Deadlock-Free Fully Adaptive Routing in Irregular Networks without Virtual Channels

Dong Xiang
School of Software
Tsinghua University
Beijing 100084, China
dxiang@tsinghua.edu.cn

Zhigang Yu
Dept. of Computer Science and Tech.
Tsinghua University
Beijing 100084, China
yuzg@live.com

Jie Wu
Dept. of Computer and Info. Sciences
Temple University
Philadelphia, PA 19122
jiewu@temple.edu

***Abstract*—This paper proposes a new flow control scheme in VCT-switched irregular networks. Based on the new scheme, a novel deadlock-free fully adaptive routing algorithm is introduced. The algorithm does not need any virtual channel. It requires that each input port of a switch holds at least two 1-packet-sized buffers. The flow control scheme is proposed based on a baseline routing scheme, where the downstream nodes check the number of safe buffers at the upstream nodes. The proposed fully adaptive routing algorithm is on the basis of different baseline routing schemes: up*/down*, and multiple spanning tree based routing schemes. Extensive simulation results validate the effectiveness of the proposed method as compared to well-known existing approaches.**

Keywords*—*Flow control, irregular networks, adaptive routing, multiple spanning trees.

I. Introduction

Networks of workstations (NOWs) or clusters are recognized as good alternatives for parallel computing due to their competitive cost/performance ratio and wire flexibility [19]. Practical networks such as Autonet [25], Myrinet [1], and Server-Net [9] are examples of high-performance with irregular interconnects. The regular networks, such as meshes/tori/hypercubes, are unrealistic due to variations in module sizes and shapes, which are not suitable for clusters or NOWs. We consider virtual cut-through (VCT) switched NOWs because we think VCT is more popular for NOWs or clusters.

It is essential to propose an effective deadlock-free adaptive routing algorithm in irregular networks [33]. Wu and Sheng [29] proposed a deadlock-free routing scheme for irregular networks using prefix routing. Three different multicast algorithms were proposed for wormhole-switched irregular networks in [11], with their respective node orders to reduce contention.

Methods in Bolotin, *et al.* in [2] and Mejia, *et al.* in [17] minimize the size of the routing table at each router for NoCs with irregular topologies. Flich, et al. in [8] proposed a routing scheme LBDR for different variations of the 2D meshes, which avoids any routing tables. It is found that the LBDR still cannot support some topologies. Rodrigo, et al. [24] proposed a new mechanism, called uLBDR, that adapts to any irregular topologies derived from 2D meshes. The method in [24] still does not need any routing tables. Cano, et al. in [3] proposed a new routing methodology and router implementation for complex SoCs. This method avoided routing table look-up by mapping the irregular network into a 2D mesh with constant or reduced logic cost, regardless of the system size.

A general methodology for the design of adaptive routing algorithms for networks with irregular topology has been proposed in [26] by extending the Duato's protocol to the irregular networks. Puente, *et al.* in [21] proposed a pseudo-Hamiltonian-cycle-based adaptive routing mechanism for irregular NOWs, which leads to apparent improvement over the classical up*/down* routing algorithm, which comes at an acceptable extra cost. Puente, et al. in [22] proposed an adaptive router architecture for irregular NOWs, which improves network capabilities by allocating more resources to the fastest and most-used virtual network. However, there may still exist traffic congestion based on the method in [21], [22], because a single spanning tree (ST) was used.

Turn models were proposed for high-performance routing in irregular networks [10], [13], [18], [27]. These methods usually set the minimum number of prohibited turns. A zone-ordered label-based routing scheme was proposed for irregular networks with multiple spanning trees (MSTs) in [18]. Any packet is delivered along the ordered zones in [18]. There may be difficulty in finding the best zones and zone order, especially when the number of STs is larger.

The Main contributions of this paper include: (1) the general framework for a fully adaptive routing algorithm in VCT-switched irregular networks is presented, based on a simple flow control scheme; (2) deadlock-free fully adaptive routing algorithms are presented based on the flow control scheme for up*/down*, and MST routing schemes.

In the rest of the paper, Section II presents the preliminaries. The new flow control scheme for irregular networks is in Section III. The deadlock-free adaptive routing algorithms are presented in Section IV. Simulation results are presented in Section V, and Section VI concludes the paper.

II. Preliminaries

Some background is provided first. The technique to select roots for the MSTs, ST assignment, and constrained turn selection proposed in [31] are presented after that.

A. Background

The up*/down* routing scheme in irregular networks was proposed for Autonet networks in [25] by selecting a single root and then establishing an ST. Its general strategy was based on selecting routes in an ST, where the packets go up the ST on leaving the source and then, come back down toward the destination.

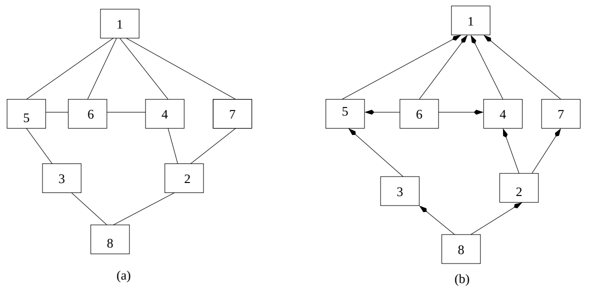

Fig. 1. An irregular network with a single ST: (a) the irregular network with eight nodes, and (b) the ST for up*/down* routing.

Let the r be the root. The distance $dist(u,r)$ from u to r is greater than that from v to r. The link (u,v) is up. Otherwise, the link is down. As the $dist(u,r) = dist(v,r)$, the direction of link (u,v) is set from the higher ID to the lower. The routing path from a source to a destination is established in such a fashion that zero or more up links (towards the root) should be traversed before zero or more down links can be traversed (away from the root) in order to reach the destination.

Fig. 1(a) presents an irregular network with eight nodes, and the ST for the up*/down* routing scheme when the node 1 is selected as the root is shown in Fig. 1(b). The drawbacks are that the selected paths may not be the shortest paths and that links near to the root can be congested and become bottlenecks.

It was shown in [20] that deadlock-free routing without using virtual channels can be successfully used in cycles of any size. A restricted injection mechanism is applied to any packet to avoid deadlocks that is trying to enter the cycle which is called Bubble Flow Control(BFC). If under no circumstances the storage spaces for packets in a cycle are allowed to become full, the packets traversing along this cycle will always be able to advance.

B. Root Selection for Multiple Spanning Trees

The method in [31] selects multiple roots for adaptive deadlock-free routing in irregular networks. It avoids congestion at the root of the ST for the traditional up*/down* routing. We use different metrics for the first root and the rest roots. This is reasonable because selection of the first root is to minimize the average distance between any pair of nodes. We have,

$$FR = min_r\{\sum_{u,v} dist_r(u,v)\}, \tag{1}$$

where $dist_r(u,v)$ is the distance between u and v according to the up*/down* routing scheme [25] in the ST with the root r. The larger the size of a cycle, the more influential it is on the performance. The remaining roots are selected to minimize the number of constrained turns for removing all cyclic input port dependencies and minimize the average distance for all pairs of nodes based on the metric as presented in Equation (2),

$$SR = \frac{\sum_{u,v} \triangle dist(u,v)}{\triangle CT}, \tag{2}$$

where $\triangle CT$ is the number of increasing constrained turns to remove all cyclic dependencies in the input port dependency graph. The method in [31] selects the node as the root that produces the maximum metric as presented in Equation (2). In Equation (2), $\sum_{u,v} \triangle dist(u,v)$ stands for reduction for distance between all pairs of nodes. In all cases, the distance between a pair of nodes does not increase after the second root is inserted and the second ST is established.

C. Spanning Tree Assignment

A simple way to select the ST for a pair of nodes is: determine the ST according to the length of the minimal path for them in the corresponding ST. The ST with shorter path is selected for the packet. However, load-balancing should also be considered when selecting the ST. That is, the number of packets with the same source should be evenly assigned to all STs. We call an ST according to its root.

Consider delivering a packet from node 5 to node 7 as shown in Fig. 2. The minimal path selected to deliver the packet in the ST 1 by the original up*/down* routing scheme is 5-1-7, where the packet is delivered along an up link (5,1) first, and a down link (1,7) after that. The length of the path is two. If the packet is delivered via the ST 8, the minimal path selected by the up*/down* routing scheme is 5-3-8-2-7. Four hops are necessary. Therefore, it is better to deliver the packet from node 5 to node 7 along the paths in the ST 1.

If a packet from node 2 to node 6 is delivered in the ST 1, the minimal path can be 2-4-6. The minimal path to deliver the packet is 2-4-6 when it is delivered across the ST 8. That is, the path lengths in both STs are equal. It is ok to deliver the packet in both STs.

D. Constrained Turn Selection for Deadlock Avoidance

Fig. 2(b) presents the input port dependency graph of the irregular network with input ports as shown in Fig. 2(a). The vertices of the graph are input ports. There is a dependency from u to v if a packet request a buffer at v when it occupies a buffer at u. There may exist some potential cyclic dependencies. We propose a new technique in [31] to constrain a small number of turns instead of introducing some prohibited turns. A scheme like the bubble flow control [20] is adopted in irregular NOWs for deadlock avoidance. That is, the constrained turns are allowed when the buffer requirement at the input port is satisfied. The following metric is used to select constrained turns.

$$m(v_i) = \sum_{v_i \in C} size(C), \tag{3}$$

where the input port i of node v is contained in the cycle C. Our method assigns a benefit metric $size(C)$, the size of the cycle C, for each input port when the node and its input port are contained in the cycle. The size of a cycle is the number of links to establish the cycle. The reason why we assign each input port the size of the cycle is that a cycle of bigger size can be more influential to the performance.

Our method selects the input port with the most metric. The benefit metric for each node and its input port is updated after a constrained turn node has been selected. That is, the benefit metric of any node and its input port, that are contained in the

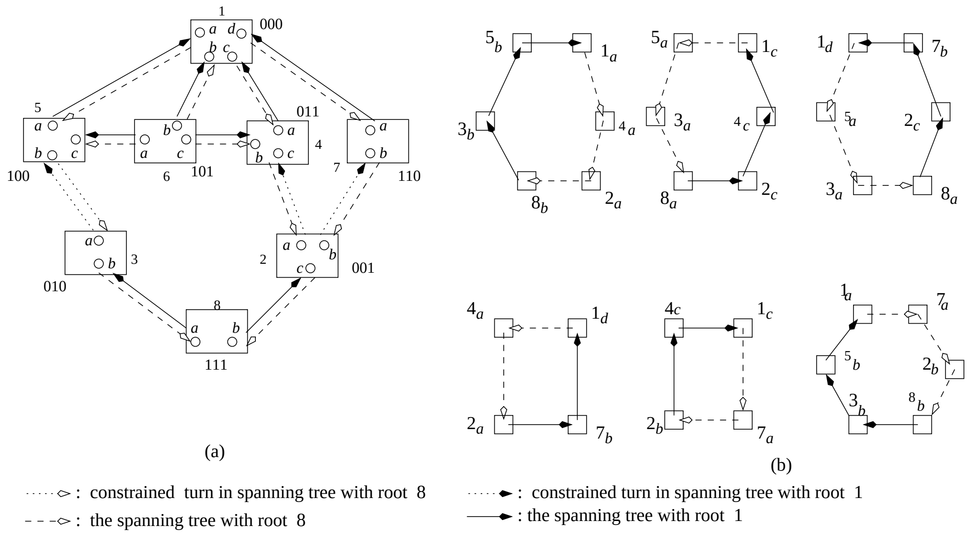

Fig. 2. Cyclic channel dependencies in an irregular network with STs: (a) the irregular network with input ports, and (b) the cyclic dependencies.

flow-control-irregular-network()

Inputs: coordinates of the current node C, coordinates of the destination D, *free* buffers: $(f_1, f_2, \ldots, f_n)$, *safe* packets: $(s_1, s_2, \ldots, s_n)$
Output: selected output channel.

1) $S := \emptyset$; $ch := null$;
2) if $C = D, ch := internal$;
3) for each next node i of C in the shortest path to d according to the baseline routing function R, do
   if (*flow-control*(i))
   $S \leftarrow S \cup \{i\}$;
4) if $S \neq \emptyset$, ch := select(S).

Fig. 3. The general framework of a fully adaptive routing algorithm in irregular networks based on the new flow control scheme.

flow-control(i)

Inputs: the number of free buffers: f_i, and the number of *safe* packets: s_i.
Output: whether the packet can route to the next node.

1) If ($f_i > 1$) return true; exit.
2) If ($f_i = 1$ and $s > 0$), return true; exit.
3) If ($f_i = 1$ and $s_i = 0$), and the next hop conforms to the baseline routing scheme R, return true; exit.
4) Return false.

Fig. 4. The general flow control function using R as the baseline routing scheme.

cycle, is reduced by the size of the cycle. The above process continues until all cycles have been removed. As shown in Fig. 2, the input port of 4_c, 3_a, 5_b, and 7_b are selected to remove all six cyclic input port dependencies.

III. The New Flow Control Scheme for Irregular Networks

Recently, Luo and Xiang [15] proposed a fully adaptive routing algorithm for tori without any virtual channels based on a new flow control scheme. That work is the initial motivation of this paper. In this section, we propose the general framework of a new fully adaptive routing algorithm for irregular networks based on the flow control as shown in Fig. 3. The baseline routing function R is deadlock-free, where it can be the up*/down* routing scheme, MST-based routing [31], and any other refined deadlock-free routing schemes. The input buffers of *flow-control-irregular-network*() are organized as dynamically allocated multi-queues. Two queues are needed to avoid deadlocks. By this case, the performance would be even better.

Two classes of packets, *safe* and *unsafe* packets, are defined in *flow-control-irregular-network*(). Based on a routing algorithm R as the baseline routing scheme in an irregular network, a packet is *safe* to the downstream node if it is delivered along a hop provided by the deadlock-free baseline routing scheme R; otherwise, it is *unsafe*. In the rest of this paper, we say a packet is *safe* or *unsafe* means that it is *safe* or *unsafe* to the current node if not specifically defined.

Fig. 3 presents the general framework of the fully adaptive routing algorithm for irregular networks based on the new flow control scheme and baseline routing scheme R. Here, f_i, and s_i stand for the number of free buffers and *safe* packets in the input port of the ith neighbor directly connected to the current node C, respectively. The input of this algorithm including coordinates of the current node and the destination, the number of free buffer numbers and special packet numbers of all neighboring input ports. The available channel set and the selected output channel are initialized as $\emptyset$ and $null$.

If the current node is equal to the destination, the internal channel is selected. Otherwise, for each upstream node v in the minimum paths from the current node to the destination checks whether the packet would be forwarded through the function *flow-control*() as presented in Fig. 4. The function *flow-control*(i) returns 1 when the packet is *safe* to the ith neighbor and a free buffer is available. Otherwise, it returns 0 when the input port is unsafe to the packet. Function *flow-control*(i) avoids filling any input port with only *unsafe* *flow-control(i)*

Inputs: the number of free buffers: f_i, and the number of *safe* packets: s_i.
Output: whether the packet can route to the next node.

1) If ($f_i > 1$) return true; exit.
2) If ($f_i = 1$ and $s > 0$), return true no matter whether the next hop conforms to up*/down* routing; exit.
3) If ($f_i = 1$ and $s_i = 0$), and the next hop conforms to the up*/down* routing scheme, return true; exit.
4) Return false.

Fig. 5. Flow control function using up*/down* routing as the baseline routing scheme.

MST-based-route(v,d)

1) Deliver the packet from v to d if v is in the minimal paths from v to d in the assigned ST in the following way until it reaches the destination d,
   a) If the link is a constrained turn, it can be delivered if the input port of the next hop contains two empty buffers.
   b) Otherwise, it can be delivered if the input port of the next hop contains at least one empty buffer.
   c) Consume the packet if the local node has been the destination d.

Fig. 6. MST-based deadlock-free routing.

packets. It checks how many free buffers (f) and *safe* packets (s) in the input buffer of the ith input port.Our method selects an output channel from S if it is not $\emptyset$. Otherwise, the packet is put in the waiting list and blocked.

The key point of the algorithm *flow-control-irregular-network*() is the *flow-control*(i) function. It avoids filling any input port with only *unsafe* packets. The input buffers are organized as dynamically allocated multi-queues. There are two queues at each input port: an *safe* packet queue and a *unsafe packet* queue. All *safe* packets to a node in an input port are linked as a *safe* packet queue. The rest are linked as an *unsafe packet* queue. So the safe packets would not be blocked by unsafe packets. In all cases, no cyclic dependencies will be introduced by the adaptive hops, therefore, the proposed fully adaptive routing algorithm is deadlock-free. We shall not provide more detailed proof on deadlock freedom of the proposed algorithm in this paper.

Fig. 5 presents the flow control function for the baseline routing scheme up*/down* routing. What is meant by the next hop conforms to the up*/down* routing is that one of the following three conditions must be met when the ith neighbor of C is in the shortest path from C to d: (1) the packet reaches the current node C via an up link and expects to reach the ith neighbor via an up link; (2) the packet reaches C by an up link and the next node via a down link; (3) the packet reaches C by a down link and the next node via a down link.

The MST based routing scheme [31] can be used as the baseline routing scheme. The preliminary section presents more details of the algorithm. The MST-based routing scheme provides deadlock-free routing because any possible cycle contains at least one constrained turn, which introduces no deadlock configuration. The following rules determine whether

flow-control-MST(i)

Inputs: the number of free buffers: f_i, and the number of *safe* packets: s_i.
Output: whether the packet can be delivered to the next node.

1) If ($f_i > 1$) return true; exit.
2) If ($f_i = 1$ and $s_i > 0$), no matter whether the next hop conforms to the baseline routing scheme R, return true; exit.
3) If ($f_i = 1$ and $s_i = 0$), the next hop conforms to the baseline routing scheme and the input port of the next node i is not a constrained turn, return true; exit.
4) Return false.

Fig. 7. The flow control function using MST based routing as the baseline routing scheme.

a packet can be delivered to the next node when one of the MST based routing schemes is used as the baseline routing scheme. Remember that a packet is assigned to a single ST. As shown in Fig. 7, any packet is delivered in the same ST based on up*/down* routing except the adaptive hops.

- $f > 1$, the packet could be delivered because there is more than one free buffers in the next node.
- $f = 1$ and $s > 0$ no matter whether input port of the next node is a constrained turn, the packet could be delivered because there is at least one *safe* packet in the next node which can always free the occupied buffer.
- $f = 1$ and $s = 0$ and the next node is not a constrained turn, the packet could be delivered if the hop conforms to the baseline routing scheme; otherwise, keep the packet in the waiting list.
- $f = 0$, keep the packet in the waiting list.

The MST-based deadlock-free adaptive routing is presented in Fig. 6. It is clear that the packet can be delivered to the neighbor in a shortest path if the input port contains more than one free buffer ($f > 1$). The packet can be delivered to the free buffer of the neighbor if it contains one free buffer and one safe buffer no matter whether the input port of the next node is not a constrained turn because the safe buffer can always be freed. The packet can be delivered to the next node if the hop conforms to the baseline routing scheme, there exists a free buffer and no safe buffer in the input port of the next node, and the input port of the next node is not a constrained turn. In all other cases, the packet cannot be delivered to the next node.

What is meant by conform to the MST based routing scheme? Two consecutive hops, if none of them is an adaptive hop provided by the MST based routing, they must follow one of the following rules: (1) the packet reaches the current node C via an up link and expects to reach the next node via an up link; (2) the packet reaches C by an up link and the next node via a down link; (3) the packet reaches C by a down link and the next node via a down link.

IV. Deadlock-Free Adaptive Routing Based on the New Flow Control Scheme

We present a new adaptive routing scheme with unfixed STs. We then introduce the fully adaptive routing algorithms *adaptive-route-up-down(C, d)*

Inputs: the current node C, and the destination d, directions for all links in the ST, and states of buffers at the neighbors.
Output: the next channel.

1) For each neighbor i of C in a shortest path to d, do steps 2), 3), 4), and 5);
2) if the packet reaches C along a up link and the link to the next node i is still a up link, and both hops are provided by the baseline routing scheme, $S \leftarrow S \cup \{i\}$ when *flow-control*(i) is true;
3) if the packet reaches C along a up link and the link to the next node i is a down link, and both hops are provided by the baseline routing scheme, $S \leftarrow S \cup \{i\}$ when *flow-control*(i) is true;
4) if the packet reaches C along a down link and the link to the next node i is still a down link, and both hops are provided by the baseline routing scheme, $S \leftarrow S \cup \{i\}$ when *flow-control*(i) is true;
5) for all other neighbor i of C in a shortest path to d, i is not in S, if *flow-control*(i) is true and i is not in S, $S \leftarrow S \cup \{i\}$;
6) if $S \neq \emptyset$, ch := select(S).

Fig. 8. Adaptive routing using up*/down* routing as the baseline routing scheme.

adaptive-route-MST(C,d)

Inputs: the current node C, and the destination d;
directions of all links in MST;
assignment of the packets to all STs.
Output: the next channel.

1) For each neighbor i of C in a shortest path to d, do steps 2), 3), 4), and 5);
2) if the packet reaches C along a up link via the baseline routing function provided hop and the link to the next node i is still a up link in the jth ST, if *flow-control-MST*(i) is true, $S \leftarrow S \cup \{i\}$;
3) if the packet reaches C along a up link and the link to the next node i is a down link in the jth ST, and both links are provided by the baseline routing function, if *flow-control-MST*(i) is true, $S \leftarrow S \cup \{i\}$;
4) if the packet reaches C along a down link and the link to the next node i is still a down link in the jth ST, and both links are provided by the baseline routing function, if *flow-control-MST*(i) is true, $S \leftarrow S \cup \{i\}$;
5) for all other neighbor i of C in a shortest path to d, if *flow-control-MST*(i) as presented in Fig. 7 returns true, and i is not in S, $S \leftarrow S \cup \{i\}$;
6) if $S \neq \emptyset$, ch := select(S).

Fig. 9. Adaptive routing using MST based routing as the baseline routing scheme.

with different baseline routing schemes without any virtual channel: (1) up*/down* routing, (2) MST based routing, and (3) unfixed MST based routing.

A. Fully Adaptive Routing Using Up/Down* Routing as the Baseline Routing Scheme*

We first propose the fully adaptive routing algorithm by using the up*/down* routing scheme as the baseline routing scheme. Fig. 8 presents the detailed routing algorithm. The inputs of the algorithm includes the coordinates of the current node, and the destination, directions of the all links of the ST, the root of the ST, and the buffer states at the input ports of the current node C's neighbors. Steps 2), 3), and 4) present available inputs port of the neighbors as the next hop which conform to the baseline routing scheme, up*/down* routing.

The routing algorithm in Fig. 8 shows three different cases when the next hop conforms to the baseline routing scheme up*/down* routing. The next node must be in the shortest paths from C to d. The three cases include: (1) the packet reaches the current node along a up link and the link to the next node i is still a up link, where both hops are provided by the baseline routing scheme; (2) the packet reaches C along a up link and the link to the next node i is a down link, where both hops are provided by the baseline routing scheme; and (3) the packet reaches C along a down link and the link to the next node i is still a down link, where both hops are provided by the baseline routing scheme.

After all next hops along the shortest paths have been identified, our method checks the next hops that do not conform to the baseline routing scheme. However, one of the buffer conditions presented by *flow-control*(i) in Fig. 4 must be satisfied. The selection function finally selects one of the next hops in S to deliver the packet.

The possibility to establish a cyclic dependency by the proposed fully adaptive routing algorithm is that one or more adaptive hops. However, any dependencies established by the adaptive hops can be eliminated because a packet can advance by a hop not provided by the baseline routing scheme when the input port of the next node is safe to the packet.

B. Fully Adaptive Routing Using Multiple Spanning Tree Based Routing as the Baseline Routing Scheme

Fig. 9 presents the fully adaptive routing algorithm based on the new flow control scheme when the MST based routing scheme is used as the baseline scheme. The MST based routing scheme is very simple, which is presented in Fig. 6. Initially, the MSTs are established and the packets are assigned to the STs for load-balancing consideration. A packet is delivered along the fixed ST. However, some of the hops, that do not conform to the baseline routing scheme, do not follow this.

Similar conditions as presented in Fig. 8 must be met when both consecutive hops in the same ST are provided by the baseline routing scheme. The three cases include: (1) the packet reaches the current node along an up link and the link to the next node i is still an up link; (2) the packet reaches C along an up link and the link to the next node i is a down link; and (3) the packet reaches C along a down link and the link to the next node i is still a down link.

Our method then checks all other next nodes in the shortest paths from the current node to the destination whether one of the buffer conditions as presented in Fig. 7 can be met. If so, the next node can be provided by the fully adaptive routing algorithm. In any case, the packet waits for one of the buffers provided by the baseline routing scheme in the fixed ST when traffic congestion occurs.

C. Deadlock Freedom Proof

The fully adaptive routing algorithms are deadlock-free because: (1) there are no cyclic dependencies in the same ST because the packets follow the deadlock-free baseline

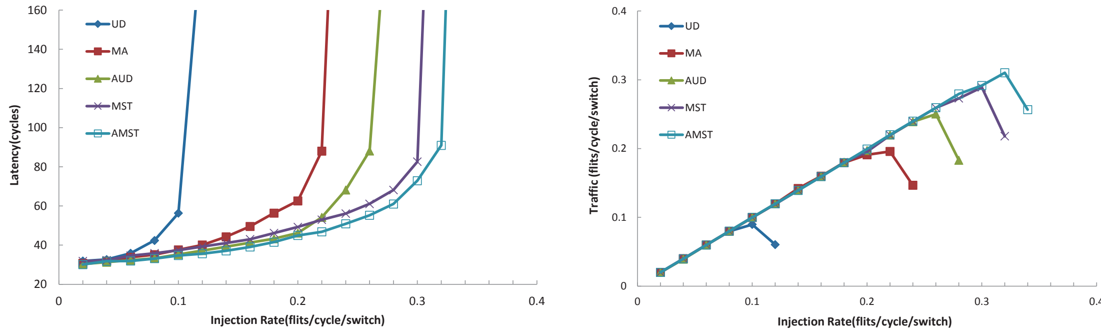

Fig. 10. Performance comparison with previous methods (UD, MA, AUD, MST) in an irregular network (32,64).

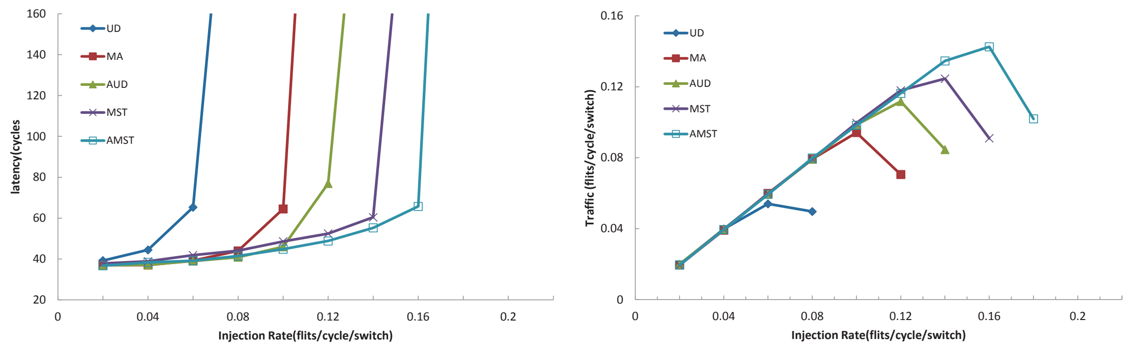

Fig. 11. Performance comparison with previous methods (UD, MA, AUD, MST) in an irregular network (64,128).

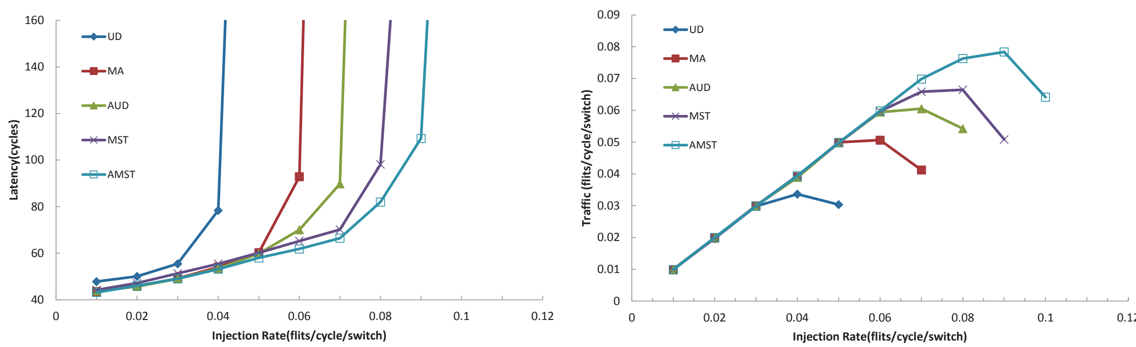

Fig. 12. Performance comparison with previous method (UD, MA, AUD, MST) in an irregular network (128,256).

routing such as: up*/down* routing or MST based routing, whose proofs can be found in [25] and [31], respectively; (2) obviously, the adaptive hops provided by the new flow control scheme cannot introduce any cyclic dependencies [15]; (3) no cyclic dependencies can be established across different STs, because any cycle contains at least one constrained turn. In all, we can conclude that the above fully adaptive routing algorithms are deadlock-free.

V. Simulation Results

The proposed routing algorithms and flow control scheme for VCT-switched irregular networks have been implemented in a cycle-accurate simulator by C++ [31]. Different baseline routing schemes are used including up*/down* routing (UD), multiple spanning tree based routing (MST). The implemented fully adaptive routing algorithms based on the proposed new flow control scheme are AUD and AMST. The previous methods, the original up*/down* routing scheme [25] (UD), the adaptive routing scheme based on Duato's protocol [26] (MA), and the MST based routing scheme (MST) [31], are also implemented.

Links are randomly added to the network, while the degree of each node is no more than four [32]. Two metrics are used to evaluate the performance of a method: the *latency* to deliver a packet (cycle) and the *accepted traffic* (flit/cycle/switch). The size of a packet is set to 16 flits for all simulation results. The topology of the irregular network is generated randomly, while we present the average results of 10 irregular networks.

We consider three STs for all simulation results related to multiple spanning trees and the *uniform* traffic pattern for all simulation results. We do not consider routing table storage and look-up for previous methods (UD and MA), and just present the performance comparison with them.

Fig. 10 presents the performance evaluation of irregular networks with 32 switches/nodes and 64 links. The up*/down* routing scheme UD reaches the saturation point very quickly when the applied load reaches 0.1. The extended Duato's protocol MA reaches the saturation point when the applied load has reached 0.22. The adaptive routing algorithm AUD with the up*/down* routing scheme as the baseline routing works better than UD and MA, whose saturation point is about 0.26, while the accepted traffic is also apparently better than the previous two methods. The most recent work (MST), based on multiple spanning tree, works better than AUD, although it is not fully adaptive.

The saturation point of MST is a little later, which is at 0.30. The adaptive routing algorithm AMST is based on the flow control scheme, which uses the multiple spanning tree based routing as the baseline routing scheme. The AMST routing algorithm apparently works better than UD, MA, AUD, and MST on accepted traffic and latency to deliver a packet in all cases.

Figs. 11 and 12, present the performance evaluation on irregular networks with 64 switches and 128 links, 128 switches and 256 links. Fig. 11 presents the performance comparison of the proposed algorithms with previous methods. The MST algorithm does not work better than MA and AUD on latency to deliver a packet when the injection rate is very low, just like that in the networks with 32 switches. As shown in Fig. 11, the fully adaptive routing algorithm AUD still works better than MST when the injection rate reaches 0.10. The reason should be that it is easy for MA and AUD to reserve adaptive channels when the injection rate is not high enough. The order for the algorithms to reach the saturation points is still UD, MA, AUD, MST and AMST .

Fig. 12 provides a performance comparison of the new algorithms with previous ones in networks with 128 switches and 256 links. The situations are still the same. The MST algorithm works worse than AUD when the injection rate is less than 0.05, which consistently works better than UD, MA, and AUD after that. The AMST algorithms work even better in all cases.

The proposed method avoids routing table look-up by implementing the routing table look-up through some very simple extra logic. The details for the router architecture are not presented in this paper. Therefore, the scalability is not a problem for the proposed method. In all simulation results for MA and UD, the extra latency to look up the routing tables is not included.

VI. Conclusions

A new flow control scheme was proposed for irregular networks, which provides a series of fully adaptive routing algorithms for VCT-switched (or packet-switched) irregular networks. Our methods provide a deadlock-free adaptive routing scheme using unfixed multiple spanning trees, that is, a packet can change its spanning tree at the intermediate nodes in order to reduce latency to deliver it. The fully adaptive routing framework is applied to up*/down* routing, multiple spanning tree, and the unfixed multiple spanning tree routing schemes. Simulation results show that the proposed flow control scheme can provide very effective fully adaptive routing algorithms without any virtual channels, and bring $\geq 10\%$ improvement when compared to the original routing algorithms.

References

[1] N. J. Boden, D. Coben, R. E. Felderman, A. E. Kulawik, C. L. Seitz, J. Seizovic and W. Su, "Myrinet - A gigabit per second local area network," *IEEE Micro*, pp. 29-36, Feb. 1995.

[2] E. Bolotin, I. Cidon, R. Ginosar, and A. Kolodny, "Routing table minimization for irregular mesh NoCs," in *Proc. of Design Automation, and Test in Europe*, pp. 942-947, 2007.

[3] J. Cano, J. Flich, J. Duato, M. Coppola, and R. Locatelli, "Efficient routing implementation in complex systems-on-chip," in *Proc. of Int. Symp. on Networks-on-Chip*, 2011.

[4] J. Cong, C. Liu, and G. Reinman, "ACES: Application-specific cycle elimination and splitting for deadlock-free routing on irregular network-on-chip," in *Proc. Design Automation Conf.*, pp. 443-448, June 2010.

[5] J. Duato, A. Robles, F. Silla, and R. Beivide, "A comparison of router architectures for virtual cut-through and wormhole switching in a NOW environment," *Journal of Parallel and Distributed Computing*, vol. 61, no. 2, pp. 224-253, 2001.

[6] J. Duato, "A necessary and sufficient condition for deadlock-free routing in cut-through and store-and-forward networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 7, no. 8, pp. 841-854, Aug. 1996.

[7] J. Flich, P. Lopez, M. P. Malumbres, and J. Duato, "Boosting the performance of myrinet networks," *IEEE Trans. Parallel and Distributed Systems*, vol. 13, no. 12, pp. 693-709, 2002.

[8] J. Flich, S. Rodrigo, and J. Duato, "An efficient implementation of distributed routing algorithms for NoCs," in *Proc. of the Second Int. Symp. on Networks-on-Chip*, pp. 87-96, 2008.

[9] R. Horst, "TNet: A reliable system area network," *IEEE Micro*, vol. 15, no. 1, pp. 36-44, Feb. 1995.

[10] A. Jouraku, M. Koibuchi, and H. Amano, "An effective design of deadlock-free routing algorithms based on 2D turn model for irregular networks," *IEEE Trans. on Parallel and Distributed Systems*, vol. 18, no. 3, pp. 320-333, March 2007.

[11] R. Kesavan, K. Bondalapti, and D. K. Panda, "Multicast on irregular switch-based networks wormhole-routing," in *Proc. of High-Performance Computer Architecture*, pp. 48-57, 1997.

[12] M. Koibuchi, A. Journaku, and H. Amano, "Descending layers routing: A deadlock-free deterministic routing using virtual channels in system area networks with irregular topologies," in *Proc. of Int. Conference on Parallel processing*, pp. 527-536, 2003.

[13] L. Levitin, K. Karpovsky, and M. Mustafa, "Minimal sets of turns for breaking cycles in graphs modeling networks," *IEEE Trans. Parallel and Distributed Systems*, vol. 21, no. 9, pp. 1342-1953, 2010.

[14] S. Y. Lin, C. H. Huang, C. H. Chao, K. H. Huang, and A. Y. Wu, "Traffic-balanced routing algorithm for irregular mesh-based on-chip networks," *IEEE Trans. on Computers*, vol. 57, no. 9, pp. 1156-1168, Sept. 2008.

[15] W. Luo and D. Xiang, "An efficient adaptive deadlock-free routing algorithm for torus networks," *IEEE Trans. Parallel and Distributed Systems*, vol. 23, no. 5, pp. 800-808, May 2012.

[16] O. Lysne, T. Skeie, S. Reiemo, and I. Theiss, "Layered routing in irregular networks," *IEEE Trans. on Parallel and Distributed Systems*, vol. 17, no. 1, pp. 51-65, 2006.

[17] A. Mejia, M. Palesi, J. Flich, S. Kumar, P. Lopez, R. Holsmark, and J. Duato, "Region-based routing: A mechanism to support efficient routing algorithm in NoCs," *IEEE Trans. on VLSI Systems*, vol. 17, no. 3, pp. 356-369, March 2009.

[18] R. Moraveiji, H. Sarbazi-Azad, and A. Y. Zomaya, "Multispanning tree zone-ordered label-based routing algorithms for irregular networks," *IEEE Trans. on Parallel and Distributed Systems*, vol. 22, no. 5, pp. 817-832, May 2011.

[19] R. Perlman, "An algorithm for distributed computation of a spanning tree in an extended LAN," in *Proc. of 9th Data Communication Symp.*, pp. 44-53, 1985.

[20] V. Puente, R. Beivide, J. A. Gregorio, J. M. Prellezo, J. Duato and C. Izu, "Adaptive Bubble Router: a design to improve performance in torus networks," in *Proc. of Int. Conf. On Parallel Processing*, Japan, Sept. 1999.

[21] V. Puente, J. A. Gregorio, R. Beivide, F. Vallejo, and A. Ibanez, "A new routing mechanism for networks with irregular topology," in *Proc. of ACM/IEEE Int. Conf. on Supercomputing*, pp. 51-58, 2001.

[22] V. Puente, J. A. Gregorio, F. Vallejo, R. Beivide, and C. Izu, "High-performance adaptive routing for networks with arbitrary topology," *Journal of Systems Architecture*, vol. 52, pp. 345-358, 2006.

[23] W. Qiao, L. Ni, and T. Rokicki, "Adaptive-trail routing and performance evaluation in irregular networks using cut-through switches," *IEEE Trans. on Parallel and Distributed Systems*, vol. 10, no. 11, pp. 1138-1158, Nov. 1999.

[24] S. Rodrigo, J. Flich, A. Roca, S. Medardoni, D. Bertozzi, J. Camacho, F. Silla, and J. Duato, "Addressing manufacturing challenges with cost-efficient fault-tolerant routing," in *Proc. of 4th Int. Symp. on Networks-on-Chip*, pp. 25-32, 2010.

[25] M. D. Schroeder, A. Birrell, M. Burrows, H. Murray, R. Needham, and T. Rodeheffer, "Autonet: A high-speed, self-configuring local area network using point-to-point links," *IEEE Journal Selected Areas in Communication*, vol. 9, no. 8, pp. 1318-1355, Oct. 1991.

[26] F. Silla and J. Duato, "High-performance routing in networks of workstations with irregular topology," *IEEE Trans. Parallel and Distributed Systems*, vol. 11, no. 7, pp. 699-719, July 2000.

[27] D. Starobinski, M. Karpovsky, and L. A. Zabrevski, "Application of network calculus to general topologies using turn prohibition," *IEEE/ACM Trans. on Networking*, vol. 11, no. 3, pp. 411-421, June 2003.

[28] I. Theiss and O. Lysne, "FRoots: A fault-tolerant and topology-flexible routing technique," *IEEE Trans. on Parallel and Distributed Systems*, vol. 17, no. 10, pp. 1136-1150, 2006.

[29] J. Wu and L. Sheng, "Deadlock-free routing in irregular networks using prefix routing," in *Proc. of Int. Conf. Parallel and Distributed Systems*, pp. 424-430, 1999.

[30] J. Wu, "A fault-tolerant and deadlock-free routing protocol in 2D meshes based on odd-even turn model," *IEEE Trans. Computers*, vol. 52, no. 9, pp. 1154-1169, Sept. 2003.

[31] D. Xiang and J. Han, "Multiple spanning tree construction for deadlock-free adaptive routing in irregular networks," in *Proc. of IEEE Int. Symp. on Parallel and Distributed Processing with Application*, pp. 9-16, July 2012.

[32] M. Koibuchi, H. Matsutani, H. Amano, D. Frank Hsu, and He. Casanova. "A case for random shortcut topologies for HPC interconnects," in *Proc. of Int. Symp. on Computer Architecture*, pp. 177-188, 2012.

[33] S. Ma, N. Enright Jerger, and Z. Y. Wang . "Whole packet forwarding: Efficient design of fully adaptive routing algorithms for networks-on-chip," in *Proc of Int. Symp. on High-Performance Computer Architecture*,pp. 1-12, 2012.