Deadlock Prevention in Network of Workstations with Wormhole Routing

Lev Levitin, Mark Karpovsky, and Mehmet Mustafa

Boston University

USA

Abstract

The problem of preventing deadlocks and livelocks in computer communication networks with wormhole routing is considered. The method to prevent deadlocks is to prohibit certain turns (i.e., the use of certain pairs of connected edges) in the routing process, in such a way that eliminates all cycles in the graph. A new algorithm that constructs a minimal (irreducible) set of turns that breaks all cycles and preserves connectivity of the graph is proposed and analyzed. The algorithm is tree-free and is considerably simpler than earlier cycle-breaking algorithms. The properties of the algorithm are proven and lower and upper bounds for minimum cardinalities of cycle-breaking connectivity preserving sets for graphs of general topology as well as for planar graphs are presented. In particular, the algorithm guarantees that not more than 1/3 of all turns in the network become prohibited.
Experimental results are presented on the fraction of prohibited turns, the distance dilation, as well as on the message delivery times and saturation loads for the proposed algorithm in comparison with known tree-based algorithms. The proposed algorithm outperforms the tree-based algorithms in all characteristics that were considered.

Deadlock Prevention in Network of Workstations with Wormhole Routing

1. Introduction and Related Work

With its simplicity, low channel setup times, and its high performance in delivering messages, wormhole routing has been widely investigated (Dally & Seitz, 1987; Boppana & Chalasani, 1993; Chalasani & Boppana, 1995; Duato, Yalamancili, & Ni, 1997; Ni & McKinley, 1993), and recently is being revisited for Networks-on-Chips technologies (Mello, Ost, Moraes, & Calazans, 2004; Hu & Marculescu, 2004). Wormhole routing and its variants, (Gaughan & Yalamanchili, 1995) virtual cut-through and pipelined circuit switching, PCS, have been used in regular topologies from chip-scale networks (Mello et al., 2004; Hu & Marculescu, 2004), to rack-packed Blue Gene (Klepacki, 2003), to irregular topologies formed by interconnecting low-cost workstations in an ad hoc manner, forming what is referred to as Network of Workstations (NOWs) (Libeskind-Hadas, Mazzoni, & Rajagopalan, 1998; Silla, Duato, Sivasubramaniam, & Das, 1998; Silla & Duato, 2000). Nodes in such networks consist of processing element connected to a router or switching element via a channel with full duplex links. Messages originate and are consumed in the processing elements. Messages that are flowing from a router towards a processing element use what is known as the consumption channel and those that are flowing away from the processing element towards the router use the injection channel. The consumption and the injection channels together form the full duplex communication link between the processor and the router. Routers are connected to other routers in the network using full duplex links as well. Messages, also known as ‘worms’, are made up of flits that are transmitted atomicly, one flit at a time, from node to node in the network. In contrast to this technique which is known as the wormhole routing, in the store and forward routing technique, the message in its entirety is received by each and every intermediate node, and only then it is transmitted to the next node. Therefore, wormhole routing provides for much faster message delivery. The header flit, containing the destination address is immediately followed by the payload or data flits (Ni & McKinley, 1993). Another aspect that makes wormhole routing and routers attractive is that each channel requires only a few flits deep buffer space (Dally & Seitz, 1986; Glass & Ni, 1992). In wormhole routed networks, messages traverse the network in a pipelined fashion, such that parts of the message occupy different network resources, while the header flit requests yet other resources. Under this policy, when there is no contention, as in lightly loaded networks, the latency of message (average delivery time) varies very slowly with the distance (Ni & McKinley, 1993). However, when a message is blocked, the header and the rest of the message wait until the blockage is removed. As a result, messages could hold up potentially large number of network resources while attempting to reserve others.

In congested networks with high injected traffic, improperly designed routing protocols can lead to a network state, in which no progress can be made in delivering, not only of the current messages but all subsequent messages in the network. This network state, in which worms are in a cyclic dependency of each other's held-up resources, is known as **deadlock**. Figure 1 depicts a section of a network in which no measures were taken to prevent deadlock. (The rest of the network where four deadlocked messages have originated is not shown.) The figure shows four-port routers with their local processors presented as circles. Assume that each message M_i is destined for node i. In the figure communication channels have been occupied by the messages shown juxtaposed next to them. The rest of the messages occupy a number of other channels in the network. It can be seen that four messages, $M_1, M_2, M_3,$ and M_4 are blocking each other, so that none of the messages can move forward. For example, message M_2 has acquired ownership of the vertical communication channel south of node 4, within node 4, and north of node 4 but is waiting for the channel between nodes 1 and 2 which has already been committed to M_3.

Figure 1. Deadlock formation in a section of network in which all turns are permitted where message M_i is destined for processor P_i, $i = 1, 2, 3,$ and 4.

When a message from a source is intended to be sent to a single destination, the delivery mechanism of such a message is known as unicast. Network resources that are being held by competing unicast messages are the communication channels in the network. Cycles in channel dependency graphs (CDG), have been identified as the root cause of deadlocks in wormhole networks for unicast messaging. In (Duato, 1993, 1994), it has been shown that necessary and sufficient condition for eliminating deadlocks is the elimination of cycles in the corresponding CDG. Given a graph, constructing the CDG for it is at best tedious. However, cycles of nodes in the CDG graph correspond to”cycles of edges”, as defined below, in the original network graph. A similar problem caused by presence of cycles, the so called”livelocks”, appears in Ethernet type networks.

Because of its susceptibility to deadlocks, considerable body of work has been dedicated to designing wormhole routing algorithms that prevent deadlocks from occurring (Dally & Seitz, 1986, 1987; Dally & Aoki, 1997; Duato, 1993), (Glass & Ni, 1994; Ni & McKinley, 1993; Zakrevski, Jaiswal, Levitin, & Karpovsky, 1999), (Zakrevski, Jaiswal, &

Karpovsky, 1999; Zakrevski, Mustafa, & Karpovsky, 2000; Lysne, Skeie, Reinemo, & Theiss, 2006; Sancho, Robles, Lopez, Flich, & Duato, 2003). In these proactive deadlock prevention schemes, either virtual channels were added (Dally & Seitz, 1986; Lysne et al., 2006), or some resources were prevented from being used.

Virtual channels have been introduced and considered as a tool to avoid deadlocks in a number of papers (Duato et al., 1997; Duato, 1991; Pifarré, Gravano, Denicolay, & Sanz, 1994), often in combination with the dimension-ordered routing (DOR) technique (Min, Ould-Khaoua, Kouvatsos, & Awan, 2004). However, as pointed out in (Duato, 1991), the use of virtual channels may have a negative effect on the message latency.

To provide deadlock-free adaptive routing, Glass and Ni (Glass & Ni, 1994), presented a method that requires neither additional physical nor virtual channels. The turn model is based on analyzing the directions in which packets can turn in regular networks and the cycles that the turns can form. By prohibiting just sufficient number of turns to break all of the cycles, produces a routing algorithm that is deadlock free and livelock free. It was determined that in an n-dimensional mesh only one quarter of the turns must be prohibited to prevent deadlocks. Only 90 degree turns in regular topologies were considered.

The motivation for seeking the minimal fraction of prohibited turns is originally due to Glass and Ni (Glass & Ni, 1994). 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 (Mustafa, Karpovsky, & Levitin, 2005; Levitin, Karpovsky, Mustafa, & Zakrevski, 2006) for irregular topologies as well. Experimental data show that there is a considerable gain of approximately 7-8% in the maximum sustainable throughput in the network, for each percentage point reduction in the fraction of prohibited turns. Similar to spanning tree approaches, prohibiting a carefully selected set of the turns in the network, provides deadlock freedom. 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 simplest deadlock prevention approach utilizes spanning tree based routing for message delivery. Since messages propagate along the tree edges deadlocks are prevented from occurring. However, in this approach a large number of network communication channels are not being used as they are not part of the spanning tree. This is not only inefficient and ineffective use of the available resources but can also lead to hot spots in the network close to the root of the spanning tree. Identification of the root node also could have a significant adverse effect on fraction of prohibited turns. In short, construction of the best spanning tree is not a trivial undertaking.

The Up/Down approach, first introduced in Autonet (Schroeder et al., 1990) routing algorithm improves this shortcoming of the spanning tree approach by using the cross links, non-tree links, under certain conditions. The rooted spanning tree is labeled in partial order determined by the spanning tree. With this labeling the root node has the smallest label and the leaf nodes have the largest labels. Because of the partial order, all edges are given either up or down directional attributes. From routing perspective, a message is permitted to traverse zero or more up links followed by zero or more down links. Once a message has made a transition to a down directed edge, then it is on the down virtual network and it will remain in the down virtual network until it reaches its destination; all down then up directed traversals are forbidden. The Up/Down approach improves on the utilization of resources but it still suffers from the other shortcomings of the spanning tree based approach.

In (Sancho & Robles, 2000) all spanning trees are explored with every node acting as the root node and then the best tree is selected. The selection of the best tree and root combination is accomplished by two heuristic rules with a run-time complexity of O(N^3), where N is the number of switches in the network. According to authors, this approach allows more messages to follow minimal paths and provide better traffic load balancing.

In (Starobinski, Karpovsky, & Zakrevski, 2003) a version of the turn prohibition algorithm was used that enabled generalizing the application of Network Calculus to arbitrary topologies. By prohibiting turns, cycles of independent packet flows were eliminated. Central issue that is tackled in Network calculus is determining conditions which lead to network stability. Because network stability establishment is easier in feed forward topologies, network calculus had been applied only to such topologies in which packets do not create cyclic dependencies. Application of turn prohibition to arbitrary topologies rendered them acyclic, which facilitated the use network calculus successfully in these arbitrary topologies. In their article, authors also demonstrated that the set of prohibited turns generated by the earlier version of the turn prohibition algorithm is not irreducible.

In (Skeie, Lysne, & Theiss, 2002) and (Lysne et al., 2006) a technique called LAyered SHortest, LASH, path algorithm was used that guarantees shortest path routing, in-order message delivery, and that avoids deadlocks by using virtual channels. In this approach a number of source and destination pairs, sd-pairs, are assigned to a virtual channel, or layer. Each sd-pair is assigned to exactly one virtual channel in a way that channel dependencies do not generate cycles in each layer. As an sd-pair is assigned to a layer, if deadlock freedom cannot be assured, another virtual channel is created and the sd-pair is assigned to that layer. It should be noted that virtual channels are not free and there is a cost, hardware and control complexity associated with virtual channels.

With a goal to reduce the restrictions imposed by the current Up/Down routing algorithm, an alternative methodology has been investigated in constructing the routing tables for NOWs. The new methodology is based on computing a depth-first search (DFS) spanning tree of the network graph which decreases the number of routing restrictions. In this approach, first a DFS spanning tree is constructed. Then nodes at the same level are labeled in a way to reduce the number of prohibited turns. Then links are given either Up or Down directionality which is different than what had been defined in (Schroeder et al., 1990). Authors in (Sancho, Robles, & Duato, 2004) defined the Up end of an edge as the edge having the larger label. Using the new methodology prevents the root from being the ”natural path” to other nodes along the DFS spanning tree and thus the hot links at or near the root node are alleviated and throughput have increased by a factor of up to 2.48.

In (Pellegrini, Starobinski, Karpovsky, & Levitin, 2004), a variant of turn prohibition called Tree-Based Turn-Prohibition, TBTP, has been investigated, in which authors show that the TBTP algorithm is of polynomial-time complexity, that it is backward compatible with the IEEE 802.1d standard. This algorithm selects a node and prohibits all turns at the selected node excluding those that involve the spanning tree links, and permits all turns that start with the selected node. A greedy criterion is used that selects a node that maximizes the difference between the number of permitted turns and the number of prohibited turns. All cross-edges at the selected node are then deleted and process is repeated on the remaining graph until there no cross-edges are left in the graph. A distributed version of the TBTP algorithm (Pellegrini, Starobinski, Karpovsky, & Levitin, 2006) is reported by using localized neighborhood concept where only information about two-hop neighborhood is maintained by nodes. The centralized version of the TBTP algorithm (Pellegrini et al., 2004) has reportedly an order of magnitude better message delivery latency performance than the spanning tree approach. The distributed approach is reported to cause some minor performance penalty. With an upper bound of 1/2 for the fraction of prohibited turns, the shortcoming of the TBTP approach is that it could potentially restrict the use of a large number of turns.

In (Shevtekar & Zakrevski, 2004) a hybrid methodology using both proactive and reactive approaches was proposed, in which, routing restrictions are adjusted dynamically based on network congestion. In this hybrid approach for routing some prohibition is imposed on routing together with a type of deadlock recovery. The main idea of the proposed method consists of selecting some of the sequential turns and prohibiting them during routing. A cost formula is proposed to estimate cost of implementing both strategies in a network. The main concern about the deadlock recovery is that identification of the deadlock condition is not a reliable process and current techniques require all messages to be short (Duato et al., 1997).

Another class of deadlock-preventing algorithms, the so-called, **tree-free cycle-breaking algorithms**, were developed in (Zakrevski, Jaiswal, Levitin, & Karpovsky, 1999; Zakrevski, Jaiswal, & Karpovsky, 1999; Zakrevski et al., 2000), (L. B. Levitin et al., 2006;

Mustafa et al., 2005; L. Levitin, Karpovsky, & Mustafa, May, 2009, 2010). These algorithms (TP and SCB algorithms) have been proved to create a minimal (irreducible) set of prohibited turns the size of which never exceeds 1/3 of the total number of turns in any graph. They have been shown to outperform the tree-based algorithms with respect to three basic characteristics: fraction of prohibited turns, distance dilation, latency and the saturation load. For some broad classes of network topologies, those algorithms provide an optimum solution of the turn prohibition problem (Levitin et al., 2010). The computational complexity of the tree-free algorithms is $O(N^2\Delta)$, where Δ is the maximum node degree (number of neighbors) in the graph. The algorithms are topology agnostic. However, the application of those general algorithms may be still unnecessarily complex in the case of graphs with certain regularities in their structure (see next chapter).

In this chapter a simplified and improved tree-free turn prohibition algorithm is proposed referred to as the Simple Cycle-Breaking (SCB) algorithm. The algorithm provides a minimal (irreducible) set of prohibited turns which breaks all cycles and preserves connectivity of the network. The number of prohibited turns is shown to satisfy an upper bound that does not exceed 1/3 of the total number of turns in the network. A better upper bound, which is tight for certain types of graphs, is also established. For certain classes of network topologies, the algorithm yields an optimal solution.

Section 2 of this chapter contains definitions, formulation of the Turn Prohibition (TP) problem and lower bounds on the number of prohibited turns. The SCB algorithm is presented in Section 3. The properties of the SCB algorithm are analyzed in Section 4. It is conjectured that the SCB algorithm is an approximation algorithm for the TP problem with a constant approximation ratio bound. An investigation of planar graphs follows in Section 5.

Experimental results and comparisons with other turn prohibition algorithms are presented in Section 6. Conclusions are given in Section 7.

2. Definitions, Formulation of Problem and Lower Bounds

Consider an undirected connected graph $G(V, E)$, with $N = |V|$ vertices (nodes), denoted by $a, b, \ldots$, and $M = |E|$ edges, denoted by (a, b), etc. that represents a communication network. A **cut node** (**articulation point**) in G is a node whose removal results in a disconnected graph ((Cormen, Leiserson, & Rivest, 1989), Ch. 23). A **turn** in G is a triplet of nodes (a, b, c) if (a, b) and (b, c) are edges in G and $a \neq c$. Assume that turns are bidirectional: turns (a, b, c) and (c, b, a) are considered to be the same turn. If the degree (the 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** $P = (v_0, v_1, \ldots, v_{L1}, v_L)$ 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, $v_0 = a$ and $v_L = b$, and every two consecutive nodes are connected by an edge. Subsequences of the form (v_i, v_k, v_i) are not permitted in a path. Nodes and edges in the path are not necessarily all different. A turn (a, b, c) **belongs** to path $P = (v_0, v_1, \ldots, v_L)$ and P is **covered** by (a, b, c), if $(a, b, c) = (v_i, v_{i+1}, v_{i+2})$, for certain $i = 0, 1, \ldots, L\text{-}2$. Path $P = (v_0, v_1, v_2, \ldots, v_k, v_0, v_1)$ in G is called a **cycle**. Note that, by this definition, a directed edge (v_0, v_1) rather than a node must be repeated in the same direction to make a cycle. Thus, this chapter considers ”cycles of edges”, instead of the more common definition of a cycle as a ”cycle of nodes”.

Consider a set $W(G)$ of turns in G. Any path in G includes at least one turn from $W(G)$ that is prohibited from being used for routing. Therefore, $W(G)$ is called the set of

prohibited turns. The set $W(G)$ is called **cycle-breaking** if every cycle in *G* includes at least one turn from $W(G)$. $W(G)$ is called **connectivity-preserving** if for any $a, b \in V$ there exists a path $P(a, \ldots, b)$ in *G*, such that does not include any turn from $W(G)$. The *minimum* cardinality of cycle-breaking and connectivity-preserving set $W(G)$ for a given graph *G* is denoted by $Z(G)$ and the fraction of prohibited turns is denoted by $z(G) = Z(G) / T(G)$.

Since turn prohibition imposes restrictions on the paths, between nodes, obviously, the smaller $W(G)$ the better. Thus, the **Turn Prohibition** (TP) problem can be formulated as follows.

Given: An undirected graph $G(V, E)$.

Find: A **connectivity-preserving cycle-breaking** (CPCB) set of turns $W(G)$ with the minimum (smallest) number of prohibited turns $Z(G)$.

Conjecture 1. The Turn Prohibition problem is NP-hard.

Though the Turn Prohibition problem is not yet proven to be NP-hard, it looks at least as hard as similar NP-hard graph problems. At the first glance, it is closely related to the well-known Feedback Arc Set (FAS) problem for directed graphs (Ausiello et al., 2003, p 374), which is to find the minimum set of arcs whose removal makes a directed graph acyclic. But in fact these two problems are quite different. The Turn Prohibition problem for a network graph is equivalent to the FAS problem for the corresponding directed channel dependency graph *under the condition that the network graph remains connected*. This additional condition changes radically the nature of the problem. It is easy to construct examples (the simplest is the bipartite graph $K_{2,3}$) where the optimal solution of the FAS problem for the channel dependency graph violates the connectivity of the network graph. Therefore, the known approximation algorithms for the FAS problem (and the related Feedback Vertex Set problem) are not applicable to the TP problem. Thus, one has to look for specific (low complexity) algorithms that would provide suboptimal solutions of the TP problem. This goal is pursued in Section 3 of this chapter.

Let G be a connected graph with minimum degree δ. Consider a set of R cycles in G such that no more than r cycles are covered by the same turn. Then (Levitin et al., 2006), 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

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

Bound (3) is tight for some values of M, N, and δ. For example, for all planar graphs without cut nodes and with degrees 2 and 3, for planar graphs with girth g 6, for the bipartite graph $K_{3,3}$, for the Petersen graph and for all two-dimensional rectangular and honeycomb meshes (see (Zakrevski, 2000) and Section 5 below).

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

In this section an algorithm, called the Simple Cycle-Breaking (SCB) algorithm, is presented that is much simpler than and at least as efficient as those in (Zakrevski, Jaiswal, &

Karpovsky, 1999; Levitin et al., 2006; Mustafa et al., 2005).

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 $c \in G \setminus H$ (i.e., if $a \in H$, $b \in G \setminus H$, and $P(a,b)$ is a path from node a to node b, then $c \in P(a,b)$).

Proof**:** Suppose G has cut nodes. Let S_i be the set of connected components of G obtained by deleting cut node $c_i (i = 1, 2, \ldots)$ from G. Consider the union $\bigcup_i S_i$. Let $H \;_i S_i$ 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 $G \setminus H$ 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

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

where d_i $(i = 1, 2, \ldots, d)$ 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

$$\sum_{i=1}^{d}(d_i-1)\ge(d'-1)(d-1)+(d-d')d+(d'-1)$$

$$\ge d(d-1)=2\binom{d}{2}.$$ +

Lemma 2 will be used below to prove properties of a new algorithm for obtaining a minimal cycle-breaking set of turns.

Given a connected graph $G(V,E)$, the SCB algorithm creates two sets: the set $W(G)$ 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, $W(G)=\varnothing$, $A(G)=\varnothing$, and all nodes are unlabeled. If $|V|=N$, the algorithm consists of $N-1$ stages. 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 $W(G)$ and $A(G)$. Otherwise, go to Step 2.

2) Select a non-cut node a of the minimum degree d, such that inequality (4) is satisfied. Prohibit all turns of the form (b,a,c) and include them in $W(G)$. Permit all turns of the form (a,b,c) and include them in $A(G)$. Label a by the smallest unused natural number $\ell(a)$.

3) Delete node a to obtain a 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 $\ell(a)<\ell(c)$. The prohibition rule for the SCB algorithm can be expressed in different terms.

Let us call an edge (a,b) **positive**, if $\ell(a) < \ell(b)$, and **negative** otherwise. Then the path *P* is prohibited *iff* it includes a pair of consecutive edges such that **the first edge is negative** and **the second edge is positive**. Then the connectivity means that SCB algorithm labels nodes in such a way that for any two nodes there exists a path between them in which all positive edges (if any) precede all negative ones (if any).

Example 1. Figure 2 demonstrates the operation of the SCB algorithm. The original graph is shown in Figure 2(A). Since there are 11 nodes the completion of the algorithm would involve 10 stages. Before the algorithm begins to execute, the two sets $W(G)$ 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, the minimum degree non-cut node is selected. Since node *f* is a cut node, it cannot be selected. The minimum degree non cut nodes are nodes *a, b, c, d, h, k, m,* and *n*. The criterion in (4) is applied and all of the candidate nodes satisfy the inequality. For example, for node *a*, both the left and right hand side of (4) evaluate to 6, hence, node *a* is selected. As shown in Figure 2(B), three turns are prohibited, denoted by three arcs, i.e., (*b, a, c*)*,* (*b, a, e*)*,* (*c, a, e*). 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 10 nodes as shown in Figure 2(D). The SCB algorithm begins executing the stage 2. In this stage, node *c* is selected, one turn (*b, c, d*) prohibited, node *c* is labeled with 2, and deleted to obtain the 9-node subgraph shown in Figure 2(D), and stage 3 of the algorithm begins. Since node *b* is of the minimum degree 1, and a non-cut node, it is selected, no turns are prohibited, node *b* is labeled 3 and deleted. In stages 4, 5, and 6, nodes *d*, *e*, and *f* are selected and labeled as shown. In stage 7, node *g* is selected, one turn, namely (*h, g, n*), is prohibited, node *g* is labeled 7, deleted, and stage 8 begins executing. The stages 8 and 9 select nodes *h* and *m* in that order, and prohibit the indicated turns. During stage 10, Step 1 of the algorithm labels nodes *k* and *n* and the algorithm terminates. In Figure 2(E) the graph is shown with all of the prohibited turns and the node labels. It is clear that in all prohibited turns (*x, y, z*) the labels satisfy $\ell(y) < \ell(x)$ and $\ell(y) < \ell(z)$. The stage-by-stage operation of the algorithm is also shown in Table 1 in which each row corresponds to a stage.

Figure 2. Example demonstrating the operation of the SCB algorithm resulting in a fraction of prohibited turns of $z(G) = 7/31$.

Table 1. Stage-by-Stage Operation of the SCB Algorithm

Selected Node	Node Label	Set of Prohibited Turns
a	1	$\{ (b,a,c),(b,a,e),(c,a,e) \}$
c	2	$\{ (b,c,d) \}$
b	3	$\varnothing$
d	4	$\varnothing$
e	5	$\varnothing$
f	6	$\varnothing$
g	7	$\{ (h,g,n) \}$
h	8	$\{ (k,h,m) \}$
m	9	$\{ (k,m,n) \}$
k h	10 11	$\varnothing$ $\varnothing$

4. Properties of the SCB Algorithm

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

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

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 $W(G)$.

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

Property 4. For any graph G, $W(G) \le T(G)/3$, 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 $x, y \in G \setminus a$. 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 exits a path between any two unlabeled nodes x and y. Consider now paths from a labeled node u, $\ell(u) \le \ell(a)$ 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 (Figure 3). Then, let

x be the first unlabeled node in the path from u to v or from u to y, and z be the last unlabeled node in the path from u to v. The part of P from x to y, or from x to z, can be replaced 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 z' be the labeled node that immediately follows z in P and P^* (in the case when such a node exists). Since all turns (x', x, w) and (w, z, z') 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. +

Figure 3. Figure depicting the state of the graph at stage $n+1$ 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 $P^* = (u, \ldots, x', x, w, \ldots, s, t, t', \ldots, v)$ is permitted since it does not involve any prohibited turns.

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, a cycle $C = (a, b, P, c, a, b)$ is obtained. 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 $W(G)$, a cycle would be created in G and violate the cycle-breaking Property 1. Thus, set $W(G)$ is minimal. +

Proof of Property 4. At the stage of the algorithm when node a is selected (stage $\ell(a)$, all turns (b, a, c) 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 (in the subgraph that remains at stage $\ell(a)$) ; the number of permitted turns is $\sum_{i=1}^{d}(d_i - 1)$, where d_i, $(i = 1,\dots,d)$ 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 $W(G) \le T(G)/3$. +

In general, the fraction of prohibited turns yielded by the 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 $|E| = n(n-1)/2$. Indeed, the closer is a graph to a complete one, the larger is the fraction of prohibited turns, as shown by the following theorem which provides a better upper bound on the ratio $|W(G)|/T(G)$.

Theorem 2. *Let $G = (V, E)$ be a connected graph with N nodes and M edges. The fraction of prohibited turns $z_{SCB}(G)$ yielded by the SCB algorithm satisfies the upper bound:*

$$z_{SCB}(G) = \frac{|W(G)|}{T(G)}$$

$$\le \frac{1}{3} - \frac{2N - 3 - \sqrt{8\beta + 1}}{3[2N + (\beta - 1)(\sqrt{8\beta + 1} + 3)]}, \qquad (5)$$

where $\beta = M - N + 1$ is the cyclomatic number.

Proof. When a node is selected in the course of the SCB algorithm, all its edges are deleted. Thus, if d_ℓ is the degree of node with label ℓ at the stage when it is selected, then

$$\sum_{\ell=1}^{N} d_\ell = M. \qquad (6)$$

The total number of prohibited turns is

$$|W(G)| = \sum_{\ell=1}^{N} \frac{d_\ell (d_\ell - 1)}{2}. \quad (7)$$

Note that, for the SCB algorithm,

$$d_{\ell+1} \geq d_\ell - 1. \quad (8)$$

(Otherwise, the nodes would be selected in the opposite order). Obviously, $d_N = 0$ $d_{N-1} = 1$. The quadratic sum (7) under the constraint (6) achieves maximum if the values of d_ℓ are maximally unequal, so that some of them are as large as possible. Looking at the sequence (d_ℓ) in the backward direction, from $\ell = N$ to $\ell = 1$, one can see that, because of (8), the sequence can increase only by 1 from one term to the another: $d_N = 0, d_{N-1} = 1, d_{N-2} \leq 2, \ldots, d_{N-i} \leq i$. Hence, there exists a subsequence (d_{ℓ_j}) such $d_{\ell_j} = j$, where j takes on all integer values from 0 to a certain k. The value of $|W(G)|$ achieves its maximum, if k is the largest integer that satisfies two conditions. On one hand,

$$M \geq \sum_{j=0}^{k} j = \frac{k(k+1)}{2}. \quad (9)$$

On the other hand, since the graph remains connected through the course of the algorithm, the number of remaining edges should be no smaller than the number of remaining nodes:

$$M - \frac{k(k+1)}{2} \geq N - (k+1). \quad (10)$$

The number of prohibited turns in the nodes of the subsequence (d_{ℓ_j}) is

$$\sum_{j=1}^{k} \frac{j(j-1)}{2} = \frac{(k+1)k(k-1)}{6}. \quad (11)$$

The upper bound on $|W(G)|$ is obtained for the value of k (not necessarily an integer) that turns (10) into equality, i.e. for the root of the equation:

$$M - N = \frac{(k-2)(k+1)}{2}. \tag{12}$$

Hence,

$$k = \frac{1+\sqrt{8(M-N+1)+1}}{2} = \frac{1+\sqrt{8\beta+1}}{2}. \tag{13}$$

Then, by (11),

$$|W(G)| \leq \frac{(M-N+1)\left(\sqrt{8(M-N+1)+1}+3\right)}{6}$$

$$= \frac{\beta(\sqrt{8\beta+1}+3)}{6}. \tag{14}$$

Now let us estimate the total number of turns. According to the proof of Property 4 in Theorem 1, if the degree of the selected nodes is k, then there exist at least k other nodes with the sum of degrees at least k^2. The total number of turns at these $k + 1$ nodes is minimal, if all degrees are equal to $d = k$. Since the graph is connected, the remaining $N - k - 1$ nodes add at least $N - k - 1$ turns. Thus the total number of turns $T(G)$ satisfies the inequality

$$T(G) \geq \frac{(k+1)k(k-1)}{2} + N - k + 1, \tag{15}$$

where k is given by (13).

It follows that the fraction of prohibited turns $z(G)$ is upperbounded by

$$z(G) \leq \frac{|W(G)|}{T(G)}$$

$$\leq \frac{1}{3}\left[1 - \frac{2N-3-\sqrt{8\beta+1}}{2N+(\beta-1)(\sqrt{8\beta+1}+3)}\right]. \qquad (16)$$

+

Bound (14) and (16) are tight for all values of M and N such that $M - N = \frac{(k-2)(k+1)}{2}$, where k is a natural number and $k < N$, in particular, for a tree $(M = N-1)$, for a ring $(M = N)$, and for a complete graph $K_N \left(M = \frac{N(N-1)}{2} \right)$.

Note that bound (16) converges to $1/3$ iff the cyclomatic number $\beta = M - N + 1 = \Omega(N^{2/3})$. It will be shown below (see Section 3) that for some classes of graphs, the SCB algorithm guarantees that the fraction of prohibited turns is substantially smaller than that given by bound (16).

Theorem 3. *The fraction of prohibited turns is* $z(G) = 1/3$ *iff* $G = K_N$.

Proof. It is seen from (16) that if $M < \binom{N}{2}$ then $z(G) < 1/3$. For a K_N, $M = \binom{N}{2}$, and then $z(G) \leq 1/3$. On the other hand, there are $\binom{N}{3}$ turn-disjoint triangles and $N\binom{N-1}{2}$ turn in the complete graph K_N. Hence, by use of bound (2), $z(K_N) \geq 1/3$ is obtained. It follows that $z(G) = 1/3$ iff G is a complete graph K_N. +

Though in general, the SCB algorithm is suboptimal, it can be readily shown by the use of the lower bounds (1)-(3) that the algorithm provides optimal solutions, for several broad classes of network topologies, in particular, for the following:

- All n-dimensional p-ary meshes and tori (including hypercubes $\mathbf{Z}_2^n$);
- All hexagonal and honeycomb meshes and tori;
- All homogeneous meshes;
- All fractahedrons;
- All graphs for which the bound (14) is achieved (including all complete graphs);
- All planar graphs without cut-nodes and with degrees not larger than 3;
- All planar graphs of girths $g \geq 6$;
- Certain special graphs, e.g. all Plato's polyhedra, the Petersen graph, the bipartite graph $K_{3,3}$, etc.

Earlier versions of a tree-free turn prohibition algorithms (the TP and CB algorithms) were presented in (Zakrevski, Jaiswal, & Karpovsky, 1999; Mustafa et al., 2005; Levitin et al., 2006). They have been shown to outperform the Up*/Down* algorithm in terms of the fraction of prohibited turns, average distance between nodes, and saturation load (see (Mustafa et al., 2005; Levitin et al., 2006)). 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 (Levitin et al., 2006)). In contrast, the SCB algorithm does not use recursive calls, but only iterations, and it has only three steps in each iteration, which is easier for implementation and reduces the memory requirements. 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 time complexity of SCB algorithm remains the same as in previous works ((Mustafa et al., 2005; Levitin et al., 2006)), the practical implementation is substantially simpler.

A straightforward evaluation of the SCB worst-case time complexity is $O(NM)$. This follows from the fact that it takes $O(M)$ time to determine all cut nodes ((Cormen et al., 1989), Ch. 23, Problem 23-2).

Conjecture 2. The SCB algorithm is a polynomial-time approximation algorithm for the turn prohibition problem with a constant approximation ratio bound:

$$\frac{Z_{SCB}(G)}{Z(G)} \leq \rho = const.$$

Here $Z_{SCB}(G)$ and $Z(G)$ are the numbers of prohibited turns in the solution of the TP problem obtained by the SCB algorithm and in the optimal (minimum) solution, respectively.

5. Turn Prohibitions for Planar Graphs

Planar graphs defined as those which can be embedded in a plane without any crossing edges form an important class of graphs. A large number of physical problems such as transportation highways (without underpasses), telecommunication networks, and physical circuit (or component) layout problems are modeled by planar graphs. For example, for proper operation, all physical layout problems in a printed circuit board as well as VLSI designs involve constructing conductive (metallic) signal pathways that must be prevented from crossing each other; therefore, these problems naturally map into planar graphs. In VLSI chips, either the entire chip or large sections of the chip are modeled by planar graphs

(Agarwal, Mustafa, & Pandya, 2006). In (Agarwal et al., 2006; Agarwal, Mustafa, Shankar, Pandya, & Lho, 2007) authors introduced turn prohibition in Network-On-Chips (NOC) architectures where multiple processing elements are networked on one VLSI chip, in which the layout is planar. In this section constructive upper bounds on the minimum fraction of turns, $z(G)$ to be prohibited to break all cycles in any planar graph G are presented.

An important characteristic of a planar graph is the number of edges in the shortest cycle known as its **girth**.

Lemma 3. *The average degree* $\bar{d}$ *in a planar graph with N nodes and girth g obeys inequality*

$$\bar{d} \le \frac{2g}{g-2} - \frac{4g}{N(g-2)}. \tag{17}$$

Proof. Let G be a planar graph with F faces and girth $g(G) = g$. Since each edge belongs to either one or two faces, it follows that $2M \ge \sum_{j=1}^{F} g_j \ge Fg$ where g_j is the number of edges of face j. Hence

$$F \le \frac{2M}{g}. \tag{18}$$

Substituting (18) into the Euler equation $F = M - N + 2$, one obtains

$$M \le \frac{g(N-2)}{g-2}. \tag{19}$$

Thus, the average node degree is

$$\bar{d} = \frac{2M}{N} \le \frac{2g}{g-2} - \frac{4g}{N(g-2)}.$$ +

It is seen that the upper bound on d given by (17) decreases monotonically with girth.

Since the probability of a deadlock in a network with a uniform traffic and without turn prohibitions decreases with increasing girth, it may be conjectured that the cost of deadlock prevention in terms of the fraction of turns required to be prohibited decreases as well as the girth increases. The following theorem proves this idea.

Theorem 4. *If G is a planar graph without triangles, then*

$$z(G) \le \frac{1}{4}. \quad (20)$$

Proof. With no triangles in a planar graph, the girth of the graph is $g(G) = g \ge 4$. Then the average degree $\bar{d}$ in (17) becomes $\bar{d} \le 4 - 8/N$, which means that a planar graph without triangles contains at least two nodes of degree less than four. Note that any subgraph of G is also a planar graph with girth at least four and an average degree $\bar{d} \le 4 - 8/n$, where n is the number of nodes in the subgraph. By Lemma 1, there exists a subgraph H of G that consists of non-cut nodes only and connected to the rest of the graph by at most one cut node. Consider a subgraph of G formed by subgraph H and this cut node. It follows that this graph contains a non-cut node of degree at most 3.

At every step of the execution of the SCB algorithm (see Section 3), a minimum degree node is selected according to the rule (4). Let $A_i (i = 1, 2, 3)$ be the number of nodes of degree i that were selected during the execution of the algorithm. Since the last node left is a node of degree zero and all edges are deleted in the course of the algorithm,

$$A_1 + A_2 + A_3 = N - 1, \quad (21)$$

$$A_1 + 2A_2 + 3A_3 = M. \quad (22)$$

Hence,

$$A_2 + 2A_3 = M - N + 1 = F - 1. \quad (23)$$

Note that the number of prohibited turns in the SCB procedure is given by

$$Z = A_2 + 3A_3. \tag{24}$$

Hence, an upper bound for Z would correspond to a maximal A_3 and a minimal A_2. Obviously, the deletion of a degree 2 node decreases the number of faces by 1, and the deletion of a degree 3 node decreases this number by 2. The algorithm terminates when the number of faces is reduced to 1. It is easy to show, using (17), (18), and the Euler equation, that for girth $g \geq 4$,

$$\overline{d} \leq \frac{4(F-2)}{F(g-2)+4} + 2 \leq \frac{4(F-2)}{2(F+2)} + 2 = 4 - \frac{8}{F+2}.$$

Hence, for any graph with girth $g \geq 4$ ($F \leq 5$), $\overline{d} = 3 - \frac{1}{7} < 3$.

Thus any such graph has non-cut nodes of degree 2 or 1. Therefore, $A_3 \leq (F-4)/2$ and $A_2 \geq 3$ (provided $F \geq 4$). Then, using (23) and (24):

$$Z = \frac{3}{2}(A_2 + 2A_3) - \frac{A_2}{2} = \frac{3}{2}(M - N + 1) - \frac{A_2}{2}.$$

Finally, since $A_2 \geq 3$, it follows that

$$Z(G) \leq Z \leq \frac{3}{2}(M - N). \tag{25}$$

Here, $Z(G)$ is the minimum number of prohibited turns for G. To estimate the total number of turns $T(G)$, we note that there are two cases; first, when the average degree $\overline{d} = 2M/N$ is $3 \leq \overline{d} \leq 4 - 8/N$ and second, when $\overline{d} < 3$. In the first case, $T(G)$ is minimal if nodes are of degree 3 and degree 4 only and in the second case if nodes are of degree 2 and degree 3 only. Assuming first that nodes are of degree 3 and 4 only, we determine that $N_3 = 4N - 2M$ and $N_4 = 2M - 3N$, where N_3 and N_4 designate the number of nodes of degree 3 and degree 4, respectively, in the graph. It follows that $T(G) = 6(M - N)$, and the fraction of prohibited turns $z(G) = Z(G)/T(G)$ is

$$z(G) \le \frac{1}{4}. \tag{26}$$

For the case when $2 \le \bar{d} < 3$, $T(G)$ is minimal if there are N_2 nodes of degree 2 and $N_3 = N - N_2$ nodes of degree 3. Then $N_2 = 3N - 2M$, and $N_3 = 2(M - N)$, and $T(G) = 4M - 3N = 6(M - N) + (3N - 2M) \ge 6(M - N)$ (since $\bar{d} = 2M / 3 < 3$). Hence in both cases the upper bound for the fraction of prohibited turns is

$$z(G) \le \frac{1}{4}. \tag{27}$$

+

Figure 4. Planar graph with girth g = 8, N = 32, M = 40, and T = 64. Prohibited turns are shown as arcs.

Theorem 5. *If G is a planar graph with N nodes and girth* $g \ge 6$*, then*

$$z(G) \le \frac{2}{g+6} - \frac{(g-2)(g-6)}{(g+6)\left[g(N-8)+6N\right]}, \tag{28}$$

and

$$z(G) \le \frac{2}{g+6}. \tag{29}$$

Proof. Note that, by (17), for girth $g = 6$ the average degree becomes $\bar{d} \le 3 - 6/N < 3$. For the case of $g \ge 6$, if there are only N_2 nodes of degree 2 and N_3 nodes of degree 3, we get $N_2 = 3N - 2M$, $N_3 - 2(M - N)$, and the total number of turns, $T(G) = N_2 + 3N_3 = 4M - 3N$. Since $T(G)$ achieves minimum if the degrees take values closest to the given average degree, it follows that $T(G) \ge 4M - 3N$. By the same argument that is given in the first paragraph of the proof of Theorem 4, there will always be non-cut nodes of degree at most 2 available for selection at every step of the algorithm; and therefore

$A_3 = 0$. From $A_1 + A_2 = N - 1$ and $A_1 + 2A_2 = M$, we obtain that $Z = A_2 = M - N + 1 = F - 1$ and the upper bound for the fraction of prohibited turns will be

$$z(G) \le \frac{M - N + 1}{4M - 3N}.$$

Substituting $x = M / N$, we get

$$z(G) \le \frac{1}{4} - \frac{1/4 - 1/N}{4x - 3}. \tag{30}$$

The right-hand side of (30) is a monotonically increasing function of x. Note that from $\overline{d} = 2M / N = 2x$ we get

$$x \le \frac{g}{g-2} - \frac{2g}{N(g-2)}.$$

Substituting the maximum value of x into (30) we obtain

$$z(G) \le \frac{2}{g+6} - \frac{(g-2)(g-6)}{(g+6)[g(N-8)+6N]},$$

and

$$z(G) \le \frac{2}{g+6}.$$ +

The bound (28) is tight as shown by the following example.

Figure 5. An infinite planar graph with average degree $\overline{d} = 10/3$

Example 2. Consider the planar graph G of girth $g = 8$ shown in Figure 4, with N = 32, M = 40, and T = 64. For this graph, $Z = M - N + 1$ and $z(G) = 9/64$ is equal to the right-hand part of inequality (28).

To avoid misunderstanding, let us point out that it is planarity and girth constraints that result in (20) and (28), but not the limits on the average degree alone. It is easy to construct graphs with average degree $\overline{d}$ arbitrarily close to 2, for which $z(G)$ is arbitrarily close to 1/3.

For girth $g(G)=5$ planar graphs, the average degree in (17) becomes

$$\overline{d} \leq \frac{10}{3} - \frac{20}{3N}. \tag{31}$$

This bound is tight and is achieved for example for the infinite graph of Figure 5.

For $N<20$ ($F<12$) the average degree $\overline{d}<3$ and it follows that such graphs would always have a node of degree 2.

Conjecture 3. If G is a planar graph with girth $g(G) = 5$, then $z(G) \leq 1/5$. Note that Theorems 4 and 5 do not apply to non-planar graphs. For example, for bipartite graph $K_{4,4}$ with $N = 8$ nodes, we have $z(K_{4,4}) = 14/48 > 1/4$.

The proofs of Theorem 4 and Theorem 5 suggest a somewhat more general result.

Theorem 6. *If in the course of the SCB algorithm, all selected nodes are of degree 2 or less, then the solution given by SCB is optimal, and,*

$$|W(G)| = M - N + 1. \tag{32}$$

Proof. The result follows immediately from the expressions (23) and (24), and lower bound (1). +

In particular, the SCB algorithm provides an optimal solution for 2-dimensional rectangular meshes and honeycomb meshes (Parhami, 1998) – two popular network topologies.

6. Experimental Results

In this section the results of simulation experiments are presented with turn prohibition for the SCB, Up*/Down*-DFS (where the spanning tree is constructed by depth-first search), Up*/Down*-BFS (where the spanning tree is constructed by breadth-first search), and L-turn algorithms. Simulation experiments involving message delivery simulations were performed using the Modeler from OPNET Technologies. Modeler provides a powerful discrete event simulation environment where flits are tracked and timed from their creation at the source processor to their consumption at the destination processor. In all of the experiments and simulations, network topologies were first generated using tools that were developed for this purpose. All of the topologies used in our work are represented by 64-node undirected graphs. Seven hundred different connected irregular graphs of various average degrees were generated, 100 graphs each for average degrees 4, … , 10. In each graph, a node could have a degree from 1 to 16. Four algorithms that were investigated are then used to prohibit turns in each graph. When determining the prohibited turns, the algorithms as defined in (Koibuchi et al., 2001) for the L-turn, in (Schroeder et al., 1990) for the Up*/Down*-BFS, and in (J. Sancho & Robles, 2000; J. Sancho et al., 2000; J. C. Sancho et al., 2004) for the Up*/Down*-DFS were used. Exactly the same spanning trees have been used for both the L-turn and Up*/Down*-BFS algorithms. In the case of the Up*/Down*-DFS, following (Sancho et al., 2004), we used the heuristic to select the next node to be added to the already constructed depth first spanning tree. The set of prohibited turns, the fraction of prohibited turns, the average distance and dilation calculations are then performed to obtain the results presented next.

Figure 6. Fraction of prohibited turns as a function of the average degree. Each point is the average of the results of 100 different random graphs.

In Figure 6 the results for the fraction of prohibited turns are shown. It can be noted that the SCB algorithm consistently performs better than the three other algorithms.

Consider now the notion of dilation in a network topology due to turn prohibitions. Paths that involve prohibited turns are prohibited and are not used for communication. Thus, one side effect of turn prohibitions is that, prohibiting certain paths from being used for message routing, may increase distances between some nodes. The net result of this is that the average distance of the network graph will be increased. To facilitate the investigation of this phenomenon, we introduced the notion of **distance dilation** which we define as the ratio of the average distance after turn prohibitions to the average distance without any turn prohibitions. When the dilation is 1 it would imply that the turn prohibitions have not caused any lengthening of the average distance. For example, in complete graphs, the fraction of prohibited turns achieves the upper bound, but the dilation is 1.

Figure 7. Average distance dilation for 64 node graphs as a function of the average degree. Each point is the average of the results of 100 different random graphs.

In Figure 7 we show the results of the dilation calculations for the four algorithms for random graphs. One can see that the SCB algorithm yields dilation in the range between 3.2%-9.6%.

Calculations for *planar* graphs follow a similar approach. We first generated 6 families of planar graphs with girths from 3, ... , 8, each family of 100 graphs. In all of our constructions, all faces (including the infinite one) are all regular and have the same number of edges equal to the girth g. After the planar graphs are generated, we then applied the SCB algorithm to break all cycles as before. Results of these calculations are shown below. In Figure 8, the fractions of prohibited turns are shown for families of planar graphs with girths 3 through 8 together with the theoretical upper bounds (20) and (28). In this figure we also show the fraction of prohibited turns for an icosahedron ($g = 3$, $z(G) = 4/15$) and the dodecahedron ($g = 5$, $z(G) = 1/5$).

Figure 8. Fraction of prohibited turns for SCB in 64 node planar graphs as a function of girth.

Results of dilation calculations for planar graphs are shown in Figure 9, where we see that as the girth increases the average dilation increases, predicting a better performance for message delivery times for planar graphs with smaller girths. In particular, for girths 3 and 4 topologies, the dilations are very small (1.0002 and 1.0009 respectively).

Figure 9. Average distance dilation in planar topologies with 64 nodes after the SCB generated turn prohibitions as a function of girth.

For message delivery experiments, we implemented wormhole node models (Mustafa et al., 2005; Levitin et al., 2006; Mustafa, Levitin, & Karpovsky, August 2006) with 16 bidirectional full-duplex ports and a local port. Messages, also known as ”worms”, are generated at a module attached to the local port of each 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 via the local channel, the router at the node determines, using a routing table, which output port to use to route the message. 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. A routing table at each node is generated using the all-pairs shortest path algorithm with an additional criterion that the selected shortest paths do not include any prohibited turns. With this approach both deadlock and live-lock conditions are proactively prevented from occurring during the actual routing of messages. We believe that using the same underlying wormhole routing models and changing only the routing tables generated identically using the prohibited turns generated by the various algorithms is the fairest way of comparing the performances of the algorithms. (Of course, it must be taken into account that turn prohibitions in the L-turn algorithm are unidirectional). The simulations are repeated with message generation rate increasing step by step until the latency becomes at least two orders of magnitude larger than for the smallest generation rate. The same experiment is repeated with all of the graphs and for all algorithms.

Figure 10. End-to-end message latency as a function of average message generation rate for 64-node degree 4 random graphs.

Figure 10 shows the dependence between the latency (message delivery time) and the average load (message generation rate). The load has been averaged over all 100 graphs of average degree 4 for each fixed latency value. For low offered loads all algorithms perform equally well. As the offered load is increased we note that the SCB algorithm maintains lower latency values for larger offered loads.

The average saturation points for all 100 graphs of each average degree have been calculated and plotted in Figure 11.

Figure 11. Saturation points (maximum sustainable message generation rates per second per node) as a function of average node degree for 64 node random graphs.

The results of the SCB algorithm applied to planar graphs are presented in Figure 12. These results are in agreement with the anticipated behavior. In particular, since the average distance increases with girth in planar networks with a given number of nodes, one would expect better performance in networks of smaller girths. Indeed as seen at Figure 12, the saturation load increases for smaller girths.

Figure 12. Saturation points (maximum sustainable message generation rates per second per node) as a function of the girth of 64-node planar graphs.

Our experiments clearly demonstrate the superior performance of the SCB algorithm as compared to the tree-based algorithms. When the SCB algorithm is used, the fraction of prohibited turns is smaller by up to 17.9%, 23.2%, and 37.2% than when Up*/Down*-DFS, Up*/Down*-BFS, and L-turn algorithms are used, respectively. The distance dilation for the SCB algorithm is smaller than for the tree-based algorithms by a factor between 1.64 and 2.72. The SCB algorithm provides for the increase in the maximum sustainable message generation rate of up to 95% over L-turn, up to 92% over Up*/Down*-BFS and up to 51% over Up*/Down*-DFS.

Perhaps, what is even more significant is the difference in the network performance in the ”working range” of loads. With the SCB algorithm, when the message generation rate increases from 0 to 90,000 messages per second per node, the latency increases only by a factor of 2.7, while with other algorithms the network becomes practically saturated. Among the tree-based algorithms, the Up*/Down*-DFS algorithm seems to be the best, but the application of heuristic rules for constructing the spanning tree involves a substantial computational cost of $O(N^3)$. Remarkably, though the fraction of prohibited turns is considerably larger for the L-turn algorithm than for the Up*/Down*-BFS algorithm, the difference between them in terms of the saturation load is not large (actually, L-turn algorithm looks better for dense networks). A possible explanation of this observation is that L-turn algorithm provides for a more even distribution of the traffic, while the Up*/Down*-BFS algorithm suffers from congestion at and near the root.

7. Conclusions

This chapter considers the problem of constructing minimal connectivity-preserving cycle-breaking (CPCB) set of turns for graphs that model communication networks. This problem is important for deadlock-free and livelock-free message routing in computer communication networks. We formulate the general Turn Prohibition (TP) problem of finding such a set with minimum number of prohibited turns and conjectured that the problem is NP-hard. We prove lower bounds on the minimum number of prohibited turns. We present a new algorithm called the Simple Cycle-Breaking (SCB) algorithm and conjectured that this algorithm satisfies a constant approximation ratio bound for the TP problem. In contrast with a number of other turn prohibition algorithms (e.g., Up*/Down*-BFS, Up*/Down*-DFS, and L-turn algorithms), the SCB algorithm is tree-free, i.e., it does not use a spanning tree for construction of the CPCB set of prohibited turns. Therefore, it is free of problems stemming from the choice of the spanning tree and its root. The SCB algorithm implements a procedure to label all nodes and to prohibit or permit turns according to this labeling.

The SCB algorithm guarantees that the fraction of prohibited turns does not exceed an upper bound that is less than or equal to 1/3, as given by (5), and that the set of prohibited turns is minimal (irreducible). Comparison with lower bounds shows that the SCB algorithm yields optimal solutions for broad classes of network topologies. An important class of network topologies, namely, planar graphs, has been investigated, and stricter upper bounds on the number of prohibited turns produced by the SCB algorithm have been obtained. In particular, for a graph without triangles, the fraction of prohibited turns does not exceed 1/4.

Experimental results show that the SCB algorithm dramatically outperforms the tree-based algorithms in terms of all four basic characteristics: the fraction of prohibited turns, the distance dilation, the message delivery time, and the saturation load.

References

Agarwal, A., Mustafa, M., & Pandya, A. S. (2006, May). *QOS driven network-on-chip design for real time systems*. In (p. 1291-1295). Ottawa: IEEE CCECE/CCGEI.

Agarwal, A., Mustafa, M., Shankar, R., Pandya, A. S., & Lho, Y. (2007, April). *A deadlock free Router design for network on chip architecture*. Journal of Korea Institute of Maritime Information and Communication Sciences, 11 (4), 696-706.

Ausiello, G., Crescenzi, P., Gambosi, G., Kann, V., Marchetti-Spaccamela, A., & Protasi, M. (2003). *Complexity and approximation*. Berlin Heidelberg New York: Springer-Verlag.

Boden, N. J., Cohen, D., Felderman, R. E., Seitz, C. L., Seizovic, J. N., & Su, W.-K. (1995). *Myrinet: A gigabit per second local area network*. IEEE Micro, 29-35.

Boppana, R., & Chalasani, S. (1993). *A comparison of adaptive wormhole routing algorithms*. Computer Architecture News, 21 (2), 351-360.

Boppana, R. V., Chalasani, S., & Raghavendra, C. (1998). *Resource deadlocks and performance of wormhole multicast routing algorithms*. IEEE Transactions on Parallel and Distributed Systems, 9 (6), 535-549.

Chalasani, S., & Boppana, R. V. (1995). *Fault-tolerant wormhole routing algorithms in mesh networks*. IEEE Transactions on Computers, 44 (7), 848-864.

Cormen, T. H., E, C., Leiserson, & Rivest, R. L. (1989). *Introduction to algorithms*. The MIT Press.

Dally, W., & Aoki, H. (1997). *Deadlock-free adaptive routing in multiprocessor networks using virtual channels*. IEEE Transactions on Parallel and Distributed Systems, 466-475.

Dally, W. J., & Seitz, C. L. (1986). *The torus routing chip*. Journal of Distributed

Computing, 1 (3), 187-196.

Dally, W. J., & Seitz, C. L. (1987). *Deadlock-free message routing in multiprocessor interconnection networks*. IEEE Transactions on Computers, 36 , 547-553.

Duato, J. (1993). A new theory of deadlock-free adaptive routing in wormhole networks. *IEEE Transactions on Parallel and Distributed Systems*, *4*, 1320-1331.

Duato, J. (1994). *A necessary and sufficient condition for deadlock-free adaptive routing in wormhole networks*. Proceedings of International Conference on Parallel Processing, 142-149.

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

Duato, J. (Dec. 1991). *Deadlock-free adaptive routing algorithms for multicomputers: Evaluation of a new algorithm*. In proceedings of the third IEEE symposium. Parallel and distributed processing. (p. 840-847). Dallas, Texas, USA: IEEE Computer Society.

Duato, J., Yalamancili, S., & Ni, L. (1997). *Interconnection networks an engineering approach*. Silver Spring, MD: IEEE Computer Society Press.

Einhoff, G., & Fidler, M. (June 2004). *The turnnet concept: Routing in feed-forward networks with prohibited turns.* In Communications, 2004 IEEE international conference on (Vol. 4, p. 2009-2013).

Fleury, E., & Fraigniaud, P. (1998). *A general theory for deadlock avoidance in wormhole-routed networks*. IEEE Transactions on Parallel and Distributed Systems, 9 (7), 626-638.

Gaughan, P., & Yalamanchili, S. (1995). *A family of fault tolerant routing protocols for direct multiprocessor networks*. IEEE Transactions on Parallel and Distributed Systems, 6 (5), 482-497.

Glass, C., & Ni, L. (1992). *The turn model for adaptive routing*. Proceedings of the 19th Annual International Symposium on Computer Architecture, 278-286.

Glass, C., & Ni, L. (1994). *The turn model for adaptive routing.* Journal of the ACM , 5, 874-902.

Horst, R. (1996). *ServerNet(TM) deadlock avoidance and fractahedral topologies.* Proceedings of IEEE International Parallel Processing Symposium, 274-280.

Hu, J., & Marculescu, R. (2004). *DyAD - Smart routing for networks-on-chip*. Proceedings of the 41st annual conference on Design Automation, 00 , 260-263.

Jayasimha, D. N., Schwiebert, L., Manivannan, D., & May, J. A. (2003). *A foundation for designing deadlock-free routing algorithms in wormhole networks.* Journal of the ACM , 50 (2), 250–275.

Jouraku, A. (2007). *An effective design of deadlock-free routing algorithms based on 2d turn model for irregular networks.* IEEE Transactions on Parallel and Distributed Systems, 18 (3), 320–333.

Klepacki, D. (2003). *Blue gene*. Retrieved from http://lobster.bu.edu/SCV/Archive/IBM/BGL-BU.pdf.

Koibuchi, M., Funahashi, A., Jouraku, A., & Amano, H. (2001, September). *L-turn routing: An adaptive routing in irregular networks.* In Proceedings of IEEE international conference on parallel processing (p. 383-392).

Levitin, L., Karpovsky, M., & Mustafa, M. (May, 2009). *Deadlock prevention by turn prohibitions in interconnection networks*. In 2009 IEEE international symposium on parallel & distributed processing (p. 1-7). Rome, Italy: IEEE Computer Society.

Levitin, L. B., Karpovsky, M. G., Mustafa, M., & Zakrevski, L. (2006). *New algorithm for finding cycle-breaking sets of turns in a graph.* Journal of Graph Algorithms and Applications (JGAA), 387-420.

Libeskind-Hadas, R., Mazzoni, D., & Rajagopalan, R. (1998). *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, 244-249.

Lysne, O., Skeie, T., Reinemo, S.A., & Theiss, I. (2006). *Layered routing in irregular networks*. IEEE Transactions on Parallel and Distributed Systems, 51-56.

Mejia, A., Flitch, J., Duato, J., Reinemo, S. A., & Skeie, T. (2006). *Segment-based routing: An efficient fault-tolerant routing algorithm for meshes and tori.* In Parallel and distributed processing symposium, 2006. IPDPS 2006. 20th international. IEEE.

Mello, A. V. de, Ost, L. C., Moraes, F. G., & Calazans, N. L. V. (2004, May). *Evaluation of routing algorithms on mesh based NoCs (Tech. Rep. No. 040)*. PUCRS, Brazil: FACULDADE DE INFORMATICA PUCRS-Brazil.

Min, G., Ould-Khaoua, M., Kouvatsos, D., & Awan, I. (2004, May). *A queuing model of dimension-ordered routing under self-similar traffic loads.* In Parallel and distributed processing symposium, 2004. proceedings. 18th international (p. 601-613). Washington, DC: IEEE Computer Society.

Mustafa, M., Karpovsky, M., & Levitin, L. (2005, August). *Cycle breaking in wormhole routed computer communication networks.* In Opnetwork 2005. OPNET Technologies.

Mustafa, M., Levitin, L., & Karpovsky, M. (August 2006). *Weighted turn prohibition in* *computer communication networks.* In OpnetWork2006. Washington, DC.: OPNET Technologies, Inc.

Ni, L. M., & McKinley, P. K. (1993). *A survey of wormhole routing techniques in directed networks.* Computer, 26, 62-76.

Parhami, B. (1998). *Introduction to parallel processing: Algorithms and architectures*. Plenum Press.

Pellegrini, F. D., Starobinski, D., Karpovsky, M., & Levitin, L. (2004). *Scalable cycle-breaking algorithms for gigabit Ethernet backbones*. Proceedings of IEEE Infocom 2004.

Pellegrini, F. D., Starobinski, D., Karpovsky, M. G., & Levitin, L. B. (2006, February). *Scalable, distributed cycle-breaking algorithms for gigabit Ethernet backbones*. Journal of Optical Networks, 5 (2), 122-144.

Pifarré, G. D., Gravano, L., Denicolay, G., & Sanz, J. L. C. (1994). *Adaptive deadlock-and livelock-free routing in the hypercube network*. IEEE Transactions of Parallel and Distributed Systems, 5 (11), 1121–1139.

Sancho, J., & Robles, A. (2000, August). *Improving the Up*/Down* routing scheme for networks of workstations.* In Proceedings euro-par 2000.

Sancho, J., Robles, A., & Duato, J. (2000). *A flexible routing scheme for networks of workstations.* In ISHPC (p. 260-267).

Sancho, J. C., Robles, A., & Duato, J. (2004, August). *An effective methodology to improve the performance of the Up*/Down* routing algorithm.* IEEE Transactions of Parallel and Distributed Systems, 15 (8), 740-754.

Sancho, J. C., Robles, A., Lopez, P., Flich, J., & Duato, J. (2003). *Routing in infiniband™ torus network topologies*. Parallel Processing, International Conference on, 0 , 509.

Schroeder, M. D., Birrell, A. D., Burrows, M., Murray, H., Needham, R. M., Rodeheer, T. L., et al. (1990, April). *AutoNet: A high-speed self configuring local area network using point-to-point links* (Tech. Rep. No. SRC Research Report 59). Digital Equipment Corporation, SRC.

Schwiebert, L. (2001). *Deadlock-free oblivious wormhole routing with cyclic dependencies.* IEEE Transactions on Computers, 50 (9), 865-876.

Schwiebert, L., & Jayasimha, D. N. (1996). *A necessary and sufficient condition for deadlock-free wormhole routing*. Journal of Parallel and Distributed Computing, 32 (1), 103–117.

Seitz, C. L. (1985, March). *The hypercube communication chip* (Display File 5182:DF:85). California Institute of Technology.

Shevtekar, A., & Zakrevski, L. (2004). *Hybrid turn-prohibition routing algorithm for the networks of workstations*. Parallel and Distributed Processing Techniques and Applications.

Silla, F., & Duato, J. (2000). *High-performance routing in networks of workstations with irregular topology*. IEEE Transactions on Parallel and Distributed Systems, 11 (7), 699-719.

Silla, F., Duato, J., Sivasubramaniam, A., & Das, C. R. (1998). *Virtual channel multiplexing in networks of workstations with irregular topology*. Proceedings of the International Conference on High Performance Computing, 147-154.

Skeie, T., Lysne, O., & Theiss, I. (2002). *Layered shortest path (LASH) routing in irregular system area networks*. In International parallel and distributed processing symposium: IPDPS 2002 workshops (p. 0162-0170). Fort Lauderdale, FL.

Starobinski, D., Karpovsky, M., & Zakrevski, L. (2003). *Application of network calculus* *to general topologies using turn prohibition.* IEEE/ACM Transactions on Networking, 11 (3), 411-421.

Sun, Y.-M., Yang, C.-H., Chung, Y.-C., & Huang, T.-Y. (2004). *An efficient deadlock-free tree-based routing algorithm for irregular wormhole-routed networks based on the turn model.* In Icpp '04: Proceedings of the 2004 international conference on parallel processing (pp. 343–352). Washington, DC, USA: IEEE Computer Society.

Zakrevski, L. (2000). *Fault-tolerant wormhole message routing in computer communication networks*. Doctoral dissertation, Boston University, College of Electrical Engineering.

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

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

Zakrevski, L., Mustafa, M., & Karpovsky, M. (2000). *Turn prohibition based routing in irregular computer networks*. Proceedings of the IASTED International Conference on Parallel and Distributed Computing and Systems, 175-179.

Zhou, J., Lin, X.-Y., & Chung, Y.-C. (2006). *A tree-turn model for irregular networks.* In Nca '06: Proceedings of the fifth IEEE international symposium on network computing and applications (pp. 11–18). Washington, DC, USA: IEEE Computer Society.