Deadlock Prevention by Turn Prohibition in Interconnection Networks

Lev Levitin
Dept. of Computer Engineering
Boston University
Boston, MA 02215, USA
Email: levitin@bu.edu

Mark Karpovsky
Dept. Computer Engineering
Boston University
Boston, MA 02215, USA
Email: markkar@bu.edu

Mehmet Mustafa
Dept. of Computer Engineering
Boston University
Boston, MA 02215, USA
Email: mmustafa@bu.edu

***Abstract*—In this paper we consider the problem of constructing minimal cycle-breaking sets of turns for graphs that model communication networks, as a method to prevent deadlocks in the networks. We present a new cycle-breaking algorithm called Simple Cycle-Breaking or SCB algorithm that is considerably simpler than earlier algorithms. The SCB algorithm guarantees that the fraction of prohibited turns does not exceed 1/3. Experimental simulation results for the SCB algorithm are shown.**

I. Introduction

Due to the availability of low cost workstations with network interface adapters that offer high-performance communications using wormhole techniques, clusters of workstations are emerging as preferred computing environments [1], [2], [3], [4]. However, as wormhole routed messages hold network resources while requesting others, as they traverse the network towards the destination, wormhole routing is prone to deadlocks under heavy network loads [5], [6], [7], [1], [3]. Deadlocks have been shown to occur due to the presence of "cycles of edges" in the graph representing the network [8]. Similar to spanning tree approaches [9], prohibiting a carefully selected set of the turns in the network, provides deadlock freedom [8], [10], [1], [11], [3], [12], [13]. However, unlike the spanning tree based approaches, the cycle-breaking approach allows all communication links in the network to be used. The only restriction is that some pairs of communication links, namely, those that form the prohibited turns, are prevented from being used sequentially. The motivation for seeking the minimal fraction of prohibited turns is due to Glass and Ni [11]. They have found that reduction in the number of prohibited turns results in a decrease of average path length and the average message delivery time, thereby increasing the throughput. After Glass and Ni showed it for regular topologies such as meshes and tori, this conclusion was confirmed by other authors [4], [3] for irregular topologies as well. Experimental data [12] show that there is a gain of approximately 7% - 8% in the maximum sustainable throughput in the network, for each percentage point reduction in the fraction of prohibited turns.

Let us consider an undirected regular graph $G(V, E)$, with $N = |V|$ vertices, denoted by $a, b, \ldots$, and $M = |E|$ edges, denoted by (a, b), etc. A **turn** in G is a triplet of nodes (a, b, c) if (a, b) and (b, c) are edges in G and $a \neq c$. In an undirected graph, turns (a, b, c) and (c, b, a) are considered to be the same turn. This undirected graph is a model of an interconnection network where nodes are computing elements and routers, and undirected edges correspond to full duplex communication channels between computing elements. If the degree (number of neighbors) of node j is d_j, the total number of turns $T(G)$ in G is given by $T(G) = \sum_{j=1}^{N} \binom{d_j}{2}$. A **path** $P = (v_0, v_1, \ldots, v_{L-1}, v_L)$ of length L, $L \geq 1$, from node a to node b in G is a sequence of nodes $v_i \in V$ such that, $v_0 = a$, $v_L = b$ and every two consecutive nodes are connected by an edge. Subsequences of the form $\ldots, v_i, v_k, v_i, \ldots$ are not permitted in a path. Nodes and edges in the path are not necessarily all different. Path $P = (v_0, v_1, v_2, \ldots, v_k, v_0, v_1)$ in G is called a **cycle**. Set $F(G)$ of **prohibited turns** in G is called **cycle-breaking** and **connectivity preserving** if every cycle in G includes at least one turn from $F(G)$, and G remains connected. If a turn is prohibited, the paths that include this turn cannot be used for routing. The minimum cardinality set $F(G)$ for a given graph G is denoted by $Z(G)$ and the fraction of prohibited turns is denoted by $z(G) = Z(G)/T(G)$.

The following lower bounds were proven in [12] and are presented here for completeness. Given a connected graph G with minimum node degree δ, the total number R of cycles in G, and the maximum number r of cycles covered by the same turn, the number of prohibited turns $Z(G)$ and fraction of prohibited turns $z(G)$ satisfy the following inequalities:

$$Z(G) \geq M - N + 1, \tag{1}$$

$$z(G) \geq \frac{R}{rT(G)}, \tag{2}$$

and

$$Z(G) \geq M - N + \binom{\delta - 1}{2} + 1, \ \delta > 2. \tag{3}$$

In this paper, we present a new turn prohibition algorithm, the Simple Cycle Breaking (SCB) algorithm and prove its properties and compare it with other deadlock preventing algorithms in Section II. Distance dilation caused by SCB is considered in Section III. Our experimental results are presented in Section IV, and conclusions in Section V.

II. General Algorithm for Construction of Minimal Cycle-Breaking Sets of Turns

Denote by $G\backslash H$ the graph obtained by deletion from graph G all nodes of subgraph H. A node $a \in V$ in a graph G is a **cut-node** if its deletion disconnects G.

Lemma 1: If a connected graph G has cut nodes, then there exists a connected subgraph H which consists of non-cut nodes only of the original graph G and is connected to the rest of G via only one cut node $c \in G\backslash H$ (i.e., if $a \in H$, $b \in G\backslash H$, and $P(a,b)$ is a path from node a to node b, then $c \in P(a,b)$).

Note that any path from H to $G\backslash H$ includes this cut node c. Of course, it can, in general, include more cut nodes.

Proof: Suppose G has cut nodes. Let S_i be the set of connected components of G obtained by deleting cut node $c_i (i = 1, 2, \dots)$ from G. Consider the union $\cup_{i=1} S_i$. Let $H \in \cup_{i=1} S_i$ be the connected component with the smallest number of nodes. This component does not include any cut nodes from the original graph (otherwise it would not be the smallest component). Thus, if H is obtained by deleting cut node c from graph G, then H is a connected subgraph which is connected to $G\backslash H$ via one cut node c only. ■

Lemma 2: In any connected graph G, there exists a non-cut node a of degree d, such that

$$2\binom{d}{2} \leq \sum_{i=1}^{d}(d_i - 1), \tag{4}$$

where d_i $(i = 1, 2, \dots, d$ are the degrees of the neighbors of a (nodes adjacent to a).

Proof: Using Lemma 1, consider a subgraph that consists of non-cut nodes and at most one cut node, connecting this subgraph to the remaining part of the graph. Select a non-cut node a of the minimum degree d among all non-cut nodes in this subgraph. If a is not adjacent to the cut node, then inequality (4) is obviously satisfied. Suppose now that all nodes with minimum degree d are adjacent to the cut node with degree $d' < d$. Then the selected node a has at most $d'-1$ neighbors of degree d, while at least $(d-1)-(d'-1) = d-d'$ of its neighbors have degrees at least $d+1$. Thus,

$$\begin{aligned} \zeta(a) \quad &\triangleq \quad \sum_{i=1}^{d}(d_i - 1) \\ &\geq \quad (d'-1)(d-1) + (d-d')d + (d'-1) \\ &\geq \quad d(d-1) = 2\binom{d}{2}. \end{aligned}$$

■

We now describe the SCB Algorithm. Given a connected graph $G(V, E)$, the SCB algorithm creates two sets: the set $F(G)$ of prohibited turns and the set $A(G)$ of permitted turns. It also labels all nodes by natural numbers starting with 1, in the order they are selected by the algorithm. In the beginning, $F(G) = \emptyset$, $A(G) = \emptyset$, and all nodes are unlabeled. If $|V| = N$, the algorithm consists of $N-1$ stages (recursive calls). Each stage consists of 3 steps described below.

1) If $|V| = 2$, label the nodes by the smallest unused natural numbers, select and delete the node with label $\ell = N-1$ and return sets $F(G)$ and $A(G)$. Otherwise, go to step 2
2) Select a non-cut node a of the minimum degree d, such that
$$2\binom{d}{2} \leq \sum_{i=1}^{d}(d_i - 1), \tag{5}$$
where d_i are the degrees of the neighbors of a (nodes adjacent to a). The existence of such a node is guaranteed by Lemma 2. Prohibit all turns at he selected node a of the form (b, a, c) (nodes b and c are neighbors of a) and include them in $F(G)$. Permit all turns of the form (a, b, d) (node d is neighbor of node b) and include them in $A(G)$. Label a by the smallest unused natural number $\ell(a)$.
3) Delete node a to obtain graph $G' = G \setminus a$ and go to step 1 for G'.

Note that at the stage of the algorithm when node a is selected, all other undeleted nodes are unlabeled. In fact, they will be labeled later. As a result, turn (b, a, c) is prohibited *iff* $\ell(a) < \ell(b)$ and $\ell(a) < \ell(c)$.

In Fig. 1 we show a simple graph demonstrating the operation of the SCB algorithm. Graph has two degree 3 cut-nodes c and e, and one degree 2 cut-node d.

The original graph is shown in Fig. 1(a). Since there are 7 nodes the completion of the algorithm would involve 6 stages. Before the algorithm begins to execute, the two sets $F(G)$ and $A(G)$ are initialized to be empty and the *label* is initialized to be 1. At the first stage, step 1 determines that the number of remaining nodes in the graph is not equal to 2 and immediately transitions to step 2. At this step, a minimum degree non-cut node is selected. Since node d is a cut node, it cannot be selected. The minimum degree non cut nodes are nodes a, b, f, and g. The criterion in (4) is applied and all of the candidate nodes satisfy the inequality. For example, for node a, the left hand side of (4) evaluate to 2 and right hand side of (4) evaluate to 3, hence, node a is selected. As shown in Fig. 1(b), one turn is prohibited, denoted by the arc, i.e., (b, a, c), which is added to the set $F(G)$. Permitted turns at this step are (a, b, c),(a, c, b), and (a, c, d) which are added to the set of permitted turns, $A(G)$. The node is assigned the label 1, and transition is made to step 3. In Step 3, the selected node a is deleted to obtain the subgraph with 6 nodes as shown in Fig. 1(c). The SCB algorithm begins executing the stage 2. In this stage, degree 1 node b is selected, no turns are prohibited, but one turn (b, c, d) is permitted, node b is labeled with 2 as shown in Fig. 1(d). After deleting the node in step 3 we obtain the 5-node subgraph shown in Fig. 1(e), and stage 3 of the algorithm begins. Since node c is a degree 1 node, it is selected, no turns are prohibited, one turn, (c, d, e), is permitted, node c is labeled 3 as shown in in Fig. 1(f) and when node c is deleted in step 3 we arrive at the 4 node sub graph of in Fig. 1(g). In stage 4, node d is selected and labeled as shown. In stage 5, all of the remaining nodes e, f, and g satisfy (4). Algorithm arbitrarily selects node e, prohibits turn (f, e, g), permits two turns, namely, (e, f, g) and (e, g, f), labels the node e as 5, and deletes it. Then stage 6 begins executing. In this stage, step 1 of the algorithm labels nodes f and g and the algorithm terminates. In Fig. 1(m) the graph is shown with all of the prohibited turns and the node labels. It is clear that in all prohibited turns (u, v, w) the labels satisfy $\ell(v) < \ell(u)$ and $\ell(v) < \ell(w)$. The stage-by-stage operation of the algorithm is also shown in Table I in which each row corresponds to a stage.

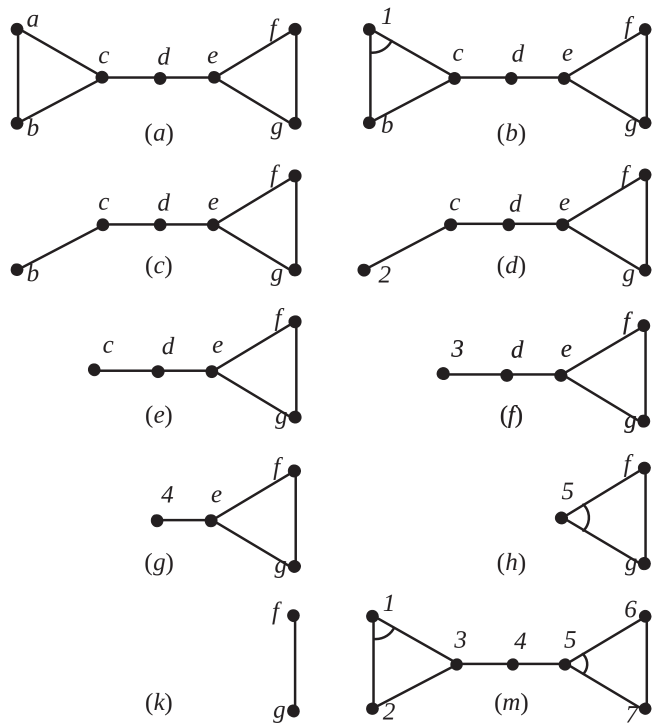

Fig. 1: Operation of the SCB algorithm on a simple graph where prohibited turns are shown as arcs, yielding $z(G) = 2/11$.

TABLE I: Stage-by-Stage Operation of the SCB algorithm. In the last stage, stage 6, nodes f and g are labeled and the algorithm terminates

Selected Node	Node Label	Set of Prohibited Turns	Set of Permitted Turns
a	1	$\{(b, a, c)\}$	$\{(a, b, c),\ (a, c, b),\ (a, c, d)\}$
b	2	$\emptyset$	$\{(b, c, d)\}$
c	3	$\emptyset$	$\{(c, d, e)\}$
d	4	$\emptyset$	$\{(d, e, f), (d, e, g)\}$
e	5	$\{(f, e, g)\}$	$\{(e, f, g),\ (e, g, f)\}$
f	6	$\emptyset$	$\emptyset$
g	7	$\emptyset$	$\emptyset$

Theorem 1: The SCB algorithm has the following four properties.

Property 1. Any cycle in G contains at least one turn from $F(G)$.

Property 2. SCB preserves connectivity; for any two nodes $a, b \in V$, there exists a path between a and b that does not include turns from $F(G)$.

Property 3. The set $F(G)$ of prohibited turns generated by SCB algorithm is *minimal* (*irreducible*).

Property 4. For any graph G, $|F(G)| \leq T(G)/3$, where $T(G)$ is the total number of turns in G.

Proof of Property 1: Consider the node a with the minimum label $\ell(a)$ in any cycle C in G. Then in the turn (b, a, c) $(b, a, c \in C)$, $\ell(a) < \ell(b)$ and $\ell(a) < \ell(c)$. Thus, turn (b, a, c) is prohibited and cycle C is broken. ■

Proof of Property 2: The proof is by induction. Consider the first selected node a, $\ell(a) = 1$. Since a is a non-cut node, after all turns of the form (b, a, c) are prohibited and node a is deleted, there still exists a path from any node x to any node y, where $x, y \in G \backslash a$. Also, since all turns of the form (a, b, c) are permitted, there exists a path from a to any node $x \in G$. Now assume that connectivity is preserved after the first n stages of the algorithm, so that the next selected node a has label $\ell(a) = n + 1$. Node a is a non-cut node in the graph that remains after deletion of the first n selected nodes. Therefore, after prohibition of all turns (b, a, c) there still exists a path between any two unlabeled nodes x and y. Consider now paths from a labeled node u, $\ell(u) \leq \ell(a)$ to another previously labeled node v, $\ell(v) < \ell(a)$, or to an unlabeled node y. If such a path P does not include a turn of the form (b, a, c), where b and c are unlabeled, it remains permitted. Now suppose P includes such a turn (Fig. 2). Then, let x be the first unlabeled node in the path from u to v or from u to y, and t be the last unlabeled node in the path from u to v. Now we can replace the part of P from x to y, or from x to t, respectively, by a path that does not include a (such a path exists, since a is a non-cut node) and obtain a path P^*. Let x' be the node already labeled that immediately precedes x in P and in P^*, and t' be the labeled node that immediately follows t in P and P^* (in the case when such a node exists). Since all turns (x', x, w) and (w, t, t') are permitted, path P^* does not contain prohibited turns, and connectivity is preserved at the $(n = 1)$th stage of the algorithm. Thus, Property 2 is proved by induction. ■

Proof of Property 3: Consider a prohibited turn (b, a, c). Since connectivity is preserved and a is a non-cut node, there exists a permitted path (b, P, c) from b to c that does not include a. Adding edges (a, b) and (c, a) to this path, we obtain a cycle $C = (a, b, P, c, a, b)$. Since turns of the form (a, b, x) and (a, c, y) are permitted, the only prohibited turn in C is (b, a, c). By removing this turn from $F(G)$, we would create a cycle in G and violate the cycle-breaking Property 1. Thus, set $F(G)$ is minimal. ■

Proof of Property 4: At the stage of the algorithm when node a is selected (recursive call $\ell(a)$), all turns (b, a, c)

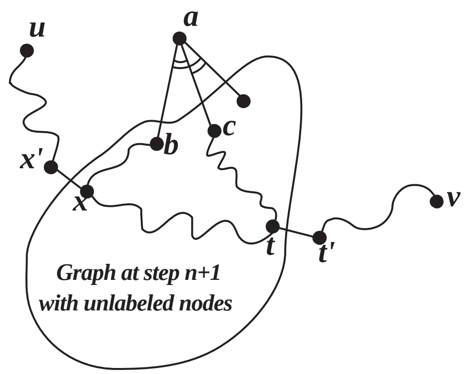

Fig. 2: Figure depicting the state of the graph at step n+1 of the SCB algorithm. Path $P = (u, \ldots, x', x, \ldots, b, a, c, \ldots, t, t', \ldots, v)$ is prohibited due to the prohibited turn at node a. Path $P^* = (u, \ldots, x', x, \ldots, t, t', \ldots, v)$ is permitted since it does not involve any prohibited turns.

become prohibited, and all turns (a, b, c) become permitted, where $\ell(a) < \ell(b)$ and $\ell(a) < \ell(c)$. The number of prohibited turns is $\binom{d}{2}$ where d is the degree of node a; the number of permitted turns is $\sum_{i=1}^{d}(d_i - 1)$, where d_i, $(i = 1, \ldots, d)$ are degrees of all neighbors of a. By Lemma 2, it is always possible to select a non cut-node such that inequality (4) is satisfied. This means that the number of permitted turns is larger than the number of prohibited turns by at least a factor of two. Since this is true for each stage of the algorithm, it follows that $|F(G)| \leq T(G)/3$. ∎

In general, the fraction of prohibited turns yielded by SCB algorithm is considerably smaller than the upper bound of $1/3$. The only class of graphs where the fraction is exactly $1/3$ is the complete graphs K_n with $|V| = n$ and $|E| = n(n-1)/2$.

Let us compare the SCB algorithm with other cycle-breaking algorithms. A widely used algorithm of this sort is the Up/Down algorithm [2]. Its complexity is $O(M)$, which is smaller than that of SCB (the worst-case complexity of SCB is $O(NM)$. This follows from the fact that it takes $O(M)$ time to determine all cut nodes ([14], Ch. 23, Problem 23-2). However, the Up/Down algorithm can turn out to be extremely inefficient. There are examples where the fraction of prohibited turns becomes arbitrarily close to 1 [15].

Another turn-prohibition algorithm, the L-turn algorithm [10], provides an improvement compared with Up/Down algorithm of about 6% in the number of prohibited turns and up to 30% in the throughput. However, the implementation of the L-turn algorithm requires checking a substantial portion of all cycles in the network for a certain condition. Since the number of cycles in a graph can grow super-exponentially with the number of nodes, the practicality of L-turn algorithm for large networks is questionable.

The earlier version of the turn prohibition algorithms (the TP and CB algorithms [16], [17], [12]) have been shown to outperform the Up/Down algorithm in terms of the fraction of prohibited turns (about 5%), average distance between nodes, and saturation load (up to 50%) (see [17], [12]). However the earlier algorithms were more complicated than the SCB algorithm. Indeed, every recursive call in TP and CB algorithms involved as many as ten steps. In particular, at every stage all connected components that appear after a node removal had to be identified, special edges had to be determined, nodes had to be examined in order to be characterized as forcing or delayed, a "halfloop" flag had to be examined and set, etc (for detail, see [12]). In contrast, the SCB algorithm has only three steps which are easier for implementation. The simplification is achieved by elimination of complexities of dealing with cut nodes and is based on theoretical results described in Lemma 1 and Lemma 2. Though the order of the worst-case asymptotic complexity of SCB algorithm remains the same as in previous works ([17], [12]), the practical implementation is substantially simpler.

III. Distance Dilation

In this section we introduce the notion of dilation in a network topology due to turn prohibitions. Paths that involve prohibited turns are themselves prohibited and are not used for communication. Thus, one side effect of turn prohibition is that, prohibiting certain paths from being taken for message routing, may increase distances between some node pairs. The net result of this is that the diameter and the average distance of a network topology will increase. To facilitate the investigation of this phenomenon, we introduced the notion of distance dilation in a graph, which we define as the ratio of the average distance after turn prohibition to the average distance without any turn prohibition. When the dilation is 1 it would imply that the turn prohibition has not caused any lengthening of the average distance. For example in complete graphs the fraction of prohibited turns reaches the upperbound, but the dilation is 1. In Fig. 3 the distance dilation due to SCB in wrap-around topologies (tori) is shown. Dilation is larger in a ring than it is in a 2D torus of same number of nodes. In another set of experiments, we compared the dilation caused by the Up/Down and the SCB algorithms. In these experiments, the topologies that were investigated were bisection width constrained. Bisection width of a graph is the minimum number of edges whose deletion would disconnect the graph into two equal sized components. These were all 64-node topologies with bisection widths of $2, \ldots, 26$. In these experiments the SCB introduced dilations were 5%-10%, and those introduced by the Up/Down algorithm were 10%-20%. Intuitively, we believe that prohibition algorithms that introduce smaller dilations would perform better in message delivery latencies.

IV. Experimental Results

In this section we present the results of our calculations for the fraction of prohibited turns using the SCB algorithm and experiments involving message delivery simulations using OPNET discrete event simulation tools. In all of our calculations and simulations, network topologies were first generated using tools that we developed. All of the topologies were represented by 64-node undirected graphs. Since usually

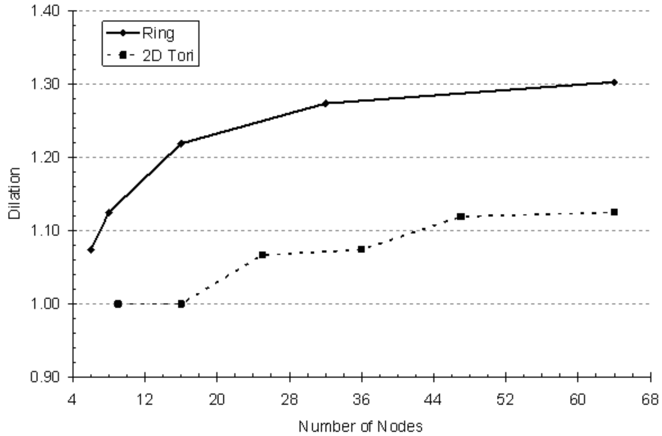

Fig. 3: Dilation in 1-dimensional and 2-dimensional tori due to SCB.

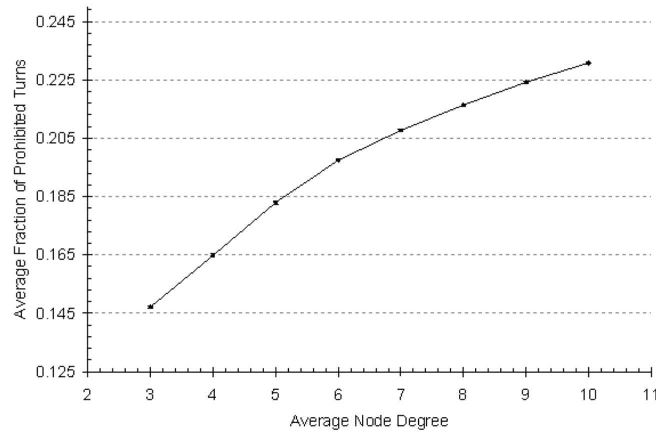

Fig. 4: Average fractions of prohibited turns resulting from the SCB algorithm in graphs with 64 nodes as a function of the average degree.

the number of router ports is small compared to the size of the network, we assumed that the average degree of nodes is small in comparison with the total number of nodes. We constructed families of graphs of average degrees 3 through 10. Each family consisted of 100 different randomly generated graphs. In Fig. 4 we show the results of applications of the SCB algorithm for 8 different families of topologies, each with a different average degree. In Fig. 5, the average distance versus the average node degree is plotted after the application of the SCB algorithm. Given a network topology, as turns are prohibited, the average distance may increase. This increase is described in terms of the dilation introduced by the turn prohibition. Results of dilation calculations are shown in Fig. 6, where we see that as the average degree increases, the average dilation decreases, predicting a better performance for message delivery for topologies with larger degrees.

In bisection-width constrained graphs with 64 nodes and the given bisection width (2-26) we calculated the average fraction of prohibited turns and the dilation. For each family, composed of 100 different graphs, we calculated the average fraction of prohibited turns and tabulated the results as shown in Table II. We see that the Up/Down algorithm not only has larger fractions of prohibited turns but also have larger variances. Dilation calculations show a similar trend as presented in Table III.

For our message delivery experiments we implemented wormhole node models [12], [13] with 16 ports and a local port. Messages, also known as worms, are generated at a module attached to the local port at the node. All messages in our simulations, 200 flits long, were generated using uniform traffic model with exponential inter-arrival times. As worms are injected into the network at the local channels, the router at the node, determines which output port to use to route the message using a routing table. If the output port is free, it is immediately committed to the incoming message port for the duration of the message, otherwise the message is blocked until the output port is freed up. The routing tables at each

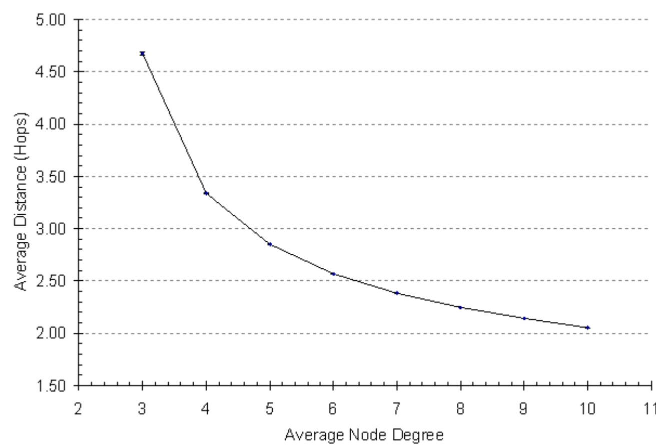

Fig. 5: Average distance in graphs with 64 nodes as a function of the average degree, after SCB turn prohibition.

TABLE II: Comparing the average fraction of prohibited turns (percentage) in bisection-width (2-26) constrained topologies.

	SCB (percent)	Up/Down (percent)
2	18.364 ± 0.062	23.260 ± 0.137
4	18.320 ± 0.065	23.226 ± 0.111
8	17.965 ± 0.073	23.153 ± 0.135
12	17.946 ± 0.068	23.387 ± 0.116
16	17.905 ± 0.058	23.295 ± 0.136
20	17.934 ± 0.060	23.221 ± 0.131
26	17.989 ± 0.061	23.241 ± 0.123

node are generated using the all-pairs shortest path algorithm with an additional constraint that the selected shortest paths do not include any prohibited turns. This way, both deadlock and live-lock conditions are proactively prevented from occurring during the actual routing of messages.

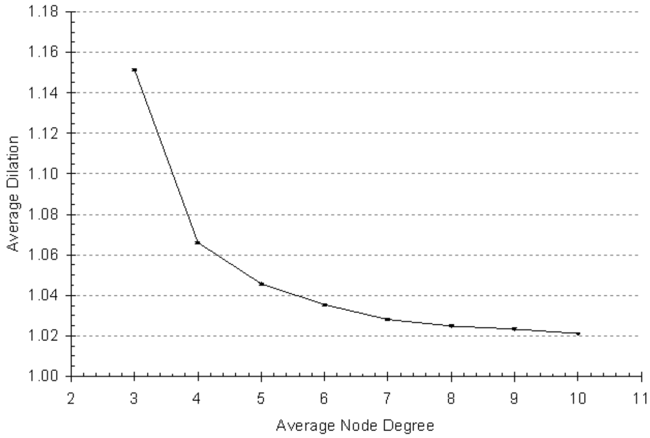

Fig. 6: The average dilation, defined as the ratio of the average distance in a topology after turn prohibitions to the average distance in the topology without any turn prohibitions.

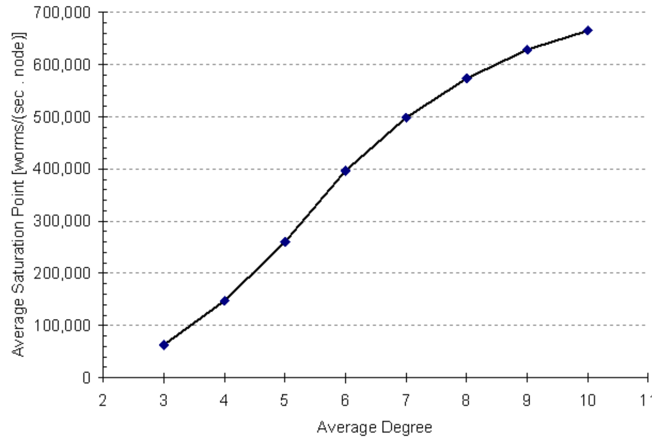

Fig. 7: Saturation points, also known as the maximum sustainable message generation rate per second per node, as a function of the average degree in Hamiltonian topologies.

TABLE III: Comparing the average dilations introduced by SCB and Up/Down algorithms, in bisection-width (2-26) constrained topologies.

	SCB	Up/Down
2	1.122	1.098
4	1.139	1.139
8	1.112	1.170
12	1.107	1.189
16	1.104	1.192
20	1.104	1.195
26	1.101	1.191

TABLE IV: Comparing the average saturation load values (worms /(s.node)) due to SCB and Up/Down algorithms, in bisection-width (2-26) constrained topologies.

	SCB ($\times 10^6$ worms/(s.node))	Up/Down ($\times 10^6$ worms/(s.node))
2	42.7 ± 0.5	46.3 ± 0.5
4	54.7 ± 1.1	57.5 ± 0.8
8	84.0 ± 1.2	70.8 ± 1.0
12	101.0 ± 1.1	73.7 ± 0.9
16	109.0 ± 1.3	77.9 ± 1.0
20	112.0 ± 1.3	79.9 ± 1.0
26	118.0 ± 1.0	82.9 ± 1.0

In Fig. 7 we show the results of average saturation points for Hamiltonian topologies obtained during message delivery experiments. The results are in agreement with the anticipated behaviors. Our results for the bisection width constrained families of graphs are presented in Table IV, again demonstrating the superior performance of the SCB algorithm.

V. Conclusion

In this paper we considered the problem of constructing minimal cycle-breaking sets of turns for graphs that model communication networks. This problem is important for message routing in computer/communication networks for preventing deadlock formation. We present a simple algorithm called the Simple Cycle-Breaking or SCB algorithm which is considerably simpler than those in [4], [12], [13] and has the same performance and time complexity. Earlier cycle-breaking algorithms were complicated, involving as many as ten steps, whereas the SCB algorithm has only three steps and is easy to understand. The complexities of having to deal with cut nodes have totally been eliminated in the SCB algorithm. We also present our simulation results for the fraction of prohibited turns, distance dilation, and the saturation loads (worm generation rate at which the latency tends to infinity) for a number of family of graphs. In all of our experiments, the performance of the SCB algorithm was superior to the performance of the Up/Down algorithm.

References

[1] E. Fleury and P. Fraigniaud, "A general theory for deadlock avoidance in wormhole-routed networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 9, pp. 626–638, 1998.

[2] R. Libeskind-Hadas, D. Mazzoni, and R. Rajagopalan, "Tree-based multicasting in wormhole-routed irregular topologies," *Proceedings of the Merged 12th International Parallel Processing Symposium and the 9th Symposium on Parallel and Distributed Processing*, pp. 244–249, 1998.

[3] L. M. Ni and P. K. McKinley, "A survey of wormhole routing techniques in directed networks," *Computer*, vol. 26, pp. 62–76, 1993.

[4] L. Zakrevski, S. Jaiswal, L. Levitin, and M. Karpovsky, "A new method for deadlock elimination in computer networks with irregular toplologies," *Proceedings of the International Association of Science and Technology for Development (IASTED) Conference, Parallel and Distributed Computing Systems (PDCS-99)*, vol. 1, pp. 396–402, 1999.

[5] R. V. Boppana, S. Chalasani, and C. Raghavendra, "Resource deadlocks and performance of wormhole multicast routing algorithms," *IEEE Transactions on Parallel and Distributed Systems*, vol. 9, no. 6, pp. 535–549, 1998.

[6] W. Dally and H. Aoki, "Deadlock-free adaptive routing in multiprocessor networks using virtual channels," *IEEE Transactions on Parallel and Distributed Systems*, pp. 466–475, 1997.

[7] W. J. Dally and C. L. Seitz, "Deadlock-free message routing in multiprocessor interconnection networks," *IEEE Transactions on Computers*, vol. 36, pp. 547–553, 1987.

[8] J. Duato, "A new theory of deadlock-free adaptive routing in wormhole networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 4, pp. 1320–1331, 1993.

[9] M. D. Schroeder, A. D. Birrell, M. Burrows, H. Murray, R. M. Needham, T. L. Rodeheffer, E. H. Satterthwaite, and C. P. Thacker, "Autonet: A high-speed self configuring local area network using point-to-point links," Digital Equipment Corporation, SRC, Tech. Rep. SRC Research Report 59, April 1990.

[10] M. Koibuchi, A. Funahashi, A. Jouraku, and H. Amano, "L-turn routing: An adaptive routing in irregular networks," in *ICPP*, 2001, pp. 383–392.

[11] C. Glass and L. Ni, "The turn model for adaptive routing," *Journal of ACM*, vol. 5, pp. 874–902, 1994.

[12] L. B. Levitin, M. G. Karpovsky, M. Mustafa, and L. Zakrevski, "New algorithm for finding cycle-breaking sets of turns in a graph," *JGAA*, pp. 387–420, 2006.

[13] M. Mustafa, M. Karpovsky, and L. Levitin, "Cycle breaking in wormhole routed computer communication networks," in *OpnetWork2005*, Opnet Technologies, Inc. Washington, DC.: Opnet Technologies, Inc., August 2005.

[14] T. H. Cormen, C. E, Leiserson, and R. L. Rivest, *Introduction To Algorithms*. The MIT Press, 1989.

[15] D. Starobinski, M. Karpovsky, and L. Zakrevski, "Application of network calculus to general topologies using turn prohibition," *IEEE/ACM Transactions on Networking*, vol. 11, no. 3, pp. 411–421, 2003.

[16] L. Zakrevski, S. Jaiswal, and M. Karpovsky, "Unicast message routing in communication networks with irregular topologies," *Proceeding of Computer Aided Design (CAD-99)*, 1999.

[17] M. Mustafa, M. Karpovsky, and L. Levitin, "Cycle breaking in wormhole routed computer communication networks," in *Opnetwork 2005*, Opnet Technologies. Opnet Technologies, August 2005.