Contents

List of Figures

List of Tables

Abstract

Next generation high performance computing (HPC) highly depends on the massively parallel computers (MPC). The overall performance of a massively parallel computer system is heavily affected by the interconnection network and its processing nodes. Continuing advances in VLSI technologies promise to delivering more power to individual nodes. But the on-chip interconnection networks consume 50% of the total power and off-chip bandwidth is limited to the maximum number of possible out going physical links. On the other hand, low performance of communication network degrades the parallel system. Hence it is important to find a suitable interconnection network with the existing technologies and at the same time it is also important to evaluate the network performance. In this research proposal, we like to introduce a new interconnection network that could reduce those problems (like- high power consumption, longer wire length, low bandwidth issue and etc.) and also like to measure the static and dynamic communication performance of our newly introduced interconnection network, comparing the performance results with other networks at different levels of hierarchy such as inter-chips, inter-nodes and inter-cabinets.

Acknowledgments

First of all, I would like to express my deepest gratitude to my supervisor, Professor Yasushi Inoguchi for his great support and effective advise during my research period. His guidance helped me to find the correct direction of my research. His constant inspiration and encouragement allow me to reach the final goal of this research.

Secondly, I like to express my sincere thanks to Assistant Professor Yukinori Sato (School of Information Science, Japan Advanced Institute of Science and Technology) and Assistant Professor M.M. HAFIZUR RAHMAN (Department of CS, KICT, IIUM) for their kind supports during my research period.

Then, I like to express my gratitude to second supervisor Associate Professor Kiyofumi Tanaka for his cordial advices during my research.

I also wish to express my honor to my sub-theme supervior Associate Professor Masato Suzuki and Professor Mineo Kaneko for their valuable advices and suggestions.

I am eagerly acknowledge the Japan Educational Exchanges and Services (JEES) for granting me the "2013 NTT DOCOMO Scholarship for Applicants Residing Abroad " which makes my study and life feasible in Japan. And also acknowledge the "Research Grants for JAIST Students" which allows me to present my research work in international conference.

Finally, I like to give my sincere acknowledgments to my family members for their huge support and patience; to my father, my mother and also to my spouse.

Chapter 1

Introduction

1.1 Background

High performance computer (HPC) is the increased demand for next generation of computers. Sequential computers fails to meet this demand and already reached to the saturation point due to the scaling difficulties of uniprocessor architectures. Hence the need for massively parallel computers (MPC) is increasing day by day. Massively parallel computers with thousand of nodes have been commercially available with tera-flops performance and efforts have been made to build MPC systems with millions of nodes for peta-flops or even for exa-flops performance. To facilitate the millions of node, interconnection networks are the key elements [1]. Interconnection network acts as a path between one node to another. Considering millions of nodes, the large diameter of conventional interconnects is completely infeasible. On the other hand, to jump into the next level there is few key constraints that limit network performance, cost-performance ratio, power consumption, throughput and latency [2].

In the future intuitive, flexible and highly interactive parallel computers with massive computation power will replace the recent sequential computers having small computation power. Parallel computers will not only be an important factor for day-to-day life, will also be a major shortcut for next generation technologies. Even now a day, parallel computer has become an integral part of society. Sectors like- Banking, Education system, Military, communication and even in research fields, usages of parallel computer has already been boosted up and removed all the traditional equipment. But the

problem resides for parallel computer like- supercomputers is its processing power, memory, inter-communication between the CPUs, power consumptions and the cooling systems.

1.2 Problem Statement

One of the main constraints for MPC systems is the suitable interconnection network that could be scaled up to millions of nodes with small diameters. Next is the cost-performance issue, increased outgoing link from each node increases the total cost of the network. Interconnection links consume the most of the power; shorter link consumes smaller power where as longer links consumes more power. Even according to the advancement of current technology to build a one exa-flop system, it needs close to 1000MW of electrical power, which indeed a major constraint for next generation computers. And also it is always desirable an interconnection network with low latency and high throughput for high network performance.

1.3 Objective

Hierarchical Interconnection networks (HIN) [3] are cost-effective way to interconnect a large processing system. Even lots of hypercube based hierarchical interconnection networks have also been available; but for MPC systems, the number of physical links is a major concern. With the early researches, TORUS networks shows better performance than the MESH network due to the tori connections but consumes more power than the MESH network due to the increased connections and even the total cost increases for the increased links. One of the recently used network is the TOFU interconnection network [4], which requires 10 outgoing links for each node; causing more cost than the other networks and even the scaling at higher level causes low performance issues. Hence it is important to find a new interconnection network with concerning the key constraints and improved network performance as well as reducing the overall power consumption than the other existing networks.

1.4 Approach

High computation power is the great challenge and also the increased demand for next generation computers. Two-dimensional (2D) networks were the main focus for recent studies; with the increase of cores, traditional 2D networks are no longer efficient for many-core processors. Even it consumes more power and shows lower performance than the 3D networks. Hence it is important to find a new interconnection network, which is suitable for next generation interconnection network with reduced power and shorter interconnects. Routing plays a vital role for the overall performance of the interconnection networks. Dimension-order routing has been popular for MPC systems due to its minimal hardware requirements and allows the design of simple and fast routers. Routing for MESH and TORUS uses the dimension-order routing. Hence it is also important to find a suitable routing logic for the new interconnection network.

1.5 Scope of the Thesis

Network topology affects the performance metrics. In previous researches, HIN networks (like- TESH [5], TTN [6] and STTN [7]) show better static network performance than the conventional MESH and TORUS networks. It is being expected for the interconnection network with low cost, low degree, low congestion, high connectivity and high fault tolerance. With fixed routing logic, it is important to measure the static network performance of new interconnection network to compare the performance result with the other conventional networks. Evaluating the static network performance consists of measuring the below parameters-
1. Diameter. 2. Average Distance.
3. Node Degree. 4. Cost. 5. Power Consumption.

The actual performance of the network can be found through the dynamic communication performance. Low latency and high network throughput are desirable for any interconnection network [8]. To compare the new network dynamic communication performance with the TORUS, MESH and even with the recent networks, it is also needed to find performance results with the common parameters for new network and others. Uniform traffic pattern will help to compare the dynamic performance of new network with the

existing networks. And the non-uniform load distribution will be helpful to observe the networks maximum tolerance capabilities. Evaluating the dynamic communication performance of new interconnection network consists of measuring the below parameters-
1. Network Latency 2. Network Throughput

1.6 Organization of the Thesis

At the end of this chapter the rest of the paper is organized as follows. In chapter 2, we deal with the relative conventional interconnection networks, their architecture and also the other hierarchical interconnection like- TESH, TTN and STTN with their architectural interconnections. In chapter 3, we deal with the concept of 3D Tori connected mESH Network (3D-TESH), his architecture, the addressing and the routing algorithm of 3D-TESH. In chapter 4, we evaluate the static network performance of 3D-TESH with respect to other conventional network performance. Chapter 5 presents the evaluation of the dynamic network performance of 3D-TESH with respect to other conventional network performance. Chapter 6 presents the conclusion of this research work and gives some direction of future works.

Chapter 2

Related Works

2.1 Introduction

Massively Parallel Computes (MPC) highly depend on the interconnection networks. But interconnection network with large diameter is completely infeasible. Hierarchical Interconnection network (HIN) are cost-effective way to interconnect a large processing system. Even lots of hypercube based hierarchical interconnection networks have also been available; but for MPC systems, the number of physical links is the major concern. A MPC system with small number of outgoing links is cost effective and desirable. To move into the next generation systems, key constraints will be the network performance, cost-performance ratio, power consumption, throughput and latency [2]. In this research, we like to develop a new interconnection network from the viewpoint of reduced power consumption; whereas one of the possible candidates for interconnection network can be the hierarchical interconnection network.

2.2 Various types of Interconnection Networks

2.2.1 MESH Network

Mesh [9] is one of the k-ary n-cube networks, which is easy to layout in on-chips due to its regular and equal-length links. It has high path diversity i.e., there is many ways to reach from one node to another. Mesh has been used in Tilera 100-core CMP and On-chip network prototypes. Average

Latency is O(sqrt(N)) and has O(N) cost. Figure 2.1 shows the architectural interconnect for 2D-Mesh network.

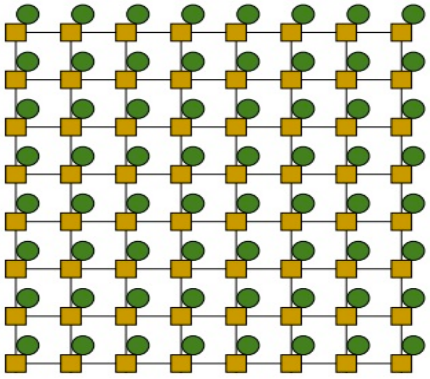


Figure 2.1: Architecture of MESH Network [9]

2.2.2 TORUS Network

Mesh is not symmetric on edges hence its performance very sensitive to placement of task on edge vs. middle. Torus [9] network avoids this problem. It has higher path diversity (& bisection bandwidth) than mesh network. It requires higher cost and harder to layout on-chip and unequal link lengths. Figure 2.2 shows the node level interconnections for 2D-Torus network.

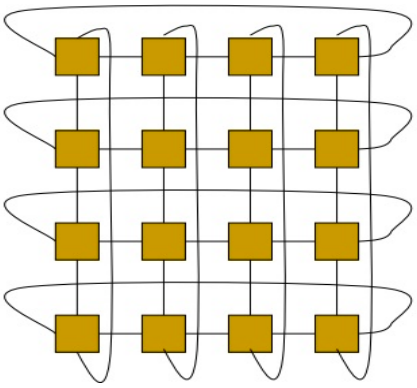


Figure 2.2: Architecture of Torus Network [9]

2.2.3 TOFU Network

TOFU interconnection network has been adopted by the Fujitsu K computer and already achieved 10.51 petaflops performance requiring power consumption of 12.6 MW [10]. TOFU network has 6 dimensional mesh/torus interconnect. 10 links are used for inter-node connection of TOFU network, where 6 links are scalable for connecting 3D-torus interconnect and another 4 links are fixed sized connecting 3D-mesh/torus topology. Figure 2.3 shows the interconnecting structure of TOFU network.

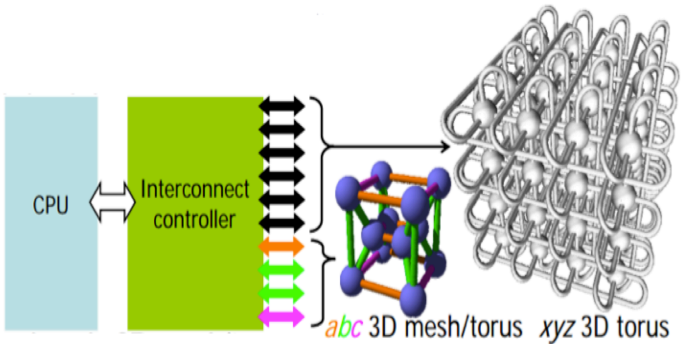


Figure 2.3: Architecture of TOFU Network [11]

2.2.4 5D-TORUS Network

5D-Torus network shows excellent performance for the MPI-type communications. In 5D-Torus network, one node is connected with 10 neighboring nodes. Recently, 5D-Torus network has been adopted in IBM Blue Gene/Q super-computer. Though 5D-Torus and Tofu interconnect uses same number of outgoing links, 5D-Torus shows better performance over the TOFU interconnect due to the greater over-provisioning of links and greater fail-proof resiliency by resulting the greater complexity and cost. Figure 2.4 shows the 5D-torus interconnect between the various nodes.

2.2.5 Hierarchical Interconnection Networks (HIN)

2.2.5.1 Tori-connected mESH (TESH) Network

The Tori-connected mESH (TESH) [5] Network is a hierarchical interconnection network consisting of Basic Modules (BM) that are hierarchically

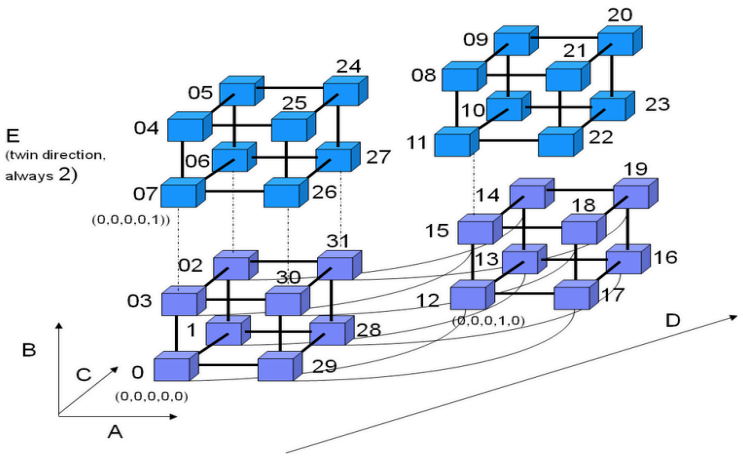


Figure 2.4: Architecture of 5D-Torus Network [12]

interconnected to form a higher level network with multiple lower level networks. The BM of the TESH network is a 2D-mesh network of size $(2^m \times 2^m)$. BM refers to a Level-1 network. Higher level TESH networks are built by recursively interconnecting immediately lower level subnetworks in a 2D-torus fachion. A higher-level network is built using a 2D-toroidal connection among (2^{2m}) immediate lower level subnetworks. If m = 2, the size of the BM is (4 $\times$ 4), and similarly if m = 3 then the size of the BM will be (8 $\times$ 8) [5]. A BM of (4 $\times$ 4) is shown in Figure 2.5. As shown in the figure, the BM has some free ports in the periphery for higher level interconnection. All ports of the interior Processing Elements (PEs) are used for intra-BM connections. All free ports of the exterior PEs, either one or two, are used for inter-BM connections to form higher level networks.

2.2.5.2 Tori-connected Torus Network (TTN)

Tori connected Torus Network (TTN) [6] is also a hierarchical interconnection network. The lowest level of TTN is the Level-1 network defined as the basic module. Multiple basic modules (BM) are hierarchically interconnected to form a higher level TTN network. A $(2^m \times 2^m)$ BM consists of a 2D-torus network of 2^{2m} processing elements (PE) having 2^m rows and 2^m columns, where m is a positive integer. Figure 2.6 shows the BM for TTN with m = 2, which defines the size of BM as (4 $\times$ 4). Each BM has 2^{m+2} free ports for the higher level interconnections. Ports of the interior nodes are used for

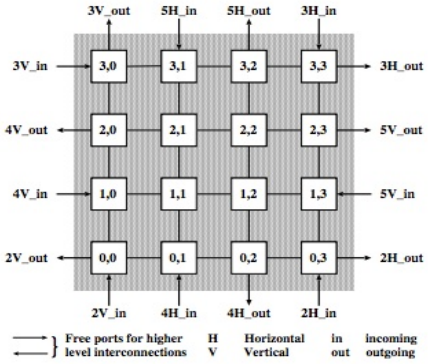


Figure 2.5: Architecture of TESH Network [5]

intra-BM level and rest of the free-ports at the exterior nodes, either one or two, are used for inter-BM connections to form higher level networks.

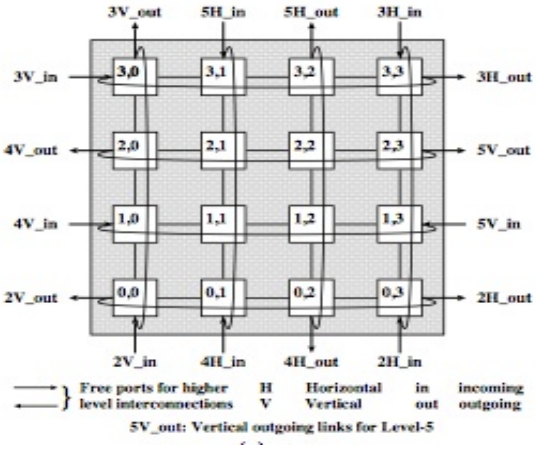


Figure 2.6: Architecture of TTN Network [6]

2.2.5.3 STTN Network

The Symmetric Tori connected Torus Network (STTN) [7] is a hierarchical interconnection network, whose lowest level of network is the Level-1 network defined as the basic module. Multiple basic modules (BM) are hierarchically interconnected to form a higher level network. A $(2^m \times 2^m)$ BM consists of a 2D-torus network of 2^{2m} processing elements (PE) having 2^m rows and 2^m

columns, where m is a positive integer. Figure 2.7 depicts the basic module for STTN with m = 2, which defines the size of BM as (4 × 4). Each BM has 2^{m+2} free ports at the contours for higher level interconnection. All ports of the interior nodes are used for intra-BM level. All free-ports of the exterior nodes, either one or two, are used for inter-BM connections to form higher level networks. Higher level networks for STTN are built by the recursive interconnection (2^{2m}) of the immediate lower level sub- networks. Considering ($m = 2$) a Level-2 STTN network can be formed by interconnecting ($2^{2\times2}$) = 16 BMs. Similarly, a Level-3 network can be formed by interconnecting 16 Level- 2 sub-networks.

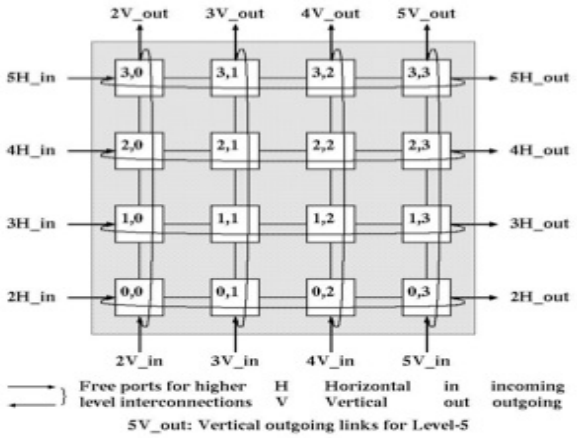


Figure 2.7: Architecture of STTN Network [7]

2.3 Importance of multi-dimensional network (3D-TESH)

Interconnection network consists of multiple nodes communicating with others directly or indirectly. In direct networks, point to point links interconnection is used and in indirect connection, communication is carried through some switches. Already large numbers of interconnection network have been proposed, ranging from conventional to hierarchical interconnects. With increased demand of high computational power, two dimensional (2D) interconnection networks are no longer sufficient for many-core processors due

to the limited capacity for single core processors. As the multi-dimensional interconnection has the advantages of shorter global interconnects, high performance & low power consumption, this is the key motivation for us to consider a multi-dimensional network. On the other hand, according to the previous researches hierarchical networks shows better performance than the conventional networks. Hence in this research plan, we have considered a new multi-dimensional hierarchical interconnection network(3D-TESH) with two dimensional higher-level of interconnections.

Regarding the consideration of two dimensional interconnection for the higher level networks, is based upon the complexity of the network. If we have considered a three or more dimensional interconnection network at the higher level, the network size will be increased much more than two dimensional interconnect. Hence the wiring complexity and the power consumption will also be increased. But in this research plan, we are trying to find a new interconnection network that will require less power consumption and show better network performance. Hence we have considered a two dimensional interconnect at the higher level of 3D-TESH network.

2.4 Virtual Channels

Virtual channels are very important factor for the analysis of dynamic communication performance evaluation. The most common method for implementing separate buffering resources at each input port is known as virtual channels (VC). The hardware cost increases with the number of virtual channels increases. The unconstrained use of virtual channels is cost-prohibitive in parallel systems. Therefore, a deadlock-free routing with minimum number of virtual channels is expected. Virtual channels are used to solve the deadlock avoidance problem, but they can be also used to improve network latency and throughput. Figure 2.8 illustrates two model situations: "assume packet P_0 arrived earlier and acquired the channel between the two routers first. In absence of virtual channels, packet P_1 arriving later would be blocked until the transmission of P_0 has been completed. Assume now that physical channels implement two virtual channels. Upon arrival of P_1, the physical channel is multiplexed between them on a flit-by-flit bases and both packets proceed with half speed. Using of 2 virtual channels for the above situation, is based upon the below two assumptions-

- Assume that P_0 is a full-length packet whereas P_1 is only a small control

packet of size of few flits. Then this scheme allows P_1 pass through both routers while P_0 is slowed down for a short time corresponding to the transmission of few packets.

- Assume that P_0 is temporarily blocked downstream from the current router. Then P_1 can proceed at the full speed of the physical channel" [13].

Deadlock is fatal for an interconnection network. When resources (buffers or channels) are occupied by deadlocked packets, other packets are also blocked by these resources and finally paralyzes network operation. To prevent this situation, networks must either use deadlock avoidance or deadlock recovery. Almost all modern network use deadlock avoidance, usually by imposing an order for the resources and insisting that packets acquire these resources in order. On the other hand, one of the common method to avoid the deadlocks by using the virtual channels. Here, lemma 1 and lemma 2 describes the required number of virtual channels for various conventional networks.

"Lemma 1: If the message is routed in the z $\rightarrow$ y $\rightarrow$ x direction in a 3D-torus/3d-mesh network, then the network is deadlock free with 2 virtual channels" [3].

"Lemma 2: If the message is routed in the y $\rightarrow$ x direction in a 2D-torus network, then the network is deadlock free with 2 virtual channels" [5].

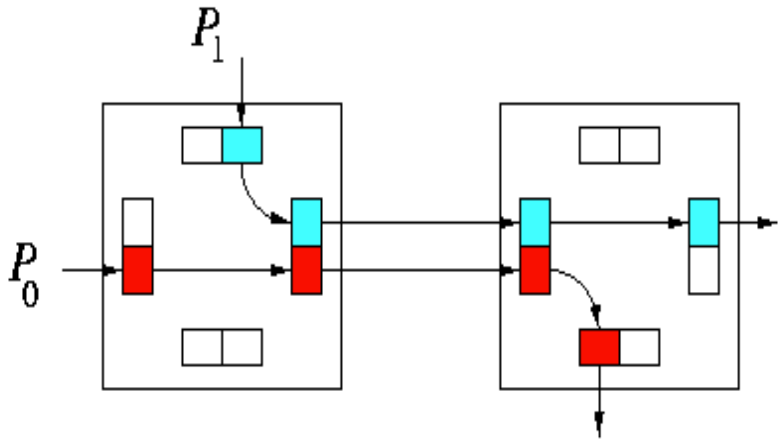


Figure 2.8: Packet transmission using 2 virtual channels[13]

In this research plan, we have also used 6 virtual channels for our dynamic communication performance. We have used Topaz NoC simulator [14] to simulate on-chip network performance with the 64 nodes. Please find the Appendix B for the virtual channel implementation code of Topaz simulator.

2.5 Summary

In this chapter, we have introduced few interconnection networks which is used for the performance comparison in this thesis. MESH and TORUS are conventional interconnection networks. Even today most of the modern super-computers uses conventional interconnection networks. Blue Gene/Q super-computer built by IBM corporation uses the 5D-Torus interconnection network. On the hand, HIN is the one of the possible solution for high cost-performance and low power consumption issues due to reduced number of outgoing link from different levels of hierarchies. Here, we have introduced three HIN networks as- TESH, TTN, STTN; their basic architecture and also explained the importance of 3D-TESH network in the field of next generation of supercomputers.

Chapter 3

Network Architecture

3.1 Introduction

Network topology refers to the static arrangement of inter-links and nodes in an interconnection network. Network topology plays a vital role in a multi-processor system because routing strategy and static network performance heavily depends on the network topology. Current MPCs with about millions of nodes already reached into 10 peta-flops performance and efforts been made for exa-scale performance. Network topology is the main factor for the next generation exa-scale system as the on-chip interconnection networks consume 50% of the total power and off-chip bandwidth is limited to the maximum number of possible out going links. In this chapter, we describe the architectural details of a new hierarchical interconnection network named as 3D-TESH network, higher level connection for 3D-TESH, the node addressing and finally the message routing algorithm for 3D-TESH network.

3.2 Architecture of 3D-TESH

Hierarchical interconnection networks (HINs) [15] are a cost-effective way to interconnect a large number of nodes. 3D-TESH is a HIN consists of basic modules (BMs) that are hierarchically interconnected for higher levels.

Definition: A 3D-TESH(m, L, q) network, by definition is built using 2^m number of 2D-TESH($2^m \times 2^m$) basic modules, where m is a positive integer, has L levels of hierarchy and q is used for the inter-level connectivity.

In 3D-TESH, a BM is similar to 3D-mesh network consists of ($2^m \times 2^m \times 2^m$) connected processing elements (PEs) having (2^m) rows and ($2^m \times 2^m$) columns. Figure 3.1 shows the BM for 3D-TESH with m = 2 which defines the size of the BM as ($4 \times 4 \times 4$). Similarly, if m = 3, then the size of the BM becomes ($8 \times 8 \times 8$) network with 512 nodes.

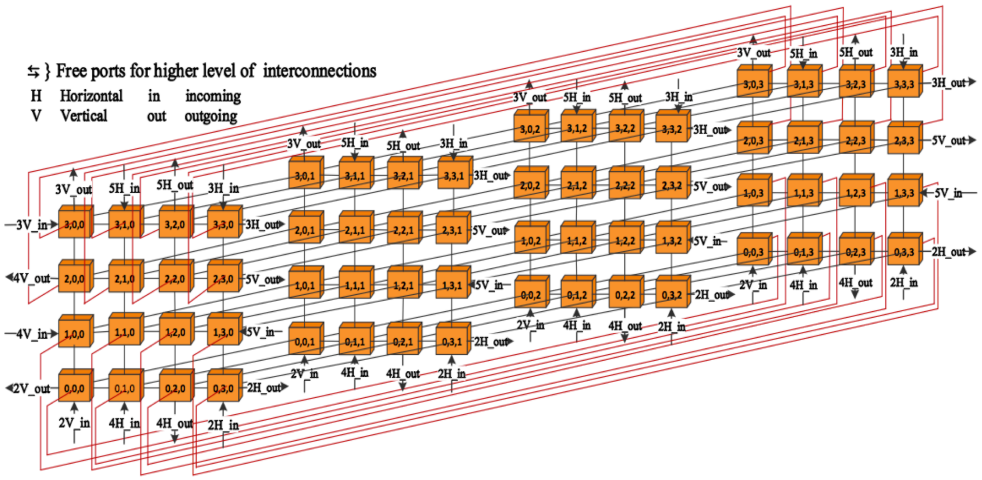


Figure 3.1: A ($4 \times 4 \times 4$) Basic Module of 3D-TESH Network

Lemma 1: Each ($2^m \times 2^m \times 2^m$) 3D-TESH BM has 2^{2m+2} free ports for its higher level of interconnections.

Lemma 1 defines the number of free ports for the higher level interconnections of 3D-TESH network. In 3D-TESH network for each higher level of interconnection, a BM uses $2^m \times 4 \times (2^q) = 2^{m+q+2}$ of its free links, where $2(2^{m+q})$ free links for vertical interconnections and $2(2^{m+q})$ free links for horizontal interconnections. Here, $q \in 0, 1, ..., m$ defined as the inter-level connectivity. q = 0 leads to the minimum inter-level connectivity, whereas q = m leads to the maximum inter-level connectivity. As from the figure 3.1, a ($4 \times 4 \times 4$) BM has $2^{2\times2+2} = 64$ free ports. if we choose m = 2 and q = 0, then $(2^{2+0+2}) = 16$ of the free ports and their associated links are used for each higher level interconnection, 8 for horizontal and 8 for vertical connection. However, if the value of L is 2 or 3, the number of vertical_in, vertical_out, horizontal_in, horizontal_out connections are more than four, this means that the inter-level connectivity (q) is increasing. Similarly, if m

= 2 and L = 3, then 8 links can be used for each vertical_in, vertical_out, horizontal_in, horizontal_out connections, when q = 1. Figure 3.2 shows the 3D-TESH(2,3,1) interconnections.

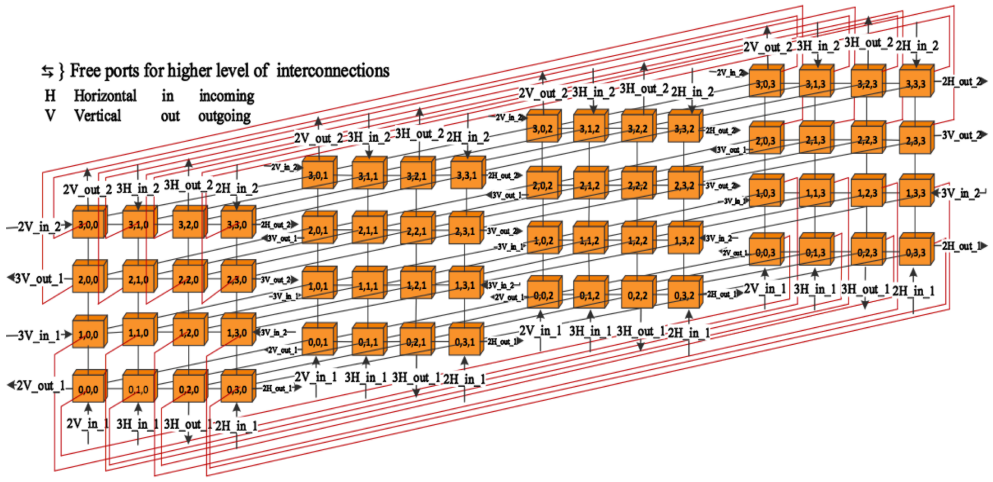


Figure 3.2: A (4 × 4 × 4) Basic Module of 3D-TESH(2,3,1) Network

3.3 Higher-level Networks of 3D-TESH

Higher level of 3D-TESH network is built by the recursive interconnection of the immediate lower level of sub-networks. Figure 3.3 shows the higher-level interconnection for 3D-TESH. A Level-2 network can be formed by $(2^{2\times2})$ 16BMs (16 level-1 3D-TESH network). Similarly, a level-3 3D-TESH network can be formed by interconnecting 16 level-2 sub-networks. As we have considered m is 2, then the number nodes at level-2 network can be defined by $N = (2^{2mL} \times 2^m)$. Hence number of nodes at level-2 network is $N = (2^8 \times 2^2) = 1024$.

The maximum level for 3D-TESH can be built by a $(2^m \times 2^m \times 2^m)$ BM is $L_{max} = 2^{m-q} + 1$. If inter-level connectivity, q = 0 & m = 2, L_{max}= 5; Level-5 is the highest possible level. The maximum number of nodes in each level of network can be defined as $N = (2^{2mL} \times 2^m)$. If m = 2 & L = 2, then N = 1024. Similarly, a level-3 3D-TESH network will be consists of $N = 2^{2\times2\times3} \times 4 = 16384$ nodes, which is equal to 16 level-2 3D-TESH networks. Table 3.1 generalizes the various parameters of 3D-TESH

Network, whereas table 3.2 compares the various levels of 3D-TESH network for m = 2 with the different inter-connectivity.

Table 3.1: Generalization of 3D-TESH Network

Basic Module	inter-connectivity	Max Levels	Number of Nodes
$(2^m \times 2^m \times 2^m)$	q	$L_{max} = 2^{m-q} + 1$	$N_L = (2^{2mL} \times 2^m)$

Table 3.2: Example of various levels of 3D-TESH network

m	inter-connectivity, q	Max Levels	Number of Nodes
2	0	$L_{max} = 2^{2-0} + 1 = 5$	$N_1 = (2^{2\times2\times1} \times 2^2) = 64$ $N_2 = (2^{2\times2\times2} \times 2^2) = 1024$ $N_3 = (2^{2\times2\times3} \times 2^2) = 16384$ $N_4 = (2^{2\times2\times4} \times 2^2) = 262144$ $N_5 = (2^{2\times2\times1} \times 2^2) = 4194304$
2	1	$L_{max} = 2^{2-1} + 1 = 3$	$N_1 = (2^{2\times2\times1} \times 2^2) = 64$ $N_2 = (2^{2\times2\times2} \times 2^2) = 1024$ $N_3 = (2^{2\times2\times3} \times 2^2) = 16384$
2	2	$L_{max} = 2^{2-2} + 1 = 2$	$N_1 = (2^{2\times2\times1} \times 2^2) = 64$ $N_2 = (2^{2\times2\times2} \times 2^2) = 1024$

3.4 Number of Channels at Various Levels of 3D-TESH

Larger scaling for supercomputers can make the total cable length enormous e.g., up to thousands of kilometers. Recent high-radix switches with dozens of ports make switch layout and system packaging more complex [16]. The recent K-computer requires cable length near about one thousand kilometers. This section of the paper defines the required number of inter-links between the different layers of 3D-TESH network and also compares result with other networks. Next generation interconnection network requires massive inter-connection between the chips, nodes and even between the cabinet chassis.

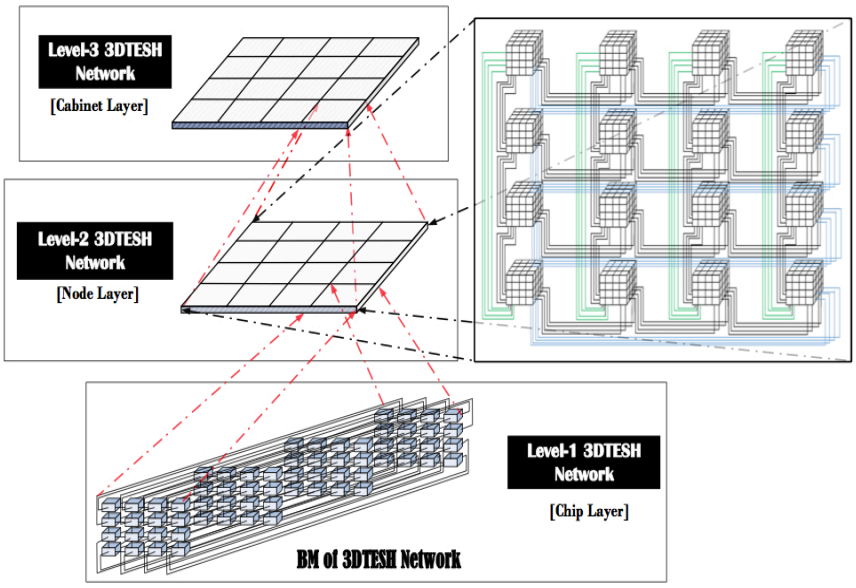


Figure 3.3: Higher-level of Interconnection for 3D-TESH Network

Hence the total cable length has become an important factor for designing the next generation supercomputers. Even also interconnection networks have a high impact on the total power consumption of a single MPC system. Due to this reason increase of inter- links between different levels of networks will also increase the total power consumption. As an example- K-computer requires the highest total power consumption of any 2011 TOP500 supercomputer (9.89 MW the equivalent of almost 10,000 suburban homes) with 80,000 (2.0GHz 8- core) SPARC64 VIIIfx processors contained in 864 cabinets, for a total of over 640,000 cores; [17]. On the other hand, hierarchical interconnection networks maintain a very small number of inter-links between the different layers of network. Figure 3.2 shows the hierarchical structure of 3D-TESH network, where level-1 network is defined as the chip layer, level-2 is the node layer, level-3 is the cabinet layer and so on for the higher levels. The various levels of interconnections for 3D-TESH network can be defined by the below equation-

$$L_N = (\text{Number of BM in current level, } L_N) \times \{\text{Number of inner links in } L_1 \text{ (160 links for m = 2) network}\} + \sum_{i=2}^{N} \{(\text{Number of BM at current level, } L_N) \times (\text{Number of outgoing links at each higher-level, } L_i)\} \quad [\text{where N} \geq 2] \tag{3.1}$$

Here, equation 3.1 defines required number of inter-connected links for various levels of networks. The total number of used links for level-1 3D-TESH(2, 1, 0) network is 160, whereas the number of BM for level-1 3D-TESH network is one. This equation has also been generalizes with the required number of links for various networks at table 3.3. On the other hand, table 3.4 shows the example of required number of interconnecting links for various levels of 3D-TESH network, in which cabinet layer the required number of links for 3D-TESH is above 45k. Table 3.4 also shows the link comparison on various networks like- 2D-MESH, 2D-TORUS, 3D-MESH and 3D-TORUS against the 3D-TESH network. The network size for level-1 network has been considered with 64 nodes, level-2 network is having 1024 nodes and level-3 network is consists of 16384 nodes. And from this table we can find that 3D-MESH and 3D-TORUS network require more number of links than the 3D-TESH network at the higher levels. This table also shows that 3D-TESH network requires much more number of outgoing links than the two dimensional networks.

Table 3.3: Generalization of number of links at various levels of 3D-TESH

Topology	Level-1 Network	Higher Level Network
3D-TESH	(Number of X-directional Links) + (Number of Y-directional Links) + (Number of Z-directional Links)	(Number of BM in current level) × (Number of inner links in level-1 network) + $\sum_{i=2}^{N}$ { (Number of BM at current level, L_N) × (Number of outgoing links at each higher-level, L_i)} [where N $\geq$ 2]

Table 3.4: Example of required number of links at various levels of networks

Topology	Level-1 Network (64 Nodes)	Level-2 Network (1024 Nodes)	Level-3 Network (16384 Nodes)
3D-TESH (m = 2, L, q = 0)	160 links	16 x 160 (L1 Links) + 128 (L2 Links) = 2688	256 x 160 (L1 Links) + 2048 (L2 Links) + 2048 (L3 Links) = 45056
2D-MESH	112 links	16 x 112 (L1 Links) + 192 (L2 Links) = 1984	256 x 112 (L1 Links) + 3072 (L2 Links) + 768 (L3 Links) = 32512
2D-TORUS	128 links	16 x 112 (L1 Links) + 256 (L2 Links) = 2048	256 x 112 (L1 Links) + 3072 (L2 Links) + 1024 (L3 Links) = 32768
3D-MESH	144 links	16 x 144 (L1 Links) + 448 (L2 Links) = 2752	256 x 144 (L1 Links) + 7168 (L2 Links) + 2816 (L3 Links) = 46848
3D-TORUS	192 links	16 x 144 (L1 Links) + 768 (L2 Links) = 3072	256 x 144 (L1 Links) + 7168 (L2 Links) + 5120 (L3 Links) = 49152

3.5 Addressing of Nodes

Nodes in a BM are addressed by three digits; the first is the Y-index, then the X-index and finally the Z-index. In general, in a Level-L 3D-TESH, the node address can be represented by:

$$A^L = \begin{cases} (a_{yL},\, a_{xL},\, a_{zL}) & \text{if } L = 1 \\ \\ (a_{yL},\, a_{xL}) & \text{if } L_{max} \geq L \geq 2 \end{cases}$$

More generally in a Level-L 3D-TESH, the node address is represented by-

$$\begin{aligned} A &= A^L\ A^{L-1}\ A^{L-2}\ ...\ ...\ ...\ A^2\ A^1 \\ &= (a_{2L},\, a_{2L-1})\ (a_{2L-2},\, a_{2L-3})...\ ...\ (a_4,\, a_3)\ (a_2,\, a_1,\, a_0) \end{aligned} \tag{3.2}$$

Here, the Level-1 is defined by node address $(a_2,\, a_1,\, a_0)$, where a_0 defines the node address for Z-axis and then followed by the X-axis and then the Y-axis. Level-2 to Level-5 Networks are two dimensional networks, so first digit defines the row index and then the next is the column index. Now if the address of a node n^1 included in BM_1 is represented as $n^1 = [(s_{2L},\, s_{2L-1})...\ ...(s_4,\, s_3)\ (s_2,\, s_1,\, s_0)]$ and the address of a node n^2 included in BM_2 is represented as $n^2 = [(d_{2L},\, d_{2L-1})...\ ...(d_4,\, d_3)\ (d_2,\, d_1,\, d_0)]$. The node n^1 in BM_1 and n^2 in BM_2 are connected if the following connections are satisfied for n^2(when m = 2, q = 0)-

- **Link for BM-**

$[(s_{2L},\, s_{2L-1})...\ ...(s_4,\, s_3)\ (s_2,\, s_1,\, s_0)]$ to $[(s_{2L},\, s_{2L-1})...\ ...(s_4,\, s_3)\ (s_2 \pm 1,\, s_1 \pm 1,\, s_0 \pm 1 \bmod\ 2^m)]$
where $2^m - 1 > s_2 > 0$, $2^m - 1 > s_1 > 0$, $2^m > s_0 \geq 0$, not both $s_1, s_2\ \pm$

$[(s_{2L},\, s_{2L-1})...\ ...(s_4,\, s_3)\ (s_2,\, s_1,\, s_0)]$ to $[(s_{2L},\, s_{2L-1})...\ ...(s_4,\, s_3)\ (s_2 + 1,\, s_1,\, s_0 \pm 1 \bmod\ 2^m)]$ where $s_2 = 0$, $2^m > s_1, s_0 \geq 0$

$[(s_{2L},\, s_{2L-1})...\ ...(s_4,\, s_3)\ (s_2,\, s_1,\, s_0)]$ to $[(s_{2L},\, s_{2L-1})...\ ...(s_4,\, s_3)\ (s_2,\, s_1 + 1,\, s_0 \pm 1 \bmod\ 2^m)]$ where $s_1 = 0$, $2^m > s_2, s_0 \geq 0$

$[(s_{2L},\, s_{2L-1})...\ ...(s_4,\, s_3)\ (s_2,\, s_1,\, s_0)]$ to $[(s_{2L},\, s_{2L-1})...\ ...(s_4,\, s_3)\ (s_2,\, s_1 - 1,\, s_0 \pm 1 \bmod\ 2^m)]$ where $s_1 = 2^m - 1$, $2^m > s_2, s_0 \geq 0$

$[(s_{2L}, s_{2L-1})... \ ...(s_4, s_3)\ (s_2, s_1, s_0)]$ to $[(s_{2L}, s_{2L-1})... \ ...(s_4, s_3)\ (s_2 - 1, s_1, s_0 \pm 1 \bmod 2^m)]$ where $s_2 = 2^m - 1$, $2^m > s_1, s_0 \geq 0$

- **Link for L2_Vertical-** $[(s_{2L}, s_{2L-1})... \ ...(s_4, s_3)\ (0, 0, s_0)]$ to $[(s_{2L}, s_{2L-1})... \ ...(s_4 \pm 1 \bmod 2^m, s_3)\ (0, 0, s_0)]$
- **Link for L2_Horizontal-** $[(s_{2L}, s_{2L-1})... \ ...(s_4, s_3)\ (0, 2^m - 1, s_0)]$ to $[(s_{2L}, s_{2L-1})... \ ...(s_4, s_3 \pm 1 \bmod 2^m)\ (0, 2^m - 1, s_0)]$
- **Link for L3_Vertical-** $[(s_{2L}, s_{2L-1})... \ ...(s_6, s_5)\ ...\ (2^m - 1, 0, s_0)]$ to $[(s_{2L}, s_{2L-1})... \ ...(s_6 \pm 1 \bmod 2^m, s_5)\ ...\ (2^m - 1, 0, s_0)]$
- **Link for L3_Horizontal-** $[(s_{2L}, s_{2L-1})... \ ...(s_4, s_3)\ (2^m - 1, 2^m - 1, s_0)]$ to $[(s_{2L}, s_{2L-1})... \ ...(s_6, s_5 \pm 1 \bmod 2^m)\ ...\ (2^m - 1, 2^m - 1, s_0)]$
- **Link for L4_Vertical-** $[(s_{2L}, s_{2L-1})... \ ...(s_8, s_7)\ ...\ (2, 0, s_0)]$ to $[(s_{2L}, s_{2L-1})... \ ...(s_8 \pm 1 \bmod 2^m, s_7)\ ...\ (1, 0, s_0)]$
- **Link for L4_Horizontal-** $[(s_{2L}, s_{2L-1})... \ ...(s_8, s_7)\ ...\ (0, 2, s_0)]$ to $[(s_{2L}, s_{2L-1})... \ ...(s_8, s_7 \pm 1 \bmod 2^m)\ ...\ (0, 1, s_0)]$
- **Link for L5_Vertical-** $[(s_{10}, s_9)... \ ...(s_4, s_3)\ (2, 3, s_0)]$ to $[(s_{10}, s_9)... \ ...(s_4, s_3)\ (1, 3, s_0)]$
- **Link for L5_Horizontal-** $[(s_{10}, s_9)... \ ...(s_4, s_3)\ (3, 2, s_0)]$ to $[(s_{10}, s_9)... \ ...(s_4, s_3)\ (3, 1, s_0)]$

Similarly, we can define various level of interconnections as above when q = 1 or q = 2 with vertical and horizontal connections. The highest level of network which can be obtained by a $(2^m \times 2^m \times 2^m)$ BM is defined by $L_{max} = 2^{m-q} + 1$. When m = 2 and q = 0, $L_{max} = 5$; level-5 is the maximum possible level. In the rest of the paper, we have considered m = 2 and interconnectivity q = 0; therefore, we focus on the class of 3D-TESH(2, L, 0).

3.6 Routing Algorithm for 3D-TESH

A simple deterministic, dimension-order routing (DOR) algorithm has been considered for 3D-TESH network. DOR routes a packet continuously in each dimension until the distance of that dimension is zero, then forwards to the

next dimension. Routing of messages for 3D-TESH is performed from top to bottom similar to TESH [5] network. For every message the routing for 3D-TESH network completes at the highest level of network first; after that the packet reaches its highest level sub-destination, routing continues with the sub-network to the next lower level sub-destination. This process is repeated until the packet arrives at its final destination BM and then completes level-1 routing at the destination BM. When a packet is generated at a source node, the node checks its destination. If the packet 's destination is the current BM, the routing is performed with the BM only. If the packet is destined to another BM, the source node sends the packet to the outlet node which connects the BM to the level at which the routing is performed. Let a source node address is s = (s_{2L}, s_{2L-1}) (s_{2L-2}, s_{2L-3}) ... ... (s_4, s_3) (s_2, s_1, s_0) and destination node d = (d_{2L}, d_{2L-1}) (d_{2L-2}, d_{2L-3}) .. ... (d_4, d_3) (d_2, d_1, d_0) considering the routing at the Y,X direction for the higher levels and Y, X, Z direction for the level-1 networks. Similarly, routing tag can be defined as t = (t_{2L}, t_{2L-1}) (t_{2L-2}, t_{2L-3})... ...(t_4, t_3) (t_2, t_1, t_0), where $t_i = d_i$ - s_i. Algorithm 1 shows the routing algorithm for 3D-TESH network, whereas algorithm 2 & 3 defines the function outlet_x, outlet_y, $receiving\_node_x$ and $receiving\_node_y$.

The function $SP\_routing$ is used to find the route direction from the source BM towards the destination BM. On the other hand, outlet_x and outlet_y are the function to get x coordinate s_1 and y coordinate s_2 of the node that link (s, d, l, dα) exists, where level l($2 \leq$ l $\leq$ L), dimension d(d $\in$ {V,H}) and direction α($\alpha \in \{+,-\}$). Hence vertical and horizontal direction are represented by V+, V-, H+ and H-. Please find the Appendix A for implemented routing code for 3D-TESH(2, L, 0) network.

Algorithm 1 Routing Algorithm for 3D-TESH Network

Routing 3D-TESH(s_{2L}, s_{2L-1}, s_{2L-2}, ……, s_1, s_0, d_{2L}, d_{2L-1}, d_{2L-2}, ……, d_1, d_0);
tag: t_{2L}, t_{2L-1}, t_{2L-2}, ……, t_1, t_0;
for i = 2L : 3;
routedir = $SP\_routing(s, d, \lfloor (i-1)/2+1 \rfloor, i)$;
if (routedir = positive) then t_i = ((d_i - s_i + 2^m) mod 2^m);
else t_i = (2^m - (d_i - s_i + 2^m) mod 2^m); endif;
while ($t_i \neq 0$) do
if (i mod 2) = 1, then $outlet\_node_x = outlet\_x(s, d, \lfloor (i-1)/2+1 \rfloor, H, routedir)$;
$outlet\_node_y = outlet\_y(s, d, \lfloor (i-1)/2+1 \rfloor, H, routedir)$;
else $outlet\_node_x = outlet\_x(s, d, \lfloor (i-1)/2+1 \rfloor, V, routedir)$;
$outlet\_node_y = outlet\_y(s, d, \lfloor (i-1)/2+1 \rfloor, V, routedir)$; endif;
$BM\_routing(s_2, s_1, 0, outlet\_node_y, outlet\_node_x, 0)$;
if (routedir = positive) then send the packet to the next BM;
else move the packet to previous BM; endif;
if ($t_i > 0$) then $t_i = t_i$ - 1; endif;
if ($t_i < 0$) then $t_i = t_i$ + 1; endif;
if (i mod 2) = 1, $s_1 = receiving\_node_x(s, d, \lfloor (i-1)/2+1 \rfloor, H, routedir)$;
$s_2 = receiving\_node_y(s, d, \lfloor (i-1)/2+1 \rfloor, H, routedir)$;
else $s_1 = receiving\_node_x(s, d, \lfloor (i-1)/2+1 \rfloor, V, routedir)$;
$s_2 = receiving\_node_y(s, d, \lfloor (i-1)/2+1 \rfloor, V, routedir)$; endif;
endwhile;
endfor;
$BM\_routing(s_2, s_1, s_0, d_2, d_1, d_0)$;
end
$BM\_routing(s_2, s_1, s_0, d_2, d_1, d_0)$;
source: s_2, s_1, s_0; destination: d_2, d_1, d_0;
BM_tag: t_2, t_1, t_0 = destination address(d_2, d_1, d_0) - source address(s_2, s_1, s_0);
if ($t_0 > 0$ and $t_0 \leq 2^{m-1}$) or ($t_0 < 0$ and t_0 = - (2^m - 1)), movedir = positive; endif;
if ($t_0 > 0$ and t_0 = (2^m - 1)) or ($t_0 < 0$ and $t_0 \geq$ -2^{m-1}), movedir = negative; endif;
if (movedir = positive and $t_0 > 0$) then $t_0 = t_0$; endif;
if (movedir = positive and $t_0 < 0$) then t_0 = m + t_0; endif;
if (movedir = negative and $t_0 < 0$) then $t_0 = t_0$; endif;
if (movedir = negative and $t_0 > 0$) then t_0 = -m + t_0; endif;
while($t_0 \neq 0$) do
if ($t_0 > 0$) then move packet to +z node; $t_0 = t_0$ - 1; endif;
if ($t_0 < 0$) then move packet to -z node; $t_0 = t_0$ + 1; endif;
endwhile;
while($t_1 \neq 0$) do
if ($t_1 > 0$) then move packet to +x node; $t_1 = t_1$ - 1; endif;
if ($t_1 < 0$) then move packet to -x node; $t_1 = t_1$ + 1; endif;
endwhile;
while($t_2 \neq 0$) do
if ($t_2 > 0$) then move packet to +y node; $t_2 = t_2$ - 1; endif;
if ($t_2 < 0$) then move packet to -y node; $t_2 = t_2$ + 1; endif;
endwhile;
end
$SP\_routing(s, d, Level, i)$;
if ((d_i - s_i + 2^m) mod 2^m) > $2^m/2$, then routedir = negative;
elseif ((d_i - s_i + 2^m) mod 2^m) = $2^m/2$, {
if (Level mod 2) = 0, then
if (i mod 2) = 0, then routedir = positive;
else routedir = negative; endif;
else
if (i mod 2) = 0, then routedir = negative;
else routedir = positive; endif; }
else routedir = positive; endif;
end

Algorithm 2 function definition for 3D-TESH Network

$outlet\_x(s, d, L, VH, rd)$; //This function is applicable for only 3D-TESH(m, L, 0)
if (VH = V and rd = positive) then {
if (L = 2) then $outlet\_node_x = 0$; elseif (L = 3) then $outlet\_node_x = 0$;
elseif ((L mod 2) = 0) then $outlet\_node_x = 0$;
elseif ((L mod 2) $\neq$ 0) then $outlet\_node_x = 2^m - 1$; else "Error"; }
if (VH = H and rd = positive) then {
if (L = 2) then $outlet\_node_x = 2^m - 1$; elseif (L = 3) then $outlet\_node_x = 2^m - 1$;
elseif ((L mod 2) = 0) then $outlet\_node_x$ = L - 2;
elseif ((L mod 2) $\neq$ 0) then $outlet\_node_x$ = L - 3; else "Error"; }
if (VH = V and rd = negative) then {
if (L = 2) then $outlet\_node_x = 0$; elseif (L = 3) then $outlet\_node_x = 0$;
elseif ((L mod 2) = 0) then $outlet\_node_x = 0$;
elseif ((L mod 2) $\neq$ 0) then $outlet\_node_x = 2^m - 1$; else "Error"; }
if (VH = H and rd = negative) then {
if (L = 2) then $outlet\_node_x = 2^m - 1$; elseif (L = 3) then $outlet\_node_x = 2^m - 1$;
elseif ((L mod 2) = 0) then $outlet\_node_x$ = L - 3;
elseif ((L mod 2) $\neq$ 0) then $outlet\_node_x$ = L - 4; else "Error"; }
end
$outlet\_y(s, d, L, VH, rd)$; //This function is applicable for only 3D-TESH(m, L, 0)
if (VH = V and rd = positive) then {
if (L = 2) then $outlet\_node_y = 0$; elseif (L = 3) then $outlet\_node_y = 2^m - 1$;
elseif ((L mod 2) = 0) then $outlet\_node_y$ = L-2;
elseif ((L mod 2) $\neq$ 0) then $outlet\_node_y$ = L - 3; else "Error"; }
if (VH = H and rd = positive) then {
if (L = 2) then $outlet\_node_y = 0$; elseif (L = 3) then $outlet\_node_y = 2^m - 1$;
elseif ((L mod 2) = 0) then $outlet\_node_y = 0$;
elseif ((L mod 2) $\neq$ 0) then $outlet\_node_y = 2^m - 1$; else "Error"; }
if (VH = V and rd = negative) then {
if (L = 2) then $outlet\_node_y = 0$; elseif (L = 3) then $outlet\_node_y = 2^m - 1$;
elseif ((L mod 2) = 0) then $outlet\_node_y$ = L - 3;
elseif ((L mod 2) $\neq$ 0) then $outlet\_node_y$ = L - 4; else "Error"; }
if (VH = H and rd = negative) then {
if (L = 2) then $outlet\_node_y = 0$;
elseif (L = 3) then $outlet\_node_y = 2^m - 1$;
elseif ((L mod 2) = 0) then $outlet\_node_y = 0$;
elseif ((L mod 2) $\neq$ 0) then $outlet\_node_y = 2^m - 1$; else "Error"; }
end
$receiving\_node_x(s, d, L, VH, rd)$; //This function is applicable for only 3D-TESH(m, L, 0)
if (VH = V and rd = positive) then {
if (L = 2) then $receiving\_nodex = 0$; elseif (L = 3) then $receiving\_nodex = 0$;
elseif ((L mod 2) = 0) then $receiving\_nodex = 0$;
elseif ((L mod 2) $\neq$ 0) then $receiving\_nodex = 2^m - 1$; else "Error"; }
if (VH = H and rd = positive) then {
if (L = 2) then $receiving\_nodex = 2^m - 1$; elseif (L = 3) then $receiving\_nodex = 2^m - 1$;
elseif ((L mod 2) = 0) then $receiving\_nodex$ = L - 3; elseif ((L mod 2) $\neq$ 0) then $receiving\_nodex$ = L - 4; else "Error"; }
if (VH = V and rd = negative) then {
if (L = 2) then $receiving\_nodex = 0$; elseif (L = 3) then $receiving\_nodex = 0$;
elseif ((L mod 2) = 0) then $receiving\_nodex = 0$; elseif ((L mod 2) $\neq$ 0) then $receiving\_nodex$ = $2^m - 1$; else "Error"; }
if (VH = H and rd = negative) then {
if (L = 2) then return $receiving\_nodex = 2^m - 1$; elseif (L = 3) then return $receiving\_nodex$ = $2^m - 1$;
elseif ((L mod 2) = 0) then $receiving\_nodex$ = L - 2; elseif ((L mod 2) $\neq$ 0) then $receiving\_nodex$ = L - 3; else "Error"; }
end

Algorithm 3 function definition for 3D-TESH Network

$receiving\_node_y(s, d, L, VH, rd)$; //THIS function is applicable for only 3D-TESH(m, L, 0)
if (VH = V and rd = positive) then {
if (L = 2) then return $receiving\_nodey = 0$; elseif (L = 3) then return $receiving\_nodey = 2^m - 1$;
elseif ((L mod 2) = 0) then $receiving\_nodey$ = L - 3; elseif ((L mod 2) $\neq$ 0) then $receiving\_nodey$ = L - 4; else "Error"; }
if (VH = H and rd = positive) then {
if (L = 2) then $receiving\_nodey = 0$; elseif (L = 3) then $receiving\_nodey = 2^m - 1$;
elseif ((L mod 2) = 0) then $receiving\_nodey = 0$; elseif ((L mod 2) $\neq$ 0) then $receiving\_nodey = 2^m - 1$; else "Error"; }
if (VH = V and rd = negative) then {
if (L = 2) then $receiving\_nodey = 0$; elseif (L = 3) then $receiving\_nodey = 2^m - 1$;
elseif ((L mod 2) = 0) then $receiving\_nodey$ = L - 2; elseif ((L mod 2) $\neq$ 0) then $receiving\_nodey$ = L - 3; else "Error"; }
if (VH = H and rd = negative) then {
if (L = 2) then $receiving\_nodey = 0$; elseif (L = 3) then $receiving\_nodey = 2^m - 1$;
elseif ((L mod 2) = 0) then $receiving\_nodey = 0$; elseif ((L mod 2) $\neq$ 0) then $receiving\_nodey = 2^m - 1$; else "Error"; }
end

To understand the routing path for 3D-TESH network using the 3D-TESH DOR routing algorithm we have considered figure 3.4 where the source node is (1,2),(1,2),(1,2,0) and the destination node is (2,1),(2,1),(2,1,0). At first routing will be done at Level-3 network, the source node will send the packet to the outlet node (1,2),(1,2),(3,0,0) of Level-3 network and will reach Level-3(2,2) from Level-3(1,2) network. Similarly, it will reach Level-3(2,1) network. Then, Level-2(1,2) routing will be started and will reach Level-2(2,1) network. After that in Level-1 network packet will reach the destination Node(2,1,0) from the destination BM Node(0,3,0).

3.7 Summary

In recent, a large number of interconnection networks have been proposed to minimize the cost and to maximize the performance. Hierarchical interconnection networks are ahead of others. In this chapter, we have described about the architecture of 3D-TESH network. The addressing, message routing of 3D-TESH and the routing algorithm of 3D-TESH were described in details. Higher level of 3D-TESH is maintained by the immediate lower level of subnetworks. Higher level interconnection of 3D-TESH is also described in this chaptor.

Figure 3.4: Routing Path for 3D-TESH Network

Chapter 4

Static Network Performance Evaluation

4.1 Introduction

The topology of interconnection network affects the performance metrics. Performance metrics can be used to evaluate and compare different network topologies. It is being expected from the interconnection network with low cost, low degree, low congestion, high connectivity, and high fault tolerance than the other networks. In this chapter, we will compare the static network performance like- the node degree, diameter, average distance, cost performance, the bisection bandwidth and arc connectivity of various interconnection networks against the 3D-TESH network.

4.2 Comparison of Static Performance of Various Networks

4.2.1 Node Degree

The node degree is defined as the maximum number of physical outgoing links from a node. Since each node of 3D-TESH network has maximum six outgoing links, the degree of 3D-TESH is 6. Constant node degree networks are easy to expand and the network interface cost remains unchanged with increasing network size. The I/O interface cost of a particular node is proportional to its degree. Table 4.1 shows the node degree for the various

networks.

Table 4.1: Node Degree of Various Networks

Parameter	2D-Mesh	2D-Torus	TESH Network	3D-TESH
Node Degree	4	4	4	6

4.2.2 Diameter Performance

A node must follow a communication path to transmit data to other node which are not directly connected. Increase of path length increases the communication delay. Shortest path is desirable. The diameter of a network is the maximum inter-node distance i.e., the maximum number of links that must be traversed to send a message to any node along the shortest path. Diameter is the maximum distance between all distinct pairs of nodes along the shortest path. Network with small diameter is preferable. If the diameter is preferable it will take less time to route a packet. Diameter is a common approach to compare the static network performance of a network topology. It has been shown from the figure 4.1 that the 3D-TESH has the diameter less than 2D-MESH, 2D-TORUS, TESH networks for any hierarchical level of network.

Diameter for 3D-TESH can also be evaluated using the below equation-

Diameter = maximum value to make the Z-directional routing + maximum value to move to the highest level of outgoing node + then make the highest level routing + maximum value to go to the next level routing out going node + make the next level routing + this loop will continue as it moves to the level-2 + from level-2 incoming node to the destination node.

For the 3D-TESH network, an upper bound for the diameter is given by-

$$Diameter = max(D_z + D_s + (\sum_{i=2}^{L}(D_{si} + D_i)) + D_d) \tag{4.1}$$

Here, where D_z is the value to move to Z-directions. D_s is to move to the

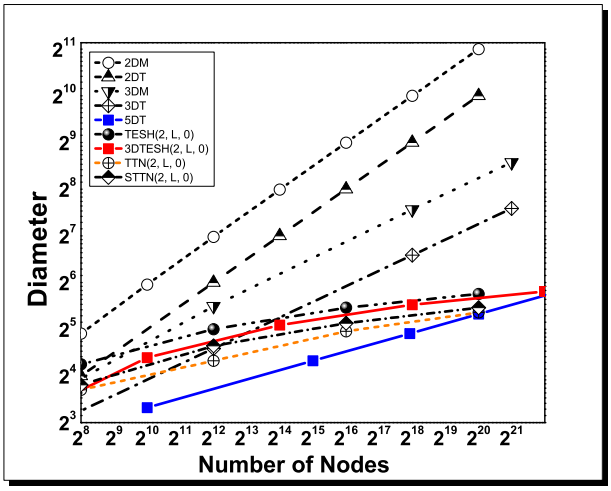


Figure 4.1: Diameter Performance for Various Networks

highest level of outgoing node. D_{si} is the value to go to the next level of routing and D_i is the corresponding level of routing. D_d is the value from level-2 to destination node. Table 4.2 shows the calculated formulation for 3D-TESH network-

4.2.3 Average Distance

It is not always preferable to compare the network performance against only the diameter because a node has to communicate with others; hence on an average, shorter path than the lower diameter is being expected. The average distance is the mean distance between all distinct pairs of nodes in a network. Small average distance is preferable which allows small communication latency. Figure 4.2 shows the average distance for various networks.

4.2.4 Cost Performance

Inter-node distance, message traffic density, and fault- tolerance are dependent on the diameter and the node degree. Hence the product of node degree and diameter is useful for measuring the relationship between cost and performance of a multiprocessor system. Figure 4.3 shows the total cost of TESH

Table 4.2: Calculated Formulation of Diameter for 3D-TESH Network.

Parameter	Result of D_z	Result of D_s	Result for D_{si} and D_i	Result for D_d	Diameter of the corresponding level
Level-1 network	2	6	$D_{si} = 0$, $D_i = 0$	0	8
Level-2 network	2	6	for i = 2; D_{si}=0, D_i=7	6	21
Level-3 network	2	6	for i = 2; D_{si}=0, D_i=7 for i = 3; D_{si}=0, D_i=7	6	34

and 3D-TESH with respect to other networks. TESH shows lower cost than 3D-TESH, as the node degree of 3D-TESH is greater than TESH.

4.2.5 Bisection Bandwidth

The Bisection Bandwidth (BW) of a network is defined as the minimum number of links that must be removed to partition the network into two equal halves. Small bisection bandwidth implies low bandwidth between the two halves and it can slow down the final merging phase. A large bisection bandwidth is undesirable for the VLSI design of the interconnection network, since it implies a lot if extra chip wires. Figure 4.4 shows the bisection bandwidth of the various networks.
The bisection bandwidth for 3D-TESH(m, L, q) is given by:

$$\begin{aligned} BW_{3D-TESH(m,L,q)} &= 2^m \times 2^{2m(L-1)-1} \times 2^{m+1} \\ &= 2^{m(2L-3)+1} \times 2^m \ [\ \mathrm{L} \geq 2\ ] \end{aligned} \tag{4.2}$$

4.2.6 Arc Connectivity

Arc Connectivity measures the robustness of a network. It is a measure of the multiplicity of paths between the processors. Arc connectivity is the minimum number of links that must be removed in order to break the network

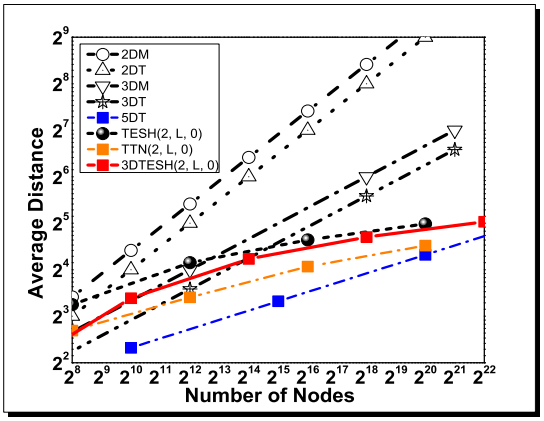


Figure 4.2: Average Distance for Various Networks

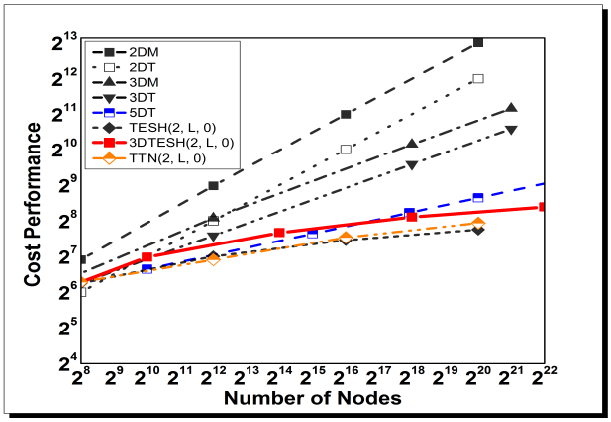


Figure 4.3: Cost Performance for Various Networks

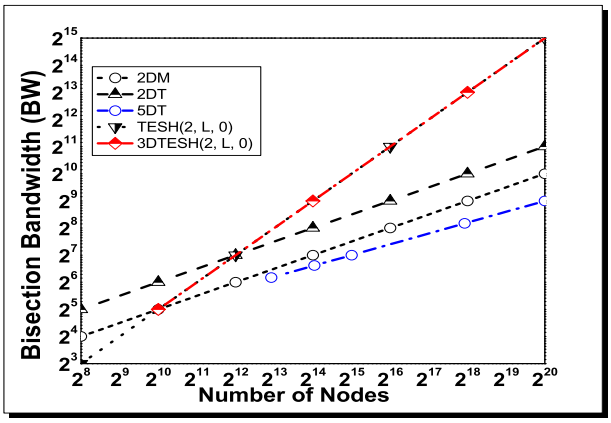


Figure 4.4: Bisection Bandwidth for Various Networks

into two disjoint parts. High arc connectivity improves performance during normal operation by avoiding link congestion and improved fault tolerance. A network is maximally fault tolerant if its connectivity is equal to the degree of that network. Table 4.3 shows arc connectivity of various networks.

Table 4.3: Arc Connectivity for Various Networks

Parameter	2D-Mesh	2D-Torus	TESH Network	3D-TESH
Node Degree	4	4	4	6
Arc Connectivity	2	4	2	4

4.3 Summary

The above six parameters are normally used to determine the static performance of a network. Diameter and average distance is the most important consideration for the evaluation of static network performance. Because lower diameter of a network takes shorter time to send a message from source node to destination node. Similarly if the average distance is short, the static network performance would be improved. Cost of a network is totally belongs to node degree. As the node degree of 3D-TESH is high, the cost performance of 3D-TESH is greater than the two dimensional TESH network.

Chapter 5

Dynamic Communication Performance Evaluation

5.1 Introduction

Bad performance of the communicational network will severely limit the speed and efficiency of the entire MPC system. The dynamic communication performance (DCP) of an interconnection network is characterized by latency and throughput. Message latency is the time required for a packet to traverse the network from source to destination. Therefore, it refers to the time elapsed from the instant when the last flit of the message is received at the destination. Latency can be described as:

$$T = T_h + L/b, T_s = L/b \tag{5.1}$$

Here, the head latency T_h, is the time required for the head of the message to traverse the network. The serialization latency T_s, is the time required for a packet of length L to cross a channel with bandwidth b. Network throughput is the rate at which packets are delivered by the network for a particular traffic pattern. It refers to the maximum amount of information delivered per unit of time through the network.

5.2 Estimation of Power Consumption

Power dissipation is a major concern for the next generation of supercomputers. The k-computer with Tofu interconnect requires about 9.89MW of

electrical power to run the complete system [4]. The power model for 3D-TESH can be defined by 2-ways; one is on-chip power model and other is the off-chip power model.

The on-chip power model is based on the Orion energy model [18] using 65nm fabrication process, considering the static and dynamic power dissipation within the routers and in the inter-router interconnects. We have used the GARNET on-chip network model [19] along with the Orion energy model. Inside the GARNET simulator, router informations are collected considering the number of reads, writes to router buffers, the activity at the local & global arbiters and finally the total number of crossbar traversals. The total link power is also measured according to Orion link power model. Networks total energy consumption is equal to the sum of energy consumption of all routers and links. The equation 5.2 shows the summation of energy consumption inside a router. The energy of each component depends on the dynamic and leakage energy. "The dynamic energy is defined by $E = 0.5\alpha CV^2$, here α is the switching activity, C is the capacitance and V is the supply voltage" [19]. The capacitance depends on various physical (transistor width, wire length, etc.) and architectural (number of ports, flit size, buffer size, etc.) parameters. GARNET transfers the pre-component activity to Orion. On the other hand, total leakage power consumed in the network is the sum of leakage power of router buffers, crossbar, arbiters and links.

$$E_{router} = E_{buffer\_write} + E_{buffer\_read} + E_{vc\_arb} + E_{sw\_arb} + E_{xb} \quad [19] \qquad (5.2)$$

Considering the clock frequency is 1GHz, supply voltage 1.0V, using 128bits message size and uniform traffic pattern; table 5.1 shows the simulation condition for level-1 network and table 5.2 shows the power analysis for 3D-TESH(2,1,0) level-1 network with 2mm link length and 1 virtual channel. Figure 5.1 shows the dynamic and static power dissipation for links with 1 virtual channel. Similarly, figure 5.2 shows the power dissipation for total router power for various networks; where 3D-Torus networks shows the worst.

Now, considering the off-chip network for 3D-TESH we have assumed electrical interconnect with 4Gb/s of link bandwidth. The power model for off-chip electrical interconnect [20] is based upon the figure 5.3, which shows board-level electrical interconnects and table 5.3 shows the board trace parameters and corresponding SPICE parameters. This electrical power model

Table 5.1: Simulation condition for level-1 3D-TESH network

Parameter	Value	Units
Fabrication process	65nm	-
number of nodes	64 nodes	-
Link length	2	[mm]
Operating frequency	1×10^9	Hz
Transistor type	NVT	-
Supply voltage	1	V
Traffic pattern	uniform traffic	-
Message injection rate	0.01	flits/cycle/node
Message size	128	bits
Simulation cycle	20000	-

Table 5.2: Power estimation for various networks(2mm link length, 1 virtual channel & 64 nodes)

Topology	Link Dynamic Power(W)	Link static Power(W)	Router dynamic power(W)	Router static power(W)	Clock Power (W)	Router total Power(W)
3D-TORUS	0.116735	0.675740	0.476532	1.502490	3.067085	5.046107
3D-MESH	0.134247	0.574379	0.451448	1.156960	2.492006	4.100415
3D-TESH	0.128555	0.608166	0.457665	1.268284	2.683699	4.409648

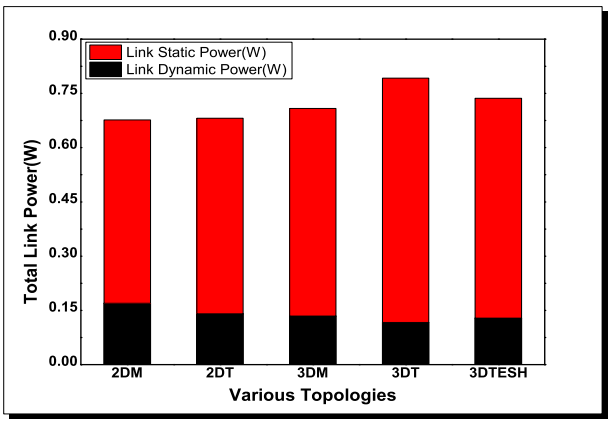


Figure 5.1: Link Power Analysis for Various Networks

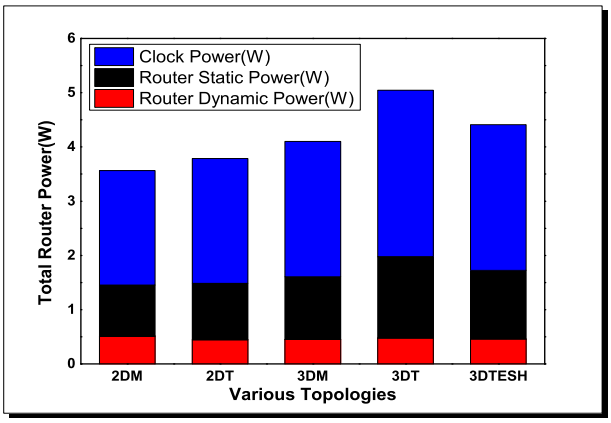


Figure 5.2: Router Power Analysis for Various Networks

considers the power dissipation at the termination resistors and depends on attenuation and noise characteristics of interconnects.

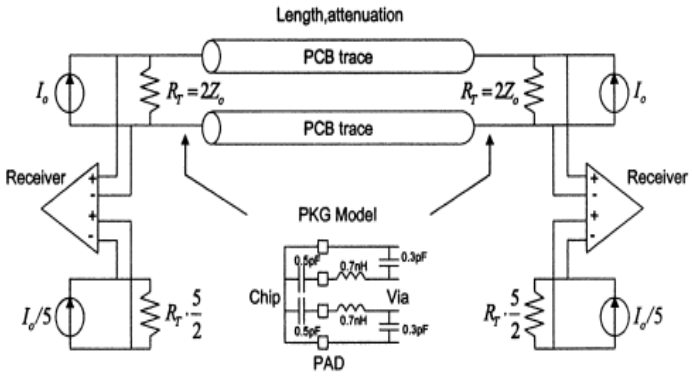


Figure 5.3: Schematic showing board-level electrical interconnects. This figure is obtained from reference [20]

The total power has been obtained from this power model [20] is the sum of the power consumed in the two termination resistances, where I_O is the required current for one way signaling and Z_0 is the impedance. Thus total power consumption has been defined as-

$$P_{term} = 1.2 I_0^2 Z_0; \quad [20] \tag{5.3}$$

Table 5.3: Board trace parameters and the corresponding SPICE parameters used for dielectric and skin effect. This table is obtained from reference [20].

Parameters	Value
L	302 [nH/m]
C	148 [pF/m]
Z_0	45 [Ω]
R_0	4.71 [Ω/m]
R_s	1.313×10^3 [$\Omega/m\sqrt{Hz}$]
G_0	0
G_d	9.929×10^{-12} [Ω/mHz]

Figure 5.4 shows the power comparison for off-chip interconnect of electrical and optical interconnects with various wire length. This figure explains the power requirement for off-chip optical interconnect is higher than the electrical interconnect up to a curtain level. But with the increase of the interconnection link length, the power consumption for the electrical interconnect has been increased much more than the optical interconnect. This figure is also been highlighted for electrical interconnect at length 100mm and 1m, which has been used for this research paper.

Now, using the above off-chip interconnect model we can also simulate the total required power for various levels of 3D-TESH network. To find the total required power at the level-2 3D-TESH network, we have assumed the off-chip wire length is 100mm, $V_{NF} = 8.8mV$, 1 virtual channel and similarly, to find the total required power at the level-3 3D-TESH network, we have assumed the off-chip wire length is 1m, $V_{NF} = 8.8mV$ & 1 virtual channel. Equation 5.4 shows the definition to calculate the required power at various level of 3D-TESH network, table 5.4 shows the parameter that have been used for calculating the power consumption for higher-level networks and figure 5.5 shows the total power comparison of various levels of network, which explains that 3D-torus network will require much more power than 3D-TESH network with the increase of the network size due to the increased number of outgoing links (section 3.4) at various levels.

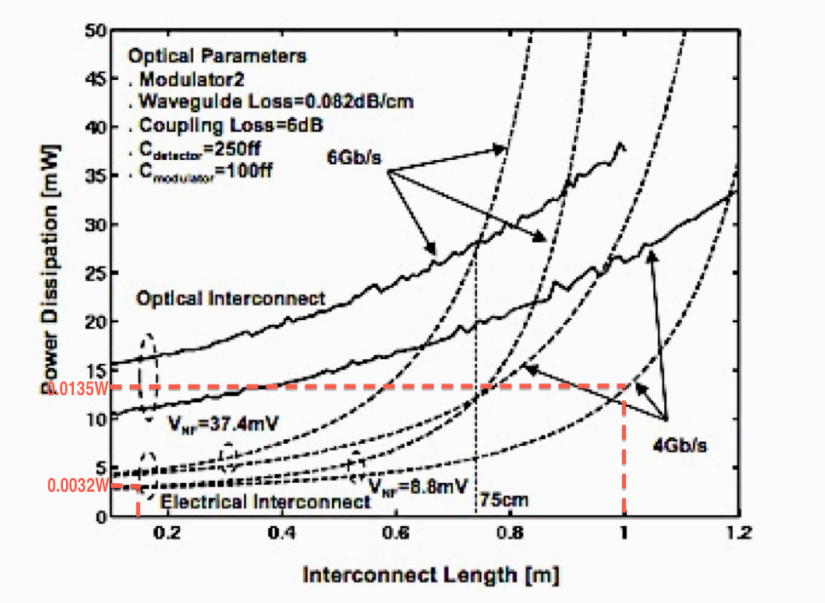


Figure 5.4: Power comparison for off-chip interconnect. This graph is obtained from reference [20]

Total power required for L_N network = (Number of BM in current level, L_N) $\times$ (power required for L_1 network) + $\sum_{i=2}^{N}$ {(Total number of outgoing links from all BM at higher-level, L_i) $\times$ (power required for each L_i link)} [where N $\geq$ 2] (5.4)

5.3 Definition of Various Traffic Patterns

Network load has a very effective influence on performance. In general, for a given distribution of destinations, the average message latency of a VCT switched network is more heavily affected by the network load than by any design parameter, provided that a reasonable choice is made for those parameters. Even throughput is also heavily affected by the traffic patterns,

Table 5.4: Parameter used for power analysis of various levels of networks

Network Size	Links length	Link bandwidth	$V_{gaussian}$	V_{NF}	Power consumption (per link)
Level-2 Network (1024 Nodes)	100mm	4 Gb/s	5mV	8.8mV	0.0032w
Level-3 Network (16384 Nodes)	1m	4 Gb/s	5mV	8.8mV	0.0135w

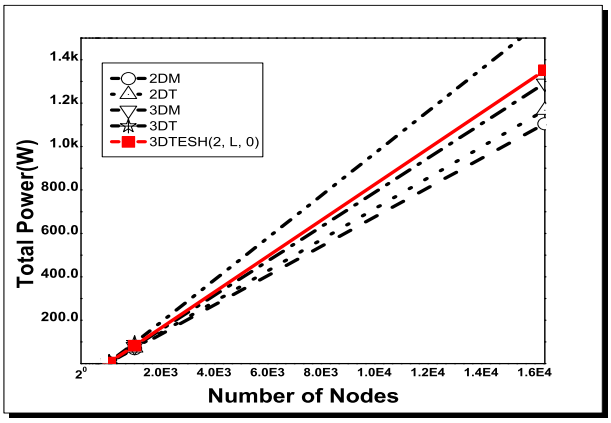


Figure 5.5: Power comparison for various networks(64 on-chip nodes)

i.e., modeling the network workload is very important. Hence we have used the following non-uniform traffic patterns along with the uniform traffic patterns.

Uniform- In the uniform traffic pattern, every node sends message to every other node with equal probability, i.e., source and destination are randomly selected for each generated message.

Perfect Shuffle- This pattern is defined as the fixed source- destination pair for every message. The node with binary value $a_{n-1}, a_{n-2}, ..., a_1, a_0$ communicate with $a_{n-2}, a_{n-3}, ..., a_0, a_{n-1}$ (rotate left 1 bit).

Matrix transpose- Fixed source-destination pair for every message. The node with binary value $a_{n-1}, a_{n-2}, ..., a_1, a_0$ communicates with the node $a_{(n-1)/2}, ..., a_0, a_{n-1}, ..., a_{n/2}$.

Tornado- Fixed source-destination pair for every message. The node with decimal coordinates [x,y] (bi-dimensional), communicates with the node [(x + (k/2-1)) mod k, (y+(k/2-1)) mod k], where k represents the size of the network in both x and y dimensions.

Bit Reversal- Fixed source-destination pair for every message. The node with binary value $a_{n-1}, a_{n-2}, a_{n-3}, a_{n-4}, ..., a_1, a_0$ communicates with the addressing node $a_0, a_1, ..., a_{n-4}, a_{n-3}, a_{n-2}, a_{n-1}$.

To stress a topology or routing algorithm, commonly used traffic pattern is matrix transpose, in which each source sends all of its traffic to a single destination. Perfect shuffle and bit-reversal traffic pattern selects its destination by selectively complementing the bits of the source address. The tornado pattern is designed as an adversary for torus topologies, whereas neighbor traffic measures a topology 's ability to exploit locality.

5.4 Simulation Environment

To evaluate dynamic communication performance, we use the TOPAZ interconnection network simulator [14]. Here, we have showed the dynamic communication performance of 3D-TESH network with regards to the 64 nodes

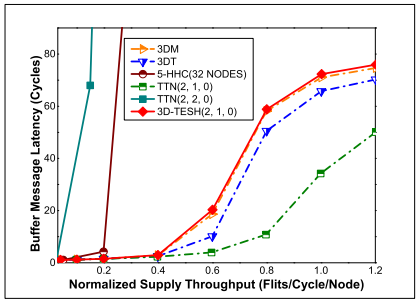
and compare the result with the other networks having the same common parameters. For all of the simulations, we have considered the 64 number of nodes for 3D-mesh and 3D-torus network, same as the 3D-TESH network. In all the simulations, we use 6 virtual channels (VCs) for per physical link and 5 flits of packet size, having 16 bytes for each flit. Hence for each message we have considered 640 bits packet length. We have used the variable load and flits are transmitted at 20,000 simulation cycles using the Virtual Cut-Through (VCT) [21] flow control and 2 cycle wire delay for each links. To evaluate the superiority of the non-uniform traffic than that of uniform counterpart, we have considered uniform traffic with the same parameters.

5.5 Comparison of Dynamic Performance of Various Networks

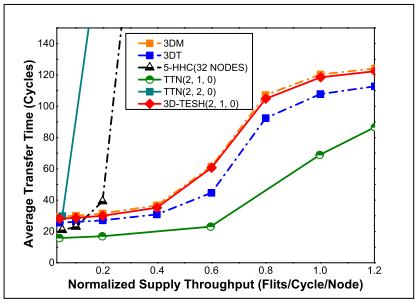
In this section, we have analyzed the performance for 3D-TESH(2,1,0) network, which is close to 3D-mesh network. Hence we have found very similar but still shows better performance than the 3D-mesh network. Here, we have compared the the dynamic performance of 3D-TESH(2,1,0) with the other networks against the Uniform, Matrix Transpose, Tornado, Perfect Shuffle and Bit-reversal traffic patterns.

Uniform Traffic: The dynamic performance of 3D-TESH under uniform traffic pattern with the variable load is shown in figure 5.6. The normalized supply throughput of the 3D-TESH network is higher than the Hierarchical Hypercube network(5-HHC) [22] having 32 nodes, takes less buffer message latency and average transfer time than the 3D-mesh. But requires more buffer message latency and average transfer time than the 3D-torus network. Hence, 3D-TESH network achieves better dynamic communication performance than the Hierarchical Hypercube alike networks; and worse performance than that of 3D-torus network. Figure has also compared the dynamic performance for TTN(2,1,0) network with 16 nodes.

Matrix Transpose: Here, 3D-TESH shows slightly better performance than the 3D-mesh network but worse than the 3D-torus and 5-HHC (32 nodes). As the comparing nodes for 5-HHC is only 32 whereas 3D-TESH uses 64 nodes so it could be possible that HHC network shows worse perfor-

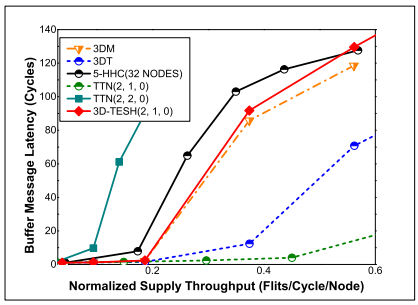


(a) Buffer Message Latency

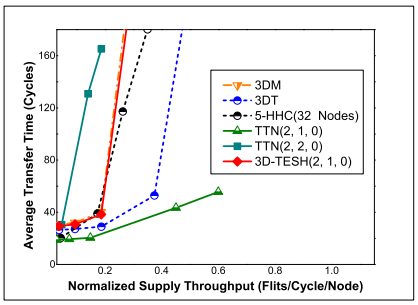


(b) Total Message Latency

Figure 5.6: Uniform traffic with 2 cycle wire delay



(a) Buffer Message Latency

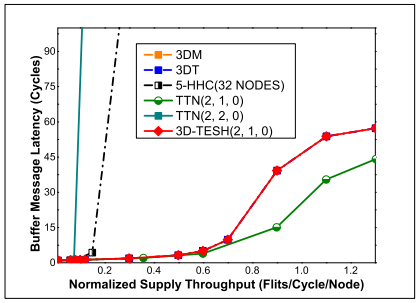


(b) Total Message Latency

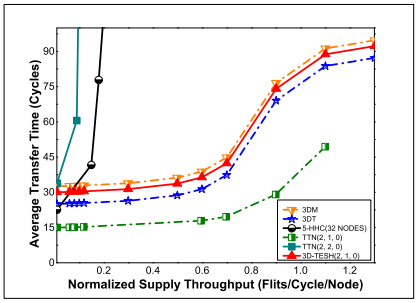
Figure 5.7: Matrix Transpose traffic with 2 cycle wire delay

mance when the HHC uses 64 nodes with the same simulation parameters. The simulation of matrix transpose traffic pattern has been shown in figure 5.7.

Tornado: Here, 3D-TESH shows better performance than the 3D-mesh and 5-HHC network but worse than the 3D-torus network in terms of both total message latency and buffer message latency. As the comparing nodes for TTN(2,1,0) is only 16, TTN should be outperform other networks. But if we increase the size of TTN network to 256 nodes with TTN(2,2,0), it shows worst performance than the others. The simulation of tornado traffic pattern has been shown in figure 5.8.

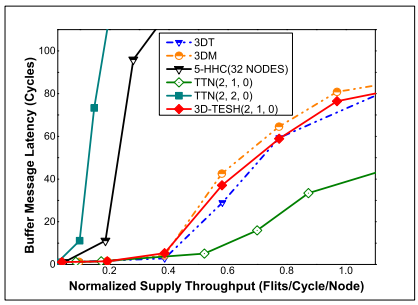


(a) Buffer Message Latency

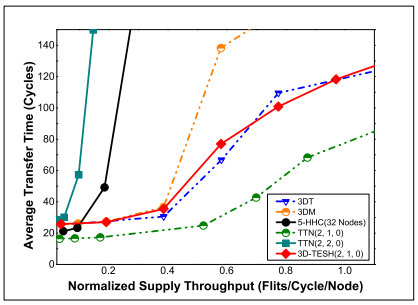


(b) Total Message Latency

Figure 5.8: Tornado traffic with 2 cycle wire delay



(a) Buffer Message Latency



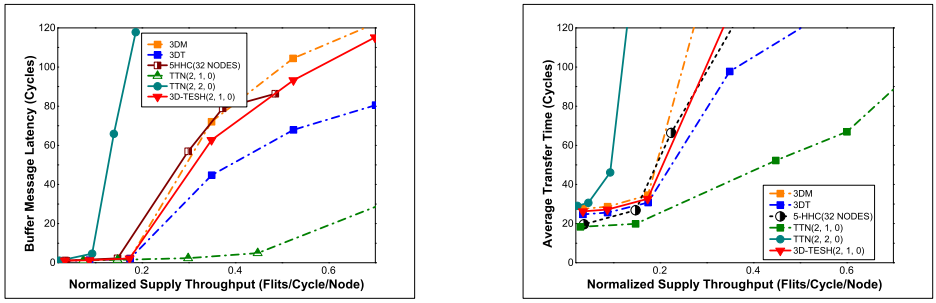
(b) Total Message Latency

Figure 5.9: Perfect Shuffle traffic with 2 cycle wire delay

Perfect Shuffle: Here, 3D-TESH shows better performance than the 3D-mesh and 5-HHC network and even almost equal to the 3D-torus network in terms of both total message latency and buffer message latency. As comparing nodes for TTN(2,1,0) is only 16, TTN should be outperform other networks. On the other hand, level-2 TTN(2,2,0) shows worst performance than the others. The simulation of perfect shuffle traffic pattern has been shown in figure 5.9.

Bit-reversal: Here, 3D-TESH shows better performance than the 3D-mesh and even almost equal to the 5-HHC network in terms of both total message latency and buffer message latency. TTN(2,1,0) also outperforms other networks due to the reduced number of processing nodes. The simulation of

bit-reversal traffic pattern has been shown in figure 5.10.



(a) Buffer Message Latency (b) Total Message Latency

Figure 5.10: Bit-reversal traffic with 2 cycle wire delay

5.6 Summary

In this chapter we have introduced power estimation and the dynamic communication performance of 3D-TESH network and also compare the communication performance against 3D-MESH and 3D-TORUS networks having the same number of nodes at the level-1 network with 64 nodes. As the hierarchy for TTN(2,1,0) is based upon 2D-mesh network, the number of nodes for TTN(2,1,0) network is lower than the three dimensional networks. Hierarchical Hypercube network shows worst performance than the all other topologies even in level-1 network with 32 nodes and 2 cycle wire delay.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

In this research plan, our main objective was to find a new interconnection network, which achieves high performance for many-core processors. And also we like to introduce a new interconnection network that could reduce the existing problems of interconnection networks such as- high power consumption, longer wire length, high cost-performance ratio, high throughput and high latency. As through our research findings, we have come to know that the performance of hierarchical networks is better than the conventional networks; hence we have introduced a new multi-dimensional hierarchical topology named as 3D-TESH network.

From our analysis we have found that the static network performance of 3D-TESH network is better than the conventional topologies of 2D-mesh, 2D-torus, 3D-mesh, 3D-torus and even better or equal than the 2D-TESH network in terms of diameter, average distance, node degree and bisection bandwidth. On the other hand, due to the torus connectivity at the basic module of the two dimensional TTN and STTN networks, shows slightly better static performance than the 3D-TESH when m is equal to 2. It is also the similar scenario for 5D-torus network, which shows high diameter and average distance performance than 3D-TESH network until 4 millions of nodes and nearly equal after that. Though the cost-performance and bisection bandwidth is worst for 5D-torus network over the 3D-TESH network. In summary, we have found that 3D-TESH network achieved about 52.08%

better diameter performance and near about 45.71% better average distance performance than the 3D-torus network with 262,144 nodes.

In case of dynamic network performance, we have able to find the buffer message latency and the total message latency along with the power consumption estimation for 3D-TESH network against the other networks. In case of power estimation, we could able to find the required power consumption for 3D-TESH network at various levels of networks, which shows that 3D-TESH network requires near about 14.81% less power than the 3D-torus networks with 16384 nodes. Both 3D-TESH and 3D-torus network requires six outgoing links for each node whereas 3D-mesh network also requires six outgoing links without any torus connectivity. Hence 3D-mesh requires less power consumption than the 3D-torus and 3D-TESH networks. Now for dynamic communication performance, 3D-TESH(2, 1, 0) network is able to show better dynamic performance than 5-HHC (about 30% better performance for uniform traffic pattern when load = 0.2 flits/cycle/node), 3D-mesh (about 2% better performance for uniform traffic pattern when load = 1.2 flits/cycle/node) in terms of uniform, tornado, perfect shuffle and bit-reversal traffic patterns even at the level-1 network with 64 nodes. As we have compared the 3D-TESH(2, 1, 0) topology at the level-1 network, it is obvious that we could get more better performance at higher levels of hierarchy for 3D-TESH network than the other networks due to hierarchical structure of 3D-TESH network.

Hence, in summary we can prefer that 3D-TESH would be a good choice for next generation supercomputers.

6.2 Future Work

There are many ways in which this thesis can be extended. Some of the possible areas are recommended below:

- Various levels of inter-connectivity has not been studied in this research, which can be a special factor for achieving the high performance of 3D-TESH network.
- The dynamic performance comparison against the 5D-torus and Tofu networks has not been studied here, which can also be an important factor in implementing 3D-TESH network in real systems.

References

1. W.J. Dally, " Performance Analysis of k-ary cube Interconnection Networks", *IEEE Trans. Comput.*, vol. 39, pp. 775-785, June 1990.

2. Daniel Sanchez, George Michelogiannakis, Christos Kozyrakis, " An analysis of on-chip interconnection networks for large-scale chip multiprocessors ", *ACM Transactions on Architecture and Code Optimization*, Vol. 7, Issue 1, Article 4, April 2010.

3. M.M. Hafizur Rahman, Susumu Horiguchi, " HTN: a new hierarchical interconnection network for massively parallel computers", *IEICE Trans. on Information and Systems*, Vol. E86-D, pp. 1479-1486, 2003.

4. Ajima, Y.; Inoue, T.; Hiramoto, S.; Takagi, Y. ; Shimizu, T. " The Tofu Interconnect ", *Micro, IEEE*, DOI. 10.1109/HOTI.2011.21, Vol. 32, pp. 21-31, JAN-FEB. 2012.

5. V.K. Jain, T.Ghirmai, and S.Horiguchi," TESH:A new hierarichical interconnection network for massively parallel computing", *IEICE Trans. on Inf. & Syst.*, vol. E80-D, pp. 837-846, 1997.

6. M.M. Hafizur Rahman, Yukinori Sato and Yasushi Inoguchi, " High and stable performance under adverse traffic patterns of tori-connected torus network", *Computers & Electrical Engineering*, vol. 39, pp. 973-983, 2013.

7. M.M. Hafizur Rahman, Yasushi Inoguchi, Faiz Al Faisal and Monz Kumar Kundu, " Symmetric and Folded Tori Connected Torus Network", *Journal of Networks, Academy Publisher*, DOI. 10.1109/ICCIT.2009.5407144, Vol. 6, No. 1, pp. 26-35, Jan., 2011.

8. M.M. Hafizur Rahman, Yasushi Inoguchi, Yukinori Sato, Yasuyuki Miura, Susumu Horiguchi, " Dynamic Communication Performance of the TESH Network under Non-uniform Traffic Patterns", *Proc. of the ICCIT*, pp. 365-370, 2008.

9. Interconnection Networks, CMU 15-418, Spring 2012, http://www.cs.cmu.edu/

10. K computer, Tofu interconnect, http://en.wikipedia.org/wiki/K_computer

11. The 6D Mesh/Torus Interconnect of K computer, http://www.fujitsu.com/downloads/TC/sc10/interconnect-of-k-computer.pdf

12. Turing, IBM Blue Gene/Q Hardware configuration, Jan 7, 2015. (http://www.idris.fr/eng/turing/hw-turing-eng.html)

13. Topics in parallel computing, Virtual channels and deadlocks, (http://pages.cs.wisc.edu/ tvrdik/8/html/Section8.html)

14. Pablo Abad, Pablo Prieto, Lucia G. Menezo, Adrian Colaso, Valentin Puente, Jose-Angel Gregorio, " TOPAZ: An Open- Source Interconnection Network Simulator for Chip Multiprocessors and Supercomputers", *Sixth IEEE/ACM International Symposium on Networks-on-Chip (NoCs)*, DOI. 10.1109/NOCS.2012.19, pp. 99-106, 2012.

15. M.M. Hafizur Rahman, Yukinori Sato, and Yasushi Inoguchi, " Dynamic Communication Performance Enhancement in Hierarchical Torus Network by Selection Algorithm ", *Journal of Networks*, DOI. 10.1109/ICCITECHN.2010.5723855, Vol. 7, No. 3, March 2012.

16. Ikki Fujiwara, Michihiro Koiuchi, Henri Casanova, " Cabinet Layout Optimization of Supercomputer Topologies for Shorter Cable Length", *13th International Conference on Parallel and Distributed Computing*, DOI. 10.1109/PDCAT.2012.86, pp. 227 − 232, Dec. 2012.

17. M. Yokokawa, F. Shoji, A. Uno, Motoyoshi Kurokawa, T. Watanabe, " The K computer: Japanese next-generation supercomputer development project", *Low Power Electronics & Design, IEEE*, DOI. 10.1109/ISLPED.2011.5993668, pp. 371-372, 2011.

18. Andrew B. Kahng, Bin Li, Li-Shiuan Peh, Kambiz Samadi, "ORION 2.0: A Power-Area Simulator for Interconnection Networks", *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, DOI. 10.1109/TVLSI.2010.2091686, vol. 20, Issue. 1, March 2011.

19. Agarwal, N., Krishna, T., Li-Shiuan Peh, Jha, N.K.," GARNET: A detailed on-chip network model inside a full-system simulator", *IEEE International Symposium on Performance Analysis of Systems and Software*, DOI. 10.1109/ISPASS.2009.4919636, pp. 33-42, 26-28 April, 2009.

20. Hoyeol Cho, Pawan Kapur, Krishna C. Saraswat, "Power Comparison Between High-Speed Electrical and Optical Interconnects for Interchip Communication", *Journal of Lightwave Technology*, DOI. 10.1109/I-ITC.2004.1345710, VOL. 22, NO. 9, September 2004.

21. J.A. Gregorio, R. Beivide, F. Vallejo," Modeling of interconnection subsystems for massively parallel computers", *Performance Evaluation 47*, DOI. 10.1016/S0166-5316(01)00058-X, pp. 105-129. February 2002.

22. Qutaibah M. Malluhi and Magdy A. Bayoumi, " The Hierarchical Hypercube: A New Interconnection Topology for Massivelv Parallel Svstems", *IEEE Transaction on parallel and distributed systems*, DOI. 10.1109/71.262585, vol. 5, no. 1, Janu. 1994.

Publications

- Faiz Al Faisal, ○ M.M. Hafizur Rahman, Yasushi Inoguchi, “ Dynamic Communication Performance of TTN with Uniform and Non-uniform Traffic Patterns using Virtual Cut-Through Flow Control ”, *International Conference on Advanced Computer Science Applications and Technologies (ACSAT2014)*, 2014, in press.

- ○ Faiz Al Faisal, M.M. Hafizur Rahman, Yasushi Inoguchi, "Dynamic Communication Performance of STTN under various Traffic Patterns using Virtual Cut-Through Flow Control", *International Conference on Frontier of Computer Science and Technology (FCST2014)*, 2014, in press.

- ○ Faiz Al Faisal, Yukinori Sato, Yasushi Inoguchi, "Introduction of a New Interconnection Network that achieves high performance for Many-Core Processors ", *Presented in JHES*, 2014, in press.

Appendix A

Routing program for 3D-TESH(2, L, 0)

```
#include <iostream>
#include <stdio.h>
#include <math.h>
#include <cstdlib>

#define DIMENSION 4

int outlet_x(int *s, int *d, int L, int VH, int routedir)
{
        int outlet_nodex;

         if (VH == 1 and routedir == 0){
                if (L == 2) return outlet_nodex = 0;
                else if (L == 3) return outlet_nodex = 0;
                else if ((L % 2) == 0) return outlet_nodex = 0;
                else if ((L % 2) != 0) return outlet_nodex = 3;
                else return -1; }

         if (VH == 0 and routedir == 0){
                if (L == 2)return outlet_nodex = 3;
           if (L == 3) return outlet_nodex = 3;
                else if ((L % 2) == 0) return outlet_nodex = L - 2;
                else if ((L % 2) != 0) return outlet_nodex = L - 3;
                else return -1;}

         if (VH == 1 and routedir == 1) {
                if (L == 2) return outlet_nodex = 0;
                else if (L == 3)  return outlet_nodex = 0;
                else if ((L % 2) == 0) return outlet_nodex = 0;
                else if ((L % 2) != 0) return outlet_nodex = 3;
                else return -1; }

        if (VH == 0 and routedir == 1) {
```

```
                if (L == 2)  return outlet_nodex = 3;
                else if (L == 3) return outlet_nodex = 3;
                else if ((L % 2) == 0) return outlet_nodex = L - 3;
                else if ((L % 2) != 0) return outlet_nodex = L - 4;
                else return -1; }

}

int outlet_y(int *s, int *d, int L, int VH, int routedir)
{
        int outlet_nodey;

        if (VH == 1 and routedir == 0) {
                if (L == 2)  return outlet_nodey = 0;
                else if (L == 3)  return outlet_nodey = 3;
                else if ((L % 2) == 0) return outlet_nodey = L - 2;
                else if ((L % 2) != 0) return outlet_nodey = L - 3;
                else return -1;
        }

        if (VH == 0 and routedir == 0)  {
         if (L == 2)  return outlet_nodey = 0;
         else if (L == 3)  return outlet_nodey = 3;
         else if ((L % 2) == 0) return outlet_nodey =  0;
         else if ((L % 2) != 0) return outlet_nodey = 3;
         else return -1;
        }

        if (VH == 1 and routedir == 1) {
         if (L == 2)  return outlet_nodey = 0;
         else if (L == 3)  return outlet_nodey = 3;
         else if ((L % 2) == 0) return outlet_nodey = L - 3;
         else if ((L % 2) != 0) return outlet_nodey = L - 4;
         else return -1;
        }

        if (VH == 0 and routedir == 1)  {
         if (L == 2)  return outlet_nodey = 0;
         else if (L == 3)  return outlet_nodey = 3;
         else if ((L % 2) == 0) return outlet_nodey = 0;
         else if ((L % 2) != 0) return outlet_nodey = 3;
         else return -1;
        }
}
```

```
int receiving_nodex(int *s, int *d, int L, int VH, int routedir)
{
        int receiving_nodex;

         if (VH == 1 and routedir == 0)  {
          if (L == 2)  return receiving_nodex = 0;
          else if (L == 3)  return receiving_nodex = 0;
          else if ((L % 2) == 0) return receiving_nodex = 0;
          else if ((L % 2) != 0) return receiving_nodex = 3;
          else return -1;
         }

         if (VH == 0 and routedir == 0)  {
          if (L == 2)  return receiving_nodex = 3;
          else if (L == 3)  return receiving_nodex = 3;
          else if ((L % 2) == 0) return receiving_nodex = L - 3;
          else if ((L % 2) != 0) return receiving_nodex = L - 4;
          else return -1;
         }

         if (VH == 1 and routedir == 1) {
          if (L == 2)  return receiving_nodex = 0;
          else if (L == 3)  return receiving_nodex = 0;
          else if ((L % 2) == 0) return receiving_nodex = 0;
          else if ((L % 2) != 0) return receiving_nodex = 3;
          else return -1;
         }

         if (VH == 0 and routedir == 1)  {
          if (L == 2)  return receiving_nodex = 3;
          else if (L == 3)  return receiving_nodex = 3;
          else if ((L % 2) == 0) return receiving_nodex = L - 2;
          else if ((L % 2) != 0) return receiving_nodex = L - 3;
          else return -1;
         }
}

int receiving_nodey(int *s, int *d, int L, int VH, int routedir)
{
        int receiving_nodey;

         if (VH == 1 and routedir == 0) {
          if (L == 2)  return receiving_nodey = 0;
          else if (L == 3)  return receiving_nodey = 3;
          else if ((L % 2) == 0) return receiving_nodey = L - 3;
```

```
          else if ((L % 2) != 0) return receiving_nodey = L - 4;
          else return -1;
         }

         if (VH == 0 and routedir == 0) {
          if (L == 2)  return receiving_nodey = 0;
      else if (L == 3)  return receiving_nodey = 3;
          else if ((L % 2) == 0) return receiving_nodey = 0;
          else if ((L % 2) != 0) return receiving_nodey = 3;
          else return -1;
         }

         if (VH == 1 and routedir == 1) {
          if (L == 2)  return receiving_nodey = 0;
          else if (L == 3)  return receiving_nodey = 3;
          else if ((L % 2) == 0) return receiving_nodey = L - 2;
          else if ((L % 2) != 0) return receiving_nodey = L - 3;
          else return -1;
         }

        if (VH == 0 and routedir == 1)  {
          if (L == 2)  return receiving_nodey = 0;
          else if (L == 3)  return receiving_nodey = 3;
          else if ((L % 2) == 0) return receiving_nodey = 0;
          else if ((L % 2) != 0) return receiving_nodey = 3;
          else return -1;
        }

}

//Defines the route direction for message routing...
int SP_Routing(int *s,int *d,int L,int i)
{
        int routedir = 0;

        if (((d[i] - s[i] + 4) % 4) > 4/2)
        {
                return routedir = 1;
        }
        else if ((((d[i] - s[i] + 4) % 4) == 4/2))
        {
                if ((L % 2) == 0)
                {
                        if ((i % 2) == 0) {return routedir = 0;}
                        else {return routedir = 1;}
```

```
                }
                else
                {
                        if ((i % 2) == 0) {return routedir = 1;}
                        else {return routedir = 0;}
                }
        }else{
                return routedir = 0;
        }
}


long BM_routing(int sy, int sx,int sZ, int dy, int dx, int dZ)
{
long diameter = 0;
int movedir = 0; //0 is for positive move and 1 is for negative move

int delz = dZ - sZ;
int delx = dx - sx;
int dely = dy - sy;

if ((delz > 0 and delz <= 2) or (delz < 0 and delz == -3)) {movedir = 0;}
if ((delz > 0 and delz == 3) or (delz < 0 and delz >= -2)) {movedir = 1;}
if (movedir == 0 and delz > 0) {delz = delz;}
if (movedir == 0 and delz < 0) {delz = delz + 2;}
if (movedir == 1 and delz < 0) {delz = delz;}
if (movedir == 1 and delz > 0) {delz = delz - 2;}

while(delz != 0)
{
 if (delz > 0) {delz = delz - 1; if (sZ + 1 >= 4) {sZ = -4 + sZ + 1;}
 else {sZ = sZ + 1;}} //move the packet to +z direction
 if (delz < 0) {delz = delz + 1; if (sZ - 1 < 0) {sZ = 4 + sZ - 1;}
 else {sZ = sZ - 1;}} //move the packet to -z direction
}
while(delx != 0)
{
 if (delx > 0) {delx = delx - 1; if (sx + 1 >= 4) {sx = -4 + sx + 1;}
 else {sx = sx + 1;}} //move the packet to +x direction
 if (delx < 0) {delx = delx + 1; if (sx - 1 < 0) {sx = 4 + sx - 1;}
 else {sx = sx - 1;}} //move the packet to -x direction
}
while(dely != 0)
{
 if (dely > 0) {dely = dely - 1; if (sy + 1 >= 4) {sy = -4 + sy + 1;}
```

```
 else {sy = sy + 1;}} //move the packet to +y direction
 if (dely < 0) {dely = dely + 1; if (sy - 1 < 0) {sy = 4 + sy - 1;}
 else {sy = sy - 1;}} //move the packet to -y direction
}

}

long Routing(int s10, int s9,int s8, int s7, int s6, int s5,
int s4,int s3,int s2, int s1,int s0,int d10, int d9,int d8,
int d7, int d6, int d5, int d4,int d3,int d2, int d1,int d0)
{
int d[11] = { d0, d1, d2, d3, d4, d5, d6, d7, d8, d9, d10};
int s[11] = { s0, s1, s2, s3, s4, s5, s6, s7, s8, s9, s10};
int t[11] = {0};
int outlet_nodex= 0;
int outlet_nodey= 0;
int diameter = 0;
int routedir = 0; //0 is for positive and 1 is for negative move

 for (int i = 10; i>=3; i--)
 {
        routedir = SP_Routing(s,d,floor((i-1)/2 + 1 ),i);

            if (routedir == 0) { t[i] = (d[i] - s[i] + 4) % 4; }
                else { t[i] = 4 - ((d[i] - s[i] + 4) % 4); }

  while(t[i] != 0)
  {
  if ((i % 2) == 1 ) {
  outlet_nodex = outlet_x(s, d, floor((i-1)/2 + 1 ), 0, routedir);
  outlet_nodey = outlet_y(s, d, floor((i-1)/2 + 1 ), 0, routedir);
  }
  else {
  outlet_nodex = outlet_x(s, d, floor((i-1)/2 + 1 ), 1, routedir);
  outlet_nodey = outlet_y(s, d, floor((i-1)/2 + 1 ), 1, routedir);
  }
        BM_routing(s[2],s[1],0, outlet_nodey,outlet_nodex,0);

  if (routedir == 0) {if (s[i] + 1 >= 4) s[i]= -4 + s[i] + 1;
  else {s[i] = s[i] + 1;}} // move the packet to the next BM
  else {if (s[i] - 1 < 0) s[i]= 4 + s[i] - 1; else {s[i] = s[i] - 1;}}
  // move the packet to the previous BM

        if (t[i] > 0) t[i] = t[i] - 1;
        if (t[i] < 0) t[i] = t[i] + 1;
```

```
        if ((i % 2) == 1 ) {
        s[1] = receiving_nodex(s,d, floor((i-1)/2 + 1 ), 0, routedir);
        s[2] = receiving_nodey(s,d, floor((i-1)/2 + 1 ), 0, routedir);
        }
        else    {
        s[1] = receiving_nodex(s,d, floor((i-1)/2 + 1 ), 1, routedir);
        s[2] = receiving_nodey(s,d, floor((i-1)/2 + 1 ), 1, routedir);
        }
  }
 }
BM_routing(s[2],s[1],s[0], d[2],d[1],d[0]);
}


int main(void)
{

  /// Source Node.....................
        int s10 = 0; int s9 = 0;
                int s8 = 0; int s7 = 0;
                        int s6 = 1; int s5 = 2;
                                int l2s4 = 1; int l2s3 = 2;
                                        int s2 = 1; int s1 = 2; int s0 = 0;

        /// Destination Node.....................
        int d10 = 0; int d9 = 0;
                int d8 = 0; int d7 = 0;
                        int d6 = 2; int d5 = 1;
                                int d4 = 2; int d3 = 1;
                                        int d2 = 2; int d1 = 1; int d0 = 0;


         Routing(s10, s9, s8, s7, s6, s5, s4, s3, s2, s1, s0,
         d10, d9, d8, d7, d6, d5, d4, d3, d2, d1, d0);

        return 0;
}
```

Appendix B

Virtual channel implementation at Topaz simulator [14]

```
int getVnet() const {
                return m_vnet;
}

Boolean cleanInterfaces() const { return m_CleanInterfaces; }

void setCleanInterfaces(Boolean value)
{ m_CleanInterfaces = value; }


 //*****************************************************************
//  f: virtual void initialize ();
//  d:
//*****************************************************************

void TPZSimpleRouterFlowTorus :: initialize()
{
   Inhereited :: initialize();
   m_vnets=((TPZSimpleRouter&)getComponent()).getVnets();
   m_ports=((TPZSimpleRouter&)getComponent()).numberOfInputs();
   unsigned ports=m_vnets*m_ports+1;
   m_changeDirection=new Boolean[ports];

   for(int i=0; i<ports; i++)
   {
      m_changeDirection[i] = false;
   }

}

Boolean TPZSimpleRouterFlowTorus :: inputReading()
{
```

```
unsigned outPort;
unsigned inPort;
unsigned virtualChannel;
cleanOutputInterfaces();

//*****************************************************
// PART 4: Loop through all output ports.
// We move the data flits for already established connections
//*****************************************************
for( outPort = 1; outPort <= m_ports; outPort++)
{
   // Find the input port assigned to outport.
   // If not assigned, go to the next.
   if( ! (inPort = m_connections[outPort] ) )
   {
            continue;
   }

   virtualChannel=(inPort-1)/m_ports+1;
   if( inPort % m_ports == 0  )
   {
       virtualChannel=m_routing[inPort]->getVnet();
           if(inPort>m_ports)continue;
   }
   if(outPort == m_ports) virtualChannel=1;

   // If there is a message routing
   if( ! ( m_routingtime[inPort]-- ) )
   {
         TPZMessage* mess=m_routing[inPort];
         m_routing[inPort] = 0;
       }
      outputInterfaz(outPort)->sendData(mess,virtualChannel);
   }
}
//*****************************************************
// PART 3: Process the connection of the crossbar
// the token is to be round robin arbitration
//*****************************************************
inPort = m_ports*m_vnets;
m_token = 0;
int i;
for(i = 0; i < m_ports*m_vnets ; i++, inPort = inPort - 1)
{
   virtualChannel=(inPort-1)/m_ports+1;
```

```
    // Extract the output port
    outPort = extractOutputPortNumber(m_routing[inPort]);

    if(inPort == m_ports)
    {
       virtualChannel=m_routing[inPort]->getVnet();
           if(inPort>m_ports)continue;
    }
    if(outPort == m_ports) virtualChannel=1;

    if( outputInterfaz(outPort)->isStopActive(virtualChannel) )
    {
       if( ! m_token ) m_token = inPort;
   continue;
    }
//if we are changing direction or injecting, bubble must be verified
    if( ! m_connections[outPort] &&
    ( ! m_changeDirection[inPort] ||
    bubbleReady(outPort+m_ports*(virtualChannel-1)) ) )
    {
       // Occupy the port
       m_connections[outPort] = inPort;
    }
    else
    {
       // If this is the first port rejected in this round
       // serve it first in the following round
       if( ! m_token )
       {
          m_token = inPort;
       }
    }

  }
  // No ports were rejected
  if( ! m_token ) m_token = inPort;
  //*********************************************************
  // PART 2: Loop through all the routing
  // Filling those that are empty with data from the fifo
  //*********************************************************
  for(inPort = 1; inPort <= m_ports*m_vnets; inPort++)
  //the in ports from the perspective of the output port
  {
     // If the routing is empty
```

```
      if( ! m_routing[inPort] )
      {
        // Take the next message from the corresponding fifo
         if(m_fifos[inPort].numberOfElements()!=0)
         {
                m_fifos[inPort].dequeue(m_routing[inPort]);
          }
      }
   }
   //*************************************************************
   // PART 1: Loop through all sync.
   //*************************************************************
   for( inPort = 1; inPort <= m_ports*m_vnets; inPort++)
   {
      // If a message at syncronizer,
      if( m_sync[inPort] )
      {
            // Put it in the corresponding buffer
            m_fifos[inPort].enqueue(m_sync[inPort]);
            // and remove it syncs
            m_sync[inPort]=0;
      }
   }
   return true;
}

//*****************************************************************
//  f: virtual Boolean bubbleReady(unsigned inPort);
//*****************************************************************
Boolean TPZSimpleRouterFlowTorus :: bubbleReady(unsigned inPort) const
{
   unsigned bubble = ((TPZSimulation*)getComponent().getSimulation())
   -> getPacketLength((TPZNetwork*)getOwnerRouter().getOwner()) * 2;

   // If there is not room for two whole packets, no bubble available.
   if(m_bufferSize - m_fifos[inPort].numberOfElements() < bubble)
   {
      return false;
   }
   else
   {
      return true;
   }
}
```