Available online at www.sciencedirect.com
SCIENCE DIRECT
Journal of Systems Architecture 52 (2006) 345–358
JOURNAL OF SYSTEMS ARCHITECTURE
www.elsevier.com/locate/sysarc

High-performance adaptive routing for networks with arbitrary topology

V. Puente [a], J.A. Gregorio [a], F. Vallejo [a], R. Beivide [a], C. Izu [b,*]

[a] Computer Architecture Group, University of Cantabria, 39005 Santander, Spain
[b] Computer Science Department, University of Adelaide, SA 50005, Australia

Available online 14 November 2005

Abstract

A strategy to implement adaptive routing in irregular networks is presented and analyzed in this work. A simple and widely applicable deadlock avoidance method, applied to a ring embedded in the network topology, constitutes the basis of this high-performance packet switching. This adaptive router improves the network capabilities by allocating more resources to the fastest and most used virtual network, thus narrowing the performance gap with regular topologies. A thorough simulation process, which obtains statistically reliable measurements of irregular network behavior, has been carried out to evaluate it and compare with other state-of-the-art techniques. In all the experiments, our router exhibited the best behavior in terms of maximum/sustained performance and sensitivity to the network topology.

Keywords: Irregular topologies; Routing, Flow control; Deadlock; Cluster computing; Networks of workstations; Local area networks

1. Introduction

Networks of workstations and other forms of cluster computing are currently emerging in the high-performance computer market as good alternatives to sophisticated parallel computers. The cost/performance ratio of the commodity hardware and the existence of affordable and scalable high-performance communication technologies justify the penetration in the market of these distributed computing platforms. Moreover, their versatility makes this kind of systems particularly attractive to solve a wide range of applications.

Computer clusters are commonly organized as switched networks in which each switch or router has several computing nodes connected to some of its input/output ports. The remaining ports are used to link other routers in order to provide a connected system. Bi-directional full-duplex links are commonly employed to exploit communication locality. Packets interchanged among computing nodes cross the network according to the rules dictated by a certain routing mechanism. It is well known that the interconnection network architecture and its associated software libraries are critical components for high-performance cluster computing.

* Corresponding author. Fax: +34 942 201479.
E-mail addresses: vpuente@atc.unican.es (V. Puente), jagm@atc.unican.es (J.A. Gregorio), fernando@atc.unican.es (F. Vallejo), mon@atc.unican.es (R. Beivide), cruz@cs.adelaide.edu.au (C. Izu).

1383-7621/$ - see front matter
doi:10.1016/j.sysarc.2005.09.003

In recent years, the adoption of packet routing techniques successfully used in multiprocessor systems has been a common technological trend for commercial high-performance interconnection networks. Nevertheless, not all the low-level network functions used in multiprocessors can be easily adapted to this new scenario. One of these critical functions, conditioning either cost or performance of the whole network, is the method of dealing with packet deadlock. There are a number of successful distributed deadlock avoidance mechanisms based on the regular multi-dimensional structure of the network topology [6,8,17]. Notwithstanding, network irregularity is a common characteristic of most distributed computer systems. Although this feature allows for an easy and flexible network design, it requires new distributed mechanisms to deal with packet deadlock.

The mechanisms used in experimental and commercial high-performance communication technologies for cluster computing either require a significant number of hardware resources [13] or it enforces a restrictive use of its hardware resources [2,21] compromising in both cases their cost/performance ratio. Moreover, there are technologies that prohibit the use of irregular topologies [7,14], which clearly restrict the system versatility and thus, its applicability.

This paper provides a new approach to avoid deadlock in irregular networks by means of a controlled packet injection technique over a virtual ring embedded into the network topology. This method has been derived from a switching technique developed by the authors for regular topologies such as the torus that can be decomposed into a set of rings [4,18] and successfully employed in CC-NUMA multiprocessor systems [17], outperforming other high-performance routing proposals that use the same amount of resources. Actually, this routing mechanism has been implemented in the IBM BlueGene/L supercomputer [1].

The remainder of this paper is organized as follows: Section 2 presents the state-of-the-art on routing mechanisms for irregular networks. Section 3 introduces our routing methodology and Section 4 describes the corresponding router architecture. Section 5 details our simulation environment and Section 6 discusses the performance differences among the different proposals analyzed. Finally, the main findings of this research are summarized in Section 7.

2. Related works

This Section reviews the deadlock management functions associated with adaptive routers in irregular networks. This provides the context to analyze the basis of the different routing alternatives.

When a packet in transit reaches a router, its header provides the information to select the output port to be forwarded. The routing mechanism must safely choose an output port such that the network is maintained free of anomalies. Packet deadlock, as commented before, is the most critical anomaly that a network can suffer. A cyclic dependency between busy and requested communication channels is the source of deadlock among packets [6]. In other words, the transit of a set of packets indirectly depends on their own movement, it is thus impossible for them to advance towards their destinations.

In regular topologies, such as meshes and tori, links are arranged into several dimensions and deadlock avoidance mechanisms can be easily distributed among these dimensions [6,17,8]. The techniques used in these methods are diverse but they are mainly based on restrictions on the routing mechanisms or on the packet injection process. Other less conservative techniques rely on the fact that packet deadlock is infrequent and apply techniques based on deadlock detection and recovery [26]. Obviously, these distributed techniques cannot be easily exported to networks with irregular topology. The irregularity obliges the utilization of more restrictive and complex mechanisms, most of them based on a particular centralized view of the network. In most cases, the performance of the network is heavily conditioned by the restrictions imposed by its respective deadlock avoidance mechanism. For example, the up*/down* routing (UDR) selects a root node and embeds a *breath-first spanning tree* (BFS) in the network topology which provides a total order of the network channels [21]. The classification of a link as ascendant or descendent is determined by the relative positions of the nodes it connects to the root. A permissible path will be one in which the packet can only use descendent links after traversing all the ascendant ones, thus avoiding any cyclic dependency. The implementation of this mechanism is relatively easy but the restrictions imposed in the use of the network resources lead to poor performance. Packets have to travel non-minimal paths and frequently congestion builds up around the root node. A related mechanism using UDR based on a *depth-first span-*

ning tree (DFS) exhibits better performance, as it reduces traffic congestion around the root [22].

Another deadlock avoidance mechanism extends the L-Turn algorithm for regular networks [9]. To exploit the same idea in an irregular topology, a link classification is carried out among classes "up, down, right, left", inspired in a bi-dimensional mesh [12]. As in the regular scenario, some packet movements are forbidden depending on the link directions. The resulting network performance is lower than the one exhibited when using UDR based on a DFS spanning tree.

Smart routing is another alternative to manage traffic in irregular networks [5]. This mechanism is based on the construction of an explicit graph in which all the cyclic dependencies among the links in the topology are registered. Each cyclic dependency is broken by limiting the routing algorithm at some point of the cycle. The behavior exhibited by this mechanism shows a balanced utilization of the network links under random traffic. Notwithstanding, the algorithm employed to break the cycles down is based on a solver that requires an elevated computing time. In fact, networks with a few hundreds of nodes cannot be managed using this technique.

A different point of view is considered in the case of adaptive-trail routing [19]. This method is based on the search for an Eulerian path[1] embedded in the network topology. This path establishes an order in the use of the links to avoid packet deadlock. Its performance under different traffic patterns is reasonably good. It is important to note, that, this method has limited applicability because it is not always possible to find an Eulerian path in an arbitrary topology.

The previous proposals provide some traffic adaptability but they are all limited by the routing restrictions imposed to avoid packet deadlock. In order to achieve higher performance, adaptive minimal routing should be employed.

One method to support fully adaptive routing in irregular networks was presented in [23]. This technique is based on a well-established routing theory for regular networks introduced in [8]. The approach utilizes two virtual networks multiplexed over the physical topology. One of the virtual networks, the escape path, must be deadlock-free. The other virtual network can be managed without restrictions employing minimal adaptive routing. Moreover, when virtual cut-through flow control is employed [10], packets can change between virtual networks without restrictions.

When using this fully adaptive routing technique, the selection of the escape virtual network is a critical design issue as it is going to impact on the overall network performance. Our work is precisely focused on selecting a simple but adequate escape virtual network for irregular topologies.

3. Adaptive routing with restricted packet injection

A new adaptive routing for irregular networks is presented in this section. To implement an adaptive routing based on two virtual networks as in [23], any of the existent mechanisms that provide deadlock-freedom in an irregular network could be considered for building the escape path. This selection can be a complex task. An extreme solution, as stated in Section 2, to assure that deadlock could not exist is to identify and break down all the possible topological cycles in the network. This process can be very time consuming and depending on the network size, impracticable. Conversely, our design approach relies on finding an escape virtual network in the simplest and most economical way. We have to provide any node that can be involved in a deadlock situation with the possibility to break down the underlying topological cycle. If we offer, in a conservative manner such a possibility to every node in the system, we will obtain a deadlock-free virtual network.

Our deadlock-free virtual network is derived from a specific tour through the network. In other words, our escape network will be a virtual ring embedded in the network topology. By providing this ring with a convenient deadlock avoidance mechanism we will obtain a simple implementation of the deadlock-free escape network. This virtual ring used in combination with a fully adaptive network leads to a high-performance deadlock-free communication system.

To determine the topology of the escape ring, in a first step, we eliminate all those nodes that will never be involved in a topological cycle. These nodes correspond to open branches in the network, such as node number 5 in Fig. 1(a). In a second step, we look for a directed circuit embedded in the previously pruned topology that visits each node of the network one or more times up to the node degree.

[1] A path through a graph which starts and ends at the same vertex and includes every edge exactly once.

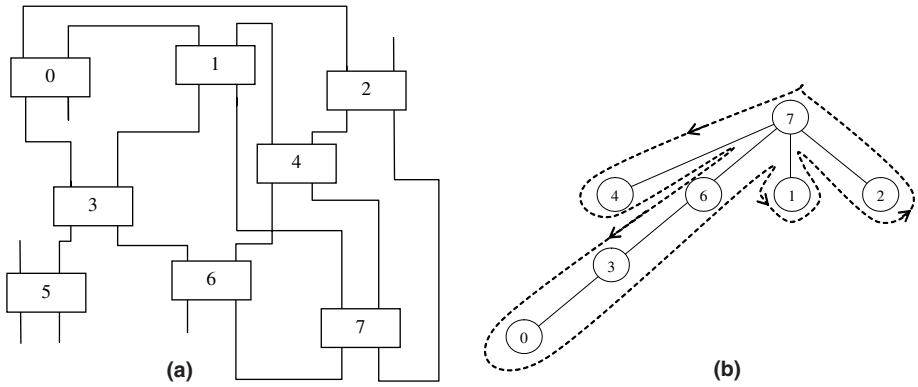

Fig. 1. (a) Irregular network topology. (b) Example of an escape virtual ring.

Our virtual ring topology is based on a peripheral tour through a spanning tree, always embedded in an arbitrary network. To illustrate how we obtain the ring, we trace, without lifting the pencil from the paper, a path through the tree, which visits the leaves as soon as possible. We may return to each vertex as many times as needed to visit all its children, returning at the end, to the starting vertex.

The directed ring in Fig. 1(b) "$7 \rightarrow 4 \rightarrow 7 \rightarrow 6 \rightarrow 3 \rightarrow 0 \rightarrow 3 \rightarrow 6 \rightarrow 7 \rightarrow 1 \rightarrow 7 \rightarrow 2 \rightarrow 7$" has been derived from the above-mentioned peripheral tour through an undirected spanning tree embedded in the network. The resulting escape virtual network based on this tour can be seen in Fig. 2(a). This undirected ring will visit all the nodes at least once and each edge twice. As the tree has $N-1$ links, the resulting directed escape ring will have $2(N-1)$ links. In fact, if we consider the spanning tree as a directed graph having two independent opposite links between nodes, our escape ring constitutes an Eulerian tour *inside* the tree. As any vertex can be visited several times, nodes will contain information about the input virtual channel and output port pairs that constitute the escape ring.

The minimization of the escape virtual ring's length has also been considered in this research in order to improve network performance. It is clear that a Hamiltonian cycle embedded in the pruned undirected network would provide us with two minimal length escape circuits (see for example, Fig. 2(b)). Nevertheless, the search for Hamiltonian paths in irregular graphs is an NP-complete problem [3]. This implies that this method can be extremely costly for medium-to-large networks. Moreover, not always does an arbitrary graph have an embedded Hamiltonian cycle. In our experiments, we will employ a backtracking algorithm to find Hamiltonian paths on undirected graphs, such

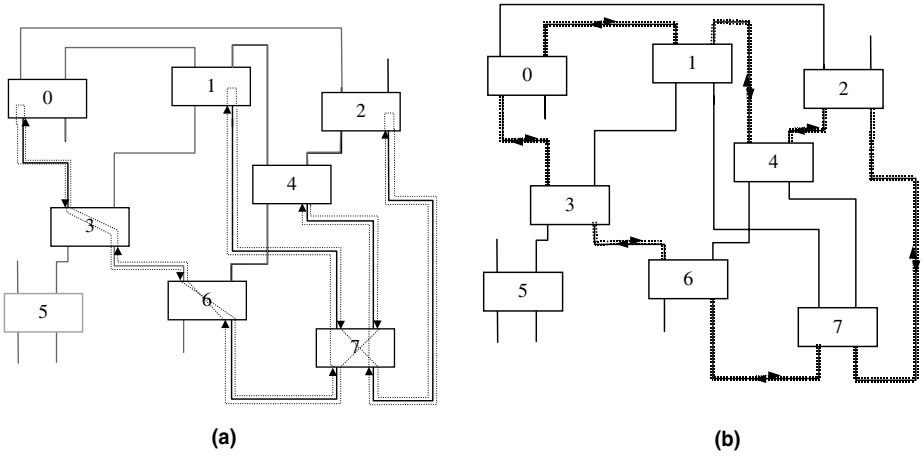

Fig. 2. (a) Tree-based escape path. (b) Hamiltonian-based escape path.

as the one proposed in [11]. As we will see later, when the backtracking algorithm does not provide a solution in a few seconds, the search is aborted and the longer escape ring based on the tree tour is used.

Although rings are deadlock prone topologies, there are simple and well-established mechanisms to avoid this anomaly. We use virtual cut-through flow control (VCT) [10] in which the only condition for packet transmission between nodes is the existence of a free buffer in the destination node to eventually store the whole packet in case it blocks at that node. We will also rely on a mechanism of restricted packet injection as in [20] which, for regular networks using VCT, exhibits better performance than the traditional mechanism based on two virtual channels [17].

In our mechanism, any node in the escape virtual ring can transmit packets as regulated by VCT but no packet can be injected by a node unless room for two packets is guaranteed in its local buffer. Even when a node injecting a packet receives a transit packet as well, the node buffer space corresponding to the virtual ring would not be exhausted. This mechanism, called bubble flow control (BFC), guarantees at least one free buffer in the ring (a *bubble* under our terminology), so that transit packets can progress and deadlock never occurs. Note that transit packets have more probability to advance in the network than the new ones trying to be injected. Consequently, this strategy if used in isolation, may lead to packet starvation. However, when this deadlock-free network is combined with another adaptive virtual network, packet starvation cannot appear (For further details see [18]). In the adaptive virtual network, all the packets, new or in transit, are treated in the same way, so all packets will progress, including those at the injection queues. This proposal has been used in the interconnection network of the BlueGene/L supercomputer [1].

To map the adaptive and escape networks into the same physical network, we will use two disjoint subsets of virtual channels. All of the virtual channels will transmit packets under a VCT flow control policy. The subset that constitutes the adaptive virtual network will manage packet injection without restrictions, just fulfilling the condition imposed by VCT flow control. The other subset, constituting the escape ring, will only accept a new packet if the BFC condition holds. Packets always try to travel through the adaptive minimal network. A packet will enter the escape ring only when all the profitable adaptive virtual channels for this packet are exhausted. Changes from the escape to the adaptive network are possible and regulated by VCT flow control. Changes from the adaptive to the escape network are considered as a new packet injection, thus regulated by bubble flow control. The resulting routing is free from both packet deadlock and starvation.

Notwithstanding, packet livelock could arise when using this switching mechanism. A packet traveling through the non-minimal routing escape ring can switch to the adaptive network at any router, provided that there is room in the selected adaptive buffer. The packet may need to enter the escape circuit again, getting further from its destination. Thus, this packet may indefinitely travel among virtual networks and never arrive at its destination. Nevertheless, livelock is eliminated by simply limiting the number of times that a packet can leave the escape ring. To implement this function, an additional field in the packet header is required to record the number of network changes. The optimal number of changes is dependent on the network size.

4. Router architecture

A scheme of a router architecture able to implement our controlled injection switching can be seen in Fig. 3. The same architecture will be employed to implement other adaptive switching mechanisms based on Up*/Down* routing, which have the same hardware costs.

In all the experiments, we consider a switch supporting 12 input/output bi-directional links. Up to 4

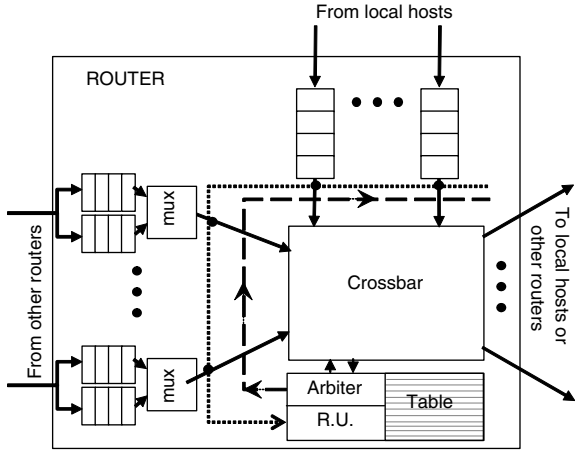

Fig. 3. Router architecture.

input links can be devoted to connecting local hosts. The remaining 8 input links, which are multiplexed between two virtual channels, are used to interconnect switches. Each pair of virtual channels belonging to each input port is multiplexed before entering the crossbar depicted in Fig. 3. Similarly, the 12 output links are split between local hosts and switch connections. Every physical link can transport in parallel 16 bits (1 phit). This routing device has a structure similar to the one of the Elite switch employed by Quadrics interconnection technology [14].

The router is a self-timed device pipelined in five stages. A first synchronization stage is required to accommodate the clock phases between neighbors. In a second stage a FIFO queue stores a phit of the incoming packet. This FIFO is able to store up to four packets of 64 phits. The routing unit, the crossbar arbitration and the switching stage consume the other three cycles. The routing unit processes the headers of the packets waiting in the first location of the FIFO queues. A scheme of this routing unit for a 4-port router can be seen in Fig. 4. To determine the profitable output ports, the destination node label recorded in the packet's header is used to address a local routing table. The number of bits required in each table entry is the number of output ports multiplied by the number of virtual channels. Once the set of profitable virtual channels in the neighbor nodes is determined, a selection routing function must choose one of them as the destination of the packet. This decision depends on both the packet routing policy and the availability of the output ports. This type of routing unit can be considered as a standard solution for irregular networks. The size of the routing table is only in the order of Kbytes, allowing for adequate network scalability under current technology trends.

Given that the escape path can cross a router through any combination of input–output ports, another table look-up is required to implement our controlled injection routing strategy. If the selected output channel belongs to the escape virtual ring (this packet movement is treated as an injection into the ring), it is necessary to check that the Bubble condition is fulfilled before sending the request to the arbiter.

To illustrate our mechanism we can focus on the example shown in Fig. 4. The escape path configuration is represented by dotted lines on the right of the figure. All the adaptive profitable channels labeled as "vc1" in the main routing table can send requests to the arbiter without any limitation. In some cases, the remaining profitable channels labeled as "vc2" must check the Bubble condition before sending requests. For example, to advance a packet stored in the "vc2" channel associated to input port 0 to the "vc2" channel associated to the output port 3, the Bubble condition is irrelevant as the packet continues traveling through the escape ring. Nevertheless, if the same packet tries to advance towards the "vc2" channel associated to output port 1, the Bubble condition must be verified as this packet movement represents a new injection in the escape

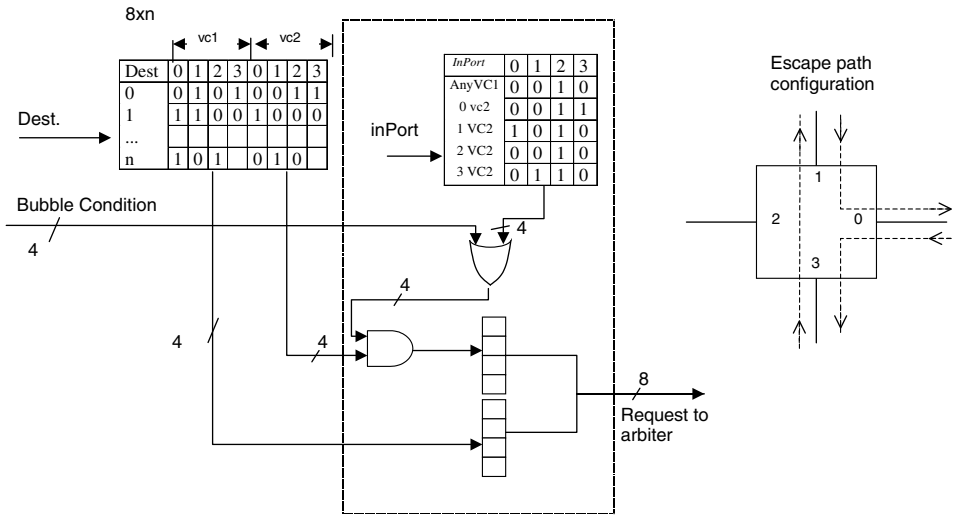

Fig. 4. Added complexity to the routing unit to support the escape network reconfiguration (example for a 4-port router).

ring. If a packet stored at any "vc1" channel tries to advance to any of the "vc2" channels associated to output ports 0, 1 or 3, again the Bubble condition must be fulfilled. Then, we use an additional small table to distinguish among all these cases. The value of any bit at position (i,j) indicates when the Bubble condition must be checked. A bit clear means that the Bubble condition fulfillment is required to advance a packet from input channel i to output port j.

As can be seen, the additional table required and its control logic are quite simple. As mentioned before, our simulation models a pipelined router with five stages, such as the one presented in [18]. Although the small table is located in the critical path of the routing stage, it is known that this pipeline stage does not determine the router clock cycle. As the crossbar arbitration stage is usually more costly, there will be no increment in the router clock cycle [15].

5. Simulation environment

The experimental environment employed in this research is based on the SICOSYS simulator [16]. This software tool has been successfully employed to assess the network performance of regular topologies [16]. SICOSYS provides accurate measurements of the network behavior with a lower computational time than that exhibited by standard hardware simulators. In our experiments the only difference among the several adaptive router implementations will be their respective routing units.

It is clear that a specific routing mechanism can provide good performance with a given irregular network and exhibit poor behavior with another. Some previous works tend to underestimate the impact of the networks analyzed on their results. In some cases, clearly structured networks are employed [19]. In other papers, a single irregular network is employed for multiple experiments without providing any reason for its use [22,23].

A reliable analysis of the behavior of different routing mechanisms in irregular networks must carefully guarantee the independence of the results obtained from the topologies considered. To achieve statistically reliable results, we consider for each experiment a set of 50 different networks with similar average topological properties. Each set of networks will be generated by predetermining the network connectivity and the host distribution. The maximum and average number of connections per router together with a value for its standard deviation characterizes the connectivity of a particular set of networks. In the same way, the host distribution in each set of networks is characterized by the maximum and average number of hosts per router plus a value for its standard deviation.

The connectivity of a particular set of networks will be modeled in a pseudo-normalized manner. Accordingly, the connectivity of the samples within a set will be distributed as a Gaussian random variable restricted to values with physical meaning. The probability distribution has an upper bound established by the network degree (in our experiments 8 input/output ports per switch) and a lower bound fixed at a value of three. Medium-to-large networks with average connectivity under three are clearly unrealistic. A similar procedure will be employed to determine the host distribution for the same set of networks.

Once the average connectivity, C, for a sample from the set of networks is set, we have to determine the number of routers using one link, two links and so on, up to the network degree. This can be modeled assuming a linear relationship between the probability that a router has i connected links, Pi, and the number of links, i.e. $Pi = a \cdot i + b$. By considering d as the network degree, we can determine a and b, through the following equations system:

$$C = \sum_{i=1}^{d} i \cdot P_i = \sum_{i=1}^{d} i \cdot (a \cdot i + b)$$

$$\sum_{i=1}^{d} P_i = \sum_{i=1}^{d} (a \cdot i + b) = 1$$

A similar procedure will be employed to determine the distribution of the number of hosts connected to each router.

Once both distributions are established, we can finally build the network. The method employed is based on a random selection of router pairs with open connections up to complete the network connectivity. No more than one connection is allowed between two specific routers.

In our experiments, we consider networks with 64, 128 and 512 routers. For each network size, a set of 50 samples will be randomly generated according to the above process and considering the initial parameters showed in Table 1.

Once a sample is generated, a first stage in the simulation process will be employed to initialize the routing tables, including those controlling the

Table 1
Basic parameters for network generation

Average connectivity per router	Standard deviation in connectivity	Average number of hosts per router	Standard deviation in the number of hosts	Network degree	Maximum number of hosts per router
6 ports	0.2	2	1	8	4

escape virtual ring. As mentioned before, we will try to find a Hamiltonian cycle in a short period of time (in the order of seconds in our tests). If the search is unsuccessful, we will use the original escape virtual ring.

In all the experiments the packet size is 64 phits and the destination headers are randomly generated. Nevertheless, the packet injection process can be either uniform or in burst mode. The former pattern attempts to model the behavior of typical parallel applications over a distributed architecture while the latter tries to reflect the behavior of a local or wider area computer network. The initial number of simulation cycles discarded will be large enough for each system size to stabilize the network regime. At this point we simulate through a length from 50,000 cycles, for the smallest networks, to 100,000 for the biggest.

6. Analysis of the performance measurements

This Section is divided into two parts. In the first one, we will analyze the impact of the virtual escape topology on the system performance. In the latter, a comparison of the behavior exhibited by adaptive routers with different escape networks will be carried out.

6.1. Performance sensitivity to the escape virtual network

The suitability of our controlled injection routing mechanism for any irregular network will depend on the impact on the overall network performance of the selected topology for implementing the escape virtual ring. Fortunately, we will see that this effect is almost negligible.

To test this, one special set of 50 networks, of 64 nodes each, was generated. For all the samples of that set both a Hamiltonian cycle and, of course, the ring that tours the embedded spanning tree were found. The Hamiltonian cycle provides a bi-directional ring as the escape network. Although a unidirectional ring is sufficient to avoid deadlock, this choice provides a drastic reduction of the escape path lengths, and it levels the number of resources required for both strategies. When the Hamiltonian cycle is found, $2N$ virtual channels are dedicated to the deadlock-free network. When no Hamiltonian path is encountered, our virtual escape ring needs $2(N-1)$ virtual channels. Remember that N is the number of nodes that, after pruning the network, can be involved in any potential network cycle. On the other hand, the maximum length of any path in the escape network when a Hamiltonian cycle is used is, obviously, $N/2$ and the average distance for a packet that travels through the escape network is $N/4$. When the longer virtual escape ring is used, the maximum distance between nodes is $2N-1$ and its average value depends on the nature of the circuit itself. Nevertheless, as we will see later, these differences in topological distances do not seem to play an important role in network performance.

The performance measurements for both approaches are shown in Fig. 5. As can be seen, the greatest difference in maximum achieved throughput was under 7%.

This behavior can be explained by the way in which the escape network is used and by considering the average length traversed by potentially blocked packets. It must be remembered that the escape network in our switching mechanism is only used as a last resort for routing. Moreover, as changes from the escape network to the adaptive one are permitted at any time, packets always try to travel through

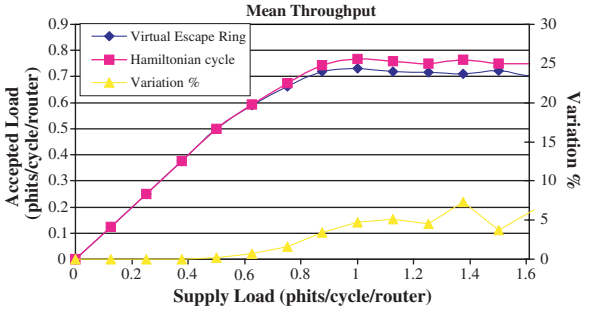

Fig. 5. Throughput exhibited by the tree-based virtual escape ring and by the Hamiltonian-based rings for a set of networks with 64 nodes.

the shorter adaptive routes. Besides, the restricted injection mechanism controlling the escape network reduces the volume of traffic this network can manage. In conclusion, the average use of the escape virtual channels is clearly lower than the use of the adaptive virtual channels.

We can illustrate this low-utilization of the escape network by analyzing the packet average distance when using the two different escape networks, as shown in Fig. 6. At low-loads, most packets travel through minimal distances using the adaptive network. As the traffic builds up, some packets must travel using the escape network, which is their last routing alternative. As the distances between nodes in the escape networks are non-minimal, the length of the path for a packet entering this network will be incremented. Notwithstanding, the differences in average distance between the two alternatives are almost negligible. In Fig. 7, a histogram showing the distribution of path distances can be seen. It can be observed that the frequencies of long paths in the network are close to zero. The probability of a packet traversing more than 7 links is under 1%; remember that in this case, the escape ring can have up to 127 links. These experimental measurements confirm that the performance of our routing mechanism is quite independent of the ring selected to implement the escape virtual network.

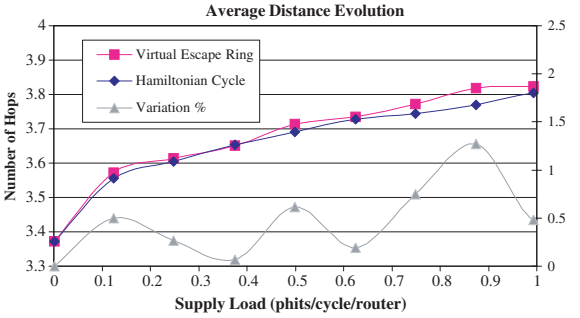

Fig. 6. Average distance evolution for the two escape path alternatives.

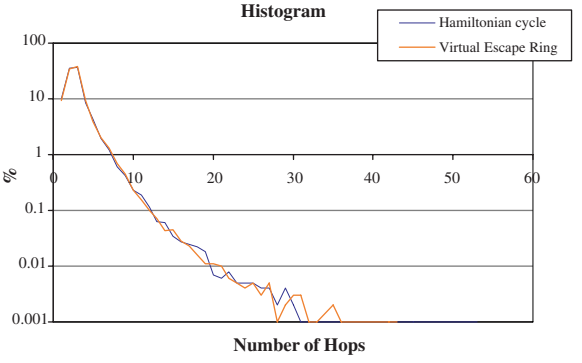

Fig. 7. Distance histogram for the two escape path alternatives.

The low-performance sensitivity to the length of the escape network is an important advantage that confirms the versatility of our controlled injection mechanism. Other studies proposing routing algorithms based on the use of specific paths show high-performance sensitivity with respect to the considered topology [19]. Furthermore, the computational effort needed to find our virtual escape ring is almost negligible and it depends quasi-linearly on the network size. Other routing mechanisms exhibiting good performance are impracticable when the considered network size is on the order of hundreds of nodes [5].

6.2. Network performance evaluation

In this sub-section we will compare the performance of our routing approach to those exhibited by other mechanisms. We will include the original up*/down* switching mechanism [21] as a baseline to determine the performance gains obtained by using minimal adaptive routing. We have also selected the adaptive up*/down* routing based on a DFS tree to compare with our controlled injection routing mechanism [22], since it exhibits high-performance and uses the same amount of resources as our method. We do not consider the L-turn routing [12] and the trip-based model presented in [25] because their performance is poorer than the one exhibited by the up*/down* DFS routing.

Other alternatives, as adaptive-trail routing [19] or smart routing [5], have not been considered in this analysis because of their high-computational costs and, in some cases, their impossibility to be applied to medium-to-large network sizes. In addition, adaptive-trail routing requires an Eulerian topology, which is not always the case.

Once the network samples are generated as described in Section 5, a complete experiment under uniform traffic was carried out. Figs. 8–10 show the average network throughput corresponding to networks of 64, 128 and 512 routers, respectively. Moreover, given that each point corresponds to an average of 50 values we have also represented their dispersion by means of the sample standard deviation divided by the average observed value.

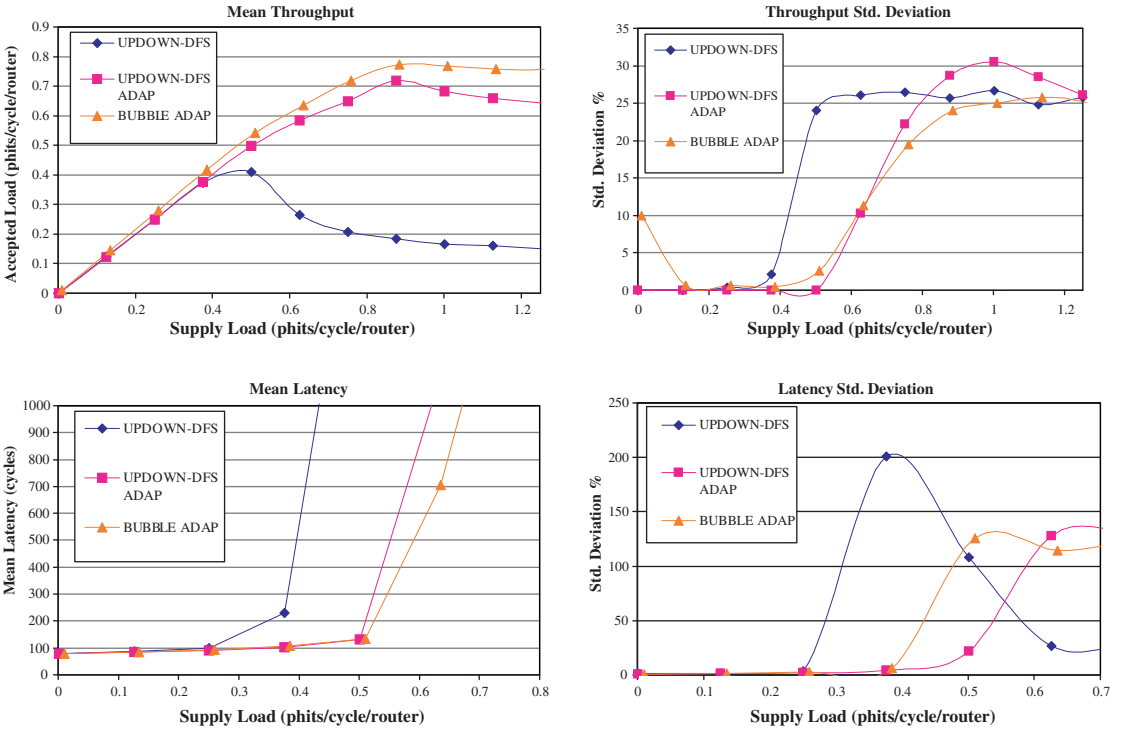

Fig. 8. Measurements for a set of networks of 64 nodes under uniform traffic.

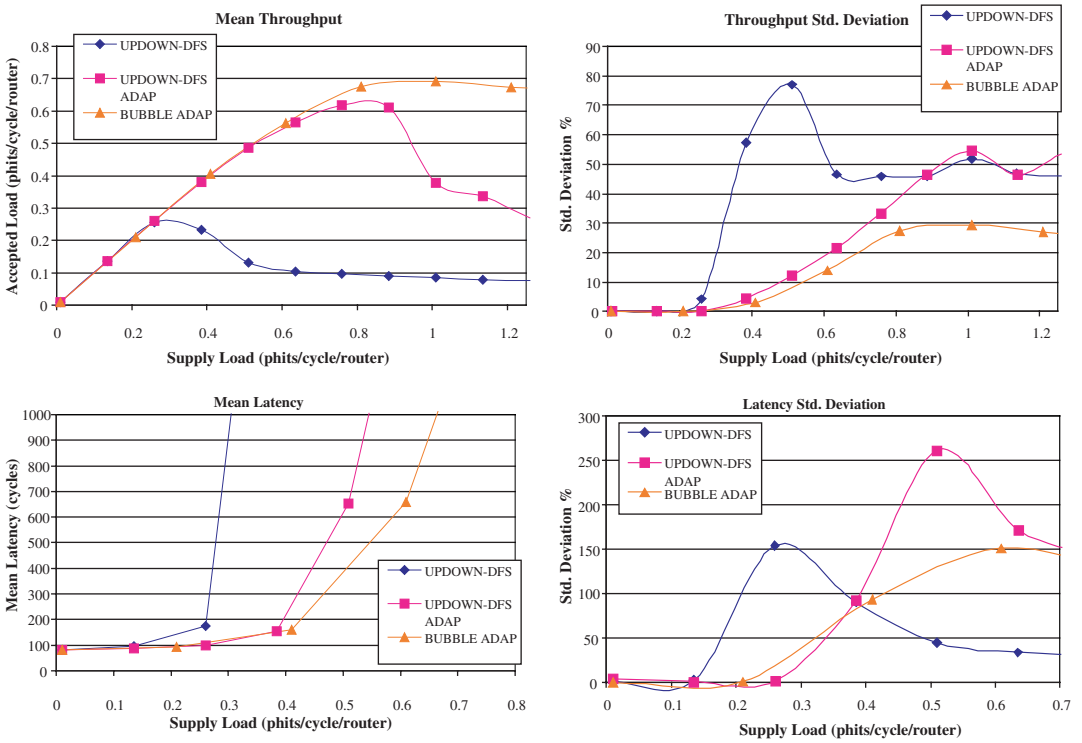

Fig. 9. Measurements for a set of networks of 128 nodes under uniform traffic.

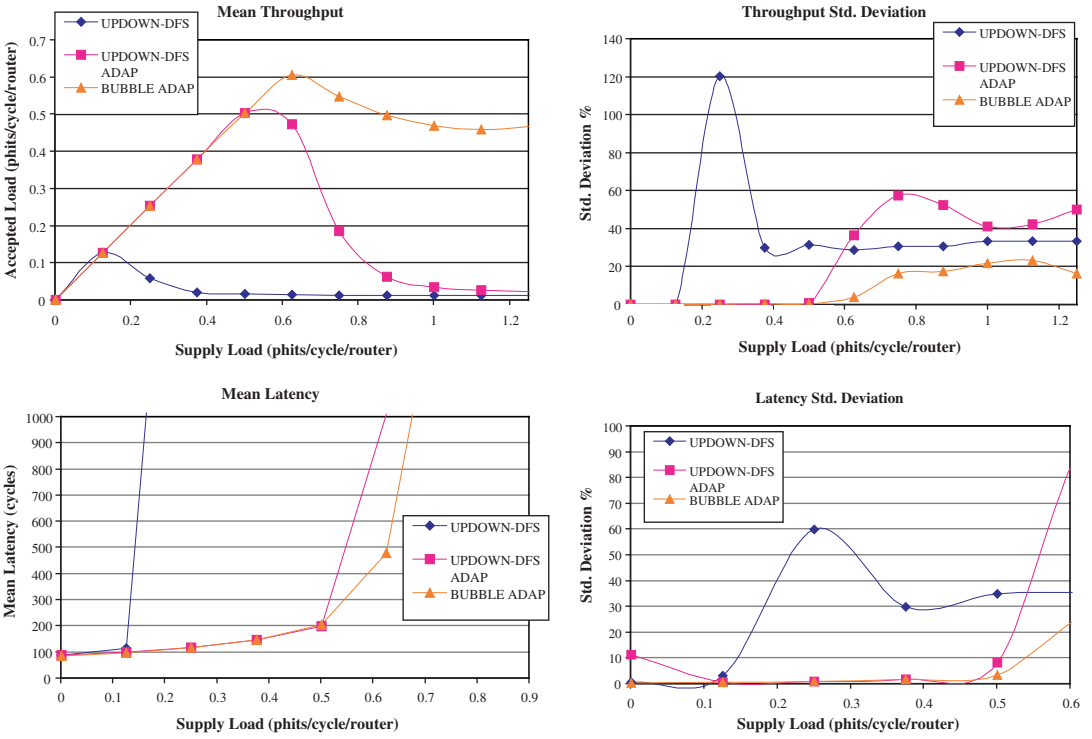

Fig. 10. Measurements for a set of networks of 512 nodes under uniform traffic.

We should note that the up*/down* mechanism does not require virtual channels, thus resulting in a simpler design compared to the two fully adaptive alternatives. However, both of them double the throughput achieved by the up*/down* router, clearly justifying the cost of implementing two virtual channels per physical link.

It is clear from the figures that our controlled injection routing outperforms the behavior shown by the adaptive up*/down* routing versions for all the network sizes. The bigger the network, the higher the performance differences. As network throughput is higher in our approach, its average packet latency is lower. These gains can be explained by looking at the resource (FIFO queues) distribution in the network. In our mechanism, out of the total of $2pN$ queues for the entire network (p stands for the number of input ports), $2N$ queues are reserved for building the escape virtual network. In contrast, the adaptive up*/down* routing employs pN queues for each one of the two virtual networks. We are favoring the adaptive virtual network in which packets travel without restrictions by the shortest paths. Thus, our performance gains are obtained by managing more traffic in the adaptive network and by using a regulated injection mechanism in the escape network that prevents network flooding.

We can also see in Figs. 9 and 10 that the volume of traffic managed by the network per time unit under our routing proposal remains more or less stable beyond the network saturation point. This sustained throughput is desirable for applications with intensive communication phases. The other two routers exhibit noticeable performance degradation. This behavior can be explained by the different nature of the escape virtual topologies. The adaptive up*/down* routing uses a tree as the escape topology and our mechanism employs a ring. A set of resources in a ring can be more homogeneously used than in a tree because there is no early saturation of links near the root.

By considering the statistical behavior of the measurements, a lower standard deviation can be observed in our routing mechanism when compared with the adaptive up*/down* alternative. This implies that our routing mechanism is less sensitive to the topological characteristics of the networks.

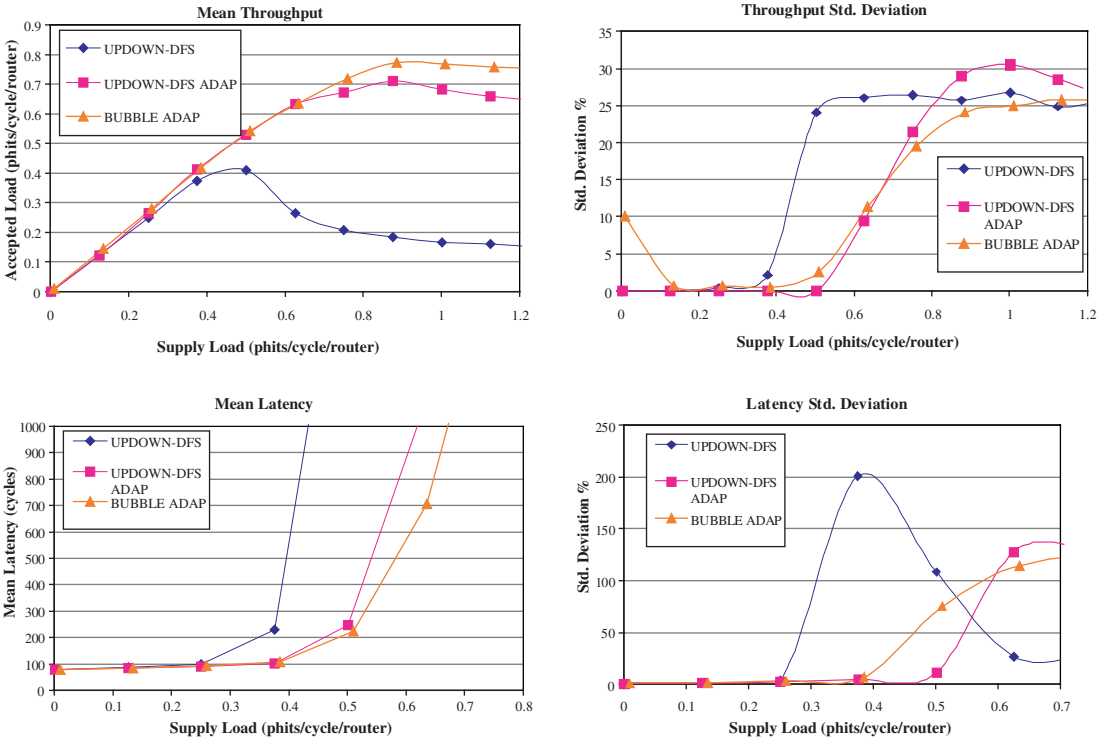

Fig. 11. Measurements for a set of networks of 64 nodes under bursty traffic.

This added advantage guarantees that no dramatic changes will be produced in the network performance when a network reconfiguration has to be carried out.

As we mentioned before, the larger the network, the bigger the differences between the two fully adaptive routing mechanisms under study. For this reason, only networks of 64 routers were employed to analyze the behavior of the different mechanisms under bursty traffic conditions. We model the bursty traffic using a switched Bernoulli process, selecting as basic traffic parameters the same values as in [24]. Fig. 11 shows the performance exhibited by the network as well as the standard deviation obtained.

As can be seen, the differences between the two adaptive routers are similar to those obtained for uniform traffic. Our controlled injection routing always behaves better than its up*/down* counterpart. The behavior is similar to that observed in the normal traffic case.

7. Conclusions

A new method to implement fully adaptive routing in networks with arbitrary topology has been presented and analyzed in this work. Our routing algorithm, based on a controlled packet injection policy applied on a ring embedded in the network, adequately balances the use of the resources and assures deadlock-free communications.

Our adaptive router has at least three clear advantages over previous proposals: versatility, efficiency and topological independence. With respect to versatility, our escape virtual ring is derived from a spanning tree that can always be found in arbitrary topologies in quasi-linear time. Thus, this technique can be successfully applied to any irregular computer network and hence, to any connected network.

Regarding efficiency, experimental simulation results have shown that our router outperforms other techniques that use the same amount of network resources. The gains are derived from a better distribution and utilization of the FIFO network queues. The use of our method implies a larger number of network queues assigned to the fastest and most used virtual network, thus favoring packet throughput. Furthermore, a subset of FIFO queues is restrictively managed in order to prevent network flooding. Another important feature of our router is

its ability to sustain peak throughput beyond the network saturation point.

In regard of the topological independence, our switching technique has been demonstrated to be the least sensitive to the network topology among the adaptive proposals considered in this paper.

Finally, the minor performance gains obtained from the use of optimized escape paths have demonstrated the suitability of our proposal in spite on using non-minimal paths.

Acknowledgments

This work has been done with the support of the Ministerio de Educación y Ciencia, Spain, under grants TIN2004-07440-C02-01 and PR2004-0572.

References

[1] N.R. Adiga, An overview of the BlueGene/L supercomputer, Supercomputing 2002, November 2002.
[2] N.J. Boden, D. Cohen, R.E. Felderman, A.E. Kulawik, C.L. Seitz, J.N. Seizovic, W.K. Su, Myrinet—A gigabit-per-second local-area-network, IEEE Micro 15 (1) (1995) 29–36.
[3] B. Bollobás, T.I. Fenner, A.M. Frieze, An algorithm for finding Hamiltonian paths and cycles in random graphs, Combinatorica 7 (4) (1987) 327–341.
[4] C. Carrion, R. Beivide, J.A. Gregorio, F. Vallejo, A flow control mechanism to prevent message deadlock in *k*-ary *n*-cube networks, HiPC97, December 1997.
[5] L. Cherkasova, V. Kotov, T. Rokicki, Fiber channel fabrics: evaluation and design, in: Proceedings of the 29th Hawaii International Conference on System Sciences, February 1995.
[6] W. Dally, C. Seitz, Deadlock-free message routing in multiprocessors interconnection networks, IEEE Trans. Comput. 36 (5) (1987) 547–553.
[7] Dophin NICs. Inc., The Dolphin SCI Interconnect—white paper, February 1996.
[8] J. Duato, A necessary and sufficient condition for deadlock-free routing in cut-through and store-and-forward networks, IEEE Trans. Parallel Distrib. Syst. 7 (8) (1996).
[9] C. Glass, L. Ni, Maximally fully routing in 2D meshes, in: International Symposium on Computer Architecture, 1992.
[10] P. Kermani, L. Kleinrock, Virtual cut-through: A new computer communication switching technique, Comput. Networks 3 (1979) 267–286.
[11] W. Kocay, An extension of multi-path algorithm for finding Hamilton Cycles, Discrete Math. (101) (1992) 171–188.
[12] M. Koibuchi, A. Funahashi, A. Jouraku, H. Amano, L-turn routing: an adaptive routing in irregular networks, in: International Conference on Parallel Processing, September 2001.
[13] H. Nishi, K. Tasho, T. Kudoh, J. Yamamoto, H. Amano, RHiNET-1/SW: an LSI switch for a local area system network, in: Cool chips III: International Symposium on Low-Power and High-Speed Chips, Kikai-Shinko-Kaikan Tokyo, Japan, 2000.
[14] F. Petrini, W. Feng, A. Hoisie, S. Coll, E. Frachtenberg, The quadrics network: high-performance clustering technology, IEEE Micro 22 (1) (2002) 46–57.
[15] V. Puente, J.A. Gregorio, C. Izu, R. Beivide, F. Vallejo, Low-level router design and its impact on supercomputer system performance, in: Proceedings of the 1999 International Conference on Supercomputing, June 1999.
[16] V. Puente, J.A. Gregorio, R. Beivide, SICOSYS: An integrated framework for studying interconnection network in multiprocessor systems, in: Proceedings of the IEEE 10th Euromicro Workshop on Parallel and Distributed Processing, January 2002.
[17] V. Puente, C. Izu, J.A. Gregorio, R. Beivide, F. Vallejo, The adaptive bubble router, J. Parallel Distrib. Comput. 61 (9) (2001).
[18] V. Puente, C. Izu, J.A. Gregorio, R. Beivide, J.M. Prellezo, F. Vallejo, Improving parallel system performance by changing the arrangement of the networks links, IN: Proceedings of the 2000 International Conference on Supercomputing, May 2000.
[19] W. Quiao, L. Ni, T. Rokicki, Adaptive-trail routing and performance evaluation in irregular networks using cut-through switches, IEEE Trans. Parallel Distrib. Syst. 10 (11) (1999) 1138–1157.
[20] A.W. Roscoe, Routing messages through networks: an exercise in deadlock avoidance. Technical report, Oxford University Computing Laboratory Report, 1987.
[21] M.D. Schroeder et al., Autonet: A high-speed, self-configuring local area network using point-to-point links, Technical Report SRC research report 59, DEC, April 1990.
[22] J.C. Sancho, A. Robles, J. Duato, New methodology to compute deadlock-free routing tables for irregular networks, CANPC'00, January 2000.
[23] F. Silla, J. Duato, Improving the efficiency of adaptive routing in networks with irregular topology, HiPC, December 1997.
[24] J. Solé, J. Domingo, J. García, Modelling the bursty characteristics of ATM cell streams, Technical Report UPC/DAC RR-90/17, Department of Computer Architecture, Polytechnic University of Cataluña 1997.
[25] Y. Tseng, D.K. Panda, T.H. Lai, A Trip-Based Multicasting Model in Wormhole-Routed Networks with Virtual Channels, IEEE Trans. on Parallel and Distributed Systems 7 (2) (1996).
[26] S. Warnakulasuriya, T.M. Pinkston, Characterization of deadlocks in irregular networks, in: Proceedings of the International Conference on Parallel Processing, September 1999.

Valentin Puente was born in Vendejo, Cantabria (Spain). He received the MD and Ph.D. degree in Physics from University of Cantabria, Spain, in 1995 and 2000, respectively. He is currently an Associate Professor in Computer Architecture at the same University. His research interests include parallel systems, simulation techniques, and interconnection network, with emphasis in performance optimization and fault-tolerant mechanism.

José Angel Gregorio was born in Bareyo, Cantabria (Spain). He received his MS and Ph.D. in Physics (Electronics) from the University of Cantabria, in 1978 and 1983, respectively. He is currently a Professor of Computer Architecture in the Department of Electronics and Computers in the same University. His research interests include parallel and distributed computers, interconnection networks, and performance evaluation of computers and communication systems. He is also a member of the IEEE Computer society.

F. Vallejo received his MS and Ph.D. degrees in Physical Science (electronics) from the Universidad de Cantabria, Spain in 1985 and 1991, respectively. Since 1985, he has been with the Departamento de Electrónica y Computadores, Universidad de Cantabria where he is currently an Assistant Professor of Computer Architecture and Technology. His current research interests are in interconnection networks for multiprocessors systems and parallel processing. Specific topics of his actual research are in interconnection mechanisms for transactional memory systems.

Ramón Beivide received the B.Sc. degree in Computer Science from the Universidad Autónoma de Barcelona in 1981 and the Ph.D. degree in Computer Engineering from the Universidad Politécnica de Catalunya (UPC) in 1985. He has been an Assistant Professor at the Universidad Politécnica de Catalunya and at the Universidad del País Vasco both in Spain. In 1991, he joined the School of Telecommunication Engineering at Universidad de Cantabria, Spain, where he is currently a Professor. His research interests include parallel computers, interconnection systems, performance evaluation and graph theory. He has published around 100 full-reviewed technical papers and he has served as referee, editor and program chair of different international magazines and conferences. He is a member of the IEEE Computer society.

Cruz Izu received the B.Sc. in Computer Science and Ph.D. in Computer Architecture from the University of the Basque Country in 1989 and 1994 respectively. She has been a lecturer at the University of Adelaide since 1996. Her research interests include parallel architectures, interconnection networks, network simulation and traffic load characterization.