Deadlock-free Routing Scheme for Irregular Mesh Topology NoCs with Oversized Regions

Xinming Duan, Jigang Wu
School of Computer Science and Software, Tianjin Polytechnic University, Tianjin, China
Email: xmduan@yahoo.cn, asjgwu@gmail.com

***Abstract*—At present, typical application-specific NoC systems often integrate a number of heterogeneous components which have varied functions, sizes and communication requirements. Instead of regular topology networks, constructing irregular mesh topology network on chip (NoCs) becomes an attractive approach to building future NoC systems with irregular structure. Deadlock-free routing control algorithm is a promising problem for irregular mesh topology. The available routing algorithms from regular mesh are not suitable for irregular mesh network. So in this paper, we introduce a hybrid scheme multiphase routing algorithm for irregular mesh integrating oversized rectangle modules. The basic idea of the scheme is borrowed from the area of fault tolerant networks, where a network topology is rendered irregular due to fault regions. The proposed scheme only employs 2 virtual channels per physical channel with fast routing decisions. In the case that the proposed two-phase routing scheme does not keep connection between some pairs of nodes, certain healthy nodes are deactivated to guarantee its deadlock-freeness. A greedy method is presented to ensure that only the minimum nodes are deactivated.**

***Index Terms*—network on chip, routing algorithm, deadlock-free, irregular mesh**

I. INTRODUCTION

Network on Chip (NoC) is a new important paradigm for implementing communication among various components in a single die [1]. Driven by the increasing application requirements, a single NoC system platform is often needed to incorporate components of different functions, sizes and communication requirements [2]-[5]. In such a network, the tile size should be able to accommodate the physically largest component, such as a shared memory. An effective method to solve the issue is to apply irregular mesh architectures combined with efficient routing algorithms [4]-[8]. Irregular mesh allows an oversized rectangular area in the mesh, larger than a tile, to be declared as a oversized cell. It turns out to be able to provide the required interconnect flexibility due to its adaptability for any component placements and its scalability inheriting from regular mesh.

In a NoC system with oversized components, routing of messages becomes more complex. Bigger rectangle cells formed by removing a certain amount of nodes and links make irregular mesh-based NoC available to accommodate oversized modules. In effect, a module acts as an obstacle to the network traffic. This not only results in higher packet latency, but deadlock free routing algorithms designed for regular mesh network are no more usable. At present, many efforts have been made to present a flexible and efficient routing algorithm for irregular mesh. Therefore, in this paper, a new routing algorithm for irregular mesh with oversized rectangle cells is proposed.

The rest of the paper is organized as follows. Section 2 introduces some definitions and theorem related to deadlock-free routing. Section 3 presents a two-phase routing algorithm for irregular mesh. Section 4 Summaries the paper.

II. DEFINITIONS AND THEOREM

Before we show that the proposed deadlock-free two-phase routing scheme, we first introduce some definitions and theorem related to the channel and channel class dependency [9]-[11].

Definition 1. An interconnection network is represented as a connected directed graph, $I = G(N, C)$, where N represents the set of nodes, and C represents the set of output channels of nodes.

Definition 2. Given an interconnection network I, a routing algorithm R and a pair of channels c_i, c_j, where $c_i, c_j \in C$ and $i \neq j$, if c_j will be used immediately after c_i on the route to any node $x \in N$, then there is a channel dependency from c_i to c_j, which is represented as $c_i \rightarrow c_j$.

Definition 3. The channel dependency graph D_c for a given interconnection network I and a routing algorithm R is a directed graph, $D_c = G(C, E_c)$, where C is the set of vertices representing the channels of I, and E_c represents the set of arcs connecting the channels which have channel dependency.

Lemma 1. Given an interconnection network I and a routing algorithm R, if the channel dependency graph D defined from I and R has no cycles, the routing algorithm R is a deadlock-free routing algorithm for I.

Lemma 1 is proved by Duato in [10].

Definition 4. Given a mesh network I, a routing function R and a pair of channel classes cc_g, $cc_h \in CC$, there is a channel class dependency from cc_g to cc_h which is

Corresponding author: Jigang Wu

doi:10.4304/jcp.8.1.27-32

represented as $cc_g \rightarrow cc_h$ if $\exists\, cc_g.c_i, cc_h.c_j \in C$, there is $cc_g.c_i \rightarrow cc_h.c_j$.

Definition 5. Given a mesh network *I* and a routing function R, channel class dependency graph D_{cc} of *R* is a directed graph, D_{cc}=G(*CC*, E_{cc}). The vertices of graph *CC* are the channel classes of *I*. The arcs of graph E_{cc} are the pairs of channel classes $<cc_g, cc_h>$ such that there is $cc_g \rightarrow cc_h$.

Definition 6. One direction function *DR*: $CC \rightarrow P$(x+, x-, y+, y-), where *P*(x+, x-, y+ ,y-) is the power set of x+, x-, y+, y-. *DR* returns the directions of channel classes belonging to the sets of channel class.

Theorem 1. Given a mesh network I and a connected routing function R, R is deadlock-free if

$\forall$ a strongly connected branch $D_{cc1}=G(CC_1, E_{cc1})$ where $D_{cc1} \subseteq D_{cc}$, D_{cc} is the channel class dependency graph for R

such that $DR(CC_1) \neq \{x+, x-, y+, y-\}$.

Proof: Theorem 1 is proved in [11].

III. Two-Phase Fault Tolerance Routing

In the area of fault tolerant routing, routing algorithms based on two-phase routing scheme have been presented [12][13]. The scheme avoids faulty nodes by selecting an intermediate node. In this paper, we extend the scheme to the irregular mesh topology NoC systems.

A. Basic Ideas

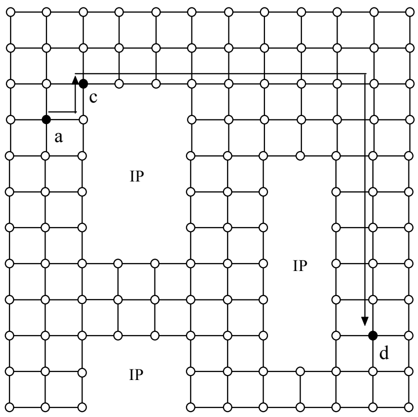

Figure 1. An irregular mesh with 3 oversized IP modules.

Since irregular mesh with oversized cells is used to connect various modules of application-specific NoCs, certain nodes are not able to reach other nodes by the available routing algorithms. So we present a two-phase routing scheme to solve the issue. The presented scheme uses available routing algorithms as basic routing algorithms. It is in fact a combination of two basic routing algorithms. The routing path from source node to a destination node is split into two phases. In the first phase, messages are routed from a source node to an intermediate node, and then from the intermediate node to a destination node in the second phase. For example, as shown in figure 1, a message uses X-Y routing algorithm to routes from a source node 'a' to a intermediate node 'c' in the first phase, and then also uses X-Y routing algorithm to routes from the intermediate node 'c' to a destination node 'd'.

It is assumed that the node 'c' do not store the whole message after a message reached it at the end of the first routing phase. It simply forwards the flits of the message in a pipelined fashion. Since a message can reserve a sequence of nodes, it is possible that the head flit of a message reserve nodes between 'c' and 'd', while its tail flit still reserve nodes between 'a' and 'c'. Furthermore, if we use different virtual channel during each of the routing phase, the routing algorithm is guaranteed to be deadlock-free.

B. Two-Phase X-Y Routing Algorithm

In this section, we introduce a new fault-tolerant routing algorithm based on the two-phase scheme. The presented fault tolerant routing algorithm denoted as two-phase *X-Y* routing algorithm is described as follows, where x_c, y_c represent x, y coordinates of the current node. x_m, y_m represent x, y coordinates of the intermediate node. x_d, y_d represent x, y coordinates of the destination node. We use one bit "current_phase" to indicate if a message uses the first phase (0) or the second phase (1) to route. x_{offs}, y_{offs} represent the offsets of x, y coordinate between the current node and the intermediate node (current_phase=0), or between the current node and the destination node (current_phase=1). Each physical channel of x or y dimension is split into 2 virtual channels $-X_0$, $+X_1$ or $-Y_0$, $+Y_1$.

Algorithm two-phase *X-Y* routing (x_c,y_c):
begin
  if current_phase=0 **then**
    $x_{offs} \leftarrow x_m - x_c$
    $y_{offs} \leftarrow y_m - y_c$
    if x_{offs}=0 and y_{offs}=0 **then**
      current_phase=1
    endif
  endif
  if current_phase=1 **then**
    $x_{offs} \leftarrow x_d - x_c$
    $y_{offs} \leftarrow y_d - y_c$
    if x_{offs}=0 and y_{offs}=0 **then**
      chn←internal
      return
    endif
  endif
  if x_{offs}<0 **then**
    chn←$-X_{current\_phase}$
  endif
  if x_{offs}>0 **then**
    chn←$+X_{current\_phase}$
  endif
  if x_{offs}=0 **and** y_{offs}<0 **then**
    chn←$-Y_{current\_phase}$
  endif
  if x_{offs}=0 **and** y_{offs}>0 **then**
    chn←$+Y_{current\_phase}$
  endif
end

According to the *two-phase X-Y* routing algorithm for irregular mesh-based NoCs, messages round oversized components through two phases. Messages are typed as the first phase message (current_phase=0) or the second phase message (current_phase=1). Each message is initially the first phase message (0). The first phase messages are routed through the first basic *X-Y* routing function. The first phase messages will not become the second phase messages until the offsets between the current node and the intermediate node equal to 0. The second phase messages are routed through the second basic X-Y routing function until they arrive at their destinations.

Theorem 2 *two-phase X-Y* algorithm is deadlock-free.

Proof:

The channel class dependency graph of basic *X-Y* algorithm $D_{cc}=G(CC, E_{cc})$ is shown in Figure 2(a), where verticals represent the channel classes -x, +x, -y and +y. Directed arcs between channel classes represent that message can be routed from one channel class to another.

The channel class dependency graph of two-phase *X-Y* algorithm D_{cc}=G(CC, E_{cc}) is shown in Figure 2(b).

All the strongly connected branch of channel class dependency graph D_{cc} is shown in Figure 2(c).

The channel classes of 8 strongly connected branch of D_{cc} are respectively $CC_1=\{-x_0\}$, $CC_2=\{+x_0\}$, $CC_3=\{-y_0\}$, $CC_4=\{+y_0\}$, $CC_5=\{-x_1\}$, $CC_6=\{+x_1\}$, $CC_7=\{-y_1\}$ and $CC_8=\{+y_1\}$. The directions contained by them are $DR(CC_1)= DR(CC_5)=\{-x\}$, $DR(CC_2)=DR(CC_6)=\{+x\}$, $DR(CC_3)= DR(CC_7)=\{-y\}$ and $DR(CC_4)= DR(CC_8)=\{+y\}$.

$\forall$ a strongly connected branch $D_{cci}=G(CC_i, E_{cci})$ such that $\{+x, -x, +y, -y\} \neq DR(CC_i)$, where i=1, ⋯, 8.

According to theorem 1, theorem 2 is completed.

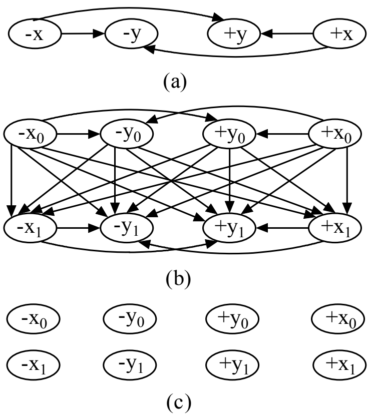

Figure 2. The channel class dependency graph of two-phase Routing.

The two-phase *X-Y* routing algorithm can also be extended to a three-phase routing algorithm. The three-phase *X-Y* routing algorithm using 3 virtual channels can reach more nodes than the two-phase *X-Y* routing algorithm using 2 virtual channels. But it still cannot be sure to keep its connection, as the number of regions increases. The routing algorithm proposed in [11] employs only 3 channels while keep its connection despite the number of the regions in irregular mesh. The 3-phase routing algorithm is not acceptable, since the virtual channels employed by it add up to 3. Some nodes of irregular mesh need to be deactivated to guarantee the connection and deadlock-freeness.

C. Selection of the Intermediate Nodes

The original *X-Y* routing algorithm may not route messages to reach its destination because of regions which act as obstacles to the network traffic. More nodes of irregular mesh may be reached through the proposed two-phase *X-Y* routing algorithm. The two-phase *X-Y* routing algorithm selects an intermediate node between a source node and a destination node. Before we show how to find the intermediate nodes between any pairs of nodes, we first introduce some definitions and formula.

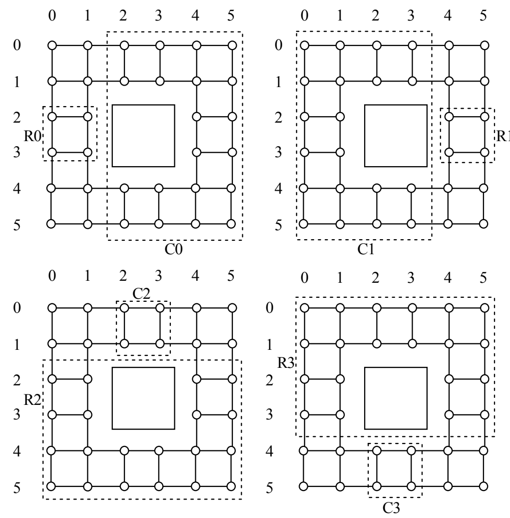

Figure 3. Eight X-zones of irregular mesh with three IP modules.

Definition 7. The rectangular area of mesh is called a zone of (x_1, y_1, x_2, y_2), where x_1,y_1 represent the upper left corner coordinates of the rectangular area and x_2, y_2 represent the lower right corner coordinates of it. The whole mesh of n×n can also be called a zone of (0, 0, n-1, n-1).

Definition 8. Given a irregular mesh of n × n with rectangular regions, for each region of (x_1, y_1, x_2, y_2), there is 4 pairs of zones for each region (*R0*, *C0*), (*R1*, *C1*), (*R2*, *C2*) and (*R3*, *C3*), where *R0* is a zone of (0, y_1+1, x_1, y_2-1), *C0* is a zone of (x_1+1, 0, n-1, n-1), *R1* is a zone of (x_2, y_1+1, n-1, y_2-1), *C1* is a zone of (0, 0, x_2-1, n-1), *R2* is a zone of (0, y_1+1, n-1, n-1), *C2* is a zone of (x_1+1, 0, x_2-1, y_1), *R3* is a zone of (0, 0, n-1, y_2-1) and *C3* is a zone of (x_1+1, y, x_2-1, n-1).

For example, figure 3 shows a mesh of 6×6 and a region of (1, 1, 4, 4). There are 4 pairs of zones which are denoted as (*R0*, *C0*), (*R1*, *C1*), (*R2*, *C2*) and (*R3*, *C3*). As shown in figure 3, *R0* is a zone of (0, 2, 3, 3), *C0* is a zone of (2, 0, 5, 5), *R1* is a zone of (4, 2, 5, 3), *C1* is a zone of (0, 0, 3, 5), *R2* is a zone of (0, 2, 5, 5), *C2* is a zone of (2, 0, 3, 1), *R3* is a zone of (0, 0, 5, 3) and *C3* is a zone of (2, 4, 3, 5).

It is easy to see that messages from the zones *R0*, *R1*, *R2* or *R3* cannot reach nodes in the zones *C0*, *C1*, *C2* or *C3* respectively through the original *X-Y* routing algorithm, since the region is larger than a tile of mesh. In order to route messages around the region, the proposed two-phase *X-Y* routing algorithm routes messages from the source node to an intermediate node in the first phase, then from the intermediate node to the destination node in the second phase. So the first step is to find a intermediate node for each pair of node between *Ri* and *Ci* shown in figure 3 before the two-phase *X-Y* routing algorithm can work, where i=0, …, 3. We use the following approach.

Step 1: Calculate the number of 1-round hops for each pair of nodes. If the two nodes is connected through 1-round *X-Y* routing algorithm, the number of 1-round hops between them equals to their total offset of dimension X and Y, otherwise, it equal to ∞.

Step 2: Calculate the number of two-phase hops for each pair of nodes through each intermediate node. The number of two-phase hops between the source node and destination node through a intermediate node equal to the number of hops between the source node and intermediate node, plus the number of hops between the intermediate node and destination node.

Step 3: For each pair of nodes, select a node as the intermediate node of them through which the number of hops between them is minimum.

TABLE I.
HOPS FROM THE SOURCE NODE (1, 2) TO THE DESTINATION NODE (X, Y)

x, y	hops	x, y	hops	x, y	hops	x, y	hops
0, 0	3	1, 3	1	3, 0	∞	4, 3	∞
0, 1	2	1, 4	2	3, 1	∞	4, 4	∞
0, 2	1	1, 5	3	3, 2	∞	4, 5	∞
0, 3	2	2, 0	∞	3, 3	∞	5, 0	∞
0, 4	3	2, 1	∞	3, 4	∞	5, 1	∞
0, 5	4	2, 2	∞	3, 5	∞	5, 2	∞
1, 0	2	2, 3	∞	4, 0	∞	5, 3	∞
1, 1	1	2, 4	∞	4, 1	∞	5, 4	∞
1, 2	0	2, 5	∞	4, 2	∞	5, 5	∞

TABLE II.
HOPS FROM THE SOURCE NODE (1, 2) TO THE DESTINATION NODE (5, 3) THROUGH THE INTERMEDIATE NODE (X, Y)

x, y	hops	x, y	hops	x, y	hops	x, y	hops
0, 0	11	1, 3	∞	3, 0	∞	4, 3	∞
0, 1	9	1, 4	7	3, 1	∞	4, 4	∞
0, 2	∞	1, 5	9	3, 2	∞	4, 5	∞
0, 3	∞	2, 0	∞	3, 3	∞	5, 0	∞
0, 4	9	2, 1	∞	3, 4	∞	5, 1	∞
0, 5	11	2, 2	∞	3, 5	∞	5, 2	∞
1, 0	9	2, 3	∞	4, 0	∞	5, 3	∞
1, 1	7	2, 4	∞	4, 1	∞	5, 4	∞
1, 2	∞	2, 5	∞	4, 2	∞	5, 5	∞

Take the selection of the intermediate node between the source node (1, 2) and the destination node (5, 3) for example. On step 1 the number of 1-round hops for each pair of nodes is calculated. Table 1 shows the number of hops from the source node (1, 2) to any destination node (x, y). We can see from it that the number of 1-round hops from node (1, 2) to node (5, 3) equals to ∞. That means messages from node (1, 2) cannot reach node (5, 3) through 1-round X-Y routing. On step 2 the number of two-phase hops for each pair of nodes through each intermediate node is calculated. Table 2 shows the number of hops from the source node (1, 2) to the destination node (5, 3) through each intermediate node. We can see from it that the number of two-phase hops from node (1, 2) to node (5, 3) through node (1, 1) or (1, 4) is minimal among all intermediate nodes. We can select one of them as the intermediate node for routing messages from node (1, 2) to node (5, 3).

D. *Connection of Two-phase X-Y Routing Algorithm*

The proposed two-phase *X-Y* routing algorithm may not keep its connection, if the number of integrated oversized IP modules increases. That is some nodes of irregular mesh may not be reached by two-phase *X-Y* routing algorithm. Certain nodes of irregular mesh need to be deactivated to guarantee the connection and deadlock-freeness.

Before minimizing the number of deactivated nodes of irregular mesh, we first introduce definitions and formula to calculate the size of deactivated nodes.

It is easy to see that each of irregular mesh as well as IP modules incident has exactly 4 borders, such as up border, bottom border, left border, and right border.

Definition 9. *X-zones* can be achieved through the following steps. Extending the up border line and bottom border line of each IP modules until it touch the left border or right border of irregular mesh or other IP modules.

For example, figure 3 shows all *X-zones* of irregular mesh shown in figure 1, eight *X-zones* R1, ⋯, R8 are achieved by extending the up and bottom border lines of three IP modules.

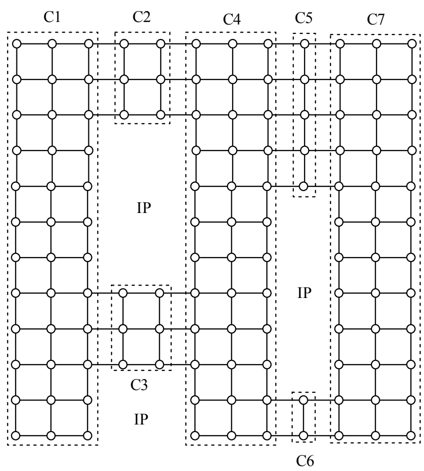

Figure 4. Eight *X-zones* of irregular mesh with three IP modules.

Definition 10. *Y-zones* can be achieved through the following steps. Extending the left border line and right border line of each IP modules until it touch the up border or bottom border of irregular mesh or other IP modules.

For example, figure 4 shows all *Y-zones* of irregular mesh shown in figure 1, seven *Y-zones* C1, ⋯, C7 are achieved by extending the left and right border lines of three IP modules.

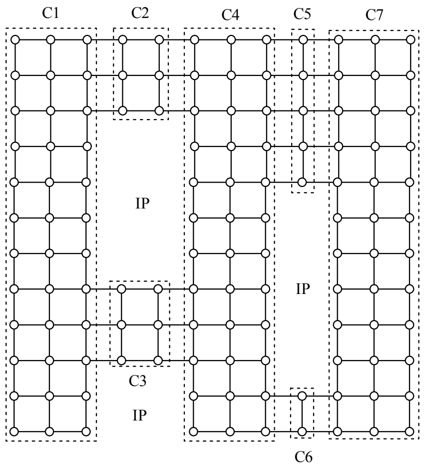

Figure 5. Seven *Y-zones* of irregular mesh with three IP modules.

Definition 11. Let *S1* and *S2* be Boolean matrix, matrix element $S1_{i,j} \in S2$, $S2_{i,j} \in S2$, where i represents the index of *X-zones*, j represents the index of *Y-zones*. $S1_{i,j}$ or $S2_{i,j}$ represents that messages could be routed from an *X-zone* R_i to a Y-zone C_j through the first or second *X-Y* routing algorithm ($S1_{i,j}$=1 or $S2_{i,j}$=1) or not ($S1_{i,j}$=0 and $S2_{i,j}$=0).

Definition 12. Let *D1* be a Boolean matrix, matrix element $D1_{j,i} \in D1$, where j represents the index of *Y-zones*, i represents the index of *X-zones*. $D1_{j,i}$ represents that message could transmit from a *Y-zone* C_j in its first routing phase to an *X-zone* R_i in its second routing phase ($D1_{i,j}$=1) or not ($D1_{j,i}$=0).

For example, Boolean matrix *S1*, *S2* and *D1* of irregular mesh shown in figure 4 and 5 are followed.

$$S1 = S2 = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$

$$D1 = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

Definition 13. Let *T* be a Boolean matrix, matrix element $T_{i,j} \in T$, where i represents the index of *X-zones*, j represents the index of *Y-zones*. $T_{i,j}$ represents that messages could be routed from an *X-zone* R_i to a *Y-zone* C_j through the two-phase *X-Y* routing algorithm ($T_{i,j}$=1) or not ($T_{i,j}$=0).

Formula 1. $T = S1 \bullet D1 \bullet S2$

Proof. It is easy to see that R_i can reach C_j through two-phase *X-Y* routing algorithm, if there is a *Y-zone* C_{j1} and an *X-zone* R_{i1} such that R_i can reach C_{j1} through the first *X-Y* routing algorithm, C_{j1} of the first phase can reach R_{i1} of the second phase and R_{i1} can reach C_j through the second *X-Y* routing algorithm. That is Boolean matrix $S1_{i,j1}$=1, $D1_{j1,i1}$=1, $S2_{i1,j}$=1.

Boolean matrix *T* computed by formula 1 is followed.

$$T = S1 \bullet D1 \bullet S2 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

TABLE III.
X-ZONE OR Y-ZONE THAT CANNOT REACH

X-zone	size	*Y-zone*	size
R2	12	C6	2
R7	6	C6	2
R5	15	C3	6

We can see from Boolean matrix *T* that *R2* cannot reach *C6*, *R5* cannot reach *C3* and *R7* cannot reach C6 through two-phase X-Y routing algorithm. Thus, either R_i or C_j in Table I is deactivated to keep the connection of irregular mesh. Our goal is now to select at least one of *X-zone* and *Y-zone* in each row of Table I while minimizing the size of selected zones. A greedy method is presented as follows. A zone is isolated if another zone in the same row is selected. We select a zone which makes the maximum sizes of zones are isolated. This step is repeated till all zones in Table I are either selected or isolated. In this example, *C3* and *C6* are selected as deactivated nodes. That is nodes of *C3* and *C6* can not be the destination in irregular mesh.

IV. CONCLUSION

As a uniform, scalable solution, irregular mesh network is proposed to meet the variety of NoC components placement. In irregular mesh network, messages have to misroute oversized modules. Thus routing algorithms for irregular mesh network employ more virtual channels than that in regular mesh [8]. In this paper, we introduce a hybrid scheme multiphase routing algorithm for irregular mesh. The proposed routing algorithm only employs 2 virtual channels. Fewer channels correspond to fewer buffers, which is desirable for router systems as buffers consume the largest proportion of the area and power in router systems [14][15]. Furthermore, due to its relatively low router intelligence, the proposed routing algorithm takes routing decisions fast. In the case that some nodes cannot reach by the routing algorithm, the minimum size of nodes is deactivated such that they cannot be the source (*X-zones*) or the destination (*Y-zones*). Therefore, the proposed algorithm is applicable to the application-specific NoC systems with a few oversized components.

ACKNOWLEDGMENT

This work was supported in part by the NSF of China under Grant No. 60970016.

REFERENCES

[1] L. Benini and G.D. Micheli, "Networks on chips: a new SoC paradigm," IEEE Computer, vol. 35, no. 1, pp. 70-78, Jan. 2002.
[2] S. Kumar, A. Jantsch, J-P. Soininen, "A network on chip architecture and design methodology," Proc. IEEE Annual Symposium on VLSI, pp. 105–112, April, 2002.
[3] T. Hollstein, R. Ludewig, C. Mager. "A hierarchical generic approach for onchip communication, testing and debugging of SoCs". Proc. of the VLSI-SoC2003, pp. 44–49, Dec. 2003
[4] Xinming Duan, Yulu Yang, Mei Yang, "Topology and Binary Routing Schemes of A PRDT-Based NoC " Proc. IEEE ITNG, pp. 920-924, 2007.
[5] Martin K.-F. Schafer, Thomas Hollstein, "Deadlock-Free Routing and component Placement for Irregular Mesh-based Networks-on-Chip," Proceedings of the 2005 IEEE/ACM International conference on Computer-aided design, pp. 238-245 , Nov. 2005
[6] R. Holsmark, and S. Kumar, Corrections to Chen and Chiu's Fault Tolerant Routing Algorithm for Mesh Networks", J. of Information Science and Engineering, Vol.23, pp.1649-1662, 2007.
[7] R. Holsmark, M. Palesi, and S. Kumar, Deadlock free routing algorithms for irregular mesh topology NoC systems with rectangular regions", J. system architecture, Vol.54, Issue 3-4, pp.427-440, 2008.
[8] Xinming Duan, Dakun Zhang, Xuemei Sun, "Routing Schemes of An Irregular Mesh-based NoC," Proc. of the NSWCTC, pp. 547-553, April, 2009.
[9] W.J. Dally, C.L. Seitz, "Deadlock-free message routing in multiprocessor interconnection networks," IEEE Trans. Computers, vol. 36, no. 5, pp. 547-553, May 1987.
[10] J. Duato, "A necessary and sufficient condition for deadlock-free adaptive routing in wormhole networks," IEEE Trans. on Parallel and Distributed Systems, vol. 6, no. 10, pp. 1055–1067, Oct. 1995.
[11] Xinming Duan, Dakun Zhang, "Fault-tolerant routing schemes for wormhole mesh," Proc. Of the ISPA2009, pp.278-287,2009.
[12] M. E. Gomez, N. A. Nordbotten, J. Flich, P. Lopez, A. Robles, J. Duato, "A routing methodology for achieving fault tolerance in direct networks," IEEE Trans. on Computers, vol. 55, no. 4, pp. 400-415, 2006.
[13] C.-T. Ho, L. Stockmeyer, "A new approach to fault-tolerant wormhole routing for mesh-connected parallel computers," IEEE Trans. on Computer, vol. 53, no. 4, pp. 427–438, april, 2004.
[14] R. Libeskind-Hadas, "A Tight Lower Bound on the Number of Channels Required for Deadlock-Free Wormhole Routing," IEEE Trans. Computers, vol. 47, no. 10, pp. 1158-1160, Oct. 1998.
[15] N. Banerjee, P. Vellanki, and K.S. Chatha, "A power and performance model for network-on-chip architectures," Proc. Of the DATE2004 Conf., vol. 2, pp. 1250-1255, 2004.

Xinming Duan received the BS degree in computer science from Tianjin normal University, China, in 1993 and the PhD degree in computer engineering from the Nankai University of China in 2007. He is an associate professor at Tianjin Polytechnic University, China. His research is focused towards specialized architectures and routing algorithms for Networks on Chip. Other areas of interest include fault-tolerant computing, parallel/distributed computing, and interconnection networks.

Jigang Wu received the BS degree in computational mathematics from Lanzhou University, China, in 1983 and the PhD degree in computer software and theory from the University of Science and Technology of China. He is successively a professor at Tianjin Polytechnic University, China. His research interests include algorithms in reconfigurable VLSI design, hardware/software codesign, parallel computing, and combinatorial search.