Physical Planning Of On-Chip Interconnect Architectures

Hongyu Chen, Bo Yao, Feng Zhou, and Chung-Kuan Cheng
Department of Computer Science and Engineering
University of California, La Jolla, CA92093-0114
{hchen, byao, zfeng, kuan}@cs.ucsd.edu

Abstract

Interconnect architecture plays an important role in determining the throughput of meshed communication structures. We assume a mesh structure with uniform communication demand for communication. A multi-commodity flow (MCF) model is proposed to find the throughput for several different routing architectures. The experimental results reveal several trends: 1. The throughput is limited by the capacity of the middle row and column in the mesh, simply enlarging the congested channel cannot produce better throughput. A flexible chip shape provides around 30% throughput improvement over a square chip of equal area. 2. A 45-degree mesh allows 17% throughput improvement over 90-degree mesh and a 90-degree and 45-degree mixed mesh provides 30% throughput improvement. 3. To achieve maximum throughput on a mixed Manhattan and diagonal interconnect architecture, the best ratio of the capacity for diagonal routing layers and the capacity for Manhattan routing layers is 5.6. 4.Incorporating a simplified via model, interleaving diagonal routing layers and Manhattan routing layer is the best way to organize the wiring directions on different layers.

1. INTRODUCTION

Mesh is a common routing architecture for many reconfigurable computing systems. Both conventional FPGAs[12] and recently proposed on-chip multi-processors systems[7] [6] use mesh networks as communication backbones. With rapid technology scaling, wires become one most precious resource on a chip. Unreasonable distribution of wire resources will result in bottlenecks that stall the data flows, meanwhile leave other routing resources wasted. Simply enlarging the channel capacity of the whole array is by no means an effective solution.

Our goal is to allocate channel capacities in the mesh routing architecture to maximize its communication capability. The communication capability is measured by the throughput, the amount of information that every pair of nodes can exchange simultaneously. Throughput is a function of channel capacity and the dimension of the processor array.

Khalid and Betz investigated the channel allocation problems for FPGAs in [5] [2]. They applied placement and routing to benchmark circuits on FPGAs with different routing track distributions. They conclude that uneven track distribution do not improve the routability of FPGA interconnects.

Multi-commodity flow (MCF) is a natural way to model the communication network traffic. Many previous works used MCF in studying the wide area communication network traffic [10]. Due to the high computing complexity of MCF, most of these works [10] adopted heuristic methods to approximate the MCF solution.

Recent advance in MCF algorithm [4] allows us to compute MCF more efficiently. In this paper, we choose MCF to model communication traffic. We extend the MCF algorithm in [4] to solve various MCF problems. Solution of MCF finds the optimal throughput for a given routing architecture. In our MCF model, the routing demand is equal for every pair of nodes. Thus, the result is independent of test cases. Moreover, the result of MCF is independent of placement and routing.

In [9], Mutsunori et al. demonstrated that on-chip diagonal routing is feasible based on state-of-the-art technologies. The progress of diagonal routing technology provides another opportunity to explore different arrangements of interconnect structure. We compare the throughput of three different mesh structures, the 90-degree mesh, the 45-degree mesh, and the 90-degree and 45-degree mixed mesh. Experimental results show that a 45-degree mesh can achieve better throughput than a 90-degree mesh. Moreover, 90-degree and 45-degree mixed mesh can further improve the throughput.

Mixed 90-degree and 45-degree mesh allows more freedom on routing directions. We explore the allocation of routing resources and the arrangement of routing resources of routing directions between 90-degree and 45-degree wires. We propose a simplified via model to derive the optimal solution.

Our contributions in this paper include:

(1). We use MCF model to analyze the detailed communication traffic on mesh interconnect networks. We find the exact traffic bottlenecks of the network and the throughput of communications in the mesh structure. This provides a feasible upper bound of communication.

(2). We extend the flow approach [4] to compute the optimal routing resource allocation for mesh interconnect structures with reasonable sizes. The results reveal some basic trends of throughput related to the scale and structure of the communication mesh:

a. For uniform capacity mesh, the congested edges lie in the center rows and columns. The total throughput of each node is inversely proportional to the dimension of the mesh
b. The re-arrangement of capacities between different columns or rows will not improve the throughput if we keep the total capacity of the columns or rows a constant.
c. A flexible chip shape provides a throughput improvement of around 30% over a square chip of equal area.
d. A 45-degree mesh structure produces a 17% more throughput than a 90-degree mesh for a processor array of 144 nodes.

e. A mixture of 90-degree and 45-degree mesh structure can achieve even a 30% more throughput. To achieve maximum throughput, the ratio of resources allocated to the 45-degree routing layers verses those to the 90-degree routing layers approaches 5.6 as number of nodes increases.
f. In the 90-degree and 45-degree mixed routing, interleaving the diagonal routing layer and Manhattan routing layers can reduce the number of vias and hence increase the communication throughput.

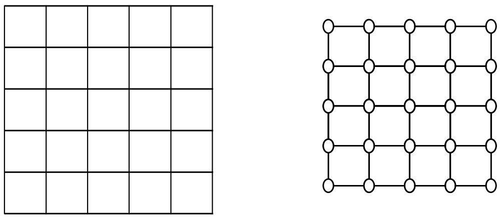
The rest of this paper is organized as follows: Section 2 presents the problem formulation in MCF model. Section 3 introduces six different interconnect structures we consider. Section 4 gives the experimental results and our observations. We draw conclusion in section 5.

2. PROBLEM FORMULATION

We decompose the communication resources into an array of n×n slots. Each slot contains a communication terminal, say, a processor. The slots are aligned in rows and columns. The slot array forms a 90-degree mesh structure. Figure 1(a) illustrates an example of a 90-degree mesh structure with 25 slots. Each square tile represents a slot. The mesh structure can be mapped to a graph G={V, E} according the following rules:

(1) Each slot corresponds to a node in the graph.
(2) The adjacency between two slots is represented by an edge connecting two corresponding nodes.
(3) The edge capacity is proportional to the length of the line segment separating the adjacent slots, and the number of routing layers.

Figure 1(b) describes the graph corresponding to the mesh in Fig.1(a).



(a) (b)

Fig. 1 A 5 by 5 communication mesh and its graph representation

We assume a uniform communication requirement i.e. every pair of nodes communicate with an equal demand. All communications happen at the same time. Note that the model can be extended to various communication demands, e.g., Poisson distribution, Rents rule, etc., depending on specific applications. In this paper, the uniform pairwise communication model is adopted because of its simplicity and genericalness. Moreover, the communication demand represents an unbiased symmetry, which makes the solution independent of the test cases, placement, and routing.

We define the throughput, z, to be the maximum amount of communication flow between every pair of nodes. We try to find the throughput using a multi-commodity flow model. The flow that starts from node i is deemed as commodity i. Commodity i starts from node i with the amount of $z\cdot(N\text{-}1)$, where $N = n^2$ is the number of nodes in the graph, to each of the rest nodes with the amount of z. We solve the multi-commodity flow problem to find the maximum value of z.

We can use following linear program to express the above MCF problem:

Maximize: z

S. t.: For each commodity v , on each node i,

$$\sum_{j \in neighbor\ of\ i} (f_{ji}^{v} - f_{ij}^{v}) = \begin{cases} -z\cdot(n^2-1) & if\ i = v \\ z & otherwise \end{cases} \quad (1)$$

and for each edge (i,j) in the graph,

$$\sum_{v=1}^{n^2} (f_{ij}^{v} + f_{ji}^{v}) \le c_{ij} \quad (2)$$

In this linear program, flow variable f_{ij}^{v} represents the flow amount of commodity v on edge *(i,j)*. The edge capacity c_{ij} represents the flow capacity of edge *(i,j)*. We set that the flow injecting to a node is positive and the flow ejecting from a node is negative.

The linear program includes two sets of constraints. Constraint (1) describes the flow conservation of each commodity v at each node i. Constraint (2) denotes that the total amount of flow on each edge is no more than the capacity of that edge.

In subsections 3.3, 3.4 and 3.5, we allow edge capacity to be changed. Thus, the edge capacities become variables in the linear constraints. Thus, allows us to optimize the capacities under the area constraints.

In [4], a fast combinatorial (1+ε)-approximation algorithm was introduced to solve the MCF problem. We extend the approach to incorporate edge capacities as variables.

According to the algorithm in [4], we adopt the primal-dual structure of the linear program. The algorithm assigns a nonnegative shadow cost [12] to each edge according to the congestion level on that edge. Initially, all the shadow costs are set to be equal. Then, the algorithm proceeds in iterations. In each iteration, we reroute a fixed amount of flow along the shortest path for every commodity. At the end of each iteration, we adjust the capacity of every edge and its shadow cost according to the dual linear program.

In our model, we all fractional flows. Note that the throughput, $\hat{z}$, of the fractional flow model, is an upper bound of the throughput, $\tilde{z}$, of the integer flow model [3][1]. In [8], Motwani and Raghavan showed that by randomized rounding, with the probability of 1- ε, we can find $\tilde{z}$ approaches $\hat{z}$ with inequality $\tilde{z} \ge \hat{z}/(1+\Delta^{+}(1/\hat{z},\varepsilon/2N))$, where N is the number of nodes in the mesh, ε is any real number between 0 and 1, and $\Delta^{+}(1/\hat{z},\varepsilon/2N)$ is the value of δ such that $\left[e^{\delta}/(1+\delta)^{(1+\delta)}\right]^{1/\hat{z}} = \varepsilon/2N$.

3. ROUTING ARCHITECTURES

[1]For packet switching network in RAW and Smart Memories, we do not require the flow to be integer. For wire switching network in FPGAs, the flow amounts can be interpreted as the number of wires, which needs to be integers.

We construct six routing architectures with different capacities and routing orientations. The first three structures are 90-degree meshes with different edge capacities. In the first architecture, every edge has a unit capacity. In the second architecture, edges on the same row or column have equal capacity. In the third architecture, edge capacities are flexible but the sum of the capacities of all the edges is fixed. The fourth architecture is a 45-degree mesh where interconnection goes in 45 degree. The fifth is a mixture of 90-degree and 45-degree mesh. And the last one is the mixed 90-degree and 45-degree mesh with different routing direction assignments.

3.1 Uniform edge capacity

For the model of uniform edge capacity, All the edge capacity is set to a unit, i.e. $c_{i,j} = 1$ for all edges *(i,j)* in the graph. This case is the basis of our experiments. We assume that the n×n array of slots is evenly distributed in a square area.

3.2 Uniform row and column capacity

In the second structure for interconnection, edge capacities c_e are set as variables. However, the capacities of edges in the same row are set to be equal. Likewise, the vertical capacities of edges in the same column are set to be equal. The sum of the vertical edge capacities in a row is set to be n, and the sum of the horizontal edge capacities in a column is also set to be n. In other words, we assume that the height and the width of the array remain to be *n*.

Let c_{Hi} be the capacity of horizontal edges in the *i*-th row, and c_{Vi} be the capacity of vertical edges in the *i*-th column. We add the *2n* variables, $c_{H1}, c_{H2}, \ldots, c_{Hn}, c_{V1}, c_{V2}, c_{Vn}$, to the linear program. The height and width constraints of the array can be expressed as:

$$\sum_{i=1}^{n} c_{Hi} = n \text{ and } \sum_{k=1}^{n} c_{Vi} = n \cdot$$

For this structure, we assume that we can adjust the row height and the column width of the array of processors.

3.3 Fixed total edge capacity

For the third structure we give the program more freedom to choose the best edge capacities. We require only that the total capacity of all edges to be a constant. This structure represents the best edge capacity we can allocate for a 90-degree mesh. The resultant throughput is an upper bound of a 90-degree mesh architecture.

We set the edge capacities, c_{ij}, as variables. The total capacity constraint is expressed as:

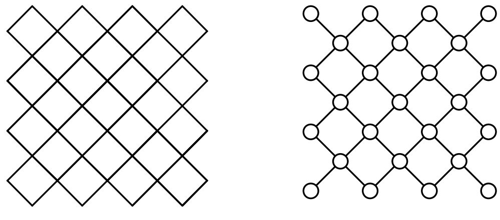
$$\sum_{\textit{for} \text{ all edges ij}} c_{ij} = 2 \cdot (n^2 - n) \cdot$$

Note that $2 \cdot (n^2 - n)$ is the number of edges in an n×n mesh.

For this structure, we assume that the area of each slot is flexible. We adjust the height and width of each individual slot so that the total area remains the same.

3.4 45-degree mesh

The fourth structure adopts a 45-degree mesh. All wires are oriented in 45 degree or 135 degree. The size of the mesh increases with *n*. For a 45-degree mesh of *n*, the number of nodes is $n^2+(n-1)^2$ and the number of edges is $4(n-1)^2$.



(a) (b)

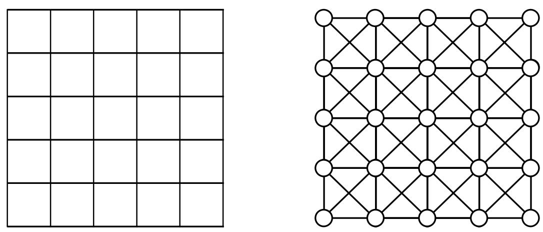
Fig. 2 Example of 45 degree mesh of n = 4

Figure 2(a) shows an example of 45-degree mesh of n = 5. Figure 2(b) illustrates the corresponding graph to the mesh. In this structure, we assume that the slots are shaped in diamonds (a square rotated by 45°) and are aligned in 45° and 135° directions. Thus, the edge capacity remains to be a unit, i.e. $c_{ij}=1$.

3.5 90-degree and 45-degree mixed mesh

In the fifth structure, we add 45-degree channels to the 90-degree mesh. Figure 3 illustrates an example of the mixed mesh for *n* = 5. Figure 3(a) shows the slots arrangement. For an *n* by *n* mixed mesh, the number of nodes is n^2 and the number of edges is $2(n-1)^2 + 2(n^2-n)$.

In Figure 3(b), the edges are oriented in 0°, 90°, 45° or 135° angle. All nodes are aligned in rows and columns. Thus the channels for 45° and 135° wires are scaled by $1/\sqrt{2}$. In other words, for a pair of routing layers, if we can allocate a capacity of *x* to 0° and 90° edges, we can only allocate a capacity of $x/\sqrt{2}$ to 45° and 135° edges. Let c_1 be the capacity of horizontal and vertical edges, c_2 be the capacity of 45° and 135° edges. The area constraints can be expressed as $c_1 + \sqrt{2} \cdot c_2 = 1$. Thus, the total area is equal to the area of uniform structure.



(a) (b)

Fig. 3 A 90-degree and 45-degree mixed mesh for n equals 5

3.6 Routing Direction Assignment

Vias become an important concern when number of routing layers increases. In [1] a global routing graph with via edges is used to model the multiplayer routing. In that model vias are not considered as routing blockages.

We propose a network flow model shown in Fig. 4 to take vias into. Our basic assumption is that each via will block one routing track. For each slot, we set an upper bound on the total number of vias and wires across the node.

Suppose there are k routing layers. Each slot is now represented by k routing cells (Fig. 4(a)). Each routing cell consists of two nodes (Fig. 4(b)): n_a and n_b. Node n_a takes all the incoming edges from the neighboring routing cells, and node n_b ejects edges to neighboring routing cells. An edge with capacity c direct from node n_a to node n_b. This edge is used to restrict the total number of vias and wires crossing the routing cell. Using this flow model, we compare the communication throughputs with different routing layer assignments.

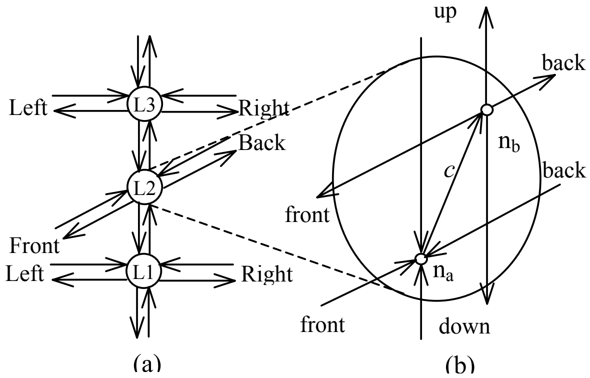


Fig. 4 A network flow model for multiplayer routing

4. EXPERIMENTAL RESULTS

We first use Matlab's linear program package on a Sun Ultra10 workstation to compute MCF solutions. For the case with 100 nodes, the run time exceeds 24 hours. We then implement the MCF algorithm [4] using C programming language. The MCF algorithm derives the MCF solutions for cases with up to 319 nodes within 12 hours.

4.1 Results for uniform edge capacity mesh

Table 1 describes the results of uniform edge capacity meshes with n = 2 to 10. We list the number of nodes and the throughput z.

Table 1. Results of uniform edge capacity mesh

n	Number of nodes	*z*
2	4	0.3750
3	9	0.3333
4	16	0.2343
5	25	0.2000
6	36	0.1620
7	49	0.1429
8	64	0.1229
9	81	0.1111
10	100	0.0990

From the experimental result, we have the following observations:

1. The throughput is $1/n$ when n is odd and $(n^2-1)/n^3$ when n is even.
2. The throughput is limited by edges on the middle column and row. When n is an even number, edges in the central row and column form the bottleneck of the flow. When n is an odd number, the two columns and two rows form the bottleneck.

Figure 5 shows the bottleneck of communication flow for n = 4 and 5. The congested edges are marked with bold lines. Note that the bottleneck form the horizontal and vertical cut sets. The cut lines are shown with dashed lines.

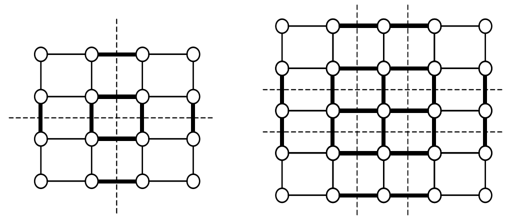


Fig. 5 Flow congestion for uniform edge capacities

4.2 Results of uniform row and column capacity mesh

For equal n, the throughput of a 90-degree mesh with uniform row and column capacities is exactly the same as that of the 90-degree mesh with fixed edge capacities. We obtain no throughput improvement because the total capacity of the edges in each column and row is fixed.

4.3 Results for fixed total edge capacities mesh

For n = 2 to 10, Table 2 shows the results of 90-degree mesh with fixed total edge capacities. The fourth column provides the throughput improvement compared to that of 90-degree mesh with uniform edge capacity. As we no longer limit the total capacity of each row or column, the average throughput improves 29.7% for n = 4 to 10.

Table 2. Results of fixed total edge capacities

N	Number of nodes	*z*	Improvement on *z* (%)
2	4	0.375	0.00
3	9	0.333	0.00
4	16	0.281	20.01
5	25	0.240	20.00
6	36	0.208	28.57
7	49	0.185	28.56
8	64	0.169	33.32
9	81	0.148	33.35
10	100	0.134	36.36

The results also show that all edges are congested. The optimal edge capacity is no longer uniform. The capacity is larger for the edges in the middle row and column. Table 3 shows the optimal edge capacities for all the vertical edges in a 6 by 6 mesh. We list the sum of all the capacities of each row. Figure 6 illustrates the optimal sums of the rows in a 9×9 mesh. Note that there are eight rows of vertical edges in a 9×9 mesh. The chip area is no longer a square, but a convex area as is shown in Figure 6.

Table 3. Optimal capacities for vertical edges in 6 by 6 mesh

Row \ Col	1	2	3	4	5	6	Sum
1	0.60	0.74	0.79	0.79	0.74	0.61	4.28
2	0.95	1.19	1.27	1.28	1.19	0.96	6.85
3	1.07	1.34	1.44	1.44	1.34	1.07	7.71
4	0.95	1.19	1.27	1.27	1.19	0.96	6.85
5	0.60	0.74	0.79	0.79	0.74	0.60	4.28

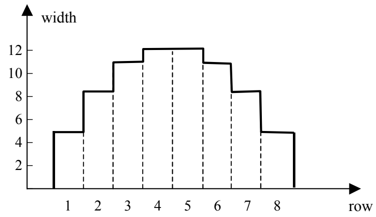


Fig. 6 Optimum row width of 9 by 9 mesh to maximize total throughput

4.4 Results for 45-degree mesh

Table 4 shows the results of 45-degree mesh for n = 2 to 12. To compare the results in table 4 and table 1, we use the cases with almost the same number of nodes. For instance, both the case of n = 4 in table 4 and the case of n = 5 in table 1 contain 25 nodes. The case with 45-degree mesh achieves the throughput of 0.209, which gains a 4.18 percent improvement. Also we compare the case of n = 7 in table 5 with the case of n = 9 in table 1. The case in table 5 contains 85 nodes, which has 4 more nodes than the case in table 1. The throughput of the 45-degree mesh case is 0.1260, which is 13.16% more than that of the 90-degree mesh case.

Table 4. Results of 45-degree mesh

N	Number of nodes	*z*
2	5	0.250
3	13	0.250
4	25	0.209
5	41	0.174
6	61	0.147
7	85	0.126
8	113	0.106
9	145	0.101
10	181	0.0828
11	221	0.0759
12	265	0.0673

The congested edges also present a different pattern: they form 4 cut sets at four corners. Figure 7 shows the flow congestion in 45-degree mesh for n = 5 and n = 6. The congested edges are in bold lines and the cut lines are in dashed lines.

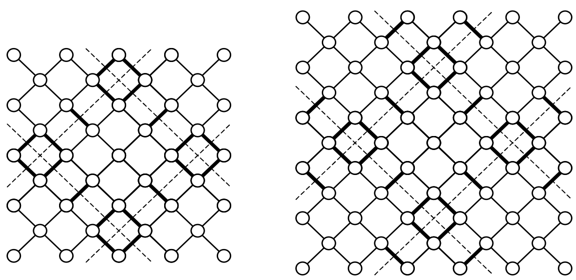


Fig. 7 Flow congestion in 45-degree mesh

Fig.8 explains why 45-degree routing is better than 90-degree routing. Assume that we have a square shaped chip with two routing layers. Figure 8(a) illustrates the case of 90-degree routing and Fig. 8(b) depicts the case of 45-degree routing. We draw a cut line for the horizontal congested edges in dashed lines in Figure 8(a). Only the wires on the horizontal routing layer could cross the cut line and the number of wires across the cut line is d/D, where d is the wire pitch and D is the dimension of the chip. We then draw a similar cut line on Fig.8 (b). The number of edges across the cut line in each layer is $d/\sqrt{2}D$. The total number of wires crossing the cut line for the two layers in Fig. 8(b) is $\sqrt{2}d/D$. Thus the upper bound of throughput increase to $\sqrt{2}$ = 1.414. However, the throughput is now limited by the cut edges at four corners.

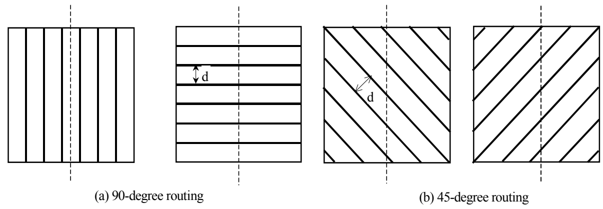


Fig 8. Explanations for throughput increase for a 45-degree mesh

4.5 Results for 90-degree and 45-degree mixed mesh

Table 5 depicts the results for the 90-degree and 45-degree mixed mesh structure. Column 2 lists the throughput z. Column 3 lists the throughput improvements over the 90-degree meshes with uniform edge capacity. Columns 5 and 6 list the best capacity for horizontal and vertical edges, c_1, and the best capacity for 45-degree edges, c_2, respectively. Column 7 lists the normalized capacity ratio of the diagonal edges to the Mahattan edges.

Table 5. Results of 90-degree and 45-degree mixed mesh

n	z	Improvement on *z* (%)	c_1	c_2	$\sqrt{2}\cdot c_2/c_1$
2	0.375	0.00	1.0000	0.0000	0.00
3	0.333	0.00	1.0000	0.0000	0.00
4	0.245	4.85	0.2290	0.5452	3.36
5	0.219	9.53	0.2577	0.5249	2.88
6	0.185	14.04	0.1853	0.5761	4.39
7	0.166	16.01	0.2022	0.5641	3.94
8	0.148	20.11	0.1614	0.5930	5.19
9	0.134	20.40	0.1696	0.5872	4.89
10	0.120	21.31	0.1553	0.5988	5.44
11	0.110	21.48	0.1608	0.5935	5.22
12	0.101	22.05	0.1527	0.5992	5.55
13	0.094	22.14	0.1562	0.5967	5.40
14	0.087	22.68	0.1510	0.6004	5.62
15	0.082	22.71	0.1536	0.5986	5.51
16	0.076	22.95	0.1504	0.6008	5.65
17	0.0723	23.02	0.1524	0.5994	5.56

We have the following observations:

1. The throughput of the mixed mesh is better than the 90-degree mesh, given the equal communication resource. The improvement in the throughput is up to 20.04% for large number of nodes. It is also better than 45-degree mesh in terms of throughput.
2. With n increasing, the optimal ratio for the capacity of the 45-degree edge to the 90-degree edge approaches 5.6.

4.6 Results for routing layer assignment for 45-degree and 90-degree mixed mesh

We use our MCF model introduced in section 3.6 to compute the optimal routing direction assignment for mixed 45-degree and 90-degree routing. Assume that there are four routing layers and each of them is assigned to a different routing direction. Fig. 9 shows four different routing layer assignments. The throughputs under four different assignments are listed in Table 6. The throughputs with assignments IV and I are about 16 percent larger than the throughputs with assignments II and III.

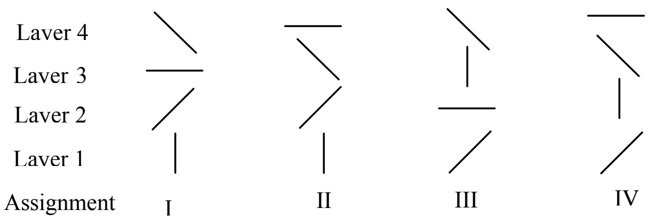


Fig. 9 Different routing directions assignments

Table 6. Throughput with different routing layer assignment

N	*z(I)*	*z*(II)	*z(III)*	*z(IV)*
5	0.0173	0.0147	0.0147	0.0171
6	0.0102	0.0083	0.0.0083	0.0101
7	0.0065	0.0053	0.0051	0.0064
8	0.0041	0.0034	0.0034	0.0041

Fig. 10 explains why interleaving the Manhattan routing layers and diagonal routing layers can produce better throughput. In Fig. 10, we can see, given two points on the plane the shortest way to connect them is always a Manhattan line plus a diagonal line. Thus if we interleave the Manhattan routing layer and diagonal routing layer, the wires can go along shortest paths without paying more vias. This will produce better throughput.

5. CONCLUSIONS

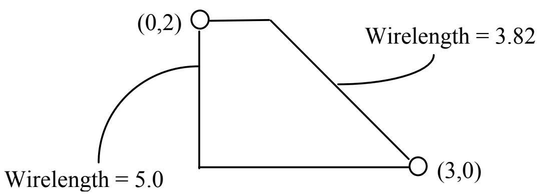


Fig. 10 Shortest path between two points on the plane

Interconnect sets important limitations for throughput of communications. For meshed interconnect structures, we propose a multi-commodity flow model to find the throughput and the best communication flow pattern. This provides a feasible upper bound of communication throughput.

We study several different interconnect structures, including 90-degree meshes with uniform and non-uniform edge capacities, the 45-degree mesh, and the 90-degree and 45-degree mixed mesh. The results reveal the following basic trends:

1. For the 90-degree mesh with uniform edge capacity, the congested edges lie in the center rows and columns. Moreover, the throughput can be improved by more than 30 percent if we choose optimal edge capacities instead of uniform edge capacities.
2. For a 45-degree mesh structure, the throughput is better than that of 90-degree mesh with the same number of nodes. The bottleneck edges lie off the diagonal lines and form 4 cut sets.
3. For a 90-degree and 45-degree mixed mesh, the throughput is better than either a 90-degree-only or a 45-degree-only mesh structure. Moreover, more resource should be allocated to 45-degree edges in a mixed structure as the number of nodes increases. With the consideration of vias in mind, we should interleave the Manhattan routing layers and diagonal routing layers.

6. ACKNOWLEDGEMENT

This work was supported in part under grants from NSF project number MIP-9987678, the California MICRO program, SRC support, and Cal-(IT)^2 graduate fellowship. The authors would like to thank all the reviewers for their comments.

7. REFERENCES

[1] C. Albrecht “Provably good global routing by a new approximation algorithm for multicommodity flow,” In Proc. Of ISPD, April 2000, pp. 19-25

[2] V. Betz and J. Rose, "Effect of the Prefabricated Routing Track Distribution on FPGA Area-Efficiency," in IEEE Trans. on VLSI, Vol. 6, No. 3, Sept. 1998, pp. 445-456

[3] R.C. Carden, IV and C. K. Cheng “Early feasibility and cost assessment for multichip module technologies.” Int’l. J. of High Speed Electronics and Systems, vol.6, (no.3), World Scientific, Sept. 1995. pp.441-58.

[4] N. Garg, and J. Konemann, “Faster and Simpler Algorithms for Multicommodity Flow and other Fractional Packing Problems,” In Proc. Of the 39th Annual Symposium on Foundations of Computer Science, pp.300-309, 1998

[5] M. Khalid and J. Rose, "Experimental Evaluation of Mesh and Partial Crossbar Routing Architectures for Multi-FPGA Systems," in IFIP IWLAS `97, Grenoble, France, December 1997, pp. 119-127.

[6] K. Mai, etal. “Smart Memories: a modular reconfigurable architecture”, in Proc. of 27th ISCA, Vancouver, BC, Canada, 2000. pp.161-71.

[7] C. A. Moritz, D. Yeung, and A. Agarwal, “Exploring optimal cost-performance designs for Raw microprocessors.” In Proc.of IEEE Symp. on FPGAs for Custom Computing Machines, Los Alamitos, CA, 1998. pp.12-27.

[8] R. Motwani and P. Raghavan, *Randomized Algorithms*, Cambridge university press, 1995, pp.79-83.

[9] I. Mutsunori, T. Mitsuhashi, A. Le, S. Kazi, Y. Lin, A. Fujimura, and S. Teig, “A Diagonal Interconnect Architecture and Its Application to RISC Core Design,” Proc. ISSCC, pp. 684- 689. San Jose, CA, Feb. 2002.

[10] I. Saniee, J. S. Sokol, “Virtual Path Design in Service-Specific ATM Networks,” in J. of Heuristics, 6: pp.-83, 2000

[11] F. Shahrokhi and D. Matula, “The maximum concurrent flow problem,” J. ACM, 37(2): pp.318-334, 1990

[12] The programmable Logic Data Book, Xilinx Inc., San Jose, 1994