Notes on Sorting and Counting Networks

(Extended Abstract)

Nikos Hardavellas, *Damianos Karakos* and *Marios Mavronicolas*[*]

Department of Computer Science, University of Crete, and Institute of Computer Science, Foundation of Research and Technology (FORTH), Heraklion 71110, Greece

Abstract. Implementing *counting networks* on shared-memory multi-processor machines often incurs a performance penalty proportional to the *depth* of the networks and the extent to which concurrent processors access the same memory location at the same time. In this work, we examine the dependence of performance on the *width* of the *balancers* used in constructing such networks.

Our main result is a construction of counting networks (and, hence, *sorting networks*) with perfect power width p^k, for any integers $p \geq 2$ and $k \geq 1$. This construction is built on balancers of width p, and generalizes in a novel way the *periodic counting network* of Aspnes, Herlihy and Shavit [3], built on balancers of width 2. This result provides a partial answer to a question of Aharonson and Attiya [1].

The depth of these networks is k^2, thus implying decrease in depth of counting networks through an increase in balancer width. Furthermore, we provide a formal analysis of the performance of our construction as measured by *contention* [8]. Through a novel use of recurrence relations, we show that our counting networks incur a contention of $\Theta(nk^2/p^{k-1})$ in the presence of n concurrent processors. This bound implies a trade-off between depth and contention.

1 Introduction

Counting networks were recently introduced by Aspnes, Herlihy and Shavit [3] as a new approach for solving multi-processor coordination problems that can be expressed as *counting problems*, i.e., problems where processes must cooperate to obtain successive values from a given range. Typical examples of such counting problems are the assignment of successive memory addresses and the robust simulation of an asynchronous algorithm on a shared-memory multi-processor architecture in a way that preserves causality of computation steps (see, e.g., [1, 3]).

Roughly speaking, a *counting network* is a collection of shared variables, called *balancers*, connected through *wires*. On a shared-memory multi-processor machine, a balancer can be implemented by a *Compare&Swap* variable, and a wire can be implemented by a memory address pointer. In more detail, a *p-balancer* can be thought of as a p-input, p-output toggle. When an input appears

[*] Also with Department of Computer Science, University of Cyprus, Nicosia, Cyprus.

on one of its input wires, it takes the output wire to which the toggle is set, and toggles the gate so that the input will leave on the next output wire. If 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 $\lceil m/p \rceil$ will exit on the top output wire, and $\lfloor m/p \rfloor$ will exit on the bottom output wire.

One can connect a collection of balancers to form a *balancing network* much in the same way 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 others. Some of the remaining unconnected input and output wires are the input and output wires, respectively, of the network. Each request for a value corresponds to a traversal of the network by a *token*, starting from any input wire, following the pointer obtained by accessing the first balancer to the next one, 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 that have entered the network on the ith input wire, $0 \leq i \leq w-1$, where w is the *width* of the network. Similarly, y_i denotes the number of tokens that have left the network on the ith output wire, $0 \leq i \leq w-1$. A balancing network of width w is a *counting network* if each time the network becomes free of tokens, i.e., all entering tokens have exited, $0 \leq y_i - y_j \leq 1$, for any i, j, $0 \leq i < j \leq w-1$. A *smoothing network* only guarantees that $|y_i - y_j| \leq 1$, for any i, j, $0 \leq i, j \leq w-1$.

It has been argued quite convincingly [3] that counting networks comprise the most appropriate solution to counting problems in situations where reducing the expected contention and enhancing the concurrency are of primary interest. (Classical solutions based on implementing a *counter* by a single shared *Fetch&Add* variable suffer from serious performance degradation, because of synchronization bottlenecks and high memory contention.) Indeed, counting networks have been successfully used in diverse applications such as implementing a producer/consumer buffer, and barrier synchronization [3].

Unfortunately, all to date known constructions of counting networks have fan-out $p2^k$, for any integers $p,\ k \geq 1$, using 2-balancers or p-balancers [1, 3]. Although, for sorting networks, "padding" can be used to transfer a sorting network of fan-out $n' \geq n$ into a sorting network of fan-out n, a similar idea would not work for counting networks. Hence, to meet width specifications imposed by specific applications, the question arises of providing general constructions of counting networks with arbitrary width, using a fixed set of balancer types.

In this work, we consider a very special case of the previous question, originally posed in [1]. More specifically, we address the problem of constructing counting networks with width p^k, for any integers $p \geq 2$ and $k \geq 1$, using p-balancers. Our main result is a construction that generalizes the *periodic counting network* of Aspnes, Herlihy and Shavit [3], built on 2-balancers. (In turn, this network is based on the *periodic balanced sorting network* of Dowd, Perl, Rudolph and Saks [7].) This is achieved through an appropriate generalization of the combinatorial notions of *chains* and *cochains* (originally introduced in [7] and subsequently used in [3]) to accomodate the use of p-balancers.

This construction implies a reduction in the depth of counting networks through increasing the width of the used balancers. At this point, it is natural to ask whether this depth reduction is not overpaid by an increase in *contention*, the extent to which concurrent processors access the same memory location at the same time. We provide a rigorous analysis of a formal complexity measure of contention, introduced by Dwork, Herlihy and Waarts [8]. We use recurrence relations to show that our construction incurs a contention of $\Theta(nk^2/p^{k-1})$ in the presence of n concurrent processors. This bound implies that decreasing k, hereby reducing depth, may not always result in improving performance, and suggests a trade-off between depth and contention, parameterized by p.

As a case study, we present several new sorting networks of width 9, built on 3-comparators. The balancing duals of these networks, obtained by replacing each 3-comparator with a 3-balancer, are conjectured to be counting networks. Larger counting networks can, however, be obtained from these balancing duals by adjuncting them with smoothing networks of fan-in and fan-out 9, built on 3-balancers. These constructions significantly improve upon previous ones given by Gerbessiotis [9] in both size and depth. More specifically, the sorting networks in [9] use 15 comparators and achieve depth 5, while ours use 12 comparators and achieve depth 4. Consequently, our counting networks of width 9 improve upon corresponding ones in [9]. Furthermore, while the constructions in [9] are quite ad-hoc and result in highly unstructured sorting and counting networks, our networks seem to possess a fine order and structure making them more amenable to generalization.

We believe that the techniques we used in our main construction should be applicable to more general situations where balancers drawn from arbitrary sets of balancer types are available. The ultimate goal of this research is to identify general techniques of designing counting networks, given an arbitrary set of balancers and a width specification, whenever that would be possible (cf. [1]). We feel that our results constitute a modest intermediate step towards this general goal.

As a by-product, the comparison duals of our counting networks of width p^k, $p \geq 2$ and $k \geq 1$, provide the first known family of sorting networks that completely avoids the use of 2-balancers and does not rely on "padding". (See, however, [2] for a discussion of the advantages of using k-comparators in constructing *halvers* and *expanders*, general building blocks of sorting networks, and [14] for a sorting *algorithm* based on three-way comparisons.)

The rest of this paper is organized as follows. In Section 2, we describe balancing, smoothing and counting networks. In Section 3, we present our main construction, while experimental performance results for this construction are outlined in Section 4. Our constructions of sorting, smoothing and counting networks of width 9 are included in Section 5. We conclude, in Section 6, with a discussion of our results and directions for further research.

Due to lack of space, the details of some of our definitions and proofs are omitted.

2 Counting Networks

In this section, we describe smoothing and counting networks; our presentation is patterned after those in [1, 3, 8].

Counting and smoothing networks belong to a larger class of networks called balancing networks, constructed from wires and computing elements called balancers. The construction is similar to that of comparison networks from wires and comparators (see, e.g., [6, 13]). We begin by describing balancing networks.

For each integer $p \geq 2$, a *p-balancer* is a computing element with p input wires and p output wires. Tokens may arrive on any of the p-balancer's input wires and at arbitrary times. Intuitively, a p-balancer is a toggle mechanism, which, given a stream of input tokens, sends them to output wires $0, 1, \ldots, p-1$, in that order and in a cyclic way; thus, a p-balancers effectively balances the number of tokens that have been output on its output wires. Denote by x_i and y_i, $0 \leq i \leq p-1$, the number of input tokens ever received on the p-balancer's ith input wire, and the number of tokens ever sent out on its ith output wire, respectively. Throughout the paper, we will abuse this notation and use x_i (resp., y_i) both as the name of the ith input (resp., output) wire and the count of the number of input (resp., output) tokens received (resp., sent out) on the wire.

Let the *state* of a p-balancer at a given time be defined as the sets of tokens on its input and output wires. A state of a p-balancer is *quiescent* if $\sum_{i=0}^{p-1} x_i = \sum_{i=0}^{p-1} y_i$; that is, the number of tokens that entered the balancer is equal to the number of tokens that left it. The following formal safety, liveness and step properties are required for a p-balancer: (1) In any state, $\sum_{i=0}^{p-1} x_i \geq \sum_{i=0}^{p-1} y_i$ (i.e., a p-balancer never creates output tokens), (2) given any finite number of input tokens $m = \sum_{i=0}^{p-1} x_i$ to the p-balancer, the p-balancer reaches within a finite amount of time a quiescent state (i.e., a p-balancer never "swallows" input tokens); in any such quiescent state, $\sum_{i=0}^{p-1} y_i = m$, and: (3) in any quiescent state, $0 \leq y_i - y_j \leq 1$ for any pair of i and j, $0 \leq i < j \leq p-1$ (i.e., the output has the step property).

Let $P = \{p_1, p_2, \ldots, p_k\}$ be a set of positive integers. A *balancing network of width w over P* is a directed graph with three kind of nodes: *w source* nodes, *w sink* nodes, and some finite number of *inner* nodes. Source nodes represent the input wires and sink nodes represent the output wires. Inner nodes represent the p-balancers of the network; directed edges between inner nodes represent the wires. Each inner node corresponds to a single p-balancer, for some $p \geq 2$, and has in-degree equal to out-degree; call it *degree*. The *size* of a balancing network is defined as the total number of its inner nodes (i.e., balancers). Assume there are m inner nodes and denote by $a_i, i = 1, \ldots, m$, the degree of the ith node. It is required that $a_i \in P$. The *depth of a wire y, $depth(y)$*, is defined to be zero if y is an input wire of the network, and $\max_{i \in [0,p-1]} x_i + 1$, if y is an output wire of a p-balancer with input wires $x_0, x_1, \ldots, x_{p-1}$. The *depth of a balancing network $\mathcal{B}$, $depth(\mathcal{B})$*, is the maximal depth of any wire.

The *state of a balancing network* is defined as the collection of states of all its component balancers. A state of a balancing network is *quiescent* if $\sum_{i=0}^{w-1} x_i =$

$\sum_{i=0}^{w-1} y_i$; that is, the number of tokens that entered the network is equal to the number of tokens that left it. The safety and liveness properties of a balancing network follow naturally from its definition and the safety and liveness properties of a p-balancer: (1) in any state, $\sum_{i=0}^{w-1} x_i \geq \sum_{i=0}^{w-1} y_i$ (i.e., a balancing network never creates output tokens). (2) given any finite number of input tokens $m = \sum_{i=0}^{w-1} x_i$ to the network, the network reaches within a finite amount of time a quiescent state (i.e., a network never "swallows" input tokens). In any such quiescent state, $\sum_{i=0}^{w-1} y_i = m$.

On an MIMD shared-memory multi-processor machine, a balancing network is implemented as a shared data structure, where balancers are records and wires are pointers from one record to another. Each of the machine's n asynchronous processors runs a program that repeatedly traverses the data structure from some input pointer to some output pointer, each time shepherding a new token through the network. Tokens generated by processor p enter the network on input wire $p \bmod w$. The limitation on the number of concurrent processors implies a limitation on the number of tokens concurrently traversing the network at any given time: $\sum_{i=0}^{w-1} x_i - \sum_{i=0}^{w-1} y_i \leq n$.

A *counting network of width w over P* is a balancing network of width w over P for which, in any quiescent state, $0 \leq y_i - y_j \leq 1$, for any i, j, $0 \leq i < j \leq w-1$ (this is the *step property*). A *smoothing network of width w over P* is a balancing network of width w over P for which, in any quiescent state, $|y_i - y_j| \leq 1$ for any i, j, $0 \leq i, j \leq w-1$. 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.

Consider an execution of a counting network entering a quiescent state after m tokens pass through it. Each time a token passes through a balancer, a *stall* is charged to all tokens currently on input wires of the balancer. The *(m,n)-contention of a counting network $\mathcal{N}$*, $\mathcal{C}_{m,n}(\mathcal{N})$, is the worst-case number of stalls charged during an execution of a counting network at concurrency n in which m tokens traverse and finally exit the network. The *amortized n-contention of a counting network $\mathcal{N}$*, $\mathcal{C}_n(\mathcal{N})$, is the (m,n)-contention of $\mathcal{N}$ divided by m.

Denote by $\mathcal{DS}$ the *balancing dual* of a sorting network $\mathcal{S}$, i.e., the isomorphic network obtained from $\mathcal{S}$ by replacing each comparator with a balancer.

3 Periodic Counting Networks with Perfect Power Width

We present a general construction of a counting network PERIODIC$[p^k]$ with perfect power width p^k, for any integers $p \geq 2$ and $k \geq 1$.

We start with definitions for p-chains and p-cochains, patterned after those in [7] modified to accomodate p-balancers.

Consider a sequence $X = x_0, x_1, \ldots, x_{n-1}$. We represent each index (subscript) of a term in X as a *p-ary string*, i.e., a string over $\{0, 1, \ldots, p-1\}$ representing the index in the p-ary arithmetic system. In our discussion, we will use terms, indices, and representations of indices interchangeably.

The *level* 0 *p-chain of* X is X itself. For any integer $i \geq 1$, a *level* i *p-chain of* X is a subsequence of X whose indices have the same i low-order p-ary digits. There are, for each integer $i \geq 0$, p^i level i p-chains of X. For $i \geq 1$, the level i p-chain of X corresponding to the i low-order p-ary digits $p_i p_{i-1} \ldots p_1$ will be denoted as $X^{p_i p_{i-1} \ldots p_1}$. In particular, X^0, X^1, ..., X^{p-1} denote the level 1 p-chains of X, namely, the subsequences of X of terms with indices $0, 1, \ldots, p-1$ modulo p, respectively. Let $\mathcal{C}$ be a set of p-chains of X. The *p-cochain of* X *defined by* $\mathcal{C}$, $X^{\mathcal{C}}$, is the subsequence of X consisting of all terms of p-chains in $\mathcal{C}$. Of special interest are p-cochains of X defined by certain combinations of level 2 p-chains of X. Specifically, let $\mathcal{C}_0 = \{X^{00}, X^{11}, \ldots, X^{p-1,p-1}\}$, $\mathcal{C}_1 = \{X^{01}, X^{12}, \ldots, X^{p-1,0}\}$, through $\mathcal{C}_{p-1} = \{X^{0,p-1}, X^{p-1,p-2}, \ldots, X^{10}\}$. Each of these sets of p-chains of X defines a corresponding p-cochain of X. These p-cochains, denoted by $X^{\mathcal{C}_0}$, $X^{\mathcal{C}_1}$, through $X^{\mathcal{C}_{p-1}}$, will be called the *special p-cochains of* X. Notice that, by definition of special p-cochains, $X^{ij} \in \mathcal{C}_k$ if and only if $j \equiv (i+k) \bmod p$. Our first lemma reveals the close relationship between p-chains and p-cochains.

Lemma 1. *For each integer* $i > 1$, *a subsequence of* X *is a level* i *p-chain of* X *if and only if it is a level* $i-1$ *p-chain of one of the special p-cochains of* X.

The next lemma establishes a trivial step property of level 1 p-chains.

Lemma 2. *Assume* l *is a multiple of* p. *If the sequence* $X = x_0, x_1, \ldots, x_{l-1}$ *has the step property, then the sums of terms in the level 1 p-chains of* x, X^0, X^1, *through* X^{p-1}, *are within 1 of each other.*

Define the balancing network BLOCK$[p^k]$ as follows. When k is equal to 1, the network BLOCK$[p^k]$ consists of a single p-balancer. For larger values of k, the network BLOCK$[p^k]$ is constructed recursively. We start with p networks BLOCK$[p^{k-1}]$ denoted by $\mathcal{N}_0$, $\mathcal{N}_1$, through $\mathcal{N}_{p-1}$. Given an input sequence x, the input to $\mathcal{N}_i$ is $x^{\mathcal{C}_i}$, where $0 \leq i \leq p-1$. Let $y^{\mathcal{C}_i}$ be the output sequence for the network $\mathcal{N}_i$, $0 \leq i \leq p-1$. The final stage of the network BLOCK$[p^k]$ combines each p-tuple $y_j^{\mathcal{C}_0}$, $y_j^{\mathcal{C}_1}$, through $y_j^{\mathcal{C}_{p-1}}$ in a single p-balancer, yielding final outputs z_{pj}, z_{pj+1}, through z_{pj+p-1}, $0 \leq j \leq p^{k-1}-1$. The balancing network PERIODIC$[p^k]$ consists of k balancing networks BLOCK$[p^k]$ joined so that the i-th output wire of one is the i-th input wire of the next, $0 \leq i \leq p^k - 1$. Figure 1 shows the balancing network PERIODIC[9].

We show that PERIODIC$[p^k]$ is a counting network. Our first lemma generalizes Lemma 4.2 in [3] to accomodate p-balancers.

Lemma 3. *Assume each of the sequences* $x^{(0)}$, $x^{(1)}$, *through* $x^{(p-1)}$ *has the step property, and p-tuples* $x_i^{(0)}$, $x_i^{(1)}$, *through* $x_i^{(p-1)}$ *enter a p-balancer yielding outputs* $y_i^{(0)}$, $y_i^{(1)}$ *and* $y_i^{(p-1)}$. *Then, each of the sequences* $y^{(0)}$, $y^{(1)}$, *through* $y^{(p-1)}$ *has the step property.*

We next show:

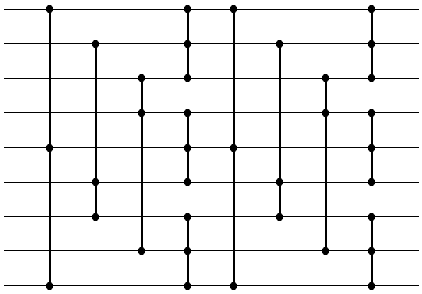


Fig. 1. The balancing network PERIODIC[9]

Lemma 4. *Let* BLOCK$[p^k]$ *be quiescent. If each of* x^0, x^1, *through* x^{p-1} *has the step property, so does* z.

Proof: By induction on k.

For the base case, where $k = 1$, BLOCK$[p^k]$ is a single p-balancer, whose output y is guaranteed, by the definition of a p-balancer, to have the step property.

For the induction step, assume the property for BLOCK$[p^{k-1}]$, $k > 1$, and consider the network BLOCK$[p^k]$. Denote by $y^{\mathcal{C}_l}$ the output sequence of the network $\mathcal{N}_l$, $1 \leq l \leq p-1$, being itself a network BLOCK$[p^{k-1}]$. In the full version of the paper, we show that for each l, $0 \leq l \leq p-1$, $y^{\mathcal{C}_l}$ has the step property, and then we show the step property for z by case analysis. ■

Lemma 5. *Let* BLOCK$[p^k]$ *be quiescent with input sequence* x *and output sequence* y. *For each integer* $i \geq 1$, *if all the level* i p*-chains of* x *have the step property, then so do all the level* $i-1$ p*-chains of* y.

Proof: By induction on i.

The base case, where $i = 1$, is established in Lemma 4. For the induction step, assume the property for input p-chains of level at most i. Let x be the input sequence to BLOCK$[p^k]$, $z^{\mathcal{C}_0}$, $z^{\mathcal{C}_1}$, through $z^{\mathcal{C}_{p-1}}$ the output sequences of the p nested BLOCK$[p^{k-1}]$ networks $\mathcal{N}_0$, $\mathcal{N}_1$, through $\mathcal{N}_{p-1}$, z the concatenation of $z^{\mathcal{C}_0}$, $z^{\mathcal{C}_1}$, through $z^{\mathcal{C}_{p-1}}$, and y the output sequence of BLOCK$[p^k]$.

For $i > 1$, Lemma 1 implies that every level i p-chain of x is entirely contained in one of the special p-cochains $x^{\mathcal{C}_0}$, $x^{\mathcal{C}_1}$, through $x^{\mathcal{C}_{p-1}}$ of x. Consider, without loss of generality, any level i p-chain of x entirely contained in $x^{\mathcal{C}_0}$. By Lemma 1, any such p-chain is a level $i-1$ p-chain of $x^{\mathcal{A}}$, and has, by hypothesis, the step property. Hence, by induction hypothesis, all level $i-2$ chains of $z^{\mathcal{C}_0}$ have the step property. Similarly, all level $i-2$ p-chains of $z^{\mathcal{C}_1}$, through $z^{\mathcal{C}_{p-1}}$ have the step property. It follows, by Lemma 1, that all level $i-1$ p-chains of z have the step property. Hence, Lemma 3 implies that all level $i-1$ p-chains of y have the step property, as needed. ■

As an immediate consequence of Lemma 5, it follows:

Theorem 6. PERIODIC$[p^k]$ *is a counting network.*

We remark that the special p-cochains $x^{\mathcal{C}_0}$, through $x^{\mathcal{C}_{p-1}}$ could be defined as, e.g., $\mathcal{C}_0 = \{x^{0\,0}, x^{1\,p-1}, \ldots, x^{p-1\,1}\}$, through $\mathcal{C}_{p-1} = \{x^{0\,p-1}, x^{1\,p-2}, \ldots, x^{p-1\,0}\}$, while still preserving the shown properties of p-chains and p-cochains, and lead to an isomorphic construction. (The reader is encouraged to verify this.) Becker, Nassimi and Pearl [5] present interesting generalizations of special 2-cochains, calling them *g-chains,* and use them to construct large classes of sorting networks out of 2-balancers. It is an important combinatorial question to collect *all* generalizations of special p-cochains, for any integer $p \geq 2$, in a systematic way, and use them to obtain new (periodic) counting networks with perfect power width p^k, $k \geq 2$. (In fact, it may be that many of our networks of width 9, presented in section 5, are just instances of such generalizations.)

For each $k \geq 1$, let $depth(\text{BLOCK}[p^k])$ and $depth(\text{PERIODIC}[p^k])$ denote the depths of the networks BLOCK$[p^k]$ and PERIODIC$[p^k]$, respectively. For $k \geq 2$,

$$depth(\text{PERIODIC}[p^k]) \;=\; k\; depth(\text{BLOCK}[p^k]) \;=\; k(depth(\text{BLOCK}[p^{k-1}]) + 1)\;,$$

with $depth(\text{PERIODIC}[p]) = depth(\text{BLOCK}[p]) = 1$. Hence, it follows:

Theorem 7. $depth(\text{PERIODIC}[p^k]) \;=\; k^2$, $k \geq 1$.

We proceed to show a tight contention bound for PERIODIC$[p^k]$.

Theorem 8. $\mathcal{C}_n(\text{PERIODIC}[p^k]) \;\leq\; (\frac{n}{p^{k-1}} - 1)k^2$

Proof: Since the PERIODIC$[p^k]$ network consists of k cascaded BLOCK$[p^k]$ networks, we have:

$$\mathcal{C}_n(\text{PERIODIC}[p^k]) \;\leq\; k \cdot \mathcal{C}_n(\text{BLOCK}[p^k])\;.$$

Hence, it suffices to prove:

Lemma 9. $\mathcal{C}_{m,n}(\text{BLOCK}[p^k]) \;\leq\; (\frac{n}{p^{k-1}} - 1)km$

Proof: Let LADDER$[p^k]$ be the largest leftmost subnetwork of BLOCK$[p^k]$ that has depth 1.

Observe that BLOCK$[p^k]$ consists of LADDER$[p^k]$ followed by p "parallel" BLOCK$[p^{k-1}]$ networks. Notice that, by construction, each of the p-balancers in LADDER$[p^k]$ equidistributes tokens in each of the BLOCK$[p^{k-1}]$ networks; hence, the concurrency of each of these networks is n/p. Hence, it follows:

$$\mathcal{C}_{m,n}(\text{BLOCK}[p^k]) \;=\; \mathcal{C}_{m,n}(\text{LADDER}[p^k]) \;+\; \sum_{i=0}^{p-1} \mathcal{C}_{m_i,\frac{n}{p}}(\text{BLOCK}[p^{k-1}])\;,$$

where m_i is the number of tokens fed through the ith BLOCK$[p^{k-1}]$ subnetwork, $0 \leq i \leq p-1$, out of the m tokens fed through BLOCK$[p^k]$; clearly, $\sum_{i=1}^{p} m_i = m$.

We start by showing:

Claim 10. $\mathcal{C}_{m,n}(\text{LADDER}[p^k]) \leq (\frac{n}{p^{k-1}} - 1)m$

Proof: Notice that LADDER$[p^k]$ consists of p^{k-1} "independent" p-balancers with concurrency $\frac{n}{p^k}$ per input line, and, hence, total concurrency $\frac{n}{p^{k-1}}$. Each token passing through one of the p-balancers may cause installation of up to $\frac{n}{p^{k-1}} - 1$ stalls. Hence, the total number of stalls due to m tokens is at most $(\frac{n}{p^{k-1}} - 1)m$. ■

Claim 10 implies:

$$\mathcal{C}_{m,n}(\text{BLOCK}[p^k]) \leq (\frac{n}{p^{k-1}} - 1)m + \sum_{i=0}^{p-1} \mathcal{C}_{m_i, \frac{n}{p}}(\text{BLOCK}[p^{k-1}])$$

By iteration, it follows:

$$\begin{aligned}
& \mathcal{C}_{m,n}(\text{BLOCK}[p^k]) \\
\leq\ & (\frac{n}{p^{k-1}} - 1)m + \sum_{i=0}^{p-1}((\frac{\frac{n}{p}}{p^{k-2}} - 1)m_i + \sum_{i=0}^{p-1} \mathcal{C}_{m_{ii}, \frac{n}{p^2}}(\text{BLOCK}[p^{k-2}]) \\
=\ & (\frac{n}{p^{k-1}} - 1)m + (\frac{n}{p^{k-1}} - 1)\sum_{i=0}^{p-1} m_i + \sum_{i=0}^{p-1}\sum_{i=0}^{p-1} \mathcal{C}_{m_{ii}, \frac{n}{p^2}}(\text{BLOCK}[p^{k-2}]) \\
=\ & (\frac{n}{p^{k-1}} - 1)2m + \sum_{i=0}^{p-1}\sum_{i=0}^{p-1} \mathcal{C}_{m_{ii}, \frac{n}{p^2}}(\text{BLOCK}[p^{k-2}]) \\
\leq\ & \cdots \\
\leq\ & (\frac{n}{p^{k-1}} - 1)(k-1)m + \underbrace{\sum_{i=0}^{p-1}\sum_{i=0}^{p-1}\cdots\sum_{i=0}^{p-1}}_{k-1 \text{ times}} \mathcal{C}_{m_{(k-1)i}, \frac{n}{p^{k-1}}}(\text{BLOCK}[p])
\end{aligned}$$

Note, however, that $\mathcal{C}_{m_{(k-1)i}, \frac{n}{p^{k-1}}}(\text{BLOCK}[p])$ is the contention due to $m_{(k-1)i}$ tokens traversing a p-balancer with concurrency $\frac{n}{p^{k-1}}$. Arguing as in the proof of Claim 10,

$$\mathcal{C}_{m_{(k-1)i}, \frac{n}{p^{k-1}}}(\text{BLOCK}[p]) \leq (\frac{\frac{n}{p^{k-1}}}{p^{1-1}} - 1)m_{(k-1)i} = (\frac{n}{p^{k-1}} - 1)m_{(k-1)i}$$

Hence,

$$\begin{aligned}
\mathcal{C}_{m,n}(\text{BLOCK}[p^k]) &\leq (\frac{n}{p^{k-1}} - 1)(k-1)m + (\frac{n}{p^{k-1}} - 1)\underbrace{\sum_{i=0}^{p-1}\sum_{i=0}^{p-1}\cdots\sum_{i=0}^{p-1}}_{k-1 \text{ times}} m_{(k-1)i} \\
&= (\frac{n}{p^{k-1}} - 1)km\ ,
\end{aligned}$$

as needed. ■

Hence,

$$\mathcal{C}_n(\text{PERIODIC}[p^k]) \ \leq \ (\frac{n}{p^{k-1}} - 1)k^2 \ ,$$

as needed ■

In the full version of the paper, we show that the worst-case contention bound shown in Theorem 8 is tight by describing an execution, in which tokens traverse "sequentially" the sequence of the k cascaded BLOCK$[p^k]$ networks, attaining this bound.

4 Performance

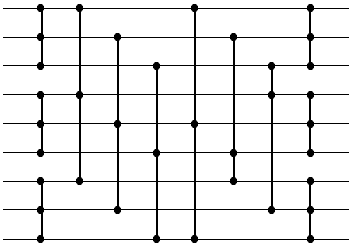
We implemented a software simulation of periodic counting networks in a general asynchronous multiprocessor. Using this simulation, we investigated the dependence of the *typical* amortized contention of these networks on factors such as the width of the used balancers, the depth of the network and the concurrency (number of processors).

This simulation helped us evaluate the performance of periodic counting networks as measured by contention. Throughout, our experimental results verified the linear dependence of contention on concurrency, formally shown in section 3. In more detail, maintaining constant depth, we observed improvement in performance under an increase in the balancer's width. This improvement should provide an excellent testbed for the comparison of our constructions to corresponding existing ones built up solely of 2-balancers. Furthermore, maintaining constant balancer's width, we observed corresponding improvement under an increase in depth. Most important, we observed a significant deviation of typical (contention) performance from the worst-case one following from our contention analysis: typical contention was found to be only about 10% of the worst-case one! Typical experimental performance results appear in Figure 2.

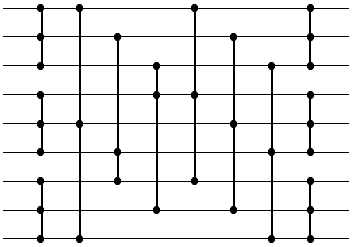
Concurrency	3	6	9	12	15	18	21	24	27	30	33	36
PERIODIC[4]	0.59	1.84	3.09	4.43	5.87	7.33	8.58	10.04	11.16	12.37	13.85	15.31
PERIODIC[8]	0.35	0.75	1.19	1.73	2.21	2.91	3.51	4.25	4.78	5.40	6.24	6.74
PERIODIC[9]	0.79	1.15	1.81	2.85	3.65	4.50	5.47	6.42	7.11	8.00	8.86	9.83
PERIODIC[27]	0.27	0.70	0.98	1.34	1.63	1.89	2.07	2.28	2.54	2.77	2.96	3.14
Concurrency	42	45	48	51	54	57	60	63	66	69	72	75
PERIODIC[4]	17.66	19.17	20.60	21.22	22.66	23.16	24.64	25.43	27.37	28.42	29.03	30.19
PERIODIC[8]	8.24	8.76	9.66	10.21	10.87	11.53	12.07	13.02	13.61	14.10	14.86	15.49
PERIODIC[9]	11.62	12.11	13.02	13.85	15.04	15.67	16.48	17.34	18.29	18.05	20.03	20.49
PERIODIC[27]	3.54	3.68	3.96	4.18	4.30	4.64	4.77	5.07	5.21	5.44	5.80	5.94

Fig. 2. Experimental values for typical amortized contention

Due to lack of space, a complete description of our simulation methodologies and discussion of the results is deferred to the full version of the paper.



(a) $\mathcal{S}^1_{9\times 9}$



(b) $\mathcal{S}^2_{9\times 9}$

Fig. 3. The sorting networks $\mathcal{S}^1_{9\times 9}$ and $\mathcal{S}^2_{9\times 9}$

5 Sorting and Counting Networks with Width 9

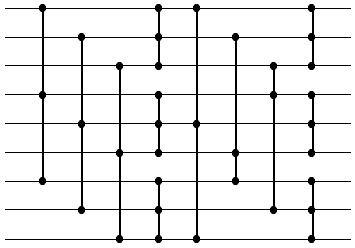
We present constructions for 5 new sorting networks of width 9, $\mathcal{S}^1_{9\times 9}$, through $\mathcal{S}^5_{9\times 9}$. These networks are depicted in Figures 3, and 4. Our constructions result from systematically permuting small "smooth-like" building blocks of width 9.

Theorem 11. *The comparison networks $\mathcal{S}^1_{9\times 9}$, $\mathcal{S}^2_{9\times 9}$, $\mathcal{S}^3_{9\times 9}$, $\mathcal{S}^4_{9\times 9}$ and $\mathcal{S}^5_{9\times 9}$ are sorting networks.*

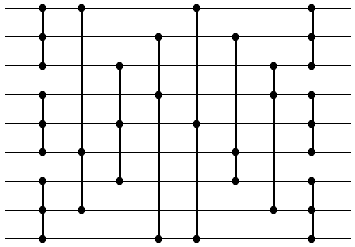
Proof: It has been verified by using a computer program that each of these networks sorts all Boolean sequences of length 9. The result follows from the 0-1 sorting principle [6, 13]. ■

Notice that each of $\mathcal{S}^1_{9\times 9}$ through $\mathcal{S}^5_{9\times 9}$ achieves depth 4 and size 12. In fact, $\mathcal{S}^3_{9\times 9}$ can be modified to achieve size 11 by eliminating the (redundant) 7th comparator from left to right. At the time of writing, we do not know whether or not 11 can be further reduced. We have run several simulation tests by feeding tokens into randomly selected input wires of the balancing dual of each of $\mathcal{S}^1_{9\times 9}$ through $\mathcal{S}^5_{9\times 9}$. None of these tests violated the counting property. This leads us to formulate the following conjecture.

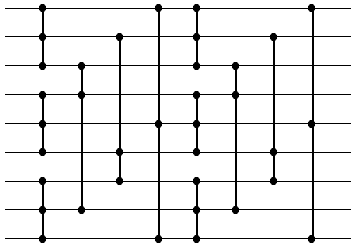
Conjecture 12. The networks $\mathcal{DS}^i_{9\times 9}$, $1 \leq i \leq 5$, are counting networks.



(a) $\mathcal{S}^3_{9\times 9}$



(b) $\mathcal{S}^4_{9\times 9}$



(c) $\mathcal{S}^5_{9\times 9}$

Fig. 4. The sorting networks $\mathcal{S}^3_{9\times 9}$, $\mathcal{S}^4_{9\times 9}$ and $\mathcal{S}^5_{9\times 9}$

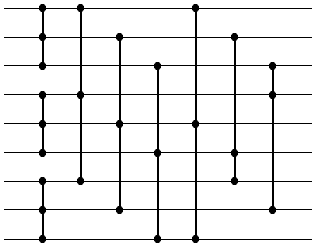
The balancing network $\mathcal{M}^1_{9\times 9}$, depicted in Figure 5(a), has been introduced by Gerbessiotis [9].

Lemma 13 (Gerbessiotis, [9]). *$\mathcal{M}^1_{9\times 9}$ is a smoothing network.*

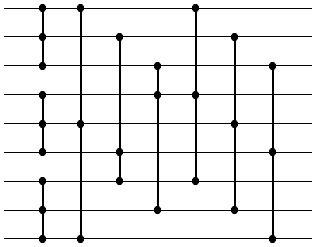
Since, as observed in [3], *smoothing + sorting = counting,* it immediately follows:

Theorem 14. *The networks $\mathcal{M}^1_{9\times 9}\mathcal{DS}^i_{9\times 9}$, $1 \leq i \leq 5$, are counting networks.*

We slightly permute the smoothing network $\mathcal{M}^1_{9\times 9}$ to obtain a balancing network $\mathcal{M}^2_{9\times 9}$, depicted in Figure 5(b). In the full version of the paper, we



(a) $\mathcal{M}^1_{9\times 9}$



(b) $\mathcal{M}^2_{9\times 9}$

Fig. 5. The smoothing networks $\mathcal{M}^1_{9\times 9}$ and $\mathcal{M}^2_{9\times 9}$

show:

Theorem 15. *The network $\mathcal{M}^2_{9\times 9}$ is a smoothing network.*

Hence, it follows:

Theorem 16. *The networks $\mathcal{M}^2_{9\times 9}\mathcal{DS}^i_{9\times 9}$, $1 \leq i \leq 5$, are counting networks.*

6 Discussion and Directions for Further Research

Generalizing the periodic counting network of Aspnes, Herlihy and Shavit [3], we presented a general construction of a counting network with width p^k, for any integers $p \geq 2$ and $k \geq 1$, using p-balancers. Our construction relies heavily on an appropriate generalization of the combinatorial notions of chains and cochains. This construction enhances the class of currently known constructible counting networks.

Through a rigorous analysis based on recurrence relations, we revealed an interesting trade-off between depth and contention incurred by our construction. We verified and further evaluated our theoretical bounds on contention by experimental means.

We also presented size and depth economical sorting networks with width 9, built on 3-comparators, and associated counting networks, built on 3-balancers, improving upon previous constructions of Gerbessiotis [9].

Our work leaves open several interesting questions. Can the *bitonic counting network* in [3] be generalized to accomodate p-balancers and achieve width p^k, $k \geq 1$? How about the "small-depth" counting networks in [12]? Aharonson and Attiya [1] present techniques for constructing counting networks of fan-in and fan-out $q2^k$, for any integers $q, k \geq 1$, using 2-balancers and q-balancers. Can their techniques be combined with our construction to yield a counting network of width qp^k, for any integers $p \geq 2$ and $q, k \geq 1$? More generally, the following question of Aharonson and Attiya [1] remains as yet unsettled: For any integers p and q, $p,\ q \geq 2$, how can a counting network of width pq be built from p-balancers and q-balancers? We are currently investigating the possibility of building such networks using our periodic constructions as main building blocks, and we have obtained some preliminary results.

We believe that the use of recurrence relations for analyzing contention, initiated by our work, is quite promising and should be applicable elsewhere. Can the contention bound for the bitonic counting network shown in [8] be obtained through formulating and solving an appropriate recurrence relation? Furthermore, the contention bound for the periodic counting network has been shown to be tight for every $p \geq 2$. (In [8], a corresponding bound for the special case $p = 2$ is reported.) Dwork, Herlihy and Waarts [8] suggest the study of contention of counting networks in general (as opposed to studying the contention of specific constructions). Does there exist a counting network achieving a better trade-off between depth and contention than ours?

Counting networks have recently attracted a lot of flourishing interest and attention [1, 3, 8, 10, 11, 12]. Our results further the applicability of the currently known techniques for constructing and analyzing the performance of such networks. Clearly, we have barely scratched the surface of this rich subject, with the vast bulk of the interesting results remaining yet to be discovered.

Acknowledgments: We are grateful to Hagit Attiya for many helpful suggestions and comments, and, especially, for raising questions on contention.

References

1. E. Aharonson and H. Attiya, "Counting Networks with Arbitrary Fan-Out," in *Proceedings of the 3rd ACM–SIAM Symposium on Discrete Algorithms,* pp. 104–113, January 1992.
2. M. Ajtai, J. Komlos and E. Szemeredi, "Halvers and Expanders," in *Proceedings of the 33rd Annual IEEE Symposium on Foundations of Computer Science,* pp. 686–692, October 1992.
3. J. Aspnes, M. Herlihy and N. Shavit, "Counting Networks and Multi-Processor Coordination," in *Proceedings of the 23rd Annual ACM Symposium on Theory of Computing,* pp. 348–358, May 1991. Expanded version: "Counting Networks," Technical Memo MIT/LCS/TM-451, Laboratory of Computer Science, MIT, June 1991.
4. K. E. Batcher, "Sorting Networks and their Applications," in *Proceedings of AFIPS Joint Computer Conference,* 32, pp. 338–334, 1968.
5. R. Becker, D. Nassimi and Y. Perl, "The New Class of g-Chain Periodic Sorters," in *Proceedings of the 5th Annual ACM Symposium on Parallel Algorithms and Architectures,* July 1993.
6. T. Cormen, C. Leiserson and R. Rivest, *Introduction to Algorithms,* Mc-Graw Hill and MIT Press, 1990.
7. M. Dowd, Y. Perl, L. Rudolph and M. Saks, "The Periodic Balanced Sorting Network," *Journal of the ACM,* Vol. 36, No. 4, pp. 738-757, October 1989.
8. C. Dwork, M. Herlihy and O. Waarts, "Contention in Shared Memory Algorithms," in *Proceedings of the 25th Annual ACM Symposium on Theory of Computing,* May 1993.
9. A. Gerbessiotis, "Sorting and Counting Networks," unpublished manuscript, Harvard University, October 1992.
10. M. Herlihy, B.-C. Lim and N. Shavit, "Low Contention Load Balancing on Large-Scale Multiprocessors," in *Proceedings of the 4th Annual ACM Symposium on Parallel Algorithms and Architectures,* July 1992.
11. M. Herlihy, N. Shavit and O. Waarts, "Low Contention Linearizable Counting Networks," in *Proceedings of the 32nd Annual IEEE Symposium on Foundations of Computer Science,* pp. 526–535, October 1991.
12. M. Klugerman and C. Plaxton, "Small-Depth Counting Networks," in *Proceedings of the 24th Annual ACM Symposium on Theory of Computing,* pp. 417–428, May 1992.
13. D. Knuth, *Sorting and Searching,* Volume 3 of *The Art of Computer Programming,* Addison-Wesley, 1973.
14. S. S. Tseng and R. C. Lee, "A new Parallel Sorting Algorithm Based upon Min-Mid-Max Operations," *BIT,* Vol. 24, pp. 187–195, 1984.

This article was processed using the LaTeX macro package with LLNCS style