Computing in Additive Networks with Bounded-Information Codes

Keren Censor-Hillel, Erez Kantor, Nancy Lynch, Merav Parter

Computing in Additive Networks with Bounded-Information Codes *

Keren Censor-Hillel[1], Erez Kantor[2], Nancy Lynch[2], and Merav Parter[2]

[1] Department of Computer Science, Technion, Haifa 32000, Israel
[2] CSAIL, Massachusetts Institute of Technology, MA 01239, USA

Abstract. This paper studies the theory of the additive wireless network model, in which the received signal is abstracted as an addition of the transmitted signals. Our central observation is that the crucial challenge for computing in this model is not high contention, as assumed previously, but rather guaranteeing a bounded amount of *information* in each neighborhood per round, a property that we show is achievable using a new random coding technique. Technically, we provide efficient algorithms for fundamental distributed tasks in additive networks, such as solving various symmetry breaking problems, approximating network parameters, and solving an *asymmetry revealing* problem such as computing a maximal input. The key method used is a novel random coding technique that allows a node to successfully decode the received information, as long as it does not contain too many distinct values. We then design our algorithms to produce a limited amount of information in each neighborhood in order to leverage our enriched toolbox for computing in additive networks.

1 Introduction

The main challenge in wireless communication is the possibility of collisions, occurring when two nearby stations transmit at the same time. In general, collisions provide no information on the data, and in some cases may not even be distinguishable from the case of no transmission at all. Indeed, the ability to merely detect collisions (a.k.a., the collision detection model) gives additional power to wireless networks, and separation results are known (e.g., [26]).

Traditional approaches for dealing with interference (e.g., FDMA, TDMA) treat collisions as something that should be avoided or at least minimized [12, 21, 23]. However, modern coding techniques suggest the ability to *retrieve information* from collisions. These techniques significantly change the notion of collisions, which now depends on the model or coding technique used. For example, in *interference cancellation* [2], the receivers may decode interfering signals

* The first author is supported in part by the Israel Science Foundation (grant 1696/14). The last three authors are supported in a part by NSF Award Numbers CCF-1217506, CCF-AF-0937274, 0939370-CCF, and AFOSR Contract Numbers FA9550-14-1-0403 and FA9550-13-1-0042. Merav Parter is also supported by Rothschild and Fulbright Fellowships.

that are sufficiently strong and *cancel* them from the received signal in order to decode their intended message. Hence, from this viewpoint, collision occurs only when neither the desired signal nor the the interfering signal are relatively strong enough.

In this paper, we consider the *additive network model*, in which colliding signals add up at the receiver and are hence *informative* in some cases. It has been shown that such models approximate the capacity of networks with high signal-to-noise ratio [3], and that they can be useful in these settings for various coding techniques, such as ZigZag decoding [11, 22], and bounded-contention coding [6]. While in practice there are limitations for implementing such networks to the full extent of the model, the above previous research shows the importance of understanding the fundamental strength of models that allow the possibility of extracting information out of collisions. In a recent theoretical work [6], the problems of local and global broadcast have been addressed in additive networks, under the assumption that the contention in the system is *bounded*.

The central observation of this paper is that in order to leverage the additive behavior of the system, what needs to be bounded is not necessarily the contention, but rather the total amount of *information* a node has to process at a given round. This observation allows us to extend the quantification of the computational power of the additive network model in solving distributed tasks way beyond local and global broadcast. Our key approach in this paper is *not to assume* a bound on the initial number of pieces of information in the system, but rather *guarantee* a bound on the number of *distinct* pieces of information in a neighborhood of every vertex. We then use a new random coding technique, which we refer to as *Bounded-Information Codes (BIC)*, in order to extract the information out of the received signals. This allows us to efficiently solve various cornerstone distributed tasks.

1.1 Contributions and Methods

On the technical side, we provide efficient algorithms for fundamental *symmetry breaking* tasks, such as leader election, and computing a BFS tree and a maximal independent set (MIS), as well as algorithms for *revealing asymmetry* in the inputs, such as computing the maximum. We also provide efficient algorithms for approximating network parameters by a constant factor. Our key methods are based on enriching the toolbox for computing in additive networks with various primitives that leverage the additive behavior of received information and our coding technique.

Main techniques: The work in [6] introduced Bounded-Contention Codes (BCC) as the main technique. BCC allows the decoding of the XOR of any collection of at most a codewords, where a is the bound on the contention. As mentioned, our key approach in this paper is not to assume a bound on the contention, but rather to make sure that the amount of distinct information colliding at a node at a given round is limited. Our main ingredient is augmenting the deterministic BCC codes with randomization, resulting in Bounded-Information Codes. BIC

allows successful decoding of any transmission of n nodes sending at most $O(a)$ distinct values altogether, with high probability.

Randomization plays a key role in the presented scheme in two different aspects. First, the drawback of the standard BCC code is that the transmission of the *same* message by an even number of neighbors is cancelled out. By increasing the message size by factor of $O(\log n)$ and using randomization, BIC codes add random "noise" to the original BCC codeword so that the probability that two BIC messages cause cancellation becomes negligible.

Another useful aspect of randomization is intimately related to the fact that our information bounds are logarithmic in n. This allows for a win-win situation: if the number of distinct pieces of information (in a given neighborhood) is small (i.e., $O(\log n)$), the decoding is successful thanks to the BIC codes. On the other hand, if the number of distinct pieces of information is large (i.e., $\Omega(\log n)$), there are sufficiently many transmitting vertices in the neighborhood which allows one to obtain good concentration bounds by, e.g., using Chernoff bounds (for example, in estimating various network parameters). It is noteworthy that our estimation technique bares some similarity to the well-known *decay strategy* [4] which is widely used in radio-networks. The key distinction between the long line of works that apply this scheme and this paper is the dimension to which this strategy in applied. Whereas so-far, the strategy was applied to the *time* axis (e.g., in round i, vertex u transmits with probability 2^{-i}), here it is applied to the *information* (or message) axis (e.g., vertex u writes the specific information in the i'th block of its message with probability 2^{-i}). This highly improves the time bounds compared to the basic radio model (i.e., the statistics are collected over the multiple blocks of the message instead of over multiple slots).

An immediate application of BIC is a simple logarithmic simulation of algorithms for networks that employ full-duplex radios (where a node can transmit and receive concurrently) by nodes who have only half-duplex radios (where a node either transmits or receives in a given round). This allows us to consider algorithms for the stronger model of full-duplex radios and obtain a translation to half-duplex radios, and also allows us to compare our algorithms to a message-passing setting. To make justice with such comparisons, we note that a message-passing setting not only does not suffer from collisions, but also is in some sense similar to having full duplex, as a node receives and sends information in the same round.

Note that in the standard radio model, collision detection is not an integral part of the model but rather an external capability that can be chosen to be added. In BIC, collision detection is an integral part of the model, where *collision* now refers to the situation where the number of distinct pieces of information exceeds the allowed bound. To avoid confusion, the collision detection in the context of BIC, is hereafter referred to as *information-overflow detection*. We show that information-overflow can be detected while inspecting the received codeword, without the need for any additional mechanisms.

Symmetry breaking: The first type of algorithms we devise are for various symmetry breaking tasks. The main tool in this context is the *select-level* function,

$\mathcal{SL}$, that outputs two random values according to a predefined distribution. Every vertex v computes the $\mathcal{SL}$ function locally, without any communication. The power of this function lies in its ability to assign random *levels* to nodes, such that with high probability[3] the maximal level contains at most a logarithmic number of nodes (i.e., below the information bound of the BIC code), and the nodes in the maximal level have different values for their second random variable.

The $\mathcal{SL}$ function allows us to elect a leader in $O(D)$ rounds, w.h.p., where D is the diameter of the network. The elected leader is the node with the maximal pair of values chosen by the $\mathcal{SL}$ function. A by-product of this algorithm is a 2-approximation of the diameter, and the analysis is done over a BFS tree rooted at the leader. We also show how to construct a BFS tree rooted at an arbitrary given node in $O(D)$ rounds, w.h.p, by employing both the $\mathcal{SL}$ function and BIC.

Apart from the above new algorithms, our framework allows relatively simple translations of known algorithms for solving various tasks in message passing systems into additive networks. This includes Luby's MIS algorithm [18], Schneider and Wattenhofer's coloring algorithm [24], and approximating the minimum dominating set of Wattenhofer and Kuhn [15], improving significantly over the known bounds for standard radio-model. We give a flavor of these translations by providing the full MIS algorithm and analysis in [7], and sketch the results for coloring and approximating the minimum dominating set.

Approximations: We design algorithms for approximating various network parameters. We show how to compute a constant approximation of the degree of a node, as well as a constant approximation of the size and diameter of the network. (Our coding scheme only requires nodes to know a polynomial bound N on the network size n.) Our algorithms naturally extend to solve the more general tasks of local-sum and global-sum approximations[4] that have been recently considered in [17]. Yet, the additive setting allows us to obtain much better bounds than those of [17].

Asymmetry revealing: In addition to the above symmetry breaking algorithms, we show that additive networks also allow for fast solutions for tasks which do not require symmetry breaking, but rather already begin with inputs whose asymmetry needs to be revealed: we give an algorithm that computes the *exact* maximal value of all inputs in the network in $O(D \cdot \log n / \log\log n)$ rounds, w.h.p. (in contrast, a 2-approximation for the maximal value can be computed within $\Theta(D)$ rounds). We obtain this because our coding scheme allows us to perform a tournament at a high rate. For example, for single-hop networks, in each round only a $O(\log n)$ fraction of the remaining competing vertices survive for the next round.

In some sense, asymmetry revealing can be viewed as the counterpart of symmetry breaking. Clearly, if we compute the maximal input in the system then

[3] We use the term *with high probability* (w.h.p.) to denote a probability of at least $1 - 1/n^c$ for a constant $c \geq 1$.

[4] These are generalizations of degree-approximation and network-size approximation, respectively.

we can obtain a leader as a by-product. However, the opposite does not hold, and indeed in our leader-election algorithm mentioned above we significantly exploit the fact that the leader need not be predetermined, and use our new toolbox to obtain a leader within only $O(D)$ rounds.

1.2 Comparison with Related Work

First, we compare our results with previous theoretical work on the additive network model. The work of [6] assumes a bound a on the contention in the system, i.e., there are at most a initial inputs in total in the network. The main method for obtaining global broadcast in the above work is random linear network coding, which can be shown to allow an efficient flow of information in the system. However, this is what requires the bound on the contention. Our BIC coding method bares some technical similarity to the approach of random linear network coding, but allows us to refrain from making assumptions on the total information present in the network.

The aforementioned global broadcast algorithm requires $O(D + a + \log n)$ rounds. While this algorithm can be used to solve many of the problems that we address in this paper, such as electing a leader and computing the maximal input, it would require $O(n)$ rounds, as for these problems it holds that a can be as large as the total number of nodes in the network. In comparison, our $O(D)$-round leader election algorithm is optimal, and our $O(D \log n / \log \log n)$-round algorithm for computing the maximal input is nearly-optimal, as $O(D)$ is a natural lower bound for both problems, even in the message-passing model.

It is important to mention that our algorithms use messages of size $O(\log^3 n)$. While a standard assumption might be that the message size is $O(\log n)$ bits, this difference is far from rendering our results easy. In comparison, the global broadcast algorithm of [6] requires a message size of $O(a \log n + \ell)$ bits for inputs of size ℓ and contention bounded by a. In our setting, we assume ℓ fits the message size (say, is logarithmic in n), but since a can be as large as n, such a message size would be unacceptable. In addition, if we compare our results to algorithms for the much less restricted message-passing setting, it is crucial to note that even unbounded message sizes do not make distributed tasks trivial. For example, it is possible to compute an MIS in general graphs in $O(\log n)$ rounds even with messages of size $O(1)$ [19], but the best known lower bound is $\Omega(\log \Delta + \sqrt{\log n})$ even with unbounded messages [14]. Recently, Barenboim at el. [5] showed a randomized MIS algorithm with $O(\log \Delta \cdot \sqrt{\log n})$ rounds using unbounded messages.

In [7], we overview results that address the same tasks as this paper in the standard radio network model and in the message-passing model. An additive network can be viewed as lying somewhere in between these two models, as it does suffer from collisions, but to a smaller extent. Nevertheless, while our coding methods assist us in overcoming collisions, the additive network model is still subject to the broadcast nature of the transmissions, and therefore it is highly non-trivial to translate algorithms for the message-passing setting that make use of the ability to send different messages on different links concurrently.

The related work overviewed in [7], include algorithms and lower bounds for various problems in radio networks, such as the wake-up problem [9], MIS with and without collision detection [20,26] or with multiple channels [8], leader election [10], and approximation of local parameters [17], as well as MIS algorithms for message passing systems [1,18,25] and lower bounds [14,16].

2 Background: Additive Networks and BCC

The Additive Network Model: A *radio network* consists of stations that can transmit and receive information. We address a synchronous system, in which in each round of communication each station can either transmit or listen to other transmissions. This is called the half-duplex mode of operation. Mainly due to theoretical interest, we also consider the full-duplex mode of operation which is considered harder to implement. We follow the standard abstraction in which stations are modeled as nodes of a graph $G = (V, E)$, with edges connecting nodes that can receive each other's transmissions.

In the standard radio network model, a node $v \in V$ receives a message m in a given round if and only if in that round exactly one of its neighbors transmits, and its transmitted message is m. In the half-duplex mode, it also needs to hold that v is listening in that round, and not transmitting. If none of v's neighbors transmit then v hears silence, and if at least two of v's neighbors transmit simultaneously then a *collision* occurs at v. In both cases, v does not receive any message.

Some networks allow for *collision detection*, where the effect at node v of a collision is different from that of no message being transmitted, i.e., v can distinguish a collision from silence (despite receiving no message in both). Other networks operate without a collision detection mechanism, i.e., a node cannot distinguish a collision from silence. It is known that the ability to detect collisions has a significant impact on the computational power of the network [26].

In contrast, in this paper, we study the *additive network model*, in which a collision of transmissions is not completely lost, but rather is modeled as receiving the XOR of the bit representation of all transmissions. More specifically, we model a transmission of a message m by node v as a string of bits. A node v that receives a collision of transmissions of messages $\{m_u \mid u \in \Gamma(v)\}$, receives their bitwise XOR, i.e., receives the message $y = \bigoplus_{u \in \Gamma(v)} m_u$. Here $\Gamma(v)$ is the set of neighbors of v. Note that the above notation does not distinguish between the case where a node u transmits to that where it does not, because we model the string of a node that does not transmit as all-zero.

The network topology is unknown, and only a polynomial upper bound $N = n^{O(1)}$ is known for the number of nodes n. Throughout, we assume that each vertex v has a unique identifier $\mathtt{id}_v$ in the range $[1, \ldots, n^c]$ for some constant $c \geq 1$. The bandwidth is $O(\text{poly} \log n)$ bits per message.

Bounded-Contention Coding (BCC): Bounded-Contention Codes were introduced in [6] for the purpose of obtaining fast local and global broadcast in

additive networks. Given parameters M and a, a BCC code is a set of M codewords such that the XOR of any subset of size at most a is uniquely decodable. As such, BCC codes can leverage situations where the number of initial messages is bounded by some number a, and can be used (along with additional mechanisms) for global broadcast in additive networks. Formally, Bounded-Contention Codes are defined as follows.

Definition 1. *An $[M, m, a]$-BCC-code is a set $C \subseteq \{0,1\}^m$ of size $|C| = M$ such that for any two subsets $S_1, S_2 \subseteq C$ (with $S_1 \neq S_2$) of sizes $|S_1|, |S_2| \leq a$ it holds that $\bigoplus S_1 \neq \bigoplus S_2$.*

Simple BCC codes can be constructed using the dual of linear codes. We refer the reader to [6] for additional details and a construction of an $[M, a \log M, a]$-BCC code for given values of M and a.

3 New Tools

In this section we enrich the toolbox for computing in additive networks with the following three techniques. The first is a method for encoding information such that it can be successfully decoded not when the number of transmitters in limited, but rather when the amount of distinct pieces of information is limited (even if sent by multiple transmitters concurrently). The second technique is a general simulation of any algorithm for full-duplex radios in a setting of half-duplex radios within a logarithmic number of rounds. Finally, we show that we can detect whether the number of distinct messages exceeds the given threshold.

Bounded-Information Codes (BIC). Using BCC and randomization allows one to control the number of distinct pieces of information in the neighborhood. Let $G = (V, E)$ be an n-vertex network and assume that all the messages are integers in the range $[0, n]$. We show that for a bandwidth of size $O(\log^3 n)$, one can use randomization and BCC codes to guarantee that every vertex v, whose neighbors transmit $O(\log n)$ *distinct* messages (i.e., hence bounded pieces of information) in a given round, can decode *all* messages correctly with high probability (i.e., regardless of the number of transmitting neighbors).[5] Let C be an $[n, \log^2 n, \log n]$-BCC code and $x \in [0, n]$. By the definition of C, the codeword $C(x) = [b_1, \ldots, b_k] \in \{0,1\}^k$ contains $k = O(\log^2 n)$ bits. Due to the XOR operation, co-transmissions of the same value even number of times are cancelled out. To prevent this, we use a randomized code, named hereafter as a *BIC* code (or BIC for short) as defined next.

Definition 2. *Let C be an $[n, \log^2 n, \log n]$-BCC code. An $[n, c \log^3 n, \log n]$-BIC code for C is a random code C^I defined as follows. The codeword $C^I(x)$ consists of $k' = \lceil c \cdot \log n \rceil$ blocks, for some constant $c \geq 4$, each block is of size $k =$*

[5] The definition of the BIC code can be given for any bound a on the number of distinct values. Since we care for messages of polylogarithmic size, we provide the definition for specific bound $a = O(\log n)$.

$O(\log^2 n)$ *(the maximal length of a BCC codeword), and the* i*'th block contains* $C(x)$ *with probability* $1/2$ *and the zero word otherwise, for every* $i \in \{1, \ldots, k'\}$.

In other words, for vertex v with value x, let $m(v) = C^I(x)$ be the message containing the BIC codeword of x and let $m_i(v)$ denote the i'th block of v's message. Then, $m_i(v) = C(x)$ with probability $1/2$ and $m_i(v) = 0^k$ otherwise. Let $m'(v) = \bigoplus_{u \in \Gamma(v)} m(u)$ be the received message obtained by adding the BIC codewords of v's neighbors. Then the decoding is performed by using BCC to decode each block $m'_i(v)$ separately for every $i \in \{1, \ldots, k'\}$, and taking a union over all decoded blocks.

Lemma 1. *Let* $V' \subseteq V$ *be a set of transmitting vertices with values* $X' = \bigcup_{v \in V'} \textsc{Val}(v)$ *where* $|X'| = O(\log n)$*. For every* $v \in V'$*, let* C_v^I *be an* $[n, c \cdot \log^3 n, \log n]$*-BIC code, for constant* $c \geq 4$*. Let* $m(v)$ *be the* C_v^I *codeword of* $\textsc{Val}(v)$*. Then, the decoding of* $\bigoplus_{v \in V'} m(v)$ *is successful with probability at least* $1 - 1/n^{c-1}$.

Proof. For every $x \in X'$, let $V_x = \{v \in V' \mid \textsc{Val}(v) = x\}$ be the set of transmitting vertices in V' with the value x. For $x \in X'$ and $i \in \{1, \ldots, k'\}$, let $V_x^i = \{v \in V_x \mid m_i(v) = C(x)\}$ be the set of vertices v whose i'th block $m_i(v)$ contains the codeword $C(x)$. We say that block i is *successful* for value $x \in X'$, if $|V_x^i|$ is odd (hence, the messages of V_x are not cancelled out in this block). Let $M_i \subseteq X'$ be the set of values for which the i'th block is successful, and let V'_i contain one representative vertex with a value in M_i. We first claim that with high probability, every value $x \in X'$ has at least one successful block $i_x \in \{1, \ldots, k'\}$. We then show that the decoding of this i_x'th block is successful. The probability that the i'th block is successful for x is $1/2$ for every $i \in \{1, \ldots, k'\}$. By the independence between blocks, the probability that x has no successful block is at most $1/n^c$. By applying the union bound over all $m \leq n$ distinct messages, we get that with probability at least $1 - 1/n^{c-1}$, every value $x \in X$ has at least one successful block i_x in the message. Let $m' = \bigoplus_{v \in V'} m(v)$ be the received message and let m'_i be the i'th block of the received message. It then holds that $m'_i = \bigoplus_{v \in V'} m_i(v) = \bigoplus_{v \in V'_i} m_i(v)$. To see this, observe that the values with even parity in the i'th block are cancelled out and the XOR of an odd number of messages with the same value $C(x)$ is simply $C(x)$. Since m'_i corresponds to the XOR of $|V'_i| = O(\log n)$ distinct messages, the claim follows by the properties of the BCC code. □

In our algorithms, the messages may contain several fields (mostly a constant) each containing a value in $[0, n^c]$ for some constant $c \geq 1$. To guarantee a proper decoding on each field, the messages are required to be aligned correctly. For example, a message containing ℓ fields where the i'th field contains $x_i \in [0, n]$ is split evenly into ℓ blocks and all bits are initialized to zero. The BIC codeword of x_i, denoted by $C^I(x_i)$, is written at the beginning of the i'th block. Hence, when the messages are added up, all codewords of a given block are added up separately. To avoid cumbersome notation, a multiple-field message is denoted by concatenation of the BIC codewords of each field, e.g., the content of a two-field message containing x_1 and x_2 is referred as $C^I(x_1) \circ C^I(x_2)$, where formally

the message is divided into two equi-length blocks and $C^I(x_1)$ (resp., $C^I(x_2)$) is written at the beginning of the first (resp., second) block.

From full-duplex to half-duplex. The algorithms provided in this paper are mostly concerned with the full-duplex setting. However, in the additive network model, one can easily simulate a full-duplex protocol P_f by half-duplex protocol P_h with a multiplicative overhead of $O(\log n)$ rounds with high probability, as explain in more details in [7].

Information-Overflow Detection. In the standard radio model, a collision corresponds to the scenario where multiple vertices transmit in the same round to a given mutual neighbor. In an additive network, this may not be a problem, since with BIC codes, the decoding is successful as long as there are $O(\log n)$ *distinct* pieces of information in a given neighborhood. In this section, we describe a scheme for detecting an event of information-overflow. Our scheme is adapted from the contention estimation scheme of [6], designed for the setting of detecting whether there are more than a certain number of initial messages throughout the network. In our setting, the nodes generate values by themselves, and we will later wish to use the fact that we can detect whether too many different values were generated. The key observation within this context, is that using a BIC code with a doubled information-limit allows one to detect failings with high probability. To see this, assume an information bound $K = c \log n$ for constant $c \geq 1$ and consider an $[n, 2K \log n, 2K]$-BCC code C. The BIC code C^I based on C supports $2K$ distinct messages. Throughout, because of space considerations, some of the proofs are omitted. However, all the proofs are given in the full version [7].

Lemma 2. *With high probability, either it is detected that the number of distinct values exceeds K, or each value w is decoded successfully.*

4 Symmetry Breaking Tasks

In this section we show how to solve symmetry breaking tasks efficiently in additive networks. As a key example, we focus on the problem of leader election. In [7], we consider additional tasks that involve symmetry breaking such as computing a BFS tree, computing an MIS and finding a proper vertex coloring. A key ingredient in many of our algorithms is having the vertices choose random variables according to some carefully chosen probabilities, which, at a high level, are used to reduce the amount of information that is sent throughout the network. We refer to this as the $\mathcal{SL}$ (Select Level) function and describe it as follows.

The $\mathcal{SL}$ function does not require communication, and only produces two local random values, an r-value and an z-value, that can be considered as primary and secondary values for breaking the symmetry between the vertices. The r-value is defined by letting $r = j$ with probability of 2^{-j}, and the z-value, z, is sampled uniformly at random from the set $\{1, ..., 2^{8r}\}$.

Note that $\mathcal{SL}$ does not require the knowledge of the number of vertices n. We next show that the maximum value of $r(v)$ is concentrated around $O(\log n)$ and that not to many vertices collide on the maximum value. Let $j^{\mathcal{SL}}_{max} = \max\{r(v) \mid v \in V\}$ and $S^{\mathcal{SL}}_{max} = \{v \in V \mid r(v) = j^{\mathcal{SL}}_{max}\}$.

Lemma 3. *With high probability, it holds that (a)* $j^{\mathcal{SL}}_{max} \leq 3\log n + 1$*; (b)* $|S^{\mathcal{SL}}_{max}| \leq 2\log n$*; and (c)* $z(v) \neq z(v')$ *for every* $v, v' \in S^{\mathcal{SL}}_{max}$.

Proof. Let $P_v = \mathbb{P}(r(v) \geq 3\log n+1)$. Then, by definition, $P_v = \sum_{i=3\log n+1}^{\infty} 2^{-i}$ $= 1/n^3$. By applying the union bound over all vertices in S, we get that with probability at least $1 - 1/n^2$, $r(v) \leq 3\log n + 1$, for every $v \in S$, as needed for Part (a). We now turn to bound the cardinality of $S^{\mathcal{SL}}_{max}$. The random choice of $r(v)$ can be viewed as a random process in which each vertex flips a coin with probability $1/2$ and proceeds as long as it gets "head". The value of $r(v)$ corresponds to the first time when it gets a "tail". We now claim that the probability that $|S^{\mathcal{SL}}_{max}| > 2\log n$ is very small. This holds since the probability that all of $2\log n$ coin flips are "tails" is exactly $2^{-2\log n}$ which is less than the probability that $|S^{\mathcal{SL}}_{max}| > 2\log n$ and none of the vertices in $S^{\mathcal{SL}}_{max}$ succeeded in getting another head (and hence in having a larger r-value). Hence, the probability that $|S^{\mathcal{SL}}_{max}| \leq 2\log n$ is at least $1 - 2^{-2\log n} = 1 - 1/n^2$, as needed for Part (b).

Finally, consider Part (c). It is sufficient to show that the z-values (of vertices of $S^{\mathcal{SL}}_{max}$) are sampled from a sufficient large range. Note that, the size of this range is $2^{8\cdot j^{\mathcal{SL}}_{max}}$. We later show that $j^{\mathcal{SL}}_{max} \geq \log n/2$ with high probability. This implies that the range size (of the z-values) is at least n^4 with high probability. Assume that $j^{\mathcal{SL}}_{max} \geq \log n/2$, then the probability that $z(v) = z(v')$, for any pair $v, v' \in S^{\mathcal{SL}}_{max}$ is at most $1/n^4$. Applying the union bound over all pairs in $S^{\mathcal{SL}}_{max}$ gives the claim, since $|S^{\mathcal{SL}}_{max}| \leq n$.

In the remaining, we show that indeed, $j^{\mathcal{SL}}_{max} \geq \log n/2$ with high probability. For every $v \in V$, let x_v be an indicator variable for the event that $r(v) \geq \log n/2$, i.e., $x_v = 1$, if $r(v) \geq \log n/2$ and $x_v = 0$, otherwise. Let $X = \sum_{v\in V} x_v$. Note that, the probability that $X \geq 1$ is the same as the probability that $j^{\mathcal{SL}}_{max} \geq \log n/2$. In addition, $\Pr[x_v = 1] = 2^{-(\log n/2)+1} \geq 2^{-\log n/2}$ and hence (by the linearity of expectation) $\mathbb{E}[X] = \sum_{v\in V} \Pr[x_v = 1] = \sqrt{n}$. By Chernoff bound, the probability that $X = 0$ is exponentially small. Hence, $X \geq 1$ and so $j^{\mathcal{SL}}_{max} \geq \log n/2$ with the high probability. Part (c) holds. □

4.1 Leader Election

A Leader-Election protocol is a distributed algorithm run by any vertex such that each node eventually decides whether it is a leader or not, subject to the constraint that there is exactly one leader. Moreover, at the end of the algorithm all vertices know the $\mathcal{SL}$ function values of the leader.

We first describe a two-round leader election protocol for single-hop networks. Let C^I be an $[N, O(\log^3 N), O(\log N)]$-BIC code sampled uniformly at random from the distribution of all random codes that are based on a particular $[N, O(\log^2 N), O(\log N)]$-BCC code C (which is used by all vertices). First,

the vertices apply the $\mathcal{SL}$ function to compute $r(v), z(v)$. To do that, in the first communication round, every vertex v transmits $C^I(r(v))$. Since with high probability, by Lemma 3(a), $j^{\mathcal{SL}}_{max} \leq 2\log n$, the information is bounded and by Claim 2, each vertex can compute $S^{\mathcal{SL}}_{max}$ w.h.p. In the second communication round, every vertex v with $r(v) = j^{\mathcal{SL}}_{max}$, transmits $C^I(z(v))$. That is, in the second phase only the vertices of $S^{\mathcal{SL}}_{max}$ transmit the codeword of their z's value. Since by Lemma 3(b), with high probability, $|S^{\mathcal{SL}}_{max}| = O(\log n)$, and by Claim 2 again, the z-values of all vertices in $S^{\mathcal{SL}}_{max}$ are known to every vertex in the network w.h.p. Finally, the leader is the vertex $v^* \in S^{\mathcal{SL}}_{max}$ with the largest z-value, i.e., $z(v^*) = \max_{v' \in S^{\mathcal{SL}}_{max}} z(v')$. In [7], we consider the general case of electing a leader in a network G with diameter D, and also show how it implies a 2-approximation of the diameter as a byproduct.

5 Approximation Tasks: Degree Approximation

In this section we consider approximation tasks. As a key example, we focus on the task of approximating the degree, i.e., each vertex v is required to compute an approximation for its degree in the graph G. We refer the reader to [7] for additional approximation schemes such as (1) an approximation for the network size; (2) an approximation for the network diameter; and (3) a 2-approximation for the maximum (or minimum).

We describe Algorithm AppDegree that computes with high probability a constant approximation for the degree of the vertices within $O(1)$ rounds. For vertex v and graph G, let $\deg(v, G) = |\Gamma(v, G)|$ be the degree of v in G. When the graph G is clear from the context, we may omit it and simply write $\deg(v)$. Recall that we assume that each vertex v has a unique identifier $\mathtt{id}_v$ in the range of $[1, \ldots, n^c]$ for some constant $c \geq 1$.

The algorithm consists of two communication rounds (which can be unified into a single round). The first round is devoted for computing the exact degree for low-degree vertices v with degree $\deg(v) \leq c \cdot \log n$. The second round computes a constant approximation for high-degree vertices v with $\deg(v) > c \cdot \log n$. In the first communication round, every vertex v uses a random instance C^I_v of an $[N, c{\cdot}\log^3 N, c{\cdot}\log N]$-BIC code to encode its ID and transmits $C^I_v(\mathtt{id}_v)$ as part of $m_1(v)$. In addition, the vertices use the Information-Overflow Detection scheme of Section 3 to verify if their BIC decoding is successful (that is, the message $m_1(v)$ consists of two fields, the first encodes the ID and the second is devoted for overflow detection). Upon receiving $m'_1(v) = \bigoplus_{u \in \Gamma(v)} m_1(u)$, the vertex applies BIC decoding to the first field of the message and applies Information-Overflow Detection to the second field to verify the correctness of the decoding. Note that by the properties of the BIC code, in this round, the low-degree vertices compute their exact degree in G.

The second round aims at computing a constant factor approximation for the remaining vertices with high-degree. Set $a = 40 \cdot \log N$ and $b = 2\log N$. Every vertex v sends an $(a \cdot b)$-bit message $m_2(v)$ defined by a collection of a random numbers in the range of $\{1, \ldots, b\}$ sampled independently by each

vertex v. Specifically, for every v and $i \in \{1,\ldots,a\}$, $r_i(v)$ is sampled according to the geometric distribution, letting $r_i(v) = j$ for $j \in \{1,\ldots,b-1\}$ with probability 2^{-j}, and $r_i(v) = b$ with probability 2^{-b+1} (the remaining probability). For every $i \in \{1,\ldots,a\}$ and every $j \in \{1,\ldots,b\}$, let $x_{i,j}(v) = 1$ if $j < r_i(v)$ and $x_{i,j}(v) = 0$ otherwise. Let $X_i(v) = x_{i,b}(v) \cdots x_{i,2}(v) \cdot x_{i,1}(v)$ and let $m_2(v) = X(v) = X_a(v) \cdots X_2(v) \cdot X_1(v)$ be the transmitted message of v. Let $Y(v) = \bigoplus_{u \in \Gamma(v)} X(u)$ be the received message of v. The decoding is applied to each of the a blocks of $Y(v)$ separately, i.e., treating $Y(v)$ as $Y(v) = Y_a(v) \cdots Y_2(v) \cdot Y_1(v)$, where $Y_i(v) = y_{i,b}(v) \cdots y_{i,2}(v) \cdot y_{i,1}(v)$, such that $y_{i,j}(v) = \bigoplus_{u \in \Gamma(v)} x_{i,j}(u)$. For every $j \in \{1,...,b\}$ and every $v \in V$, define $\mathrm{SUM}(j,v) = \sum_{i=1}^{a} y_{i,j}(v)$. Finally, define $j^*(v) = \min\{j \mid \mathrm{SUM}(j,v) \leq 0.2 \cdot a\}$, if there exists an index j such that $\mathrm{SUM}(j,v) \leq 0.2 \cdot a$ (we later show that such index do exists with high probability) and $j^*(v) = 0$, otherwise as a default value. The approximation $\delta(v)$ is then given by $2^{j^*(v)-1}$. This completes the description of the algorithm.

As mentioned earlier, the correctness for low-degree vertices follows immediately by the properties of the BIC code and the information-overflow detection (Lemma 1 and Lemma 2). We then show that in the second round, for high-degree vertices v, we have $\delta(v)/\deg(v) = O(1)$, with high probability. We thus have the following.

Theorem 1. *There exists an $O(1)$-round algorithm that computes w.h.p. the exact degree $\deg(v)$ for vertices with $\deg(v) = O(\log n)$ and a constant approximation if $\deg(v) = \Omega(\log n)$.*

6 Revealing Asymmetry – Distributed Tournament

Consider the setting where every vertex is given an input value (corresponding to its rank, for example) and the goal is to find the vertex with the maximum value. We will show that BCC codes with message size of $O(\log^3 n)$ allow one to perform many simultaneous competitions between $\Omega(\log n)$ candidates, which result in a tournament process of $O(D \cdot \log n/\log\log n)$ rounds for a network of diameter D. Specifically, the fact that the BCC code provides successful decoding when there are $O(\log^2 n)$ concurrent transmitting neighbors, allows us to reduce the number of competitors by a factor of $\Omega(\log n)$ in every round, and hence the winner is found within $O(D \cdot \log n/\log\log n)$ rounds. Because of space considerations, we presenting here only the protocol for single-hop networks. The protocol for any network of diameter $D > 1$, which requires some subtle modifications is presented in the full version [7].

Single-hop network. Let $V = \{v_1,\ldots,v_n\}$ be the vertices of the network and let $X = \{x_1,\ldots,x_n\}$, where $x_i \in \{1,\ldots n^2\}$ for all i, be the set of integral inputs such that vertex v_i holds the input x_i. Let $\max(X) = \max_{i=1}^{n} x_i$ be the maximum value in X. Note that by Section 5, a 2-approximation for the maximum can be computed within a single round, w.h.p. The main contribution of this section is the *exact* computation of the maximum value.

Theorem 2. *The maximum value* $\max(X)$ *can be computed within* $O\left(\frac{\log n}{\log\log n}\right)$ *rounds, with high probability.*

Algorithm CompMaxSH consists of $O(\log n/\log\log n)$ communication rounds. For simplicity, assume that the input values are distinct. This can be obtained by appending to every input value $\lceil \log n \rceil$ least significant bits corresponding to the ID of the vertex. Let $c \geq 2$ be an upper bound on the approximation ratio of Algorithm ApproxNetSize and set $\tau = \lceil c \cdot \log n/\log\log n \rceil$. Initially, all vertices are active. In round $t = \{1, \ldots, \tau\}$, let n_t be a constant approximation for the number of active vertices at the beginning of round t, and let C be an $[n_1, 32c\cdot\log^3 n_1, 32c\cdot\log^2 n_1]$-BCC code[6]. After computing n_t, every active vertex v_j transmits $C(x_j)$ with probability $p_t = 4c\cdot\log^2 n_1/n_t$. If a vertex v_i receives an input $x_j > x_i$ in round t, it becomes inactive. The final result $\max(v_i)$ of every vertex v_i corresponds to the maximum input value x_j it received throughout the algorithm. This completes the description of the algorithm.

We now analyze the algorithm and begin with correctness. Let A_t be the active vertex set at the beginning of round t. Note that $A_\tau \subseteq \ldots \subseteq A_1 = V$. Let v_m be a vertex with maximum input, i.e., $x_m = \max(X)$.

Lemma 4. *For each round* $t \in \{1, \ldots, \tau\}$*, with high probability it holds that* $|A_t| = O(n_1/\log^{t-1} n_1)$ *and* $x_m \in A_t$*.*

Proof. The claim is shown by induction. For the base of the induction $t = 1$, we have that $A_1 = V$, and $n_1 \leq c \cdot n$ since by the properties of Algorithm ApproxNetSize it holds that with high probability $n_1 \in [n/2, c \cdot n]$ for some constant $c \geq 2$. Assume that the claim holds up to step $t - 1 \geq 1$ and consider step t. Order the values of the vertices in A_{t-1} in increasing order of their inputs and consider the subset $H_{t-1} \subset A_{t-1}$ of the $\lceil |A_{t-1}|/\log n_1 \rceil$ vertices with the highest input values in A_{t-1}. We first claim that with high probability, at least one of the vertices in H_{t-1} transmits in round $t-1$. Since every vertex in A_{t-1} transmits with probability of $p_{t-1} = 4c\log^2 n_1/n_{t-1}$ and $n_{t-1} \leq c \cdot |A_{t-1}|$, in expectation there are at least $4\log n_1$ transmitting vertices in H_{t-1} and hence, by a Chernoff bound, w.h.p there is at least one transmitter in H_{t-1}.

We proceed by showing that the number of transmitting vertices in round $t - 1$ is $O(\log^2 n)$. In expectation, the number of transmitting vertices in A_{t-1} is at most $8c \cdot \log^2 n_1$, and hence by Chernoff bound, with high probability there are less than $32c\log^2 n_1$ transmitters. By the properties of the BCC code, all messages received in round $t - 1$ are decodable. This implies that all vertices know the value of at least one vertex in H_{t-1} and as a result all vertices in $V \setminus H_{t-1}$ become inactive. In other words, $A_t \subseteq H_{t-1}$ and hence $n_t \leq |H_{t-1}| = |A_{t-1}|/\log n_1 = O(n_1/\log^t n_1)$, where the last equality holds w.h.p by the induction assumption. Finally, by the induction assumption for $t - 1$, $v_m \in A_{t-1}$, since all messages were decoded successfully in round $t - 1$ w.h.p, it holds that v_m remains in A_t as well. The claim follows. □

[6] This approximation for the size of the network can be obtained by applying Algorithm ApproxNetSize or simply Algorithm AppDegree in the case of single-hop networks (where only the active vertices participate in these algorithms).

We thus have the following, which proves Theorem 2.

Lemma 5. *With high probability* $\max(v_i) = \max(X)$ *for every vertex* $v_i \in V$.

7 Discussion

It is clear that computing in the additive network model should be doable faster than in the standard radio network model. In this paper we quantify this intuition, by providing efficient algorithms for various cornerstone distributed tasks. Our work leaves open several important open questions for further research. First, it is natural to ask whether our algorithms can be improved. Specifically, most of our algorithms apply for the full-duplex model and translate into half-duplex by paying an extra factor of $O(\log n)$. It would be interesting to obtain better bounds for half-duplex radios without using the full-duplex protocol as a black box. An additional axis that requires investigation is the multiple channels model. It would be interesting to study the tradeoff between running time, message size and the number of channels. Note, that whereas most of our algorithms are optimal for full-duplex radios (up to constant factors), some leave room for improvements. For example, in the problem of computing the maximum input, we believe that some pipelining of the simulation of phases should be able to give a round complexity of $O(D + \log n/\log\log n)$, instead of the current $O(D \cdot \log n/\log\log n)$. However, this is not immediate. Designing lower bounds for this model is another important future goal. It seems that the problem of computing the maximum input in a single-hop network, should be a good starting point, as we believe that this task requires $\Omega(\log n/\log\log n)$ rounds. Another interesting future direction involves the implementation of an abstract MAC layer over *additive* radio network model. Such an implementation was provided recently [13] for the standard radio network model. Finally, we note that all our algorithms are randomized, as opposed to the original definition of BCC codes. Is randomization necessary? What is the computational power of the additive network model without randomization?

References

1. N. Alon, L. Babai, and A. Itai. A fast and simple randomized parallel algorithm for the maximal independent set problem. *J. Algorithms*, 7(4):567–583, 1986.
2. J. Andrews. Interference cancellation for cellular systems: a contemporary. *SIAM Journal on Computing*, 12(1):19 – 2, 2005.
3. A. S. Avestimehr, S. N. Diggavi, and D. Tse. Wireless network information flow: A deterministic approach. *IEEE Trans. on Info. Theory*, 57(4):1872–1905, 2011.
4. R. Bar-Yehuda, O. Goldreichh, and A. Itai. On the time-complexity of broadcast in multi-hop radio networks: An exponential gap between determinism and randomization. *J. of Compt. Syst. Sciences*, 45:104 – 126, 1992.
5. L. Barenboim, M. Elkin, S. Pettie, and J. Schneider. The locality of distributed symmetry breaking. In *FOCS*, pages 321–330, 2012.

6. K. Censor-Hillel, B. Haeupler, N. A. Lynch, and M. Médard. Bounded-contention coding for wireless networks in the high snr regime. In *DISC*, pages 91–105, 2012.
7. K. Censor-Hillel, E. Kantor, N. A. Lynch, and M. Parter. Computing in additive networks with bounded-information codes. arxiv.org/abs/1508.03660, 2015.
8. S. Daum, M. Ghaffari, S. Gilbert, F. Kuhn, and C. Newport. Maximal independent sets in multichannel radio networks. In *PODC*, pages 335–344, 2013.
9. M. Farach-Colton, R. J. Fernandes, and M. A. Mosteiro. Lower bounds for clear transmissions in radio networks. In *LATIN: Theoretical Informatics*, pages 447–454, 2006.
10. M. Ghaffari and B. Haeupler. Near optimal leader election in multi-hop radio networks. In *SODA*, pages 748–766, 2013.
11. S. Gollakota and D. Katabi. Zigzag decoding: combating hidden terminals in wireless networks. In *SIGCOMM*, pages 159–170, 2008.
12. P. Gupta and P. Kumar. The capacity of wireless networks. *IEEE Trans. on Info. Theory*, pages 388–404, 2000.
13. F. Kuhn, N. Lynch, and C. Newport. The abstract mac layer. *Distributed Computing*, 24:187–206, 2011.
14. F. Kuhn, T. Moscibroda, and R. Wattenhofer. What cannot be computed locally! In *Proc. PODC*, pages 300–309, 2004.
15. F. Kuhn and R. Wattenhofer. Constant-time distributed dominating set approximation. *Distributed Computing*, 17(4):303–310, 2005.
16. N. Linial. Locality in distributed graph algorithms. *SIAM Journal on Computing*, 21(1):193–201, 1992.
17. Z. Liu and M. Herlihy. Approximate local sums and their applications in radio networks. In *DISC*, pages 243–257, 2014.
18. M. Luby. A simple parallel algorithm for the maximal independent set problem. *SIAM Journal on Computing*, 15:1036–1053, 1986.
19. Y. Métivier, J. Robson, N. Saheb-Djahromi, and A. Zemmari. An optimal bit complexity randomized distributed mis algorithm. *Distributed Computing*, 23(5-6):331–340, 2011.
20. T. Moscibroda and R. Wattenhofer. Maximal independent sets in radio networks. In *PODC*, pages 148–157, 2005.
21. A. Ozgur, O. Leveque, and D. Tse. Hierarchical cooperation achieves optimal capacity scaling in ad hoc networks. *IEEE Trans. on Info. Theory*, pages 3549–3572, 2007.
22. A. ParandehGheibi, J.-K. Sundararajan, and M. Médard. Collision helps - algebraic collision recovery for wireless erasure networks. In *WiNC*, 2010.
23. K. N. Ramachandran, E. M. Belding-Royer, K. C. Almeroth, and M. M. Buddhikot. Interference-aware channel assignment in multi-radio wireless mesh networks. In *INFOCOM*, pages 1–12, 2006.
24. J. Schneider and R. Wattenhofer. Coloring unstructured wireless multi-hop networks. In *PODC*, pages 210–219, 2009.
25. J. Schneider and R. Wattenhofer. An optimal maximal independent setalgorithm for bounded-independence graphs. *Distributed Computing*, 22(5-6):349–361, 2010.
26. J. Schneider and R. Wattenhofer. What is the use of collision detection (in wireless networks)? In *DISC*, pages 133–147, 2010.