Spring 5-31-1994

Deterministic and adaptive routing algorithms for mesh-connected computers

Yikui Cai

New Jersey Institute of Technology

ABSTRACT

DETERMINISTIC AND ADAPTIVE ROUTING ALGORITHMS FOR MESH-CONNECTED COMPUTERS

by
Yikui Cai

The two-dimensional mesh topology has been widely used in many multicomputer systems, such as the AMETEK Series 2010, Illiac IV, MPP, DAP, MasPar MP-1 and Intel Paragon. Its major advantages are its excellent scalability and simplicity. New generation multicomputer uses a switching technique called wormhole routing. The essential idea of wormhole routing is to advance a packet directly from incoming to outgoing channel without sorting it, as soon as enough information has been received in the packet header to select the outgoing channel. It has advantages of low latency and low error rate. The problems addressed by this thesis are to evaluate existing routing algorithms for the 2D mesh based on the wormhole model and to design a new routing algorithm that performs better from existing algorithms.

In this thesis, the performance of both deterministic and adaptive algorithms, as functions of network size, router buffer size, packet length, is evaluated by computer simulation under different traffic model. Also, a new algorithm, called the west-north-first algorithm, is proposed and tested. It contains both characteristics of deterministic and adaptive algorithm, and hence has a better overall performance under various network traffic models.

The results of this study can be applied to the design of parallel processing network system.

DETERMINISTIC AND ADAPTIVE ROUTING ALGORITHMS FOR MESH-CONNECTED COMPUTERS

by
Yikui Cai

A Thesis
Submitted to the Faculty of
New Jersey Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of
Master of Science in Electrical Engineering

Department of Electrical and Computer Engineering

May 1994

APPROVAL PAGE

DETERMINISTIC AND ADAPTIVE ROUTING ALGORITHMS FOR MESH-CONNECTED COMPUTERS

Yikui Cai

Dr. Michael A. Palis, Thesis Advisor — Date
Associate Professor of Electrical and Computer Engineering
New Jersey Institute of Technology

Dr. David Nassimi, Committee Member — Date
Associate Professor of Computer and Information Science
New Jersey Institute of Technology

Dr. Sotirios Ziavras, Committee Member — Date
Assistant Professor of Electrical and Computer Engineering
New Jersey Institute of Technology

BIOGRAPHICAL SKETCH

Author: Yikui Cai

Degree: Master of Science in Electrical Engineering

Date: May 1994

Undergraduate and Graduate Education:

- Master of Science in Electrical Engineering,
  New Jersey Institute of Technology, Newark, NJ, USA, 1994

- Bachelor of Science in Electrical Engineering,
  Shanghai University of Technology, Shanghai, P.R. China, 1985

Major: Electrical Engineering

This thesis is dedicated to
My Parents and Sister
for their loves

ACKNOWLEDGEMENT

The author would like to express her sincere gratitude to her advisor, Professor Michael A. Palis, for his guidance, support, kindness and encouragement throughout the process of producing this thesis.

My thanks are due to Professors David Nassimi and Sotirios Ziavras for serving as members of the thesis committee.

I am thankful to Xi Li for his constant support and encouragement.

TABLE OF CONTENTS

TABLE OF CONTENTS
(Continued)

LIST OF TABLES

LIST OF FIGURES

Figure **Page**

LIST OF FIGURES
(Continued)

CHAPTER 1

INTRODUCTION

1.1 Motivation

Massively parallel computers with thousands of processors are considered the most promising technology to achieve teraflops computational power. Such large-scale multiprocessors are usually organized as ensembles of nodes, where each node has its own processor, local memory, and other supporting devices. These nodes are connected via some form of interconnection network and communicate with each other by passing messages over the network.

There are two general classes of interconnection networks: direct networks and indirect networks. Direct network has become a popular architecture because it scales well, that is, as the number of nodes in a system increases, the total communication bandwidth, memory bandwidth and processing capability of the system also increase.

There are three major issues in designing a direct network system, system topology, flow control and routing.

The topology of a network, represented as a graph, defines how nodes are interconnected. N-dimensional mesh is one of the examples.

A network consists of many channels and buffers. Flow control deals with the allocation of channels and buffers to a message as it travels along a path through the network. A good flow control policy should avoid channel congestion while reducing the network latency. Wormhole routing has been a popular flow control technique in new-generation direct networks. The pipeline nature of wormhole routing has two advantages. First, the absence of network contention makes the network latency relatively insensitive to path length. Second, large packet buffer at each intermediate node is obviated; only a small FIFO (first in, first out) flit buffer is required.

A direct network topology must allow every node to send packets to every other node. Without complete connectivity in a network, routing determines the path used by a packet to reach its destination. Efficient routing is critical to the performance of direct networks. Routing can be classified as *deterministic* or *adaptive*. With deterministic routing, the path a message follows depends only on its source and destination nodes. This method is also referred to as oblivious routing. Because the source and the destination of a packet are fixed after it's "born", its path is determined. It can't change its route to avoid path blocking. In adaptive routing, for a given source and destination, the path taken by a particular packet depends on the dynamic network conditions, such as the presence of faulty or congested channel. Due to its adaptive nature, misrouting occurs in network congestion situation. One major problem with adaptive routing is potential deadlock in which a set of packets may become blocked forever in the network.

Because of these short-come there is a constant interest in new routing algorithms which aim at improving the overall performance of wormhole routing technique.

1.2 Statement of Purpose

The purposes of this thesis are:

1. to evaluate existing deterministic and adaptive routing algorithms;
2. to design a new routing algorithm that is deadlock free and better than the existing ones with respect to different network traffic conditions;
3. to identify the best routing algorithm with respect to following performance metrics:
   - hardware cost (buffer size, network size, packet size and hardware complexity of router);
   - *network throughput.*

1.3 Contributions

The specific contributions of this thesis are as follows:

1. From our experimental data, it showed that the deterministic routing algorithm is superior to adaptive routing algorithms in uniformly distributed traffic condition and inferior to adaptive routing algorithms under non-uniformly distributed traffic condition.
2. The new developed routing algorithm gives an alternative way to adaptive routing. Its performance has been compared with the existing deterministic and adaptive algorithms and the result showed that it is between the deterministic one and the old adaptive one in both uniform and non-uniform traffic conditions.

CHAPTER 2

INTERCONNECTION NETWORKS

The processing nodes of a massively parallel computer exchange data and synchronize with one another by passing messages over an interconnection network(see Figure 2.1).

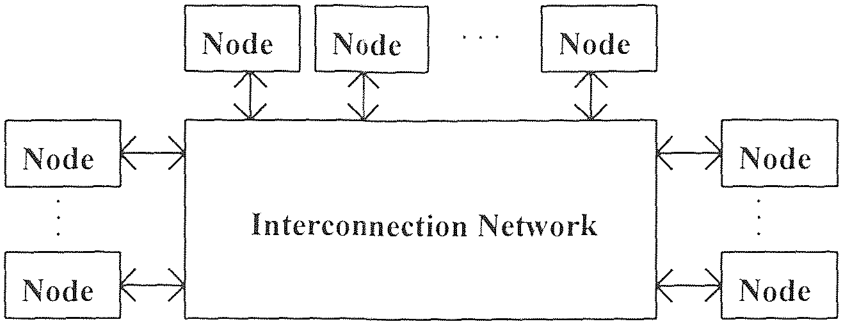


Figure 2.1 A generic multiprocessor based on an interconnection network.

There are two general classes of interconnection networks: (1) *direct or single-stage networks*, and (2) *indirect or multistage networks*.

- In a direct network, every communication link in the network connects a pair of processing nodes; i.e., there are no intermediate switching nodes. All switching is done in the processing nodes. Figure 2.2 shows examples of direct networks.
- In an indirect network, each pair of processing nodes is connected by a path consisting of one or more switching nodes. Figure 2.3 shows examples of indirect networks.

In this thesis, we will only be concerned with direct networks. In a direct network, each node contains a separate router to handle communication-related tasks. Figure 2.4 shows the architecture of a generic node. The router supports some number of input and output channels. Normally, every input channel is paired with a corresponding output channel. External channels are used for communication between routers while an internal channel connects the local processor to the router. In this thesis, the term channel will refer to an external channel.

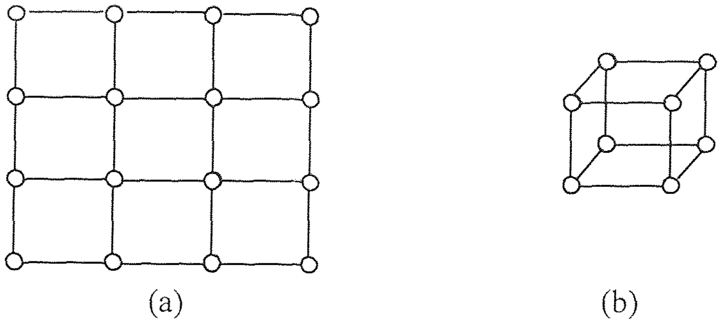


Figure 2.2 Direct network topologies: (a) 4 x 4 2D mesh; (b) 3D hypercube.

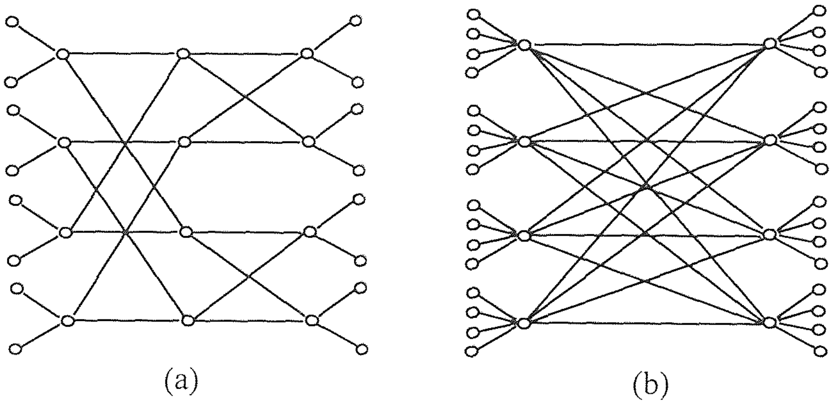


Figure 2.3 Indirect network topologies (a) a 2-ary 3-fly; (b) a 4-ary 2-fly.

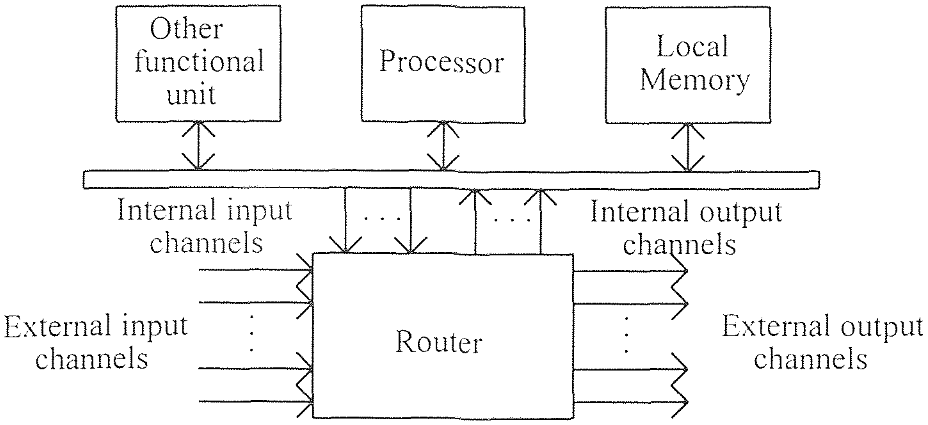


Figure 2.4 A generic node architecture

A direct network is characterized by three factors: *topology*, *routing*, and *flow control*.

2.1 Topology

The topology of a network, modeled as a graph, defines how the nodes are interconnected by channels. If every node is connected directly to every other node, the network topology is fully connected, or complete. Although complete topologies obviate forwarding of messages by intermediate nodes, they are practical only for very small networks because the number of physical connections per node is limited by rigid constraints.

Therefore, many direct networks use a fixed, multiple-hop topology, such as a hypercube or two-dimensional mesh (see the examples in Figure 2.2). In multiple-hop topologies, a message may traverse one or more intermediate nodes before reaching its destination.

Several parameters are used to evaluate a topology, such as *bisection width*, *channel width*, *channel bandwidth* ***W***, *channel rate* and *bisection density* ***B***. The *bisection width* ***Bw*** of a topology is the minimum number of channels that must be removed, or cut, to partition the network into two subnetworks, each containing half the nodes in the network. *Channel width* is the number of bits transmitted simultaneously on a physical channel between two adjacent nodes, and *channel rate* is the peak rate at which bits can be transferred over each individual line of a physical channel. The *bisection density* is the product of bisection width and the channel width ($\boldsymbol{B}=\boldsymbol{B}w\times\boldsymbol{W}$), it is commonly used as measure of network cost.

For a given number of network nodes, low-dimensional mesh networks have much lower bisection width than others; consequently, they can offer wider channels and a higher channel bandwidth for a given bisection density. The disadvantage of a low dimension network is the relatively large distance between nodes. However, in systems that support "wormhole routing" (discussed later), the network latency is almost independent of the path length when there is no contention and when the message size is relatively large[2]. Low-dimensional meshes are popular topologies for such systems because the negative effects of their large internode distance are minimized.

2.2 Routing

A direct network topology must allow every node to send messages to every other node. In the absence of a complete topology, routing determines the path selected by a message to reach its destination. Efficient routing is critical to the performance of direct networks.

Routing can be classified in several ways. In source routing, the source node selects the entire path before sending the message. Each message must carry this routing information, thus increasing the message size. Furthermore, the path cannot be changed after the message has left the source. Most direct network systems use distributed routing. In this approach, each router, upon receiving the message decides whether it should be delivered to the local processor or forwarded to a neighboring router. In the latter case the routing algorithm is invoked to determine which neighbor the message should be sent to. In a practical router design, the routing decision process must be as fast as possible to reduce the network latency.

Routing can also be classified as *deterministic or adaptive*. With deterministic routing, the path a message follows depends only on its source and destination nodes. This method is also referred to as oblivious routing. A routing technique is adaptive if, for a given source and destination, the path taken by a particular message depends on dynamic network conditions, such as the presence of faulty or congested channels.

A routing algorithm is said to be minimal if the path selected is one of the shortest paths between the source and destination pair. A non-minimal routing algorithm allows message to follow a longer path, usually in response to current network conditions.

2.3 Flow Control

A network consists of many channels and buffers. Flow control deals with the allocation of channels and buffers to a message as it travels along a path through the network. A resource conflict occurs when a message cannot proceed because some resource that it requires is held by another message. Whether the message is dropped, blocked in place, buffered, or rerouted through another channel depends on the flow control policy. A good flow control policy should avoid channel congestion while reducing the network latency.

Figure 2.5 shows the three information units important to understanding flow control [3].

- **Message** The logical unit of communication. Two objects communicate by sending a message. This is the only unit seen by clients of the network service.
- **Packet** A message is divided into one or more packets. A packet is the smallest unit that contains routing information, e.g., the destination address. Long messages must be broken into many packets to avoid degrading network performance.
- **Flit** A packet can be further divided into flow control digits of *flits,* the smallest unit on which flow control is performed, that is, communication resources, wires

and buffers, are allocated on a flit-by-flit basis. In a packet, only the header flit contains routing information. The remaining flits must follow the header flit to determine their route.

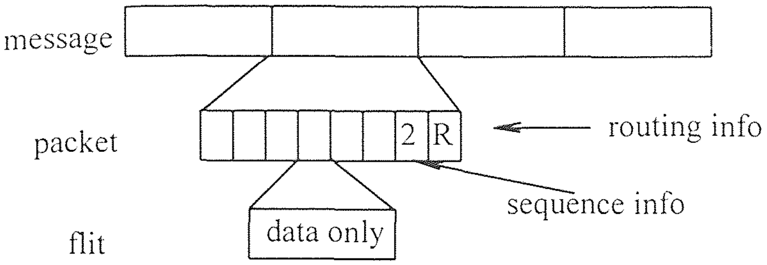


Figure 2.5 Packet and flit units

Network latency is highly dependent on the flow control method used. There are three popular flow control methods used in direct networks: store-and-forward, wormhole routing, and virtual cut-through.

In store-and-forward routing, when a packet reaches an intermediate node, the entire packet is stored in a packet buffer. The packet is then forwarded to a selected neighboring node when the next channel is available and the neighboring node has an available packet buffer. In multicomputer networks, however, the latency of store-and-forward routing is unacceptable. Newer multicomputers use *wormhole routing* [3], where channels and buffers are allocated to flits which are significantly smaller than a packet.

In wormhole routing, a packet is divided into a number of flits, (flow control digits) for transmission. Figure 2.5 shows a packet and its flit units. The header flit of a packet carries the routing information and governs the route. As the header advances along the specified route, the remaining flits follow in a pipeline fashion, as shown in Figure 2.6. If the header flit encounters a channel already in use, it is blocked until the channel becomes available. The trailing flits are blocked and remain in their flit buffers along the established route. Once a channel has been acquired by a packet, it is reserved for the packet. The channel is released when the last, or tail, flit has been transmitted on the channel.

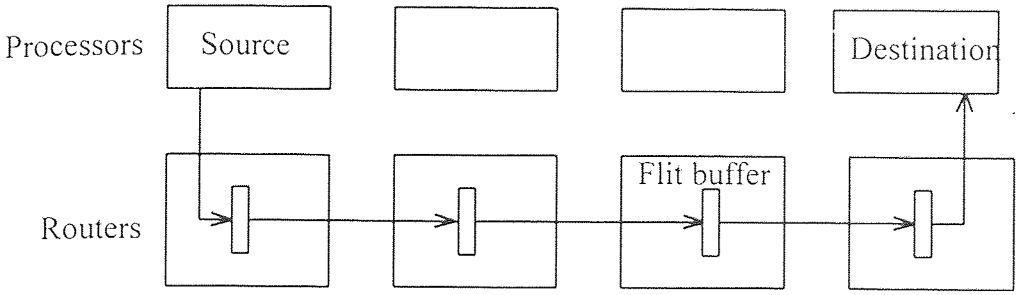


Figure 2.6 Wormhole routing

Performing flow control on a flit-by-flit basis obviates the need for large packet buffers; only a small FIFO flit-buffer is required. Moreover, it reduces latency, as shown in Figure 2.7. With store-and-forward routing, a packet is received in its entirety before being transmitted to the next channel. Dally [3] gives a simple estimate for the routing time of a packet in a store-and-forward network. Let L be the size of a packet in bits, W be the channel bandwidth in bits/cycle and T_C be the cycle time. If the packet must cross D channels, then the zero-load latency (i.e., no network contention) of store-and-forward routing is the product of the time to transmit the packet across a single channel, L/W, and the number of transmissions, D, required to reach the destination.

$$T_{sf} = T_c(\frac{L}{W} \times D) \qquad (2\text{-}1)$$

With wormhole routing, packets are divided into flits. Channels and buffers are allocated flit-by-flit. A flit can advance as soon as it is allocated the resources it requires. It need not wait for the entire packet to be received. With this pipelined routing, the latency becomes the sum of amount of time required to transmit a packet across a single channel, L/W, and the amount of time required for each flit to reach the destination.

$$T_{wh} = T_c(\frac{L}{W} + D) \qquad (2\text{-}2)$$

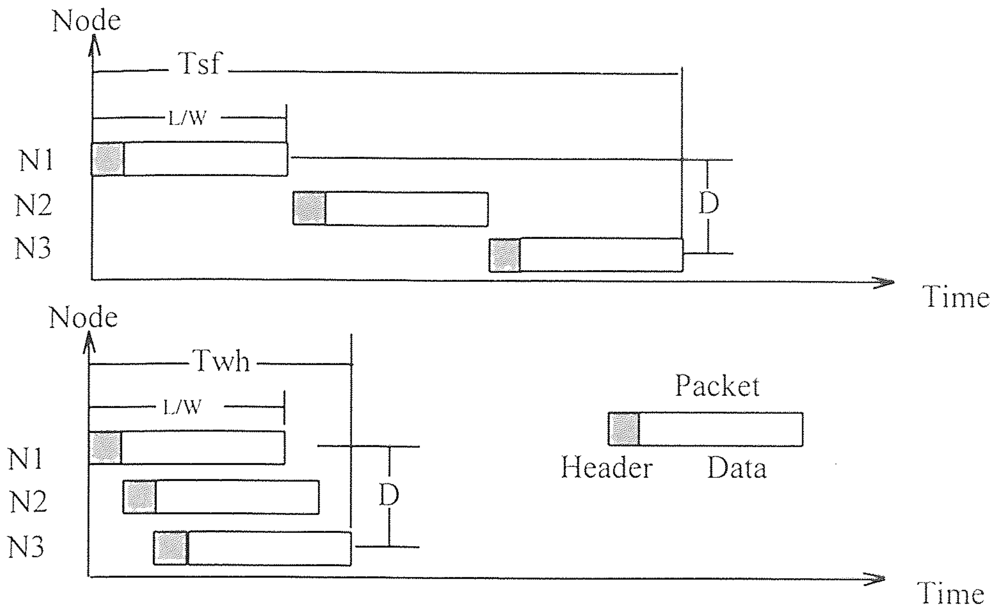


Figure 2.7 Comparison the communication latency of store-and-forward routing (top) and wormhole routing (bottom).

A hybrid strategy, *virtual cut-through* [14], allocates storage buffers to packets as in store-and-forward, but pipelines the transmission of flits as in wormhole. The header flit is examined upon arrival at an intermediate node. If the next required channel is busy, the trailing flits are allowed to advance into the node and are stored in a packet buffer. On the other hand, if the next channel is free, the flits are forwarded immediately without buffering.

Virtual cut-through has the latency properties of wormhole routing, T_{wh}, but requires blocked packets to be buffered [3]. Consequently, the buffer size should be as large as the packet size, as in store-and-forward routing.

CHAPTER 3

WORMHOLE ROUTING ALGORITHMS FOR THE TWO-DIMENSIONAL MESH

The focus of this thesis is the two-dimensional mesh (or 2D mesh, for short). The 2D mesh is a direct network arranged as an n x n grid of processing model as in Figure 2.2 (a).

In [15], Seitz explains why low-dimensional networks, such as a 2D mesh is better than others for wormhole routing. As mentioned in chapter 2, the network latency of wormhole routing is given by equation 2-2, repeated here:

$$T_{wh} = T_c(\frac{L}{W} + D),$$

where L is the size of a packet in bits, W is the channel bandwidth in bits/cycle and T_C is the cycle time, and D is the number of transmissions required for the packet to reach its destination.

Let us compare a 256-node binary 8-cube and a 16 x 16 2D mesh. Table 3.1 show the their parameters with the same bisection density.

Table 3.1 The Parameters of Binary 8-cube and 2D Mesh

Parameter	Binary 8-cube	2D mesh
Number of nodes ***N***	256	256
Bisection density ***B***	256	256
Bisection width ***Bw***	256	16
Channel width ***W=B/Bw*** (bits/cycle)	1	8
Average distance ***D***	4	11.6

Now we can tabulate the number of network cycles, $L/W+D$, required to route packets of length L. With Tc=0.035μs (Ametek Series 2010), the time is shown in Table 3.2.

Table 3.2 Routing Time as function of Packets Length

L (bits)	binary 8-cube	2D mesh
0	4(0.1μs)	11.6 (0.4μs)
8	12(0.4μs)	12.6(0.4μs)
16	20(0.7μs)	13.6(0.5μs)
32	36(1.3μs)	15.6(0.5μs)
64	68(2.4μs)	19.6(0.7μs)
128	132(4.6μs)	27.6(1.0μs)
256	260(9.1μs)	43.6(1.5μs)
...	...	...
2048	2052(72μs)	267.6(9.4μs)

What we observe, is that the network latency dependent on message distance is so much smaller than the network latency dependent on message length provides an important clue that we can afford to make the average *D* larger with little impact on message latency. Also because lower dimension networks are more wireable, we can afford to make *W* larger which in turn makes T_CxL/W smaller. With a low dimension network, we can also keep all of the wires short to reduce $T_{C.}$

Dally already showed his analysis [3] of latency versus network dimension for the class of k-ary n-cube that the optimal number of dimension for machines in the range N=256 is two, that is, a 2-D mesh. Based on this analysis, we believe that 2-D mesh is better than other network topologies for wormhole routing, and therefore we chose it as our network topology.

Wormhole routing has been a popular flow control technique in new-generation direct networks. As discussion in chapter 2, the pipeline nature of wormhole routing has two advantages. First, the absence of network contention makes the network latency relatively insensitive to path length. Second, large packet buffer at each intermediate node is obviated; only a small FIFO (first in, first out) flit buffer is required. Wormhole routing is our choice for network flow control.

The goal of this thesis is to evaluate existing routing algorithms for the 2D mesh based on wormhole model. Routing is the method used for a message to choose a path over the network channels. There are two mainly routing methods: deterministic and adaptive. With deterministic routing, the path a particular packet follows depends only on its source and destination node. When the path taken by a particular packet depends on the state of the network it is in, this routing method is called adaptive routing.

In this thesis, we investigate two routing algorithms, one deterministic and the other adaptive. These routing algorithms are described in the following subsection.

3.1 X-Y Deterministic Routing for the 2D Mesh

In a 2D mesh, each node is represented by its (x,y) coordinates, where x is the row number and y is the column number. A packet is sent from its source to its destination via a unique path which is obtained as follows: first send the packet along the X (or row) dimension until it reaches the same column as the destination. Then send the packet along the Y (or column) dimension until it reaches the destination. This routing algorithm is sometime referred to as the x-y deterministic routing algorithm.

If a packet's outgoing channel is occupied by another when its head flit arrives, it blocks until the channel is free. In the situation when two or more packets arrive at a node and contend for the same output channel at the same time, the packet with the farthest destination is given higher priority. The other packets are blocked until the output channel becomes available.

Figure 3.1 is an example of X-Y deterministic routing in a 4 x 4 2D mesh.

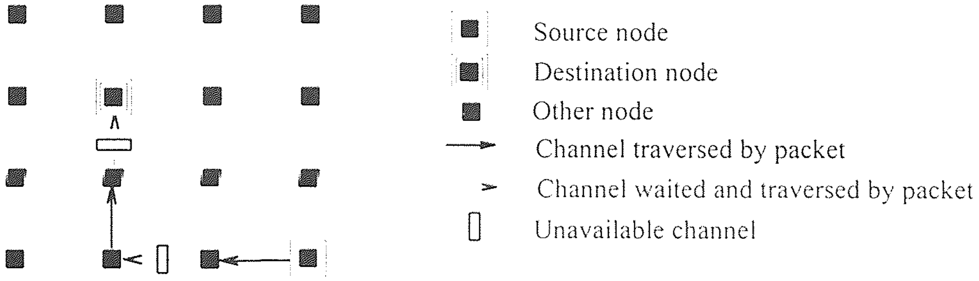

Source node
Destination node
Other node
Channel traversed by packet
Channel waited and traversed by packet
Unavailable channel

Figure 3.1 An example of X-Y deterministic routing in a 4 x 4 2D mesh.

The main disadvantage of deterministic routing is that it cannot respond to dynamic network conditions, such as congestion. The packet's source-to-destination path is completely determined prior to routing the packet and cannot be changed once it is launched in the network. Consider the example shown in Figure 3.2. Nodes (0,0), (0,1) and (0.2) have packets to send to nodes (3,3), (2,3) and (1,3) respectively. Using the x-y deterministic routing algorithm, all packets would have to go through node (0,3) enroute to their destinations as indicated by solid lines. A congestion situation occurs in node (0,3) if the packet from (0,2) has not been sent completely before the packet from (0,1) arrives. This, in turn, delays the transmission of the packet from (0,0).

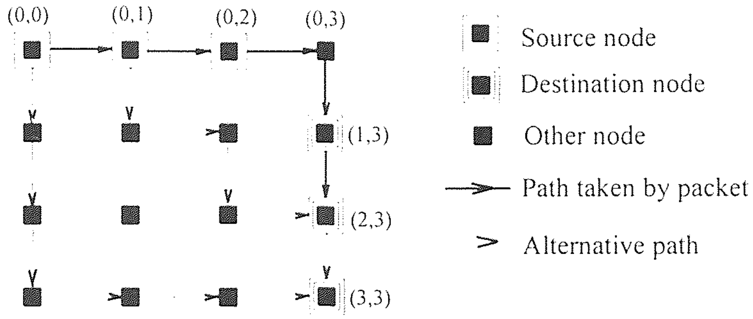


Figure 3.2 An example of congestion in a 4 x 4 2D mesh.

3.2 Adaptive Routing Algorithm for the 2D Mesh

Adaptive routing allows the path taken by a packet to be determined by the current state of the network instead of just the source and destination addresses, (as in X-Y deterministic routing). For example, if two packets contend for the same output channel, one packet is given higher priority, (e.g., the packet with the furthest destination), while the other packet is sent to another free output channel (i.e., the packet is "misrouted"). By allowing packets to change paths on-the-fly, congestion could be avoided. For example, in Figure 3.2, packets from nodes (0,0) and (0,1) can choose the dash lines other than the solid lines to approach their destinations.

3.2.1 Deadlock

One major problem with adaptive routing is the potential for deadlock. In wormhole routing, deadlock can occur if blocked packets hold channels (and their corresponding flit buffers) which are requested by other packets. Figure 3.3 shows an example of channel deadlock involving four routers and four packets. Each packet is holding a flit buffer which is requested by another packet.

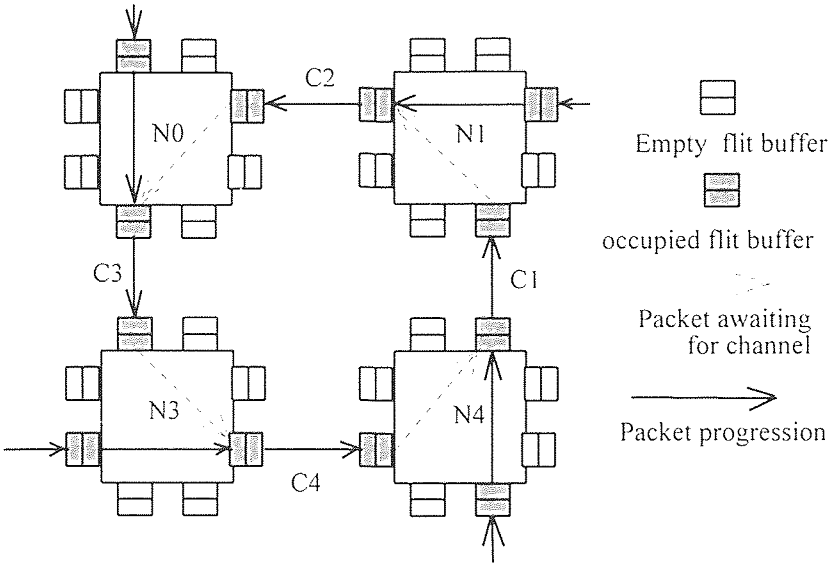


Figure 3.3 An example of channel deadlock involving four packets

By ordering network resources (i.e., channels and flit buffers) and requiring that packets request and use these resources in strictly monotonic order, circular wait - a necessary condition for deadlock - is avoided. Hence, deadlock involving these resources can not arise.

3.2.2 Channel Dependence Graph

To develop a deadlock-free routing algorithm, a *channel dependence graph* [6] can be used. For a given interconnection network $I = G(N,C)$ (N represent the set of processing nodes and C represent the channels of I) and routing function $R:C \times N \rightarrow C$ which maps the current channel c_c and destination node n_d to the next channel c_n on the route from c_c

to n_d, $\boldsymbol{R}(c_c\ , n_d)$=$c_n$, a channel dependence graph $\boldsymbol{D}$ = $\boldsymbol{G}(\boldsymbol{C},\boldsymbol{E})$. The edges of $\boldsymbol{D}$ are the pairs of channels connected by $\boldsymbol{R}$:

E=$\{(c_i\ , c_j)|\boldsymbol{R}(c_i\ , n)$=$c_j$ for some $n \in N\}$.

In [6], Dally and Seitz proved that a routing algorithm is deadlock-free if and only if there are no cycles in the channel dependence graph.

Figure 3.4(a) is channel dependency graph of Figure 3.3. From this graph, because all "turns" are allowed, a cycle is formed and a deadlock can occur. One way to avoid deadlock is to disallow packet to be forwarded from channel C1 to channel C2, that is, to prohibit turning from south to west. The resulting channel dependence graph is shown in Figure 3.4 (b), which is acyclic. Therefore, to send a packet from node 4 to node 0, the packet must be forwarded through node 3.

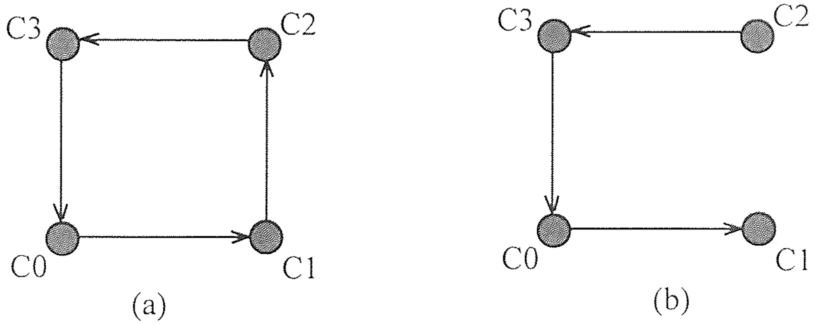


Figure 3.4 (a) channel dependency graph
(b) channel dependence graph based on restricted minimal routing

3.3 The Turn Model and the West-First Adaptive Routing Algorithm

Given a network topology and the associated set of channels, adaptive routing algorithms are usually developed in an ad hoc way. The *turn model* by Glass and Ni [12] provides a systematic approach to the development of adaptive routing algorithms for a given network without adding channels. As Figure 3.3 shows, deadlock occurs because the packet routes contain turns that form a cycle. The following steps can be used to develop adaptive routing algorithms for n-dimensional meshes.

- Classify channels according to the direction in which they route packets.

- Identify the turns that occur between one direction and another, omitting 0-degree and 180-degree turns.
- Identify the simple cycles these turns can form.
- Prohibit one turn in each cycle.
- Add 180-degree and 0-degree turns, which are needed for non-minimal routing algorithms or if there are multiple channels in the same direction.

The case of a 2D mesh illustrates the use of the turn model. There are eight possible turns and two possible abstract cycles, as shown in Figure 3.5 (a). Cycles among packets may result if the turns are not restricted, as illustrated in Figure 3.3. The deterministic XY routing algorithm prevents deadlock by prohibiting four of the turns, as shown in Figure 3.5 (b). The remaining four turns can not form a cycle. That is, its channel dependency graph is acyclic, and hence a deadlock situation will not occur.

The fundamental concept behind the turn model is to prohibit the smallest number of turns such that cycles are prevented. Although the X-Y routing is deadlock free, it does not allow any adaptiveness. Figure 3.5(c) shows how the channel dependency graph can be made acyclic by prohibiting only two turns. These two turns are turns that cause the packet to turn west from either north or south. Therefore, to travel west, a packet must begin in that direction for as long as necessary. Thereafter, it can travel adaptively in any direction as long as it does not travel west again. This routing algorithm was introduced by Ni and McKinley[1] and is referred as the *west-first routing algorithm*.

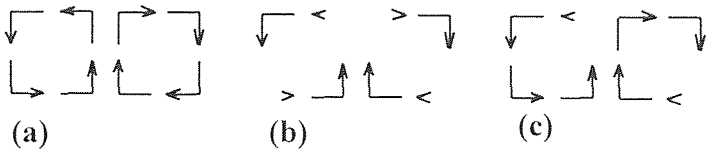


Figure 3.5 An illustration of the turn model in a 2D mesh: (a) abstract cycles in a 2D mesh; (b) four turns (solid arrows) allowed in X-Y routing; (c) six turns (solid arrows) allowed in west-first routing.

Figure 3.6 shows an example of the west-first routing algorithm. The destination node of the packet is the west of its source. Consequently, the packet has to travel west first as long as necessary, i.e., until it reaches the correct column. If, while moving west, the packet encounters a busy channel, it blocks until the channel is free. Once the packet reaches the right column, it can move adaptively. For example, in the Figure 3.6, the packet, upon reaching the correct column, wishes to turn north but encounters a busy channel. Therefore, it chooses one of the free channels, for example, the west channel. Upon arriving at the next node, it turns north to approach its destination. However, it encounters a busy channel while attempting to turn east when it reaches the row of its destination. Hence, it adaptively chooses one of the free channels - the north channel, for example. From thereon, it is able to reach its destination, without further blocking, by following the shortest path.

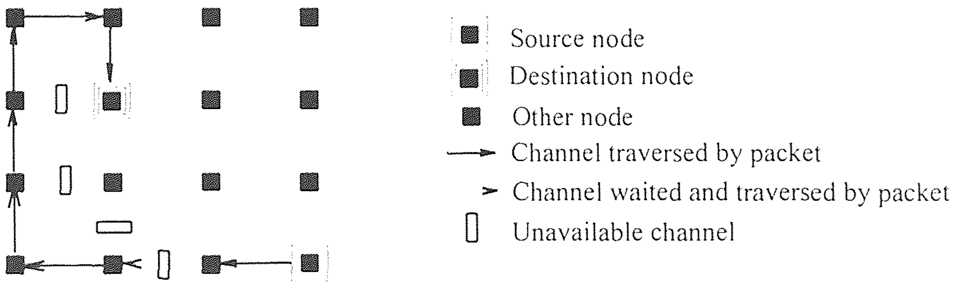

Source node
Destination node
Other node
Channel traversed by packet
Channel waited and traversed by packet
Unavailable channel

Figure 3.6 An examples of west-first adaptive routing in a 4 x 4 2D mesh.

Because cycles are avoided, west-first routing is deadlock-free. Note, however, that the algorithm is deterministic while moving west; then becomes adaptive once its destination is no longer to its west.

CHAPTER 4

PERFORMANCE COMPARISON BETWEEN X-Y DETERMINISTIC AND WEST-FIRST ADAPTIVE ROUTING ALGORITHMS

This section describes the simulator we developed to investigate both the x-y deterministic and west-first adaptive routing algorithms. The performances of the algorithms were measured in term of the network throughput. For each algorithm, the network throughput was measured by varying several parameters, specifically, the network size, the packet length, and the offered traffic.

4.1 The Router Model

A router is a hardware unit, as seen in Figure 4.1. It includes a control logic unit, 8 channels and input and output circular flit buffers.

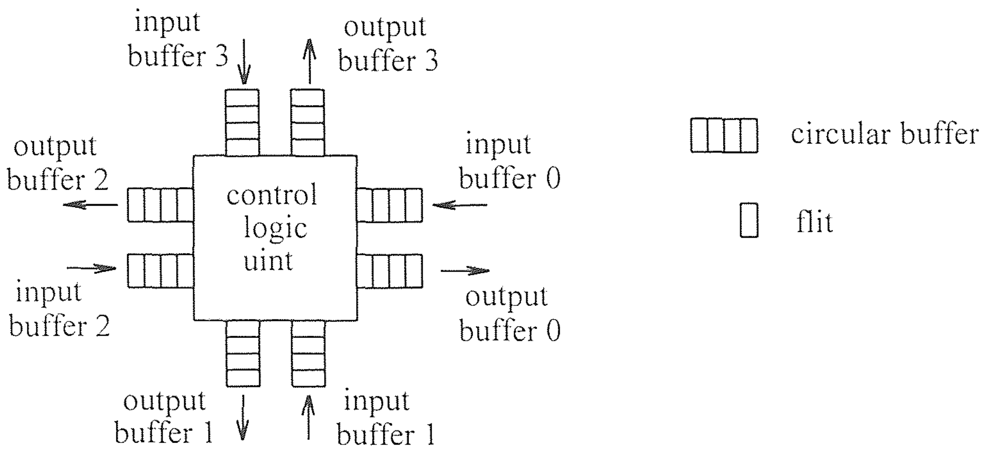


Figure 4.1 The configuration of a router

- ***Control Logic Unit*** The function of this unit is to assign packets arriving from the input buffers to output buffers. This assignment is determined by the routing algorithm. The unit determines the output buffer for a given packet when it receives the header flit of the packet. Trailing flits of the packet are by default assigned to the same output buffer of the header flit.

- ***Buffers*** Each buffer is implemented as a circular FIFO queue. The circular buffer has a head and tail pointer. The difference between two pointers is the number of occupied flits, and its maximum value is the length of the buffer. Figure 4.2 shows a circular buffer.
- ***Channels*** A channel connects an input buffer of a node to an output buffer of a neighboring node. Figure 4.3 illustrates the channels for a 2x2 mesh. A flit at the head of the output buffer is transmitted over the channel only if the corresponding input buffer in the neighboring node is not full.

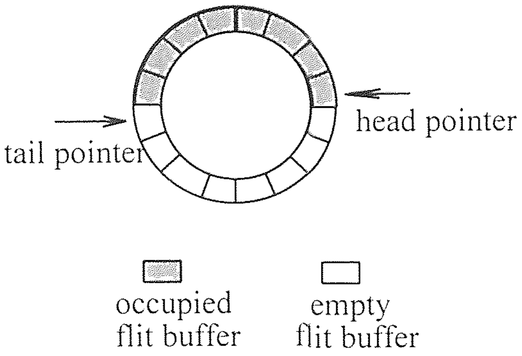


Figure 4.2 A flit circular buffer

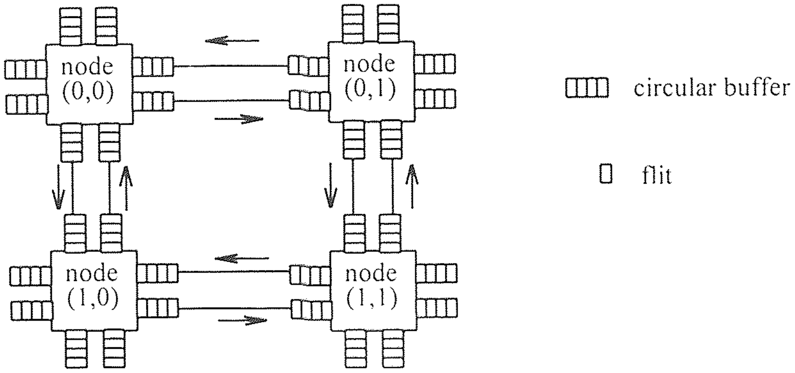


Figure 4.3 A 2 x 2 mesh

4.2 Simulator

Our simulator is a software program with following features:

- The router model described in the previous subsection is used. Each router can accept up to five incoming packets: four from neighboring routers and one from the local processor connected to the router. Packets from neighboring routers always take precedence over the packet from the local processor.
- The mesh size is varied from 4 x 4 (16 nodes) to 64 x 64 (4096 nodes).
- Packet lengths are used 16, 128 and 1024 flits.
- 8-flit buffers are used for each port. (we tested the x-y deterministic routing algorithm under the uniform traffic model, and the result showed that the buffer size has no significant effect on routing time and therefore we fixed the buffer size in the following simulation).
- The offered traffic is defined in terms of packet generation rate and distribution of packet destinations. The packet generation is modeled as a Poisson process. The number of packets generated in each time unit is Poisson distributed. For the packet destination, the following are used:

  Uniform distribution For a given packet, the destination node is chosen with equal probability among the nodes of the network (including the source node).

  Non-uniform distribution We contrived a distribution of packet destinations that forced congestion in some nodes of the network. Specifically, the *transpose permutation* is used, i.e., a packet whose source node is (x,y) has its destination node, (y,x).

  In what follows we will refer to the uniform distribution as the uniform traffic model and the non-uniform distribution as the non-uniform traffic model.

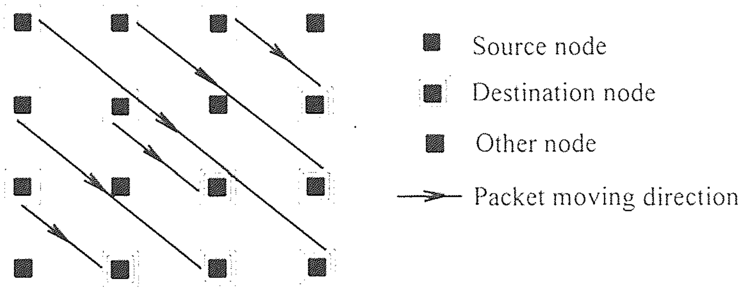


Figure 4.4 An example of transpose distribution

4.3 Experiment Result

Network performance may be unstable before traffic reaches its maximum value. Once the network has reached a steady state, the packet generation rate is equal to the packet reception rate (i.e. the throughput), unless the network is saturated.

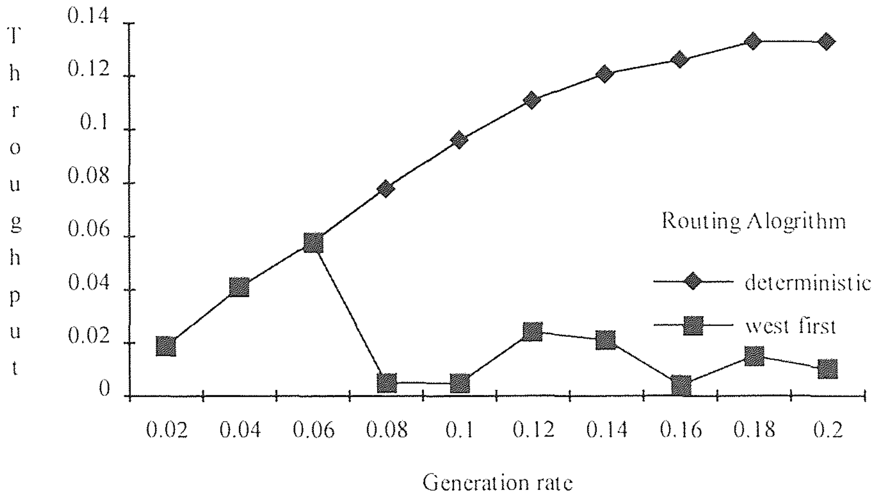


Figure 4.5 Throughput vs generation rate under uniform traffic model for deterministic and west-first algorithms: packet length = 128 flits for 8 x 8 mesh

The throughput of a system is usually defined as the maximum amount of packet delivered per time unit. Under different traffic models, we have measured the throughput of the x-y deterministic and west-first algorithms as function of packet generation rate.

In Figure 4.5, the throughputs of the x-y deterministic and west-first algorithms under the uniform traffic model are shown. It is clear that the x-y deterministic algorithm has a better performance than the west-first algorithm. Note further that the throughput of the west-first algorithm drops rapidly after the generation rate reaches a certain point. The reason is that, as more packets enter the network, more and more packets get misrouted. The undesirable effect is that these misrouted packet get farther away from their destinations, to the point that the time needed to reach their destination is much worse than if the packet simply blocked (as in the x-y deterministic algorithm) instead of being misrouted.

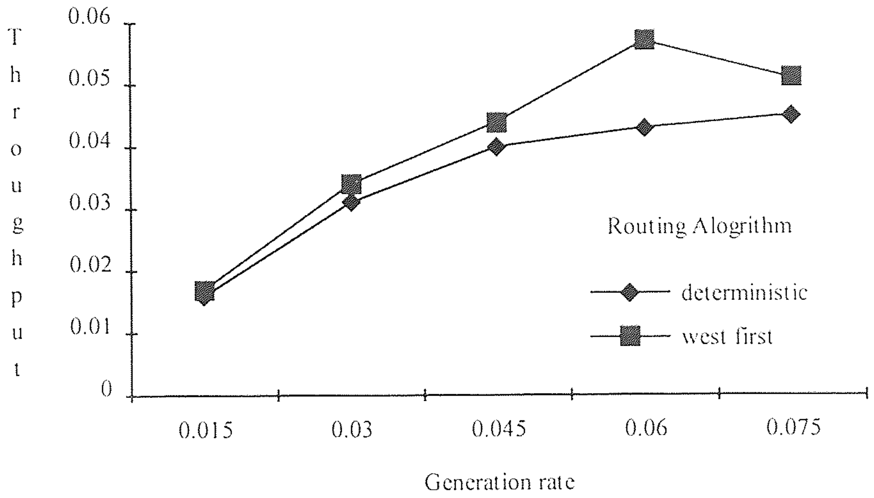


Figure 4.6 Throughput vs generation rate under non-uniform traffic model for deterministic and west-first algorithms: packet length = 128 flits for 8 x 8 mesh.

In Figure 4.6, it can be seen that the west-first algorithm performs better than the deterministic algorithm under the non-uniform traffic model. For non-uniform traffic, some nodes become highly congested with a large number of incoming packets competing for the same output channel. In the deterministic algorithm, all of these packets, except one, block until the channel become free. Thus, packets get closer to their destinations at a slow rate, and throughput suffers. In the adaptive algorithm, packets in highly congested nodes are misrouted to less congested nodes, thereby spreading the traffic evenly across the nodes as time progresses. This, in turn, reduces the number of misroutings and thereby increases throughput.

4.4 The Scalability and Packet Length Effect of the Algorithms

In our experiment, we also examined the scalability of the algorithms (as the network size increases) and the effect of packet length under uniform and non-uniform traffic models.

Figures 4.7 and 4.8 show the network throughput for the x-y deterministic and the west-first algorithms under various mesh sizes under uniform traffic model. We can see that both algorithms scale pretty well, especially the x-y deterministic one. Results for the non-uniform traffic model can be found in Figures 4.9 and 4.10. The reason for which the throughput increases as the mesh size increases is that the packet density in the network decreases when the mesh size goes higher with respect to a fixed packet generation rate, and lower packet density means less blocking and fewer misroutings, or higher throughput.

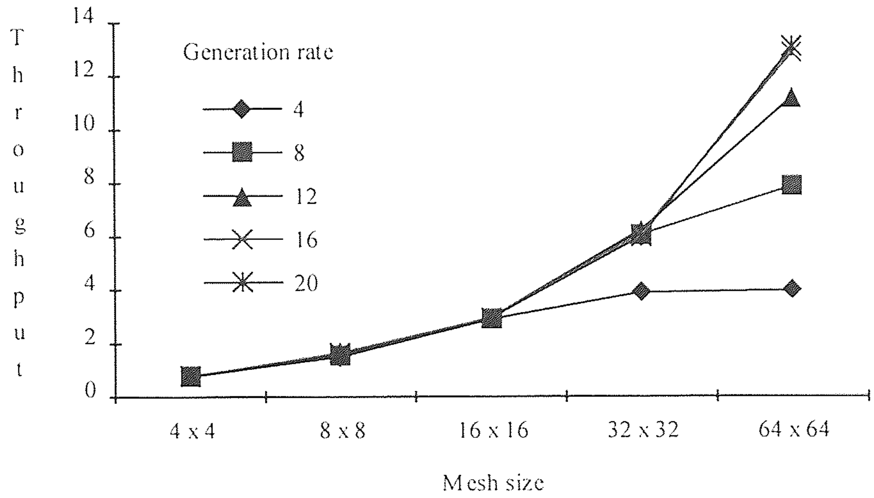


Figure 4.7 Throughput vs mesh size in x-y deterministic algorithm under uniform traffic model: packet length =16 flits

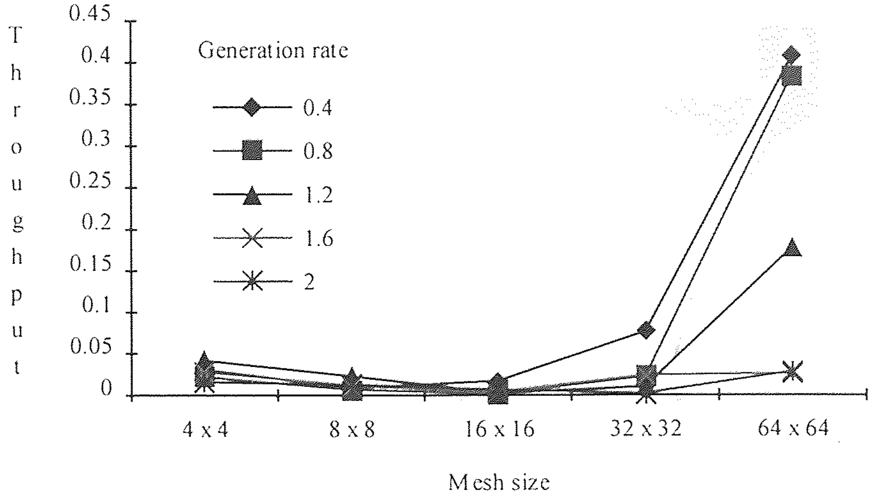


Figure 4.8 Throughput vs mesh size in west-first adaptive algorithm under uniform traffic model: packet length =128 flits

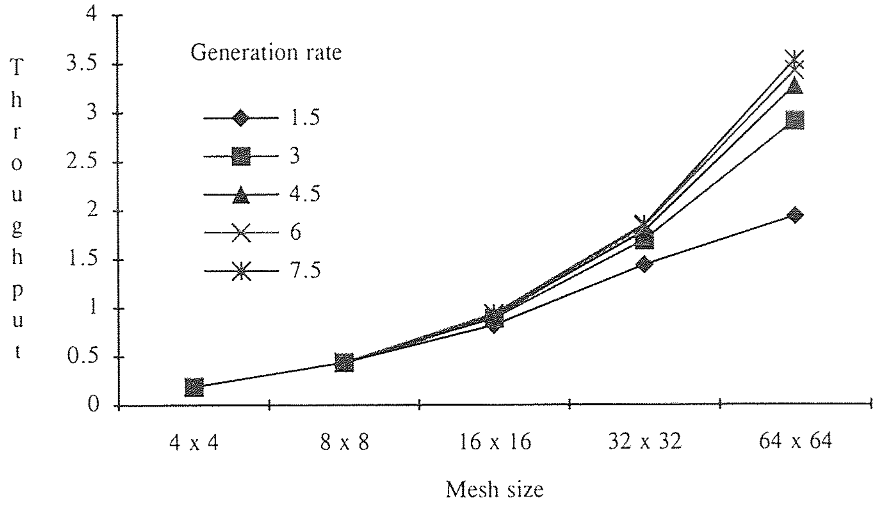


Figure 4.9 Throughput vs mesh size in x-y deterministic algorithm under non-uniform traffic model: packet length =16 flits

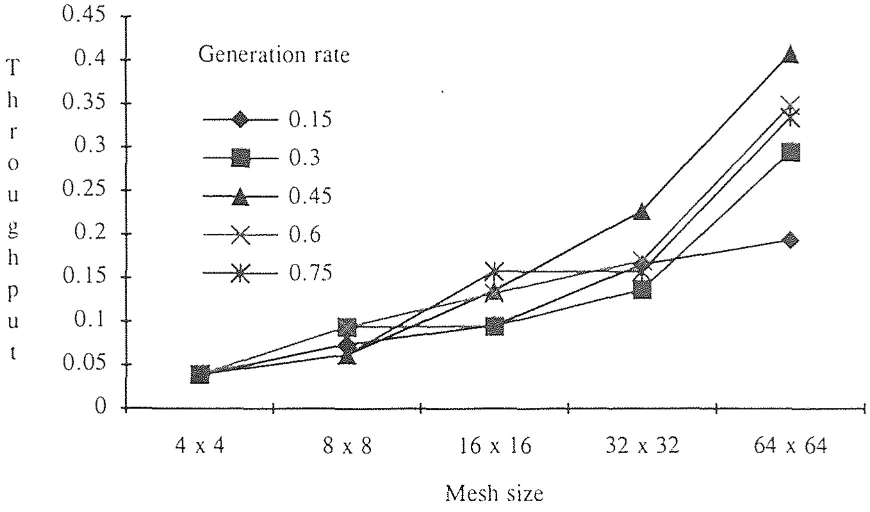


Figure 4.10 Throughput vs mesh size in west-first adaptive algorithm under non-uniform traffic model: packet length =128 flits

Figures 4.11 and 4.12 show the throughputs of the algorithms with respect to different packet lengths under uniform traffic model. They indicate that as the packet length increases, the throughput drops. This phenomena happens because in wormhole routing after the header flit of a packet is sent through a channel, it is reserved for all other flits of the packet until the transmission of the packet is completed. This increases the blocking time of the channel for other packets, and therefore decreases the throughput.

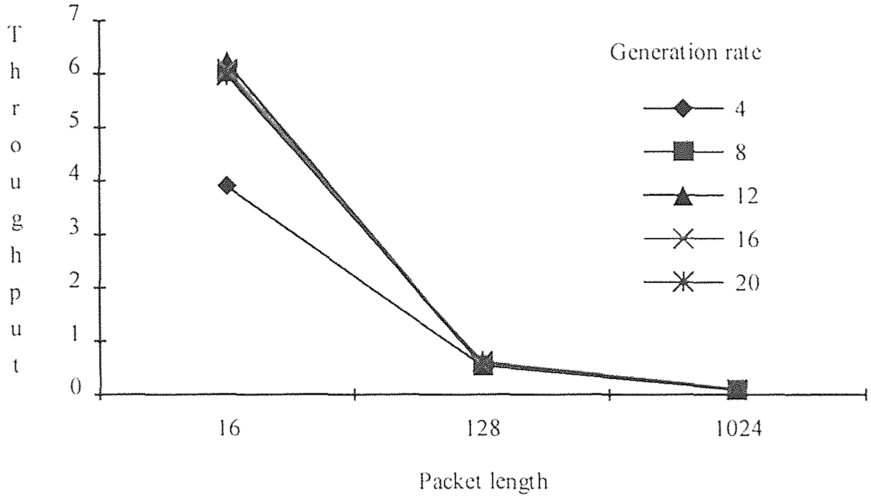


Figure 4.11 Throughput vs packet length in x-y deterministic algorithm under uniform traffic model for 32 x 32 mesh

For the non-uniform traffic model case, a similar result was found, as displayed in Figures 4.13 and 4.14.

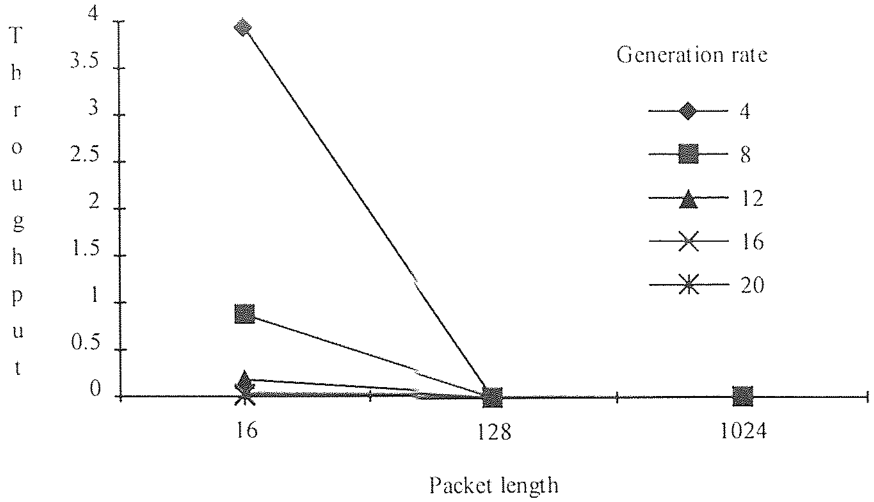


Figure 4.12 Throughput vs packet length in west-first adaptive algorithm under uniform traffic model for 32 x 32 mesh.

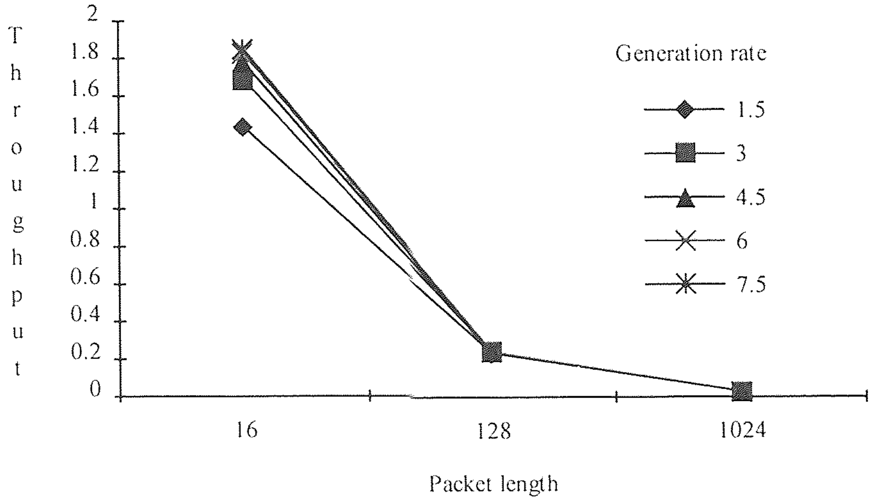


Figure 4.13 Throughput vs packet length in x-y deterministic algorithm under non-uniform traffic model for 32 x 32 mesh

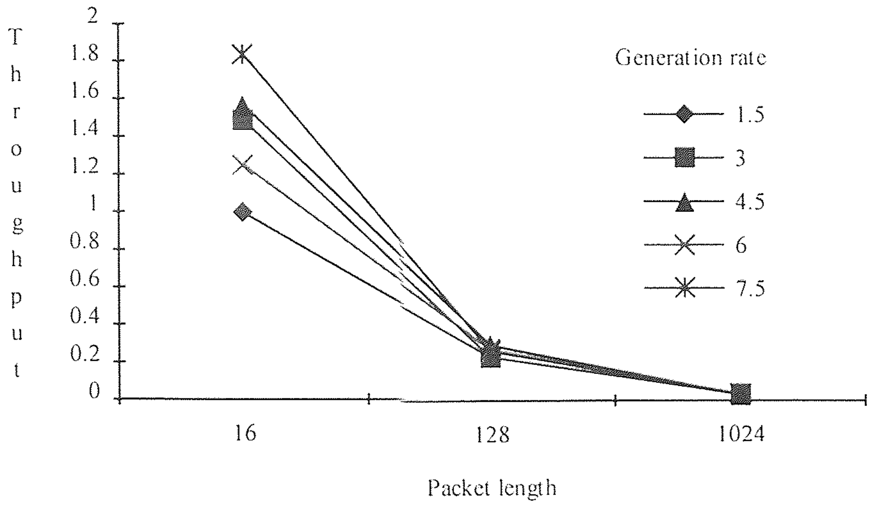


Figure 4.14 Throughput vs packet length in west-first adaptive algorithm under non-uniform traffic model for 32 x 32 mesh.

Observe that the x-y deterministic algorithm has better performance than the west-first algorithm in both scalability and packet length effect tests with uniform traffic model. On the other hand, with the non-uniform traffic model, the west first is better than the x-y deterministic algorithm. This is consistent with the results we have found in section 4.2.

CHAPTER 5

A NEW ADAPTIVE ROUTING ALGORITHM

From the performance results in section 4.2, it is obvious that the x-y deterministic algorithm fits the uniform traffic model better than the west-first, while the west-first algorithm performs more intelligently than the x-y deterministic algorithm in the non-uniform traffic model. The reason is that the deterministic algorithm has completely fixed the way in which a packet is routed, but the adaptive one has not. Hence for non-uniform traffic, where more flexible routing is preferred, the deterministic algorithm performs poorly. On the other hand, the adaptive algorithm routes a packet adaptively, so its chance of misrouting a packet in uniform traffic is higher than in non-uniform traffic.

Intuitively, an algorithm that has a proper balance between determinism and adaptiveness will deliver a better overall performance. In the following section, we introduce a new routing algorithm, called the *west-north-first* routing algorithm.

5.1 West-north-first Adaptive Routing Algorithm

The west-north-first routing algorithm works like this: if a packet destination is to the west and /or north of the source (i.e. it can be west-north, north, north-east, west or west-south), the algorithm routes the packet to west then north, (or only west or north if movement in the other direction is not needed). It stops once the packet is "aligned" with its destination (i.e., either in the same row or the same column). After that, the algorithm routes the packet adaptively, obeying the turn model shown in Figure 5.1(b): if the packet destination is to the south and / or east, the algorithm routes the packet adaptively and follows the turn model. The turn model of the algorithm is shown in Figure 5.1(b), together with the turn models of the other two routing algorithms.

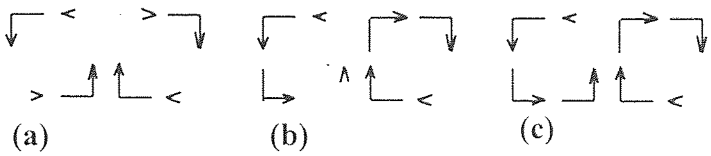


Figure 5.1 An illustration of the turn model in a 2D mesh: (a) four turns (solid arrows) allowed in X-Y routing; (b) five turns (solid arrows) allowed in west-north routing; (c) six turns (solid arrows) allowed in west-first routing.

In Figure 5.1(b), the turns prohibited are turns from north to west, from east to north and from south to west. Therefore, to travel west and north, a packet must begin in west then north directions. Because cycles are avoided, west-north-first routing is deadlock-free. The algorithm is deterministic while moving west and / or north; henceforth it is fully adaptive. Figure 5.2 shows four examples of west-north-first in a 2D 8x8 mesh.

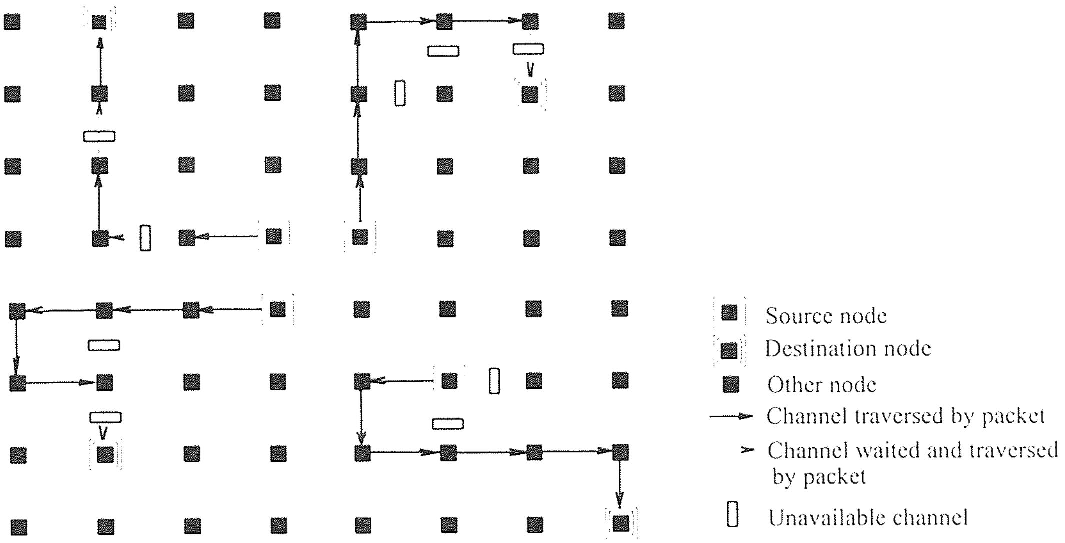

Source node
Destination node
Other node
Channel traversed by packet
Channel waited and traversed by packet
Unavailable channel

Figure 5.2 Four examples of west-north-first routing in a 8 x 8 2D mesh.

Observer that the west-north-first routing algorithm is deterministic in two directions (west and north) and adaptive in the other two. On the other hand, the west-first algorithm is deterministic in one direction (west) and adaptive in the other three.

Hence we say that the west-north-first algorithm is balanced with respect to determinism and adaptiveness.

5.2 Experiment Results

As implied by the performance results of x-y deterministic and west-first algorithms, under the uniform traffic model, algorithms with more deterministic sense have better performance than those with less because misrouting in these algorithms doesn't exist or happens less likely. In contrast, algorithms with more adaptiveness, in the non-uniform traffic model, will be better than those with less.

To give the performance result of the west-north-first algorithm and compare it with the x-y deterministic, and the west-first algorithms, we merge the throughput curves of these three algorithms into one chart with respective to uniform and non-uniform traffic model, as in Figure 5.3 and Figure 5.4.

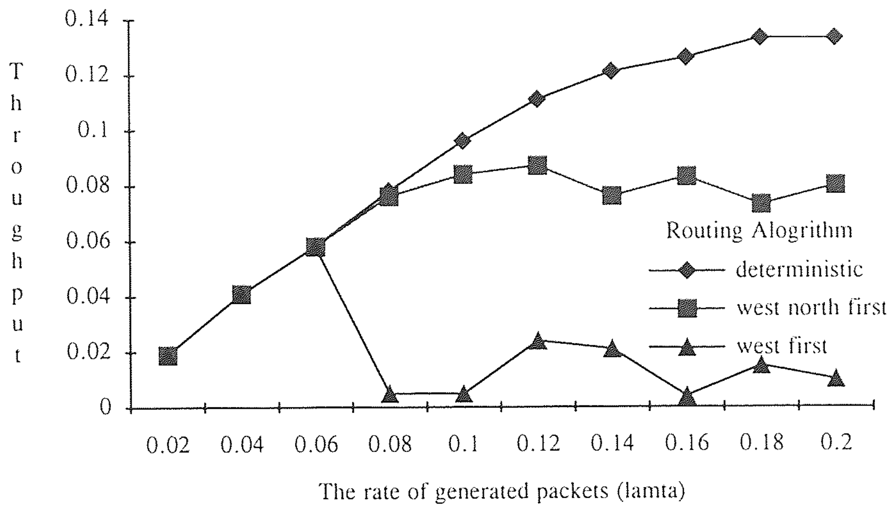


Figure 5.3 Throughput vs generation rate under uniform traffic model for deterministic and adaptive algorithms: packet length = 128 flits for 8 x 8 mesh

The x-y deterministic algorithm has fixed the way in which one packet is to be routed. The west-first algorithm routes a packet toward west at the very beginning if going west is necessary, then to the other directions adaptively as needed. The west-north-first algorithm stands between the x-y deterministic and the west-first algorithms in the sense that it is more adaptive than the x-y deterministic algorithm, and more deterministic than the west-first algorithm. Because of this characteristic, the west-north first algorithm should be better than the x-y deterministic algorithm in the non-uniform traffic model and better than west-first in the uniform traffic model. Our experimental results validate this hypothesis. In Figure 5.3 (uniform traffic model), the west-north-first algorithm is superior to west-first algorithm, but inferior to the x-y deterministic algorithm. For the non-uniform traffic model, as Figure 5.4 indicates, the west-north-first algorithm is better than the x-y deterministic algorithm and close to the west-first algorithm.

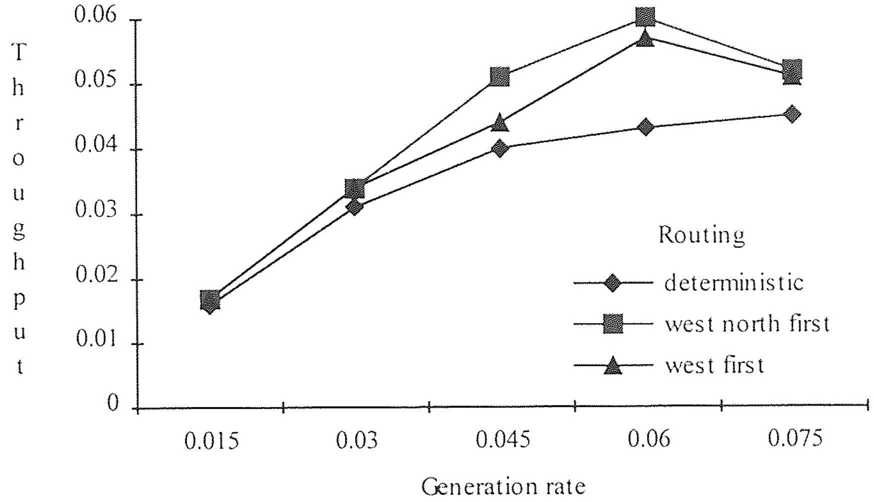


Figure 5.4 Throughput vs generation rate under non-uniform traffic model for deterministic and adaptive algorithms :
packet length = 128 flits

5.3 The Scalability and Packet Length Effect of the Algorithm

The scalability of the west-north-first algorithm is as good as those of x-y deterministic and west-first algorithm, as shown in Figures 5.5 and 5.6.

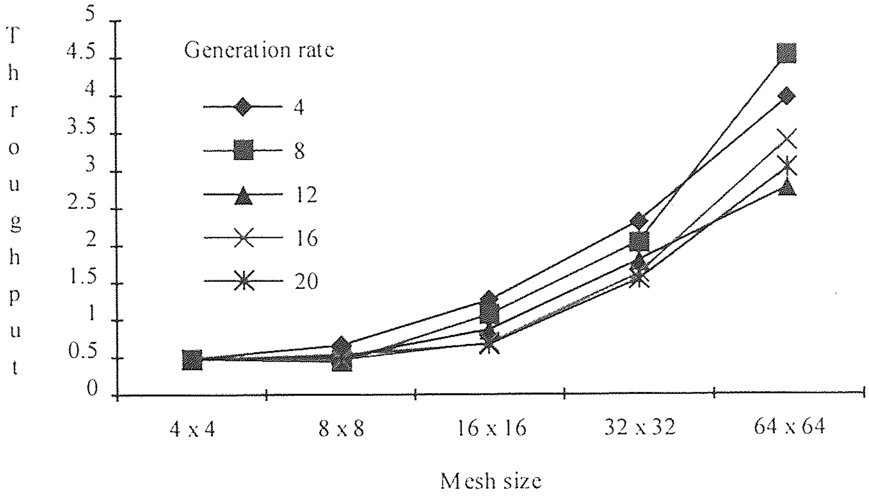


Figure 5.5 Throughput vs mesh size in west-north-first adaptive algorithm under uniform traffic model: packet length = 16 flits

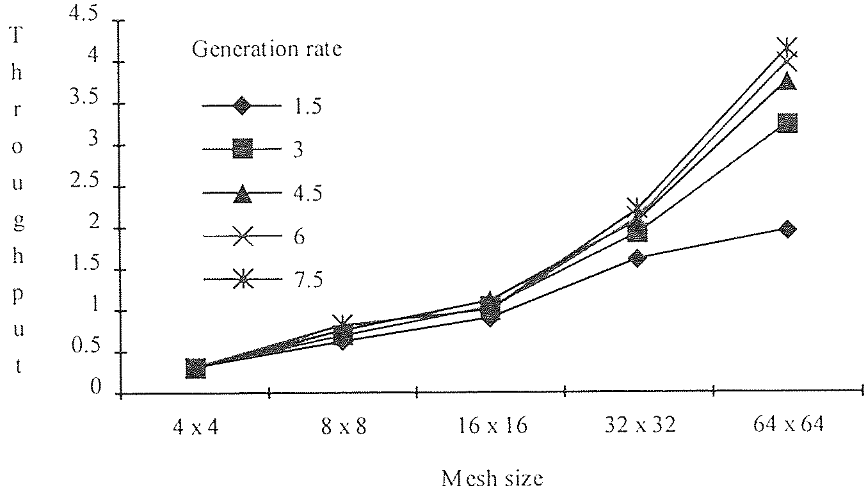


Figure 5.6 Throughput vs mesh size in west-north-first adaptive algorithm under non-uniform traffic model: packet length = 128 flits

Figures 5.7 and 5.8 show the throughputs of the algorithm with respect to different packet lengths in uniform and non uniform traffic model.

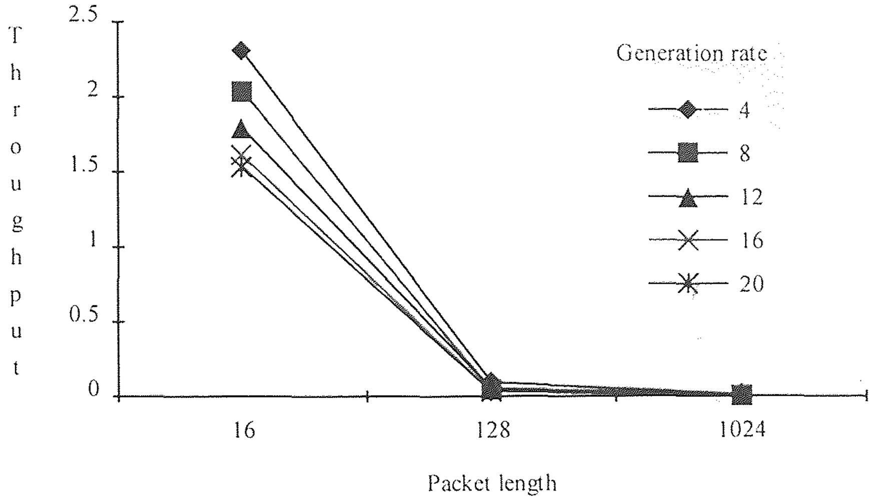


Figure 5.7 Throughput vs packet length in west-north-first adaptive algorithm under uniform traffic model for 32 x 32 mesh

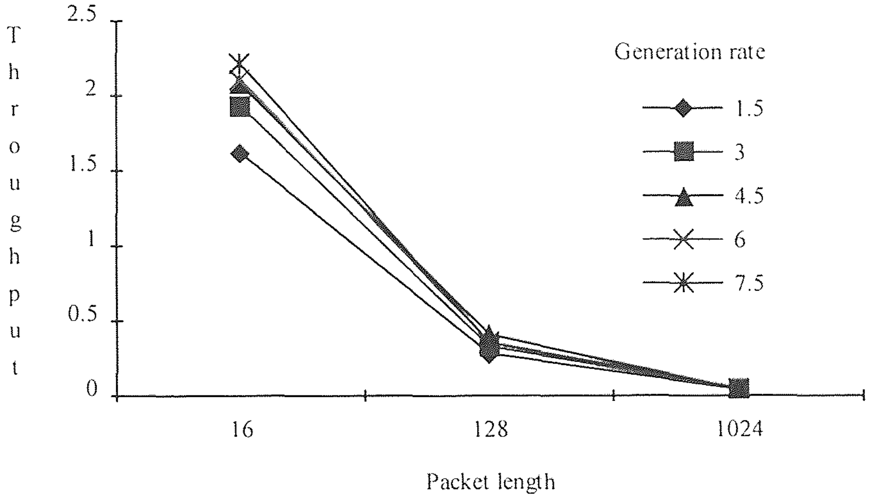


Figure 5.8 Throughput vs packet length in west-north-first adaptive algorithm under non-uniform traffic model for 32 x 32 mesh

CHAPTER 6

CONCLUSIONS

In this thesis, we have proposed and studied a new routing algorithm, the west-north-first algorithm. We have also evaluated some other algorithms, such as the x-y deterministic and the west-first adaptive routing algorithms and compared them with the west-north-first one.

From the performance results, the deterministic algorithm is more efficient under uniform traffic model than under the non-uniform traffic model, such as the transpose permutation. Our simulation shows that the network throughput in this case is about 50 percent higher.

On the other hand, the west-first algorithm, which is adaptive except for the west direction, can much better handle non-uniform traffic than uniform traffic. The throughput of the west-first algorithm declines as the number of packets in the network increases and is unstable when facing the non-uniform traffic model.

The west-north-first algorithm, combining the characteristics of deterministic and adaptive routing algorithms, has good performance in both uniform and non-uniform traffic models. The performance of the west-north-first is close to the one from the deterministic algorithm in the uniform traffic model, while in non-uniform traffic model, its performance is even better than the one of west-first algorithm. From this, we can conclude that the west-north-first algorithm gives a better overall performance than the other two. The major reason is that it has a good balance between deterministic and adaptive algorithms in the sense of reducing misrouting and latency.

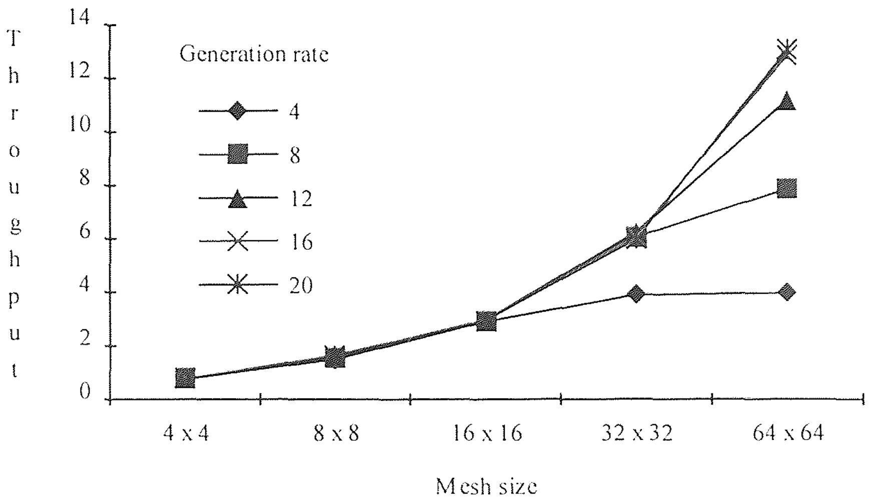
For further research, using virtual channel technique on top of our west-north-first adaptive routing is very interesting and worth pressured. The virtual channel technique is to divide a flit buffer associated with each network channel into several virtual channels and each virtual channel is dedicated to one packet. With virtual channel, deadlock can be avoided by making routing relation acyclic.

APPENDIX A

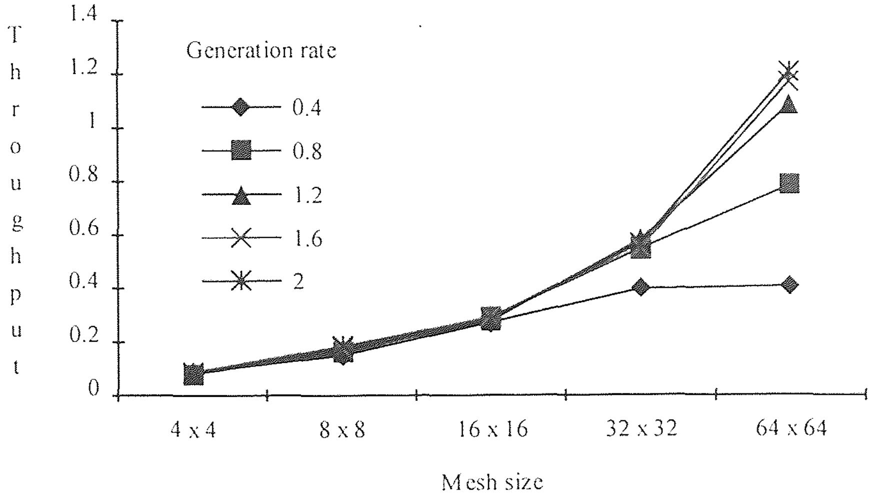
COMPARISON OF THE THROUGHPUT BY VARYING MESH SIZE

1. X-Y Deterministic Routing with Uniform Traffic Model

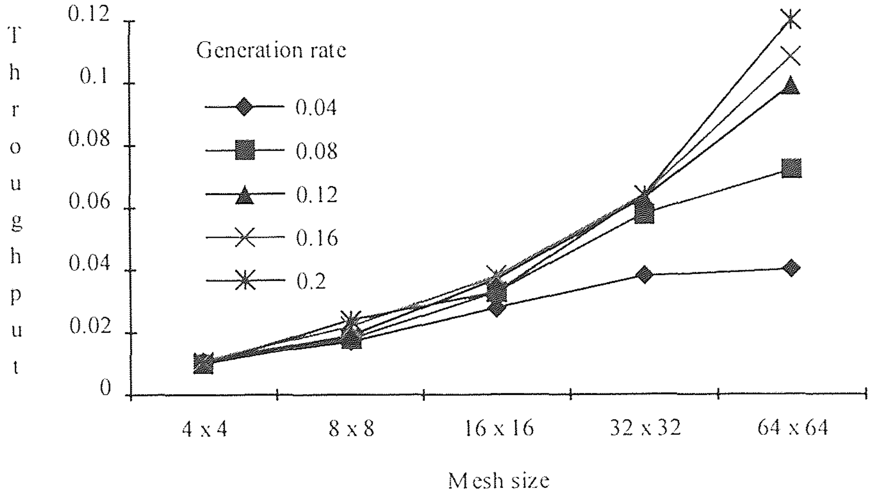
X-Y (uniform)	λ (the number of packets generated per time unit)				
Packet Len. 16	4	8	12	16	20
4 x 4	0.767	0.770	0.777	0.767	0.760
8 x 8	1.501	1.566	1.604	1.635	1.642
16 x 16	2.901	2.963	2.957	2.953	2.957
32 x 32	3.895	6.059	6.225	6.099	5.977
64 x 64	3.969	7.875	11.20	12.83	13.06



X-Y(uniform)	λ (the number of packets generated per time unit)				
Packet Len.128	0.4	0.8	1.2	1.6	2.0
4 x 4	0.085	0.080	0.089	0.089	0.088
8 x 8	0.150	0.161	0.170	0.182	0.184
16 x 16	0.273	0.291	0.279	0.294	0.289
32 x 32	0.397	0.548	0.580	0.550	0.568
64 x 64	0.408	0.786	1.088	1.174	1.208

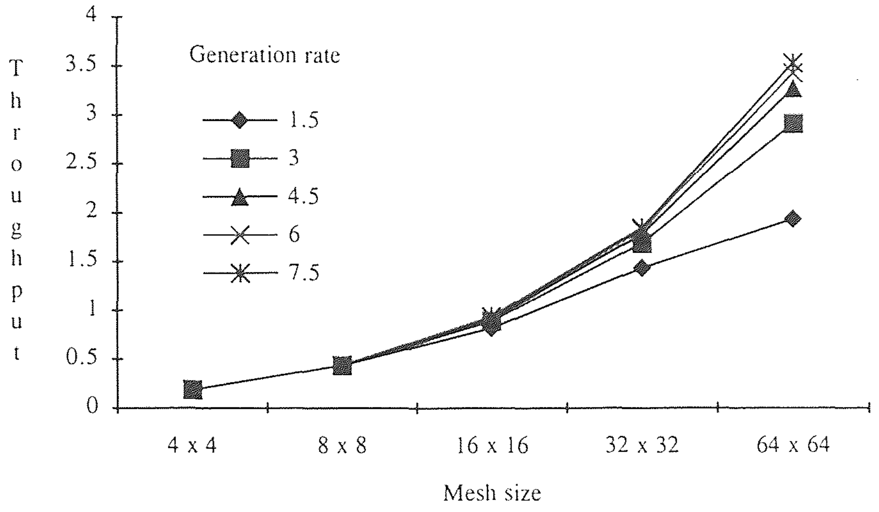


X-Y(uniform)	λ (the number of packets generated per time unit)				
Packet Len.1024	0.04	0.08	0.12	0.16	0.20
4 x 4	0.011	0.010	0.011	0.011	0.010
8 x 8	0.017	0.018	0.019	0.022	0.024
16 x 16	0.028	0.033	0.037	0.038	0.033
32 x 32	0.038	0.058	0.063	0.064	0.064
64 x 64	0.040	0.072	0.099	0.108	0.120

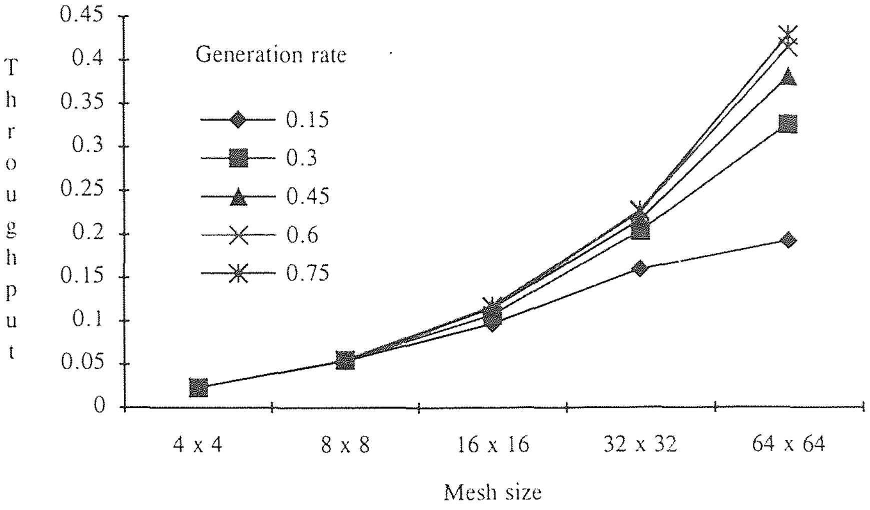


2. X-Y Deterministic Routing with Non-uniform Traffic Model

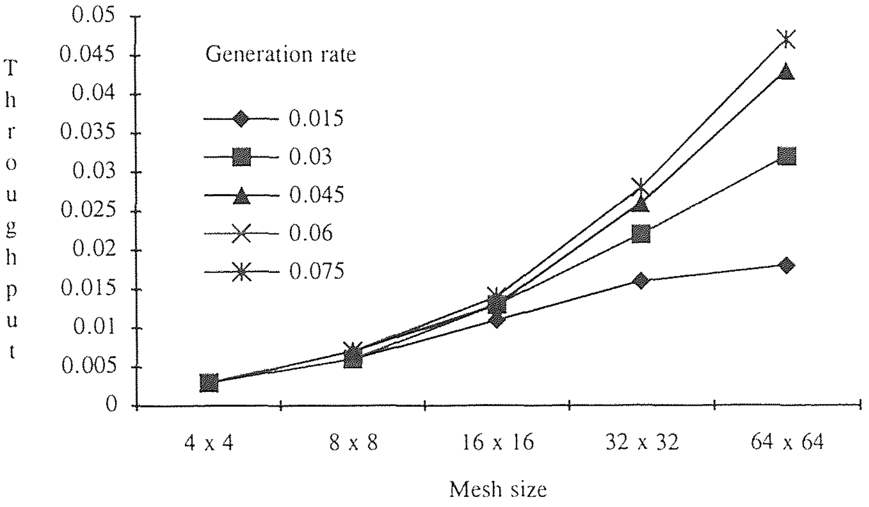
X-Y (non-uniform) Packet Len 16	λ (the number of packets generated per time unit)				
	1.5	3.0	4.5	6.0	7.5
4 x 4	0.188	0.188	0.188	0.188	0.188
8 x 8	0.437	0.438	0.438	0.438	0.438
16 x 16	0.820	0.888	0.916	0.937	0.937
32 x 32	1.434	1.689	1.779	1.828	1.849
64 x 64	1.935	2.912	3.270	3.425	3.532



X-Y (non-uniform) Packet Len 128	λ (the number of packets generated per time unit)				
	0.15	0.30	0.45	0.60	0.75
4 x 4	0.023	0.023	0.023	0.023	0.023
8 x 8	0.054	0.055	0.055	0.055	0.055
16 x 16	0.097	0.107	0.115	0.116	0.117
32 x 32	0.160	0.204	0.217	0.226	0.228
64 x 64	0.193	0.327	0.382	0.415	0.429

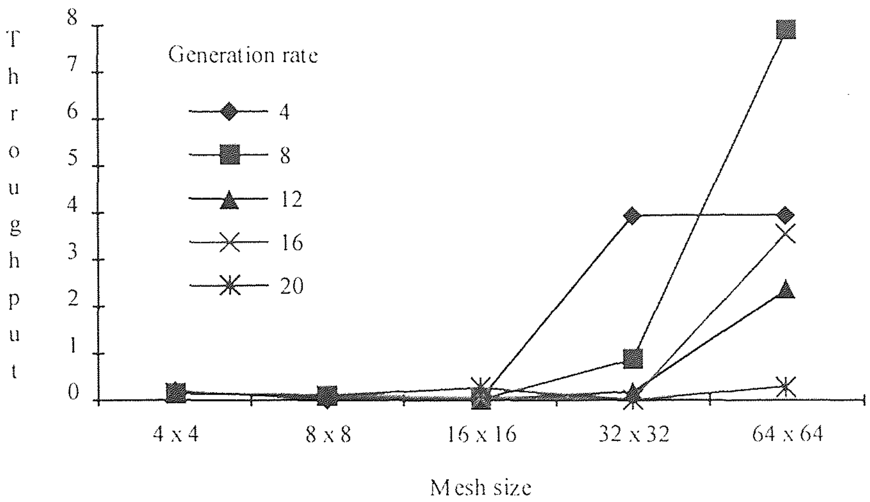


X-Y (non-uniform) Packet Len 1024	λ (the number of packets generated per time unit)				
	0.015	0.03	0.045	0.060	0.075
4 x 4	0.003	0.003	0.003	0.003	0.003
8 x 8	0.006	0.006	0.007	0.007	0.007
16 x 16	0.011	0.013	0.013	0.014	0.014
32 x 32	0.016	0.022	0.026	0.028	0.028
64 x 64	0.018	0.032	0.043	0.047	0.047

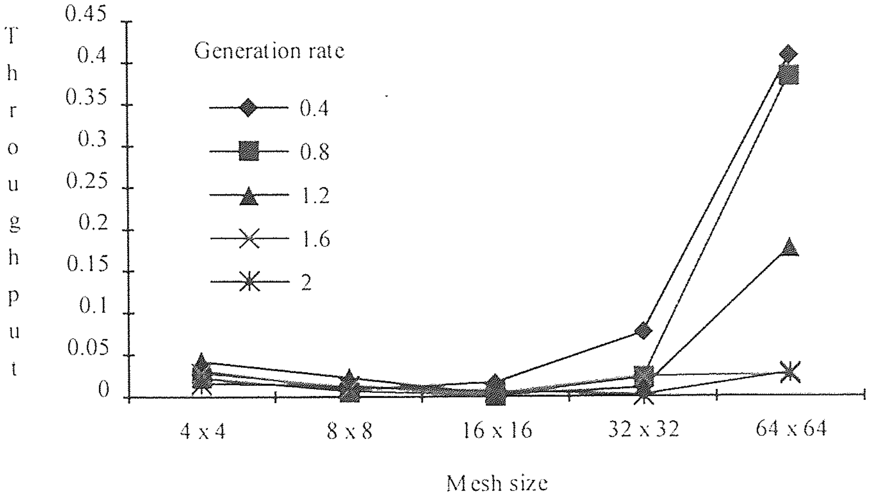


3. West-First Adaptive Routing with Uniform Traffic Model

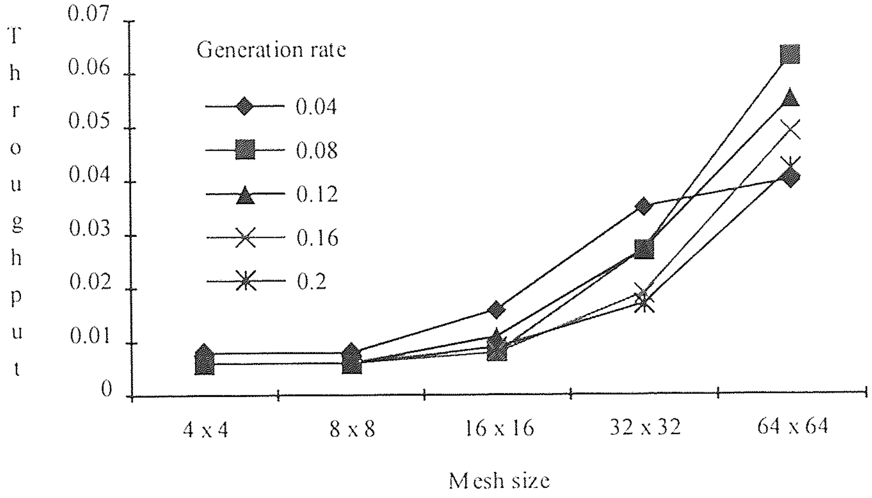
W-F (uniform)	λ (the number of packets generated per time unit)				
Packet Len. 16	4	8	12	16	20
4 x 4	0.197	0.151	0.152	0.151	0.150
8 x 8	0.026	0.092	0.125	0.123	0.118
16 x 16	0.008	0.018	0.033	0.058	0.269
32 x 32	3.943	0.871	0.181	0.039	0.011
64 x 64	3.962	7.911	2.354	3.557	0.296



W-F (uniform)	λ (the number of packets generated per time unit)				
Packet Len. 128	0.4	0.8	1.2	1.6	2.0
4 x 4	0.031	0.022	0.042	0.028	0.016
8 x 8	0.009	0.007	0.023	0.012	0.013
16 x 16	0.017	0.002	0.003	0.004	0.007
32 x 32	0.076	0.022	0.010	0.024	0.002
64 x 64	0.408	0.385	0.178	0.026	0.028

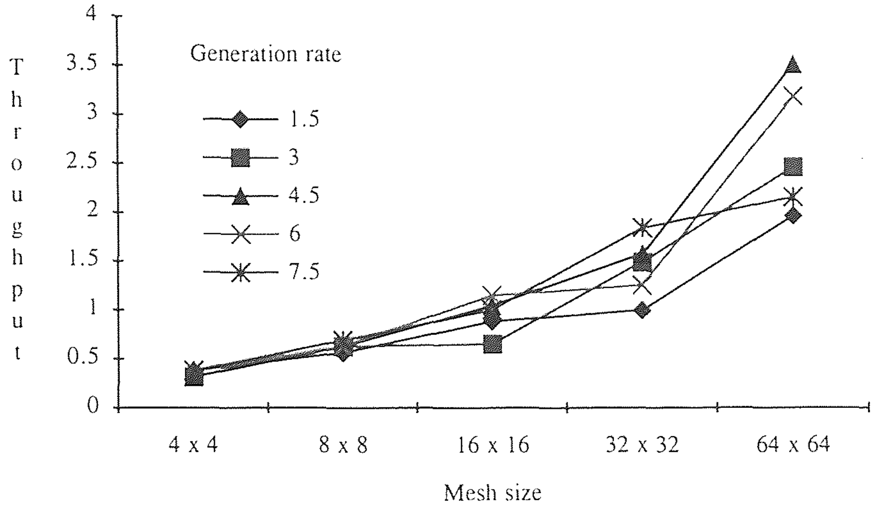


W-F (uniform) Packet Len. 1024	λ (the number of packets generated per time unit)				
	0.04	0.08	0.12	0.16	0.20
4 x 4	0.008	0.006	0.006	0.006	0.006
8 x 8	0.008	0.006	0.006	0.006	0.006
16 x 16	0.016	0.008	0.011	0.008	0.009
32 x 32	0.035	0.027	0.027	0.019	0.017
64 x 64	0.040	0.063	0.055	0.049	0.042

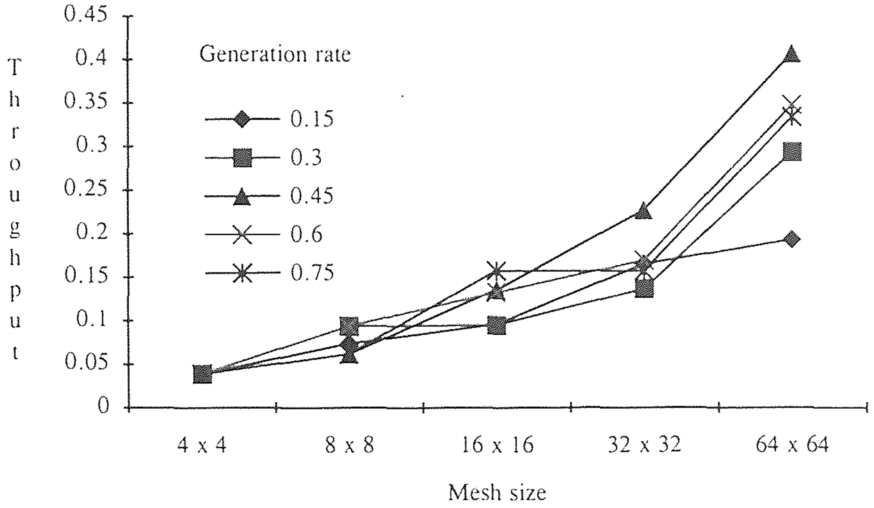


4. West-First Adaptive Routing with Non-uniform Traffic Model

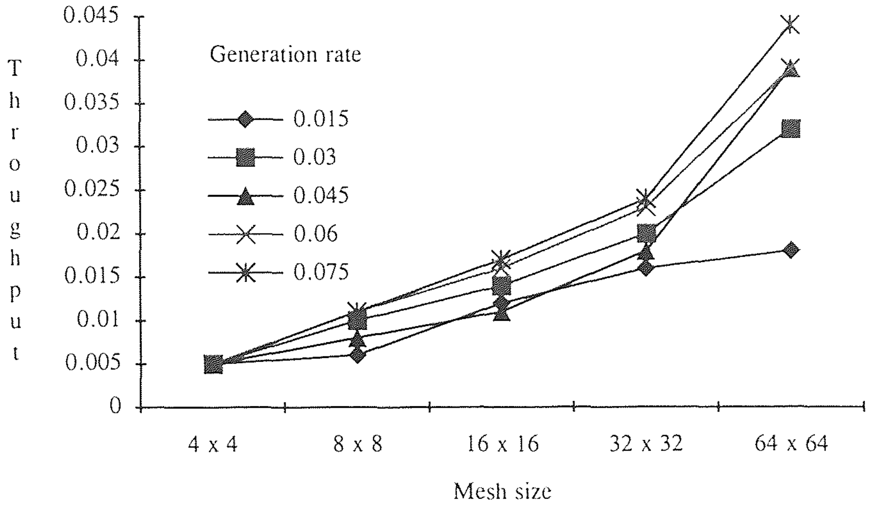
W-F (non-uniform) Packet Len 16	λ (the number of packets generated per time unit)				
	1.5	3.0	4.5	6.0	7.5
4 x 4	0.375	0.313	0.313	0.376	0.376
8 x 8	0.559	0.625	0.625	0.626	0.688
16 x 16	0.881	0.652	1.043	1.149	1.000
32 x 32	0.995	1.488	1.568	1.251	1.836
64 x 64	1.962	2.462	3.509	3.181	2.154



W-F (non-uniform) Packet Len 128	λ (the number of packets generated per time unit)				
	0.15	0.30	0.45	0.60	0.75
4 x 4	0.039	0.039	0.039	0.039	0.039
8 x 8	0.074	0.094	0.062	0.094	0.062
16 x 16	0.096	0.095	0.136	0.133	0.158
32 x 32	0.166	0.137	0.228	0.170	0.158
64 x 64	0.194	0.295	0.408	0.349	0.335

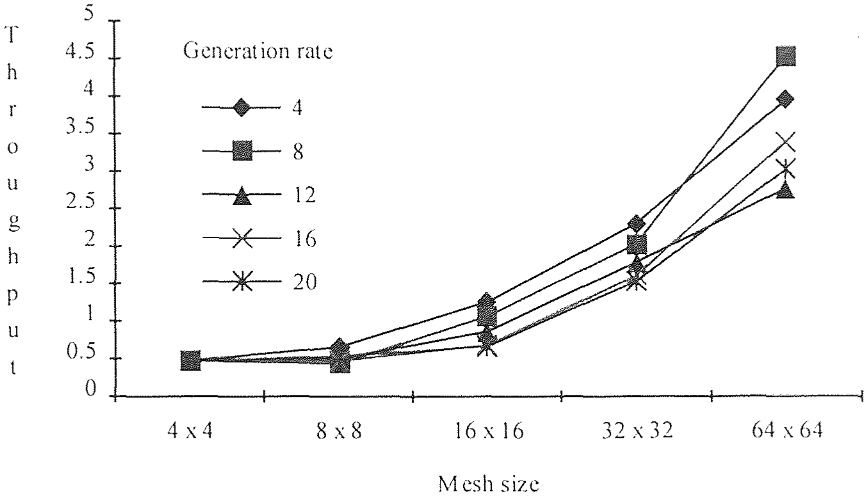


W-F (non-uniform) Packet Len 1024	λ (the number of packets generated per time unit)				
	0.015	0.03	0.045	0.060	0.075
4 x 4	0.005	0.005	0.005	0.005	0.005
8 x 8	0.006	0.010	0.008	0.011	0.011
16 x 16	0.012	0.014	0.011	0.016	0.017
32 x 32	0.016	0.020	0.018	0.023	0.024
64 x 64	0.018	0.032	0.039	0.039	0.044

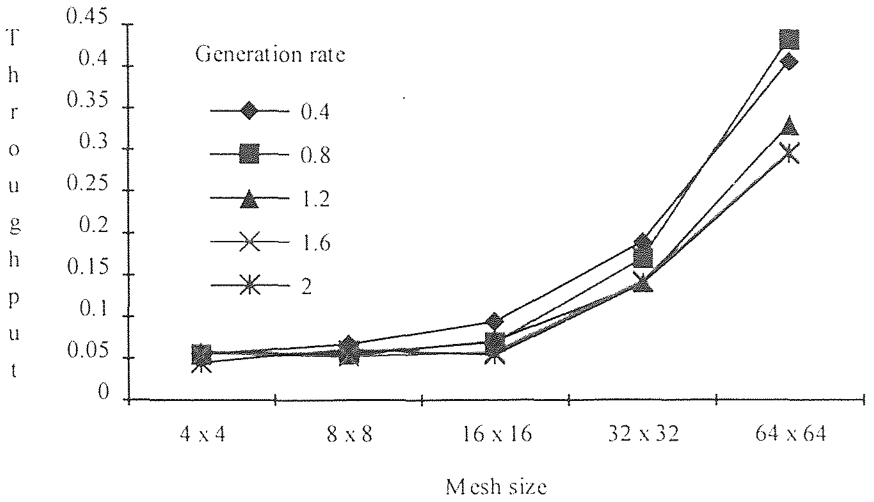


5. West-North-First Adaptive Routing with Uniform Traffic Model

W-N-F (uniform)	λ (the number of packets generated per time unit)				
Packet Len. 16	4	8	12	16	20
4 x 4	0.480	0.474	0.475	0.472	0.469
8 x 8	0.658	0.441	0.502	0.468	0.540
16 x 16	1.261	1.070	0.866	0.686	0.669
32 x 32	2.307	2.038	1.794	1.611	1.536
64 x 64	3.955	4.547	2.773	3.406	3.027

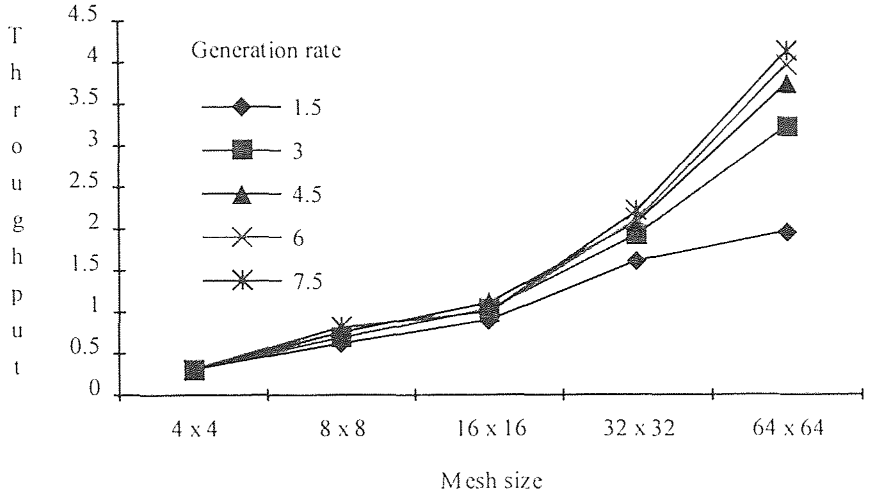


W-N-F (uniform)	λ (the number of packets generated per time unit)				
Packet Len. 128	0.4	0.8	1.2	1.6	2.0
4 x 4	0.055	0.055	0.058	0.056	0.045
8 x 8	0.067	0.057	0.054	0.052	0.061
16 x 16	0.094	0.069	0.071	0.058	0.054
32 x 32	0.190	0.170	0.140	0.143	0.140
64 x 64	0.407	0.434	0.331	0.297	0.295

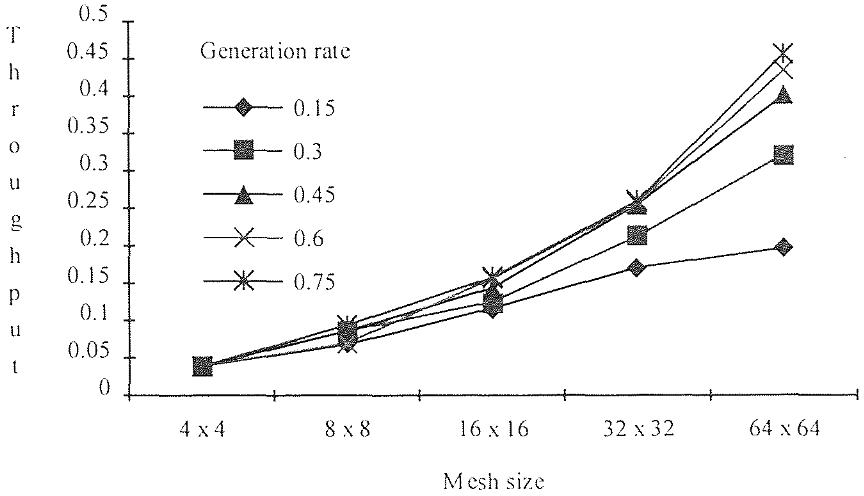


6. West-North-First Adaptive Routing with Non-uniform Traffic Model

W-N-F (non-uniform) Packet Len 16	λ (the number of packets generated per time unit)				
	1.5	3.0	4.5	6.0	7.5
4 x 4	0.313	0.313	0.313	0.313	0.313
8 x 8	0.619	0.688	0.751	0.814	0.814
16 x 16	0.902	1.037	1.111	1.000	1.000
32 x 32	1.616	1.930	2.077	2.109	2.211
64 x 64	1.959	3.227	3.736	3.969	4.125



W-N-F (non-uniform) Packet Len 128	λ (the number of packets generated per time unit)				
	0.15	0.30	0.45	0.60	0.75
4 x 4	0.039	0.039	0.039	0.039	0.039
8 x 8	0.068	0.086	0.086	0.070	0.094
16 x 16	0.117	0.125	0.144	0.156	0.158
32 x 32	0.171	0.213	0.255	0.257	0.260
64 x 64	0.197	0.321	0.401	0.435	0.457

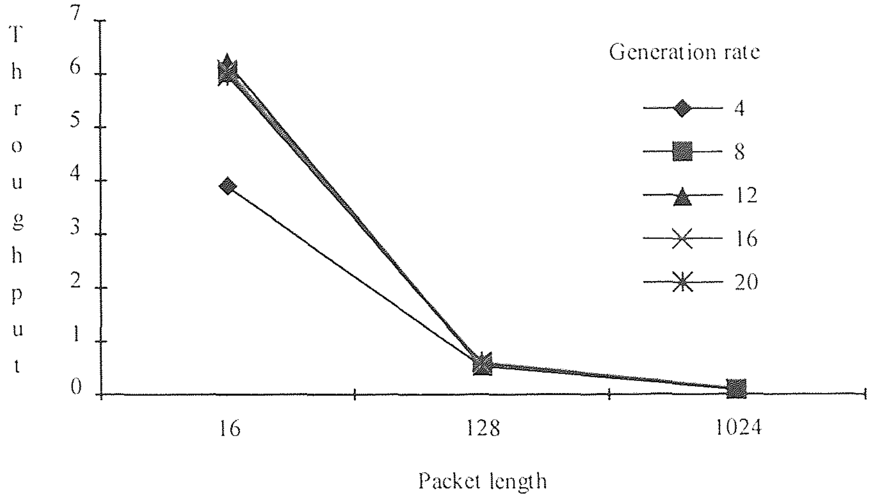


APPENDIX B

COMPARISON OF THE THROUGHPUT BY VARYING PACKET LENGTH

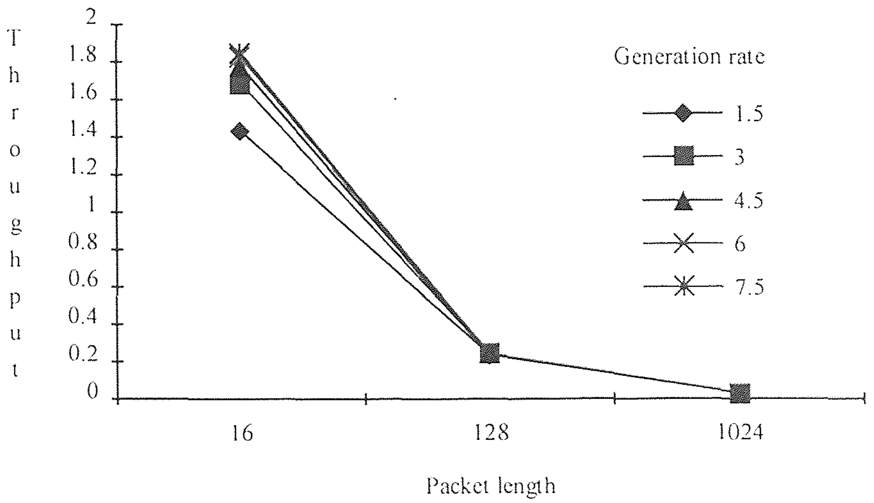
1. X-Y Deterministic Routing with Uniform Traffic Model

Mesh Size		λ (the number of packets generated per time unit)				
32 x 32		4	8	12	16	20
Packet	16	3.895	6.059	6.225	6.099	5.977
Length	128	0.541	0.544	0.553	0.580	0.615
(flit)	1024	0.085	0.093	0.097	0.100	0.106



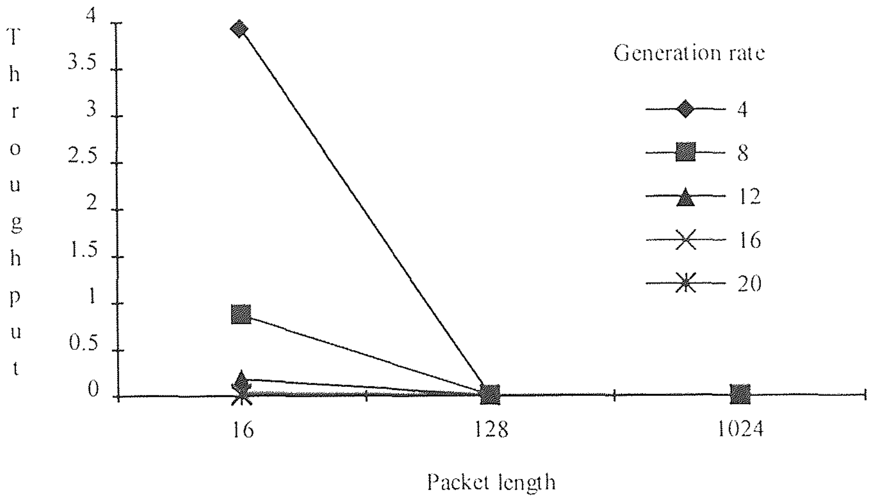
2 X-Y Deterministic Routing with Non-Uniform Traffic Model

Mesh Size		λ (the number of packets generated per time unit)				
32 x 32		1.5	3.0	4.5	6.0	7.5
Packet	16	1.434	1.689	1.779	1.828	1.849
Length	128	0.236	0.240	0.242	0.242	0.242
(flit)	1024	0.031	0.031	0.031	0.031	0.031



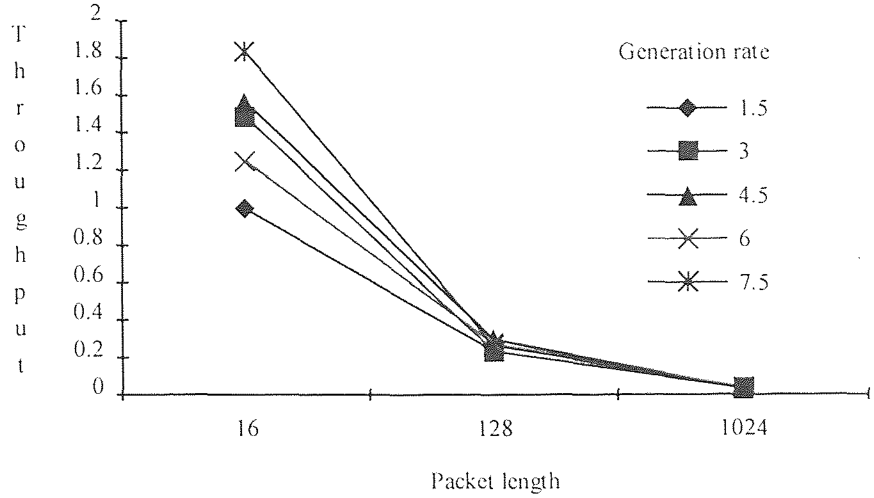
3. West-First Adaptive Routing with Uniform Traffic Model

Mesh Size 32 x 32		λ (the number of packets generated per time unit)				
		4	8	12	16	20
Packet Length (flit)	16	3.943	0.871	0.181	0.039	0.011
	128	0.005	0.008	0.009	0.015	0.013
	1024	0.008	0.007	0.005	0.006	0.006



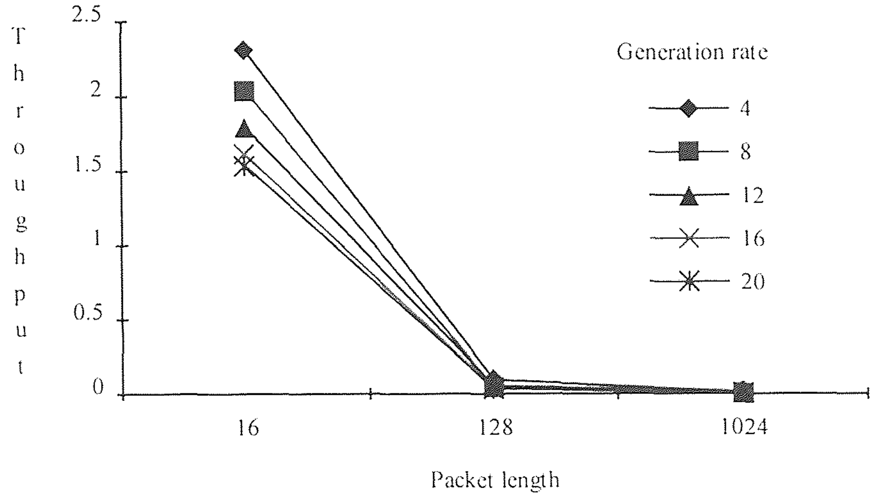
4. West-First Adaptive Routing with Non-Uniform Traffic Model

Mesh Size 32 x 32		λ (the number of packets generated per time unit)				
		1.5	3.0	4.5	6.0	7.5
Packet Length (flit)	16	0.995	1.488	1.568	1.251	1.836
	128	0.234	0.233	0.297	0.273	0.265
	1024	0.037	0.033	0.031	0.036	0.036



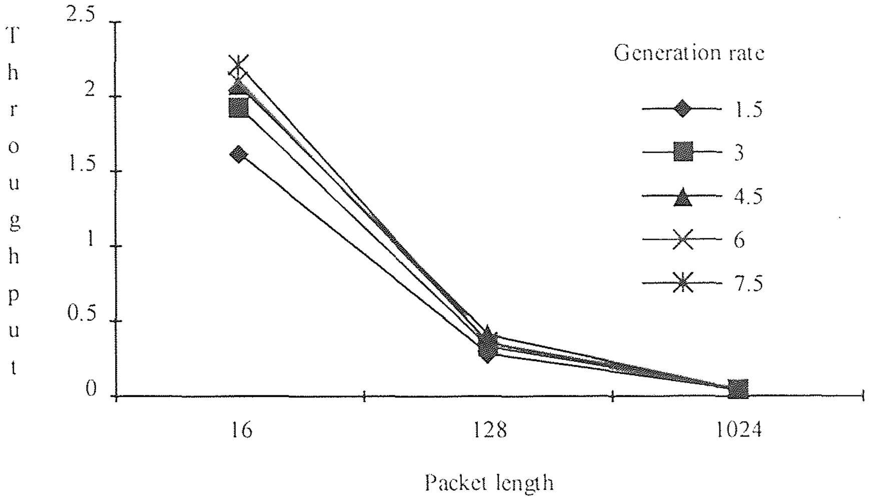
5. West-North-First Adaptive Routing with Uniform Traffic Model

Mesh Size 32 x 32		λ (the number of packets generated per time unit)				
		4	8	12	16	20
Packet Length (flit)	16	2.307	2.038	1.794	1.611	1.536
	128	0.096	0.048	0.056	0.035	0.033
	1024	0.014	0.010	0.011	0.010	0.009



6. West-North-First Adaptive Routing with Non-Uniform Traffic Model

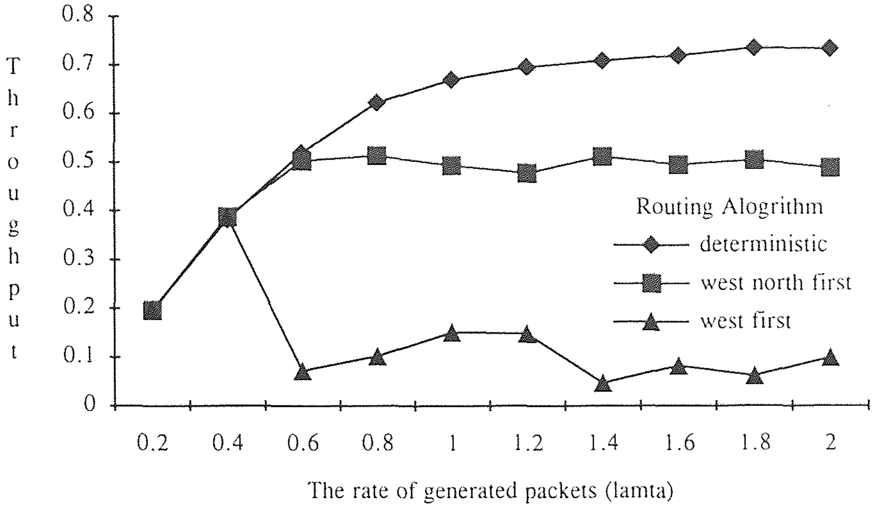
Mesh Size 32 x 32		λ (the number of packets generated per time unit)				
		1.5	3.0	4.5	6.0	7.5
Packet Length (flit)	16	1.616	1.930	2.077	2.109	2.211
	128	0.285	0.333	0.411	0.359	0.359
	1024	0.043	0.045	0.044	0.043	0.050



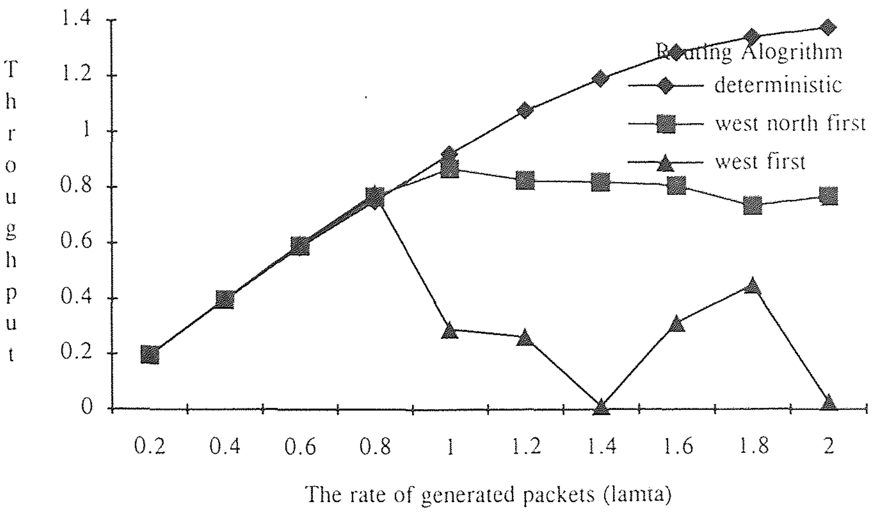
APPENDIX C

COMPARISON OF THROUGHPUT UNDER UNIFORM TRAFFIC MODEL

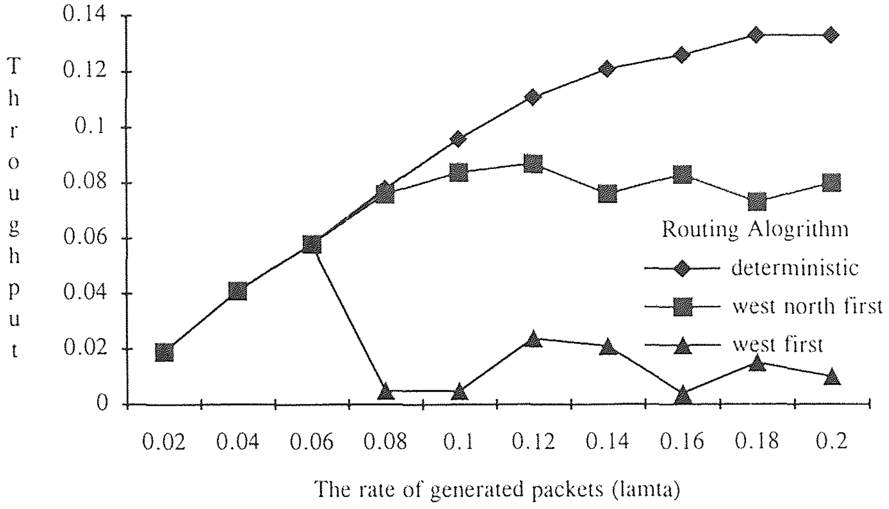
Mesh 4x4	**λ (the number of packets generated per time unit)**									
Packet Len.16	0.2	0.4	0.6	0.8	1.0	1.2	1.4	1.6	1.8	2.0
deterministic	0.196	0.383	0.520	0.624	0.672	0.697	0.710	0.720	0.736	0.735
west north first	0.196	0.388	0.504	0.515	0.495	0.479	0.513	0.496	0.505	0.490
west first	0.197	0.388	0.072	0.103	0.152	0.149	0.047	0.083	0.062	0.100



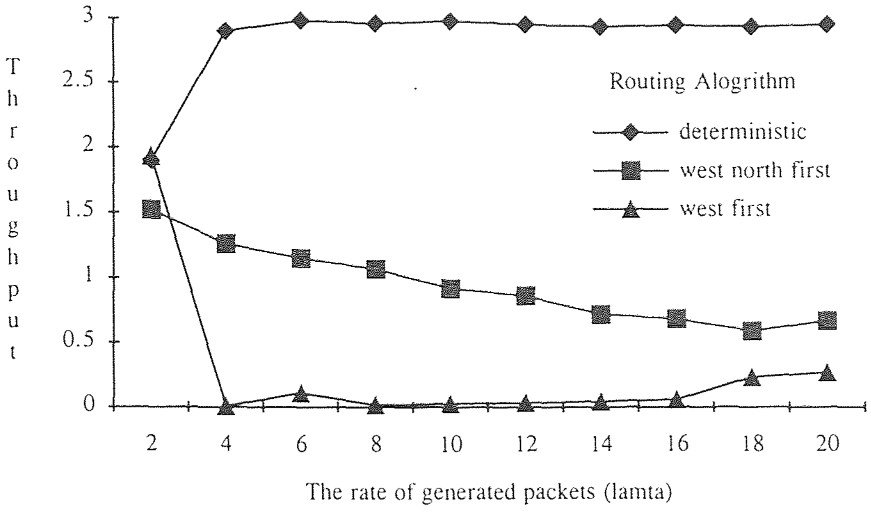
Mesh 8x8	**λ (the number of packets generated per time unit)**									
Packet Len.16	0.2	0.4	0.6	0.8	1.0	1.2	1.4	1.6	1.8	2.0
deterministic	0.196	0.396	0.588	0.753	0.922	1.079	1.193	1.288	1.342	1.376
west north first	0.197	0.398	0.592	0.769	0.870	0.828	0.822	0.810	0.736	0.770
west first	0.197	0.397	0.597	0.780	0.292	0.263	0.011	0.314	0.450	0.027



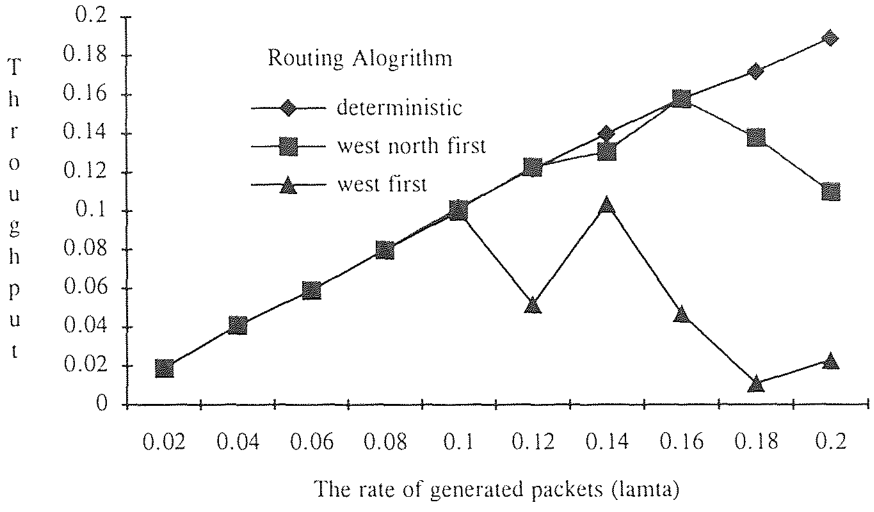
Mesh 8x8	λ (the number of packets generated per time unit)									
Packet Len.128	0.02	0.04	0.06	0.08	0.10	0.12	0.14	0.16	0.18	0.20
deterministic	0.019	0.041	0.058	0.078	0.096	0.111	0.121	0.126	0.133	0.133
west north first	0.019	0.041	0.058	0.076	0.084	0.087	0.076	0.083	0.073	0.080
west first	0.019	0.041	0.058	0.005	0.005	0.024	0.021	0.004	0.015	0.010



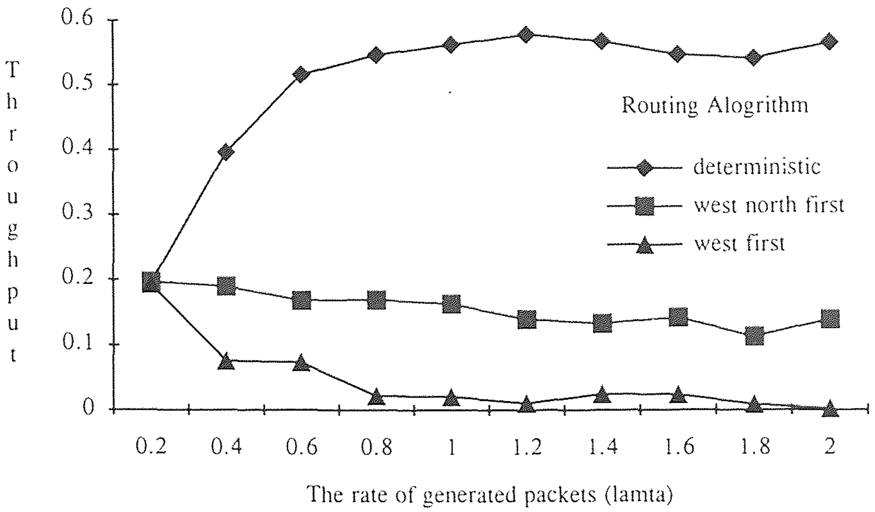
Mesh 16x16	λ (the number of packets generated per time unit)									
Packet Len.16	2	4	6	8	10	12	14	16	18	20
deterministic	1.901	2.901	2.981	2.963	2.982	2.957	2.940	2.953	2.940	2.957
west north first	1.523	1.261	1.151	1.070	0.921	0.866	0.719	0.686	0.592	0.669
west first	1.934	0.008	0.110	0.018	0.028	0.033	0.043	0.058	0.234	0.269



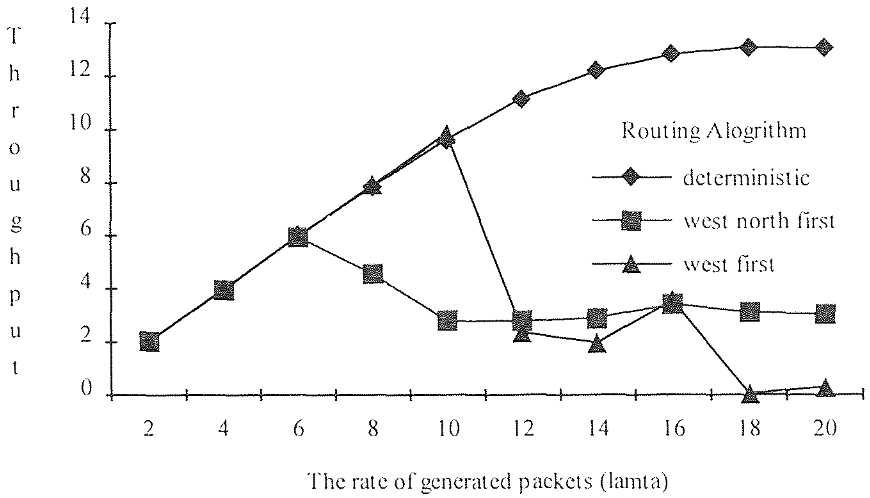
Mesh 16x16	λ (the number of packets generated per time unit)									
Packet Len.128	0.02	0.04	0.06	0.08	0.10	0.12	0.14	0.16	0.18	0.20
deterministic	0.019	0.041	0.059	0.080	0.102	0.122	0.140	0.158	0.172	0.189
west north first	0.019	0.041	0.059	0.080	0.101	0.123	0.131	0.158	0.138	0.110
west first	0.019	0.041	0.059	0.080	0.100	0.052	0.104	0.047	0.011	0.023



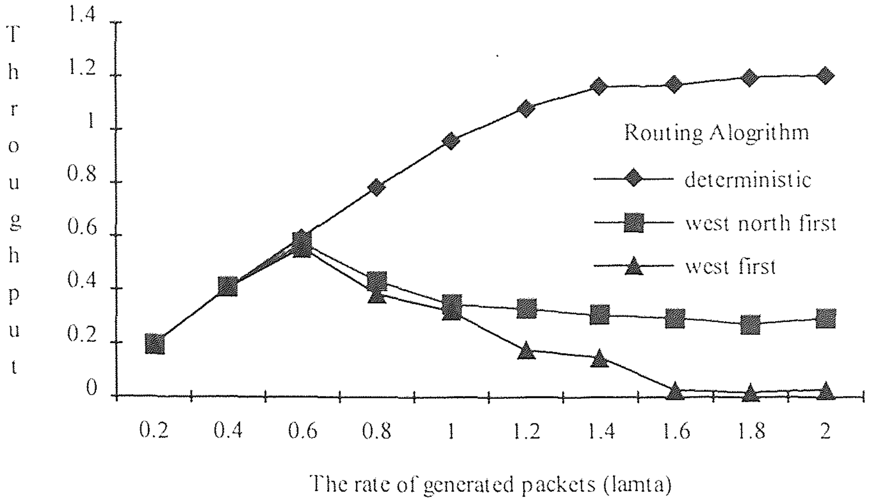
Mesh 32x32	λ (the number of packets generated per time unit)									
Packet Len.128	0.2	0.4	0.6	0.8	1.0	1.2	1.4	1.6	1.8	2.0
deterministic	0.196	0.397	0.517	0.548	0.564	0.580	0.570	0.550	0.544	0.568
west north first	0.197	0.190	0.169	0.170	0.164	0.140	0.134	0.143	0.114	0.140
west first	0.193	0.076	0.074	0.022	0.021	0.010	0.025	0.024	0.009	0.002



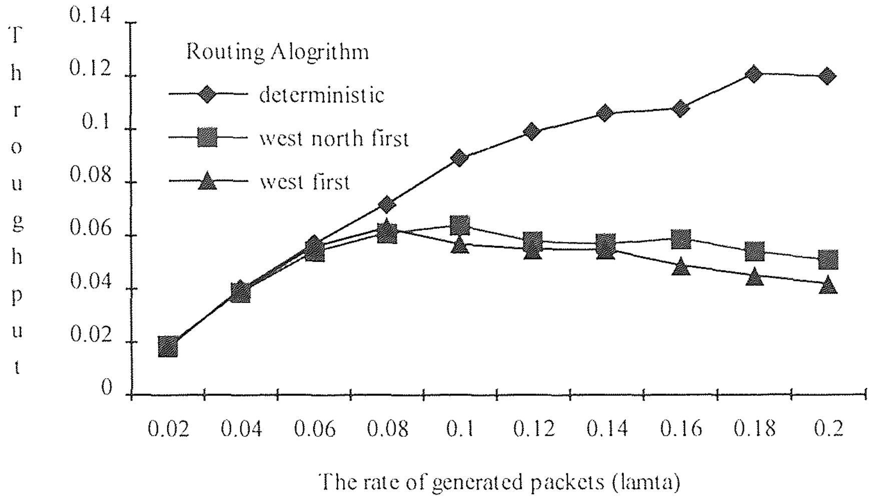
Mesh 64x64	λ (the number of packets generated per time unit)									
Packet Len.16	2	4	6	8	10	12	14	16	18	20
deterministic	2.010	3.969	5.984	7.875	9.648	11.20	12.21	12.83	13.08	13.06
west north first	2.006	3.955	5.975	4.547	2.783	2.773	2.909	3.406	3.096	3.027
west first	2.006	3.962	5.983	7.911	9.879	2.354	1.970	3.557	0.058	0.296



Mesh 64x64	λ (the number of packets generated per time unit)									
Packet Len.128	0.2	0.4	0.6	0.8	1.0	1.2	1.4	1.6	1.8	2.0
deterministic	0.196	0.408	0.598	0.786	0.964	1.088	1.167	1.174	1.204	1.208
west north first	0.195	0.407	0.576	0.434	0.347	0.331	0.307	0.297	0.273	0.295
west first	0.196	0.408	0.557	0.385	0.322	0.178	0.150	0.026	0.017	0.028



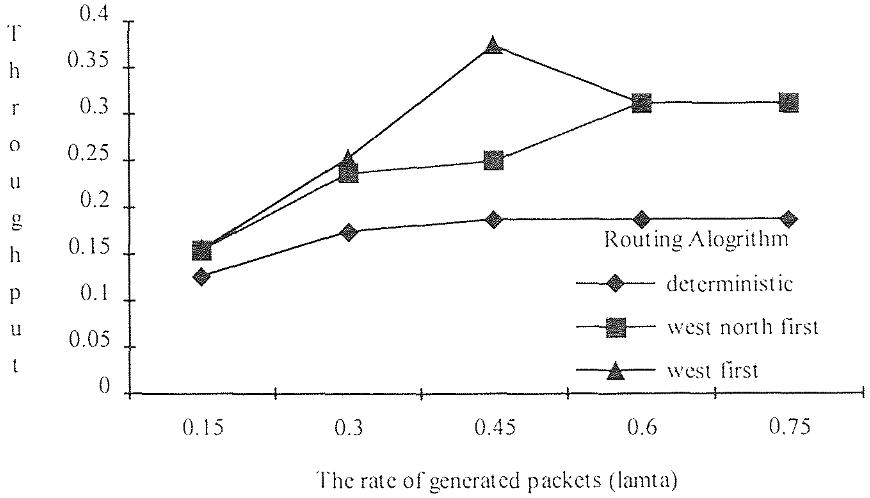
Mesh 64x64	λ (the number of packets generated per time unit)									
Packet Len.1024	0.02	0.04	0.06	0.08	0.1	0.12	0.14	0.16	0.18	0.2
deterministic	0.018	0.040	0.057	0.072	0.089	0.099	0.106	0.108	0.121	0.120
west north first	0.019	0.039	0.054	0.061	0.064	0.058	0.057	0.059	0.054	0.051
west first	0.018	0.040	0.056	0.063	0.057	0.055	0.055	0.049	0.045	0.042



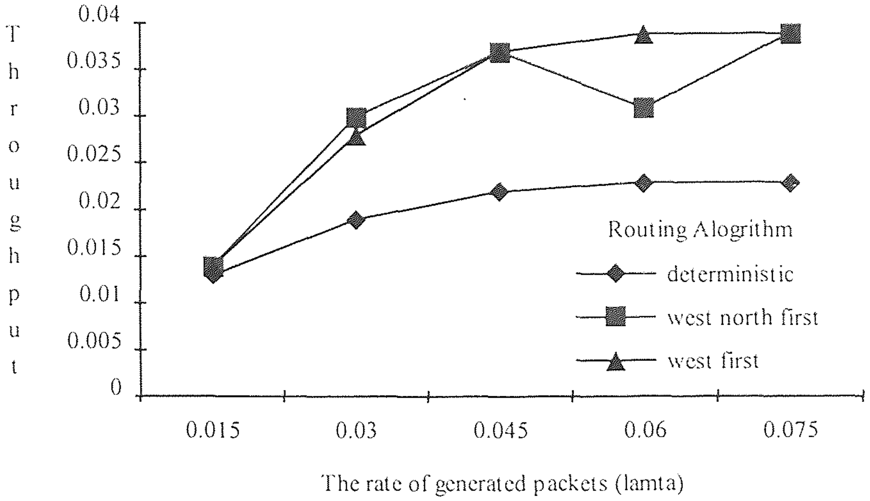
APPENDIX D

COMPARISON OF THROUGHPUT UNDER NON-UNIFORM TRAFFIC MODEL

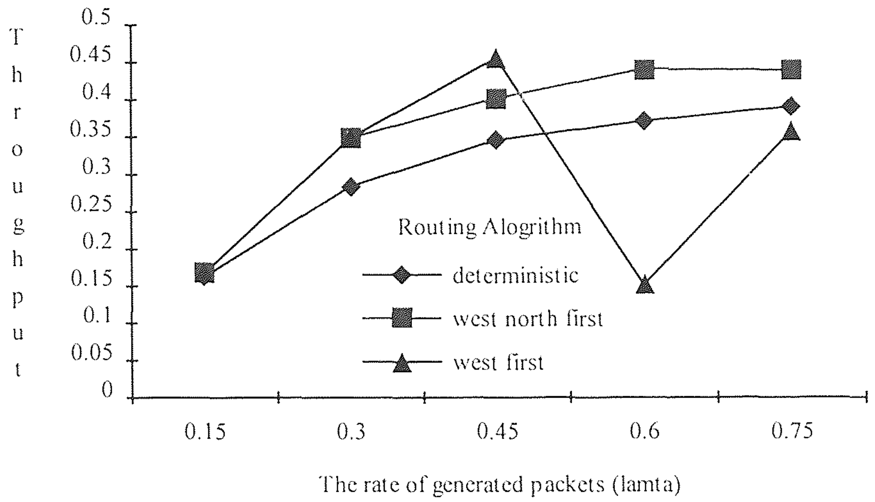
Mesh Size 4 x 4	Lamta (the number of packets generated per time unit)				
Packet Length 16	0.15	0.3	0.45	0.6	0.75
deterministic	0.127	0.175	0.188	0.188	0.188
north west first adaptive	0.155	0.237	0.250	0.313	0.312
west first adaptive	0.156	0.253	0.375	0.313	0.312



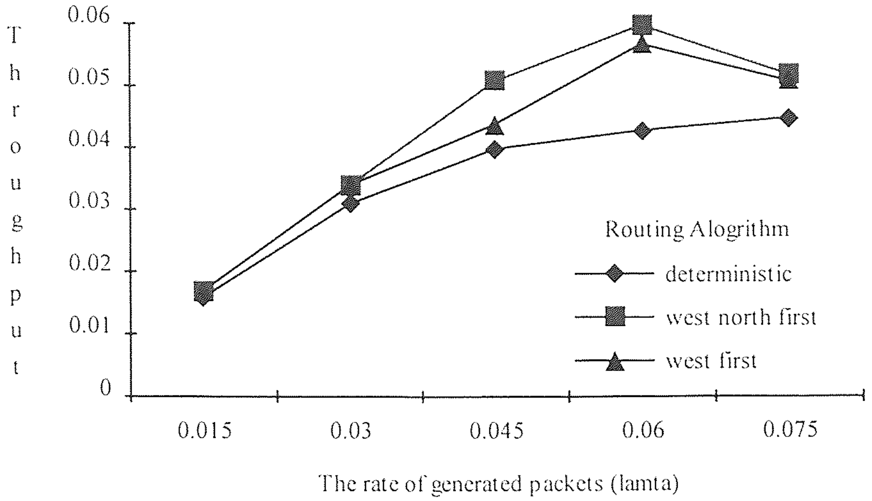
Mesh Size 4 x 4	Lamta (the number of packets generated per time unit)				
Packet Length 128	0.015	0.030	0.045	0.060	0.075
deterministic	0.013	0.019	0.022	0.023	0.023
north west first adaptive	0.014	0.030	0.037	0.031	0.039
west first adaptive	0.014	0.028	0.037	0.039	0.039



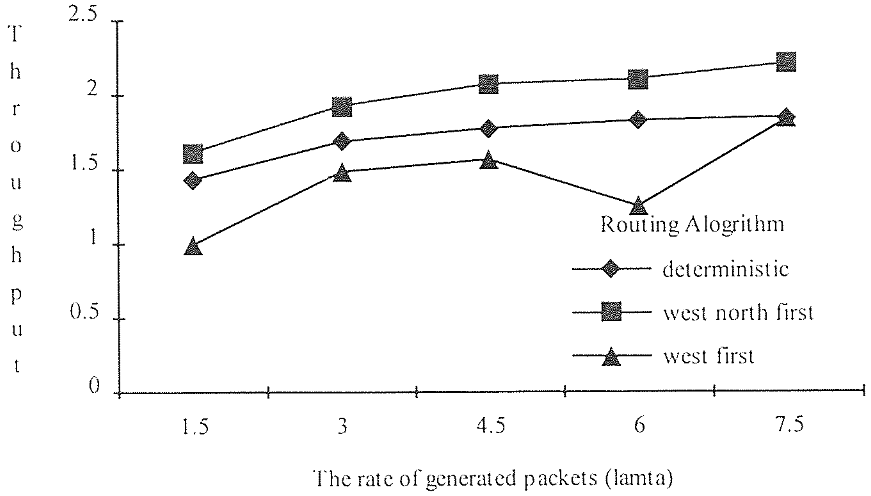
Mesh Size 8 x 8 Packet Length 16	Lamta (the number of packets generated per time unit)				
	0.15	0.3	0.45	0.6	0.75
deterministic	0.163	0.284	0.346	0.371	0.390
north west first adaptive	0.168	0.349	0.402	0.441	0.439
west first adaptive	0.168	0.349	0.456	0.151	0.358



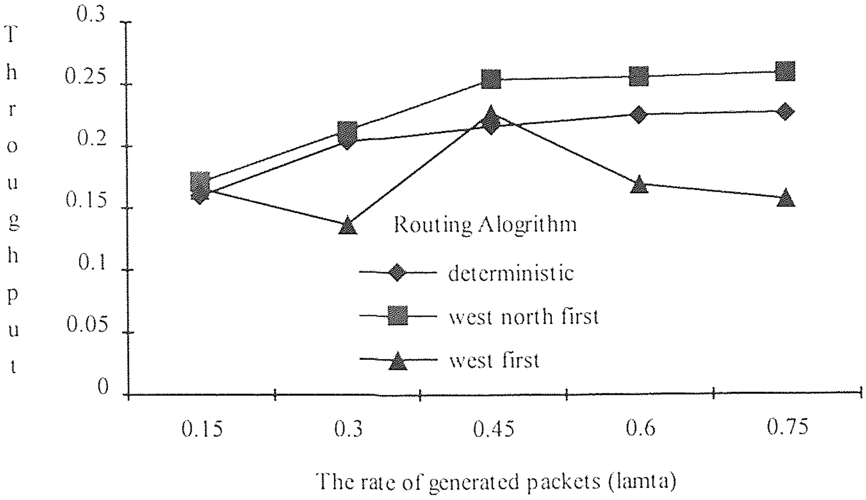
Mesh Size 8 x 8 Packet Length 128	Lamta (the number of packets generated per time unit)				
	0.015	0.030	0.045	0.060	0.075
deterministic	0.016	0.031	0.040	0.043	0.045
north west first adaptive	0.017	0.034	0.051	0.060	0.052
west first adaptive	0.017	0.034	0.044	0.057	0.051



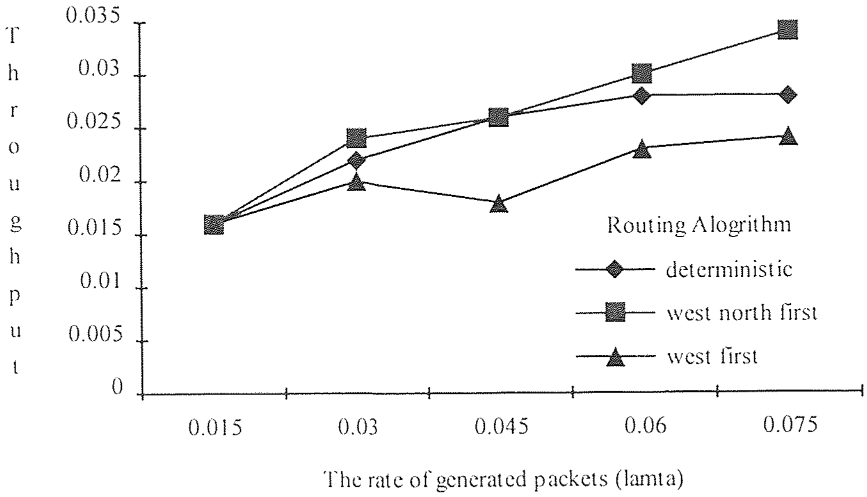
Mesh Size 32 x 32 Packet Length 16	Lamta (the number of packets generated per time unit)				
	1.5	3.0	4.5	6.0	7.5
deterministic	1.434	1.689	1.779	1.828	1.849
north west first adaptive	1.616	1.930	2.077	2.109	2.211
west first adaptive	0.995	1.488	1.568	1.251	1.836



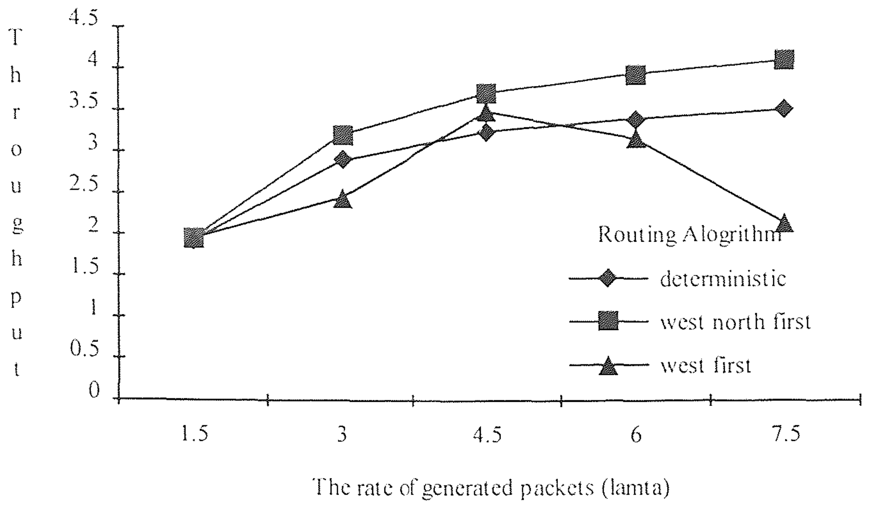
Mesh Size 32 x 32 Packet Length 128	Lamta (the number of packets generated per time unit)				
	0.15	0.3	0.45	0.6	0.75
deterministic	0.160	0.204	0.217	0.226	0.228
north west first adaptive	0.171	0.213	0.255	0.257	0.260
west first adaptive	0.166	0.137	0.228	0.170	0.158



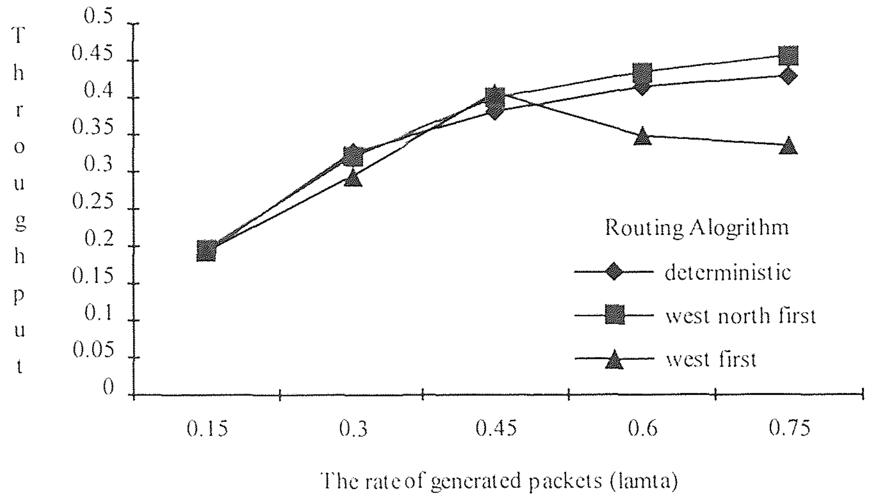
Mesh Size 32 x 32 Packet Length 1024	Lamta (the number of packets generated per time unit)				
	0.015	0.030	0.045	0.060	0.075
deterministic	0.016	0.022	0.026	0.028	0.028
north west first adaptive	0.016	0.024	0.026	0.030	0.034
west first adaptive	0.016	0.020	0.018	0.023	0.024



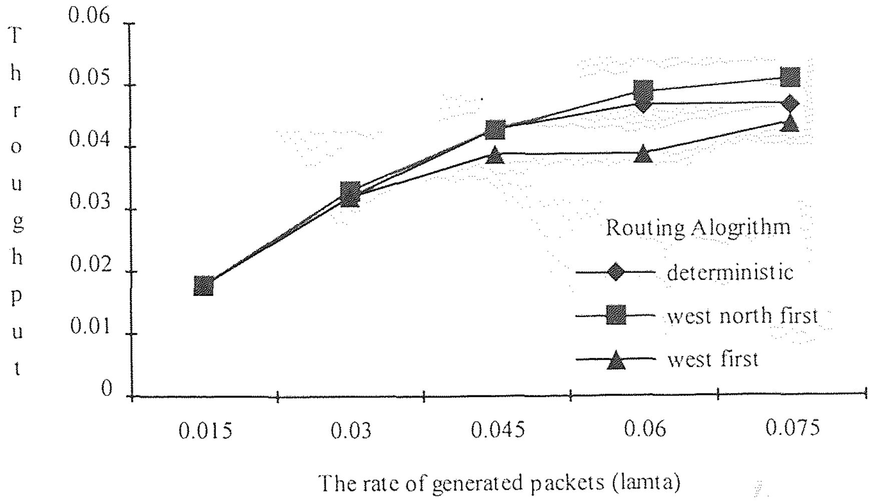
Mesh Size 64 x 64 Packet Length 16	Lamta (the number of packets generated per time unit)				
	1.5	3.0	4.5	6.0	7.5
deterministic	1.935	2.912	3.270	3.425	3.532
north west first adaptive	1.959	3.227	3.736	3.969	4.125
west first adaptive	1.962	2.462	3.509	3.181	2.154



Mesh Size 64 x 64 Packet Length 128	Lamta (the number of packets generated per time unit)				
	0.15	0.30	0.45	0.60	0.75
deterministic	0.193	0.327	0.382	0.415	0.429
north west first adaptive	0.197	0.321	0.401	0.435	0.457
west first adaptive	0.194	0.295	0.408	0.349	0.335



Mesh Size 64 x 64 *Packet Length 1024*	Lamta (the number of packets generated per time unit)				
	0.015	0.030	0.045	0.060	0.075
deterministic	0.018	0.032	0.043	0.047	0.047
north west first adaptive	0.018	0.033	0.043	0.049	0.051
west first adaptive	0.018	0.032	0.039	0.039	0.044


T h r o u g h p u t
0.06
0.05
0.04
0.03
0.02
0.01
0
0.015
0.03
0.045
0.06
0.075
The rate of generated packets (lamta)
Routing Alogrithm
deterministic
west north first
west first

REFERENCES

1. Ni, L.M. and P.K. McKinley. "A Survey of Wormhole Routing Techniques in Direct Networks." *Computer,* February 1993, pp. 62-76.

2. Dally, W.J. and C.L. Seitz. "The Torus Routing Chip." *J. Distributed Computing,* Vol.1, No.3, 1986, pp.187-196.

3. Dally, W.J. *VLSI and Parallel Computation,* San Matro, CA, Morgan Kaufmana Publisher, 1990, pp. 140-222.

4. Dally, W.J. and H. Aoki. "Deadlock-Free Adaptive Routing in Multicomputer Networks Using Virtual Channels." *IEEE Transactions Parallel and Distributed System,* Vol.4, No.4, April 1993, pp. 466-475.

5. Lin, X and L.M. Ni. "Deadlock-Free Multicast Wormhole Routing in Multicomputer Networks." *Proc. 18th Int'l Symp. Computer Architecture,* May 1991, pp. 116-125.

6. Dally, W.J. and C.L. Seitz. "Deadlock-Free Message Routing in Multiprocessor Interconnection Networks." *IEEE Transactions Computer,* Vol.C-36, No.5, May 1987, pp. 547-553.

7. Ngai, J.Y. and C.L. Seitz. "A Framework for Adaptive Routing in Multicomputer Networks." *Computer Architecture News,* Sigarch Publication. Vol.19, No.1, March 1991, pp. 6-14.

8. Dally, W.J. "Virtual-Channel Flow Control." *IEEE Transactions Parallel and Distributed Systems,* Vol.3, No.2, March 1992, pp. 194-205.

9. Konstantinidou, S. and L. Snyder. "Chaos Router: Architecture and Performance" *18th Int'l Symp. Computer Architecture,* May 1991, pp. 212-221.

10. Gunther, K.D. "Prevention of Deadlocks in Packet-Switched Data Transport Systems." *IEEE Transactions on Communication,* Vol. Com-29, No.4, April 1981, pp. 512-524.

11. Law, A.M. and W.D. Kelton. *Simulation Modeling and Analysis,* New York, NY, McGraw-Hill, 1982, pp. 240-278.

12. Glass, C.J. and L.M. Ni. "The Turn Model for Adaptive Routing." *Proc. 19th Int'l Symp. Computer Architecture,* IEEE CS Press. Los Alamitos, Calif., Order No.2940. 1992, pp. 278-287.

13. Duato, Jose. "A New Theory of Deadlock-Free Adaptive Routing in Wormhole Networks." *IEEE Transactions on Parallel and Distributed System*, Vol.4, No.12, December 1993, pp.1320-1331.

14. Kermani, Parviz and L. Kleinrock. "Virtual Cut-Through: A New Computer Communication Switching Technique." *Computer Networks*, Vol.3, 1979, pp.267-286.

15. Seitz, Charles L. *VLSI and Parallel Computation,* San Matro, CA, Morgan Kaufmana Publisher, 1990, pp. 38-42.