A New Communication Mechanism for Cluster Computing

Andres Ibañez, Valentin Puente, Jose Angel Gregorio and Ramón Beivide

Computer Architecture Group
Universidad de Cantabria
39005 Santander, Spain
{*andres,vpuente,jagm,mon*}*@atc.unican.es*[**]

Abstract. A new fully adaptive routing algorithm for irregular networks is proposed in this paper. When compared to the most relevant routing proposals for networks of workstations with irregular topology, our routing algorithm has the characteristic of avoiding the existence of packet deadlock without using virtual channels. For a 256-node network, uniform traffic, and virtual cut-through flow control, our mechanism can outperform the classic up*/down* algorithm by a factor of 10. In fact, for medium size networks, the new technique can obtain better performance than its virtual channel-based counterparts even though it has a lower hardware complexity.

Keywords: irregular networks, routers, routing algorithm, deadlock, virtual channels, bubble method.

1 Introduction

Networks of workstations (NOWs) or other forms of cluster computing currently appear to be good alternatives for parallel computing due to their competitive cost/performance ratio. They are normally organized as switched-networks with irregular topology. It is precisely that irregularity which makes the packet routing and deadlock avoidance mechanisms more complex than in regular networks. Classical solutions impose some artificial order on the network nodes, normally forming a "tree", and route the packets by using non-minimal paths. In this way, the routing algorithms are simpler and the possible cyclic dependencies, responsible for packet deadlock, are eliminated [2]. However, these techniques have the drawbacks of the increase of packet latencies and waste of resources.

With the aim to avoid these limitations, in this paper we propose a new fully adaptive routing algorithm for irregular networks based on a solution, first proposed for multiprocessor systems [4], which avoids deadlock in regular networks. To achieve this, the new routing proposal selects a subset of physical links forming a Pseudo-Hamiltonian (PH) cycle, i.e., a cycle made up of those links and

[**] This work has been supported by Spanish CICYT, project TIC98-1162-C02-01.

nodes which could generate cyclic dependencies in the network. Subsequently, the so-called "bubble flow control method" is applied, thus avoiding the exhaustion of the storage resources belonging to the PH-cycle. Without modifying the current routers[1], this new method outperforms other standard techniques, obtaining a notably higher performance for network sizes over 64 nodes.

The rest of the paper is organized as follows. In Section 2, the main characteristics of the irregular networks and the router model used in this work are shown. Section 3 reviews classical and newer routing proposals for irregular networks. The new routing mechanism is presented in Section 4. The simulation environment is described in Section 5 and comparative results are shown in Section 6. Finally, we present the main conclusions in Section 7.

2 Irregular Networks

NOWs are normally organized as switched networks with irregular topology. Each switch or router is shared among several workstations connected to it through its ports. The rest of the switch ports are used for connecting the switch with other switches, facilitating network connectivity. Network switches or routers are connected by means of physical channels, generally bidirectional point-to-point links. The messages interchanged among nodes cross the network following paths that fulfill the rules of a routing algorithm. The specific route a packet will follow can be determined either in the emitting workstation or in the intermediate routers. The first method, known as source routing, includes in the packet header enough information to get to the destination. On the contrary, in distributed routing, each router has information (usually as a look-up table) for selecting the most profitable output channel for each incoming packet. In both cases, before the network is prepared to receive traffic, the routing tables have to be initialized with the appropriate information, depending on the selected routing algorithm.

With respect to the switching methodology, the router can implement one of the following three techniques: store-and-forward (SF), virtual cut-through (CT) and wormhole (WH). However, for high-performance networks, only virtual cut through and wormhole are good candidates. Both techniques have pros and cons and the selection is an important decision due to the consequences for the whole network. A comparison between the two techniques can be found, for example, in [2].

2.1 Router structures

The basic router model used is shown in Figure 1.a. It consists of an internal crossbar able to switch every input link to every output link, allowing multiple messages to cross the router simultaneously without interference. The number of input and output ports is generally the same, and for simplicity, the temporary

[1] Throughout this work, we will use the terms *router* or *switch* indistinctly.

storage (buffers) are located at the input links. However, the results of this work are not affected by buffer location.

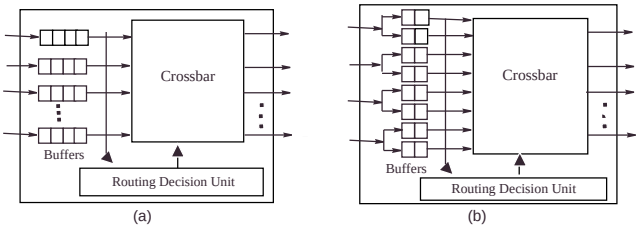

Fig. 1. Router models with input buffers, (a)without virtual channels (b)with two virtual channels per link

The switch has a Routing Decision Unit (RDU) responsible for routing each incoming packet toward its destination and selecting the most convenient output link. This profitable link is selected from a local look-up table (distributed routing) addressed by the input port and the final packet destination, taking also into account the neighboring router status.

In order to compare the results of our technique, another router structure based on virtual channels, such as the one in Figure 1.b, has also been used. This router allows several packets to share the same physical link in a multiplexed way. The virtual channels are implemented using separated buffers for each virtual channel. This scheme facilitates the design of deadlock-free routing algorithms and improves throughput significantly. Nevertheless, it requires additional hardware, because the RDU is more complex and the crossbar either has more inputs to arbitrate or these inputs must be multiplexed.

3 Classical routing algorithms for irregular networks

Routing algorithms can be deterministic or adaptive. The first type always provides the same path for any packet traveling between the same pair of nodes. On the contrary, adaptive routing algorithms determine the packet route depending not only on the source-destination pair but also on the network status. In any case, deadlock-free routing must be provided by every practical algorithm. A deadlock refers to a situation in which a set of packets is blocked forever because each packet of the set holds some resources (links or buffers) that are also needed by another packet. Next, we will focus on two deadlock-free minimal routing algorithms for irregular networks with different adaptability degrees.

3.1 Up*/Down* Algorithm

The up*/down* algorithm was proposed for Autonet networks [5]. It is a distributed deadlock-free routing scheme that provides partial adaptability in irreg-

ular networks. Its general strategy is based on routing packets in a tree, where the routes go up the tree on leaving the source and then, come back down at the destination. One of the nodes is chosen as the root of the tree (usually, the one closest to the rest of the nodes) and all links of the topology are designated as up* or down* links with respect to this root. The up*/down* state of a link is relative to a spanning tree computed in background by a distributed algorithm. A link is up* if it points from a lower to a higher-level node in the tree (i.e. to a node closer to the root). Otherwise, it is down*. For nodes at the same level, nodes IDs break the tie.

The routing from a source to a destination is established in such a fashion that zero or more up* links (towards the root) are traversed before zero or more down* links are traversed (away from the root) in order to reach the destination. This prevents cyclic dependencies among packets and thus, the routing is deadlock-free.

The advantage of this approach is that each node's hardware and software are simple and it provides some adaptability. The drawbacks are that the selected paths are generally not the shortest paths and that links near the root get congested and become bottlenecks that lead to low throughput. Moreover, these problems become critical when the network size increases.

3.2 Adaptive Up*/Down* Algorithm

Recently, a general methodology for the design of adaptive routing algorithms for networks with irregular topology has been proposed in [6]. This methodology attempts not only to provide minimal routing between every pair of nodes, but also to increase the adaptability. To summarize, this methodology starts from a deadlock-free routing algorithm for a given interconnection network, and shares physical links in the network by two virtual channels: escape and adaptive channels. The latter are used for fully adaptive routing, while escape channels are used in the same way as in the original routing function. A packet arriving at an intermediate router first tries to reserve an adaptive channel. If all the suitable outgoing adaptive channels are busy, then an escape channel is selected. If none of these provides a minimal path to the packet destination, the shortest path is chosen. The routing algorithms designed with this methodology are deadlock-free provided that the original routing algorithm is deadlock-free [1].

4 Bubble Routing

In this section, a new fully adaptive minimal routing algorithm is proposed for irregular virtual cut-through networks. The algorithm called "Bubble Routing" (BR), makes use of flow control for avoiding storage exhaustion in all those physical channels which could generate cyclic dependencies among packets and, therefore, produce deadlock. The aim of the BR mechanism is to obtain full adaptability allowing packets to follow minimal routes without any restriction,

provided it is always possible that, in case of blocking, packets are routed to their destination through some deadlock-free route.

To achieve this goal, BR selects a subset of physical links forming a Pseudo-Hamiltonian (PH) cycle, i.e., a cycle made up of those links and nodes that could generate cyclic dependencies in the network. Figure 2.a shows an example of an irregular network where a PH cycle has been defined. Note that those nodes forming open branches in the network, such as node number 7, do not belong to the Pseudo-Hamiltonian cycle. Messages stored in such nodes can never generate deadlock because their links can not form cyclic dependencies with other links.

Once a PH cycle is defined including all nodes and links that could generate cyclic dependencies, it is possible to achieve full adaptability. A PH cycle can always be obtained. In the worst case, a path traversing all the network nodes could be used. Links belonging to the PH cycle might be used as an adaptive option to follow minimal paths or as an escape route if there are no more choices. It is necessary that deadlock never occurs in this cycle so packets can never be indefinitely blocked. To achieve this goal, Bubble Routing is applied to avoid packet deadlock in links belonging to the PH cycle. A condition must be fulfilled to allow any packet to enter in this cycle. There must be room for at least two packets, one for the packet itself and another one to establish a "bubble". This "Bubble Condition" guarantees that the storage resources of the PH cycle are never exhausted and therefore packets inside the cycle can always advance. Once a packet is in the PH cycle, for advancing it to the next router, space is only necessary for the packet itself, i.e. using classical flow control techniques. It can be noted that the Bubble Condition can be verified at any storing resource (buffer) belonging to the PH cycle. In particular, the condition can be tested locally, in the same router where the routing decision is taking place [4].

It should be remarked that it is possible to achieve fully adaptive routing avoiding deadlocks without using virtual channels. The routing algorithm is deadlock-free as long as a deadlock-free PH cycle is always offered as an escape route. As with the adaptive up*/down* algorithm, packets can switch between escape channels and adaptive ones and vice versa (obviously, for entering the escape channels, the Bubble Condition must always be fulfilled). This packet movement freedom can give rise to livelock[2]. However, the Bubble Routing algorithm gives preference to minimal paths over non-minimal ones. This guarantees that as time tends to infinity, the livelock probability tends to zero. A similar strategy was used in [3] to prove that the Chaos routing algorithm, which allows packets to follow non-minimal paths when all the minimal paths are busy, was livelock-free. Next, we are going to describe the simulation environment used for comparing the different strategies against our new proposal.

5 Simulation framework

For comparison purposes, simulation techniques have been used to evaluate the performance achieved by different routing algorithms. A general-purpose inter-

[2] Packets traveling in the network that never reach their destination nodes.

connection network simulator, named NOWSIM (Network of Workstations Simulator), has been implemented as an extension of the network simulator NETSIM [7], developed at Rice University in the YACSIM environment [7].

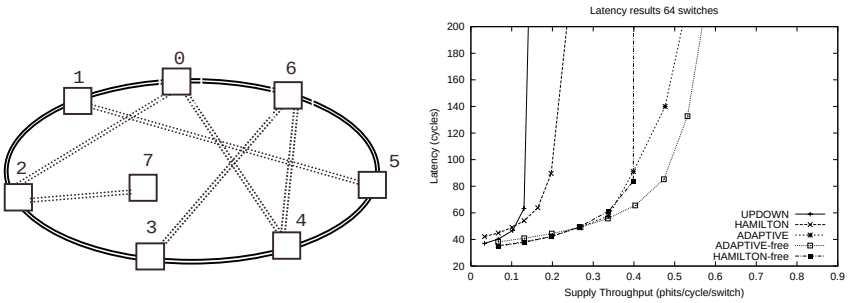

Fig. 2. (a) Pseudo-Hamiltonian (PH) cycle in an irregular network. (b) Average packet latency versus applied traffic for a 64-nodes.

Network topology is completely irregular and randomly generated. However, for the sake of simplicity, three restrictions to possible topologies are imposed. First, it is assumed that all the routers have a structure such as that of Figure 1, with the same size, 5 input and 5 output ports. Also, there is one processor connected to each router, thus leaving 4 ports available to connect to other routers. Finally, two neighboring routers are connected by means of a single link. The analyzed routing algorithms being partially or fully adaptive, all of them offer several routing choices. Therefore, all the algorithms require accessing a routing table, selecting among several options, and determining the most suitable output channel. Thus, it is assumed that it takes one clock cycle to compute the routing algorithm in all cases. Also, two cycles are needed to transmit one phit across both the crossbar and the buffer. And, finally, another cycle is spent in travelling between routers.

A virtual cut-through switching technique is assumed in the simulations. Messages are one-packet sized divided into 16 phits and we assume that each one can be transferred across a physical link per cycle. Buffers can store 8 packets. When multiplexing physical links between two virtual channels, buffers can only store 4 packets in order to maintain constant the buffer capacity per physical link.

6 Comparative results

In this section, a performance comparison of the routing algorithms described in Section 3 against Bubble Routing has been carried out. Irregular networks of different sizes (just 64 and 256 nodes are shown) have been simulated. Under the same evaluation methodology and simulation environment, results for the different routing algorithms have been obtained.

However, in order to clearly identify the performance range, we have considered two extreme situations for packet adaptability. On one hand, complete freedom for routing packets to their destinations following any minimal route and, on the other hand, to enforce any packet to follow the safe path determined by the deadlock-free algorithm though this is not minimal. Figure 2.b shows the average packet latency versus the applied load for 64 switches under uniform traffic. This Figure shows results of the up*/down* algorithm (UPDOWN), the two virtual channel adaptive algorithm with and without freedom for routing the packets (ADAPTIVE-free and ADAPTIVE), and the new fully adaptive proposal also with and without freedom (HAMILTON-free and HAMILTON). In these cases, the maximum number of hops necessary for a packet to reach its destination will always be the corresponding to the safe route (up*/down*-path for ADAPTIVE and Hamilton-path for HAMILTON). While in the ADAPTIVE-free and in the HAMILTON-free cases, that number of hops will usually be lower but only statistically limited.

As shown in Figure 2.b, for a 64-switches network, HAMILTON-free outperforms the throughput achieved by the classical strategy UPDOWN in a factor of 3.5. In fact, for this network size, the BR performance is close to that obtained by the ADAPTIVE-free, though for this case the router complexity is greater because it is necessary to implement two virtual channels per link. Even without doing misrouting respect to the safe routes, both algorithms behave better than UPDOWN because packets do not concentrate around the root. The cost is a little increase on base latency. However, the new proposal presents the more amazing results when network size is larger. Figure 3 shows the average packet latency and throughput for a 256-nodes network under uniform traffic. HAMILTON-free outperforms UPDOWN in a factor greater than 9, and surprisingly, it duplicates the performance of the more costly ADAPTIVE-free strategy.

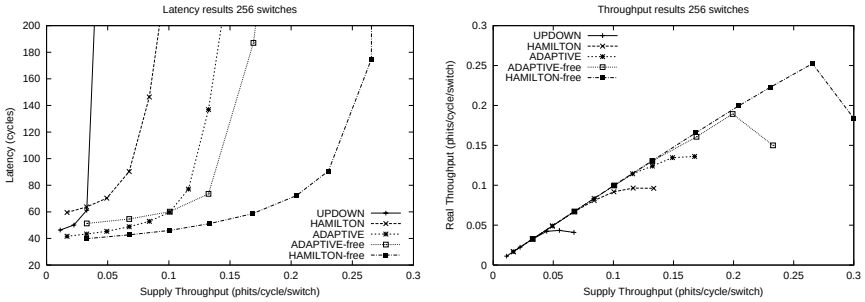

Fig. 3. Average packet latency and throughput versus applied traffic for a 256-nodes network

Obviously, the saturation points correspond to those in which the slope of the latency curves goes to infinite. But beyond these points, Figure 3 shows

the misrouting effect on the HAMILTON-free and on the ADAPTIVE-free algorithms. If we try to inject more packets than the network is able to handle, the performance of these algorithms will drop bellow the maximum achievable throughput. HAMILTON and ADAPTIVE algorithms eliminates this throughput decreasing by limiting the adaptability and by selecting non-minimal routes, therefore diminishing the maximum achievable throughput. However, it is possible to control the decay effect limiting the number of times a packet can leave the safe route (up*/down* or Hamilton). Depending on this limit, the maximum achievable performance will be greater, but the misrouting effect will also has more impact beyond the saturation point.

7 Conclusions

Bubble Routing for irregular networks means an important improvement over the classical up*/down* algorithm at, practically no extra cost. Avoiding the concentration of packets around the root node, the performance improvement can be as high as 10 times for a 256-node network under random traffic.

Bubble Routing even outperforms the adaptive up*/down* algorithm in spite of the fact that in our case it is not necessary to implement virtual channels in the router. The Bubble condition can be tested locally in the routers with a practically negligible hardware cost [4].

References

1. 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, vol. 7, no. 8, pp. 841-854, August 1996.
2. J. Duato, A. Robles, F. Silla and R. Beivide, "A comparison of router architectures for virtual cut-through and wormhole switching in a NOW environment", Proc. of 13th Int. Parallel Processing Symp., pp. 240-247, April 1998.
3. S. Konstantinidou and L. Snyder, "The Chaos Router", IEEE Trans. on Computers, Dec. 1994.
4. V. Puente, R. Beivide, J.A. Gregorio, J.M. Prellezo, J. Duato and C. Izu, Adaptive Bubble Router: a design to improve performance in torus networks, Proc. Of International Conf. On Parallel Processing, Japan, Sep. 1999.
5. M. D. Schroeder et al., "Autonet: A high-speed, self-configuring local area network using point- to-point links," Tech. Report SRC 59, DEC, April 1990.
6. F. Silla, M. Malumbres, A. Robles, P. López and J. Duato, "Efficient adaptive routing in networks of workstations with irregular topology", Proc. of the Workshop on Comm. and Arch. Support for Parallel Computing, pp. 46-60, Feb. 1997.
7. J.R. Jump, "YACSIM (Ver.2.1), NETSIM(Ver.1.0) Reference Manuals", Electrical & Computer Engineering Department, RICE University, March 1993.