A New Routing Mechanism for Networks with Irregular Topology

V. Puente, J.A. Gregorio, R. Beivide, F. Vallejo and A. Ibañez

***Abstract*-- Selecting a Pseudo-Hamiltonian cycle in any irregular network and applying a restricted packet injection mechanism to avoid the exhaustion of the storage resources, a new fully adaptive routing algorithm has been developed and tested. Our new routing mechanism outperforms the most relevant routing proposals for networks with irregular topology. In all the tested cases a significant improvement has been obtained. The most spectacular gains were obtained for big networks. For a 512-node network, uniform traffic, and virtual cut-through flow control, our mechanism can outperform, in some cases, the classic up*/down* algorithm by almost a factor of 2.**

***Index Terms*—routing algorithm, routers, virtual channels, Bubble method, irregular networks.**

I. 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. Autonet [8] or Myrinet [1] are examples of high-performance interconnection technologies derived from the convergence between local area networks and massively parallel multicomputers.

NOWs are normally organized as switched networks where each switch or router is shared among several workstations connected to it through its ports. The rest of the switch ports are used for interconnecting to other switches, facilitating network connectivity. Messages interchanged among nodes cross the network following paths fulfilling the rules of a routing algorithm and their advance will be in either virtual cut-through or wormhole mode.

In these networks, irregularity is a common characteristic that allows an easy design of scalable and flexible wired systems. However, 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 for visiting the network nodes, normally forming a "tree". Packets are routed using non-minimal paths, thus increasing latency and wasting resources.

In this paper we propose a new fully adaptive routing algorithm for irregular networks, which provides a very good cost/performance ratio. This new mechanism has been derived from a strategy firstly proposed for parallel computers using regular interconnection networks [6]. The new routing proposal selects a subset of physical links forming a Pseudo-Hamiltonian (PH) cycle. This PH cycle is made up of links connecting those nodes that could generate cyclic dependencies in the network. The purpose of this PH cycle is to act as an escape path for blocked packets. Subsequently, a technique based on a restricted injection policy will be used in order to avoid the exhaustion of the storage resources belonging to the PH-cycle. The resulting routing methodology outperforms any relevant routing proposals for NOWs with irregular topology.

The rest of the paper is organized as follows. The next Section reviews the essentials of classical and recent routing proposals for irregular networks. Section III presents the new routing mechanism. The simulation framework is described in Section IV and comparative results are shown in Section V. Finally, we present the main conclusions in Section VI.

II. Classical routing algorithms for irregular networks

Routing algorithms can be deterministic or adaptive. The former always provide 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 both cases, the routing algorithm can be minimal, if it only selects shortest paths towards the destinations, or non-minimal, if packets can follow routes moving away from the destination (misrouting).

In any case, every practical algorithm must provide deadlock-free routing. 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

This work was supported in part by the Spanish CICYT under Grant No. TIC-1162-C02-01.

The authors are with the University of Cantabria, Spain, e-mail: {vpuente,jagm,mon,fernando}@ atc.unican.es).

networks with different adaptability degrees.

A. Up*/Down* Algorithm

The up*/down* algorithm was first proposed for Autonet networks [8]. It is a distributed deadlock-free routing scheme that provides partial adaptability in irregular 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 arbitrarily 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 some adaptability is provided. The drawbacks are that the selected paths are generally not the shortest paths and that links near the root get congested and become bottlenecks leading to low throughput. Moreover, these problems become critical when the network size increases.

B. Adaptive Up*/Down* Algorithm

Recently, a general methodology for the design of adaptive routing algorithms for networks with irregular topology has been proposed in [9]. This methodology attempts not only to provide minimal routing between every pair of nodes, but also to increase 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. The routing algorithms designed with this methodology are deadlock-free provided that the original routing algorithm is deadlock-free [3].

Other recent routing algorithms, such as adaptive-trail routing [7] and smart-routing [2], showing certain improvements over the adaptive up*/down* algorithm, will be considered in forthcoming papers by the authors. Nevertheless, the significant performance improvement of our proposal seems to indicate that our algorithm could clearly outperform both of them.

III. PH ROUTING. A NEW PROPOSAL

In this section, a new fully adaptive routing algorithm for irregular networks under virtual cut-through flow control is proposed. The algorithm is based on avoiding storage exhaustion in all those physical links forming a Pseudo-Hamiltonian (PH) cycle. This PH cycle basically visits all the network nodes and its management must ensure deadlock-free communications. The combination of deadlock-free routing through this PH cycle, which acts as an escape path, together with other fully adaptive paths constitutes the basis of our proposal. First of all, we will establish the flow control mechanism to ensure deadlock-free communication in any network cycle and after that, the characterization of a PH cycle will be considered.

Bubble flow control

In an interconnection network, any cyclic dependency among resources (buffers) can give rise to deadlock among packets. Although traditionally, the way to avoid them was through virtual channels, recent results have shown that deadlock-free routing without using virtual channels can be successfully used in cycles (rings) of any size [6]. For avoiding packet deadlock, a restricted injection mechanism is applied to any packet that is trying to enter in this cycle. We usually denote this mechanism as Bubble Flow Control (BFC). The idea is simple. If under no circumstances the storage spaces for packets in a cycle are allowed to become full, the packets traveling along this cycle will always be able to advance.

To verify this situation, BFC imposes the contition that, for injecting a packet in the cycle there must be room for at least two packets, one for the packet itself and another one to maintain a hole or "Bubble" in the cycle. Once a packet is inside the cycle, for advancing it to the next router, space is only necessary for the packet itself, i.e. using classical virtual cut-through flow control.

Given that BFC guarantees that deadlock does not exist in any cycle, a cycle containing all the nodes of the network can be considered. This cycle can be used as an escape route to reach any destination node. Thus, if a packet situated in any buffer of any node in the network can reach this cycle, it will always have at least one safe path to get to its destination.

Obviously, this path around all the nodes only uses a subset of the network‘s links (and the associated buffers). The rest can be used by the packets to try to reach their destination adaptively.

At this point we should mention that this approach, although perfectly valid, is rather conservative. In fact, it is sufficient that any cyclic dependency that can occur shares a link to a safe path. In this way, it is also guaranteed that there will not be deadlock in any of them since this link would cut the corresponding cycle. Although in this analysis the conservative option is studied, it should be highlighted that there are other alternatives based on the same idea.

The PH cycle

A cycle for an irregular network going around all the nodes can be obtained in several ways. One of the most suitable ways corresponds to a Hamiltonian path. Therefore, given the graph representing an irregular network, any of the algorithms for determining a Hamiltonian path can be used.

Nevertheless, not all networks can embed a Hamiltonian path (thus the name Pseudo-Hamiltonian). Fortunately, there are other alternatives for finding a cycle that can act as an escape path. For example, any connected graph embeds a spanning tree. Hence, traversing, for example in preorder, all the network nodes would be visited forming a circuit. We can ensure that the number of visits to each node is always less than or equal to the node degree. Assuming that buffers for packets are located at the input links, the above spanning tree route never visits any buffer more than once. Therefore, in this way, a single cycle can always be obtained, ensuring that no sub-cycles exist in such a cycle.

In fact, there is an even simpler algorithm which can find a safe path to all the nodes. One eliminates, in any order, any link which does not separate the network in two parts until the nodes are left with minimal connectivity. Then, making use of the bidirectional character of the links, a cycle is created by simply going around all the network‘s nodes in a round trip.

For example, Figure 1 represents a simple irregular network. Firstly, it should be mentioned that those nodes (representing switches) forming open branches in the network, such as node number 5, do not have to belong to the escape cycle because messages stored in such nodes can never generate deadlock. Considering the rest of the nodes, there is a Hamiltonian path (0, 1, 4, 2 7, 6, 3, 0) represented by thick lines (really two paths given the bidirectionality of the links). Considering the simplest algorithm, an usefull circuit is shown in Figure 2 (0 3 5 3 6 7 1 7 2 7 4 7 6 3 0). Obviously, this path sometimes obliges the packet to go back over itself provoking a loss in performance as a consequence of its greater length, but it guarantees the existence of a cycle to form an escape path in any network and with any topology.

PH routing

The way the complete algorithm works is as follows. Links belonging to the PH cycle will be used as an escape route if there are no more choices for a packet traveling to its destination. The objective is to have a safe path between any pair of nodes of the network. BFC guarantees that the storage resources of the PH cycle are never exhausted and therefore packets inside the cycle can always advance.

Once a PH cycle is defined including all nodes that could generate cyclic dependencies, it is possible to achieve full adaptability 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. However, without using virtual channels, this mechanism can give rise to two anomalies: livelock[1] and starvation. The former can be eliminated easily giving preference to minimal paths over non-minimal ones and limiting the number (which can be very large) of times a packet can leave the PH-cycle. From this moment, the packet can no longer use adaptive paths and must follow the escape route until its destination. Nevertheless, the starvation of the packets trying to enter the PH cycle is more difficult to eliminate without using more complex injection mechanisms.

To obtain a routing mechanism absolutely free from anomalies, two virtual channels per physical link can be used. Links belonging to the PH cycle are shared between adaptive and escape channels. Virtual channels not belonging to links forming the PH cycle are handled as adaptive ones. Packets traveling through adaptive channels use minimal routes. The presence of two virtual channels eliminates the two previously mentioned anomalies. Livelock is eliminated in the same way: limiting the number of times a packet can abandon the escape route. Starvation does not exist since the packets from adaptive channels have the same priority as those from injection points.

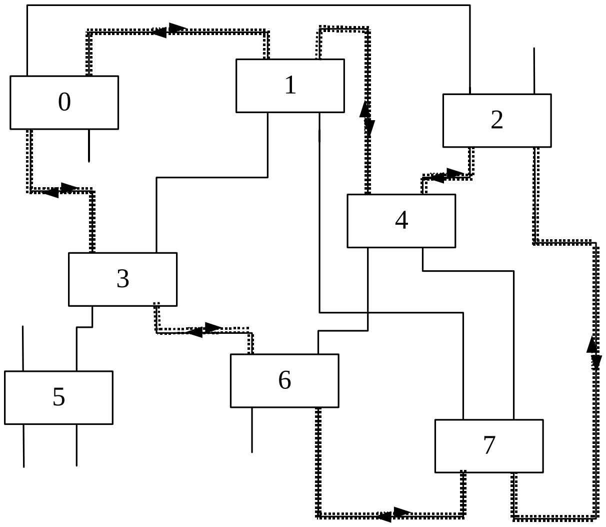


Fig. 1. Pseudo-Hamiltonian (PH) cycle in an irregular network.

It should be noted that the existence of a “Bubble” (space for a whole packet) must be guaranteed at any storing resource (buffer) belonging to the PH cycle. The location of such a bubble is irrelevant. In particular, the establishment of the bubble can be carried out locally, in the same switch where the routing decision is taking place [6]. This is an important hardware implementation aspect. For example, in Figure 1, a packet injected in router #6 towards router #7, must first ensure that there is room at the buffer of router #7 for the

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

packet itself. But the establishment of the bubble (room for another packet) can be done at the buffer of router #6, corresponding to the counterclockwise direction of the PH cycle. In this way, no centralized control is needed for maintaining the bubble because each router ensures no packet will enter into the PH cycle without fulfilling the BFC conditions.

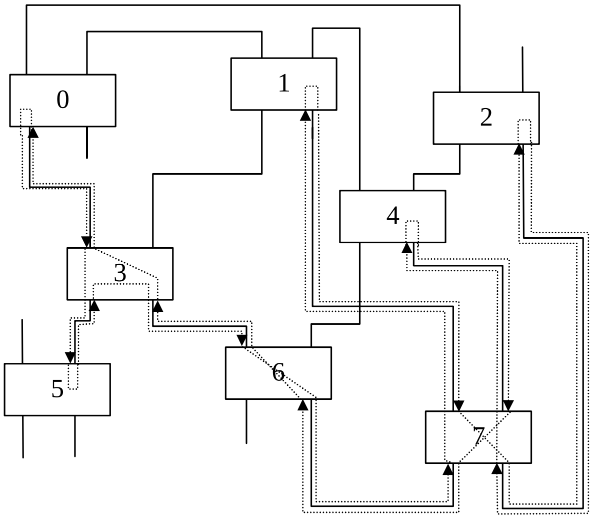


Fig. 2. Closest to worst-case Pseudo-Hamiltonian (PH) cycle in the irregular network of Figure 1.

IV. FRAMEWORK

Simulation techniques have been used to evaluate the performance achieved by the previously mentioned routing algorithms. A general-purpose interconnection network simulator, called SICOSYS, able to capture essential aspects of the low-level implementation has been employed [6]

SICOSYS is able to simulate a wide variety of packet routers in a precise way. Results are very close to those obtained by using hardware simulators but at lower computational cost. In order to make the tool easily comprehensible, extensible and reusable, the design of the tool is object-oriented and its implementation is in C++ language.

SICOSYS has been designed to construct and simulate any network structure (regular or irregular topologies). Users may also specify the number of input and output ports, buffer size, number of virtual channels per physical link, the node's processing capability, packet length, module delay, traffic load, ending simulation conditions, etc. Different traffic distributions (random, bit-reversal, hot spot, local, etc.) can also be adequately modeled. As output data, the simulator is able to collect information about message latency, network throughput, messages sent and received, average distance, buffering storage occupation and other relevant statistics.

A. Router Structures

In order to evaluate the performance of our technique, two router structures have been considered. The first one is the router model shown in Figure 3. It consists of an internal crossbar able to switch every input link to every output link simultaneously. This router does not employ virtual channels and it will be used to check the important effects they have.

The number of input and output ports is generally the same, and for simplicity, the temporary storage (buffers) is located at the input links. Nevertheless, the results of this work are not affected by buffer location.

The switch has a Routing Decision Unit (RDU), responsible for routing each incoming packet toward its destination, and an arbitration unit, responsible for 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, also taking into account the neighboring router status and the local crossbar utilization. If none of the requested output links is free, the packet will wait until one is released. To implement a fair procedure, the next packet to be selected, among those that are blocked, is chosen on the basis of a round-robin policy.

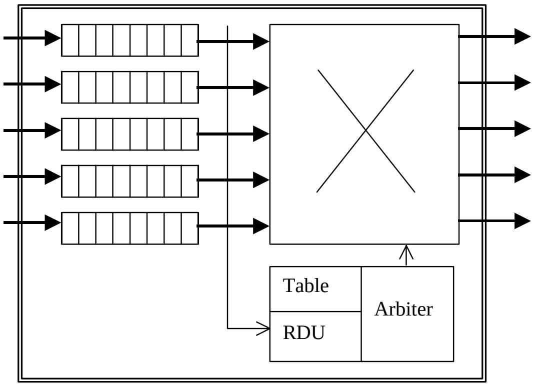


Fig. 3. Input-buffer router model without virtual channels.

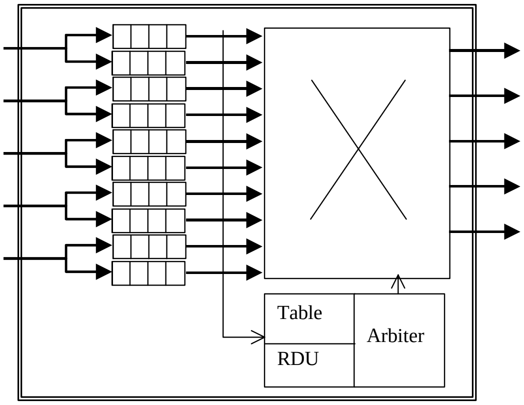


Fig. 4. Input-buffer router model with two virtual channels.

The second router structure considered is based on virtual channels, such as the one in Figure 4. As is known, this type of router allows several packets to share the same physical link in a multiplexed way. Thus, if a packet is blocked at an input buffer, another packet coming from the same physical link can advance through the other virtual channel provided that routing rules are fulfilled. This mechanism facilitates the design of deadlock-free routing algorithms and improves throughput significantly. 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. However, these different hardware characteristics have not been considered.

B. Basic Parameters

Network topology is irregular and randomly generated. These conditions represent the worst case. 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 those of Figure 3 or 4, with the same size, 5 input and 5 output ports. Also, there is one host connected to each router, thus leaving 4 ports available to connect to other routers. Finally, no more than one link can be used to connect two neighboring routers.

A virtual cut-through switching [4] technique is assumed in the simulations. Messages are one-packet length divided into 16 phits, obviously assuming that each phit can be transferred across one 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 the buffer capacity per physical link constant.

All the analyzed routing algorithms, either partially or fully adaptive, offer several routing choices. Therefore, all the algorithms require accessing to 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 decision in all cases. Also, one cycle is needed to transmit one phit across both the buffer and the crossbar respectively. Finally, another cycle is spent in traveling between routers.

Latency and throughput are the main performance metrics. Message latency lasts from when the packet is introduced into the network until the last phit is received at the destination node. Latency is measured in clock cycles. Throughput is the amount of information delivered per time unit and expressed in phits per cycle per switch.

V. Comparative results

In this section, a performance comparison of the routing algorithms described in Section II against the PH routing has been carried out for irregular networks of different sizes and connectivities. The resulting type of network is exactly like the one shown in Figure 1. In the maximal connectivity case, all the links are used. When the connectivity is partial, certain links remain open.

The traffic pattern of the applied load corresponds to a uniform distribution. Each node generates packets destined to all the other nodes of the network with equal probability. Although this traffic does not have any characteristics of locality, it is usual for evaluating the performance of the interconnection network under worst-case conditions.

Figure 5 shows the average packet latency versus the applied load for 64, 128 and 512 switches under uniform traffic and maximum connectivity (every router link is used). This Figure shows results for the same network under the up*/down* algorithm (UPDOWN), the two virtual channel adaptive up*/down* algorithm (UPDOWN-2vc) and under the two versions of the PH routing algorithm: without virtual channels (HAMILTON) and using two virtual channels per link (HAMILTON-2vc). This means that in the UPDOWN and HAMILTON cases we used the router represented in Figure 3 and in the other cases (UPDOWN-2vc and HAMILTON-2vc) we employed the router of Figure 4.

As Figure 5 shows, HAMILTON-2vc remarkably outperforms the classical UPDOWN strategy. As the network size increases this difference is even greater. This means that the new proposal not only presents a good scalability but also improves performance. The same applies for the throughput (phits/cycle/switch) the networks are able to support, as Figure 6 shows. In the cases not using virtual channels, HAMILTON leads to a significant gain compared to the UPDOWN alternative with no extra cost. Nevertheless, it leads to a strong loss in performance when the network reaches saturation levels. When virtual channels are used, HAMILTON-2vc again outperforms UPDOWN-2vc.

With the aim of finding out how the new routing proposal behaves when the network topology varies, the network connectivity was modified. Each switch still has one host attached to it, but not all the other 4 ports are used to provide connection between switches. As a particular case, Figure 6 shows latency and throughput values for the above-mentioned networks but with the following connectivity conditions: 20% of the switches are fully connected, 60% have three ports connected, 15% are connected to two other switches and the other 5% of the switches are only connected to one. As can be seen, the absolute values of latency and throughput are obviously worse because of the lower connectivity. But comparatively, the improvement of our proposal is maintained.

For all the cases, the reason for the improvement in performance is the better traffic distribution throughout the network. It is well-known that the principal problem of the up*/down* algorithm is the concentration of packets around the root node. The new routing technique specifically confronts this problem by distributing the packets throughout the network. This means that, although the average distance traveled by the packets is greater, the network supports an greatly increased level of traffic.

VI. Conclusions

The new PH routing mechanism for irregular networks leads to 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 twice for a 512-node network under random traffic. Consequently,it has been shown that it is very important to avoid the above-mentioned packet concentration although the route to be followed by the packets on the escape path is longer.

PH routing even outperforms the recently introduced adaptive up*/down* algorithm. Further comparisons with other schemes like smart-routing are in progress. Nevertheless, knowing the performance differences between this technique and up*/down*, we can conjecture that the PH strategy will also outperform the performance exhibited by other schemes.

REFERENCES

[1] N.J. Boden, D. Coben, R. E. Felderman, A. E. Kulawik, C. L. Seitz, J. Seizovic and W. Su, "Myrinet - A gigabit per second local area network", IEEE Micro, pp. 29-36, February 1995.

[2] L. Cherkasova, V. Kotov, and T. Rokicki, “Fibre channel fabrics: evaluation and design”, Proc. of the 29th Hawaii International Conference on System Sciencies, vol.1, pp. 53 – 62, Jan. 1996.

[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, vol. 7, no. 8, pp. 841-854, August 1996.

[4] P. Kermani and L. Kleinrock, "Virtual Cut-Through: a new computer communication switching technique", Computer Networks 3, pp. 267-286, 1979.

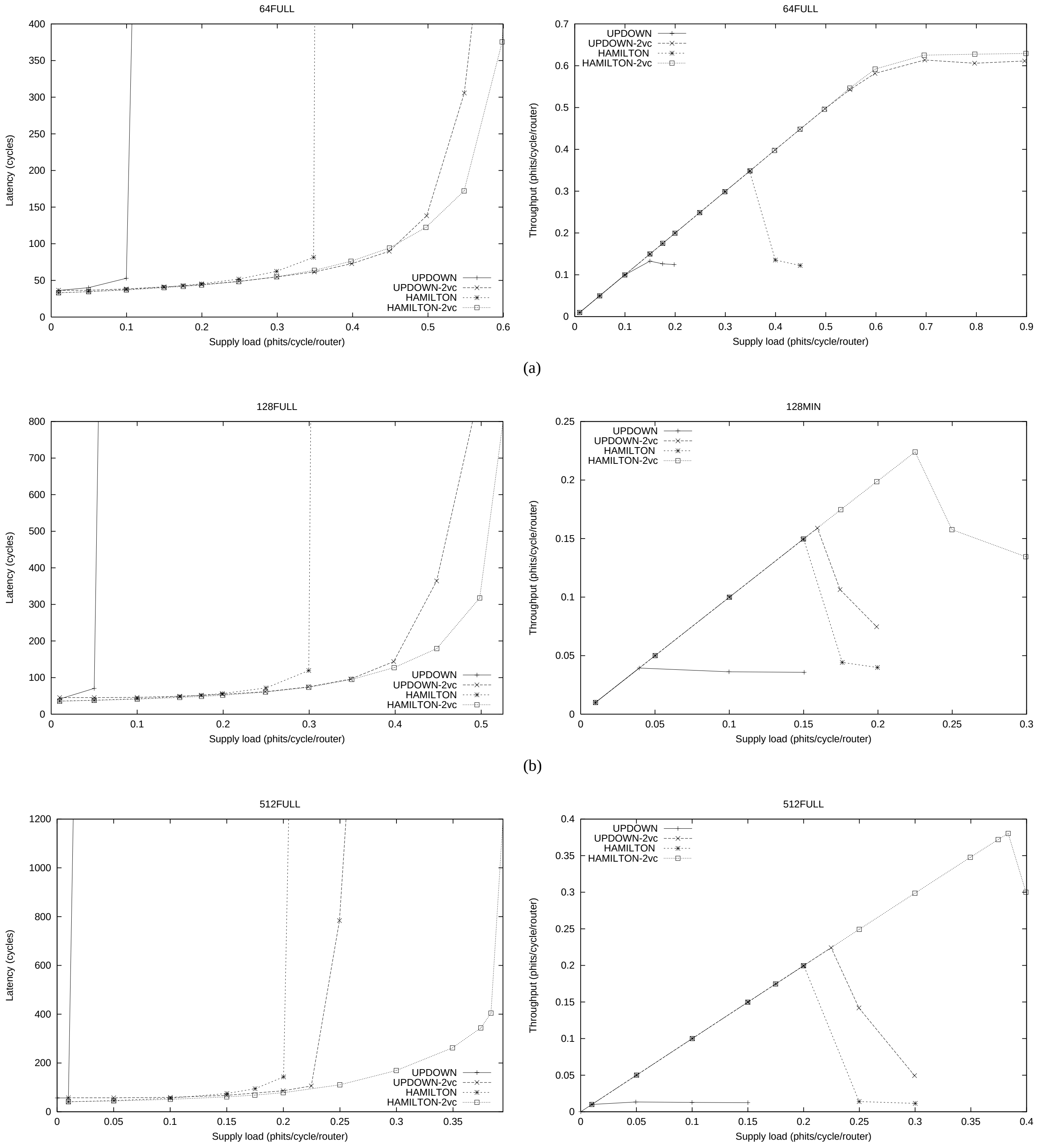
[5] S. Konstantinidou and L. Snyder, "The Chaos Router", IEEE Trans. on Computers, Dec. 1994.

[6] 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.

[7] W. Qiao; L.M Ni, and T. Rokicki,, “Adaptive-trail routing and performance evaluation in irregular networks using cut-through switches”. IEEE Transactions on Parallel and Distributed Systems, vol. 10, no. 11, pp. 1138 -1158, Nov. 1999.

[8] 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.

[9] F. Silla, M. Malumbres, A. Robles, P. López and J. Duato, "Efficient adaptive routing in networks of workstations with irregular topology", Proceedings of the Workshop on Communication and Architectural Support for Network-based Parallel Computing, pp. 46-60, Feb. 1997.

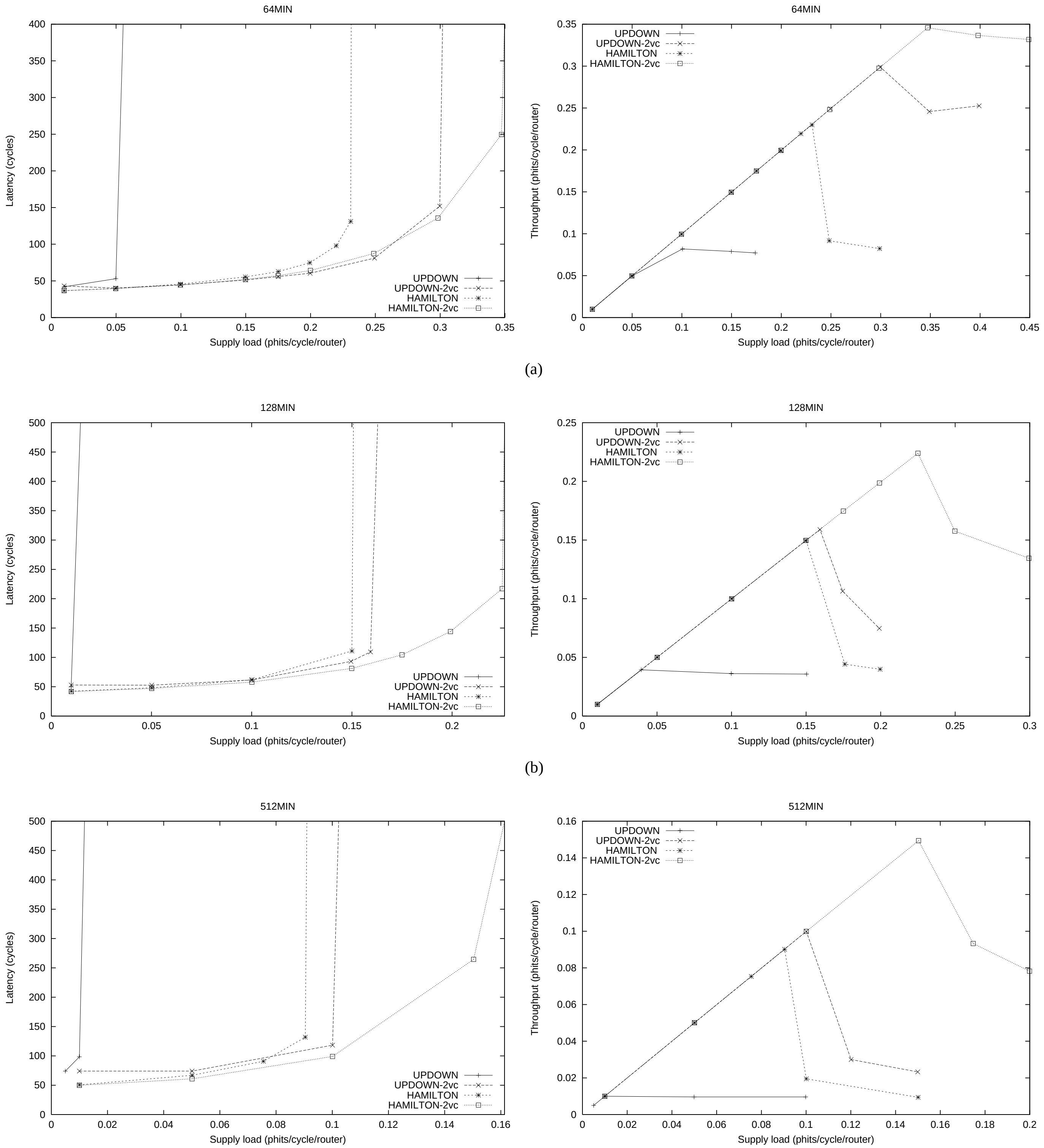


(a)

(b)

(c)

Fig. 5. Latency and Throughput under random traffic for a fully-connected network size of (a) 64-node network; (b) 128 nodes and (c) 512 nodes.



(a)

(b)

(c)

Fig. 6.- Latency and Throughput under random traffic for a partially-connected network size of (a) 64 nodes; (b) 128 nodes and (c) 512 nodes.