International Journal of Innovative Research in Information Security (IJIRIS)
Issue 9, Volume 2 (November 2015)
ISSN: 2349-7017(O)
ISSN: 2349-7009(P)
www.ijiris.com

Literature Review: Convey the Data in Massive Parallel Computing

Mohd. Kalamuddin Ahamad
Computer Science, Mewar University,
Chhittorghara,
India

Mohd. Husain
Department of CS,
APJ Tech. University Lucknow,
(Formerly UPTU, Lucknow) India,

ABSTRACT-- In this paper we have studied several works on direct network architectures which are well-built contestant for useful in many successful cost-effective, experimental massive parallel computers and well scale up shared memory of multiprocessors. The uniqueness of direct networks, as reflected by the communication latency and routing latency metrics are significant to the performance of such systems. A multiprocessor system can be used for the wormhole routing for the most capable switching method and has been adopted in several new massive parallel computers. This technique is unique technical challenges in routing and flow control in particular system, and avoid deadlock. The highly scale up network is a combination of topology and hypercube. Due to the being of concurrent multiple mesh and hypercubes, this network provides a great architectural support for parallel processing. The growth of the network is more efficient in terms of communication, interconnection network is scaled up the network and will be more reliable and also the unreliability of the interconnection network to get minimized. This is very desirable characteristic for the interconnection network as the network remains equipped for more failure of adjoining nodes or links in parallel computer architecture. Formulations to optimize the performance of throughput of networks through queuing theory M\M\1 concept.

Keywords – Communication, Scalability, Queueing theory, Hypercube network, Parameters, Embedded network, Arrival rate, Departure rate.

I. INTRODUCTION

Parallel processing is a vital part of everyday life. The concept is so inbuilt in our existence that we benefit from it without realizing. When faced with a tough problem, we involve others to solve it more easily. This co-operation of more than one worker to facilitate the solution of a particular problem may be termed as parallel processing. The goal of parallel processing is thus to solve a given problem more rapidly, or to enable the solution of a problem that would otherwise be impracticable by a single worker. The principles of parallel processing are, however, not new, as evidence suggests that the computational devices used over thousand years ago by the Greeks recognized and exploited such concepts *[2],[3] [4]*.To the following key for scale up to the design of the multicomputer architecture

- A lot of research effort has been dedicated during the last decade to improve the performance of multicomputer.
- A key architectural issue is the PE-interconnection networks. Since the number of nodes in the multicomputer network is increasing, the time required to move data between the nodes is important in total system performance.
- Whether a direct network system is used with message-passing or a shared-memory concepts, the transmission time become critical. Also, it will affect the possible granularity level of parallelism in executing an application program.
- One of the most powerful architectural schemes used in PE-interconnection networks is wormhole routers and the related routing algorithms.

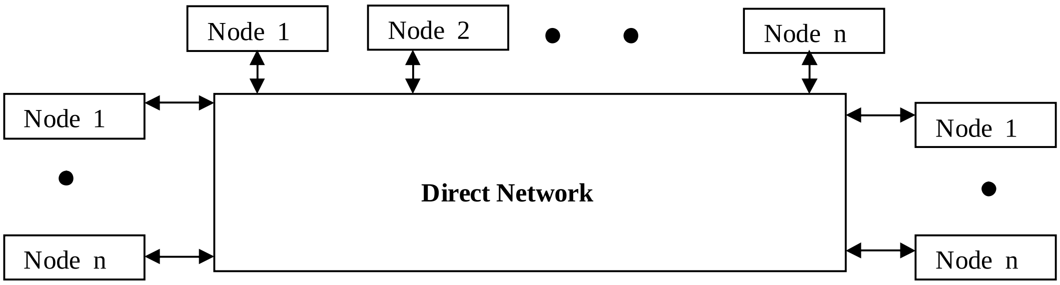


Fig.1. A general multiprocessor based on direct network

The approaches the every node are connected to one another various among machines. In direct network architecture, each node has a point to point, or direct, connection to some number of nodes, called the neighboring nodes. Direct network have become popular architecture for constructing massively parallel computers because they scale well, that is the number of nodes in the system increases, the total communication bandwidth and processing capability of the system also increase. In fig. 1, shows a generic multiprocessor with a set of nodes inters connected via direct network or PE-interconnecting network *[1], [9], and [4]*.

II. DIRECT NETWORKS

In direct network architecture, each node has a point-to-point, or direct, connection to some number of other nodes, called neighboring nodes. Neighboring nodes may send packets to one another directly, while nodes that are not directly connected must rely on other nodes in the network to transfer packets from source to destination. While a router's function could be performed by the corresponding local processor, dedicated routers are used to allow overlapped computation and communication within each node. Each router supports some number of input and output channels. Internal channels connect the local processor memory to the router. External channels are used for communication between routers, and, therefore nodes. By connecting the input channels of one node to the output channels of other nodes, the topology of the direct network will be defined *[9], [10]*. In fig.2 shows the few direct networks.

For topologies in which packets may have to traverse some intermediate nodes, the routing algorithm determines the path selected by a packet to reach its destination. At each node, the routing algorithm indicates the next channel to be used. Efficient routing is critical to the performance of interconnection network *[11], and [13]*.

When a message or packet header reaches an intermediate node, a switching mechanism determines how and when the router switch is set, i.e. the input channel is connected to the output channel selected by the routing algorithm. The switching mechanism determines how network resources are allocated for message transmission.

Popular direct networks are:

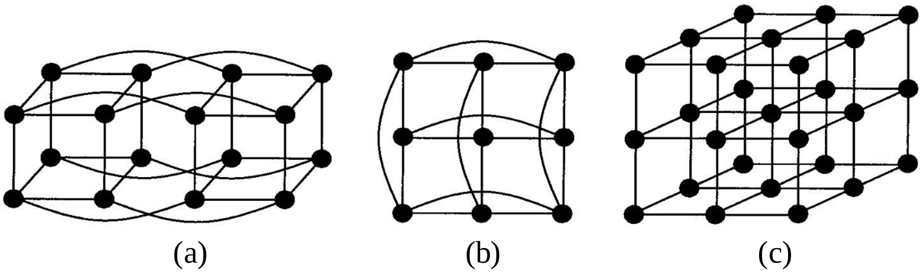
- *k*-ary *n*-cube or torus
- n-dimensional mesh
- hypercube

A. n- Dimensional mesh: Formally has $k_0 \times k_1 \times \ldots \times k_{n-2} \times k_{n-1}$ nodes, k_i nodes along each dimension i, where $k_i \geq 2$ (nodes have from n to 2n neighbors, depending on their location in the mesh).

B. k- ary n-cube: all nodes have the same number of neighbors (all k_i are equal) There are wraparound channels in *k*-ary *n*-cube, which are not present in the n-dimensional mesh. A *k*-ary *n*-cube contains k^n nodes.

- If k=2, Every node n neighbors, one in each dimension.
- If k > 2, Every node has 2n neighbors, two in each dimension.
- If n =1, k-ary n-cube collapses to a ring with k nodes.

C. Hypercube: Hypercube is a special case of both the *n*-dimensional mesh and the *k*-ary *n*-cube hypercube is an *n*-dimensional mesh in which $k_i = 2$ for all $0 \leq i \leq n-1$, i.e., a *2*-ary *n*-cube.

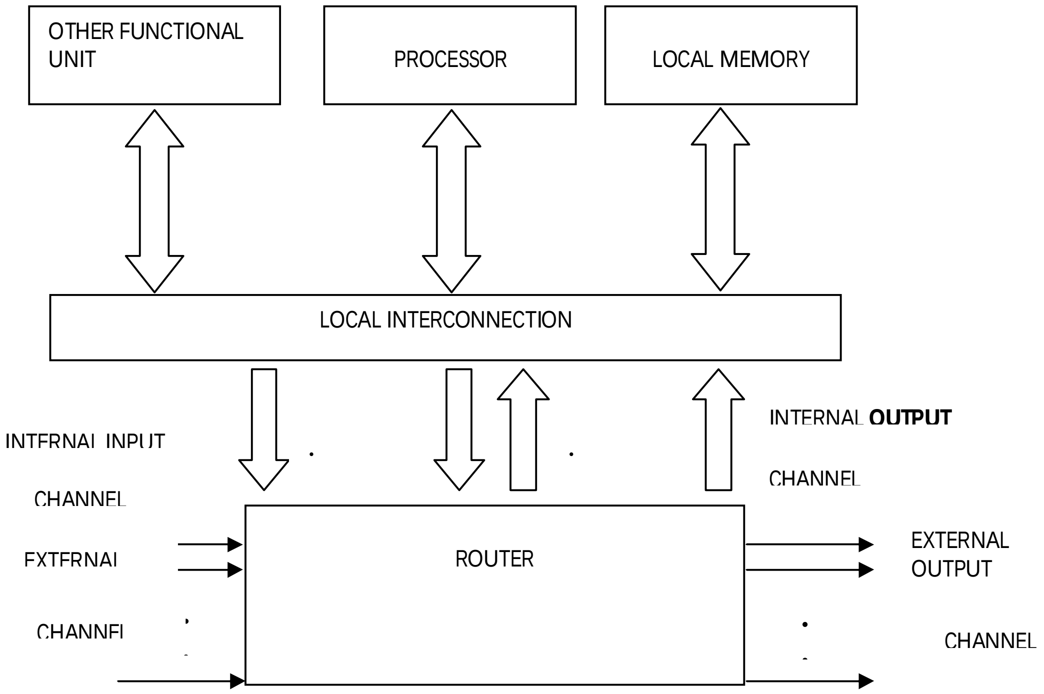


(a) 2-ary 4-cube (hypercube), (b) 3-ary 2-cube (torus), (c) 3-ary 3D-Mesh

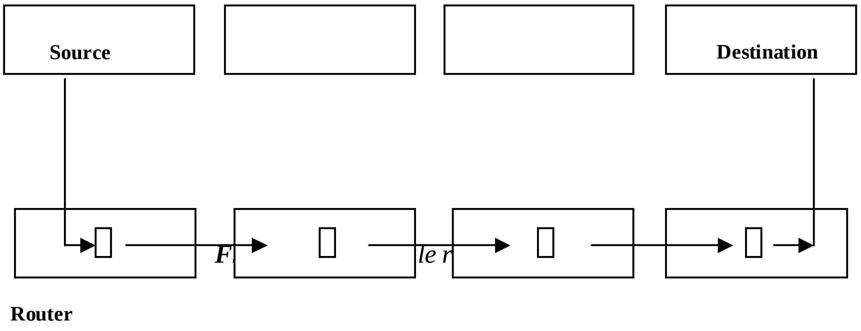
Fig. 2 *(a), (b), and **(c)** Direct network topologies*

III. WORMHOLE ROUTING

Wormhole routing is a particular case of cut-through switching. Instead of storing a packet then transmitting it to the next node, wormhole routing operates by advancing the head of a packet directly from incoming to outgoing channels of the routing chip. A packet is divided into a number of *flits* (flow control digits) for transmission. The size of a flit depends on system parameters, in particular, the channel width. The header flit (or flits) governs the route. As soon as a node examines the completely in a node and header flit(s) of a message, it selects the next channel on the route and begins forwarding flits down that channel. As the header advances along the specified route, the remaining flits follow in a pipeline fashion in fig.3.(b) and generic node architecture show in fig. 3. (a).



Processing Element



Because most flits contain no routing information, the flits in a message must remain in contiguous channels of the network and cannot be interleaved with the flits of other messages. When the header flit of a message is blocked, all of the flits of a message stop advancing and block the progress of any other message requiring the channels they occupy.

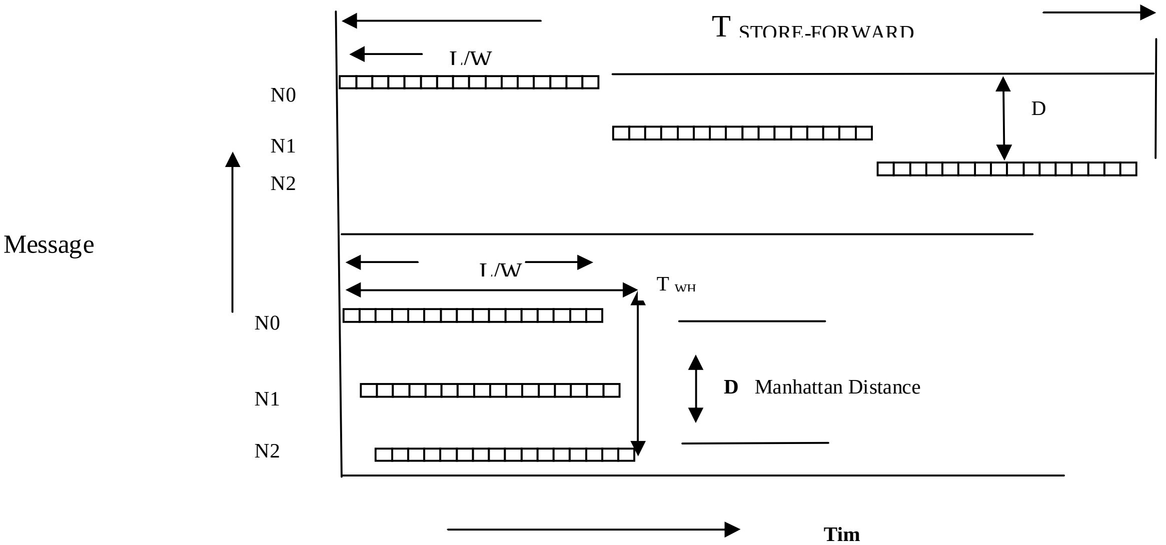


Fig. *3. (c) Wormhole routing Vs store and forwara*

Wormhole routing avoids memory bandwidth in the nodes through which messages are routed. Only a small FIFO flit buffer can be used. It also makes the network latency largely insensitive to path length. On the other hand, in order to reduce the effect of message blocking, physical channels may be split into virtual channels and these will be used to increase the total throughput of the physical channel *[2]*. Virtual channels are logical entities associated with a physical link used to distinguish multiple data streams traversing the same physical channel. They are multiplexed over a physical channel in a demand-driven manner, with bandwidth allocated to each virtual channel as needed *[11], [13].*

In fig. 3.(c) , Illustrate the advantage of wormhole routing. There are the two components of latency, distance and message aspect ratio. The distance D is the number of hops required to get from the source to the destination. The message aspect ratio is the number of channel cycles to transmit the message across one channel. The top half of the figure show the store and forward routing. The message is entirely transmitted from node N0 to N1, the from N1 to N2 and so on . With store and forward routing, latency is the product of D and L/W.

$$\mathbf{T}_{\textbf{Store-Forward}} = T_c\,(D * L/W)$$

The bottom half of figure.3 shows the wormhole routing. As soon as a flit arrives at a node, it is forwarded to the next node. With wormhole routing, latency is reduced to the sum of D and L/W.

$$\mathbf{T}_{\textbf{Wormhole-Routing}} = T_c\,(D + L/W)$$

In both of these equation, T_c is the channel cycle time, the amount of time required to perform a transaction on a channel. Some of the direct networks that use wormhole routing are Ncube-2 (hypercube), Intel Touchstone Delta (2D mesh), Paragon (2D mesh), MIT J-Machine (3Dmesh) and Cray T3D (3D torus).

A. *ROUTING ALGORITHMS*

In an intercommunication network, routing algorithms that are used for determining the path to the destination node can be classified according to their [11]:

Number of destinations- Uncast: packets may have a single destination, Multicast: packets may hay multiple destinations. Place where routing decisions are taken- Centralized: by centralized controller, Source: by the source node, Distributed: determined in a distributed manner while the packet travels.

Multiphase: Hybrid, source node computes some destinations, path established in a distributed manner

Way of implementation- Table-Lookup: looking at a routing table, Finite-State Machine: executing a routing algorithm in software or hardware according to a finite-state machine.

Adaptivity- Deterministic: always supply the same path between a source/destination pair.

Progressiveness- Progressive: move the header forward, reserving a new channel at each routing operation, Backtracking: allow header to backtrack, releasing previously reserved channels (used for fault-tolerant routing).

Minimality- Profitable supply channels that bring the packet closer to it destination, Misrouting (non-minimal): may also supply channels that send the packet away from its destination. Number of alternative paths- Fully adaptive, Partial adaptive

IV. NETWORK PPROPERTIES

In this section the properties of a k- ary n cube network *[1-3], [13]*-

IV. I *Illustrate the Notation in* k- ary n cube network

In a k ary n cube network:
n: Illustrate the number of dimension in hypercube
k: Illustrate the number of nodes per dimension in hypercube
N: Illustrate the number of nodes in hypercube
C: Illustrate the number of bidirectional channel in hypercube
w: Illustrate the width of bidirectional channel in hypercube
W: Illustrate the wiring of k-ary n-cube network
b: Illustrate the bisection width of a k-ary n-cube network
B: Illustrate the bisection density of a k-ary n-cube network
D: Illustrate the diameter of a k-ary n-cube network

Nodes: In a k-ary n-cube network, the number of nodes $N=k^n$ for torus, $N= 2^n$ for hypercube.

***Node degree*:** In a k-ary n-cube network, the node degree directly proportional to dimensionality of network. If we count the number of bidirectional linked to a node then the node degree is 2n for the torus, and n for the hypercube.

Channels: In a k-ary n-cube network, the total number of bidirectional channels is $C= n*k^n$ for the torus, $C= n*2^n$ for the hypercube.

Channel width: The channel width is defined as the number of physical wires per channel. Although a channel consisting of data and control lines. We will ignore the control lines assume that the data lines dominate the wiring. To have an equal packaging cost for nodes, the number of wires per node should be the same in the k-ary n-cube network. In k-ary n-cube network, the channel width w is the number of data lines. In a physical bidirectional channel, w' = n*w for torus, w'=n*w/2 hypercube.

Total wiring: The total wiring is defined as the number of data wires in a network, it is a measure of the tot bandwidth (or capacity) of network. It is the product of the total number of physical channels and channel width.

***Network Diameter*:** The network diameter is defined as the maximum distance between any two nodes in the network. It is calculated by counting the number of hops between the two most distant nodes in the network. In a k-arry n-cube network, Diameter D=nk/2 for torus and D=n for hypercube.

A. *Properties of Embedded Network*

The interconnection network is a vital role in a parallel processing. A good interconnection network is expected to have least number of links, topological network cost and more reliable. The interconnection network must be able to built scale up .The data routing functions in embedded hypercube network could be analyzed [6-10].

(1). *Mesh Embedded*

A Single stage recirculating represent network. In network, each PE_i is allowed to send to any one of PE_{i+1}, PE_{i-1}, PE_{i+r} and PE_{i-r}, Where r= N in one circulating steps through entire network [5] and [8], and [11]. In fig 4 Illustrate the mesh embedded hypercube.

The following routing function from (1) - (5) apply on simple mesh network [8] –

$$R_{+1}(i) = (i+1) \bmod N \qquad (1)$$

$$R_{-1}(i) = (i-1) \bmod N \qquad (2)$$

$$R_{+r}(i) = (i+r) \bmod N \qquad (3)$$

$$R_{-r}(i) = (i-r) \bmod N \qquad (4)$$

$$R_C(k_{n-1} \ldots\ldots k_{d+1}\, k_d\, k_{d-1} \ldots\ldots k_0) = (k_{n-1} \ldots\ldots k_{d+1}\ k'_d\ k_{d-1} \ldots k_0) \qquad (5)$$

Where 0≤ i ≤ N-1, Commonly N know as perfect Square.

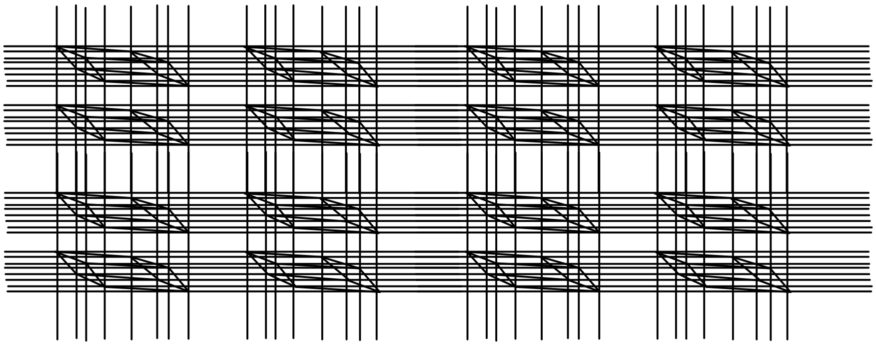


Fig. ***4.*** *A (4, 4, 8)-Mesh embedded hypercube Network*

(2). *Torus Embedded*

Suppose $l \times m$ be the size of several concurrent torus networks with l number of rows and m number of columns and N being the number of nodes connected in the hypercube, the torus embedded hypercube network can be designed with the size of (l, m, N) *[6,7,8]*. Nodes with identical positions in the torus networks will be a group of N number of nodes connected in the hypercube configuration and can be addressed with three parameters such as row number i, column number j of torus and address of node k in hypercube where the addressed node is residing. Hence, a (l, m, N)–Torus embedded hypercube network will have $l \times m \times N$ number of nodes and a node with address as (i, j, k) where $0 \leq i < l$, $0 \leq j < m$ and $0 \leq k < N$. The data routing functions of torus embedded hypercube

$$T_1(i, j, k) = (i, (j+1) \bmod m, k) \qquad (6)$$

$$T_2(i, j, k) = (i, (m+j-1) \bmod m, k) \qquad (7)$$

$$T_3(i, j, k) = ((i+1) \bmod l, j, k) \qquad (8)$$

$$T_4(i, j, k) = ((l+i-1) \bmod l, j, k) \qquad (9)$$

$$T_C(k_{n-1} \ldots\ldots k_{d+1}\, k_d\, k_{d-1} \ldots\ldots k_0)$$

$$= (k_{n-1} \ldots\ldots k_{d+1}\ k_d^{\,‘}\ {}_{d-1} \ldots k_0) \qquad (10)$$

Two nodes (i_1, j_1, k_1) and (i_2, j_2, k_2) are said to be connected if following connection rules are satisfied:

***Rule* 1:** A *hypercube link* in the network will exist if

(a) $i_1 = i_2$ and

(b) $j_1 = j_2$ and

(c) k_1 and k_2 differ by one bit position in their binary address.

***Rule* 1:** generates $l \times m$ hypercube with dimension N and these hyper cubes are separated from each other until the *rule* 2 is applied.

***Rule* 2:** A *mesh link* will exist if

(a) $k_1 = k_2$ and

(b) i and j differ by one in one component while the other component is identical.

***Rule* 3:** A *Torus link* will exist if follows the rule2 and wraparound the row and column.

3. *Double Loop Embedded Hypercube*

The total number of nodes in the DLH (m,d) is $4m \times 2^d$. When m=2, the DLH(m,d) can be constructed by combining the positive features of the hypercube topology, such as small diameter, high connectivity, symmetry and simple routing, and the scalability and constant node degree of the DL(2m) topology as follows [2],[5],[13]:

1) The 2^d nodes can be connected to be a d dimensional hypercube network according routing equation (5), where d is denoted the dimensional hypercube networks.

2) The 4*m* such kinds of *d* dimensional hypercube networks can be divided into 2*m* groups, in which any group can be coded with a group-id, which adopts Johnson code from 0 to 2*m*, and any dimensional hypercube network in a same group can also be coded with a net-id using 0 or 1.
3) The nodes with both the same node-id in different groups can be connected to a DL(2*m*) according to *d* .
4) The code of nodes in the DLH(*m*,*d*)*:* When there is just one bit different between any two nodes, there will exist a link between them, that is to say, these two nodes are neighboring to each other.

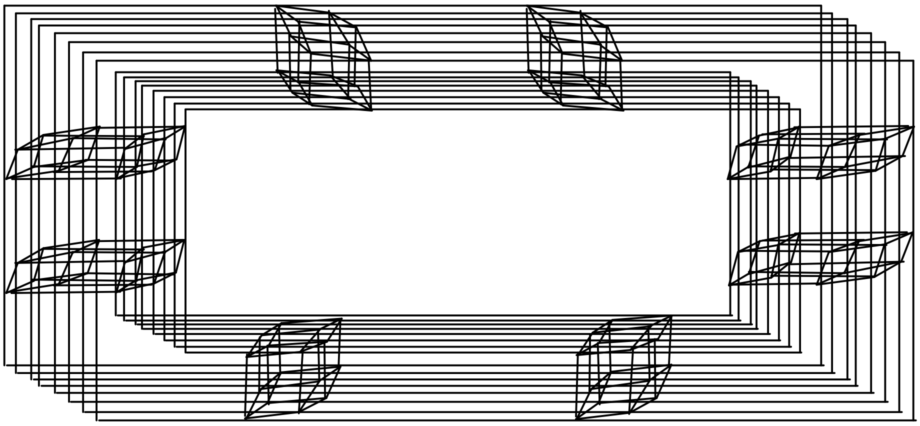


Fig.5. *Double Loop Embedded Hypercube*
Fig.5 shows a DLH (m,d) interconnection where solid lines represent hypercube links .

V. DEADLOCKS in a WORMHOLE-SWITCHEDTORUS

Deadlocks can occur in wormhole-switched tori based on the proposed routing algorithm for VCT-switched networks. Fig.6 shows a deadlock pattern. In this figure, clue is applied to a wormhole-switched 2D-torus and dimension order routing is applied to the mesh sub network. A packet Xa (from A to A1) occupies channels from AE to D$_1$B, and requests the channel BF. Its header flit is at node Band a flit is still at E. Packet X$_b$ (from B to B$_1$) occupies the channels from BF to Ai C, and requests the channel CG, whose header flit is at node C and a flit is still at F. Packet Xc (from C to C1) occupies channels from CG to B1D, and requests channel DH. Packet X$_d$ (from D to D$_1$) occupies channels from DH to C$_1$A, and requests the channel AE. Each packet occupies several R1channels. All of them can request only R$_1$ channel and none of them can advance. However, in a VCT-switche2D-torus, packets at node E in the set S2, which can escape its right node in the set S1in this pattern. VCT-switching allocates buffers to the entire packet, while wormhole may not. This difference leads to a possible deadlock in wormhole switching *[14-17]*.

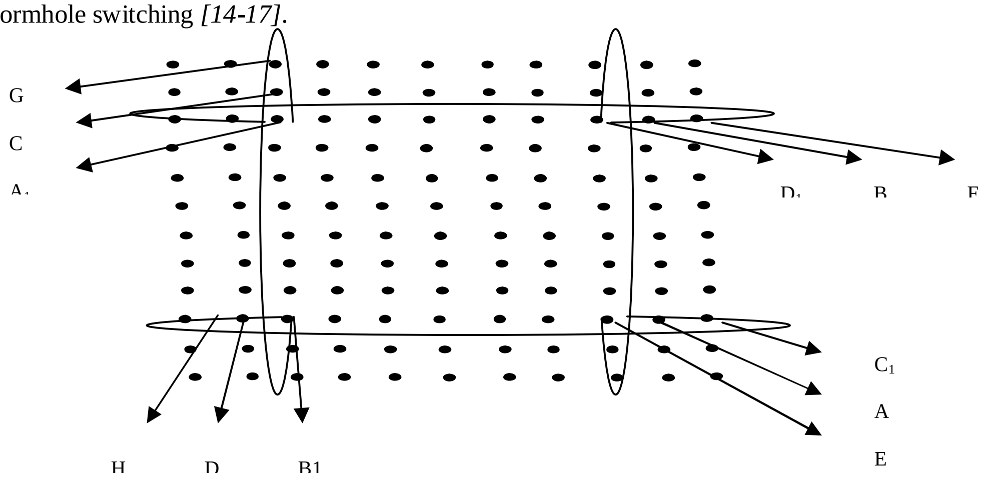


Fig. *6. Shows a deadlock pattern*

→ R_1 channel occupied by Xa
→ R_1 channel occupied by Xb
→ R_1 channel occupied by Xc
→ R_1 channel occupied by Xd

V.I *Evidence of Wormhole*

In order to avoid deadlocks in a wormhole-switched torus, a minimal deadlock-free algorithm named wormhole-clue is proposed. The difference between clue and wormhole-clue is that, a constraint is added to R channels in wormhole-clue. Let a packet need traverse one or more wraparound links from the current node to the destination, it requests R_1 channels. The packet can request only R channels from nodes of class S_i to nodes of class S_j where (j <= i) in wormhole-clue. The rules of wormhole clue can be stated as follows:

(1). A packet need traverse one or more wraparound links.
    (a).R1 channels. It can request only R channels from nodes of class S to nodes of class S_1 where (j<= i).
    (b).R channels. The next hop of a packet is to traverse a wraparound link of dimension d, and d is the lowes of the dimensions in which the packet need traverse wraparound links from the current node to the destination. The packet can request the R2channel of that wraparound link.2. A packet need not traverse wraparound links.
(2). A packet need not traverse wraparound links.
    (a).R_1 channels. It can request any of the R channels available.
    (b).R_2 channels. It can request a R channel of the mesh sub network links. However, it must follow a Deadlock-free routing algorithm for meshes, such as, negative-first or dimension order routing when being delivered via $R_{2.}$

To simplify the routing logic, we replace the restriction above as follows: when packets need cross wraparound links, they can only request R channels of the dimensions which they need cross wrap-around links.

VI. QUEUING THEORY

Queuing technique is an important factor in switching network architectures, since it strongly influences the aggregated bandwidth of the network. In the simple input queuing technique the packets queue at the switch input awaiting the availability of the desired switch output; higher performance is offered by output queuing which in turn is difficult to implement. A solution is to form multiple queues at each switch input or to create a central buffer shared among all switch inputs and output.

Queuing theory is an appropriate and useful modeling tool for system analysis and performance evaluation in computer and telecommunications network [14]. Since our proposed model has been constructed on the M/M/1 priority queue [3], in this section we give a quick review on the M/M/1 queue and priority queue concepts.

VI.I M/M/1 Queue

The M/M/1 queuing model has a single service facility with one server, unlimited waiting room and the first-come first-served queue discipline. The service times are independent and identically distributed with a general distribution, the inter_arrival times of customers are also independent and identically distributed with a general distribution, and the inter_arrival times are independent of the service times. It is assumed that the general inter_arrival time and service time distributions are each partially specified by their first two moments. We should remind here that the nth moment of a random variable X is defined as the average of $X^n(\overline{X^n} = \Sigma_{i=1}^{k}(X_i)^n/k)$. All descriptions of this model thus depend only on the basic parameter 4-tuple$(\overline{a}, a^2, \overline{s}, s^2)$, where $\overline{a}$ and $\overline{a^2}$ are the first and second moments of the customers' inter_arrival time, and similarly, $\overline{s}$ and $\overline{s^2}$ are the first and second moments of the service time. Also in this work we consider the arrival rate and service rate as $\lambda = 1/\overline{a}$ and $\mu = 1/\overline{s}$, respectively. The mean delay time of a M/M/1 queuing model system can be approximate by Allen – Cunneen formula [3]

$$W_{MM/1} \approx \frac{\rho(C_A^2 + C_S^2)}{2\mu(1-\rho)}$$

Where ρ is the utilization factor of the server and equal to λ/μ, C_A and C_S are the coefficient of variation of the inter_arrival time and service time respectively [3].We remind that the relationship between Coefficient variation of random variable X and its moments is represented by

$$C_X^2 = \overline{x^2}/\overline{x}^2 - 1.$$

A. *Priority Queue*

We consider a system with one server in which the customers have preferential treatment based on priorities associated with them. We assume that the priority of a customer is an integer fixed at arrival time, and a customer with priority i (i=1,2,3………………..,p) belongs to class i. We say one customer has higher priority than another if it belongs to a priority class with lower index.

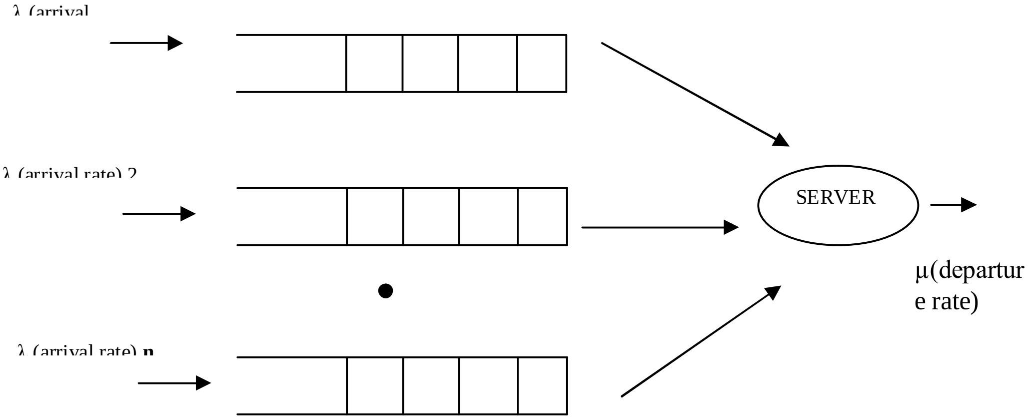


Fig. 7. Priority queuing system

In other words, the lower the index, the highest priority. The priority queuing system to be studied is depicted in Fig.7, where the different queue levels correspond to the different priority classes. For the service discipline, we assume that whenever a customer is completed, the server is next assigned to that customer at the head of the highest priority nonempty queue. Once a customer begins on the server, it is allowed to run to completion; i.e., the service discipline is non preemptive. Independent and identically distributed arrivals and service times are assumed for the ith class with the arrival and service rate denoted by and μ_i, respectively. In all the analysis we have reviewed so far, and the queue size of each class was infinite. However, in the case of wormhole switching this is not a true assumption, because in wormhole switching each buffer can hold only finite number of flits.

VII. CONCLUSION

In this paper we have discussed designing of multiprocessing computer and, how to connect each and every node to each other in the PE-Interconnecting network device. In the massive computing system the message is incoming and outgoing from one node to the next node, and we got avoidance of the deadlock by wormhole clue. Further different topologies are using the data communication with embedded hypercube to resolve the conflict by applying various switching technique and routing algorithms. We study the concepts of routing delay, properties of hypercube, queuing theory, and optimizing technique etc.for different journals. In our research work we scale up network in a combination of static topology and hypercube network. The analysis of results show that embedded hypercube interconnecting network is highly scalable and configuration of the existing node is not required. Due to the existence of concurrent embedded hypercube network, this network provides a great architectural support for parallel processing. The growth of the network is more efficient in terms of different parameter of used in data communication in future.

Further, we analysis the delay of torus embedded hypercube interconnection network has shown that as the interconnection network is scaled up the network will be more reliable than without embedded system. This is very desirable feature for the interconnection network as the network remains operational for more failure of neighboring nodes or links in parallel computer architecture.

Analysis of the dependability of embedded hypercube PE-Interconnection network has exposed that as the interconnection network is scaled up the network will be more reliable. Formulations to optimize the performance of various result like that latency metric, analysis of mesh embedded hypercube interconnecting network, reliability of torus embedded hypercube.

REFERENCES

[1]. Zhao yang Li, Yi Zhang, Yu Chen and Ruichun Tang, “Design and implementation of a high-performance interconnection network,” Proceedings of the Fourth International Conference on Parallel and Distributed Computing, Applications and Technologies, 2003, PDCAT'2003, 27-29 Aug.2003, pp.17- 20.

[2]. K. Hwang, "Advanced Computer Architecture: Parallelism, Scalability, Programmability," New York McGraw-Hill, 1993.
[3]. Hesham El-Rewini and Mostafa Abd-El-Barr, "Advanced Computer Architecture and Parallel Processing," John Wiley & Sons, Inc., Hoboken, New Jersey, 2005.
[4]. Ahmed Louri and Hongki Sung, "An Optical Multi-Mesh Hypercube: A Scalable Optical interconnection Network for Massively Parallel Computing," Journal of Light wave Technology, Vol.12, No.4, Apr.1994, pp.704-716.
[5]. Ahmed Louri and Hongki Sung, "A scalable optical hypercube-based interconnection network for massively parallel computing," Applied optics, Vol.33, No.11, Nov.1994.
[6]. N. Gopalakrishna Kini, M. Sathish Kumar and Mruthyunjaya H.S., "A Torus Embedded Hypercube Scalable Interconnection Network for Parallel Architecture," IEEE explore conference publications, Mar.2009, pp.858-861.
[7]. N. Gopalakrishna Kini, M. Sathish Kumar and Mruthyunjaya H. S., "Analysis and Comparison of Torus Embedded Hypercube Scalable Interconnection Network for Parallel Architecture," International journal of Computer Science and Network Security, Vol.9, No.1, Jan.2009, pp.242- 247.
[8]. N. Gopalakrishna Kini, M. Sathish Kumar and Mruthyunjaya H.S., "Design and Comparison of Torus Embedded Hypercube with Mesh Embedded Hypercube Interconnection Network," International journal of Information Technology and Knowledge Management, Vol.2, No.1, Jun.2009, pp.87-90.
[9]. J.F.Fang, J.Y.Hsiao and C.Y.Tang, "Embedding Meshes and TORUS Networks onto degree-four chordal rings," IEE Proc.-Comput. Digit. Tech., Vol. 45, No.2, Mar.1998.
[10]. Liu Youyao, Han Jungang and Du Huimin, "A Hypercube-based Scalable Interconnection Network for Massively Parallel Computing," Journal of Computers, Vol.3, No.10, Oct.2008, and pp.58-65, ISSN: 0975-3397.
[11]. Ayse Yasemin Seydim "Wormhole Routing in Parallel Computers", School of Engineering and Applied Sciences Southern Methodist University e- mail:yasemin@seas.smu.edu May, 1998.
[12]. Axel Jantesh, Zhonghai Lu ,"An Analytical Latency Model for Networks-on-Chips", IEEE Transaction.
[13]. WilliamJ. Dally,"Performance analysis of k-arry n-cube interconnection networks", IEEE Transaction on computers, vol 39, No. 6 June 1990.
[14]. L.M. Ni, and P.K. McKinley, "A Survey of Wormhole Routing Techniques in Direct Networks", IEEE Computer, vol.26, no.2, pp. 62-76, Feb. 1993.
[15]. C. Glass and L. Ni, "Fault-Tolerant Wormhole Routing in Meshes", Proc. 23rd Int'l Symp. Fault-Tolerant Computing, pp. 240-249, June1993.
[16]. L. Ni and P. McKinley, "A Survey of Wormhole Routing Techniques in Direct Networks", Computer, vol. 26, no. 2, pp. 62-76, Feb.1993.
[17]. L. Schwiebert and D.N. Jayasimha, "A Necessary and Sufficient Condition for Deadlock-Free Wormhole Routing", J. Parallel and Distributed Computing, vol. 32, no. 1, pp. 103-117, Jan. 1996.