Analytical Performance Modelling of Adaptive Wormhole Routing in the Star Interconnection Network

Abbas Eslami Kiasari[1,2], Hamid Sarbazi-Azad[1,2], and Mohamed Ould-Khaoua[3]

[1]IPM School of Computer Science
Tehran, Iran
{kiasari,azad}@ipm.ir

[2]Sharif University of Technology
Dept. of Computer Engineering
Tehran, Iran

[3]University of Glasgow
Dept. of Computing Science
Glasgow G12 8RZ, UK
mohamed@dcs.gla.ac.uk

Abstract

The star graph was introduced as an attractive alternative to the well-known hypercube and its properties have been well studied in the past. Most of these studies have focused on topological properties and algorithmic aspects of this network. Although several analytical models have been proposed in the literature for different interconnection networks, none of them have dealt with star graphs. This paper proposes the first analytical model to predict message latency in wormhole-switched star interconnection networks with fully adaptive routing. The analysis focuses on a fully adaptive routing algorithm which has shown to be the most effective for star graphs. The results obtained from simulation experiments confirm that the proposed model exhibits a good accuracy under different operating conditions.

1. Introduction

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

Several researchers have recently proposed analytical models of popular interconnection networks, e.g. k-ary n-cubes, tori, hypercubes, and meshes [1][6][17]. The most difficult part in developing any analytical model of adaptive routing is the computation of the probability of message blocking at a given router due to the number of combinations that have to be considered when enumerating the number of paths that a message may have used to reach its current position in the network. Almost all studies on star interconnection networks focus on topological properties and algorithmic issues. There has been hardly any study on performance evaluation and analytical modelling of such networks. In this paper, we discuss performance issues of star graphs by introducing the first mathematical model to predict the average message latency as a performance measure in wormhole star networks using the high-performance routing algorithm proposed in [13].

The rest of this paper is organised as follows. In Section 2, the structure of star graph is described. In Section 3, adaptive wormhole routing in the star graph is discussed. Section 4 proposes a mathematical performance model for adaptive routing in wormhole star graph. Validation of the proposed performance model is realized in Section 5 using results obtained from simulation experiments. Finally, Section 6 concludes the paper.

2. The star graph

1-4244-0054-6/06/$20.00 ©2006 IEEE

Let V_n be the set of all $n!$ permutations of symbols 1, 2, 3,…, n. For any permutation $v \in V_n$, if we denote the i^{th} symbol of v by v_i, v can be written as $v_1v_2\ldots v_n$. A *star graph* defined on n symbols, $S_n = (V_n, E_n)$, is an undirected graph with $n!$ nodes, where each node v is connected to $n - 1$ nodes which can be obtained by interchanging the first and i^{th} symbols of v, i.e. $[v_1v_2\ldots v_iv_{i+1}\ldots v_n, v_iv_2\ldots v_1v_{i+1}\ldots v_n] \in E_n$, for $2 \leq i \leq n$. We call these $n - 1$ connections as *dimensions*. Thus each node is connected to $n - 1$ nodes through dimensions 2, 3,…,n.

The star graph is an attractive alternative to the hypercube [3], and compares favourably with it in several aspects [9]. For example, the degree and diameter of S_n is $n - 1$ and $\lfloor 3(n\text{-}1)/2 \rfloor$, i.e. sub-logarithmic in the number of nodes of S_n while a hypercube with $\Theta(n!)$ nodes has a degree and a diameter of $\Theta(\log n!) = \Theta(n\log n)$, i.e. logarithmic in the number of nodes. Much work has been done to study both the topological properties and parallel algorithms of the star graph in the past.

Each node, in the star graph, is uniquely indexed by an n-tuple using the n numbers corresponding to a permutation on the symbol set {1, 2, …, n}. We assume the adjacent nodes are connected by two unidirectional communication links (or a bi-directional channel). Each physical channel has some, say V, virtual channels that share the bandwidth of the physical channel in a multiplexed fashion. Also each input/output virtual channel has incoming/outgoing buffers. In this paper, a communication channel or communication link should be taken to mean a physical channel. Every physical channel, virtual channel, and message originating from a node can be given unique numbers based on the address of the node.

3. Adaptive wormhole routing in star graphs

Several fully adaptive routing algorithms on star graph have been evaluated in [13] of which the one using negative hop-based deadlock free routing, augmented with a new idea called bonus card, has shown to have the best performance.

In the negative-hop algorithm [5], the network is partitioned into several subsets, such that no subset contains adjacent nodes (this is equivalent to the well-known graph colouring problem). If C is the number of subsets, then the subsets are labelled as 0, 1, …, C-1, and nodes in subset i are labelled (or coloured) as i. A hop is a negative hop if it is from a node with a higher label to a node with a lower label; otherwise, it is a positive hop. A message occupies a buffer of virtual channel i at an intermediate node if and only if the message has taken exactly i negative hops to reach that intermediate node. If H is the diameter of the network and C is the number of colours, then the maximum number of negative hops that can be taken by a message is $H_N = \lceil H(C-1)/C \rceil$ [4] [5].

The structure of S_n is a bipartite graph, and its nodes can be partitioned into two subsets; therefore, it can be coloured using only two colours [5]. Because adjacent nodes are in distinct partitions, the maximum number of negative hops a message may take is at most half the diameter of S_n, which equals $\lceil H/2 \rceil = \lceil \lfloor 3(n-1)/2 \rfloor / 2 \rceil$. Hence, negative-hop schemes with $\lceil \lfloor 3(n-1)/2 \rfloor / 2 \rceil$ virtual channels per physical channel can be designed for S_n.

The negative-hop (*NHop*) algorithm has an unbalanced use of virtual channels because messages start their journey starting from virtual channel 0. However, very few messages take the maximum number of hops and use all the virtual channel 0...$\lceil \lfloor 3(n-1)/2 \rfloor / 2 \rceil$, and thus virtual channels with high numbers will be used rarely. The NHop scheme can be improved by giving each header flit a *number bonus card* [4]. For negative-hop scheme it is equal to the number of virtual channel level minus the number of required negative hops to reach the destination node. At each node, the header flit has some flexibility in the selection of virtual channels. The range of virtual channels that can be selected for each physical channel is equal to the number of bonus cards available plus one. The resulting deadlock-free routing algorithm using negative-hop routing scheme and the bonus card is named *Nbc* which has been well-evaluated and investigated against other routing algorithms for the star graphs in [13].

In [13], the Nbc routing scheme has been used and a routing algorithm, named *Enhanced-Nbc* with high performance and minimum virtual channel requirements was resulted. Investigations showed that Enhanced-Nbc has a better performance [13] compared to other algorithms reported in the literature and the other algorithms proposed in [16].

4. The analytical model

In this section, we derive an analytical performance model for wormhole fully adaptive routing in a star graph. Due to the superior performance of enhanced-Nbc algorithm [13], our analysis focuses on this routing algorithm but the modelling approach used here can be equally applied for other routing schemes after few changes in the model.

The measure of interest in our model is the average message latency as a representative for network performance.

The following assumptions are made when developing the proposed performance model. These assumptions have been widely used in similar modelling studies [1][6][7][10][11][14][17][18] Messages are broken into some packet of fixed length of M flits which are the unit of switching. The flit transfer time between any two routers is assumed to one cycle over physical channels.

a) Message destinations are uniformly distributed across the network nodes.

b) Nodes generate traffic independently of each other, and follow a Poisson process, with a mean rate of λ_g messages/cycle.

c) Messages are transferred to the local processor through the ejection channel once they arrive at their destination.

d) V virtual channels per physical channel are used. These virtual channels are used according to enhanced-Nbc routing algorithm.

The model computes the mean message latency as follows. First, the mean network latency, $\overline{S}$, that is the time to cross the network is determined. Then, the mean waiting time seen by a message in the source node to be injected into the network, $\overline{W_s}$, is evaluated. Finally, to model the effect of virtual channels multiplexing, the mean message latency is scaled by a factor, $\overline{V}$, representing the average degree of virtual channels multiplexing that takes place at a given physical channel [8]. Therefore, the mean message latency can be written as

$$Latency = (\overline{S} + \overline{W_s})\overline{V} \cdot \tag{1}$$

The average number of hops that a message makes across the network, $\overline{d}$, is given by [3]

$$\overline{d} = (n - 4 + \frac{2}{n} + \sum_{i=1}^{n} \frac{1}{i}) \times \frac{n!}{n!-1} \cdot \tag{2}$$

Fully adaptive routing allows a message to use any available channel that brings it closer to its destination resulting in an evenly distributed traffic rate on all network channels. A router in the S_n has n-1 output channels and the PE generates, on average, λ_g messages in a cycle. Since each message travels, on average, $\overline{d}$ hops to cross the network, the rate of messages received by each channel, λ_c, can be calculated as [2]:

$$\lambda_c = \frac{\lambda_g \overline{d}}{n-1} \cdot \tag{3}$$

Since the star graph is symmetric averaging the network latencies seen by the messages generated by only one node for all other nodes gives the mean message latency in the network. Let $S = 123\ldots n$ (identity permutation) be the source node with linear address 0 and i denotes linear address of the destination node, where $1 \le i \le n!-1$. The network latency, S_i, seen by the message crossing from node 0 to node i consists of two parts: one is the delay due to the actual message transmission time, and the other is due to the blocking time in the network. Therefore, S_i can be written as

$$S_i = M + h_i + \sum_{k=1}^{h_i} B_{i,k} \ , \tag{4}$$

where M is the message length, h_i is the distance between the node 0 and node i, and $B_{i,k}$ is the mean blocking time seen by a message form node 0 to node i on its k^{th} hop. Averaging over all the possible destination nodes destined made by a typical message yields the mean network latency as

$$\overline{S} = \frac{\sum_{i=1}^{n!-1} S_i}{n!-1} \cdot \tag{5}$$

Under the uniform traffic pattern and due to the symmetry of the star graph topology, adaptive routing results in an evenly distributed traffic rate on all network channels. Furthermore, a message sees the same mean waiting time and same mean service time across all channels regardless of their positions in the network. However, the message sees a different probability of blocking at each channel as the number of alternative paths, that can be selected, changes from one channel to the next along the path from the source to destination node. The probability of blocking depends on the number of output links, and thus on the virtual channels that a message can use at its next hop. We define $f(i,j,k)$ as the number of output channels for k^{th} hop of j^{th} path set (of all possible paths) for the destination node *i,*

Let $P_{block_{i,k}}$ be the average probability blocking seen by a message form node 0 to node i on its k^{th} hop, and w be the mean waiting time when blocking occurs. The mean blocking time, $B_{i,k}$, is giving by

$$B_{i,k} = P_{block_{i,k}} w \ \cdot \tag{6}$$

The probability of blocking, $P_{block_{i,k}}$, can therefore be calculated as

$$P_{block_{i,k}} = \frac{1}{N_{set_i}} \sum_{j=1}^{N_{set_i}} P_{block_{i,j,k}} \ , \tag{7}$$

where $P_{block_{i,j,k}}$ is the probability of blocking for k^{th} hop of j^{th} path set for the destination node i and N_{set_i} is the number of path sets for the destination node *i.*

A message is blocked at a given channel when all the adaptive virtual channels of class a and also $V_2 - \left\lceil d/2 \right\rceil + 1$ virtual channels of class b (that are used for Nbc routing algorithm) are busy, where d is the number of remaining hops to the destination node. When blocking occurs a message has to wait for all V_1 fully adaptive virtual channels of class a and $V_2 - \left\lceil d/2 \right\rceil + 1$ virtual channels of class b [13]. For computing $P_{block_{i,j,k}}$ messages are divided into 3 groups based on the last hop and next hop.

***Group A*:** This group contains messages that used a virtual channel of class *a* in their last hop (with probability of blocking P_{block_A}).

Group $\boldsymbol{B^-}$**:** This group contains messages that used a virtual channel of class *b* in the last hop and the next hop is negative (with probability of blocking $P_{block_{B^-}}$).

Group $\boldsymbol{B^+}$**:** This group contains messages that used a virtual channel of class *b* in the last hop and the next hop is positive (with probability of blocking $P_{block_{B^+}}$).

Since the number of messages in group B^- is equal to the number of messages in group B^+, then we can write the probability of blocking as

$$P_{block_{i,j,k}} = \left(P_{block_A} + \frac{1}{2} P_{block_{B^-}} + \frac{1}{2} P_{block_{B^+}} \right)^{f(i,j,k)} \tag{8}$$

Note that *f*(*i,j,k*) is the number of output channels for k^{th} hop in the j^{th} path set for destination node *i,* 1≤ *i*≤ *n*!.

Let us now compute P_{block_A}, $P_{block_{B^-}}$ and $P_{block_{B^+}}$. When a message used a virtual channel from class *a* in its last hop, it can use any of V_1 virtual channels of class *a* and also $V_2 - \left\lceil d/2 \right\rceil + 1$ virtual channel of class *b*. Therefore blocking occurs, when all $V - \left\lceil d/2 \right\rceil + 1$ virtual channels of selectable physical channel are busy. Now, we can write

$$P_{block_A} = \sum_{v = V - \left\lceil d/2 \right\rceil + 1}^{V} P_{vc_0} P_v \frac{\begin{pmatrix} \left\lceil d/2 \right\rceil - 1 \\ v - V + \left\lceil d/2 \right\rceil - 1 \end{pmatrix}}{\begin{pmatrix} V \\ v \end{pmatrix}}, \tag{9}$$

where P_v, $0 \le v \le V$ and P_{vc_0} are the probability that *v* virtual channels at a physical channel are busy, and the probability that the message use a virtual channel of class *a* in the last hop, respectively.

Also, a message from group B^- and B^+ that used virtual channel *l* at the last hop, can use any of V_1 virtual channels of class *a* and also $V_2 - \left\lceil d/2 \right\rceil - l + 1$ and $V_2 - \left\lceil d/2 \right\rceil - l + 2$ virtual channels of class *b*, respectively. Thus,

$$P_{block_{B^-}} = \sum_{l=1}^{V_2 - \left\lceil d/2 \right\rceil} \sum_{v = V_1 + n_l}^{V} P_{vc_l} P_v \frac{\begin{pmatrix} V_2 - n_l \\ v - V_1 - n_l \end{pmatrix}}{\begin{pmatrix} V \\ v \end{pmatrix}}, \tag{10}$$

and

$$P_{block_{B^+}} = \sum_{l=1}^{V_2 - \left\lceil d/2 \right\rceil + 1} \sum_{v = V_1 + n_l + 1}^{V} P_{vc_l} P_v \frac{\begin{pmatrix} V_2 - n_l - 1 \\ v - V_1 - n_l - 1 \end{pmatrix}}{\begin{pmatrix} V \\ v \end{pmatrix}}, \tag{11}$$

where P_{vc_l} represents the probability that a message uses a virtual channel of class *b* with number *l* in its last hop. To determine the mean waiting time, *w*, to acquire a virtual channel a physical channel is treated as an M/G/1 queue with a mean waiting time of [15]

$$w = \frac{\rho \overline{S} (1 + C_{\overline{S}}^2)}{2(1 - \rho)} \tag{12}$$

$$\rho = \lambda_c \overline{S} \tag{13}$$

$$C_{\overline{S}}^2 = \frac{\sigma_{\overline{S}}^2}{\overline{S}^2} \tag{14}$$

where λ_c is the traffic rate on the channel given by equation 3, $\overline{S}$ is its service time calculated by equation 5, and $\sigma_{\overline{S}}^2$ is the variance of the service time distribution. Since the minimum service time at a channel is equal to the message length, *M*, following a suggestion given in [10], the variance of the service time distribution can be approximated as $\sigma_{\overline{S}}^2 = (\overline{S} - M)^2$. Hence, the mean waiting time becomes

$$w = \frac{\lambda_c \overline{S}^2 (1 + (1 - M / \overline{S})^2)}{2(1 - \lambda_c \overline{S})} \tag{15}$$

Similarly, modelling the local queue in the source node as an M/G/1 queue, with the mean arrival rate λ_g / V and service time $\overline{S}$ with an approximated variance $(\overline{S} - M)^2$ yields the mean waiting time seen by a message at the source node as [15]

$$W_s = \frac{\frac{\lambda_g}{V} \overline{S}^2 (1 + (1 - M / \overline{S})^2)}{2(1 - \frac{\lambda_g}{V} \overline{S})} \tag{16}$$

The probability, P_v, that *v* virtual channels are busy at a physical channel can be determined using a Markovian model. State π_v (0≤ *v*≤ *V)* corresponds to *v* virtual channels being busy. The transition rate out of state π_v to state π_v+1 is the traffic rate λ_c (given by equation 7) while the rate out of state π_v to state π_{v-1} is $\frac{1}{\overline{S}}$ ($\overline{S}$ is given by equation 5). The transition rates out of state π_v are reduced by λ_c to account for the arrival of messages while a channel is in this state.

The Markovian model results in the following steady state probability (derivation explained in [15]), in which the service time of a channel has been approximated as the network latency of that channel:

$$P_v = \begin{cases} (1-\lambda_c \overline{S})(\lambda_c \overline{S})^v, & 0 \le v < V \\ (\lambda_c \overline{S})^v, & v = V . \end{cases} \qquad (18)$$

When multiple virtual channels are used per physical channel, they share the physical bandwidth in a time-multiplexed manner. The average degree of multiplexing of virtual channels, that takes place at a given physical channel, can then be estimated by [8]:

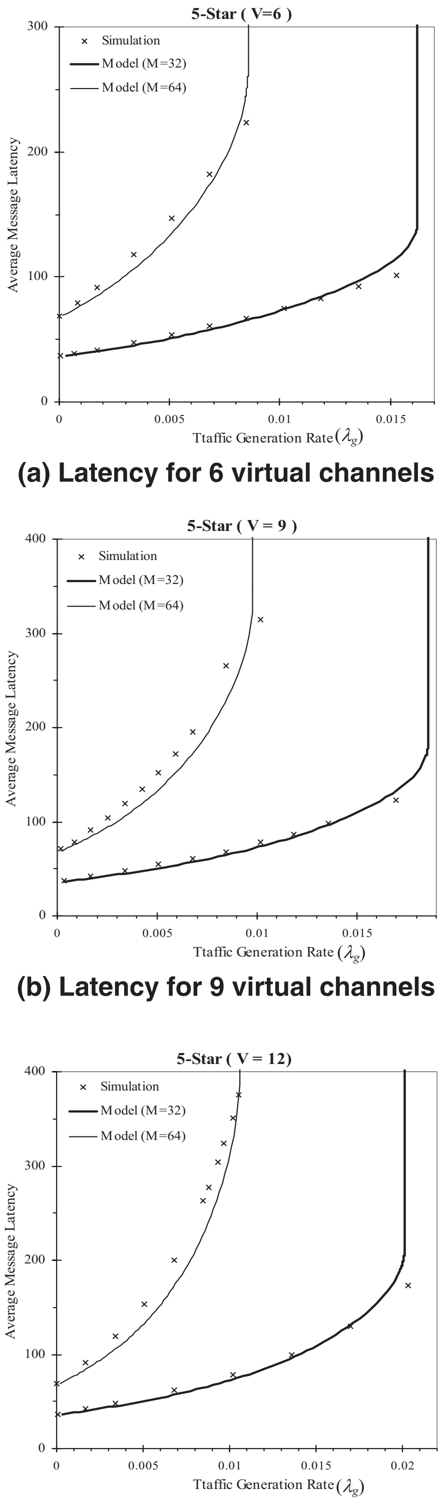
$$\overline{V} = \frac{\sum_{v=1}^{V} v^2 p_v}{\sum_{v=1}^{V} v p_v} . \qquad (19)$$

The above equations reveal that there are several inter-dependencies between the different variables of the model. For instance, Equations 4, 5 and 6 reveal that $\overline{S}$ is a function of w while equation 12 shows that w is a function of $\overline{S}$. Given that closed-form solutions to such inter-dependencies are very difficult to determine, the different variables of the model are computed using an iterative technique.

5. Validation of the model

The proposed analytical model has been validated through a discrete-event simulator that mimics the behaviour of the described routing algorithms in the network at the flit level. The simulator uses the same assumptions as the analysis, and some of these assumptions are detailed here with a view to making the network operation clearer. The network cycle time is defined as the transmission time of a single flit from one router to the next. Messages are generated at each node according to a Poisson process with a mean inter-arrival rate of λ_g messages/cycle. Message length is fixed at M flits. Destination nodes are determined using a uniform random number generator. The mean message latency is defined as the mean amount of time from the generation of a message until the last data flit reaches the local processor at the destination node. The other measures include the mean network latency, the time taken to cross the network, the mean queuing time at the source node, and the time spent at the local queue before entering the first network channel. Numerous validation experiments have been performed for several combinations of network sizes, message lengths, and number of virtual channels to validate the model.

Figure 1 depict latency results predicted by the model explained in the previous section, plotted against those provided by the simulator for the S_5 interconnection networks with 120 nodes, with V = 6, 9 and 12 virtual channels per physical channel, and 2 different message



(a) Latency for 6 virtual channels

(b) Latency for 9 virtual channels

(c) Latency for 12 virtual channels

Figure 1: The average message latency predicted by the model against simulation results for a 5-Star with (a) *V*=6 (b) *V*= 9 and (c) *V*=12 virtual channels and messages length *M*=32 and 64 flits.

lengths M=32 and 64 flits. The horizontal axis in the figure shows the traffic generation rate at each node while the vertical axis shows the mean message latency. The figures reveal that in all cases the analytical model predicts the mean message latency with a good degree of accuracy in the steady-state regions. Moreover, the model predictions are still good even when the network operates in the heavy traffic region, and when it starts to approach the saturation region. However, some discrepancies around the saturation point are apparent. These can be accounted for by the approximations made to ease the derivation of different variables, e.g. the approximation made to estimate the variance of the service time distribution at a channel. Such an approximation greatly simplifies the model as it allows us to avoid computing the exact distribution of the message service time at a given channel, which is not a straightforward task due to inter-dependencies between service times at successive channels as wormhole routing relies on a blocking mechanism for flow control.

6. Conclusion and future work

Star interconnection networks have gained much attention during the last decade. However, most of studies in this line have focused on topological properties and algorithmic aspects of these networks. In this paper, we introduced the first mathematical performance model of adaptive wormhole-routed star graphs and validated it through simulation experiments. We saw that the proposed model manages to achieve a good degree of accuracy while maintaining simplicity, making it a practical evaluation tool that can be used by the researchers in the field to gain insight into the performance behaviour of fully adaptive routing in wormhole-routed star graphs.

Our next objective is to compare the performance merits of the star graphs and their equivalent hypercubes under different technological constraints to conduct a fair comparison.

References

[1] S. Abraham and K. Padmanabhan. Performance of the direct binary n-cube networks for multiprocessors. *IEEE Transactions on Computers*, 37(7):1000-1011, 1989.

[2] A. Agarwal. Limits on interconnection network performance. *IEEE Transactions on Parallel and Distributed Systems*, 2(4):398-412, 1991.

[3] S. B. Akers, D. Harel and B. Krishnamurthy. The Star Graph: An Attractive Alternative to the *n*-cube. *Proceedings of International Conference on Parallel Processing*, pages 393-400, 1987.

[4] R. V. Boppana and S. Chalasani. A Comparison of Adaptive Wormhole Routing Algorithms. *Proceedings of the 20th Annual International Symposium on Computer Architecture (ISCA'93)*, pages 351-360, 1993.

[5] R. V. Boppana and S. Chalasani, A Framework for Designing Deadlock-Free Wormhole Routing Algorithms. *IEEE Transactions on Parallel and Distributed Systems*, 7(2):169-183, 1996.

[6] Y. Boura, C. R. Das and T. M. Jacob. A performance model for adaptive routing in hypercubes. *Proceedings of the International Workshop on Parallel Processing*, pages 11-16, 1994.

[7] B. Ciciani, M. Colajanni and C. Paolucci. An accurate model for the performance analysis of deterministic wormhole routing. *Proceedings of the 11th International Parallel Processing Symposium*, pages 353-359, 1997.

[8] W. J. Dally. Virtual channel flow control. *IEEE Transactions on Parallel and Distributed Systems,* 3(2):194-205, 1992.

[9] K. Day and A. Tripathi. A Comparative Study of Topological Properties of Hypercubes and Star Graphs. *IEEE Transactions on Parallel and Distributed Systems*, 5(1):31–38, 1994.

[10] J. T. Draper and J. Ghosh. A Comprehensive analytical model for wormhole routing in multicomputer systems. *Journal of Parallel and Distributed Computing* 23(2): 202-214, 1994.

[11] R. Greenberg and L. Guan. Modelling and comparison of wormhole routed mesh and torus networks. *Proceedings of the 9th IASTED International Conference on Parallel and Distributed Computing and Systems*, pages 501-506, 1997.

[12] R. E. Kessler and J. L. Schwarzmeier. CRAY T3D: A new dimension for Cray Research. *CompCon*, pages 176-182, 1993.

[13] A. E. Kiasari, H. Sarbazi-Azad and M. Rezazad. Performance Comparison of Adaptive Routing Algorithms in the Star Interconnection Network. *Proceedings of the 8th International Conference on High Performance Computing in Asia Pacific Region (HPC-Asia'05),* pages 257-264, 2005.

[14] J. Kim and C. R. Das. Hypercube communication delay with wormhole routing. *IEEE Transactions on Computers,* 43(7):806-814, 1994.

[15] L. Kleinrock. Queueing Systems. Volume 1, John Wiley, New York, 1975.

[16] J. V. Misic and Z. Jovanovic. Routing Function and Deadlock Avoidance in a Star Graph Interconnection Network. *Journal of Parallel and Distributed Computing* 22(2):216-228, 1994.

[17] H. H. Najafabadi, H. Sarbazi-Azad and P. Rajabzadeh. Performance Modeling of Fully Adaptive Wormhole Routing in 2-D Mesh-Connected Multiprocessors. *Proceedings of the 12th Annual Meet-*

ing of the IEEE / ACM International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS'04), pages 528-534, 2004.

[18] H. Sarbazi-Azad, M. Ould-Khaoua and L. M. Mackenzie. An accurate analytical model of adaptive wormhole routing in *k*-ary *n*-cube interconnection networks. *Performance Evaluation*, 43(2):165-179, 2001.