High-Performance Routing in Networks of Workstations with Irregular Topology

Federico Silla and José Duato, *Member*, *IEEE*

Abstract—Networks of workstations are rapidly emerging as a cost-effective alternative to parallel computers. Switch-based interconnects with irregular topology allow the wiring flexibility, scalability, and incremental expansion capability required in this environment. However, the irregularity also makes routing and deadlock avoidance on such systems quite complicated. In current proposals, many messages are routed following nonminimal paths, increasing latency and wasting resources. In this paper, we propose two general methodologies for the design of adaptive routing algorithms for networks with irregular topology. Routing algorithms designed according to these methodologies allow messages to follow minimal paths in most cases, reducing message latency and increasing network throughput.

As an example of application, we propose two adaptive routing algorithms for AN1 (previously known as Autonet). They can be implemented either by duplicating physical channels or by splitting each physical channel into two virtual channels. In the former case, the implementation does not require a new switch design. It only requires changing the routing tables and adding links in parallel with existing ones, taking advantage of spare switch ports. In the latter case, a new switch design is required, but the network topology is not changed. Evaluation results for several different topologies and message distributions show that the new routing algorithms are able to increase throughput for random traffic by a factor of up to 4 with respect to the original up*/down* algorithm, also reducing latency significantly. For other message distributions, throughput is increased more than seven times. We also show that most of the improvement comes from the use of minimal routing.

Index Terms—Networks of workstations, irregular topologies, wormhole switching, deadlock avoidance, adaptive routing.

1 Introduction

Wormhole switching [8] has become the most widely used switching technique for multicomputers and distributed shared-memory multiprocessors (DSMs). Wormhole switching only requires small buffers in the routers through which messages are routed. Also, it makes message latency largely insensitive to the distance in the network. See [30] for a detailed analysis of wormhole.

The main drawback of wormhole switching is that blocked messages remain in the network, therefore wasting channel bandwidth and blocking other messages. In order to reduce the impact of message blocking, physical channels may be split into virtual channels by providing a separate buffer for each virtual channel and by multiplexing physical channel bandwidth. The use of virtual channels can increase throughput considerably by dynamically sharing the physical bandwidth among several messages [9]. However, it has been shown that virtual channels are expensive, increasing node delay [7].

An alternative but complementary approach to improve network performance consists of using adaptive routing [20]. This routing strategy provides alternative paths to route messages, thus avoiding congested regions in the network and increasing throughput. However, deadlocks may appear if the routing algorithms are not carefully designed. A deadlock occurs in an interconnection network when no message is able to advance toward its destination because network buffers are full. As routing decisions must be taken in a few nanoseconds in wormhole routers, a practical way to avoid deadlock is to design deadlock-free routing algorithms. A simple and effective approach to avoiding deadlocks consists of restricting routing so that there are no cyclic dependencies between channels [8]. A more efficient approach consists of allowing the existence of cyclic dependencies between channels while providing some escape paths to avoid deadlock [14], [15]. The resulting routing algorithms are much more flexible, usually increasing performance. As a consequence, some recent router implementations, like the MIT Reliable Router [11], [12] and the Cray T3E router [35], are based on these techniques, implementing escape paths as dedicated virtual channels. Additionally, there has been considerable interest on these issues. Several researchers developed alternative theories of deadlock avoidance, proposing sufficient conditions [2], [26], [37], [22] and necessary and sufficient conditions [16], [17], [34], [27] for deadlock-free adaptive routing.

Networks of workstations (NOWs) have received considerable attention during recent years. Due to the high cost of parallel computers and the rapidly increasing computation power of microprocessors, NOWs are becoming a cost-effective alternative to parallel computers. In this way, NOWs behave more like distributed multicomputers and DSM machines spread around entire buildings than like simple LANs. NOWs may not provide the closely coupled environment of multicomputers and multiprocessors. However, they provide a similar programming environment and

• *The authors are with the Grupo de Arquitecturas Paralelas, Dpto. Informática de Sistemas y Computadores, Universidad Politécnica de Valencia, Escuela Universitaria de Informática, Camino de Vera, 46022, Valencia, Spain. E-mail: {fsilla, jduato}@gap.upv.es.*

Manuscript received 9 Sept. 1998; revised 8 Nov. 1999; accepted 21 Dec. 1999. For information on obtaining reprints of this article, please send e-mail to: tpds@computer.org, and reference IEEECS Log Number 107367.

meet the needs of a great variety of parallel computing problems at a lower cost (see [1] as an example). Additionally, they provide an efficient and more natural environment for some highly demanding applications, like distributed multimedia databases.

Recently, several switch-based interconnects like Autonet[1] [33], Myrinet [4], ServerNet [23], and ServerNet II [21] have been proposed to build networks of workstations. Typically, these switches support networks with irregular topologies. Such irregularity provides the wiring flexibility required in local area networks, also allowing the design of scalable systems with incremental expansion capability. The irregular connections between switches also make routing and deadlock avoidance on these networks quite complicated. Current proposals avoid deadlock by removing cyclic dependencies between channels. As a consequence, routing is considerably restricted and many messages must follow nonminimal paths, thereby increasing latency and wasting resources. Thus, more powerful routing strategies are required for NOWs. These strategies should route messages along minimal paths and take advantage of alternative paths available in the network while being flexible enough to be implemented on any topology.

In this paper, we take such a challenge. In order to increase the adaptivity of routing algorithms for irregular networks, the design methodology presented in [14], [15] for regular networks could be applied to irregular ones, routing messages along minimal routes and using escape paths to avoid deadlock. However, that methodology cannot be directly applied to irregular topologies. In this paper, we propose two general methodologies for the design of adaptive routing algorithms for networks with irregular topology. The first one is very similar to the methodology proposed in [14], [15], but modified in such a way that it guarantees deadlock freedom for any topology. The second methodology refines the first one, reducing the amount of messages that follow nonminimal paths. As an application of the new design methodologies, we use them to derive two adaptive routing algorithms for Autonet networks. These algorithms increase adaptivity while allowing messages to follow minimal paths in most cases, reducing message latency and increasing network throughput. The proposed routing algorithms do not require a new switch design. They can be implemented simply by changing the routing tables and adding links in parallel with existing ones, taking advantage of spare switch ports. They can also be implemented by splitting physical channels into virtual ones. In this case, the network topology remains unaltered. However, this approach would require a new switch design supporting virtual channels. We will present a detailed performance study of the proposed routing algorithms, comparing them with the up*/down* routing scheme used in Autonet networks. Also, we will analyze in depth the proposed routing algorithms in order to know the reasons for the performance improvement over up*/down* routing. Finally, in this paper, we analyze the influence of the variations in the network topology on the performance of the different routing algorithms.

1. Also called AN1.

The rest of the paper is organized as follows: Section 2 introduces networks of workstations, focusing on the interconnection network. Section 3 describes two general methodologies for the design of adaptive routing algorithms for networks with irregular topology. As an example, these methodologies are applied to the Autonet routing algorithm. Section 4 describes the routing algorithm proposed for Autonet networks, showing how its performance can be improved by applying the new methodologies in Section 5. The routing algorithms designed according to the proposed methodologies are evaluated in Section 6. Finally, some conclusions are drawn.

2 Networks of Workstations

Networks of workstations are usually arranged as switch-based networks, which consist of a set of switches, each one having several ports. Each port provides bidirectional connectivity through an external link. A set of ports in each switch are either connected to different workstations or left open, whereas the remaining ports are connected to ports in other switches to provide connectivity between the workstations. Such connectivity is typically irregular and the only thing that is guaranteed is that the network is connected. Typically, all links are bidirectional full-duplex and multiple links between two switches are allowed. Such a configuration allows a system with a given number of workstations to be built using fewer switches than a direct network, while allowing a reasonable number of external communication ports per switch. Fig. 1a shows a typical NOW using a switch-based interconnect with irregular topology. In this figure, it is assumed that switches have eight ports and each workstation has a single port.

Switches may implement different switching techniques, like wormhole, virtual cut-through, or ATM. However, wormhole switching is used in recently proposed networks like Myrinet [4] and ServerNet [23]. So, we will restrict ourselves to wormhole switching in this paper. Nevertheless, the design methodologies proposed in this paper are also valid for packet switching and virtual cut-through. Several deadlock-free routing schemes have been proposed in the literature for irregular networks [33], [4], [23], [32]. Routing in irregular topologies can be performed by using source routing or distributed routing. In the former case, each workstation has a routing table that indicates the sequence of ports to be used at intermediate switches to reach the destination. That information is stored in the message header [4]. In the latter case, switches require routing tables. These tables store the alternative output ports that can be taken by the incoming message. Table lookup routing allows the use of the same switch fabric for different topologies. However, some network mapping algorithm must be executed in order to fill those tables before routing can be performed. The details of the mapping algorithm greatly depend on the underlying hardware support.

Once a message reaches a switch directly connected to its destination workstation, it can be delivered as soon as the corresponding link becomes free. Thus, we are going to

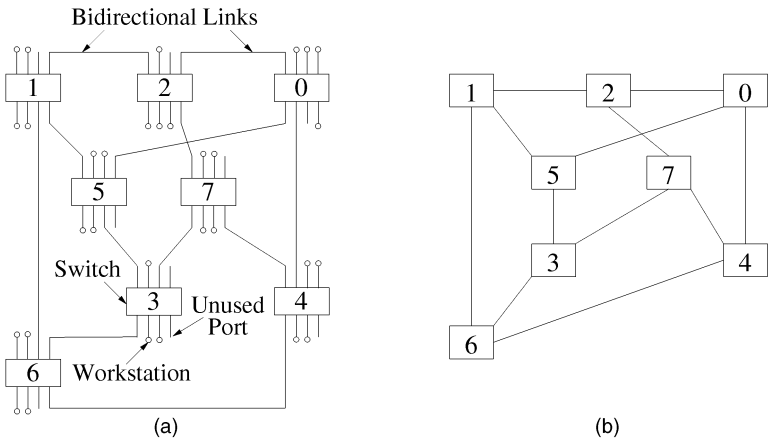


Fig. 1. (a) A network of workstations with switch-based interconnect and irregular topology; (b) the corresponding graph G.

focus on routing messages between switches, modeling the interconnection network I by a multigraph $I = G(N, C)$, where N is the set of switches, and C is the set of bidirectional links between switches. Fig. 1b shows the graph for the irregular network in Fig. 1a.

3 Design Methodologies for Adaptive Routing Algorithms

In this section, we propose two general methodologies for the design of adaptive routing algorithms for networks with irregular topology. The first one is similar to the methodology proposed in [14], [15]. However, it restricts the use of adaptive channels to messages that did not use escape paths at previous routing steps. This constraint guarantees deadlock freedom on any topology, as will be seen later. The resulting design methodology increases adaptivity while allowing messages to be routed following minimal paths in most cases. The second design methodology refines the first one in order to reduce the amount of messages that follow nonminimal paths.

3.1 Increasing Adaptivity

The first design methodology is very simple. Given an interconnection network and a deadlock-free routing function defined on it, it is possible to duplicate all the physical channels in the network, taking advantage of spare switch ports.[2] Alternatively, it is possible to split physical channels into two virtual channels. In both cases, the graph representation of the network contains the original and the new channels. Then, the routing function is extended in such a way that newly injected messages can use new channels without any restriction as long as the original channels are used exactly in the same way as in the original routing function. New channels can be used for fully adaptive minimal (or nonminimal) routing. Minimal routing usually provides better performance because messages occupy fewer resources on average. Messages coming through a new channel can reserve either new or original channels. However, once a message reserves one of the original channels, it can no longer reserve any of the new channels. The latter constraint is the main difference between the methodology proposed in [14], [15] and the one proposed here. This constraint drastically simplifies the proof of deadlock freedom. Moreover, as will be seen in Section 5, deadlock may occur if that constraint is not enforced.

Deadlock-freedom can be proven informally by contradiction as follows: Suppose that there is a deadlocked configuration. In this configuration, messages cannot occupy the original channels because the original routing function is deadlock-free, and once a message has entered the set of original channels, it cannot return to new channels. Thus, all blocked messages must occupy new channels. However, those messages can escape from deadlock by using the original channels because the original routing function is able to deliver messages from any source to any destination. Therefore, there cannot be a deadlocked configuration. Deadlock freedom can be formally proven by using the theory proposed in [15].

This design methodology is valid for any topology. It is also valid regardless of whether the additional channels are physical or virtual ones. This routing strategy is similar to the one proposed in [10], except that it is valid for any topology and it does not require keeping track of the number of dimension reversals.

3.2 Providing Minimal Paths

According to the extended routing function presented in the previous section, new channels provide more routing flexibility than original channels. Moreover, they can be used to route messages through minimal paths. However, once a message reserves an original channel, it will be routed only through original channels until delivered. This may degrade performance because original paths are not

2. The cost of duplicating physical channels depends on the initial network configuration. In the case for a network containing switches with spare ports, the cost would be that of cables. For example, in a 16-switch Myrinet network built by using the new 16-port switches, where four ports are used to connect to other switches and four ports are used to connect to workstations, the cost of duplicating all the physical channels would be $4,480.00 (October 1999 prices). However, in the case for a network with no spare switch ports, the increment in cost is higher because larger switches should be acquired. If the 16-switch Myrinet network in the previous example had switches with eight ports, the cost of duplicating all the physical channels would be $84,480.00 (replacing 8-port switches with 16-port switches).

minimal in most cases. Also, routing through original channels produces a loss of adaptivity.

Following this reasoning, the general methodology proposed in the previous section can be refined by restricting the transition from new channels to original channels, since the latter provides less adaptivity and nonminimal paths. In particular, the extended routing function can be redefined in the following way. Newly injected messages can only leave the source switch using new channels belonging to minimal paths, and never using original channels. When a message arrives at a switch from another switch through a new channel, the routing function gives a higher priority to the new channels belonging to minimal paths. If all of them are busy, then the routing algorithm selects one of the original channels belonging to minimal paths (if any). This original channel will be used as an escape path. To ensure that the new routing function is deadlock-free, if none of the original channels provides minimal routing, then the original channel that provides the shortest path will be used, guaranteeing that at least one escape path exists at each switch. In case several original channels provide shortest paths, only one of them will be supplied by the routing function. Once a message reserves an original channel, it will be routed using these channels according to the original routing function until it is delivered. By restricting the use of original channels in this way, we allow most messages to follow minimal paths, and therefore, a more efficient use of resources is achieved. Note that routing algorithms designed according to the methodology described in Section 3.1 provide greater adaptivity, while with this new design methodology we try to restrict routing to only minimal paths, reducing adaptivity at the current switch. Nevertheless, this methodology may increase adaptivity at the remaining switches by keeping messages on new channels.

The improved design methodology also supplies deadlock-free routing algorithms, assuming that the original routing function is deadlock-free. The proof of deadlock freedom differs from the one for the methodology proposed in the previous section, because newly injected messages can only leave the source switch using new channels. The following theorem formally proves deadlock freedom.

Theorem 1. *The improved design methodology supplies deadlock-free routing functions.*

Proof. We proceed by contradiction. Suppose that there is a deadlocked configuration. In this configuration, each message is waiting for channels occupied by other messages in the configuration. However, messages in the configuration cannot occupy original and new channels, because once a message reserves an original channel, it cannot request a new channel again. Also, messages cannot occupy only original channels because the original routing function is deadlock-free. Therefore, all the blocked messages must occupy new channels. However, those messages can escape from deadlock by using the original channels because the original routing function is able to deliver messages from any switch to any destination. The only exception is that newly injected messages can only leave the source switch using new channels. Thus, in a configuration consisting of newly injected messages only, no message is able to escape from deadlock through original channels. However, if one of those messages is waiting for a new channel, it is occupying a channel directly connected to a workstation. Therefore, no other message in the configuration can request the channel it occupies and that message is not involved in the deadlocked configuration. Thus, there is no deadlocked configuration, contrary to the initial assumption. □

This result is valid for any topology and any original deadlock-free routing algorithm. Deadlock freedom also can be proven by using the theory proposed in [15].

4 The Autonet Routing Algorithm

As an example, we are going to apply the design methodologies proposed in Section 3 to Autonet networks. For this purpose, we first describe the routing algorithm proposed for those networks. The routing scheme used in Autonet, commonly known as up*/down* routing [33], is deadlock-free. It provides partially adaptive communication between the nodes of an irregular network. The Autonet routing algorithm is distributed, and implemented using table-lookup. When a message reaches a switch, the destination address stored in the message header is concatenated with the incoming port number, and the result is used to index the routing table at that switch. The table lookup returns the outgoing port number that this message should be routed through. When multiple valid routes exist from the current switch to the destination, the routing table returns all the alternative outgoing ports. In the case that multiple outgoing ports are free, the routing scheme selects the one with the lower identifier.

The routing table in each switch must be filled before messages can be routed. To do so, a breadth-first spanning tree (BFS) on the graph G is computed first using a distributed algorithm. This algorithm has the property that all the switches in the network will eventually agree on a unique spanning tree. Routing is based on an assignment of direction to the operational links, including the ones that do not belong to the tree. In particular, the "up" end of each link is defined as: 1) the end whose switch is closer to the root in the spanning tree; 2) the end whose switch has the lower ID, if both ends are at switches at the same tree level (see Fig. 2). Links looped back to the same switch are omitted from the configuration. The result of this assignment is that each cycle in the network has at least one link in the "up" direction and one link in the "down" direction.

To eliminate deadlocks while still allowing all links to be used, the up*/down* routing algorithm uses the following rule: a legal route must traverse zero or more links in the "up" direction followed by zero or more links in the "down" direction. Thus, cyclic dependencies between channels are avoided because a message cannot traverse a link along the "up" direction after having traversed one in the "down" direction. Such routing not only guarantees deadlock-freedom, but also provides some adaptivity. The lookup tables can be constructed to support both minimal and nonminimal adaptive routing. However, in some cases, up*/down* routing is not able to supply any minimal path

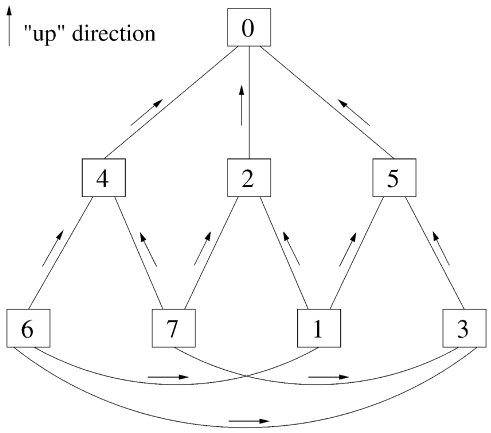


Fig. 2. Link direction assignment for the network in Fig. 1.

between some pairs of switches, as shown in the following example.

Fig. 2 shows the example irregular network shown in Fig. 1a. Switches are arranged in such a way that all the switches at the same tree level are at the same vertical position in the figure. The root switch for the corresponding BFS spanning tree is switch 0. The assignment of "up" direction to the links in the network is illustrated. The "down" direction is along the reverse direction of the link. Note that every cycle has at least one link in the "up" direction and one link in the "down" direction. It can be observed that all the alternative minimal paths are allowed in some cases. This is the case when the destination host is connected to the root switch. For example, a message transmitted from switch 7 to switch 0 can be routed either through switch 4 or switch 2. In some other cases, however, only some minimal paths are allowed. For example, a message transmitted from switch 2 to switch 5 can be routed through switch 0 but it cannot be routed through switch 1. It should be noted that any transmission between adjacent switches is always allowed to use the link(s) connecting them, regardless of the direction assigned to that link. However, when two switches are located two or more links away, it may happen that all the minimal paths are forbidden. This is the case for messages transmitted from switch 4 to switch 1. The only minimal path (through switch 6) is not allowed, because it requires one transition from "down" to "up" direction. All the allowed paths (through switches 0, 2, and through switches 0, 5) are nonminimal since they require three hops, while the illegal path through switch 6 requires only two hops.

This problem with minimal paths becomes more important as network size increases. In general, up*/down* concentrates traffic near the root switch, providing minimal paths only between switches that are located near the root switch. In most cases, only nonminimal paths are provided between nonadjacent switches located far from the root switch. Thus, the percentage of nonminimal paths increases with network size. Additionally, the concentration of traffic in the vicinity of the root switch produces a premature saturation of the network near the root switch, thus reducing network throughput. Also, it leads to uneven channel utilization.

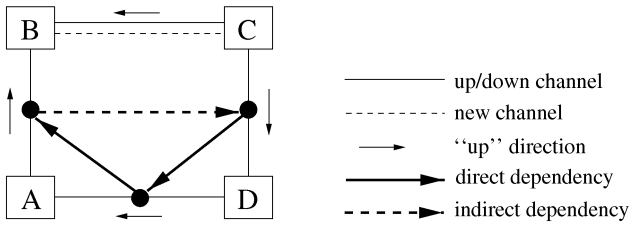


Fig. 3. Extended channel dependency graph.

5 Applying the Design Methodologies

The routing scheme used in Autonet networks allows partially adaptive communication between switches. However, as seen in the previous section, it does not always supply minimal paths. Some minimal paths are not allowed because messages traveling along them should go through a link in the "up" direction after having traversed a link in the "down" direction, contrary to the basic routing rule. This "down" to "up" conflict may occur more than once in these minimal paths.

In this section, we present two routing algorithms that allow messages to use these forbidden routes. These algorithms also increase adaptivity and throughput. They are based on the methodologies proposed in Section 3. We can follow two different approaches. In the first approach, physical channels are not split into virtual channels. Instead, the channels in the network are duplicated, taking advantage of spare switch ports. This approach requires adding more wires, but it does not require a new switch design. In the second approach, physical channels are split into virtual channels. This approach does not require more wires, but current switches do not support virtual channels. In both cases, the additional channels will be used to circumvent the "down" to "up" conflicts that prevent routing using minimal paths. For example, in Fig. 2, messages going from switch 4 to switch 1 would be able to arrive at their destination with only two hops through switch 6, instead of using the three-hop paths provided by the up*/down* algorithm. Once channels have been either duplicated or split into virtual channels, the routing algorithm must be extended in order to use the new channels.

New channels are directionless and can be used in both directions, as a shortcut. However, if directionless channels were used without restrictions, the new routing algorithm might not be deadlock-free, as can be seen in Fig. 3. This figure shows part of the extended channel dependency graph [15] for a fragment of an irregular network consisting of four switches. The "up" direction of the channels is also displayed. A new directionless channel connecting switches B and C has been added (shown with a dashed line). If a message is routed through the "up" channel from A to B, the new directionless channel from B to C, and the "up" channel from C to D, it produces the indirect dependency shown in the figure. Thus, there is a cycle in the extended channel dependency graph. Hence, a routing algorithm that imposes no restriction on the use of new channels may not be deadlock-free. This is the reason why both methodologies in Section 3 do not allow messages to be routed

through new channels after having been routed through the original ones.

5.1 Applying the First Methodology: Increasing Adaptivity

This methodology is intended to overcome the routing restrictions resulting from removing the cyclic dependencies between channels to avoid deadlock. We are interested in providing alternative routing choices that allow messages to use the paths that the Autonet routing scheme does not allow. At the same time, this design methodology considerably increases adaptivity.

When applying the design methodology presented in Section 3.1 to the up*/down* routing scheme, the new routing algorithm can be stated as follows: Original (up*/down*) channels can be used according to the up*/down* rule revisited in Section 4. New channels can only be used if the message arrives at the switch through a new channel or if the message is injected into the network at the current switch. Additionally, a new channel can be used by a message to leave the current switch only if the routing distance from the next switch up to the destination is shorter than the routing distance from the current switch to the destination. This rule prevents messages from being routed through paths that are longer than the original up*/down* paths. Once a message has reserved an "up" or "down" channel, it cannot use any of the new channels again. Therefore, when the routing table provides both, up*/down* and new channels, a higher priority has been assigned to the new ones, since they allow messages to be routed with higher flexibility.

Once channels have been duplicated or split into virtual channels, new channels can be used by messages traveling to any destination. For example, in Fig. 2, messages traveling from switch 5 to switch 6 can be transmitted through switch 1 using new channels until they reach their destination. Messages can also move to the up*/down* channel in the hop from switch 1 to switch 6, or they can even reach their destination through up*/down* channels from their source. The same occurs for messages from switch 6 to switch 5. Routing messages in this way increases adaptivity considerably, since the number of possible paths between two switches increases.

As indicated in Section 3.1, the proposed routing scheme is deadlock-free. It is also livelock-free. Effectively, we know that the up*/down* routing scheme revisited in Section 4 is livelock-free. New channels can only be used by a message if it has arrived at the switch through one of the new channels. Additionally, new channels are used to forward messages along paths that are not longer than the basic up*/down* paths. Every time a message advances through a new channel, it gets closer to its destination. Therefore, the new routing scheme is livelock-free.

It is important to note that adaptivity does not increase the complexity of the switch. As switches become faster, they tend to be simpler, and therefore adaptivity could introduce an additional complexity, reducing the maximum performance the switch could achieve. However, since our adaptive routing algorithm is implemented using table-lookup, decisions are taken by looking up in a programmable memory. Thus, modifying the routing algorithm in order to introduce adaptivity does not add any extra delay, since the basic operation (accessing a table) is the same, and routing tables are programmed at the network initialization phase.[3] With respect to the selection of a single output channel from a set of candidates, since up*/down* routing is partially adaptive, it requires making a selection in case that the routing table returns several outgoing channels. When the degree of adaptivity is increased by applying this design methodology, the selection function has to select among a larger set of choices, increasing its delay. However, the delay of the new selection function can be reduced and be almost as small as the one for up*/down* routing if it is implemented efficiently. We can divide the selection function into two concurrent circuits: one to select among original channels and the other one to select among new channels. Each of these two circuits has the same delay as the up*/down* selection function. The only extra delay introduced by our selection function is a multiplexer to select from the set of new channels or the set of original channels, depending on the availability of each of them.

5.2 Applying the Refined Methodology: Providing Minimal Paths

The algorithm that results from the application of the first design methodology to the Autonet routing scheme increases adaptivity considerably. However, most messages are routed through nonminimal paths. On the contrary, routing algorithms designed according to the refined methodology allow most messages to follow minimal routes.

When applying the design methodology of Section 3.2 to the Autonet routing scheme, the original up*/down* routing algorithm must be extended in order to use the new channels efficiently. The new routing function is a direct application of the design methodology, and can be stated as follows: When a message enters the network, it can only leave the source switch by using those new channels that provide a minimal path toward the destination. At intermediate switches, messages arriving on new channels are routed through those outgoing channels (new or original ones) that provide minimal routing. However, once a message reserves an original channel, it can no longer reserve a new one, being routed according to the original up*/down* routing algorithm until delivered. Thus, when the routing table provides both up*/down* and new channels, a higher priority is assigned to the new ones, since they allow messages to follow minimal paths and also allow them to be routed with higher adaptivity. In case all the new channels belonging to minimal paths are busy and none of the original channels offers a minimal route, the original channel belonging to the shortest path will be provided by the routing table. This guarantees that one escape path exists at each switch. In case several outgoing

3. Following the definitions proposed in [15], we distinguish between routing (providing a set of candidate output channels) and selection function (selecting a single candidate based on output channels status).

original channels belong to shortest paths, only one of them will be provided by the routing table. By restricting the use of original channels in this way, we allow most messages to follow minimal paths, and therefore, a more efficient use of resources is made. Also, adaptivity is increased with respect to the original Autonet routing algorithm. This new routing algorithm is inherently different from the one proposed in the previous section, since that algorithm allows the use of original channels to leave the source switch and does not limit their use at intermediate switches to only those original channels that provide minimal or shortest paths. Moreover, that routing algorithm provides a greater adaptivity, while in this new algorithm we try to restrict routing to only minimal paths, thus reducing adaptivity. We would like to remark that this routing algorithm provides fully adaptive minimal routing between all pairs of workstations until messages are forced to move to original channels. When a message starts using original channels, both algorithms provide the same adaptivity as the up*/down* routing scheme.

Regarding the complexity of the switch, this new routing algorithm does not introduce any extra delay, since the basic routing operation is the same (looking up in a table). With respect to the selection function for this new routing algorithm, it may be implemented as explained in the previous section. In fact, the selection function is exactly the same, the only difference between both algorithms being the information stored in the routing table.

As indicated in Section 3.2, the proposed routing scheme is deadlock-free. It is also livelock-free. Effectively, new channels can only be used if a message has arrived at a switch through one of the new channels. Additionally, new channels are only used to forward messages along minimal paths. Therefore, the number of new channels that can be crossed by a message is bounded by the diameter of the network. Moreover, the up*/down* routing scheme revisited in Section 4 is livelock-free. Thus, the new routing scheme is livelock-free.

Finally, a variation of the new routing algorithm could be as follows: At intermediate switches, instead of routing through an original channel when all the new channels belonging to minimal paths are busy, we could reduce the use of original channels even more by routing the message through a new channel belonging to a nonminimal path that conforms to the up*/down* rule. That is, we could route the message through the virtual channel labeled as new that belongs to the same physical channel as the escape path (when duplicating links, it would be routed through the new duplicated channel). If this new channel is also busy, then try the original channel (escape path). However, this routing algorithm increments the complexity of the selection function.

6 Performance Evaluation

In this section, we evaluate the performance of the routing algorithms proposed in Section 5. We will refer to the first routing scheme, proposed in Section 5.1, as A-2 ("A" for adaptive) when implemented by duplicating physical links. Since the routing scheme proposed in Section 5.2 provides minimal adaptive routing, we will refer to it as MA-2 when links are duplicated. We have also evaluated the up*/down* routing scheme for comparison purposes. We will refer to this routing scheme as UD.

As both algorithms, A-2 and MA-2, require the duplication of all the physical channels in the network, twice as much network bandwidth is available for them with respect to the UD scheme. Thus, in order to perform a fair comparison, a third routing scheme was evaluated. This scheme also duplicates all the channels. The additional channels are used in exactly the same way as the original ones. Both the original and the new channels can be used interchangeably, regardless of the channel reserved at the previous switch. We will refer to this routing scheme as UD-2.

As mentioned in Section 3, an alternative approach consists of splitting physical channels into two virtual channels. We have also considered this option. When the A-2, MA-2, and UD-2 routing algorithms are implemented by splitting each physical channel into two virtual channels, the corresponding routing schemes will be referred to as A-2vc, MA-2vc, and UD-2vc, respectively. Note that these routing algorithms do not duplicate physical channels. Thus, total network bandwidth for the A-2vc, MA-2vc, and UD-2vc routing algorithms is the same as for the UD routing scheme.

Instead of analytic modeling, simulation was used to evaluate the routing algorithms. Our simulator models the network at the flit level. The evaluation methodology used is based on the one proposed in [15]. The most important performance metrics are latency and throughput. Message latency lasts since the message is introduced in the network until the last flit is received at the destination workstation. Latency is measured in clock cycles. Traffic is the flit reception rate, measured in flits per cycle per switch. When the network is not saturated, received traffic will be equal to injected traffic. However, when the network is beyond saturation, received traffic will be lower than the injected one. Moreover, we define throughput as the maximum amount of information delivered by the network per time unit (maximum received traffic).

Message latency should also include software overhead at the source and destination workstations (system call, buffer allocation, buffer-to-buffer copies, etc.). This overhead traditionally accounted for a high percentage of message latency [28], [24]. However, we did not consider such an overhead because some recent proposals reduce and/or hide that overhead, thus exposing hardware latency [18], [3], [25], [13]. We only considered network hardware latency in this study.

6.1 Switch Model

Each switch has a routing control unit that selects the output channel for a message as a function of its destination workstation, the input channel and the status of the output channels. Table look-up routing is used. UD, UD-2, A-2, MA-2, UD-2vc, A-2vc, or MA-2vc routing strategy can be chosen. The routing control unit can only process one message header at a time. It is assigned to waiting messages in a demand-slotted round-robin fashion. When a message gets the routing control unit but it cannot be routed because all of the feasible output channels are busy, it must wait in the input buffer until its next turn. A crossbar inside the switch allows multiple messages to traverse it simultaneously without interference. It is configured by the routing control unit each time a successful routing is made. As all the routing algorithms analyzed in this paper are partially or fully adaptive, all of them offer several routing choices. Therefore, in order to execute the routing algorithm, all the algorithms require accessing a routing table, selecting among several options, and arbitrating for the selected output channel. Thus, we assumed that it takes one clock cycle to compute the routing algorithm in all the cases. We also assumed that it takes one cycle to transmit one flit across the crossbar or through a physical channel. With respect to the input and output buffer sizes, their size has been set to 2 flits.

It should be noted that links in a NOW may be long and may have different lengths. As a consequence, flow control strategies differ from those used in multicomputers and DSMs, usually requiring deep buffers [4]. Those issues will not be considered in this paper because performance does not differ significantly from the case considered here (fixed-length short wires) if flow control is efficiently implemented [36].

Also, when physical channels are split into two virtual channels, we assumed flit-by-flit multiplexing [9]. It is important to note that we assume that virtual channel multiplexing can be efficiently implemented. In practice, implementing virtual channels is not trivial because link wires can be long, increasing signal propagation delay and making flow control more complex. This issue would require further research, analyzing techniques like channel pipelining and/or block acknowledgment, and it is out of the scope of this paper, which focuses on routing. See [36] for a description of an efficient flow-control protocol that supports long wires and different wire lengths, as well as the corresponding performance evaluation.

6.2 Network Model

The network is composed of a set of switches. Network topology is completely irregular and has been generated randomly. However, for the sake of simplicity, we imposed three restrictions to the topologies that can be generated. First, we assumed that there are exactly four workstations connected to each switch. Also, two neighboring switches are initially connected by a single link. That link is duplicated later in order to implement the A-2, MA-2, and UD-2 routing algorithms. Finally, all the switches in the network have the same size. In case links are duplicated, we assumed 12-port switches, thus leaving eight ports (four pairs) available to connect to other switches. On the other hand, when channels are split into two virtual channels, we assumed that switches have eight ports, using four of them to connect to other switches. We have evaluated networks with a size ranging from 16 switches (64 workstations) to 64 switches (256 workstations). For each network size, several distinct irregular topologies have been analyzed.

In order to apply the up*/down* routing algorithm, the spanning tree for the network must be computed. The root switch is chosen as the switch whose average distance to the rest of the switches is the smallest one. After rooting the network, links are labeled with their corresponding direction, according to the rules described in Section 4. Then, routing tables for the up*/down* scheme are computed. In order to apply the A-2 and the MA-2 schemes, all the channels in the network are duplicated, and routing tables are computed as explained in Section 5. Finally, routing tables for the UD-2 scheme are also computed. Routing tables for the UD-2vc, A-2vc, and MA-2vc routing schemes are the same as for UD-2, A-2, and MA-2, respectively, but considering virtual channels instead of physical ones.

6.3 Message Generation

Message traffic and message length depend on the applications. We have run simulations with synthetic traffic, as well as execution-driven simulations. For each simulation run with synthetic traffic, we considered that message generation rate is constant and the same for all the workstations. Once the network has reached a steady state, the flit generation rate is equal to the flit reception rate (traffic). We have evaluated the full range of traffic, from low load to saturation. With respect to message destination, we have considered that it is randomly chosen among all the workstations in the network. This pattern has been widely used in other performance evaluation studies [9], [6], [5]. We have also considered the bit reversal, perfect shuffle, and transpose distributions. For message length, 16-flit, 32-flit, 64-flit, 128-flit, and 256-flit messages were considered. Also a mixture of short and long messages, with 80 percent 16-flit messages and 20 percent 256-flit messages, was used. Simulations were run, after a transient period long enough to deliver 60,000 messages, for a number of cycles sufficient for obtaining steady values of network throughput, or, when the network is close to saturation, for a number of cycles high enough to deliver 200,000 messages.

With respect to execution-driven simulations, we have evaluated the behavior of an irregular network using the traffic generated by some SPLASH-2 applications as the input load (see [19] for a detailed description of the execution-driven simulator and application parameters).

6.4 Simulation Results

Here we present the results obtained by simulation. We divide this section into three parts. In the first one, we will compare and analyze the performance of the proposed routing algorithms. In the second part, we will study in more detail some of the characteristics of the MA-2 (MA-2vc) routing scheme. In the last part of this section, we will analyze how much variations in the network

topology influence the performance of the different routing algorithms.

6.4.1 Comparing the Routing Algorithms

Fig. 4 shows the average message latency versus traffic for each of the routing schemes when message size is 16 flits. Fig. 4a displays the results for a network composed of 16 switches. As can be seen, when doubling the available bandwidth, the A-2 routing scheme triples throughput when compared with the UD strategy, while the UD-2 routing scheme only doubles throughput. Moreover, the latency achieved by A-2 is lower than the one achieved by UD-2 for the whole range of traffic. Thus, most of the improvement achieved by A-2 is due to the additional routing flexibility provided by this scheme and the use of shorter paths. This behavior of the A-2 scheme reduces contention and balances channel utilization. On the other hand, when the MA-2 routing algorithm is used, throughput is almost increased by a factor of 4 with respect to UD, outperforming the A-2 algorithm, but latency is slightly higher than when using the A-2 routing scheme. This is due to the routing restrictions of the MA-2 scheme, which, at the source switch, provides much less adaptivity than the A-2 algorithm. This loss of adaptivity is due to the use of only minimal routes. However, in the 16-switch network we analyzed, the longest distance between switches is four hops when using the up*/down* routing algorithm, and three hops considering only the topology, and the average distances are about 2.1 and 1.8 hops, respectively. With these maximum distances, each message crosses very few channels, and therefore, the main advantage provided by the MA-2 routing algorithm is not used. Moreover, both average distances are similar, meaning that the nonminimal routes are not much longer than the minimal ones.

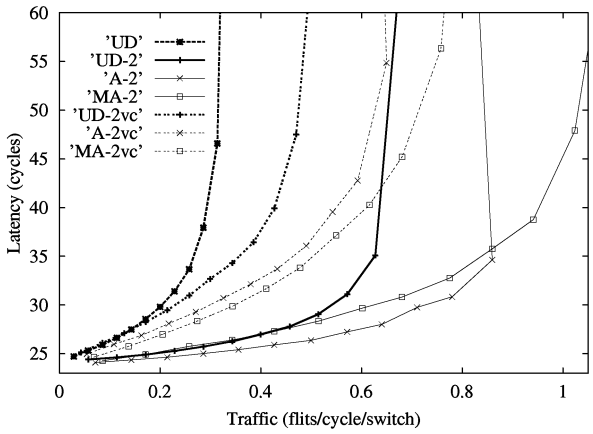
When virtual channels are used instead of duplicating all the network links, UD-2vc, A-2vc, and MA-2vc perform better than the UD strategy. A-2vc doubles the throughput achieved by the UD scheme. MA-2vc performs even better than A-2vc. However, the improvement with respect to A-2vc obtained when using the MA-2vc algorithm is smaller than when physical links were duplicated. Also, the MA-2vc routing strategy achieves the lowest latency for the whole range of traffic. Most of the improvement achieved by MA-2vc is due to the use of shorter paths, besides the increment of adaptivity with respect to the UD scheme. Note that the MA-2vc routing scheme achieves higher throughput than the UD-2 algorithm, despite the fact that the latter requires doubling network bandwidth. Obviously, the MA-2vc scheme exhibits a higher latency.

Differences in the relative behavior of both the MA-2vc routing scheme and the A-2vc strategy, with respect to the MA-2 and A-2 algorithms, respectively, arise both in latency and throughput. Differences in latency are due to the fact that when channels are duplicated, messages crossing a pair of parallel physical channels can advance toward their destination independently from each other because each of them can use the whole channel bandwidth. Therefore, wasting some resources in a few cases due to the use of nonminimal paths is profitable (especially in small networks because of the small differences in the average distances). In this case, for small networks, it is more important to provide a greater adaptivity. This is the reason why message latency is lower when using A-2 than when using MA-2 for low and medium network loads. However, for high network loads, when links are multiplexed into two virtual channels, the use of nonminimal paths degrades performance more than in the previous case because the message traveling along one of these paths hinders the progress of the other message that is waiting in the other virtual channel multiplexed onto the same physical channel. Therefore, the use of minimal routing increases the achieved throughput.

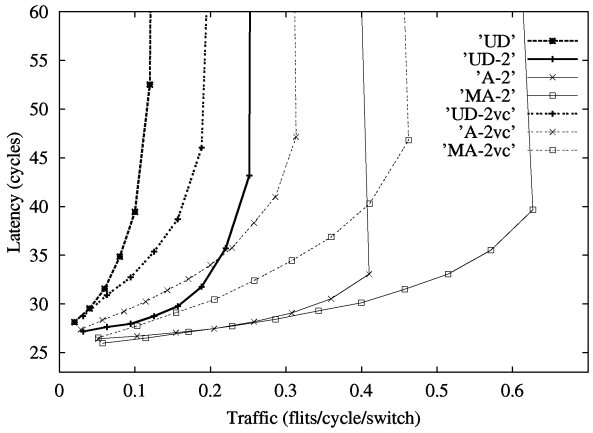
The proposed routing algorithms scale very well with network size. Figs. 4b and 4c show the results obtained for networks with 32 and 64 switches, respectively (note that the X axis in Fig. 4b is one third smaller than in Fig. 4a, and also in Fig. 4c is two thirds smaller). The message size is 16 flits. For 64 switches, when physical channels are duplicated, the A-2 routing algorithm increases throughput with respect to the UD algorithm by a factor of 4.1, while the UD-2 schemes does by a factor of 2.1. Latency is reduced for the whole range of traffic. This is due to the use of adaptivity and shorter paths. In the case of the MA-2 strategy, the improvement is much greater, increasing throughput by a factor of 6. This surprising improvement is due to a more balanced distribution of traffic across the network and the use of minimal routing in most cases. In large networks, the average routing distance imposed by the up*/down* mechanism is about 4 hops, while it is 3.1 hops considering only the topology. Therefore, it is worth using minimal routing at the cost of losing some adaptivity, mainly at the source switch. Note that unlike in the case for 16-switch networks, in this case the MA-2 algorithm achieves the lowest latency for the whole range of traffic.

Regarding the use of virtual channels in the 64-switch network, it can be seen that the UD-2vc routing scheme almost doubles throughput and reduces latency with respect to the UD scheme. The A-2vc routing scheme triples throughput and also reduces latency with respect to the UD scheme. However, the best performance is achieved by the MA-2vc algorithm, whose throughput is almost five times greater than the one achieved by the UD scheme, at the same time that latency is noticeably reduced. Note that for large networks, the MA-2vc scheme achieves a higher throughput than the A-2 scheme, despite the fact that network bandwidth is half the bandwidth available for the latter scheme. This clearly indicates the importance of using minimal routing.

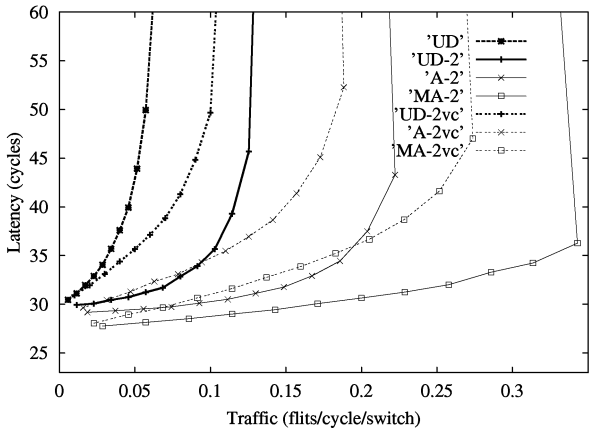
Fig. 5 shows the influence of message size on the behavior of the routing schemes. Figs. 5a, 5b, and 5c show the average message latency versus traffic for a network with 64 switches when message length is 64, 128, and 256 flits, respectively. These results show the robustness of the proposed routing algorithms against message length variation. As can be seen, the UD and UD-2 routing schemes achieve a slightly better throughput when message size is increased. In these routing schemes, link contention is significant. The A-2 and MA-2 routing schemes also achieve a slightly better throughput when message size is



(a)

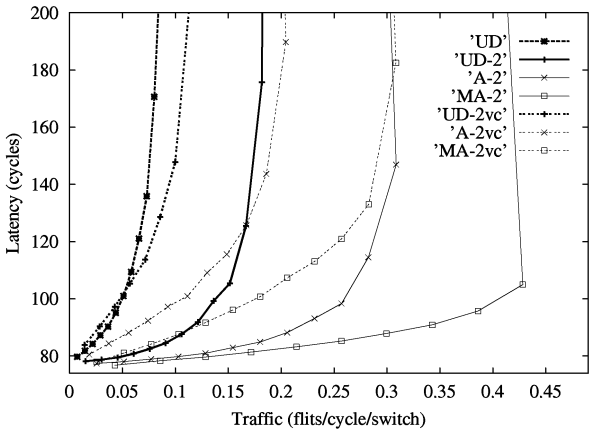


(b)

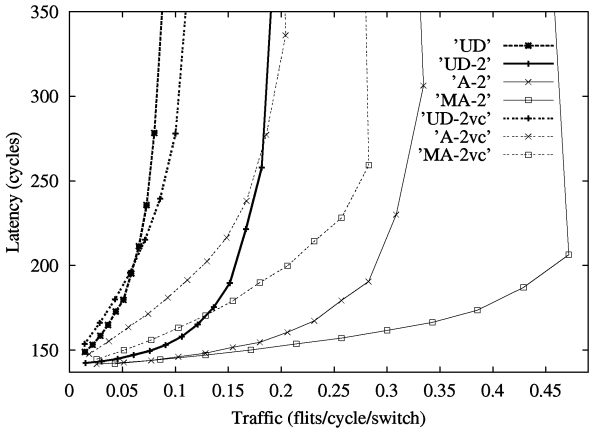


(c)

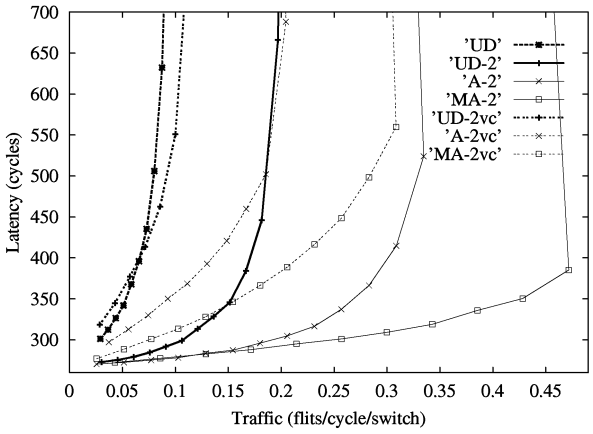
Fig. 4. Average message latency versus traffic for an irregular network with (a) 16 switches, (b) 32 switches, and (c) 64 switches. Message length is 16 flits.



(a)



(b)



(c)

Fig. 5. Average message latency versus traffic for an irregular network with 64 switches. Message length is (a) 64 flits, (b) 128 flits, and (c) 256 flits.

larger. Anyway, the MA-2 routing scheme achieves the maximum performance and lowest latency for all the message sizes.

When virtual channels are used, UD-2vc, A-2vc, and MA-2vc schemes achieve a throughput similar to that achieved with 16-flit messages (Fig. 4c). This fact shows that their behavior is almost independent of message size. However, as routing algorithms based on duplicating physical channels increase throughput when message size increases, the relative behavior of these algorithms with respect to those based on virtual channels improves with message size.

In the case of using a mixture of short and long messages (80 percent 16-flit messages and 20 percent 256-flit

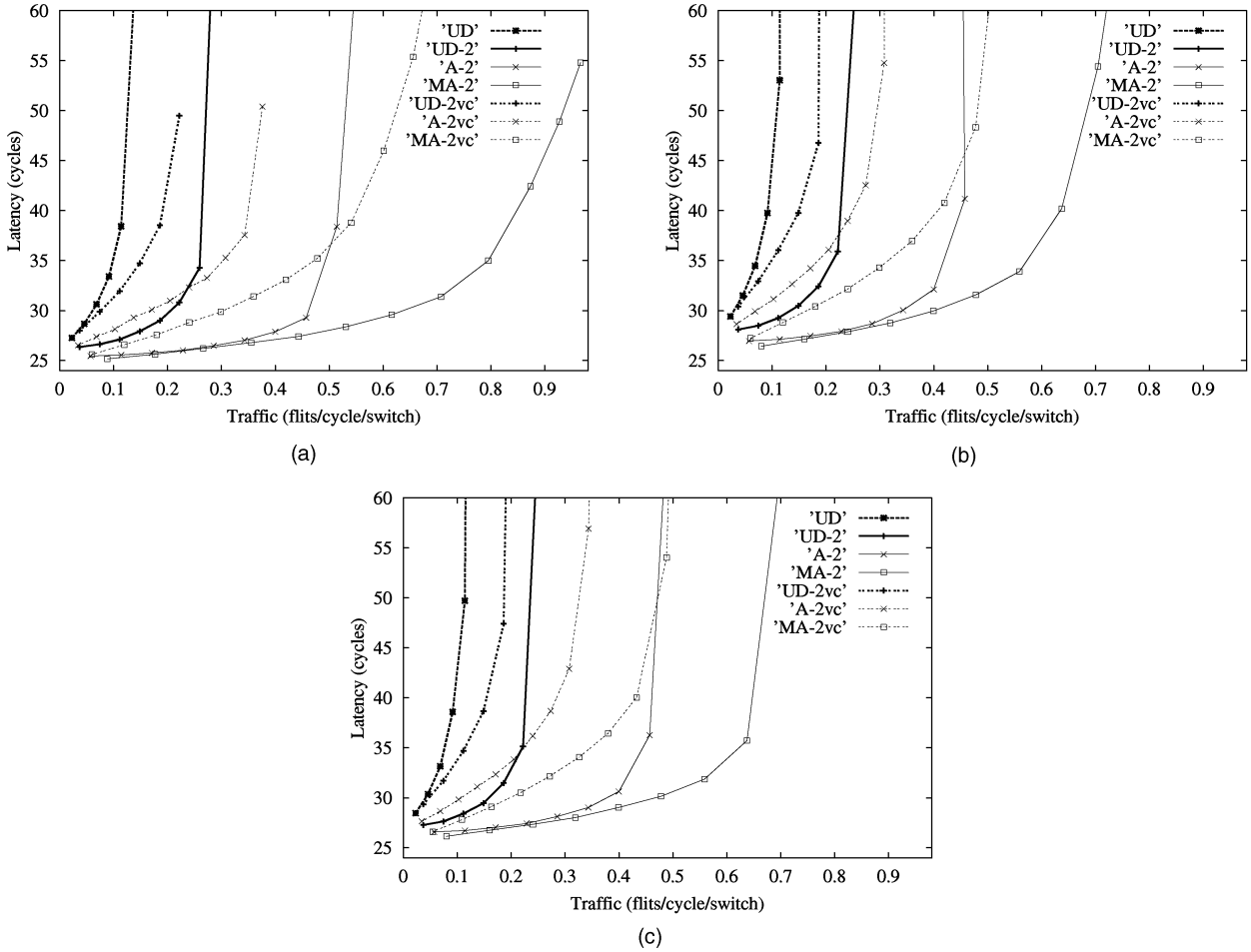


Fig. 6. Average message latency versus traffic for an irregular network with 32 switches. Message length is 16 flits. Message destination distribution is (a) bit reversal, (b) perfect shuffle, and (c) transpose.

messages), simulation results (not presented) show similar behavior as for other message sizes for all seven routing algorithms under study.

When the distribution of message destinations is not uniform, the performance achieved by the different routing algorithms increases. Fig. 6 shows the simulation results for the seven routing schemes in a network composed of 32 switches when message destination follows the (a) bit reversal, (b) perfect shuffle, and (c) transpose distributions. These plots can be compared with Fig. 4b. It can be seen that the UD, UD-2, and UD-2vc routing algorithms do not improve throughput significantly with respect to the case of using the uniform distribution. For the remaining routing algorithms (A-2, A-2vc, MA-2, and MA-2vc), the achieved throughput is noticeably increased, especially when the bit reversal distribution is used. Note that at the same time that throughput is increased, the relative improvement of the four proposed algorithms with respect to UD is also greater. This result is also valid for the rest of network sizes and message lengths, especially in the case of long messages.

Note that when using message distributions like bit reversal, perfect shuffle, and transpose, communication between workstations follows a fixed pattern. Moreover, this pattern is sometimes symmetric. This is true in the case of bit reversal and transpose distributions, where workstation A sends messages to workstation B, while workstation B sends its messages to workstation A. Thus, we could reduce network traffic by connecting workstations A and B to the same switch. In this case, no traffic would exist between network switches. However, our objective is to stress the network in order to study the behavior of routing algorithms, and therefore we need messages to be routed through the network. We have allocated workstation n at switch n DIV 4. With this allocation policy, some traffic is internal to the switches, as shown in Table 1, but the percentage of this internal traffic is quite small.

On the other hand, one is always left wondering if the proposed routing algorithms really make a difference in the real world. To answer this question, we have run several execution-driven simulations. We have analyzed whether the interconnection network is able to handle the traffic generated in a NOW with the shared memory model. In particular, we have evaluated the behavior of a 16-switch irregular network using the traffic generated by some SPLASH-2 applications as the input load (see [19] for a detailed description of the execution-driven simulator and application parameters). Some simulation results are presented in Figs. 7 and 8 for FFT and RADIX, respectively. As can be seen, the proposed routing algorithms reduce the average message latency, and as a result, significantly

TABLE 1
Percentage of Internal Traffic for Different Destination Distributions and Network Sizes

	Bit Reversal	Shuffle	Transpose
16 switches	12.5%	12.5%	12.5%
32 switches	12.5%	6.2%	6.2%
64 switches	6.2%	3.1%	6.2%

reduce the overall execution time for the applications. In the case for FFT, MA-2vc reduces execution time by 32 percent with respect to UD. In the case for the RADIX application, the reduction in execution time is even larger, reaching 43 percent when MA-2vc is used. The main reason for this significant improvement is that the network saturates for short intervals during the execution of the applications (see the sharp latency increments in Figs. 7 and 8). During these intervals, the higher throughput achieved by MA-2vc and A-2vc with respect to UD helps draining messages and reducing network congestion at a faster rate. It should be noticed that the network saturates because there are four workstations connected to each switch, thus sharing the bandwidth provided by its links. This kind of configuration is very common in current NOWs. These results agree with the ones obtained for ccNUMA multiprocessors when several processors are attached to each network router [29].

Finally, it should be noted that the improvement achieved by using the theory proposed in [14], [15] for the design of adaptive routing algorithms is much higher in irregular topologies (results presented in this paper) than in regular ones [15]. This is mainly due to the fact that in irregular networks, routing algorithms like up*/down*

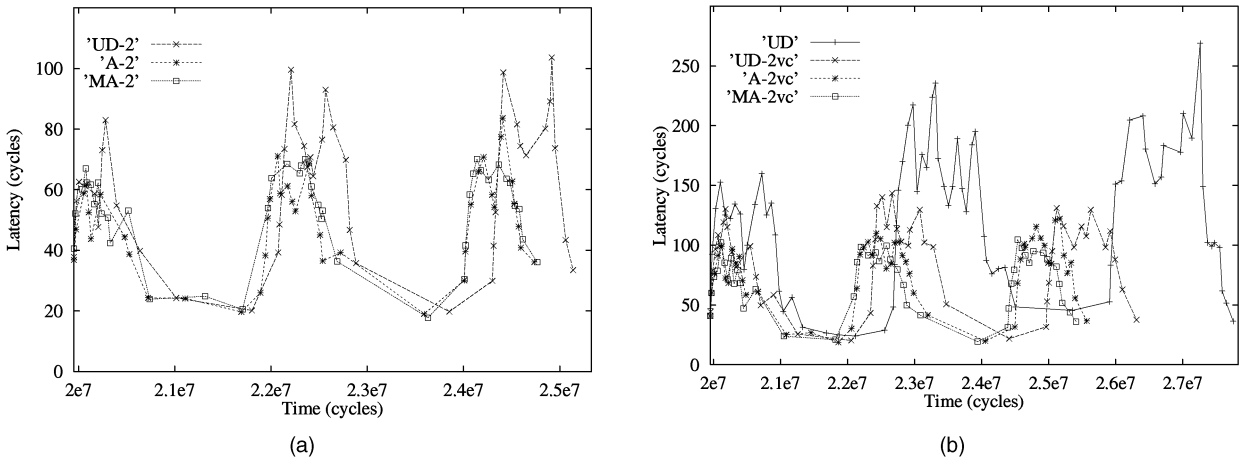


Fig. 7. Average message latency during the execution of FFT in an irregular network composed of 16 switches.

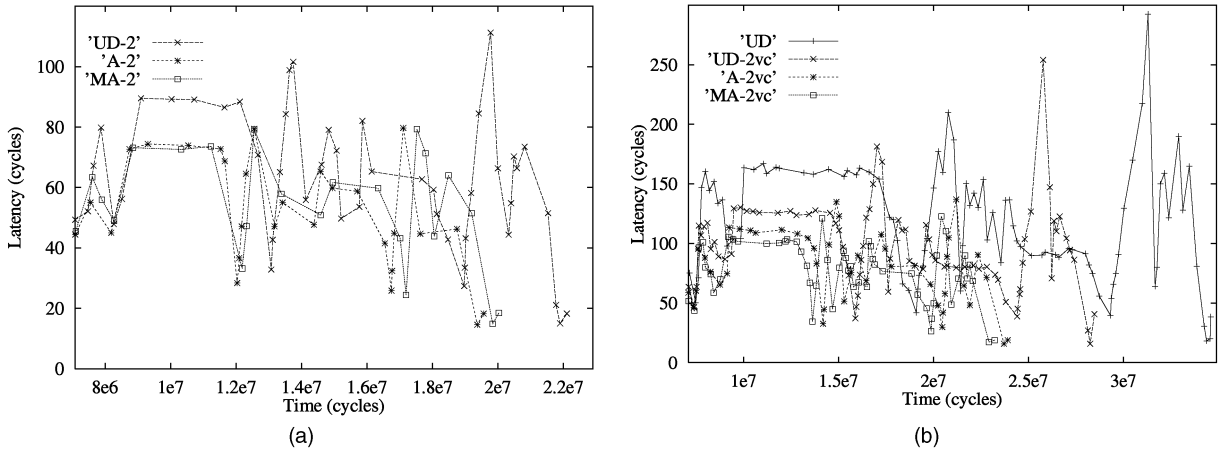


Fig. 8. Average message latency during the execution of RADIX in an irregular network composed of 16 switches.

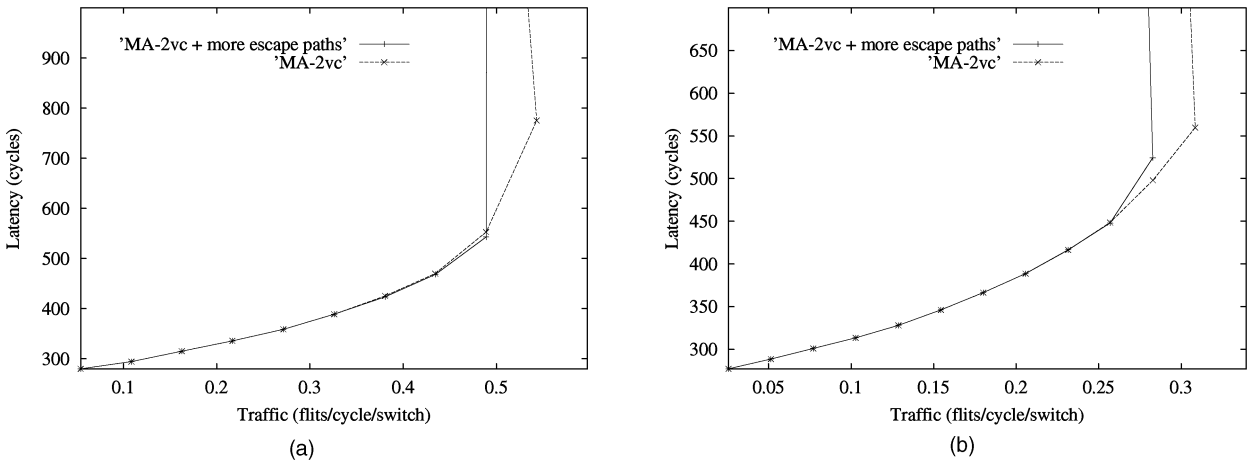


Fig. 9. Average message latency versus traffic for an irregular network with two virtual channels per physical channel. Network size is (a) 32 switches and (b) 64 switches. Message length is 256 flits.

routing do not provide minimal paths in most cases, and also concentrate traffic near the root switch of the network. However, these problems are avoided if the mentioned theory is used to design routing algorithms.

6.4.2 The MA-2 Routing Algorithm in Detail

Figs. 4, 5, and 6 clearly show that the MA-2 routing algorithm achieves the best performance when duplicating physical links. The same happens with MA-2vc in the case of using virtual channels. This happens despite the fact that these routing algorithms provide less adaptivity than the A-2 and A-2vc schemes. There are three main differences between the MA-2 (MA-2vc) and the A-2 (A-2vc) routing algorithms. These three differences are:

- With the MA-2 (MA-2vc) scheme, a new outgoing channel can be used by a message to leave the current switch only if it belongs to a minimal path, while with the A-2 (A-2vc) strategy, it can be used if the routing distance from the next switch up to the destination is less than that from the current switch to the destination.
- In the MA-2 (MA-2vc) routing algorithm, newly injected messages can only leave the source switch using new channels belonging to minimal paths, and never using original channels. With the A-2 (A-2vc) scheme, messages can leave the source switch through new and original channels.
- At intermediate switches, the MA-2 (MA-2vc) routing function gives a higher priority to the new channels belonging to minimal paths. As escape paths, the routing algorithm provides the original channels belonging to minimal paths. If none of the original channels provides minimal routing, then the one that provides the shortest path will be used. The A-2 (A-2vc) algorithm can use any of the original channels.

It would be interesting to know which of these differences has a larger effect on the performance improvement achieved by MA-2 (MA-2vc) with respect to A-2 (A-2vc). To do so, we first analyze the influence on performance produced by the third difference. We have simulated a variation of the MA-2 (MA-2vc) routing algorithm in which when all of the outgoing new channels that provide minimal routing are busy, messages are allowed to use any of the original channels, independently of whether they provide minimal routing. This variation will increase the adaptivity of the routing algorithm, because there are more choices to route messages. Therefore, the overall network performance may increase. However, messages routed through the additional choices will, in most cases, follow nonminimal paths, thus diminishing network performance.

Fig. 9 shows some of the simulation results. In Fig. 9a, one can see the comparison between the performance achieved by the variation of the algorithm and the one obtained with the MA-2vc scheme. Network size is 32 switches and message length is 256 flits. As can be seen, both strategies behave similarly, but the MA-2vc algorithm achieves a slightly higher throughput. In the case of a 64-switch network (in Fig. 9b), the worsening of the new algorithm is greater. When shorter messages are used, only with large networks and 64-flit messages (not shown), the MA-2vc scheme performs better. In the rest of the cases, there is no difference between the two algorithms. When links were duplicated instead of using virtual channels, none of the simulations revealed any difference.

We can conclude that the small worsening in performance is due to the fact that when several original channels that belong to nonminimal paths are provided by the routing function, messages have a higher probability to find a free original channel, thus following a nonminimal path and acquiring more resources than necessary. This is more noticeable in large networks, where the differences in path lengths are greater. Therefore, it is better for messages to wait for some cycles until any of the outgoing channels belonging to a minimal path becomes free (this channel will probably be a new one) than being directly routed through a probably nonminimal path that, in addition, is less adaptive.

We have seen that the third difference between the MA-2 (MA-2vc) and the A-2 (A-2vc) routing algorithms has a small influence on the improvement achieved by the former. Let us analyze the second difference. To do so, we

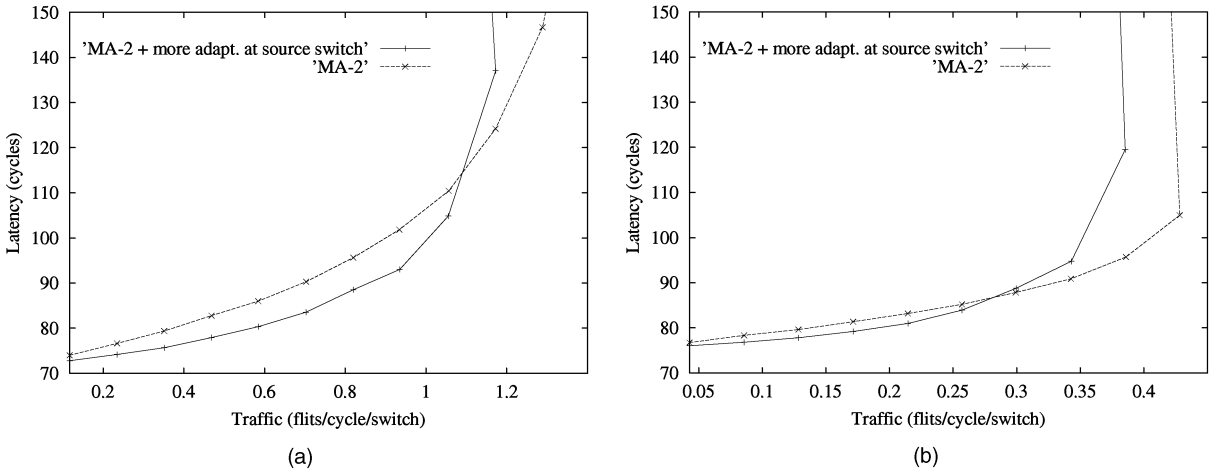


Fig. 10. Average message latency versus traffic for an irregular network. Message length is 64 flits. Physical channels are duplicated. Network size is (a) 16 switches and (b) 64 switches.

have modified the routing tables of the MA-2 (MA-2vc) scheme in order to allow messages to leave the source switch through both new and original channels. Messages can use only those new channels that provide minimal routing, while they are allowed to use any of the original channels provided by the up*/down* routing function, independently of whether they provide minimal routing or not. At intermediate switches, we keep the basic behavior of the MA-2 (MA-2vc) strategy. Note that this variation of the MA-2 (MA-2vc) routing algorithm considerably increases adaptivity at the source switch.

Fig. 10a shows the average message latency versus traffic for a network with 16 switches when message length is 64 flits. Physical links have been duplicated. As can be seen, when messages are allowed to leave the source switch using both new and original channels, message latency is slightly lower. This is due to the higher adaptivity of the new algorithm, that reduces contention and increases performance more than the use of nonminimal routing worsens it. However, the throughput achieved by the new version of the algorithm is lower than the one for the original function. The reason for this is that when the network is close to saturation, the use of nonminimal routing leads to a wasting of network resources, which does not compensate the increase in performance due to the higher adaptivity. Fig. 10b shows the case for a network composed of 64 switches. The results are similar. In this case, differences in latency are smaller, while the difference in throughput is greater due to the higher differences between minimal and nonminimal routes in this larger network. Also, the MA-2 algorithm provides more adaptivity in a large network than in a small one. Thus, the additional adaptivity provided by the modified routing algorithm does not help significantly. For the remaining network sizes and message lengths, we obtained similar results.

Fig. 11 shows the same case study for a network where physical channels have been split into two virtual channels. In this case, allowing messages to leave the source switch using both new and original channels worsens performance considerably. Restricting routing at the source switch to only new channels reduces message latency and at the same time increases performance significantly.

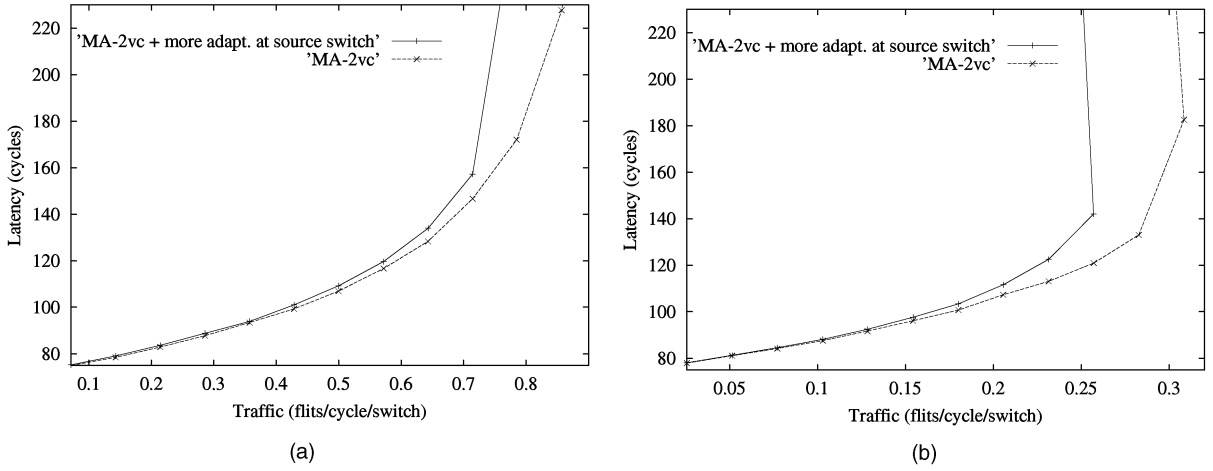


Fig. 11. Average message latency versus traffic for an irregular network with two virtual channels per physical channel. Message length is 64 flits. Network size is (a) 16 switches and (b) 64 switches.

TABLE 2
Average Adaptivity of the A-2 (A-2vc) and MA-2 (MA-2vc) Algorithms for Several Network Sizes

	MA-2 and MA-2vc		A-2 and A-2vc	
	At source switch	At intermediate switches	At source switch	At intermediate switches
16 switches	1.48	1.77	2.86	1.57
32 switches	1.49	1.59	2.86	1.48
64 switches	1.48	1.46	2.87	1.40

Analyzing the results presented in Fig. 9, we can conclude that the third difference between MA-2 (MA-2vc) and A-2 (A-2vc) routing algorithms has a small influence on the overall network performance. We have seen that restricting the use of escape paths to only those ones that belong to minimal routes makes a difference only in large networks. On the other hand, analyzing Figs. 10 and 11, we see that the second difference, that is, allowing messages to leave the source switch only through new channels, has a big impact on network throughput. Finally, after comparing the mentioned figures with Figs. 4 and 5, we conclude that the first difference (the restriction that allows using only those new channels that belong to minimal paths) is the one that has the larger effect on performance improvement. In order to analyze the reasons for that improvement in more detail, Table 2 shows the average adaptivity[4] of A-2 (A-2vc) and MA-2 (MA-2vc) routing algorithms for several network sizes. For each routing algorithm, two different adaptivities have been calculated: The first one, referred to as "At source switch," is the average adaptivity provided by the routing algorithms when routing takes place at the source switch. The second one is the average adaptivity provided when routing at intermediate switches. As expected, the A-2 (A-2vc) routing algorithm provides much more adaptivity at the source switch, because messages can leave the switch through new and original channels and also messages are allowed to leave the switch through new channels that do not provide minimal routing. However, the smaller adaptivity of the MA-2 (MA-2vc) routing algorithm produces a much more efficient use of resources, since all the provided routes are minimal. On the other hand, the MA-2 (MA-2vc) algorithm provides an adaptivity greater than the one provided by the A-2 (A-2vc) routing scheme at intermediate switches. The reason is the following: The A-2 (A-2vc) scheme allows messages to leave intermediate switches through new channels belonging to nonminimal paths and also through all the original channels supplied by the up*/down* routing function. Therefore, it should provide more adaptivity than the MA-2 (MA-2vc) routing algorithm. However, once a message is routed through an original channel at a switch, in the following switches it will be routed with a very small adaptivity, if any. Thus, the average adaptivity provided by the A-2 (A-2vc) scheme decreases considerably, being even lower than the one supplied by MA-2 (MA-2vc). On the other hand, the higher adaptivity provided by the MA-2 (MA-2vc) algorithm also provides, almost always, minimal routing, while the A-2 (A-2vc) scheme is not able to provide minimal routing in many cases. In practice, the lower adaptivity of A-2 (A-2vc) is even smaller because messages consume more resources than strictly necessary because they travel along nonminimal routes. As a consequence, output channels remain busy for longer periods of time, avoiding incoming messages to be successfully routed. In summary, the MA-2 (MA-2vc) routing algorithm achieves a much better use of resources than the A-2 (A-2vc) scheme does.

Finally, we proposed at the end of Section 5.2 a variation of the MA-2 (MA-2vc) routing algorithm that, at intermediate switches, instead of routing through an original channel once all of the new channels belonging to minimal paths are busy, would route the message through a new channel belonging to a nonminimal path that conforms to the up*/down* rule. This variation was intended to reduce the use of original channels even more. We have also analyzed this variation.

Fig. 12a shows the average message latency versus traffic for a network with 32 switches when message length is 64 flits. Physical links have been duplicated. It can be seen that using nonminimal routing on new channels has a small effect on network performance. Latency is only slightly reduced at high network loads, while no increment in throughput is achieved. In the case for a 64-switch network, there is no improvement in latency, while throughput is increased by less than 5 percent. When splitting physical channels into two virtual channels, the plot for a 32-switch network shown in Fig. 13a reveals a slight improvement in

4. The average adaptivity has been computed as the average number of routing options provided by each of the routing algorithms at every switch in the network. In order to make this computation representative, we have averaged the results for 10 different topologies for each of the network sizes analyzed.

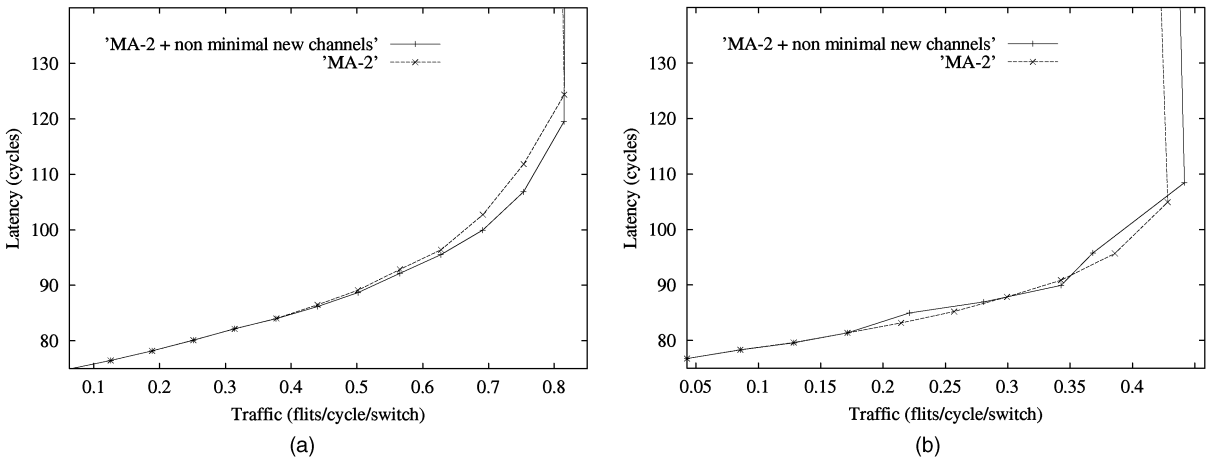


Fig. 12.Average message latency versus traffic for an irregular network. Message length is 64 flits. Physical channels are duplicated. Network size is (a) 32 switches and (b) 64 switches.

both latency and throughput. However, in the case for a larger network displayed in Fig. 13b, the new version of the MA-2vc algorithm exhibits almost the same latency, while throughput is slightly lower. In summary, as simulation results show, the increment in performance obtained by this new routing function is negligible. This is due to the use of nonminimal paths. Moreover, we have not taken into account in the simulations carried out that this new routing strategy needs a more complex selection function, which may increase routing time. Also, note that routing tables need to be larger, because they must store, besides the new and original channels as in MA-2vc, the nonminimal new channels that can be selected when all the minimal new channels are busy.

6.4.3 The Randomness of the Network

When someone builds a NOW with an irregular topology, she or he is supposed to arrange the workstations and switches in such a configuration that the network has good performance. If this performance is not as good as expected, she or he will probably tune the network topology, changing its layout if possible. Therefore, the topology of a NOW is rarely defined in a completely random way. The location of the workstations greatly influences the topology, but the customer can also play an important role in defining the topology. Thus, this process is not completely random. However, not all the existing networks will have the same topology (or the best topology), mainly due to space constraints in the arrangement of workstations and switches.

The results presented above have been obtained from the simulation of the routing algorithms in networks whose topologies have been generated in a completely random way. As discussed above, this is not the best way to model the topology of a NOW, but generating it randomly prevents us from performing a tendentious study.

Because not all the networks will have the same topology, we can wonder whether varying the irregular

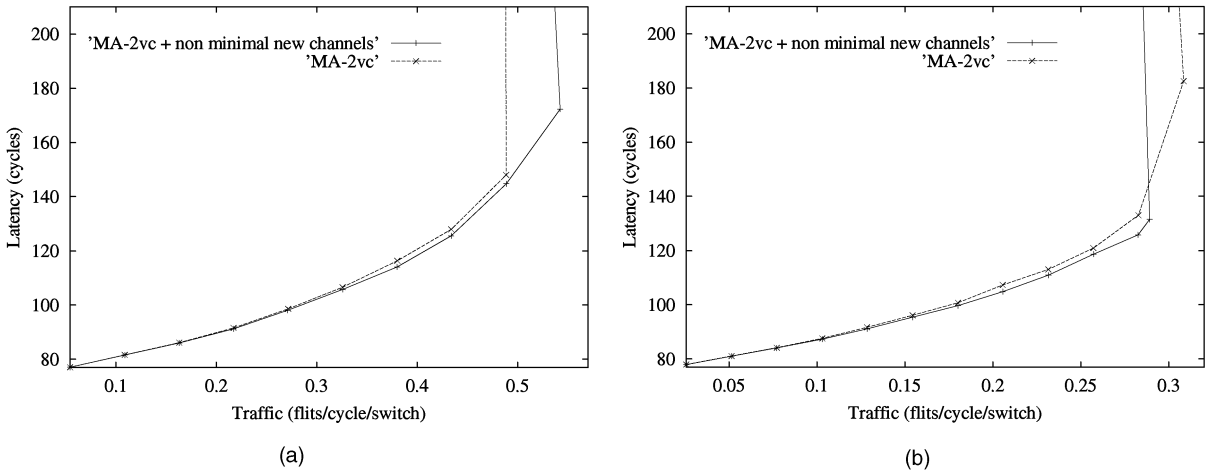


Fig. 13. Average message latency versus traffic for an irregular network with two virtual channels per physical channel. Message length is 64 flits. Network size is (a) 32 switches and (b) 64 switches.

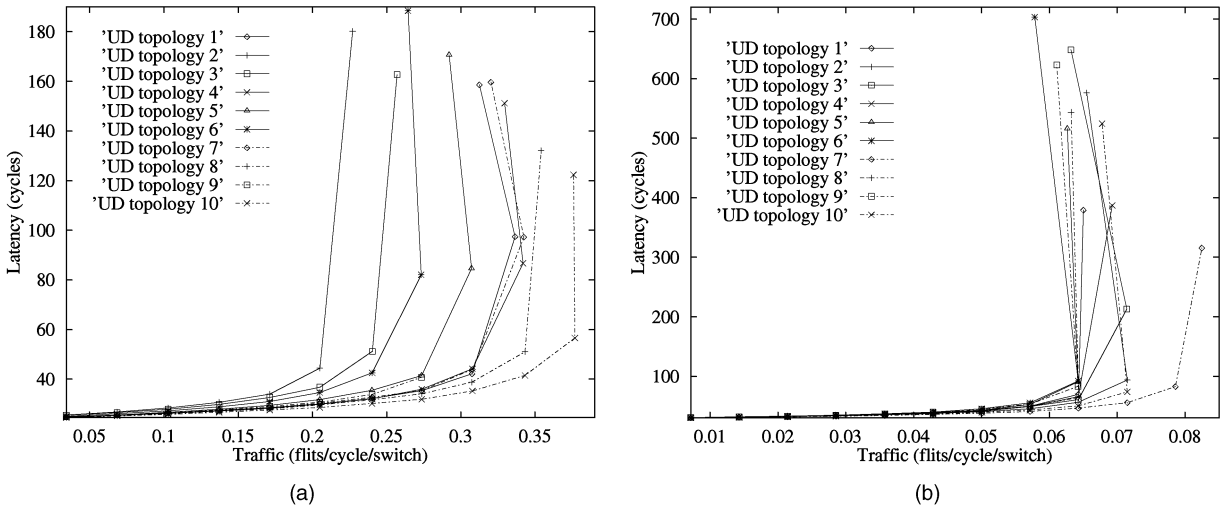


Fig. 14. Average message latency versus traffic for 10 different irregular networks when using the UD routing algorithm. Message length is 16 flits. Network size is (a) 16 switches and (b) 64 switches.

topology influences the network performance. In other words, will all the networks generated in a completely random way have the same behavior? Obviously, we cannot expect all of them to present the same performance. Therefore, how much do variations in the topology affect the performance achieved by the studied routing algorithms? Will the best routing algorithms presented above continue being the best ones in other irregular topologies?

In order to study how the randomness of the network affects the performance of the routing algorithms, we generated, in a completely random way, 10 different irregular topologies for each of the network sizes analyzed. To build these topologies, we imposed the same three restrictions mentioned in Section 6.2. We carried out this study for networks with 16 switches (64 workstations) and 64 switches (256 workstations). For each of the topologies generated, we ran simulations for all of the seven routing algorithms studied in the previous sections, evaluating the full range of traffic. Message destination was randomly chosen among all the workstations in the network.

Fig. 14 presents the average message latency versus traffic for the 10 different generated topologies when using the UD routing algorithm. In Fig. 14a, one can see the results for a 16-switch network. Message length is 16 flits. As can be seen, network performance varies with network topology. For low loads, all the topologies present the same message latency. Differences arise when the networks are heavily loaded, close to saturation. Depending on the particular topology used, throughput can vary by as much as 44 percent. The topology that achieves the lowest performance is the second one. In the case for 64-switch networks (Fig. 14b), performance is also influenced by the topology of the network. In this case, the variation in throughput is not larger than 21 percent. Again, differences in performance are noticeable only when networks are close to saturation.

For the rest of routing algorithms, results also confirm that performance depends, in part, on the topology of the network. In the case for the MA-2vc scheme (shown in Fig. 15), the variation is much smaller than for the UD strategy. In fact, the MA-2vc routing algorithm presents, among the seven routing algorithms we evaluated, the smallest variation in performance when changing the topology. Fig. 15a shows the results for a 16-switch network. All the topologies, except the second one, present almost the same performance, varying by no more than 15 percent. The second topology exhibits the worst behavior for all the routing algorithms. Fig. 15b shows the comparison for 64-switch networks. In this case, variation in throughput is not larger than 16 percent. In general, the average variation in throughput due to the topology for all the routing schemes is not larger than 25 percent.

As seen, the performance of the different routing algorithms depends, in part, on the underlying network topology. The question now is whether the relative benefits of the best routing algorithms presented above are kept in other irregular topologies. In order to answer this question, we compared the relative performance of the distinct routing schemes in several topologies. We observed that the relative performance of the several routing algorithms is not significantly affected by the particular topology used. Using throughput as the performance metric, when we duplicate the links in the network, the MA-2 routing algorithm always performs better than the A-2 algorithm, and the latter performs better than UD-2. In the case of splitting physical channels into two virtual channels, results are identical: MA-2vc performs better than A-2vc, which performs better than UD-2vc. The worst performance is always achieved by the UD scheme. With respect to the

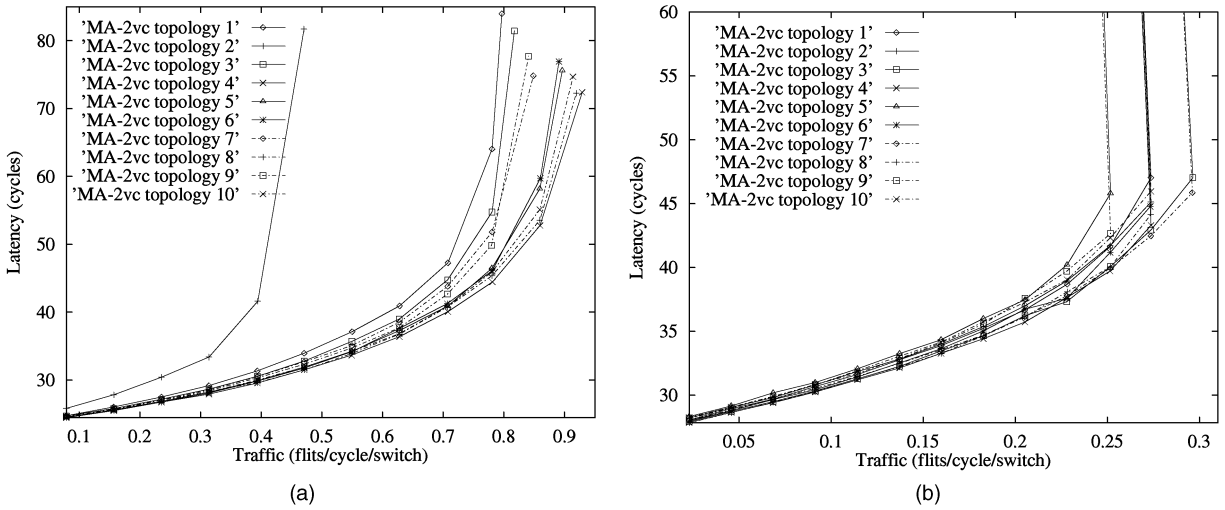


Fig. 15. Average message latency versus traffic for 10 different irregular networks when using the MA-2vc routing algorithm. Message length is 16 flits. Network size is (a) 16 switches and (b) 64 switches.

factor of improvement for the different algorithms, it varies on average less than 18 percent when changing from one topology to another.

Also, note that the average factor of improvement for the MA-2 routing algorithm with respect to the UD scheme is about 6 with uniform traffic, while in the case for the MA-2vc strategy is about 4. Thus, it is worth splitting physical channels into virtual ones when designing new switches because this alternative presents a better cost/performance ratio.

Finally, some topologies have been especially designed for up*/down* routing to achieve good performance [31]. We have evaluated the performance of the different routing algorithms in these topologies. Some of them have the property that all of the paths provided by up*/down* routing are minimal. For the sake of brevity, we will not describe these topologies in this paper. The reader can refer to [31] for a detailed description. Simulation results are presented in Fig. 16. For all the plots, traffic distribution follows a client/server model (also proposed in [31]). In this model, a few servers are uniformly distributed all round the network. As can be seen, in the torus and hypercube, using the routing algorithms proposed in this paper improves performance noticeably. However, for the rest of topologies, we obtain no benefit when using virtual channels or duplicating physical links (except for the mstage topology). The reason for this is the traffic pattern used. All traffic is between clients and servers, and therefore, the network is congested due to the fact that servers cannot absorb messages fast enough. For the same topologies, we have also run simulations where message destinations are randomly chosen among all the workstations in the network. The obtained results are similar to those presented in Figs. 4 and 5.

7 Conclusions

Networks of workstations (NOWs) are becoming increasingly popular. In particular, they are a cost-effective alternative to parallel computers. Typically, workstations in a NOW are connected using irregular topologies. Irregularity provides the wiring flexibility, scalability, and incremental expansion capability required in this environment. However, the irregular connections between switches also make routing and deadlock avoidance on these networks quite complicated. Current proposals avoid deadlock by removing cyclic dependencies between channels. As a consequence, routing is considerably restricted and many messages must follow nonminimal paths, therefore increasing latency and wasting resources.

In this paper, two simple and general methodologies for the design of adaptive routing algorithms for switch-based interconnects with irregular topology have been proposed. Routing algorithms designed according to these methodologies are intended to increase the adaptivity of the original routing function while allowing more messages to follow minimal paths. The first design methodology focuses on maximizing adaptivity. Routing algorithms designed according to the second methodology route most of the messages through minimal paths, even at the cost of losing some adaptivity, especially at the source switch. Additionally, it has been shown that the resulting routing algorithms are deadlock and livelock-free. As an example of the application of these methodologies, they have been used to improve the performance of Autonet networks. To apply each design methodology, two alternative implementations have been analyzed. The first one does not require changing the switch design. It takes advantage of spare switch ports, duplicating the links in the network. Also, routing tables should be updated according to the new algorithms. The second implementation approach consists of splitting

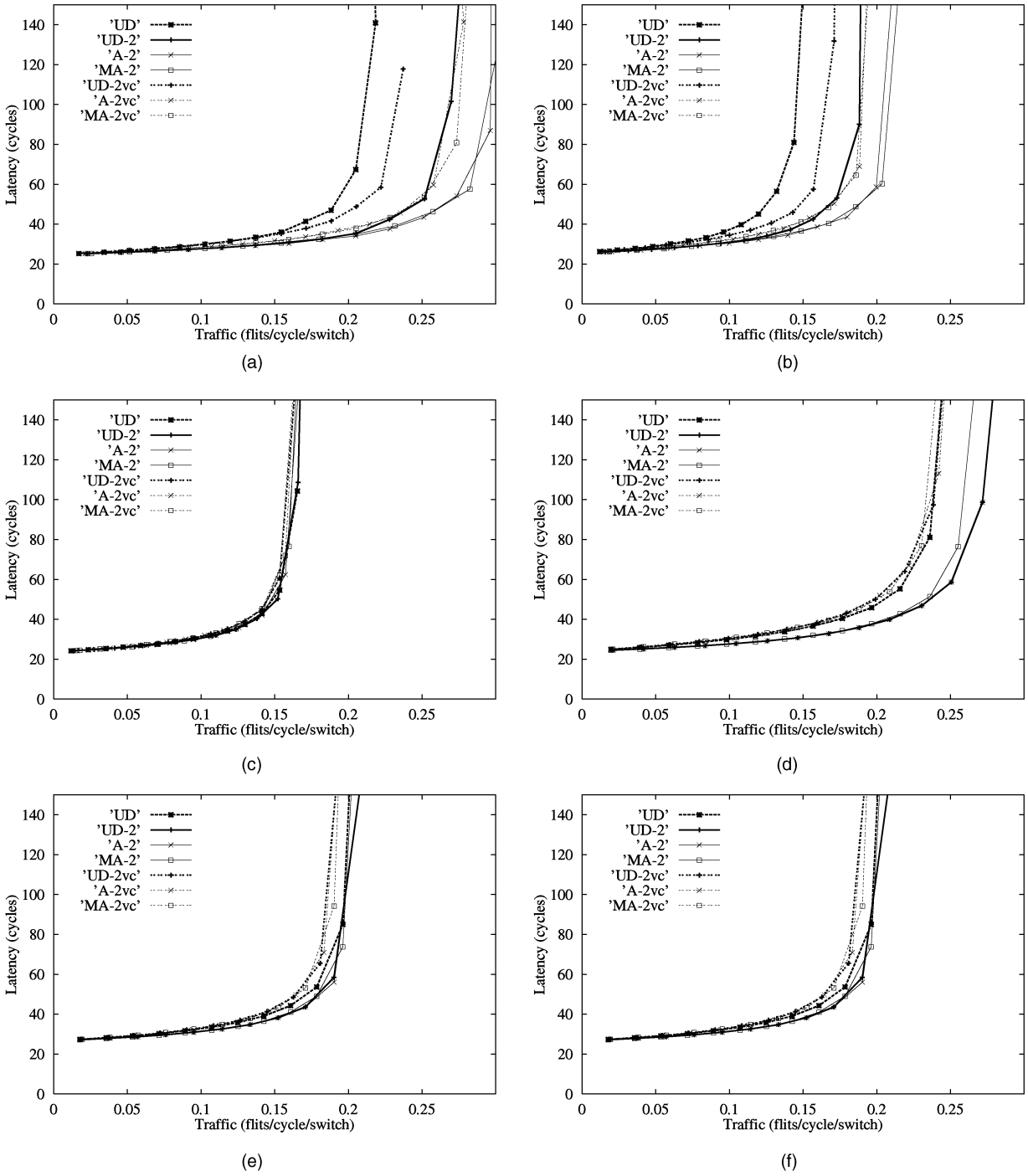


Fig. 16. Simulation results for some nonrandom topologies: (a) Torus, (b) Hypercube, (c) Corefringe, (d) Mstage, (e) Floors, and (f) Floors+ (refer to [31] for a description of these topologies). Message length is 16 flits. All traffic is between clients and servers.

physical channels into two virtual channels. In this case, the topology remains unchanged, but a new switch design is required.

The performance of the new routing algorithms has been evaluated, comparing it with the original up*/down* algorithm as well as with an extension of up*/down* that uses the new channels according to the same basic routing rules. The evaluation study was performed on networks with 16, 32, and 64 switches, using several distributions of traffic and different message lengths. Despite the simplicity of the proposed approaches, the new routing algorithms increase throughput by a factor between 2 and 4.5 with respect to the original up*/down* routing algorithm when physical channels are split into two virtual channels. When physical channels are doubled, the new routing algorithms improve throughput up to seven times the original one. Latency is also considerably reduced in all the cases for the whole range of network load. Different message sizes have

been evaluated, showing that the throughput achieved by the new algorithms is almost independent of message length, especially when physical channels are split into virtual ones. For large networks, the improvement achieved by the new routing schemes is much larger than for small networks, especially in the case for the MA-2 (MA-2vc) routing algorithm. This is due to the higher differences in length between minimal and nonminimal routes.

We have analyzed in detail the use of minimal versus nonminimal routing, concluding that the use of minimal routing is considerably advantageous. Nonminimal routing leads to unnecessary wasting of network resources, increasing latency and reducing the maximum traffic delivered.

Finally, we have analyzed how the variations in the network topology influence the performance of the different routing algorithms. Results show that network throughput varies less than 25 percent when changing from one topology to another. They also reveal that the relative performance of the different routing schemes is not changed significantly.

ACKNOWLEDGMENTS

This work was supported by the Spanish CICYT under Grant TIC97-0897-C04-01.

REFERENCES

[1] A.C. Arpaci-Dusseau, R.H. Arpaci-Dusseau, D.E. Culler, J.M. Hellerstein, and D.A. Patterson, "High–Performance Sorting on Networks of Workstations," *Proc. ACM SIGMOD '97* May 1997.

[2] P.E. Berman, L. Gravano, G.D. Pifarré, and J.L.C. Sanz, "Adaptive Deadlock- and Livelock-Free Routing with All Minimal Paths in Torus Networks," *Proc. Fourth ACM Symp. Parallel Algorithms and Architectures,* June 1992.

[3] M.A. Blumrich, K. Li, R. Alpert, C. Dubnicki, E.W. Felten, and J. Sandberg, "Virtual Memory Mapped Network Interface for the SHRIMP Multicomputer," *Proc. 21st Int'l Symp. Computer Architecture,* pp. 142–153, Apr. 1994.

[4] N.J. Boden, D. Cohen, 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, Feb. 1995.

[5] R.V. Boppana and S. Chalasani, "A Comparison of Adaptive Wormhole Routing Algorithms," *Proc. 20th Int'l Symp. Computer Architecture,* May 1993.

[6] A.A. Chien and J.H. Kim, "Planar-Adaptive Routing: Low-Cost Adaptive Networks for Multiprocessors," *Proc. 19th Int'l Symp. Computer Architecture,* May 1992.

[7] A.A. Chien, "A Cost and Speed Model for k-Ary n-Cube Wormhole Routers," *Proc. Hot Interconnects '93,* Aug. 1993.

[8] W.J. Dally and C.L. Seitz, "Deadlock-Free Message Routing in Multiprocessor Interconnection Networks," *IEEE Trans. Computers,* vol. 36, no. 5, pp. 547–553, May 1987.

[9] W.J. Dally, "Virtual-Channel Flow Control," *IEEE Trans. Parallel and Distributed Systems,* vol. 3, no. 2, pp. 194–205, Mar. 1992.

[10] W.J. Dally and H. Aoki, "Deadlock-Free Adaptive Routing in Multicomputer Networks Using Virtual Channels," *IEEE Trans. Parallel and Distributed Systems,* vol. 4, no. 4, pp. 466–475, Apr. 1993.

[11] W.J. Dally, L.R. Dennison, D. Harris, K. Kan, and T. Xanthopoulus, "The Reliable Router: A Reliable and High-Performance Communication Substrate for Parallel Computers," *Proc. Workshop Parallel Computer Routing and Comm.,* pp. 241–255, May 1994.

[12] W.J. Dally, L.R. Dennison, D. Harris, K. Kan, and T. Xanthopoulos, "Architecture and implementation of the Reliable Router," *Proc. Hot Interconnects II,* Aug. 1994.

[13] B.V. Dao, S. Yalamanchili, and J. Duato, "Architectural Support for Reducing Communication Overhead in Pipelined Networks," *Proc. Third Int'l Symp. High Performance Computer Architecture,* Feb. 1997.

[14] J. Duato, "On the Design of Deadlock-Free Adaptive Routing Algorithms for Multicomputers: Design Methodologies," *Proc. Parallel Architectures and Languages Europe 91,* June 1991.

[15] J. Duato, "A New Theory of Deadlock-Free Adaptive Routing in Wormhole Networks," *IEEE Trans. Parallel and Distributed Systems,* vol. 4, no. 12, pp. 1,320–1,331, Dec. 1993.

[16] J. Duato, "A Necessary and Sufficient Condition for Deadlock-Free Adaptive Routing in Wormhole Networks," *Proc. 1994 Int'l Conf. Parallel Processing,* Aug. 1994.

[17] J. Duato, "A Necessary and Sufficient Condition for Deadlock-Free Adaptive Routing in Wormhole Networks," *IEEE Trans. Parallel and Distributed Systems,* vol. 6, no. 10, pp. 1,055–1,067, Oct. 1995.

[18] T. von Eicken,, D.E. Culler, S.C. Goldstein, and K.E. Schauser, "Active Messages: A Mechanism for Integrated Communication and Computation," *Proc. 19th Int'l Symp. Computer Architecture,* June 1992.

[19] J. Flich, M.P. Malumbres, P. López, and J. Duato, "Performance Evaluation of Networks of Workstations with Hardware Shared Memory Model Using Execution-Driven Simulation," *Proc. 1999 Int'l Conf. Parallel Processing,* Sept. 1999.

[20] P.T. Gaughan and S. Yalamanchili, "Adaptive Routing Protocols for Hypercube Interconnection Networks," *Computer,* vol. 26, no. 5, pp. 12–23, May 1993.

[21] D. Garcia, "ServerNet II," *Proc. 1997 Parallel Computing, Routing, and Comm. Workshop,* June 1997.

[22] L. Gravano, G.D. Pifarré, P.E. Berman, and J.L.C. Sanz, "Adaptive Deadlock- and Livelock-Free Routing with All Minimal Paths in Torus Networks," *IEEE Trans. Parallel and Distributed Systems,* vol. 5, no. 12, pp. 1,233–1,251, Dec. 1994.

[23] R. Horst, "ServerNet Deadlock Avoidance and Fractahedral Topologies," *Proc. Int'l Parallel Processing Symp.,* pp. 274–280, Apr. 1996.

[24] J.-M. Hsu and P. Banerjee, "Performance Measurement and Trace Driven Simulation of Parallel CAD and Numeric Applications on a Hypercube Multicomputer," *IEEE Trans. Parallel and Distributed Systems,* vol. 3, no. 4, pp. 451–464, July 1992.

[25] V. Karamcheti and A.A. Chien, "Do Faster Routers Imply Faster Communication?" *Proc. Workshop Parallel Computer Routing and Comm.,* May 1994.

[26] X. Lin, P.K. McKinley, and L.M. Ni, "The Message Flow Model for Routing in Wormhole-Routed Networks," *Proc. 1993 Int'l Conf. Parallel Processing,* Aug. 1993.

[27] X. Lin, P.K. McKinley, L.M. Ni, "The Message Flow Model for Routing in Wormhole-Routed Networks," *IEEE Trans. Parallel and Distributed Systems,* vol. 6, no. 7, pp. 755–760, July 1995.

[28] R.J. Littlefield, "Characterizing and Tuning Communications Performance for Real Applications," *Proc. First Intel DELTA Applications Workshop,* Feb. 1992.

[29] J.F. Martinez, J. Torrellas, and J. Duato, "Improving the Performance of Bristled CC-NUMA Systems Using Virtual Channels and Adaptivity," *Proc. 1999 ACM Int'l Conf. Supercomputing,* 1999.

[30] L.M. Ni and P.K. McKinley, "A Survey of Wormhole Routing Techniques in Direct Networks," *Computer,* vol. 26, no. 2, pp. 62–76, Feb. 1993.

[31] S.S. Owicki and A.R. Karlin, "Factors in the Performance of the AN1 Computer Network," SRC Research Report 88, June 1992.

[32] W. Qiao and L.M. Ni, "Adaptive Routing in Irregular Networks Using Cut-Through Switches," *Proc. 1996 Int'l Conf. Parallel Processing,* Aug. 1996.

[33] M. D. Schroeder, A.D. Birrell, M. Burrows, H. Murray, R.M. Needham, T. Rodeheffer, E. Satterthwaite, and C. Thacker, "Autonet: A High-Speed, Self-Configuring Local Area Network Using Point-to-Point Links," Technical Report, SRC Research Report 59, DEC, Apr. 1990.

[34] L. Schwiebert and D.N. Jayasimha, "A Universal Proof Technique for Deadlock-Free Routing in Interconnection Networks," *Proc. Symp. Parallel Algorithms and Architectures,* pp. 175–184, July 1995.

[35] S.L. Scott and G. Thorson, "The Cray T3E Networks: Adaptive Routing in a High Performance 3D Torus," *Proc. Hot Interconnects IV,* Aug. 1996.

[36] F. Silla and J. Duato, "On the Use of Virtual Channels in Networks of Workstations with Irregular Topology," *Proc. 1997 Parallel Computing, Routing, and Comm. Workshop,* June 1997.
[37] C. Su and K.G. Shin, "Adaptive Deadlock-Free Routing in Multicomputers Using Only One Extra Channel," *Proc. 1993 Int'l Conf. Parallel Processing,* Aug. 1993.

Federico Silla received the MS and PhD degrees in computer engineering from the Technical University of Valencia, Spain, in 1995 and 1999, respectively. He is currently a lecturer in the Department of Information Systems and Computer Architecture at the Technical University of Valencia. His research addresses high performance interconnects for networks of workstations and image compression.

José Duato received the MS and PhD degrees in electrical engineering from the Technical University of Valencia, Spain, in 1981 and 1985, respectively. He is currently a professor in the Department of Information Systems and Computer Architecture, Technical University of Valencia, and an adjunct professor in the Department of Computer and Information Science, The Ohio State University. He is currently researching multiprocessor systems, networks of workstations, interconnection networks, and multimedia systems. His theory on deadlock-free adaptive routing for wormhole networks has been used in the design of the routing algorithms for the MIT Reliable Router and the Cray T3E. He coauthored the text *Interconnection Networks: An Engineering Approach* with Sridhar Yalamanchili and Lionel M. Ni (IEEE CS Press). Dr. Duato served as a member of the editorial board of *IEEE Transactions on Parallel and Distributed Systems* from 1995 to 1997. Also, he has been or is a member of the program committee for several major conferences (ICPADS, ICDCS, Europar, HPCA, ICPP, MPPOI, HiPC, PDCS, ISCA, IPPS/SPDP). Dr. Duato is a member of the IEEE.