Total-Exchange on Wormhole k-ary n-cubes with Adaptive Routing

Fabrizio Petrini[*]
Oxford University Computing Laboratory
Wolfson Building, Parks Road
Oxford OX1 3QD, England
e-mail: fabp@comlab.ox.ac.uk

Abstract

The total-exchange is one of the most dense communication patterns and is at the heart of numerous applications and programming models in parallel computing. In this paper we present a simple randomized algorithm to efficiently schedule the total-exchange on the family of k-ary n-cubes with adaptive routing and wormhole switching. This algorithm is based on an important property of the wormhole networks that reach high throughput under uniform traffic.

The experimental results, conducted on a 256 nodes bi-dimensional cube using an adaptive routing algorithm based on the Duato's methodology, show that the proposed exchange algorithm reaches a very high throughput with small and medium-sized packets, around 85% of the optimal bound, and is more efficient than other algorithms presented in the literature.

1 Introduction

Communication between nodes in a parallel machine can generally be described as x-to-y communication where x and y can be substituted by *one*, *all* and *many*. Communication implies nodes sending and receiving messages: x being *one*, *all* and *many* respectively, indicates that only one of the nodes, that all nodes, and that only some nodes send data. Similarly, y being *one*, *all* and *many* indicates that from each of the senders one, all and many nodes receive data respectively. Communication can be further distinguished as a *broadcast*/*accumulation* or as a *personalized* communication. For example, one-to-all communication could be either a one-to-all broadcast (*single-node broadcast*) where a single node sends out the same message to all nodes, or a one-to-all personalized communication (*single-node scatter*) where a single node sends out different messages to each node. Communication with multiple senders and multiple receivers is also referred as *collective* communication. The nature of the messages to be sent can be also classified as *personalized* or *non-personalized*. The all-to-all personalized communication, or simply *total-exchange*, is an important communication pattern that is at the heart of many applications, such as matrix transposition and the fast Fourier transform (FFT), and programming models as the BSP [8]. The *total-exchange* is a collective communication pattern where every node has to send a distinct message to any other node. The efficient implementation of the total-exchange has been extensively studied in the past few years [9, 10, 11, 12].

[*]Part of this work was done while the author was at the International Computer Science Institute, Berkeley, CA

Wormhole switching has been adopted by many new-generation parallel computers, such as the Intel Touchstone Delta, Intel Paragon, MIT J-Machine, Stanford Flash and the Cray T3D and T3E. In such networks, a packet is partitioned in a sequence of elementary units called *flits*, which are sent in a worm-like or pipelined manner. Network throughput of wormhole networks can be increased by organizing the flit buffers associated with each physical channel into several virtual channels. These virtual channels are allocated independently to different packets and compete with each other for the physical bandwidth. This decoupling allows active messages to pass blocked messages using network bandwidth that would otherwise be wasted. *Adaptive routing* is very important to provide a network performance which is less sensitive to the communication pattern. In this case, the paths can be chosen according to the degree of congestion in the network.

In this paper we present a simple and innovative randomized direct algorithm to schedule the total-exchange in a wormhole-routed cube with adaptive routing. This algorithm (1) is able to exploit the increased throughput provided by the adaptive routers under uniform traffic, (2) works for any k-ary n-cube, irrespective the values of k and n, (3) does not require explicit synchronization between the processors, (4) is very simple to describe and implement and (5) is bandwidth-efficient.

On the other hand, most of the literature that deals with the total-exchange tends to strictly divide the transmission in phases that proceed in lockstep. They also assume the presence of deterministic routing.

Our work is motivated by the fact that wormhole networks can reach a high throughput under uniform random traffic if packets are routed adaptively and if the packet size is properly chosen [2]. For this reason we adopt a randomized strategy that reproduces a uniform random traffic inside the network. The experimental results, conducted on a bi-dimensional cube with 256 nodes, using a detailed simulation model, show that it is possible to achieve a network throughput that is very close to optimality. Also, we provide an in-depth analysis of the network behavior during the execution of the total-exchange algorithms.

The rest of this paper is organized as follows. Section 2 describes the minimal adaptive routing algorithm for the class of k-ary n-cubes. Section 3 reviews some total-exchange algorithms presented in the literature and Section 4 motivates and describes our randomized algorithm. The experimental results are shown in Section 5. Finally, some concluding remarks are in given in Section 6.

2 Routing on the k-ary n-cubes

A k-ary n-cube is characterized by its dimension n and radix k, and has a total of k^n nodes. The k^n nodes are organized in an n-dimensional mesh, with k nodes in each dimension and wrap-around connections at the borders. The binary hypercube is a special case of k-ary n-cube with $k = 2$. Also, the two dimensional torus is another special case with $n = 2$.

Routing algorithms on the k-ary n-cubes are deadlock-prone and require sophisticated strategies to avoid or to recover from deadlocks. In this paper we consider an adaptive algorithm based on the Duato's methodology [1]. This methodology only requires the absence of cyclic dependencies on a connected channel subset. The remaining channels can be used in almost any way. We associate four virtual channels to each link: on two of these channels, called *adaptive* channels, packets are routed along any minimal path between source and destination. The remaining two channels are *escape* channels where packets are routed deterministically when the adaptive choice is limited by network contention. A similar algorithm has been recently adopted by the Cray T3E [7].

A central point of this adaptive algorithm is the interface between the processor and the router. We assume that packets can enter the network using only a subset of the adaptive channels [5]. This limitation, known as *source throttling*, makes the network throughput stable when the network operates above saturation, which is the typical case when we execute a total-exchange algorithm.

When the adaptive routing function can choose between two distinct directions the algorithm picks one of them randomly. In our simulation model, we equalize the routing delay (the time needed to open a path for an incoming packet) the link delay (the flit transmission latency over a physical link) and the switch delay (the router traversal time of a generic non-header flit) to a single clock cycle.

3 Total-exchange algorithms

The algorithms that can be used to implement the total-exchange on a given network can be roughly classified into two classes: (1) *direct* algorithms, in which data are sent directly from source to destination and (2) *indirect* algorithms, in which data are sent from source to destination through one or more intermediate nodes.

The optimal direct algorithm for a hypercube architecture is the *pairwise* exchange algorithm as it guarantees no link contention at every step. This algorithm has also been shown to perform well on the fat tree architecture of the CM-5 [3]. It requires $N - 1$ steps, where N is the number of nodes, and the communication schedule is as follows. In step i, $1 \leq i \leq N-1$, each node exchanges data with the node determined by taking the exclusive-or of its number with i. Therefore, this algorithm has the property that the entire communication pattern is decomposed into a sequence of pairwise exchanges. Unfortunately this algorithm generates conflicts on the bi-dimensional cubes.

Recently Tseng and Gupta introduced an algorithm for wormhole-routed k-ary n-cubes that divides the transmission in a number of congestion-free phases [11]. This algorithm uses an optimal number of phases if the arity of the cube is an integer multiple of eight or an asymptotically optimal number otherwise. It assumes that phases can be strictly routed in lockstep, a condition that cannot be easily achieved in a distributed environment. Also, the messages must be long enough to hide the initial delay of the potentially asymmetric message transmission in each phase.

In the indirect algorithms a message is sent from the source node to the destination through one or more intermediate nodes. These algorithms combine the messages of several phases in a single packet, so they can reduce the startup overhead in those machines that are injection-limited rather than bandwidth-limited.

The *indirect pairwise* exchange algorithm [10] aims at reducing the the link contention of the pairwise exchange algorithm on the bi-dimensional cubes. In this algorithm each node communicates only with the nodes in its row and column. Each exchange along a row is followed by a complete exchange along a column. The *diagonal-propagation* [12] approach includes a set of total-exchange algorithms for the bi- and three-dimensional cubes whose arity is a multiple of four. This approach develops a communication schedule consisting of several congestion-free phases in which the processors are grouped along the diagonals. On a bi-dimensional cube, the diagonal-propagation algorithm is two times bandwidth-optimal, that is it takes twice the time determined by the lower bound of the bisection bandwidth, but half the time determined by the startup overhead of a direct algorithm. A further reduction (to a small multiple of the logarithm of the cube arity k) of the startup overhead is provided by two algorithms introduced by Suh and Yalamanchili [9].

In the experimental evaluation we will focus our attention on the direct total-exchange algorithms. An in-depth discussion of indirect algorithms is outside the scope of this paper. Although the vast majority of existing machines are dominated by the message startup (and so, indirect algorithms can be potentially more efficient), there is a collaborative effort in the scientific community to reduce this overhead to negligible values [4].

4 A randomized total-exchange algorithm

The total-exchange in the k-ary n-cubes, as the uniform random traffic, is limited by the network bisection, when the startup overhead is not the limiting factor. The network capacity can be determined by considering that 50% of the traffic crosses the bisection of the network. Thus if a network has bisection bandwidth B, each of the N nodes can inject $2B/N$ traffic at the maximum load.

Our algorithm is based on an important property of the wormhole routed networks. In Figure 1 we can see the saturation points, under uniform traffic, of a 256 nodes bi-dimensional cube, using the adaptive algorithm. The flit size is four bytes.

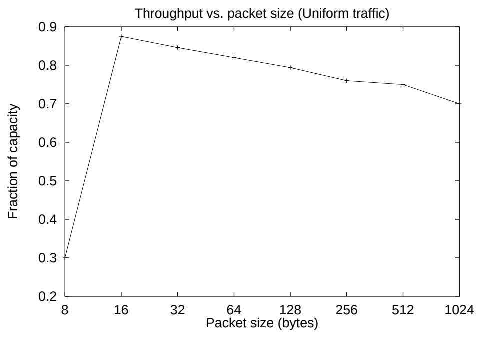

Figure 1. Saturation points under uniform traffic.

We can see in Figure 1 that the network throughput is sensitive to the packet size with a preference for short packets. The network reaches around 80% of the capacity for packets ranging from 16 to 256 bytes, with a peak throughput of 89% of the capacity when the packet size is 16 bytes.

Short packets reduce the network contention and increase the overall throughput, but when the packets are too short (e.g. 8 bytes = 2 flits) the routing overhead dominates and there can be bubbles in the internal pipelines of the router. When the packets are long (> 256 bytes) there is less parallelism in the utilization of the virtual channels, with a consequent performance disadvantage.

While the high saturation throughput provided by the adaptive algorithm is well known in the wormhole routing community, it has not still been exploited to schedule collective communication patterns. As already noted, a viable solution to implement the total-exchange is to synthetically reproduce inside the network a uniform random traffic, by properly choosing the packet size.

Starting from this observation, we propose a simple *randomized* algorithm to implement the total-exchange. In outline, given m the grain size of the total-exchange (i.e. the amount of information exchanged between any pair of nodes) and p the packet size, both expressed in bytes, we can logically schedule the transmission in $\lceil \frac{m}{p} \rceil$ steps. In each step i, $i \in \{0, \ldots, \lceil \frac{m}{p} \rceil - 1\}$:

1. each node j generates an independent permutation $\Pi_{i,j}$ of the remaining $N - 1$ nodes
2. and send the packets following the order suggested by the permutation.

Even if we have a sequence of steps, strict synchronization between processing nodes is not required, i. e. the processing nodes can proceed autonomously after the beginning of the communication pattern. This algorithm can be considered as complementary to the other algorithms in the literature, because it replaces a deterministic scheduling with a random scheduling.

5 Experimental results

We have implemented two total-exchange algorithms in the SMART simulation environment [6]. They are the algorithm by Tseng and Gupta [11] that uses an optimal number of phases when the cube arity is an integer multiple of eight and our randomized algorithm. In the Tseng and Gupta algorithm all the processors start the exchange algorithm in the same instant (a condition that is not easy to meet in a distributed environment) and each phase is separated by $m/4 + 2 * k$ cycles, a delay that includes the conflict-free average propagation delay of $2 * k$ cycles and the time needed to absorb a packet at the destination, $m/4$ cycles, considering the message size m and the data path width of 4 bytes. There is no explicit barrier synchronization between the processors, and the phases are logically separated by an interval of $m/4 + 2 * k$ cycles implemented with local timeouts.

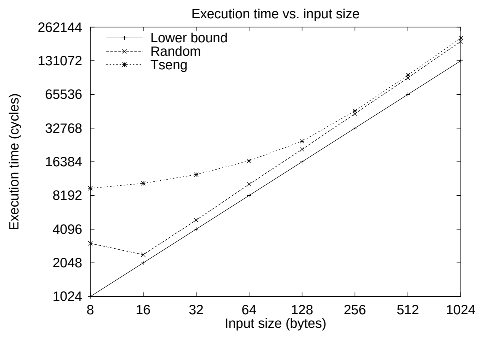

Figure 2. The execution times of the total-exchange algorithms. Both axes use a logarithmic scale.

Figure 2 shows the performance of these algorithms on a 256 nodes bi-dimensional cube. The graph shows the execution time of the total-exchange algorithms on the vertical axis and the grain size of the total-exchange on the horizontal axis. The

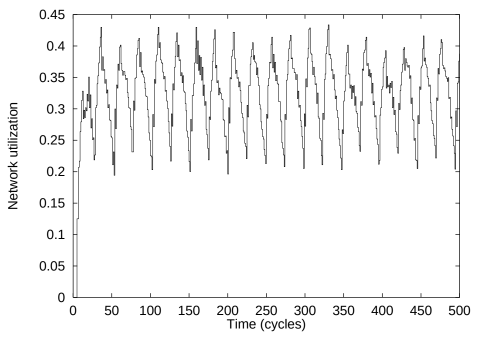

Figure 3. Network utilization during the execution of the Tseng and Gupta algorithm. The grain size of the total-exchange is 32 bytes.

graph in Figure 2 also reports a lower bound, that is computed by considering the usual topological limitation of the bisection bandwidth. In fact, half of the packets must cross the network bisection.

The Tseng algorithm is very inefficient when the grain size of the total exchange is small. For example when $m = 32$ bytes, the execution time is three times the asymptotic lower bound. As expected, this algorithm gives better results with larger packets.

We provide more insight on these results by observing the network utilization, i. e. the fraction of active links, during the execution of the total-exchange algorithms. The cube has the property that with the total-exchange (and uniform traffic in general) all the links are active when the network operates at capacity. In fact, all the $2k^{n-1}$ links of the bisection are active on both directions. Assuming that messages are long enough, each message occupies on the average a number of links equal to the average distance d_m. Thus the number of active links (in the single direction) $Links$ is

$$Links = 8d_m k^{n-1} = 2nk^n \qquad (1)$$

that is exactly the number of links in the networks, because each node has $2n$ outgoing unidirectional links. From this result follows that network utilization and network throughput are strongly related.

In Figure 3 we can inspect the network behavior of the Tseng algorithm when the packet size is 32 bytes. The network utilization is very low, around 30% and this explains the poor performance results.

Figure 4 describes the network behavior when the packet size is 1024 bytes. We can see that the adaptive algorithm breaks the congestion-free nature of the communication patterns designed for a deterministic routing algorithm. Getting back to Figure 2, we can see that the randomized algorithm obtains excellent performance with small- and medium-sized packets.

The near-optimal performance of the randomized algorithm can be explained looking at Figure 5. This algorithm exploits some performance characteristics of the adaptive routing algorithms under uniform traffic. When the grain size is 32 bytes, (1) the network reaches the steady state of about 85% active links after very few cycles (just 25 cycles in the example); (2) once in the steady state, the network utilization and throughput are stable with small fluctuations (less than 2%): a high percentage of links are used in a profitable way; (3) at the end of the steady state packets are consumed by the network in a short period (400 cycles).

When the grain size is larger that 32 bytes, we can organize the exchange algorithm in a sequence of sweeps, each using the smaller grain size and another independent permutation. We pay the inefficiency of the initial and final periods just once, achieving 85% of the theoretical throughput. We did not use this solution in the experimental evaluation of Figure 2, so the randomized algorithm can obtain an even better performance with larger packets.

In Figure 6 we study, for a fixed packet size of 32 bytes, how the saturation throughput is influenced by the network arity k and the network dimension n. We can see that the saturation throughput is insensitive to the network arity (the small fluctuations are within the confidence interval, which is 1%), and slightly sensitive to the network dimension. The difference between the uni-dimensional and the bi-dimensional cube is less than 1% and the difference between the bi-dimensional and the three-dimensional cube is only 2%. This means that we can efficiently apply our algorithm to the entire family of k-ary n-cubes, for any value of k and n.

The execution time can be easily estimated by dividing the total amount of information for the steady state bandwidth. The optimal packet size is a peculiar characteristic of the routing algorithm in use: we have seen that packets between 16 and 64 bytes are good choices.

These results suggest a methodology to implement all the algorithms where each node sends and receives the same amount of information on the cubes: rather than scheduling in detail the communication pattern, we can look at the problem from the flow control point of view. Each routing algorithm has a particular fingerprint in terms of network throughput with uniform traffic which tells us the maximum throughput that can be achieved for any packet size. By properly choosing a convenient size and by randomizing the message transmission we can exploit the network performance in a simple, efficient, robust and predictable way.

6 Conclusion

We have presented a simple and efficient randomized algorithm to implement the total-exchange on the family of k-ary n-cubes equipped with adaptive routing. This algorithm utilizes an interesting, and not previously used, property of the interconnection network that, with the uniform traffic, (1) reaches a steady state after few cycles (about 25), (2) can get a stable and high throughput (85% of the capacity) in the steady state, with oscillations in the network utilization/throughput that are less than 2%, and (3) is drained in few hundred cycles at the end of the communication pattern.

This scheme has several interesting features. First, it is very simple, requires no additional hardware to synchronize the processing nodes and can be used efficiently on the whole family of k-ary n-cubes, irrespective the values of k and n. It also exploits the characteristics of adaptive routers, which will eventually replace the deterministic routers that are currently in use in many existing multicomputers. Finally, it paves the way to a new strategy to efficiently implement dense collective communication pattern other that the total-exchange, as broadcast or personalized communication.

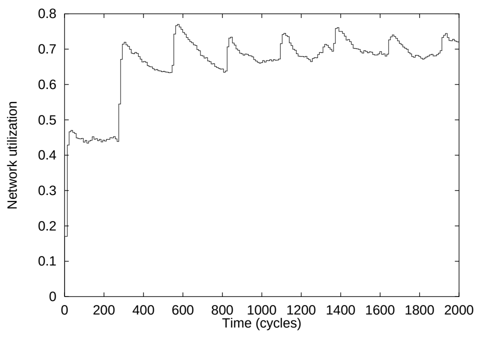

Figure 4. Network utilization during the execution of the Tseng and Gupta algorithm. The grain size of the total-exchange is 1024 bytes.

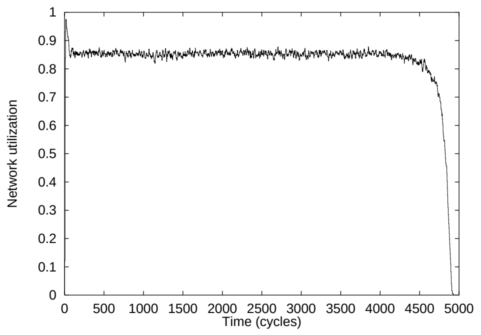

Figure 5. Network utilization during the execution of the randomized algorithm. The grain size of the total-exchange is 32 bytes.

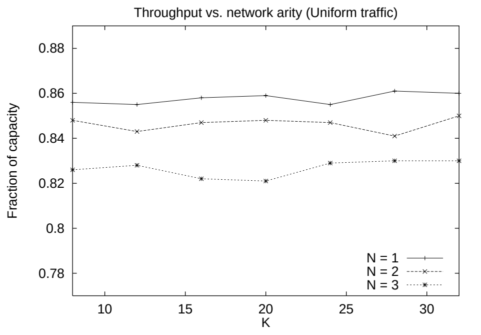

Figure 6. Saturation throughput under uniform traffic, varying the network arity and dimension in a k-ary n-cube. The packet size is fixed at 32 bytes.

References

[1] J. Duato. A New Theory of Deadlock-Free Adaptive Routing in Wormhole Networks. *IEEE Transactions on Parallel and Distributed Systems*, 4(12):1320–1331, December 1993.

[2] J. Duato and P. López. Performance Evaluation of Adaptive Routing Algorithms for k-ary n-cubes. In K. Bolding and L. Snyder, editors, *First International Workshop, PCRCW'94*, volume 853 of *LNCS*, pages 45–59, Seattle, Washington, USA, May 1994.

[3] S. Heller. Congestion-Free Routing on the CM-5 Data Router. In K. Bolding and L. Snyder, editors, *First International Workshop, PCRCW'94*, volume 853 of *LNCS*, pages 176–184, Seattle, Washington, USA, May 1994.

[4] S. S. Mukherjee and M. D. Hill. A Case for Making Network Interfaces Less Peripheral. In *Hot Interconnects V*, Stanford University, CA, August 1997.

[5] F. Petrini and M. Vanneschi. Minimal Adaptive Routing with Limited Injection on Toroidal k-ary n-cubes. In *Supercomputing 96*, Pittsburgh, PA, November 1996.

[6] F. Petrini and M. Vanneschi. SMART: a Simulator of Massive ARchitectures and Topologies. In *International Conference on Parallel and Distributed Systems Euro-PDS'97*, Barcelona, Spain, June 1997.

[7] S. L. Scott and G. M. Thorson. The Cray T3E Network: Adaptive Routing in a High Performance 3D Torus. In *HOT Interconnects IV*, Stanford University, August 1996.

[8] D. B. Skillicorn, J. M. D. Hill, and W. F. McColl. Questions and Answers about BSP. *Journal of Scientific Programming*, 1998. To appear.

[9] Y.-J. Suh and S. Yalamanchili. Algorithms for All-to-All Personalized Exchange in 2D and 3D Tori. In *Proceedings of the 10th International Parallel Processing Symposium, IPPS'96*, pages 808–814, April 1996.

[10] R. Thakur and A. Choudary. All-to-All Communication on Meshes with Wormhole Routing. In *Proceedings of the 8th International Parallel Processing Symposium, IPPS'94*, pages 561–565, Cancun, Mexico, April 1994.

[11] Y.-C. Tseng and S. K. S. Gupta. All-to-All Personalized Communication in a Wormhole-Routed Torus. *IEEE Transactions on Parallel and Distributed Systems*, 7(5):498–505, May 1996.

[12] Y.-C. Tseng, T.-H. Lin, S. K. S. Gupta, and D. K. Panda. Bandwidth-Optimal Complete Exchange on Wormhole Routed 2D/3D Torus Networks: A Diagonal-Propagation Approach. *IEEE Transactions on Parallel and Distributed Systems*, 8(4):380–396, April 1997.