International Journal of Innovative Research in Information Security (IJIRIS) ISSN: 2349-7017(O)
Issue 2, Volume 5 (May 2015) ISSN: 2349-7009(P)
www.ijiris.com

Data Transmit in Communication with Multiprocessor Interconnection: Literature Review

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

Mohd. Husain
Department of CSE
UttarPradeshTech.University
Lucknow, India,

A.A. Zilli
Department of CSE,
Integral University,
Lucknow, India

ABSTRACT-- This paper explore direct network architectures are strong entrant for use in many successful profitable, experimental multicomputers and scalable shared memory multiprocessors. The characteristics of direct networks, as reflected by the communication latency metric, are critical to the performance of such systems. In multiprocessor system use the wormhole routing for the most capable switching method, has been adopted in several new huge 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 will be more reliable and also the unreliability of the interconnection network gets minimized. This is very enviable aspect for the interconnection network as the network remains operational for more failure of neighboring nodes or links in parallel computer architecture. Formulations to optimize the performance of single queues, networks of queues of different parameters are minimization (arrival rate, departure rate, waiting time in queue and etc.)

Keywords - Hypercube network, Parameters, Embedded network, Scalability, Arrival rate, Departure rate, waiting time

I. INTRODUCTION

A lot of research effort has been dedicated during the last decade to improve the performance of multicomputer. A key architectural issue is the 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 interconnection networks is wormhole routers and the related routing algorithms.

The way the nodes 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. Fig.1shows a generic multiprocessor with a set of nodes inters connected via direct network or PE-interconnecting network *[1], [2], [3], and[4].*

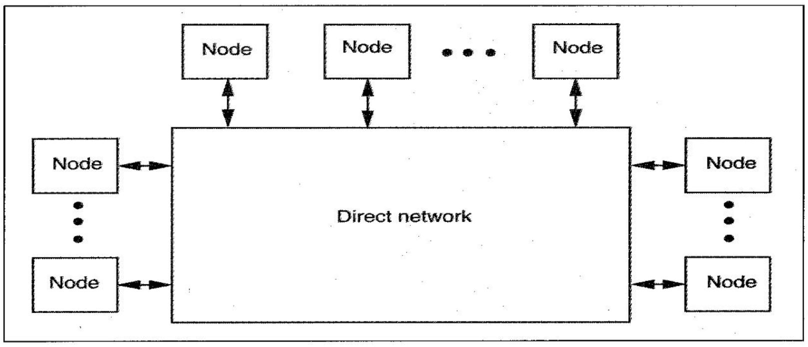

Fig.1 A general multiprocessor based on direct network

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. Although 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. 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 *[9], [10], and [11].*

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:

- n-dimensional mesh
- *k*-ary *n*-cube or torus
- 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 $k_i >= 2$ (nodes have from n to 2n neighbors, depending on their location in the mesh).

B. k- arry 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. A hypercube is an *n*-dimensional mesh in which $k_i = 2$ for all $0 \le i \le n-1$, i.e., a *2*-ary *n*-cube.

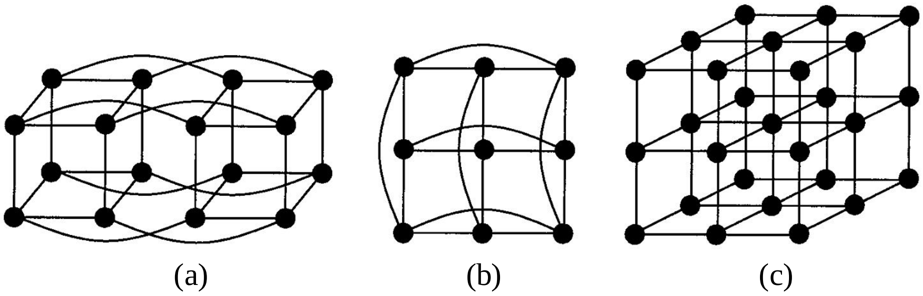
(a) (b) (c)

Fig. 2 Direct Network Topologies
(a) 2-ary 4-cube (hypercube), (b) 3-ary 2-cube (torus), (c) 3-ary 3D mesh

III. WORMHOLE ROUTING

Wormhole routing is a special 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) govern 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.

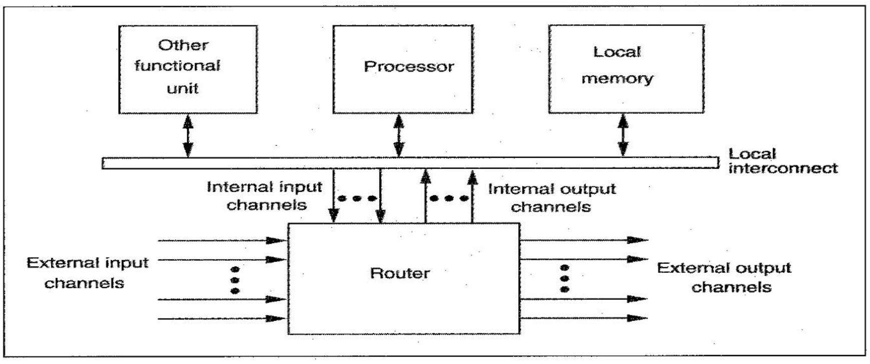

Fig. 3.1 Generic node architecture

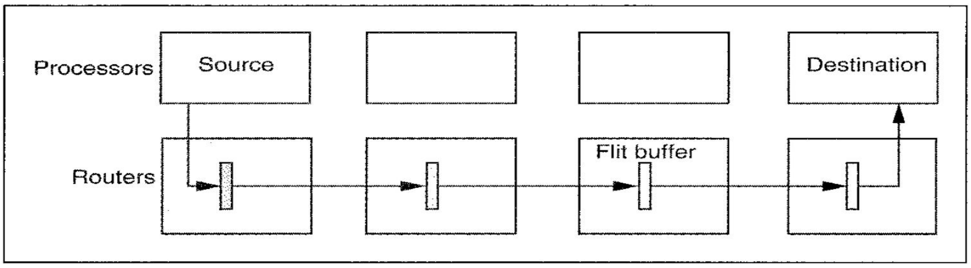

Fig. 3.2 Wormhole routing

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.

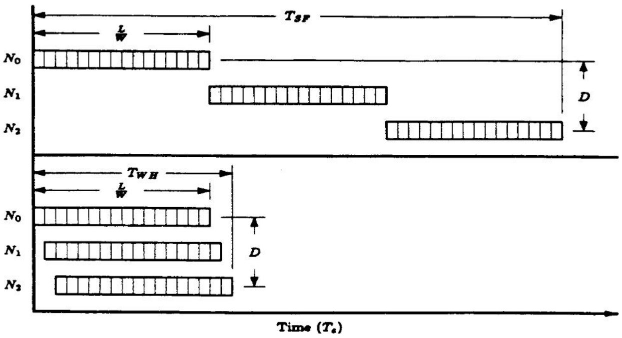

Fig. 3 Wormhole routing Vs store and forward

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]*.

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.

$$\mathbf{T_{SF} = T_c\,(D \times L/W)} \qquad (1)$$

The bottom half of fig.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.

$$\mathbf{T_{WF} = T_c\,(D + L/W)} \qquad (2)$$

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- Unicast: 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.

Adaptive: use information about network traffic and/or channel status to avoid congested faulty regions of the network.

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, Partially Adaptive

IV. NETWORK PPROPERTIES

In this section the properties of a k- arrry n cube network *[13]*-

Notation

In a k arry n cube network:

n: number of dimension
k: number of nodes per dimension

N: Total number of nodes
C: Total number of bidirectional channel
w: width of bidirectional channel
W: Total wiring of k-arry n-cube network
b: bisection width of a k-arry n-cube network
B: bisection density of a k-arry n-cube network
D: diameter of a k-arry n-cube network

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

Node degree**:** In a k-arry 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-arry 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-arry n-cube network.

In k-arry n-cube network, the channel width w is the number of data lines. In a physical bidirectional channel, w' **= nw for torus, w'=nw/2 hypercube.**

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

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

$$T_1 (i, j, k) = (i, (j+1) \,mod\, m, k) \qquad (3)$$
$$T_2 (i, j, k) = (i, (m+j-1) \,mod\, m, k) \qquad (4)$$
$$T_3 (i, j, k) = ((i+1) \,mod\, l, j, k) \qquad (5)$$
$$T_4 (i, j, k) = ((l+i-1) \,mod\, l, j, k) \qquad (6)$$
$$T_C (k_{n-1} \ldots\ldots k_{d+1}\, k_{d}\, k_{d-1} \ldots\ldots k_0)$$
$$= (k_{n-1} \ldots\ldots k_{d+1}\; k_{d}' \;_{d-1} \;\ldots k_0) \quad (7)$$

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
(i) $i_1 = i_2$ and
(ii) $j_1 = j_2$ and
(iii) k_1 and k_2 differ by one bit position in their binary address.

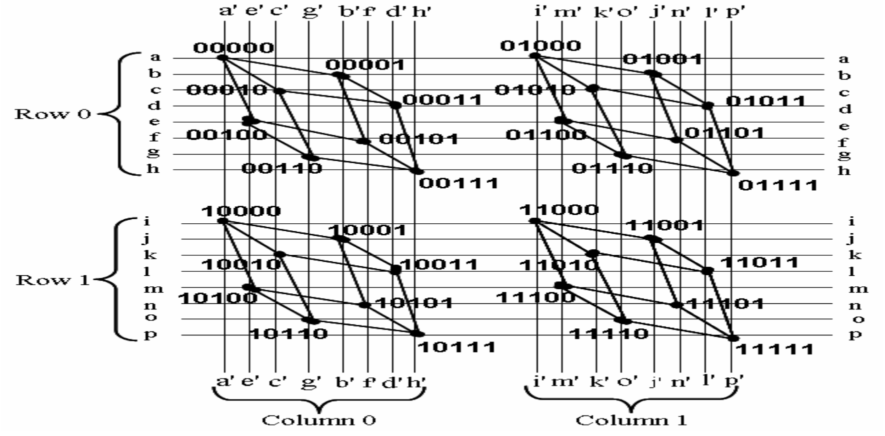

Fig. 4 A (2, 2, 8)-torus embedded hypercube Network

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
(i) $k_1 = k_2$ and
(ii) i and j differ by one in one component while the other component is identical.

IV. CONCEPT OF QUEUEING MODEL

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 in this section we give a quick review on the M/M/1 queue and priority queue concepts *[12]*.

A. 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 $(\bar{a}, \overline{a^2}, \bar{s}, \overline{s^2})$, where $\bar{a}$ and $\overline{a^2}$ are the first and second moments of the customers' inter_arrival time, and similarly, $\bar{s}$ 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 $\lambda = 1/\bar{a}$ and $\mu = 1/\bar{s}$, respectively. The mean delay time of a M/M/1 queuing model system can be approximate by Allen – Cunneen formula .

$$\bar{W}_{\mathrm{M/M}/1} \approx \frac{\rho\left(C_A^2 + C_S^2\right)}{2\mu(1-\rho)} \qquad (8)$$

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

B. 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.

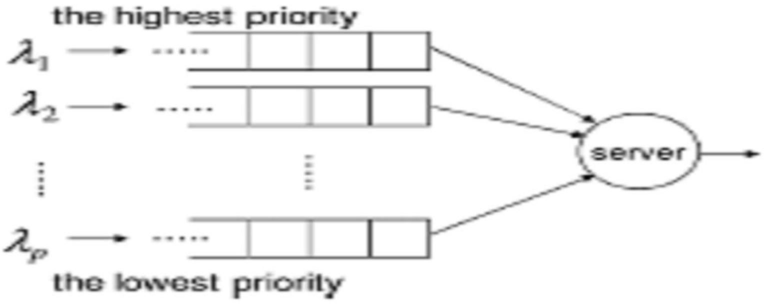

Fig. 5 Priority queuing system

In other words, the lower the index, the highest priority. The priority queuing system to be studied is depicted in Fig. 2, 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. The mean delay time of random arrivals to the ith queue W_i'an be written as *[22]:*

$$\bar{W}_i = \frac{\bar{R}}{\left(1-\Sigma_{k=1}^{i-1}\rho_k\right)\left(1-\Sigma_{k=1}^{i}\rho_k\right)} \qquad (9)$$

Where $\bar{R}$ is the residual service time seen by an incoming customer. In a M/M/1 queuing system, is approximated by [3]

$$\bar{R} \approx \Sigma_{k=1}^{p}\frac{\rho_k}{2\mu_k}\left(C_{A_k}^2 + C_{S_k}^2\right) \qquad (10)$$

Where μ_k and ρ_k are average service rate and utilization factor of class k, respectively.

C. OPTIMIZING FOR AVERAGE DELAY AND QUEUE OCCUPANCY

Queuing systems form a fundamental part for different types of networks, including computer multiprocessor networks and communications data networks. Queuing systems are also an integral part of various network elements, such as the input and output buffers of a packet switch. We often would like to optimize some performance metrics of queuing systems, for example, buffer occupancy, overall delay, jittering, workload, and probabilities of certain states. In a network of queues, we may also have multiple conflicting objectives that need to be optimally balanced. However, optimizing the performance of even simple queues like the *M=M=m=m* queue is in general a difficult problem because of the nonlinearity of the performance metrics as functions of the arrival and service rates. Nonlinear optimization in general takes running time that scales exponentially with the problem size.

We start the optimization formulations with a simple example of minimizing the service load of a *M / M /* 1 queue with constraints on average queuing delay *W*, total delay *D*, and queue occupancy *Q[14],[15],[16],[17],[18]:*

Proposition: The following nonlinear optimization is a geometric program, and therefore can be turned into a convex optimization and efficiently solved for its global optimum:

$$
\begin{array}{ll}
\text{Minimize} & \mu / \lambda \\
\text{Subject to} & W \leq W_{max}, \\
 & D \leq D_{max}, \\
 & Q \leq Q_{max}, \\
 & \lambda \geq \lambda_{min}, \\
 & \mu \geq \boldsymbol{\mu}_{max},
\end{array}
\qquad (11)
$$

Where the optimization variables are the arrival rate λ ¸ and the service rate μ. The constant parameters are the performance upper bounds W_{max}, D_{max} and Q_{max}, and practical constraints on the maximum service rate $\boldsymbol{\mu}_{max}$ of the queue that cannot be exceeded, and the minimum incoming traffic rate λ_{min} that must be supported. The objective is to minimize the service load. We can also show that even a joint optimization over both (λ , μ) and (W_{max}, D_{max} , Q_{max}) is still a geometric program. The above formulation can be extended to a Markovian queuing system with N queues sharing a pool of service rate bounded by $\boldsymbol{\mu}_{max}$ (for example, connected to a common outgoing link). The arrival rate to be supported for each individual queue i is bounded by λ i,min. There are delay and queue occupancy bounds $Wi;max$, $Di;max$ and $Qi;max$ for each queue i. The objective now becomes minimizing a weighted sum of the service loads for all the queues:

Corollary: The following nonlinear optimization is a geometric program:

$$
\text{Minimize} \quad \sum_{i=1}^{N} \mu_i / \lambda_i
$$

$$
\begin{array}{ll}
\text{Subject to} & Wi \leq Wi,max; \\
 & Di \leq Di,max; \\
 & Qi \leq Qi,max; \\
 & \lambda \geq \lambda_{i,\,min}, \\
 & \sum_{i=1}^{N} \mu_i \geq \boldsymbol{\mu}_{max}
\end{array}
\qquad (12)
$$

Where the optimization variables are the arrival rates λ_i and the service rates μ_i . A simple numerical example for $N = 2$ with weights $\alpha_1 = 1$; $\alpha_2 = 2$ is summarized as follows. If we set the delay and queue occupancy constraints as $Q1;max = 4;Q2;max = 5;W1;max = 2:5;W2;max = 3;D1;max = 2;D2;max = 2$, and service and arrival rate constraints as $\lambda\, 1;min = 0:5$; $\lambda\, 2;min = 0:8$; $\mu max = 3$, geometric programming gives the optimizers: $\mu_1^* = 1:328$; $\mu_2^* = 1:672$; $\mu_3^* = 0:828$; $\mu_4^* = 1:172$ and the optimized objective value is $4:457$.

V. CONCLUSION

Analysis of the dependability of embedded hypercube interconnection network has exposed that as the interconnection network is scaled up the network will be more reliable and also the unreliability of the interconnection network gets minimized. 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.

Formulations to optimize the performance of single queues, networks of queues, and large deviation theoretic bounds on blocking probability minimization. Based on various result like that communication latency metric, analysis of mesh embedded hypercube interconnecting network, reliability of torus embedded hypercube and we work to get optimize latency, density, minimize waiting time in queue and also unreliability interconnection gets minimize.

REFERENCES

[1] Zhaoyang 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*," NewYork 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, 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 Netrwork s-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]Mung Chiang, Arak Sutivong, and Stephen Boyd," *Efficient Nonlinear Optimizations of Queuing Systems*", Stanford University, CA 94305.

[15]S. Boyd and L. Vandenberghe, Convex Optimization Stanford University EE 364 Course Reader 2001.

[16]RS Yadav, MM Tripathi, "*Image encryption techniques: A critical comparison*", International Journal of Computer Science Engineering and Information, Vol 3, Issue 1, 2013.

[17]M. Chiang and S. Boyd, "Shannon *duality through Lagrange duality: efficient computation and free energy interpretations for channel capacity and rate distortion,*" Proc. 40th Allerton Conference, Oct. 2002.

[18] Mohammad husain, Mohammad haroon, "*Different types of systems model for Dynamic Load balancing*", international journal of engineering research technology Vol 2, Issue 3, 2013.