Characterization of Deadlocks in Irregular Networks

Sugath Warnakulasuriya
WorkPlace Systems, 36 West Colorado Boulevard, Suite 206, Pasadena, California 91105
E-mail: sugath@workplace-systems.com

and

Timothy Mark Pinkston
SMART Interconnects Group, University of Southern California, Los Angeles, California 90089-2562
E-mail: tpink@charity.usc.edu

Received July 12, 1999; revised June 27, 2001; accepted July 12, 2001

This work characterizes how various network parameters influence message blocking and deadlocks in irregular networks. Information on blocking behavior is provided that is useful in making design trade-offs between restricting routing freedom and allowing the possibility for deadlocks to form in irregular networks. This paper also identifies ways in which a network’s susceptibility to deadlock can be reduced and provides guidelines for designing irregular networks that maximize routing flexibility and resource utilization. Finally, a new empirical evaluation methodology for classifying irregular topologies and relating network behavior to various classes of network topologies is introduced.

1. INTRODUCTION

The recent emergence of powerful workstations and high-performance off-the-shelf network switches has made it possible to use networks of workstations (NOWs) as low-cost alternatives to custom-built multiprocessors. NOWs connect small groups of processors to a network of switching elements that can form

1 This research was supported in part by NSF Career Award, Grant ECS-9624251, and an NSF grant, CCR-9812137.
irregular topologies, thus accommodating the wiring flexibility and incremental expansion requirements of local area network (LAN) settings. Examples of commercially available switches for use in these types of networks include Autonet [1], Myrinet [2], ServerNet [3], SGI Spider [4], and the Intel Cavallino router [5]. In designing NOWs, one of the primary goals is to provide high message throughput while maintaining low message latency by efficiently utilizing network resources. However, the irregularity of the interconnection patterns in NOWs make this a challenging task.

A number of important techniques for accomplishing efficient message routing in regular networks for multicomputers, including cut-through switching, virtual channels, and adaptive routing, have been developed in recent years [6]. All three of these techniques have migrated into high-performance switches intended for implementing NOWs. In implementing these techniques, the critical issue of deadlock behavior must be addressed. Deadlocks in routing can occur when messages block waiting for channel resources while holding onto other channel resources. Cyclic dependencies formed by groups of such blocked messages can lead to the indefinite postponement of messaging activities and, if allowed to persist, can bring the entire system to a halt. Routing approaches for addressing deadlock can be based either on avoiding deadlock or on recovering from deadlock.

Deadlock avoidance techniques proposed for irregular networks range from those that impose partial ordering of channel resources in order to altogether prevent resource dependency cycles [1, 7] to those that provide some adaptivity along with a set of escape resources [8]. Very few unrestricted deadlock recovery routing schemes have been proposed for irregular networks [9]. The primary distinction between deadlock avoidance and deadlock recovery approaches is the decision made in trading off routing freedom and deadlock formation. The circumstances in which either deadlock avoidance or deadlock recovery routing is preferable depend both on the frequency with which deadlocks occur and on the degradative effects of correlated message blocking behavior that can lead to deadlock. Therefore, designing an optimal approach to handling deadlock requires thorough understanding of the relationship between network design parameters, deadlock, and other harmful correlated message blocking phenomena. Previous approaches for handling deadlock in irregular networks have been proposed without the benefit of this important insight.

In this paper, network design parameters that significantly influence deadlock formation are identified and their effects on message blocking and deadlock are empirically evaluated through simulation for a class of irregular networks representing an “average” case. This increases our understanding of the nature and likelihood of deadlocks and provides insight into the implications of deadlock forming phenomena on network and routing algorithm design. The next section provides background on the type of message blocking and deadlock phenomena that can occur in irregular interconnection networks. Section 3 presents the evaluation approach and empirical results. Section 4 discusses the results. Section 5 presents related work, and, finally, conclusions are presented in Section 6. Implementation of the network simulator used in this study is described briefly in the Appendix.
2. MASSAGE BLOCKING AND DEADLOCKS IN IRREGULAR NETWORKS

2.1. Depicting Deadlocks

A model of consumer–resource interaction developed in previous work [10] uses channel wait-for graphs (CWGs) to represent resource allocations and requests existing within the network at a given point in time. The model supports various switching techniques including wormhole, buffered wormhole, and virtual cut-through as well as various policies for buffer usage, including atomic and non-atomic buffer allocation [11]. In this work, CWGs are used in a similar fashion for describing deadlocks and correlated message blocking, and for detecting and characterizing deadlocks in irregular networks.

Figure 1 illustrates a CWG depicting resource allocations and requests existing at a certain point in time in an irregular network. The source and destination of message \( m_i \) are labelled \( s_i \) and \( d_i \) in the network graph, and vertices \( v_c \) in the CWG represent channels (physical or virtual). Solid and dashed arcs between vertices represent the dependencies between resources caused by messages acquiring and requesting them, respectively. The resource model represented by this CWG corresponds to the “OR request model” of the consumer–resource interaction [12]. It has been established for distributed systems in general [13] and interconnection networks in particular [10] that, for the OR request model, a wait-for graph containing a knot contains a deadlock. Each instance of a knot formed within the corresponding CWG of a network is considered a unique deadlock event. Hence, the terms knot and deadlock are used interchangeably.

Previous work [14, 15] further establishes that deadlocks can be characterized by three attributes: deadlock set, resource set, and knot cycle density. The deadlock set is the set of messages that own the virtual channels involved in the knot and represents the expanse of the deadlock in terms of the messages involved. The resource set is the set of all virtual channels owned by members of the deadlock set and represents the expanse of the deadlock in terms of the resources involved. The knot cycle density represents the number of unique cycles within a knot. It is a useful attribute for representing the complexity in correlated resource dependency required for deadlock to form.

2.2. Reducing the Probability of Deadlock

A key factor that influences the probability of deadlock formation in irregular interconnection networks is routing freedom. Routing freedom corresponds to the number of routing options available to a message being routed at a given node within the network. It can be increased by adding physical channels, implementing multiple virtual channels per physical channel, and/or increasing the adaptivity of the routing algorithm. In irregular networks, the total number of channels can be increased by adding them either arbitrarily in a random fashion (e.g., to satisfy minimum and maximum node degree constraints) or uniformly in a regular fashion (i.e., using virtual channels). The former does not necessarily increase routing
FIG. 1. (a) A "multicycle deadlock" consisting of messages $m_1 \cdots m_7$ formed under minimal adaptive routing in an irregular network. (b) The set of vertices $\{vc_1 \cdots vc_{10}\}$ forms a knot in the CWG.
FIG. 2. (a) A single-cycle deadlock consisting of messages $m_1, \cdots, m_4$ formed under minimal adaptive routing in an irregular network. (b) The corresponding CWG.

Shown in Fig. 2 is a single-cycle deadlock formed under minimal adaptive routing. The amount of adaptivity provided by a routing algorithm for packets in irregular network topologies can be very much location dependent due to a possibly uneven distribution of physical channels [16]. Because of this, the same deadlock can still form even when the number of channels is doubled and when channels are placed in the network so as to quadruple the minimum node degree of the network, as shown in Fig. 3. This is because the routing freedom of only some messages in the network is increased when this blocked configuration forms. However, by uniformly increasing the number of channels in the network (i.e., using virtual channels or uniform placement of physical channels), the routing options for all packets in the network whereas the latter does, as illustrated in Figs. 2–4.

FIG. 3. (a) The same deadlock forms even when the number of channels has doubled and the minimum node degree has quadrupled. (b) The corresponding CWG.
FIG. 4. (a) A cyclic nondeadlock consisting of messages \( m_1 \cdots m_4 \) formed under minimal adaptive routing in the irregular network of Fig. 2 with two virtual channels (alternatively, uniform doubling of physical channels). (b) The corresponding CWG.

packets in the network are increased—including those in this blocked configuration, as shown in Fig. 4. Increasing the routing adaptivity increases flexibility in the use of physical and virtual channel resources by packets, reducing the chances of deadlock formation. Although it is still possible for a multicycle deadlock to form (as shown in Fig. 5), the number of messages and degree of correlation required amongst the messages increase so much that this occurrence becomes highly improbable.

As routing freedom is increased, the probability that messages block in the network decreases. More importantly, the degree of correlation required among

FIG. 5. (a) A multicycle deadlock consisting of messages \( m_1 \cdots m_9 \) formed under minimal adaptive routing in the irregular network of Fig. 2 with two virtual channels (alternatively, uniform doubling of physical channels). (b) The corresponding CWG.
blocked messages to form a knot increases substantially, although the number of cycles that can form when messages block also increases (i.e., at network saturation). Given enough routing freedom, this correlation requirement offsets the opposing effect on deadlock probability caused by the increase in the number of resource dependency cycles. For instance, networks with little-to-no routing freedom have a one-to-one correspondence between cycles and deadlocks, i.e., single-cycle deadlocks form. However, networks with greater routing freedom can offset the opposing effects from cyclic blocking of messages as a large number of cycles can exist without the formation of deadlock; i.e., cyclic nondeadlocks form. This effect of routing freedom is substantiated empirically in the next section by quantifying the influence of various network design parameters on deadlock-forming phenomena for a class of irregular networks.

3. EMPIRICAL STUDY OF DEADLOCKS IN IRREGULAR NETWORKS

3.1. Evaluation Methodology

The approach used for precisely detecting deadlocks is based on the theoretical framework described in [10]. A deadlock detection algorithm implemented in IRFlexSim is used to identify knots within the CWG of an ongoing network simulation. It correctly distinguishes between congestion and deadlocks, precisely detecting and characterizing all deadlocks occurring during simulation. A number of topological analysis tools are also used to narrow the scope of networks evaluated to specific classes of irregular networks. This allows for more meaningful results to be obtained.

3.1.1. Classification of Irregular Topologies

A new approach that classifies irregular network topologies based on relevant network attributes is used that strongly associates empirical results to certain classes of irregular topologies [17]. In this approach, network configurations specify the number of router switches, physical links, and minimum/maximum switch degree constraints. Six network attributes shown in Table 1 are used to reflect the quality of an interconnection pattern. The average number of routing options at each node and the average number of alternate minimum paths per node pair give an indication of routing freedom. The network diameter and the average internode distance give an indication of routing resource requirements. Finally, the intercluster bandwidth index (ICBI) and the intercluster link-cost index (ICLI) give an indication of link distribution within and between regions of high connectivity (i.e., clusters) and traffic load distribution on links between clusters, respectively [16]. The smaller the ICBI and the larger the ICLI, the more strongly clustered the network is and the

IRFlexSim is a flit-level simulator for irregular networks that is based on the FlexSim regular network simulators previously developed by the SMART Interconnects Group at USC.
### TABLE 1

**Summary of Minimum, Maximum, and Mean (in bold) Values for Selected Attributes of Randomly Generated Topologies of Each Configuration**

<table>
<thead>
<tr>
<th>Network Configuration Identifier</th>
<th>Average number of routing options at each node</th>
<th>Average number of alternative minimal paths per node pair</th>
<th>Average diameter</th>
<th>Average internode distance</th>
<th>Intercluster bandwidth index [16]</th>
<th>Intercluster link-cost index [16]</th>
<th>Percent of topologies with all attributes within ±5% of mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>64:64</td>
<td>1.00</td>
<td>1.00</td>
<td>10.00</td>
<td>5.45</td>
<td>0.1034</td>
<td>0.9548</td>
<td>3.0%</td>
</tr>
<tr>
<td>64:128</td>
<td>1.01</td>
<td>1.03</td>
<td>12.90</td>
<td>6.06</td>
<td>0.2761</td>
<td>1.1359</td>
<td>3.2%</td>
</tr>
<tr>
<td>64:256</td>
<td>1.02</td>
<td>1.30</td>
<td>17.00</td>
<td>7.66</td>
<td>0.4545</td>
<td>1.2930</td>
<td>1.0%</td>
</tr>
<tr>
<td>64:512</td>
<td>1.37</td>
<td>1.64</td>
<td>5.00</td>
<td>3.01</td>
<td>0.0079</td>
<td>0.6532</td>
<td>3.2%</td>
</tr>
<tr>
<td></td>
<td>1.46</td>
<td>1.91</td>
<td>6.29</td>
<td>3.12</td>
<td>0.1087</td>
<td>0.9768</td>
<td>1.0%</td>
</tr>
<tr>
<td></td>
<td>1.55</td>
<td>2.24</td>
<td>9.00</td>
<td>3.34</td>
<td>0.3763</td>
<td>1.2931</td>
<td>1.0%</td>
</tr>
<tr>
<td>64:128</td>
<td>2.16</td>
<td>2.68</td>
<td>4.00</td>
<td>2.20</td>
<td>0.0159</td>
<td>0.7871</td>
<td>1.0%</td>
</tr>
<tr>
<td></td>
<td>2.25</td>
<td>2.92</td>
<td>4.01</td>
<td>2.25</td>
<td>0.0811</td>
<td>1.0217</td>
<td>1.0%</td>
</tr>
<tr>
<td></td>
<td>2.36</td>
<td>3.23</td>
<td>5.00</td>
<td>2.31</td>
<td>0.2075</td>
<td>1.2617</td>
<td>1.0%</td>
</tr>
<tr>
<td>64:512</td>
<td>2.76</td>
<td>3.23</td>
<td>3.00</td>
<td>1.79</td>
<td>0.0492</td>
<td>0.8625</td>
<td>1.0%</td>
</tr>
<tr>
<td></td>
<td>2.98</td>
<td>3.54</td>
<td>3.00</td>
<td>1.80</td>
<td>0.0871</td>
<td>1.0046</td>
<td>1.0%</td>
</tr>
<tr>
<td></td>
<td>3.12</td>
<td>3.85</td>
<td>3.00</td>
<td>1.82</td>
<td>0.1797</td>
<td>1.1304</td>
<td>1.0%</td>
</tr>
</tbody>
</table>

*Note.* The Network Configuration Identifier gives the number of switches followed by the number of links.

More heavily loaded the intercluster links are. In what follows, the topologies that represent the average case for each network configuration are evaluated (shown in bold).

In order to select topologies that represent the average case for their respective network configurations, hundreds of connected topologies of each network configuration listed in Table 1 were randomly generated. All topologies were then analyzed to determine their values for each of the six attributes. Only those network topologies that met the narrow criterion that each of its attributes fall within ±5% of its respective mean value were selected to represent the respective "normal" cases of each network configuration.

Shown in Table 1 are the average as well as minimum and maximum values of the attributes for the randomly generated topologies of the network configurations evaluated. This table also indicates the percentage of topologies within each network configuration that met the selection criterion mentioned above. The evaluation of these topologies is intended to demonstrate the average deadlock and message blocking behavior of each network configuration. This approach not only allows one to observe message blocking and deadlock for the particular class of networks examined in this study, but also provides a basis for studying other classes of topologies by varying one or more of the normalized attributes [17].
3.1.2. Simulation Parameters

The default network configuration assumed consists of 64 switches, 128 physical channels, a minimum switch degree of 1, and a maximum switch degree of 8 (not counting the locally connected processors). For all network configurations used, each switch is connected locally to four processors (bristling factor of 4) using independent physical injection/delivery channels. Only one virtual channel per physical channel is assumed unless mentioned otherwise. Networks of other configurations containing different numbers of physical channels, and/or switch connectivity are used to evaluate the effects of various parameters on deadlock. For example, in evaluating the effect of physical channels on deadlock behavior, networks with 64 switches and 64, 128, 256, 384, and 512 physical channels are assumed. The minimum switch degree is always set to one physical channel. The maximum switch degree allowed for switch connectivity is always set to twice the average switch degree. This assumption on switch degree differs from previous evaluation approaches [8, 9] that require all switches to have uniform switch degree, thereby eliminating much of the irregularity of randomly generated topologies.

All simulations are run for normalized loads up to full network bisection capacity or until the network saturates with respect to the number of resource dependency cycles, whichever occurs first. In most cases, this saturation load exceeds the loads at which network performance (latency and throughput) saturates. The performance saturation points are denoted in the figures by vertical lines. The saturation loads represent the worst case scenarios for the network as attempts to inject additional traffic into the network does not yield additional throughput or other performance gains. However, in many cases, deadlocks do not begin to form until the network is deeply saturated. Therefore, simulations are performed using loads well into deep saturation in order to properly observe and characterize deadlock behavior. Each simulation is run for a duration of 200,000 simulation cycles beyond steady state. For those experiments in which no deadlock initially formed, simulations are further extended for an additional duration of 200,000 cycles to provide greater opportunity for deadlock to occur.

During simulation, the deadlock detection algorithm is invoked every 100 cycles. If deadlock is detected, it is immediately “broken” by removing a single randomly selected message in the deadlock set from the network in a flit-by-flit manner so as to emulate a recovery procedure similar to that described in [18]. Deadlock frequency is measured in relative terms by the normalized number of deadlocks—which is the ratio of the number of deadlocks to the number of messages delivered. Deadlock complexity is measured in terms of the size of the deadlock sets, resource sets, and knot cycle density. Measurements such as the total number of resource dependency cycles formed and the number and percentage of blocked messages are also used to measure message blocking as well as to estimate the likelihood of deadlock.

Other default parameters used for simulation include 32 flit messages, 8 flit buffers, wormhole switching, atomic buffer allocation, and a random channel selection policy. Uniform traffic patterns are used for most simulations. Additionally,
nonuniform “hot-spot” traffic that emulates client–server behavior found in LAN-type settings is also used. For adaptive deadlock recovery routing, a minimal true fully adaptive routing (TFAR) algorithm that allows unrestricted use of any available virtual channel along any profitable path is used [19]. For nonadaptive deadlock recovery routing, an algorithm that allows the use of any virtual channel along a single minimal path between any node pair is used (Static). This singular path allowed between a given node pair is predetermined and does not change during simulation. Given that no other routing restrictions are enforced, deadlocks are possible in both Static and TFAR.

3.2. Impact of Routing Freedom

The impact that routing freedom has on deadlock formation is evaluated by varying the amount of physical channels, routing adaptivity, and virtual channels in the simulated networks.

3.2.1. Effect of Arbitrarily Adding Physical Channels and Routing Nonadaptively

The effect on deadlock probability of arbitrarily adding physical channels to an irregular network that uses static (nonadaptive) routing is evaluated by varying the number of bidirectional physical channels randomly added, following the minimum/maximum node degree constraints discussed in the previous section. All other network parameters are set to default values. Fig. 6 plots the normalized number of deadlocks versus load rate for the 64:64, 64:128, 64:256, 64:384, and 64:512 networks. Both the absolute number of deadlocks and normalized deadlocks initially increase as the number of physical channels increase to a point, then, subsequently, decrease as the number of physical channels is further increased.

As static routing supplies each message with only a single routing option at each intermediate node, each resource dependency cycle formed constitutes a deadlock. Formation of such single cycle deadlocks requires that there be cycles within the physical topology along which these deadlocks could form. A connected network topology with \( n \) nodes and \( n \) links such as the 64:64 network can contain, at most, a single topological cycle—all deadlocks must form in one of the two directions along this topological cycle. Therefore, the number of deadlocks that can form within the 64:64 network is constrained by the availability of topological cycles along which resource dependency cycles can form. The relatively large number of deadlocks that occur in the 64:64 network despite this limitation suggests that the resolution of most deadlocks is quickly followed by the formation of another deadlock along the same topological cycle.

These results indicate that increasing the number of physical channels arbitrarily placed yields only a moderate reduction in message blocking and deadlocks when using static routing, even when substantial numbers of channels are added. This is due to static routing’s inability to exploit the increased channel resources and the random placement of additional channels. Although the total number of physical channels are increased, the number and severity of “blocking points” in
the network are not necessarily decreased. Thus, routing freedom largely remains unaffected.

3.2.2. Effect of Arbitrarily Adding Physical Channels and Routing Adaptively

To examine the effects of better utilizing available physical channel resources, minimal TFAR is used in irregular networks with varying numbers of additional arbitrarily placed bidirectional channels. All other network parameters are set to default values. Figure 7 plots the number of normalized deadlocks versus load rate for
FIG. 7. (a) Number of deadlocks and (b) normalized deadlocks versus load rate for 64:64, 64:128, 64:256, and 64:384 network configurations using TFAR (no virtual channels).

the 64:64, 64:128, 64:256, and 64:384 networks. Very few cycles or deadlocks occurred for the 64:512 network, so this network is not included in the figures. These figures indicate that deadlocks increase as the number of physical channels is increased initially, but reduce substantially as the number of channels is further increased.

Unlike static routing, which always supplies only a single routing option, TFAR allows routing over all minimal paths between all nodes and is, therefore, able to better utilize the increased number of physical channel resources. However, the
results indicate that the increased routing freedom gained by arbitrarily adding physical channel resources from 64 to 128 is insufficient to overcome the increased opportunities for deadlocks posed by the greater number of topological cycles. Nevertheless, as routing freedom is further increased by arbitrarily adding physical channels beyond 128, the resulting decrease in message blocking helps to reduce deadlock frequency.

Despite the higher average number of routing options made available at each node as physical channels are increased, a relatively high number of deadlocks (mostly single-cycle) appear in all but the 64:512 network. The predominance of single-cycle deadlocks in all network configurations indicate that most deadlocks form when messages exhaust adaptivity and/or reach the boundaries of “clusters” where only a single physical channel option remains along the chosen minimal path. Results indicate that for all of the networks evaluated, 92 to 99% of the messages block waiting for only a single channel option at saturation.

Finally, the results indicate that a large number of additional, randomly connected physical channels (i.e., a channel-to-switch ratio of 8:1) is needed to significantly reduce deadlock frequency. Although adaptive routing increases routing freedom to some extent, the number and severity of “blocking points” in the network are not significantly decreased by the addition of randomly placed channels. Hence, the main deadlock-reducing benefit comes from adaptive routing.

### 3.2.3. Effect of Uniformly Adding Channel Resources and Routing Adaptively

The effects on deadlock probability of uniformly adding channel resources are evaluated by considering routing that allows unrestricted use of 1, 2, 3, and 4 virtual channels along nonadaptive predetermined paths (Static1, Static2, Static3, and Static4) as well as along minimal TFAR paths (TFAR1, TFAR2, TFAR3, and TFAR4). All other network parameters are set to default values.

Figure 8a plots the absolute number of deadlocks (not normalized) and cycles versus load rate for Static1, Static2, Static3, and Static4. As no deadlocks occurred for Static routing with 3 or 4 virtual channels, only cycles are shown for these networks. Static1 is inherently limited to only single-cycle deadlocks when deadlocks form; therefore a single curve represents both cycles and deadlocks for this network. For each node pair, static routing allows the use of only those virtual channels that belong to physical channels along a single predetermined path. The increase in the degree of message correlation required for deadlock as caused by increasing routing freedom using two virtual channels eliminates most deadlocks. Furthermore, with an increase in routing freedom using 3 and 4 virtual channels, no deadlocks formed during the entire simulation period (400,000 cycles).

Figure 8b plots the average deadlock set and resource set sizes for the deadlocks formed in Static1 and Static2. The average deadlock set size for Static2 is roughly three to four times larger than for Static1. Correspondingly, the average resource set size for Static2 is also three to four times larger than for Static1. Static1 leads to single-cycle deadlocks only while Static2 leads to multicycle deadlocks with knot densities as large as 6000 cycles. These deadlock characteristics indicate that when routing freedom is increased, the fewer deadlocks that form are much larger and
FIG. 8. (a) Number of deadlocks and cycles versus load rate and (b) average deadlock and resource set sizes versus load rate for Static routing with 1, 2, 3, and 4 VCs per physical channel in 64:128 networks.

more complex due to the higher degree of correlation among messages required for deadlock to form.

In contrast to Static, minimal TFAR can use all virtual channels along all profitable paths that lead to a destination, thus yielding much greater routing freedom. Figure 9a plots the percent of messages blocked versus load rate for TFAR1, TFAR2, TFAR3, and TFAR4. These figures also include information regarding Static1, Static2, Static3, and Static4 for reference. In this figure, two curves are shown for
both TFAR1 and Static2; the upper curve shows the number of cycles and the lower shows the number of deadlocks. For Static1, the number of cycles is equal to the number of deadlocks, so only one curve is shown. As no deadlocks occurred for TEAR routing with 2, 3, and 4 virtual channels, only cycle information is shown for these networks.

As with regular networks [15], combining adaptive routing with multiple virtual channels has a drastic effect on reducing deadlocks. Results also indicate that
adaptive routing, when used with only a single virtual channel, does not provide enough routing freedom to offset the opposing effects of blocked messages and resource dependency cycles. However, when using two virtual channels, the increased routing freedom in adaptive routing increases the degree of correlation required for deadlock so as to minimize those deadlocks suffered in static routing with two virtual channels.

Results indicate that TFAR2, TFAR3, and TFAR4 as well as Static2, Static3, and Static4 lead to a large number of cyclic nondeadlocks immediately following saturation. As shown in Fig. 9a, the number of cycles begin to grow rapidly at the point where a large number of messages begin to block. This is shown in Fig. 9b, which plots congestion (percentage of messages in the network that are blocked) for all of the networks evaluated. This indicates that the addition of each virtual channel reduces congestion and allows higher loads to be applied before a large number of cyclic nondeadlocks form. It also indicates the congestion-reducing benefits of using any given number of virtual channels in an unrestricted, true fully adaptive manner. The sharp growth in cycles observed in the figure is due to the combined effects of a large number of messages that block at saturation and the increased routing freedom of virtual channels that allow each message to wait on a larger number of channel resources. In the absence of deadlock for these networks, the large number of cycles indicate the formation of large cyclic nondeadlocks.

In lieu of virtual channels, a similar significant reduction in deadlock frequency can also be achieved by uniformly adding physical channels in a regular redundant fashion. When used unrestrictedly, redundant physical channels provide the same routing freedom as virtual channels while allowing greater performance as no channel multiplexing is required. This approach can straightforwardly be implemented by utilizing unused switch ports and implementing multiple physical channel links between switch pairs. In this case, the size (number of ports) of each switch would have to be sufficiently large so as not to limit the scalability of the network.

3.2.4. Effect of Nonuniform Traffic

In addition to uniform traffic, a nonuniform traffic pattern that emulates client–server interaction found in LAN settings is also used. In this traffic pattern, 10 to 20% of the nodes in the network with the highest node degree were identified as being servers. All other network nodes were assumed to be clients. A disproportionate amount of traffic generated by all nodes were sent to the nearest server. Almost all of the experiments performed using this nonuniform traffic pattern resulted in nearly identical deadlock frequencies and characteristics as those found using uniform traffic. One notable exception was for TFAR with two virtual channels. None of the 64:128 networks with TFAR and 2 VCs led to any deadlock when using uniform traffic. However, a small number of deadlocks (1 for every 750,000 messages delivered) were found for some cases of 64:128 networks under client–server traffic. An examination of these deadlocks reveals that they were caused by a
high rate of messages exchanged between a small number of server nodes that were close to each other in proximity. The use of a third virtual channel resulted in no deadlocks during simulation.

3.3. Impact of Nonatomic Allocation

Trends in implementation technology make the use of large channel buffers feasible in irregular network switches. In many instances, the relatively large physical distances between network nodes in NOW settings require the use of large buffers for efficient flow control. However, large buffers cannot be efficiently utilized under assumptions of atomic buffer allocation. Also, the heterogeneous nature of traffic in NOWs containing messages of many different sizes make the efficient use of large buffers even more critical. Allowing messages to share channel buffers (nonatomic buffer allocation) can greatly increase the utilization of channel resources. However, it can also reduce the routing freedom of messages that block before arriving at the heads of shared queues.

To examine the effects of nonatomic buffer allocation on deadlock behavior, minimal TFAR is used with two virtual channels and a fixed message length of 16 flits in 64:128 networks with channel buffer depths of 16, 32, 48, and 64 flits (NAT 16:16, NAT 16:32, NAT 16:48, and NAT 16:64). All other network parameters are set to default values. A network that uses atomic buffer allocation with 16 flit messages and 16 flit channel buffers (AT 16:16) is also evaluated for comparison. (Note that larger buffer configurations for the atomic buffer allocation network (i.e., AT 16:64) have the same behavior as only one message is allowed in each VC queue at any one time.)

No deadlocks appeared for any of the irregular networks for loads well into saturation. Thus, Fig. 10 shows message blocking behavior: the number of messages in the network (upper curve)—which, at saturation, nearly all become blocked—and the number of messages blocked prior to reaching the head of a VC queue (lower curve). The NAT 16:16 network is able to support nearly 200% more messages than the AT 16:16 network at their respective saturation points and approximately 20% more messages in deep saturation. As compared to atomic buffer allocation, nonatomic buffer allocation improves the load at which the network saturates by about 66%. Results also indicate that although the use of larger buffers (as opposed to smaller buffers) with nonatomic allocation allows more messages to be in the network, it does not significantly improve the throughput saturation load of the network, which is approximately 0.125 for all buffer sizes.

The percentage of messages that block at the heads of queues for the nonatomic networks decreases as the buffer size increases, but a significant percentage (25-40% in deep saturation) remains regardless of buffer size. These messages wait for multiple channels resources when blocked, thus preserving routing freedom. The routing freedom of these messages serves to decrease the probability of deadlock due to the high degree of correlation required for deadlock to form. The absence of deadlock, especially of the single-cycle variety, indicates that each resource dependency cycle contains at least one message that has reached the head of its queue, as formally proved in [11].
FIG. 10. (a) Messages in the network (upper curve) and messages blocked before reaching the head of a queue (lower curve) vs load rate for 64:128 networks with 2 VCs and buffer depths of 16, 32, 48, and 64 flits using wormhole routing with nonatomic buffer allocation. Show, in (b) is the number of cycles that form. TFAR with uniform traffic and a message length of 16 flits is assumed.

The effects of nonatomic buffer allocation for virtual cut-through routing is also examined (but not plotted) using 64:128 networks with a fixed message length of 16 flits, channel buffer depths of 32 and 64 flits, and two virtual channels. As with the nonatomic wormhole networks, no deadlocks occurred for the virtual cut-through networks with two virtual channels. The overall message blocking behavior is seen to be similar to that of the nonatomic wormhole networks.
Apparently, nonatomic buffer allocation in wormhole switching offers similar intermessage compaction within buffers as does virtual cut-through switching while not significantly affecting the degree of correlation in message blocking required for deadlock, even though messages commit earlier to virtual channels as compared to atomic buffer allocation. As mentioned previously, at least one message in each resource dependency cycle reaches the head of its queue, thus preserving the freedom to adaptively choose among multiple channel resources.

4. DISCUSSION

The uneven distribution of channels in irregular networks limits the routing freedom of messages at various points in the network. Blocking points make congestion and deadlock more likely in irregular networks than in regular networks as less correlated resource dependencies are needed for deadlock to form. Arbitrarily adding channel resources only moderately alleviates congestion and reduces deadlock frequency. For more effective results, channel resources should be added at points in the network where the irregular topology creates blocking points. This serves to increase routing freedom at the points in the network where it is lowest, thus increasing the degree of correlation in resource dependency needed for deadlock to form.

The use of virtual channels is a step in this direction. With this technique, channel resources are increased uniformly across the network, including at the blocking points of a network. Those messages that block waiting for a single physical channel benefit from having multiple (virtual) channel resources from which to choose, assuming unrestricted routing within physical channels. The benefits of this technique in increasing the resource correlation required for deadlock to form is amplified when unrestricted routing is allowed across physical channels. Results indicate that uniformly increasing the number of existing channel resources between switches by a factor of 2 is much more effective at reducing deadlock frequency than increasing the number of channel resources in an arbitrary fashion between switches by a factor of 4. Thus, it can be concluded that how channels (physical or virtual) are added to reduce deadlock frequency is more important than how many channels are added.

Furthermore, improving resource utilization by allowing the sharing of channel buffers for both wormhole and virtual cut-through switching yields some benefits in reducing congestion, as seen by the higher saturation load rate. However, this technique does not increase routing freedom, which is key to reducing the probability of deadlock. Therefore, nonatomic buffer allocation should be used in conjunction with other techniques, such as virtual or redundant physical channels and adaptive routing, to gain maximum benefit.

5. RELATED WORK

The work in [20] models the probability of deadlock in operating systems. However, no such work has been performed for deadlock in interconnection
networks. Original studies that measured deadlock frequency in interconnection networks did so using crude approximations based on network congestion and message blocking via time-outs as the primary means for detecting deadlocks [19, 21, 22]. Subsequent work [14, 15] has improved upon this by not only precisely measuring actual deadlock, but by relating key network parameters to their influence on deadlock for regular interconnection networks. That work is extended here to characterize deadlocks for irregular networks.

Previous work based on the use of a relatively small sample of randomly generated topologies implicitly assumes that all irregular networks of a particular size behave similarly regardless of their interconnection patterns [8, 9]. In addition, these studies limit their investigation to only those irregular topologies that have uniform switch degree, thus restricting the applicability of their results. In contrast, this work quantitatively classifies topologies so as to relate results to the crucial properties of the interconnection patterns of a topology. Also, this study includes topologies that have nonuniform switch degree, an important characteristic that contributes significantly to the irregularity of a topology.

6. CONCLUSIONS AND FUTURE WORK

This work characterizes the effects of various network parameters on the frequency of blocked messages, resource dependency cycles, and deadlock. The interrelationships between routing freedom, blocked messages, correlated resource dependencies, and deadlock formation have been empirically quantified for a class of irregular interconnection networks.

It has been shown that as routing freedom is uniformly increased, i.e., through the unrestricted use of multiple virtual channels, the complexity in the correlation of resource dependencies required for deadlock substantially increases so as to substantially reduce the occurrence of deadlock. However, nonuniform increase in routing freedom, i.e., through the addition of physical channels in arbitrary locations in the network, may actually accentuate "blocking points" in the network instead of relieving them, thus having little-to-no effect on increasing the complexity in correlation of resource dependency needed to abate deadlock formation. It is safe to conclude that correlated message blocking and deadlocks in irregular networks can be highly improbable when sufficient routing freedom is properly provided by the network and fully exploited by the routing algorithm. Unrestricted routing on irregular networks with two or more virtual channels or, alternatively, redundant (uniformly added) physical channels is seen to significantly minimize the occurrence of deadlock.

Research that builds on the results of this study can be explored in future work. This includes empirical analysis using hybrid bursty traffic loads generated by real applications, evaluation of other classes of irregular topologies, development of an improved method of classifying irregular topologies, and investigation of efficient techniques for injection limitation, deadlock detection, and resolution required for practical implementation of deadlock recovery routing.
APPENDIX

The model of resource allocations and dependencies described briefly in Section 2 has been implemented in a flit-level irregular interconnection network simulator, *IRFlexSim* (an extension of *FlexSim 1.2*). The simulator implements an algorithm that detects each and every instance of true deadlock and does not detect false deadlocks. The algorithm involves maintaining a CWG, detecting cycles within the CWG, and identifying groups of messages and cycles that form knots. This algorithm is intended for use in a simulation environment for evaluation purposes and is not intended (nor suitable) for implementation in real networks as it requires global information.

Building and Maintaining a Channel Wait-for Graph

During run-time of an on-going network simulation, a CWG that reflects the resource allocations and requests dynamically occurring can be built and maintained. The CWG can be implemented as an array of linked lists. Each array element represents a message \((m_i)\) within the network, and each linked list represents the set of channels (physical or virtual) owned by a particular message \((\text{owns}(m_i))\), with additional links to desired channels for blocked messages \((\text{requests}(m_i))\). These channel resources are represented as “VCnodes” (or virtual channel nodes) in the list. Fig. 11 provides an overview of the data structures used to maintain the CWG, cycle list, and deadlock list.

When a channel is allocated to a message, the VCNode data structure for the channel is appended to the linked list for that message and is annotated with its new owner. Similarly, when a channel is released, the VCNode is removed from the beginning of the linked list and ownership information is removed accordingly. When a message blocks, special request links are placed from the end of the linked list to the VCNodes representing each of the alternate channels the message may use to continue routing. When a message resumes, these request links are removed and replaced with a link to the VCNode representing the newly acquired channel.

![Diagram of Channel Wait-for Graph, Cycle List, and Deadlock List](image)

**FIG. 11.** The CWG, cycle list, and deadlock list data structures. The state of the data structures reflects the example in Fig. 1.
The direct links from the elements of the array of messages to the last VCNodes of each linked list are used as an optimization for efficient traversal of the linked lists.

Detecting Cycles

Cycles in the CWG can be detected by performing a depth-first search of the graph with backtracking (starting with each linked list in linear order). Each node visited is marked so that the second visitation indicates the presence of a cycle. Annotations indicating visitation are removed upon backtracking. To avoid detection of each unique cycle more than once, the partial order of message IDs can be used to prune the search (i.e., restrict visiting a VCNode owned by a message with a lower message ID). For example, the VCNodes in the CWG in Fig. 11 are visited in the following order where “f” indicates cycle detection, b indicates a backtracking step, and p indicates a pruning measure: [m₁], vc₀, vc₁, vc₃, vc₅, vc₇, vc₁, f, b, b, b, b, b, b, [m₂], vc₂, vc₃, vc₅, vc₇, p, b, b, b, b, [m₃], vc₄, vc₅, vc₇, p, b, b, b, b, b, b, [m₄], vc₆, vc₇, p, b, b, b, b, b, [m₅], vc₈, vc₉, vc₁₀.

When a unique cycle is detected, it is placed in the cycle list as shown in Fig. 11. A list of messages involved and their corresponding “branches” are sufficient to uniquely identify a cycle (i.e., m₁:1 in the figure indicates the first branch of message m₁, representing the first of possibly many channels the message is waiting for). When previously detected cycles are again detected during a different run of the algorithm, their entries in the cycle list can be updated to indicate the time of their most recent detection. The cycle detection algorithm guarantees detection of each unique cycle in the CWG during each invocation. Therefore, those cycles that were not detected during the most recent invocation of the algorithm are removed from the cycle list. Since a very large number of cycles can exist within a congested network, additional pruning using advanced indexing functions and search techniques to compare cycles can be done in determining their uniqueness.

Detecting Deadlocks

Once the cycle detection algorithm identifies all cycles within the CWG, a deadlock detection algorithm can be invoked to identify groups of these cycles that form a knot. A number of heuristics can be used to make this possible such as the following. First, a group of blocked messages whose request arcs are all involved in cycles is identified using information provided by the cycle detection algorithm (e.g., messages m₁, m₂, m₃, and m₄ in Fig. 11). Each message in this group is examined to see if all of the channels they are waiting for are owned by messages also in this group. The subset of messages that meet this condition is further distinguished based on whether the messages own at least one channel requested by a message also in the group. Note that these conditions hold for messages m₁, m₂, m₃, and m₄ in Fig. 11. Cycles are then examined to identify a group of them that includes only messages from this group. A group of cycles that meets this condition forms a knot and is therefore identified as a deadlock and placed in the deadlock list, as shown in Fig. 11. Detected deadlocks can be “broken” by removing a single
message in the deadlock set in a flit-by-flit fashion so as to simulate a progressive recovery procedure.

**Time and Space Complexity**

The operations to build and maintain the CWG are invoked automatically by our simulator on an as-needed basis, and each of the operations requires constant time. The deadlock and cycle detection times are invoked based on user-specified time intervals. For a simulation run length of $T_{sim}$ simulation cycles (sim-cycles), the cycle detection algorithm is invoked once every $T_{cd}$ sim-cycles, $T_{cd} \leq T_{sim}$, and the deadlock detection algorithm is invoked once every $(T_{cd} \times n)$ sim-cycles, $n \geq 1$.

The computational complexity of the cycle detection algorithm can be expressed as $O((R_f/2)^M + C^3_a)$ where $R_f$ corresponds to the routing freedom, $M$ corresponds to the average hop distance, and $C_a$ corresponds to the number of active cycles in the network. The term $(R_f/2)^M$ corresponds to the time required for full traversal of the CWG while the term $C^3_a$ corresponds to the time required for “processing” (determine uniqueness, record characteristics, etc.) of each cycle found within the CWG.

Currently, we are able to detect all cycles formed in networks that provide a moderate degree of routing freedom (i.e., true fully adaptive routing [22] in 2D and 3D networks with as many as four virtual channels per physical channel) for up to and slightly beyond saturation loads. As an example, the execution time for a single invocation of the cycle detection algorithm on an IBM SP/2 node for a saturated 16-ary 2-cube network with four virtual channels can take up to 3 h. Given the theoretical worst case for $C$ and $C_a$ of $(R_f)^M$, where $M$ is the number of blocked messages within the network, large simulations can become impractical as a very large number of messages with high fanout may block within some networks at loads beyond network saturation.

The computational complexity of the deadlock detection algorithm can be expressed as $O(D \times C \times M_c)$, where $D$ is the average number of deadlocks found during an invocation of the deadlock detection algorithm, $C$ is the average number of cycles found during an invocation of the cycle detection algorithm, and $M_c$ is the average number of messages involved in a cycle. Thus, our simulation capabilities are not limited by the complexity of this algorithm. In comparison to the cycle detection algorithm, the execution times for invocations of the deadlock detection algorithm are negligible. However, this algorithm has been greatly optimized by using heuristics based on information provided by the cycle detection algorithm, and since it relies on all cycles being detected, it is of limited use for detecting deadlock in deep saturation for some networks. We are currently implementing an alternative deadlock detection scheme that does not require that all cycles be identified and that therefore, can be used to detect deadlock behavior during network conditions beyond saturation.

The space complexity for the algorithm is also dictated by the number of cycles that can form. For example, a network containing one million unique cycles within its CWG requires approximately 75 MB of virtual memory.
REFERENCES


