The Generalized Turn Prohibition Model for Multicast Routing in Irregular Networks

M.Karpovsky[1], L.Zakrevski[2], M.Mustafa[1], A.Agarwal[1]

[1] - Boston University, ECE Department, [2] – New Jersey Institute of Technology, ECE Department

ABSTRACT

In this paper a universal model for breaking cycles is described. A method applicable to nondirected graphs is first developed and subsequently applied to communication networks. Loops or cycles are particularly important in communication networks. Specifically, communication networks using wormhole routing techniques [1,2,3] and local area networks with backpressure flow control mechanisms [4, 5, 6] are particularly vulnerable to network states known as deadlock and livelock. An efficient approach to handle deadlocks and livelocks in such networks is based on preventing them by selecting routing policies that are deadlock and livelock free. Cycles in channel dependency graph [7] in such networks have been used as an indicator for vulnerability for deadlocks. Construction of the channel dependency graph is at best difficult for all but very small networks. Our model provides a simpler alternative. It operates in the network graph domain instead of the channel dependency graph. We present a simple routing strategy for deadlock and livelock free unicasting and multicasting by using minimal number of turn prohibitions to break all cycles in the network graph. Performance of the model is evaluated by simulation experiments for irregular communication networks using unicast and multicast wormhole routing.

KEY WORDS:- Turn Model, Turn Prohibition, Wormhole Routing, Unicasting, Multicast Routing

1. INTRODUCTION

Existence of loops or cycles in nondirected graphs that represent network topologies could lead to problems in the underlying network. Interconnection networks have been relying on routing algorithms that restrict the use of channel resources by packets or worms in order to avoid deadlocks [3,7,8]. Deadlock is a network state in which packets or worms hold channel resources while waiting for other channel resources indefinitely. Duato [7] has shown that presence of cycles in the channel dependency graph (CDG) leads to deadlocks. Therefore, to prevent deadlocks all cycles in the CDG must be broken. Since CDG depends not only on the topology of the original network but also on selected routing protocol, CDG is a much more complex directed graph with considerably more vertices and edges than the original graph representing network's topology. With the generalized turn model (GTM), all simple and compound cycles are broken in the original graph using turn prohibitions. In this paper we first introduce the GTM and then study its performance.

Several routing methods currently exist for regular topologies, such as 2-D meshes, tori and hypercubes [1, 2, 9, 20, 21, 22]. For irregular topologies most of the existing routing strategies are based on spanning trees and (up/down) routing [11, 12]. According to this strategy, once a spanning tree is constructed, any two nodes can communicate with each other along the tree without any deadlocks. Main drawbacks of this approach are the long message paths and high load on the edges near the root node. This method can be improved by allowing shortcuts using edges not belonging to the spanning tree, but this could result in deadlocks due to the formation of cycles in the channel dependency graph [11].

For irregular topologies it was shown in [15, 19] that reduction in the number of prohibited turns results in a decrease of average path lengths of messages and in a reduction of average delivery time accompanied by an increase in throughput.

We note that a set of prohibited turns for deadlock prevention does not completely specify the routing strategy, i.e. several routing strategies can satisfy the same set of restrictions on turns in the network graph. In [14], a decentralized algorithm for constructing local routing tables based on the set $Z(\boldsymbol{G})$. This algorithm minimizes average message path lengths and thus average delivery time. Our goal therefore is to select the shortest routing path $\boldsymbol{a}_1, \boldsymbol{a}_2 \ldots \boldsymbol{a}_m$ ($\boldsymbol{a}_1=\boldsymbol{s}$, $\boldsymbol{a}_m=\boldsymbol{d}$) among all paths, satisfying the routing restrictions imposed by the set $Z(\boldsymbol{G})$ for any source $\boldsymbol{s}$ and destination $\boldsymbol{d}$. We assume that these computations are infrequent as they are performed only when network topology changes are detected which we assume do not occur often.

In both regular and irregular topologies, routing algorithms prevent deadlocks or cycles from being formed by restricting the use of network resources. In 2D meshes, simple cycles are broken by using for example *xy* routing by permitting turns coming in from positive or negative *x* direction, out onto positive or negative *y* directions, and forbidding all other 90 degree turns [8,9]. In irregular topologies, cycles and deadlocks are prevented by restricting the use of edges that are not on a spanning tree [10-13] by a technique called up/down. In this technique vertices are labeled in a partial order determined by a selected spanning tree and edges are classified into up or down directed tree edges and up or down directed cross edges types for those edges that are not on the spanning tree. The technique prevents cycles by preventing a down edge followed by an up edge, known as Down-Up turn. In this technique permitted turns are those corresponding to “zero or more up edges followed by zero or more down edges”. With this approach fraction of turns (pairs of incoming and outgoing channels for routers) to be prohibited at the routers depends on the selected spanning tree and in the worst case (see below) can be close to 1.

With GTM the number of prohibited turns is minimized while breaking all cycles in the network graph. In doing so, GTM guarantees that no more than 1/3 of all possible turns will be prohibited. This upper bound is attained only in topologies with complete graphs, K_n [14]. Furthermore, the set of prohibited turns is irreducible in the sense that deletion of any one of the turns from this set will introduce a cycle in the graph.

Construction of the set of prohibited turns constitutes the first of the two stages of the routing process. At this stage we are trying to construct a minimal set of prohibited turns since small reduction in a number of prohibited turns may result in a drastic decrease in the average delivery time for messages in the network [15, 16]. At the second stage distance based routing tables are constructed that conform to turn prohibitions selected during the first stage. Routing tables constructed this way provide optimal or near optimal, i.e. shortest path between any source and destination in the graph. Each node in the network has its own routing table that excludes the turns that are prohibited at this node [16]. Significant gains have been observed in our experiments based on two virtual channel adaptive routing [17]. In the rest of the paper we cover theoretical aspects of the GTM in Section 2. In Section 3 we illustrate the performance of the algorithm for interconnection networks with deterministic wormhole routing for unicast messages in irregular networks. In Section 4 multicast routing is studied in irregular networks.

2. DEFINITIONS AND MATHEMATICAL MODEL

We assume that G is a connected multigraph of N vertices and E edges where, we denote the graph as $G(V,C)$, a channel as $c_i = (\boldsymbol{v_j},\boldsymbol{v_k})$, and the set of edges or channels as $C=\{c_1,c_2,\ldots,c_E\}$. Similarly the set of vertices is denoted by $V=\{v_1,v_2,\ldots,v_N\}$. The notation $c_i = (\boldsymbol{v_j},\boldsymbol{v_k})$ denotes an undirected edge c_i between vertices $\boldsymbol{v_j}$ and $\boldsymbol{v_k}$. Edges and vertices are either ordinary or special. When a vertex $\boldsymbol{v_i}$ is deleted from the graph G, the vertex $\boldsymbol{v_i}$ and all edges incident on it are also deleted from the graph. This may result in a disconnected graph with components $G_1,G_2,\ldots,G_l$ such that $G - \boldsymbol{v_i} = \cup_j G_j$.

Definition 1. *When a vertex $\boldsymbol{v_i}$ is deleted from a graph G resulting in components $G_1,G_2,\ldots,G_l$, then one of the edges connecting $\boldsymbol{v_i}$ to component G_j , $j=2,3,\ldots,l$ in the original graph G is defined as a special edge.*

Definition 2. *Assume a vertex $\boldsymbol{v_i}$ is deleted from a graph G that results in components G_1 and G_i for $i=2,3,\ldots,l$. If there exists a closed walk $\boldsymbol{v_i,x_1,x_2,\ldots,x_m,v_i}$, such that $\boldsymbol{x_1,x_2,\ldots,x_m} \in G_{i-1}$ then a vertex in G_i connected to $\boldsymbol{v_i}$ in the original graph G with a special edge is defined as a special vertex.*

Thus for a vertex to be special it must be connected to the vertex that is just being deleted by a special edge in the original graph. As will be seen soon, special vertices are privileged vertices that fewer turn restrictions are imposed on them. Vertices that are not special are ordinary vertices.

Definition 3. *A turn is a 3-tuple of vertices ($\boldsymbol{a, b, c}$) where $\boldsymbol{a, b, c} \in V$ and ($\boldsymbol{a, b}$), ($\boldsymbol{b, c}$) $\in C$ are edges incident on the middle vertex $\boldsymbol{b}$.*

For our analysis we consider symmetric graphs where ($\boldsymbol{a,b}$), ($\boldsymbol{b,c}$) $\in C$. Furthermore, we assume that if a turn ($\boldsymbol{a,b,c}$) is prohibited then so is the turn ($\boldsymbol{c,b,a}$). Because of this symmetry, when enumerating them, turns ($\boldsymbol{a,b,c}$) and ($\boldsymbol{c,b,a}$) would be counted as one. For a vertex $\boldsymbol{v_i}$ of degree d_i, number of symmetric turns involving $\boldsymbol{v_i}$ is given by

$$T_i = d_i(d_i-1)/2.$$

Similarly for graph G, total number of symmetric turns is

$$T = \sum T_i = ½ \sum d_i(d_i-1)$$

where the summation is over all vertices.

To study the effectiveness of GTM for graph G, we compute the ratio $z(G) = |Z(G)| / |T(G)|$ as the number of prohibited turns to the total number of turns. We use the following recursive algorithm to construct a set of prohibited turns $Z(G)$ in an arbitrary graph G. For explanation of the $TP(G)$ algorithm please refer to the pseudo code listing below.

TP procedure has been described in [14]. This is a recursive procedure, at each step one node is selected by a special rule and all turns, including this node, are either prohibited, or permitted. Some nodes are labeled special, and cannot be selected. After deleting a node and all adjacent links, *TP* algorithm is recursively used for each of components of original graph.

Following theorems can be proven for the *TP* and for the set *Z* of prohibited turns generated by the *TP* algorithm [14].

Theorem 1. Any cycle in the original graph $\boldsymbol{G}$ has at least one turn included in Z.

Theorem 2. Graph remains connected after turn prohibition using *TP*, i.e. for any two vertices there exists a path connecting them that does not contain prohibited turns.

Theorem 3. The set of prohibited turns Z generated by *TP* is irreducible for any topology.

Theorem 4. *TP* procedure prohibits at most 1/3 of all turns in the graph.

Theorem 5. If a graph $\boldsymbol{G}$ has N vertices, M edges and $T(\boldsymbol{G})$ turns then fraction of turns that are prohibited $z(\mathbf{G})$ is lower bounded by $z(\mathbf{G}) \geq (N - M + 1) / T$.

Theorem 1 guarantees that all cycles will be broken and Theorem 2 assures that this is done without loss of connectivity of the graph. With Theorem 3 we are assured that the set Z of prohibited turns contains no redundant elements. An irreducible set Z implies that if any one of the turns in the set is deleted a cycle will be introduced into the graph. Theorem 4 provides an upper bound on the fraction of turns that are prohibited in any graph and the last theorem provides a lower bound on number of prohibited turns. We note that the complexity of the *TP* algorithm is of the order of $O(N^2d)$, where N is the number of vertices in $\boldsymbol{G}$ and d is the maximal degree of the vertices. It is also worth mention that the *TP* algorithm does not guarantee that set Z of prohibited turns is unique or minimal.

Example of using TP algorithm is shown on Fig.1. Nodes are selected according to their numbers. In this graph nodes labeled ***1, 2, 6, 7, 8, 9,*** and ***10*** are all equally likely to be chosen as the first vertex. Here we have chosen ***1*** which when deleted creates two components, G_1 and G_2. Vertices {***8, 9, 10, 11***} belong to G_2 since it has fewer edges connecting to node ***1***. Edge (***1, 11***) is labeled as special and shown with heavy line. After node ***1*** is deleted *TP* is now invoked with G_1 that has vertices {***2, 3, 4, 5, 6, 7***}. In this set only node 2 is of minimal degree that is subsequently selected and all turns about it are forbidden. Similarly node ***3, 4,*** and ***5*** are chosen and turns about them are forbidden as shown. We note that *TP* algorithm did not prohibit all turns about node ***1*** since doing that would have violated Theorem **2.** After *TP* is applied to G_1 as shown, node ***11*** is labeled as *special*. All nodes are of minimal degree three but the algorithm would not select node 11 since it is a special node. Then node ***8*** is selected followed by nodes ***9*** and ***10.*** In this example fraction of prohibited turns is z = 12/49 which is less than 1/3.

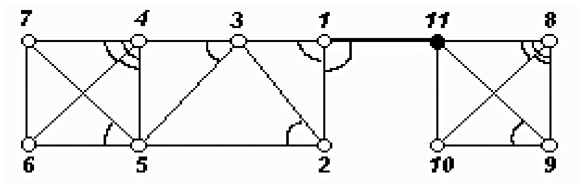


Fig. 1 An Example Graph Showing a Special Edge and a Special Vertex

Although in many cases the *TP* procedure yields a minimal number of prohibited turns this cannot always be guaranteed. For example if the sequence of selections made by the *TP* procedure for the graph in Fig. 1 is (***7, 6, 4, 5, 3, 2, 1, 8, 9, 10, 11***) the number of prohibited turns becomes 11 instead of 12. The lower bound on the fraction of prohibited turns for Fig. 1 is 9.

3. PERFORMANCE OF GTM FOR UNICAST ROUTING

In this section we study the performance of GTM by comparing it with the up/down approach for randomly generated connected irregular graphs. To illustrate the efficiency of GTM we show in Fig. 2 percentages of prohibited turns generated for GTM and up/down approaches for graphs with 256 vertices. For each average degree, one thousand random graphs are generated and the average of the fraction of prohibited turns calculated and plotted as shown. It can be seen from Fig. 2 that moving from the up/down curve to the *TP* algorithm curve results in a 15% to 50% reduction of the fraction of prohibited turns.

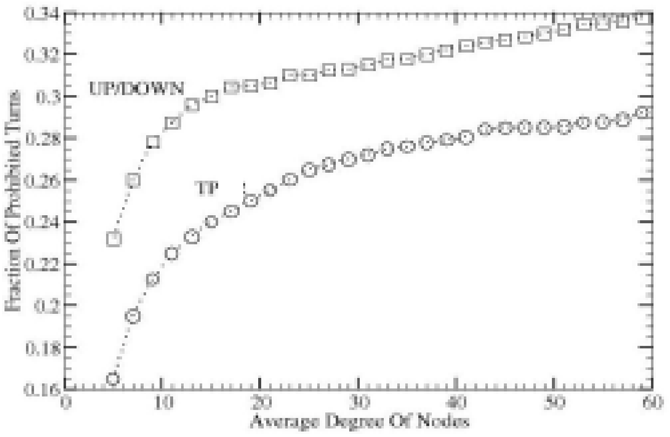


Fig. 2 Fraction of Prohibited Turns in Up/Down and *TP* Algorithms

In Fig. 3 we show simulation results comparing the average network latency versus message generation rates for *TP* and up/down approaches. For both algorithms we simulated for randomly generated connected graphs with 256 nodes of degree 4. For each graph we first found the Breadth First spanning tree for the graph and labeled it according to up/down algorithm. Subsequently set of turn prohibitions implied by the labeling for the spanning tree is constructed. For each graph, 10,000 worms were generated and forwarded using the appropriate routing tables. Then the average latency is computed for each message generation rate. Similar to [23], we assume that when the header flit of a worm reaches its destination, the worm is consumed in finite time. Results of these experiments shown in Fig. 3 demonstrate the superiority of the GTM algorithm over the up/down approach.

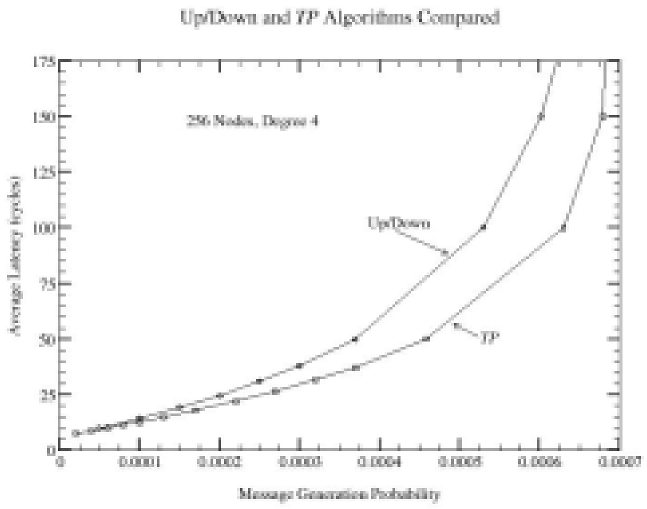


Fig. 3 Average Latencies Compared for Up/Down and *TP* Algorithms.

4. USING OF GTM FOR MULTICAST ROUTING

In this section we extend the *TP* based routing to multicasting. There have been several different approaches to multicasting including trip-based approach [25], U-Mesh algorithm by McKinley et al [26] and modified U-Mesh algorithm [27] in which chain ordering has been considered. These algorithms do not require modifying routers but suffer from repeated re-starts, which are time consuming. In our approach we assume that routers have the capability of selectively replicating worms on several of its output ports if necessary. We do not consider this to be a significant impediment due to the fact that such features could very easily be accommodated by virtue of the larger in-system programmable FPGA chip technologies. Routers with this capability would modify the multi destination header of an incoming worm in such a way that subsets of the destination would be present on outgoing worms on different ports. Our approach to multicasting is based on constructing and using minimum multicast trees. A multicast tree is a spanning tree that covers the source node and all destination nodes and possibly some other nodes in the network graph. Intervening nodes that are not in the destination set but are part of the multicast tree simply forward the incoming worm after replication if necessary. We have used three different approaches in implementing the multicast routing. The simplest approach utilizes only the local information, which we call the *Zero-Algorithm*. Here the multicast tree is constructed by combining the unicast paths for the message. Nodes at which branching is encountered would replicate the message and forward it onto multiple output ports as necessary. For next multicast algorithm, called *One-Algorithm*, source node constructs the multicast tree based on knowledge of not only local information but also information about its immediate neighbors. In this approach the node looks ahead by 1-hop and determines the minimal tree based on this information. Finally in the *Infinity-Algorithm* node does exhaustive search of the best possible tree with global knowledge of all nodes in the network.

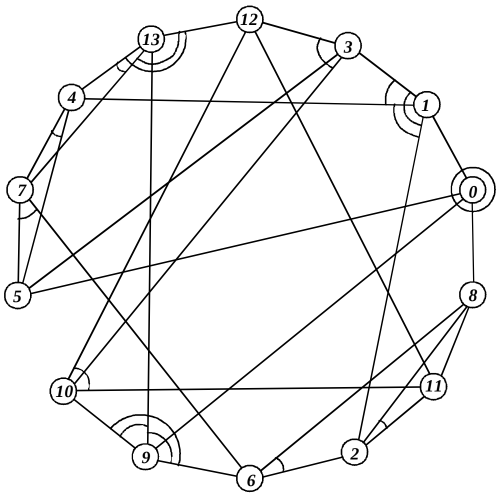
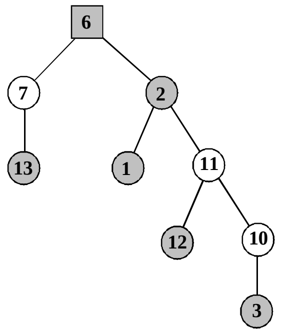


Fig. 4 Graph with 14 nodes of degree 4 showing prohibited turns generated by *TP*. All turns around node 0 are prohibited.

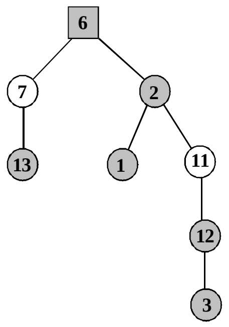
Let us consider the connected network of 14 nodes shown in Fig. 4. After *TP* operation prohibited turns are shown as arcs connecting two edges that represent the turn. For example we used full circle around node ***0*** to indicate that all turns around this node are forbidden. Similarly turn (***8, 2, 11***) along with others as marked are prohibited. In this network node ***6*** is multicasting a message to nodes *D* = {***1, 2, 3, 12, 13***}.

With the *Zero-Algorithm* source node replicates message and routes one message to its output port leading to node ***7*** and the other one to its output port leading to node ***2***. At this point multicast message leading to node ***7*** has only one destination address in it, whereas the message leading to node ***2*** has four destination addresses in it. Node ***2*** subsequently replicates the incoming message into two and forwards one to node ***1*** and the other to node ***11*** and so on. This algorithm yields a tree length of 8 hops.

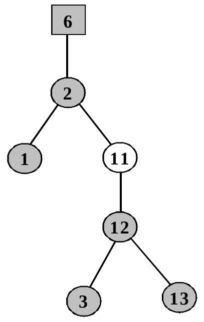
One-Algorithm behaves identical until node ***11*** is reached where worm is not replicated. Incoming worm is sent out to node ***12*** with two destination addresses in the header, ***12*** and ***3***. *One-Algorithm* is best understood with the aid of the covering matrix shown in Table. 1. In this matrix, top row enumerates the nodes in the destination set, and the leftmost vertical column lists the next nodes of the source. Referring to Fig. 5 we see that these are nodes ***2, 7, 8,*** and ***9*** as listed in the table**.** Matrix elements are distances from neighbors of the source node. For example we see that no destination node is reachable from node ***9*** and all entries in the matrix are therefore **X** which stands for infinite distance. A distance value of 0 indicates that next node is also the destination. In this matrix the goal is to cover all destinations with fewest number of output ports, or equivalently next nodes. This algorithm shrinks the tree size by one to yield a tree with 7 edges.



(a) *Zero-Algorithm*



(b) *One-Algorithm*



(c) *Infinity-Algorithm*

Fig. 5 Three Multicast Algorithms Showing Three Tree Lengths.

Nxt Node \ Dst Node	1	2	3	12	13
2	①	⓪	③	②	3
7	2	X	X	X	①
8	2	1	3	2	3
9	X	X	X	X	X

Table 1. Covering Matrix for *One-Algorithm*

Finally the global, *Infinity-Algorithm* improves the results of the *One-Algorithm* further by one edge for the given graph yielding a tree length of 6 edges as shown in Fig. 5.

In previous paper we have described the results of high-level experiments, comparing the size of the constructed transmission tree for different approaches [17]. In particular, we have seen the up/down and *TP*-approaches resulted in similar tree lengths.

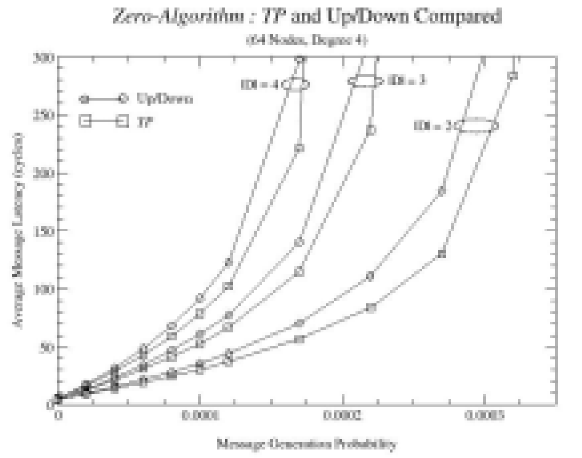


Fig. 6 *Zero-Algorithm* Based Multicast Experimental Results.

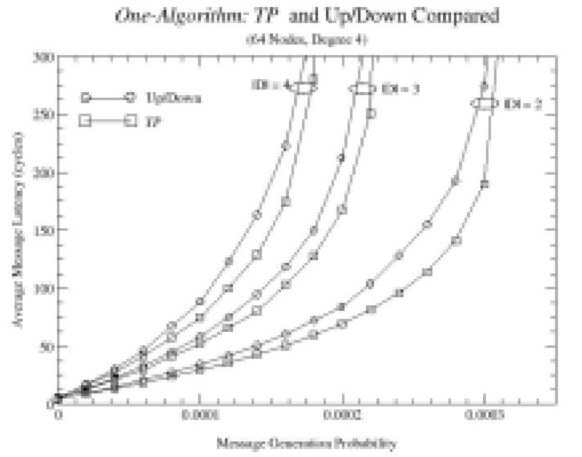


Fig. 7 *One-Algorithm* Based Multicast Experimental Results.

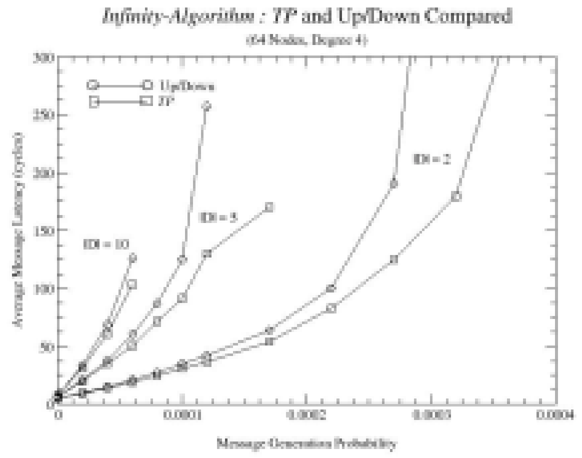


Fig. 8 *Infinity-Algorithm* Based Multicast Experimental Results.

Our low level multicast experimental results are shown next. We have studied multicasting based on up/down and the *TP* algorithm as discussed above for destination sets of size 2, 3, and 4 for randomly generated graphs of 64 nodes. When selecting members of the destination set uniform distribution was used. We run our experiments for 100 different random graphs and averaged the results for the latencies observed. We considered the latency to be the time difference between when the last flit of the worm is received by the last recipient and the time the original worm was created. For each graph we transmitted 10,000 messages before starting to collect the statistics as in all of our experiments. In Fig.6 our results are for *Zero-Algorithm* based trees. In this set of experiments no significant performance gain was observed between the up/down and the *TP* algorithms. In Fig. 7 we show the results of *One-Algorithm* with similar results. Next, in Fig. 8 latency times for multicasting based on the *Infinity-Algorithm* is shown for 2, 5 and 10 destinations. It is evident that *TP* algorithm outperformed the up/down approach.

5. ACKNOWLEDGEMENTS

Authors wish to thank Shameek Gupta and Raman Mathur from Boston University for their help in our simulation experiments. This research was partly supported by International Science and Technology Center (Project B-104-98) and by New Jersey Institute of Technology (SBR grant).

9. REFERENCES:

[1] W. Dally and H. Aoki "Deadlock-Free Adaptive Routing in Multiprocessor Networks Using Virtual Channels," *IEEE Trans. on Parallel and Distributed Systems* vol. 8, pp. 466-475, 1997.

[2] W. Dally and C. Seitz, L. "Deadlock-Free Message Routing in Multiprocessor Interconnection Networks," *IEEE Trans. on Comput.* vol. 36, pp. 547-553, 1987.

[3] E. Fleury and P. Fraigniaud "A General Theory for Deadlock Avoidance in Wormhole-Routed Networks," *IEEE Trans. on PDS* vol. 9, pp. 626-638, 1998.

[4] Gellernter "A DAG-based Algorithm for Prevention of Store and Forward Deadlock in Packets Networks," *IEEE Trans. on Comput.* vol. C-30, pp. 709-715, 1981.

[5] K. Gunther, D. "Prevention of Deadlocks in Packet-switched Data Transport Systems," *IEEE Trans. Commun.* vol. COM-29, pp. 512-524, 1981.

[6] M. Karol, J. Golestani and D. Lee "Prevention of Deadlocks and Livelocks in Lossless, Backpressured Packet Networks," *INFOCOM* vol. 3, pp. 1333-3342, 2000.

[7] J. Duato "A New Theory of Deadlock-Free Adaptive Routing in Wormhole Networks," *IEEE Trans. on PDS*, vol. 4, pp. 1320-1331, 1993.

[8] L. Ni, M. and P. McKinley, K. "A Survey of Wormhole Routing Techniques in Directed Networks," *Computer* vol. 26, pp. 62-76, 1993.

[9] C. Glass and L. Ni "The Turn Model for Adaptive Routing," *Journal of ACM* vol. 5, pp. 874-902, 1994.

[10] R. Libeskind-Hadas, Mazzoni and R. "Optimal Contention-Free Unicast based Multicasting in Switch-Based Networks of Workstations," *Proc. of the First Merged Int. Parallel Processing Symposium and Symposium on Parallel and Distributed Processing* pp. 358-364, 1998.

[11] R. Libeskind-Hadas, D. Mazzoni and R. ajagopalan "Tree-Base Multicast Routing in the Mesh with No Virtual Channels," *Proc. of the First Merged Int. Parallel Processing Symposium and Symposium on Parallel and Distributed Processing* pp. 244-249, 1998.

[12] M. Schroeder and t. al. "Autonet: A High-Speed self configuring Local Area Network Using Point-to-point Links," 1990.

[13] R. Sivaram, D. Panda and C. Stunkel "Multicasting in Irregular Networks With Cut-Through Switches Using Tree-Based Multidestination Worms," *Proc. of the 2nd Parallel Computing, Routing and Communication Workshop*, 1997.

[14] L. Zakrevski "Fault-Tolerant Wormhole Message Routing in Computer Communication Networks," *PhD Thesis, College of Engineering* pp. 21-27, 2000.

[15] L. Zakrevski and M. Karpovsky, G. "Fault-Tolerant Message Routing in Computer Networks," *Proc. of Int. Conf. on PDPA-99* pp. 2279-2287, 1999.

[16] L. Zakrevski, S. Jaiswal and M. Karpovsky "Unicast Message Routing in Communication Networks With Irregular Topologies," *Proc. of CAD-99* 1999.

[17] L. Zakrevski, M. Mustafa and M. Karpovsky "Turn Prohibition Based Routing in Irregular Computer Networks," *Proc. of the PDCS-2000* pp. 175-179, 2000.

[18] N. Boden and e. al. "Myrinet: A Gigabit per second Local Area Network," *IEEE Micro* pp. 29-35, 1995.

[19] L. Zakrevski, S. Jaiswal, L. Levitin and M. Karpovsky "A New Method for Deadlock Elimination in Computer Networks With Irregular Toplologies," *Pro. of the IASTED Conf. PDCS-99* vol. 1, pp. 396-402, 1999.

[20] J. Duato, S. Yalamanchili and L. Ni, M. "Interconnection Networks: An Engineering Approach," 1997.

[21] R. Boppana, V. and S. Chalasani "Fault-Tolerant Wormhole Routing Algorithms in Mesh Networks," *IEEE Trans. on Comput.* vol. 44, pp. 848-864, 1995.

[22] C. Glass and Ni,L. "Fault-Tolerant Wormhole Routing in Meshes," *Proc. of Int. Symp. on Fault-Tolerant Computing* 1993.

[23] J. Ngai, Y. and C. Seitz, L. "A Framework for Adaptive Routing in Multicomputer Networks," *Proc. First Symp. Parallel Algorithms and Architectures* pp. 1-9, 1989.

[24] R. Boppana, V., S. Chalasani and C. Raghavendra, S. "Resource Deadlocks and Performance of Wormhole Multicast Routing Algorithms," *IEEE Transactions on PDS* vol. 9, no. 6, pp. 535-549, 1998.

[25] Y.-C. Tseng, D. Panda, K. and T. Lai, H. "A Trip-Based Multicasting Model in Wormhole-Routed Networks With Virtual Channels," *IEEE Trans. on PDS* vol. 7, no. 2, pp. 138-150, 1996.

[26] P. McKinley, K., H. Xu, A. Esfahanian, H. and L. Ni, M. "Unicast-Based Multicast Communication in Wormhole-Routed Direct Networks," *IEEE Transactions on PDS* vol. 5, no. 12, pp. 1254-1265, 1994.

[27] R. Kesavan, K. Bodalapati and D. Panda, K. "Multicast on Irregula Switch-based Networks with Wormhole Routing," *Third Int. Symp. on High Performance Computer Architecture* no. Feb, pp. 48-57, 1997.