On Arbitrary Waksman Networks and their Vulnerability

Bruno Beauquier, Eric Darrot

To cite this version:

Bruno Beauquier, Eric Darrot. On Arbitrary Waksman Networks and their Vulnerability. RR-3788, INRIA. 1999. <inria-00072871>

HAL Id: inria-00072871
https://hal.inria.fr/inria-00072871

Submitted on 24 May 2006

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

On Arbitrary Waksman Networks and their Vulnerability

Bruno Beauquier — Eric Darrot

N° 3788

Octobre 1999

THÈME 1

Rapport de recherche

ISSN 0249-6399 ISRN INRIA/RR--3788--FR+ENG

INRIA
SOPHIA ANTIPOLIS

On Arbitrary Waksman Networks and their Vulnerability

Bruno Beauquier * , Eric Darrot *

Thème 1 — Réseaux et systèmes
Projet Sloop

Rapport de recherche n° 3788 — Octobre 1999 — 13 pages

Abstract: Motivated by problems in telecommunication satellites, we investigate rearrangeable permutation networks made of binary switches. A simple counting argument shows that the number of switches necessary to build a $n \times n$ rearrangeable network (i.e. capable of realizing all one-to-one mappings of its n inputs to its n outputs) is at least $\lceil \log_2(n!) \rceil = n \log_2 n - n \log_2 e + o(n)$ as $n \to \infty$. For $n = 2^r$, the r-dimensional Beneš network gives a solution using $n \log_2 n - \frac{n}{2}$ switches. Waksman, and independently Goldstein and Leibholz, improved these networks using $n \log_2 n - n + 1$ switches. We provide an extension of this result to arbitrary values of n, using $\sum_{i=1}^{n} \lceil \log_2(i) \rceil$ switches. The routing algorithm used in Beneš networks is also generalized for arbitrary values of n. Finally the fault-tolerance issue of these networks is discussed.

Key-words: switching networks, multistage networks, rearrangeable networks, permutation networks, fault tolerance, vulnerability.

Submitted to Parallel Processing Letters (World Scientific).

An extended abstract of this paper has been accepted in ALGOTEL'99, Roscoff, France, 5–7 mai 1999.

This work has been supported by Alcatel Space Industries (Toulouse, France).

* SLOOP is a joint project with the CNRS and the University of Nice-Sophia Antipolis (I3S laboratory). E-mail: {Bruno.Beauquier,Eric.Darrot}@inria.fr.

Unité de recherche INRIA Sophia Antipolis
2004, route des Lucioles, B.P. 93, 06902 Sophia Antipolis Cedex (France)
Téléphone : 04 92 38 77 77 - International : +33 4 92 38 77 77 — Fax : 04 92 38 77 65 - International : +33 4 92 38 77 65

Réseaux de Waksman Généralisés et leur Vulnérabilité

Résumé : Motivée par des problèmes de conception de réseaux embarqués dans des satellites de télécommunications, notre étude porte sur les réseaux de permutation réarrangeables composés de commutateurs binaires. Un simple argument de comptage montre que le nombre de commutateurs nécessaires pour construire un réseau réarrangeable $n \times n$ (capable de réaliser toutes les permutations de ses n entrées vers ses n sorties) est au moins $\lceil \log_2(n!) \rceil = n \log_2 n - n \log_2 e + o(n)$, quand $n \to \infty$. Pour $n = 2^r$, le réseau de Beneš de dimension r fournit une solution utilisant $n \log_2 n - \frac{n}{2}$ commutateurs. Waksman, et indépendamment Goldstein et Leibholz, ont amélioré cette construction en utilisant $n \log_2 n - n + 1$ commutateurs. Nous donnons une généralisation de ce résultat pour toute valeur de n, en utilisant $\sum_{i=1}^{n} \lceil \log_2(i) \rceil$ commutateurs. L'algorithme de routage utilisé dans les réseaux de Beneš est également généralisé pour notre construction. Enfin, nous abordons la tolérance aux pannes de ces réseaux.

Mots-clés : commutation, réseaux multi-étages, réseaux de permutation, réarrangeabilité, tolérance aux pannes.

1 Motivation

The following problem arises in the design of on-board networks in telecommunication satellites. The purpose of these satellites is to relay incoming audio or video signals, via determined output amplifiers, according to the geographical areas targeted, the traffic variations and/or the compatibility between signals and amplifiers. Incoming signals are routed to output amplifiers through a network made of switching elements interconnected by waveguides. Each switching element can perform mechanically (by rotation) the two following connections of its 4 ports:

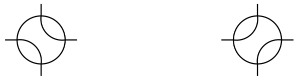

Figure 1: Possible connections (states) of a rotative switch

Furthermore, once on board, amplifiers can fail and switches can lock definitively in one state. In [1, 2] it is assumed that amplifier failures can occur and that any input signal can be routed possibly to any output amplifier. In contrast, we consider in this paper the case where amplifiers do not fail and where each incoming signal needs a specific amplifier.

For many reasons (layout properties, reliability, energy saving, etc), but mainly to decrease launch costs, it is crucial to minimize the network physical weight, i.e. to minimize the number of switches, the number of links (waveguides) and their length. As launch costs are dramatically high, it is worth saving even one switch. In case of switch locking the problem becomes minimizing the number of additional switches needed to guarantee the satellite's purpose, despite a number of possible locked switches predetermined by the expected satellite lifetime. This difficult problem is tackled at the end of this paper in the case of only one locking and will be the subject of a forthcoming paper [3] in the case of an arbitrary number of faults.

A practical way to realize such networks is to refer to the classical multistage permutation networks made of 2 × 2 switching elements (also called *binary switches*), which have been well studied in the literature for both telecommunication and parallel applications. In this model, each switch can be set in a direct or a crossed connection state as follows:

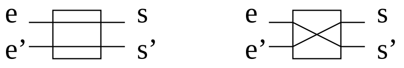

Figure 2: Settings of the binary switch

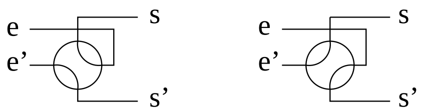

Figure 3: Emulation of a binary switch by a rotative switch

Figure 3 shows that rotative switches of figure 1 and binary switches are functionally equivalent. However, note that the implicit orientation (from left to right) of binary switches as they are used in multistage networks restricts the use of rotative switches regarding their actual capabilities.

The paper is organized as follows. In section 2 we recall some definitions and known results on rearrangeable permutation networks. In section 3 we present a constructive way of building an arbitrary size permutation network and we give the associated routing algorithm. This construction is the best known at this time in terms of switch count and improves previous results as shown by Table 1 (section 2) and by computation (section 4). Finally, in section 5 we extend our results in case of one faulty switch.

2 Rearrangeable Permutation Networks

A *switching network* is an arrangement of switches and transmission links allowing some input terminals (input signals in our case) to be connected simultaneously to some output terminals (the amplifiers) by edge-disjoint paths. Such a network can potentially perform all or only some of the possible connections of its inputs to its outputs, but is usually restricted to one-to-one connections. A switching network with n inputs and n outputs is said to be a *rearrangeable permutation network* if, for any one-to-one mapping π of the set of the inputs on the set of the outputs, there exist a set of edge-disjoint paths connecting the input i to the output $\pi(i)$, for each $1 \leq i \leq n$.

As specified in section 1, we restrict ourselves to *binary permutation networks* constructed solely from binary switches. However, our results can be easily extended to k-ary permutation networks where every switch can realize any mapping of its k inputs to its k outputs.

Since one binary switch has two possible settings, s switches have 2^s settings. However, note that this does not imply that s switches will realize 2^s distinct network mappings, as different settings may produce the same network mapping. But at least $\lceil \log_2 (n!) \rceil$ ($n \log_2 n - 1.443n$ from Stirling's formula) switches are needed to realize all $n!$ possible mappings of a network with n inputs and n outputs.

The first rearrangeable binary permutation networks were designed by Beneš [4, 5, 6] and were based on the Clos 3-stage network [7]. The r-dimensional Beneš network can realize any permutation of its $n = 2^r$ inputs to its $n = 2^r$ outputs on edge-disjoint paths, through $2r - 1$ levels of 2^{r-1} switches, for a total of $n \log_2 n - \frac{n}{2}$ switches. Beneš networks are thus asymptotically optimal in terms of the switch count. The 3-dimensional 8×8 Beneš network is shown in Figure 4.

Further works by different authors [8, 9, 10, 11] have shown that one switch could be spared at each step of the recursive construction of Beneš networks, without affecting its rearrangeability. We call Waksman networks the corresponding optimized networks, having $n \log_2 n - n + 1$ switches, n being a power of two. The case $n = 8$ is shown in Figure 5.

Recently Chang and Melhem generalized Beneš networks for any size n [12] and called them AS-Beneš (arbitrary size) networks. Although they wished to reduce the number of

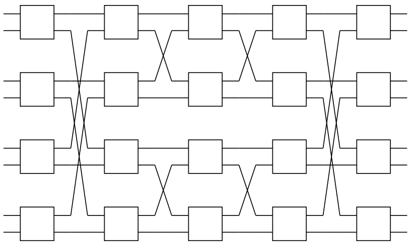

Figure 4: 8 × 8 Beneš network

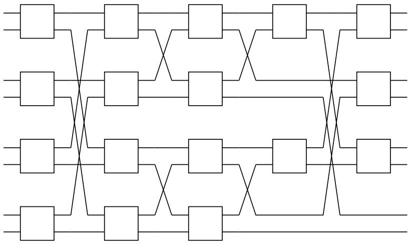

Figure 5: 8 × 8 Waksman network

switches used, their construction does not take into account the possibility of sparing one switch in the even case of the recursion, presented here.

We call AS-Waksman networks our permutation networks. By denoting by $S(n)$ the number of switches used for any number n of terminals, we obtain:

$$S(n) \;=\; S\left(\left\lceil\frac{n}{2}\right\rceil\right)+S\left(\left\lfloor\frac{n}{2}\right\rfloor\right)+n-1 \;=\; \sum_{i=1}^{n}\lceil\log_2(i)\rceil. \tag{1}$$

Note that this result is mentioned in a footnote on page 447 of [10] and is attributed to M. W. Green, no publication being known though. Table 1 presents the values obtained by the different constructions.

3 Network Construction

Three binary switches can be used to construct a 3 × 3 permutation network as shown in Figure 6. This network can be seen as being built from a 2×2 network (a switch) and a 1×1 network (a link). Figure 7 shows the 4 × 4 Waksman network using five binary switches. It can be seen as being built from two 2 × 2 permutation networks.

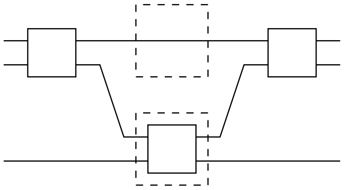

Figure 6: 3 × 3 AS-Waksman network

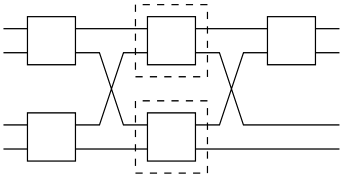

Figure 7: 4 × 4 Waksman network

The procedures used to construct these networks can be generalized to recursively construct a network of any size. Specifically, a $n \times n$ AS-Waksman network is constructed recursively from a $\lceil\frac{n}{2}\rceil \times \lceil\frac{n}{2}\rceil$ AS-Waksman network and a $\lfloor\frac{n}{2}\rfloor \times \lfloor\frac{n}{2}\rfloor$ AS-Waksman network.

n	*Lower bound*	*Beneš* [4]	*AS-Beneš* [12]	*Waksman* [11]	*AS-Waksman*
2	1	1	1	1	1
3	3		3		3
4	5	6	6	5	5
5	7		8		8
6	10		12		11
7	13		15		14
8	16	20	20	17	17
9	19		22		21
10	22		26		25
11	26		30		29
12	29		36		33
13	33		39		37
14	37		44		41
15	41		49		45
16	45	56	56	49	49
32	118	144	144	129	129

Table 1: Comparative switch count.

When n is even, the construction is similar to that of Waksman. The n inputs are connected to $\frac{n}{2}$ switches and each switch is connected to two $\frac{n}{2} \times \frac{n}{2}$ AS-Waksman networks. Any $n-2$ outputs are connected to $\frac{n}{2}-1$ switches and each of them is connected to the two AS-Waksman subnetworks. The last two outputs are connected directly to the two subnetworks as shown in Figure 8(a).

To construct an AS-Waksman network when n is odd, any $n-1$ inputs are connected to $\lfloor \frac{n}{2} \rfloor$ switches and each switch is connected to a $\lfloor \frac{n}{2} \rfloor \times \lfloor \frac{n}{2} \rfloor$ AS-Waksman and a $\lceil \frac{n}{2} \rceil \times \lceil \frac{n}{2} \rceil$ AS-Waksman. Similarly, any $n-1$ outputs are connected to $\lfloor \frac{n}{2} \rfloor$ switches and each switch is connected to the two AS-Waksman subnetworks. The last input and the last output are connected directly to the $\lceil \frac{n}{2} \rceil \times \lceil \frac{n}{2} \rceil$ AS-Waksman subnetwork as shown in Figure 8(b).

This recursive process is illustrated in Figure 9 where a 9×9 AS-Waksman is built from a 4×4 AS-Waksman and a 5×5 AS-Waksman. In general, a $n \times n$ AS-Waksman network may be constructed in this way, for any n.

Remark. Notice that our connection rules allow many possible constructions, according to which and how inputs/outputs are connected, and thus they define actually a family of

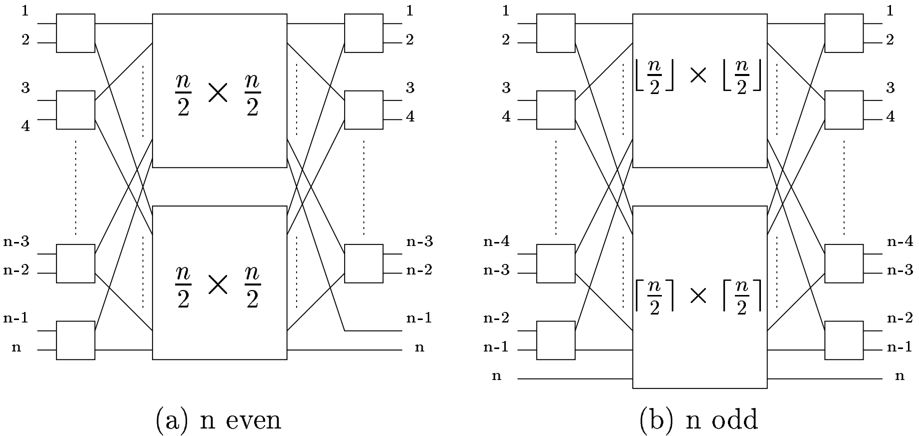
1
2
3
4
n-3
n-2
n-1
n
$\frac{n}{2} \times \frac{n}{2}$
$\frac{n}{2} \times \frac{n}{2}$
1
2
3
4
n-3
n-2
n-1
n
(a) n even
1
2
3
4
n-4
n-3
n-2
n-1
n
$\lfloor \frac{n}{2} \rfloor \times \lfloor \frac{n}{2} \rfloor$
$\lceil \frac{n}{2} \rceil \times \lceil \frac{n}{2} \rceil$
1
2
3
4
n-4
n-3
n-2
n-1
n
(b) n odd

Figure 8: General construction of AS-Waksman networks

AS-Waksman networks. Nevertheless, in the figures and for the main proof, we suppose for convenience that inputs $2k-1$ and $2k$ are connected to the same switch for $1 \leq k \leq \frac{n}{2}$, as well as outputs $2l-1$ and $2l$ for $1 \leq l \leq \frac{n-1}{2}$, while the remaining connections are direct as shown in Figure 8.

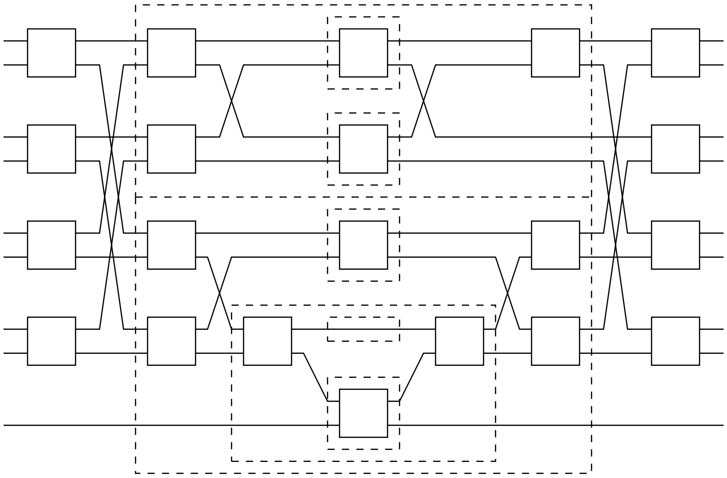

Figure 9: A 9×9 AS-Waksman network

Theorem 1 *Given any one-to-one mapping π of n inputs to n outputs in a $n \times n$ AS-Waksman network, there is a set of edge-disjoint paths from the inputs to the outputs connecting input i to output $\pi(i)$ for $1 \leq i \leq n$.*

Proof. The proof is by induction on n. If $n = 1$ or $n = 2$, the AS-Waksman network consists of a single link or a single binary switch respectively and the result is obvious. Hence, we assume that the result is true for the AS-Waksman networks of sizes $\lfloor \frac{n}{2} \rfloor$ and $\lceil \frac{n}{2} \rceil$. The key of the induction is to observe that the middle part of an AS-Waksman network comprises two AS-Waksman subnetworks. Hence, it will be sufficient to decide whether each path is to be routed through the upper or the lower subnetwork.

The only constraints that we have on whether paths use the upper or lower subnetworks are that paths from inputs $2k-1$ and $2k$ must use different subnetworks for $1 \leq k \leq n/2$, as well as paths to outputs $2l-1$ and $2l$ for $1 \leq l \leq (n-1)/2$. This is because each switch on the first and last levels of the AS-Waksman network has precisely one connection to each of the upper and lower subnetworks. There is no choice when either the input or the output is connected directly to a subnetwork.

The routing problem to realize any given permutation π can be reduced to a bipartite edge coloring problem as follows. If n is even then let $n = 2p$ and if n is odd then let $n = 2p-1$. Consider the bipartite multi-graph $G_\pi = (V, E)$ with vertex set $V = \{u_k, v_k\}_{1 \leq k \leq p}$ and with edge $[u_k, v_l] \in E$ if there are $i \in \{2k-1, 2k\}$ and $j \in \{2l-1, 2l\}$ such that $\pi(i) = j$. From this definition, it follows that G_π has maximum degree 2. It is a classical result from graph theory (see e.g. [13]), known as König-Hall's theorem, that the edges of a bipartite multi-graph of maximum degree Δ can be colored using exactly Δ colors, so that adjacent edges are assigned different colors.

Now we assume that the edge coloring problem is solved for G_π using two colors and we show how this leads to a routing of the permutation π in the $n \times n$ AS-Waksman network. Let color 1 be the color assigned to the edge $[u_{\lceil \pi^{-1}(n)/2 \rceil}, v_p]$. Note that the path connecting input $\pi^{-1}(n)$ to output n must use the lower subnetwork. Thus, for each i we decide to route the path from input i to output $\pi(i)$ through the lower subnetwork if the edge $[u_{\lceil i/2 \rceil}, v_{\lceil \pi(i)/2 \rceil}]$ has color 1 and through the upper subnetwork if it has color 2. In case of n odd, another constraint is that the path connecting input n to output $\pi(n)$ must use the lower subnetwork. However, the edge $[u_p, v_{\lceil \pi(n)/2 \rceil}]$ has necessarily color 1, due to a parity argument.

In this manner, all paths are assigned the upper or lower subnetworks without conflict, i.e. we can set the switches at the first and last levels of the AS-Waksman network so that both ends of every path are connected to the same subnetwork. The remainder of the path routing and switch setting is handled by induction in the subnetworks. Hence, we have established the inductive hypothesis, thereby proving the theorem. □

As an example, we illustrate the routing algorithm in Figure 10 for the mapping

$$\pi = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 8 & 7 & 6 & 9 & 1 & 4 & 3 & 2 & 5 \end{pmatrix}$$

in a 9×9 AS-Waksman network, color 1 being black. The associated bipartite multi-graph G_π is drawn on the left and the edge coloring induces the subnetwork assignment for each connection between i and $\pi(i)$. The routing inside each subnetwork can be obtained recursively by the same procedure.

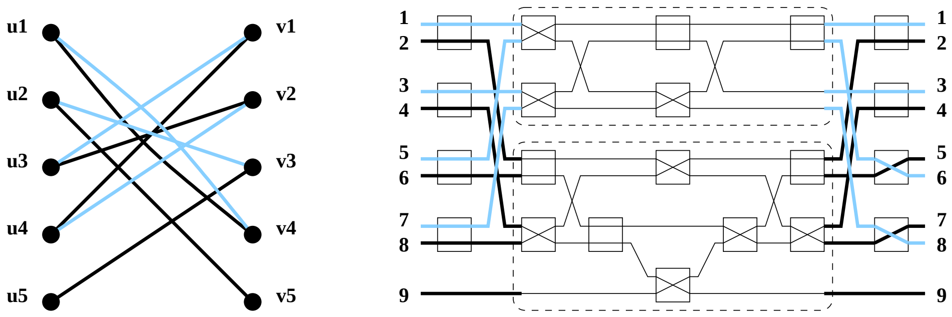

Figure 10: Routing a permutation in a 9×9 AS-Waksman by bipartite edge coloring

4 Switch Count

In this section, we compute $S(n)$, the number of binary switches used for a $n \times n$ AS-Waksman network.

Theorem 2 *For any $n \geq 1$, the number of binary switches used for the rearrangeable $n \times n$ AS-Waksman permutation network is $S(n) = \sum_{i=1}^{n} \lceil \log_2(i) \rceil$.*

Proof. The proof is by induction on n. As $S(1) = 0$, the result is true for $n = 1$. From the recursive construction given in section 3, we have, for any $n \geq 2$:

$$S(n) \;=\; S\left(\left\lceil \frac{n}{2} \right\rceil\right) + S\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + n - 1. \tag{2}$$

Note that for any two positive integers i and r, if we have $2^{r-1} < i \leq 2^r$, then we have $2^r < 2i - 1 < 2i \leq 2^{r+1}$, hence $\lceil \log_2(2i-1) \rceil = \lceil \log_2(2i) \rceil = \lceil \log_2(i) \rceil + 1$. Thus, recalling that $\log_2(1) = 0$, we have:

$$\begin{aligned} S(n) &= \sum_{i=1}^{\lceil n/2 \rceil} \lceil \log_2(i) \rceil + \sum_{i=1}^{\lfloor n/2 \rfloor} \lceil \log_2(i) \rceil + n - 1 && (3) \\ &= \sum_{i=2}^{\lceil n/2 \rceil} (\lceil \log_2(i) \rceil + 1) + \sum_{i=1}^{\lfloor n/2 \rfloor} (\lceil \log_2(i) \rceil + 1) && (4) \\ &= \sum_{i=2}^{\lceil n/2 \rceil} \lceil \log_2(2i-1) \rceil + \sum_{i=1}^{\lfloor n/2 \rfloor} \lceil \log_2(2i) \rceil = \sum_{j=1}^{n} \lceil \log_2(j) \rceil. && (5) \end{aligned}$$

The inductive hypothesis being satisfied, the proposition holds. □

Corollary 3 *For any $n \geq 1$, $S(n) \leq n \log_2 n - 0.91n + 1$.*

Indeed, a different computation approach can give a more expressive value: $S(n) = n \log_2 n - n(2^{\lambda} - \lambda) + 1$, where $\lambda = \lceil \log_2 (n) \rceil - \log_2 n$. This allows to estimate the worst case obtained for $\lambda = -\log_2 (\log_2 e) \sim 0.53$. Note that when n is a power of two, it can be computed by induction that $S(n) = n \log_2 n - n + 1$, as already obtained by Waksman [11].

5 Fault-Tolerance Issue

Now, we consider the possibility for switches to lock and to remain definitively in the same state. A switching network with n inputs/outputs is said to be a *k-tolerant permutation network* if, for any one-to-one mapping π of the inputs on the outputs and for any k switches locked each in some state, there is a set of edge-disjoint paths connecting the input i to the output $\pi(i)$, for each $1 \leq i \leq n$. We denote by $N(n, k)$ the minimum number of switches for such a network. In this paper, we restrict ourselves to 1-tolerant permutation networks.

Theorem 4 *For any $n \geq 1$, $N(2n, 1) \leq 2N(n, 1) + 2n$.*

Proof. The proof is based on the construction illustrated in Figure 11. Both the $2n$ inputs and the $2n$ outputs are connected two by two to $2n$ switches and each switch is connected to two 1-tolerant permutation networks of size n. We claim that this $2n \times 2n$ network, made of two $n \times n$ 1-tolerant networks using each $N(n, 1)$ switches, is a 1-tolerant permutation network.

Actually, we have a stronger result: any permutation can be realized despite one faulty switch in each subnetwork, plus one faulty switch at the first or last level. Indeed, in this case each subnetwork can still handle by assumption any mapping of its n inputs to its n outputs. Moreover, one switch locking at the first or last level corresponds to have the two associated inputs or outputs directly connected to the two subnetworks, similarly to the construction of even AS-Waksman networks. Thus, the arguments used in the proof of Theorem 1 can be easily adapted to decide through which subnetwork each path is to be routed, thereby proving our claim. □

Unfortunately, no similar good construction is known for the odd case. Note also that the construction for the even case is not necessarily the best possible, in particular for $n = 4$. The last result is thus interesting only to build a twice larger 1-tolerant permutation network when a good one is already known. For this purpose, we present now in Figures 12, 13, 14 and 15 some 1-tolerant networks obtained by hand for $n =$ 3, 4, 5 and 6 respectively. For space reasons, proofs are omitted but can be done by restricted case analysis. These constructions together with the proof of Theorem 4 give rise to the summary of results presented in Table 2.

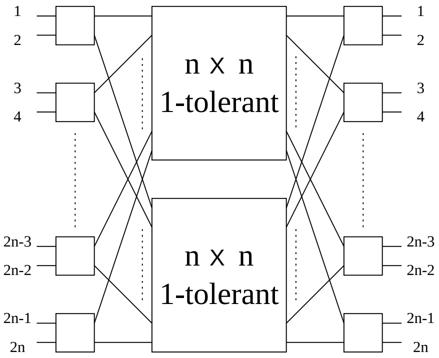

Figure 11: Construction of even 1-tolerant permutation networks

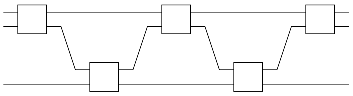

Figure 12: 1-tolerant 3 × 3 network

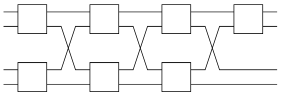

Figure 13: 1-tolerant 4 × 4 network

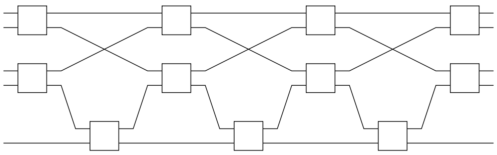

Figure 14: 1-tolerant 5 × 5 network

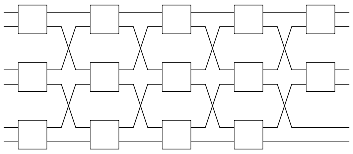

Figure 15: 1-tolerant 6 × 6 network

Size	2	3	4	5	6	8	10	12	16
Switches	2	5	7	11	14	22	32	40	50

Table 2: Number of switches for 1-tolerant permutation networks.

6 Conclusion

This paper has provided a simple way for building rearrangeable permutation networks of arbitrary size n using $\sum_{i=1}^{n} \lceil \log_2 (i) \rceil$ binary switches. An efficient algorithm for routing any permutation in such networks has been given. The fault-tolerance issue has also been tackled in the case of one locked switch. It is worth pointing out that all these results can easily be adapted for rearrangeable networks using p-ary switching elements.

In a forthcoming paper [3], the fault-tolerance results will be extended by presenting constructive ways for building k-tolerant permutation networks, for any number k of switch faults.

Acknowledgements

We thank Alcatel Space Industries for financial support and especially B. Vaillant who introduced us the problem. We are also grateful to Jean-Claude Bermond, Pavol Hell and Stéphane Pérennes for their helpful remarks and corrections.

References

[1] J-C. Bermond, E. Darrot and O. Delmas, Design of fault-tolerant on-board networks in satellites, Manuscript.

[2] S. Pérennes and D. Tóth, Design of fault-tolerant flow networks, Manuscript.

[3] B. Beauquier and E. Darrot, Rearrangeable switching networks tolerant to switch locks, Manuscript (title may change).

[4] V. Beneš, Permutation groups, complexes, and rearrangeable multistage connecting networks, *Bell Sys. Tech. Journal* **43** (1964) 1619–1640.

[5] V. Beneš, *Mathematical Theory of Connecting Networks and Telephone Traffic*, Academic Press, New York (1965).

[6] V. Beneš, Applications of group theory to connecting networks, *Bell Sys. Tech. Journal* **54** (1975) 407–420.

[7] C. Clos, A study of non-blocking switching networks, *Bell Sys. Tech. Journal* **32** (1953) 406–424.

[8] L. Goldstein and S. Leibholz, On the synthesis of signal switching networks with transient blocking, *IEEE Transactions on Electronic Computers* **16** (1967) 637–641.

[9] A. Joel, On permutation switching networks, *Bell Sys. Tech. Journal* **47** (1968) 813–822.

[10] W. Kautz, K. Levitt and A. Waksman, Cellular interconnection arrays, *IEEE Transactions on Computers* **17** (No.5, May 1968) 443–451.

[11] A. Waksman, A permutation network, *Journal of the ACM* **15** (1969) 159–163.

[12] C. Chang and R. Melhem, Arbitrary size Beneš networks, *PPL* **7** (No. 3, Sep. 1997) 279–284.

[13] C. Berge, *Graphs*, North-Holland (1985).

Contents

Unité de recherche INRIA Sophia Antipolis
2004, route des Lucioles - B.P. 93 - 06902 Sophia Antipolis Cedex (France)

Unité de recherche INRIA Lorraine : Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - B.P. 101 - 54602 Villers lès Nancy Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38330 Montbonnot St Martin (France)
Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - B.P. 105 - 78153 Le Chesnay Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, B.P. 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399