Proceedings of the IASTED International Conference on Parallel and Distributed Computing and Systems, November 3-6, 1999, Cambridge, USA

A New Method for Deadlock Elimination in Computer Networks with Irregular Topologies

Lev Zakrevski, Sharad Jaiswal, Lev Levitin, Mark Karpovsky*

Research Lab. on Reliable Computing,
Boston University, Department of Computer Engineering,
8 Saint Mary's St., Boston, MA 02215, USA
zakr@bu.edu

Abstract In this paper we consider the problem of deadlock-free wormhole unicast routing in networks with irregular topologies. To prevent deadlocks, for each router some input/output pairs (turns) have to be prohibited. We analyze the problem of minimizing the number of prohibited turns for providing deadlock-free routing. A new algorithm is proposed, which guarantees that this number does not exceed 1/3 of the total number of turns and that it is still possible to send messages between any two initially connected nodes. To the best of our knowledge this is the first meaningful upper bound on the fraction of turns to be prohibited to prevent deadlocks for networks with an irregular toplogy. Also, the problem of routing in the presence of prohibited turns is considered and appropriate methods developed.

1. Introduction

We are considering computer/ communication networks, consisting of computing nodes (which do not have to be identical) connected by bi-directional links. Each node can be viewed as the combination of a router and a processor with some RAM, bus and I/O circuitry. In this paper we concentrate on networks with arbitrary (irregular) topologies, such as networks of workstations (NOWs).

Many existing methods for message transmission utilize *wormhole routing*, where each message (or each packet of the message) is divided into flits, all of which follow the same path [21]. In wormhole routing, a router begins forwarding a packet as soon as the header is received and the required channel buffer in the next router can accept one or more flits of the packet. Flits are transmitted from one router to the next in a pipelined fashion and may occupy several channels along the path from source to destination. Only the header flit of a packet contains information required for routing. If the header flit is blocked because the required buffer in the next router along it's path is full, all of the flits in the packet are blocked, and, therefore, so are the channels that they occupy. If more than one flit can be buffered at a node, flits behind the header can "catch up" until the available buffer space is filled. At that point, they block and can continue only after the header is unblocked.

Wormhole routing is efficient because it allows low channel-setup time, low-latency communications and reduced communication overhead [2,21]. It has been adopted in almost all existing direct networks [5]. Gradually, variations of this technique, known as cut-through routing [16], are being used in commercial NOW implementations like Myrinet [1,14]. However, wormhole routing is very susceptible to *deadlocks* because packets are allowed to hold many resources while requesting others. Recently, NOWs have emerged as an inexpensive alternative to massively parallel multiprocessors. In order to minimize network latency and achieve high bandwidth communications, recent experimental and commercial switches for NOWs implement wormhole routing [1,18,22]. A flexible router architecture that implements a variety of routing and switching schemes by dedicating a microprogrammable routing engine to each incoming link was presented in [15]. Design of efficient deadlock-free routing algorithms in irregular topologies introduces new challenges, which we are going to address in this paper.

One way to solve the deadlock problem is to allow the preemption of packets [21]. However, because of requirements for low latency and reliability, packet preemption is not used in most direct network architectures and deadlocks are avoided by the routing algorithms. To avoid deadlocks, the routing algorithm itself must be shown to be deadlock-free.

Overall, wormhole routing strategies can be divided into adaptive and non-adaptive techniques [7,8,9,10,21,23]. For non-adaptive routing the route followed by a packet is completely defined by the source and destination addresses. For adaptive routing the route depends also on dynamic network conditions, such as the presence of faulty or congested channels. In this paper we consider mostly non-adaptive methods. However, the proposed methods are extendable for adaptive routing as well. For non-adaptive methods, deadlocks are absent if and only if there are no cycles in the channel dependency graph [9,10].

Several routing methods currently exist for routing in regular topologies, such as 2-dimensional and multidimensional meshes and tori and hypercubes [4,6,24]. In addition, approaches to the more difficult problem of routing in the presence of faults (fault-tolerant routing) also have been developed for some of these topologies for the cases when the number of faults is relatively small compared to the number of nodes [6,24]. Very few papers have been published on wormhole routing for irregular networks such as NOWs [20]. Some

* This work has been supported by NSF grant MIP-9630096

of these algorithms require complex signaling hardware at the routers. Even for regular topologies finding a good trade-off between communication performances and hardware costs is an open problem [10]. In this paper we take on these challenges.

For the case when the number of faults is large, fault-tolerant routing becomes very similar to routing in an irregular topology. Efficient and scalable fault-tolerant routing techniques for meshes and tori developed by the authors can be find in [24,25]. If the network graph can be embedded in a mesh (or hypercube) and the difference between the number of nodes in the mesh and the number of nodes in the original network is small, then this embedding with the corresponding fault-tolerant routing can be used for routing in the original network.

For the approaches that we will outline in this paper, packets will be delivered without deadlocks for all sources and destinations.

For the case of a general topology, several routing strategies based on the spanning tree approach have been developed [19,22]. In this case, a spanning tree must be first constructed. Any two nodes can communicate with each other along the tree without any deadlocks. The main drawbacks of these approaches are the long packet paths and high loads for the edges near the root node. The method can be improved by allowing shortcuts using edges, not belonging to the spanning tree, but the above drawbacks cannot be eliminated. The problem of finding deadlock-free tree-based wormhole algorithms has been widely regarded in the literature as a difficult one. We will outline our approach for solution of this problem in the next sections.

2. Mathematical model

We assume that the original network consists of N nodes, connected by E edges. Also we assume that all nodes are connected (for any two nodes there exists a path between them). In the general case network graph G can be considered as a multigraph with several edges between two nodes. In particular, if V virtual networks [9,10,11] are used, any two nodes are connected either by 0, or by V edges. (Each physical channel is split into V virtual channels, using time multiplexing. Each virtual channel has its own buffer. Sum of the capacities of these virtual channels is restricted by the capacity of the original physical channel, so large V will lead to performance degradation. The number of virtual channels can vary for different links).

We will consider in this paper mostly the case of deterministic (non-adaptive) *unicast* routing. A routing strategy will then be a function on the set of pairs ($\boldsymbol{s}$, $\boldsymbol{d}$), where $\boldsymbol{s}$ and $\boldsymbol{d}$ are source and destination nodes. For each such pair the value of this routing function will be either 0 (packet will not be transmitted), or a vector of edges representing a path from $\boldsymbol{s}$ to $\boldsymbol{d}$. The set of all path vectors will be denoted by $\boldsymbol{P}$.

The condition for deadlock elimination can be checked by analyzing P. Based on P, the channel dependency graph can be constructed, nodes of this graph correspond to edges in G and edges in the channel dependency graph correspond to "turns" in G. For the deterministic case there must be no cycles in the channel dependency graph to prevent deadlocks [9,10].

For every routing strategy some of the turns in network graph G are prohibited. The turn in G is a 3-tuple of nodes ($\boldsymbol{a},\boldsymbol{b},\boldsymbol{c}$) such that ($\boldsymbol{a},\boldsymbol{b}$) and ($\boldsymbol{b},\boldsymbol{c}$) are edges of G, $\boldsymbol{a}\neq\boldsymbol{c}$. (Nodes of G are labeled by $\boldsymbol{1,2,\ldots,N}$, and we do not distinguish between a node and its label). In order to correctly model existing switch-based networks as the DEC Autonet [22] and Myrinet [1] we assume that G is symmetric, i.e. if ($\boldsymbol{a,b}$) is an edge in G , then $\boldsymbol{(b,a)}$ is also an edge. In Autonet and Myrinet these symmetrical channels can be used simultaneously without contention. We assume that if ($\boldsymbol{a},\boldsymbol{b},\boldsymbol{c}$) is prohibited, then ($\boldsymbol{c},\boldsymbol{b},\boldsymbol{a}$) is also prohibited, and we will consider these two turns as one. The total number of turns in $T=\Sigma(d_i\ (d_i+1))/2$, where d_i is a degree (number of neighbors) of node $\boldsymbol{i}$. For example, for the up/down routing a spanning tree for G is constructed, nodes are labeled preserving the partial order defined by the tree (the root has label $\boldsymbol{1}$) and turn ($\boldsymbol{a},\boldsymbol{b},\boldsymbol{c}$) is prohibited if $\boldsymbol{b}>\boldsymbol{a}$ and $\boldsymbol{b}>\boldsymbol{c}$.

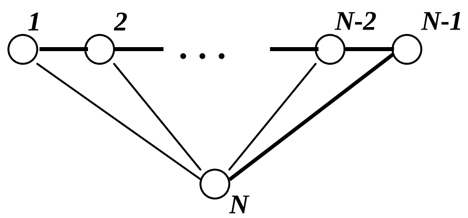

Fig.1. *Example of a network with z»1-6/N for the spanning tree (up/down) approach and z»2/N for the proposed approach (the edges of the selected spanning tree are shown in bold).*

As it is illustrated in Fig.1, reduction in a number Z of prohibited for deadlock elimination turns results in a decrease of average path length t of packets. (If turns ($\boldsymbol{i,N,j}$) ($\boldsymbol{i,j}$=$\boldsymbol{1,2,\ldots,N-1}$) are prohibited then $Z=N\ (N-1)/2$ and $t\approx N/2$, and if turns $\boldsymbol{(i+1,\ i,\ N)}$ ($\boldsymbol{i=1,2,\ldots,N-2}$) are prohibited, then $Z=N-1$ and $t\approx2$). We note that for another selection of the spanning tree Z can be drastically reduced for this example, but the problem of construction of an optimal tree is NP-hard and in the general case even for the best selection of a spanning tree and the corresponding labeling of nodes Z can be rather large.

This reduction in the average path length results in a reduction of average delivery time and increase in throughput. For example, for 2-d meshes and (x-y) routing $z=Z/T=1/3$ and for North-Last routing [15] $z=1/6$ and North-Last has better performance in terms of delivery time and throughput. In [11,12] Glass and Ni investigated in depth the turn model, which prohibits some turns in the multidimensional meshes to break all the possible deadlocks. (In their works only 90-degree turns have been considered which is sufficient for meshes). Our simulation results for random graphs with N=256 nodes also illustrate a strong correlation between fractions of prohibited turns and average delivery time for networks with irregular topologies.

In this paper we will use the turn model for irregular networks. Fractions $z=Z/T$ of prohibited turns will be used as one of the criteria of efficiency of a routing strategy.

We note that a set of prohibited turns preventing deadlocks does not specify completely the routing strategy, i.e. several routing strategies can satisfy the same set of restrictions on turns in the network graph.

In Section 3 we describe techniques for constructing minimal sets of prohibited turns to prevent deadlocks. Section 4 is devoted to routing strategies satisfying selected sets of prohibited turns and minimizing average path lengths and average delivery time for given restrictions on local memories in routers.

As far as it is known to us, for all known routing strategies for irregular topologies z may be close to 1 for some network graphs. For example, for routing with up/down restrictions z depends on a selection of the spanning tree for the network graph and in some cases (as it is shown by Fig.1) may be close to 1. We are not aware of any technique for construction of spanning trees, which will guarantee a meaningful upper bound on fraction z of prohibited turns.

In the next section we will outline our approach for unicast non-adaptive deadlock-free wormhole routing with a fraction of prohibited turns not exceeding 1/3. (For some topologies this upper bound cannot be improved). We will describe an approach for construction of the corresponding spanning tree and restrictions on turns defined by this tree. We will denote the algorithm implementing this approach as "**z-algorithm**". To the best of our knowledge, *z*-algorithm is the first algorithm providing a meaningful upper bound on a fraction of turns in network, which have to be prohibited to prevent deadlocks.

In Section 4 we will outline local, global and mixed (hierarchical) approaches for routing satisfying restrictions on turns generated by *z*-algorithm. The goal of these approaches is to minimize average delivery time and to provide for a good trade-off between average delivery time and sizes of local memories in routers. We will also describe in this section local, mixed and global approaches for the case of routing with two virtual networks combining advantages of *z*-algorithm and the up/down approach.

The complexities of the algorithms described in Sections 3 and 4 do not exceed $O(N^3)$ (N is a number of nodes). These algorithms have to be implemented *only when there is a change in the topology* of the original network (faults are detected in some nodes, new users join the system, etc.)

3. Deadlock elimination by turn prohibition

In this section we will describe our results on lower and upper bounds on fractions $z=z(N,E)$ of turns which have to be prohibited to prevent deadlocks in a given network graph. The proposed upper bound is constructive, i.e. its proof generates a simple algorithm (we call it *z*-algorithm) for construction of a tree and labeling of nodes by $\boldsymbol{1,2,\ldots,N}$ such that turn $\boldsymbol{(a,b,c)}$ is prohibited iff at least one of the edges $\boldsymbol{(a,b)}$ or $\boldsymbol{(b,c)}$ does not belong to the tree and $\boldsymbol{a>b}$ and $\boldsymbol{c>b}$.

Denote by $z(N,E)$ a minimal fraction of prohibited turns for prevention of deadlocks in network graph G with N nodes and E edges. Let d_i be a number of neighbors (degree) of node $\boldsymbol{i}$ and T be the total number of turns ($2E=\sum_{i=1}^{N} d_i$, $T=\sum_{i=1}^{N} d_i(d_i-1)/2$).

The following lower bounds for $z(N,E)$ will be useful to estimate performances of the routing strategies that will be described later in this section.

First we note, that

$$z(N,E) \geq (E-N+1)/T. \qquad (1)$$

This bound follows from the fact that there are $b = E-N+1$ linearly independent cycles in G (b is the cyclomatic number for G [13]) and each one of these cycles has to contain at least one prohibited turn to prevent deadlocks.

For example, for 2-d $r\times r$ meshes $N=r^2$, $E=2(r-1)r$, $T=6(r-2)^2+12(r-2)+4$,

$$z(N,E)\geq (r^2-2r+1)/(6r^2-12r+4) \qquad (2)$$

and the lower bound for $z(N,E)$ in this case is close to 1/6 for large meshes. We note that for the North-Last algorithm [6] and 2-d meshes the fraction of prohibited turns is equal to lower bound (2), which proves optimality of the North-Last algorithm.

For the example shown in Fig.2, $N=10$, $E=17$, $T=44$ and by (1) $z(N,E)\geq 8/44$.

Let $C=\{C_1, \ldots C_R\}$ be a system of cycles in G and m is a maximal number of cycles from C containing the same turn. Then

$$z(N,E) \geq R/mT \qquad (3)$$

If $m=1$, then $z(N,E) \geq R/T$, where R is a maximal number of cycles in the network graph such that every turn belongs to at most one cycle.

For the example shown in Fig.2 one can select as C the system of all 9 triangles and one cycle of length 5 ($R=10$). In this case cycles don't have common turns ($m=1$), and by (3) $z(N,E)\geq 10/44$.

Bound (1) is useful when a number of cycles in G is small and (3) can be used for a networks with large numbers of cycles.

We will describe below an algorithm (*z*-algorithm) for prevention of deadlocks (selection of sets of prohibited turns) such that for any network graph $z\leq 1/3$.

At the first step, a node with the minimal degree is selected and labeled by ***1.*** If after deletion from G of node ***1*** and all edges neighboring ***1,*** the remaining graph G-***1*** is still connected, then we prohibit all $d_1(d_1-1)/2$ turns $(\boldsymbol{a,1,b})$ and permit all turns $(\boldsymbol{1,b,c})$. If after deletion from G of node ***1*** and all edges adjacent to ***1*** the remaining graph G-***1*** consists of disconnected subgraphs $G_1,\ldots,G_s$, (this procedure is used also for $s=1$) then, we select nodes

$a_1,\dots,a_s$ (called ***tree*** nodes) such that a_i is a node of G_i and $(a_i,1)$ is an edge in G. All edges $(a_i,1)$ are added to the spanning tree. All constructed tree nodes, except one, are added to the set of ***basic*** nodes B (initially $B=\emptyset$). At the next step we repeat the same procedure to the remaining graph G-1, labeling non-basic node with a minimal degree by ***2***. If several non-basic nodes have the same degree, a node non-adjacent to a tree node is selected (if it exists). At each step, basic nodes are selected in such a way, that every component of connectivity has only one basic node. Process is finished, when all nodes are labeled.

By this procedure, turn (a,b,c) is prohibited if and only if $a>b$, $c>b$ and at least one of the edges (a,b) or (b,c) does not belong to the constructed spanning tree. For example, after the first step we permit all $s(s-1)/2$ turns $(a_i,1,\ a_j)$ ($i,j=1,\dots,s,\ i\neq j$), prohibit all remaining turns $(a,1,b)$ (where a,b Ï $\{a_1,\dots,a_s\}$) and permit all turns $(1,b,c)$.)

To prove that for this procedure $z\leq 1/3$ we note that at step j we prohibit some of the prohibited turns (a,j,b) ($a>j$, $b>j$) and permit all turns (j,b,c) ($b>j$, $c>j$). The number of prohibited turns at step j is at most $d_j(d_j-1)/2$ and the number of permitted turns (j,b,c) is $W=\Sigma(d_i-1)$ where summation is made for all d_j nodes $\boldsymbol{i}$, adjacent to $\boldsymbol{j}$. If node $\boldsymbol{j}$ has a minimal degree in the remaining graph $\boldsymbol{H}$ or if it is non-connected with a basic node, which has a minimal degree (we assume $\boldsymbol{H}$ to be connected), then $W\geq d_j(d_j-1)$. The only remaining case is, when $\boldsymbol{j}$ is connected with a basic node with minimal degree d', at most d'-1 nodes with degree greater or equal than d_j, and at least d_j-d' nodes with degree greater or equal than d_j+1.

In each case, the number of permitted turns is larger than number of prohibited turns by at least a factor of two (we note, that it is true for at least two nodes in any graph). In view of this, we can prove that $z\leq 1/3$ using induction by N.

We note that the proposed z-algorithm provides for at least one permitted path (along the constructed spanning tree) between any two nodes in G.

The complexity of z-algorithm does not exceed $O(N^2)$

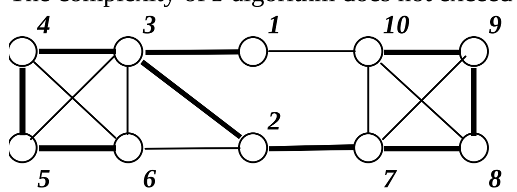

Fig.2. *Example of a network with node labeling (the spanning tree shown in bold.)*

We will illustrate now the proposed algorithm for deadlock elimination by the example shown in Fig.2. At the first step we delete node ***1*** and edges (***3***,***1***) and (***10***,***1***), prohibit turn (***3***,***1***,***10***) and permit turns (***4***,***3***,***1***), (***5***,***3***,***1***), (***6***,***3***,***1***), (***2***,***3***,***1***), (***1***,***10***,***7***), (***1***,***10***,***8***), (***1***,***10***,***9***). Edge (***1***,***3***) is added to the spanning tree (nodes ***3*** and ***10*** are tree nodes). At the second step, after deletion of node ***2***, we have two disconnected graphs, and two edges (***2***,***3***) and (***2***,***7***) are added to the spanning tree (***7*** is selected as a basic node). After 9 steps the total number Z of prohibited turns is 11 and since there are T=44 turns in the network we have z=1/4. The resulting labeling of nodes is given in Fig.2. (Edges of the resulting spanning tree are presented in Fig.2 by bold lines).

In many cases z-algorithm is optimal in terms of a number of prohibited turns. For example, one can prove by this algorithm and lower bounds (1)-(3) that for infinite 2-d meshes z=1/6 and for 3-d meshes z=1/5.

For the full bipartite graph $K_{3,3}$ with N=6 (see Fig.3) z=5/18 (turns (***2***,***1***,***4***), (***2***,***1***,***6***), (***4***,***1***,***6***), (***3***,***2***,***5***), (***4***,***3***,***6***) are prohibited), and z-algorithm is optimal in this case since for any full bipartite graph $K_{n,n}$ with $N=2n$ by (2)-(3) $z\geq 1/4$.

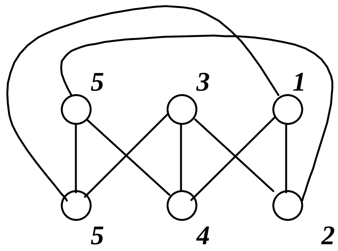

Fig.3. *Routing for bipartite graph* $K_{3,3}$

In Fig.4 we present results of computer simulation of average fraction z_{av} of turns, prohibited by z-algorithm and the up/down approach for deadlock elimination in networks with N=256 nodes as a function of average degree d (number of neighbors for every node). The results are averaged over 1,000 randomly selected connected networks. One can see from Fig.4 that z_{av} is growing with increase in d and z-algorithm is outperforming the up/down approach by at least 10% for $d\geq 10$.

We note also that our approach guarantees that $z\leq 1/3$ but does not guarantee the minimum number of prohibited turns (for the example shown in Fig.2 in the solution presented above turn (***7***,***2***,***8***) does not have to be prohibited, and z can go down to 10/44).

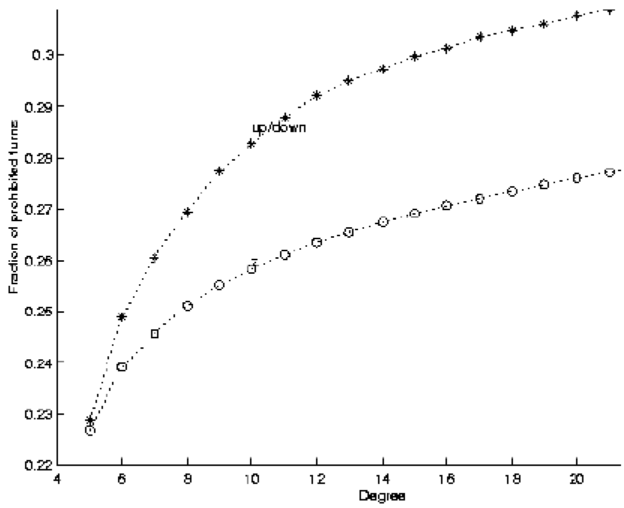

Fig.4 *Fraction* z_{av} *of prohibited turns versus average degree d of nodes for networks with N=256 nodes for z-algorithm (lower curve) and for up/down approach.*

We are planning to investigate the problem of minimization of a number of prohibited turns for deadlock prevention. We are planning also to modify our approach for the case of fault-tolerant routing. In this case some nodes (or links) will be identified as faulty and the set of prohibited turns should be dynamically reconfigured on-line. Our initial results for fault-tolerant wormhole routing for networks with regular topologies can be found in [24]. This fault-tolerant routing will be considered as a special case of routing in networks with changing topology (new users are added to the system, or some users are disconnected). We believe that as long as the rate of the change is low the proposed approach for deadlock-elimination will be efficient.

4. Deadlock-free unicast routing

In the previous section we described the approach for construction of spanning tree *T(G)* and labeling of nodes for any given network graph *G* such that to prevent deadlocks it is sufficient to prohibit turns (***a,b,c***) where ***a***>***b***, ***c***>***b*** and at least one of the edges (***a,b***) and (***b,c***) does not belong to *T*(*G*). (The fraction of these turns does not exceed 1/3).

We will describe in this section several routing strategies satisfying these restrictions and minimizing average path lengths between sources and destinations for given restrictions on local memories in routers. These strategies do not require virtual networks. Generalizations of these strategies to the case of two virtual networks will be also given at the end of this section.

For any intermediate node ***i*** of a packet path a routing protocol estimates length of the shortest path between neighbors of ***i*** and the destination, satisfying the restrictions on turns imposed by *z*-algorithm, and routes the packet to neighbor ***j***, which has the lowest estimation (providing that the corresponding turns in ***i*** and ***j*** are permitted). Sizes of local memories in routers will determine accuracy of these estimations and performance of the corresponding routing strategies.

For the *local* approach the distance between any two nodes is estimated as the tree distance (in links) in *T*(*G*). In this case, the size of the local memory in routers required for storing *T*(*G*) and node labeling is $O(N)$ (*N* is a number of nodes) and $O(N^2)$ steps will be required to compute the distances.

For the *global* approach the distance between two nodes is the length of the shortest path between these nodes such that this path satisfy the restrictions imposed by *z*-algorithm. The size of the local memory required for the global approach is $O(N^2)$ and $O(N^3)$ steps are needed to compute the distances.

We will also describe the *mixed* (hierarchical) approach, which will be a combination of the local and global approaches.

We will illustrate now local, global and mixed approaches for the network shown in Fig.5. (The spanning tree and labeling of nodes generated by *z*-algorithm are also given in Fig.5).

Suppose that the source is node ***5*** and the destination is node ***2***. Then for the local approach path ***5-9-7-6-2*** of length *l*=4 will be selected, since in this case there is no information in the local memory of node ***5*** about edges (***2,4***) and (***6,8***). For the global approach path ***5-4-2*** of length *l*=2 will be selected.

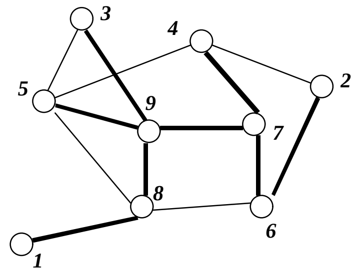

Fig.5. *Example of routing using local, global and mixed approaches.*

For the mixed approach we partition spanning tree *T*(*G*) into *Q* disjoint connected components (clusters). Every node will have global information about structures of all clusters. Also, the structure of *T*(*G*) with node labels is stored in local memories. (The local and global approaches for estimating distances outlined above are special cases of the mixed approach with *Q*=*N* and *Q*=1).

We note that the mixed approach allows efficient use of the available memory in routers to achieve the best possible routing. More memory is available, smaller number of clusters *Q* will be, with the proportional increase in the size of each cluster. This will provide for more information about shortcuts in *T*(*G*).

5. Conclusions

Our approach for constructing a deadlock-free routing strategy consists of two stages.

At the first stage using the turn model we are constructing a set of prohibited turns to prevent deadlocks. We outlined in Section 3 two approaches for solving this problem. For the first approach (*z*-algorithm) we can guarantee that a fraction of prohibited turns do not exceed 1/3 for any network topology. Using the lower bounds on a fraction of prohibited turns we can show that *z*-algorithm is optimal or close to optimal in many cases. Our initial simulation results show that *z*-algorithm has better performance (and lower number of prohibited turns) than the approach based on up/down restrictions. The second approach for constructing a set of prohibited turns is based on two virtual networks. Since the restrictions on turns imposed by *z*-algorithm and the up/down approach are complimenting each other, we are using two virtual networks V_z and V_{updown} for routing based on *z*-algorithm and the up/down approach. A packet which have been moving in V_z can be transferred to V_{updown} (but not otherwise)

At the second stage our goal is to construct an optimal routing strategy satisfying a given set of prohibited turns and given sizes of local memories. We outlined in the Section 4 three approaches to solve this problem. These approaches (local, global and mixed) differ in the way distances in the network graph are estimated by using local information stored in routers. The global-information-based approach provides for the shortest path and smallest delivery times but requires local memories in routers of the order $O(N^2)$. For the local approach memory of the size $O(N)$ is sufficient but this result in an increase in delivery time. The mixed approach provides for a good trade-off between sizes of local memories and delivery time. We are planning to investigate these approaches and compare them with the existing techniques.

We also plan to generalize our routing techniques to the case of adaptive routing taking into account queue sizes at each node and to the case of fault-tolerant routing when some of the nodes and/or links in the network are identified as faulty.

Literature

1. N.J. Boden et al., "Myrinet: A Gybabit per Second Local Area Network," *IEEE Micro*, pp.29-35, February 1995.
2. R.V. Boppana and S. Chalasani, "A Comparison of Adaptive Wormhole Routing Algorithms," *Computer Architecture News*, 21(2), pp. 351-360, May 1993.
3. K. Chakrabarty, M.G. Karpovsky, L.B. Levitin, "Fault Isolation and Diagnosis in Multiprocessor Systems with Point-to-Point Connections," *Fault Tolerant Parallel and Distributed Systems*, Kluwer Academic Publishers, 1998, pp.285-301.
4. S. Chalasani and R V. Boppana, "Communication in Multicomputers with Nonconvex Faults," *IEEE Trans. on Computers*, vol.46, pp.616-622, May 1997.
5. C.M. Chiang and L. Ni, "Deadlock-Free Multihead Wormhole Routing," *Proc. of First High Performance Computing Asia Conf.*, 1995.
6. C. Cunningham and D. Avresky, "Fault-Tolerant Adaptive Routing for Two-Dimensional Meshes." *Proc. of First Int. Symp. on High Performance Computing Architecture*, January 1995.
7. W.J. 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, April 1997.
8. J. Duato, "A New Theory of Deadlock-Free Adaptive Routing in Wormhole Networks," *IEEE Trans. on Parallel and Distributed Systems*, vol. 4, pp. 1,320-1,331, 1993.
9. J. Duato, "A Necessary and Sufficient Condition for Deadlock-Free Adaptive Routing in Wormhole Networks," *Proc. of Int. Conf. on Parallel Processing*, vol. I., pp. 142-149, August 1994.
10. E. Fleury and P. Fraigniaud, "A General Theory for Deadlock Avoidance in Wormhole-Routed Networks," *IEEE Trans. on Parallel and Distributed Systems*, vol. 9, pp.626-638, July 1998.
11. C. Glass and L. Ni, "The Turn Model for Adaptive Routing," *Proc. of the 19th Annual Int. Symp. on Computer Architecture*, pp. 278-286, May 1992.
12. C. Glass and L. Ni, "The Turn Model for Adaptive Routing," *J. of ACM*, vol.5, pp.874-902, 1994.
13. F. Harary, "Graph Theory", *Addison-Wesley*, Reading, Mass, 1969.
14. R.W. Horst, "ServerNet^{TM} Deadlock Avoidance and Fractahedral Topologies", *Proc. of IEEE Int. Parallel Processing Symp.*, pp.274-280, 1996.
15. Kang Shin, Stuart Daniel and Sang Kyun Yun, “A Router Architecture for Flexible Routing and Switching in Multihop Point-Point Networks”, *IEEE Trans. on Parallel and Distributed Systems*, vol.13, No. 1, Jan. 1999.
16. P. Kermani and L. Kleinrock, "Virtual Cut-Through: A New Computer Communication Switching Technique," *Computer Networks*, vol 3, pp.267-286, 1979.
17. R. Kesavan, K. Bondalapati and D. Panda, "Multicast on Irregular Swith-Based Networks with Wornhole Routing”, *Proc. of the Third Int. Symposium on High Performance Computer Architecture*, pp 48-57, Feb. 1997.
18. P. LeMahieu, V. Bohossian, J. Bruck, "Fault-Tolerant Switched Local Area Networks," *Proc. of the First Merged Int. Parallel Processing Symposium and Symposium on Parallel and Distributed Processing*, pp. 747-751, 1998.
19. R. Libeskind-Hadas, D. Mazzoni and R. Rajagopalan, "Tree-Based Multicasting in Wormhole-Based Irregular Topologies," *Proc. of the First Merged Int. Parallel Processing Symp. and Symp. on Parallel and Distributed Processing*, pp.244-249, 1998.
20. R. Libeskind-Hadas, D. Mazzoni and R. Rajagopalan, "Optimal Contention-Free Unicast-Based Multicasting in Switch-Based Networks of Workstations," *Proc. of the First Merged Int. Parallel Processing Symp. and Symp. on Parallel and Distributed Processing*, pp. 358-364, 1998.
21. L. M. Ni and P.K. McKinley, "A Survey of Wormhole Routing Techniques in Directed Networks," *Computer*, vol. 26, pp. 62-76, 1993.
22. M. Schroeder et al, "Autonet: A high-speed self configuring local area network using point to point links”, *Technical Report 59, DEC SRC*, April 1990.
23. L. Schwiebert and D. Jayasimha, "A Necessary and Sufficient Condition for Deadlock-free Wormhole Routing”, *J. Parallel and Distributed Computing*, vol. 32, pp 103-117, 1996.
24. L. Zakrevski and M.G. Karpovsky, "Fault-Tolerant Message Routing for Multiprocessors," *Parallel and Distributed Processing* (Editor J.Rolim), Springer, 1998, pp.714-731.
25. L. Zakrevski, M.G. Karpovsky,"Fault-Tolerant Message Routing in Computer Networks," *Proc. of Int. Conf. on PDPA-99*, 1999.