*Proceedings of the 13th IASTED International Conference on Parallel and Distributed Computing and Systems, August 2001*

# The *k*-ary *n*-cube Network and its Dual: a Comparative Study

Muhammed Mudawwar and Aya Saad

Computer Science Department
The American University in Cairo

mudawwar@aucegypt.edu
aysaad@aucegypt.edu

## Abstract

In this paper we compare the *k*-ary *n*-cube network with its dual. The dual is obtained by interchanging the role of nodes and channels. The resulting nodes in the dual network are connected to two channels only and the resulting channels are multi-way. One of the major topological advantages that can be observed in the dual network is the fixed node degree that is always equal to two irrespective of the network topology or dimension. This property enables the dual network to have wider channels. This comparative study includes the network properties, the cost, and the performance. The network performance is evaluated through simulation. The simulation results show that the dual network has a lower latency and is more efficient (makes better utilization of channels) than the *k*-ary *n*-cube direct network with same number of nodes and similar packaging costs.

**Keywords:** Interconnection Networks, *k*-ary *m*-way networks, multi-way channels, *k*-ary *n*-cube, network properties.

## 1. Introduction

An interconnection network is a critical part of a parallel computer since application performance is affected by the network latency and throughput. Among the popular networks that are well studied in the literature and used in parallel architectures are the *k*-ary *n*-cube strictly orthogonal wormhole-routed networks [4] [8]. The term *k* refers to the number of nodes per dimension and the term *n* represents the network dimension. Nodes along each dimension can have a linear arrangement resulting in a multi-dimensional mesh topology. Alternatively, they can have a ring arrangement resulting in a torus topology. We can also restrict the number of nodes to two per dimension. This will result in a hypercube topology. Other interesting direct interconnection networks have also been proposed in the literature. These include the hierarchical networks in [12] and the recursive cube of rings in [13].

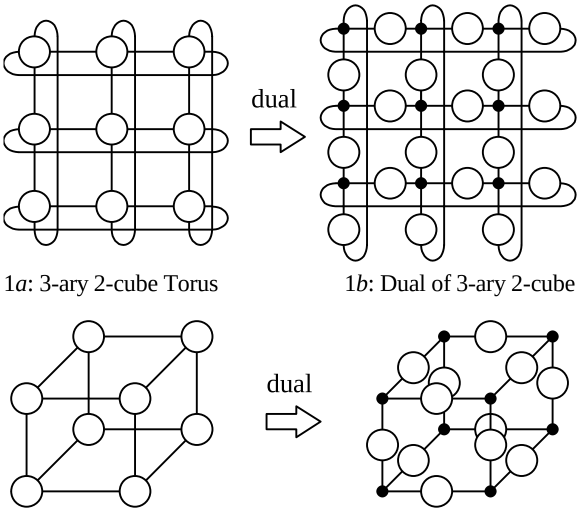
### 1.1 Motivation

The dual of a *k*-ary *n*-cube network, called *k*-ary *m*-way, was proposed in [9] as an alternative interconnection structure that can compete with *k*-ary *n*-cube networks. In this paper, we study the properties and compare the performance of both networks. Routers in the dual network have a degree of two (linked to two multi-way channels only) irrespective of the network dimension or topology. They have a simple design and can be implemented efficiently as suggested in [10].

## 2. Background

### 2.1 The Dual of a *k*-ary *n*-cube Network

The dual of a *k*-ary *n*-cube network, called *k*-ary *m*-way network, is obtained by interchanging the nodes and channels in a network [9]. Nodes become channels and channels become nodes. Figure 1 shows two examples of direct orthogonal networks and their duals. Figure 1*a* is 3-ary 2-cube torus with 9 nodes and 18 bi-directional channels. Figure 1*b* is the dual of the 3-ary 2-cube torus with 18 nodes and 9 channels. Each channel (shown as a black dot) is a 4-way channel because it links 4 nodes together.



1*a*: 3-ary 2-cube Torus

1*b*: Dual of 3-ary 2-cube

Figure 1*c*: 3D cube

1*d*: Dual of 3D cube

Figure 1*c* is a 3D cube with 8 nodes and 12 channels. Figure 1*d* is the dual of the 3D cube with 12 nodes and 8 channels. Each channel is a 3-way channel because it links 3 nodes together. Nodes in the dual networks are linked to *two multi-way channels only*, irrespective of the network topology or dimension.

### 2.2 Multi-way Channels

A bi-directional channel in a direct network can be implemented as two unidirectional channels (full-duplex organization), or as a shared channel (half-duplex organization), as shown in Figures 2*a* and 2*b*. A shared bi-directional channel must be arbitrated to select the driving node at each clock cycle (*arb* line in Figure 2*b*). In the dual network, a multi-way channel, called also *m*-way channel, is used to link several nodes together. It can be implemented as a set of *m* parallel links as shown in Figure 2*c*, or as a shared bus as shown in Figure 2*d*.

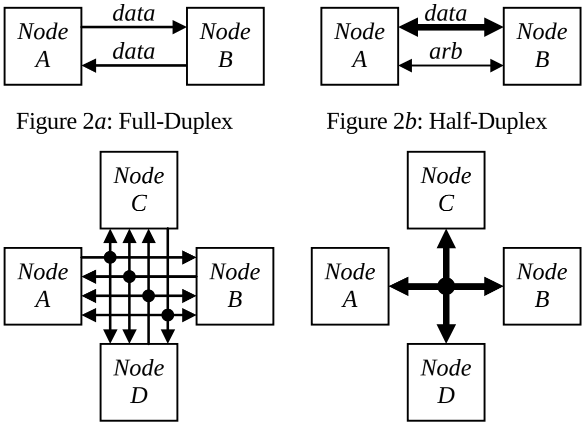


Figure 2*a*: Full-Duplex

Figure 2*b*: Half-Duplex

Figure 2*c*: 4-way Parallel

Figure 2*d*: 4-way Shared

In Figure 2*c*, Four parallel links are used to implement a 4-way channel. Each link connects one node to the remaining three nodes. The links can operate in parallel and four flits can be transmitted simultaneously during one cycle. A shared *m*-way channel, one the other hand, uses a bus to link the four nodes. The shared *m*-way channel can be made wider, but must be arbitrated to select one of the *m* nodes to drive the shared channel at each clock cycle. Multi-way channels (both shared and parallel) facilitate broadcasting and multicasting. A flit can be transmitted to multiple destinations in one cycle.

In the remaining part of this paper, we will restrict our discussion to shared *m*-way channels only because of their wide data path, and because our network simulator is based on them.

### 2.3 Related Work

A shared *m*-way channel is a bus and a *k*-ary *m*-way network with shared channels can be classified as a bus-based network. Many bus interconnection structures were discussed in the literature. They are modeled as *hypergraphs* [1]. A hypergraph, however, does not identify the buffer resources of a network, and hence cannot be used to study deadlocks.

Examples of bus interconnection networks are the hypermesh [14], [6], hypergrid (hypertorus) [5], and hyperbus [2]. In a hypermesh, each node is connected to all the nodes in each dimension through a bus. The hypergrid and hypertorus structures are defined as the Cartesian product of hyperpaths and hyperrings [5]. The node degree is not a constant, but is twice the network dimension. The hyperbus is defined as the dual of a generalized hypercube [2].

In [11], we defined a class of strictly orthogonal scalable network topologies, based on the concept of shared multi-way channels. In [9], we proposed two approaches of constructing *k*-ary *m*-way networks and linking processing nodes with their local memories and/or caches to a *k*-ary *m*-way network. The first approach is to link nodes to multi-way channels. Routers and nodes are two separate entities. The second approach is to integrate routers within nodes. The performance of a *k*-ary *m*-way network, with nodes linked to channels was studied in that paper. In [10], we presented the design of a switch-free router that can be used to build *k*-ary *m*-way networks of various topologies and dimensions.

## 3. Network Properties

In this section we compare the properties of a *k*-ary *n*-cube network with those of its dual.

### 3.1 Notation

In a *k*-ary *n*-cube network:

$n$: number of dimensions
$k$: number of nodes per dimension
$N$: total number of nodes
$C$: total number of bi-directional channels
$w$: width of a bi-directional channel
$W$: total wiring of a *k*-ary *n*-cube network
$b$: bisection width of a *k*-ary *n*-cube network
$B$: bisection density of a *k*-ary *n*-cube network
$D$: diameter of a *k*-ary *n*-cube network

In the dual network:

$n'$: number of dimensions
$k'$: number of multi-way channels per dimension
$N'$: total number of nodes
$C'$: total number of multi-way channels
$w'$: width of a multi-way channel
$W'$: total wiring of the dual network
$b'$: bisection width of the dual network
$B'$: bisection density of the dual network
$D'$: diameter of the dual network

### 3.2 Nodes

In a *k*-ary *n*-cube network, the number of nodes $N = k^n$ for the torus and $N = 2^n$ for the hypercube. In the dual network, the number of nodes $N' = n'\,k'^{\,n'}$ for the torus and $N' = n'\,2^{n'-1}$ for the hypercube. To have the same number of nodes $N = N'$ in the *k*-ary *n*-cube network and its dual:

$k^n = n'\,k'^{\,n'}$ for the torus, and Equation 1

$2^n = n'\,2^{n'-1}$ for the hypercube Equation 2

Solving Equation 1 with $n = n'$ gives:

$$\frac{k}{k'} = \sqrt[n]{n} \qquad \text{Equation 3}$$

Hence, there are more nodes per dimension in a torus than that of its dual with equal dimensionality and number of nodes.

Solving Equation 2 yields:

$$n = n' + \log_2(n'/2) \qquad \text{Equation 4}$$

Thus, the dimensionality of a hypercube is greater than of its dual with an equal number of nodes. For example, there are 1024 nodes in a 10-D hypercube, as well as in the dual hypercube with 8 dimensions.

### 3.3 Node Degree

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

In the dual network, we count the number of multi-way channels linked to a node. This number is 2, irrespective of the network dimensionality or topology. Thus, multi-way channels can be much wider than bi-directional channels.

### 3.4 Channels

In a *k*-ary *n*-cube network, the total number of bi-directional channels is $C = n\ k^n$ for the torus and $C = n\ 2^{n-1}$ for the hypercube. In the dual network, the number of multi-way channels is $C' = k'^{\,n'}$ for the torus and $C' = 2^{n'}$ for the hypercube.

If the dimensionalities of a torus and its dual are equal, $n = n'$, and for an equal number of nodes:

$$\frac{C}{C'} = \frac{nk^n}{k'^n} = n\left(\frac{k}{k'}\right)^n = n^2 \qquad \text{Equation 5}$$

For a hypercube:

$$\frac{C}{C'} = \frac{n2^{n-1}}{2^{n'}} = \frac{n \times n'}{4} = \frac{n'(n' + \log_2(n'/2))}{4} \qquad \text{Eq. 6}$$

Therefore, the number of bi-directional channels in the *k*-ary *n*-cube network exceeds the number of multi-way channels in the dual network by a factor proportional to the square of the dimension.

### 3.5 Channel Width

The channel width is defined as the number of physical wires per channel. Although a channel consists of data and control lines, we will ignore the control lines and 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 and its dual. In the *k*-ary *n*-cube network, the channel width, *w*, is the number of data lines in a physical bi-directional channel. In the dual network, the channel width, *w'*, is the number of data lines in a physical multi-way channel. Our assumption here is that Multi-way channels are shared, although they can be implemented differently as indicated in Section 2.2.

To have an equal wiring per node:

$$w' = n \times w \text{ for the torus} \qquad \text{Equation 7}$$

$$w' = n \times w / 2 \text{ for the hypercube} \qquad \text{Equation 8}$$

Clearly, the width of a multi-way channel in the dual network greatly exceeds the width of a bi-directional channel, especially in high-dimensional networks.

### 3.6 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 a network. It is the product of the total number of physical channels and the channel width.

For a torus network and its dual with an equal number of nodes and same dimensionality:

$$\frac{W}{W'} = \frac{C \times w}{C' \times w'} = \frac{n^2}{n} = n \qquad \text{Equation 9}$$

For a hypercube network and its dual with an equal number of nodes:

$$\frac{W}{W'} = \frac{C \times w}{C' \times w'} = \frac{n \times n'}{4 \times (n/2)} = \frac{n'}{2} \qquad \text{Equation 10}$$

Therefore, the total wiring of a *k*-ary *n*-cube network greatly exceeds the total wiring of its dual by a factor proportional to the number of dimensions.

### 3.7 Bisection Width

The bisection width is defined as the number of channels that must be crossed in order to cut the network into two equal sub-networks. The bisection width of a *k*-ary *n*-cube torus is $b = 2k^{n-1}$. The factor 2 is due to the ring arrangement in all dimensions. The bisection width of a hypercube is $b = 2^{n-1}$. In the dual network, the bisection width is $b' = 2k'^{\,n'-1}$ for a torus, and $b' = 2^{n'-1}$ for a hypercube.

For a torus and its dual network with an equal number of nodes and same dimensionality:

$$\frac{b}{b'} = \frac{k^{n-1}}{k'^{n-1}} = \frac{k^n}{k'^n} \times \frac{k'}{k} = \frac{n}{\sqrt[n]{n}} \qquad \text{Equation 11}$$

For a hypercube network and its dual with an equal number of nodes:

$$\frac{b}{b'} = \frac{2^{n-1}}{2^{n'-1}} = \frac{n'}{2} \qquad \text{Equation 12}$$

Therefore, the bisection width of a *k*-ary *n*-cube network is larger than that of its dual with an equal number of nodes.

### 3.8 Bisection Density

The bisection density of a network is defined as the number of wires that must be crossed in order to cut the network into two equal sub-networks. It is the product of the bisection width and the channel width.

For a torus and its dual network with an equal number of nodes, same dimensionality, and equal wiring per node:

$$\frac{B}{B'} = \frac{b \times w}{b' \times w'} = \frac{n}{\sqrt[n]{n}} \times \frac{1}{n} = \frac{1}{\sqrt[n]{n}} \qquad \text{Equation 13}$$

For a hypercube network and its dual with an equal number of nodes, and equal wiring per node:

$$\frac{B}{B'} = \frac{b \times w}{b' \times w'} = \frac{n'}{n} = \frac{n'}{n' + \log_2 (n'/2)} \qquad \text{Equation 14}$$

Therefore, the bisection density of a *k*-ary *n*-cube network is slightly less than that of its dual with an equal number of nodes and equal wiring per node.

### 3.9 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-ary n-cube network, the diameter $D = nk/2$ for a torus, and $D = n$ for a hypercube. In the dual network the diameter $D' = n'\ k'\ /\ 2$ for a torus, and $D' = n'$ for a hypercube.

If the dimensionalities of a torus and its dual are equal, $n = n'$, and for an equal number of nodes:

$$\frac{D}{D'} = \frac{k}{k'} = \sqrt[n]{n} \qquad \text{Equation 15}$$

For a hypercube:

$$\frac{D}{D'} = \frac{n}{n'} = \frac{n' + \log_2 (n'/2)}{n'} \qquad \text{Equation 16}$$

Hence, the diameter of a k-ary n-cube network is larger than that of its dual with an equal number of nodes.

## 4. Network Simulation and Performance

A time-driven simulator has been implemented to compare the performance of a *k*-ary *n*-cube and its dual network. In the implementation, we assume that a bi-directional channel is shared and arbitrated as shown in Figure 2*b*. Similarly, we assume that a multi-way channel is shared and arbitrated in the dual network as shown in Figure 2*d*.

### 4.1 The Simulator

The simulator is a C++ program that simulates *k*-ary *n*-cube networks and their duals at the flit level. A flit transfer between two adjacent nodes is assumed to take place in one clock cycle. The network is simulated synchronously, moving all flits that have been granted channels in one clock cycle and then advancing time to the next cycle. The simulator can be configured to support different network sizes, topologies, dimensionalities, number of buffers (virtual channels), buffer sizes, routing algorithms, messages lengths, message generation rates, and traffic patterns. The simulator can generate various statistics, such as average message latency, maximum latency, latency standard deviation, latency histogram, channel utilization rate, node injection rate, and node ejection rate.

### 4.2 Routing Algorithms

The comparison is conducted under unicast-based wormhole routing algorithms. Two deterministic and two minimal adaptive routing algorithms have been employed in the torus and in the mesh (hypercube) topologies. To avoid deadlocks, we divided buffers (or virtual channels) into classes. For dimension-order routing in a mesh or in a hypercube, one buffer class is sufficient and any buffer can be allocated without any restrictions. For dimension-order routing in a torus, 2 buffer classes are required, irrespective of the number of dimensions, to avoid deadlocks in the rings along each dimension. This algorithm is presented in [9] and makes efficient use of buffers. For minimal adaptive routing in a mesh or a hypercube, 2 buffer classes are also required, irrespective of the number of dimensions. We can also make efficient use of buffers by allocating any buffer when we route along the least dimension, and restrict routing to one class of buffers (the adaptive class) when we don't route along the least dimension. The selection of the next buffer class does NOT depend on the current buffer class, but rather on the next selected routing dimension when more than one exists. Finally, for minimal adaptive routing in a torus, four buffer classes are required, irrespective of the number of dimensions. We note that all the routing algorithms have been proven to be deadlock and livelock free.

### 4.3 Setting up the Simulation Parameters

In our comparison study, six networks are utilized. The first one is a 32 by 16, 2D torus with 512 nodes. The second one is the dual of a 16 by 16 torus with 256 4-way channels and 512 nodes. The width of a multi-way channel in the dual network is chosen to be 64 bits. To have an equal wiring cost per node in both networks, the width of a bi-directional channel in the 2D torus network is 32 bits, according to Equation 7.

The third network is a 4D torus with 1024 nodes (8×8×4×4 nodes). The fourth one is the dual of a 4×4×4×4 torus with 256 8-way channels and 1024 nodes as well. The width of a multi-way channel in the dual network is chosen again to have 64 bits of data, but the width of a bi-directional channel is 16 bits in the 4D torus to have an equal wiring cost per node according to Equation 7.

The fifth network is a 10D hypercube with 1024 nodes. The sixth network is the dual of an 8D hypercube with 1024 nodes as well. The width of a multi-way channel is chosen again to have 64 bits of data, but the width of a bi-directional channel in the 10D hypercube is almost 13 bits, according to Equation 8.

All of the experiments are conducted under uniform traffic distribution and using four buffers (virtual channels) per routing direction. Short messages, each carrying 64 bytes of data, are generated.

### 4.4 Simulation Results

The simulation results are shown in Figures 3 through 6. The *latency* is measured from the time a message is generated at a source node until the tail flit is ejected at a destination node. It is measured in terms of simulated cycles, rather than real time, to make it independent of the clock or channel speed. Source queuing time is included in the latency measurement. The *ejection rate* of a node is a measure of normalized throughput and is the number of bits that are ejected per node per clock cycle. *Channel utilization* is a measure of network efficiency and is the percentage of cycles a channel is used to transfer flits.

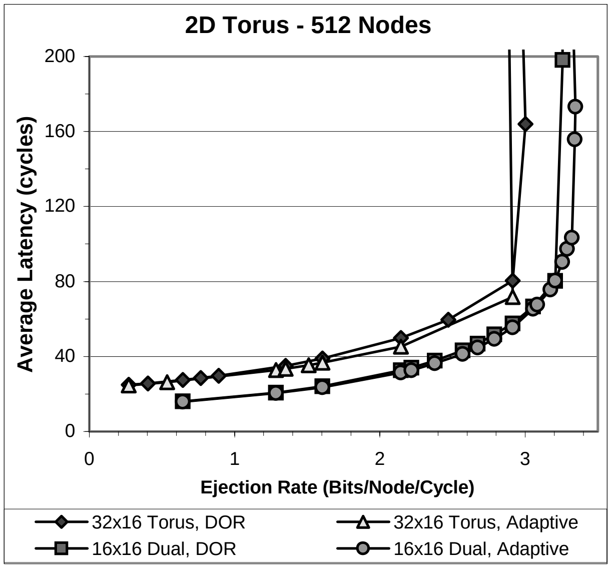


Figure 3: Performance comparison of a 2D torus and dual network

Figure 3 is a performance comparison between a 2D torus and a 2D torus dual, both with 512 nodes. Dimension order routing (DOR) and adaptive routing are used in both networks. The obtained graphs show that the dual network performs better than the direct network. The dual network has a lower average latency and saturates at a higher ejection rate (better throughput). At saturation point, the latency increases sharply in both networks, because messages end up waiting a long period of time in the source queues before being injected into the network. Figure 3 also reveals that adaptive routing is only slightly better than deterministic routing in both networks. The use of adaptive routing is not as significant as changing the topology.

Figure 4 shows a performance comparison between a 4D torus and a 4D torus dual, both with 1024 nodes. Dimension order routing (DOR) and adaptive routing are used in both networks. Figure 4 reveals that the average latency of the dual network is less than that of the direct network before saturation point. However, the direct 4D torus network saturates at a higher ejection rate and has a higher throughput. This is because the total bandwidth of the 4D torus is 4 times the total bandwidth (total wiring) of the dual network according to Equation 9. The ejection rate is 41% higher in the 4D torus than in the dual network, although the total bandwidth is 400% (4×) more.

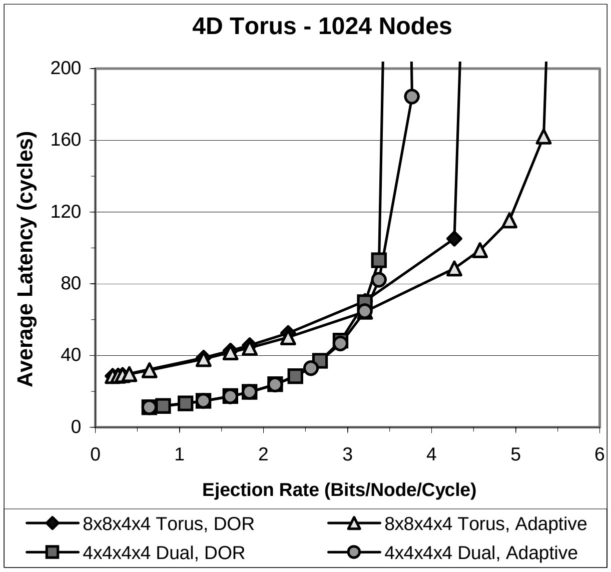


Figure 4: Performance comparison of a 4D torus and dual network

In addition to average latency, we compute the latency standard deviation as a measure of variation in latency. We also compute the channel utilization as a measure of efficiency. This is shown in Figure 5. The latency standard deviation increases slowly with channel utilization below saturation, but increases sharply at saturation point. Figure 5 clearly indicates that the dual network is much more efficient than the direct 4D torus. Channel utilization exceeds 95% in the dual 4D torus, and barely reaches 50% in the direct network.

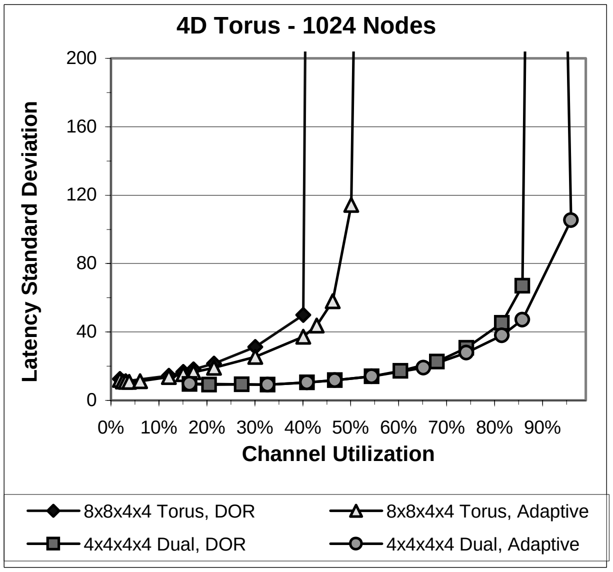


Figure 5: Efficiency of a 4D torus and dual network

Figure 6 shows a performance comparison between a 10D hypercube and an 8D dual network, both with 1024 nodes. As in Figure 4, the average latency in the dual network is much less than that of the direct network before saturation point. However, the 10D hypercube network saturates at a higher ejection rate and has a higher throughput. The ejection rate is 29% higher in the 10D hypercube than in the dual network, although the total bandwidth is 400% (4×) more in the 10D hypercube, according to Equation 10.

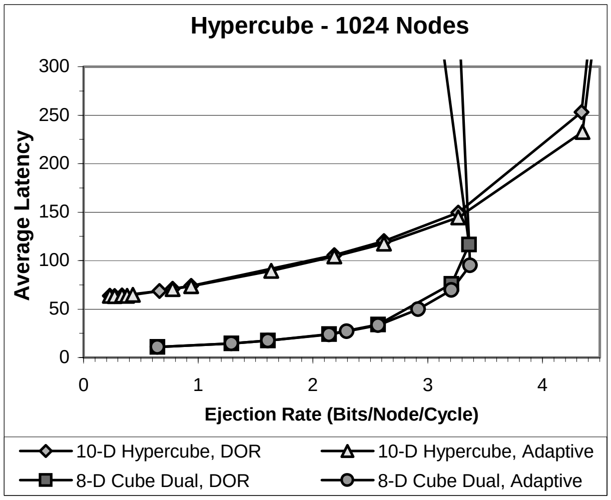


Figure 6: Performance of a 10D hypercube and dual network

## 5. Conclusion and Future Work

We have shown that *k*-ary *m*-way networks, which are the dual of *k*-ary *n*-cube networks, have a lower latency and a higher efficiency (channel utilization) than *k*-ary *n*-cube networks of same number of nodes and similar cost. Although the total network bandwidth and wiring is higher in the *k*-ary *n*-cube network and grows linearly with the dimensionality according to Equations 9 and 10, the ejection rate (and throughput) of the 2D torus network was shown to be worse than that of the dual network, and becomes only slightly better for higher-dimensional networks as shown in Figures 4 and 6.

The results that we have obtained encourage further study of *k*-ary *m*-way networks and their applicability in the design of future parallel computers. As for future work, we will investigate broadcasting, multicasting, and routing in the presence of faults in *k*-ary *m*-way networks, and compare it with *k*-ary *n*-cube networks. We believe that *k*-ary *m*-way networks are especially useful for broadcasting and multicasting because of the nature of a multi-way channel.

## References

[1] J.-C. Bermond and F. Ergincan, Bus Interconnection Networks, *Discrete Applied Mathematics*, 68, pages 1-15, 1996.

[2] L. Bhuyan and D. Agrawal, "Generalized Hypercube and Hyperbus Structures for a Computer Network," *IEEE Transactions on Computers, v. c-33, no. 4, pp. 323-333,* April 1984.

[3] A. Chien, "A Cost and Speed Model for k-ary n-Cube Wormhole Routers," *IEEE Transactions on Parallel and Distributed Systems, v. 9, no. 2*, February 1998.

[4] J. Duato, S. Yalamanchili, and L. Ni, "Interconnection Networks: An Engineering Approach", *IEEE Computer Society Press*, 1997.

[5] A. Ferreira, A. G. vel Lejbman, and S.W. Song, Broadcasting in bus interconnection networks, *CONPAR 94*, Lecture Notes in Computer Science, Springer-Verlag, Sept. 1994.

[6] S. Kim and K. Chwa, "Optimal Embeddings of Multiple Graphs into a Hypermesh," *International Conference on Parallel and Distributed Systems*, 1997.

[7] S. Latifi and P. Srimani, "A New Fixed Degree Regular Network for Parallel Processing," *Proceedings of Eighth IEEE Symposium on Parallel and Distributed Processing, New orleans,* October 1996.

[8] P. Mohapatra, "Wormhole Routing Techniques for Directly Connected Multicomputer Systems," *ACM Computing Surveys, v. 34 no. 3,* September 1998.

[9] M. Mudawwar, "K-ary M-way Networks: the Dual of K-ary N-cubes," *in Proceedings of the 12th IASTED International Conference on Parallel and Distributed Computing and Systems,* November 2000.

[10] M. Mudawwar, "A Switch-Free Router for k-ary m-way Networks," *in Proceedings of the 2000 International Conference on Parallel and Distributed Processing Techniques and Applications, pp. 977-983,* June 2000.

[11] M. Mudawwar, "Multiway Channels in Interconnection Networks," *Proceedings of the ISCA 12th International Conference on Parallel and Distributed Computing Systems, pp. 506-513,* August 1999.

[12] H. Park and D. Agrawal, "WICI: An Efficient Hybrid Routing Scheme for Scalable and Hierarchical Networks," *IEEE Transactions on Computers, v. 45, no. 11,* November 1996.

[13] Y. Sun, P. Cheung and X. Lin, "Recursive Cube of Rings: A New Topology for Interconnection Networks," *IEEE Transactions on Parallel and Distributed Systems, v. 11, no. 3, pp. 275-286,* March 2000.

[14] T. Szymanski, Hypermeshes: Optical interconnection networks for parallel processing, *Journal of Parallel and Distributed Computing*, vol. 26, pages 1-23, January 1995.