ELSEVIER

Computer Communications 25 (2002) 997–1008

computer communications

www.elsevier.com/locate/comcom

A new injection limitation mechanism for wormhole networks

M.S. Obaidat[a,*], Z.H. Al-Awwami[b], M. Al-Mulhem[b]

[a]*Department of Computer science, Monmouth University, West Long Branch, NJ 07764, USA*

[b]*College of Computer Science and Engineering, King Fahd University of Petroleum and Minerals (KFUPM), Dhahran 31261, Saudi Arabia*

Received 1 August 2001; revised 30 October 2001; accepted 28 November 2001

Abstract

Wormhole switching has been widely applied to the interconnection networks of parallel systems as well as System Area Networks, and Local Area Networks, largely because of its efficiency and performance merits. Examples include the Myrinet of Myricom Inc as well as most of the newly developed parallel systems. True Fully Adaptive Routing (TFAR) Algorithms have demonstrated their suitability for wormhole switched networks due to their unrestricted Adaptivity and moderate resource requirements. Wormhole switching has proven to be the most popular switching technique targeted for interconnection networks of message-passing multicomputers as well as SANs, and LANs. TFAR Algorithms have also been gaining favor for application in wormhole switched networks due to their highly adaptive and moderate hardware requirements. Wormhole switched networks have associated drawbacks however, as they generally suffer from performance degradation beyond the saturation point due to channel congestion. Fully adaptive algorithms are vulnerable to cyclic dependencies, which are precursors to deadlock formations. Consequently the frequent occurrence of deadlocks can further degrade the performance and stability characteristics of these networks. Injection limitation techniques were recently introduced in an attempt to countermeasure these drawbacks and effectively contain their impact on the performance of the network. This paper proposes a new injection limitation mechanism and its performance evaluation. The new mechanism is named Congestion Level Injection Control (CLIC). This mechanism attempts to provide a solution for these problems and improve the overall performance of the network. The new mechanism is centered on congestion level estimation in the network using only local information at each node. The mechanism subsequently prevents the injection of new packets if the network is deemed to be highly congested or possibly close to its saturation point. The performance of the CLIC mechanism has been compared with other competing schemes. Our results have shown that CLIC has superior performance when compared to other competing schemes.

Keywords: Wormhole networks; Injection limitation; Wormhole switching; Performance evaluation; True fully adaptive routing

1. Introduction

Massively Parallel Processors (MPP) are composed of many nodes that cooperate amongst themselves in order to subdue large problems. The nodes communicate with each other over a set of communication links and switching fabrics that are collectively known as the interconnection network. These parallel processing units need to achieve extremely fast communication amongst them. The performance of the interconnection network is therefore of significant importance to the performance of the entire message-passing multicomputer. The most common direct interconnection network topologies are the class of k-ary n-cube networks, which encompasses rings, meshes, and tori.

Wormhole switching being the most popular switching technique applied to these interconnection networks is fundamental to the efficiency achieved by these multicomputers. Wormhole switching is widely adopted in multiprocessors due to its high performance and low-cost properties. More specifically, it is well suited for applications in multicomputer interconnection networks as it allows for the design of simple, fast, and low-cost hardware router nodes while providing low latency, high-bandwidth communication [1–6]. Wormhole switched interconnection networks were originally developed and used in message-passing multicomputers. The success of these interconnects, mainly due to their scalability and performance characteristics caused the technology to be transferred to many other fields such as distributed shared-memory multiprocessors. Today the technology is applied to a plethora of applications such as internal networks for Asynchronous Transfer Mode (ATM) switches, SANs, Network of Workstations (NOWs), LANs, Metropolitan Area Networks (MANs), Wide Area Networks (WANs), telephone switches, backplane buses, processor-to-memory interconnects in supercomputer class machines,

[*] Corresponding author. Tel.: +1-732-571-4482; fax: +1-732-263-5202.
E-mail address: obaidat@monmouth.edu (M.S. Obaidat).

0140-3664/02/$ - see front matter © 2002 Elsevier Science B.V. All rights reserved.
PII: S0140-3664(02)00014-2

and very recently to Network Attached Storage (NAS) specialized servers [1].

Routing is a major characteristic of the underlying interconnection network. In the absence of a complete topology in which every node has a direct connection to every other node in the network, routing determines the path that a packet eventually follows in order to reach its destination. The routing algorithm utilized along with the switching mechanism determines how packets are routed over the interconnection network and strongly influences the performance of that network. Routing algorithms are classified according to the amount of adaptivity they provide and the approach they adopt in order to overcome the problematic deadlock issue.

Adaptive routing algorithms are favored for implementation in these networks and are designed according to varying degrees of adaptivity. *Deterministic* routing algorithms provide no adaptivity. *Fully adaptive* routing algorithms allow routing on all possible paths provided by the routing function and therefore require the most costly hardware resources. *Partially adaptive* routing algorithms attempt to curtail the amount of hardware requirements by restricting packets to be routed on a subset of all possible path selections [5]. Fully adaptive routing algorithms that implement deadlock-recovery techniques truly utilize all channel resources for routing packets. None of these resources are excluded for the sake of avoiding deadlock. These routing algorithms are referred to as *True Fully Adaptive Routing* (*TFAR*) algorithms [1,2,4]. Routing algorithms are also classified according to the way they protect against deadlock, which is the main obstacle that adaptive routing algorithms in wormhole switched networks have to overcome. *Deadlock-avoidance* algorithms restrict packet routing in a way that avoids the occurrence of deadlock. They define a necessary condition for preventing deadlocks, and they restrict routing so as to satisfy this condition at all times. This process insures freedom of deadlock. All the deterministic algorithms belong to this class of routing algorithms [3,6]. TFAR algorithms route packets without any restrictions, and are therefore susceptible to deadlock formations. These algorithms are accordingly classified as *deadlock-recovery* routing algorithms. TFAR algorithms must therefore be accompanied by a deadlock-recovery mechanism, which is invoked whenever deadlocks are suspected amongst a set of two or more packets [7–9]. Recent research efforts have determined that deadlocks are generally rare events, and that they are more likely to occur as the network reaches or is beyond its saturation point [7]. This observation has caused deadlock-recovery techniques to be more widely accepted as a viable alternative to deadlock-avoidance algorithms.

Although wormhole switching can achieve very fast switching of flits due to its light and efficient buffer requirements, it has some associated drawbacks. It generally suffers from high channel contention at high traffic load rates. This occurs because of the intertwined dependencies between blocked packets and their reserved channels, and buffer resources. This consequently leads to lower achievable throughputs for wormhole switched networks. TFAR algorithms with their efficient designs suffer some drawbacks as well. In certain cases, they exhibit severe performance degradation when the network is beyond its saturation point. Particularly, when using traffic patterns that tend to instigate more cyclic dependencies while routing. The performance degradation problem can be further exacerbated when deadlocks start to occur. If these deadlocks are not drained from the network as quickly as they form, they may lead to more deterioration of the performance and the stability of the network [8,9].

Injection limitation techniques were recently introduced in order to overcome the limitations of wormhole switching and fully adaptive algorithms. Injection limitation, as the name suggests, attempts to control or limit the injection of packets into networks that are suspected to be near their saturation points. These mechanisms strive to overcome the aforementioned problems, while still retaining the efficient nature of TFAR wormhole switching algorithms. The mechanisms should also be as dynamic and as intelligent as possible so as to cope with fluctuating network and traffic conditions, without causing any undesirable results such as reduced network throughput.

The proposed mechanism is designed for use in conjunction with TFAR algorithms. The Congestion Level Injection Control (CLIC) mechanism is implemented here in association with ZOMA, an efficient TFAR algorithm that we proposed in Ref. [9]. ZOMA implements an efficient deadlock-recovery mechanism that was shown to have equal performance to other popular and more expensive deadlock-recovery mechanisms [9]. The performance of our CLIC scheme is compared to that of a preferred injection limitation mechanism reported in the literature using simulation. The remaining sections in this paper are organized as follows. Section 2 provides the details about the network model and assumptions used in the simulation study. In Section 3, the proposed CLIC mechanism and associated logic is explained, Section 4 presents the details related to out performance evaluation including the simulation results and discussions. Finally, Section 5 contains the conclusions.

2. Network model

The interconnection network topology considered in this paper is a 256-nodes two-dimensional (2D), 16×16, mesh topology network. The network size was selected for reasonable simulation times. Each physical channel is composed of a FIFO buffer that is two-flits deep. Packets used are all 32-flits long, while flits are 32-bits wide. The selection function, which selects a single path from the set of all possible paths to route the packet through, will attempt to locate a free channel for the packet first. If that is not possible, it selects amongst the choices according to the *straight-first* flavor of selection. The timeout value, after which a packet is marked as deadlocked, is 10 clock cycles for all traffic patterns except the hot spot pattern, where a timeout value of 35 cycles is used. The traffic *generation rate* is used as the input parameter that determines the rate at which each node in the network generates packets. All the traffic rates are normalized with respect to the maximum wire capacity of the network topology. The maximum wire capacity is calculated as in Ref. [5].

The following are the assumptions made about the operation of the system and its interconnection network, as it will be used throughout the simulation modeling in this paper:

- A single injection channel between the processor and the router is used. If the processor generates a new packet while another is being injected, the latter packet is queued at the tail of the injection queue.
- Injection queues are allowed to grow without bounds. Latency figures calculated include source queuing time.
- A single delivery channel between the processor and the router is used. If more than one packet is being delivered, delivery channel is used in a round robin fashion amongst all of the delivering packets.
- Flits passed to the delivery channel are assumed to be consumed immediately by the processor unit.
- The router is connected to neighboring routers via dual uniplex channels, one for each incoming and outgoing direction, in a full-duplex fashion.
- Each uniplex channel is associated with at least one multiflit FIFO queue. This queue acts as the input queue of the receiving node (input buffered architecture.)
- A physical channel is associated with a configurable number of virtual channels. Physical channel bandwidth is shared amongst all the virtual channels in a demand-slotted round robin fashion.
- The phit size is equal to the flit size; therefore a flit can be transferred across the physical channel within one clock cycle.
- A header flit can be processed by the node's routing unit within one clock cycle.
- Only one header is processed by the routing unit at a time. If more than one header requires the routing unit, they are serviced in a round robin fashion.
- The crossbar of the router is non-blocking.

3. Congestion level injection control mechanism

Injection limitation techniques were recently developed to be used along adaptive routing algorithms in wormhole switched networks. In what follows we attempt to shed some light on some of the earlier efforts found in the literature related to this technique.

A basic injection limitation mechanism was used in Ref. [10]. The paper proposed using a fully adaptive routing algorithm based on the Duato model [6]. Virtual channels are divided into two classes, deterministic and adaptive channels. Injection of new packets is only allowed on a subset of the adaptive channels, which are referred to as the *source throttling* lanes. This mechanism introduces a crude injection limitation to the packet injection rate. This tends to alleviate the performance degradation problem at high network load rates, albeit in a basic or crude fashion.

Most recent injection limitation mechanisms proposed in the literature are based on controlling the number of utilized virtual channels within a node. Injection of new packets is only permitted when the number of occupied virtual channels in a particular node is less than a predetermined threshold value [11,12]. The value of the threshold is usually determined empirically via simulation just as the network enters the saturation point. By attempting to control the maximum allowed number of occupied virtual channels throughout the network, it is generally feasible to alleviate the performance degradation exhibited beyond the saturation point.

Injection limitation mechanisms can generally be classified into different categories depending mainly on how dynamic and adaptive the mechanisms are. Earlier injection limitation mechanisms were mainly *static* in nature. The threshold value must be modified for various traffic patterns and network conditions, which makes selecting the optimal value for this threshold quite difficult. If the threshold value imposes maximum restrictions, then there may be a performance penalty of higher latencies at normal network loads. While if there are less restrictions on the threshold value, then the mechanism may not be able to achieve its goal of preventing the performance degradation beyond the saturation point of the network [11,12]. To overcome the shortcomings of these injection limitation mechanisms in relation to requiring different settings for different traffic distributions, some mechanisms update the threshold value once, after guessing the traffic distribution being used to drive the network. The threshold value is computed using a linear function. These mechanisms are accordingly referred to as *semi-dynamic* injection limitation mechanisms [13]. When the injection limitation mechanism attempts to be dynamic and adaptive to current network load, networks conditions, and traffic distributions, then the mechanism is referred to as a *dynamic* injection limitation mechanism [8]. These mechanisms attempt to dynamically determine the appropriate threshold value depending on the current network and traffic conditions. Among the recent dynamic mechanisms reported in the literature is the DRIL mechanism, which exhibits excellent performance. A brief description of DRIL is provided next.

The *Dynamically Reduced message Injection Limitation* (*DRIL*) mechanism is a dynamic injection limitation mechanism. It is based on approximating network traffic by counting the number of busy virtual channels used within the nodes of the network, and dynamically adapting to this calculated network load. The DRIL mechanism has been

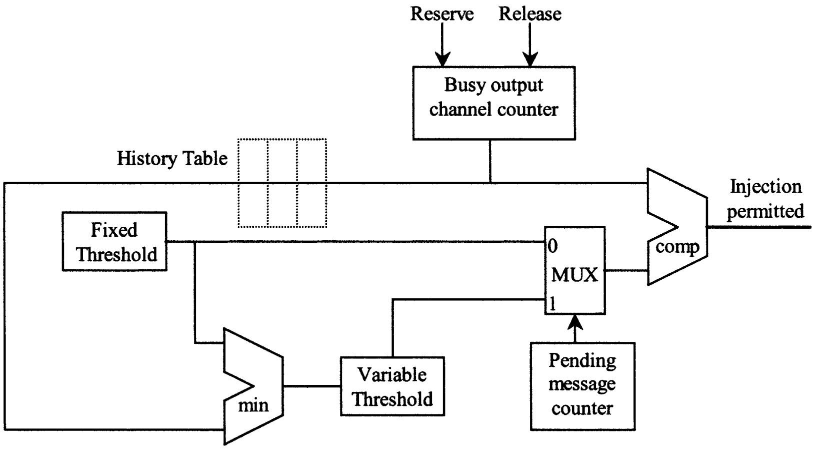

Fig. 1. DRIL injection limitation mechanism.

shown to outperform the other mechanisms that are based on counting the number of busy virtual channels, and we will therefore use it for comparison. The mechanism works simply by preventing the injection of new packets into the network if the number of busy virtual channels surpasses a predetermined threshold value. Since the optimal threshold value is dependent on several variables that range from network topology to packet length, it is determined empirically via simulation. The number of busy virtual channels is calculated just before the network reaches its saturation point. This value is then used as the basis for the initial threshold value of the mechanism. The threshold value is dynamically updated as a function of network load. Each node dynamically determines its own optimal value for the number of busy virtual channels by observing their count as the network enters its saturation point. Being near the saturation point is estimated by the number of packets that are queued for injection in a particular node. When the number of packets in local injection queues exceeds an empirically predetermined value, then a count of the number of busy virtual channels is performed. The obtained value will be used to dynamically update the initial threshold value. This threshold value is then used to limit the injection of new packets in order to maintain the network traffic to an acceptable level below the saturation point. Whenever traffic decreases then the fixed initial threshold value is used instead of the dynamically obtained value, as it imposes lower restrictions on the injection of packets allowed into the network. This allows the mechanism to dynamically adapt to changing network load and traffic. Also the newly calculated threshold value is used only if it leads to a more restrictive injection limitation policy, and is higher than the minimum needed to guarantee the possibility of injecting packets and therefore the elimination of starvation. All the empirical values will be determined via simulation. This mechanism also requires storing the previous *m* samples of the busy virtual channel count in order to have a better calculation of the dynamic threshold. Whenever a new packet is injected, the count of the number of busy virtual channels is stored in this table. The threshold value is then obtained by averaging the table contents when the network is determined to be near the saturation point. Fig. 1 provides an illustration of how the DRIL mechanism is implemented [8].

3.1. Proposed CLIC mechanism

The proposed *CLIC* mechanism is a dynamic mechanism that estimates the network load by sensing the relative congestion of packets as calculated by each channel of each node in the network. The operation of the CLIC mechanism is composed of distinguished components that are reviewed next.

3.1.1. Congestion level indicator (CLI)

Each node attempts to sense the congestion in the network by calculating the number of cycles consumed to route a particular packet on each of its output channels. The objective of this feature is to associate a particular congestion value with each output channel of every node in the network. This value is referred to as the *Congestion Level Indicator* (*CLI*) value and it is more precisely calculated using a counter that is reset when the header of the packet is routed through the channel. The counter is then incremented at each clock cycle until the tail flit of that packet is finally passed through the channel. At this point, the value of the counter is committed to a register that is used to indicate the CLI value associated with the channel at any particular point in time. Output channels with high CLI values indicate that there is a relatively high level of congestion or traffic that exist ahead of that channel. If the CLI value is high enough then it may indicate that the network is approaching its saturation point. Since each channel is composed of several virtual channels that are used to route different packets by multiplexing them on the same channel. We elect to calculate the CLI value for only one of the virtual channels in order to reduce the hardware requirements. The CLI value for only one of the virtual channels is representative of the

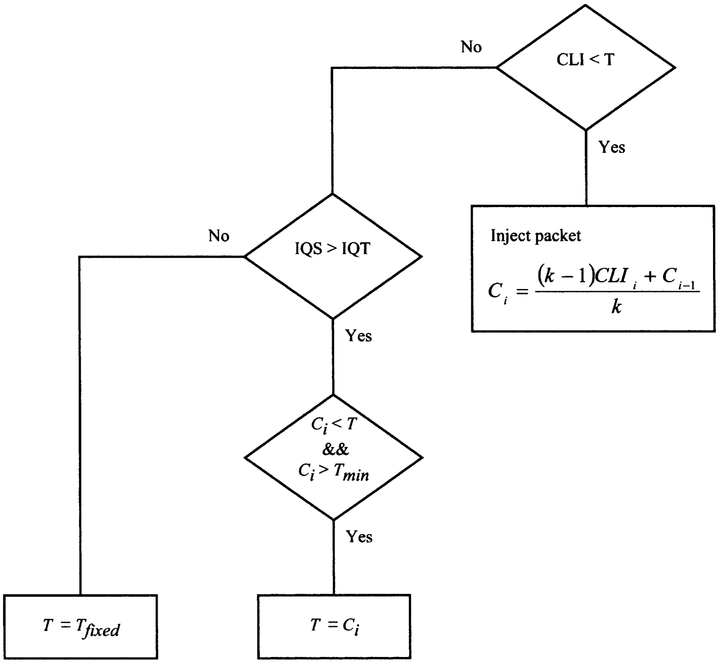
No
CLI < T
Yes
Inject packet
$C_i = \frac{(k-1)CLI_i + C_{i-1}}{k}$
No
IQS > IQT
Yes
$C_i < T$ && $C_i > T_{min}$
Yes
$T = T_{fixed}$
$T = C_i$

Fig. 2. CLIC injection limitation mechanism.

overall congestion level of the output channel because we are using TFAR, where all virtual channels of a particular channel are used in a similar fashion without any restrictions. Also all the virtual channels are multiplexed over the same physical channel, which factors the congestion level for all virtual channels within a single CLI value for a particular virtual channel. This mechanism requires, as many counters as there are channels. Therefore four counters and registers are required in our particular network configuration. In this mechanism virtual channel number 0 of each physical channel is used to represent that physical channel. Also all the CLI values are initially set to the packet length in flits, as it is the minimum possible value for a CLI in the absence of any contention or traffic from other packets in the network.

3.1.2. Injection control

The injection control or limitation mechanism operates at the packet routing level. When a new packet is selected for injection from the head of the injection queue, the routing unit that implements the adopted routing algorithm processes the packet. The routing function generates a list of profitable output channels that the packet can be routed through. The selection function then selects one of those output channels that the packet should be injected on. The injection control logic is then invoked on this selected output channel. If the CLI value for the selected channel is lower than a predetermined threshold T (initially set to T_{fixed}), then packet injection is allowed; otherwise the injection of the new packet is prevented during the current cycle. This logic is illustrated in Fig. 2.

The mechanism attempts to accurately calculate the CLI value at which the network is deemed to be near its saturation point by keeping track of the CLI value when packets are injected into the network. This is accomplished using the formula shown in Fig. 2. The value of C represents an indicator or average of all previous CLI values. It is an efficient method of acquiring an average of previous and recent CLI values without having to allocate many registers to hold several previous CLI values as implemented in the DRIL mechanism. The k variable used is referred to as the weight factor. The value of k used in the formula is 9, which provides more weight to the recent value of the CLI on the overall C expression.

When the network is suspected to be near its saturation point, as indicated by a predetermined number of packets that accumulate in the injection queue then the threshold is modified to be the latest calculated value of C. This is indicated by the *Injection Queue Size* (*IQS*) > *Injection Queue Threshold* (*IQT*) decision block. This branch is not executed unless the C value leads to a more restrictive injection policy, and it is above a predetermined minimum congestion threshold. This minimum threshold exists because certain levels of congestion are normal due to the packet length and the existence of other traffic in the network. The last branch of the logic in the chart is there to insure that the mechanism is dynamic in case the network load has decreased. If that happens then the network is not close to its saturation point and the threshold value is reset to the initial predetermined threshold value, which is referred to as T_{fixed} and it allows for a less restrictive injection control policy.

3.1.3. Injection selection function

The CLIC injection control mechanism prevents the

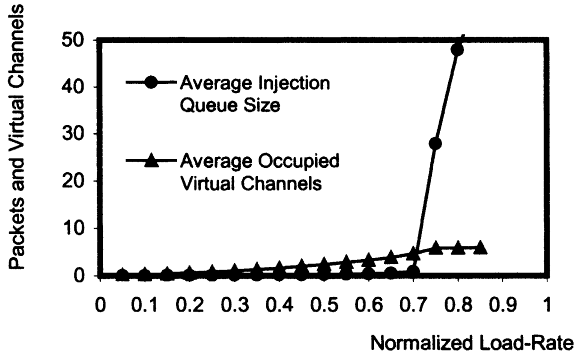

Fig. 3. 2D 16 × 16 mesh (uniform traffic).

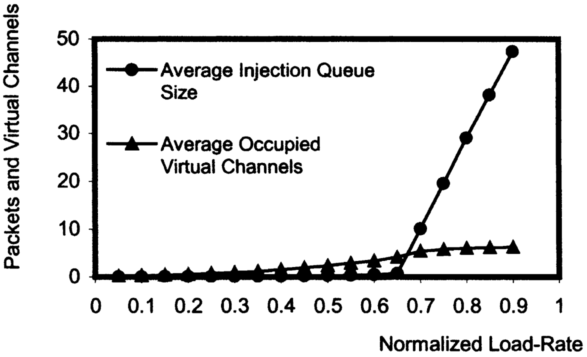

Fig. 4. 2D 16 × 16 mesh (bit-reversal traffic).

injection of a packet on a requested output channel if the CLI value associated with that channel is higher than a predetermined threshold. This mechanism should be complemented by a packet selection function that selects the channel with the lowest CLI value amongst all the profitable channels passed to it by the routing function. This is to safeguard against the unnecessary restriction on packet injection in the presence of profitable channels that have acceptable CLI values. This feature does not conflict with the straight-first selection function that is used throughout this effort. The straight-first selection function influences the channel selection decision that a packet in transit should be directed through, and not a newly injected packet.

4. Performance evaluation

This section evaluates the performance of the CLIC injection limitation mechanism against the performance of DRIL and the plain ZOMA algorithm. The simulator used is known as *WormSim* and it was developed as part of this effort. The two most important performance evaluation measures for interconnection networks are the latency and throughput metrics. The simulator only collects these two metrics after the network has reached the *steady state*. For more information with regard to the steady state detection refer to Ref. [9].

4.1. Empirical threshold determination

This section will briefly discuss the proper threshold values for both of the DRIL and the CLIC mechanisms.

4.1.1. DRIL thresholds

The DRIL mechanism depends mainly on the number of occupied virtual channels per node. The proper threshold value for the virtual channels count should be the number of virtual channels occupied just before the network enters into saturation. To determine this value using simulation, the average number of occupied virtual channels per node is plotted against the normalized load rate. Also, as both mechanisms need to determine a threshold for the number of packets that are pending in the injection queue when the network is deemed to be saturated, simulation is used to obtain the average injection queue size as a function of the normalized load rate. The injection queue size threshold will also be used for the CLIC mechanism. These two threshold values are plotted on the same charts. Fig. 3 shows the simulation results for both of these threshold variables plotted as a function of the normalized load rate for the uniform traffic pattern. From the figure we can determine that the network enters its saturation point at approximately 0.7 of the normalized load rate. The number of occupied virtual channels threshold value for this traffic pattern is approximated to be 6, while the injection queue size threshold value is approximately 10. It is important to note here that the threshold values obtained from the chart are only approximate values. The proper threshold values should be obtained by actual simulation run trials. The chart provides minimum asymptotic values for these thresholds. Table 1 provides the actual threshold values as determined by the simulation trials. Similarly, Figs. 4–6 display the simulation threshold values for the Bit-reversal, Dimension-reversal, and Hot spot traffic patterns, respectively.

Table 1
Injection limitation mechanisms threshold values

Threshold	Uniform	Bit-reversal	Dimension-reversal	Hot spot
Busy flit buffers	8	9	11	4
Injection queue size	10	10	10	10
Congestion threshold	170	180	170	130
Minimum congestion threshold	65	60	120	32

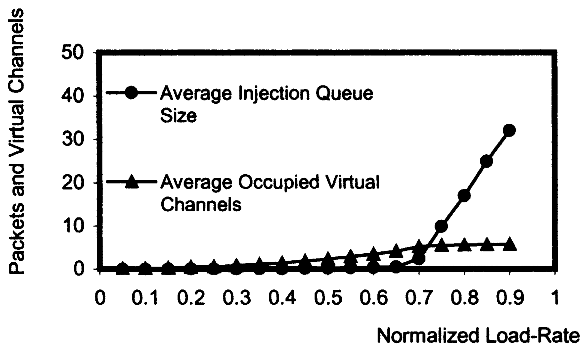

Fig. 5. 2D 16 × 16 mesh (dimension-reversal traffic).

4.1.2. CLIC thresholds

The CLIC mechanism needs to establish a congestion level threshold that occurs in the network just before it enters into the saturation phase. This value is determined empirically by calculating the average value for all those channels with the maximum congestion values per node in the network throughout a particular run. Accordingly, this threshold is referred to as the *Maximum* CLI threshold and it is plotted as a function of the normalized load rate. Certain levels of registered congestion are normal because of the packet length and multiplexing delays due to the presence of other packets in the network. Therefore we also need to empirically determine this normal level of congestion as described by the CLIC mechanism. This threshold is determined by calculating the average congestion values for all channels of all nodes in the network throughout a particular run. This threshold value is referred to as the *Average CLI* threshold and it is also plotted as a function of the normalized load rate. Figs. 7–10 show the plots of these threshold values in the case of uniform, bit-reversal, dimension-reversal, and hot spot traffic patterns, respectively. It is important to note that these plots only provide asymptotic boundaries for the threshold values. The precise values of these thresholds still require some simulation trials in order to find their optimum value. Table 1 shows these threshold values that were determined via simulation and were used in order to produce the simulation results of the next section.

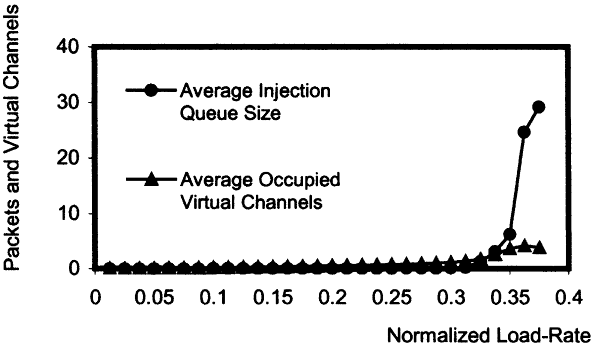

Fig. 6. 2D 16 × 16 mesh (hot spot traffic).

4.2. Uniform traffic

This is the most popular traffic distribution used in evaluating interconnection networks. Packet destinations are chosen uniformly amongst all the nodes in the network. More specifically the probability of a node i sending a packet to node j is the same for all i and j, where $i \neq j$. The performance results of the ZOMA algorithm along with the CLIC and DRIL injection limitation mechanisms are shown in Figs. 11 and 12.

The performance charts show that major performance gains are achieved when applying the CLIC or DRIL injection limitation mechanisms over the plain ZOMA algorithm. The plain ZOMA algorithm saturates around 0.7 of wire capacity. CLIC saturates around 0.85, while DRIL saturates around 0.9 of wire capacity. This corresponds to a 21 and 29% of performance improvement over the plain ZOMA, respectively. In addition to the performance improvement demonstrated, the injection limitation mechanisms are effective in overcoming the steep performance degradation behavior displayed by the ZOMA algorithm in Fig. 12.

Reviewing Fig. 11, we notice that the DRIL mechanism demonstrates favorable performance results than the CLIC mechanism. The figure suggests that DRIL has a 6% performance improvement over CLIC. But by carefully examining Fig. 12, we notice that CLIC demonstrates higher throughput characteristics than DRIL, which makes the direct comparison of both of these mechanisms non-trivial. In Fig. 12, CLIC exhibits roughly 2% higher throughput than DRIL. The performance gain in throughput and latency are not directly comparable. The injection limitation mechanism attempts to throttle the amount of traffic in the network in order to allow for the smooth delivery of packets throughout the network. In other words these mechanisms attempt to balance the amount of throughput delivered by the network as opposed to the latency incurred by the packets that are being routed through the network. A good injection limitation mechanism should attempt to maximize the achievable throughput in relation to the obtained latencies. Therefore in order to evaluate the performance of these injection limitation mechanisms, both of these metrics have to be considered simultaneously. Experimental observation shows that even slight improvements in throughput are more difficult to achieve than the corresponding latency metric. Empirical results suggest that a 1% improvement in throughput corresponds roughly to 10% improvement in the latency figures. If we calculate the various latency figure improvements between the two injection limitation mechanisms, we arrive at an average of 18% improvement in latency for DRIL over CLIC. Weighing that against a 2% throughput enhancement for CLIC over DRIL, we can arrive to the conclusion that the overall performance characteristics of both of these injection limitation mechanisms are somewhat similar. But we should also tip the performance characteristics slightly in favor for DRIL especially at high load rates. We can explain this observation as

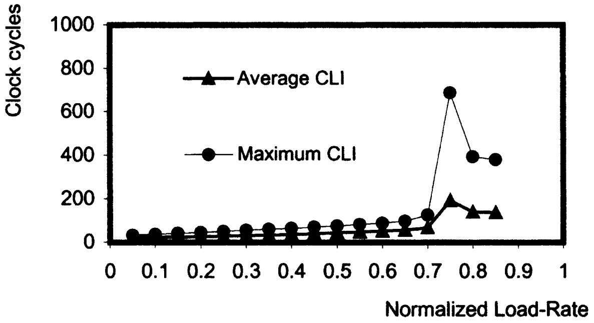

Fig. 7. 2D 16 × 16 mesh (uniform traffic).

follows. The random traffic pattern demonstrates sharp saturation curves due to the cyclic routing behavior of fully adaptive routing when used with the uniform pattern. Presented with this traffic pattern, a strict injection limitation policy such as DRIL may be more effective in avoiding saturation especially at high load rates. This is because when DRIL detects a high number of busy virtual channel buffers, it prevents the entire node from injecting any packets. While in CLIC each channel individually detects the high level of congestion ahead of it and prevents the injection of packets on that channel only. Other channels that are connected to the same node may not yet detect the same level of congestion and therefore continue to inject packets into the network. This behavior by design will have a slower reaction than that in DRIL in response to a high level of traffic in the network, but it allows for the network to accept more traffic. This explains why CLIC demonstrates higher throughput but somewhat higher latency figures. This observation will also be used to explain the behavior for the other traffic patterns used in the next section.

4.3. Non-uniform traffic

This section will present the simulation results for the different non-uniform traffic, namely, the bit-reversal, dimension-reversal, and the hot spot traffic patterns.

Bit-reversal traffic. Using bit-reversal traffic, a node with binary address a_{n-1}, a_{n-2},…,a_1, a_0 sends a packet to a node with binary address a_0, a_1,…,a_{n-2}, a_{n-1}. This traffic pattern causes nodes on certain rows to send packets to nodes on certain columns, causing various pockets of conflicts near the center of the network. The simulation results obtained using this traffic pattern are shown in Figs. 13 and 14. From the figures, we can make the following observations. First, the performance improvement of the injection limitation mechanisms is substantial. Second, the CLIC mechanism outperforms DRIL in this non-uniform traffic pattern in contrast to what was observed earlier in the uniform traffic pattern. CLIC saturates approximately around 0.9, while DRIL saturates around 0.85, and plain ZOMA saturates around 0.65 of the normalized load rate. This corresponds to a 6 and 38% of performance gain for CLIC over DRIL and the plain ZOMA, respectively.

We can also notice from Fig. 14 that the throughput performance is improved over the plain ZOMA, just as in the uniform traffic pattern, although the bit-reversal traffic pattern does not actually demonstrate sharp performance degradation beyond saturation in the plain ZOMA case. This is due to the previously mentioned behavior that this traffic pattern does not lend itself to forming cyclic dependencies while routing. This means that these injection limitation mechanisms improve the performance of the plain ZOMA algorithm even for those traffic patterns that do not cause serious performance degradation beyond the saturation point. More importantly, as can be observed by the chart, the CLIC mechanism achieves slightly higher throughput than DRIL, even that it already demonstrates better latency results. This makes the performance

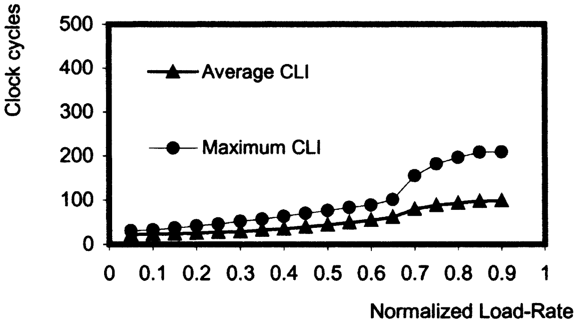

Fig. 9. 2D 16 × 16 mesh (dimension-reversal traffic).

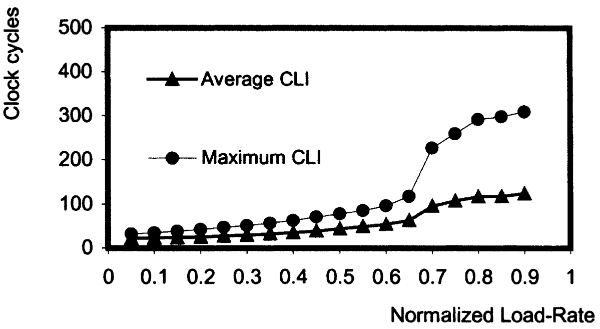

Fig. 8. 2D 16 × 16 mesh (bit-reversal traffic).

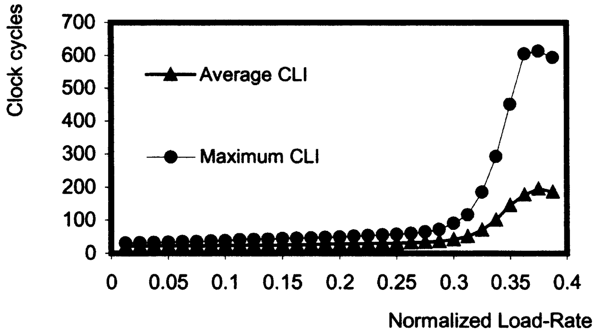

Fig. 10. 2D 16 × 16 mesh (hot spot traffic).

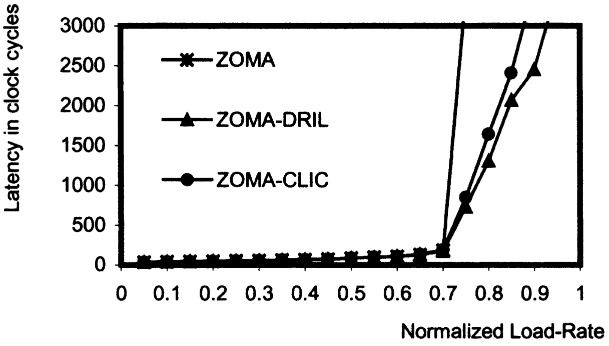

Fig. 11. 2D 16 × 16 mesh (uniform traffic).

characteristics of CLIC even more favorable than those displayed by DRIL.

Dimension-reversal traffic. The dimension-reversal traffic pattern causes a node with address *xy* to send packets to node *yx*. For 2D networks this is the same as the matrix transpose traffic pattern. It has a similar effect to the bit-reversal pattern, but concentrates the conflicts along the diagonal line of the network. The simulation results are shown in Figs. 15 and 16.

The charts show that CLIC has a clear performance advantage over DRIL, and that both of them outperform the plain ZOMA algorithm. Plain ZOMA saturates around 0.7, DRIL around 0.8, while CLIC saturates at around 0.9 of wire capacity. This corresponds to about a 29 and a 13% performance gain for CLIC over plain ZOMA and DRIL, respectively. Fig. 16 shows that the throughput is not only generally improved at high load rates using these mechanisms, but the throughput achieved by CLIC is even slightly higher than that demonstrated by DRIL. This leverages the performance gain of CLIC even higher. These and the previous results of the bit-reversal traffic pattern confirm the trend that CLIC outperforms DRIL for the non-uniform traffic patterns for the same reasons that the latter outperformed CLIC for the uniform traffic pattern. More precisely it is the fact that CLIC senses the abnormally high congestion per each individual channel of each node, which allows each node to effectively prevent or continue injection on different channels depending on the state of the network near those channels. Non-uniform traffic patterns, which more closely resemble the behavior of real parallel applications, tend to concentrate traffic non-uniformly throughout the network. This causes the channels to be loaded variably. Manipulating the injection of packets on a more granular level than the node unit, which is the channel unit in this case can lead to achieving much higher performance characteristics as the simulation results confirm.

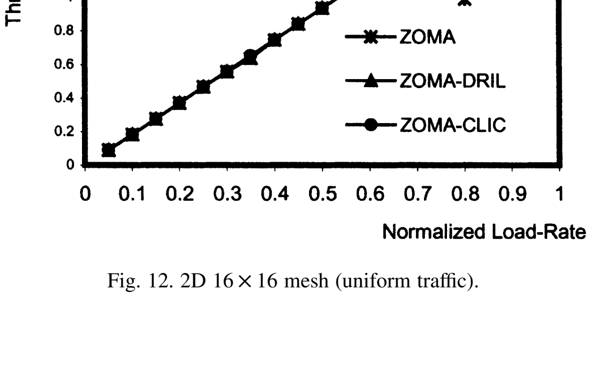

Fig. 12. 2D 16 × 16 mesh (uniform traffic).

There is yet another unfavorable characteristic that can be observed about the DRIL mechanism. Fig. 15 shows that the performance of DRIL is inconsistent as evident by the anomaly in the latency graph around the 0.85-loading factor. This behavior could not be avoided using different thresholds without seriously sacrificing the performance gains of the DRIL mechanism. This occurs in DRIL as the whole node is prevented from injecting packets into the network when a certain threshold is reached. If the ongoing calculations of the threshold values are not dynamic enough, it may lead to long periods of dormant node behavior. When this occurs throughout the network, it can collectively disturb the obtained results into being inconsistent or can even more drastically lead to the starvation of certain nodes. The starvation possibility was actually mentioned in Ref. [8], but neither its effects on the stability of the network were mentioned and nor a solution for the problem were offered in the paper. In order to solve this problem, extra hardware is needed in the form of timeout counters and registers. The function of these counters is to reactivate a particular node if it has been dormant longer than a predetermined timeout value regardless of the injection limitation threshold value. The DRIL mechanism did not include this extra hardware in its logic. This problem is avoided in CLIC by having the injection control mechanism operate at the channel level. This provides ample flexibility to the node as it is allowed to control the injection of packets along different channels.

Hot spot traffic. The hot spot traffic pattern used directs 5% of all the network traffic toward a single node that is randomly selected, while the remaining traffic is uniformly distributed. This pattern causes serious congestion near the hot spot node, which eventually propagates throughout the network. This causes all the routing algorithms to have early saturation points. The *hot spot* case is the most difficult traffic pattern for the network to manage. The simulation results are shown in Figs. 17 and 18.

The charts show that CLIC outperforms DRIL on both accounts of latency and throughput. Taking the throughput as well as the latency figures into account, it is approximated that CLIC saturates at around 0.375, while DRIL saturates at 0.35, and plain ZOMA saturates at 0.3375 of the normalized load rate. This provides CLIC with a 7 and 11% performance improvement over DRIL and the plain ZOMA, respectively. The performance advantage is more apparent when observing the throughputs achieved by the two mechanisms. CLIC demonstrates higher and more stable throughput behavior than DRIL. This is in line with previous observations that CLIC outperforms DRIL for

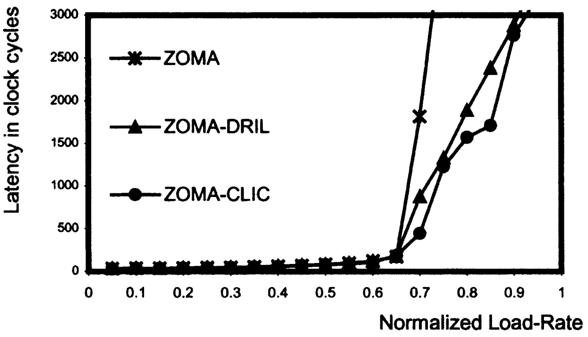

Fig. 13. 2D 16 × 16 mesh (bit-reversal traffic).

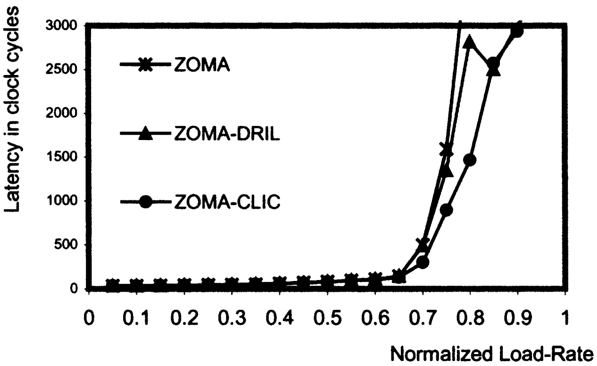

Fig. 15. 2D 16 × 16 mesh (dimension-reversal traffic).

the non-uniform traffic patterns and for the same reasons mentioned earlier.

Also the performance figures confirm the previous notion that the performance obtained by the DRIL mechanism is inconsistent as apparent by the anomaly observed at 0.325-load factor. Around that point, the performance of DRIL is surprisingly even worse than that of the plain ZOMA algorithm. Experimenting with various other DRIL threshold values did not eliminate this behavior, but rather worsened it. This behavior can be explained by the harsh reaction of preventing the entire node from injecting any packets, while only certain channels of the node are being overloaded. To support this argument, it can be further observed that the throughput provided by DRIL is lower than that provided by the plain ZOMA algorithm at that mentioned traffic rate. This confirms that DRIL allowed less traffic to be injected into the network, as did the plain ZOMA algorithm. In addition to this weak performance, DRIL is not successful in totally eliminating the degraded performance behavior that occurs after reaching the saturation point, as evident by the dip in throughput at high load rates.

Another essential function of an injection limitation mechanism, especially as it is used along TFAR algorithms, is its ability to reduce the frequency of deadlocks in the network. To assess the performance of CLIC in this regard, the frequency of deadlocks detected by the two injection limitations mechanisms is collected and compared against that obtained for the plain ZOMA algorithm. This is obtained for the traffic distribution that causes more deadlocks to occur, namely the hot spot traffic pattern. Fig. 19 shows this comparison. More specifically what is shown is the number of deadlocks detected normalized to the total number of packets delivered by the network for the various load rate factors. From the chart it is clear that CLIC provides the best performance with regard to the frequency of deadlocks detected. As can be seen both mechanisms were able to reduce the frequency of deadlocks, with CLIC providing the lowest number of deadlocks detected in the network. This is in spite of the fact that CLIC provided the highest throughput for all traffic patterns evaluated. This combination is significant since CLIC did not reduce the frequency of deadlocks by the reducing the overall traffic accepted by the network. Rather it allowed more traffic into the network, and was able to deliver this higher payload with less number of deadlocks. This again highlights the intelligent logic of the CLIC injection limitation mechanism that is due to controlling the injection at the individual channel level.

5. Conclusions

This paper presented a new injection limitation mechanism that we named CLIC. The CLIC mechanism can be used in conjunction with any TFAR algorithm in order to stabilize and boost its performance. CLIC prevents the injection of an excessive amount of traffic in the network by locally monitoring the level of congestion experienced along each

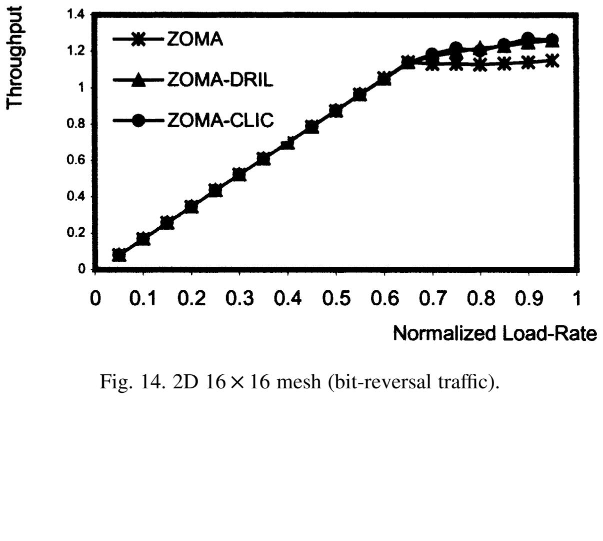

Fig. 14. 2D 16 × 16 mesh (bit-reversal traffic).

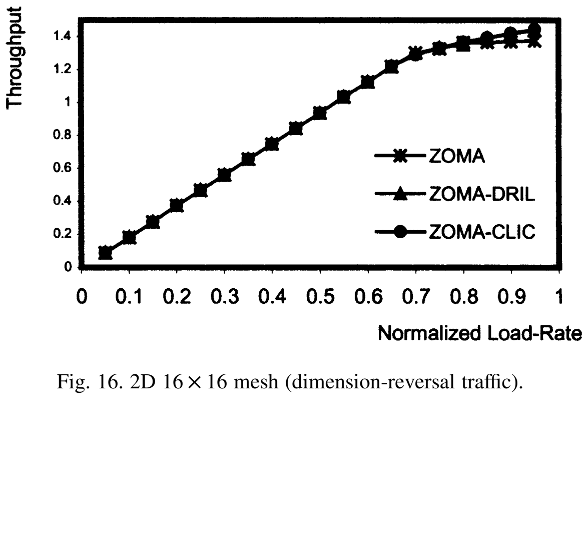

Fig. 16. 2D 16 × 16 mesh (dimension-reversal traffic).

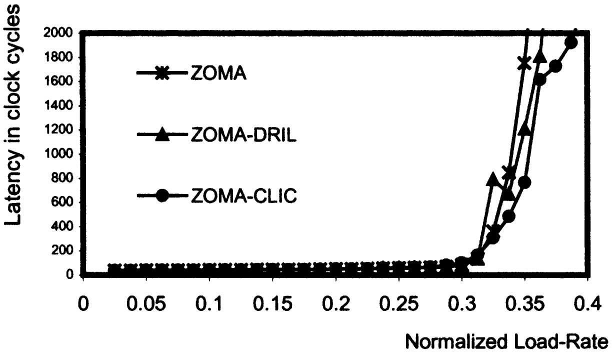

Fig. 17. 2D 16 × 16 mesh (hot spot traffic).

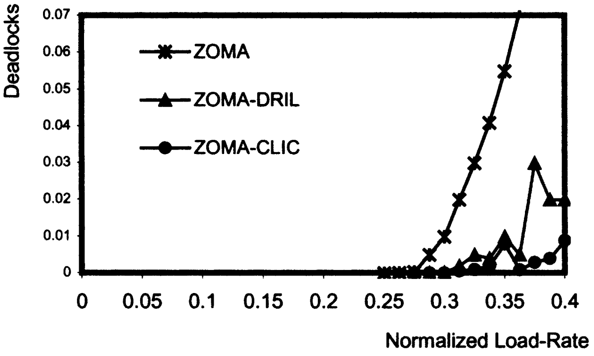

Fig. 19. Frequency of deadlocks in hot spot traffic.

individual channel by those packets that are being routed through it, and subsequently preventing the injection of new packets on that channel if the congestion level is determined to be higher than the network can manage efficiently. CLIC have demonstrated superb performance improvements over DRIL in almost all traffic pattern cases and performance metrics as shown by the simulation results. The excellent performance characteristics of CLIC were obtained due to its smart logic. More specifically applying injection limitation at the channel level rather than the node level allowed CLIC to achieve superior performance in all traffic patterns that load the network non-uniformly. These traffic patterns more closely resemble the behavior of real parallel applications.

CLIC also successfully demonstrated that it achieved the two most important objectives of any injection limitation mechanism. Mainly, it maintained the performance stability of the network at high load rates, and was able to eliminate severe performance degradation behavior beyond the saturation point. Second, CLIC was also successful in reducing the frequency of deadlocks detected by the network. This was achieved by preventing the network from reaching a highly congested state where cyclic dependencies start to frequently form. These two significant features of this injection limitation mechanism compliment true fully adaptive algorithms in general. This will enhance the adoption of TFAR algorithms with low-cost and efficient deadlock-recovery design, such as ZOMA, and they will be even more widely acceptable for application in wormhole networks.

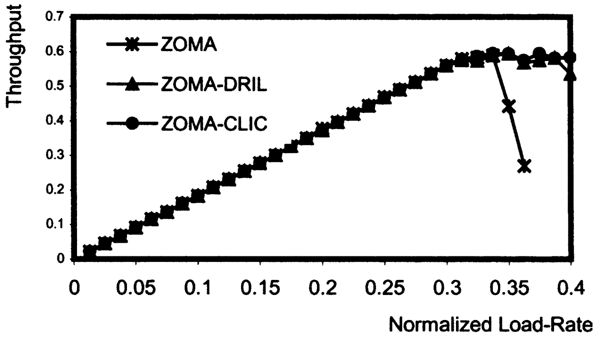

Fig. 18. 2D 16 × 16 mesh (hot spot traffic).

For future research in this area, we are investigating a new application for the CLI counters that were used as part of the CLIC mechanism. These counters could be utilized to create a new selection function. The suggested selection function will select from the possible set of channels that channel with the least CLI value associated with it in order to route the packet. Research has shown that no single set of selection functions provided superior performance consistently for all conditions and traffic patterns. Moreover, tuning the selection function to the applied workload can significantly improve the performance. The suggested future work proposed here attempts to solve this problem by enabling the selection function to always select the output channel with the least congestion that may be presented to the packet as it travels through the network. We expect that the implementation of this selection function may yield substantial performance improvement results in addition to what has been demonstrated by the CLIC mechanism.

References

[1] J. Duato, S. Yalamachili, L. Ni, Interconnection Networks: an Engineering Approach, IEEE Computer Society Press, 1997.

[2] L.M. Ni, P.K. McKinley, A survey of wormhole routing techniques in direct networks, IEEE Computer 26 (2) (1993) 62–76.

[3] W.J. Dally, C.L. Seitz, Deadlock-free message routing in multiprocessor interconnection networks, IEEE Transactions on Computers C-36 (5) (1987) 547–553.

[4] A.A. Chien, A cost and speed model for *k*-ary *n*-cube wormhole routers, IEEE Transactions on Parallel and Distributed Systems 9 (2) (1998) 150–162.

[5] J.H. Kim, Planar-Adaptive Routing (PAR): low-cost adaptive networks for multiprocessors, Master Thesis, University of Illinois at Urbana-Champaign, 1993.

[6] J. Duato, A necessary and sufficient condition for deadlock-free adaptive routing in wormhole networks, IEEE Transactions on Parallel and Distributed Systems 6 (10) (1995) 1055–1067.

[7] S. Warnakulasuriya, T.M. Pinkston, Characterization of deadlocks in interconnection networks, Proceedings of the 11th International Parallel Processing Symposium, April 1997, pp. 80–86.

[8] P. Lopez, J.M. Martinez, J. Duato, DRIL: dynamically reduced message injection limitation mechanism for wormhole networks,

Proceedings of the 1998 International Conference on Parallel Processing, August 1998, pp. 535–542.

[9] Z.H. Al-Awwami, M.S. Obaidat, M. Al-Mulhem, A new deadlock recovery mechanism for fully adaptive routing algorithms, Proceedings of the 19th IEEE International Performance, Computing, and Communications Conference, February 2000, pp. 132–138.

[10] F. Petrini, M. Vanneschi, Minimal adaptive routing with limited injection on toroidal *k*-ary *n*-cubes, Proceedings of the 1996 Supercomputing Conference, Article No. 23, 1996.

[11] P. Lopez, J. Duato, Deadlock-free adaptive routing algorithms for the 3D-Torus: limitations and solutions, Proceedings of the Parallel Architectures and Languages Europe 93, June 1993, pp. 684–687.

[12] F. Petrini, J. Duato, P. Lopez, J.-M. Martinez, LIFE: a limited injection, fully adaptive, recovery-based routing algorithm, Proceedings of the Fourth International Conference on High Performance Computing, December 1997, pp. 589–595.

[13] P. Lopez, J.M. Martinez, and J. Duato, F. Petrini, On the reduction of deadlock frequency by limiting message injection in wormhole networks, Proceedings of the Parallel Computer Routing and Communications Workshop, June 1997, pp.295–307.