Source misrouting in King topologies

E. Stafford, C. Martinez, J. L. Bosque,
F. Vallejo, C. Camarero, B. Perez and R. Beivide

August 2014

King networks were proposed as higher degree alternatives to 2D tori and meshes. These networks offer improved throughput and latency with minimum distance routing in benign traffic patterns. As a solution to performance problems in adverse traffic pattern of the minimal routing, this article presents a missrouting solution. It uses two parameters to control the generation of routing tables. Using the paths on these tables, packets reach their destination nodes through non-minimal paths, without deadlock, livelock or starvation. Optimal values for the parameters are empirically determined. This routing strategy slightly improves the throughput while hardly increasing the base latency and extending the linear behaviour of the network. Experimental results confirm the good properties of this algorithm compared to minimal routing and Valiant algorithms.

1 Introduction

Interconnection networks are becoming fundamental part of any modern computer. Their design heavily conditions the performance of the computers built around them. Therefore they receive constant attention in the literature[**?**].

King networks were proposed as a compromise between high-radix indirect networks and low-radix direct networks. They are eight-degree evolutions of well known two-dimensional networks, meshes and tori. Previous studies revealed their tantalizing topological properties, such as low diameter, average distance and high bisection bandwidth[**?**]. Other interesting properties include straightforward partitioning and resource placement. There is a folding scheme to avoid scalability problems with wrap-around links. And also, their suitability for fault tolerant applications was studied in [**?**].

Although several minimal routing algorithms were proposed with different degree of success, none seemed able to exploit the full potential of king networks. With uneven traffic distributions, the apparition of hotspots caused partial congestion of the network while leaving areas unused, limiting its performance.

This paper proposes a misrouting scheme for king networks, denoted $King\epsilon\delta$, that is based on source routing. Thus, the capability of the packets to diverge from the minimum path is set at injection time. This will allow packets to flow through unused areas of the network and reduce the negative effect of uneven traffic. The use of non-minimal paths increases the individual delay of the packets, but a better load-balance is achieved and therefore, the overall performance is improved. Moreover, since it uses source routing, it avoids starvation and livelock, which are common problems misrouting techniques have to address [?].

Particularly, the main achievements of the present paper are:

- A new source misrouting technique for king topologies is proposed. This method is deadlock-free and avoids communication anomalies such as packet livelock and starvation. The proposal is defined by two parameters, one influences the balance of the links in different directions, while the other limits the wandering of the minimal distance path.
- A complete study of the effects on performance of both parameters. As a conclusion it gives the best values for the parameters as a function of the diameter of the network.
- Finally, it shows an implementation of the misrouting mechanism based on lookup tables. The implementation design rules out the possibility of communication anomalies, like deadlock, livelock or starvation. With this implementation a full evaluation of the proposed mechanism has been carried out and a discussion of the experimental results is presented.

The King$\epsilon\delta$ routing algorithm has been tailored to exploit the topological richness of king topologies [?, ?]. These were conceived as a higher degree evolution of the classic 2D mesh and torus. These topologies have added links allow packets to advance in eight directions like the king on a chessboard. Thus, they double the degree (or radix) of their 2D counterparts. Nevertheless, they still present a straightforward layout. Furthermore, they exhibit excellent topological properties such as high throughput, low latency, easy partitioning, good scalability and fault-tolerance properties. The use of topologies with diagonal links has been considered in the past, in the fields of microprocessor design [?], FPGAs [?] and interconnection networks [?]. Also mesh and toroidal topologies with added diagonals have been considered, both with degree six [?, ?, ?] and eight [?].

Misrouting is not a novel idea. Through the years different authors have proposed solutions to the load-balance vs. latency increase trade-off. Chiefly among them is the well known Valiant routing [?], which selects a random intermediate destination for each packet. It gives a very good load-balance at the cost of doubling the average latency. In addition, the arbitrary change of direction produced at the intermediate node usually complicates the implementation of deadlock avoidance mechanisms. More modern proposals [?] [?], offer better latency but have to make concessions to avoid deadlocks.

The King$\epsilon\delta$ algorithm increases the path diversity of the minimal routing by using tables. The packets obtain a specially constructed routing record from the table that allows them to wander from the minimal path. This is implemented as a replacement of the routing record generation algorithm, and does not require further changes in the router logic. Routing tables have been widely used in the literature, both in system networks [**?**] and NOCs [**?**, **?**]. Deadlock avoidance is achieved by using the Adaptive Bubble Router(ABR) [**?**], which will be explained in section 3.3.

The remainder of this paper is organized as follows. Section 2 outlines some fundamentals of king networks. Then Section 3 explains the King$\epsilon\delta$ proposal and its implementation features, as well as the selection of the optimal values for the algorithm parameters. Finally Section 4 shows an evaluation, followed by conclusions and future lines of research.

2 King Torus Network

In [**?**] king networks were introduced and a first performance evaluation under different traffic patterns was presented. This section presents a summary of the king networks and their distance properties, as well as the bisection bandwidth and path diversity.

As usual, networks are modeled by graphs, where graph vertices represent processing elements and edges represent the communication links among them. The *king torus network* can be seen as a square 2D torus with extra diagonal links that turn the four degree torus into an eight degree network. In the next definition we give the formal description of the topology, and a graphical representation of this network is shown in Figure 1. As it can be observed, three different kind of links compose king networks. On one hand, the orthogonal links (directions X and Y) form a 2D torus. On the other hand, diagonal links (directions Z and T) form two disconnected components of the graph. Altogether, the three graphs shown in the figure form the king torus network.

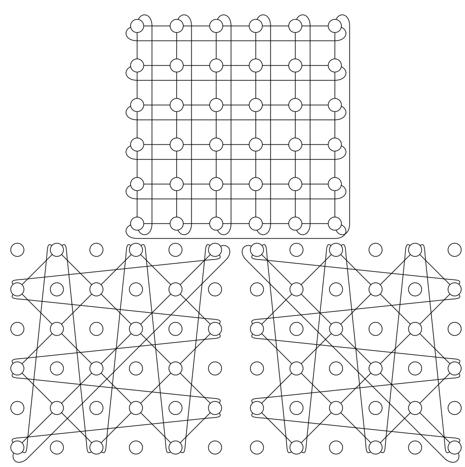

Figure 1: King torus network of side 8.

Definition 1 *Let s be a positive integer. Then, the king torus network of side s, denoted as KT_s, is defined as the graph such that:*

- *The set of nodes is $\{(x, y) \in \mathbb{Z} \times \mathbb{Z} \mid x, y = 0, \ldots, s-1\}$,*
- *Any node (x, y) is connected to the eight different nodes:*

$$(x, y) \pm (1, 0), \pm(0, 1), \pm(1, 1) \pm (-1, 1) \pmod{(s, s)},$$

  where $\pmod{(s, s)}$ means to take modulo in each component, that is, $(x, y) \pmod{(s, s)} = (x \pmod{s}, y \pmod{s})$.

Although nodes are labelled by means of two-dimensional vectors, we will consider this network as four-dimensional, since there are links in four different directions. Therefore, we will call to any link obtained by $\pm(1, 0)$ a link of direction X, $\pm(0, 1)$ a link of direction Y, $\pm(1, 1)$ of direction Z and $\pm(-1, 1)$ of direction T.

Topological properties are usually considered as a first indicator of the network behavior. In Table 1, expressions for the diameter, average distance and bisection bandwidth of both torus and king torus networks are shown. Note that these parameters are widely known for torus networks, [**?**], while the distance properties of king torus were presented in [**?**], and its bisection bandwidth was first calculated in [**?**]. As it can be seen, the effect of doubling the degree of the torus network to obtain the king torus produces a clear improvement of the topological properties. First, the diameter is halved while the average distance is decreased in a factor of $\frac{2}{3} \approx 0.67$. Theses values predict smaller maximum and average delays in ideal situations. Moreover, king torus network doubles the number of links of its degree four counterpart but triplicates its bisection bandwidth, which represents an upper bound for throughput under uniform traffic.

Network	T_s	KT_s
Diameter	s	$\lfloor \frac{s}{2} \rfloor$
Average Distance	$\frac{s}{2}$	$\frac{s}{3}$
Bisection Bandwidth	$4s$	$12s$

Table 1: Topological Parameters

3 Routing over King Torus Networks

This section presents an alternative routing scheme over king networks. Its key idea is balancing the use of links of different directions by allowing misrouting. As a consequence, this routing scheme will exploit all the path diversity available in king torus networks. The first subsection recalls how to route over king networks using minimum paths. While the second, presents the King$\epsilon\delta$ algorithm.

3.1 Minimal Path Routing

Routing in direct networks is usually done with algorithmic routing. That is, when a packet is injected, the path that lead to the destination will be computed using a simple algorithm or *routing function*. This will be coded into a vector, or *routing record*, with integer components. In common direct networks, like hypercubes or k-ary n-cubes, each component indicates the number of hops that must be made in each dimension to reach the destination. This routing record can be processed in a precise order, forcing packets to follow a specific path, or in any order, which gives packets a limited ability to adapt to local congestion. In a side s 2D torus, this routing record is calculated by subtracting the coordinates of the destination and source nodes, and applying modulo s to each component. However, nodes in a king torus are also labelled by two component vectors, but there are four possible travel directions, then let us concrete in the next definition what a routing record means in this context.

Definition 2 *Let a and b be two nodes of KT_s. A four-dimensional integer vector $(\Delta_X, \Delta_Y, \Delta_Z, \Delta_T)$ is called a* routing record *for nodes a and b if any path from node a obtained by Δ_X jumps in direction $(1, 0)$, Δ_Y jumps in direction $(0, 1)$, Δ_Z jumps in direction $(1, 1)$ and Δ_T jumps in direction $(1, -1)$ reaches node b. Moreover, this routing record is called* minimum *if $|\Delta_X|+|\Delta_Y|+|\Delta_Z|+|\Delta_T|$ equals the minimum distance between nodes a and b.*

Two different approaches for minimum routing calculation in king torus were considered in [**?**]. The first one was named *Knaive* and obtained minimum routing records with at most two non-zero components. This can be done since any two nodes can be joined with minimum distance using at most two directions: one orthogonal and another diagonal. The virtue of this algorithm is that it balances the use of all directions, giving best results in uniform traffic. However, experiments with adverse traffic patterns showed the performance of Knaive was not as good as expected. The reason for this was that it did not exploit all the path diversity available in the king network.

To this aim an enhancement of Knaive, named *EKnaive*, was developed that used routing records with three non-zero components. This can be done by applying the notion that two jumps in one orthogonal direction (X or Y) can be replaced by a jump in Z plus one in T without altering the length of the path. Routing records using this algorithm were still minimum and gave more path diversity but not all that is available, and it did not show the same the balance exhibited by Knaive.

3.2 Misrouting over King Networks

Two basic ideas are contained in this subsection. The first is trying to balance the use of all the directions, as it was observed with EKnaive that balance is important for uniform and other benign traffic patterns. To achieve this it is desirable that the absolute value of the components in the routing records as close as possible. The second idea is to equilibrate the use of all the network by allowing non-minimum routing records, that is, to perform misrouting to balance use of the links. These concepts are formally described in the following definition.

Definition 3 *Let ϵ, δ be two positive integers. Let a and b be two nodes of KT_s at a minimum distance d. Let $(\Delta_X, \Delta_Y, \Delta_Z, \Delta_T)$ be a routing record of the nodes a and b. Then,*

- *The routing record is said to be ϵ-balanced if:*

$$\begin{array}{lll} ||\Delta_X| - |\Delta_Y||, & ||\Delta_X| - |\Delta_Z||, & \\ ||\Delta_X| - |\Delta_T||, & ||\Delta_Y| - |\Delta_Z||, & \leq \epsilon, \\ ||\Delta_Y| - |\Delta_T||, & ||\Delta_T| - |\Delta_Z||, & \end{array}$$

- *The routing record is said to be δ-diverted if*

$$|\Delta_X| + |\Delta_Y| + |\Delta_Z| + |\Delta_T| \leq d + \delta.$$

First of all, note that any ϵ-balanced 0-diverted routing record gives minimum paths. Therefore, the search for ϵ-balanced routing records would include the two previous minimal options, that is, Knaive and EKnaive routing algorithms. However, as we will show in the results section, better performance is obtained when misrouting is allowed. Hence, for a king torus network this paper explores more suitable values of ϵ and δ so that a better performance in different traffic configurations can be obtained.

A preliminary study of 3-balanced routing tables was considered in [**?**]. As it will be shown in the following section, selecting $\epsilon = \delta = \frac{d}{2}$, where d denotes the diameter of the king network, gives the best trade-off between latency and throughput.

Both, ϵ and δ represent the two ideas presented at the beginning of the subsection. The ϵ parameter represents the component balance. The smaller the ϵ, the more balanced are the components, thus increasing the possible paths. Therefore, especially in adverse traffics, the profitable channels are increased and traffic is spread over a wider are of the network. Whereas, increasing δ allows packets to wander farther from the minimum distance path, therefore allowing potentially empty areas of the network to be used.

Finally, the following example illustrates how this new technique entails an increase of the path diversity between a pair of nodes. Consider nodes $(0, 0)$ and $(13, 14)$ in KT_{16}. For three different configurations of ϵ and δ we have considered all the ϵ-balanced δ-diverted routing records between both nodes.

Then, in Table 2 the average path diversity obtained thanks to these routing records is shown. Note that a different choice of ϵ and δ implies an increase of the average number of paths between any pair of nodes. In the next section will show how this increase translates to great performance gains.

(ϵ, δ)	$(3, 0)$	$(4, 5)$	(6, 7)
Average path diversity	0.89×10^4	0.46×10^6	0.98×10^6

Table 2: Average path diversity

3.3 Routing Implementation

In this subsection the technical aspects of the routing implementation are discussed. King networks are vertex-symmetric graphs. In an intuitive way, this means that the appearance of the network from a vertex is the same for any other vertex. Therefore, calculating routing records depends on the relative position of the destination node from the source node, rather than the absolute positions of nodes themselves. As this algorithm relies on tables, the vertex symmetry of the network means that there is only one table, of which every node has a copy. Note that the table is calculated off-line once for each network size.

Given the network dimensions, the table is defined by the two parameters, ϵ and δ, and is indexed by the relative position of the destination node. For each node there is a set of possible ϵ-equilibrated and δ-diverted routing records that are chosen at random when the packets are injected.

To find the best value for the parameters, an initial evaluation of all the tables with $0 \leq \epsilon \leq d$ and $0 \leq \delta \leq d$, were d is the diameter of the network, was performed. After observing the results, the scope was thought wide enough, as local maximums were found. The results for individual traffic patterns showed a performance improvement in all traffic patterns except uniform, as will be explained later.

From the above exploration it was concluded that, as the parameter values increase, the performance in adverse traffic patterns improves, but the average latency also increases. Analysing these results in detail allows selecting values for both parameters that maximise the throughput and minimise the latency. These values were found to depend on the diameter d of the network, $\epsilon = \delta = \frac{d}{2}$.

An interesting conclusion from this study is that the more adverse a traffic pattern is, the more significant is the improvement with large ϵ and δ. For instance with transpose traffic pattern the best results were obtained with $\epsilon = \delta = d$.

One concern related to the implementation of tables is their size. Presently the selected tables have a large amount of entries per destination in order to attain maximum path diversity. This is impractical, so a way to reduce the size was required. The method devised to reduce the size was to choose a subset of the entries of a table at random , such that the resulting table had a precise average number of entries per destination, or *multiplicity*. The selection was made ensuring that each destination had at least one entry. Then, tests were run to explore the performance vs. multiplicity trade-off. As can be seen in figure 2, the performance stabilises as the multiplicity increases. For the experiments a value of eight was chosen, since it gave acceptable results with reasonable table sizes. For example, multiplicity eight tables for 16×16 networks have 255 destination nodes, and that an entry can be packed in two bytes. Then, the total size of the table is almost 4KB. However, slightly better performance can be achieved with larger tables.

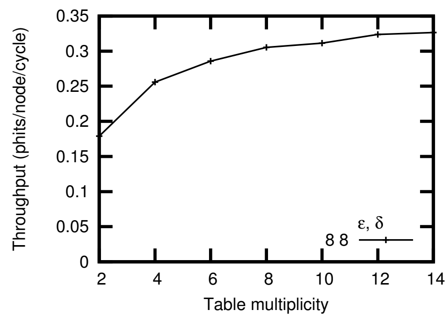

Figure 2: Representation of throughput vs. table multiplicity in a 32×32 king torus. The traffic pattern is uniform and load is 0.6 phits/cycle/node.

The flow control mechanism emploied is Virtual Cut-Through, with virtual channels. Through the use of ABR, the router needs only two virtual channels, and therefore two queues per physical link. One is used for adaptively routed packets and the other for statically routed ones. In ABR, the static virtual channels are managed under dimension-order routing (DOR), constituting a safe virtual network in which packet deadlock never occurs[**?**]. The remaining virtual channels are configured as a fully adaptive virtual network in which packet deadlock is not a concern. As long as there are available adaptive queues, ABR always routes packets through the adaptive network. Safe queues are only requested when all the profitable adaptive queues are full. This routing/flow-control mechanism is used by IBM BlueGene supercomputers [**?**].

4 Evaluation

In this section we present the experimental evaluation carried out to verify the better performance and scalability of the proposed misrouting approach. This is done by comparing with the Knaive arithmetic routing and the well known Valiant algorithm [**?**]. The evaluation will show the advantages of using King$\epsilon\delta$ technique selecting $\epsilon = \delta = \frac{d}{2}$ against the aforementioned algorithms. This study has been made with 16×16 and 32×32 networks. Although only the results for 32×32 are shown, similar conclusions can be extracted from the results of 16×16 networks.

All the experiments have been done on a functional simulator called INSEE[**?**]. The router model is based on the bubble adaptive router presented in [**?**] with two virtual channels. An important factor in the evaluation of networks are the traffic patterns. The evaluation has been performed with synthetic workloads using typical traffic patterns. These were applied on the networks by injecting packets of 8 phits at a constant rate, or load, measured in number of phits per cycle per node. To ensure the validity of the results, measurements of performance parameters were taken only when the network reached a steady state. The increased degree of some topologies theoretically allows the throughput to rise above one phit per cycle per node. In order to take advantage of this, each router had up to three injectors per node. The metrics considered are throughput, average latency and also maximum latency.

Figures 3 and 4 show the results in 32×32 networks, thus selecting $\epsilon = \delta = \frac{16}{2} = 8$, when stressed with well known traffic patterns. The tables used have multiplicity eight as was explained in section 3.3. The values measured are throughput and latency vs. increasing load. As will be shown, the behaviour of the different algorithms is clearly divided in two tendencies. First, in uniform and reversal traffic patterns the best performance is obtained with minimum distance routing, these patterns are considered *benign*. Second, the remaining traffic, are considered *adverse* and therefore benefit from the misrouting techniques.

It is known that uniform and reversal traffic patterns naturally balance the use of all the network resources and hotspots do not appear. In addition, it has been shown that the Knaive routing does not disturb this equilibrium[**?**]. Then, although the misrouting approaches are also balanced, they force each packet to stay longer in the network, so throughput can not be as high as with a minimal routing. Furthermore, uniform traffic rarely occurs in real situations and the bursty nature of applications do not usually force networks to operate in a sustained saturation.

The Valiant algorithm completely randomizes the communication independently of the used traffic pattern. So it is assumed that it gives an upper bound for the througput in adverse traffic patterns. Consequently the graphs show that this algorithm always reaches the best throughput in these traffic patterns. It is noticeable that the technique presented in this paper always improves on the minimum distance routing.

The advantage of the Valiant algorithm in throughput is obtained at the expense of doubling the average distance of the packets, and therefore increasing the latency. The diminished throughput of our proposal is a tradeoff of its benefits. Namely, that the average distance is only increased in 50%, thus the latency increase is much smaller. Especially at low loads, which is the most common scenario, the latency is almost equal to that of Knaive. Furthermore, the asimptotic growth of the latency occurs at higher loads than the best of the other algorithms.

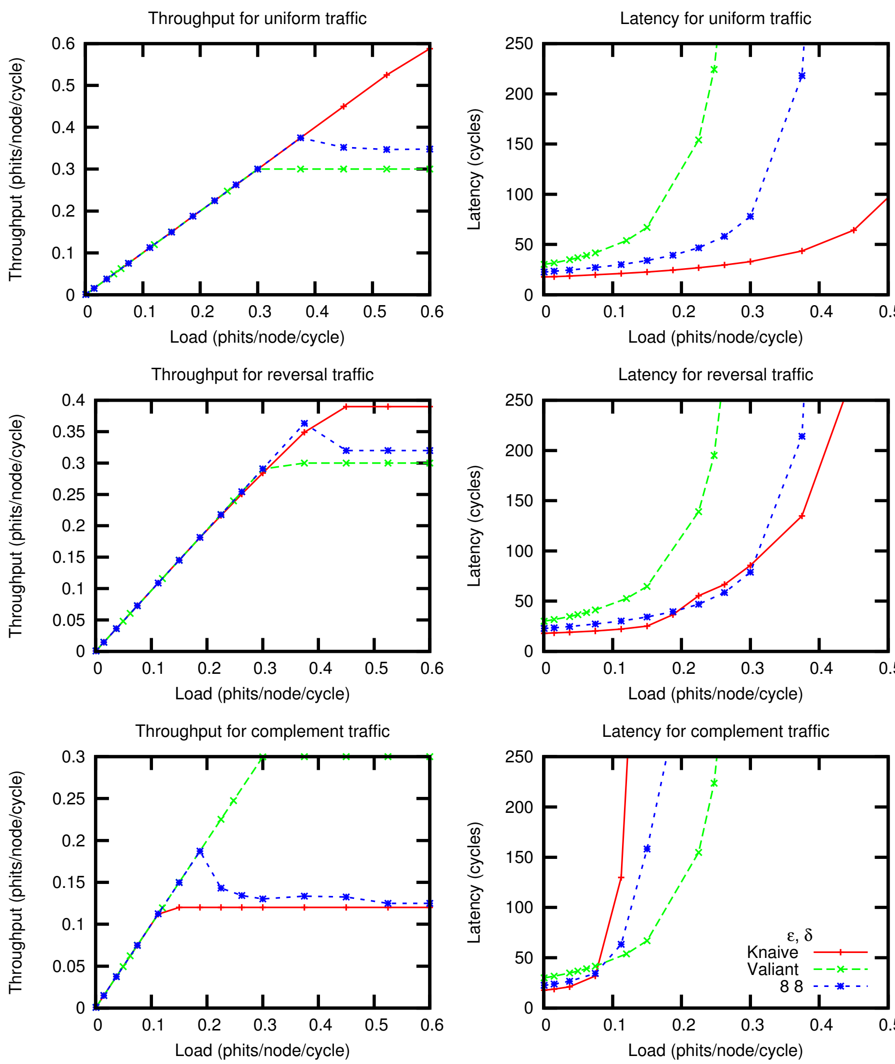

Figure 3: Throughput and latency comparison of King$\epsilon\delta$ with Knaive Valiant on a 32×32 king torus under various traffic patterns.

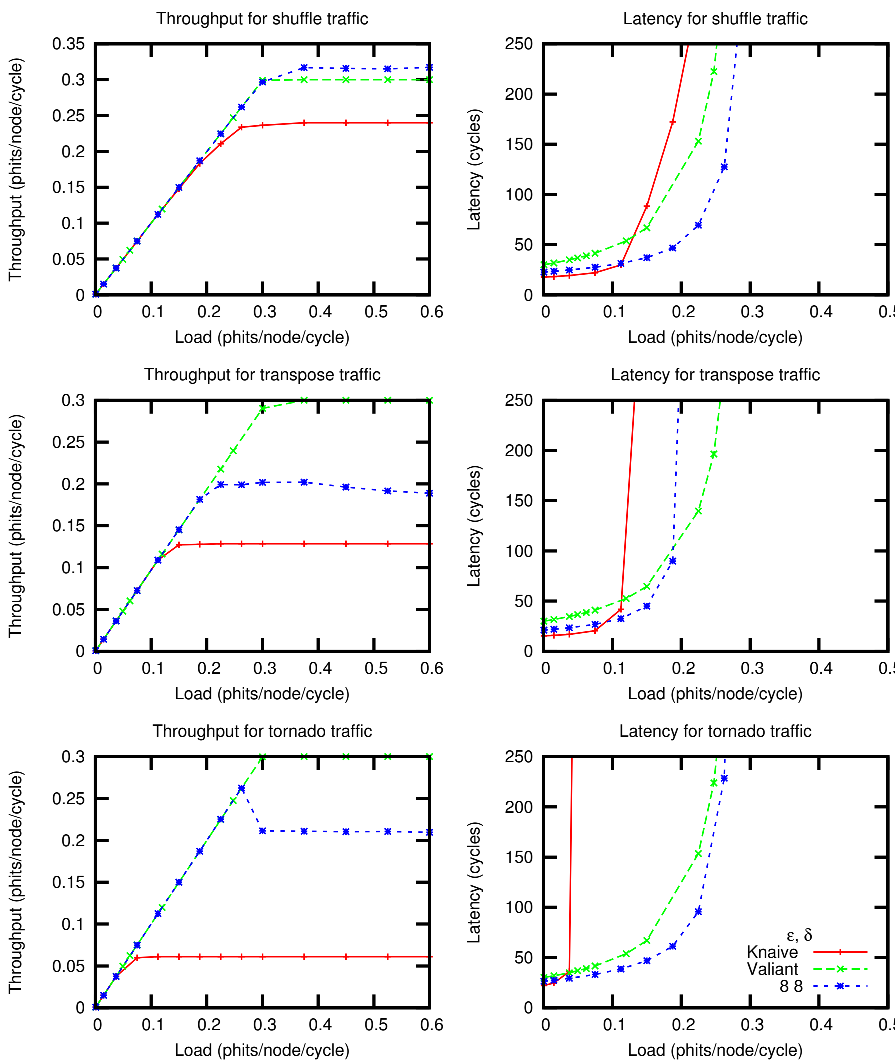

Figure 4: Throughput and latency comparison of King$\epsilon\delta$ with Knaive Valiant on a 32×32 king torus under various traffic patterns.

5 Conclusions

Interconnection networks are key components in modern computers. As one of the main bottlenecks they strongly influence the performance of these machines. Therefore, maximizing the use of the network resources is a top priority in the design of future systems. King networks show a large amount of paths connecting any two nodes. However, the use of minimal routing, like Knaive, does not make use of all the available paths. This causes that, in adverse traffic, only a few links are highly used, limiting the performance of the system. In these situations there are a large amount of idle links that could be used to improve the performance.

This paper presents a misrouting strategy that overcomes the above situation. It is based on two main ideas, direction balance and path expansion. These are synthesized by two parameters ϵ and δ. A complete study has been carried out of the impact of these two parameters in the network performance. As a consequence of this, the best values for these parameters have been found to be a function of the network diameter.

The strategy is implemented with look-up tables that are used at packet injection time. The paper presents a way of producing the tables with a realizable size, while preserving the virtues of the thechnique. On the other hand, being a source routing algorithm it prevents the appearance of typical misrouting artifacts like livelock or starvation. Moreover, it is deadlock-free as it uses the ABR mechanism. The main achievement of this proposal is improving the throughput of the Knaive routing in adverse traffic patterns without significantly increasing the latency at low loads.

Experimental results show significant improvements, both in throughput and latency. These are more notorious in adverse traffic patterns. In benign situations, like uniform or reversal, the proposed strategy reaches similar performance to Knaive when the network is not saturated. Therefore it is important to consider congestion control techniques. The remaining patterns show very good results, duplicating or triplicating the throughput.

The cost of this solution is the slight increase of individual packet latency. It is controlled by the δ parameter, so it is always bounded. Interestingly, at medium loads, the misrouting allows using more network resources, reducing hotspots and avoiding time to be wasted in delays around them, and therefore giving better performance.

Acknowledgment

This work has been partially funded by the Spanish Ministry of Education and Science (grant TIN2010-21291-C02-02 and Consolider CSD2007-00050), as well as by the HiPEAC European Network of Excellence.

References

[1] N. R. Adiga et al. An Overview of the BlueGene/L Supercomputer. In *Supercomputing, ACM/IEEE 2002 Conference*, page 60, nov. 2002.

[2] W. Dally and B. Towles. *Principles and Practices of Interconnection Networks*. Morgan Kaufmann Publishers Inc., 2003.

[3] J. Duato. A necessary and sufficient condition for deadlock-free routing in cut-through and store-and-forward networks. *IEEE Transactions on Parallel and Distributed Systems*, 7(8):841, 1996.

[4] Wen-Hsiang Hu, Seung Eun Lee, and Nader Bagherzadeh. DMesh: a Diagonally-Linked Mesh Network-on-Chip Architecture. *NoCArc, First International Workshop on Network on Chip Architectures to be held in conjunction with MICRO-41*, 2008.

[5] M. Igarashi, T. Mitsuhashi, A. Le, S. Kazi, Yang-Trung Lin, A. Fujimura, and S. Teig. A diagonal-interconnect architecture and its application to RISC core design. In *2002 IEEE International Solid-State Circuits Conference. Digest of Technical Papers (Cat. No.02CH37315)*, volume 1, page 210, 2002.

[6] Darren J. Kerbyson, Mike Lang, and Gregory Johnson. InfiniBand Routing Table Optimizations for Scientific Applications. *j-PARALLEL-PROCESS-LETT*, 18(4):589–608, dec 2008.

[7] J. Kim. Low-cost router microarchitecture for on-chip networks. In *Microarchitecture, 2009. MICRO-42. 42nd Annual IEEE/ACM International Symposium on*, pages 255–266, dec. 2009.

[8] Igor Loi, Federico Angiolini, and Luca Benini. Synthesis of Low-overhead Configurable Source Routing Tables for Network Interfaces. In *Proceedings of the Conference on Design, Automation and Test in Europe*, DATE '09, pages 262–267. European Design and Automation Association, 2009.

[9] Alan Marshall, Tony Stansfield, Igor Kostarnov, Jean Vuillemin, and Brad Hutchings. A reconfigurable arithmetic array for multimedia applications. In *Proceedings of the 1999 ACM/SIGDA 7th international symposium on Field programmable gate arrays - FPGA '99*, page 135, 1999.

[10] C. Martinez, E. Stafford, R. Beivide, C. Camarero, F. Vallejo, and E. Gabidulin. Graph-based metrics over QAM constellations. In *2008 IEEE International Symposium on Information Theory*, page 2494, 2008.

[11] Javier Navaridas, Steve Furber, Jim Garside, Xin Jin, Mukaram Khan, David Lester, Mikel Luján, José Miguel-Alonso, Eustace Painkras, Cameron Patterson, et al. SpiNNaker: Fault tolerance in a power-and area-constrained large-scale neuromimetic architecture. *Parallel Computing*, 39(11):693–708, 2013.

[12] Javier Navaridas, Mikel Luján, Jose Miguel-Alonso, Luis A. Plana, and Steve Furber. Understanding the interconnection network of SpiNNaker. In *Proceedings of the 23rd international conference on Conference on Supercomputing - ICS '09*, page 286, 2009.

[13] Javier Navaridas, Jose Miguel-Alonso, Jose A. Pascual, and Francisco J. Ridruejo. Simulating and evaluating interconnection networks with INSEE. *Simulation Modelling Practice and Theory*, 19(1):494, 2011.

[14] V. Puente, C. Izu, R. Beivide, J.A. Gregorio, F. Vallejo, and J.M. Prellezo. The Adaptive Bubble Router. *Journal of Parallel and Distributed Computing*, 61(9):1180, 2001.

[15] Rohit Sunkam Ramanujam and Bill Lin. Randomized Throughput-Optimal Oblivious Routing for Torus Networks. *Computers, IEEE Transactions on*, 62(3):561–574, march 2013.

[16] Ali Shahabi, Nima Honarmand, Hassan Sohofi, and Zainalabedin Navabi. Degradable mesh-based on-chip networks using programmable routing tables. *IEICE Electronics Express*, 4(10):332–339, 2007.

[17] K. G. Shin. HARTS: a distributed real-time architecture. *Computer*, 24(5):25–35, may 1991.

[18] Esteban Stafford, Jose L. Bosque, Carmen Martínez, Fernando Vallejo, Ramon Beivide, and Cristobal Camarero. A first approach to king topologies for on-chip networks. In *Euro-Par 2010 - Parallel Processing*, volume 6272 of *Euro-Par'10*, pages 428–439, Berlin, Heidelberg, 2010. Springer-Verlag.

[19] Esteban Stafford, Emilio Castillo, Fernando Vallejo, José Luis Bosque, Carmen Martínez, Cristobal Camarero, and Ramón Beivide. King Topologies for Fault Tolerance. In *HPCC-ICESS*, pages 608–616, 2012.

[20] K.W. Tang and S.A. Padubidri. Diagonal and toroidal mesh networks. *IEEE Transactions on Computers*, 43(7):815, 1994.

[21] Brian Towles, William J. Dally, and Stephen Boyd. Throughput-centric routing algorithm design. In *Proceedings of the fifteenth annual ACM symposium on Parallel algorithms and architectures*, SPAA '03, pages 200–209, New York, NY, USA, 2003. ACM.

[22] L. G. Valiant and G. J. Brebner. Universal schemes for parallel communication. In *Proceedings of the thirteenth annual ACM symposium on Theory of computing - STOC '81*, page 263, 1981.