THESIS ABSTRACT

THE OHIO STATE UNIVERSITY
GRADUATE SCHOOL

NAME: Roshan M. Rao QUARTER/YEAR: Autumn 1994

DEPARTMENT: Computer and Information Science DEGREE: M.S.

ADVISER'S NAME: D. N. Jayasimha

TITLE OF THESIS: Utilization Imbalance in Wormhole Routed Networks

Wormhole routing is a popular switching technique for interprocessor communication in multiprocessor systems. It has been experimentally observed that wormhole routed networks saturate at low to moderate loads. Adaptive routing algorithms with multiple virtual channels do not help very much in improving the saturation behavior, since the restrictions they place on the choice of channels for achieving deadlock freedom create an *imbalance* in the utilization of channels. This imbalance forces a few links to attain full utilization first and the entire network saturates, though most of the links have yet to carry their full capacity of traffic. In this thesis, we study this problem analytically, define a way to measure imbalance, and use the results of this analysis to present an output selection policy to alleviate imbalance. Reducing imbalance has the twin benefits of improving the saturation behavior of the network and reducing message latencies. We substantiate this claim through simulation by applying this policy to two routing algorithms.

Adviser's Signature

Utilization Imbalance in Wormhole Routed Networks

A THESIS

Presented in Partial Fulfillment of the Requirements for

the Degree Master of Science in the Graduate

School of The Ohio State University

By

Roshan Rao, B.E.,

* * * * *

The Ohio State University

1994

Master's Examination Committee:

Professor D. N. Jayasimha

Professor D. K. Panda

Approved by

Adviser

Department of Computer and Information Science

To my parents, with love and gratitude

Acknowledgements

I am deeply grateful to Professor D. N. Jayasimha, my thesis advisor for his guidance and insight. I thank Professor D. K. Panda for serving on my thesis committee. I was fortunate to be a member of Medusa, a very supportive research group - my thanks to Loren Schwiebert for many fruitful discussions, to Jeff May and to Kant Patel for helping develop the simulator that I used to test my ideas and for some good suggestions. I also wish to thank Manas Mandal for helping out with the intricacies of LaTeX and CSIM.

VITA

February 5, 1970	Born - New Delhi, India.
1991	Bachelor of Engineering (Electronics & Communication Engg.) Univ. Visvesvaraya College of Engg., Bangalore, India.
1991-1992	Research Assistant, Indian Institute of Science, Bangalore, India.
1992-present	Graduate Teaching Associate, The Ohio State University, Columbus, Ohio, USA.

Fields of Study

Major Field: Computer and Information Science

Studies in:

Parallel Computing
Operating Systems

Table of Contents

List of Tables

List of Figures

CHAPTER I

INTRODUCTION

1.1 Background

Parallel processing is considered to be the most promising technology to achieve high performance computing. Massively parallel machines are organized as a collection of many nodes, each with its own processor, memory and other devices. In a direct network, these nodes are connected to each other in a suitable topology. Popular topologies include the 2D and the 3D meshes, hypercubes and tori. The performance of the system as a whole is heavily dependent on the time required to move information between different nodes and this is where the technique used for communication plays an important part. Packet switching and circuit switching have been used in the past. *Wormhole routing* [2] has however gained popularity in recent years.

1.2 Wormhole Routing

Many modern parallel computers like the Intel Paragon, the MIT J-machine, the Symult 2010, the Caltech MOSAIC and the nCUBE-2 have adopted wormhole routing. In this switching technique, a message packet is divided into a number of packets called *flits*. The *header* flit, the beginning of the message, carries information required to route the message. The remaining flits bear the information being transferred. The header flit controls the route taken by the entire message. As it advances along a particular route, the rest of the flits follow in a pipelined manner as shown in Figure 1. If the header finds that a channel is in use, it is blocked until the channel becomes free. The trailing flits are stored in flit buffers along the route. Thus, this technique eliminates the need for large packet buffers at each node and offers a network latency almost independent of path length for long messages under negligible contention. The path taken by the header and hence by the rest of the message is decided by the *routing algorithm* used by the network. Routing algorithms may be *static* (also known as *deterministic*) (e.g., X-Y Routing) where the path taken by the message is independent of current network conditions or *adaptive* (e.g., West-First Routing [4], Optimal Fully Adaptive Routing [5], etc.) where multiple paths may be provided so that the best route (according to dynamic network conditions) can be chosen. The algorithm imposes some constraints on the route taken by the message so that deadlock doesn't occur.

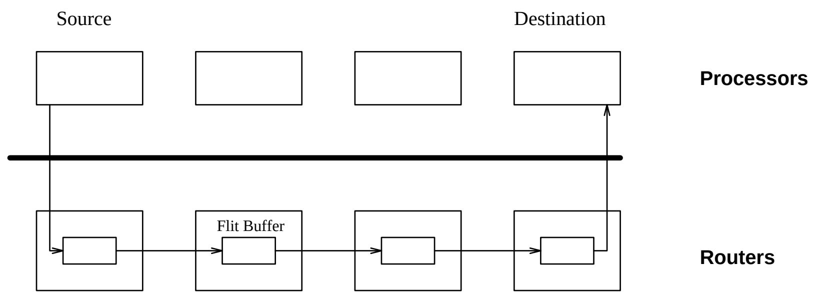

Figure 1: Wormhole routing

1.3 Problem Statement

It has been experimentally observed that wormhole-routed networks saturate at low to moderate loads. Hence the overall network utilization remains low. Adaptive routing algorithms with multiple virtual channels do not completely alleviate this problem, since the restrictions they place on the choice of channels for achieving deadlock freedom create an *imbalance* in the utilization of channels. This imbalance forces a few links to attain full utilization first and because of the nature of wormhole routing, the entire network saturates, though most of the links have yet to carry their full capacity of traffic. We study this problem analytically, define a way to measure imbalance, and use the results of this analysis to present an output selection policy to alleviate imbalance.

1.4 Organization of the Thesis

In Chapter 2, we describe a simple analytical model to carry out a link-level study of potential loads on a 2D mesh. This technique can be applied to other topologies, also. We apply it to many routing algorithms and graphically portray the imbalances. Also, the results of the analysis are presented in a way that gives an insightful picture of the static/adaptive policies of the routing algorithms.

In Chapter 3, we introduce the notion of "Imbalance" and a way to measure this quantity. Some simulation results to back up our earlier analytical observations are presented.

In Chapter 4, we describe how an intelligent selection policy may be fashioned out of the raw data gained from the analysis of Chapter 2. A method to alleviate imbalance by classifying links into different regions based on utilization is presented.

In Chapter 5, we present results of implementing this scheme on a simulator. The results show an improvement in the saturation behavior of the network, an increase in utilization and a decrease in latency. We generally get good results at higher injection rates.

In Chapter 6, we conclude the thesis by presenting a summary of the work and also give some pointers to future research in this area.

1.5 Terminology

The following terms appear frequently in the thesis :

1. **Latency :** The time elapsed after the head of a message has entered the network at the source node until its tail emerges from the network at the destination.

2. **Utilization :** For a given channel, this is the percentage of time when the channel is being used. The *overall* utilization refers to the average of the values for each channel.

3. **Steady state :** This represents a condition of the network in which the injection rate is equal to the consumption rate.

4. **Saturation :** This represents a condition of the network when the consumption rate consistently remains much lower than the injection rate.

CHAPTER II

ANALYSIS

2.1 Introduction

In this chapter, we describe an analytical technique to determine the potential utilization of each channel in a network. Message generation time is assumed to be uniformly randomized as is the choice of destinations. Using this technique, routing algorithms such as X-Y (static), West-First, North-Last, Negative-First (examples of partially adaptive algorithms [4]) and Optimally Fully Adaptive (an example of a fully adaptive algorithm [5]) are studied and compared.

2.2 The Analytical Technique

The aim of this analysis is to gain an insight into the potential utilization of each channel of a wormhole routed network. The analysis that follows is for a 2D mesh(Figure 2), but the technique can be used for any topology. We want to determine the contribution of each node to the traffic on each link.

Each node generates a message to every other node (except itself) with a probability of $1/(N^2 - 1)$ where the mesh is of size NXN. We assume that there are potentially infinite number of virtual channels for each physical channel. Thus many messages could be using the same link simultaneously - *blocking* of channels is not considered. Given a routing algorithm, the traffic that would travel on each link for a given direction is calculated. The figure that we arrive at, $util(i)$, is an upper bound on the load that link i would ever be subjected to and could be a good indicator of the potential utilization of the link. This gives a picture of the potential load on each link thus portraying any imbalances in the network.

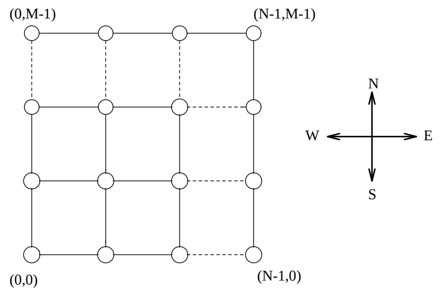

Figure 2: A MXN mesh

2.2.1 General Technique to Calculate $util(i)$:

Let $\{< s_1, d_1 >, < s_2, d_2 >, \ldots, < s_n, d_n >\}$ be the list of Source-Destination pairs that use link i when messages are routed according to the routing algorithm being studied. Let $actual(s_j, d_j, i)$ be the number of routes from s_j to d_j suggested by the routing algorithm that actually contain link i and $total(s_j, d_j)$ be the total number of possible paths from s_j to d_j. These functions are easily calculated from the recurrences shown in the next section.

Then we define,

$$util(i) = 1/(N^2 - 1) * \sum_{j=1}^{n} \frac{actual(s_j, d_j, i)}{total(s_j, d_j)} \tag{2.1}$$

For a deterministic decision,

$$actual(s_j, d_j, i) = total(s_j, d_j) \tag{2.2}$$

therefore,

$$util(i) = n/(N^2 - 1) \tag{2.3}$$

2.2.2 Recurrence

In this section, we derive some mathematical equations which are used to calculate the functions $actual()$ and $total()$.

Referring to Figure 2, $m + n$ represents the number of links needed for a minimal path while m or n represents the distance in terms of the number of links between $(0, 0)$ and (n, m) in the respective dimension. Using the formula for combinations, the total number of minimal paths from $(0, 0)$ to (n, m) where $0 < n \leq N$ and $0 < m \leq M$ is given by

$$P(n, m) =^{m+n} C_m = \frac{(m + n)!}{m!n!} \tag{2.4}$$

For a x-link $(lx, ly) \longrightarrow (lx + 1, ly)$, let $\Psi(lx, ly, lx + 1, ly)$ be the number of *possible* routes that use the link when a message is routed from $(0, 0)$ to $(n - 1, m - 1)$. So, we have,

$$\begin{aligned} \Psi(lx, ly, lx + 1, ly) + \Psi(lx, ly, lx, ly + 1) &= \Psi(lx - 1, ly, lx, ly) + \\ &\quad \Psi(lx, ly - 1, lx, ly) \end{aligned} \tag{2.5}$$

That is, total number of routes entering node (lx, ly) = total number of routes exiting node (lx, ly). Also, the number of routes exiting the node is divided in the ratio of the number of remaining nodes (or links) in the x and y directions. Using this result, we can recursively describe Ψ for an outgoing link as a fraction of the sum of Ψ values for the incoming links.

This leads us to the following formula:

$$\Psi(lx, ly, lx+1, ly) = \begin{cases} \frac{(n-1-lx)}{(m+n-2-lx-ly)}(\Psi(lx-1, ly, lx, ly)+ \\ \Psi(lx, ly-1, lx, ly)) & \text{if } lx \neq 0 \text{ and } ly \neq 0 \\ \frac{(n-1)}{(m+n-2)}P(n,m) & \text{if } lx = 0 \wedge ly = 0 \\ \frac{(n-1)}{(m+n-2-ly)}\Psi(lx, ly-1, lx, ly) & \text{if } lx = 0 \\ \frac{(n-1)}{(m+n-2-lx)}\Psi(lx-1, ly, lx, ly) & \text{if } ly = 0 \end{cases} \tag{2.6}$$

$\Psi(lx, ly, lx, ly+1)$ for the y-link originating from (lx, ly) can be described in a similar manner. These equations enable us to calculate $actual()$ and $total()$ and hence $util(i)$ for each link i.

The results obtained above are applied to some static and adaptive routing algorithms. To simplify calculations, the algorithms are analyzed along each of the 4 directions of the mesh separately. The results which represent utilizations for each link are graphically portrayed in 3D. In the graphs, the x-axis represents the West$\leftrightarrow$East dimension, the y-axis represents the North$\leftrightarrow$South dimension and the z-axis is an indicator of the potential utilization of the channel $(x, y) \longrightarrow (x+1, y)$, $(x, y) \longleftarrow (x+1, y)$, $(x, y) \longrightarrow (x, y+1)$ or $(x, y) \longrightarrow (x, y-1)$ depending on the direction being studied.

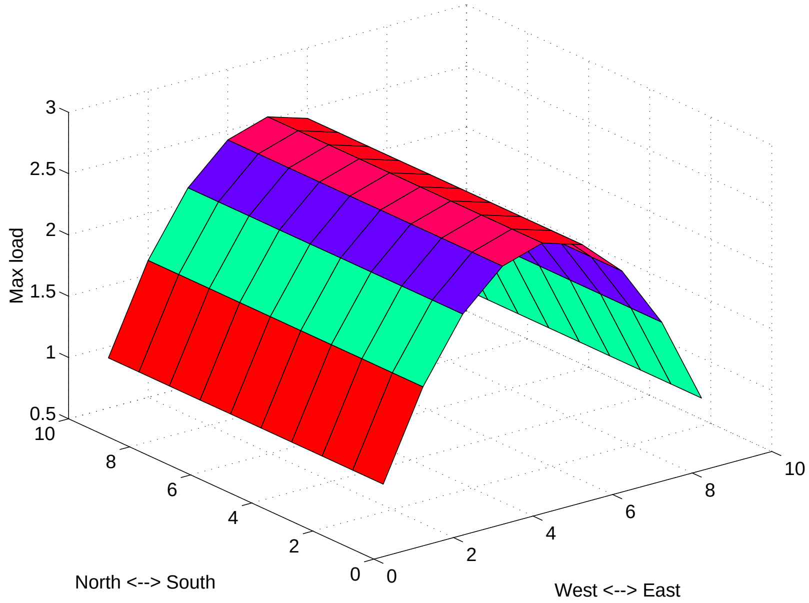

Figure 3: X-Y Routing - Any Direction

2.3 X-Y Routing

This is a simple, deterministic routing algorithm which is an example of dimension order routing. The message is routed completely in the x-dimension first and then in the y-dimension.

Consider the x-links traveling in the East direction, first. Consider a link $(i, j) \longrightarrow (i+1, j)$. Its sources are all nodes (a, j) where $0 \leq a \leq i$. The destinations are (x, y) where $(i+1) \leq x \leq (N-1)$ and $0 \leq y \leq (M-1)$. Routing is deterministic.

Using these source-destination pairs and the formulas from the previous sections, we study the network. The results are in Figure 3.

Because of the symmetry of X-Y Routing, it turns out that the equations for x-links traveling West are identical to those given above. The same holds true for both

Northbound and Southbound y-links. Hence, the results are the same as for the first case.

2.4 West-First Routing

This is a partially-adaptive algorithm. The packet is routed first West, if necessary and then adaptively South, East and North. The analysis that follows assumes *minimal* routing though the proposed algorithm /citeglass:turn does not.

Consider the x-links for messages traveling in the East direction, first. Consider a link $(i, j) \longrightarrow (i+1, j)$. Its sources are all nodes (a, b) where $0 \leq a \leq i$ and $0 \leq b \leq (M-1)$. Now, depending on the relative positions of the source and the link, the following 3 cases present themselves:

1. $b < j$: The destinations are (x, y) where $(i+1) \leq x \leq (N-1)$ and $j \leq y \leq (M-1)$.

2. $b = j$: The destinations are (x, y) where $(i+1) \leq x \leq (N-1)$ and $0 \leq y \leq (M-1)$.

3. $b > j$: The destinations are (x, y) where $(i+1) \leq x \leq (N-1)$ and $0 \leq y \leq j$.

Due to the *adaptiveness* of the algorithm, there are many paths to choose for a given source-destination pair and the probability of choosing any particular path is the same.

The results are presented in Figure 2.3. Note that *unlike* X-Y Routing, the values for various y-levels are *different*.

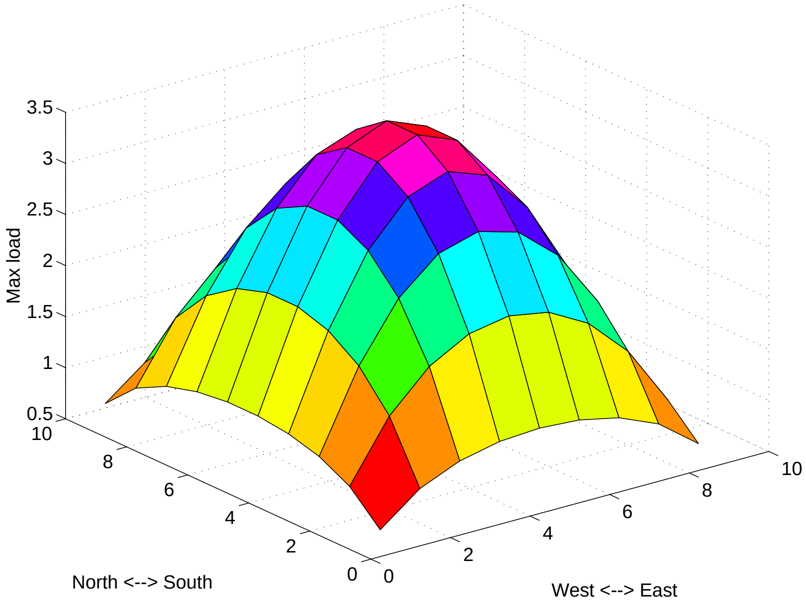

Figure 4: West-First - East

Consider a link $(i,j) \longleftarrow (i+1,j)$ in the West direction. Its sources are all nodes (a,j) where $i+1 \leq a \leq N-1$. The destinations are (x,y) where $0 \leq x \leq i$ and $0 \leq y \leq (M-1)$. Thus, the analysis is identical to X-Y Routing[1]. So, the results are identical to those in Figure 4.

The y-links in the South direction present a different picture. Consider a link $(i,j) \longrightarrow (i,j-1)$. Its sources are all nodes (a,b) where $0 \leq a \leq N-1$ and $j \leq b \leq M-1$. We have the following 2 cases:

1. $a \leq i$: The destinations are (x,y) where $i \leq x \leq (N-1)$ and $0 \leq y \leq j-1$ and routing is adaptive.

[1]Actually, all the three partially adaptive routing algorithms, when analyzed in this way, present a combination of deterministic and adaptive routing policies. Please see Table 1.

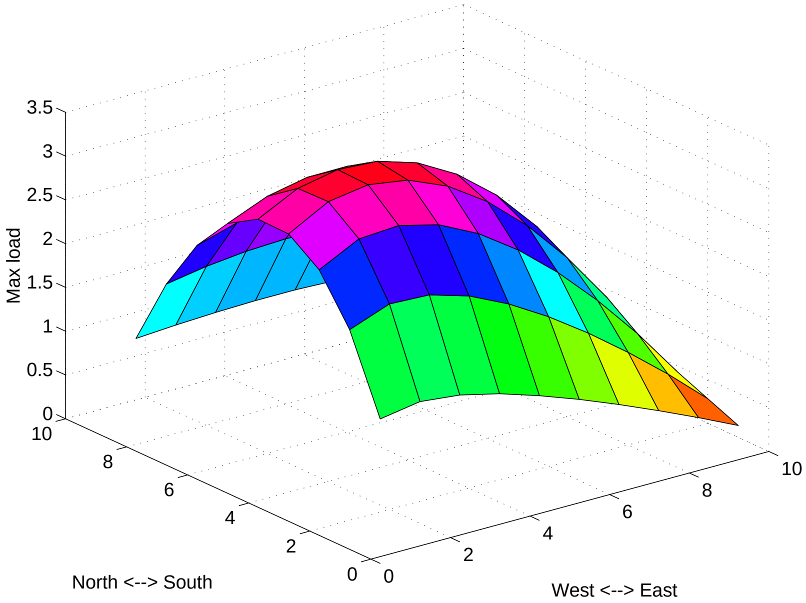

Figure 5: West-First - South

2. $a > i$: The destinations are (i, j) where $0 \leq y \leq (j-1)$. Note that this case is deterministic.

The results are shown in Figure 5.

Since the algorithm algorithm treats travel in the North and the South symmetrically, the analysis for y-links in the North direction is exactly the converse of that for travel in the South direction. The results are shown in Figure 6.

These 3D graphs give us a geographical perspective of the utilization of the network. It is clear that utilization is highest towards the center of the mesh and is much lower in the extremities. This non-uniformity is prevalent in other routing algorithms, too.

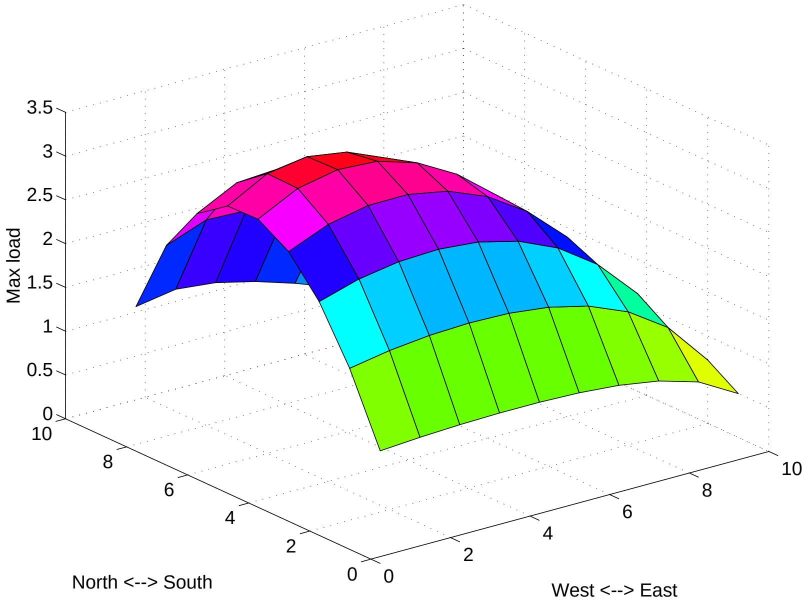

Figure 6: West-First - North

2.5 North-Last Routing

This is another partially adaptive algorithm. Here, the packet is routed first adaptively West, South and East and then North. Hence, the packet travels north last. The analysis that follows assumes *minimal* routing.

Consider a link $(i, j) \longrightarrow (i+1, j)$ in the East direction. Its sources are all nodes (a, b) where $0 \leq a \leq i$ and $0 \leq b \leq j$. Now, depending on the relative positions of the source and the link, we have the following 2 cases:

1. $b > j$: The destinations are (x, y) where $(i+1) \leq x \leq (N-1)$ and $0 \leq y \leq j$ and routing is fully-adaptive.

2. $b = j$: Here, we have two sub-cases:

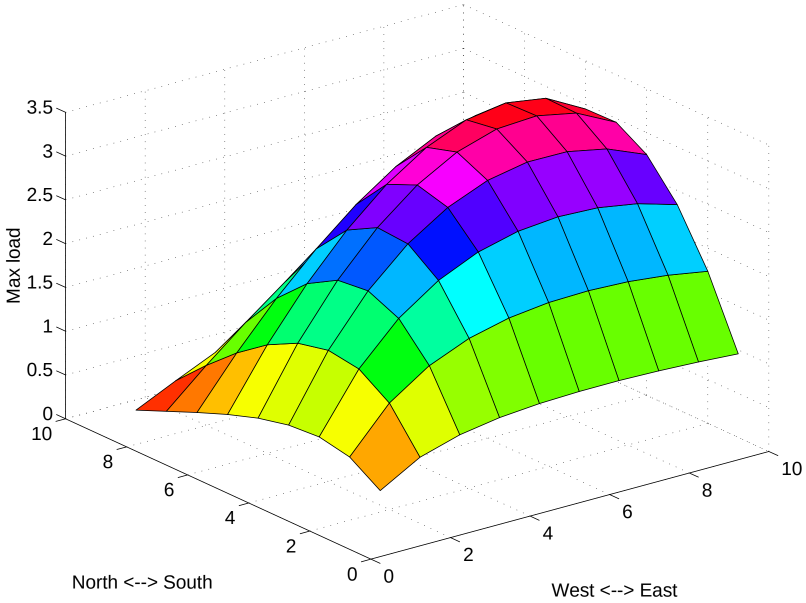

Figure 7: North-Last - East

(a) The destinations are (x, y) where $(i+1) \leq x \leq (N-1)$ and $j \leq y \leq M-1$ and routing is *not* adaptive.

(b) The destinations are (x, y) where $(i+1) \leq x \leq (N-1)$ and $0 \leq y \leq j$ and routing is fully-adaptive. The results are shown in Figure 7.

This algorithm treats travel in the West and the East in a symmetrical manner. So, the analysis for travel in the West is the exact converse of that for travel in the East and the results are shown in Figure 8.

Consider a link $(i, j) \longrightarrow (i, j-1)$ in the South direction. Its sources are all nodes (a, b) where $0 \leq a \leq N-1$ and $0 \leq b \leq j$. Again, depending on the relative positions of the source and the link, we have the following 3 cases:

1. $a < i$: The destinations are (x, y) where $i \leq x \leq (N-1)$ and $0 \leq y \leq (j-1)$.

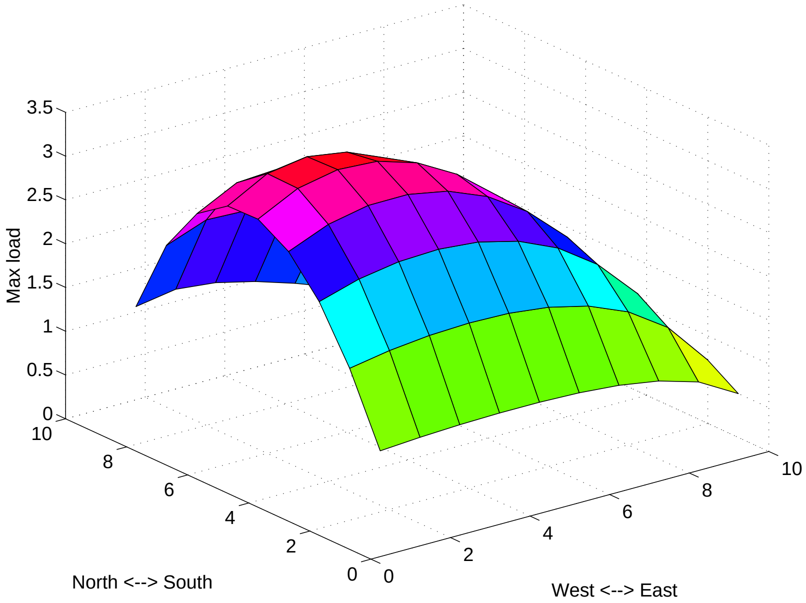

Figure 8: North-Last - West

2. $a = i$: The destinations are (x, y) where $0 \leq x \leq (N-1)$ and $0 \leq y \leq (j-1)$.

3. $a > i$: The destinations are (x, y) where $0 \leq x \leq i$ and $0 \leq y \leq (j-1)$.

All the above cases correspond to fully-adaptive routing and the results are identical to West-First routing in the East direction and are displayed in Figure 4.

We are faced with a different picture when we look at y-links in the North direction. Here, routing is deterministic. Consider a link $(i, j) \longleftarrow (i, j-1)$. Its sources are all nodes (i, b) where $0 \leq b \leq j-1$. And the destinations are (x, y) where $0 \leq x \leq (N-1)$ and $j \leq y \leq M-1$. So, the picture is identical to Figure 3.

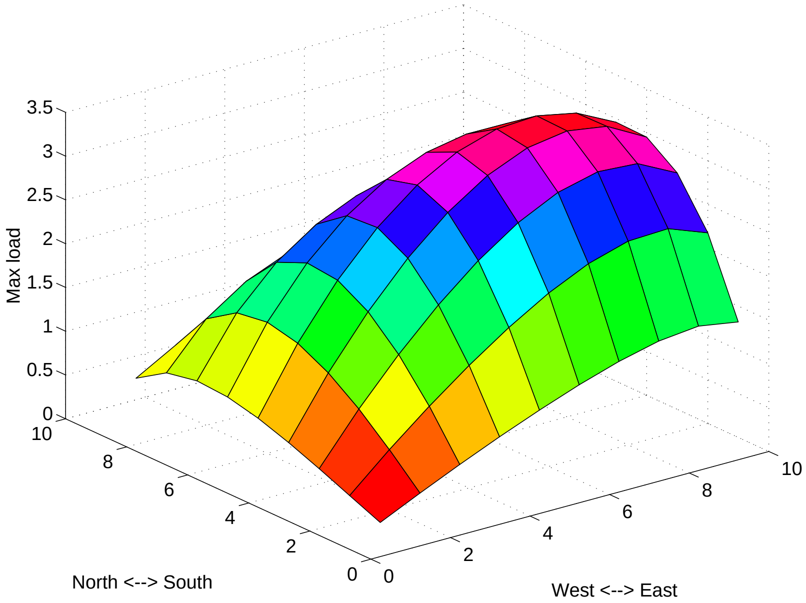

Figure 9: Negative-First - East

2.6 Negative-First Routing

In this partially adaptive routing algorithms, the packet is routed first adaptively West and South, and then adaptively East and North. So, it travels in the negative directions first. The analysis that follows assumes *minimal* routing.

Consider a link $(i,j) \longrightarrow (i+1,j)$ in the East direction. Its sources are all nodes (a,b) where $0 \le a \le i$ and $0 \le b \le (M-1)$. Now, depending on the relative positions of the source and the link, we have the following 2 cases:

1. $b > j$: The destinations are (x,j) where $(i+1) \le x \le (N-1)$ and routing is deterministic.

2. $b \le j$: The destinations are (x,y) where $i+1 \le x \le (N-1)$ and $0 \le y \le j$ and routing is adaptive.

The results are shown in Figure 9.

Next, consider a link $(i, j) \longleftarrow (i+1, j)$ in the West direction. Its sources are all nodes (a, b) where $i+1 \leq a \leq N-1$ and $0 \leq b \leq j$. Again, we have two cases:

1. $b > j$: The destinations are (x, j) where $0 \leq x \leq i$ and routing is fully-adaptive.

2. $b = j$: Here, there are two sub-cases:

   (a) The destinations are (x, y) where $(i+1) \leq x \leq (N-1)$ and $j \leq y \leq M-1$ and routing is *not* adaptive.

   (b) The destinations are (x, y) where $(i+1) \leq x \leq (N-1)$ and $0 \leq y \leq j$ and routing is fully-adaptive.

The results are shown in Figure 10.

Next, consider a link $(i, j) \longrightarrow (i, j-1)$ in the South direction. Its sources are all nodes (a, b) where $0 \leq a \leq N-1$ and $j \leq b \leq M-1$.Again, depending on the relative positions of the source and the link, we have the following 2 cases:

1. $a = i$: Here, we have two sub-cases:

   (a) The destinations are (x, y) where $0 \leq x \leq i$ and $0 \leq y \leq j$ and routing is fully-adaptive.

   (b) The destinations are (x, y) where $i < x \leq N-1$ and $0 \leq y \leq j$ and routing is deterministic.

2. $a > i$: The destinations are (x, y) where $0 \leq x \leq i$ and $0 \leq y \leq j$ and routing is fully-adaptive.

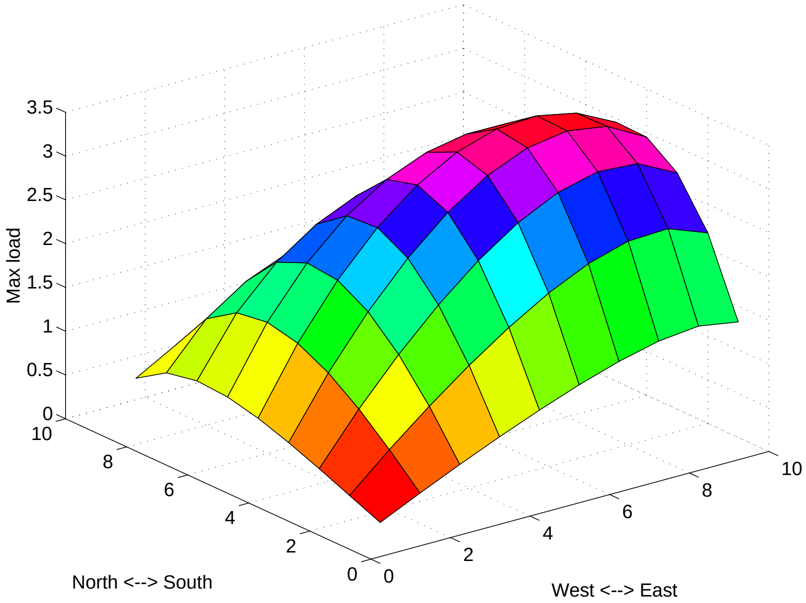

Figure 10: Negative-First - West

The results are shown in Figure 11.

Finally, consider a link $(i, j) \longleftarrow (i, j-1)$ in the North direction. Its sources are all nodes (a, b) where $0 \le a \le N-1$ and $0 \le b \le j$. And we have the following 2 cases:

1. $a \le i$: The destinations are (x, y) where $i \le x \le (N-1)$ and $j \le y \le M-1$ and routing is adaptive.

2. $a > i$: The destinations are (i, y) where $j \le y \le M-1$.

The results are shown in Figure 12.

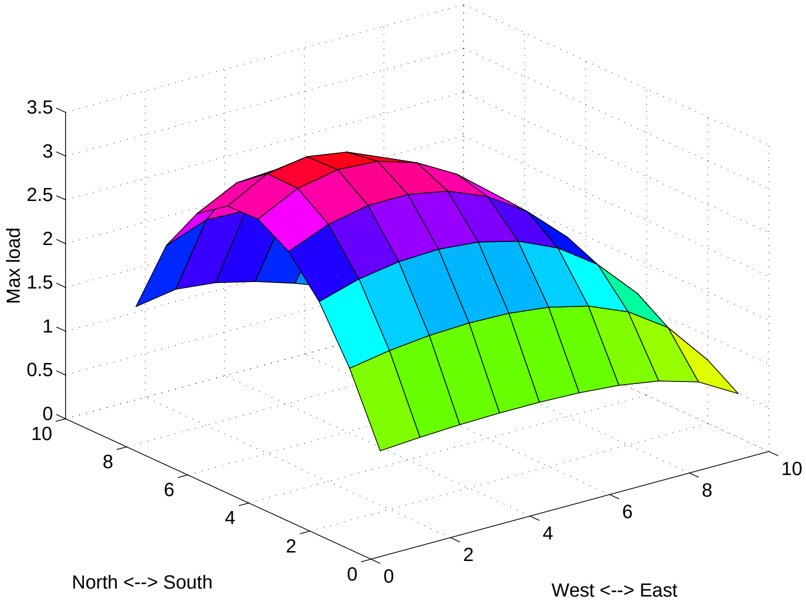

Figure 11: Negative-First - South

2.7 Optimal Fully Adaptive Routing

This is different from the other algorithms analyzed in that there are two virtual channels in the Y-direction. Hence we consider the following six directions: $X \rightarrow$, $X \leftarrow, Y_1 \uparrow, Y_2 \uparrow, Y_1 \downarrow$ and $Y_2 \downarrow$. The analysis that follows assumes *minimal* routing.

Since this algorithm is similar to the West-First routing algorithm without the presence of the extra virtual channels, many a time, the analysis is similar.

Next, we consider each link separately.

Consider x-links for messages traveling East. Consider a link $(i, j) \longrightarrow (i+1, j)$. Its sources are all nodes (a, b) where $0 \leq a \leq i$ and $0 \leq b \leq (M-1)$. Now, depending on the relative positions of the source and the link, we have the following 3 cases:

1. $b < j$: The destinations are (x, y) where $(i+1) \leq x \leq (N-1)$ and $j \leq y \leq$

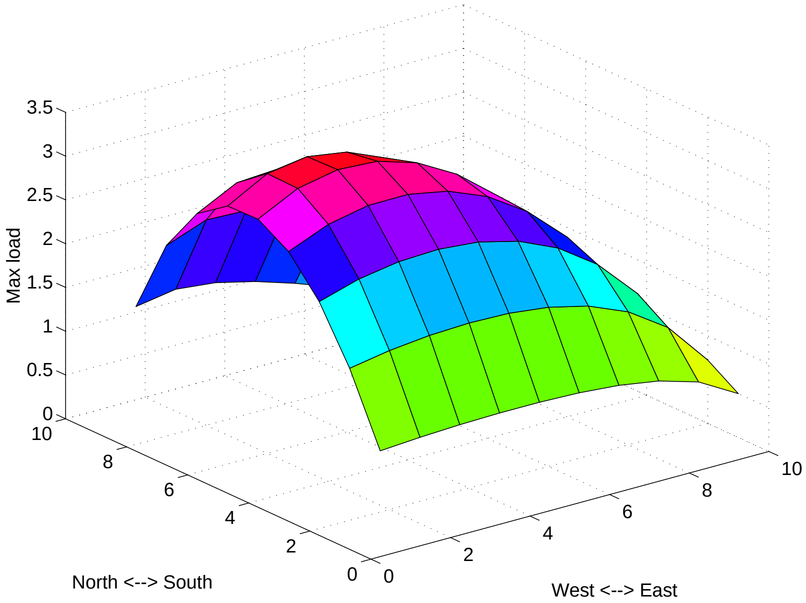

Figure 12: Negative-First - North

$(M-1)$.

2. $b = j$: The destinations are (x, y) where $(i+1) \leq x \leq (N-1)$ and $0 \leq y \leq (M-1)$.

3. $b > j$: The destinations are (x, y) where $(i+1) \leq x \leq (N-1)$ and $0 \leq y \leq j$.

Note that all these cases are fully adaptive. This is identical to West-First except that for each instance, the possible number of paths that use a particular link increases because of the two channels in the Y-direction. However it should be noted that the *total* number of paths increases too. Hence when we calculate the imbalance, these two balance each other out and the imbalance figures for this case are precisely identical to those for West-First. Figure 13 portrays the results.

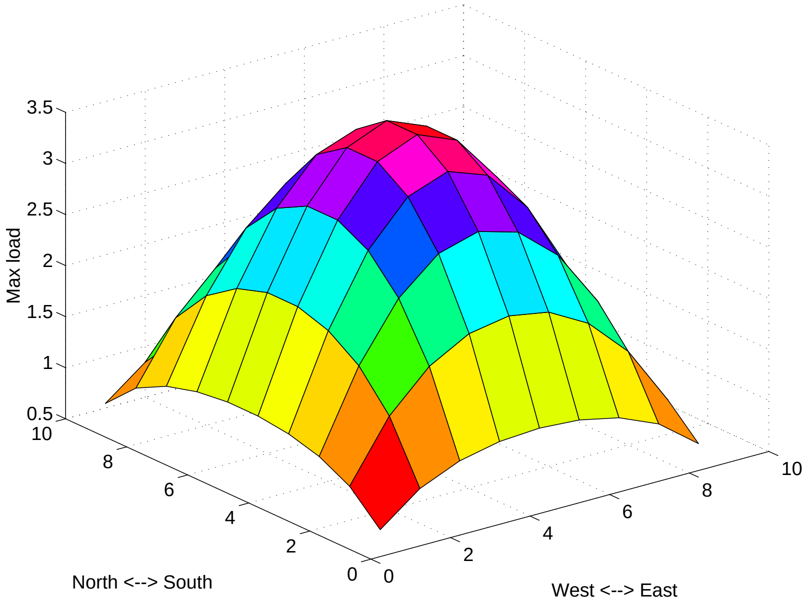

Figure 13: Optimally Fully Adaptive - East

Next, we analyze travel to the West. A key point to remember here is that $Y_1 \uparrow$ and $Y_1 \downarrow$ can be used only when the message does not need to route further West. Consider a link $(i, j) \longleftarrow (i+1, j)$. Its sources are all nodes (a, b) where $i+1 \leq a \leq N-1$ and $0 \leq b \leq (M-1)$[2] Now, depending on the relative positions of the source and the link, we have the following 3 cases:

1. $b < j$: The destinations are (x, y) where $0 \leq x \leq i$ and $j \leq y \leq (M-1)$.

2. $b = j$: The destinations are (x, y) where $0 \leq x \leq i$ and $0 \leq y \leq (M-1)$.

3. $b > j$: The destinations are (x, y) where $0 \leq x \leq i$ and $0 \leq y \leq j$.

---

[2]Note that this is different from West-First where this particular case was completely deterministic. Here, $Y_2 \uparrow$ and $Y_2 \downarrow$ which have no restrictions on them play an important role.

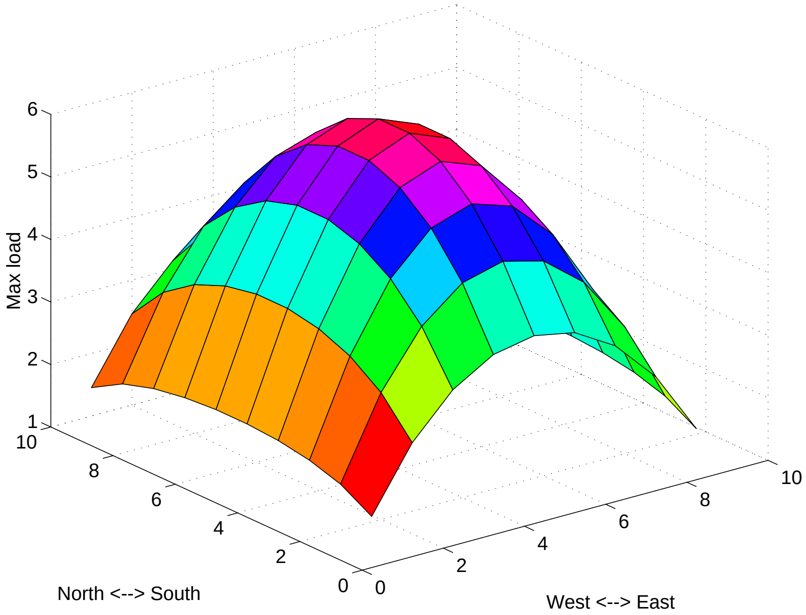

Figure 14: Optimally Fully Adaptive - West

Because of the restrictions on Y_1, it can be used only when the message has reached the column of its destination. So we have some kind of a constrained adaptive routing here:

1. When a message uses Y_2, routing is fully adaptive.

2. But, it can also use Y_1 when it reaches the column of its destination (does not need to route further West).

Figure 14 shows the analytical results.

There are 2 Virtual Channels in the Northern direction. Consider a channel $Y_1 \uparrow$ (link $(i, j) \longleftarrow (i, j-1)$). Its sources are all nodes (a, b) where $0 \leq a \leq N-1$ and $0 \leq b \leq j-1$. Now, depending on the relative positions of the source and the link, we have the following 3 cases:

1. $a < i$: The destinations are (x, y) where $i \leq x \leq (N-1)$ and $j \leq y \leq M-1$. Now, this case is fully adaptive and both Y-channels can be freely used[3].

2. $a = i$: The destinations are (x, y) where $0 \leq x \leq (N-1)$ and $j \leq y \leq M-1$. Here, depending on the need for Westward travel, we have two subcases:

   (a) $x < i$: Here, messages have to travel West. So that means that this link cannot be used.

   (b) $x \geq i$: This is fully-adaptive as there is no Westward travel.

3. $a > i$: The destinations are (i, y) where $j \leq y \leq M-1$ because no Westward travel is allowed. This is deterministic.

Hence we find that it's a combination of both static and adaptive routing (see Figure 15).

Analyzing the second channel in the North direction. Its sources are all nodes (a, b) where $0 \leq a \leq N-1$ and $0 \leq b \leq j-1$. Now, depending on the relative positions of the source and the link, we have the following 3 cases (see Figure 16):

1. $a < i$:The destinations are (x, y) where $i \leq x \leq (N-1)$ and $j \leq y \leq M-1$. This case is fully adaptive.

2. $a = i$: The destinations are (x, y) where $0 \leq x \leq (N-1)$ and $j \leq y \leq M-1$. This case is also fully adaptive.

[3]Similar to $X \rightarrow$

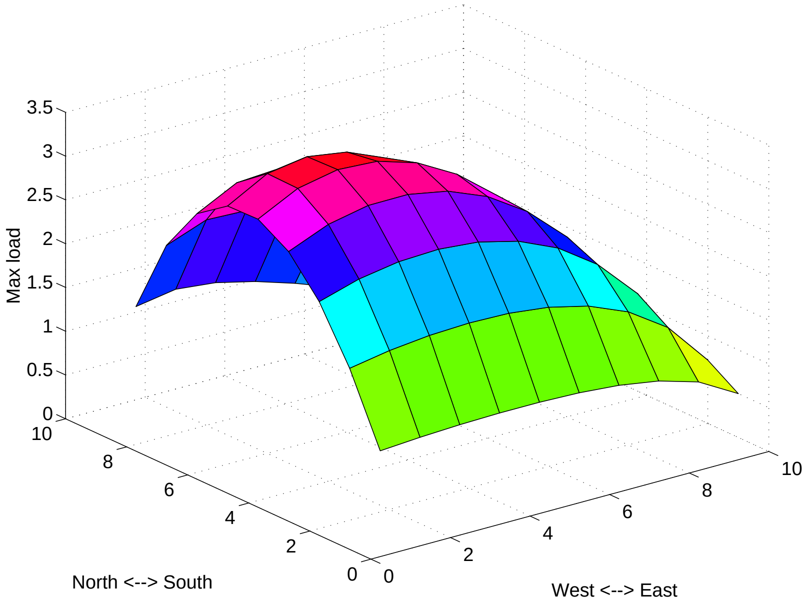

Figure 15: Optimally Fully Adaptive - North1

3. $a > i$: The destinations are (x, y) where $0 \leq x \leq i$ and $j \leq y \leq M - 1$. This case is also fully adaptive.

There are 2 Virtual Channels in the South too. Because of the North-South symmetry of this algorithm, we find that when analyzed, $Y_1 \downarrow$ represents the exact converse of $Y_1 \uparrow$. Similarly $Y_2 \downarrow$ is a mirror-image of $Y_2 \uparrow$. Figures 17 and 18 give a graphical perspective.

2.8 Observations

2.8.1 Directional Analysis

Observe that each partially-adaptive algorithm, when studied direction-wise is actually *fully adaptive (FA)* in some directions , *deterministic (DT)* in others, and a combination of the two *(FA/DT)* in yet others. This is shown in Table 1. The varia-

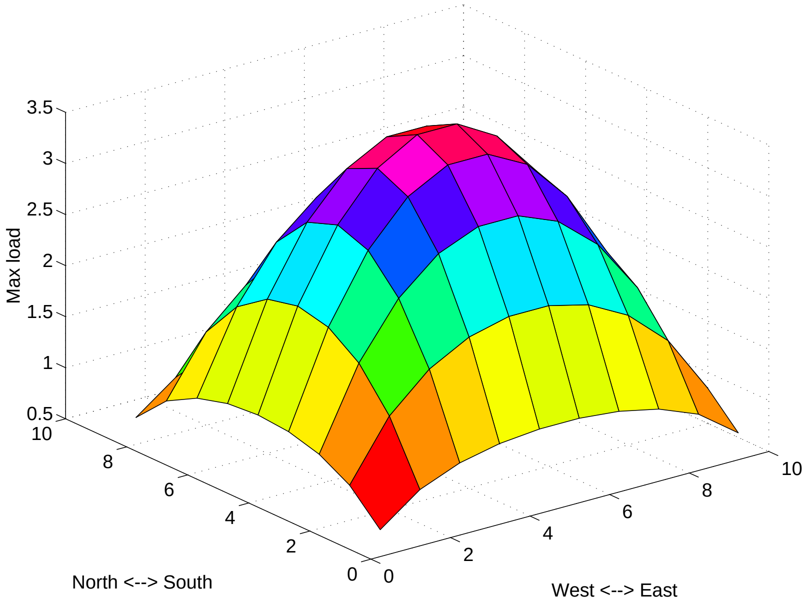

Figure 16: Optimally Fully Adaptive - North2

Table 1: Routing Algorithms - Directional Analysis

Algorithm	$X \rightarrow$	$X \leftarrow$	$Y1 \downarrow$	$Y2 \downarrow$	$Y1 \uparrow$	$Y2 \uparrow$
X-Y	DT	DT	DT	-	DT	-
West-First	FA	DT	FA/DT	-	FA/DT	-
North-Last	FA/DT	FA/DT	FA	-	DT	-
Negative-First	FA/DT	FA/DT	FA/DT	-	FA/DT	-
Optimal F.A.	FA	FA/DT	FA/DT	FA	FA/DT	FA

tion in adaptivity, ranging from determinism to fully-adaptiveness is a consequence of the constraints imposed by the algorithms in order to prevent deadlock. For example, West-First prohibits turns *to* the West. So, it is deterministic in that direction. It allows turns from any direction *to* the East - therefore it is fully-adaptive in the East. Similar reasoning holds for the other directions and also for the other algorithms.

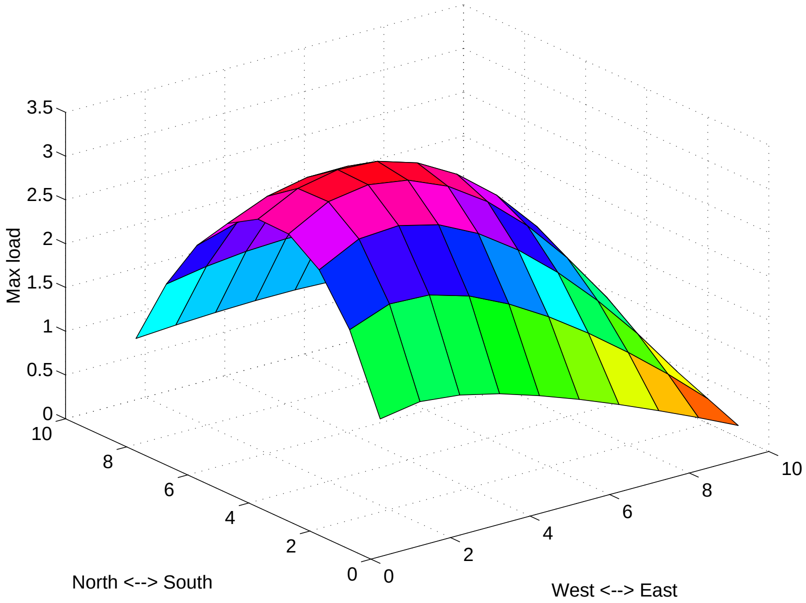

Figure 17: Optimally Fully Adaptive - South1

This decomposition is equivalent to the Turn Model [4] and is an excellent way to study the utilization of the network since it allows us to concentrate on one direction at a time.

2.8.2 Utilization

The utilization graphs bring out clearly the fact that the maximum potential load and hence the utilization for a channel varies drastically depending on the position of the channel in the mesh. The graphs for fully-adaptive routing are perfectly symmetrical and the links in the center have the highest load. The graphs for static routing show that the load for each link is the same as that as its corresponding link in the next row (if travel along the x-axis is being considered) or next column (if we are looking at travel along the y-axis) and the links in the central row (or column) have

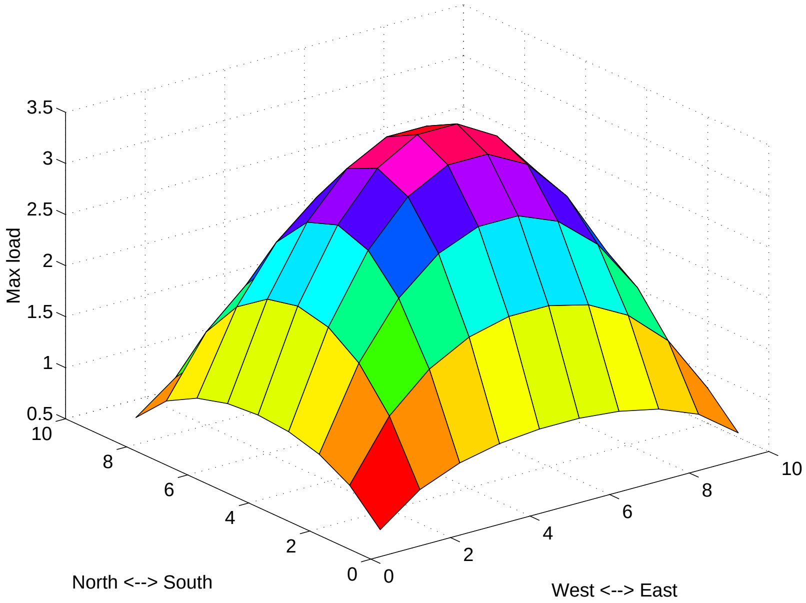

Figure 18: Optimally Fully Adaptive - South2

the highest loads. When routing is a combination of both static as well as adaptive policies, the links with highest utilization are not at the center of the mesh but away from it, depending on the direction of travel being considered. Generally, links in the peripheral regions are less utilized than those in the center.

For X-Y routing, for a $10X10$ mesh, the ratio of the highest load on any channel to lowest load on any channel is 2.8, while for West-First (or any of the other partially adaptive algorithms), this ratio is as high as 23. The variation in this ratio can be easily discerned from the graphs where the curve for X-Y routing is flatter than those for the adaptive routing algorithms. We can use our technique to find this ratio for any sized mesh. This kind of uneven load becomes more pronounced as the size of the mesh increases because the traffic increases and causes problems of early saturation and low utilization.

Significantly, the maximum loads for adaptive routing are *higher* than those for static routing. This suggests a possibility for improving the performance of adaptive routing, a possibility which is explored further in subsequent chapters.

CHAPTER III

IMBALANCE

3.1 Introduction

In this chapter, we explain the term "Imbalance" and a way to measure it. Also, the problems caused by imbalance are discussed. Finally, we show some simulation[1] results that bring out the disparities in network utilization.

3.2 Imbalance and Saturation

In a network, the saturation behavior is controlled by the links that attain full utilization first. When some links are loaded fully, messages start getting queued at links that feed into these. This behavior spreads rapidly through the network and the entire network saturates. It is clear from the figures in Chapter 2 that only a few links reach full utilization first and hence cause saturation of the entire network, while most of the other links have yet to realize their full potential. Thus this imbalance in utilization causes saturation at low to moderate loads. If it could be evened out, then the traffic would get balanced, the throughput would increase and the network would saturate at a much higher load than it currently does.

[1]Please refer Appendix A for details on the simulator.

3.3 Measuring Imbalance

Consider a 2D mesh with n rows and n columns. There are $2n(n-1)$ links in this network. Let the utilizations (normalized by the highest) for a given direction of travel be $u_1, u_2, u_3 ..., u_i, ...u_{2n(n-1)}$ and let there be VC virtual channels in the mesh. Then, we define the Imbalance, I, for that direction of travel to be

$$I = \frac{\sum_{i=1}^{2n(n-1)}(1-u_i)}{VC} \tag{3.1}$$

Each term in the summation measures the under-utilization of the link. If the link is fully-utilized, its contribution to the summation is 0. If all links are fully utilized, then $I = 0$ and there is no imbalance in the network.

Figure 19 represents a network which uses West-First routing with state ($I = 0$) in the East direction. From the analysis, the maximum load on any link traveling East was 3.7. If all links attain this level, then the network is balanced. Table 2 contains imbalance measures for the routing algorithms analyzed in Chapter 2. It is observed that deterministic routing has lesser imbalance than adaptive routing.

Table 2: Routing Algorithms - Imbalance

Algorithm	$X \rightarrow$	$X \leftarrow$	$Y1 \downarrow$	$Y2 \downarrow$	$Y1 \uparrow$	$Y2 \uparrow$
X-Y	0.27	0.27	0.27	-	0.27	-
West-First	0.46	0.27	0.44	-	0.44	-
North-Last	0.44	0.44	0.46	-	0.27	-
Negative-First	0.44	0.44	0.44	-	0.44	-
Optimal F.A.	0.46	0.38	0.44	0.44	0.46	0.46

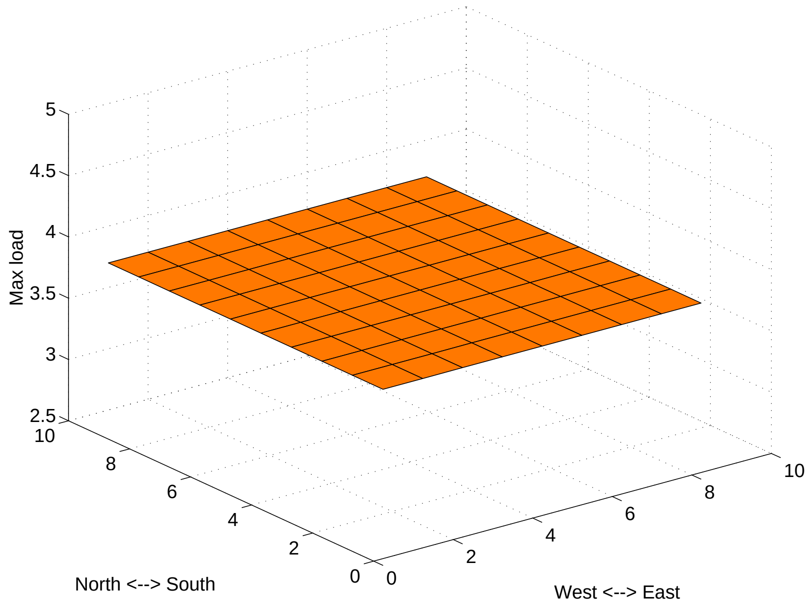

Figure 19: A Network with no Imbalance

3.4 Correspondence between Analytical and Simulation Results

We simulated many different routing algorithms varying the injection rates. At higher injection rates, the results obtained corresponded to our analytical findings. For example, Figure 20 represents the utilizations (in the East) direction for X-Y Routing using an injection rate of 9.6 flits/cycle for a 10 X 10 mesh. Figure 21 represents the utilizations (in the East) direction for West-First Routing using an injection rate of 14.4 flits/cycle for a 10 X 10 mesh. These graphs are similar to the corresponding graphs in Chapter 2. Some discrepancy is only to be expected in view of the assumptions made in the analysis, but the general form of the graphs clearly indicates a correspondence between the analytical results and those obtained by simulation.

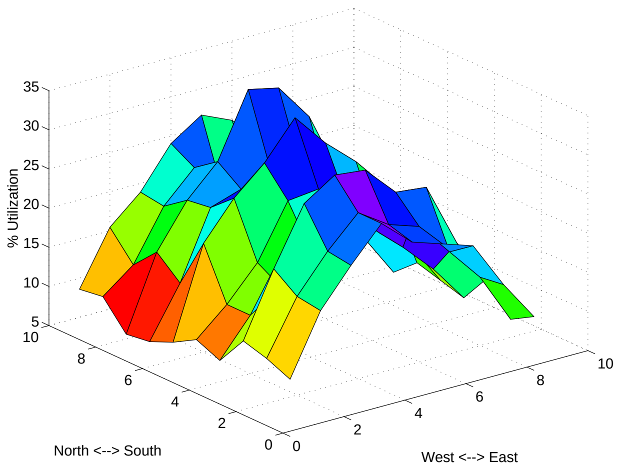

Figure 20: X-Y Routing - Utilizations

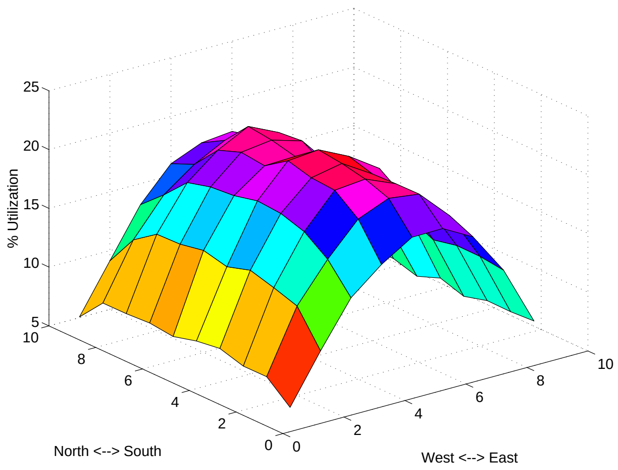

Figure 21: West-First Routing - Utilizations in the East

CHAPTER IV

THE SELECTION POLICY

4.1 Introduction

In this chapter, we propose a scheme to correct the imbalance in utilization that has been discussed in the previous chapter. An adaptive routing algorithm specifies a set of channels, any of which lead ultimately to the destination. The router uses an *output selection policy* to decide the ultimate channel chosen by the header flit. Some output selection policies [3] are, *xy* which chooses the output channel in the lowest dimension and the *no-turn* which tries to avoid turns as far as possible. Our scheme uses the information gathered by the Analysis (Chapter 2) to implement an intelligent output selection policy in the router. The traffic gets evenly balanced out in spite of the restrictions placed by the routing algorithm for deadlock freedom. This increases utilization in the low-utilization regions so that the high-utilization regions don't reach full capacity quickly and the network saturates at a much higher injection rate than it would have, with a naive selection policy.

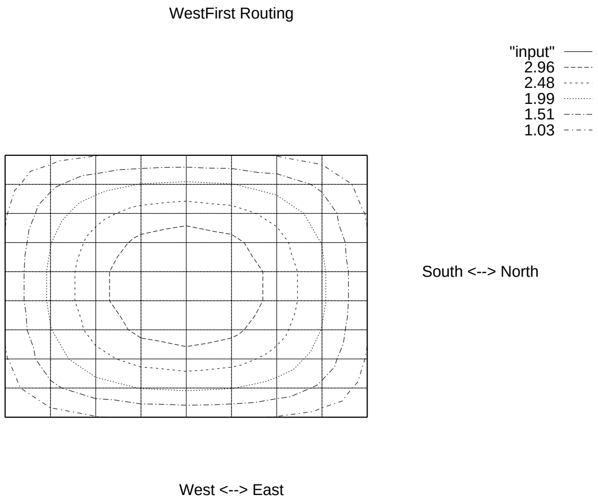

Figure 22: West-First - East

4.2 Contours

An interesting fact is that regions of approximately similar utilizations can be clustered together. This is brought out by the contour representation (see Figures 22 - 25) of the results obtained by analyzing the West-First routing algorithm[1]. We use this fact to propose a solution to the Imbalance problem.

Consider any of the above figures. In each case, the mesh can be divided into different *regions*, where a region is defined as a grouping of links with approximately similar utilizations. So we have r regions $R_1, R_2, \ldots R_r$. The regions are sorted in increasing order of utilization. This grouping is made beforehand using the results of the analysis. So, each router *knows* what region it belongs to, for *each* direction of travel. It uses this knowledge to route messages. The routing algorithm suggests a

[1]We chose West-First to test our technique - however, it is applicable to any routing algorithm.

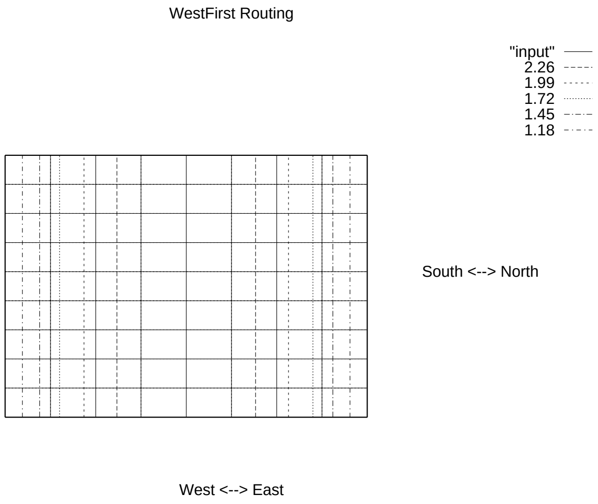

Figure 23: West-First - West

set of possible routes. The router then follows these rules (in order):

1. As far as possible, route through a region with lower utilization.

2. If that's not possible, route within the current region.

3. Route to a region with higher utilization, only when forced by the routing algorithm to do so.

These rules, by themselves, are not sufficient to erase the imbalances since if we favor the low-utilization regions all the time, we run the risk of turning them into high-utilization ones. Hence the router needs to take an intelligent, informed decision and balance traffic between different regions. This is explained clearly in the next section.

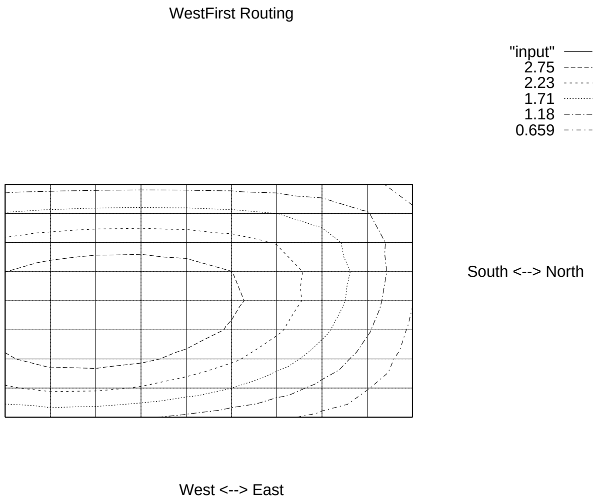

Figure 24: West-First - South

4.3 The Output Selection Policy

The technique is elaborated below for a 2D Mesh with 1 physical channel between each node. However it can be easily extended for other topologies with more channels, also.

1. **Initialization :** Each node has 4 outgoing links, one in each direction. So, each node belongs to 4 regions, one for travel in each of the 4 directions. And, from the contours data, each region has a *minimum* and *maximum* utilization. Using these figures, the amount of *extra* traffic to be directed into the low-utilized regions is calculated. Thus, for each node, for each direction, we have a *max* number of messages to be sent in that direction. And it is this maximum limit set on a channel that actually decides which channel we want to favor (a

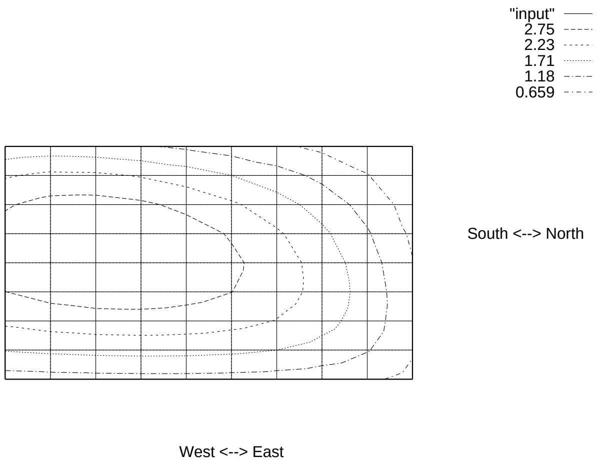

Figure 25: West-First - North

high *max* indicates that this channel is to be preferred). Also for each node, in each direction, a counter is maintained to keep track of the number of messages sent so far that way.All these variables are initialized to their pre-calculated values, *once* at startup time. Note that they are *not* calculated for each routing decision.

2. **Routing and Output Selection :** The routing algorithm decides on the set of channels that are appropriate for that routing decision. If any of these are free, the router chooses to use it. However, if it has to *wait* for a channel, the differences between the counters and the *max* values for each channel are calculated and the channel whose counter is *least*, relative to its *max* is the chosen channel and is the one chosen to be waited upon. All the counters are reset at the same time when all of them have reached their maximum values.

Hence, if C_i has a limit max_i and C_j has a limit max_j then the above scheme will ensure that over a large time interval, for every max_i messages passed through C_i, max_j messages will pass through C_j. Thus, this implements a probability-based output selection policy where the ratio of the number of messages allowed through different channels can be controlled by information provided by the user.

CHAPTER V

IMPLEMENTATION AND RESULTS

5.1 Introduction

We implemented the selection policy described in Chapter 4 using a CSIM-based simulator (described in Appendix A) for two routing algorithms, West-First and Optimal Fully Adaptive Routing. In this chapter, the simulation environment as well as the results obtained from the experiments are described.

5.2 Implementation

We simulated West-First Routing on a 10 X 10 mesh with 1 physical channel between each node. Messages were generated with a Uniform Distribution with injection rates ranging from 1.6 flits/cycle to 10.24 flits/cycle. Optimally Fully Adaptive Routing was simulated on a 10 X 10 mesh with 2 virtual channels between each node on the North/South axis and 1 virtual channel on the East/West axis. Messages were generated with a uniform distribution with injection rates ranging from 1.6 flits/cycle to 15.2 flits/cycle.

Table 3: Simulation Parameters for Saturation Tests - West-First Routing

Parameter	Value(s)
Topology	2D Mesh
Nodes	100
Virtual Channels/Physical channel	1
Network cycle	3.55 ns
Flits/packet	16,32,64
Injection Rate	1.6...10.24 flits/cycle

Table 4: Simulation Parameters for Saturation Tests - Optimal Fully Adaptive Routing

Parameter	Value(s)
Topology	2D Mesh
Nodes	100
Virtual Channels/Physical channel	1 or 2
Network cycle	6.12 ns
Flits/packet	16,32,64
Injection Rate	1.6...15.2 flits/cycle

5.2.1 A Note on Detection of Steady State or Saturation

When the number of messages that have reached their destination exceed the number of nodes(sources), and the completion rate is almost equal to the injection rate within a tolerance of 0.9, then it means that the network has attained *steady-state.* If it doesn't come close, then simulation is continued till the number of messages completed is equal to 5 times the total number of nodes. If, even after this, the completion rate remains constant within a tolerance of 0.001, then it means that latencies are very high and the number of messages that get pumped in is more than that of messages that reach their destinations and we consider the network to have *saturated.*

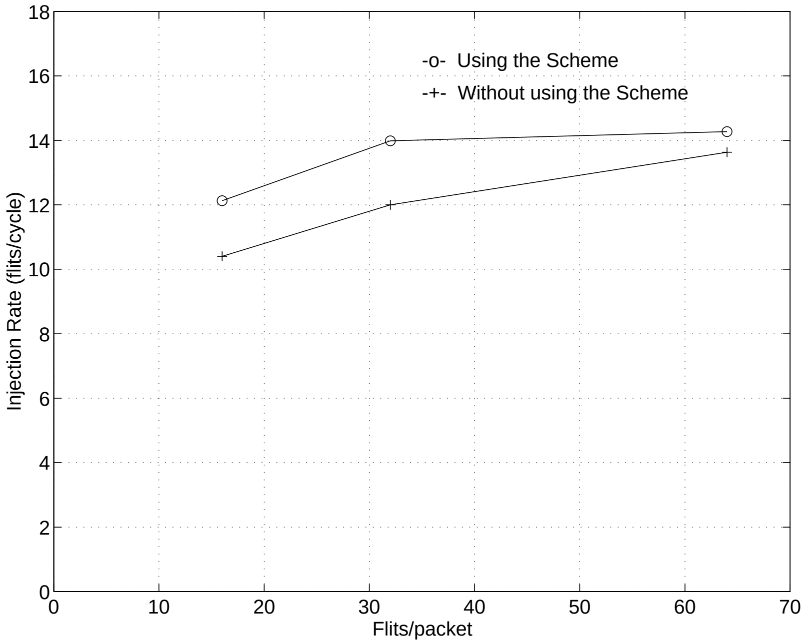

Figure 26: Saturation Behavior - West-First Routing

5.3 Results

5.3.1 Behavior at High Traffic

Our aim was to improve the behavior of the network at high traffic, or in other words, to increase the injection rate at which the network saturated. We simulated the network for various message-lengths and measured the injection rate at which the network saturated. Table 3 shows the various parameters that controlled the simulations for West-First Routing, while Figure 26 shows the results. The enhanced wormhole router performs much better and at times is able to take about traffic at about 16 % higher injection rate without saturating. This is because messages are routed so that the erstwhile low-utilized channels get more traffic and the load is more

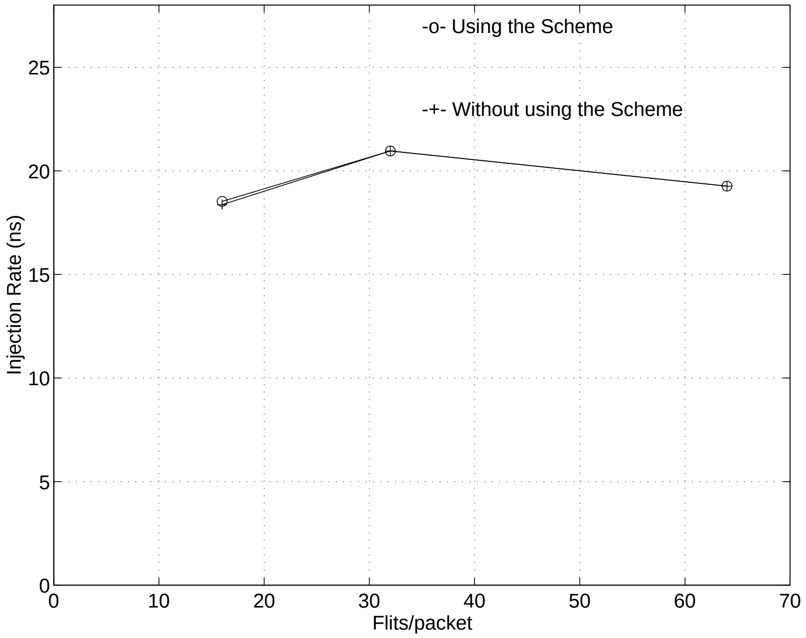

Figure 27: Saturation Behavior - Optimal Fully Adaptive Routing

balanced - hence the network saturates at a higher injection rate. Table 4 shows the simulation parameters for Optimal Fully Adaptive Routing while Figure 27 displays the results.. It should be noted that this algorithm itself incorporates an efficient selection policy. So the improvements here are minimal.

5.3.2 Utilization

We were also able to improve the utilization. In these experiments, the packet-size was kept fixed at 16 flits/packet and the injection rate was varied. The simulation parameters for West-First Routing are in Table 5 and the results (direction-wise breakup as well as the mean) are plotted in Figures 28 - 32. Utilization increases by about 5-7 % at high injection rates. Since we attempt to raise utilization in

Table 5: Simulation Parameters for Latency/Utilization Measurements - West-First Routing

Parameter	Value(s)
Topology	2D Mesh
Nodes	100
Channels	1
Network cycle	3.55 ns
Flits/packet	16
Injection Rate	1.6...10.24 flits/cycle

Table 6: Simulation Parameters for Latency/Utilization Measurements - Optimal Fully Adaptive Routing

Parameter	Value(s)
Topology	2D Mesh
Nodes	100
Channels	1 or 2
Network cycle	6.12 ns
Flits/packet	16
Injection Rate	1.6...15.2 flits/cycle

the low-utilized channels, the mean utilization increases. The Regions Scheme is effective at higher injection rates where contention for channels is considerable. The corresponding parameters for Optimal Fully Adaptive Routing are shown in Table 6. The mean utilization is plotted in Figure 33 and the graph shows an increase of about 4 % at injection rates greater than 14 flits/cycle.

5.3.3 Latency

At low injection rates, using this scheme does not affect the latency appreciably, but at higher injection rates, it starts to decrease. Figure 34 shows the results for West-First

Routing. The improvement in latencies is abour 25 %. Figure 35 represents Optimal Fully Adaptive Routing and the improvement here is about 5 %. The Regions Scheme is effective at higher injection rates where demand for channels is more.

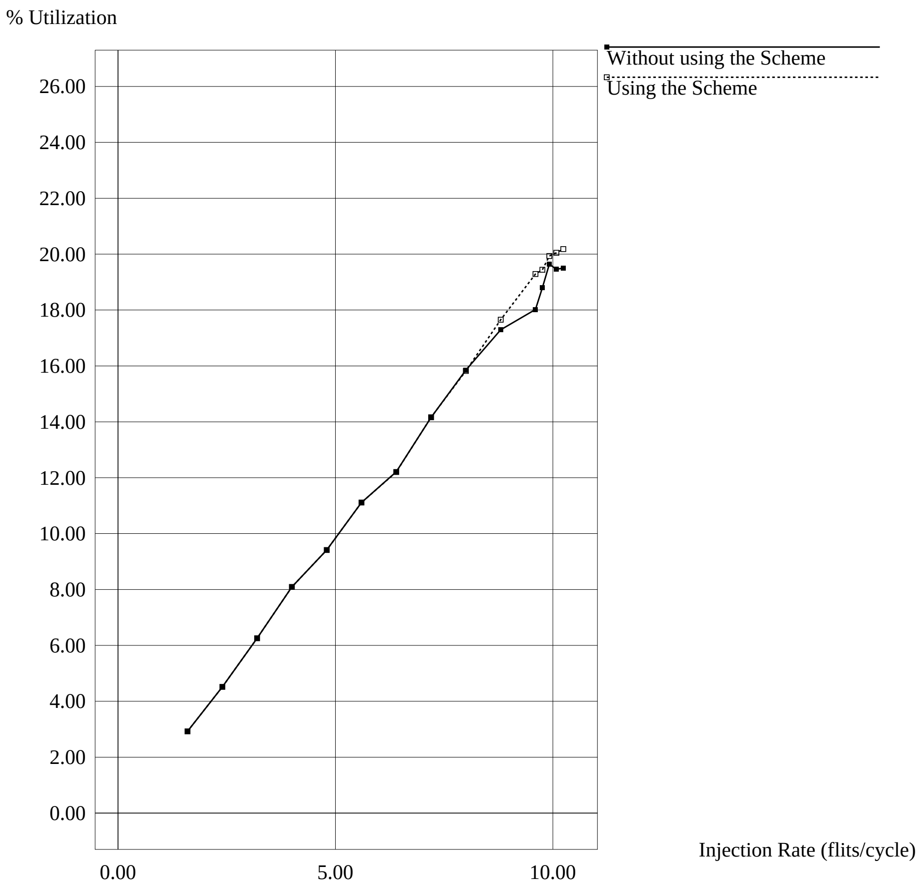

Figure 28: Utilization : West-First Routing - East

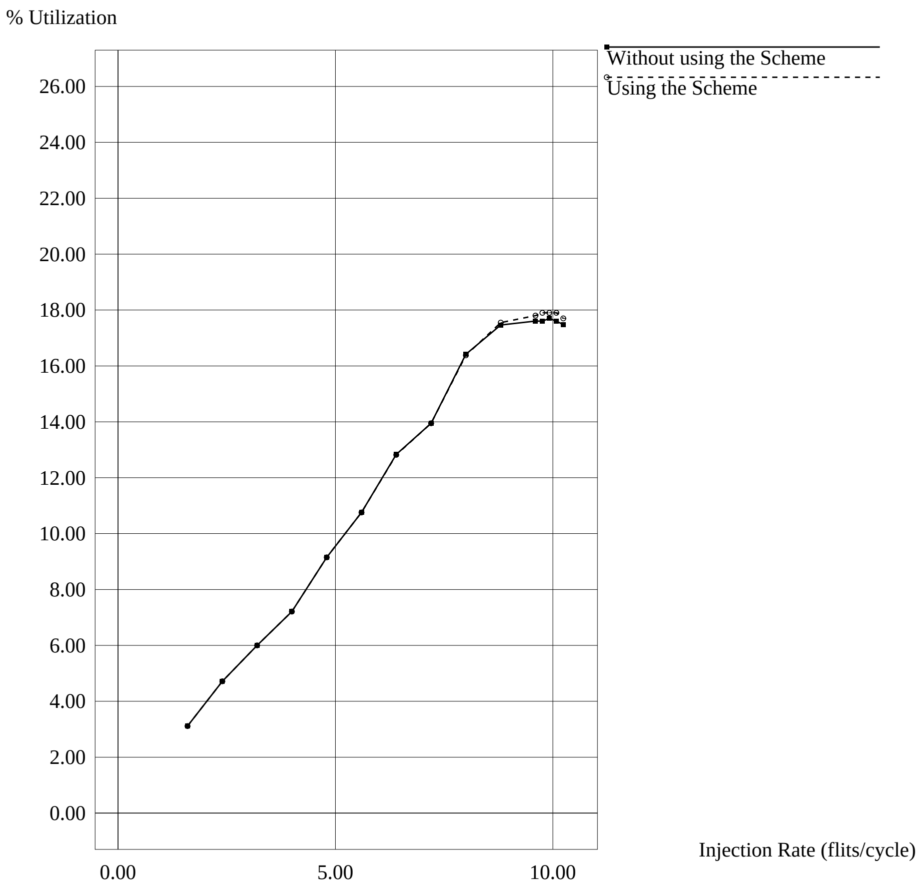

Figure 29: Utilization : West-First Routing - West

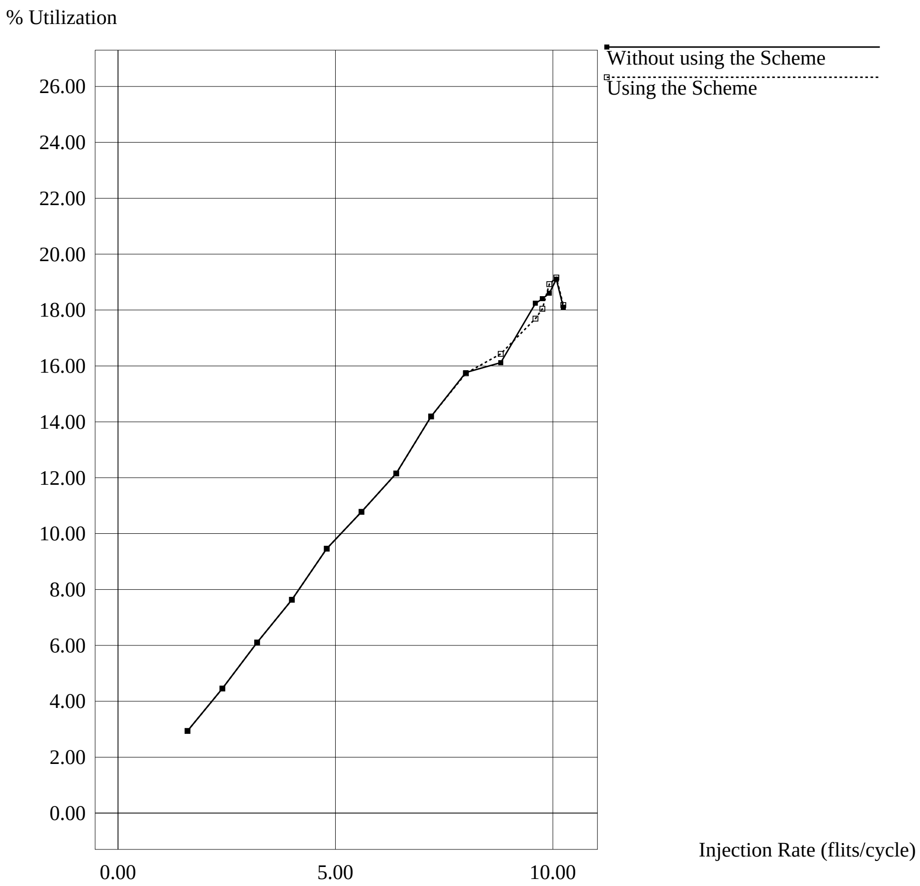

Figure 30: Utilization : West-First Routing - North

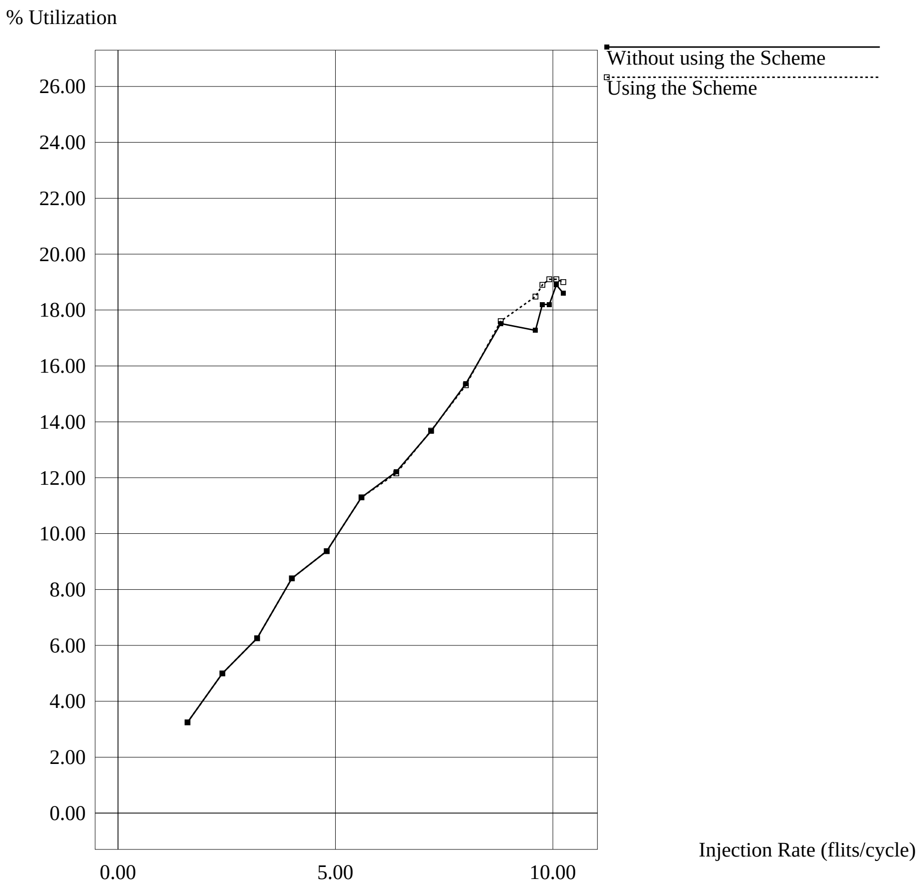

Figure 31: Utilization : West-First Routing - South

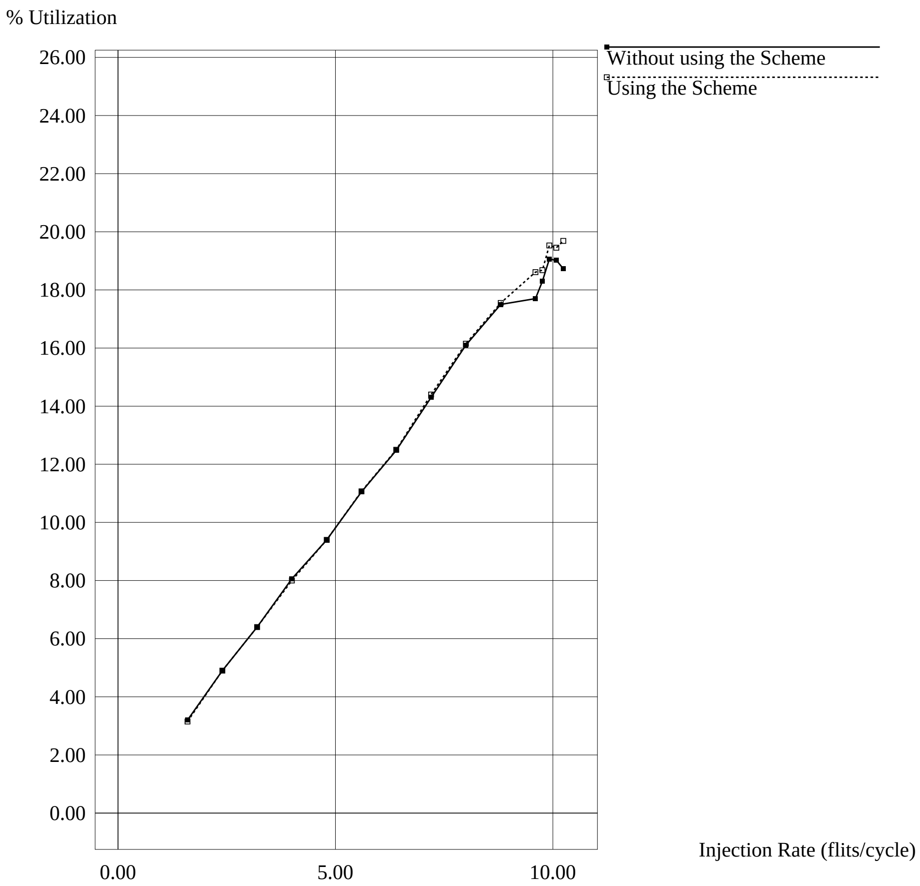

Figure 32: Utilization : West-First Routing - Mean

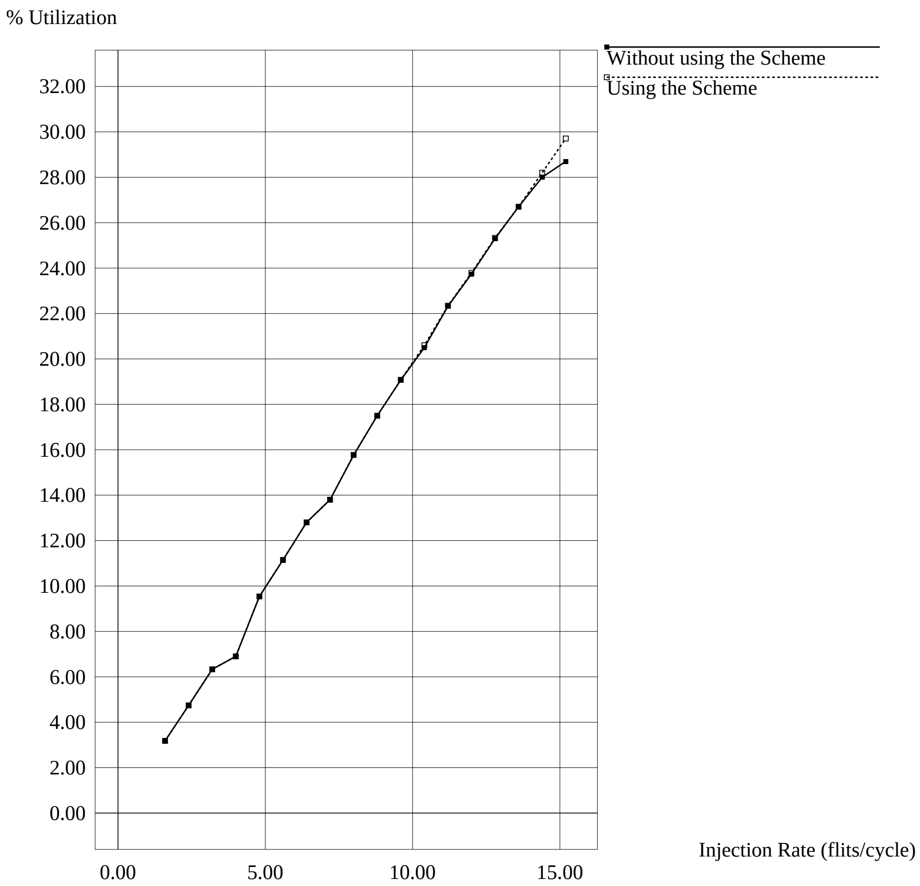

Figure 33: Utilization : Optimal Fully Adaptive Routing - Mean

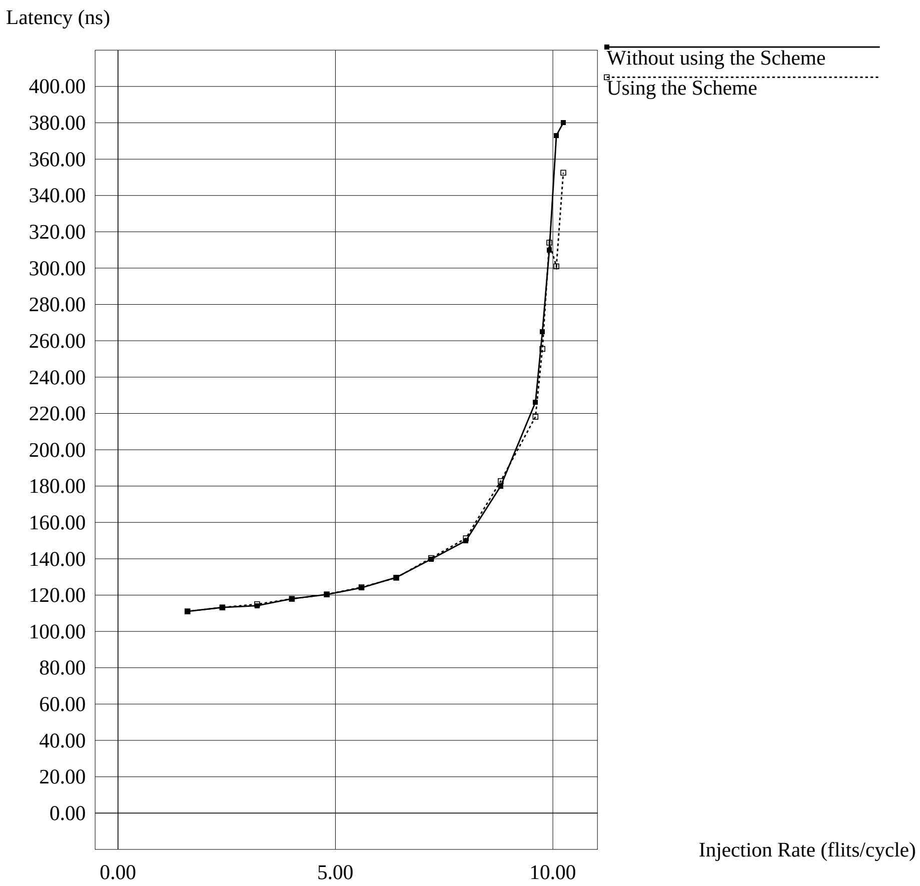

Figure 34: Latency : West-First Routing - Mean

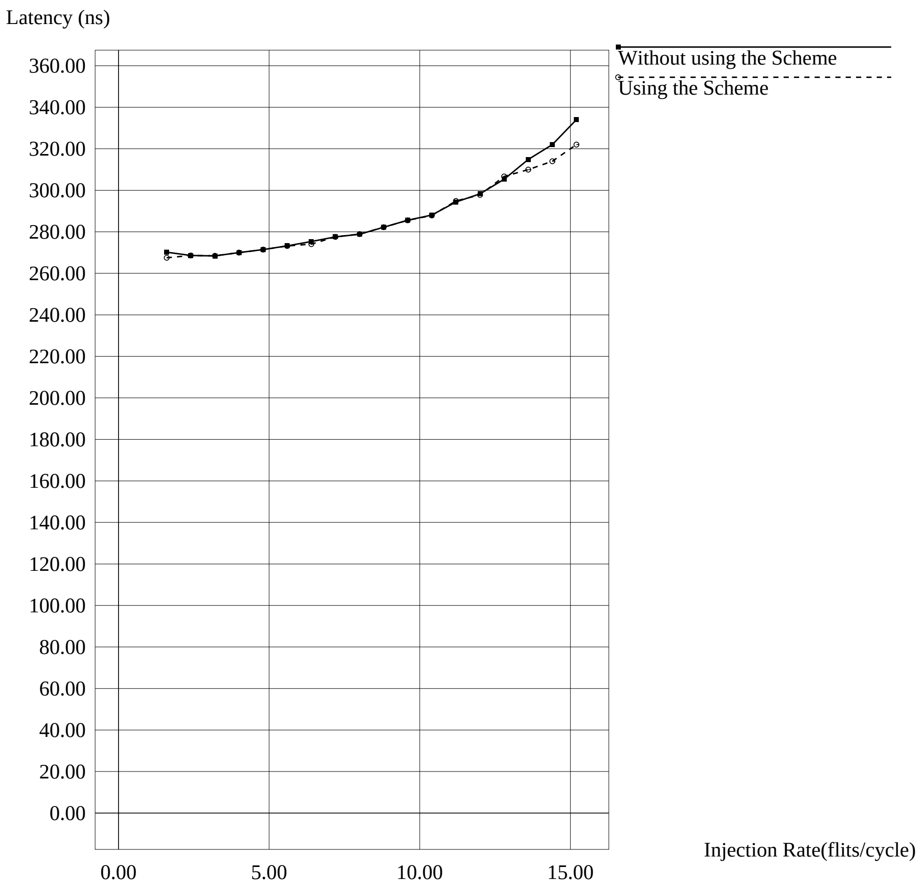

Figure 35: Latency : Optimal Fully Adaptive Routing - Mean

CHAPTER VI

CONCLUSION

6.1 Contributions

Our goals in this thesis were to study the imbalance in channel-utilization over different parts of the mesh, to quantify it and to provide a method to balance the utilizations so that the saturation behavior of the network improved. A simple analytical model was presented to study the utilizations at the link-level. This model was used to analyze routing algorithms that were static, partially-adaptive or fully-adaptive. The results of the analysis provided potential utilizations for each link in each direction. Also, each algorithm was studied direction-wise and this gave a clear view of its constituent routing policies. "Imbalance" was defined as a measurable quantity. Using this measure, the algorithms were compared. We described a technique which uses analytical results to effect an intelligent output selection policy to balance the traffic and thus improve the behavior of the network at high injection rates. The policy was implemented on a simulator for Westfirst and Optimal Fully Adaptive routing algorithms and the results were presented. The results show an improvement in the performance of the routing algorithm at high frequencies.

6.2 Future Work

While the improvement in the saturation behavior is good, the increase in the utilizations or the decrease in the latencies is not very appreciable. More work needs to be done so that latency is further reduced while utilization is increased. This scheme could be implemented with different levels of granularity i.e., the number of *regions* used to classify the links. Having more regions would lead to more precise control over routing but may involve additional computation since the *counts* need to be incremented more frequently. This tradeoff between precision and performance needs to be studied further.

Research needs to be done on the hardware aspects of actual implementation. Each routing decision requires a few additions and comparisons. The router should be designed efficiently so that the computation does not significantly affect the overall clock cycle time.

APPENDIX A

ABOUT THE SIMULATOR

The simulator[1] used in our experiments was implemented using CSIM. It simulates wormhole routed networks (using time multiplexing as well as demand multiplexing[2]) for a variety of topologies (meshes, toruses and hypercubes). Traffic patterns that can be generated include uniform distribution, hotspot traffic and transpose traffic. The user can input variables like flit-size, size and topology of the network, number of physical and virtual channels, network cycle time and header processing time. The network cycle time was calculated using Chien's Instantiated Cost Model [1]. This ensures that the router complexity is taken into account. The header processing time is generally taken as equal to twice the network cycle time. Collection of statistics forms an important part of any simulation. The program gives out measures of latency, utilization and other quantities such as the number of messages generated, the injection and completion rates etc. It has the capability to present statistics for each link[3] also. As explained in Chapter 5, our simulator takes a conservative approach to detect steady state.

[1] The original software was written by Jeff May, Kant Patel and Roshan Rao as a class project.

[2] Demand multiplexing was added by Jeff May.

[3] This enhancement was made by Kant Patel and Roshan Rao.

Bibliography

[1] A. A. Chien. A cost and speed model for k-ary n-cube wormhole routers. *Proceedings of Hot Interconnects*, 1993.

[2] W. J. Dally and C. L. Seitz. Deadlock-free message routing in multiprocessor interconnection networks. *IEEE Trans. Comput.*, C-36(5):547–553, May 1987.

[3] C. Glass and L. Ni. Adaptive routing in mesh-connected networks. *Proceedings of the 12th International Conference on Distributed Computing Systems*, June 1992.

[4] C. Glass and L. Ni. The turn model for adaptive routing. *Proceedings of the 19th Annual International Symposium on Computer Architecture*, pages 278–287, 1992.

[5] Loren Schwiebert and D. N. Jayasimha. Optimal fully adaptive wormhole routing for meshes. *Supercomputing'93*, pages 782–791.