



Performance Modeling and Analysis of OTIS Networks

A Thesis Submitted by

Hashem Hashemi Najaf-abadi

for

The Degree of Master of Science

to

The Department of Computer Engineering

Sharif University of Technology

June 2004

Abstract

The most determining factors of the performance of a uniprocessor system are its architecture and the technology in which it is implemented. But when a number of such processors are interconnected to form a multiprocessor system, the performance of the whole system is significantly influenced by the nature in which the processors transfer data between each other; e.g. the technology, topology, switching technique and routing algorithm of the interconnection network. The present study focuses on a specific optoelectronic architecture for the interconnection of processors in a multiprocessor system, the OTIS (stands for Optical Transpose Interconnection System) architecture. In the OTIS architecture, free space optics is used for the interconnection of physically distant processing nodes and electronic interconnections are used for nodes that are closer.

OTIS-based architectures appear to have the potential to be an interesting option for future generations of multiprocessing systems. But before any further observations on the potentials of these architectures can be made, further analysis of their performance is necessary. This is the objective of the work presented in this thesis.

Previous work has been mainly focused on the algorithmic properties of OTIS networks. The aim of this project is to model and evaluate the performance of OTIS networks in sight of realistic parameters such as average message latency and network bandwidth. Of the different classes of OTIS networks, this study has been concentrated on the OTIS-hypercube network. Initially, adaptive and deterministic deadlock-free routing algorithms are suggested for this network. The effect of different design parameters on the performance of this architecture is evaluated through simulation experiments for different network sizes, routing algorithms, numbers of virtual channels, traffic patterns, traffic generation rates and message lengths.

Many analytical models have been proposed for the performance of networks such as the hypercube and mesh networks. Such models can serve as a more appropriate and compact method to display, and refer to, the performance characteristics of the corresponding networks. But no such model has been reported, to our best knowledge, for OTIS-based networks in the literature. An accurate analytical performance model of such networks has been proposed in this work for wormhole switching in the cubical OTIS networks. The proposed model is validated through simulation experiments to serve as verification of its accuracy. We then utilize the proposed model to further study performance issues in OTIS-cubes under different technological implementation constraints.

Content

1.	INTRODUCTION	1
1.1.	INTERCONNECTION NETWORKS.....	1
1.1.1.	Network Topologies.....	2
1.1.2.	Switching Methods	3
1.1.3.	Routing Algorithms.....	4
1.1.4.	Interconnection Technology.....	5
1.2.	OPTIC-BASED MULTICOMPUTER SYSTEMS	5
1.2.1.	Optoelectronic Networks	5
1.2.2.	OTIS Interconnection Networks	6
1.3.	RELATED WORK.....	6
1.4.	OUTLINE OF THE THESIS.....	7
2.	OTIS INTERCONNECTION NETWORKS.....	8
2.1.	SOME ROUTING PROPERTIES OF CUBICAL OTIS NETWORKS	9
2.2.	OTIS-HYPERCUBE NETWORKS	11
2.3.	MESSAGE ROUTING IN THE OTIS-HYPERCUBE	12
2.3.1.	Inter-group Routing.....	13
2.3.2.	Intra-group Routing.....	14
2.3.3.	Deadlock-free Routing.....	14
3.	EMPIRICAL EVALUATION OF OTIS-HYPERCUBES.....	18
3.1.	THE SIMULATOR	18
3.2.	SIMULATION OF DETERMINISTIC ROUTING	19
3.2.1.	The Effect of Channel Cycle Ratio	19
3.2.2.	The Effect of the Number of Virtual Channels.....	19
3.2.3.	Performance-cost Analysis.....	22
3.3.	ADAPTIVE ROUTING.....	25
3.3.1.	The Effect of Channel Cycle Ratio	25
3.3.2.	The Effect of the Number of Virtual Channels.....	25
3.3.3.	Performance/cost Analysis.....	28
3.3.4.	An Adaptive-Deterministic Comparison.....	28
3.4.	THE EFFECT OF TRAFFIC PATTERNS.....	31
3.4.1.	Uniform Traffic.....	32
3.4.2.	Bit-flip Traffic.....	32

3.4.3. Bit-reverse Traffic.....	33
3.4.4. Butterfly Traffic	33
3.4.5. Complement Traffic	36
3.4.6. Perfect-shuffle Traffic.....	36
3.5. SUMMARY AND CONCLUDING REMARKS	39
4. MODELING THE PERFORMANCE OF THE OTIS-HYPERCUBE	40
4.1. MODELING THE PERFORMANCE OF WORMHOLE SWITCHING	40
4.2. MODEL VALIDATION.....	45
4.3. ANALYSIS OF WORMHOLE ROUTING IN THE OTIS-HYPERCUBE	48
4.4. TECHNOLOGICAL CONSTRAINT-BASED PERFORMANCE COMPARISON.....	53
4.4.1. The Effect of Bisection-bandwidth on Performance.....	53
4.4.2. The Effect of Pin-out on Performance	54
5. CONCLUSIONS.....	57
5.1. SUMMARY OF RESULTS	57
5.2. FUTURE WORK.....	58
REFERENCES.....	59

List of Figures

Figure 2.1: A free-space 3x12 optical interconnection system.....	8
Figure 2.2: An OTIS computer with basis of 3-star graph.....	9
Figure 2.3: A 3-dimensional OTIS-hypercube, OTIS- H_3	12
Figure 2.4: The node structure in the OTIS-hypercube with virtual channels.....	13
Figure 2.5: Pseudo-code of different inter-group routing schemes.....	15
Figure 2.6: Adaptive deadlock-free routing algorithm for the OTIS-hypercube.....	16
Figure 3.1: Average message latency in OTIS-hypercubes with dif. cycle ratios	20
Figure 3.2: Average message latency of OTIS-hypercubes for different numbers of virtual channels	21
Figure 3.3: The average latency of equiv. hypercubes and OTIS-hypercubes.....	23
Figure 3.4: The average message latency of a 6-dimensional hypercube and its equivalent 3-dimensional OTIS-hypercube for different values of the network cycle time.....	23
Figure 3.5: Performance to cost ratio of adaptive routing in the hypercube compared to that of the OTIS-hypercube.....	24
Figure 3.6: Average message latency in OTIS-hypercubes for different channel cycle ratios.....	26
Figure 3.7: Average message latency of OTIS-hypercubes for different numbers of virtual channels per physical channel.....	27
Figure 3.8: The average message latency of equivalent hypercubes and OTIS- hypercubes.....	29
Figure 3.9: The average message latency of a 6-dimensional hypercube and its equivalent 3-dimensional OTIS-hypercube for different values of the network cycle time.....	29
Figure 3.10: Performance to cost ratio of the hypercube compared to that of the OTIS-hypercube with a large number of virtual channels.....	30
Figure 3.11: OTIS-hypercube average message latency with different numbers of virtual channels per physical channel for adaptive and deterministic routing.....	31
Figure 3.12: Average message latency of uniform traffic.....	34
Figure 3.13: Average message latency of bit-flip traffic.....	34
Figure 3.14: Average message latency of bit-reverse traffic.....	35
Figure 3.15: Average message latency of butterfly traffic.....	35
Figure 3.16: Average message latency of complement traffic (low gen. rates).....	37
Figure 3.17: Average message latency of complement traffic (high gen. rates).....	37
Figure 3.18: Average message latency of perfect-shuffle traffic (low gen. rates).....	38
Figure 3.19: Average message latency of perfect-shuffle traffic (high gen. rates).....	38
Figure 4.1: Average message latency of 4-dimesnional OTIS-hypercubes for 4 and 6 virtual channels per physical channel.....	46
Figure 4.2: Average message latency of 6-dimesnional OTIS-hypercubes for 4 and 6 virtual channels per physical channel.....	47
Figure 4.3: Average message latency in an OTIS-H5 for different e/o ratios.....	49

Figure 4.4: Average message latency in an OTIS-H ₅ for different message lengths.	49
Figure 4.5: Average message latency in an OTIS-H ₅ for different locality factors.	50
Figure 4.6: The effect of e/o ratio and network size on the network saturation point. ...	51
Figure 4.7: The performance/cost ratio of the equivalent hypercube and OTIS-hypercube networks.	52
Figure 4.8: The average message latency of equivalent Hypercube and OTIS-hypercube networks when the channel cycle time of the hypercube is two times that of the OTIS-hypercube.	55
Figure 4.9: The average message latency of equivalent Hypercube and OTIS-hypercube networks when the channel cycle time of the hypercube is such that the pin-out of both networks are equal.	56

Chapter 1

Introduction

The speed of a program can be defined as the time it takes the program to execute. This can be measured in any increment of time. Speedup is defined as the time it takes a program to execute in serial (with one processor) divided by the time it takes to execute in parallel (with many processors) [1]. Amdahl's law [2] states that the maximum possible speedup obtainable through the parallel processing of a single task is limited to the degree of parallelism of that task. Therefore, had there been no tasks with a high degree of parallelism, large scale parallel systems would be of no use. But that is not the case and tasks currently exist, especially in the field of the simulation of physical phenomenon, that have very high degrees of parallelism. To exploit such high degrees of parallelism there always has been a need for high-performance parallel machines to overcome the physical limitations of increasing the computing power.

A parallel processing system may have a shared-memory or a distributed-memory structure. In the shared-memory structure, processors communicate with each other through a shared memory, while in the distributed-memory structures each processor has local memory and communicates with other processors through direct or indirect communication channels by sending messages. Architectures of the first kind, also known as multiprocessors, are not scalable because memory access time includes the latency of the interconnection network and this latency increases with system size. Distributed-memory architectures, also known as multicomputers, on the other hand have exhibited more scalability and attracted more research and development during the past years. Such systems are organized as an ensemble of nodes, each consisting of a processor, local memory and other supporting devices, comprising a processing element (PE) and a router or switching element (SE) which communicates with other nodes via an *interconnection network*.

1.1. Interconnection Networks

More important than the speed of the processing elements in a parallel processing system is the performance of the interconnection network linking them. Different classifications exist for interconnection networks. A traditional way of classifying them is according to the operating mode and another, according to the network control. The first classification divides networks into synchronous and asynchronous, and the second divides them into networks with centralized, decentralized or distributed control. Most of today's interconnection network architectures are implemented as asynchronous networks with distributed control.

Other classifications are those of [3] in which networks are divided into shared medium, direct and indirect networks. In the first class, the transmission medium is shared by all processors in the network. In the second, point-to-point links directly connect a processor to a subset of other processors in the network (known as its neighbors). Direct interconnection networks have been widely employed by recent machines [4]. But in the last class, nodes are connected to other nodes (or to memory banks in shared-memory architectures) through multiple intermediate stages of switching elements (SE). Indirect interconnection networks have also been employed by many experimental and commercial parallel machines [3]. Examples are the Hitachi SR2201 [5, 6], Cedar [7], CrayX/Y-MP, DEC GIGA-switch, Cenju-3, IBM RP3 [8] and SP2 [9], Thinking machine CM-5 [10] and Meiko CS-2. Multistage interconnection networks (MINs) [7] and crossbars [5] are examples of indirect networks.

Direct networks can exploit traffic locality more effectively. Consequently, most recent multicomputers, including the Intel iPCS [11 - 13], Intel Delta [14], Intel Paragon [15], Cosmic Cube [16], nCube [17, 18], MIT Alewife [19] and J-machine [20, 21], iWarp [22], Stanford DASH [23], Stanford FLASH [24], Cray T3D [25], Cray T3E [26, 27], SGI Origin [28], employ these networks. OTIS interconnection networks, the networks that are the focus of this study, are of the class of direct networks.

Several aspects of an interconnection network are highly influential on the performance of a parallel processing system. These factors consist of the *topology*, the *switching method* and *routing algorithm* used in the network, and the *implementation technology*. Each aspect is described in the following sub-sections.

1.1.1. Network Topologies

The structure of an interconnection networks can be mathematically modeled by a graph. The vertices of this graph represent the processor nodes and the edges represent the links between the processors. The topology of a graph determines the way in which vertices are connected by edges. From the topology of a network, certain properties can easily be determined. The *diameter* is determined as the maximum distance between any two nodes in the network. The number of links connected to a node determines the *degree* of that node. If this number is the same for all nodes in the network, the network is called *regular*.

Numerous topologies have been suggested in the literature and implemented in real parallel processing systems. Of those networks, superior performance belongs to those that are regular and symmetric and have a small diameter. Examples of network topologies proposed (and/or employed) for multicomputer systems include the star [29, 30], cube-connected cycles [31], generalized hypercube [32], Pyramid, and k -ary n -cube networks. Some of the most commonly used direct network topologies are the ring (employed by KSR 1st-level ring [4]), 2-dimensional torus (used in iWARP [33]), 3-dimensional torus (used in the Cray T3D [25] and Cray T3E [26, 27]) and the hypercube (employed in iSPC [11, 13] and nCUBE machines [17, 18]). These all belong to a major family of networks, called strictly orthogonal networks, which have desirable topological properties including ease of implementation, modularity, low diameter and node degree, plus the ability to exploit locality exhibited by many parallel applications [34]. A network topology is *orthogonal* if and only if nodes can be arranged in an orthogonal n -dimensional space, and every link can be arranged in such a way that it produces a displacement in a single dimension. Orthogonal topologies can be classified as strictly orthogonal and weakly orthogonal. In a *strictly orthogonal* topology, every node has at least one link crossing each dimension. In a *weakly orthogonal* topology, some nodes may not have any link in some dimensions.

Strictly Orthogonal Topologies

An interesting property of strictly orthogonal topologies is their simple routing. Therefore, the routing algorithm can be efficiently implemented in hardware. The most popular direct networks are the n -dimensional mesh and n -dimensional torus. Both of these networks are strictly orthogonal. Formally, an n -dimensional mesh has $k_0 \times k_1 \times \dots \times k_{n-2} \times k_{n-1}$ nodes, k_i node along each dimension i , where $k_i \geq 2$ and $0 \leq i \leq n-1$. Each node X is identified by n coordinates, $(x_{n-1}, x_{n-2}, \dots, x_1, x_0)$, where $0 \leq x_i \leq k_i - 1$ for $0 \leq i \leq n-1$. Two nodes, X and Y , are neighbors if and only if $y_i = x_i$ for all i , $0 \leq i \leq n-1$, except one, j , where $y_j = x_j \pm 1$. Thus, each node has

between n to $2n$ neighbors, depending on its location in the network. Therefore, this topology is not regular.

In a bidirectional torus, all nodes have the same number of neighbors. The definition of a torus differs from that of an n -dimensional mesh in that two nodes X and Y are neighbors if and only if $y_i = x_i$ for all i , $0 \leq i \leq n-1$, except one, j , where $y_j = (x_j \pm 1) \bmod k$. When $n=1$, the torus collapses to a ring. The k -ary n -cube is a special case of the torus network in which for all $0 \leq i \leq n-1$, k_i is a fixed at k . The hypercube is a special case of both the n -dimensional mesh and the torus network. When, in the definition of these two networks, $k_i = 2$ for all $0 \leq i \leq n-1$, the resulting network is a hypercube or binary n -cube.

The structure of the k -ary n -cube and hypercube networks are suitable for a variety of applications including matrix computation, image processing and other problems whose task graphs can be embedded naturally into these topologies [35].

1.1.2. Switching Methods

The switching method in a network determines the way in which fragments of messages are transferred between nodes of the network. Several methods have been described in the literature of which the two most important are *packet* (or *store-and-forward*) and *wormhole* switching. In store-and-forward switching, a node will not forward incoming fragments of a message until it has received the entire message. Most first generation multicomputers have employed packet switching. But in more recent multicomputer systems, due to low buffering requirements and good performance, wormhole switching (also known as wormhole routing) has been widely used [4]. In this switching scheme, messages are fragmented into *flits* (the smallest logical unit of data transferable in one or few network cycles) and there is a buffer space, the size of a single flit, associated with each channel entering (or exiting) the node. The first flit of a message, the *header* flit, includes the routing information and is followed by the data flits. The header flit is routed through the network and, as it goes, channels are allocated to the message. The data flits following the header are then transmitted through the allocated channels in pipeline fashion. Once all the data flits have been transmitted through a channel, the channel is de-allocated for other messages to use. If the header cannot be routed in the network due to contention of resources (channels and buffers), the data flits are blocked in situ, keeping all the allocated buffers occupied. Because of the pipelined nature of wormhole routing, it can perform well even in high diameter networks. Many experimental machines, such as iWARP [33], J-Machine [21] and Caltech Mosaic [36], and commercial ones, including Intel paragon [15], Cray T3D [25], Cray T3E[27], CM-5 [10] and nCUBE 2/3 [17,18] use wormhole routing.

Since a message, allocating a number of channels, can be blocked with wormhole routing, this method requires careful deadlock control [37]. A solution to this problem is the use of *virtual channel flow control* [38]. A flow control technique is concerned with resolving multiple message requests for the same channel [39]. A good flow control policy should reduce congestion, be fair and retain low latency. The flow control policy used in a network is very dependant on the switching scheme employed [40]. The use of virtual channels is the most common flow control strategy employed with wormhole routing.

Virtual Channels

The buffers in which the flits of a message are temporarily stored in, when being transmitted through an interconnection network, commonly operate as FIFO (first-in-

first-out) queues. As a result, once a message occupies a buffer of a channel, no other message can access the channel [4]. This chained blocking property of wormhole switching causes the bandwidth of interconnection networks to be limited to a fraction of the total available physical bandwidth (the maximum theoretical bandwidth achievable) [41, 42]. But with several separate buffers associated with each physical channel, such that several messages can use a single channel in a flit-by-flit multiplexed manner, the contention caused by messages being blocked can be ameliorated. In this technique, a physical channel is in fact divided into virtual channels. A virtual channel consists of a buffer capable of holding one or more flits of a message (and associated state information) [38]. Initially introduced in [38] to prevent deadlocks in wormhole networks based on the torus routing chip [43], virtual channels have been shown to also improve network performance and latency by relieving contention [38, 4].

1.1.3. Routing Algorithms

A routing algorithm establishes the paths that messages must follow to reach their destination. The performance of an interconnection network is deeply influenced by certain properties of the routing algorithm used. Among these properties, two are of greater importance, *deadlock and livelock freedom*, and *adaptivity*.

Adaptivity is the ability of a routing algorithm to rout packets through alternative paths in the presence of contention or faulty components. This is opposed to deterministic routing in which a message, originating at a specific source node, is always routed through the same path to reach a specific destination.

Deadlock and livelock freedom is the ability to guarantee that messages will not block or wander across the network forever. The deadlock situation occurs when no message can advance towards its destination because of being blocked by other messages that can not advance towards their destination in a similar manner. The deadlock situation occurs when messages indefinitely wander across the network never reaching their destination.

Livelock may only occur when the routing algorithm in a network is non-minimal. *Minimal* routing algorithms always rout messages through the shortest path to their destination while *non-minimal* routing algorithms do not necessarily do so. Deterministic routing [44] has been used in many practical muticomputer systems using virtual channels to ensure deadlock avoidance. This is achieved by forcing messages to visit the virtual channels in a strict order [4]. This form of routing has the advantage of being simple, but is unable to adapt to conditions such as congestion or failure. *Dimension-order routing* is a typical example of deterministic routing where messages visit network dimensions in a pre-defined order. However, if any channel along the path happens to be heavily loaded, the message will experience a large delay and if any channel along the path is faulty the message will not be delivered at all. This is while other minimal paths may exist through which the message can be routed without excessive delay.

Adaptive routing usually has the ability to provide better performance which is less sensitive to the communication pattern [37] and paths can be chosen according to the degree of channel congestion at the node where the routing decision is being taken. Many adaptive routing algorithms (minimal and non-minimal) have been reported in the literature for torus networks. These algorithms display interesting tradeoffs between their degree of adaptivity and the number of virtual channels needed for deadlock-free operation.

1.1.4. Interconnection Technology

Channel cycle time (the amount of time it takes a unit of data to be transmitted over a single channel) is greatly influential on the performance of an interconnection network. This is while network layout (which is closely related to network topology) and the interconnection technology of a network are the main factors determining the channel cycle time.

The interconnection between processing elements in a multiprocessor system is usually brought about by electronic interconnect (electrical conductance). But as the length of an electrical conductor increases, so does its resistance and capacitance. This results in the adverse low-pass filtering of signals and, inevitably, the lengthening of channel cycle time. In addition, long electrical conductors are subject to EMI distortion.

Another option for the interconnection of processing elements is the use of *optical* interconnection technology. In this technology, photonic rays which are immune to distortion and degradation (over typical distances in interconnection networks) convey the data signals.

1.2. Optic-based Multicomputer Systems

The requirement of parallel processing for high-speed communication has led to much interest in optical networks, as they possess a very high bandwidth, orders of magnitude greater than the bandwidth of copper wire [45]. Due to the small diameter of optical fibers and the option of free space propagation, networks based on this technology also have packaging advantages. Moreover, ground lines are not needed to reference signal levels. Such considerations have led to several suggestions for using optical interconnects in parallel systems [46, 47, 48].

Examples of fully optical networks are the WMCH and DSB(N) networks. Dowd has proposed the wavelength division multiple access channel hypercube (WMCH) [49]. In this network, there are multiple nodes in each dimension which are connected through an optical passive star [50, 51] by *wavelength division multiplexing* (WDM). Thompson has proposed the dilated slotted banyan switching network DSB(N) [52]. This network works in a time multiplexed mode, i.e. *time division multiplexing* (TDM), to provide N different configurations in N consecutive time slots. In other work, a combination of these two have been suggested [53, 54], known as *time and wavelength division multiplexing* (TWDM).

In these fully optical networks, the main objective is to meet cost constraints by optimally utilizing the bandwidth made available by optic interconnect through multiplexing. However, the physical distance between adjacent network nodes are considered to be equal, while this may not actually be the case. In fact, because of the 2-dimensional nature of the VLSI and board layout of multiprocessor systems, the physical distance between adjacent nodes inevitably increases unevenly along different dimensions as the dimensionality of the network increases. Thus, another approach to achieving the cost constraints of an optical interconnection network may be to use optical interconnect only for adjacent nodes that are physically distant. In this manner, the need for complex wavelength or time division multiplexing methods may also be alleviated.

1.2.1. Optoelectronic Networks

It is only when communication distance exceeds a few millimeters that optical interconnect provides speed and power advantages over electronic interconnect [55, 56]. Therefore, in the design of very large multiprocessor systems, it is justified to be in

search of techniques to exploit this property by interconnecting physically close processors using electronic interconnect and to use optical interconnect for pairs of processors that are more distant. Such networks, in which communication channels are brought about by a combination of optical and electronic interconnect, are referred to as optoelectronic networks. Marsden et al. [57], Hendrick et al. [58] and Zane et al. [59] have proposed such a technique in the form of an optoelectronic architecture, the OTIS (Optical Transpose Interconnect System) architecture.

1.2.2. OTIS Interconnection Networks

In order to interconnect physically close processors using electronic interconnect and distant processors with optical interconnect, various combinations of interconnection networks have been proposed. In OTIS computers, optical interconnects are realized via a free space optical interconnect system [57]. In this system, processors are partitioned into groups. Processors within each group are interconnected with each other through electronic interconnect while connections between processors of differing groups are brought about by optical interconnect.

Krishnamoorthy et al. [60] have shown that, in general, when the number of groups is equal to the number of processors within each group, bandwidth and power efficiency are maximized, and system area and volume are minimized. Thus, in an N^2 -processor OTIS computer, processors are partitioned into N groups of N processors each. Besides the electronic connections between the processors in each group, processor i in group j is optically connected to processor j of group i . The OTIS-hypercube and OTIS-mesh are two of the most widely studied instances of the OTIS architecture.

The attempt of the present study is to perform a thorough analysis and modeling of the performance of OTIS-hypercube networks under different traffic loads and technological constraints.

1.3. Related Work

The value of an architecture lies in our ability to use that architecture to effectively solve problems that are of interest. Much attention has been focused on the development of algorithms for OTIS-based computers. Algorithms for OTIS-hypercubes have been developed by Sahni and Wang [61]. Algorithms for OTIS-mesh computers have been studied more extensively by Zane *et al* [59], Sahni and Wang [62, 63, 64, 65], Rajasekeran and Sahni [66] and Osterloh [67]. However, no work has, to our best knowledge, investigated the appropriateness of these systems for general purpose applications using realistic implementation assumptions, i.e. these studies have considered topological and algorithmic issues in OTIS computers and no study has been conducted to evaluate the performance of these systems in sight of important parameters such as message latency and network bandwidth. In addition, no work has focused on the task of modeling the performance of these interconnection networks.

Analytical models are cost-effective and versatile tools for evaluating system performance under different design alternatives. The significant advantage of analytical models over simulation is that they can be used to obtain performance results for large systems and behavior under networking configurations and working conditions which may not be feasible to study using simulation on conventional computers due to extensive computation demands.

1.4. Outline of the Thesis

The main purpose of the work presented here has been to conduct an extensive evaluation of the performance of OTIS networks. We have also aimed at comparing the performance of these networks with that of other networks with comparable size and implementation constraints. Obtaining an analytical model for the performance of these networks and exploring their performance merits using the model has also been one of our objectives.

Deterministic and adaptive deadlock-free routing schemes for the OTIS networks are initially developed. The performance of the network is, under realistic conditions and structural constraints, evaluated through extensive simulation experiments, with different routing algorithms, traffic patterns, message generation rates, network sizes, message lengths and number of virtual channels. An analytical model is then developed for wormhole routing and validated by comparing the results obtained from the model with those obtained through simulation experiments. The characteristics of these models are then compared with those of other networks of the same size and same cost.

The rest of the thesis is organized as follows. Chapter 2 presents a more complete description of OTIS networks, while paying more attention to cubical OTIS networks. Some routing properties of these networks are then presented and schemes for deadlock free inter-group routing are presented.

An in-depth empirical evaluation of the performance of wormhole switching in cubical OTIS networks, based on results obtained from the simulation of these networks for adaptive and deterministic routing, is presented in Chapter 3. In this chapter, the behavior of different traffic patterns in these networks is studied.

In Chapter 4, an analytical model for the performance of adaptive wormhole routing in OTIS-hypercube networks is proposed and validated. Based upon this model, a number of observations on the performance of OTIS-cube networks are made. In This chapter, the effect of the cycle time of optical channels, the number of virtual channels and the effect of different traffic patterns, on the performance of these networks are studied.

Finally, in Chapter 5, this thesis is concluded with a summary of the results obtained and some suggestions for future work in this line.

Chapter 2

OTIS Interconnection Networks

Interconnection networks based on the OTIS architecture utilize conventional electronic interconnect for short distance connections and optical interconnect for distant connections.

The optical connections in this architecture are transmitted through a free space optical interconnection system. In a free space interconnection system, photonic rays are transmitted through free-space, not optic fiber. The connections of a typical free-space optical interconnection system, similar to that used to bring about the optical connections in an OTIS computer, are shown in Figure 2.1. As can be observed, each transmitter has a number and group number and is directed by lenses to a receiver with the transpose of the transmitter number and group numbers. In this specific figure, there are three sets of twelve transceivers on the left and twelve sets of three transceivers on the right.

An N^2 processor OTIS computer is however partitioned into N groups of N processors each. The processors within each group are connected by electronic interconnect in the form of a conventional topology such as a hypercube, star or mesh. Processors of different groups, on the other hand, are connected via a free-space optical interconnection system known as the *optical transpose interconnect system* (OTIS), and OTIS computers are named after this interconnection system. This system is such that processor (i, j) , processor j of group i , is connected to processor (j, i) . The electronic and optical connections of an OTIS computer based on a 3-star graph (OTIS-Star₃) are displayed in Figure 2.2.

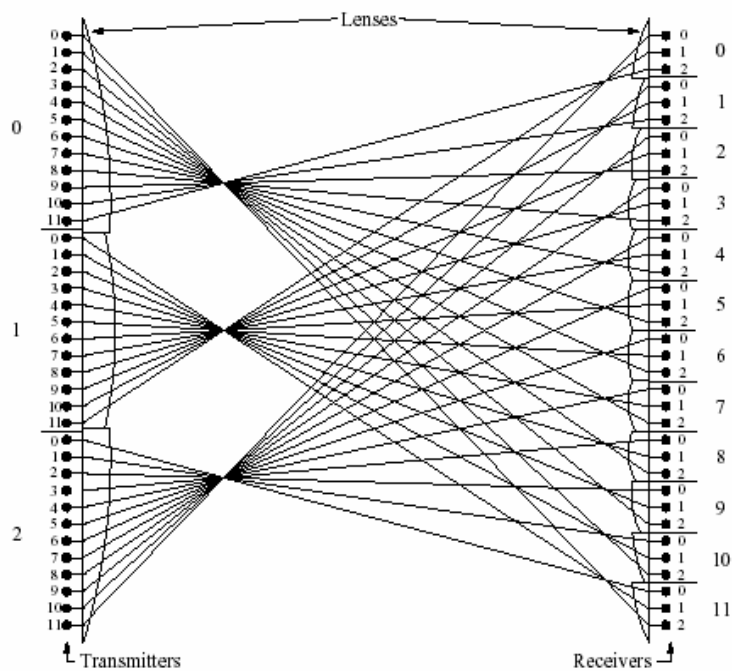


Figure 2.1: A free-space 3x12 optical interconnection system.

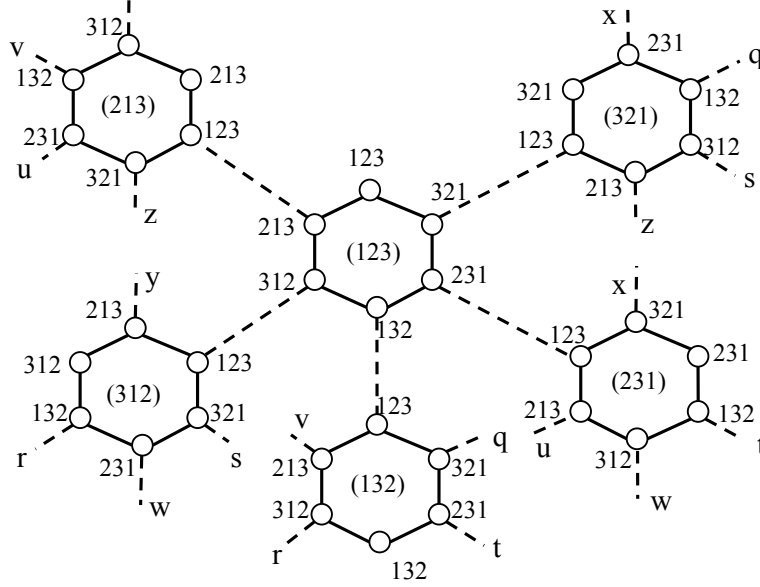


Figure 2.2: An OTIS computer with basis of 3-star graph; optical connections are shown as dashed lines (numbers inside parenthesis are group addresses).

2.1. Some Routing Properties of Cubical OTIS Networks

Definition 1. An n -dimensional shuffle-exchange network, SX_n , has 2^n nodes each addressed with an n -bit vector. In this network a node addressed $x_1x_2 \cdots x_n$ is connected to the node $x_1x_2 \cdots \bar{x}_n$ via an exchange link and to the node $x_nx_1x_2 \cdots x_{n-1}$ via a shuffle link.

Theorem 1. An OTIS- SX_n network can simulate a SX_{2n} with a slowdown factor of at most 6.

Proof. The node address in the OTIS- SX_n is given by $(G, P) = (g_1g_2 \cdots g_n, p_1p_2 \cdots p_n)$, where $g_i, p_i \in \{0,1\}$. Let $(g_1g_2 \cdots g_n, p_1p_2 \cdots p_n) \equiv x_1x_2 \cdots x_{2n}$, i.e. $g_1g_2 \cdots g_n = x_1x_2 \cdots x_n$ and $p_1p_2 \cdots p_n = x_{n+1}x_{n+2} \cdots x_{2n}$. The exchange movement in the simulated network SX_{2n} , i.e. $x_1x_2 \cdots x_{2n} \rightarrow x_1x_2 \cdots \bar{x}_{2n}$, can still be done using one movement via exchange edges over the shuffle-exchange group $x_1x_2 \cdots x_n$ as $(g_1g_2 \cdots g_n, p_1p_2 \cdots p_n) \rightarrow (g_1g_2 \cdots g_n, p_1p_2 \cdots \bar{p}_n)$.

By shuffle movement we mean a movement from processor $x_1x_2 \cdots x_{2n}$ to processor $x_{2n}x_1x_2 \cdots x_{2n-1}$. To do so, we can use the following routing steps:

From $(g_1g_2 \cdots g_n, p_1p_2 \cdots p_n) \equiv x_1x_2 \cdots x_{2n}$

- i. to $(g_1g_2 \cdots g_n, p_1p_2 \cdots p_{n-1}g_n)$ via exchange edges, if $g_n \neq p_n$,
- ii. to $(g_1g_2 \cdots g_n, g_np_1 \cdots p_{n-1})$ via shuffle edges,
- iii. to $(g_np_1 \cdots p_{n-1}, g_1g_2 \cdots g_n)$ via optical links,
- iv. to $(g_np_1 \cdots p_{n-1}, g_1g_2 \cdots g_{n-1}p_n)$ via exchange edges, if $g_n \neq p_n$,
- v. to $(g_np_1 \cdots p_{n-1}, p_ng_1g_2 \cdots g_{n-1})$ via shuffle edges,
- vi. to $(p_ng_1g_2 \cdots g_{n-1}, g_np_1 \cdots p_{n-1}) \equiv x_{2n}x_1x_2 \cdots x_{2n-1}$ via optical links. \square

Definition 2. A (k, n) DeBruijn network, dB_n^k , has k^n nodes, where every node $x_1x_2 \cdots x_n$ is connected to nodes $x_2x_3 \cdots x_n a$, $a \in \{0, 1, \dots, k-1\}$.

Theorem 2. An OTIS- dB_n^k network can simulate a dB_n^k with a slowdown factor of at most 4.

Proof. The node address in the OTIS- dB_n^k is given by $(G, P) = (g_1g_2 \cdots g_n, p_1p_2 \cdots p_n) \equiv x_1x_2 \cdots x_{2n}$, where g_i, p_i , and $x_i \in \{0, 1, \dots, k-1\}$. We require movements of the form $x_1x_2 \cdots x_{2n} \rightarrow x_2 \cdots x_{2n}a$ where $a \in \{0, 1, \dots, k-1\}$. To do so, we may use the following routing steps:

From $(g_1g_2 \cdots g_n, p_1p_2 \cdots p_n) \equiv x_1x_2 \cdots x_{2n}$

- i. to $(p_1p_2 \cdots p_n, g_1g_2 \cdots g_n)$ via optical links,
- ii. to $(p_1p_2 \cdots p_n, g_2 \cdots g_n p_1)$ via electronic edges,
- iii. to $(g_2 \cdots g_n p_1, p_1p_2 \cdots p_n)$ via optical links,
- iv. to $(g_2 \cdots g_n p_1, p_2 \cdots p_n a) \equiv x_2 \cdots x_{2n}a$ via electronic edges. \square

Definition 3. An n -D mesh, $M_{k_1 \times k_2 \times \cdots \times k_n}$, has $N = \prod_{i=1}^n k_i$ nodes arranged in n dimensions, where k_i is called the radix of the i th dimension, $1 \leq i \leq n$. Each node can be identified by an n -digit mixed-radix address (a_1, a_2, \dots, a_n) where $0 \leq a_i \leq k_i - 1$. The i^{th} digit of the address vector, a_i , represents the node position in the i^{th} dimension. Nodes with addresses (a_1, a_2, \dots, a_n) and (b_1, b_2, \dots, b_n) are connected if and only if there exists an i , $(1 \leq i \leq n)$, such that $a_i = (b_i \pm 1)$ and $a_j = b_j$ for, $i \neq j$. Thus, each node is connected to two neighbouring nodes in each dimension, except for nodes at borders.

When wrap-around links are used to connect nodes at borders at different dimensions we have an n -D torus, $T_{k_1 \times k_2 \times \cdots \times k_n}$. The k -ary n -cube is a restricted case of the torus network where the radix of each dimension equals k . A hypercube is also a mesh with dimensions of radix 2. Lets call all these mesh-based networks as n -D Grids, $G_{k_1 \times k_2 \times \cdots \times k_n}$.

Theorem 3. An opto-electronic n -dimensional grid network, OTIS- $G_{k_1 \times k_2 \times \cdots \times k_n}$, can simulate a $2n$ -dimensional grid, $G_{k_1 \times k_2 \times \cdots \times k_n \times k_1 \times k_2 \times \cdots \times k_n}$, with a slowdown factor of at most 3.

Proof. The node address in the OTIS- $G_{k_1 \times k_2 \times \cdots \times k_n}$ is given by $(G, P) = (g_1g_2 \cdots g_n, p_1p_2 \cdots p_n) \equiv x_1x_2 \cdots x_{2n}$, where $0 \leq g_i, p_i, x_i \leq k_i - 1$ for $1 \leq i \leq n$. Movements of the form $x_1x_2 \cdots x_i \cdots x_{2n} \rightarrow x_1x_2 \cdots (x_i \pm 1) \cdots x_{2n}$ are required in target network $M_{k_1 \times k_2 \times \cdots \times k_n \times k_1 \times k_2 \times \cdots \times k_n}$. For $i > n$, it can be realized via electronic edges over the P part of the node address (within a given group) in one communication step. For $i \leq n$, we may use the following communication steps:

From node $x_1x_2 \cdots x_i \cdots x_{2n}$

- i. to node $x_{n+1}x_{n+1} \cdots x_{2n}x_1x_2 \cdots x_i \cdots x_n$ via optical links,
- ii. to node $x_{n+1}x_{n+1} \cdots x_{2n}x_1x_2 \cdots x_i \cdots x_n$,
- iii. to node $x_{n+1}x_{n+1} \cdots x_{2n}x_1x_2 \cdots (x_i \pm 1) \cdots x_n$ via electronic edges
- iv. to node $x_1x_2 \cdots (x_i \pm 1) \cdots x_{2n}$ via optical links. \square

Theorem 4. An n -dimensional OTIS hypercube, OTIS- H_n , can simulate a $T_{k_1 \times k_2 \times \dots \times k_m}$ or an $M_{k_1 \times k_2 \times \dots \times k_m}$ with a slowdown factor of at most 3, where $k_i = 2^{l_i}$, $1 \leq i \leq m$, $l_i \in S_1 \cup S_2$, $Sum(S_1) = Sum(S_2) = n$.

Proof. A node in OTIS- H_n has an address in the form of $(p_1, p_2, \dots, p_n, g_1, g_2, \dots, g_n)$. We can arrange the address bit pattern into m different groups as shown below:

$$\left(\overbrace{g_1 \dots g_{l_1}}^{l_1 \text{ bits}} \overbrace{g_{l_1+1} \dots g_{l_1+l_2}}^{l_2 \text{ bits}} \dots \overbrace{g_i \dots g_{i+l_2}}^{l_k \text{ bits}} \overbrace{p_1 \dots p_{l_{k+1}}}^{l_{k+1} \text{ bits}} \overbrace{p_{l_{k+1}+1} \dots p_{l_{k+1}+l_{k+2}}}^{l_2 \text{ bits}} \dots \overbrace{p_i \dots p_{i+l_j}}^{l_m \text{ bits}} \right)$$

Now, simulation of a $2^{l_1} \times 2^{l_2} \times \dots \times 2^{l_j}$ torus (or a $2^{l_1} \times 2^{l_2} \times \dots \times 2^{l_j}$ mesh) can be realized as follows. For movements in dimensions $k+1, k+2, \dots$, and m , we can use a Grey code node numbering and do any mesh movement with an electronic intra-group movement. For movements in dimensions 1, 2, \dots , and k , we first do a transpose operation using optical links. Then an electronic movement using a Grey code assignment is done. After fixing the group number, an optical movement is realized. Therefore, simulation of the desired torus (or mesh) can be done with a slowdown factor of at most 3. \square

2.2. OTIS-Hypercube Networks

In the OTIS-hypercube parallel computer, there are 2^{2n} processors organized as 2^n groups of 2^n nodes each. The processors in each group form an n dimensional hypercube that employs electrical interconnect and the inter-group interconnections are realized by optics. A partial 3-dimensional OTIS-hypercube is illustrated in Figure 2.3. In this figure, the optical interconnections corresponding to group 0 are shown by dashed lines. Electronic interconnections in each group are shown by solid lines. The address of each group is placed in parentheses above the group. The address of each node in group (001) is displayed near the node, and the nodes in other groups are assigned addresses in the same order.

In general, the full address of a node in any OTIS network consists of two parts: the *node address* and the *group address*. The group addresses of all the nodes in a group are equal to the address of that group.

Node Structure in OTIS Networks

A node, in the n -dimensional OTIS-hypercube, or OTIS- H_n for short, consists of a processing element (PE) and a switching element (SE), as illustrated in Figure 2.4. The PE contains a processor and some local memory. A node is connected, through its SE, to its intra-group neighboring nodes using n input and n output electronic channels. Two electronic channels are used by the PE to inject/eject messages to/from the network. Messages generated by the PE are transferred to the router through the injection channel. At the destination node, messages are transferred to the local PE through the ejection channel. The optical channel is used to connect a node to its transpose node in some other group for inter-group communication. The router contains flit buffers for each incoming channel. A number of flit buffers are associated with each physical channel. The flit buffers associated with each channel may be organized into several

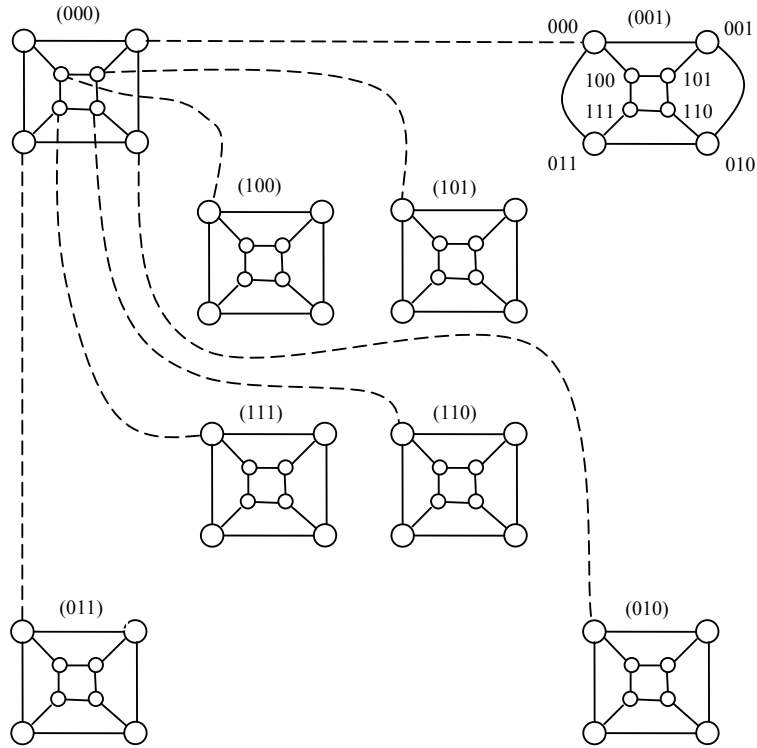


Figure 2.3: A 3-dimensional OTIS-hypercube, $OTIS-H_3$, with the optical connections exiting one of the groups (numbers inside parenthesis are group addresses).

lanes (or virtual channels), and the buffers in each virtual channel can be allocated independently of the buffers in any other virtual channel [38]. The concept of virtual channels has been first introduced in the context of the design of deadlock free routing algorithms, where the physical bandwidth of each channel is multiplexed between a number of messages [37, 38]. However, virtual channels can also reduce network contention. This is while it has been shown that virtual channels are expensive, increasing node delay considerably [68]. So, the number of virtual channels per physical channel should be reasonable. The input and output virtual channels are connected by a crossbar switch that can simultaneously connect multiple input channels to multiple output channels given that there is no contention over the output channels.

2.3. Message Routing in the OTIS-Hypercube

The routing scheme used for inter-group routing and the routing algorithm used for intra-group routing collectively determine the exact routing algorithm in an OTIS-hypercube network.

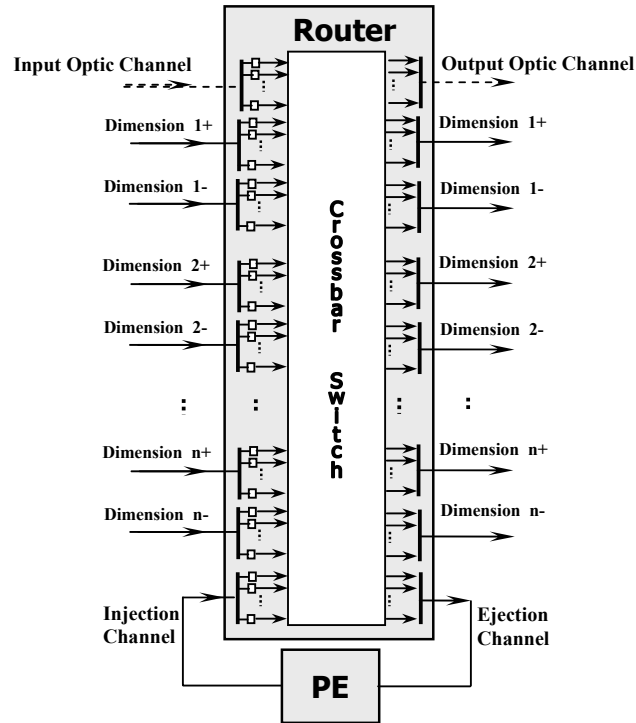


Figure 2.4: The node structure in the OTIS-hypercube with virtual channels.

2.3.1. Inter-group Routing

In what follows, we refer to different *routing schemes* in order to identify only the manner in which a message travels between different sub-graphs (groups) of the network to reach its destination. Two basic routing schemes can be suggested for any source-destination pair of nodes in an OTIS-network. In the first scheme, a message is routed in the local sub-graph, at which it is initially inserted into the network, until it reaches the node that has the same node address as the destination node. From that node, the optical channel is taken into another sub-graph. In this sub-graph, the message is routed until it reaches a node that has the same node address as the sub-graph address of the destination node. Once there, the message takes its final optical hop to the destination node. In the second basic scheme, a message is first routed to a node that has a node address equal to the sub-graph address of the destination. Once there, the optical channel takes the message to the sub-graph of the destination node. The message is then routed to the destination node within this sub-graph.

Of the two former routing schemes, the one that takes a shorter path depends on the full address of the source and destination nodes. When considering the OTIS-hypercube, this can be determined easily. If the number of differing bits of the full address of the source and destination nodes is less than that of the source node and the transpose of address of the destination node, the first routing scheme will result in a shorter path. Otherwise, the second scheme will. However, it should be obvious that in the first routing scheme, once the first optical channel has been taken, the remainder of routing can be conducted by the second scheme. Therefore, if in each intermediate node, a

message is routed according to the basic scheme that takes a shorter path to the destination of the message (without considering the source node), a minimal-path routing scheme, the third scheme, is obtained.

The descriptions of these routing schemes in pseudo-code are for lucidity presented in Figure 2.5. In this code, S_n is the source node-address, S_g is the source group-address, D_n the destination node-address and D_g is the destination group-address. The function **rou1** corresponds to the first inter-group routing scheme, **rou2** to the second scheme and **rou_minimal** to the minimal routing scheme.

2.3.2. Intra-group Routing

Routing within a hypercube may be deterministic, partially adaptive or fully adaptive. With deterministic routing, messages originating at a specific network node and destined to another specific node always take the same path. With fully adaptive routing, any path that brings the message closer to its destination may be taken, and with partially adaptive routing a subset of those paths may be taken. Deterministic routing results in a simple routing algorithm and therefore a simple router implementation [44]. Adaptive routing usually results in better traffic distribution, and thus better performance, at the price of a more complex router. The router complexity for partially adaptive routing is less than that of adaptive, and greater than that of deterministic routing. Any of these routing techniques may be used for intra-group routing in the OTIS-hypercube network.

2.3.3. Deadlock-free Routing

In order for a routing algorithm to be deadlock-free, cyclic buffer dependencies between messages and the virtual channels they allocate, must not occur. In the hypercube network, dimension order routing, a well-known deterministic routing algorithm, is inherently deadlock-free. Partially adaptive routing algorithms based on the turn model [69], such as p-cube routing, are also deadlock free. For fully adaptive routing to be deadlock free, virtual channel utilization must be restricted in a way, such as that suggested in by Duato [70]. But in an OTIS-hypercube, cyclic buffer dependencies between channels may also occur through the optical connections between groups.

To prevent the occurrence of such cyclic buffer dependencies, messages that enter a group through an optical channel must traverse that group through a separate set of virtual channels from those of messages originating in that group. Therefore, we suggest that the virtual channels of each electronic channel be split into two equal sets, i.e. each group be split into two groups, v_1 and v_2 . After being injected into the network, a message traverses the source group through v_1 . But once an optical channel has been taken and the message has entered another group, that group is traversed through v_2 .

Since there is no need for virtual channels when routing deterministically in a hypercube, the minimum number of virtual channels needed for each electronic channel in a deterministically routed OTIS-hypercube is equal to two (one belonging to group v_1 and the other to v_2). The same holds for partially adaptive routing based on the turn model. But when the approach of [70] is used for deadlock-free fully adaptive routing in the hypercube, at least two virtual channels are needed. Thus, the minimum number of virtual channels needed for deadlock-free fully adaptive routing in the OTIS-hypercube is equal to four (two belonging to group v_1 and the other two belonging to v_2).

```

 rout_1( $S_n, S_g, D_n, D_g$ ) {
     traverse current group to node  $D_n$ 
     take the optical channel at node  $D_n$ 
     traverse current group to node  $D_g$ 
     take the optical channel at node  $D_g$ 
}

 rout_2( $S_n, S_g, D_n, D_g$ ) {
     traverse current group to node  $D_g$ 
     take the optical channel at node  $D_g$ 
     traverse current group to node  $D_n$ 
}

 rout_minimal( $S_n, S_g, D_n, D_g$ ) {
     if the Hamming distance of ( $S_n, S_g$ )  and ( $D_n, D_g$ )  < the Hamming distance of ( $S_n, S_g$ )  and ( $D_g, D_n$ )
     then
         rout_1( $S_n, S_g, D_n, D_g$ )
     else
         rout_2( $S_n, S_g, D_n, D_g$ )
    }

```

Figure 2.5: pseudo-code of different inter-group routing schemes.

When messages traverse only one optical channel in their path (the second routing scheme), no restriction is necessary on the utilization of the virtual channels of optical channels. But when messages traverse two optical channels (the first routing scheme), cyclic dependencies may still occur if all the virtual channels of optical channels are allowed to be utilized by messages taking their first optical hop. Thus, for the first routing scheme (and consequently the minimal scheme), one of the virtual channels of all optical channels must be reserved for messages that are traversing a second optical channel (entering their destination node). All the other virtual channels of optical channels can be allowed to be traversed with no restriction.

Since a message that has traversed its second optical channel has definitely entered its destination node, it can not be part of a cyclic buffer dependency. It is for this reason that reserving one of the virtual channels of each optical channel, specifically for such messages, eliminates the possibility of the occurrence of cyclic buffer dependencies through the optical channels. In this manner, the deadlock-free nature of the specific hypercube routing algorithm, used for inter-group routing, will be preserved in the OTIS-hypercube.

Algorithm: Adaptive deadlock-free routing for an n-D OTIS-hypercube

Inputs: Coordinates of current node ($Current_{Sub}, Current_{Node}$) and destination node ($Dest_{Sub}, Dest_{Node}$) and input virtual network

$InNet$.

Output: Selected output physical and virtual channel, $[P_c, V_c]$.

Procedure:

$$Offset_{sub-sub} = Current_{Sub} \oplus Dest_{Sub}$$

$$Offset_{node-node} = Current_{Node} \oplus Dest_{Node}$$

$$Offset_{node-sub} = Current_{Node} \oplus Dest_{Sub}$$

$$Offset_{sub-node} = Current_{Sub} \oplus Dest_{Node}$$

If $InFromOptic$ **then**

$OutNet := 1$;

Else

$OutNet := InNet$;

Endif

If $\|Offset_{sub-sub}\| + \|Offset_{node-node}\| \leq \|Offset_{sub-node}\| + \|Offset_{node-sub}\|$ **then**

If $Offset_{node-node} \neq 0$ **then**

$P_c := \text{SelectOne}(Offset_{node-node})$;

If $P_c = \text{LeastSignificantOne}(Offset_{node-node})$ **then**

$V_c := \text{SelectVirtualChannel}(\text{ChannelsOfNet}(OutNet))$;

Else

$V_c := \text{SelectVirtualChannel}(\text{ChannelsOfNet}(OutNet) - \{0\})$;

Endif

Else

If $Offset_{sub-sub} \neq 0$ **then**

If $OutNet = 1$ **then**

$P_c := \text{Optical}$; $V_c := \text{SelectVirtualChannel}(Any)$;

Else

$P_c := \text{Optical}$;

$V_c := \text{SelectVirtualChannel}(Any - \{Channel_{reserved}\})$;

Endif

Else

$P_c := \text{Internal}$; $V_c := \text{SelectVirtualChannel}(Any)$;

Endif

Endif

Else

If $Offset_{node-sub} \neq 0$ **then**

$P_c := \text{SelectOne}(Offset_{node-sub})$;

If $P_c = \text{LeastSignificantOne}(Offset_{node-sub})$ **then**

$V_c := \text{SelectVirtualChannel}(\text{ChannelsOfNet}(OutNet))$;

Else

$V_c := \text{SelectVirtualChannel}(\text{ChannelsOfNet}(OutNet) - \{0\})$;

Endif

Else

If $OutNet = 1$ **then**

$P_c := \text{Optical}$; $V_c := \text{SelectVirtualChannel}(Any)$;

Else

$P_c := \text{Optical}$;

$V_c := \text{SelectVirtualChannel}(Any - \{Channel_{reserved}\})$;

Endif

Endif

Endif

Figure 2.6: Adaptive deadlock-free wormhole routing algorithm for the OTIS-hypercube.

Figure 2.6 displays the pseudo code of the minimal adaptive routing algorithm. In this code, all messages are considered to be inserted into the 0th virtual network, i.e. $InNet = 0$. Let $||offset||$ return the number of one's in the binary representation of $offset$, and the $SelectOne()$ function adaptively selects a dimension, that has a free virtual channel, corresponding to a one in the binary representation of the input parameter. The $ChannelsOfNet()$ function is considered to return the set of virtual channels of the virtual network determined by the parameter passed to it. The $SelectVirtualChannel()$ function selects one of the virtual channels, passed to it as parameters, that are free. The names of the other functions are descriptive of their operation.

In the next chapter, using the above routing strategies, we evaluate the performance of OTIS-hypercubes with deterministic and adaptive routing under different working conditions.

Chapter 3

Empirical Evaluation of OTIS-Hypercubes

The main performance metrics of an interconnection network consist of the average message latency (average amount of time it takes a message to completely reach its destination) and the network bandwidth (the maximum injection rate at which the average message latency is still bounded). In a thorough analysis, these performance metrics are analyzed under different traffic models. The *traffic model* in an interconnection network is characterized by three basic parameters: *message injection rate*, *message length* and *distribution of message destinations*.

Analysis of this kind can be conducted through results obtained from a real implementation of the network. But a cost effective alternative is to use a simulation of the system. To evaluate the functionality of the OTIS-hypercube network under different conditions, a discrete-event simulator has been developed that mimics the behavior of the routing algorithms, described in Chapter 2, at the flit level. In this chapter, the simulator is briefly described initially and results obtained from the simulations are then presented.

3.1. The Simulator

The simulator has been coded in C++ and consists of a number of different classes which are, at the highest hierarchical level, used to define the *network* class. Specifically, in the constructor of the *network* class, objects of the *node* and *channel* classes are defined. Each *node* has a number of pointers to its input and output channels, and each *channel* has a pointer to the *node* it is an input channel to. The way these pointers are initialized determines the topology of the network to be simulated. Once the *network* has been constructed, messages are injected into the network by *injection* events.

Four different types of *event* classes have been defined, namely, *injection*, *header-routing*, *handshaking* and *channel-switching* events. An event is specified to occur at a particular time and location. At each instance of time, all the events that must be executed at that time are completed. Only then is the time counter incremented and the events of the next time instance executed. This is while the execution of an event may generate another event to be executed at a future time. For instance, the execution of an injection event generates a header-routing event. That event, when executed, causes a channel-switching and a handshaking event.

For each virtual channel, there is a buffer the size of a single flit at the input of the corresponding physical channel to a node. When a flit is transferred from one buffer to the next, a counter associated to the corresponding virtual channel is decremented and a handshaking event, to be executed at the next time unit, is produced. This event notifies the preceding buffer allocated by the message that there is an empty buffer space ahead into which it can transfer its flit. A header-routing event, to be executed at a time determined by the channel cycle time of the network, is also produced. Once routed to a specific channel, a message waits until that channel is switched by a channel-switching event. If there is a free virtual channel available at the time of switching, it is allocated to that message and a counter corresponding to the virtual channel is initialized to the length of the message. The virtual channel on a physical channel is switched in a round-robin manner and every time a virtual channel is switched onto the physical channel, the corresponding counter is decremented if the buffer of that channel has a new flit and that of the next allocated virtual channel is

empty. The execution of all these events put together, results in the simulation of the functionality of the entire interconnection network.

3.2. Simulation of Deterministic Routing

Numerous experiments have been performed for several combinations of network size, message length, number of virtual channels and *channel cycle ratio*, i.e. the ratio of the channel cycle time of optical channels to that of electronic channels.

In each simulation experiment, a minimum of 120,000 messages were delivered, in 12 batches of 10,000 messages, and the average message latency calculated. Statistics gathering was inhibited for messages of the first batch to avoid distortions due to startup transience. The mean message latency is defined as the average amount of time from the generation of a message until the last data flit of that message is consumed at the local PE at the destination node. The network cycle time is defined as the transmission time of a single flit from one router to the next, through an electronic channel. The transmission time of a flit, through an optical channel is however a fraction of the network cycle time. Messages are generated at each node according to a Poisson process with a mean inter-arrival rate of λ_g messages per cycle. All messages have a fixed length of M flits. The destination node of each message has been determined through a uniform random number generator to simulate a uniform traffic pattern. Other traffic patterns are generated according to their mathematical definitions as will be seen.

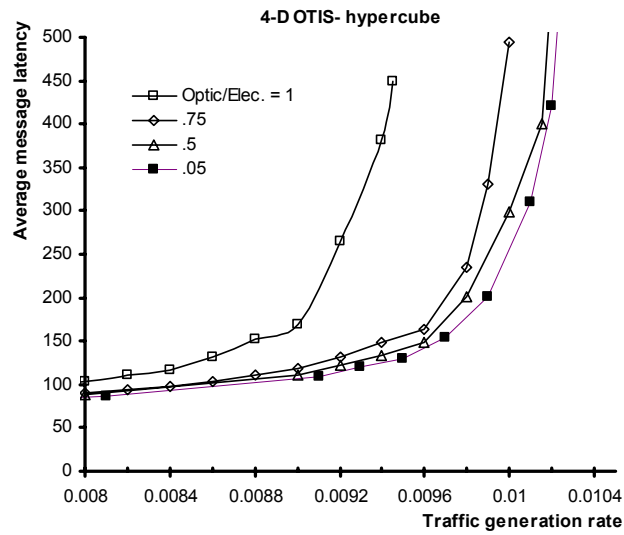
3.2.1. The Effect of Channel Cycle Ratio

Figure 3.1 depicts message latency results for the 4-dimensional and 6-dimensional OTIS-hypercubes, with respectively 256 and 4096 nodes, for different cases of the *channel cycle ratio*, o/e , with the number of virtual channels per physical channel equal to 3. It is evident from these figures that decreasing the channel cycle ratio results in an increase in the generation rate for which saturation occurs. But the effect gradually diminishes and from a point onwards, reducing the ratio any further has no effect on the saturation point. From these figures it is also apparent that for small networks the channel cycle ratio for which this occurs is greater than that for larger networks.

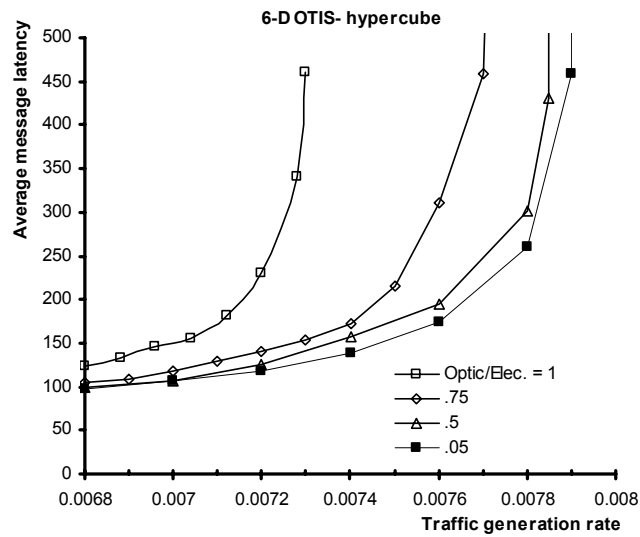
In the results of Figure 3.2, explained in the following sub-section, it is also evident that the maximum increase in the generation rate for which saturation occurs (compared to the case where o/e ratio is equal to one) increases as the number of virtual channels per physical channel is increased.

3.2.2. The Effect of the Number of Virtual Channels

Figure 3.2 shows the average message latency of 4-dimensional and 6-dimensional OTIS-hypercubes, with an o/e ratio of 0.1, for different numbers of virtual channels per physical channel and a message length of $M=32$ flits. It is observed that, increasing the number of virtual channels initially causes a considerable increase in the generation rate for which saturation occurs, but gradually loses its effect. Eventually, the saturation point reaches the bandwidth of the system. At this point, increasing the number of virtual channels, no longer has any effect on the saturation point.

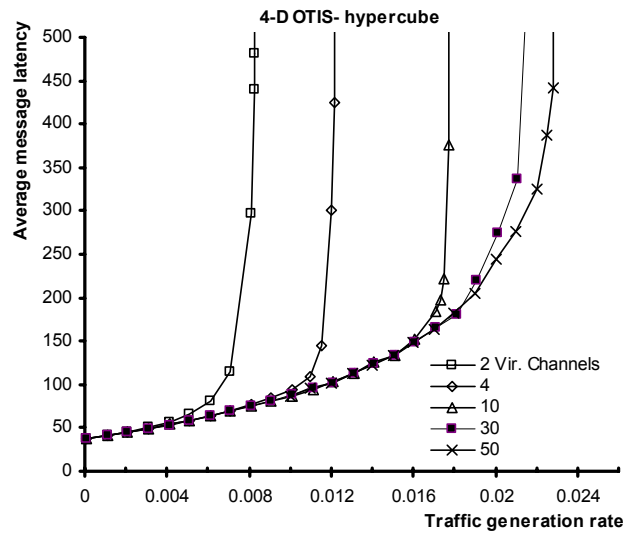


(a)

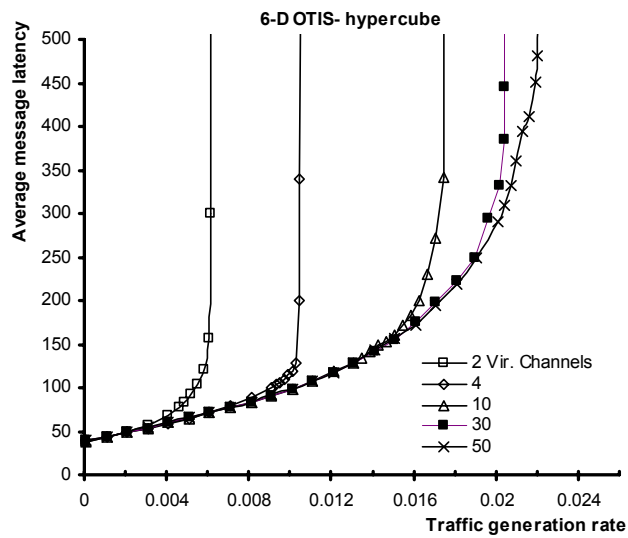


(b)

Figure 3.1: Average message latency in OTIS-hypercubes with 3 virtual channels per physical channel, message length of 32 flits, and different channel cycle ratios; (a) 4-dimensional OTIS-hypercube, and (b) 6-dimensional OTIS-hypercube.



(a)



(b)

Figure 3.2: Average message latency in OTIS-hypercubes with a channel cycle ratio of 0.1 and a message length of 32 flits, for different numbers of virtual channels per physical channel; (a) 4-dimesnional OTIS-hypercube, and ; (b) 6-dimesnional OTIS-hypercube.

It is evident from Figure 3.2 that the generation rate of the saturation point becomes equal to the bandwidth of the corresponding network when the number of virtual channels per physical channel is in the range of 20 to 30. With this number of virtual channels, the network saturates with a generation rate of 0.017 messages per node per cycle. It can therefore be concluded that the bandwidth of a 4-D OTIS-hypercube (with messages 32 flits long) is approximately equal to 0.017. The average message latency in a number of different sized OTIS-hypercubes (with a large number of virtual channels and an o/e ratio of 0.1), are depicted in Figure 3.3. From this figure, it is evident that the bandwidth of the OTIS-hypercube is almost independent of its size. Therefore, in sight of performance, the OTIS-hypercube can be considered to be a well-scalable architecture.

Another network that possesses a very high degree of performance scalability is the well-known hypercube. This can also be observed in the results of Figure 3.3, where the average message latencies in 6-D and 8-D hypercubes are depicted. These results show that when not considering implementation constraints (considering the channel cycle time of the fully-electronic hypercube and OTIS-hypercube to be the same), the OTIS-hypercube possesses less bandwidth than a hypercube with the same number of nodes. This is due to the smaller number of channels in an OTIS-hypercube compared to an equivalent hypercube (with the same number of nodes).

When implementation constraints are brought into account, considerable degradation in the performance of the hypercube becomes apparent as a result of the lengthy transmission time of long wires. But in the OTIS-hypercube, long electronic interconnections do not exist. The maximum channel transmission time of the OTIS-hypercube is therefore a fraction of that of a same sized hypercube. In Figure 3.4, the average message latency in the 3-dimensional OTIS-hypercube (with an o/e ratio of 1.0) is once again compared with that of an equivalent hypercube. This time, however, the network cycle time of the OTIS-hypercube has been scaled to different fractions of the cycle time of the hypercube. Considering the performance scalability of the hypercube and OTIS-hypercube, and the fact that similar results to that of Figure 3.4 have been obtained for different message lengths, it can be concluded that for the bandwidth of an OTIS-hypercube (with $o/e = 1.0$) to be comparable to that of an equivalent hypercube, it is sufficient that the network cycle time of the OTIS-hypercube be approximately half that of the hypercube. This is while decreasing o/e ratio will result in even better performance for the OTIS-hypercube.

3.2.3. Performance-cost Analysis

When the performance to cost ratio of the OTIS-hypercube is compared to that of an equivalent hypercube, it is observed that, for low generation rates, the OTIS-hypercube is superior to the hypercube, when the channel cycle ratio, o/e , is 1.0 (assuming the use of electronic channels for transpose communication). This is shown in Figure 3.5, where the inverse of the average message latency is considered to be representative of performance, and the number of physical channels (entering or exiting the nodes of the network) is considered to be representative of cost. From these results, it can be concluded that compared to a hypercube, the OTIS-hypercube topologically performs better at a lower cost for low generation rates.

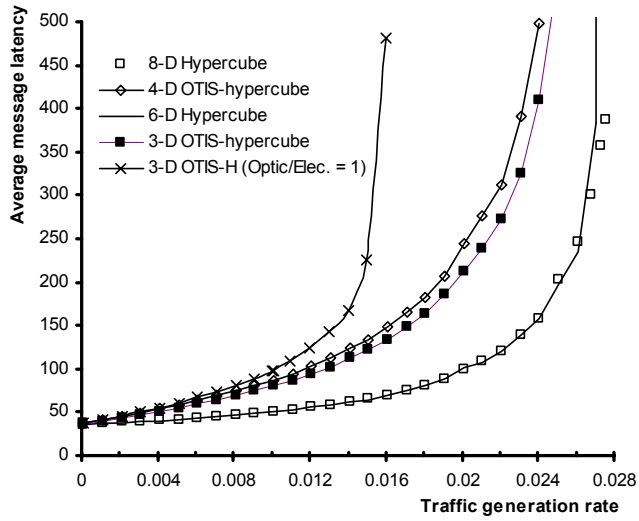


Figure 3.3: The average message latency of equivalent hypercubes and OTIS-hypercubes with 30 virtual channels per physical channel (the default channel cycle ratio is 0.1).

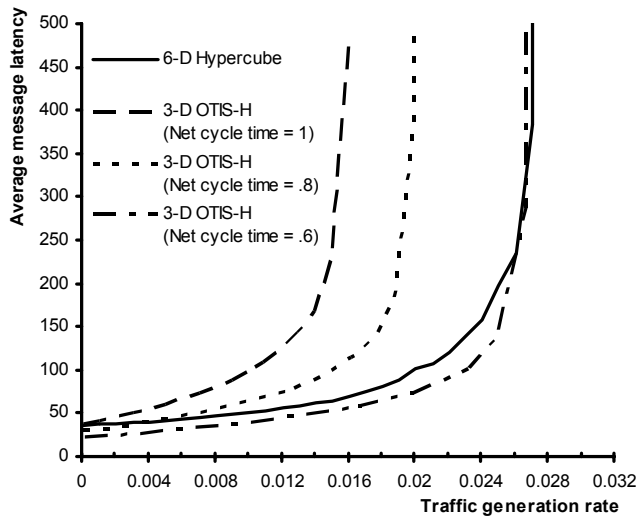
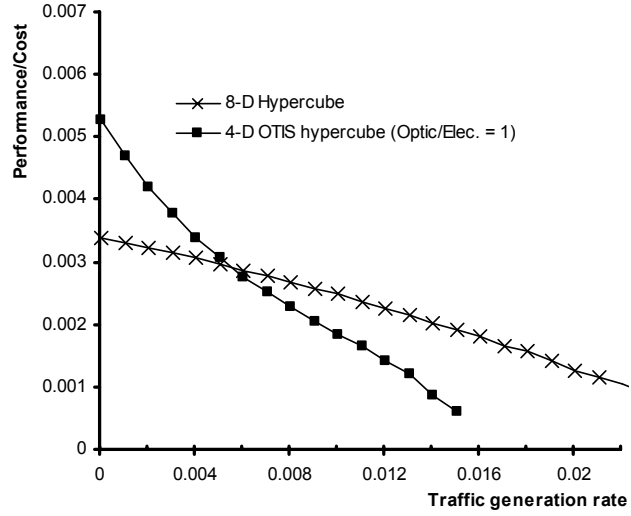
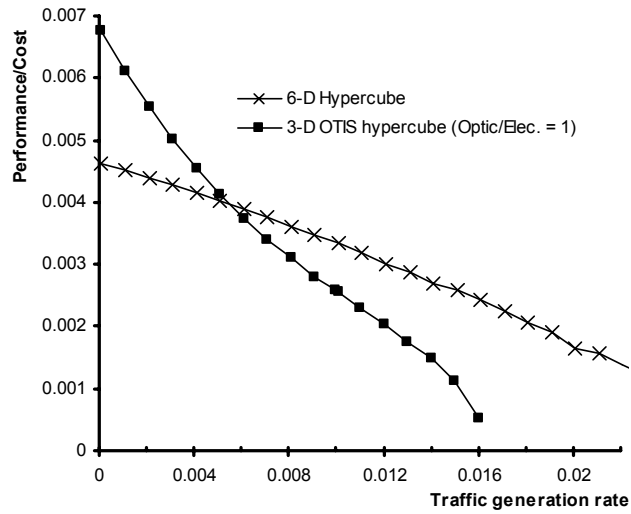


Figure 3.4: The average message latency of a 6-dimensional hypercube and its equivalent 3-dimensional OTIS-hypercube (with 30 virtual channels) for different values of the network cycle time.



(a)



(b)

Figure 3.5: Performance/cost ratio of adaptive routing in the hypercube compared to that of the OTIS-hypercube with a large number of virtual channels and the channel cycle ratio equal to 1.0; (a) 256 nodes, (b) 4096 nodes.

3.3. Adaptive Routing

As in the case of deterministic routing, numerous scenarios were considered and the performance results were obtained and compared. The conclusions drawn are quite similar to those obtained for deterministic routing.

3.3.1. The Effect of Channel Cycle Ratio

Figure 3.6 depicts message latency results for the 4-dimensional and 6-dimensional OTIS-hypercubes for different o/e ratios, and with 4 virtual channels per physical channel (the minimum possible). It is evident from these figures that, similar to deterministic routing, decreasing the channel cycle ratio results in an increase in the generation rate for which saturation occurs. But the effect gradually diminishes and from a point onwards, reducing the ratio any further has no effect on the saturation point.

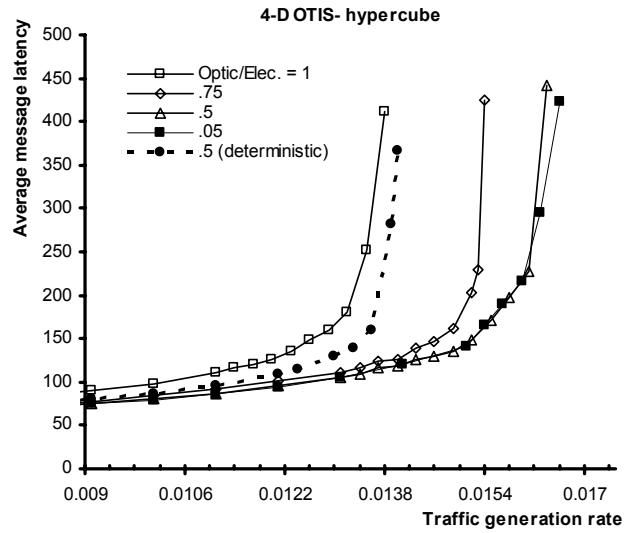
In these figures, results obtained from deterministic routing, with maximum effect of decreasing the cycle ratio, are also displayed to illustrate the fact that the effect of adaptivity is generally greater than the maximum effect of decreasing the channel cycle ratio.

3.3.2. The Effect of the Number of Virtual Channels

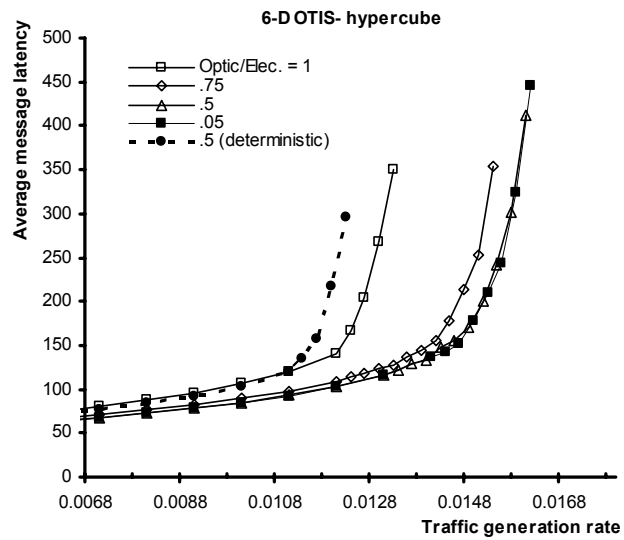
Figure 3.7 shows the average message latency of 4-dimensional and 6-dimensional OTIS-hypercubes, with $o/e = 0.1$, for different numbers of virtual channels per physical channel and the message length $M=32$ flits. It is observed that, increasing the number of virtual channels initially causes a considerable increase in the generation rate for which saturation occurs, but gradually loses its effect. Eventually, the saturation point reaches the bandwidth of the system. At this point, increasing the number of virtual channels, no longer has any effect on the saturation point.

In Figure 3.7 it is observed that the saturation point reaches the bandwidth of the corresponding network when the number of virtual channels per physical channel is equal to 10. With this number of virtual channels, the network saturates at a generation rate of about 0.022 messages per node per cycle. It can therefore be concluded that the bandwidth of a 4-dimensional OTIS-hypercube (with messages of 32 flits) is approximately equal to 0.022. This also shows that *with adaptive routing, fewer virtual channels are needed for the network to attain its maximum saturation point.*

The average message latency of a number of different sized OTIS-hypercubes (with a large number of virtual channels and $o/e = 0.1$), are depicted in Figure 3.8. From this figure, it is evident that the bandwidth of the OTIS-hypercube is almost independent of its size. Therefore, the OTIS-hypercube is a scalable architecture also for adaptive routing.

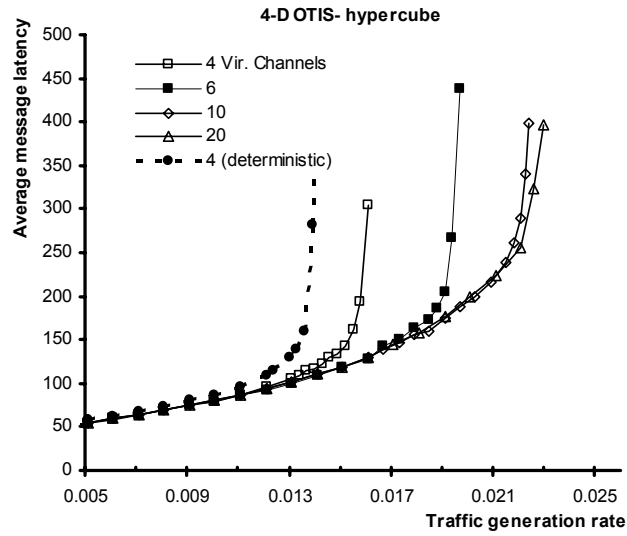


(a)

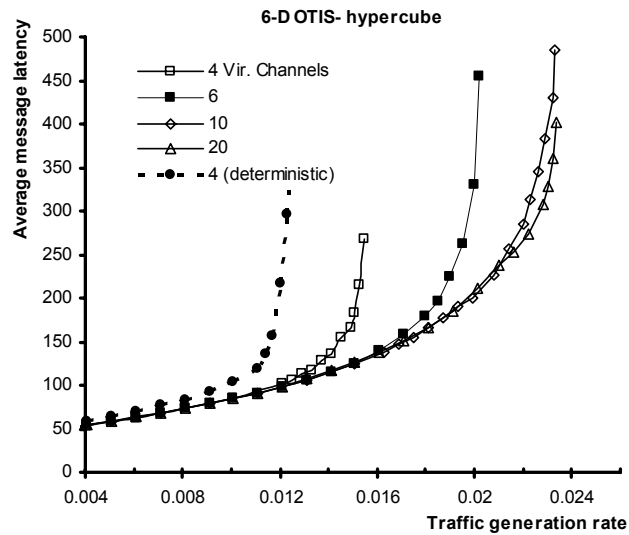


(b)

Figure 3.6: Average message latency in OTIS-hypercubes with 4 virtual channels per physical channel, message length of 32 flits, and different channel cycle ratios; (a) 4-dimensional OTIS-hypercube, and (b) 6-dimensional OTIS-hypercube.



(a)



(b)

Figure 3.7: Average message latency of OTIS-hypercubes with a channel cycle ratio of 0.1 and a message length of 32 flits, for different numbers of virtual channels per physical channel; (a) 4-dimensional OTIS-hypercube, and ; (b) 6-dimensional OTIS-hypercube.

The performance scalability of the hypercube for adaptive routing can also be observed in the results of Figure 3.8, where the average message latencies in 6-dimensional and 8-dimensional hypercubes are depicted. These results show that when not considering implementation constraints (considering the channel cycle time of the fully-electronic hypercube and OTIS-hypercube to be the same), the OTIS-hypercube possesses less bandwidth than a hypercube with the same number of nodes. This is due to the smaller number of channels in an OTIS-hypercube compared to an equivalent hypercube (with the same number of nodes). But the interesting point is that *when the channel cycle ratio is reduced, the maximum bandwidth of the OTIS-hypercube becomes almost equal to the bandwidth of an equivalent hypercube*. In a similar comparison of deterministic routing in hypercube and OTIS-hypercube networks (shown in Figure 3.2), the maximum bandwidth of the OTIS-hypercube was found to be considerably less than that of the equivalent hypercube.

In Figure 3.9, the average message latency of the 3-dimensional OTIS-hypercube (with $o/e = 1.0$) is once again compared with that of an equivalent hypercube and the network cycle time of the OTIS-hypercube has been scaled to different fractions of that in the hypercube. Considering the performance scalability of both networks, as in the case for deterministic routing, it can generally be concluded that for the bandwidth of an OTIS-hypercube (with $o/e = 1.0$) to be comparable to that of an equivalent hypercube, it is sufficient that its network cycle time be approximately half that of the hypercube. This is while decreasing the channel cycle ratio will result in even better performance by the OTIS-hypercube.

3.3.3. Performance/cost Analysis

Similar results obtained from the comparison of the performance/cost ratio of equivalent OTIS-hypercube and hypercube networks with deterministic routing have been observed for adaptive routing. The corresponding results are shown in Figure 3.10, where the inverse of the average message latency is considered to be representative of performance, and the number of physical channels in the network is considered representative of cost. From these results, it can be concluded that also with adaptive routing, the OTIS-hypercube topologically performs better at a lower cost for low generation rates, in comparison to a hypercube. The only apparent difference compared to deterministic routing is that, *for a greater portion of network bandwidth the performance-cost ratio of the OTIS-hypercube is greater than that of the hypercube* (please compare with results of Figure 3.5).

3.3.4. An Adaptive-Deterministic Comparison

As a rule, the bandwidth of a network is independent of the adaptivity of the routing algorithm used in the network. In other words, with a large number of virtual channels, the generation rate for which the network saturates is roughly the same for deterministic and adaptive routing algorithms. This can be observed in the results of Figure 3.11, where the average message latencies of deterministic and adaptive routing in the OTIS-hypercube are compared. The difference between adaptive and deterministic routing is more noticeable when there are a small number of virtual channels per physical channel. It is evident from the results of Figure 3.11 that the network saturates at a higher generation rate with adaptive routing. But since the routing algorithm has no influence on the bandwidth of the network, a straightforward conclusion is that with adaptive routing fewer virtual channels are needed for the saturation point of the network to reach its maximum (the bandwidth of the network).

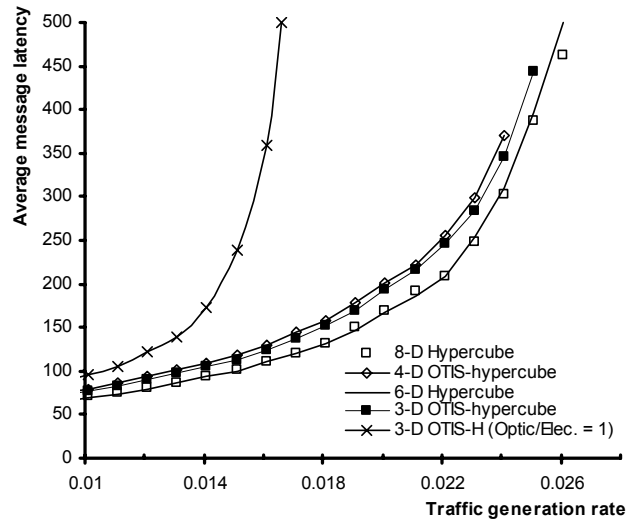


Figure 3.8: The average message latency of equivalent hypercubes and OTIS-hypercubes with 30 virtual channels per physical channel (the default channel cycle ratio is 0.1).

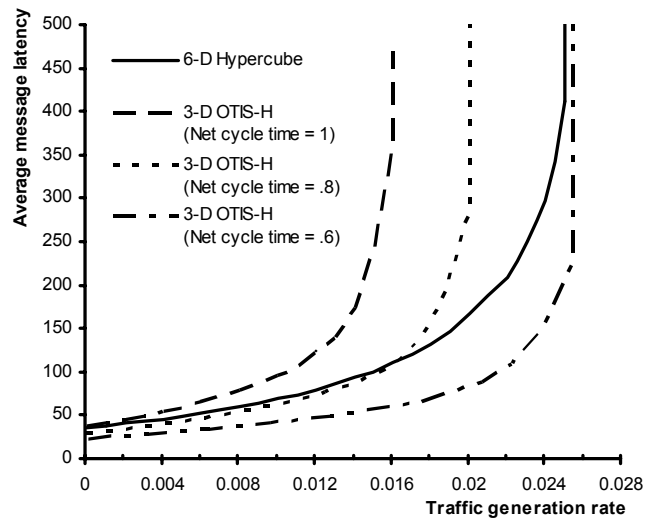
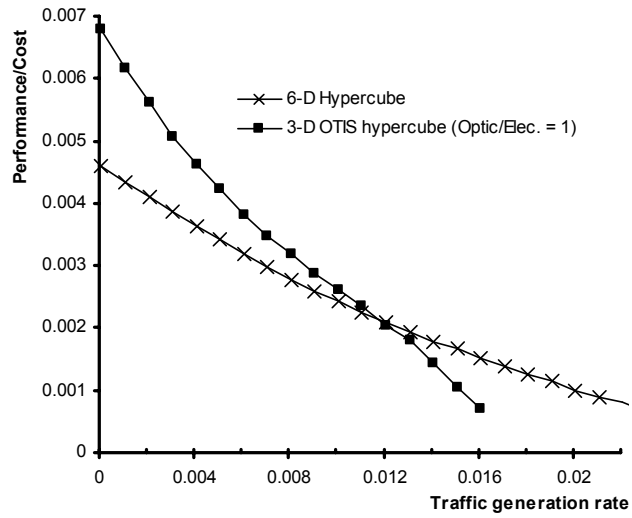
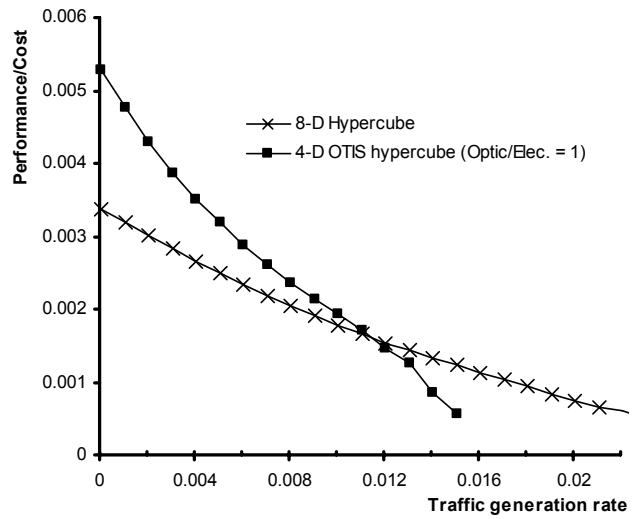


Figure 3.9: The average message latency of a 6-dimensional hypercube and its equivalent 3-dimensional OTIS-hypercube (with 30 virtual channels) for different values of the network cycle time.



(a)



(b)

Figure 3.10: Performance to cost ratio of the hypercube compared to that of the OTIS-hypercube with a large number of virtual channels and the channel cycle ratio equal to 1.0.

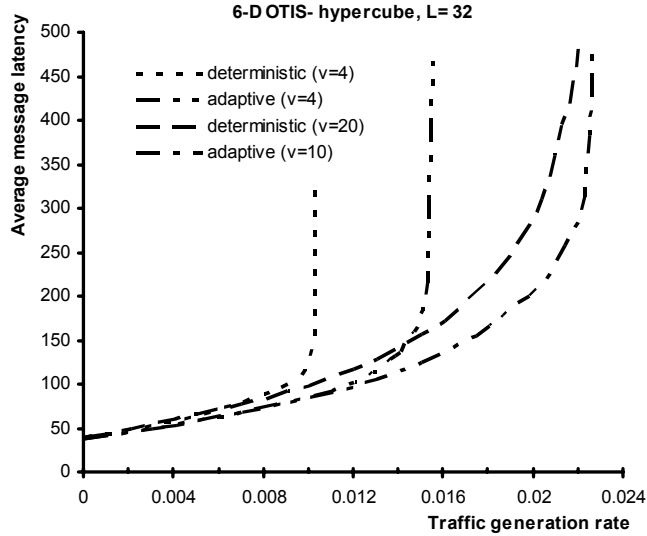


Figure 3.11: OTIS-hypercube average message latency with different numbers of virtual channels per physical channel for adaptive and deterministic routing (the default channel cycle ratio is 0.1).

3.4. The Effect of Traffic Patterns

Different tasks on a multiprocessor system can generate very diverse traffic patterns. The traffic pattern generated by an algorithm is one of the most influential factors on performance. In fact, one way of optimizing performance is by means of optimizing the mapping of tasks to nodes by the operating system.

The destination distribution of traffic (or traffic pattern) in a network determines the relationship between the source and destination nodes addresses of messages. The most frequently used traffic model is the *uniform* model in which a message is destined to different nodes of the network with an equal probability. Extensions to the uniform traffic model have also been proposed, in which messages being destined to a specific sub-section of the network is of a higher probability.

Other traffic models may determine a single destination for each source node. For instance, with the *complement* traffic pattern, messages produced at a source node with address $(a_{n-1} \dots a_1 a_0)$ are destined to the node with bit-complement of that address, i.e. $(\bar{a}_{n-1} \dots \bar{a}_1 \bar{a}_0)$. With *bit-reverse* traffic, the destination address is determined by reversing the order of the bits of the source node address, resulting in $(a_0 a_1 \dots a_{n-1})$. The destination address of a message in the bit-flip traffic pattern is defined as $(\bar{a}_0 \bar{a}_1 \dots \bar{a}_{n-1})$, the source node address after being bit-reversed and complemented. The address of the destination node with *butterfly* traffic is determined by swapping the least

significant and the most significant bits of the source node address, i.e. $(a_0 a_{n-2} \dots a_1 a_{n-1})$. With *perfect-shuffle* traffic, the destination address of a message is the source node address rotated one bit to the left (towards the most significant bit), that is $(a_{n-2} a_{n-3} \dots a_0 a_{n-1})$. The low and high halves of the source address are swapped, resulting in $(a_{n/2-1} \dots a_0 a_{n-1} \dots a_{n/2})$, to determine the destination address of a message in the *matrix-transpose* traffic pattern.

In the following subsections we study the effect that different routing schemes have on the performance of deterministic, partially adaptive and fully adaptive wormhole routing in the OTIS-hypercube under different traffic patterns.

The candidate routing algorithm for partially adaptive routing used for intra-group routing in the OTIS-hypercube is P-cube [65]. Let us briefly describe P-cube routing algorithms in the hypercube. Let $s = s_{n-1} s_{n-2} \dots s_0$ and $d = d_{n-1} d_{n-2} \dots d_0$ be the source and destination nodes respectively, in a hypercube. The set E consists of all the dimension numbers in which s and d differ. The size of E is the hamming distance between s and d . Thus, $i \in E$ if $s_i \neq d_i$. E is divided into two disjoint subsets, E_0 and E_1 , where $i \in E_0$ if $s_i = 0$ and $d_i = 1$, and $j \in E_1$ if $s_j = 1$ and $d_j = 0$. The fundamental concept of P-cube routing is to divide the routing selection into two phases. In the first phase, a packet is routed through the dimensions in E_0 in any order. In the second phase, the packet is routed through the dimensions in E_1 in any order. If E_0 is empty, the packet can be routed through any dimension in E_1 [4].

3.4.1. Uniform Traffic

In an OTIS-hypercube with uniform traffic, regardless of the routing scheme, the performance of adaptive routing is superior to that of deterministic and P-cube routing. This can be observed in the results of Figure 3.12. Furthermore, the minimal routing scheme performs better than the first and second routing schemes. But the interesting point is that deterministic routing saturates at a higher generation rate than that of P-cube routing. The OTIS-hypercube inherits this performance characteristic for uniform traffic from the hypercube network (Glass and Ni have reported such a characteristic for the performance of the hypercube network [69]).

Due to the fact that, when used individually, the first and second schemes do not always rout messages through an optimal path, one would expect the minimal routing scheme to saturate at much higher generation rates. But for adaptive routing, the difference between the performance of minimal routing and that of the first or second routing scheme is less than what may have been predicted. Thus, considering the extra complexity of implementing the minimal routing scheme, this scheme may not be an efficient option for such a system in which traffic is uniform. But, as will be shown in the following sections, for other traffic patterns, minimal routing may even result in performance poorer than that of the first or second schemes.

3.4.2. Bit-flip Traffic

It is observed in the results of Figure 3.13 that, compared to the minimal routing scheme, the network saturates at a much higher generation rate when the second scheme is used, with bit-flip traffic for all three routing algorithms. This is while messages travel a longer average distance with the second scheme. An explanation for this is that, with the first routing scheme, all messages generated by bit-flip traffic in a specific

group, exit that group through the same optical channel (the optical channel exiting a node whose address is the bit-flip of the group address), creating a bottleneck in the network.

It is also apparent from the results that, with the second routing scheme, the generation rate for which the network saturates is greater for P-cube routing than that for deterministic routing. The reason for this is that with bit-flip traffic in an OTIS-hypercube, when the second routing scheme is used, the traffic within each sub-graph is also bit-flip traffic. As shown in [69], a hypercube network with P-cube routing saturates at a higher generation rate than that of deterministic routing for bit-flip traffic. This is while P-cube routing saturates at a lower rate than deterministic routing for uniform traffic.

3.4.3. Bit-reverse Traffic

In the results obtained for bit-reverse traffic, shown in Figure 3.14, it is observed that the generation rate for which the second routing scheme saturates is greater than that for minimal routing. However, the difference between P-cube and deterministic routing is less than that for bit-flip traffic.

The reason why the performance of the minimal routing scheme is so poor with bit-reverse traffic stems from the inefficiency of the first routing scheme. Similar to bit-flip traffic, the first routing scheme (not shown in this figure) causes all messages that are injected into a group to exit that group through a single optical channel (the optical channel exiting a node whose address is the bit-reverse of the group address). But this is not the case for the second routing scheme where messages use the optical channels to exit their source group evenly.

With the second routing scheme used for bit-reverse traffic in an OTIS-hypercube, the traffic within each group is also bit-reverse traffic. This explains the superior performance of P-cube routing over deterministic routing, when the second scheme is used.

3.4.4. Butterfly Traffic

With Butterfly traffic, results of which are depicted in Figure 3.15, the performance of minimal routing is unquestionably better than that of the second routing scheme. Left out from this figure to preserve clarity, are the results of the first routing scheme. These results have, however, shown the performance of the first routing scheme to be very close to that of the minimal routing scheme.

But the interesting point is that, with the minimal scheme, there seems to be hardly any difference between the different routing algorithms. This is due to the fact that with butterfly traffic, the Hamming distance between the source and destination nodes of any message is equal to 2. It is thus, unsurprising that the degree of adaptivity with which those two hops are traversed is almost of no effect on the performance of the network.

Another point is that, since the Hamming distance between the source and destination nodes is so small, the first routing scheme will almost always be selected by the minimal routing scheme. This explains why, for butterfly traffic, the performance of minimal routing is so close to that of the first routing scheme and why these two schemes are superior to the second scheme.

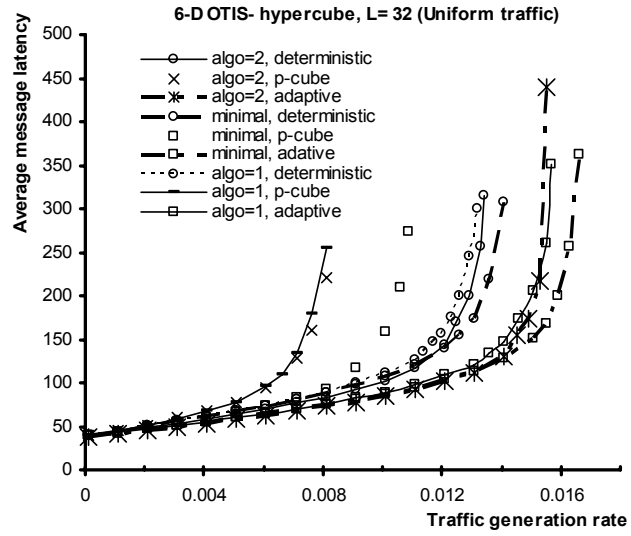


Figure 3.12: Average message latency of uniform traffic.

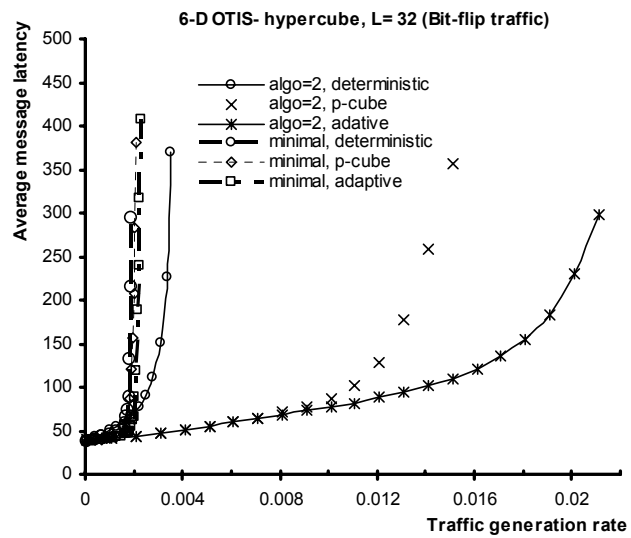


Figure 3.13: Average message latency of bit-flip traffic.

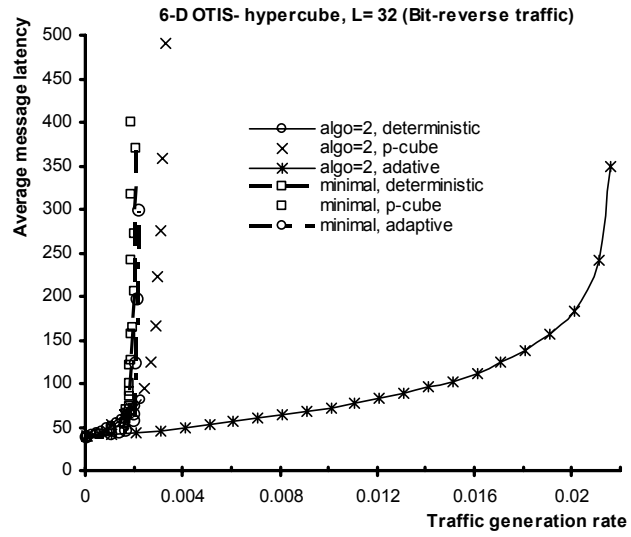


Figure 3.14: Average message latency of bit-reverse traffic.

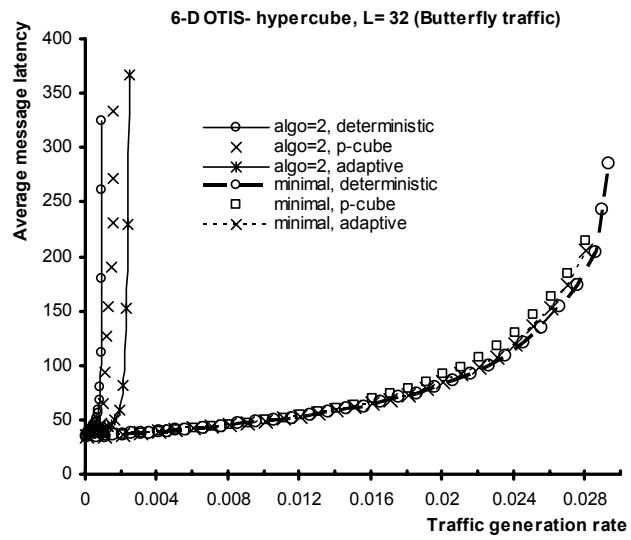


Figure 3.15: Average message latency of butterfly traffic.

3.4.5. Complement Traffic

With complement traffic, hardly any difference can be observed between the performance of minimal routing and that of the second routing scheme. This can be observed in the results of Figure 3.16. But the first routing scheme saturates at a much higher generation rate than the other two schemes.

The first routing scheme results in the path from source to destination of a message to be equal to the diameter of the network. Therefore, the second routing scheme will never rout messages through a longer path than that of the first scheme. Thus, with minimal routing, the second scheme will always be selected. This explains why the minimal and second schemes perform comparably.

With a complement traffic pattern, all messages injected into a specific group are destined to the same destination group (the address of which is complement to that of the source group address). Thus, with the second routing scheme, all messages are routed to the same node in the source group, i.e. they all exit the source group through the same optical channel. As a result, excessive traffic load is imposed on some optical channels while others are left absolutely unused. Even the traffic load on the electronic channels becomes unequally distributed.

But this is not the case for the first routing scheme, by which complement traffic is distributed evenly over the optical channels. This explains why, as depicted in Figure 3.17, the first routing scheme saturates at a much higher generation rate than that of the second scheme, even though messages traverse a longer average distance with the first scheme. In an OTIS-hypercube with complement traffic, there is also complement traffic within each group when the first routing scheme is used.

3.4.6. Perfect-shuffle Traffic

When the second routing scheme is used for perfect-shuffle traffic, all messages injected into a sub-graph exit that sub-graph through one of two optical channels and the other optical channels exiting that sub-graph are left unused. This, as in the case of complement traffic, results in the uneven distribution of traffic on optical channels, and consequently the second routing scheme suffers from early saturation. But unlike complement traffic, the minimal routing of perfect-shuffle traffic does not always utilize the second scheme. Nevertheless, the poor performance of the second routing scheme does affect the performance of the minimal routing scheme. As a result, minimal routing saturates at a generation rate only slightly higher than that of the second scheme. The results of Figure 3.18, obtained for perfect-shuffle traffic, reveal this fact.

In contrast to complement traffic, even the first routing scheme does not distribute perfect-shuffle traffic equally over the optical channels. Since the MSB (most significant bit) of the group address is rotated into the LSB (least significant bit) of the node address, the first routing scheme causes all messages of the same source group to exit that group through the optical channels of nodes with either even or odd addresses, depending on the MSB of the group address. Nonetheless, the first scheme does maintain superior performance over the second scheme. This can be observed in the results of Figure 3.19. The results obtained for adaptive intra-group routing based on the second routing scheme, shown in Figure 3.18, have been included in Figure 3.19 once again to facilitate the comparison of the performance of the different routing schemes in this traffic pattern.

When the first scheme is used for routing perfect-shuffle traffic in an OTIS-hypercube, the traffic pattern within each group becomes somewhat similar to the perfect-shuffle

pattern. The only difference corresponds to the LSB of the destination address. Therefore, considering that results presented in [69] show that with perfect-shuffle traffic in the hypercube, p-cube routing saturates at a lower generation rate than that of deterministic routing, it is acceptable that similar results be obtained for the first routing scheme in the OTIS-hypercube. This fact is also verified in the results of Figure 3.19.

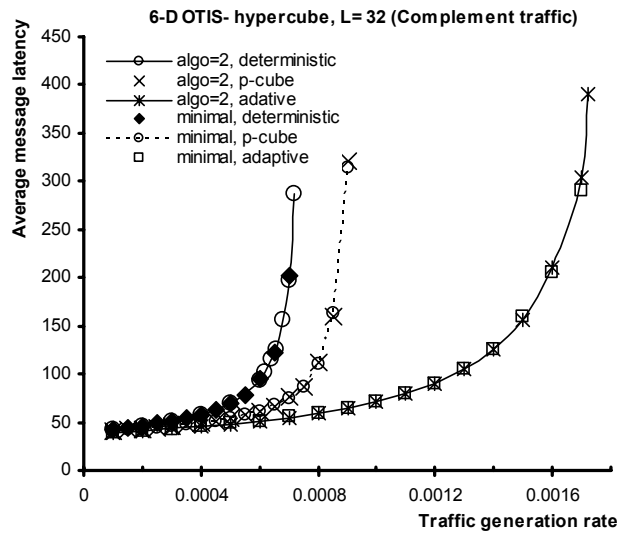


Figure 3.16: Average message latency of complement traffic (low generation rates).

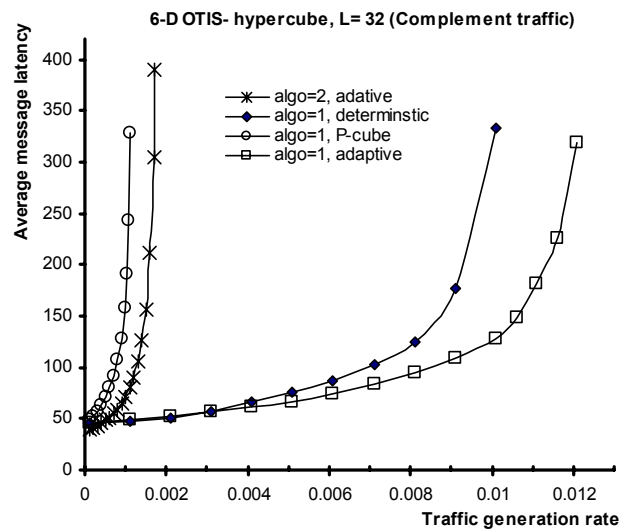


Figure 3.17: Average message latency of complement traffic (high generation rates).

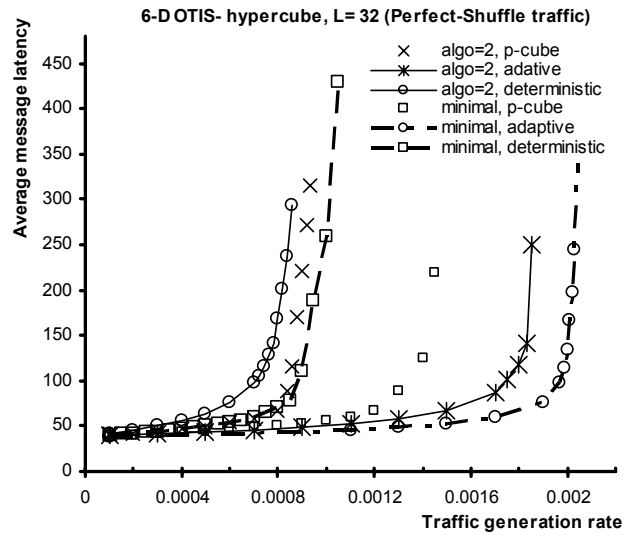


Figure 3.18: Average message latency of perfect-shuffle traffic (low generation rates).

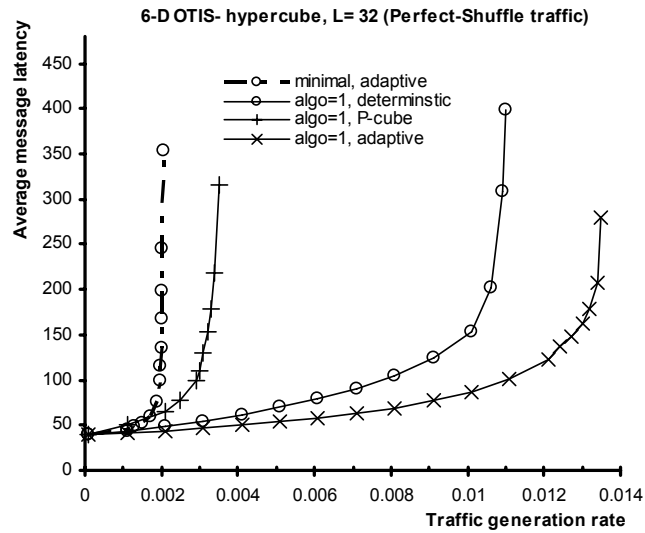


Figure 3.19: Average message latency of perfect-shuffle traffic (high generation rates).

3.5. Summary and Concluding Remarks

A simulation-based evaluation of the performance of cubical OTIS networks has been conducted. From the results obtained it has been observed that decreasing the channel cycle time of optical channels is of limited effect on performance. It is found sufficient that the channel cycle time of an OTIS hypercube be approximately half that of an equivalent hypercube for its saturation point to be, in the worst case, equal to that of the hypercube. The performance/cost ratio of the OTIS-hypercube has been found to be greater than that of the hypercube for low generation rates; while for adaptive routing, this condition has been found to hold for a greater portion of the network bandwidth than deterministic routing.

A simulation-based evaluation of the performance of the OTIS-hypercube network has also been conducted for three different inter-group routing schemes that we have defined (*first*, *second* and *minimal* schemes), three different intra-group routing algorithms (deterministic, fully adaptive and partially adaptive routing) and six different traffic patterns (uniform, complement, bit-reverse, bit-flip, butterfly, perfect-shuffle). We have shown that the method of routing messages between different groups of the network (the inter-group routing scheme) and the intra-group routing algorithm are of considerable influence on the performance of the OTIS-hypercube. However, we observe that (with the exception of uniform traffic) the inter-group routing scheme is generally of greater effect on performance than intra-group routing.

Traffic patterns have also been found to be deeply influential on performance. It is found that with bit-flip and bit-reverse traffic patterns, the network saturates at higher generation rates when the second inter-group routing scheme is used, whereas poor performance is attained with the first routing scheme. The converse holds for butterfly, complement and perfect-shuffle traffic patterns. This is while minimal routing is of superior performance only with uniform traffic.

Consideration of these characteristics can serve as a guideline to the optimal mapping of tasks to nodes by the operating system of multiprocessor systems.

In the next chapter, we further study the performance characteristics of OTIS-hypercubes and derive a mathematical performance model of wormhole routing in the OTIS-hypercube, and validate its prediction accuracy using simulation experiments.

Chapter 4

Modeling the Performance of the OTIS-Hypercube

In this chapter, an analytical performance model for minimal adaptive routing in an OTIS-hypercube that uses wormhole switching is derived and validated. Using the proposed model, the performance of OTIS-cube network architecture is evaluated. The measure that is of interest, as a measure of network performance, is the average message latency. We have initially presented a performance model for packet switching in the OTIS-hypercube [71] (not described here). We have then adjusted the model for it to be applicable for wormhole routing. In this thesis, we present only the description of the wormhole routing model. Validation of the accuracy of the model is presented through comparing the model with simulation results in a number of different network configurations. We conclude this section with a number of observations on the performance of the OTIS-hypercube attained from the analytical model.

4.1. Modeling the Performance of Wormhole Switching

The following assumptions are made when developing the performance model. These assumptions have been widely used in similar modeling studies.

- a) Messages are broken into packets with a fixed length of M flits, which is the unit of switching. The transfer time of a flit between any two routers is assumed to be $T_{c_{electronic}}$ unit cycles over electronic channels and $T_{c_{optic}}$ unit cycles over optical channels.
- b) The destination node of a message is randomly chosen among other network nodes.
- c) Messages are generated at each node according to a Poisson function with an average of λ messages per cycle.
- d) The routing algorithm is assumed to be fully adaptive to achieve maximum performance. With a fully adaptive routing algorithm a message can take any channel which is free and can bring it closer to its destination. To implement the adaptive routing algorithm V virtual channels are used per physical channel which are used according to the routing suggested in Chapter 2. These V virtual channels are divided into two equal sets, v_1 and v_2 . Inside each set, the virtual channels are used to implement fully-adaptive intra-group routing using Duato's methodology [4]. That is, one virtual channel is preserved for messages traversing the lowest of dimensions to be traversed (as would be done with deterministic routing), and the remaining $V/2-1$ virtual channels can be used freely in any order to achieve maximum adaptivity within a group.

The Outline of the Model

The messages generated by each processor may be destined for an intra-group processor with a probability of

$$\xi_1 = \frac{N-1}{N^2-1} = \frac{1}{N+1} \quad (4.1)$$

and for an inter-group processor with a probability of

$$\xi_2 = 1 - \frac{N-1}{N^2-1} = \frac{N}{N+1} \quad (4.2)$$

Therefore, the overall message generation rate for intra-group and inter-group processors in the network are respectively given by

$$\lambda_1 = N^2 \xi_1 \lambda \quad (4.3)$$

$$\lambda_2 = N^2 \xi_2 \lambda \quad (4.4)$$

Now, let us calculate the traffic rate arrived over each electronic and optical link. An intra-group message is a t -hop message with the probability of [72]

$$p_{t_1} = \frac{\binom{n}{t}}{N-1} \quad 1 < t < n \quad (4.5)$$

where $N-1$ accounts for all possible destination nodes in the group except for the source node itself. The average number of hops that an intra-group message may traverse to reach its destination is then given by [79]

$$h_{e1} = \sum_{t=1}^n t \cdot p_{t_1} \quad (4.6)$$

Such messages incorporate into the total traffic rate on network electronic channels by

$$T_{e1} = \lambda_1 h_{e1} \quad (4.7)$$

The average number of electronic hops taken by an inter-group message is then obtained by considering that, depending on which path is shorter, a message may take one of two paths to the destination node. Therefore, this parameter is equal to the aggregate of the length of the shortest paths between all possible source and destination nodes, divided by the number of all such paths.

$$h_{e2} = \frac{\sum_{S \in G} \sum_{D \in G - \{(S, *)\}} d_{S,D}}{N^2 - N} \quad (4.8)$$

where S and D are respectively the source and destination nodes, determined by a group number and a node number within that group, i.e. $S = (S_g, S_p)$ and $D = (D_g, D_p)$; $N^2 - N$ is the total number of different inter-group source-destination pairs, and G is the set of all the nodes of the network. The $d_{S,D}$ function determines the length of the shortest path from node S to node D as

$$d_{S,D} = \min \{ H(S_g, D_g) + H(S_p, D_p), H(S_g, D_p) + H(S_p, D_g) \} \quad (4.9)$$

where $H(x, y)$ returns the Hamming distance between x and y binary patterns.

Therefore, such messages incorporates into the traffic rate on electronic channels as

$$T_{e2} = \lambda_2 h_{e2} \quad (4.10)$$

The total number of electronic channels is nN^2 . Therefore, the traffic rate over each electronic channel will be given by

$$\lambda_{c_{electronic}} = \frac{T_{e1} + T_{e2}}{nN^2} \quad (4.11)$$

In a similar manner, the average number of optical channels traversed can also be determined as

$$h_o = \frac{\sum_{S \in G} \sum_{D \in G - \{(S, *)\}} \varepsilon_{S,D}}{N^2 - N} \quad (4.12)$$

where

$$\varepsilon_{S,D} = \begin{cases} 2, & \text{if } H(S_g, D_g) + H(S_p, D_p) < H(S_g, D_p) + H(S_p, D_g) \\ 1, & \text{otherwise} \end{cases} \quad (4.13)$$

is the number of optical hops to be made by the message generated by node S and destined for node D . Thus, the traffic rate over an optical channel is given by

$$\lambda_{c_{optic}} = \frac{h_o \lambda_2}{N^2 - N} \quad (4.14)$$

where $N^2 - N$ is the number of optical links in the network.

The average message latency for an intra-group message is given by

$$\bar{S}_1 = \sum_{t=1}^n p_{t_1} (S_{t_{electronic}} + (M + t)T_{c_{electronic}}) \quad (4.15)$$

where $S_{t_{electronic}}$ is the time a t -hop intra-group message spends in the network due to blocking.

The blocking time of a t -hop intra-group message is given by

$$S_{t_{electronic}} = \sum_{j=0}^{t-1} B_{t,j_{electronic}} W_{c_{electronic}} \quad (4.16)$$

where $B_{t,j_{electronic}}$ is the probability that a t -hop message is blocked at its j -th hop and $W_{c_{electronic}}$ is the waiting time to acquire a channel when blocked. The probability $B_{t,j_{electronic}}$ is given by

$$B_{t,j_{electronic}} = p_{a_{electronic}}^{t-j-1} p_{a\&d_{electronic}} \quad (4.17)$$

where $p_{a_{electronic}}$ is the probability of all the adaptive virtual channels of an electronic channel (corresponding to the virtual network currently being traversed) being busy. Similarly, $p_{a\&d_{electronic}}$ is the probability that all adaptive and deterministic virtual channels

of an electronic channel (corresponding to the current virtual network) are busy. Therefore, the probability $p_{a\&d_{electronic}}$ is, in fact, equal to the sum of $p_{a\&d_{electronic}}$ and the probability that all the $V/2-1$ adaptive virtual channels are busy. among the different combinations of $V/2-1$ out of V virtual channels being busy, only one corresponds to all the adaptive virtual channels of the physical channel in the current virtual network being busy. Therefore, we can write

$$P_{a\&d_{electronic}} = \frac{P_{V/2-1_{electronic}}}{\binom{V}{V/2-1}} + P_{a\&d_{electronic}} \quad (4.18)$$

in which $P_{v_{electronic}}$ is the probability of v virtual channels of an electronic channel being busy. $p_{a\&d_{electronic}}$ can also be determined in a similar manner. The only difference is that this parameter corresponds to any number of virtual channels greater than $V/2$. Thus, we have

$$P_{a\&d_{electronic}} = \sum_{v=V/2}^V \frac{P_{v_{electronic}} \binom{V/2}{V-v}}{\binom{V}{v}} \quad (4.19)$$

The average inter-group message latency can be determined by averaging the message latency of messages generated at all possible source nodes, destined to all possible destination nodes. Thus

$$\bar{S}_2 = \frac{\sum_{S \in G} \sum_{D \in G - \{S_g\}} T_{S,D}}{N^2 - N} \quad (4.20)$$

where S and D are respectively the source and destination nodes, $S = (S_g, S_p)$ and $D = (D_g, D_p)$, and $T_{S,D}$ represents the latency of the message taking the shortest path from node S to node D , and is given as

$$T_{S,D} = (d_{S,D} + M)T_{c_{electronic}} + S_{i_{electronic}} + S_{j_{electronic}} + \epsilon_{S,D} (T_{c_{optic}} + B_{optic} W_{c_{optic}}) \quad (4.21)$$

In Equation 21, term $(d_{S,D} + M)T_{c_{electronic}}$ accounts for the actual transmission time of a message over $d_{S,D}$ electronic channels, respectively, and $B_{optic} W_{c_{optic}}$ accounts for the time spent for blocking at the optical channel. The probability that the message gets blocked at the optical channel is B_{optic} and the waiting time to acquire the channel, when a message is blocked, is W_{optic} .

Averaging over all intra-group and inter-group message latencies with their proper weights, and adding the average waiting time, $W_{ejection}$, to obtain access to the ejection channel, will result in the average message latency in the network as

$$\bar{S} = \xi_1 \bar{S}_1 + \xi_2 \bar{S}_2 + W_{ejection} \quad (4.22)$$

The probability, $p_{v_{\text{electronic}}}$, can be determined using a Markovian model as used in [38]. The steady-state solutions of the Markovian model yields the probability as [73]

$$p_{v_{\text{electronic}}} = \begin{cases} \frac{1}{\sum_{i=0}^V Q_{i_{\text{electronic}}}}, & \text{if } v = 0 \\ p_{0_{\text{electronic}}} Q_{v_{\text{electronic}}}, & \text{if } 1 \leq v \leq V \end{cases} \quad (4.23)$$

where

$$Q_{v_{\text{electronic}}} = \begin{cases} (\lambda_{c_{\text{electronic}}} \bar{S})^v, & \text{if } 0 \leq v \leq V-1 \\ \frac{(\lambda_{c_{\text{electronic}}} \bar{S})^V}{1 - \lambda_{c_{\text{electronic}}} \bar{S}}, & \text{if } v = V \end{cases} \quad (4.24)$$

In a similar manner, $p_{v_{\text{optical}}}$, the probability of v virtual channels of an optical channel being busy can be determined. The probability that an optical channel is blocked, B_{optic} , is given by

$$B_{\text{optic}} = p_{V_{\text{optical}}} \quad (4.25)$$

To calculate the waiting time to access an electronic channel when blocking occurs, we can model a channel as an $M/G/1$ queue with an arrival rate of $\lambda_{c_{\text{electronic}}}$ and service time of \bar{S} . Approximating the service time variance as $(\bar{S} - T_{c_{\text{electronic}}} M)^2$, as suggested in [74], we can calculate the waiting time to access an electronic channel as

$$W_{c_{\text{electronic}}} = \frac{\lambda_{c_{\text{electronic}}} \bar{S}^2 \left(1 + \frac{(\bar{S} - T_{c_{\text{electronic}}} M)^2}{\bar{S}^2} \right)}{2(1 - \lambda_{c_{\text{electronic}}} \bar{S})} \quad (4.26)$$

In a similar way, we can approximate the waiting time to access an optical channel as

$$W_{c_{\text{optical}}} = \frac{\lambda_{c_{\text{optical}}} \bar{S}^2 \left(1 + \frac{(\bar{S} - T_{c_{\text{optical}}} M)^2}{\bar{S}^2} \right)}{2(1 - \lambda_{c_{\text{optical}}} \bar{S})} \quad (4.27)$$

The waiting time at a source node to inject a message into the network can be calculated assuming an $M/G/1$ queue with arrival rate λ/V and service time \bar{S} as

$$W_{\text{source}} = \frac{\frac{\lambda}{V} \bar{S}^2 \left(1 + \frac{(\bar{S} - T_{c_{\text{electronic}}} M)^2}{\bar{S}^2} \right)}{2(1 - \frac{\lambda}{V} \bar{S})} \quad (4.28)$$

The waiting time at the ejection channel of the destination node can also be calculated in a similar manner assuming an $M/D/1$ queue with arrival rate λ and fixed service time of $T_{c_{\text{electronic}}} M$ (the time required to feed the whole message to the local PE at the destination node) as

$$W_{\text{ejection}} = \frac{\lambda (T_{c_{\text{electronic}}} M)^2}{2(1 - \lambda T_{c_{\text{electronic}}} M)} \quad (4.29)$$

The final value for the average message latency is then given by

$$Latency = (\bar{S} + W_{source}) \bar{V} \quad (4.30)$$

in which, \bar{V} is the average degree of multiplexing of virtual channels that takes place at a given physical channel and can be given as [75]

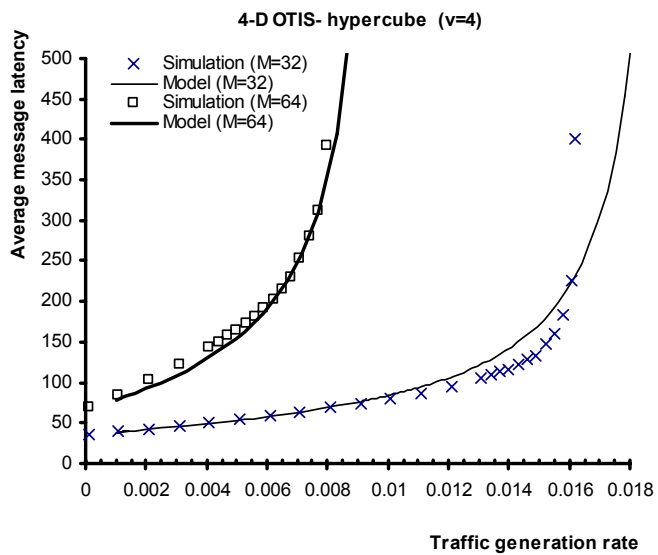
$$\bar{V} = \frac{\sum_{v=1}^V v^2 p_v}{\sum_{v=1}^V v p_v} \quad (4.31)$$

4.2. Model Validation

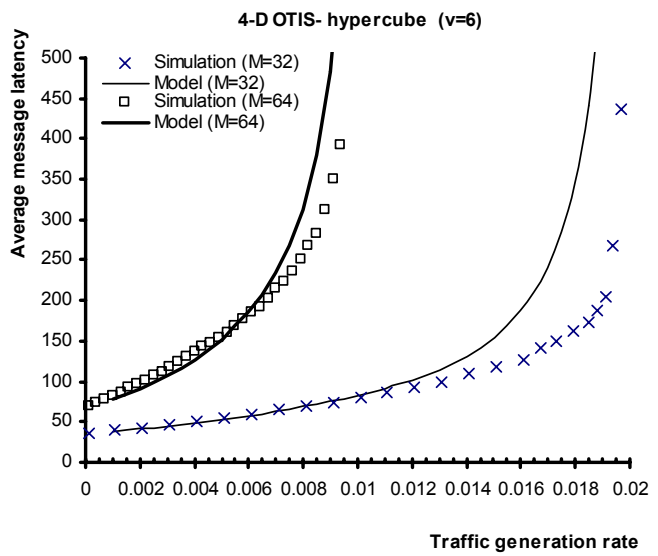
The analytical model has been validated through a discrete-event simulator that mimics the behaviour of minimal path fully adaptive wormhole routing at the flit level in OTIS-hypercube networks based on Duato's methodology [4] used for intra-group routing. In each simulation experiment, a minimum number of 120,000 messages are delivered in 12 batches of messages. Statistics gathered for the first batch of messages was inhibited to avoid distortion due to initial start-up conditions. The simulator uses the same assumptions that analysis, and some of these assumptions are detailed here in hope of making the network operation clearer. The network cycle time is defined as the transmission time of a single flit from one router to the next. It is assumed that the electronic channel cycle time is equal to one network cycle time and $o/e = 0.1$. Messages are generated at each node according to a Poisson process with a mean inter-arrival rate of λ messages/cycles. Message length is fixed at M flits. Destination nodes are determined using uniform random number generation. The mean message latency is defined as the mean amount of time from the generation of a message until the last data flit reaches the local PE at the destination node. Other measures include the mean network latency, the time taken to cross the network, the mean queuing time at the source node, and the time spent at the local queue before entering the first network channel.

Numerous validation experiments have been performed for several combinations of network sizes, message lengths, and number of virtual channels to validate the model. Figures 4.1 and 4.2 depict latency results predicted by the model explained in the previous section, plotted against those obtained by the simulation for 4-dimensional and 6-dimensional OTIS-hypercube networks with $V=4$ and 6 virtual channels and two different message lengths $M=32$ and 64 flits. The horizontal axis in the figures shows the traffic generation rate at each node while the vertical axis shows the average message latency.

The figures reveal that in all cases, the analytical model predicts the average message latency with a good degree of accuracy in the steady-state regions. Moreover, the model predictions are still good even when the network operates in the heavy traffic region, and when it starts to approach the saturation region. However, some discrepancies around the saturation point are apparent. These can be accounted for by the approximations made to estimate the variance of the service time distribution at a channel (term $(\bar{S} - T_{c_{electronic}} M)^2$ in Equations 4.26, 4.27, and 4.28). This approximation greatly simplifies the model as it allows us to avoid computing the exact distribution of the message service time at a given channel, which is not a straightforward task due to interdependencies between service times at successive channels as wormhole routing relies on a blocking mechanism for flow control.

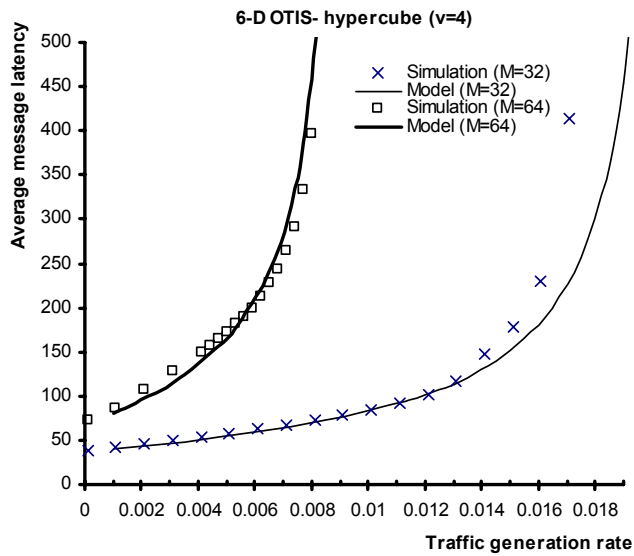


(a)

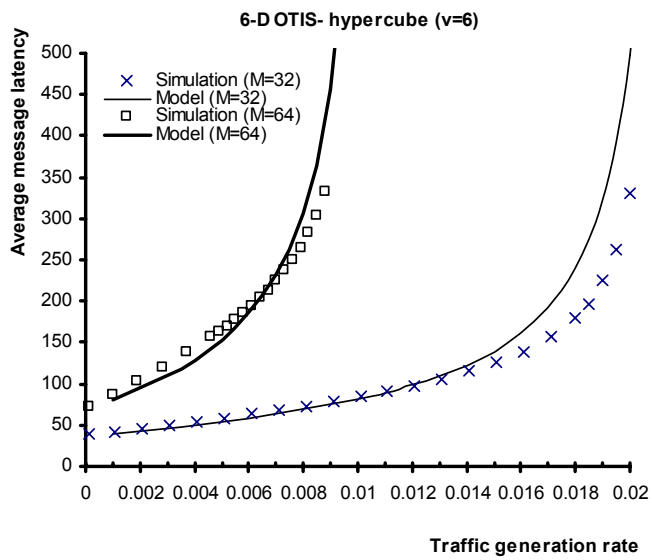


(b)

Figure 4.1: Average message latency of 4-dimesnional OTIS-hypercubes for 4 and 6 virtual channels per physical channel.



(a)



(b)

Figure 4.2: Average message latency of 6-dimesnional OTIS-hypercubes for 4 and 6 virtual channels per physical channel.

4.3. Analysis of Wormhole Routing in the OTIS-Hypercube

In this section, we use the proposed analytical model to study the performance merits of the OTIS-hypercube with adaptive routing. The 5-dimensional OTIS-hypercube, OTIS- H_5 , is used for the sake of the present discussion, but the conclusions reached here have been found to hold for other network configurations as well. In our analysis we assume a unit channel cycle for the optical links and normalize the channel cycle of electronic links to that of an optical link. The *optic to electronic channel cycle time ratio* is denoted by o/e (when the electronic channel cycle time is the time unit).

Figure 4.3 illustrates the mean message latency curves as a function of the traffic rate injected by each node into the network when the message length is $M = 32$ flits and the electronic to optic channel cycle time ratio is varied in an OTIS- H_5 . As can be seen in the figure, the e/o ratio does not have as great impact on overall performance as one would expect. An increase in the ratio slightly increases the average message latency and saturates the network soon.

Figure 4.4 depicts the mean message latency curves as a function of the traffic rate injected by each node into the network (λ) when the message length is varied with $o/e=0.1$ in an OTIS- H_5 . Although not so precious, a conclusion that can be made from the results given in this figure is that, like in all other networks, increasing the message length will result in larger average message latency.

In many applications, the traffic generated is not uniformly distributed between network nodes. Instead, there are some localities in the distribution of destination node addresses. To model traffic locality in the network, we may use the model reported in [75] to define traffic locality. According to this model a node generates local traffic (i.e. intra-group messages) with a probability of f , and thus global traffic (i.e. inter-group messages) with a probability of $1-f$. The above performance model can be easily changed to take care of locality by changing only Equations 4.1 and 4.2 as

$$\xi_1 = f \tag{4.32}$$

$$\xi_2 = 1-f \tag{4.33}$$

Figure 4.5 investigates the effect of traffic locality on overall performance. The locality factors of $f = 0.01, 0.1, 0.2, 0.5,$ and 1.0 are used when drawing the mean message latency curves as a function of the traffic rate injected by each node into the network when $M=32$ flits and $o/e = 0.1$ in an OTIS- H_5 . It can be seen in the figure that having more locality in the traffic can result in a better performance.

Figure 4.6.a and Figure 4.6.b show the impact of o/e ratio and network dimensionality (or size), respectively, into the network saturation point. A network is assumed to be saturated at a given generation rate when average message latency for the given generation rate becomes larger than a predefined large value; the predefined value is assumed to be 2,000,000 unit cycles here. Figure 4.6-a displays that once again increasing the o/e ratio causes early network saturation.

The results of Figure 4.6-b are both interesting and important. It is shown in this figure that increasing the network size has no noticeable effect on the saturation point. As can be seen in the figure, less than 4% degradation results in network saturation when the network size has been increased from 2^4 nodes to 2^{20} nodes. This means that the OTIS-hypercube is a truly scalable network from the performance point of view, as its performance does not drop when its size increases, the case for some networks such as the torus and mesh.

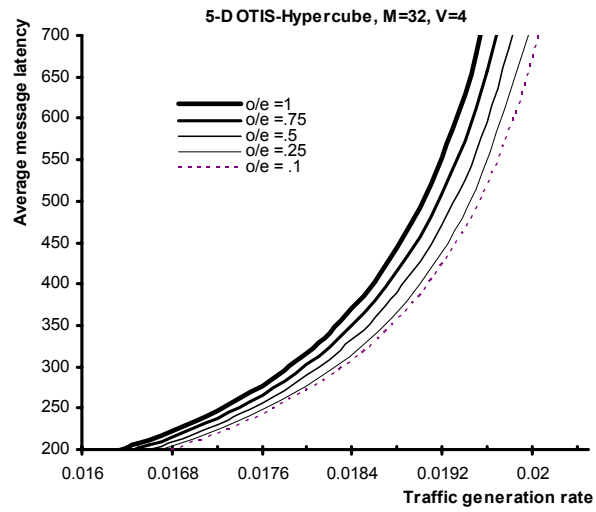


Figure 4.3: Average message latency in an OTIS-H5 with M=32 flits, for different e/o ratios.

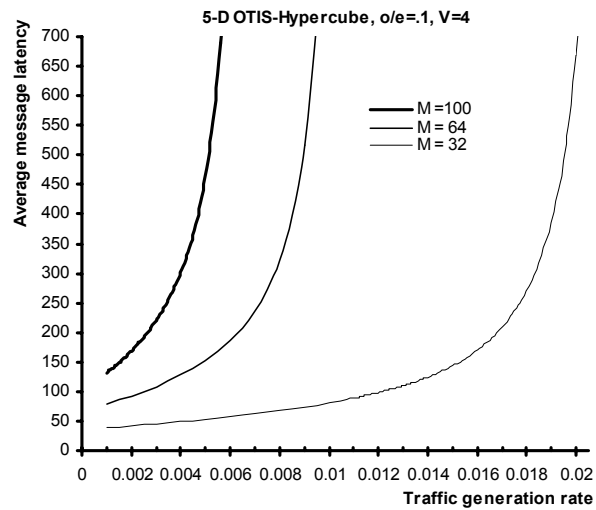


Figure 4.4: Average message latency in an OTIS-H5 with o/e=0.1, for different message lengths.

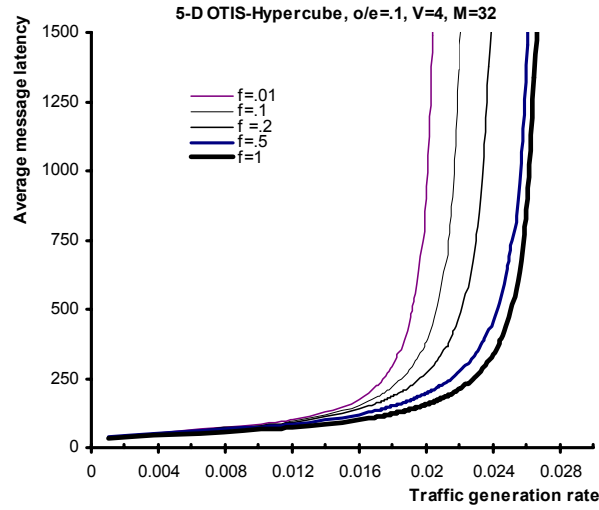
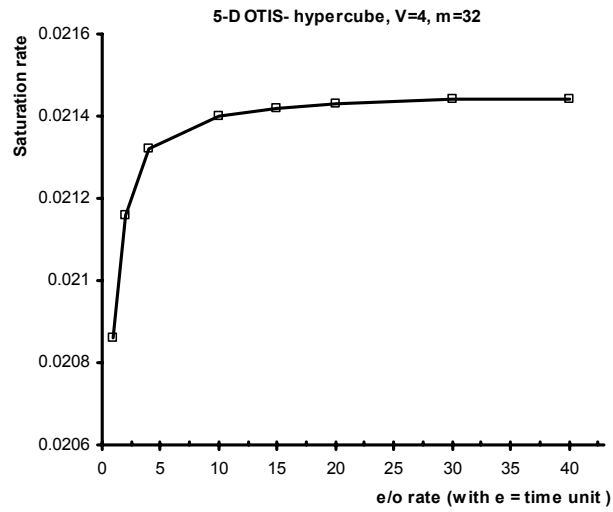


Figure 4.5: Average message latency in an OTIS-H₅ with M=32 flits, and e/o =10, for different traffic locality factors.

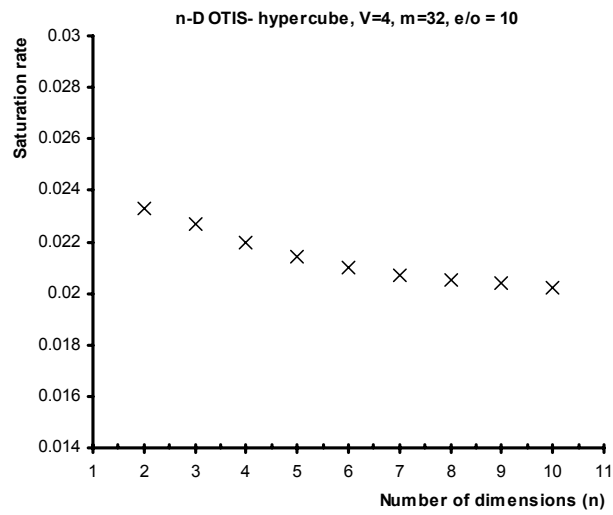
A Cost-Performance Analysis

In this section, we compare the OTIS-hypercube and its equivalent hypercube in view of a cost-performance metric. The reciprocal of the average message latency is generally considered to be representative of the performance of a network and the node degree to be representative of the cost of the network (number of channels in the network). The node degree of an n -dimensional OTIS-hypercube is equal to $n+1$ (n electronic channels and one optical channel) and that of an n -dimensional hypercube is equal to n . But the number of nodes of an OTIS-hypercube is the square of that of a hypercube with the same dimensionality. Therefore, a hypercube and OTIS-hypercube are equivalent when the dimensionality of the hypercube is two times that of the OTIS-hypercube. With these assumptions in mind, the performance to cost ratio of equivalent OTIS-hypercube and hypercube networks are compared in Figure 4.7 for different network sizes and message lengths.

In these figure, it is observed that with different generation rates both the hypercube and OTIS-hypercube have greater performance to cost ratio for shorter message lengths. Independent of message length, the OTIS-hypercube is also observed to be of a greater performance to cost ratio, compared to the hypercube, when both networks are of the same size. This means that at an equal cost the performance of the OTIS-hypercube should be superior to that of the hypercube.

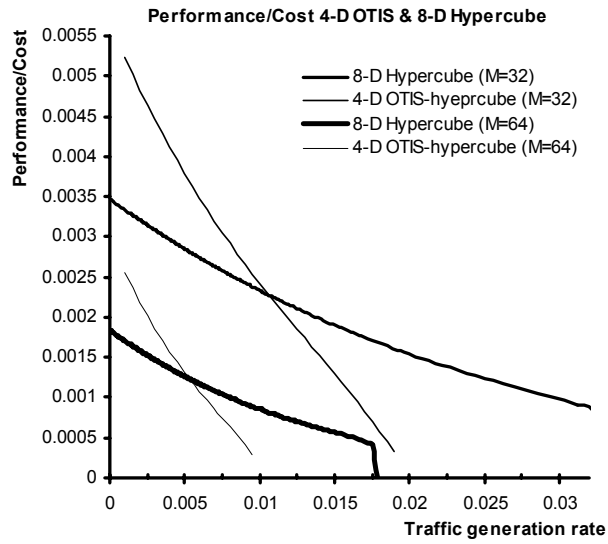


(a)

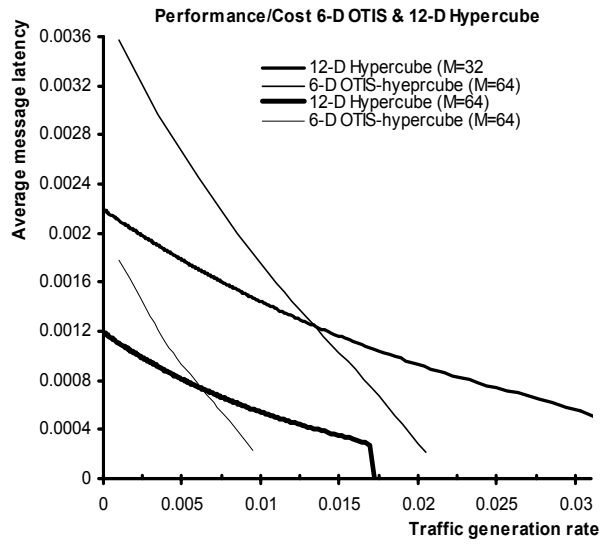


(b)

Figure 4.6: The effect of e/o ratio and network size on the network saturation point $M=32$ flits. (a) the effect of e/o ration, and (b) the effect of network size.



(a)



(b)

Figure 4.7: The performance/cost ratio of the equivalent hypercube and OTIS-hypercube networks.

4.4. Technological Constraint-based Performance Comparison

Extensive examination of interconnection networks has been conducted over the last decade, both with a view to studying fundamental graph-theoretic properties and feasibility of implementation in various technologies [72]. The latter consideration is of crucial importance since in practice, implementation technology puts bandwidth constraints on network channels, and these are important factors in determining how well the theoretical properties of a particular network topology can be exploited. When a multiprocessor system is implemented on a single VLSI-chip, the wiring density of the network is descriptive of the overall system cost and performance [4]. For instance, Dally [42] has used the bisection width, i.e. the number of wires that cross the middle of the network, as a rough measure of the network wiring density in a pure VLSI implementation.

Other researchers, including Abraham [76] and Agrawal [41], have conducted similar studies to Dally's and have arrived at the same conclusions. However, they have also argued that while the wiring density constraint is certainly applicable where an entire network is implemented on a single VLSI-chip, this is not the case in the currently more realistic situation where a network has to be partitioned over many chips. In such circumstances, they have identified that the most critical bandwidth constraint is imposed by the chip's I/O pins through which any data entering or leaving the chip must travel. In other work [77, 78, 79], the performance of multidimensional tori using both of the aforementioned cost constraints, have been compared. In these studies, different wire delay models have also been accounted for.

In the following subsections we compare the effect of bisection-bandwidth and pin-out on the performance of the OTIS-hypercube and hypercube networks.

4.4.1. The Effect of Bisection-bandwidth on Performance

Due to the limited channel bandwidth imposed by implementation technology, a message is broken into channel words (or phits), each of which is transferred in one cycle. If the channel width (i.e. number of wires of each channel) is C_w bits, a message of B bits is divided into $M=B/C_w$ phits [4]. Let us define n_{cube} and n_{otis} to be, respectively, the dimensionality of the hypercube and the OTIS-hypercube. The bisection bandwidth of the OTIS-hypercube and the hypercube, B_{otis} and B_{cube} , with a channel width of $C_{w_{otis}}$ and $C_{w_{cube}}$, can be expressed as

$$B_{cube} = \frac{2^{n_{cube}}}{2} \times C_{w_{cube}} = 2^{n_{cube}-1} C_{w_{cube}} \quad (4.34)$$

and

$$B_{otis} = \left(\frac{2^{n_{otis}}}{2} \right)^2 \times C_{w_{otis}} = 2^{2n_{otis}-2} C_{w_{otis}} \quad (4.35)$$

For a fixed bisection bandwidth, the ratio of the channel width of the OTIS-hypercube to that of the hypercube is given by

$$\mu_{\text{bisection width}} = \frac{C_{w_{otis}}}{C_{w_{cube}}} = \frac{2^{n_{cube}-1}}{2^{2n_{otis}-2}} \quad (4.36)$$

Therefore, $\mu_{\text{bisection width}}$ is the factor by which the channel cycle time of the hypercube grows compared to that in an equivalent OTIS-hypercube, when bisection width is held fixed for both networks.

When both networks have the same number of nodes, it is obvious that the ratio of the channel width of the OTIS-hypercube to that of the hypercube becomes equal to 2. In other words, with an equal bisection bandwidth, the channel cycle time of the hypercube is two times that of an equivalent OTIS-hypercube. Figure 4.8 shows the average message latency of the hypercube and OTIS-hypercube as a function of message generation rate at each node for network sizes of 256 and 4096 nodes, and for messages of $M=32$ and 64 flits, under constant bisection bandwidth constraint.

Observation of these results reveals that the average message latency of an OTIS-hypercube, independent of the message length and network size, is approximately half that of an equivalent hypercube at low generation rates and the OTIS-hypercube saturates at approximately two times the generation rate at which an equivalent hypercube saturates. Thus the OTIS-hypercube performs considerably better than an equivalent hypercube with the same bisection bandwidth.

4.4.2. The Effect of Pin-out on Performance

Similarly, in multiple-chip implementations, where a complete node is fabricated on a chip, pin-out, which is the number of I/O pins (i.e. node degree \times channel width), is a more suitable metric. The node pin-out for the hypercube and OTIS-hypercube networks, $P_{\text{hypercube}}$ and $P_{\text{OTIS-hypercube}}$, can be written as

$$P_{\text{otis}} = (n_{\text{otis}} + 1)C_{w_{\text{otis}}} \quad (4.37)$$

and

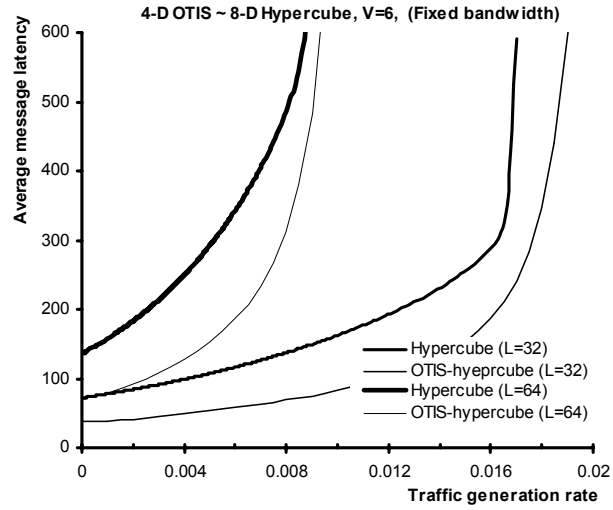
$$P_{\text{cube}} = n_{\text{cube}}C_{w_{\text{cube}}} \quad (4.38)$$

Assuming a constraint of constant node pin-out, the channel cycle ratio of the hypercube and its equivalent OTIS-hypercube graph can be given as

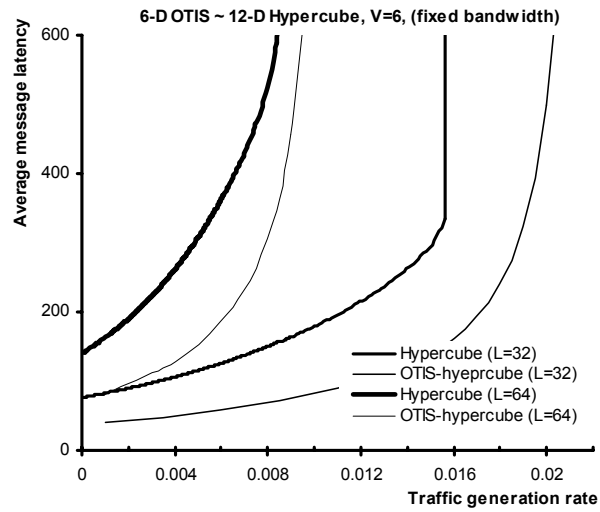
$$\mu_{\text{pin-out}} = \frac{C_{w_{\text{otis}}}}{C_{w_{\text{cube}}}} = \frac{n_{\text{cube}}}{n_{\text{otis}} + 1} \quad (4.39)$$

When the networks have the same number of nodes, we have $\mu_{\text{pin-out}} = 2n/(n+1)$.

Under a constant pin-out constraint, similar to the case of fixed bisection bandwidth, Figure 4.9 shows the average message latency of the hypercube and OTIS-hypercube as a function of message generation rate at each node, for network sizes of 256 and 4096 nodes, and for messages of $M=32$ and 64 flits. The performance (average message latency and saturation point) of the OTIS-hypercube is observed to be superior to that of a same sized hypercube. But the superiority is not as great it was in the case of equal bisection bandwidths.

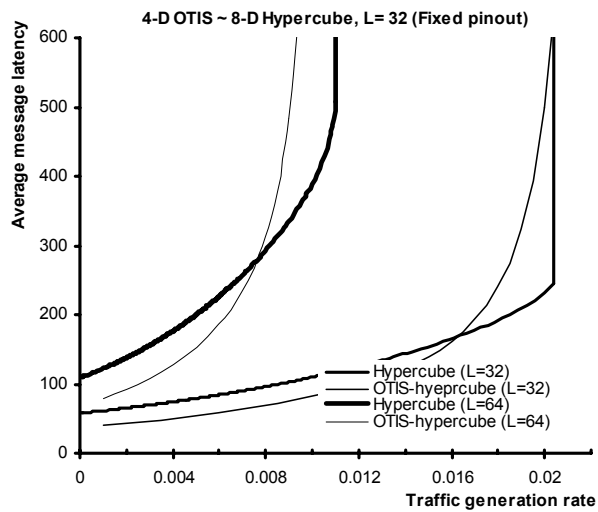


(a)

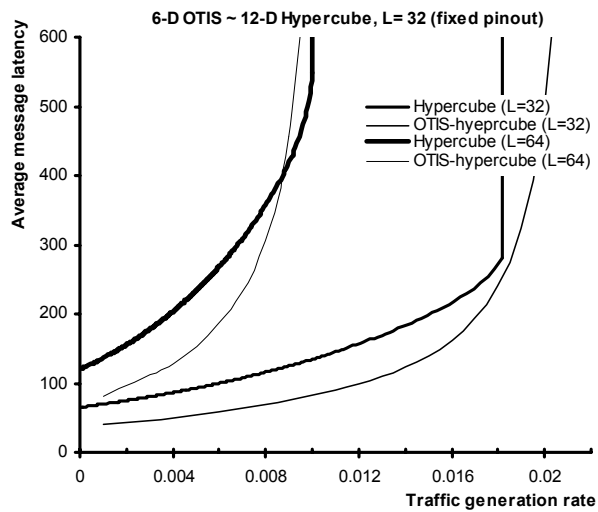


(b)

Figure 4.8: The average message latency of equivalent Hypercube and OTIS-hypercube networks when the channel cycle time of the hypercube is two times that of the OTIS-hypercube.



(a)



(b)

Figure 4.9: The average message latency of equivalent Hypercube and OTIS-hypercube networks when the channel cycle time of the hypercube is such that the pin-out of both networks are equal.

Chapter 5

Conclusions

The OTIS architecture is an interesting option for the implementation of optoelectronic networks in which optical interconnect is used for connecting distant processors. Although the algorithmic properties of the OTIS network have been studied in the literature extensively, the performance of this class of network in realistic conditions, however, has not been studied adequately.

This thesis has confronted the task of evaluating the performance of OTIS interconnection networks. The work has mainly focused on OTIS-cube networks that employ wormhole switching. A number of conclusions about the performance of such interconnection networks have been extracted from the simulation results and performance models.

5.1. Summary of Results

From the simulation results, it has been observed that, both for adaptive and deterministic wormhole routing, the channel cycle time of optical channels are of limited effect on the overall performance of OTIS-hypercube networks. This effect is such that decreasing the channel cycle time of optical channels compared to electronic channels initially results in considerable performance enhancement. But as the optical channel cycle time decreases, the effect of this parameter diminishes. A similar characteristic has been observed for the number of virtual channels, such that the effect of increasing the number of virtual channels, which initially induces performance enhancement, gradually diminishes.

Cubical OTIS networks are found to be, in view of performance, generally scalable networks, such that their bandwidth is independent of their size or dimensionality. It is also found sufficient that the channel cycle time of an OTIS-hypercube (even with the channel cycle time of optical channels equal to that electronic channels) to approximately be half that of an equivalent hypercube for the bandwidth of the two networks to be equal. Considering that long electronic channels do not exist in the OTIS network, such a condition may easily be achievable.

For low message generation rates, it has been shown that the OTIS-hypercube is of superior performance/cost ratio compared to an equivalent hypercube, even when the channel cycle time of the optical channels are assumed to be equal to that of electronic channels.

It has been shown that depending on the traffic pattern, minimal path inter-group routing in the OTIS-hypercube network may not always result in optimal performance, and one of the non-minimal inter-group routing schemes described (in Section 2.3) may be of superior performance. The cause of such behavior by certain traffic patterns has been found to be related to the uneven traffic load distribution over optical links, as explained for each instance observed in the corresponding sections.

The mathematical model obtained for wormhole switching performance has been validated through simulation experiments and can, therefore, serve as a very useful and cost-effective tool for predicting the performance of OTIS-hypercube networks under different structural and traffic conditions.

5.2. Future Work

A number of issues in connection with the OTIS architecture remain uninvestigated. A selection of these problems is presented below.

The hypercube is a special case of the mesh network. But since the size of a mesh may grow by increasing its radices, meshes are usually referred in the context of low dimensional networks. Meshes are also an interesting option for the base topology of the OTIS architecture, resulting in the OTIS-mesh. Currently, no evaluation of the performance of OTIS-mesh networks has been conducted. Such an evaluation may even compare the OTIS-mesh and OTIS-hypercube architectures under different constraints. Another interesting base topology is the DeBruijn network and its comparison to the hypercube when employed as the basis graphs in OTIS-networks.

Because of the minimal-path nature of the inter-group routing schemes presented in this project, both the schemes presented are somehow deterministic (not considering intra-group routing). But non-minimal path inter-group routing need not be deterministic, and depending on traffic load, a message may travel through multiple groups, by taking multiple optical channels (non-consecutively), before reaching its destination. This kind of routing scheme may result in better traffic distribution, but techniques will have to be carefully devised to prevent deadlock and livelock occurrence. An evaluation of such routing schemes may also prove to be rewarding.

Inevitably, the performance of the above mentioned scenarios can also be mathematically modeled to serve as a more lucid source of reference to their performance characteristics. Such models are important and useful tools for designers and researchers working in the area of interconnection networks of massive multicomputers.

References

- [1] T. G. Lewis, H. El-Rewini, "Introduction to Parallel Computing", *Prentice-Hall Inc.*, 1992, pp. 31-32.
- [2] Amdahl, G.M., "Validity of single-processor approach to achieving large-scale computing capability", *Proceedings of AFIPS Conference*, Reston, VA. 1967. pp. 483-485
- [3] L. M. Ni, "Issues in designing truly scalable interconnection networks", *Proceeding of the 1996 ICPP Workshop on Challenges of Parallel Processing*, August 1996, pp. 74-83.
- [4] J. Duato, C. Yalamanchili, L. Ni, "Interconnection networks: an engineering approach", *IEEE computer society press*, 1997.
- [5] H. Fujii, Y. Yasuda, H. Akashi, Y. Inagami, M. Koga, O. Ishihara, M. Kashiya, H. Wada, T. Sumimoto, "Architecture and performance of the Hitachi SR 2201 massively parallel processor system", *Proceedings of the 11th International Parallel Processing Symposium*, 1997, pp. 233-241.
- [6] Y. Yasuda, H. Fujii, H. Akashi, Y. Inagami, T. Tanaka, J. Nakagoshi, H. Wada, T. Sumimoto, "Dealock-free fault tolerant routing in the multidimensional crossbar network and its implementation for the Hitachi SR2201", *Proceedings of the 11th International Parallel Processing Symposium*, 1997, pp. 346-352.
- [7] J. Konicek et al., "The organization of the Cedar system", *Proc. Int'l. Conf. on Parallel Processing*, 1991, pp. 49-56.
- [8] Gregory F. Pfister, William C. Brantley, David A. George, Steve L. Harvey, Wally J. Kleinfelder, Kevin P. McAuliffe, Evelin A. Melton, V. Alan Norton and Jodi Weiss, "The IBM Research Parallel Processor Prototype (RP3): Introduction and Architecture", *Proceedings of the International Conference on parallel processing*, 1985, pp. 764-771.
- [9] M. Banikazemi, V. Moorthy, L. Hereger, D. K. panda, and B. Abali, "Efficient Virtual Interface Architecture Support for IBM SP Switch-Connected NT Clusters", *Proceedings of the International Parallel and Distributed Processing Symposium (IPDPS 2000)*, 2000, pp. 33-42.
- [10] Charles Leiserson, Zahi S. Abuhamdeh, David C. Douglas, Carl R. Feynman, Mahesh N. Ganmukhi, Jeffrey V. Hill, W. Daniel Hillis, Bradley C. Kuszmaul, Margaret A. St. Pierre, David S. Wells, Monica C. Wong, Shaw-Wen Yang, Robert Zak, "The Network Architecture of the Connection Machine CM-5", *Journal of Parallel and Distributed Computing*, Vol. 33, No. 2, 1996, pp. 145-158.
- [11] R. Arlanskas, "iPSC/2 system: a second generation hypercube", *Proceedings of the 3rd ACM Conference on Hypercube Concurrent Computers and applications*, 1988, pp. 38-42.

- [12] Intel Corp., iPSC/1 reference manual, 1986.
- [13] S.F. Nugent, "The iPSC/2 direct-connect communication technology", *Proceedings of the conference on Hypercube Concurrent Computers and applications*, 1988, pp. 51-60.
- [14] Intel Corp., "A Touchstone DELTA system description", 1991.
- [15] Intel Corp., "Paragon XP/S product overview", Supercomputer systems division, Beaverton, Oregon, 1991.
- [16] C.L. Seitz, "The Cosmic cube", *Communication of the ACM*, Vol. 28, No. 1, 1985, pp. 22-33.
- [17] nCUBE systems, "n-Cube handbook", 1986.
- [18] nCUBE systems, "nCUBE 3", at <http://www.ncube.com>
- [19] Anant Agarwal, Ricardo Bianchini, David Chaiken, Kirk L. Johnson, David Kranz, John Kubiawicz, Beng-Hong Lim, Ken Mackenzie, and Donald Yeung. "The MIT Alewife Machine: Architecture and Performance", *Proceedings of the 22nd Annual International Symposium on Computer Architecture*, 1995, pp. 2-13.
- [20] M. Noakes, D. A. Wallach, W. J. Dally, "The J-machine multicomputer: an architectural evaluation", *Proceeding of the 20th International Symposium on Computer Architecture*, 1993, pp. 224-235.
- [21] M. Noakes, W. J. Dally, "System design of the J-machine", *Proceedings of Advanced research in VLSI*, MIT Press, 1990, pp. 179-192.
- [22] C. Peterson, J. Sutton, P. Wiley, "iWARP: a 100-MOPS VLIW microprocessors for multicomputers", *IEEE MICRO*, Vol. 11, No. 13, 1991.
- [23] D. Lenoski et al. "The Stanford DASH Multiprocessor", *IEEE Computer*, Vol. 25, No. 3, Mar 1992, pp. 63-79.
- [24] Jeffrey Kuskin, David Ofelt, Mark Heinrich, John Heinlein, Richard Simoni, Kourosh Gharachorloo, John Chapin, David Nakahira, Joel Baxter, Mark Horowitz, Anoop Gupta, Mendel Rosenblum, and John Hennessy, "The Stanford Flash Multiprocessor", *In Proceedings of 21st International Symposium on Computer Architecture*, April 1994, pp. 302-313.
- [25] R.E. Kessler, J.L. Swarszmeier, "Cray T3D: a new dimension for Cray research", *In Proceedings of the 38th IEEE Computer Society International Conference (COMPCON)*, February 1993, pp. 176-182.
- [26] E. Anderson, J. Brooks, C. Grassl, S. Scott, "Performance of the Cray T3E multiprocessor", *Proceedings of the Supercomputer Conference*, 1997, pp. 19.
- [27] Cray Research Inc., "The Cray T3E scalable parallel processing system", at http://www.cray.com/PUBLIC/product-info/T3E/CRAY_T3E.html.

- [28] J. Laudon, D. Lenoski, "The SGI Origin: A ccNUMA highly scalable server", *Proceedings of the 24th International Symposium on Computer Architecture*, 1997, pp. 241-251.
- [29] S. B. Akers, D. Harel, B. Krishnamurthy, "The star graph: an attractive alternative to the n-cube", *Proceeding of the International Conference on parallel Processing*, 1987, pp. 393-400.
- [30] S. B. Akers, D. Harel, B. Krishnamurthy, "A group-theoretic model for symmetric interconnection networks", *IEEE Transaction on Computers*, Vol. 38, No. 4, 1989, pp. 555-566.
- [31] F. P. Preparata and J. Vuillemin, "The cube-connected cycles: a versatile network for parallel computation", *Communication of ACM*, Vol. 24, 1981, pp. 300-309.
- [32] L. N. Bhuyan, D. P. Agrawal, "Generalized hypercube and hyperbus structures for a computer network", *IEEE Transaction on Computers*, Vol. 33, 1984, pp. 323-333.
- [33] C. Peterson, J. Sutton, P. Wiley, "iWARP: a 100-MOPS VLIW microprocessor for multicomputers", *IEEE MICRO*, Vol. 11, No. 13, 1991.
- [34] D. A. Reed, R.M. Fujimoto, "Multicomputer networks: message based parallel processing", *MIT press*, 1987.
- [35] D. Nassimi, S. Sahni, "Finding connected components and connected ones on a mesh-connected parallel computer", *SIAM journal on computing*, Vol. 9, 1980, pp. 744-757.
- [36] C.L. Seitz, "Mosaic C: an experimental, fine-grain multicomputer", *In Proc. International Conference Celebrating the 25th Anniversary of INRIA, SpringerVerlag, LNCS 653*, December 1992, pp. 69-85.
- [37] W. J. Dally, H. Aoki, "Deadlock-free adaptive routing in multicomputer networks using virtual channels", *IEEE Transaction on Parallel and Distributed Systems*, Vol.4, No. 4, 1993, pp. 66-74.
- [38] W. J. Dally, "Virtual channel flow control", *IEEE Transaction on Parallel and Distributed Computing*, Vol. 3, No. 2, 1992, pp. 194-205.
- [39] S. Rammy, "Routing in wormhole networks", PHD Dissertation, Computer Science Department, University of Saskatchewan, 1995.
- [40] G. L. Frazier, "Buffering and flow control in communication switches for scalable multicomputers", PHD Thesis, University of California, Los angles, 1995.
- [41] A. Agrawal, "Limits on interconnection network performance", *IEEE Transaction on Parallel and Distributed Systems*, Vol. 2, No. 4, 1991, pp. 398-412.

- [42] W. J. Dally, "Performance analysis of k-ary n-cube interconnection networks", *IEEE Transaction on Computers*, Vol. C-39, No. 6, 1990, pp. 775-785.
- [43] W. J. Dally, C. L. Seitz, "The torus routing chip", *Journal of Distributed Computing*, Vol. 1, No. 3, 1986, pp. 187-196.
- [44] J. Duato, "Why commercial multicomputers do not use adaptive routing", *IEEE Technical Committee on Computer Architecture Newsletter*, 1994, pp. 20-22.
- [45] A. Jourdan, F. Masetti, M. Garnot, G. Soulage, M. Sotom: "Design and Implementation of a Fully Reconfigurable All-optical Crossconnect For High Capacity Multiwavelength Transport Network", *IEEE Journal of Lightwave Technology*, June 1996, vol.14, no. 6, pp. 1198-1206.
- [46] Amnon Barak, Eugen Schenfeld, "Embedding Classical Communication Topologies in the Scalable OPAM Architecture", *IEEE Transactions on Parallel and Distributed Systems*, Vol. 7, Issue 9, pp. 979 – 992, 1996.
- [47] Andreas G. Nowatzky, Paul R. Prucnal, "Are crossbars really dead?: the case for optical multiprocessor interconnect systems", *Proceedings of the 22nd annual international symposium on Computer architecture*, pp. 106-115, 1995.
- [48] Brian Webb, Ahmed Louri, "A Class of Highly Scalable Optical Crossbar-Connected Interconnection Networks (SOCNs) for Parallel Computing Systems", *IEEE Transactions on Parallel and Distributed Systems*, pp. 444-458, 2000.
- [49] P.W. Dowd, "Wavelength Division Multiple Access Channel Hypercube Processor Interconnection", *IEEE Transaction on Computers*, 41(10), pp. 1223-1241, 1992.
- [50] C. Dragone, "Efficient n x n star coupler based on Fourier optics", *Electronics Letters*, 24(15), 1988, pp. 942-944.
- [51] R.A. Linke, "Frequency division multiplexed optical networks using heterodyne detection", *IEEE Network Mag.*, 3(2), 1989, pp. 13-20.
- [52] R. A. Thompson, "The dilated slipped Banyan switching network architecture for use in an all-optical local area network", *IEEE J. Lightwave Technol.*, 9(12), pp. 1780-1787.
- [53] T. Q. Dam, K.A.K. A. Williams, D. H. C. Du, "A media-access protocol for time and wavelength division multiplexed passive star networks", Tech. Report 91-63, Computer science dept., University of Minnesota.
- [54] M. G. Hluchyj, M. L. Karol, "ShuffleNet: an application of generalized perfect shuffles to multihop lightwave networks", *J. Lightwave Technology*, 9(10), 1991, pp. 1386-1396.
- [55] M. Feldman, S. Esener, C. Guest and S. Lee, "Comparison between electrical and free space optical interconnects based on power and speed considerations", *applied optics*, May 1988, 27(9): 1742-1751.

- [56] F. Kiamilev, P. Marchand, A. Krishnamoorthy, S. Esener, and S. Lee, "Performance comparison between optoelectronic and VLSI multistage interconnection networks", *journal of lightwave technology*, Dec. 1991, 9(12): 1674-1692.
- [57] G. C. Marsden, P. J. Marchand, P. Harvey, and S. C. Esener, "Optical transpose interconnect system architectures", *Optical Letters*, July 1993, 18(13): 1083-1085.
- [58] W. Hendrick, O. Kibar, P. Marchand, C. Fan, D. V. Blerkom, F. McCormick, I. Cokgor, M. Hansen, and Esener, "Modeling and optimization of the optical transpose interconnection system", *In optoelectronic technology Center, Program Review*, Cornell University, Sept. 1995.
- [59] F. Zane, P. Marchand, R. Paturi, and S. Esener, "Scalable network architectures using the optical transpose interconnection system (OTIS)", *In proceedings of the second International Conference on Massively Parallel Processing using Optical Interconnections (MPPOI'96)*, San Antonio, Texas, 1996, pp. 114-121.
- [60] Krishnamoorthy, P. Marchand, F. Kiamilev, and S. Esener, "Grain-size considerations for optoelectronic multistage interconnection networks", *Applied Optics*, Sept. 1992, 31(26): 5480-5507.
- [61] S. Sahni, C.-F. Wang, "BPC permutations on the OTIS-hypercube optoelectronic computer", *Informatica*, 1998, 22: 263-269.
- [62] S. Sahni and C.-F. Wang, "BPC permutations on the OTIS-mesh optoelectronic computer", *In proceedings of the fourth international conference on massively parallel processing using optical interconnections (MPPOI'97)*, 1997, pp. 130-135.
- [63] C.-F. Wang and S. Sahni, "Matrix multiplication on the OTIS-mesh optoelectronic computer", *In Proceedings of the sixth international conference on Massively Parallel Processing using Optical Interconnections (MPPOI'99)*, 1999, pp. 131-138.
- [64] C. -F. Wang and S. Sahni, "Image processing on the OTIS-mesh optoelectronic computer", *IEEE transaction on parallel and distributed systems*, 2000, 11(2): 97-107.
- [65] C. -F. Wang and S. Sahni "Basic operations on the OTIS-mesh optoelectronic computer", *IEEE transaction on parallel and distributed systems*, 1998, 9(12): 1226-1236.
- [66] S. Rajasekeran and S. Sahni "Randomized routing, selection and sorting on the OTIS-mesh", *IEEE transaction on parallel and distributed systems*, 1998, 9(9): 833-840.
- [67] A. Osterloh, "Sorting on the OTIS-mesh", *In Proceedings of the 14th International Parallel and Distributed Processing Symposium (IPDPS'2000)*, 2000, pp. 269-274.

- [68] A. A. Chien, "A cost and speed model for k-ary n-cube wormhole routers", *In Proceedings of Hot Interconnects '98*, August 1993.
- [69] C. J. Glass and L. M. NI, "The Turn model for adaptive routing", *J. ACM*, vol. 5, 1994, pp. 874–902.
- [70] J. Duato, T. Pinkston, "A general theory for deadlock-free adaptive routing using a mixed set of resources", *IEEE Transaction on Parallel and Distributed Systems*, Vol. 12, 2001, pp. 1219-1235.
- [71] H. H. Najaf-abadi, H. Sarbazi-azad, "Routing Capability and Performance of Cubical OTIS Networks", *to appear in proceeding of the International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS'04)*, San Jose, California, 2004.
- [72] M. Ould-Khaoua, H. Sarbazi-Azad, "An analytical model of adaptive wormhole routing in hypercubes in the presence of hotspot traffic", *IEEE Transactions on Parallel and Distributed System*, 2001, 12(3): 283-292.
- [73] M. Ould-Khaoua, "An analytical model of Duato's adaptive routing algorithm", *IEEE Transaction on Computers*, 48(12), 1999, pp. 1-8.
- [74] L. Kleinrock, "Queueing Systems", Vol. 1, John Wiley, 1975.
- [75] H. Sarbazi-Azad, A. Khonsari, M. Ould-Khaoua, "Analysis of deterministic routing in k-ary n-cubes with virtual channels", *Journal of Interconnection Networks*", 2002, 3(1-2):85-101.
- [76] S. Abraham, K. Padmanabhan, "Performance of the multicomputer networks under Pin-out constraints", *Journal of Parallel and Distributed Computing*, Vol. 12, No. 3, 1991, pp. 237-248.
- [77] D. Basak, D. Panda, "Designing clustered multiprocessor systems under packaging and technological advances", *IEEE Transaction on Parallel and Distributed Systems*, Vol. 7, No. 9, 1996, pp. 962-978.
- [78] J. Duato, P. Lopez, "Performance evaluation of adaptive routing algorithms for k-ary n-cubes", *Proceedings of Parallel Computer Routing and Communication, LNCS 853*, 1994, pp. 45-59.
- [79] S. L. Scott, J. R. Goodman, "The impact of pipeline channels on k-ary n-cube networks", *IEEE Transaction on Parallel and Distributed Systems*, Vol. 5, No. 1, 1994, pp. 2-16.

Publications During the Course of This Research

1. H. Hashemi-Najafabadi, H. Sarbazi-Azad, **Comparative evaluation of adaptive and deterministic routing in the OTIS-hypercube**, *Proceedings of Ninth Asia-Pacific Computer Systems Architecture Conference*, September 7-9, 2004, Beijing, China, *Lecture Notes on Computer Science*, Springer Verlag.
2. H. Hashemi-Najafabadi, H. Sarbazi-Azad, **The effect of adaptivity on the performance of the OTIS-hypercube under different traffic patterns**, to appear, *Proceedings of IFIP International Conference on Network and Parallel Computing*, October 18-20, 2004, Wuhan, China, *Lecture Notes on Computer Science*, Springer Verlag.
3. H. Hashemi, H. Sarbazi-azad, **On routing capabilities and performance of cube-based OTIS-systems**, to appear, *Proceedings of International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS'04)*, July 25 - 29, 2004, San Jose, CA, USA.
4. H. Hashemi-Najafabadi, H. Sarbazi-Azad, **A new analytical performance model for n-d Tori**, to appear, *Proceedings of IEEE ICCD'2004 conference*, CA, U.S.A.
5. H. Hashemi-Najafabadi, H. Sarbazi-Azad, **An accurate combinatorial model for performance prediction of deterministic wormhole routing in the n-d torus network**, under review in *Applied Mathematical Modelling*.
6. H. Hashemi-Najafabadi, H. Sarbazi-Azad, **An empirical performance analysis of wormhole routing in cube-based optoelectronic multicomputers**, under review in *Parallel Computing*.
7. H. Hashemi-Najafabadi, H. Sarbazi-Azad, **Mathematical performance evaluation of adaptive wormhole routing in optoelectronic hypercubes**, under review in *IEEE Transactions on Parallel and Distributed Systems*.
8. H. Hashemi-Najafabadi, H. Sarbazi-Azad, **Performance analysis of OTIS-hypercubes**, under review in *Performance Evaluation*.
9. H. Hashemi-Najafabadi, H. Sarbazi-Azad, **An empirical comparison of the OTIS-hypercube and OTIS-mesh networks**, submitted to *IEEE/ACM MASCOTS'2004 conference*, CA, U.S.A.