

# Analysis of Buffer Arrangements in Low and High Dimensional Networks

N. Alzeidi

Department of Computer Science, College of Science, Sultan Qaboos University, P.O. 36, Al-Khodh, Muscat-123, Sultanate of Oman

Received 2 May 2009; accepted 11 March 2010

تحليل ترتيبات المصدات في الشبكات ذات الابعاد الواطئة والعالية  
ن. الزيدي

الخلاصة: أدخلت القنوات الظاهرية على شبكات الور مهول من اجل تحسين الاداء. تتكون هذه القنوات الظاهرية عن طريق توزيع كمية المساحة التخزينية المؤقتة المخصصة للوصلات الفيزيائية الى عدة مساحات متوازية. تشترك هذه المساحات المتوازية في نقل البيانات على نفس الوصلة المادية بحسب الحاجة لكل منها. كانت هناك محاولات قليلة لدراسة الطريقة الانسب لتوزيع هذه المساحة المؤقتة على القنوات الظاهرية الا ان هذه الدراسات اعتمدت على المحاكاة وركزت على التوجيه الثابت للبيانات. في هذه الدراسة نستخدم نماذج رياضية لتحليل الاداء لدراسة الطريقة الانسب لتكوين القنوات الظاهرية عند استخدام التوجيه المتكيف لنقل البيانات في شبكات الور مهول المترابطة.

المفردات المفتاحية: الشبكات، القنوات الظاهرية، المساحة التخزينية المؤقتة والمحدودة، تحليل الاداء

**Abstract:** Virtual channels have been introduced to enhance the performance of wormhole-switched networks. They are formed by arranging the buffer space dedicated to a given physical channel into multiple parallel buffers that share the physical bandwidth on a demand driven time-multiplexed manner. The question to be answered is: given a fixed amount of finite buffer what is the optimal way to arrange it into virtual channels. There have been few studies attempting to address this issue, however, these studies have so far resorted to simulation experiments and focused on deterministic routing algorithms. In this paper we use analytical performance models to investigate the optimal arrangement of the available buffer space into multiple virtual channels when adaptive routing is used in wormhole-switched  $k$ -ary  $n$ -cubes.

**Keywords:** Interconnection networks, Wormhole switching, Virtual channels, finite buffers, Performance analysis

## 1. Introduction

The performance of most digital systems today is often limited by communication or interconnection, not by logic or memory (Dally and Towles, 2004). Interconnection networks, the hardware fabric supporting communication among individual components of these systems, have been developed as an effective solution to this communication bottleneck. Hence, the performance of the interconnection network must be

fully understood and analysed to harness the full computational power of any high-performance computing system. Among other factors, the performance of an interconnection network is greatly influenced by its topology, switching method and routing algorithm. These factors are now considered in turn.

In the  $k$ -ary  $n$ -cube topology, nodes are arranged in  $n$  dimensional grid with  $k$  nodes per dimension. Each node consists of a processing element, PE, (with some memory) and a router (to handle message transmis-

\*Corresponding author's e-mail: alzidi@squ.edu.om

sion). Injection and ejection channels connect the router to the local PE where messages generated at the local PE are transferred to the network via the injection channel, while messages destined to the local PE are consumed from the network through the ejection channel. Each node's router is connected to a neighboring node in each dimension using input and output channels. The input and output channels are connected by a crossbar switch that can simultaneously connect multiple inputs and to multiple outputs (Dally and Towles, 2004 and Duato *et al.* 2003).

A critical requirement for any routing algorithm is to avoid deadlock situations, which occur when messages cannot advance towards their destinations. Deterministic routing always route messages through a predefined path. It, consequently, cannot take advantage of the path diversity that is usually provided by the network. In adaptive routing, on the other hand, messages can use any of the available paths between a given pair of nodes (Mohapatra, 1998). Duato's fully adaptive routing has been widely deployed in practical systems because it requires a minimum number of virtual channels (Duato *et al.* 2003).

The switching method dictates when and how messages get access to network resources, namely physical links and buffers. In wormhole switching (Duato *et al.* 2003), a message is divided into small flits for transmission and flow control. The header flit governs the route of the message and the remaining data flits follow it in a pipeline fashion. When the header flit is blocked due to contention for output channels or due to insufficient buffer space, all other data flits wait at their current nodes forming a chain of flits that spans over multiple nodes. Wormhole is an attractive switching method as it makes the end-to-end delay insensitive to the message destination due to the pipelining of flits. Another advantage of wormhole switching especially for system level and on-chip networks is the simplicity of router design as switches requires only minimum amount of buffer space. However, as traffic increases, the performance of wormhole switching can degrade significantly due to the chains of blocked messages. To overcome this, the flit buffers associated with every physical channel are arranged into several smaller size buffers forming virtual channels (Dally 1992). Each virtual channel then will has its own flit buffer and control logic and they all compete with each other for the same physical bandwidth in a time-multiplexing manner.

There have been few attempts to study the optimal arrangement of virtual channels (*ie.* given a fixed amount of finite buffer, what is the optimal way to arrange it into virtual channels). Dally 1992, concluded that with the total amount of buffer per physical channel held constant, adding virtual channels to a net-

work is significantly a more effective use of buffer space than adding depth to a single virtual channel. More recently, (Rezazad and Sarbazi-azad 2005 and Rezazad and Sarbazi-Azad, 2005), have conducted a another simulation study and arrived at similar conclusions. They have identified that increasing the buffer depth, and therefore decreasing the number of virtual channels, results in better performance for low traffic rates while the converse (*ie.* increasing the number of virtual channels can improve the performance of the network) is true under high traffic conditions. All of these studies (Dally 1992; Rezazad and Sarbazi-azad 2005 and Rezazad and Sarbazi-Azad, 2005) have suggested that increasing the number of virtual channels beyond a threshold value have adverse effect in network performance. For example, in a 16-ary 2-cube network with constant message size and constant buffer size of 32 flits, the author in (Dally, 1992) has observed that the best performance is achieved when the available buffer space is arranged into 4 virtual channels (each with 8 flits buffers), suggesting an optimal number of virtual channels for performance-cost tradeoffs.

In this paper we use recently developed analytical models (Alzeidi *et al.* 2006 and Alzeidi *et al.* 2007) of adaptive routing in  $k$ -ary  $n$ -cubes to conduct the first analytical performance comparison of deep versus parallel buffers in wormhole-switched  $k$ -ary  $n$ -cubes when the total amount of buffer associated with each physical channel is kept constant. The assumption of keeping the physical buffer size constant is necessary to give a fair cost-performance comparison.

The rest of this paper is organized as follows. Section 2 provides necessary background information about the topologies studied and the used routing algorithm. Section 3 lists the assumptions used in the analysis and the main equations of the analytical model while Section 4 presents the cost-performance model. Section 5 compares the performance merits of  $k$ -ary  $n$ -cubes with different virtual channels arrangements. Finally, Section 6 concludes the paper.

## 2. Preliminaries

This paper considers two widely studied networks; hypercube (a high dimensional network) and a 2D torus (low dimensional network). In this section we first briefly highlight their topological properties and then introduce the adaptive wormhole routing used in the comparisons.

### 2.1 The Hypercube and Torus Networks

An  $n$ -dimensional hypercube ( $H_n$ ) packs  $2^n$  nodes into  $n$  dimensions which are collectively connected to

each other via  $n \times 2^n$  channels. Each node in the hypercube is addressed with  $n$  digits binary string and linked to  $n$  neighboring nodes (one node in each dimension) through  $n$  channels. Two nodes  $X = x_n x_{n-1} \dots x_1$  and  $Y = y_n y_{n-1} \dots y_1$  are said to be connected (ie. neighbors) via a channel in dimension  $j$  if their addresses differ at bit position  $j$  only (ie.  $X_j \neq Y_j$ ). An edge in  $H_n$  can also be represented by an  $n$ -character string with a hyphen (-) and  $n - 1$  binary digits. For example in  $H_4$ , the string 00-1 denotes the edge at dimension 2, connecting nodes 0001 and 0011. The hypercube is characterized by its low diameter (ie. the maximum value of the minimum distance between any pair of nodes) which is equal to  $n$ .

In the 2-dimensional torus (or  $k$ -ary 2-cube), there are  $N = k^2$  nodes arranged into two dimensions (referred to as  $X$  and  $Y$  dimensions) with  $k$  nodes per dimension. Each node can be identified by its  $(x, y)$  coordinates where  $x$  and  $y$  represent the node position in the  $X$  and  $Y$  dimensions respectively. Nodes with address  $(x_1, y_1)$  and  $(x_2, y_2)$  in the torus are connected if and only if  $x_1 = x_2 \pm 1$  and  $y_1 = y_2 \pm 1$  (Dally and Towles, 2004 and Duato *et al.* 2002). Thus, each node is connected to two neighboring nodes in each dimension.

As can be seen from Fig. 1 (c), a node in a hypercube or a torus network consists of a processing element (PE) and a router. The PE contains a processor and some local memory. The router has input and output channels and is responsible for forwarding packets to their destinations according to the used routing algorithm. The PE is connected to the router to inject/eject messages to/from the network. The router contains finite amount of flit buffers for each input. The input and output channels are connected by a crossbar switch that can simultaneously connect multiple inputs to multiple output channels. Figure 1 illustrates a hypercube and a torus network with their node structure.

## 2.2 Adaptive Wormhole Routing

Routing is the process of determining which path a message should take to advance from its source to its destination. A critical requirement for any routing algorithm is to avoid deadlock situations, which occur when messages can not advance towards their destinations. Adaptive routing algorithms, which enable messages to explore alternative network paths, have been suggested to overcome the performance limitation of deterministic routing (Dally and Towles, 2004; Duato *et al.* 2002 and Mohapatra, 1998). In adaptive routing, messages reaching a given router have typically several alternative channels to choose from, consequently improving the performance by balancing evenly the traffic in the network channels.

Duato's algorithm (Duato, 1993 and Duato *et al.* 2003) has been studied extensively (Boura *et al.* 1994; Duato *et al.* 2002; Mohapatra 1998; Ould-Khaoua, 1999 and Sarbazi-Azad *et al.* 2001) and adopted widely in practical systems like the IBM Blue Gene/L (Adiga *et al.* 2005), Cray T3E (Scott and Thorson, 1996) and the Reliable Router (Dally *et al.* 1994). In this algorithm, the virtual channels divide the network into two separate virtual networks. A message is routed adaptively without any restriction in the first virtual network. If the message is blocked, it switches to using virtual channels in the second virtual network, which is deadlock-free and therefore provides escape routes for messages to break any deadlock that may occur in the first virtual network. A routing sub-function is a restriction of a routing algorithm that supplies these escape channels needed by a deadlock-free adaptive routing algorithm. Such a routing sub-function can itself be deterministic (Duato *et al.* 2003).

For the 2D torus, the algorithm requires  $V$ , ( $V > 2$ ) virtual channels per physical channel, which are split into two sets:  $VC_1 = \{v_3, v_4, \dots, v_V\}$  and  $VC_2 = \{v_1, v_2\}$ . The two virtual channels in  $VC_2$  (also called deterministic virtual channels) are used to implement a deadlock-free routing sub-function (ie. escape routes). The other virtual channels in  $VC_1$  (also called adaptive virtual channels), can be visited adaptively in any order that brings the message closer to its destination. At any routing step, a message firstly checks the adaptive virtual channels (channels in  $VC_1$ ) of the remaining dimension to be visited. If more than one adaptive virtual channels are available, one of them is chosen randomly to route through. If all virtual channels in  $VC_1$  are busy, the message is routed through the deterministic virtual channels (channels in  $VC_2$ ) of the lowest (or alternatively highest) dimension to be visited. If the deterministic virtual channel is also busy, then the message is blocked and waits for that virtual channel to become free.

For the hypercube, the same methodology is applied but the virtual channels in  $VC_2$  can be reduced to one. That is, the virtual channels are divided into two sets:  $VC_1 = \{v_3, v_4, \dots, v_V\}$  and  $VC_2 = \{v_1\}$ . This is because only one virtual channel is needed to implement a deadlock-free deterministic routing sub-function for the hypercube (Duato *et al.* 2003). As described earlier, the remaining virtual channels in  $VC_1$  are used adaptively by messages to get closer to their destinations. If all adaptive virtual channels are busy, then the message is routed through the deterministic virtual channel.

## 3. The Analytical Model

The presented analysis uses the analytical models

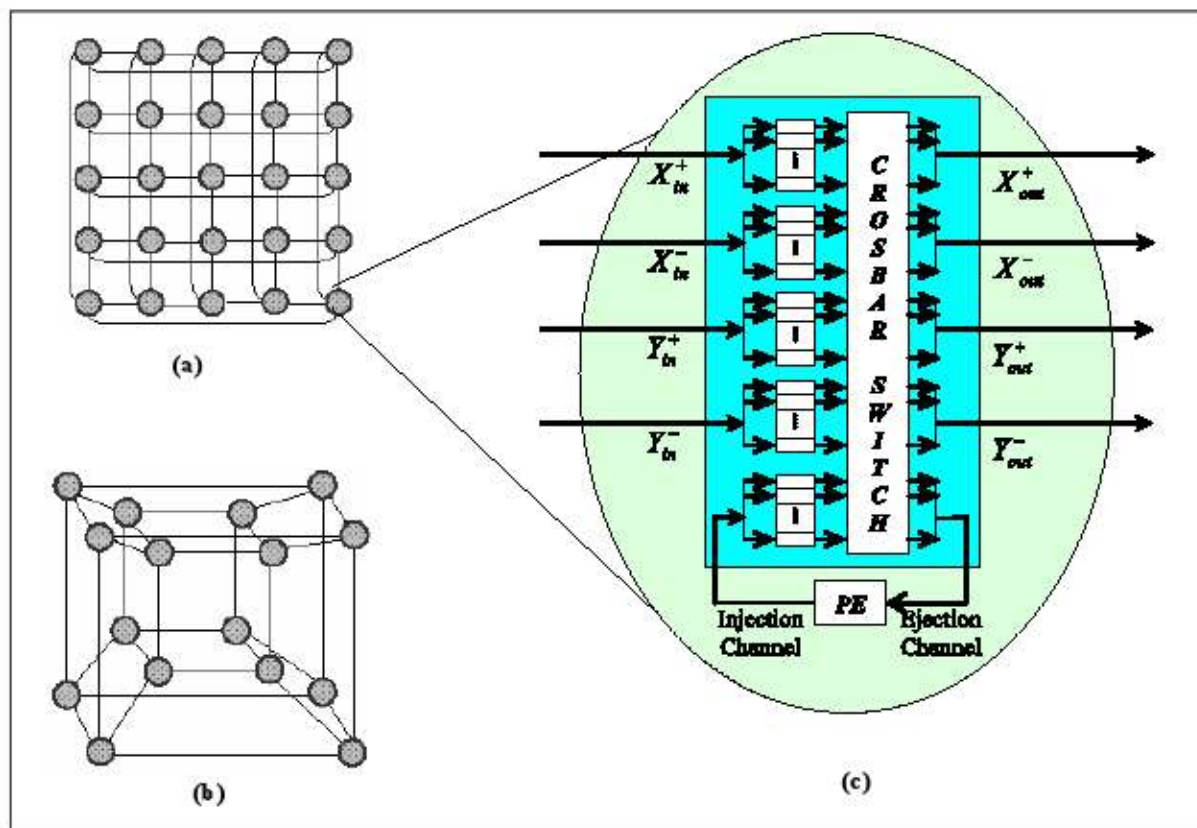


Figure 1 (a) 2D torus (5 x 5 nodes, (b) 4D hypercube (or 4-cube) and (c) their node structure.

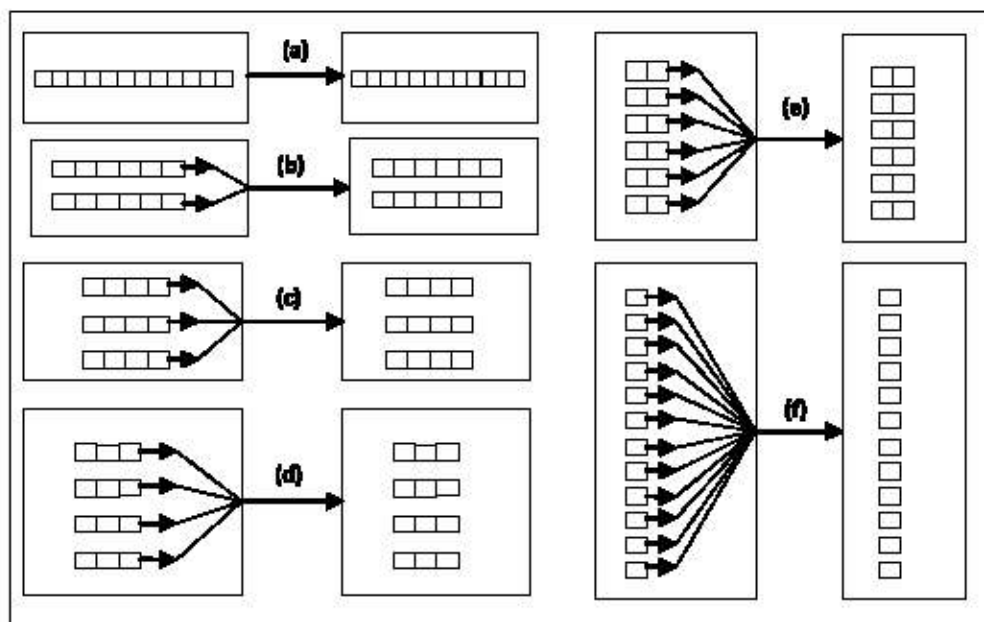


Figure 2. Arranging a 12-flit buffer in several ways; (a) when no virtual channels are used, the buffer is organised as one queue, while networks using virtual channels may organise it into several arrangements each with different queue size, namely (b) 2 x 6 flits, (c) 3 x 4 flits, (d) 4 x 3 flits, (e) 6 x 2 flits, and 12 x 1 flit.

developed in (Alzeidi *et al.* 2006). These models capture the behavior of adaptive routing in the torus and hypercube networks when wormhole switching is used with virtual channels and finite amount of flit buffers. The model in (Alzeidi *et al.* 2007) presented a new technique for computing the virtual channels occupancy probabilities while the model in (Alzeidi *et al.* 2006 and Alzeidi *et al.* 2007) concentrates in capturing the effects of finite buffers in adaptive routing. The models were based on the following assumptions which have been widely used in existing studies (Chien, 1998; Dally 1992; Duato and Lopez, 1994; Miller and Najjar, 1997; Rezazad and Sarbazi-azad, 2005 and Rezazad and Sarbazi-azad, 2005).

- Message length is fixed and is equal to  $M$  flits each of which takes one network cycle to cross from one router to the next using wormhole switching.
- Messages are routed according to Duato's adaptive routing and message destinations are uniformly distributed across the network nodes.
- Nodes generate traffic independently of each other and according to a Poisson process with a mean rate of  $\lambda_s$  messages per cycle.
- The local queue at the ejection channel in the source node has infinite capacity. Moreover, messages are transferred to the local processing element immediately after they arrive to their destination.

In these models, the mean network latency,  $\bar{S}$ , (*ie.* the time required by a message to cross the network from the source to the destination), is first calculated. Then the waiting time at the queue of the source node,  $\bar{W}$ , is approximated using an M/G/1 queue. Finally the average degree of multiplexing,  $\bar{V}$ , is obtained using a Markov model to capture the effects of multiplexing multiple virtual channels in a physical channel. Hence, the mean message latency is expressed as:

$$\text{Mean Message Latency} = (\bar{S} + \bar{W}) \bar{V} \quad (1)$$

The network latency seen by a message crossing from node  $S$  to node  $D$  is composed of three components: the message transmission time  $M$ , the switching time at each router, and the blocking delay. Hence, it can be written as:

$$S_H = |H| + M + \sum_{j=1}^{|H|} B_j \quad (2)$$

Averaging over the  $N-1$  possible destination nodes in the network yields the mean network latency as

$$\bar{S} = \frac{1}{N-1} \sum_{D \in G - \{S\}} S_H \quad (3)$$

In the above equations,  $|H|$  is the total number of

hops between node  $S$  and node  $D$  and  $B_j$  is the blocking time seen by a message on its  $j^{\text{th}}$  hop. Details of the analytical models used in this study are not presented here for the sake of clarity. Nevertheless, interested readers can find the details of the proposed models in (Alzeidi *et al.* 2006 and Alzeidi *et al.* 2007). The equations of the models are listed in Appendix A of this paper.

#### 4. The Cost Performance Model

To make a fair and concrete comparison, the intra-router delay (*ie.* the time to cross the router) must be considered as the complexity of the router might affect the overall performance. This is especially true for wormhole-switched networks as their performance is sensitive to the intra-router delay (Dally and Towles, 2004 and Duato *et al.* 2003).

Two main components contribute to the intra-router delay, the switching delay and routing delay (Chien, 1998). The switching delay is composed of the delay involved in the internal flow control, the delay to cross the crossbar switch and the time to setup the output channel. Hence, according to the studies in (Chien, 1998; Duato and Lopez, 1994 and Miller and Najjar, 1997), the switching delay is given by

$$t_s = \frac{3.4 + 0.6 \log_2(P)}{4.9} \quad (4)$$

In the above equation,  $P$  is the number of ports in the crossbar switch. For the 2D torus,  $P=2V+1$ , while for the  $n$ -dimensional hypercube,  $P=nV+1$ .

Similarly, the routing delay involves address decoding, routing decision and updating the header of the message. According to (Chien 1998 and Duato and Lopez, 1994), the routing delay is given by

$$t_r = \frac{3.4 + 0.6 \log_2(P)}{4.9} \quad (5)$$

In the above equation,  $R$  is the degree of freedom, or the number of alternative channels offered by the routing algorithm to route the message through. When Duato's adaptive routing algorithm is used, a message may be routed to any adaptive virtual channel of the remaining dimensions or to one of the deterministic virtual channels (*ie.* when all adaptive virtual channels are busy). Moreover, a message may also be routed to the local PE. Therefore, the routing delay for the 2D torus and hypercube are, respectively, given by

$$t_r = \frac{4.7 + 1.2 \log_2(2(V-1)+2)}{4.9} \quad (6)$$

$$t_r = \frac{4.7 + 1.2 \log_2((n-1)(V-1)+2)}{4.9} \quad (7)$$

It should be mentioned that the original equations presented in (Chien, 1998; Duato and Lopez, 1994 and

Miller and Najjar, 1997) computes these delays in terms of time units (namely nanoseconds) and are not divided by 4.9 as in the above equations. However, because our models measure different delays in network cycles (instead of nanoseconds), the equations have to be normalised to network cycles (*ie.* the time to transmit one flit across a physical channel) which, for the studies in (Chien, 1998; Duato and Lopez, 1994 and Miller and Najjar, 1997), has been found to be 4.9 nanoseconds.

Several combinations of experiments have been conducted for different network sizes, message sizes, and buffer sizes. However, we present comparison results for several arrangements of virtual channels in 256-nodes 2D torus and hypercube when the total buffer associated with each physical channel is 24, 48 and 96 flits. The switching and routing delays have been calculated using the equations 4 to 7 for all presented results.

## 5. Results and Discussions

Since the buffer size allocated to each physical channel has been kept constant, increasing the number of virtual channels will inevitably result in a decrease in the buffer size allocated to each virtual channel. For instance, when the total buffer size per a physical channel is 24 flits, by having 3 virtual channels per a physical channel, the buffer size of each of these virtual channels will be 8 flits. This means that a 48 flits message occupy the buffers of 6 consecutive channels. But by selecting 8 virtual channels per a physical channel, the buffer space allocated to each of them will be only 3 flits. Consequently, in this case, a 48 flits message requires 16 consecutive channels to be fully accommodated. The advantage of increasing the number of virtual channels is that the physical bandwidth is more optimally utilized (Dally 1992). But decreasing the buffer depth of the virtual channels causes messages to be distributed over a greater number of routers, resulting in higher contention probabilities. Figure 2 illustrates some possible arrangements of a 16-flits buffer into different number of virtual channels. In this section, the performance merits of the 2D torus (*ie.* low dimensional network) and the hypercube (*ie.* high dimensional network) are assessed when different arrangements of the available buffer space per physical channel are used.

### 5.1 Torus Network

In low dimensional networks (in this case the 2D torus), Fig. 3 reveals that increasing the buffer size (and therefore decreasing the number of virtual channels) results in better performance (*ie.* lower latency) for low to moderate traffic rates. This can be observed for almost all cases regardless the total amount of

buffer per a physical channel and/or the message size. Similar results have also been reported in (Dally 1992; Rezazad and Sarbazi-azad, 2005 and Rezazad and Sarbazi-azad, 2005) when deterministic routing is used. In low traffic loads, there are sufficient virtual channels and it is the depth of the virtual channel buffers that have the major impact, as deeper buffers reduces the number of channels occupied by messages and hence reduces the probability of blocking.

However, as the traffic rate increases, increasing the number of virtual channels cause some enhancement in the performance (*ie.* the network saturates at higher traffic generation rates). Nevertheless, from a certain threshold, the increase in the number of virtual channels (and hence the decrease in the buffer size of each virtual channel) exhibits adverse effect on the performance, as can be noticed from Fig. 3. For example, for messages of size 48 flits, increasing the number of virtual channels beyond 8 (when the total buffer space is 24 flits) or beyond 6 (when the total buffer space is 96 flits), substantially decreases the saturation traffic rate. Two factors contributing to this effect are apparent. First, the increase in the switching and the routing delays diminishes the advantages of having greater number of virtual channels. This is because as the number of virtual channels increases, the complexity of the router increases too and hence the routing and the switching delays increases (this can be seen from equations 4 and 5). The second factor is that increasing the number of virtual channels, decreases the buffer sizes, which results in messages occupying larger number of channels and, therefore, higher blocking probabilities.

### 5.2 Hypercube

Figure 4 shows that the hypercube (as an example of high dimensional networks) favours the increase in the number of virtual channels as opposed to the increase in the buffer size per virtual channel especially under moderate and high traffic conditions. This is basically because the low diameter (and hence the smaller average number of hops that messages traverse in the hypercube) offsets the role of moderate size buffers in reducing the number of occupied channels between the source and destination nodes. To further visualize the effect of the buffer size on the performance of the hypercube and torus, we have plotted, in Fig. 5, the saturation traffic rate as a function of the buffer size for 256-nodes systems and 48 flits messages. Two important observations can be deduced from this figure. The figure reveals that the maximum throughput of the hypercube remains almost unchanged as the buffer size increases. Again this can be accounted for the small average number of hops that a message needs to make to cross the hypercube compared to the torus.

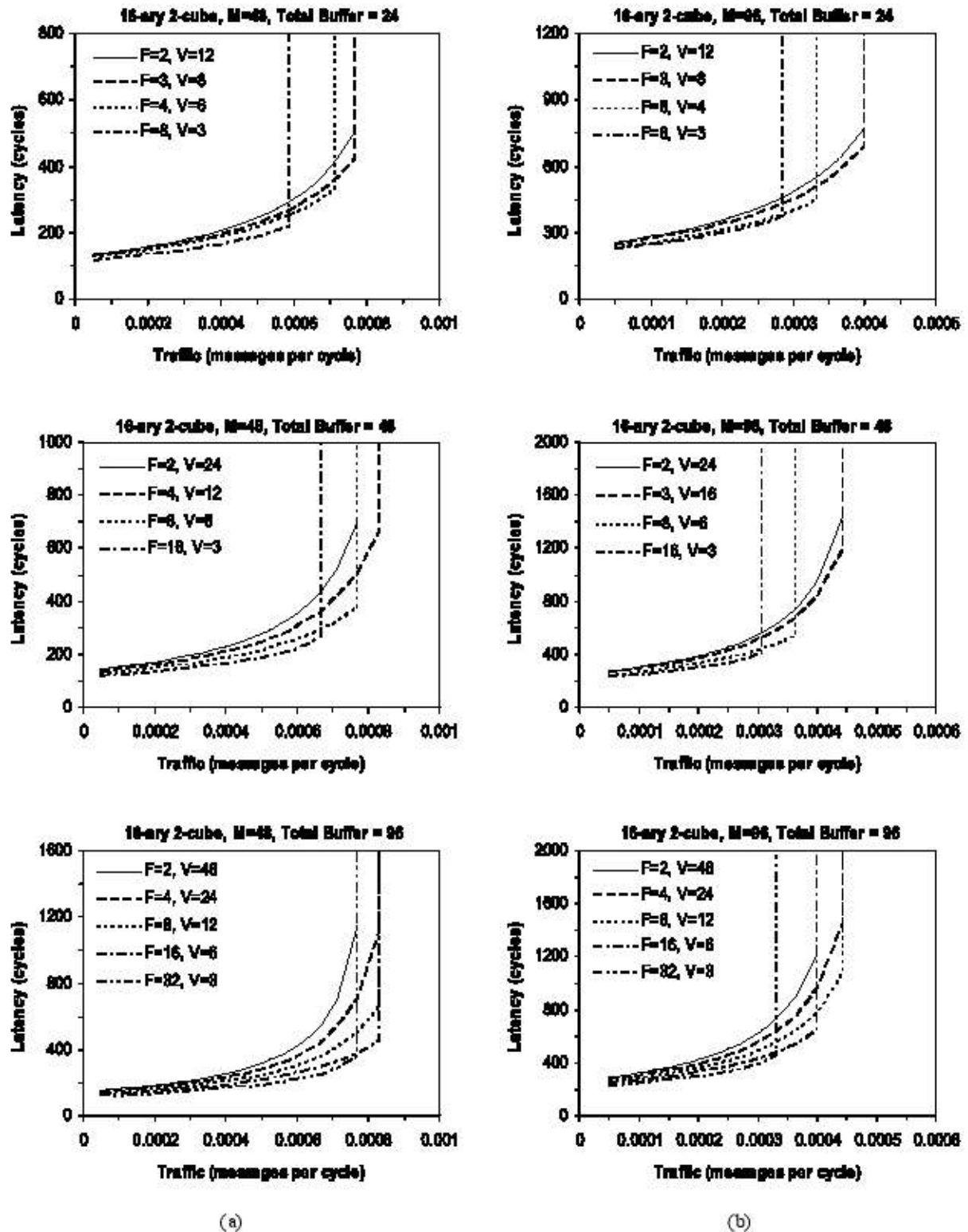


Figure 3. The average message latency in a 16-ary 2-cube 2D torus ( $N=256$ ) for messages of size (a)  $M=96$  flits when different amount (24, 48 and 96 flits) of total buffer space is associated with each physical channel of the network.

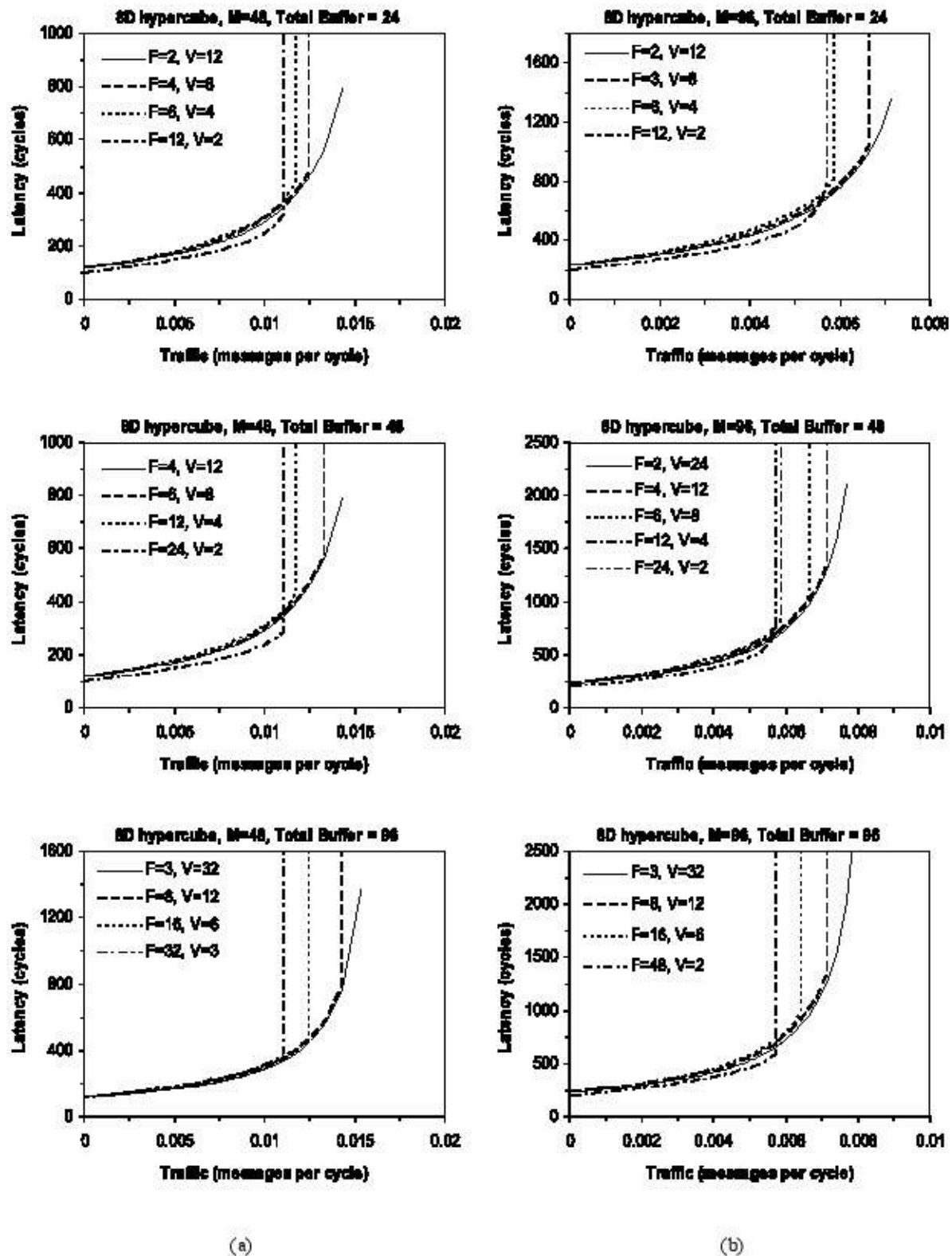


Figure 4. The average message latency in a 8-dimensional hypercube ( $N=256$ ) for messages of size (a)  $M=96$  flits when different amount (24, 48 and 96 flits) of total buffer space is associated with each physical channel of the network.



Moreover, Fig. 5 reveals that increasing the buffer size even to a point where the message can entirely be accommodated in small number of links (for example by using only 2 virtual channels), does not give any substantial advantage in the performance especially under moderate and high traffic conditions. This is because under high traffic conditions, the number of virtual channels (which is minimised here) becomes the important factor in enhancing the performance, as higher number of virtual channels allow for optimal utilization of the available physical bandwidth. However, it should be noticed that even though lower latencies are achieved with larger buffer sizes, this decrease on the average latency is quite marginal as can be observed from the curves in Fig. 4. For instance, the average latency of 48-flit messages in the 3D hypercube with 2 virtual channels and 24 flits of buffer per virtual channel is less than that of the same network with 12 virtual channels and 4 flits of buffers per virtual channel by less than 70 network cycles (20%).

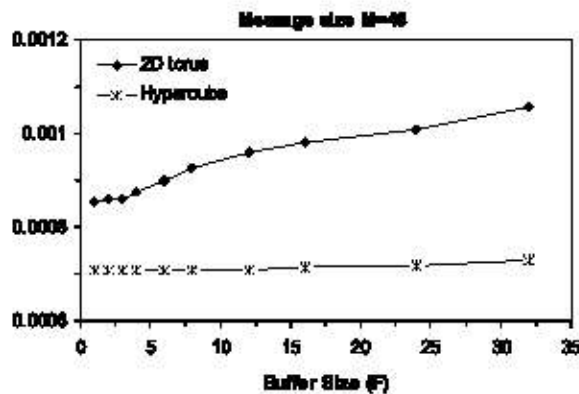


Figure 5. The effects of buffer size on the saturation traffic rate in 256-nodes 2D torus and hypercube with messages of size 48 flits.

## Conclusions

Several researchers (Dally, 1992, Rezazad and Sarbazi-azad, 2005) have studied the optimal arrangement of the available buffer into virtual channels (*ie.* given a fixed amount of finite buffer what is the optimal way to arrange it into virtual channels). However, these studies have so far resorted to simulation experiments and focused on deterministic routing algorithms. In this paper we have used analytical performance models to investigate the optimal arrangement of the available buffer space into multiple virtual channels when adaptive routing is used in wormhole-switched  $k$ -ary  $n$ -cubes. Moreover, we have considered different intra-router delays that affect the performance of the networks under question.

In low dimensional networks our results have

revealed that increasing the buffer size (and hence decreasing the number of virtual channels) results in better performance under low traffic rates. However, as traffic rates increase, we have observed that increasing the number of virtual channels causes some performance enhancement. Nevertheless, our findings agree with previous studies (Dally, 1992, Rezazad and Sarbazi-azad, 2005) that from a certain threshold, the increase in the number of virtual channels causes performance degradation. Higher dimensional networks, however, favors the increase in the number of virtual channels as opposed to the increase in the buffer size. We have also shown that increasing the buffer size in the hypercube have marginal improvement to the network performance.

## References

- Adiga, N.R., Blumrich, M.A., Chen, D., Coteus, P., Gara, A., Giampapa, M.E., Heidelberger, P., Singh, S., Steinmacher-Burow, B.D., Takken, T., Tsao, M. and Vranas, P., 2005, "Blue Gene/L Torus Interconnection Network," IBM Journal of Research and Development, Vol. 49, pp. 265-27.
- Alzeidi, N., Khonsari, A., Ould-Khaoua, M. and Mackenzie, L.M., 2006, "On the Probability Distribution of Busy Virtual Channels," presented at 20th International Parallel and Distributed Processing Symposium, Rhodes Island, Greece, pp. 7.
- Alzeidi, N., Ould-Khaoua, M., Mackenzie, L.M. and Khonsari, A., 2007, "Performance Analysis of Adaptively-Routed Wormhole-Switched Networks with Finite Buffers," presented at IEEE International Conference on Communications (ICC 2007), Glasgow, UK, pp. 38-43.
- Boura, Y., Das, C.R. and Jacob, T.M., 1994, "A Performance Model for Adaptive Routing in Hypercubes," presented at International Workshop on Parallel Processing, pp. 11-16.
- Chien, A.A., 1998, "Cost and Speed Model for  $k$ -ary  $n$ -cube Wormhole Routers," IEEE Transactions on Parallel and Distributed Systems, Vol. 9, pp. 150-162.
- Dally, W.J., 1992, "Virtual Channel Flow Control," IEEE Transactions on Parallel and Distributed Systems," Vol. 3, pp. 194-205.
- Dally, W.J., Dennison, L.R., Harris, D., Kan, K. and Xanthopoulos, T., 1994, "The Reliable Router: a Reliable and High-performance Communication Substrate for Parallel Computers," presented at First International Workshop on Parallel Computer Routing and Communication, Washington, Seattle, USA, pp. 241-255.
- Dally, W.J. and Towles, B., 2004, "Principles and Practices of Interconnection Networks

- Amsterdam," London: Elsevier/Morgan Kaufmann.
- Duato, J., 1993, "A New Theory of Deadlock-Free Adaptive Routing In Wormhole Networks," *IEEE Transactions On Parallel And Distributed Systems*, Vol. 4, pp. 1320-1331.
- Duato, J. and Lopez, P., 1994, "Performance Evaluation of Adaptive Routing Algorithms for k-ary-n-cubes," presented at First International Workshop on Parallel Computer Routing and Communication, Seattle, Washington, USA, pp. 45-59.
- Duato, J., Yalamanchili, S. and Ni, L.M., "Interconnection Networks: An Engineering Approach," San Francisco: Morgan Kaufmann Publishers Inc.
- Duato, J., Yalamanchili, S. and Ni, L.M., 2002, "Interconnection Networks: An Engineering Approach," Los Alamitos: Morgan Kaufmann Publishers Inc.
- Miller, D. and Najjar, W.A., 1997, "Preliminary Evaluation of a Hybrid Deterministic/Adaptive Router," presented at Parallel Computing, Routing and Communication Workshop, Atlanta, Georgia, USA, pp. 21-32.
- Mohapatra, P., 1998, "Wormhole Routing Techniques for Directly Connected Multicomputer Systems," *ACM Computing Surveys*, Vol. 30, pp. 374-410.
- Ould-Khaoua, M., 1999, "A Performance Model for Duato's Fully Adaptive Routing Algorithm in k-ary n-cubes," *IEEE Transactions On Computers*, Vol. 48, pp. 1297-1304.
- Rezazad, M. and Sarbazi-azad, H., 2005, "The Effect of Virtual Channel Organization on the Performance of Interconnection Networks," presented at 19th IEEE International Parallel and Distributed Processing Symposium, Denver, Colorado, USA, pp.
- Rezazad, M. and Sarbazi-Azad, H., 2005, "Performance Evaluation of Fully Adaptive Routing under Different Workloads and Constant Node Buffer Size," presented at International Conference on Parallel and Distributed Systems, Fukuoka, Japan, pp. 510-514.
- Sarbazi-Azad, H., Ould-Khaoua, M. and Mackenzie, L.M., 2001, "Analytical Modeling of Wormhole-Routed k-ary n-cubes in the Presence of Hot-spot Traffic," *IEEE Transactions on Computers*, Vol. 50, pp. 623-634.
- Scott, S.L. and Thorson, G.M., 1996, "The Cray T3E Network: Adaptive Routing in a High Performance 3D Torus," presented at Symposium on High Performance Interconnects, Stanford, CA, USA, pp. 147-156.

### Appendix A: Equations of the Analytical Model

$$\text{Mean Message Latency} = (\bar{S} + \bar{W}_s) \bar{V},$$

$$S_H = |H| + M + \sum_{j=1}^{|H|} B_j,$$

$$\bar{S} = \frac{1}{N-1} \sum_{D \in G \setminus \{S\}} S_H,$$

$$|H| = h_1 + h_2, \quad h_1 = \begin{cases} d_i & d_i \leq \lceil k/2 \rceil \\ k - d_i & \text{otherwise} \end{cases} \quad i = 1, 2,$$

$$B_j = \theta_j \cdot \Upsilon_j \cdot w_j, \quad w_j = \frac{\lambda_c T_j^2 (1 + (T_j - M)^2 / T_j^2)}{2(1 - \lambda_c T_j)}$$

$$\theta_j = \text{Pass}_j^0 \cdot P_e \cdot P_d + \text{Pass}_j^1 \cdot P_d,$$

$$\Upsilon_j = \sum_{x=2}^{F+1} P_x^{M/GI/F+1}$$

$$\theta_j = \text{Pass}_j^0 \cdot P_e \cdot P_d + \text{Pass}_j^1 \cdot P_d, \quad \text{Pass}_j^0 = 1 - \text{Pass}_j^1,$$

$$\text{Pass}_j^1 = \frac{\sum_{l=1}^2 N_0^{H'(l)}(j - h_l)}{N_0^H(j)}, \quad H'(1) = \{0, h_2 - 1\}, \quad H'(2) = \{h_1 - 1, 0\},$$

$$N_0^H(j) = \begin{cases} 1 & j = 0 \\ 0 & j < 0 \text{ or } n_H < 1 \\ \sum_{m=0}^{h_H} N_0^{H - (h_H)}(j - m) & \text{otherwise} \end{cases}$$

$$T_j = A_j + M + \sum_{i=j}^{j+A_j-1} B_i,$$

$$A_j = \begin{cases} \lceil M/F \rceil & |H| - j + 1 \geq \lceil M/F \rceil \\ |H| - j + 1 & \text{otherwise} \end{cases},$$

$$\bar{W}_s = \frac{\lambda \bar{S}^2 / V}{2(1 - \lambda \bar{S} / V)} \left( 1 + (\bar{S} - M)^2 / \bar{S}^2 \right),$$

$$P_v = \begin{cases} (1 - \lambda_c T_j) (\lambda_c T_j)^v & 0 \leq v < V \\ (\lambda_c T_j)^V & v = V \end{cases}, \quad \bar{V} = \left( \sum_{v=1}^V v^2 P_v \right) / \left( \sum_{v=1}^V v P_v \right).$$