Efficient Total-Exchange in Wormhole-Routed Toroidal Cubes

Fabrizio Petrini and Marco Vanneschi

Dipartimento di Informatica, Università di Pisa,
Corso Italia 40, 56125 Pisa, Italy,
tel +39 50 887228, fax +39 50 887226
e-mail: {petrini,vannesch}@di.unipi.it

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 a toroidal mesh with wormhole switching. This algorithm is based on an important property of the wormhole networks that reach high performance under uniform traffic using adaptive routing.
The experimental results, conducted on a 256 nodes bi-dimensional torus, show that this algorithm reaches a very high level of performance, around 90% of the optimal bound, and is more efficient than other algorithms presented in the literature.

1 Introduction

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.The efficient implementation of the total-exchange has been extensively studied in a variety of networks [TC94] [RSTG95].

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 pipelined manner. Network throughput of wormhole networks can be increased by organizing the flit buffers associated with each physical channel into several virtual channels [Dal92].

In this paper we present a simple randomized algorithm to schedule the total-exchange on a wormhole-routed toroidal mesh. Our work is motivated by the fact that wormhole networks can reach high throughput under uniform random traffic if packets are routed adaptively and if the packet size is properly chosen. 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 torus with 256 nodes using a detailed simulation model show that it is possible to achieve a network throughput that is very close to optimality.

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, based on Duato's methodology, that we consider in our study. Section 3 reviews some total-exchange algorithms presented in the literature and Section 4 motivates and describes our randomized algorithm. The results of the experimental evaluation are shown in Section 5. Finally, some concluding remarks are in given in Section 6.

2 Adaptive routing on the k-ary n-cubes

The toroidal meshes belong to the general class of the k-ary n-cube networks. 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, with wrap around connections on the borders.

Adaptive routing algorithms on the k-ary n-cubes are deadlock-prone and require sophisticated strategies for deadlock-avoidance. In this paper we utilize a minimal adaptive algorithm based on *Duato*'s methodology [Dua95]. This methodology only requires the absence of cyclic dependencies on a connected subset of the virtual channels. 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 [ST96].

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: *direct* algorithms, in which data are sent directly from source to destination and *indirect* algorithms, in which data are sent from source to destination through one or more intermediate nodes.

The *pairwise* exchange is a direct algorithm which requires $N-1$ steps, where N is the number of nodes. 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. An algorithm that works for non power-of-two cubes is the *shift* exchange, and requires only $N-1$ steps for any value of N. In this algorithm node pairs do not exchange messages with each other. Instead, at step i, a node j sends data to node $(i+j) \bmod N$ and receives data from node $(N+j-i) \bmod N$.

With indirect algorithms a message is sent from the source node to the destination through one or more intermediate nodes. The *indirect pairwise* exchange algorithm [TC94] aims at reducing the link contention of the pairwise exchange algorithm on the two-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.

4 A randomized total-exchange algorithm

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

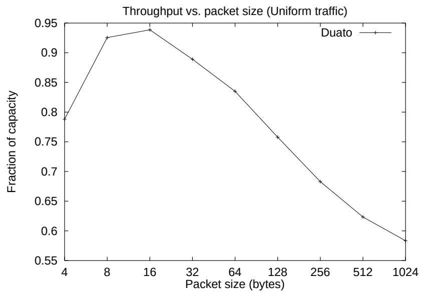

Fig. 1. Network throughput varying the packet size.

We can see that the network throughput is very sensitive to the packet size. The algorithm reaches about 90% of the optimal performance with packets between eight and 32 bytes. While this property is well known in the wormhole routing community, it has not still been exploited to schedule collective communication patterns. A viable solution to implement the total-exchange is to synthetically reproduce inside the network a uniform random traffic using an appropriate 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, $i \in \{0, \ldots, \lceil \frac{m}{p} \rceil - 1\}$ each node j generates an independent permutation $\Pi_{i,j}$ of the remaining $N-1$ nodes and sends 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.

5 Experimental results

Figure 2 shows the performance of these algorithms on a 256 nodes bi-dimensional cube. We use two graphs: in the first one we show the execution time of the total-exchange algorithms and in the second one we relate the throughput achieved at the end of the total-exchange with the bisection bandwidth. In all the graphs the input size on the x-axis represents the grain size, in bytes, of the information exchanged between any pairs of nodes. The graph in Figure 2 a) reports a lower bound, that is computed by considering the topological limitation of the bisection bandwidth. We can see that there is a gap between the randomized algorithm, that is very close to optimality with 16 and 32 bytes and the other algorithms.

In Figure 2 b) we can see that the fraction of network throughput achieved at completion of the total-exchange is around 90% with medium sized packets. The performance of the other algorithms is between 30% and 45%.

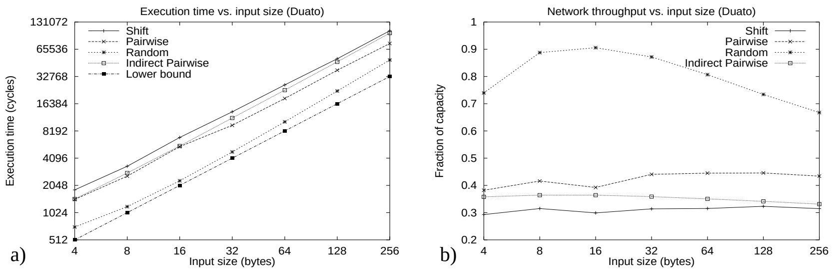

Fig. 2. Execution time and fraction of the network capacity achieved at completion of the total-exchange.

The near-optimal performance of the randomized algorithm can be explained looking at Figure 3 c). This algorithm exploits some basic characteristics of the uniformly distributed traffic with fixed packet size. When the grain size is 32 bytes, the network reaches the steady state of 90% active links after very few cycles (just 25 cycles in the example). 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. 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 90% of the throughput. 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.

The other algorithms, that impose a deterministic scheduling, clashes against the flow control characteristics of the adaptive routing algorithm. The network utilization is not stable during the execution of the communication pattern.

6 Conclusion

We have shown that a simple randomized algorithm that exploits the low level communication characteristics of the interconnection network can be more efficient than the deterministic approaches. 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 (90% of the capacity) in the steady state, with oscillations in the network utilization/throughput that are less than 2%, (3) and is drained in few hundred cycles at the end of the communication pattern.

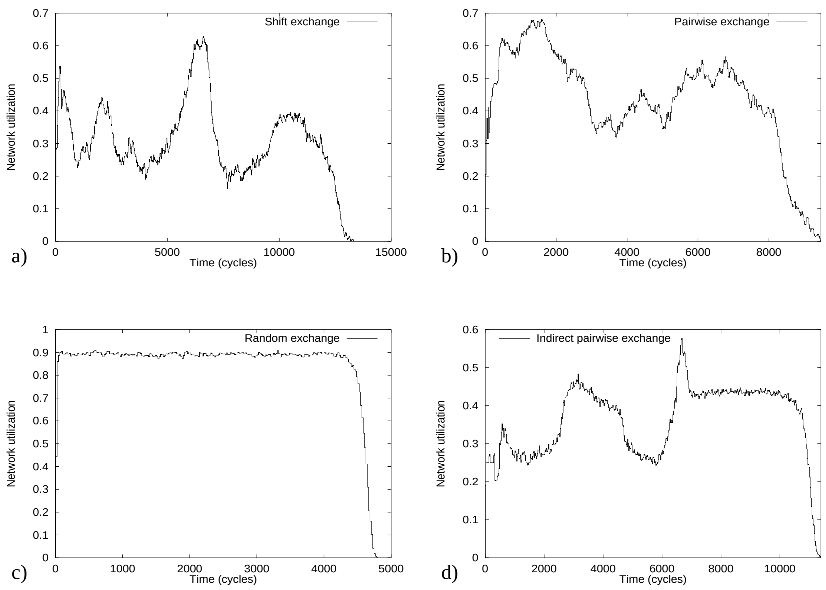

Fig. 3. Network utilization during the execution of the four exchange algorithms with the Duato's routing algorithm. The packet size is 32 bytes.

This scheme has several interesting features. First, it is very simple and requires no additional hardware to synchronize the processing nodes. It also exploits the characteristics of adaptive routers, which will eventually replace the deterministic routers that are currently in use in many existing multicomputers.

References

[Dal92] William J. Dally. Virtual Channel Flow Control. *IEEE Transactions on Parallel and Distributed Systems*, 3(2):194–205, March 1992.

[Dua95] José Duato. A Necessary and Sufficient Condition for Deadlock-Free Adaptive Routing in Wormhole Networks. *IEEE Transactions on Parallel and Distributed Systems*, 6(10):1055–1067, October 1995.

[RSTG95] Satish Rao, Torsten Suel, Thanasis Tsantilas, and Mark Goudreau. Efficient Communication Using Total-Exchange. In *Proceedings of the 9th International Parallel Processing Symposium, IPPS'94*, Santa Barbara, CA, April 1995.

[ST96] Steven L. Scott and Gregory M. Thorson. The Cray T3E Network: Adaptive Routing in a High Performance 3D Torus. In *HOT Interconnects IV*, Stanford University, August 1996.

[TC94] Rajeev Thakur and Alok 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.

This article was processed using the LaTeX macro package with LLNCS style