Randomized Routing and the r-Truncated Benes Networks

Hoda El-Sayed
EECS Department
The George Washington University
Washington, D.C., 20052

Abdou Youssef
EECS Department
The George Washington University
Washington, D.C., 20052

Abstract

Benes networks have the potential for balanced traffic, fewer conflicts, and can route any permutation in one pass due to their multiplicity of paths. Omega networks , on the other hand, are favored, due to their fast set-up and low hardware cost, although they could take more than one pass to route a permutation. This paper introduces a new class of networks referred to as r-truncated Benes networks, which combines the benefits from Omega networks and Benes networks. An r-truncated Benes network is created by eliminating the first r stages of a Benes network, where $0 \leq r \leq n-1$ in a Benes network with $2n-1$ stages. When $r = n-1$, the network becomes an Omega. Using randomized routing, we will show that r-truncated Benes networks is an excellent trade-off between Omega and Benes networks. In particular, we will show that with the small cost of one stage of randomization over Omega, r-truncated Benes networks are superior to Omega in performance.

1 Introduction

The study of interconnection networks has become one of the most popular research areas in parallel processing, as communications overhead is one of the most important factors affecting the performance of parallel computer systems. Multistage interconnection networks (MINs) [7,5,17] can be divided into three classes: blocking, nonblocking, and rearrangeable networks [6]. In blocking networks, simultaneous connections of more than one terminal pair, input and output, may result in conflicts over communication links. Examples of blocking networks are Omega[8] and baseline networks. Nonblocking networks, such as certain Clos [3] networks, can handle all possible one-to-one connections without conflict. A network is rearrangeable if it can perform all possible connections between inputs and outputs by rearranging its existing connections so that a connection path for a new input-output pair can be established. Benes network is a well-known network that belongs to this class [2]. Although Benes/Clos networks have received less attention because of the aforementioned two disadvantages, recent advances in technology and routing techniques have largely eliminated those disadvantages, thus making Benes/Clos networks competitive alternatives worthy of further study. Specifically, large crossbar switches can be built affordably due to the advances in VLSI technology. An implementation of optical Clos network has been recently reported by Lin, Krile, and Walkup [12]. In routing, a recent study by Youssef [19] has developed efficient universal randomized self-routing algorithms for Clos networks.

The two main disadvantages of the Clos/Benes networks have been high hardware cost and slow routing speed. Many Benes routing algorithms have been developed. Waksman introduced the best known Benes routing algorithm that runs in O(NlogN) time on a single processor machine [15]. Several other parallel algorithms were developed that take less time. Lee developed a Benes control algorithm that takes O(N) parallel time, which is still slow [9]. Nassimi and Sahni came up with a self-routing parallel algorithm for fast set-up in switches that takes O(logN) time; however, the algorithm does not work for all permutations [14]. Lenfant followed another approach that restricts the algorithm to a set of frequently used bijections. Hence the algorithm still does not realize all classes of permutation [10]. Lev and Pippenger implemented an algorithm that

takes $O(N\log^2 N)$ serial time for one processor and $O(\log^2 N)$ parallel time [11]. Later, Nassimi and Sahni developed another parallel routing algorithm for Benes networks. The algorithm routes Lenfant class of permutations and takes $O(\log^2 N)$, and does realize all permutations. Youssef and Arden introduced a new approach for fast routing control. The algorithm self-routes several classes of frequently used permutations, and takes $O(\log^2 N)$ time to route arbitrary permutations on Benes-Clos networks [18]. Raghavendra and Boppana developed a self-routing algorithm for self-routing the linear class of permutations, but does not realize all permutations[16]. Mitra and Cieslak [13] introduced a randomized routing scheme on extended Omega networks. Their algorithm, however, uses single randomization and in this paper we use multiple randomization on r-truncated Benes networks.

Omega networks are more frequently used due to their fast routing and low hardware cost. Those networks, however, are blocking and therefore could take more than one pass (which means longer delay) in routing a permutation. This is mainly due to their single path property, which leaves no room for randomization. On the other hand, Benes networks realize all permutations in single passes. Benes networks[1], however, have more stages than Omega networks[8]. This has given rise to having multiple paths between any pair of nodes in a Benes network, which allows randomization. On the other hand, the increased number of stages caused Benes networks to have longer routing delay and more hardware cost than Omega networks. Due to those conflicting properties, cost of hardware and the ability of randomization, we are motivated to study a new network. This network, which is called here the *r* -truncated Benes network, is a trade-off between Omega and Benes networks.

Our performance evaluation shows that with only one stage of randomization, r-truncated Benes network is superior to Omega networks.

The rest of the paper is organized as follows. The next section gives a brief overview of r-truncated Benes networks. Section 3 will discuss the routing algorithm used on r-truncated Benes networks. Section 4 will present the details of the performance analysis. Section 5 contains the conclusions and some ideas for future work.

2 Overview of r-truncated Benes Network

A Benes network, $B(q,n)$, has $N\ (= q^n)$ inputs and N output terminals, and consists of $2n-1$ stages of N/q crossbar switches each, where each switch is size qxq crossbar. Such network can be defined recursively. For $n = 1$, $B(q,1)$ is a single qxq crossbar. For $n \geq 2$, the recursive structure is shown in Figure 1 specifically,

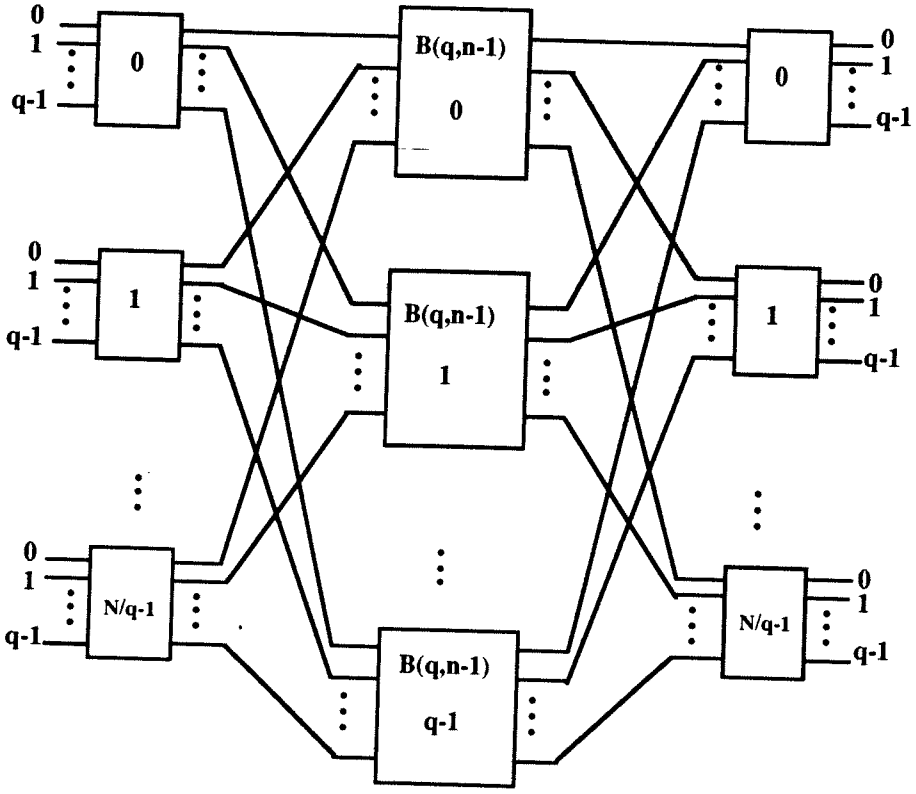

Figure 1

Benes Network B(q,n)

the middle stage consists of q copies of $B(q,n-1)$ numbered $0,1,\ldots,q-1$ from top to bottom; the first stage connects the i-th output port of the j-th switch (in the first column) to the j-th input of the i-th $B(q,n-1)$ of the middle stage; and the last stage is the mirror image of the first stage. The stages are numbered from 1 to $2n-1$ from left to right.

In this paper a high-performance routing algorithm for r-truncated Benes networks is developed through randomized self-routing algorithms. In this randomized approach, switches

that are in stages from *1* to *n-1* are set randomly. On the other hand, switches that are in the stages *n* through *2n-1* of the network are set according to a self-routing mechanism using the destination address [4]. It is known that after a message crosses the first *n-1* stages, the message can be self-routed using the destination address. This can be illustrated by considering a three stage Benes network *B(q,2)* where the input or output *y* of a switch *x* in any stage has a global label *[xy]*. Suppose that a message is to be sent from output port *[xy]* in the first stage, to the output terminal *[x'y']* in the last stage. The message first enters the input port *[yx]* in the middle stage, then, using the digit *x'* of the destination address *[x'y']*, the message exits through output port *[yx']* in the middle stage. Afterwards, it enters the input port *[x'y]* of the last stage and then using digit y' of the destination address *[x'y']*, it reaches output port *[x'y']* of the last stage, which is the desired destination. Consequently, to go from any input terminal *[vu]* of *B(q,2)* to any output terminal *[x'y']*, we can select a digit z', $0 \leq z' \leq q-1$, and form the control tag $z'x'y'$ to find a path from *[vu]* to *[x'y']* in *B(q,2)* [18]. The same is applied generally to any number of stages in *B(q,n)*. To go from a source *s* to a destination *d*, *s* forms a control tag $c_1c_2...c_{2n-1}$ where the part $c_1c_2...c_{n-1}$ is selected uniformly randomly and is called the *randprt*, and the part $c_nc_{n+1}...c_{2n-1}$ is the destination address *d* in *q*-ary, and is called the *fixedprt*. Every c_i is a *q*-ary digit and is used by stage *i* to exit the paths through output port c_i of the appropriate switch.

A truncated Benes network is derived mainly from a Benes network and we will refer to it as an *r*-truncated Benes *B(q,n,r)* network. By deleting the first *r* stages of a Benes(*q,n*), where $0 \leq r \leq n-1$, we obtain the r-truncated Benes network. When $r = n-1$, the network becomes Omega, and when $r = 0$, the network is a Benes network. Figure 2 shows a complete 5-stage Benes network B(2,3), with a 2x2 switch size. The first two stages are for randomization, in this case r = 0, 1, or 2, when r = 0 it is a Benes network and when r = 2 it becomes an Omega.

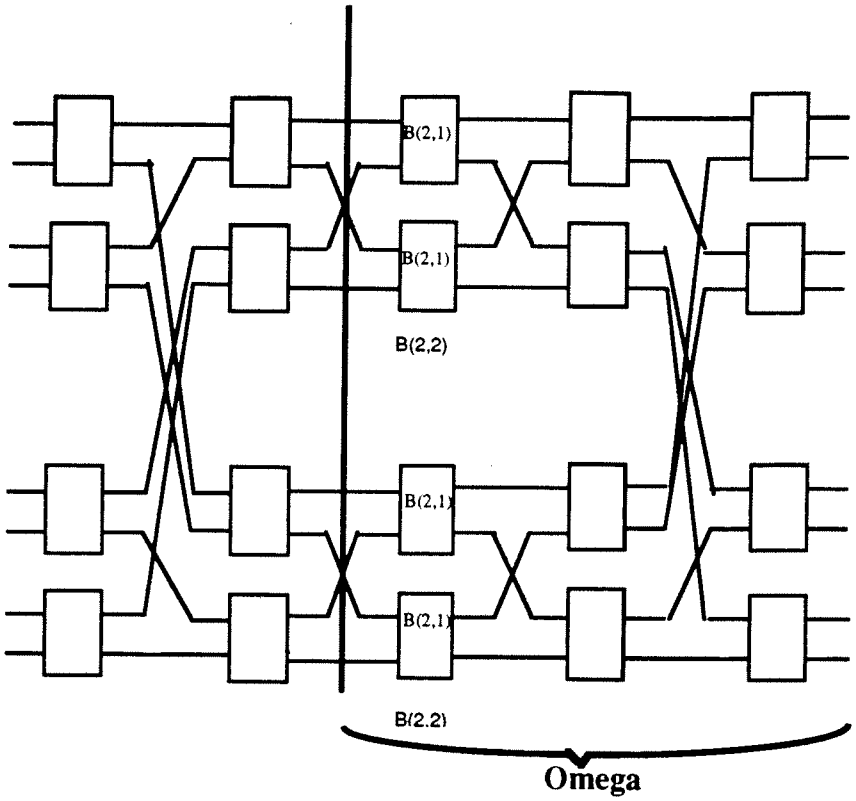

Figure 2

3 Randomized Routing Algorithm on r-Truncated Benes Networks

Multiple randomized circuit-switched routing is used with different values of r (number of stages to be truncated) to determine the optimal value of r. In this randomization approach, switches that are in stages 1 to $n-1-r$ are set randomly and r is any value $0 \leq r \leq n-1$. The number of stages for randomization in an r-truncated Benes network is $n-1-r$ as opposed to $n-1$ stages in a Benes network. On the other hand, switches that are in the remaining stages of the network are set according to a self-routing mechanism using the destination address. . In multiple randomization, new output ports are selected randomly for the stages $1 \leq stage \leq n-1$ for each message after each failed attempt to send the message due to a conflict. Randomized routing is performed on SIMD routing (permutations) for different values of r .

4 Performance Analysis

4.1 Simulation Set Up

The following parameters are used in the simulation:

N = machine size (number of input/output processors)

L = length of the message (in flits)

M = message size (in bits)

p = number of pins on each side of a switch

q = number of ports on each side of a switch (qxq is the logical switch size)

W = width of the channel (i.e., number of wires per channel, where each wire holds one bit)

Note that $p = qw$ and $L = \frac{M}{W} = \frac{Mq}{p}$

For practical considerations, we have considered the following values for p, N, and M :

- $p = 256$ pins
- $N = 1024$ and 4096
- $M = 128$ bits for $N = 1024$, and $M = 64$ bits for $N = 4096$.

Accordingly, $L = \frac{q}{2}$ for $N = 1024$, and $L = \frac{q}{4}$ for $N = 4096$.

The table below indicates the different legitimate values for q, L, and n ($N = q^n$) when $N = 1024$:

q	32	4	2
n	2	5	10
L	16	2	1

Table 1. Parameters used with 1024-processor Benes Networks

The table below indicates the different legitimate values for q, L, and n ($N = q^n$) when $N = 4096$

q	64	16	8	4
n	2	3	4	6
L	16	4	2	1

Table 2. Parameters used with 4096-processor Benes Networks

4.2 Performance Evaluation Results

In this section we study permutation routing using randomized routing on r-truncated Benes networks to determine the optimal value of r. This study does not only compare randomized routing on Omega versus Benes, but also quantifies the trade-off and identifies the optimal hybrid.

In the simulation, latencies are measured for several permutations with respect to the different parameter values of q, n, L, and r. Each permutation is performed using multiple randomized circuit switching. Figures 3 and 4 show the average latencies determined for several different random permutations when $N = 1024$ and $N = 4096$, respectively, as a function of the different values of r (truncated stages) for different networks. With $N = 1024$, the network is capable of having 3-stages, 9-stages, and 19-stages, each with different channel width and switch sizes. From the graph of figure 3, where the permutations are randomly chosen with $N = 1024$, the following observations are made :

- The latencies decrease as the number of stages decreases for $r = 0$ to $r = n\text{-}2$, and then moves up when $r = n - 1$ (corresponding to Omega).
- For the 3-stage network, r can be either 0 or 1. The latency is lower when $r = 0$, than when $r = 1$. As mentioned before, when $r = 0$, the network is a Benes network, while with $r = 1$, the

network becomes an Omega. In other words, keeping the randomized stage will give better performance in the network.

- For the 9-stage network, r can have different values from 0 up to 4. The latency has the lowest value when $r = 3$,. that is, when having, only one stage for randomization.
- For the 19-stage network , r ranges from 0 to 9. It is obvious from the graph that the latency decreases as the number of eliminated stages increases (r values). However, the latency has the lowest value when $r = 8$, which corresponds to only one stage for randomization.
- The optimal design parameter with $N = 1024$ on random permutations is with a 4x4 switch size in a 9-stage network and the optimal r value is with $r = 3$.

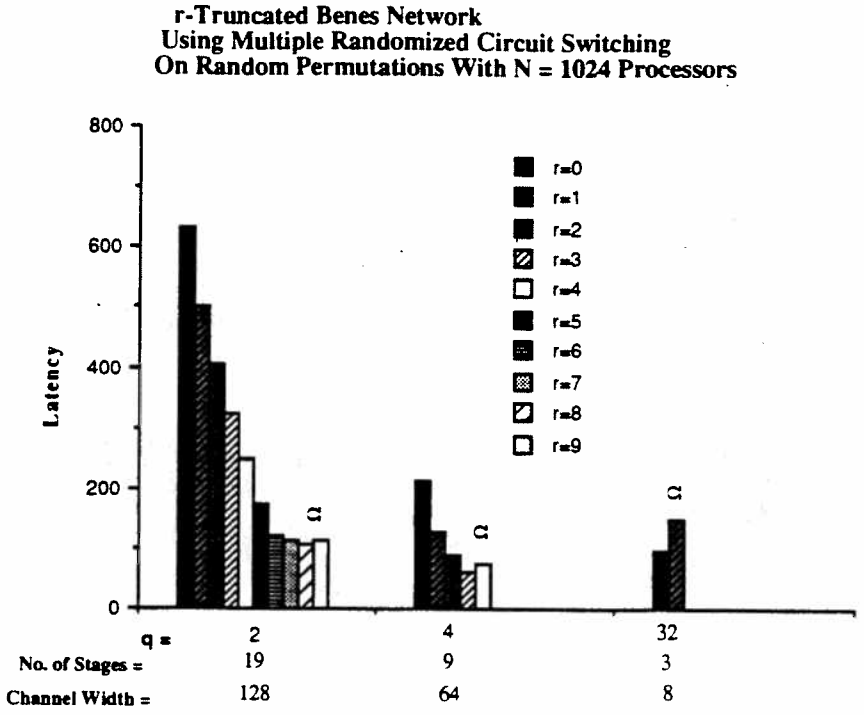

Figure 3

From the graph of figure 4, the following observations are made :

- In the 3-stage network, $r = 0$ has lower latency than when $r = 1$, which is the Omega.
- For the 5-stage network, $r = 1$ has the lowest latency, that is, with one stage for randomization.
- In the 7-stage network, $r = 2$ has the lowest latency, that is lower than the Omega when $r = 3$.

  In the 11-stage network, $r = 4$ has the lowest latency, which is with one stage of randomization.
- The optimal design parameter is with an 8x8 switch size of 7-stages, and the optimal value of r is with $r = 2$.

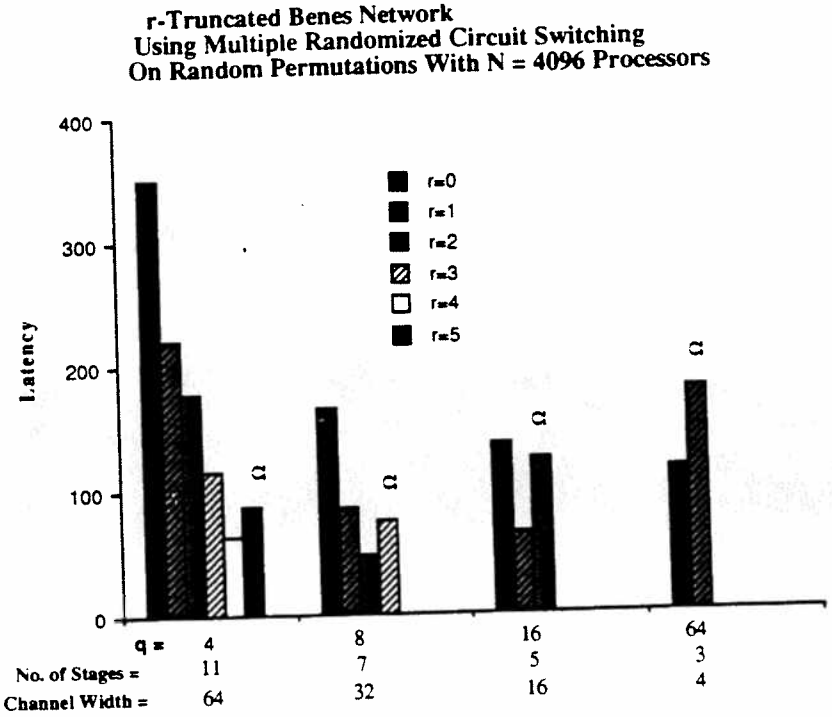

Figure 4

5 Conclusions

In this paper we introduced and investigated a new network called the r-truncated Benes network as a trade-off between Benes networks and Omega networks. The network is derived from a Benes network by eliminating r stages where $0 \leq r \leq n-1$, when $r = n-1$, the network becomes an Omega and when $r = 0$, the network is a Benes network. Simulations using the randomized routing algorithm was performed on the r-truncated network for permutation routing. The results show that with permutation routing, an optimal value of r is $r = n-2$. In other words, with only one stage of randomization, the message delay is even lower than the Omega network and the r-truncated network becomes superior to Omega. It was also observed that when using random permutations for *1024* processors, the 4x4 switch size is optimal, and for *4096* processors, the 8x8 switch size is optimal.

Randomized routing has also been used on r-truncated Benes networks using uniform MIMD traffic, our preliminary results show that Omega performs better than Benes and r-truncated Benes network. That is due to the fact that the network traffic is already uniform and randomization will be of no use in this case (as randomization provides natural traffic load balancing). It is anticipated that when using non-uniform MIMD traffic, r-truncated Benes network will perform better than Omega. This is beyond the scope of this paper, and will be pursued in future work.

References

[1] G.B. Adams III, "Fault-Tolerant Multistage Interconnection Networks, " *Computer*, vol. 20, no. 6, pp. 14-27, June 1987.

[2] V.B. Benes, *Mathematical theory on connecting networks and telephone traffic*, Academic Press, New York, 1965.

[3] C. Clos, "A Study of Non-blocking Switching Networks," *Bell System Tech Journal,* vol. 32, pp. 406-424, 1953.

[4] V.B. Benes, *Mathematical theory on connecting networks and telephone traffic*, Academic Press, New York, 1965.

[5] T. Feng, " A Survey of Interconnection Networks," *Computer* vol. 14, pp. 12-27, 1981.

[6] K. Hwang and Briggs, *Computer architecture and parallel processing*," McGraw-Hill, NY, 1984.

[7] Kai Hwang, *Advanced computer architecture: Parallelism salability programmability*, McGraw-Hill, NY, 1993.

[8] D. H. Lawrie, "Access and Alignment of Data in an Array Processor," *IEEE Trans. on Computers*, C-24, pp. 1145-1155, 1975.

[9] K.Y.Lee, "A new Benes network control algorithm," *IEEE Trans. Comput.*, vol.c-36, no. 6, pp. 768-772, June 1987.

[10] J. Lenfant, "Parallel permutation of data: A Benes network control algorithm for frequently used permutations, " *IEEE Trans. Comput.*, vol. c-27, no. 7, pp. 637-647, July 1978.

[11] G.F. Lev, N. Pippenger, and L.G. Valiant, " A fast parallel algorithm for routing in permutation networks, " *IEEE Trans. Comput.* vol. c-30, no. 2, pp. 93-100, Feb. 1981.

[12] Shing-Hong Lin, T.F.Krile, and J.F. Walkup, "Two-dimensional Optical Interconnection Network and its Uses," *Applied Optics*, vol. 27, no. 9, 1988.

[13] 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.

[14] D. Nassimi and S. Sahni, "A self-routing Benes network and parallel permutation algorithms, " *IEEE Trans. Comput.*, vol. c-30, no.5, pp. 332-340, May 1981.

[15] D. Nassimi and S. Sahni, "Parallel algorithms to set up the Benes permutation network, "*IEEE Trans. Comput.*, vol.c-31, no.2, pp. 148-154, Feb. 1982.

[16] C.S. Raghavendra and R.V. Boppana, "On self-routing in Benes and Shuffle-Exchange networks, " *IEEE Trans. Comput.*, vol. 40, no. 9, pp. 1057-1064, Sept. 1991.

[17] C. Wu and T. Feng, "On a class of multistage interconnection networks," IEEE Trans. Comput., vol. c-29, pp.694-704, Aug. 1980.

[18] A. Youssef and B. Arden, "A new approach to fast control of r^2xr^2 3-stage Benes networks of rxr crossbar switches," Proc. 17th Intl'l Symp. Comp. Arch., Seattle, pp.50-59, May 1990.

[19] A. Youssef, "Randomized Self-Routing Algorithms for Clos Networks," *Computers & Electrical*