OPTIMIZATION OF RANDOMIZED ROUTING ON BENES NETWORKS

Abdou Youssef and **Hoda El-Sayed**

Department of EECS
The George Washington University
Washington, DC 20052
youssef@seas.gwu.edu
Fax: (202) 994-0227

ABSTRACT

Randomized self-routing on Benes-Clos networks has been shown to yield high routing performance in SIMD and MIMD systems, by evenly distributing the traffic load. The performance can be further improved by optimizing the Benes design parameters and by refining the randomization process. In this paper we will optimize the Benes design parameters for maximum randomized routing performance. The design parameters are: the switch size, the link width, the total number of columns, and the number of randomizing columns. Due to cost and technological constraints, the optimization is carried out assuming a fixed number of processors and a fixed number of IO pins per switch-chip. To control the number of randomizing columns, we introduce the notion of r-truncated Benes, derived by removing r randomizing columns from Benes. This was motivated by the intuitive hypothesis that the randomization advantage is obtained from just a few randomizing columns, and that having additional randomizing columns contributes nothing to randomization while increasing source-destination distances. Our performance evaluations will show that (1) the optimal switch sizes are in the low-to-mid-level range; (2) the optimal number of randomizing columns is just one, which is highly desirable because of the reduced hardware cost; and (3) multiple randomization offers great performance improvement over standard randomization.

KEYWORDS

Benes networks, truncated Benes networks, randomized routing, design optimization, latency

INTRODUCTION

Massively parallel systems have a great potential for satisfying in a cost-effective manner the ever increasing demand for high processing speed. At the heart of those systems is the interconnection network. Standard networks include static networks such as meshes and hypercubes, and multistage interconnection networks (MIN's) such as the Omega network [5] and Benes networks [1].

Benes networks enjoy several advantages over other networks. They are superior to banyan MIN's for their ability to route all permutations and for their fault-tolerance potential. They are also superior to static networks in two respects. First, they require their processors to have only one input port and one output port, which is less than what is required in most static networks. Second, with efficient Benes routing algorithms, the problem of task-to-processor assignment is straightforward in Benes networks but is often very costly in static networks.

To achieve efficient Benes routing, several algorithms have been developed, all of which are for permutation routing. For arbitrary permutations, the standard sequential looping algorithm takes $O(N \log N)$ time to control the switches of an N-processor Benes network [10] (i.e., time to determine the appropriate switch settings to realize a given permutation). This algorithm is too costly for run-time control. Although two parallel routing algorithms have been developed where one takes $O(N)$ time [6] and the other $O(\log^2 N)$ time [4], they are of theoretical value only — the first is still too costly for run-time control and the second requires a prohibitive fully-connected static network of N nodes to run. Fast Benes routing algorithms have been found for special classes of permutations [7, 9, 11, 12], but these algorithms do not apply to arbitrary permutations. The inadequacy of these permutation routing algorithms in terms of speed or generality, and the lack of algorithms for efficient asynchronous routing on Benes networks, stress the need for alternative approaches to Benes Routing.

A better alternative approach is randomized routing. In this approach, whenever a path is to be established between a source-destination pair (s, d) in an m-column MIN, the source randomly selects an m-digit *control tag* which is then locally used by the switches to establish a path from s to d. This approach has three advantages. First, it is self-routing and thus costs a small constant time for control. Second, it applies to arbitrary permutations and to asynchronous (MIMD) routing. Third, the routing delay incurred by conflicts is very small as shown by Youssef [13], and by Bhatia and Youssef [3]. In [13] the first author applied randomized routing on Clos networks (3-stage Benes) and showed that the delay per permutation is at at most 5 network cycles with overwhelming probability on a circuit-switched Clos network, and that was confirmed experimentally in [3]. (A network cycle is the time interval during which non-conflicting source-destination paths can be established and the corresponding messages transferred.) These highly encouraging results prompted us to extend randomized routing to Benes networks of arbitrary switch sizes and numbers of columns.

In this paper we will optimize the Benes design parameters for maximum randomized routing performance. The design parameters are: the switch size, the link width, the total number of columns, and the number of randomizing columns. Due to cost and technological constraints, the optimization is carried out assuming a fixed number of processors and a fixed number of IO pins per switch-chip.

To control the number of randomizing columns, we introduce the notion of r-truncated Benes, derived by removing r randomizing columns from Benes. This was motivated by the intuitive hypothesis that the randomization advantage is obtained from just a few randomizing columns, and that having additional randomizing columns contributes nothing to randomization while increasing source-destination distances.

Our performance evaluations will show that (1) the optimal switch sizes are in the low-to-mid-level range; (2) the optimal number of randomizing columns is one, which is highly desirable because of the considerably reduced hardware cost; and (3) multiple randomization offers great performance improvement over standard randomization.

OVERVIEW OF BENES NETWORKS

Throughout the paper, let q and n be two arbitrary integers such that $q \geq 2$ and $n \geq 1$, and let $N = q^n$. A q-ary Benes network of size N, denoted here by $B(q, n)$, is a multistage network that has N input terminals (or sources) representing processors, N output terminals (or destinations) representing processors or memory modules, and $2n-1$ columns of $\frac{N}{q}$ switches of type $q \times q$ crossbar (q is the *switch size*). The input terminals and the output terminals are labeled 0 through $N-1$ top to bottom. The switches in each column are labeled 0 through $\frac{N}{q} - 1$ from top to bottom, and the input ports and the output ports of each switch are internally labeled 0 through $q-1$ top to bottom. The columns are labeled $2n-2,\ 2n-3,\ \ldots\ ,0$ from left to right. The inter-column connectivity of $B(q, n)$ is defined recursively as follows. For $n = 1$, $B(q, 1)$ is a $q \times q$ switch. For $n > 1$, $B(q, n)$ is synthesized by inserting q copies of $B(q, n-1)$ between two columns of $\frac{N}{q}$ switches as shown in Figure 1-(a): Output port j of switch i of the leftmost column is linked to the i-th input of the j-th copy of $B(q, n-1)$, and, symmetrically, the i-th output of the j-th copy of $B(q, n-1)$ is linked to input port j of switch i of the rightmost column. Figure 2-(b) shows $B(2, 3)$. An important special case is when $n = 2$: The network $B(q, 2)$ is a Clos network which has 3 columns [2].

As will be discussed in the next section, the randomization in our routing algorithms takes place in the leftmost $n-1$ columns. Clearly, there is a tradeoff between the amount of randomization possible (i.e., number of randomizing columns) and the source-destination latency. The more randomizing columns, the more one can randomize —and thus reduce contention— but the longer the signal has to travel from source to destination. To optimize this tradeoff, we extend the definition of Benes networks to truncated Benes whereby the number of randomizing columns can be reduced.

Definition: An *r-truncated Benes*, denoted $rB(q, n)$, is derived by removing the leftmost r columns from $B(q, n)$, where $r = 0, 1, ..., n-1$. Figure 1-(c) and 1-(d) show two examples of truncated Benes.

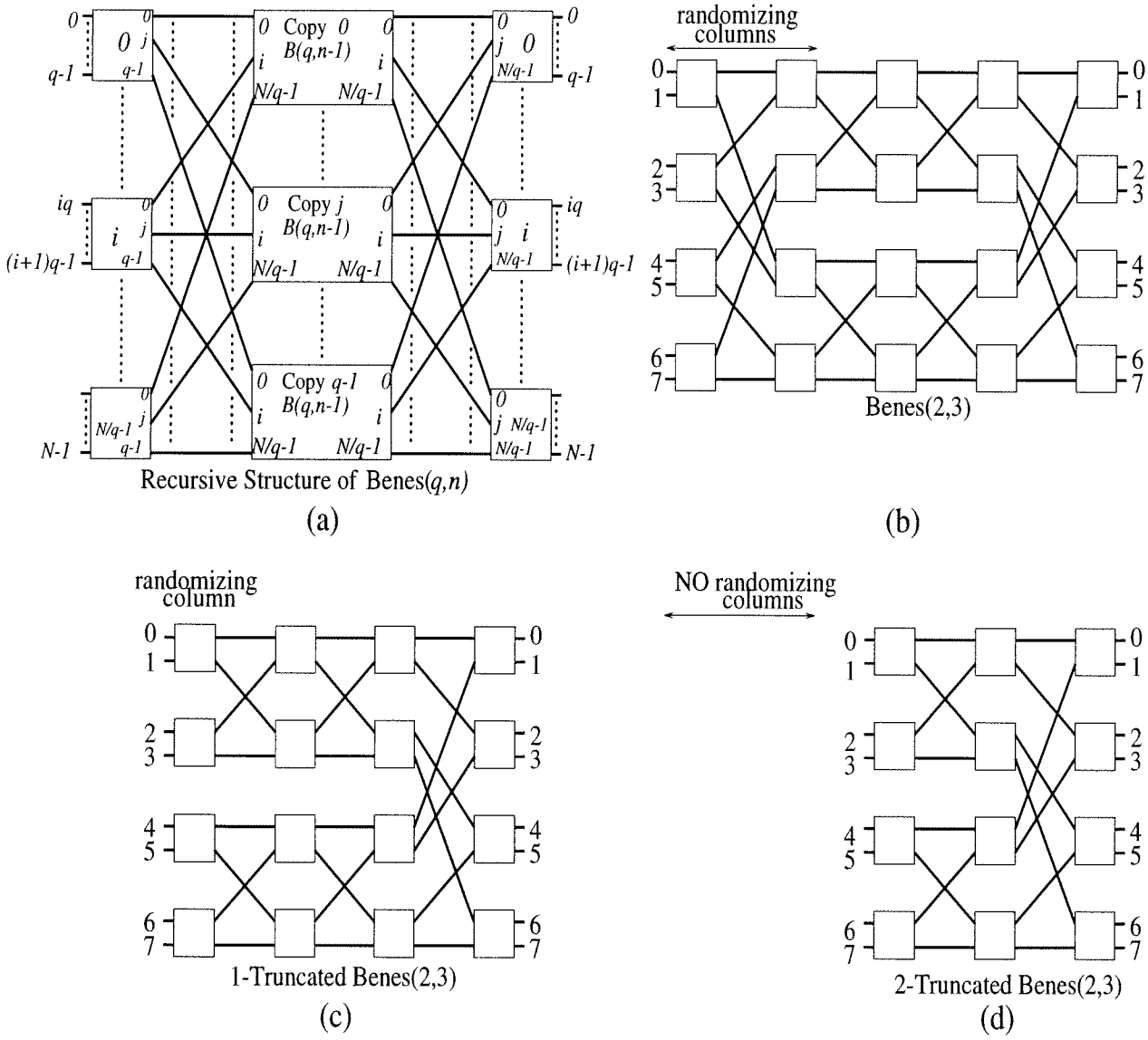

Figure 1: Structure of Benes and Truncated Benes Networks

Note two special cases, $r = 0$ and $r = n - 1$. When $r = 0$, no truncation takes place, that is, the network is the whole $B(q, n)$. When $r = n - 1$, all the randomizing columns are removed, and the remaining network is an Omega-equivalent network [5]. No more than $n - 1$ columns can be truncated because otherwise the network would lose full connectivity, that is, certain input terminals would not be able to reach certain output terminals.

RANDOMIZED ROUTING ON r-TRUNCATED BENES

Every source-destination path in r-truncated $B(q, n)$, from any source s to any destination d, is characterized by a $(2n-1-r)$-*digit control tag* (CT) $z_{2n-r-2}z_{2n-r-3}...z_0$ (in base q) which is locally used by the switches to establish the path: a switch in column i uses z_i to link the incoming path with output port z_i of the switch. It can be easily shown that the n rightmost digits $z_{n-1}...z_0$ of the CT form the q-ary representation of the destination address d [13]. The remaining (leftmost) $n-1-r$ digits are arbitrary, and can thus be chosen randomly. This justifies the generic randomized routing approach presented next.

Generic Randomized Routing: Whenever a source s has to send a message to destination d in

a circuit-switched r-truncated $B(q,n)$, the following steps are performed:

1. The source s forms a control tag by generating uniformly randomly $n-1-r$ q-ary digits and appending them to the q-ary representation of d.
2. The source s attempts to establish the s–d path characterized by the generated CT. If the path is established, then the corresponding message is sent; the time duration for establishing a conflict-free path and sending the message is called a *network cycle*. If the path is blocked because of conflict, the source attempts again in the next network cycle.

When a permutation is to be routed, all the sources execute the algorithm above simultaneously, and the delay (or latency) is measured in the number of networks cycles (i.e., maximum number of attempts) needed for all the sources in the permutation to send their messages successfully.

At this point a distinction can be made in the way the repeated attempts are made, resulting in two randomization algorithms: *Single Randomization* (SR), and *Multiple Randomization* (MR). In SR repeated attempts to establish a path use the same control tag generated at the outset (before the first attempt). In MR every repeated attempt uses a new randomly generated control tag. Note that SR is similar to an algorithm studied in [8], but MR is a new algorithm.

This author has proved in another paper [13] that for any arbitrary permutation in any arbitrary Clos network $B(q,2)$, the permutation delay under Single Randomization is at most 6 network cycles with probability $\geq .995$. This was later verified experimentally in [3], where also a comparison was performed between SR and MR. The results are plotted Figure 2-(a) and 2-(b), for average and maximum permutation delay, respectively, computed over the several hundred randomly selected permutations to measure the performance of the algorithms. The results clearly show that the delay is truly small, and that MR is better than SR.

OPTIMIZATION OF r-TRUNCATED BENES FOR MR ROUTING

For a given machine size (number of processors) N, different Benes (truncated or not) networks can be defined by varying n and q while keeping $N = q^n$. Also, by introducing truncation, one can vary the number of truncated columns r. Another parameter that can be varied is the link (or channel) width W; assuming that every switch is a single VLSI chip, the number of IO pins is then fixed by technology. Consequently, the number P of pins for the input ports of a switch is fixed. Clearly, $P = W * q$, implying that as the (logical) switch size q varies, the channel width W must also vary (so that $W * q$ remains constant), thus affecting the number of packets into which a fixed-size message is subdivided.

We considered two machine sizes, $N = 1024$ and $N = 4096$, assuming pin size $P = 256$, and varying q, n, r, and W, subject to $W * q = 256$, $q^n = N$, and $r = 0, 1, ..., n-1$. Several hundred randomly selected permutations were routed using the multiple randomization algorithm,and the latencies of the permutations were measured and the average latency computed. Note that because the number of columns vary, the time unit used for measurement is the clock cycle rather than the network cycle. The clock cycle is the time duration needed to set a switch; it is also long enough to allow a signal to travel from source to destination once a circuit-switched path is established.

The performance results (in terms of average latency) are shown in Figure 2-(c) and (d) for $N = 1024$ and $N = 2096$, respectively. The figures correspond to all possible combinations of (q, n, w, r). In both cases of N, the best performance is achieved when $r = n-2$, that is, when the number of randomizing columns is just one. This is a very favorable finding because of the low hardware cost advantage. Another conclusion drawn from the figures is that the optimal switch size is 4×4 for $N = 1024$, and 8×8 for $N = 4096$.

CONCLUSIONS

In this paper we applied randomized routing to Benes networks, extended the definition of Benes networks to truncated Benes networks, and optimized the design parameters of truncated Benes

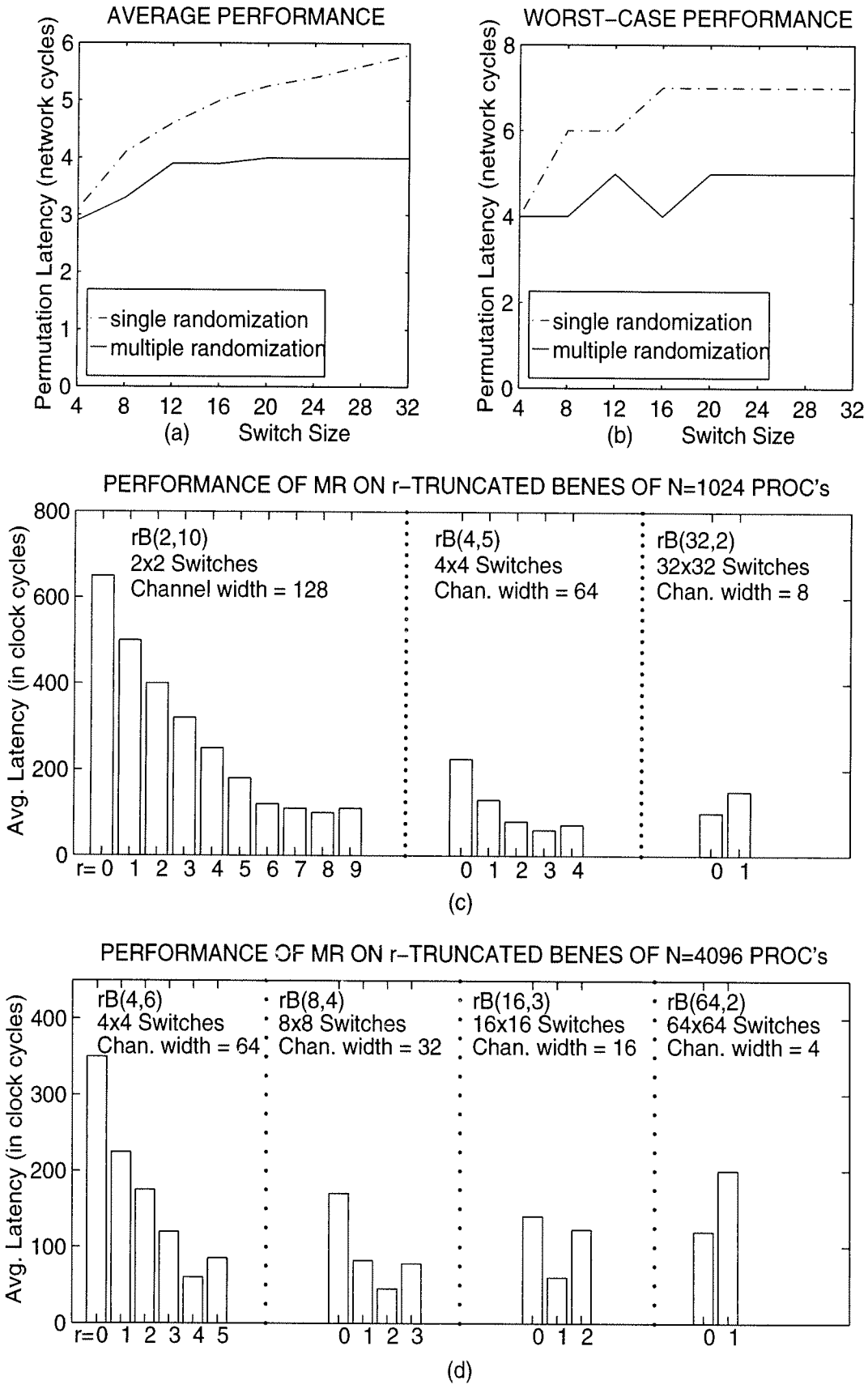

Figure 2: Performance and Optimization of Randomized Routing on Benes

under randomized routing. The universality, self-routedness and very low latency of randomized routing, and the low hardware cost of optimized truncated Benes, make optimized Multiple Randomization and Benes networks quite desirable as interconnection networks for parallel systems.

Multiple Randomization has also other advantages, such as high fault tolerance capabilities, not only with fault diagnosis but also without fault diagnosis. These capabilities and other randomized routing extensions have been considered by the authors and other researchers [3], and their findings will appear elsewhere.

References

[1] V. E. Benes, *Mathematical Theory on Connecting Networks and Telephone Traffic,* Academic Press, New York, 1965.

[2] C. Clos, "A Study of Non-Blocking Switching Networks," *Bell System Tech. J.*, Vol. 32, pp. 406–424, 1953.

[3] Bhatia and A. Youssef, "Performance Analysis and Fault Tolerance of Randomized Routing on Clos Networks," *6th Symposium of the frontiers of Massively Parallel Computation*, October 1996, Annapolis, MD.

[4] T. Feng and W. Young, "An $O(\log^2 N)$ Control Algorithm," *Proc. of the Int'l Conf. Parallel Processing*, pp. 334–340, 1985.

[5] D. H. Lawrie, "Access and Alignment of Data in an Arrary Processor," *IEEE Trans. Comput.,* C-24, pp. 1145–1155, Dec. 1975.

[6] K. Y. Lee, "A New Benes Network Control Algorithm," *IEEE Trans. Comput.,* C-36, pp. 768–772, May 1987.

[7] J. Lenfant, "Parallel Permutations of Data: A Benes Network Control Algorithm for Frequently Used Permutations," *IEEE Trans. Comput.,* C-27, pp. 637–647, July 1978.

[8] D. Mitra and R. A. Cieslak, "Randomized Parallel Communications on an Extension of the Omega Network," *J. ACM*, Vol. 34, No. 4, pp. 802–824, Oct. 1987.

[9] D. Nassimi and S. Sahni, "A Self-Routing Benes Network and Parallel Permutation Algorithms," *IEEE Trans. Comput.,* C-30, pp. 332–340, May 1981.

[10] D. C. Opferman and N. T. Tsao-Wu, "On a Class of Rearrangeable Switching Networks, Part I: Control Algorithm," *Bell Syst. Tech. J.*, Vol. 50, pp. 1579–1600, May–June 1971.

[11] C. S. Raghavendra and R. Boppana, "On Self-Routing in Benes and Shuffle-Exchange Networks," *IEEE Trans. Comput.*, Vol. 40, No. 9, pp. 1057–1064, Sept. 1991.

[12] A. Youssef and B. Arden, "A New Approach to Fast Control of $r^2 \times r^2$ 3-Stage Benes Networks of $r \times r$ Crossbar Switches," *Proc. 17th Int'l Symp. Computer Architecture,* Seattle, pp. 50–59, May 1990.

[13] A. Youssef, "Efficient Randomized Routing on Clos Networks," *the 5th Int'l Parallel Processing Symposium*, Anaheim, CA, April 1991, pp. 410–415.