The Benes network is $\frac{q(q-1)}{2n}$ almost q-set-wise independent*

Efraim Gelman[1] and Amnon Ta-Shma[2]

1 **Tel-Aviv University**
Tel-Aviv, Israel
efigelman@gmail.com.
2 **Tel-Aviv University**
Tel-Aviv, Israel
amnon@tau.ac.il

Abstract

A switching network of depth d is a layered graph with d layers and n vertices in each layer. The edges of the switching network do not cross between layers and in each layer the edges form a partial matching. A switching network defines a stochastic process over $\mathbb{S}_n$ that starts with the identity permutation and goes through the layers of the network from first to last, where for each layer and each pair (i, j) in the partial matching of the layer, it applies the transposition (ij) with probability half. A switching network is *good* if the final distribution is close to the uniform distribution over $\mathbb{S}_n$.

A switching network is ε-almost q-permutation-wise independent if its action on any ordered set of size q is almost uniform, and is ε-almost q-set-wise independent if its action on any set of size q is almost uniform. Mixing of switching networks (even for q-permutation-wise and q-set-wise independence) has found several applications, mostly in cryptography. Some applications further require some additional properties from the network, e.g., the existence of an algorithm that given a permutation can set the switches such that the network generates the given permutation, a property that the Benes network has.

Morris, Rogaway and Stegers showed the Thorp shuffle (which corresponds to applying two or more butterflies one after the other) is q-permutation-wise independent, for $q = n^\gamma$ for γ that depends on the number of sequential applications of the butterfly network. The techniques applied by Morris et al. do not seem to apply for the Benes network.

In this work we show the Benes network is almost q-set-wise independent for q up to about $\sqrt{n}$. Our technique is simple and completely new, and we believe carries hope for getting even better results in the future.

1998 ACM Subject Classification C.2 COMPUTER-COMMUNICATION NETWORKS

Keywords and phrases switching network, mixing, Benes

Digital Object Identifier 10.4230/LIPIcs.xxx.yyy.p

1 Introduction

The subject of this paper is the generation of a random permutation through the composition of (few) simple building blocks. The question is quite old and appears in many different contexts, one prominent example is the application to CCA-security in cryptography. Generating an almost random permutation on $\log n$-bit strings in such a way, ensures CCA-security to almost 2^n queries (see for example [14]). The problem comes in two main flavors:

* Supported by the Israel science Foundation (grants no. 1090/10 and 994/14) and by the United States- Israel Binational Science Foundation grant no. 2010120.

Conference title on which this volume is based on.
Editors: Billy Editor and Bill Editors; pp. 1–12

Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Mixing with simple permutations : Here, we have a small subset of (usually simple) permutations, and we ask how fast the process of composing random elements from the subset converges to the uniform distribution over $\mathbb{S}_n$. Some typical examples are, e.g., when:

- The subset that contains all transpositions,
- The subset that contains all permutations whose cycle decomposition is $2^{n/2}$, i.e., they are a product of $n/2$ disjoint transpositions, and,
- The subset that contains all permutations in some conjugacy class of $\mathbb{S}_n$.

A seminal result of Diaconis and Shahshahani [6] analyzes the case where the conjugacy class is the set of all transpositions, and shows that the mixing time is about $\frac{1}{2}n \log n + \Theta(n)$ steps. This result was later extended to the conjugacy class of r-cycles [11]. The technique uses non-commutative Fourier transform and combinatorics of Young Tableaux, and has been used with great success in many works.

switching networks : A switching network of depth d is a layered graph with d layers and n vertices in each layer. The edges of the switching network do not cross between layers and in each layer the edges form a partial matching. A switching network defines a stochastic process over $\mathbb{S}_n$ that starts with the identity permutation and goes through the layers of the network from first to last, where for each layer and each pair (i, j) in the partial matching of the layer, it applies the transposition (ij) with probability half. A switching network is *good* if the final distribution is close to the uniform distribution over $\mathbb{S}_n$.

One may view the switching network problem as a *derandomized* version of mixing with simple permutations. For example, Diaconis and Shahshahani showed that composing $d = \frac{1}{2}n \log n + \Theta(n)$ *random* transpositions is sufficient and necessary for getting a (close to) uniform permutation. However, a switching network would show a specific sequence $\sigma_1, \ldots, \sigma_t$, such that picking $i_1, \ldots, i_t \in \{0, 1\}$ at random and applying $\sigma_t^{i_t} \ldots \sigma_1^{i_1}$ would generate a close to uniform distribution over $\mathbb{S}_n$.

Similarly, the *matching-exchange* process goes as follows. A random perfect matching on the n elements is chosen uniformly at random, and for each pair (i, j) in the matching, the transposition (i, j) is applied with probability half. One way to analyze this process is by choosing k (the number of transpositions) between 0 and $n/2$ according to the binomial distribution, and then choosing a random permutation from the conjugacy class $2^k 1^{n-2k}$. Using non-commutative Fourier analysis and character estimates [12] proved the process converges to uniform in $O(\log n)$ time. Another proof of the same result, but with inferior constants, was obtained using delayed path coupling [4].

Building a switching network would show a specific sequence of perfect matchings that can be used in the matching-exchange process. Thus, constructing a switching network essentially amounts to de-randomizing the matching-exchange process, thus limiting the use of randomness to the bare minimum needed for generating a distribution close to uniform over $\mathbb{S}_n$.

Moreover, it turns out in some important applications (e.g., CCA security) it is crucial to have the second variant (of a mixing network) rather than the first one (of mixing with simple permutations). Another application is for the security of electronic voting and the efficient zero knowledge proofs for the decryption of the encrypted votes (see for example: [2, 1] and [10]). In this application both variants may be used.

However, the question whether good shallow switching networks exist is still wide open. In 1981, David Chaum [3], suggested a cryptographic protocol that guarantees anonymous communication, provided that shallow switching networks exist. In 1993, Rackoff and Simon [15] claimed an explicit switching network of depth $poly(\log n)$ is good. Rackoff and Simon gave only a short sketch of the proof while the full proof of the theorem, which had been delayed to the journal version of the paper,

was never published. In 1999, Czumaj et al. [4] claimed the existence of a good switching network of depth $O(\log^2 n)$, however, the correctness proof was deferred to the full version of the paper and has not appeared to date. Finally, it has been proven in [5] using delayed path coupling that:

▶ **Theorem 1.** *[5] There exists an explicit construction of a good switching network of depth* $polylog(n)$.[1]

Morris [13] constructed, using totally different techniques, an explicit switching network of depth $O(\log^4 n)$ that builds upon Thorp shuffle. We expand on this shortly. Still, in our current state of knowledge, it is not known whether good switching networks of *logarithmic* depth exist.

1.1 q-wise independence

One way to visualize the action of a switching network is as follows. Assume we have a switching network with n vertices and d layers. Put n numbered balls on the n vertices of the first layer. Go through the layers from first to last, and for each layer, and each pair (i, j) in the partial matching of the layer, with probability half switch the balls that are currently at positions i and j. A network is good if the distribution of the n numbered balls at the last layer is close to uniform.

One may weaken this definition by considering the action of the network on only $q \leqslant n$ numbered balls. Further weakening is considering the case where the balls are *identical* balls.

- We say a network is ε-almost *q-permutation-wise* independent if for any possible way of putting q numbered balls, the distribution of the balls at the last layer is ε-close to the uniform distribution on all the $\frac{n!}{(n-q)!}$ possible end configurations.
- We say a network is ε-almost *q-set-wise* independent if for any possible way of putting q identical balls, the distribution of the balls at the last layer is ε-close to the uniform distribution on all the $\binom{n}{q}$ possible end configurations.

A network that is q permutation–wise independent is in particular q set–wise independent.

Mathematically, the q-wise independence problem is natural. The *permutation-wise* problem compares the action of the network and the action of $\mathbb{S}_n$ on the set $X = [n]_q$ of all *ordered* subsets of size q, and the *set-wise* problem does the same for the set $X = \binom{[n]}{q}$ of all subsets of size q. The q-wise question was extensively studied, e.g., by Gowers [7] and later by Hoory et al. [8], who show, using canonical paths, that composing few random simple permutations from a certain fixed family is ε-almost q-wise. Morris, Rogaway and Stegers [14] showed the Thorp shuffle, that we soon discuss, is almost q permutation–wise independent for a large q, and we soon expand on their result.

1.2 The Thorp shuffle and the butterfly network

We first introduce some notation. For $0 \leq x < n$ let $x = x_{\log n}, \ldots, x_1$ be its binary representation. Let $\underline{x}$ (resp. $\overline{x}$) be the integer that has the same binary representation as x except that its most significant bit is 0 (resp. 1) independent of the most significant bit of x. I.e.,

$$\underline{x} = 0, x_{\log n-1}, \ldots, x_1$$
$$\overline{x} = 1, x_{\log n-1}, \ldots, x_1$$

For example, if $n = 16$, $x = 11$ then $\underline{x} = 3$ and $\overline{x} = 11$.

Now we introduce the Thorp shuffle. At each stage of the Thorp shuffle we do the following:

- For all $0 \leq x < n/2$, with probability half we switch the elements in cells number $\underline{x}$ and $\overline{x}$ (and keep them in place with probability half).

[1] It seems that the constant in the exponent is currently at least 7.

- We permute the elements as follows: an element that is at cell number $x = x_{\log n}, \ldots, x_1$ is moved to cell number $x_{\log n - 1}, \ldots, x_1, x_{\log n}$. Notice that this forms a permutation over the n elements.

This process is equivalent to the butterfly switching network. In the butterfly switching network we have $\log n$ layers with n vertices in each layer. We index the n elements by their binary representation in $\{0,1\}^{\log n}$. The edges of the i'th layer, for $i = 1, \ldots, \log n$, connect $x, y \in \{0,1\}^{\log n}$ if and only if they differ only in the i'th coordinate. The Thorp shuffle for $T = r \log n$ stages is equivalent to applying r butterfly switching networks one after the other.

One way to visualize what the butterfly switching network does is as follows. Take the $(\log n)$ dimensional cube with its usual edges, i.e. $x, y \in \{0,1\}^{\log n}$ are neighbors if and only if they differ only in one coordinate. For every i and edge $(x, x + e_i)$ color the edge with color i. As before, put n numbered balls on the n vertices of the cube, go sequentially over $i = 1, \ldots, \log n$, and for each edge (x, y) colored i, switch the balls on x, y with probability half.

The butterfly switching network (or equivalently the thorp shuffle with $\log n$ stages) induces a probability distribution over permutations of $\mathbb{S}_n$. What can we say about it?

Clearly, the butterfly network is 1-wise (set and permutation) *perfect*, i.e. if we put *one* ball at any starting vertex $x \in \{0,1\}^{\log n}$ and apply the network, the ball will have the same $(\frac{1}{n})$ probability to finish at any vertex. However, if we look at two balls, things are not that good. Suppose we start with two balls on vertices x_1, x_2 that are connected by an edge colored 1, i.e., $x_2 = x_1 + e_1$. Notice that after stage 1 the two balls have different first coordinate, and this does not change later on. Thus, the two balls must end up at vertices that differ in their first bit. A random permutation, however, would do that only with probability half. Thus, the induced distribution over $\mathbb{S}_n$ is far from uniform (set and permutation wise).

In [14] , Morris et al. analyze the Thorp shuffle. They show using a clever coupling argument, that applying the Thorp shuffle $T = 2r \log n$ stages, is $\frac{q}{r+1} \cdot (\frac{4q \log n}{n})^r$-almost q-wise-permutation independent. For example running the butterfly twice (i.e., $r = 1$) we get $\varepsilon = \frac{2q^2 \log n}{n}$ and we may have q almost as large as $\sqrt{n}$.

1.3 The Benes network

Another classical switching network is the Benes network. The Benes network is applying the butterfly twice, first in its original order, and then in reversed order. Thus, the Benes network is similar to the Thorp shuffle in that it repeatedly uses the butterfly network, but it differs from it in the way it orders the layers. An example of the Benes network on eight vertices is given in Figure 1.

It is well known that the Benes network gives positive probability to each permutation $\pi \in \mathbb{S}_n$, see., e.g., [9, Section 3.2]. Other than that, not much was known about the distribution induced over $\mathbb{S}_n$ by the Benes network. The techniques of Morris et al. do not seem to work for the Benes network.[2]

The Benes network is mathematically very natural (as it makes the operator Hermitian, see Sec 2.1). It also induces a clean recursive structure, i.e., a Benes network on n elements is composed of simple first and last layers, and two parallel Benes networks on $n/2$ elements, see Section 2.1 and Figure 1. This clean recursive structure is also behind the proof that every permutation may be obtained by the Benes network.

---

[2] One key ingredient in the proof is the fact that using the update rule described in the paper, for time $t \geqslant \log n - 1$, the probability that any two elements are adjacent at time t is $\leqslant 2^{1-\log n}$. This does not hold for the Benes network since with this update rule the probability can only be bounded by $\leqslant 2^{1-0.5 \log n}$.

Also, the Benes network appears in many applications, and often the reason is exactly this clean structure it possesses. For example, Abe [2] uses the Benes network for mixnets (and electronic voting) and a key property that is required is the ability to easily route any permutation on the network, a property that, as far as we know, the thorp shuffle lacks.

There are therefore two main reasons to study the Benes network: the first is that it appears in protocols that require some specific property of it. The second is that it is mathematically elegant, and the hope that one may be able to use its elegant recursive structure to finally give a construction of a good logarithmic-depth switching network. In particular, it is possible that applying sequentially a constant number of Benes networks, defines a distribution that is close to uniform over $\mathbb{S}_n$.

In this work we use the elegant structure of the network and prove that the Benes network is ε-almost q-set-wise independent, for q up to about $\sqrt{n}$. Specifically, we prove:

▶ **Theorem 2.** *The Benes network is $\frac{q(q-1)}{2n}$-almost q-set-wise independent.*

Parameter-wise, the result we obtain is slightly better than the one obtained in [14] for the corresponding Thorp shuffle (with $2\log n$ stages). On the downside, [14] show q-**permutation**-wise independent and also show that applying a series of Thorp shuffle sequentially significantly reduces the error, while we do not show the corresponding fact for the Benes network.

More importantly, we believe the proof technique, that is completely different than the one in [14], is of independent interest, and reveals the delicate and beautiful structure of the Benes network. We give a recursive formula for the probability the Benes moves a given set of q elements to another, and we identify the crucial parameters on which it depends. The question then reduces to a combinatorial question, that can be solved by analyzing two related experiments. Using this we prove that any set of q elements is obtained with probability at least $\frac{q!}{n^q}$ (and notice how close this is to the uniform probability of $\frac{1}{\binom{n}{q}}$) which is a strong and surprising result by itself. We believe there might be a way to tighten the analysis given in the paper and achieve much better results (e.g., proving q above $\sqrt{n}$). We view this work as a first step of understanding the Benes switching network and believe it sheds light on the way it operates and we hope it may possibly lead to solving the challenging problem of constructing a good switching network of logarithmic depth.

2 Definitions and Notation

For a set S, $\binom{S}{q}$ is the set of all q-subsets of S, i.e., $\{A \subseteq S \mid |A| = q\}$. We let $[n]$ denote the set $\{0,\ldots,n-1\}$. The symmetric group $\mathbb{S}_n$ acts on $\binom{[n]}{q}$ in a natural way, $\pi(A) = \{\pi(a) \mid a \in A\}$. Clearly the action is transitive.

Suppose $\mathbb{S}_n$ acts on a set X. We define π_X to be the $|X| \times |X|$ matrix, where $(\pi_X)_{i,j} = \delta_{\pi(x_j),x_i}$. We specify two special cases:

- When $X = \mathbb{S}_n$ and the action is by left multiplication, we denote the matrix π_X by π.
- When $X = \binom{[n]}{q}$ and the action is the natural action defined above, we denote the matrix π_X by π_q.

For a distribution $\mathfrak{D}$ over $\mathbb{S}_n$ we let D_X denote the $|X| \times |X|$ matrix

$$D_X = \sum_{\pi \in \mathbb{S}_n} \mathfrak{D}(\pi) \cdot \pi_X. \tag{1}$$

If H is a subset of $\mathbb{S}_n$, we identify it with the flat distribution over H, and let H_X be the corresponding matrix.

Notice that $(D_X)_{i,j} = \Pr_{\pi \in \mathfrak{D}}[\,\pi(x_j) = x_i\,]$, i.e., (D_X) describes the transition matrix of the stochastic process that picks π according to the distribution $\mathfrak{D}$ and applies it on x (or a distribution over X).

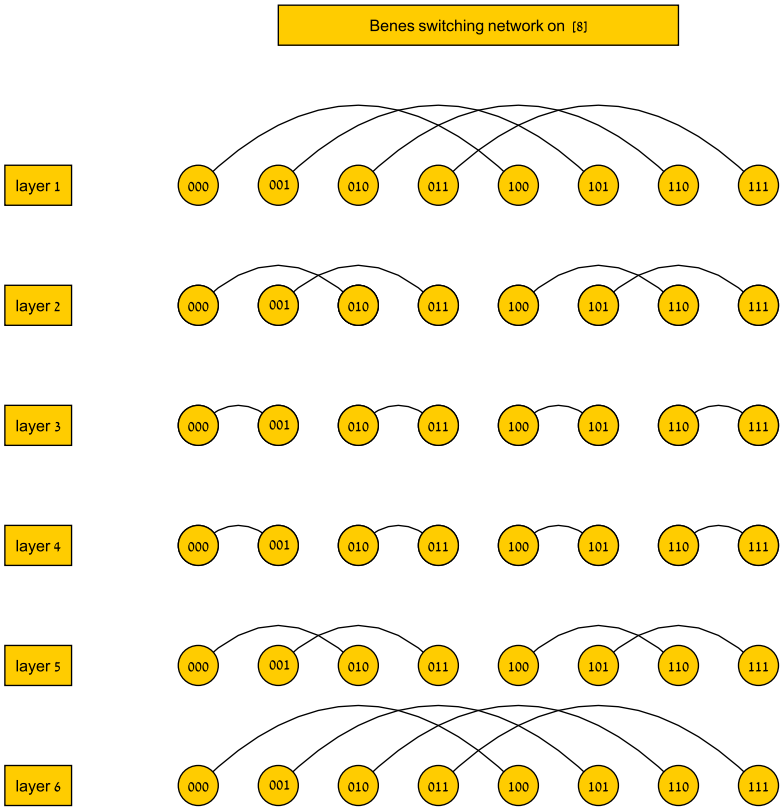

Figure 1 The Benes switching network on eight vertices. The nodes are the elements of $\{0,\ldots,7\}$ in binary representation. Notice that if we look at layers 2-5 and omit the most significant bit, we get the Benes network on four vertices at the left half and right half of the network.

2.1 The Benes network

Given $x, y \in [n]$, we say $x \overset{i}{\sim} y$ if their binary representation differs only on bit number i (starting with the least significant bit). For example $13 \overset{2}{\sim} 15$ since their binary representations are 1101 and 1111.

A Benes network is a layered graph with $2\log n$ layers, indexed from 1 to $2\log n$ and n vertices in each layer. In layers i and $2\log n + 1 - i$ (for $i = 1,\ldots,\log n$), the matching is formed by all the edges (x, y) s.t. $x \overset{\log n+1-i}{\sim} y$.

Alternatively, let H^i (for $i = 1, 2, \ldots, \log n$) be the abelian subgroup of $\mathbb{S}_n$ generated by the set of $n/2$ disjoint transpositions

$$\left\{(x \quad y) \mid x \overset{i}{\sim} y\right\}.$$

A Benes network is a series $H^{\log n},\ldots,H^1,H^1,\ldots,H^{\log n}$ indexed by n.

A Benes network defines a distribution $\mathfrak{B}^{(n)}$ on $\mathbb{S}_n$ as follows: pick $\pi_{\log n} \in H^{\log n},\ldots,\pi_1 \in H^1, \sigma_1 \in H^1,\ldots,\sigma_{\log n} \in H^{\log n}$ and output $\sigma_{\log n}\cdot\ldots\sigma_1\cdot\pi_1\cdot\ldots\pi_{\log n} \in \mathbb{S}_n$. We denote

$$B_X^{(n)} \quad = \quad \Sigma_{\pi\in\mathbb{S}_n}\mathfrak{B}^{(n)}(\pi)\cdot\pi_X. \tag{2}$$

As before, when $X = \mathbb{S}_n$ and the action is by left multiplication, we denote $B_X^{(n)}$ simply by $B^{(n)}$, and if $X = \binom{[n]}{q}$ and the action is the natural action, we denote $B_X^{(n)}$ by $B_q^{(n)}$. When n is clear from

the context we omit it and write $\mathfrak{B}, B_X, B_q$ and B instead of $\mathfrak{B}^{(n)}, B_X^{(n)}, B_q^{(n)}$ and $B^{(n)}$ respectively. Notice that

$$B_X \quad = \quad H_X^{\log n} \cdot \ldots \cdot H_X^1 \cdot H_X^1 \ldots H_X^{\log n}. \tag{3}$$

Looking at Eq (3) we see that for every X, B_X is Hermitian and positive. Looking at Eq (2) we see that B_X is stochastic. In particular B_X is also doubly stochastic.

As B_X is Hermitian we have that $(B_X^n)_{\alpha,\beta} = (B_X^n)_{\beta,\alpha}$ for any $\alpha, \beta \in X$. Since for any $\pi \in \mathbb{S}_n$ and $A \subseteq [n]$ we have that $\pi(A^c) = (\pi(A))^c$, then $(B_q^n)_{\alpha,\beta} = (B_{n-q}^n)_{\alpha^c,\beta^c}$ for any $A_1, A_2 \in [n]$, where c denotes set complement, i.e., $A^c = [n] - A$.

2.2 Almost q-set-wise independence

▶ **Definition 3.** (almost q-set-wise independence) Let $\mathfrak{D}$ be a distribution over $\mathbb{S}_n$, and q an integer. We say $\mathfrak{D}$ is ε–almost q–set-wise independent in norm ℓ, if for any initial distribution v_0 over $X = \binom{[n]}{q}$,

$$\|D_X v_0 - \mathfrak{U}_X\|_\ell \leq \varepsilon,$$

where D_X is the $|X| \times |X|$ matrix defined in Eq (1) and $\mathfrak{U}_X$ is the uniform distribution over X.

As usual, it is enough to show that

$$\|D_X v_0 - \mathfrak{U}_X\|_\ell \leq \varepsilon,$$

for initial distributions v_0 that are 1 on one element of X and zero otherwise and use the convexity of the norm.

It is well known that $\mathfrak{B}$ is perfectly one-wise independent, i.e

$$B_1 = \frac{1}{n} J, \tag{4}$$

where J is the all-one matrix.

3 Benes is $\frac{q(q-1)}{2n}$-almost q-set-wise independent

▶ **Theorem 4.** *For every q and n such that $\binom{q}{2} \leqslant n$, the Benes network is $\frac{q(q-1)}{2n}$-almost q-set-wise independent*

Proof. Fix $X = \binom{[n]}{q}$. Take $\beta \in \binom{[n]}{q}$ and let v_0 be the distribution that is 1 on β. By our previous remark , it is enough to show that:

$$\|(B_q^n)v_0 - \mathfrak{U}_X\|_1 \overset{def}{\geqslant} \sum_{\alpha \in X:\ (B_q^n)_{\alpha,\beta} < U_q} (U_q - (B_q^n)_{\alpha,\beta}) \quad \leq \quad \varepsilon,$$

where $U_q = \frac{1}{\binom{n}{q}}$. The central ingredient in the proof is showing that:

▶ **Lemma 5.** *(Main) For every $q \leq n$ where n is a power of 2 and every $\alpha, \beta \in \binom{[n]}{q}$ we have that $(B_q^n)_{\alpha,\beta} \geqslant \frac{q!}{n^q}$*

Having the lemma we see that,

$$\begin{aligned}
\sum_{\alpha:\ (B_q^n)_{\alpha,\beta}<U_q} (U_q - (B_q^n)_{\alpha,\beta}) &< \sum_{\alpha\in\binom{[n]}{q}} (U_q - \frac{q!}{n^q}) \\
&= \sum_{\alpha\in\binom{[n]}{q}} U_q - \sum_{\alpha\in\binom{[n]}{q}} \frac{q!}{n^q} \\
&= 1 - \binom{n}{q}\frac{q!}{n^q} \\
&= 1 - (1-\frac{1}{n})(1-\frac{2}{n})\dots(1-\frac{q-1}{n}) \\
&< \sum_{i=0}^{q-1} \frac{i}{n} = \frac{q(q-1)}{2n}
\end{aligned}$$

Whereas the first inequality is using the lemma and summing over all $\alpha \in \binom{[n]}{q}$, the equalities are simple algebra and the second inequality is simply inclusion-exclusion principle: Think of independent events $A_i, i = 1 \dots m$,with probabilities p_i. Then the probability that at least one of the event occurs is given by $1 - (1-p_1)(1-p_2)\dots(1-p_m)$ and is smaller then $\sum_{i=1}^{m} p_i$. Now take $m = q-1$ and $p_i = \frac{i}{n}$ and the inequality follows.

◀

Before proving the main lemma we will look into the recursive structure of the Benes network and obtain a recursive formula for $(B_q^n)_{\alpha,\beta}$.

4 The recursive structure of the Benes network

Looking at the Benes network, we notice that if we take the series

$$\underline{H}^{\log n-1}, \dots, \underline{H}^1, \underline{H}^1, \dots, \underline{H}^{\log n-1}$$

where $\underline{H}^i$ (for $i = 1, \dots, \log n - 1$) is the subgroup of $\mathbb{S}_n$ generated by the set of $n/4$ transpositions $\left\{(x \quad y) \mid x \overset{i}{\sim} y\ , x < n/2\right\}$, we get a Benes network on $[n/2]$.
Taking $\overline{H}^{\log n-1}, \dots, \overline{H}^1, \overline{H}^1, \dots, \overline{H}^{\log n-1}$, where $\overline{H}^i$ (for $i = 1, \dots, \log n - 1$) is the subgroup of $\mathbb{S}_n$ generated by the set of $n/4$ transpositions $\left\{(x \quad y) \mid x \overset{i}{\sim} y\ , x \geqslant n/2\right\}$, gives us a switching network on $\{n/2, \dots, n-1\}$ that is isomorphic to the Benes network on $[n/2]$.

We now want to use this recursive structure to get a recursive formula for $(B_q^n)_{\alpha,\beta}$, which is the probability that if we choose $\pi_{\log n} \in H^{\log n}, \dots, \pi_1 \in H^1, \sigma_1 \in H^1, \dots, \sigma_{\log n} \in H^{\log n}$ (all with flat probability $\frac{1}{2^{\frac{n}{2}}}$) then $\sigma_{\log n} \cdot \dots \sigma_1 \cdot \pi_1 \cdot \dots \pi_{\log n}(\beta) = \alpha$.

The first and last layers of the Benes network connect two inputs that differ only in the most significant bit. We use the notation of $\underline{x}$ and $\overline{x}$ introduced earlier. We remind the reader that $\underline{x} \overset{\log n}{\sim} \overline{x}$ and that either $x = \underline{x}$ or $x = \overline{x}$

Given $\alpha \subseteq [n]$, We denote

$$\begin{aligned}
\underline{\alpha} &= \{\underline{x} | x \in \alpha\} \\
\overline{\alpha} &= \{\overline{x} | x \in \alpha\}
\end{aligned}$$

We say $\beta \in X = \binom{[n]}{q}$ contains a pair $(\underline{x}, \overline{x})$ if both $\underline{x}, \overline{x}$ belong to β. We observe that if $\pi \in H^{\log n}$ then $\pi\{\underline{x}, \overline{x}\} = \{\underline{x}, \overline{x}\}$. An element $x \in [n]$ is called *paired* in β if β contains $(\underline{x}, \overline{x})$.

▶ Proposition 1. (recursive formula)

Fix α and β in $X = \binom{[n]}{q}$. Assume now β contains r pairs $(\underline{x}, \overline{x})$ and α contains d such pairs. Denote R and D the set of *paired* elements in β and α respectively. Either r or d may be zero. Denote by R_1 the set of *paired* elements in β that are on the left side of the network (i.e. belong to $[n/2]$). Symmetrically denote by R_2 the set of *paired* elements in β that are on the right side of the network (i.e. belong to $\{n/2, \ldots, n-1\}$). Notice that $R_2 = \overline{R_1}$ and that $R = R_1 \bigcup R_2$. We denote respectively the sets D_1, D_2 for α. So $|R_1| = |R_2| = r$ and $|D_1| = |D_2| = d$. In this setting we have:

$$(B_q^n)_{\alpha,\beta} \quad = \quad \frac{1}{2^{q-2r}} \cdot \frac{1}{2^{q-2d}} \sum_{\beta_L:\, R_1 \subseteq \beta_L \subseteq \beta - R_2} \; \sum_{\alpha_L:\, D_1 \subseteq \alpha_L \subseteq \alpha - D_2} (B^{\frac{n}{2}}_{|\beta_L|})_{\underline{\alpha}_L, \underline{\beta}_L} (B^{\frac{n}{2}}_{q-|\beta_L|})_{\overline{\alpha - \underline{\alpha}_L}, \overline{\beta - \underline{\beta}_L}}$$

Proof. (of Proposition 1)

The action of the first layer of the Benes network on β is captured by the set $\beta_L \subseteq \beta$ of elements of β that are moved by the first layer to the left side. Clearly, for any $\pi_{\log n} \in H^{\log n}$ we have that $\pi_{\log n}(R) = R$. Thus, paired elements in β are left in place, so R_1 stays on the left side and R_2 on the right side. Non-paired elements, i.e., the set $\beta - R$ can be either moved to the left or to the right, and each possibility occurs with equal probability.

Thus, to transfer β to α through the Benes network we have to go over all the possibilities to choose β_L s.t. $R_1 \subseteq \beta_L \subseteq \beta - R_2$. Once we choose β_L, we have to route $\underline{\beta_L}$ through the left Benes network to some $\underline{\alpha_L}$ and $\overline{\beta - \beta_L}$ through the right Benes network to $\overline{\alpha - \underline{\alpha_L}}$. The last layer then routes $\underline{\alpha_L} \cup \overline{\alpha - \underline{\alpha_L}}$ to α.

Let us see what paths are possible. First, as we saw before, we have freedom to choose any $R_1 \subseteq \beta_L \subseteq \beta - R_2$. Next, we have freedom to choose any α_L such that $|\alpha_L| = |\beta_L|$ and $D_1 \subseteq \alpha_L \subseteq \alpha - D_2$. The probability of such an event is $\frac{1}{2^{q-2r}}$ for the probability the first layer is captured by β_L, $\frac{1}{2^{q-2d}}$ for the probability the last layer is captured by α_L, and for each α_L, β_L as above, $(B^{\frac{n}{2}}_{|\beta_L|})_{\underline{\alpha}_L, \underline{\beta}_L}$ for the probability the left Benes network routes $\underline{\beta}_L$ to $\underline{\alpha}_L$, and $(B^{\frac{n}{2}}_{q-|\beta_L|})_{\overline{\alpha - \underline{\alpha}_L}, \overline{\beta - \underline{\beta}_L}}$ for the probability the right Benes network routes $\overline{\beta - \underline{\beta}_L}$ to $\overline{\alpha - \underline{\alpha}_L}$. ◀

We now prove the main lemma.

5 Proof of main lemma

Proof. (of Lemma 5). We prove the lemma by induction on q using the recursive structure of the Benes network. The base case is $q = 1$ and arbitrary n and is very simple: $(B_1^n)_{\alpha,\beta} = \frac{1}{n}$ for every $\alpha, \beta \in \binom{[n]}{1}$. We now assume for all $q \leq q_0$ and we prove for $q = q_0 + 1$.

We prove the case $q = q_0 + 1$ by induction on n. The base case is for $n = n_q$ where n_q is the first power of 2 such that $q \leqslant n_q$, i.e. $\frac{n_q}{2} < q \leqslant n_q$. In this case

$$(B_q^{n_q})_{\alpha,\beta} \quad = \quad (B^{n_q}_{n_q - q})_{\alpha^c, \beta^c} \;\geqslant\; \frac{(n_q - q)!}{n_q^{n_q - q}} \;\geqslant\; \frac{q!}{n_q^q},$$

where we have used the simple fact that $n_q - q \leq q_0$ and the induction hypothesis on q.

Now, fix α and β in $X = \binom{[n]}{q}$ with the same setting as proposition 1. we get:

$$\begin{aligned}
(B_q^n)_{\alpha,\beta} &= \frac{1}{2^{q-2r}} \cdot \frac{1}{2^{q-2d}} \sum_{\beta_L:\, R_1 \subseteq \beta_L \subseteq \beta - R_2} \sum_{\alpha_L:\, D_1 \subseteq \alpha_L \subseteq \alpha - D_2} (B_{|\beta_L|}^{\frac{n}{2}})_{\underline{\alpha}_L, \underline{\beta}_L} (B_{q-|\beta_L|}^{\frac{n}{2}})_{\overline{\alpha - \underline{\alpha}_L}, \overline{\beta - \underline{\beta}_L}} \\
&= \frac{1}{2^{q-2r}} \cdot \frac{1}{2^{q-2d}} \sum_{i=r}^{q-r} \sum_{\beta_L:\, R_1 \subseteq \beta_L \in \binom{\beta - R_2}{i}} \sum_{\alpha_L:\, D_1 \subseteq \alpha_L \in \binom{\alpha - D_2}{i}} (B_i^{\frac{n}{2}})_{\underline{\alpha}_L, \underline{\beta}_L} (B_{q-i}^{\frac{n}{2}})_{\overline{\alpha - \underline{\alpha}_L}, \overline{\beta - \underline{\beta}_L}} \\
&\geqslant \frac{1}{2^{q-2r}} \cdot \frac{1}{2^{q-2d}} \sum_{i=r}^{q-r} \binom{q-2r}{i-r} \binom{q-2d}{i-d} \frac{i!}{(\frac{n}{2})^i} \frac{(q-i)!}{(\frac{n}{2})^{q-i}} \\
&= \frac{(q-2d)!}{n^q} \frac{(d!)^2}{2^{q-2r-2d}} \cdot \sum_{i=r}^{q-r} \binom{q-2r}{i-r} \binom{i}{d} \binom{q-i}{d}
\end{aligned}$$

where the inequality is by induction on n and the final equality is simple algebra.

We want to show that $\frac{(q-2d)!}{n^q} \frac{(d!)^2}{2^{q-2r-2d}} \cdot \sum_{i=r}^{q-r} \binom{q-2r}{i-r} \binom{i}{d} \binom{q-i}{d} \geqslant \frac{q!}{n^q}$. This follows from our main combinatorial lemma:

▶ **Lemma 6.** *(Combinatorial lemma) Let $d \leqslant r$ then:*

$$(q-2d)! \frac{(d!)^2}{2^{q-2r-2d}} \cdot \sum_{i=r}^{q-r} \binom{q-2r}{i-r} \binom{i}{d} \binom{q-i}{d} \geqslant q!$$

◀

We now prove the combinatorial lemma.

Proof. (of Lemma 6)

We denote $(N)_l = N(N-1) \cdot \ldots (N-l+1) = \frac{N!}{(N-l)!}$, so if $l > N$ then $(N)_l = 0$. The inequality is equivalent to

$$\sum_{i=r}^{q-r} \frac{\binom{q-2r}{i-r}}{2^{q-2r}} (i)_d (q-i)_d \geqslant \frac{(q)_{2d}}{2^{2d}} \tag{5}$$

We consider two experiments. In both experiments we have q numbered balls, and two colors black and white. In the first experiment we color all the q balls uniformly at random. In the second experiment r of the balls are already colored white and r are colored black and we color the remaining $q-2r$ balls uniformly at random. In both experiments we define a random variable X whose value is the number of possibilities to choose an ordered sequence of d white balls and an ordered sequence of d black balls. I.e., if C is the random variable counting the number of balls that are colored white, then $X = (i)_d (q-i)_d$ given that $C = i$. Now we compute the expected value of X in each of the two experiments.

In the first experiment $E(X) = \sum_{i=d}^{q-d} \frac{\binom{q}{i}}{2^q} (i)_d (q-i)_d$, and rearranging:

$$\begin{aligned}\sum_{i=d}^{q-d}\frac{\binom{q}{i}}{2^q}(i)_d(q-i)_d &= \frac{1}{2^q}\sum_{i=d}^{q-d}\frac{q!}{i!(q-i)!}(i)_d(q-i)_d \\ &= \frac{1}{2^q}\sum_{i=d}^{q-d}\frac{q!}{(i-d)!(q-i-d)!} = \frac{(q)_{2d}}{2^q}\sum_{i=d}^{q-d}\frac{(q-2d)!}{(i-d)!(q-i-d)!} \\ &= \frac{(q)_{2d}}{2^q}\sum_{i=d}^{q-d}\binom{q-2d}{i-d} = \frac{(q)_{2d}}{2^q}2^{q-2d} = \frac{(q)_{2d}}{2^{2d}}\end{aligned}$$

Thus, this is exactly the right term in inequality (5). Also, in the second experiment $E(X) = \sum_{i=r}^{q-r}\frac{\binom{q-2r}{i-r}}{2^{q-2r}}(i)_d(q-i)_d$, which is the left term in the inequality. Thus, we need to prove that $E(X)$ in the second experiment is larger than $E(X)$ in the first experiment.

Let t_i (p_i) denote the probability that $C = i$ in the first (second) experiment respectively. Namely, $t_i = \frac{\binom{q}{i}}{2^q}$ and $p_i = \frac{\binom{q-2r}{i-r}}{2^{q-2r}}$. Let $X_i = (X|C = i) = (i)_d(q-i)_d$. It is easy to see that:

▶ Claim 1. If i, j are integers, $(d \leqslant i < j \leqslant \frac{q}{2})$, then $t_i < t_j$ and $p_i < p_j$.

Also,

▶ **Lemma 7.** *If i, j are integers, $(d \leqslant i < j \leqslant \frac{q}{2})$ then $X_i < X_j$*

▶ **Lemma 8.** *If $d \leqslant i < j \leqslant \frac{q}{2}$ then $\frac{p_j}{p_i} > \frac{t_j}{t_i}$*

Both lemmas are proven easily by induction. Take $j = i + 1$, and by simple algebra one can see that the inequalities in both lemmas are equivalent to $2i < q - 1$ which is true since i is an integer and $i < \frac{q}{2}$. For example, $\frac{p_i}{p_{i+1}} = \frac{i+1-r}{q-r-i}$ and therefore $p_i < p_{i+1}$ iff $i + 1 - r < q - r - i$ iff $2i < q - 1$.

Now, from these two lemmas together with the symmetry around $\frac{q}{2}$ of our variables and probabilities ($X_j = X_{q-j}, t_j = t_{q-j}, p_j = p_{q-j}$), we get that there exists some index $i_0 \leqslant \frac{q}{2}$ such that $p_i \geqslant t_i$ if and only if $i \in S = \{i_0, i_0 + 1 \ldots, q - i_0\}$. We also know that since $\{X_i\}, i = 1, \ldots, \frac{q}{2}$ is monotonically increasing, it must be that for any $i \in S, j \in S^c$ we have that $X_i \geqslant X_j$. We now prove this implies that the expected value of the second experiment is larger then that of the first experiment:

▶ **Lemma 9.** *Let $\{X_i\}_{i\in A}$ be a set of non negative variables. Assume $S \subset A$ such that:*

- *For any $i \in S, j \in S^c$ we have that $X_i \geqslant X_j$, i.e $min_{i\in S}\{X_i\} \geqslant max_{i\in S^c}\{X_i\} = X_{max}(S^c)$. And,*
- *Let $\{p_i\}_{i\in A}$ and $\{t_i\}_{i\in A}$ be two distributions such that $p_i \geqslant t_i$ if and only if $i \in S$.*

Then $\sum_{i\in A} p_i X_i \geq \sum_{i\in A} t_i X_i$.

To see the lemma, notice that:

$$\begin{aligned}\sum_{i\in A} p_i X_i &= \sum_{i\in S} p_i X_i + \sum_{i\in S^c} p_i X_i = \sum_{i\in S} t_i X_i + \sum_{i\in S}(p_i - t_i)X_i + \sum_{i\in S^c} t_i X_i + \sum_{i\in S^c}(p_i - t_i)X_i \\ &= \sum_{i\in A} t_i X_i + \sum_{i\in S}(p_i - t_i)X_i + \sum_{i\in S^c}(p_i - t_i)X_i\end{aligned}$$

However, $p_i - t_i$ is positive for $i \in S$ and negative otherwise, and $X_i \geq X_{max}(S^c)$ for $i \in S$ and $X_i \leq X_{max}(S^c)$ for $i \in S^c$. Thus,

$$\sum_{i \in A} p_i X_i \quad \geq \quad \sum_{i \in A} t_i X_i + \sum_{i \in S} (p_i - t_i) X_{max}(S^c) + \sum_{i \in S^c} (p_i - t_i) X_{max}(S^c) = \sum_{i \in A} t_i X_i$$

◀

References

1 Masayuki Abe. Mix-networks on permutation networks. In *Proceedings of the International Conference on the Theory and Applications of Cryptology and Information Security: Advances in Cryptology*, ASIACRYPT '99, pages 258–273, London, UK, UK, 1999. Springer-Verlag.

2 Masayuki Abe and Fumitaka Hoshino. Remarks on mix-network based on permutation networks. In Kwangjo Kim, editor, *Public Key Cryptography*, volume 1992 of *Lecture Notes in Computer Science*, pages 317–324. Springer Berlin Heidelberg, 2001.

3 David L Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 24(2):84–90, 1981.

4 Artur Czumaj, Przemka Kanarek, Mirosław Kutyłowski, and Krzyztof Loryś. Delayed path coupling and generating random permutations via distributed stochastic processes. In *Proceedings of the tenth annual ACM-SIAM symposium on Discrete algorithms*, pages 271–280. Society for Industrial and Applied Mathematics, 1999.

5 Artur Czumaj, Przemka Kanarek, Krzysztof Lorys, and Miroslaw Kutylowski. Switching networks for generating random permutations, 2001.

6 Persi Diaconis and Mehrdad Shahshahani. On the eigenvalues of random matrices. *Journal of Applied Probability*, pages 49–62, 1994.

7 WT Gowers. An almost m-wise independent random permutation of the cube. *Combinatorics Probability and Computing*, 5:119–130, 1996.

8 Shlomo Hoory, Avner Magen, Steven Myers, and Charles Rackoff. Simple permutations mix well. In *Automata, Languages and Programming*, pages 770–781. Springer, 2004.

9 Frank Thomson Leighton. *Introduction to parallel algorithms and architectures*. Morgan Kaufmann San Francisco, 1992.

10 Helger Lipmaa. Efficient nizk arguments via parallel verification of benes networks. In *To appear in SCN (9th Conference on Security and Cryptography for Networks) 2014*, 2014.

11 Nathan Lulov. *Random walks on the symmetric group generated by conjugacy classes*. PhD thesis, Harvard University, 1996.

12 Nathan Lulov and Igor Pak. Rapidly mixing random walks and bounds on characters of the symmetric group. *Journal of Algebraic Combinatorics*, 16(2):151–163, 2002.

13 Ben Morris. Improved mixing time bounds for the thorp shuffle and l-reversal chain. *The Annals of Probability*, pages 453–477, 2009.

14 Ben Morris, Phillip Rogaway, and Till Stegers. How to encipher messages on a small domain. In *Proceedings of the 29th Annual International Cryptology Conference on Advances in Cryptology*, CRYPTO '09, pages 286–302, Berlin, Heidelberg, 2009. Springer-Verlag.

15 Charles Rackoff and Daniel R Simon. Cryptographic defense against traffic analysis. In *Proceedings of the twenty-fifth annual ACM symposium on Theory of computing*, pages 672–681. ACM, 1993.