmitted on that physical channel. Thus the counter indicates the number of cycles of since the last flit transmission across any virtual channel. The physical channel is considered blocked if this number is after a given threshold.

An in-depth description of the deadlock-detection and recovery mechanisms utilized in this paper can be found in [13].

4 Experimental Results

In this section we evaluate by simulation the behavior of LIFE, using the injection limitation mechanism proposed in this paper. The most important performance measures are latency (time required to deliver a message, including source queuing delay) and throughput (maximum traffic accepted by the network). Latency is measured in cycles and traffic in flits per node per cycle. Taking into account the size of current multicomputers and the studies about the optimal number of dimensions [16], we have evaluated the performance of the new algorithm on a 8-ary 3-cube network with 512 nodes.

4.1 Traffic patterns

In our model, each node generates packets independently according to an exponential distribution and the destinations are chosen according to the following traffic patterns: *Uniform*, *Complement*, *Bit reversal*, and *Perfect-shuffle*. These traffic patterns illustrate different features. The uniform one is a standard benchmark used in network routing studies. This generation pattern can be considered representative of well-balanced shared memory computations. The complement, bit reversal and perfect-shuffle traffics are important because they occur in many numerical computations [17]. The experimental results analyze the network performance with messages of 16 flits.

4.2 Performance Evaluation

In the performance analysis we take in consideration a partially adaptive routing algorithm with 2 virtual channels (PAR 2vc) and a fully adaptive algorithm with 3 virtual channels (FAR 3vc) [8], representative of the state of the art of deadlock-avoidance strategies. A similar fully adaptive algorithm has been recently adopted by the Cray T3E [18]. We also consider two versions of concurrent Disha [6] with 2 and 3 virtual channels. Based on the results shown in [6] and empirical evaluations, we respectively chose timeouts of 800 and 1000 cycles for these two versions. All these algorithms are compared with two versions of LIFE with 2 and 3 virtual channels using the same timeouts of Disha. PAR 2vc, FAR 3vc, Disha and LIFE algorithms use 3 injection channels. Also, these algorithms share the same channel selection policy which favors the less loaded link and, in case of equally loaded links, favors continuing routing in the current dimension over turning. Each input and output buffer has capacity for 4 flits.

We have experimentally found that, setting the injection threshold $Free = 3$, *is strict enough to eliminate deadlock with all the traffic patterns listed above, even if we push the network for very long periods of time at full load*. This threshold represents the best compromise for the injection policy of LIFE: looser thresholds tend to generate deadlocks with non-uniform patterns (in particular with the bit reversal and the complement traffics), while stricter injection thresholds do not fully utilize the interconnection network. In particular, we have found that at network capacity (the offered load is one flit per cycle per node) none of these traffic patterns generate any deadlock (all packets are delivered at destination) for a period of $100,000$ clock cycles if we adopt this injection policy and the non-atomic channel allocation policy.

The interesting point is that this property is not sensitive to the packet size and the number of virtual channels: we get the same result using the same injection policy with 2 and 3 virtual channels and with 16-flit packets and longer packets (64-flits). The weakest ring of the chain is the bit reversal traffic, the most deadlock-prone communication pattern, in particular when we use 2 virtual channels. The other communication patterns allow looser injection configurations. Also, the injection policy becomes more effective when we use longer packets.

Using this injection policy and injection configuration, we compared the communication performance of LIFE with Disha concurrent and the deadlock avoidance-based algorithms. The results for packets of 16 flits are summarized in Figure 2. We also obtained similar results with 64-flit packets. These results are not included here for brevity.

In Figure 2 a) we can see that, with uniform traffic, both versions of LIFE provide better message latency at low and intermediate loads, thanks to the non-atomic virtual channel allocation policy. LIFE with 3 virtual channels outperforms the corresponding avoidance- and recovery-based algorithms with the same number of virtual channels (FAR 3vc and DISHA 3vc), both in terms of latency and throughput, reaching a considerable saturation point at 87% of the capacity, while the other algorithms stops at about 73% and 69%, respectively. Also, LIFE with 2 virtual channels provides a slightly better saturation throughput than DISHA 3vc. The sharp increase of the latency of LIFE 2vc when it is close to the saturation point is caused by the injection limitation mechanism, that detects the presence of congestion in the network and delays the entrance of new packets.

The behavior of the two variants of Disha is very similar with all the three non-uniform traffic patterns. The latency is almost stable until the saturation point. Above saturation the network throughput collapses because these patterns generate deadlocked configurations that must be drained by the limited resources dedicated to deadlock-recovery. The throughput in all these cases drops to 10%.

The selective injection limitation mechanism adopted by LIFE makes the network stable above saturation. This comes at a price, with the increase of the message latency at intermediate loads. The injection mechanism conservatively delays the transmission of new packets when it senses a potential congestion. We think that it is important to have a stable throughput because the recovery from a short period of congestion is typically much longer than the onset and the network performance can be difficult to predict in the presence of non-uniform or hot-spot traffic without a proper injection limitation mechanism.

It is worth noting that both versions of LIFE outperform FAR with 3 virtual channels. This can be explained noting that, on the average, LIFE provides more routing choices: for example, in FAR with 3 virtual channels we can choose between at most 4 possible alternatives (3 adaptive channels plus a single escape channel), while LIFE 2vc can provide an output set of 6 virtual channels. This number increases to 9 if we use 3 virtual channels per link. The only exception is the complement traffic, where the partially adaptive algorithm outperforms the fully adaptive ones. This phenomenon has been observed in [19] too. The complement is unusual since dimension order routing helps prevent conflicts in this traffic pattern.

5 Conclusion

In this paper we have presented LIFE, a limited injection, fully adaptive routing algorithm that is based on deadlock recovery rather than deadlock avoidance. A central point of LIFE is the limited injection mechanism. We have shown that it is possible to reduce the deadlocks to negligible values under the most common traffic patterns. Also, the injection mechanism is robust in terms of packet size and number of virtual channels.

The experimental results conducted on a 8-ary 3-cube have shown that, under uniform traffic, LIFE with 3 virtual channels outperforms Disha with the same number of virtual channels because the combined approach of using more injection channels and a selective injection mechanism provides better throughput. It also outperforms the avoidance-based, fully adaptive algorithm under most communication patterns, due to the increased degree of freedom offered during the message routing and the better usage of the resources. We have also shown that, by properly tuning the injection mechanism, it is possible to virtually eliminate deadlocks under uniform and non-uniform traffic patterns.

In contrast with Disha, LIFE provides stable throughput under non-uniform traffic patterns even when the network is deeply saturated. We have higher latency at low and intermediate loads in these cases, due to the selective injection mechanism that delays the incoming packets when it senses congestion in the network. But the network behavior is more robust because it can tolerate impulsive or sustained congestion offering a stable throughput.

Of particular interest are the results obtained with 2 virtual channels, that represent a very interesting trade-off between performance and router complexity. These results suggest that recovery-based algorithms are natural candidates

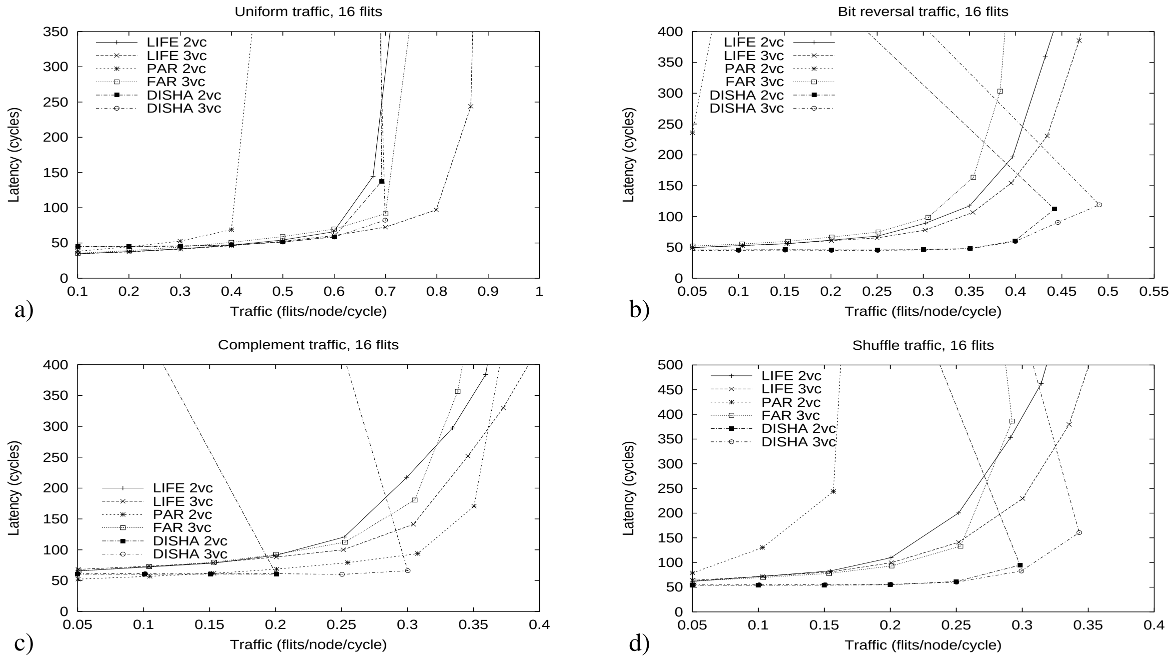

Figure 2: Average message latency vs. offered load for a 512-node 3-D torus for different routing algorithms and traffic patterns.

for the implementation of high performance switches for massively parallel architectures.

Acknowledgments

We thank all the reviewers, and in particular reviewer number three, for their insightful comments.

References

[1] J. H. Kim, Z. Liu, and A. A. Chien, "Compressionless Routing: A Framework for Adaptive Fault-tolerant Routing," *Computer Architecture News (Special Issue ISCA'21 Proceedings)*, vol. 22, pp. 289–300, April 1994.

[2] S. Warnakulasuriya and T. M. Pinkston, "Characterization of Deadlocks in Interconnection Networks," in *Proc. of the 11th Int. Parallel Processing Symp., IPPS'97*, pp. 80–86, April 1997.

[3] T. M. Pinkston and S. Warnakulasuriya, "On Deadlocks in Interconnection Networks," in *Proc. of the 24th Int. Symp. on Computer Architecture*, June 1997.

[4] Anjan K. V. and T. M. Pinston, "DISHA: A Deadlock Recovery Scheme for Fully Adaptive Routing," in *Proc. of The 9th Int. Parallel Processing Symp.*, April 1995.

[5] Anjan K. V. and T. M. Pinkston, "An Efficient, Fully Adaptive Deadlock Recovery Scheme: DISHA," in *Proc. of The 22nd Int. Symp. on Computer Architecture*, June 1995.

[6] Anjan K. V., T. M. Pinkston, and J. Duato, "Generalized Theory for Deadlock-Free Adaptive Wormhole Routing and its Application to DISHA Concurrent," in *Proc. of the 10th Int. Parallel Processing Symp.*, April 1996.

[7] V. S. Adve and M. K. Vernon, "Performance Analysis of Mesh Interconnection Networks with Deterministic Routing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 5, pp. 225–246, March 1994.

[8] J. Duato and P. López, "Performance Evaluation of Adaptive Routing Algorithms for k-ary n-cubes," in *First Int. Workshop, PCRCW'94* (K. Bolding and L. Snyder, eds.), vol. 853 of *LNCS*, pp. 45–59, May 1994.

[9] P. López and J. Duato, "Deadlock-Free Adaptive Routing Algorithms for the 3D Torus: Limitations and Solutions," in *Proc. of Parallel Architectures and Languages Europe 93*, June 1993.

[10] A. A. Chien, "A Cost and Speed Model for k-ary n-cube Wormhole Routers," in *Hot Inteconnects '93*, August 1993.

[11] W. J. Dally and H. Aoki, "Deadlock-Free Adaptive Routing in Multicomputer Networks Using Virtual Channels," *IEEE Transactions on Parallel and Distributed Systems*, vol. 4, pp. 466–475, April 1993.

[12] F. Petrini and M. Vanneschi, "Performance Analysis of Minimal Adaptive Wormhole Routing with Time-Dependent Deadlock Recovery," in *Proc. of the 11th Int. Parallel Processing Symp., IPPS'97*, pp. 589–595, April 1997.

[13] J. M. Martínez, P. López, J. Duato, and T. M. Pinkston, "Software-Based Deadlock Recovery Technique for True Fully Adaptive Routing in Wormhole Networks," in *Proc. of the Int. Conf. on Parallel Processing 1997, ICPP'97*, August 1997.

[14] P. López, J. M. Martínez, J. Duato, and F. Petrini, "On the Reduction of Deadlock Frequency by Limiting Message Injection in Wormhole Networks," in *Parallel Computing, Routing, and Communication Workshop, PCRCW'97*, June 1997.

[15] W. J. Dally, L. R. Dennison, D. Harris, K. Kan, and T. Xanthopoulos, "The Reliable Router: A Reliable and High-Performance Communication Substrate for Parallel Computers," in *First Int. Workshop, PCRCW'94* (K. Bolding and L. Snyder, eds.), vol. 853 of *LNCS*, pp. 241–255, May 1994.

[16] A. Agarwal, "Limits on Interconnection Network Performance," *IEEE Transactions on Parallel and Distributed Systems*, vol. 2, pp. 398–412, October 1991.

[17] F. T. Leighton, *Introduction to Parallel Algorithms and Architectures: Arrays, Trees, Hypercubes*. San Mateo, CA, USA: Morgan Kaufmann Publishers, 1992.

[18] S. L. Scott and G. M. Thorson, "The Cray T3E Network: Adaptive Routing in a High Performance 3D Torus," in *HOT Interconnects IV*, August 1996.

[19] M. L. Fulgham and L. Snyder, "A Comparison of Input and Output Driven Routers," in *Second Int. Euro-Par Conf., Volume I*, no. 1123 in LNCS, pp. 195–204, August 1996.