A Performance Evaluation of Fully Adaptive Wormhole Routing including Selection Function Choice

Loren Schwiebert
Department of Electrical and Computer Engineering
Wayne State University
Detroit, MI 48202–3902
Email: loren@ece.eng.wayne.edu

Abstract

Many adaptive routing algorithms have been proposed for wormhole-routed interconnection networks. Comparatively little work, however, has been done on determining how the selection function affects the performance of an adaptive routing algorithm. In this paper, we present a detailed simulation study of various selection functions for four different fully adaptive wormhole routing algorithms for 2D meshes. The simulation results show that the choice of selection function has a significant effect on the average message latency and saturation behavior. In fact, it appears that changing the selection function can have a greater effect than changing the routing algorithm. In addition, the best selection function for one adaptive routing algorithm may not be the best choice for another adaptive routing algorithm. An explanation and interpretation of the results is provided.

1 Introduction

Wormhole routing [7] has become the switching technique of choice in most modern distributed-memory multiprocessors, including switches for networks of workstations. Wormhole routing propagates messages through the network by dividing each message into flits, where a flit is a small multiple of the physical channel width. The header flit of a message contains the routing information and the data flits of the message follow the header flit through the network. The major advantage of wormhole routing is that when the header arrives at an intermediate router, the router forwards the message header to a neighboring router as soon as an output channel the message can use is available.

Since the flits of a message are forwarded as soon as possible, the message latency is largely insensitive to the distance between the source and destination. In addition, wormhole routing requires only enough storage at each router to buffer a few flits, rather than the entire message. The low minimum latency and modest buffer requirements account for the popularity of wormhole routing in distributed-memory multiprocessors. See Ni and McKinley [14] for a detailed explanation of wormhole routing.

Wormhole routing is susceptible to contention even with moderate traffic, which results in higher message latency. Because message headers are forwarded immediately, a message can span many channels simultaneously. Furthermore, a message that is blocked remains in the network. Dally [5] proposes a cost-effective method of reducing contention by allowing multiple *virtual* channels to share the same physical channel. Contention can be further reduced by permitting a message to choose from among the multiple paths in the network.

Adaptive routing algorithms support multiple paths between a source and destination. Adaptive routing is supported on the nCUBE3 and Cray T3E. Adaptive routing algorithms can be differentiated by the number of shortest paths allowed. *Fully adaptive* routing algorithms allow all messages to use any shortest path. Mohapatra [13] provides a recent survey of adaptive routing algorithms.

When an adaptive routing algorithm is used, many messages can choose from multiple output channels. The output selection function chooses one of these output channels for the message.[1] When multiple output channels are available, one of these channels is chosen based on the selection function. Global knowledge of the network status is too costly to obtain, so the selection function makes a decision using local knowledge of the network traffic.

Previous research on routing algorithms for wormhole routing has produced many different adaptive routing algorithms. Little research has been done, however, on how different selection functions affect the performance of an adaptive routing algorithm. In this paper, we present a simulation study of different selection functions with four different fully adaptive routing algorithms. The results clearly show that the choice of selection function has a significant impact on not only the average message latency of the routing algorithms but also the saturation behavior. Furthermore, the best selection function depends on the routing algorithm.

[1] Because of the focus of this paper, references to selection functions should be understood to mean output selection functions.

2 Previous Work

Many adaptive routing algorithms have been proposed, including [2, 4, 8, 11, 12, 18, 19]. Comparatively little work, however, has been done on selection function design and performance analysis.

Badr and Podar [1] proved that the zigzag selection function is optimal for meshes, in the sense that it maximizes the probability of a message reaching the destination without delay. Glass and Ni [12], and Dally and Aoki [6] have both made limited studies of how different selection functions affect the routing algorithm performance. Feng and Shin [10] have also studied the impact of three selection functions on the routing algorithm performance. Specific routing algorithms are not evaluated, but both oblivious and fully adaptive routing is considered.

Duato [9] presents simulation results for a time-dependent selection function, which prevents a message from using certain virtual channels until the time a message has been waiting exceeds some threshold value. Rao [15] proposes loading the virtual channels differently based on their relative expected utilizations. Although the techniques proposed by Duato and by Rao improve performance, both add complexity to the router, which could increase the network cycle time [3].

Restrictions imposed by the routing algorithm to achieve deadlock freedom cause an unbalanced load on the channels. Upadhyay, *et al.* [19] have argued that their routing algorithm, although less adaptive, obtains better performance because their routing algorithm makes more balanced use of the virtual channels. Similarly, we show that the selection function also affects the balance of traffic in the network and hence the performance.

In this paper, we extend earlier work [17] by conducting detailed simulations of how the selection function influences the performance of four fully adaptive routing algorithms. The first, *opt-y*, was proposed by Schwiebert and Jayasimha [18]. Since opt-y has the fewest restrictions of any deadlock-free fully adaptive routing algorithm for meshes, it allows more latitude in the selection function. The second, *balanced*, is claimed by Upadhyay, *et al.* [19] to obtain a better balance for the network traffic. We study if the selection function choice can also affect the balance. The third, *mad-y*, is the algorithm proposed by Glass and Ni [11] and is the most adaptive 2D mesh routing algorithm without cyclic dependencies. Finally, the fourth routing algorithm, *planar*, is the planar-adaptive routing algorithm proposed by Chien and Kim [4]. This algorithm divides the mesh into two separate sub-networks (East and West) and prevents traffic from switching between sub-networks. Although other interesting mesh routing algorithms have been proposed, we believe these four routing algorithms have a sufficiently diverse set of characteristics to allow meaningful conclusions to be drawn about the general effect of selection function choice.

3 Simulation Methodology

In order to model the interconnection network, a modular simulator was written in C++ using CSIM18 [16]. The simulator is structured so that modules, such as the routing algorithm or message traffic, can be changed without any modifications to the remaining components.

For accuracy, a flit-level simulation using demand multiplexing is performed. To prevent starvation, the input selection function processes requests in the order of arrival. We record both the average message latency (measured in network clock cycles) and the channel utilization (averaged over all the network channels). A 16×16 mesh is used with message sizes of 16 flits or 128 flits. Five traffic patterns, which were also used by Glass and Ni [12], are simulated:

- **Uniform** – The source sends to any other node with equal probability.
- **Matrix Transpose** – A source at (x, y) sends to a destination at $(k - y - 1, k - x - 1)$, where k is the arity of the mesh. ($k = 16$ in these simulations.)
- **Center Reflection** – A source at (x, y) sends to a destination at $(k - x - 1, k - y - 1)$.
- **Matrix Rotate** – A source at (x, y) sends to a destination at $(k - y - 1, x)$.
- **Tree Reduction** – A source at (x, y) sends to a destination at $((x + k)/2, y/2 + ((x + 1) \bmod 2) * k/2)$.

The simulator has three phases: start-up, steady-state, and termination. The start-up phase is used to ensure the network is in steady-state before measuring message latency. When the number of active messages is nearly constant over a sufficiently long period of time, the network is judged to be in steady-state. On the other hand, if the number of active messages consistently increases, the network is saturated and the simulation is halted. The average message latency and channel utilization are recorded during steady-state and enough messages are generated to achieve a 95% confidence interval of well under $\pm 1\%$, except in cases where the network is extremely close to saturation, when the 95% confidence interval occasionally exceeds 5%. After generating this many messages, the termination phase begins. During the termination phase, messages continue to be generated at the same rate, so that the network traffic remains constant. Messages generated during the termination phase are *not* included in the results. The termination phase continues until *all* the messages generated during the steady-state phase have been delivered.

To provide as fair a comparison as possible, opt-y, mad-y, and planar are extended to use two virtual lanes in the x dimension. A uniform network cycle time is assumed, because we do not want to bias the results by our differential success in optimizing the various routing algorithms.

In table 1, the routing restrictions of each routing algorithm, depending on the location of the destination relative to the source, are presented. X1 is used to represent the first virtual channel in the x dimension and similarly for the other channels. A comma separates channels in a set from which the selection function can select a channel. A slash divides two sets when a channel from the first set cannot be used after a channel from the second set.

4 Simulation Results

The selection functions only affect the choice of output channel when *multiple* output channels are free. When only one output channel is free, that channel is selected. When no output channel is free, the simulator tries repeatedly to route the message on any of the permitted channels until a channel becomes free. Since the selection function controls the routing only when multiple channels are free, the choice of selection function may seem unimportant. As the simulation results show, however, the choice has a dramatic effect on both message latency and saturation behavior.

For opt-y and mad-y, eight selection functions were simulated. The difference among the first six selection functions is the priority placed on the virtual channels. For example, when multiple output channels are available for a message, the x,y1,y2 selection function uses one of the X channels if needed and available, followed by Y1, and uses Y2 only when there is no other option. The other two selection functions tested are no turn and zigzag. No turn attempts to continue routing in the current direction, while zigzag tries to use the direction with the furthest remaining distance. Planar never allows a message to choose between Y1 and Y2, so some selection functions are redundant.

For balanced, X1 and X2 have different restrictions, so there are 24 different combinations. Instead of testing all 24, we chose a set of selection functions to test a variety of combinations. Selection functions giving priority to the x dimension or the y dimension, those giving priority to channels the message (based on direction) is always permitted to use (preferred) or the channels a message is sometimes prohibited from using (unpreferred). The no turn and zigzag selection functions are also tested.

Simulations were run for both 16-flit and 128-flit messages. Since the results are very similar, only the 16-flit results will be presented and discussed. Detailed analysis and simulation results of only uniform traffic are presented. Summarized data from the other traffic patterns is shown, but not analyzed in detail.

The simulation results clearly show that the selection function can have a significant effect on the performance of the routing algorithm. For all the 2D graphs, the x-axis is the average channel utilization and the y-axis is the average message latency experienced by the messages.

4.1 Uniform Traffic

Figures 1 – 4 depict the performance of the four routing algorithms using various selection functions with uniform traffic. For opt-y, no turn shows the best average message latency with higher utilization rates and sustains the highest traffic load before saturation. Some of the selection functions yield relatively poor performance. The saturation point for balanced varies by up to 15% depending on the selection function. Even at low channel utilization rates, the average message latency is affected by the selection function. For balanced, zigzag shows the best performance and no turn is the worst. This is the exact opposite of the performance with opt-y. For mad-y, the saturation point differs by about 17% depending on the selection function. The average message latency differs even at low rates of channel utilization and increases with the utilization rate. The performance of the selection functions falls into two distinct groups. For planar, the saturation behavior of the selection functions varies up to about 18%. Planar is the least adaptive of the four routing algorithms, but still shows a clear difference among selection functions.

4.2 Performance Analysis

So far, an implicit assumption has been made that the improved performance of some selection functions is due to a better balance of traffic in the network. We now justify this assumption with a brief illustration of the channel utilization with different selection functions.

A straightforward analysis [15] suggests that with uniform traffic the channel utilization should form a bell shape with peak utilization at the center. As the traffic concentrates in the center of the mesh, however, adaptive routing attempts to spread the traffic more evenly.

Figures 5 and 6 present the channel utilization characteristics for the best (zigzag) and the worst (no turn) selection functions for balanced with uniform traffic. The channel utilization for the South-bound channels is shown, but the results are similar for all four directions. The z-axis represents the channel utilization and the x-axis and y-axis show the relative position of each channel in the network. These 3D graphs illustrate the difference in traffic distribution and explain the cause of the performance difference. These graphs depict the case where the *average* channel utilization was approximately 12%. As the channel utilization increases, the imbalance becomes more extreme, but the overall pattern does not change.

Zigzag appears to have done an acceptable job of balancing the traffic. With uniform traffic, the traffic crossing the middle of the mesh will be higher than the traffic on the edges of the mesh, but at least the traffic is not concentrated in the center. No turn concentrates traffic in the center, and hence saturates relatively quickly. Note that no turn does not produce this behavior with all adaptive routing algorithms. For example, figure 7 shows the traffic distribution when no turn is used with opt-y. The difference is dramatic

Table 1: Routing Algorithm Restrictions

Algorithm	N or S	E or W	NE	NW	SE	SW
Opt-y	Y1, Y2	X1, X2	X1, X2, Y1, Y2	X1, X2, Y2	X1, X2, Y1, Y2	X1, X2, Y2
Balanced	Y1, Y2	X1, X2	X1, X2, Y1, Y2	X2, Y1	X1, Y2	X1, X2, Y1, Y2
Mad-y	Y1/Y2	X1, X2	Y1/X1, X2, Y2	X1, X2, Y1/Y2	Y1/X1, X2, Y2	X1, X2, Y1/Y2
Planar	Y1	X1, X2	X1, X2, Y1	X1, X2, Y2	X1, X2, Y1	X1 X2, Y2

and is due to the differing restrictions imposed by the two routing algorithms. Similar effects of traffic balance were observed with mad-y and planar.

4.3 Relative Effect of Selection Function Choice

Similar behavior was observed with the other traffic patterns. We now briefly consider whether the selection function can have more effect on performance than the routing algorithm does. Figures 8 – 12 depict the performance range of each routing algorithm for the five traffic patterns. In each graph, the best and worst performing selection function is shown for all four routing algorithms. The best and worst selection function is listed in table 2.

For uniform traffic, the best selection functions for balanced and opt-y have similar performance – opt-y has lower average message latency at high utilization rates, but saturates slightly earlier. The worst selection functions are noticeably inferior. In this case, which routing algorithm is best depends on the selection function. The best selection function for mad-y outperforms the worst choice for balanced and opt-y in terms of average message latency, but saturates sooner. Planar shows performance similar to mad-y, and outperforms mad-y and balanced depending on which selection functions are used. Transpose shows a similar distribution in performance. Matrix rotate and center-reflection traffic show a more substantial difference among the routing algorithms. The best routing algorithm depends *entirely* on the selection function. Tree-reduction traffic shows a different result – the relative ranking of opt-y and balanced is interchangeable, but always superior to mad-y and planar. Similarly, the ranking between mad-y and planar depends on the selection function.

4.4 Summary

For all traffic patterns and all routing algorithms, the selection function choice results in a significant performance impact. Opt-y always performs best with the no turn selection function, because no turn reduces blocking in different rows and columns. By maximizing the routing choices, opt-y increases routing options. no turn helps control the use of these options. For balanced, zigzag consistently achieves the best performance. Zigzag complements the routing restrictions imposed by balanced, which seeks to distribute the restrictions evenly across the network. Unfortunately, the situation with mad-y and planar is more murky. No single selection function provides consistently good performance. Mad-y performs best when used with either no turn or y1,y2,x, but sometimes there is a significant performance difference between the two. Planar usually performs best with no turn, although y1,y2,x is slightly better for transpose and tree reduction.

5 Conclusion

We have presented a detailed simulation study of various selection functions with four fully adaptive mesh routing algorithms. The four routing algorithms were chosen based on their diverse characteristics in order to study the relationship between selection function and adaptive routing under varying circumstances. The simulation results clearly support our hypothesis that the choice of selection function can have a substantial impact on the average message latency and saturation behavior of an adaptive routing algorithm. The results also show, however, that the best selection function depends on the routing algorithm.

Because traffic tends to concentrate in the center of a mesh, the selection function and routing algorithm must cooperate to move traffic away from the center. Since minimal routing is used in these simulations, the ability to avoid the center may seem limited. Nevertheless, the results show that balancing the network traffic is possible.

The simulations provide evidence that the choice of selection function can effect performance more than the choice of routing algorithm. Of course, a complete comparison of routing algorithms would require careful consideration of their implementation complexity and resulting cycle times, but the relative variance in performance among the selection functions seems unlikely to change.

The 3D graphs provide a picture of the relationship between the selection function and the utilization of channels. The graphs show that the choice of selection function has a dramatic effect on the distribution of network traffic. Selection functions that enhance the balance lead to better saturation behavior and lower message latency. Studies are currently underway to fully explain this phenomenon. A thorough understanding of this issue may lead to the design of even better performing selection functions.

Acknowledgments

The author thanks Sumit Roy and Chengzhong Xu for insightful comments and suggestions of future work.

Table 2: Best and Worst Performing Selection Functions

Traffic Pattern	Opt-y		Balanced		Mad-y		Planar	
	Best	Worst	Best	Worst	Best	Worst	Best	Worst
Uniform	no turn	zigzag	zigzag	no turn	y1,y2,x	x,y2,y1	no turn	zigzag
Matrix Transpose	no turn	x,y2,y1	zigzag	x1,x2,y1,y2	y1,y2,x	x,y2,y1	y1,y2,x	x,y1,y2
Matrix Rotate	no turn	zigzag	zigzag	no turn	no turn	x,y2,y1	no turn	x,y1,y2
Center Reflection	no turn	zigzag	zigzag	no turn	no turn	zigzag	no turn	zigzag
Tree Reduction	no turn	y2,y1,x	zigzag	y1,y2,x1,x2	y1,y2,x	x,y1,y2	y1,y2,x	x,y1,y2

References

[1] S. Badr and P. Podar. An Optimal Shortest-Path Routing Policy for Network Computers with Regular Mesh-Connected Topologies. *IEEE Transactions on Computers*, 38(10):1362–1371, October 1989.

[2] R. V. Bopanna and S. Chalasani. New Wormhole Routing Algorithms for Multicomputers. In *International Parallel Processing Symposium*, pages 419–423, 1993.

[3] A. A. Chien. A Cost and Speed Model for k-ary n-cube Wormhole Routers. In *Proceedings of Hot Interconnects '93*, August 1993.

[4] A. A. Chien and J. H. Kim. Planar-Adaptive Routing: Low-cost Adaptive Networks for Multiprocessors. In 19^{th} *Annual International Symposium on Computer Architecture*, pages 268–277, 1992.

[5] W. J. Dally. Virtual-Channel Flow Control. *IEEE Transactions on Parallel and Distributed Systems*, 3(2):194–205, March 1992.

[6] W. J. Dally and H. Aoki. Deadlock-Free Adaptive Routing in Multicomputer Networks Using Virtual Channels. *IEEE Transactions on Parallel and Distributed Systems*, 4(4):466–475, April 1993.

[7] W. J. Dally and C. L. Seitz. Deadlock-Free Message Routing in Multiprocessor Interconnection Networks. *IEEE Transactions on Computers*, C-36(5):547–553, May 1987.

[8] J. Duato. On the Design of Deadlock-Free Adaptive Routing Algorithms for Multicomputers: Design Methodologies. In *Parallel Architectures and Languages Europe 91*, pages 390–405, June 1991.

[9] J. Duato. Improving the Efficiency of Virtual Channels with Time-Dependent Selection Functions. In *Parallel Architectures and Languages Europe 92*, pages 635–650, 1992.

[10] W. Feng and K. Shin. Impact of Selection Functions on Routing Algorithm Performance in Multicomputer Networks. In *ACM International Conference on Supercomputing*, pages 132–139, 1997.

[11] C. Glass and L. M. Ni. Maximally Fully Adaptive Routing in 2D Meshes. In *International Conference on Parallel Processing*, volume I, pages 101–104, 1992.

[12] C. Glass and L. M. Ni. The Turn Model for Adaptive Routing. *Journal of the Association for Computing Machinery*, 41(5):874–902, September 1994.

[13] P. Mohapatra. Wormhole Routing Techniques for Directly Connected Multicomputer Systems. *ACM Computing Surveys*, 30(3):374–410, September 1998.

[14] L. M. Ni and P. K. McKinley. A Survey of Wormhole Routing Techniques in Direct Networks. *IEEE Computer*, 26(2):62–76, February 1993.

[15] R. Rao. Utilization Imbalance in Wormhole Routed Networks. Master's thesis, The Ohio State University, 1994.

[16] H. Schwetman. CSIM: A C-Based, Process-Oriented Simulation Language. In *Proceedings of the 1986 Winter Simulation Conference*, pages 387–396, 1986.

[17] L. Schwiebert and R. Bell. The Impact of Output Selection Function Choice on the Performance of Adaptive Wormhole Routing. In 10^{th} *International Conference on Parallel and Distributed Computing Systems*, pages 539–544, 1997.

[18] L. Schwiebert and D. N. Jayasimha. Optimal Fully Adaptive Minimal Wormhole Routing for Meshes. *Journal of Parallel and Distributed Computing*, 27(1):56–70, May 1995.

[19] J. Upadhyay, V. Varavithya, and P. Mohapatra. A Traffic-Balanced Adaptive Wormhole-Routing Scheme for Two-Dimensional Meshes. *IEEE Transactions on Computers*, 46(2):190–197, February 1997.

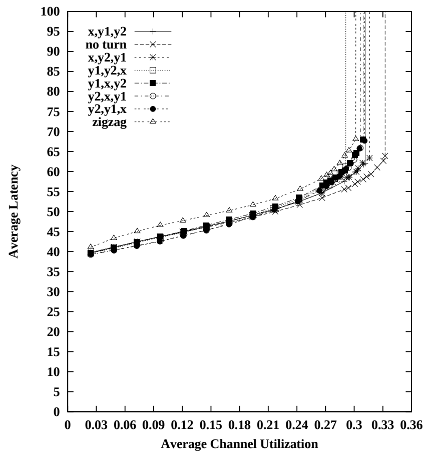

Figure 1: Opt-y – Uniform

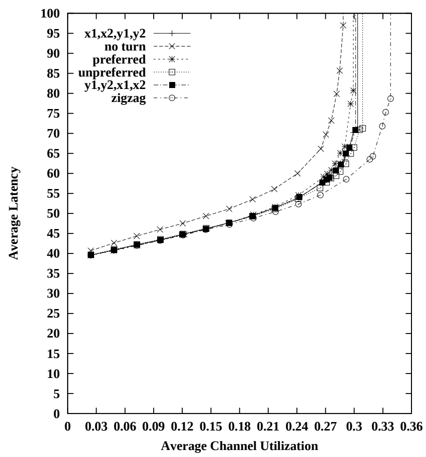

Figure 2: Balanced – Uniform

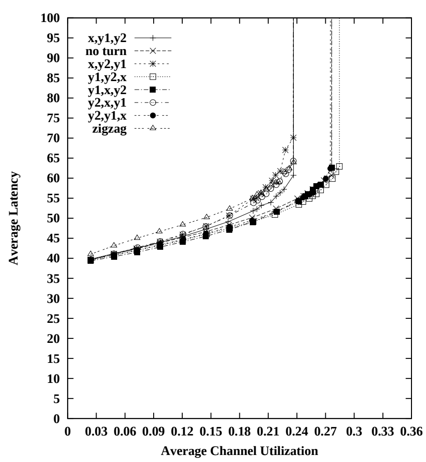

Figure 3: Mad-y – Uniform

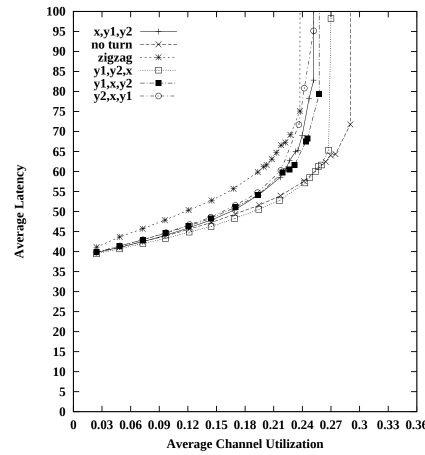

Figure 4: Planar – Uniform

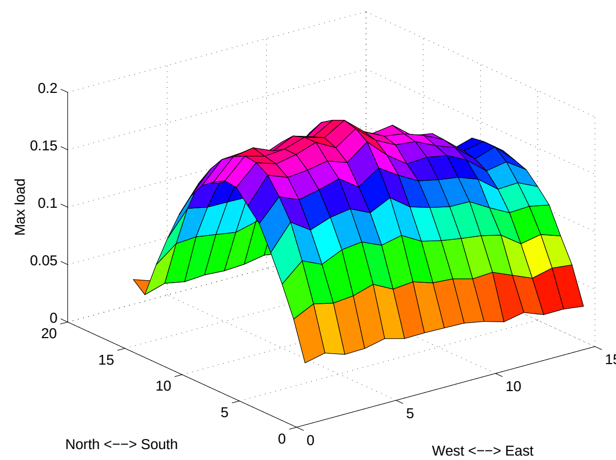

Figure 5: Channel utilization for balanced and zigzag

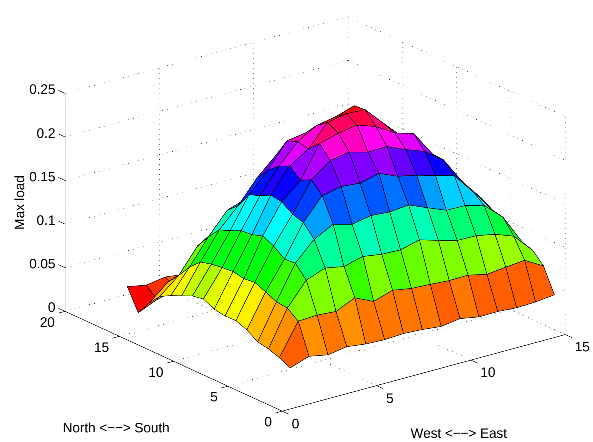

Figure 6: Channel utilization for balanced and no turn

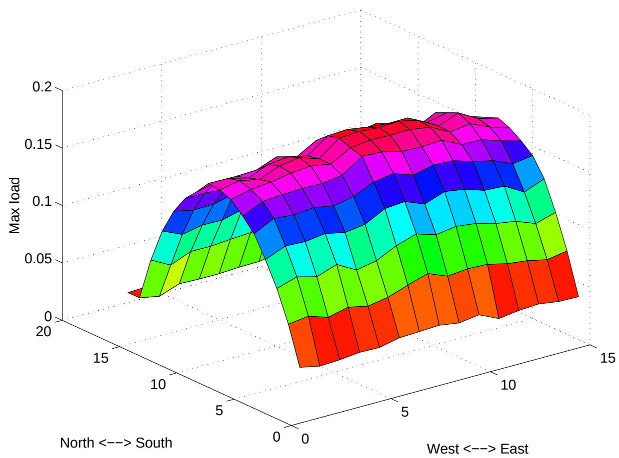

Figure 7: Channel utilization for opt-y and no turn

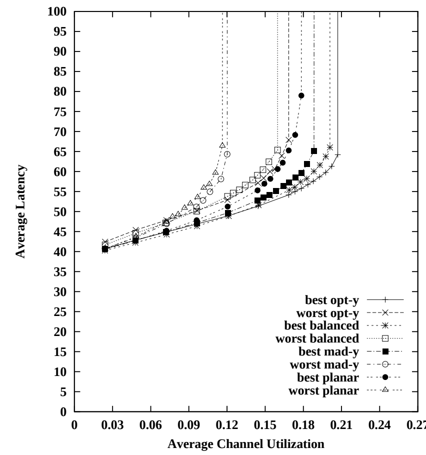

Figure 10: Comparison – Matrix Rotation

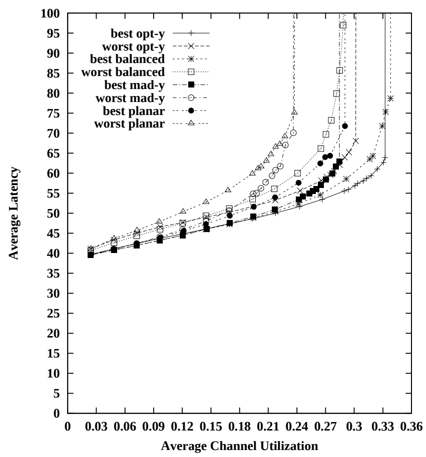

Figure 8: Comparison – Uniform

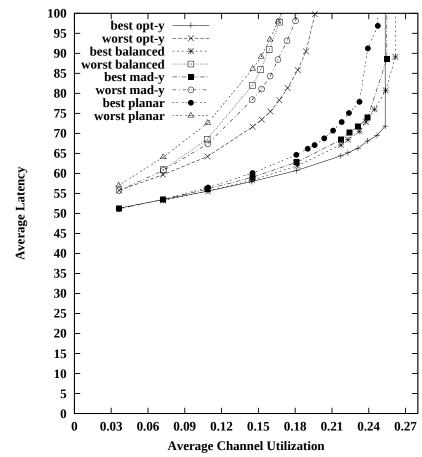

Figure 11: Comparison – Center-Reflection

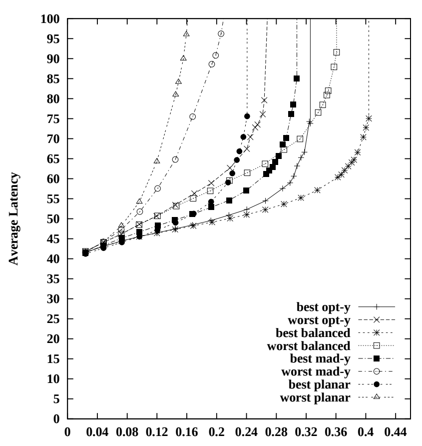

Figure 9: Comparison – Transpose

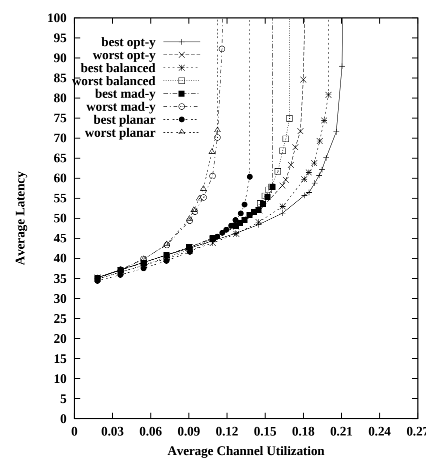

Figure 12: Comparison – Tree-Reduction