A quasi-nonblocking self-routing network which routes packets in $\log_2 N$ time.

Giuseppe A. De Biase Claudia Ferrone
Annalisa Massini

Dipartimento di Scienze dell'Informazione, Università di Roma la Sapienza
Via Salaria 113, 00198 Roma, Italy

March 4, 2010

Abstract

In this work a self-routing multi-$\log N$ permutation network is presented and studied. This network has $\log_2 N$ depth and $N(\log_2^2 N + \log_2 N)/2$ nodes, where N is the number of network inputs. Its parallel routing algorithm runs in $\log_2 N$ time. The network architecture guarantees that only a negligeable quantity of information is blocked, while the quasi-totality of the information synchronously arrives in $\log_2 N$ steps at the network outputs. Thanks to its distributed algorithm this network works in pipeline, furthermore it has a modular architecture suitable for very large N, it can be used as very high performance fast packet switching fabric and it is suitable for information exchange in very large scale multiprocessor systems.

1 Introduction

One way to build space-division packet switches is the banyan based fabric which consists of a structure based on banyan interconnection networks (a class of multistage interconnection networks) [1][2]. The classical Shuffle, Omega, Delta, Butterfly, etc. are all banyan interconnection networks which are isomorphic among themselves. These networks have $\log_2 N$ stages and can be considered $\log_2 N$ depth directed graphs, where N is the number of the vertices (inputs) and of the edges (outputs) of the network. In space-division packet switches multiple concurrent paths from the inputs to the outputs are needed; also if multicasting is considered (each output port can be addressed by more than one input), the switching fabric can be based on a permutation network, where one-to-one connections between every input and every output are possible. When in a permutation network all input-to-output paths can always be estabilished simultaneously, this network is called *nonblocking*; banyan networks are *blocking*, but they can be used as components of more complex structures (such as *dilated* networks, *fused* networks and *replicated* or *stacked* networks which have $O(N \log_2 N)$ or $O(N \log_2^2 N)$ nodes [3]-[8]) which may become nonblocking or strictly nonblocking under certain conditions [3][9].

A lot of effort has been expended on the problem of fast routing on $\log_2 N$

or multi-log N permutation networks. One can distinguish two different approaches: deterministic and probabilistic. Results of deterministic approach in nonblocking networks are: a) the Batcher algorithm, which takes time $O(\log_2^3 N)$[1] on a $O(N \log_2^2 N)$ nodes network [10], and the Oruç algorithm which runs in the same time but with minor multiplicative constants [11]; b) the Upfal algorithm which runs on a *Multi-butterfly* network with $O(\log_2 N)$ time complexity, but with large constants [12]; on the same network Upfal's routing time was reduced by Leighton and Maggs [7].

In the probabilistic approach (applicable also to blocking networks) only part of the information arrives at its destination. In this approach the goal is the search of a structure on which a suitable algorithm minimizes the *blocking probability*. Several Authors studied $O(\log_2 N)$ time routing algorithms on various networks. Szymanski and Hamacher examined the behaviour of d-dilated and r-replicated banyan networks under the assumptions of: a) permutation requests, and b) random requests (uniform traffic) by means of an analytic model which gives the blocking probability [13]. Yoon and Lee examined the behaviour of the *B-Banyan* network (a butterfly in which backward links are introduced for blocked requests) [15], and recently Venkatesan and

[1]Generally the Batcher algorithm is considered to have $O(\log_2^2 N)$ time complexity. In a recent paper Oruç points out that its real time complexity is $O(\log_2^3 N)$ [11].

Mouftah present the behaviour of the balanced gamma network, a gamma network where forward links and buffers in the last stage are added [14]. These networks either have low efficiency[2] [13] or destroy the syncronization of information [15], or are based on a too rough analytic model [14].

In the present work a $N(\log_2^2 N + \log_2 N)/2$ nodes permutation network, built by means of $\log_2 N$ k-butterfly networks, is presented. On this network runs a parallel self-routing algorithm with $\log_2 N$ time complexity. A very high overall efficiency is reached (quasi-nonblocking network) and packets arrive synchronously in $\log_2 N$ steps at the network outputs. With this network, one can obtain a very high performance switching fabric for fast packet switching in broadband communication systems, or an efficient permutation device suitable for information exchange in large scale multiprocessor systems.

2 Quasi-nonblocking permutation networks

Many permutation networks presented in the literature are blocking, in particular the banyan ones. In blocking networks the ratio $\eta = r_{out}/r_{in}$, between the number of nonblocked requests r_{out} and the number of input requests r_{in}, is its efficiency. η must be computed taking into account the whole set of input-to-output permutations and it is a measure of permutation capability of

[2]In the next section the efficiency of a blocking network is defined.

such a network [13]. Thus the efficiency, η, is a measure of the nonblocking capability of a network, and nonblocking permutation networks have always $\eta = 1$. The efficiency can be enhanced increasing the number of physical paths between inputs and outputs as in d-dilated and r-replicated networks [3]-[8].

Definition 1 *A permutation network where:*

$$\eta = 1 - \varepsilon \qquad and \qquad \varepsilon \ll 1$$

is quasi-nonblocking.

From the usual definition of blocking probability of a network, $P_b = (r_{in} - r_{out})/r_{in}$ [8][13], follows $P_b = 1 - \eta$ and, consequently, a quasi-nonblocking network is a network where P_b is very close to zero.

3 A multi-$\log N$ permutation network

A new multi-$\log N$ quasi-nonblocking permutation network can be obtained by a particular superposition of $k = \log_2 N$ butterflies. On such a structure a probabilistic algorithm with $\log_2 N$ time complexity has been implemented.

3.1 Network description

The vertical section of the proposed network (for $N = 2^5$) is presented in Fig. 1a. The network is a three-dimensional structure, the planes of which are shown in Fig. 1b. One can see that:

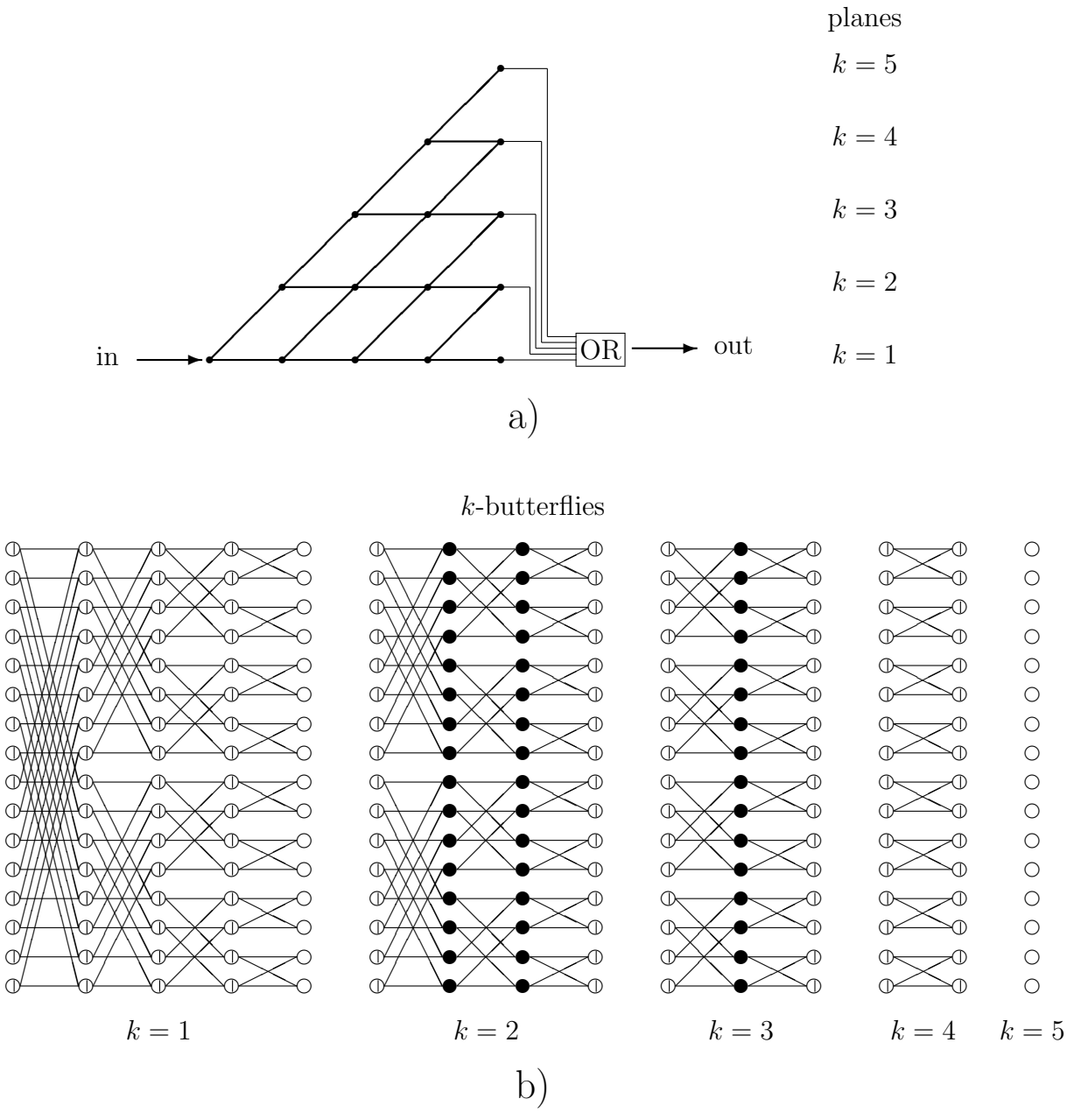

Figure 1: a) Vertical section of the permutation network for $N = 2^5$. The corresponding outputs of each plane are ORed. b) The five planes (k-butterflies) of the structure. The symbols ∘, ∘, ∘, • represent respectively 2×2, 2×4, 4×2, 4×4 crossbars.

- the k-th plane of the structure is a k-butterfly (a butterfly network without the initial $k-1$ stages, $k = 1, ..., \log_2 N$); in such a way the network on the first plane is a complete butterfly and the network on the last plane is made by only one column of nodes (see Fig. 1b),

- the output terminals of the nodes of each stage of a k-butterfly are connected with the input terminals of the nodes of the subsequent stage on the plane k and plane $k+1$,

- the nodes are 4×4 crossbars; on the boundary of the structure the nodes have two idle input or output terminals (see Fig. 1b),

- the corresponding outputs of all planes are ORed.

As one can see from Fig. 1, the depth of this network is $\log_2 N$ and its number of nodes is $N(\log_2^2 N + \log_2 N)/2 \quad < \quad N \log_2^2 N$.

3.2 Parallel self-routing algorithm

The distributed self-routing algorithm is very simple. On each k-butterfly it routes in parallel and in forward direction all information to the output ports following the binary representation $(d_1 d_2 \ldots d_n)$ of the destination addresses. The states of the nodes at the stage m are set such that packets are routed to right or left node terminal if $d_m = 0$ or $d_m = 1$ respectively. If two packets

simultaneously ask for the same output terminal, a conflict (2-conflict) occours, this conflict is resolved at node level sending one packet on a plane and the other on another plane. If the node has four inputs (internal nodes), 3-conflicts or 4-conflicts may also occour and, in these cases, some packets are blocked. The network will have high efficiency if the total number of 3-conflicts or 4-conflicts will be negligeable.

The $\log_2 N$ depth of the network and the forward direction of the information flux guarantee that the information wavefront synchronously passes through the network in $\log_2 N$ steps. Thus the network can work in pipeline, infact packets can be presented on the network inputs at each time interval Δt, where Δt is the stage-to-stage propagation time.

4 Simulation and results

The overall behaviour (network+algorithm) has been studied by numerical simulation, in a wide range of N, under the assumption of a) permutation requests and b) random requests (uniform traffic) [13]. Simulations are made by the simple algorithm presented in the previous section and give the values of network efficiency, η, and blocking probability, P_b, versus $\log_2 N$.

A uniform information flux through the network minimizes 3-conflict and 4-conflict occurence. To obtain an acceptable uniformity, the simulation program

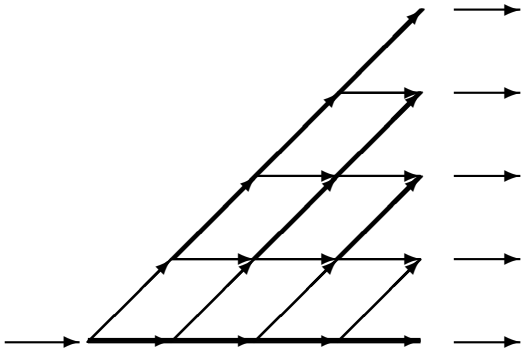

Figure 2: Flux of information through the network planes in the simulations. The thick and thin lines represent respectively the main flux (k-butterfly non-blocked packets) and the secondary flux (k-butterfly blocked packets).

sends information through planes as shown in Fig. 2. In this figure thick lines represent the direction of the main flux − the packets nonblocked by k-butterfly stages − while thin lines represent the information flux which should be blocked by the k-butterfly stages whenever 2-conflicts occour.

4.1 Permutation requests

The simulation program takes as input a set of random chosen permutations. Because of the rapid increase with N of the number of permutations ($N!$) generating very large computation time, a sufficient number of attempts have been executed at every step to obtain sufficiently steady mean values. The result of simulations in the range $N = 2^3$ to $N = 2^{13}$ is shown in Table

N	η_{per}	η_{per}^{extr}	$\Delta\eta_{per}$ (%)
2^2	1.		0.
2^3	1.		0.
2^4	1.		0.
2^5	0.99999		0.001
2^6	0.99992		0.008
2^7	0.99981		0.019
2^8	0.99966		0.034
2^9	0.99951		0.049
2^{10}	0.99934		0.066
2^{11}	0.99920		0.080
2^{12}	0.99907		0.093
2^{13}	0.99890		0.110
2^{14}		0.99878	
2^{15}		0.99863	
2^{16}		0.99848	
2^{17}		0.99834	
2^{18}		0.99819	

Table 1: Network efficiency η_{per} versus N under permutation requests.

1. It is important to examine the network behaviour for large N. Because of the prohibitive computation time involved, simulations have been stopped at $N = 2^{13}$. Several values of efficiency η_{per}^{extr} have been computed from the simulated values by extrapolation, these values are also shown in Table 1. In this Table the deviation from the crossbar efficiency, $\Delta\eta_{per}$, is also presented only for simulated efficiency values. Notice that the greatest P_b value of the simulated cases under permutation requests is $P_b = 0.00110$, corresponding to the efficiency $\eta = 0.99890$ for a network with $N = 2^{13}$ inputs.

4.2 Random requests (uniform traffic)

The simulation program takes as input a set of N random chosen requests; also in this case, to obtain sufficiently steady mean values, a sufficient number of attempts have been executed at every step. The result of simulations in the range $N = 2^3$ to $N = 2^{12}$ is shown in Table 2.

Under random requests several requests can choose the same network output, but just one request can be honoured. For this reason, in the case of random requests, the efficiency values are lower than those under permutation requests. Also the crossbar network, which is strictly nonblocking, and then presents the better efficiency under permutation requests ($\eta^C_{per} = 1$), presents in the case of random requests the efficiency behaviour shown in Table 2 (η^C_{ran}).

Because the efficiency of the presented network, η_{per}, is very close to 1, the values η_{ran} can be computed as a deviation from crossbar efficiency values used as reference. In fact the η_{ran} values can be obtained, in first approximation, by:

$$\eta_{ran} \simeq \eta^{comp}_{ran} = \eta^C_{ran}\eta_{per} \tag{1}$$

where η^C_{ran} is the efficiency of crossbar network under random requests. In Table 2 the behaviour of η^{comp}_{ran} computed by Eq. 1 is also presented. Notice

N	η_{ran}	η_{ran}^{comp}	η_{ran}^{C}	$\Delta\eta_{ran}$ (%)
2^2	0.68367	0.68367	0.68367	0.
2^3	0.65632	0.65632	0.65632	0.
2^4	0.64407	0.64410	0.64409	0.003
2^5	0.63783	0.63785	0.63786	0.005
2^6	0.63492	0.63495	0.63500	0.012
2^7	0.63345	0.63345	0.63357	0.019
2^8	0.63264	0.63260	0.63282	0.028
2^9	0.63226	0.63219	0.63250	0.038
2^{10}	0.63200	0.63187	0.63228	0.044
2^{11}	0.63182	0.63172	0.63222	0.063
2^{12}	0.63176	0.63161	0.63220	0.069
2^{13}		0.63145	0.63215	
2^{14}		0.63136	0.63213	
2^{15}		0.63125	0.63212	
2^{16}		0.63116	0.63212	
2^{17}		0.63107	0.63212	
2^{18}		0.63096	0.63210	

Table 2: Network efficiency η_{ran} versus N under random requests. The η_{ran}^{comp} values are computed by Eq. 1.

that for $N = 2^{12}$ the deviation of network efficiency from crossbar efficiency, $\Delta\eta_{ran}$, is less than 0.1%.

4.3 Internal buffers

A significant performance improvement can be achieved by adding one buffer to each node of the network. With single buffers the network can resolve 3-conflicts and partially 4-conflicts. In this case a little part of information arrives at the destination unsyncronized. Simulations have been performed for the presented network under permutation requests and in pipeline mode for

N	η_{per}^{buf}	η_{ran}^{buf}	$\Delta\eta_{per}^{buf} = \Delta\eta_{ran}^{buf}$ (%)
2^2	1.	0.68367	0.
2^3	1.	0.65632	0.
2^4	1.	0.64409	0.
2^5	0.999994	0.63786	0.0006
2^6	0.999977	0.63500	0.0023
2^7	0.999970	0.63355	0.0030
2^8	0.999966	0.63280	0.0034
2^9	0.999962	0.63247	0.0038
2^{10}	0.999953	0.63225	0.0047
2^{11}	0.999951	0.63219	0.0049
2^{12}	0.999949	0.63216	0.0051

Table 3: Efficiency η_{per}^{buf} and η_{ran}^{buf} versus N of buffered network.

$N = 2^2$ to $N = 2^{12}$ by means of the same self-routing algorithm described in the previous sections. The results in the case of random requests are computed by means of Eq. 1 and are presented in Table 3; as one can see, in this case, the values of the deviation of network efficiencies from crossbar efficiencies are negligeable.

5 Conclusions

The network described in this work has a very simple and modular architecture which gives multiple physical channels for every logical path and, in this way, solves the quasi-totality of packet conflicts. As one can see from Table 1, the network can be considered quasi-nonblocking and for this reason its behaviour under random requests is very close to that of crossbar network (see Table

2). Other important features are: i) the network routes packets by a simple distributed self-routing procedure, ii) the packets pass through the network in a synchronous manner and they can be pipelined, iii) the routing time ($\log_2 N$) is the fastest obtainable on banyan networks, iv) the topological complexity (number of the nodes) is smaller than $N \log_2^2 N$, v) because the molteplicity of paths, the network presents a good fault tolerance degree.

If buffering capability is added, the network efficiency can be enhanced. A very high performance switching fabric suitable for the future broadband integrated services digital networks can be built with the presented architecture. At present the behaviour of this switching fabric is also studied from the standpoint of a precise analysis of its fault tolerance capability.

References

[1] F.A. Tobagi, T.C. Kwok, "The tandem Banyan Switching Fabric: A simple High-Performance Fast Packet Switching", *IEEE INFOCOM*, 1245-1253, 1991.

[2] F.A. Tobagi, "Fast Packet Switch Architectures for Broadband Integrated Services Networks", *Proceedings of the IEEE*, 78, 1, 133-167, 1990.

[3] Chin-Tau Lea, "Multi-$\log_2 N$ Self-Routing Networks and Their Applications in High Speed Electronic and Photonic Switching Systems", *IEEE Trans. on Communication*, 38, 1740-1749, 1990.

[4] C. -T. Lea and D. -J. Shyy, "Tradeoff of Horizontal Decomposition Versus Vertical Stacking in Rearrangeble Nonblocking Networks", *IEEE Trans. on Communication*, 39, 899-904, 1991.

[5] D. -J. Shyy and C. -T. Lea, "$\text{Log}_2(N, m, p)$ Strictly Nonblocking Networks", *IEEE Trans. on Communication*, 39, 1502-1510, 1991.

[6] R. Melen, "A General Class of Rearrangeable Interconnection Networks", *IEEE Trans. on Communication*, 39, 1737-1739, 1991.

[7] T. Leighton, B. Maggs, "Expander Might Be Practical: Fast Algorithms for Routing Around Faults on Multibutterflies" *Proceedings of the 30th annual Symposium on the Foundations of Computer Science*, Research Triangle, North Carolina, 384-389, 1989.

[8] E.T. Brushnell, J.S. Meditch, "Dilated Multistage Interconnection Networks for Fast Packet Switching", *IEEE INFOCOM*, 1264-1273, 1991.

[9] E. Valdimarsson, "Blocking in Multirate Networks", *IEEE INFOCOM*, 579-588, 1991.

[10] K.E. Batcher, "Sorting networks and their application", *Proceeding of Spring Joint Computer Conference*, 307-314, 1968.

[11] D.M. Koppelman, A.Y. Oruç, "A Self-Routing Permutation Network", *Journal of Parallel and Distributed Computing*, 140-151, 1990.

[12] E. Upfal, "An $O(\log N)$ deterministic packet scheme" *Proceeding of the 21st Annual ACM Symposium on the Theory of Computing*, 241-250, 1989.

[13] T. H. Szymanski, V. C. Hamacher, "On the Permutation Capability of Multistage Interconnection Networks", *IEEE Trans. on Computers*, c-36, 7, 1987.

[14] R. Venkatesan, H. T. Mouftah, "Balanced Gamma Network - A New Candidate for Broadband Packet Switch Architectures", *IEEE INFOCOM*, 2482-2488, 1992.

[15] K.Y. Lee, H. Yoon, "The B-networks: A Multistage Interconnection Network with Backward Links", *IEEE Trans. on Computers*, 39, 7, 966-969, 1990.