Counting Networks with Arbitrary Fan-Out
(Preliminary Version)

by

E. Aharonson and H. Attiya

Technical Report #579
June 1991
Abstract

It is shown that an acyclic smoothing network (and hence counting network) of fan-out $n$ cannot be constructed from balancers of fan-out $b_1, \ldots, b_k$, if there exists a prime factor $p$ of $n$, such that $p$ does not divide $b_i$, for all $i$, $1 \leq i \leq k$. This holds regardless of the depth of the network, as long as it is finite. In particular, only smoothing networks of fan-out $2^\ell$ (for a non-negative integer $\ell$) can be constructed out of 2-balancers. A very simple construction of cyclic counting networks with fan-out $n$ is presented. Proving the correctness of this construction involves a novel technique of arguing about counting networks in non-quiescent states.
1 Introduction

Consider the following simulation problem. We are given an algorithm written for a system of \( n \) processors; in the algorithm, each of the processors is supposed to execute (asynchronously) a sequence of computation steps. Now assume we wish to robustly simulate this algorithm on a system with \( n \) processors which communicate via shared memory. It is required that the \( k \)th step of the \( i \)th (simulated) processor will be executed only after its \((k - 1)\)th step was executed. Since simulated processors must perform the same number of steps, it is desired that the simulation will proceed evenly with all processors, so as not to waste work.

A very simple way to solve this problem is by using a counter. Each of the simulating processors accesses the counter, and obtains a value \( I \); it then performs step \( \lfloor I/n \rfloor \) of processor \((I \mod n)\). Such a counter can be easily implemented using a single shared Fetch&Add variable. However, empirically, the time to access such shared variable grows at least linearly with the contention - the number of processors concurrently trying to access the variable. Since all \( n \) processors access the same variable, the high degree of contention creates a bottleneck which degrades the performance of the simulation.

Recently, Aspnes, Herlihy and Shavit ([2]) suggested a completely different approach to such counting problems. Their idea is to use a collection of shared variables, called balancers, each having low expected contention, in such a way that a processor need to access only a few variables in order to obtain a value from the counter. A \( b \)-balancer can be thought of as a \( b \)-input, \( b \)-output toggle. When an input appears on one of the input wires, it takes the output wire to which the toggle is set, and toggles the gate, so that the next input will leave on the next output wire. Assuming the balancer is initialized so that the first input to pass through will take the top output wire, then after \( m \) inputs have passed through the toggle exactly \( \lfloor m/b \rfloor \) have exited on the top output wire, and \( \lceil m/b \rceil \) on the bottom output wire. (Figure 1 depicts a 3-balancer; we draw wires as horizontal lines with the balancer stretched vertically.)

One can "connect" a collection of balancers to form a balancing network much in the same way that a sorting network is obtained by connecting a collection of comparators. This is done by connecting output wires from some balancers to input wires of a balancer. Some of the remaining unconnected input wires are the input wires of the network, and some of the unconnected output wires are the output wires of the network. On a shared-memory multiprocessor, each balancer can be implemented with a Compare&Swap variable. One can think of the output wires as pointers to other balancers or shared variables. Each request corresponds to a token which traverses the network starting from an initial balancer (input wire), following the pointer obtained by accessing this balancer to the next balancer, and so on. Clearly, the number of steps a token takes in order to traverse the network is at most the depth of the network.

Let \( x_i \) denote the number of tokens having entered the network on the \( i \)th input wire,

---

1 The cost of contention vary according to the architecture of the system and the specific arbitration protocols used (cf. [1]).
2 Our exposition is borrowed from the survey by Cynthia Dwork ([5]).
3 The length of the longest simple path from an input wire to an output wire.
1 ≤ i ≤ n. Similarly, $y_i$ denotes the number of tokens having left the network on the $i$th output wire, 1 ≤ i ≤ n. A balancing network of fan-out $n$ is a counting network if, in any quiescent state (a state in which no tokens are currently in the network), 0 ≤ $y_i - y_j$ ≤ 1, for any $i,j, 0 ≤ i < j ≤ n$. A smoothing network only guarantees that $|y_i - y_j| ≤ 1$, for every $i,j, 0 ≤ i, j ≤ n$.

Similarly, $Y_i$ denotes the number of tokens having left the network on the $i$th output wire, 1 ≤ $i$ ≤ n. A balancing network of fan-out $n$ is a counting network if, in any quiescent state (a state in which no tokens are currently in the network), 0 ≤ $y_i - y_j$ ≤ 1, for any $i,j, 0 ≤ i < j ≤ n$. A smoothing network only guarantees that $|y_i - y_j| ≤ 1$, for every $i,j, 0 ≤ i, j ≤ n$.

Clearly, a counting network (or even a smoothing network) of fan-out $n$, can be used to solve our simulation problem. Each of the simulating processors pushes a token through the network, as described above. If the token exits on the $i$th wire, the processor simulates the next step of the $i$th simulated processor.

Unfortunately, the only known constructions of counting networks have fan-out $2^\ell$, for $\ell ≥ 1$, using 2-balancers ([2, 3, 7]). For sorting networks, "padding" can be used to transfer a sorting network of fan-out $n' ≥ n$ into a sorting of fan-out $n$. A similar idea would not work for counting networks. One of the properties of counting network is that it balances the number of tokens on each output wire. The tokens will be output on all $n'$ output wires and not only on the $n$ desired output wires.

This paper addresses the problem of constructing counting networks with arbitrary fan-out. More precisely, what is the fan-out of counting (or smoothing) networks that can be construct using a finite (but unbounded) number of balancers whose fan-out is in the set \{b_1, ..., b_k\}.

Our first major result is negative. If there exists a prime factor $p$ of $n$, such that $p$ does not divide $b$, for all $b ∈ \{b_1, ..., b_k\}$, then there is no smoothing network (and hence no counting network) with fan-out $n$ constructed only from balancers with fan-out $b_1, ..., b_k$. (This holds regardless of the depth or the size of the network, as long as it is finite.) In particular, only networks with fan-out $2^\ell$ (for a non-negative integer $\ell$) can be constructed out of 2-balancers.

Our proof is based on the Unique Representation Theorem.

The original definition in [2] requires balancing networks to be acyclic. Our second major result deals with cyclic networks. We show a very simple construction of cyclic counting networks with fan-out $n$. The construction takes an arbitrary acyclic network with fan-out $2^{\log_2 n} (≥ n)$ and "feeds back" some of its output wires into its input wires. Proving the correctness of this construction involves a novel technique of arguing about counting networks in a non-quiescent state.

In order to bound the number of steps it takes a token to traverse an acyclic balancing network, we only need to calculate the depth of the network. Analyzing the performance
of cyclic counting networks is more involved. It is possible for a token to keep cycling in
the network; however, this happens only if other tokens keep exiting the network. We show
that, when amortized over a finite collection of tokens, the number of steps a token takes
in order to traverse our cyclic networks is at most twice the depth of the embedded acyclic
network. Using the networks constructed in [2, 3] as the embedded acyclic networks, we get
an amortized complexity of \( O(\log^2 n) \). Using the network constructed in [7] as the embedded
acyclic networks, we get an amortized complexity of \( O(\log n \log \log n) \).

Our impossibility result does not require balancers to have fan-in equal to fan-out. This
only make the result stronger. On the other hand, the construction of cyclic counting networks
uses balancers with fan-in equal to fan-out.

Counting networks have attracted a lot of interest and study recently. Our results further
the understanding of the mathematical features and the operational behavior of smoothing
and counting networks.

The rest of the paper is organized as follows. In Section 2, we formally define balancing,
smoothing and counting networks. In Section 3, we present the impossibility result for acyclic
counting networks. Section 4 contains our construction of cyclic counting networks of arbitrary
fan-out, together with proof of correctness and performance analysis. We conclude, in Section 5,
with a discussion of our results and directions for future research.

2 Counting and Smoothing Networks

Counting and smoothing networks belong to a larger class of networks called balancing net­
works, constructed from wires and computing elements called balancers. The construction
is similar to the way comparison networks are constructed from wires and comparators (see,
e.g., [4]). In this section, we describe balancing, counting and smoothing networks; our de­
scription follows the description in [2, 6], modified to accommodate balancers with fan-in not
equal to fan-out.

Formally, a \( b \)-balancer is a computing element with finite number of input wires, \( h \), and
\( b \) output wires. Denote by \( x_i, i \in \{1, \ldots, h\} \), the number of tokens input on the balancer's
ith input wire, and by \( y_i, i \in \{1, \ldots, b\} \), the number of tokens output on the balancer's ith
output wire. Tokens can arrive through any input wire and at arbitrary times. Intuitively, a
balancer is a toggle mechanism, which, gives a stream of input tokens, sends them to output
wires 1, 2, \ldots, \( b \), in that order and in a cyclic way. That is, it balances the number of tokens
on each output wire. (See Figure 2.)
Let the state of a balancer at a given time be defined as the sets of tokens that have passed on its input and output wires. The formal properties of a balancer with fan-in $h$ and fan-out $b$ are as follows:

1. In any state $\sum_{i=1}^{h} x_i \geq \sum_{i=1}^{b} y_i$. (A balancer never creates tokens.)

2. Given any finite number of tokens $\sum_{i=1}^{h} x_i$ to the balancer, the balancer reaches a quiescent state within a finite amount of time; that is, a state in which $\sum_{i=1}^{h} x_i = \sum_{i=1}^{b} y_i$. (A balancer never swallows tokens.)

3. In any quiescent state, $0 \leq y_i - y_j \leq 1$ for any $i < j$. (The output has the step property.)

Let $B = \{b_1, \ldots, b_k\}$ be a set of positive integers. A balancing network with fan-out $n$ over $B$ is a directed graph with three kinds of nodes: $n$ source nodes, $n$ sink nodes and some finite number of inner nodes. Source nodes represent input wires; sink nodes represent the output wires. Inner nodes represent the balancers of the network; directed edges between inner nodes represent the wires. A $b$-balancer has out degree exactly $b$, for some $b \in B$. The size of a network is defined as the total number of inner nodes (i.e., balancers) in the network.

The state of network is defined by the collection of tokens on all wires. A state is quiescent if $\sum_{i=1}^{h} x_i = \sum_{i=1}^{b} y_i$; that is, the number of tokens that entered the network is equal to the number of tokens that left it.

A counting network with fan-out $n$ is a balancing network with fan-out $n$ for which, in any quiescent state, $0 \leq y_i - y_j \leq 1$, for any $i,j$, $1 \leq i < j \leq n$ (this is called the step property). To illustrate this property, consider an execution in which tokens traverse the network sequentially, one after the other. Figure 3 shows such an execution on the COUNTER[6] network, built out of 2-balancers and 3-balancers. As can be seen, the network moves tokens from input wires to output wires in increasing order modulo 6.

A smoothing network with fan-out $n$ is a balancing network for which, in any quiescent state, $|y_i - y_j| \leq 1$ for any $i,j$, $1 \leq i < j \leq n$. That is, in any quiescent state the number of tokens on each output wire is one of two consecutive integer numbers. Clearly, a counting network is also a smoothing network. More generally, for any integer $K \geq 1$, a $K$-smoothing

![Figure 2: A 3-balancer (with fan-in 4).](image-url)
network is a balancing network for which, in any quiescent state, \(|y_i - y_j| \leq k'\), for any \(i, j, 1 \leq i, j \leq n\).

Note that the correctness of smoothing and counting networks is defined only for quiescent states - no guarantee is given for non-quiescent states. However, the unpredictability of inputs implies the following stronger property, which should hold in any state:

Lemma 2.1 Let \(G\) be a smoothing network. If at some state there are at most \(k\) tokens inside \(G\) then, at this state, \(|y_i - y_j| \leq k + 1\), for any \(i, j, 1 \leq i, j \leq n\).

3 Impossibility Result for Acyclic Networks

In this section, we present a necessary condition for constructing acyclic smoothing networks with a specific fan-out using balancers with specific fan-outs.

The following technical lemma shows that, for any balancing network, the number of outputs on a specific wire takes a particular form, depending on the network's size and the fan-out of the balancers used. Let \(G\) be a balancing network with fan-out \(n\) over \(B = \{b_1, \ldots, b_k\}\), of size \(m\). Define \(\ell(G) = n \prod_{i=1}^{m} a_i\), where \(a_i\) is the fan-out of the \(i\)th balancer, \(a_i \in B\); that is, \(\ell(G)\) is the product of the fan-outs of all the balancers in \(G\) and the fan-out of \(G\).

Lemma 3.1 Let \(G\) be a balancing network with fan-out \(n\) over \(B = \{b_1, \ldots, b_k\}\). For any \(i, 1 \leq i \leq n\), there exist non-negative integers \(n_i, e_i, 1, \ldots, e_i, k\), such that for any integer \(c \geq 1\), if \(c \cdot \ell(G)\) tokens are input on the first wire, then in a quiescent state,

\[
y_i = c \cdot \ell(G) \frac{n_i}{\prod_{j=1}^{k} b_j^{e_{i,j}}}.
\]
Proof: The proof is by induction on the size of the network, \( m \). The base case is a network of size 0; that is, a network without any balancers. In this case, \( \ell(G) = n \). Define \( e_{i,j} = 0 \), for all \( 1 \leq i \leq n, 1 \leq j \leq k \); also define \( n_1 = 1 \) and \( n_i = 0 \), for \( i \neq 1 \). For any integer \( c \geq 1 \), if \( c \cdot \ell(G) = c \cdot n \) tokens are input on the first wire, then there are \( c \cdot n \) tokens on the connected output wire and zero tokens on all the other wires, as needed.

For the inductive step, let \( G \) be a balancing network of size \( m + 1 \) over \( B \). Since the network is acyclic, there exists a \( k \)-balancer whose output wires are also outputs of the network, for some \( t, 1 \leq t \leq k \). Let \( h \) be the fan-in of this balancer. Let \( G' \) be the network obtained by removing this balancer. Note that size of \( G' \) is \( m \) (see Figure 4).

For any \( i, 1 \leq i \leq n \), let \( n_i, \ell_i, \ldots, \ell_{ik} \) be the parameters defined for \( G' \). Number the input wires of the \( h \)-balancer by \( i_1, \ldots, i_h \); they are all connected to output wires of \( G' \). For any \( l = 1, \ldots, h \), define \( e_{i_l,t} = 1 + \sum_{r=1}^{k} \ell_{r,t} \) and \( e_{i_l,j} = \sum_{r=1}^{k} \ell_{r,j} \), for any \( j \neq t \); also define

\[
\frac{\sum_{r=1}^{k} n_r \prod_{j=1}^{h} b_j^{e_{r,j}}}{\prod_{j=1}^{h} b_j^{e_{r,t}}} = n_i
\]

Assume \( c \cdot \ell(G) \) tokens are input on the first wire of \( G \). Since \( \ell(G) = b_t \cdot \ell(G') \), it follows that \( c \cdot b_t \cdot \ell(G') \) tokens are input on the first wire of \( G' \). By the inductive hypothesis on \( G' \), in a quiescent state there are exactly

\[
\hat{\gamma}_i = c \cdot b_t \cdot \ell(G') \frac{n_i}{\prod_{j=1}^{h} b_j^{e_{r,j}}}
\]

(1)

tokens on the \( i \)th output wire of \( G' \).
Each output wire of $G$ is either an output wire of $G'$ or an output wire of the $b_t$-balancer. If the $i$th output wire of $G$ is also an output wire of $G'$, the claim is immediate from the induction hypothesis and the definition of $n_i, e_{i,1}, \ldots, e_{i,k}$. So, assume the $i$th output wire of $G$ is an output wire of the $b_t$-balancer.

By definition, if the total number of tokens input to a $b_t$-balancer is a multiple of $b_t$, in a quiescent state, the number of tokens on all output wires is exactly the same. Thus, for any $l, i \leq l \leq b_t, y_{i,l}$, the number of tokens on each of the output wires of the $b_t$-balancer is exactly $1/b_t \sum_{r=1}^{h} \hat{y}_{i,r} = \sum_{r=1}^{h} \hat{y}_{i,r}/b_t$. For each of the terms (by Equation (1)),

$$\hat{y}_{i,r} = \frac{i}{b_t} \cdot c \cdot b_t \cdot \ell(G') \cdot \frac{\hat{n}_{i,r}}{\prod_{j=1}^{k} b_j^{e_{i,j}}} = \frac{c \cdot \ell(G') \cdot \hat{n}_{i,r}}{\prod_{j=1}^{k} b_j^{e_{i,j}}}$$

is an integer number, by the inductive step. Hence, in a quiescent state, the number of tokens on each output wire is exactly

$$y_{i,l} = \sum_{r=1}^{h} \frac{\hat{y}_{i,r}}{b_t} = \sum_{r=1}^{h} c \cdot \ell(G') \cdot \frac{\hat{n}_{i,r}}{b_t \cdot \prod_{j=1}^{k} b_j^{e_{i,j}}} = c \cdot b_t \cdot \ell(G') \sum_{r=1}^{h} \frac{\hat{n}_{i,r}}{b_t \cdot \prod_{j=1}^{k} b_j^{e_{i,j}}}.$$

By the definition of $n_i$ and $e_{i,j}$, we get that

$$y_{i,l} = c \cdot \ell(G') \frac{n_i}{\prod_{j=1}^{k} b_j^{e_{i,j}}}$$

as needed.

The proof of the main impossibility result makes use of the following fundamental theorem (repeated here for completeness).

**Theorem 3.2 (Unique Representation)** Any integer $a$ can be written in exactly one way as a product of the form $a = p_1^{e_1} p_2^{e_2} \cdots p_r^{e_r}$, where $p_1 < p_2 < \cdots < p_r$ are primes, and $e_1, \ldots, e_r$ are positive integers.

To prove the main result of this section, we consider a counting network with fan-out $n$ over $B$. By feeding $\ell(G)$ tokens into the network and applying the previous lemma we get that the number of tokens on each output wire is divisible by a multiplication of powers of the elements in $B$. On the other hand, this number is also divisible by any prime factor of $n$. Using the Unique Representation Theorem, we derive a contradiction. The precise details follow.

**Theorem 3.3** If there exists a prime factor of $n$, $p$, such that $p \mid b$, for all $b \in B$, then there is no acyclic smoothing network with fan-out $n$ over $B$.
Proof: Assume, by way of contradiction, that \( G \) is an acyclic smoothing network with fan-out \( n \) over \( B \). Input \( \ell(G) \) tokens into the network on the first wire (\( \ell(G) \) is defined before Lemma 3.1). By Lemma 3.1, \( y_i \), the number of tokens on the \( i \)th output wire in a quiescent state is

\[
y_i = \frac{\ell(G)}{\prod_{j=1}^{k} b_j^{e_{i,j}}} \cdot \frac{n_i}{n} \cdot \prod_{j=1}^{k} b_j^{e_{i,j}}
\]

(for some non-negative integers \( e_{i,j} \) and \( n_i \)).

Since \( n \not| \ell(G) \) and the network is in a quiescent state, the number of tokens on each output wire is the same and equal to \( \ell(G)/n \). Fix some \( i, 1 \leq i \leq n \), it follows that

\[
y_i = \frac{\ell(G)}{\prod_{j=1}^{k} b_j^{e_{i,j}}} = \frac{\ell(G)}{n} \cdot \prod_{j=1}^{k} b_j^{e_{i,j}}
\]

Therefore,

\[
n_i \cdot n = \prod_{j=1}^{k} b_j^{e_{i,j}}
\]

By assumption, there exists a prime factor \( p \) of \( n \), such that \( p \not| b_i \), for all \( b_i \in B \). By the Unique Representation Theorem, \( n \cdot n_i \) can be uniquely represented as a product of prime powers. Since \( p \) is factor of \( n \) it must appear in this representation. However, \( p \not| b_i \), for all \( b_i \in B \), thus, \( p \) is not a prime factor of the right side of the equality. A contradiction.

Corollary 3.4 Let \( B \) be a set of prime numbers. There exists an acyclic smoothing network with fan-out \( n \) over \( B \), only if \( B \) includes all prime factors of \( n \).

Proof: Assume, by way of contradiction, that there exists a smoothing network with fan-out \( n \) over \( B = \{b_1, \ldots, b_k\} \), and that there is a prime factor \( p \) of \( n \), such that \( p \not| b_i \). By Theorem 3.3, \( p \) must divide some \( b_i \), \( 1 \leq i \leq k \). Since \( p \) and \( b_i \) are prime numbers, this implies that \( p = b_i \). A contradiction.

This implies that only smoothing networks with fan-out \( 2^l \), \( l \geq 1 \), can be constructed for 2-balancers. A more careful look at the proof of Theorem 3.3 allows us to prove a slightly stronger result:

Theorem 3.5 If there exists a prime factor of \( n \), \( p \), such that \( p \not| b \), for all \( b \in B \), then there is no \( K \)-smoothing network with fan-out \( n \) over \( B \), for any \( K \geq 1 \).

Proof: Assume, by way of contradiction that a \( K \)-smoothing network, \( G \), exists. From Lemma 3.1 it follows that for any \( 1 \leq i \leq n \), there exist non-negative integers \( n_i, e_{i,1}, \ldots, e_{i,k} \), such that, if \( \ell(G) \) tokens are input on the first wire of \( G \), then, in a quiescent state, on each output wire there are exactly

\[
y_i = \frac{\ell(G)}{\prod_{j=1}^{k} b_j^{e_{i,j}}} \cdot \frac{n_i}{n} \cdot \prod_{j=1}^{k} b_j^{e_{i,j}}
\]
Using the networks constructed in [2, 3] as the embedded acyclic network, we get an amortized complexity of $O(\log^2 n)$. Recently, Karchmar and Klugerman ([7]) presented a better construction of counting networks with depth $O(\log n \log \log n)$. Employing their network as the embedded acyclic network yields an amortized complexity of $O(\log n \log \log n)$.

5 Concluding Remarks

Using the Unique Representation Theorem, we have shown that an acyclic smoothing network with fan-out $n$ over $B$ does not exist if there is a prime factor of $n$, $p$, such that $p \not| h$, for all $b \in B$. In particular, this implies that if $B$ contains only prime numbers, then there exists a counting network with fan-out $n$ over $B$ only if $B$ contains all prime factors of $n$. Note that in the proof of the impossibility result, all tokens were input on the same wire. This implies that the result holds regardless of the network’s fan-in.

By “feeding back” some output wires of an acyclic counting network with fan-out $2^{\log n}$, we have constructed cyclic networks with fan-out $n$, using only 2-balancers, for any positive $n$. Proving the correctness of this construction requires a new method of arguing about counting networks that are not quiescent.

Our work leaves open several interesting questions. In Section 2, an acyclic counting network with fan-out 6 was built out of 2- and 3-balancers. Is it possible to generalize this construction to build an acyclic counting network with fan-out $p \cdot q$ from $p$- and $q$-balancers? More generally, can an acyclic counting network with fan-out $n$ over $B$ be constructed if $B$ includes all prime factors of $n$? In order to prove our impossibility result we input a large number of tokens into the network. What happens if the network is only required to handle a bounded number of tokens?

Acknowledgments: The authors thank Maurice Herlihy, Ophir Rachman, Nir Shavit and Irit Yadin for helpful discussions and comments.

References


