# Reliable Computation by Networks in the Presence of Noise

Tomás Feder*

**Abstract**

We prove lower bounds on the depth of Boolean networks that can compute reliably in the presence of randomly occuring failures. We also give a bound on the reliability that error-tolerant networks can achieve: this bound implies a limit strictly smaller than 1/2 on the failure probability per gate that can be tolerated. Our results improve recently published bounds of Pippenger on the depth of error-tolerant formulas, and extend those bounds to the case of reliable computation by networks.

## I. Introduction

In his study of error-tolerant computation by Boolean formulas [4], Pippenger shows that if formulas are used to compute Boolean functions in the presence of randomly occuring failures, then (1) there is a limit strictly less than 1/2 to the failure probability per gate that can be tolerated, and (2) formulas that tolerate failures must be deeper (and therefore compute more slowly) than those that do not. (A formula is simply a network in which the output of each gate is used as the input to at most one gate.) Thus, at least in the case of formulas, two striking features of von Neumann's solution to the problem of reliable computation by networks [3] are shown to be essential.

*This work was done in part by the author as a Research Student Associate at the IBM Almaden Research Center. It is also supported in part by a graduate scholarship from AT&T Bell Laboratories. The author is a Ph.D. student in the Department of Computer Science at Stanford University, Stanford, CA 94305, USA.

Pippenger's work leaves open the question of whether these two limitations still hold when networks rather than formulas are used to compute Boolean functions in the presence of noise. As pointed out in [4], the information-theoretic argument presented there requires that the inputs to each gate be independent, a condition that holds in the case of formulas but fails in the case of more general networks.

In this paper we show that, even for general networks, the two features of von Neumann's construction mentioned above are essential. We also prove a bound on the reliability that an error-tolerant network can achieve; this bound generalizes condition (1) above by showing that the error probability for the entire network approaches 1/2 as the failure probability for each gate approaches a limit strictly smaller than 1/2.

Our proofs follow the outline of [4], but replace the information-theoretic argument with a simpler probabilistic argument that does not require independence among the inputs to a gate and thus applies to the network case as well. The main idea in this argument is to bound the probability of a special event which, in a certain sense, disconnects the output of the network from one of the input variables. This method also yields better bounds, replacing parameter $\Delta = 1 + \delta \log_2 \delta + (1-\delta)\log_2(1-\delta)$ used in [4] with a larger value $1 - 2\delta$.

It remains an open problem to obtain stronger bounds. In the case considered by von Neumann, the failure probability $\epsilon$ of each gate must be less than 1/6, while our argument only gives a bound of 1/3. More significantly, in von Neumann's construction, reliability is achieved by adding to the network correction layers that account for a fraction of the total depth asymptotic to $2/\log_3(1/\epsilon)$, as $\epsilon \to 0$; our results only show that this fraction must be at least asymptotic to $2\epsilon/\ln 3$. A more detailed comparison with von Neumann's construction can be found in [4].

# II. The Depth Bound

We recall some definitions from [4]. Suppose that among the kinds of gates available there is one that computes a Boolean function $f$ that depends essentially on $k \geq 2$ arguments, but none that depends essentially on more than $k$ arguments. By $d$-fold composition of $f$ with itself, we obtain a Boolean function $f_d$ that depends essentially on $n = k^d$ arguments.

A network may be regarded as a directed acyclic graph in which the nodes of in-degree zero are either Boolean constants (0 or 1) or references to arguments ($x_1$,$x_2$,...) and the nodes of in-degree $l \geq 1$ are gates computing Boolean functions of $l$ arguments. Furthermore, there is a unique node of out-degree zero, the ouput of the network. The *depth* of a network is then the number of gates on a longest path from an argument reference to the output; it reflects the delay between the time that the values of the arguments are presented and the time the value of the function is produced. In the absence of failures, the function $f_d$ is computed by a formula of depth $d$. We show that any network that computes $f_d$ with high probability in the presence of failures must be significantly deeper.

**Theorem 1** *Let $N$ be a network with the following property: if each input of $N$ operates correctly, and if each gate of $N$ independently operates correctly with probability $1-\epsilon$, where $\epsilon > 0$, then $N$ operates correctly (that is, produces the value of $f_d$) with probability at least $1 - \delta$, where $\delta < 1/2$. Then, for $\epsilon < (k-1)/2k$, the depth $c$ of $N$ must satisfy*

$$c \geq Rd - S,$$

*where*

$$R = \frac{\log k}{\log((1-2\epsilon)k)}$$

*and*

$$S = \frac{-\log(1-2\delta)}{\log((1-2\epsilon)k)}.$$

*(Since $R > 1$, computation proceeds more slowly in the presence of failures.) If $\epsilon \geq (k-1)/2k$, then*

$$d \leq T,$$

*where*

$$T = \frac{\log(1-2\delta)}{\log(1-2\epsilon)}.$$

*(Thus, if $\epsilon$ is too large, only relatively simple functions can be computed reliably in the presence of failures.)*

PROOF. For $1 \leq m \leq n$, let $N_m$ denote the network obtained from $N$ as follows. Since $f_d$ depends essentially on $x_m$, there exist Boolean constants

$c_1, \ldots, c_{m-1}, c_{m+1}, \ldots, c_n$ such that $f_d(c_1, \ldots, c_{m-1}, x_m, c_{m+1}, \ldots, c_n)$ is either $x_m$ or its complement $\overline{x_m}$. Substitute these constants for all references to arguments other than $x_m$ in $N$ to obtain $N_m$.

Let $X$ be a Boolean random variable that assumes the values 0 and 1 with equal probabilities. For any network $M$, let $Y(M)$ be the Boolean random variable produced by $M$ when $X$ is substituted for all argument references in $M$, and each gate in $M$ fails with probability $\epsilon$, independently of $X$ and of all other gates. Furthermore, let $\Phi(M, \alpha)$ be the generating function for the paths from argument references to the root in $M$, classified according to the number of gates they contain. That is, if $L(P)$ denotes the number of gates on the path $P$, then

$$\Phi(M, \alpha) = \sum_P \alpha^{L(P)}.$$

A gate which produces an incorrect output value with probability $\epsilon$ can be equivalently described as a gate which *fails* with probability $2\epsilon$ and produces, in case of failure, an output value of 0 or 1 with equal probabilities and independent of the values of the inputs to the gate. If for every path $P$ in a network $M$ from an argument reference to the output, there is a gate along path $P$ which fails (i.e., produces an output independent of the values of its inputs), we say that $M$ is *disconnected*, and we denote this event by $D(M)$.

Now, if $N_m$ is not disconnected, then some path $P$ in $N_m$ from an argument reference to the output has no failing gates. The event that path $P$ has no failing gates occurs with probability $(1-2\epsilon)^{L(P)}$; summing this probability over all possible paths $P$ gives an upper bound on the probability that $N_m$ is not disconnected, namely

$$1 - \Pr\{D(N_m)\} \leq \Phi(N_m, 1 - 2\epsilon). \tag{1}$$

On the other hand, if $N_m$ is disconnected, then the output $Y(N_m)$ produced by $N_m$ is independent of $X$, because all failing gates produce outputs independent of their inputs. Thus, since $X$ assumes the values 0 and 1 with equal probabilities, the output $Y(N_m)$ will be incorrect with probability 1/2 in the case where $N_m$ is disconnected. But, by assumption, $Y(N_m)$ is incorrect with probability at most $\delta$, so

$$\frac{1}{2}\Pr\{D(N_m)\} \leq \delta. \tag{2}$$

Combining inequalities (1) and (2) yields

$$\Phi(N_m, 1-2\epsilon) \geq 1-2\delta. \tag{3}$$

This strengthens the bound $\Phi(F_m, 1-2\epsilon) \geq \Delta$ obtained by Pippenger for a formula $F$, with $\Delta$ as defined above. The distinction between networks and formulas is unessential for the rest of the proof, because the network $N$ can be "unfolded" into a formula $F$ by making multiple copies of inputs and gates, and this operation preserves the length of paths from argument references to the output so that $\Phi(F_m, \alpha) = \Phi(N_m, \alpha)$. Therefore the argument presented in [4] goes through and completes the proof. □

# III. The Reliability Bound

In this section we give a lower bound on the error probability $\delta$ that must be tolerated for the entire network, given the failure probability $\epsilon$ of each gate.

Given $\epsilon > 0$, let $\beta$ be the smallest positive root of the equation

$$\beta = 2\epsilon + (1-2\epsilon)\beta^k. \tag{4}$$

Thus $\beta$ satisfies $\beta + \beta^2 + \cdots + \beta^{k-1} = 2\epsilon/(1-2\epsilon)$ when $\epsilon < (k-1)/2k$, and in particular $\beta = 2\epsilon/(1-2\epsilon)$ for $k = 2$. On the other hand, if $\epsilon \geq (k-1)/2k$, then $\beta = 1$. The following result generalizes the second part of Theorem 1 for an arbitrary error probability $\epsilon > 0$.

**Theorem 2** *Let $f_d$, $N$, $\epsilon$ and $\delta$ be as in the statement of Theorem 1, and suppose that $\delta < \beta/2$. Then*

$$d \leq T',$$

*where*

$$T' = \frac{\log(1-2\delta/\beta)}{\log(1-2\epsilon/\beta)}.$$

*(Thus given $\epsilon$, if $\delta$ is too small, only relatively simple functions can be computed reliably in the presence of failures.)*

PROOF. We will prove a relation similar to (3), namely

$$\Phi(N_m, 1-2\epsilon/\beta) \geq 1-2\delta/\beta. \tag{5}$$

Then the theorem follows by replacing $1-2\epsilon$ and $1-2\delta$ with $1-2\epsilon/\beta$ and $1-2\delta/\beta$ respectively in the proof of the second part of Theorem 1. In particular, the condition $1-2\delta>0$ becomes $1-2\delta/\beta>0$, and the condition $1-2\epsilon \leq 1/k$ becomes $1-2\epsilon/\beta \leq 1/k$, which is true for all $\epsilon>0$. (This inequality follows from the convexity of $g(z)=2\epsilon+(1-2\epsilon)z^k$ for $0\leq z\leq 1$ and the fact that $g(\beta)=\beta$ and $g(1)=1$, which imply that $g'(\beta)\leq 1$, where $g'(\beta)=k(1-2\epsilon/\beta)$ by (4).)

To prove (5), we will show that

$$1-\Pr\{D(M)\}/\beta \leq \Phi(M,1-2\epsilon/\beta) \tag{6}$$

for any network $M$, where $D(M)$ is the event that the network $M$ is disconnected, which was defined in the proof of Theorem 1. Letting $M$ be $N_m$ in (6) and using (2) will then give (5), as desired.

The proof of (6) is by induction on the structure of $M$. If $M$ is a constant input, then $\Pr\{D(M)\}=1\geq\beta$ and $\Phi(M,1-2\epsilon/\beta)=0$. If $M$ is an argument reference, then $\Pr\{D(M)\}=0$ and $\Phi(M,1-2\epsilon/\beta)=1$. For the inductive step, suppose that the output of $M$ is the output of a gate $g$ that computes a Boolean function of the values produced by the subnetworks $M_1,\ldots,M_k$. Notice that $M$ is disconnected if and only if either $g$ fails (an event that has probability $2\epsilon$) or $g$ does not fail and $M_1,\ldots,M_k$ are all disconnected. Therefore

$$\Pr\{D(M)\}=2\epsilon+(1-2\epsilon)\Pr\{D(M_1)\cdots D(M_k)\}. \tag{7}$$

Since the conditional probability that one of the $M_i$ is disconnected given that one or more of the other $M_j$ are disconnected is at least as large as the unconditional probability that $M_i$ is disconnected (see [2] or Lemma 4.1 in [1] for a proof of this fact), we have

$$\Pr\{D(M_1)\cdots D(M_k)\}\geq \Pr\{D(M_1)\}\cdots\Pr\{D(M_k)\}. \tag{8}$$

Combining (7) and (8), and subtracting (4) from the resulting inequality, we obtain

$$\Pr\{D(M)\}-\beta \geq (1-2\epsilon)(\prod_{i=1}^{k}\Pr\{D(M_i)\}-\beta^k). \tag{9}$$

To bound the right hand side of this equation, we will use the fact that if $0\leq z_1,\ldots,z_k\leq\beta$, then

$$\prod_{i=1}^{k} z_i-\beta^k \geq \beta^{k-1}\sum_{i=1}^{k}(z_i-\beta). \tag{10}$$

Equation (10) follows from the convexity of the function $f(h) = \prod_{i=1}^{k}(\beta + h(z_i - \beta))$ for $0 \leq h \leq 1$, because this property implies that $f(1) - f(0) \geq f'(0) = \beta^{k-1}\sum_{i=1}^{k}(z_i - \beta)$. We use (10) with $z_i = \min(\Pr\{D(M_i)\}, \beta)$, so that (9) becomes

$$\Pr\{D(M)\} - \beta \geq (1 - 2\epsilon)\beta^{k-1}\sum_{i=1}^{k}(\min(\Pr\{D(M_i)\}, \beta) - \beta). \tag{11}$$

This equation can be rewritten using (4) as

$$1 - \Pr\{D(M)\}/\beta \leq (1 - 2\epsilon/\beta)\sum_{i=1}^{k}\max(1 - \Pr\{D(M_i)\}/\beta, 0). \tag{12}$$

By the inductive hypothesis, for each $i$, the corresponding term of the summation in the right hand side is at most $\Phi(M_i, 1 - 2\epsilon/\beta)$, so the right hand side is at most $\Phi(M, 1 - 2\epsilon/\beta)$ and (6) follows, concluding the proof. □

# IV. Conclusion

We have shown that fault tolerant computation of Boolean functions by networks can only be achieved if the reliability $1 - 2\epsilon$ of the gates exceeds $1/k$, and that in this case networks that tolerate failures must be deeper than those that do not. Furthermore, our results expose a natural relationship between the reliability $1 - 2\epsilon$ of each gate and the reliability $1 - 2\delta$ of the entire network.


# Acknowledgment

The author would like to thank Nicholas Pippenger for his valuable suggestions during this work.


# References


[1] T. E. Harris, "A lower bound for the critical probability in a certain percolation process," Proc. Cambridge Phil. Soc., 56 (1960) 13-20.

[2] C. M. Fortuin, P. W. Kasteleyn and J. Ginibre, "Correlation Inequalities on Some Partially Ordered Sets," Comm. Math. Phys., 22 (1971) 89-93.

[3] J. von Neumann, "Probabilistic logics and the synthesis of reliable organisms from unreliable components," in C. E. Shannon and J. McCarthy (Eds.), Automata Studies, Princeton University Press, 1956, 43-98.

[4] Nicholas Pippenger, "Reliable computation by formulas in the presence of noise," IEEE Trans. Inform. Theory 34(1988) 194-197.