An Improved Mesh Topology and Its Routing Algorithm for NoC

Yang Quansheng
School of Computer Science & Engineering
Southeast University
Nanjing, China
yangqs@seu.edu.cn

Wu Zhekai
School of Computer Science & Engineering
Southeast University
Nanjing, China
zhekaiwu1985@yahoo.com.cn

***Abstract*—Network on chip (NoC) is an effective solution to complex on-chip communication problem. The mesh topology is one of the most popular NoC. It has completely regular topology which can be implemented easily, but the communication delay between remote nodes is large. In this paper, we propose an improved topology called Tmesh, which is based on the standard mesh network by inserting four long links. We also present a deadlock-free routing algorithm for Tmesh named TXY algorithm. Indeed, our experimental results demonstrate a certain reduction in the average packet delay and routing hops.**

Keywords-mesh; routing algorithm; NoC

I. INTRODUCTION

Continuous scaling of CMOS technology makes it possible to integrate a large number of cores on a chip multiprocessor (CMP) system. As a result, inter-core connection and communication becomes an important factor that affecting processor performance. Now there are three interconnection manners for CMP, which are bus, cross-switch and NoC [1-5]. When the number of cores is small, bus and cross-switch can satisfy the communication requirements. As the number of cores increasing, bus can not effectively deal with multiple data stream's concurrent transmission, and cross-switch causes exponential growth of hardware overhead. NoC becomes an effective solution to the interconnection of many cores.

Compared with bus and cross-switch, NoC is based on macro computer network and has better reusability and scalability. At present, much research is based on mesh, torus topology, especially the mesh. Mesh is easy for physical implement and routing. It has been applied in practical design, such as Godson-3[5], TILE64 Processor [6]. However, it has large communication delay between remote nodes, increasing the possibility of network congestion [7]. Ogras proposed a method to optimize the performance of NoC [8], by inserting a few additional long-range links in mesh topology to reduce the communication delay between some nodes.

This paper presents an improved mesh topology called Tmesh based on the above idea. We also give a routing algorithm named TXY algorithm. The experimental results demonstrate a certain reduction in the average packet delay and routing hops. When the network has 8×8 nodes, the average delay and routing hops of Tmesh are 2.92% and 3.53% lower than those of mesh respectively.

This paper is organized as follows. In Section II, we describe Tmesh topology, and compare it with mesh from the network diameter, the ideal throughput and the ideal average delay. In Section III, present a routing algorithm for Tmesh and discuss the strategy of deadlock avoidance. In Section IV, the simulation environment and the performance analysis is given. Section V concludes the paper and indicates possible directions for future work.

II. TMESH TOPOLOGY

A. The basic architecture

An $n \times n$ mesh consists of N ($n \times n$) nodes, where each node has an associated integer coordinate pair (x, y) such that $0 \le x < n$ and $0 \le y < n$ [9]. Two nodes with coordinates (x_i, y_i) and (x_j, y_j) are connected by a communication channel if and only if their distance is equal to one, which means $|x_i - x_j| + |y_i - y_j| = 1$. Fig. 1(a) shows the mesh topology with 16 nodes. Mesh has no overlap links and is easy for physical implementation, but communication delay between remote nodes is large. For example (see Fig. 1(a)), the distance between (0, 0) and (3, 3) is six.

Before describing the Tmesh topology, we define several important concepts listed as follows

Definition1. In an $n \times n$ Tmesh, a node is called *vertex* if its coordinate pair is belong to the set {(0, 0), (0, n-1), (n-1, 0), (n-1, n-1)}.

Definition2. The edge connects *vertices* named *long link*.

Definition3. An $n \times n$ Tmesh is divided into four areas called *Area 0*, *Area 1*, *Area 2*, and *Area 3*, each area contains one *vertex*. Their regions are set as

Area 0: $0 \leqslant x \leqslant (n-1)/2$, $0 \leqslant y \leqslant (n-1)/2$
Area 1: $(n-1)/2 < x < n$, $0 \leqslant y \leqslant (n-1)/2$
Area 2: $(n-1)/2 < x < n$, $(n-1)/2 < y < n$
Area 3: $0 \leqslant x \leqslant (n-1)/2$, $(n-1)/2 < y < n$

Definition4. In Tmesh, the distance between two nodes S and D is called *L(S, D)*. Let V be the *vertex* of the area containing S. Let T be the *vertex* of the area containing D. If the *long link* can reduce the distance, $L(S, D) = | x_v - x_s | + | y_v - y_s | + | x_t - x_d |$

Fund Projects: Support by Ministry of Education – Intel Course Construction (No. MOE-INTEL-Course-09-09) and Natural Science Foundation of Jiangsu Province (No.BK2007513)

$+ \mid y_t - y_d \mid + m$. Otherwise, $L(S, D) = \mid x_s - x_d \mid + \mid y_s - y_d \mid$. Here, m is equal to two if the combination of the areas which S and D in are *(Area0, Area2)*, *(Area2, Area0)*, *(Area1, Area3)*, *(Area3, Area1)*. Otherwise m is equal to one.

Tmesh is based on mesh and inserts four *long links* to connect the *vertices* to reduce the communication delay between some remote nodes. Fig. 1(b) shows the mesh topology with 16 nodes. The distance between (0, 0) and (3, 3) is two.

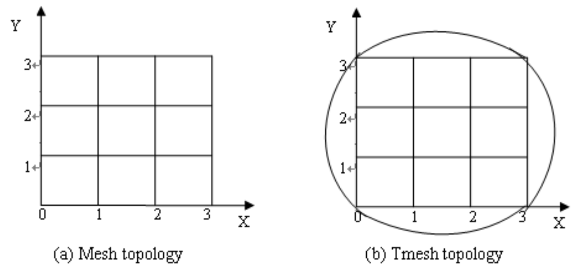

Figure 1. Mesh and Tmesh topology

B. Comparision

We make a comparison between Tmesh and mesh from three aspects, the network diameter, the ideal average delay and the ideal throughput. We give the details as follows.

- Network diameter

Network diameter means the maximum network distance between two arbitrary nodes [10]. Typically, to improve the quality and speed of network transmission, it needs to reduce the network diameter [11].

In an n×n network, the network diameter of mesh is 2(n-1) [12], which is the distance between two nodes at diagonal end. For Tmesh, if n is an odd number, the network diameter is (n-1), otherwise is n. So the network diameter of Tmesh is smaller than that of mesh. Here we give a brief explanation.

Let d be the network diameter. When n is odd, d is equal to the distance between (0, 0) and ((n-1)/2, (n-1)/2). When n is even, d is equal to the distance between (n/2-1, n/2-1) and (n-1, n-1).

Fig. 2 shows the network diameter of the Tmesh with 5×5 and 4×4 nodes. We don't mark four *long links*.

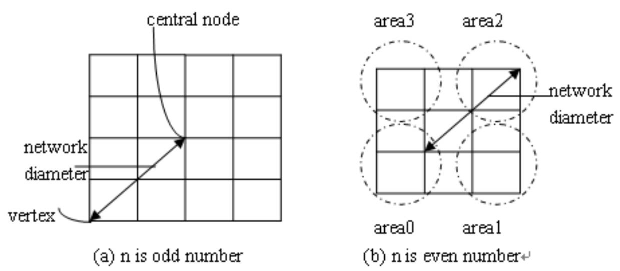

Figure 2. Network diameter of 5×5 and 4×4 network

- The ideal average delay

The ideal average delay is defined as the average routing delay time between all nodes when there is no congestion in network. Formula 1[13] gives the method to calculate the ideal average delay.

$$T = H \times Tr + D/v + L/b \quad (1)$$

H is the average hops from the source node to destination, Tr is the delay of the routing switch, D is the average distance from the source to the destination, usually, H is equal to D, v is the transmission rate on-line, L is the packet length and b is the bandwidth. For some kinds of topology, calculating the average delay is to compute H or D mainly. The average distance of mesh is equal to $2(n^2-1)/3n$ [14]. When n is equal to 4, D_{mesh} is equal to 2.5.

Tmesh has four more *long links* than mesh, so the average distance is less than mesh's. Here we take the network with 16 nodes as an example (see Fig. 1(b)). As Tmesh topology is symmetrical, we can divide all nodes into three classes according to their coordinates. For each class, we only need to select one representation as the source node, and then calculate its distance to other nodes, and finally multiplied by the number of nodes within this class. Table 1 shows the distance from one node of the class to others.

TABLE I. DISTANCE FROM ONE NODE OF THE CLASS TO OTHERS

Source nodes	(0,0)	(1,0)	(1,1)
Nodes from which distance equals 1	(1,0), (0,1), (3,0), (0,3)	(0,0), (1,1), (2,0)	(1,0), (0,1), (1,2), (2,1)
Nodes from which distance equals 2	(0,2),(2,0), (3,1),(1,3), (1,1), (3,3)	(0,1), (2,1),(1,2), (3,0), (0,3)	(0,0),(2,2), (2,0),(0,2), (3,1), (1,3)
Nodes from which distance equals 3	(1,2), (2,1), (2,3), (3,2)	(1,3), (3,1), (2,2),(3,3),(0,2)	(3,0), (0,3), (2,3), (3,2)
Nodes from which distance equals 4	(2,2)	(2,3), (3,2)	(3,3)

The total distance from one node of the class to others shows as follows

$$D(0,0) = 1 \times 4 + 2 \times 6 + 3 \times 4 + 4 \times 1 = 32$$

$$D(1,0) = 1 \times 3 + 2 \times 5 + 3 \times 5 + 4 \times 2 = 36$$

$$D(1,1) = 1 \times 4 + 2 \times 6 + 3 \times 4 + 4 \times 1 = 32$$

In Fig. 1(b), the number of nodes in each class is 4, 8, and 4 respectively. So

$$D_{Tmesh} = (D(0,0) \times 4 + D(1,0) \times 8 + D(1,1) \times 4)/(16 \times 16) = 2.125$$

- The ideal throughput

The ideal throughput means the maximum throughput of the network in a perfect flow control and routing mechanism. Formula 2[13] gives the method to calculate the ideal throughput. TP is the throughput, b is the bandwidth of each channel, Bc is the number of links required to divide the whole network into two equal half, and N is the total number of the nodes.

$$TP \leqslant 2b \times Bc/N \quad (2)$$

For an n×n mesh, we need n links to divide the mesh into two halves. Each link is two-way, so Bc is equal to 2n. While for an n×n Tmesh, we need (n+2) links to do that, so Bc is equal to 2(n+2). Now we can give the ideal throughputs of mesh and Tmesh. $TP_{mesh} \leqslant 4b/n$, $TP_{tmesh} \leqslant 4(n+2)b/n^2$. Obviously, Tmesh's throughput is a little larger than mesh's.

III. ROUTING ALGORITHM

A. The basic idea

For the regular topology, a simple arithmetic algorithm is enough, such as Dimension Ordered Routing which determines how the packet flows through the network according to the coordinates of the communication channel. For mesh topology, it is called XY algorithm. The packet firstly arrives at the column where the destination is, then at the row. Let (x_c, y_c), (x_d, y_d) be the coordinates of the current and destination node. Let move right, left, up and down denote the directions +x, –x, +y, –y. Fig. 3(a) gives the pseudo codes of the XY algorithm.

```
XY(C, D)
C: current node (xc, yc)
D: destination node (xd, yd)
IF (|xc – xd |! = 0)
{
   IF (xc < xd) move right
   ELSE move left
}
ELSE IF (yc < yd) move up
ELSE IF (yc > yd) move down
ELSE arrive at the destination
```

(a) XY algorithm

```
TXY(S, D, C)
C: current node (xc , yc)
D: destination node (xd , yd)
S: source node (xs ,ys)
Computer L0, L1
IF (C is not vertex)
    XY(C, D)
ELSE IF (L1 < L0)
    move along the long link
ELSE
    XY(C, D)
```

(b) TXY algorithm

Figure 3. The pseudo codes of XY and TXY algorithm

For Tmesh, the shortest path algorithm is not suitable, which easily leads to the congestion at the four *vertices*. A lot of the shortest paths need to pass through the four *vertices*. So the algorithm Tmesh uses is a deterministic non-shortest path routing algorithm. Although the path TXY gets may not be the shortest one, it will not be longer than the path XY gets.

The basic idea of TXY algorithm is as follows: When the packet arrives at one node, we check whether it is a *vertex*. If not, the default XY algorithm is used. If so, the distance to the destination with and without the *long link* will be computed. If the *long link* produces a shorter distance to the destination, the packet will move along the *long link*. Otherwise, XY algorithm is used. Fig. 3(b) gives the pseudo codes of the TXY routing algorithm.

B. The details

Let n×n be the number of nodes in the network. Let S, D and C be the source, destination and current node whose coordinates is (x_s, y_s), (x_d, y_d) and (x_c, y_c) respectively. Let L0, L1 be the distance to the destination without and with *long link*. The keys of the TXY routing algorithm are to computer L0 and L1 and select the *long link*.

Firstly, we give the method to computer L0 and L1. Let T be the *vertex* of the area where D is, whose coordinate is (x_t, y_t). We computer L0, L1 as follows

$$L0 = | x_c – x_d| + |y_c – y_d|$$

$$L1 = | x_t – x_d | + | y_t – y_d | + m$$

The definition of area and m are defined in *Definition 3* and *Definition 4*. C is the *vertex*, so the computation expression of L is a little different from the *Definition 4*.

Secondly, we give the method to select the *long link*. Every *vertex* is connected to two *long links*. If C and T are in the same column, the right *long link* or left *long link* is selected. Otherwise, the top *long link* or bottom *long link* is selected. For example, if the coordinates of C and T are (0, 0), (0, n-1) respectively, they are in the same column and connected by right *long link*. So the right *long link* is selected.

C. Deadlock avoidance

Deadlock occurs when a packet waits for an event that cannot happen. For example, a packet may wait for a network to be released by a packet that is, in turn, waiting for the first packet to release some resource [15]. One efficiency strategy of deadlock avoidance is to break circular wait. For mesh, there are eight possible turns which will form two cycles. By prohibiting same of the turns, we can avoid deadlock. West-First, North-Last, Negative-First [15] prohibit two kinds of turns.

To simplify the TXY algorithm, we prohibit four turns (see Fig. 4). The virtual turn is illegal, while the solid turn is legal.

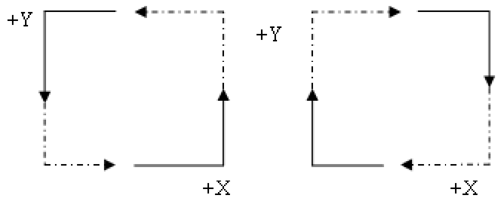

Figure 4. Four allowed turns in Tmesh

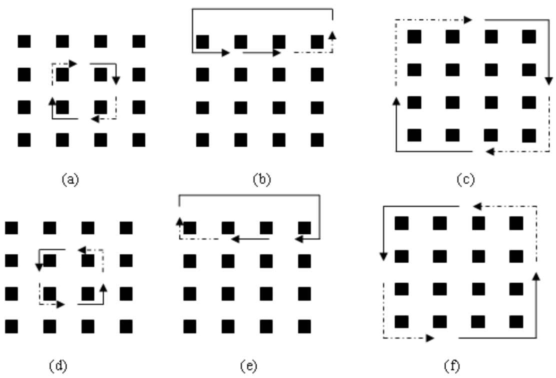

Figure 5. Six kinds of possible cycles in Tmesh

In Tmesh, there are six kinds of possible cycles (see Fig. 5). We give a detail analysis as follows: Fig. 5(a) does not appear, because the virtual turn is illegal according to the TXY algorithm. Fig. 5(b) does not appear, when the packet passes the *vertex*, it has been arrived at destination, so the virtual turn is illegal. Fig. 5(c) does not appear too. The virtual turns are illegal according to the selection of the *long link*. The analysis of Fig. 5(d), 5(e) and 5(f) is the same. So, we can conclude that TXY algorithm is deadlock-free.

IV. SIMULATION AND PERFORMANCE ANALYSIS

A. Simulation environment

We implement the Tmesh topology and TXY algorithm by modifying source codes of the gpNoCsim simulator which is developed by the Bangladesh University. GpNoCsim is a simulator for NoC which can simulate mesh, torus, and buffer fly fat tree and so on.

We simulate mesh and Tmesh topology with different nodes in the network, focusing on comparing the average delay and average hops. The parameters of simulation are shown in Table 2.

TABLE II. THE CONFIGURATION OF THE SIMULATOR

Parameters	Configuration
Virtual channel number	4
Input buffer size	4flits
Output buffer size	4flits
Packet size	8flits
Node number	16、36、64、100
Simulate Cycle number	100000
Warm Cycle	5000

B. Performance analysis

Fig. 6 and Fig. 7 shows the average delay and routing hops contrast between mesh and Tmesh. The number of the nodes in the network is 16, 36, 64 and 100. The results demonstrate a certain reduction in the average packet delay and routing hops. When the network has 64 nodes, the average delay and routing hops of Tmesh are 2.92% and 3.53% lower than those of mesh respectively.

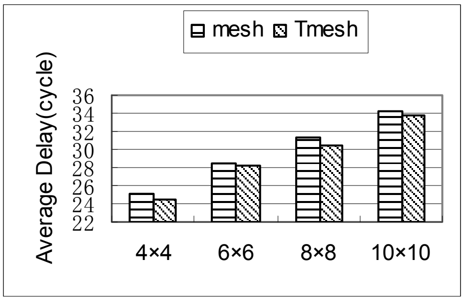

Figure 6. The average delay contrast

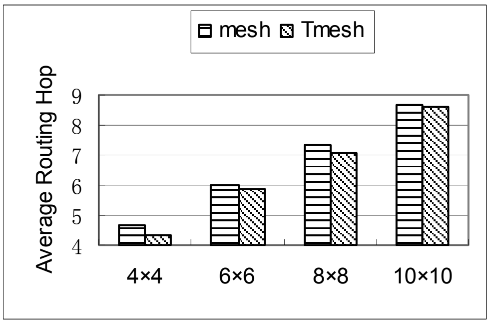

Figure 7. The average routing hops contrast

For different network scale, Tmesh's performance advantage is not so obvious. Because the source and destination are generated randomly, most of the routing path is determined by XY algorithm and the effect of the *long links* does not play out. In that case, the average delay and routing hops are similar between mesh and Tmesh. When the packet arrives at the *vertex* and the distance can be reduced by *long link*, Tmesh performs well.

V. CONCLUSION

In this paper we propose an improved mesh topology called Tmesh. We also present a routing algorithm suitable for the Tmesh. We compare Tmesh with mesh from network diameter, the ideal average delay and the ideal throughput. Further more, we simulate the Tmesh and TXY in gpNoCsim. The experimental results demonstrate a certain reduction in the average packet delay and routing hops.

TXY algorithm is not based on the shortest path, so there are differences between theoretical value and experimental results. It is an important work that improving the routing algorithm to reduce the average delay and routing hops in the future.

REFERENCES

[1] Kunle Olukotun, Lance Hammond, James Laudon. Chip Multiprocessor Architecture Techniques to Improve Throughput and Latency, 2009

[2] J.A.Kahle, M.N. Day, C.R.Johns, et al. Introduction to the Cell multiprocessor. IEEE, 2005: pp.589-604

[3] Lance Hammond, Benedict A.Hubbert, Michael Siu et al.. The Stanford Hydra CMP. IEEE, 2000: pp.71-84

[4] Wang Wei, Tang Zhizhong, Qiao Lin. Summary of the Inter-connection in Chip Multiprocessor. Computer Science. 2008, 35(9): pp.7-9

[5] Wang Huandong, Gao Xiang, Chen Yunji, Hu Weiwu. Interconnection of Godsom-3 Multi-Core Processor. 2008, 45(12): pp.2001-2010

[6] Tilea Company. TILE64 Multicore PCIe card connectivity solutions. The world of electronic components. 2008: pp.31-32

[7] Jesus Camacho Villanueva, Jose Flich, Jose Duato,etc. A Performance Evaluation of 2D-Mesh,Ring, and Crossbar Interconnects for Chip Multi-Processors. ACM, 2009

[8] Umit Y.Ogras, Radu Marculescu. "It's a Small World After All": NoC Performance Optimization Via Long-Range Link Insertion. IEEE, 2006, 14(7): pp. 693-706

[9] M.Daneshtalab, M.Ebrahimi, S.Mahammadi, A.Afzali-Kusha. Low-distance path-based multicase routing algorithm for network-on-chips. The institutino of Engineering and Technology. 2009, 3(5): pp. 432-442

[10] Shen Z. Average diameter of network structures and its estimation. In: Pro. Of the 1998 ACM Symp. On Applied Computin.1998: pp.593-597

[11] Zhu Xiaojing, Hu Weiwu, Ma ke, Zhang Longbing. Xmesh: A Mesh-Like Topology for Network on Chip. Journal of Software. 2007, 18(9): pp.2194-2204

[12] Chen Guoliang, Wu Junmin, Zhang feng et al.. Parallel Computer Architecture. 2002: pp.158-159

[13] Dally J, Towles B. Principles and Practices of Interconnection Network. Morgan Kaufmann Publishers.2003

[14] Dong Yingfei, Wang Dingxing, Zheng Weimin. Exact Computation of the Mean Minimal Path Length of N-Mesh and N-Torus. J.Chinese Computers. 1997, 20(4): pp. 376-380

[15] Chrisopher J.Glass, Lionel M Ni. The Turn Model For Adaptive Routing. ACM.1992: pp. 278-287