

Fast Distributed Algorithm for Convergecast in Ad Hoc Geometric Radio Networks

Alex Kesselman

Max Planck Institut für Informatik, Saarbrücken, Germany.

Email: akessel@mpi-sb.mpg.de

Dariusz Kowalski

Instytut Informatyki, Uniwersytet Warszawski Poland and

Max Planck Institut für Informatik, Saarbrücken, Germany.

Email: darek@mpi-sb.mpg.de

Abstract—Wireless ad hoc radio networks have gained a lot of attention in recent years. We consider geometric networks, where nodes are located in a euclidean plane. We assume that each node has a variable transmission range and can learn the distance to the closest neighbor. We also assume that nodes have a special collision detection (CD) capability so that a transmitting node can detect a collision within its transmission range. We study the basic communication problem of collecting data from all nodes called *convergecast*. Recently, there appeared many new applications such as real-time multimedia, battlefield communications and rescue operations that impose stringent delay requirements on the convergecast time. We measure the *latency* of convergecast, that is the number of time steps needed to collect the data in any n -node network. We propose a very simple randomized *distributed* algorithm that has the expected running time $\mathcal{O}(\log n)$. We also show that this bound is tight and any algorithm needs $\Omega(\log n)$ time steps while performing convergecast in an arbitrary network.

One of the most important problems in wireless ad hoc networks is to minimize the energy consumption, which maximizes the network lifetime. We study the trade-off between the energy and the latency of convergecast. We show that our algorithm consumes at most $\mathcal{O}(n \log n)$ times the minimum energy. We also demonstrate that for a line topology the minimum energy convergecast takes $n - 1$ time steps while any algorithm performing convergecast within $\mathcal{O}(\log n)$ time steps requires $\Omega(n)$ times the minimum energy.

Keywords: Radio Networks, Convergecast, Randomized Algorithms, Energy/Latency Trade-off.

The work of the first author is supported by AvH-Stiftung. A part of this work has been done while visiting the Dipartimento di Scienze dell'Informazione at Università di Roma La Sapienza, Italy. Supported in part by EYES project and MIUR grant FIRB WebMinds.

The work of the second author is supported in part by EU DELIS project and by KBN grant 4T11C04425.

I. INTRODUCTION

The next generation communication networks are likely to be a combination of wireline and ad hoc networks, which are expected to fulfill a critical role where wired backbone networks are not available or not economical to build [30]. A communication session in a wireless network is achieved either through a single-hop transmission if the communication parties are close enough, or through relaying by intermediate nodes otherwise. Depending on its power level and on the nature of environmental interference, a node can reach all nodes in a certain range. Typically, the signal power falls as $1/d^\alpha$, where d is the distance from the transmitter antenna and α is a constant between 2 and 6 depending on the environment [31]. All receivers have the same power threshold for signal detection, which is typically normalized to one. Under the above assumptions, the power required to establish a link between two nodes at distance d is d^α . In this paper we assume that $\alpha = 2$. In ad hoc networks devices are usually equipped with battery that has a limited power. Thus, among the most crucial issues is that of developing energy-efficient topology control algorithms, which maximize the network lifetime [16].

In this paper we assume that nodes are located in a Euclidean space and consider the symmetric energy model. We also assume that nodes are equipped with Global Position System (GPS) and each node can learn the distance to the closest neighbor (see [5], [22], [26], [29]). Similarly to many papers in the domain of radio communication [3], [9], [12], [13], we assume that nodes send messages in synchronous *steps* (time slots) controlled by a global clock. During a time step, each node acts either as a *transmitter* or as a *receiver*. A node acting as a transmitter chooses a transmission range and sends a message which can potentially reach all of nodes whose distances from this node are at most the chosen transmission range. In this case we say that the message *reaches* a node within the (current) transmission range of a sender. A node acting as a receiver in a given step gets a message, if and only if, it is the only one message that reaches this node in the current step. If more than one message reaches a node u then none of the messages is received by u in the current step. In this case we say that a *collision* occurred at node u . We assume that nodes have a special

collision detection (CD) capability so that a transmitting node can detect a collision within its transmission range.

An important role of a wireless sensor network is to collect data from the environment [16]. Thus, one of the fundamental communication tasks is *convergecast*, where one node has to collect rumors which are originally stored at all nodes [2], [9], [10], [12], [19], [20], [21]. These rumors can be later broadcasted to the other nodes, as in the gossiping problem. Note that the complexity of the gossiping problem is dominated by convergecast. We assume that rumors can be merged and sent as one message without any restrictions. In practice, the maximum message size is upper-bounded by the total size of all rumors, that is $O(n \log n)$. We measure the *latency* of convergecast, which is among the most popular and natural measures for communication problems.

The rapidly increasing capabilities and low costs of computing and communication devices have made it possible to use wireless networks in a wide range of applications such as real-time multimedia, battlefield communications, and rescue operations. The above applications impose stringent latency requirement on the convergecast time. Unfortunately, in the existing minimum-energy convergecast algorithms the latency tends to be quite large since the minimum energy is attained when the number of relay nodes is maximized. There is a trade-off between reaching a node directly using higher power and reaching a node via multiple hops using lower power. We study the trade-off between the energy and the latency of convergecast.

In this work we study randomized algorithms, which have a number of advantages. Randomized algorithms are usually faster than deterministic ones. In addition, randomized algorithms are very attractive due to their simplicity, which makes them amenable to efficient implementation. The algorithm proposed in this paper is also *distributed*, in the sense that it only uses information about the distance to the closest neighbor.

Our results. We propose a very simple randomized distributed algorithm for convergecast that has the expected running time $\mathcal{O}(\log n)$. We also show a matching lower bound of $\Omega(\log n)$ on the running time of any algorithm in an arbitrary network. This bound holds even for centralized algorithms. Our results show that having variable transmission ranges allows to break the linear lower bound for the convergecast time, which is inherent in one-hop radio networks. Then we study the trade-off between the energy and the latency of convergecast. We describe a minimum energy convergecast algorithm with a polynomial running time. We also show that our *distributed* convergecast algorithm in the worst case consumes $\mathcal{O}(n \cdot \log n)$ times the minimum energy. Finally, we demonstrate that for a line topology the minimum energy convergecast takes $n - 1$ time steps while any algorithm performing convergecast within $\mathcal{O}(\log n)$ time steps requires $\Omega(n)$ times the minimum energy.

Paper organization. Related work is given in Section II. Section III describes our model. Our algorithm for the convergecast problem is presented in Section IV. The trade-off between the energy and the latency of convergecast is analyzed in Section V. We have some concluding remarks in Section VI.

II. RELATED WORK

In this section we give an overview of the related work.

A. Gossiping

The most closely related problem to this work is gossiping. In the context of radio networks, this problem was considered only for fixed ranges, and most of the solutions used convergecast as a procedure to perform gossiping. The first sub-quadratic deterministic distributed algorithm for the gossiping problem in ad-hoc radio networks was the $\tilde{\mathcal{O}}(n^{3/2})$ ¹ time algorithm proposed by Chrobak et al. [12]. Gasieniec, Radzik and Xin [21] presented a gossiping algorithm with running time $\tilde{\mathcal{O}}(n^{4/3})$. Another class of algorithms depending on the maximum degree Δ and the diameter D of the network, was considered by Clementi, Monti and Silvestri [13] and subsequently by Gasieniec and Lingas [18] – the former algorithm performs gossiping in time $\tilde{\mathcal{O}}(D\Delta^2)$ and the latter algorithm in time $\tilde{\mathcal{O}}(D\Delta^{3/2})$. A study on distributed deterministic gossiping with messages of limited size can be found in the work by Christersson et al. [10].

The gossiping problem in ad hoc radio networks attracted also studies on efficient randomized distributed algorithms. Chrobak et al. [12] proposed an $\mathcal{O}(n \log^4 n)$ expected time randomized gossiping algorithm. This time was later reduced to $\mathcal{O}(n \log^3 n)$ by Liu and Prabhakaran [28], and then to $\mathcal{O}(n \log^2 n)$ by Czumaj and Rytter [14]. This shows a relatively large gap between the best known deterministic and randomized algorithms for distributed gossiping.

In the context of centralized solutions, the gossiping problem was studied by Gasieniec and Potapov [19] and recently by Gasieniec et al. [20]. In the first paper each node transmission was limited to unit message. The authors presented several optimal and almost optimal $\mathcal{O}(n)$ time gossiping algorithms for various standard network topologies, including lines, rings, stars and free trees. The authors also proved that there exists a radio network topology in which the gossiping (with unit messages) requires $\Omega(n \log n)$ time. In the second work, the authors considered the model with unlimited message size, and analyzed families of graph where sublinear-time gossiping is possible.

Oblivious algorithms (both deterministic and randomized) for the task of gossiping (all-to-all broadcast) were considered by Chlebus et al. [9] and Kowalski and Pelc [25].

B. Broadcast

Another related problem is broadcast, which was also studied only in case of fixed transmission ranges. Deterministic centralized broadcast assuming complete knowledge of the network was considered by Chlamtac and Weinstein [8], where a $\mathcal{O}(D \log^2 n)$ -time broadcast algorithm was given for all n -node networks of diameter D . Gaber and Mansour [17] proposed $\mathcal{O}(D + \log^5 n)$ -time broadcasting scheme. On the other hand, Alon et al. [1] proved the existence of a family of n -node networks of radius 2, for which any broadcast requires time $\Omega(\log^2 n)$.

Bar-Yehuda et al. [3] studied deterministic distributed broadcast in radio networks whose nodes have only limited knowledge of the topology. The authors assumed that nodes know

¹Notation $\tilde{\mathcal{O}}(f(n))$ denotes a function in $\mathcal{O}(f(n) \log^c n)$ for a (small) constant c .

only their own label and labels of their neighbors. Concerning the weaker model where nodes know only their own label, the currently fastest deterministic distributed algorithm, designed by Czumaj and Rytter [14], performs broadcast in time $\mathcal{O}(n \log^2 D)$. On the other hand, Clementi et al. [13] proved a lower bound $\Omega(n \log D)$ on broadcast time for directed n -node networks of diameter D .

Randomized broadcast algorithms in radio networks were extensively studied in recent time [3], [9], [14], [24], [27]. For these algorithms, no topological knowledge of the network was assumed. Bar-Yehuda et al. [3] designed a randomized broadcast algorithm running in expected time $\mathcal{O}(D \log n + \log^2 n)$. Kowalski and Pelc [24], and independently Czumaj and Rytter [14], improved this upper bound by presenting a broadcasting algorithm with expected time $\mathcal{O}(D \log(n/D) + \log^2 n)$. The matching lower bound $\Omega(D \log(n/D) + \log^2 n)$ was proved by Kushilevitz and Mansour [27] and Alon et al. [1].

Dessmark and Pelc [15] studied broadcasting protocols in geometric networks, but unlike our paper they considered fixed transmission ranges, no collision detection, and wider information range about the neighborhood (we assume only knowledge of the distance to the closest active node). The authors were interested in the trade-off between the latency and the information range, rather than a power consumption.

C. Energy Efficient Network Design

The problem of minimum-energy broadcast in which the transmission power of each node has to be determined so that the total power is minimized has received recently a great deal of attention. Cagalj et al. [4] give a proof of NP-hardness of the minimum-energy broadcast problem in Euclidean space. Wieselthier et al. [34] propose three greedy heuristics, namely the minimum spanning tree (MST), the shortest path tree (SPT) and the broadcast incremental power (BIP), and evaluate them through simulations. Wan et al. [33] present the first analytical results for this problem. In particular, they prove that the approximation ratio of MST is between 6 and 12 while the approximation ratio of SPT is at least $n/2$. Cartigny et al. [6] develop localized algorithms for minimum-energy broadcast.

In wireless sensor networks, many applications involve both convergecast and broadcast. These tasks can be accomplished by constructing an efficient tree for both broadcast as well as convergecast and allocating wireless communication channels to ensure collision-free communication. Upadhyayula et al. [32] and Annamalai et al. [2] propose low-latency energy-efficient centralized heuristic algorithms for convergecast and broadcast.

III. MODEL AND NOTATION

We assume that there are n nodes, which are located on a Euclidean plane. Each node knows its own ID and the total number of nodes. We assume that nodes are equipped with Global Position System (GPS) and each node can learn the distance to the closest neighbor. We consider the synchronized model, where the nodes have a common clock.

During a time step, a node may either send a message or listen to the channel. We assume that one message can carry data

collected from multiple nodes. A node successfully receives a message if and only if exactly one of its in-neighbors transmits during a time step, otherwise *collision* occurs. During a collision the messages sent are scrambled so that the node cannot retrieve any part of a message. We assume that each node can detect whether a collision occurred within its transmission range.

Let V be the initial set of nodes. Some nodes may become *inactive* later. Consider a time step t . A wireless network is represented by a directed *communication graph* $G^t = (V^t, E^t)$, where V^t is the set of *active* nodes and E^t is the set of edges. The neighbors of a node u are determined by its *transmission power* P_u . Namely, node u can reach all nodes within its *transmission range* $R_u^t = (P_u^t)^{1/2}$. Thus, edge (u, v) belongs to E if the distance between u and v , $d(u, v)$, is at most R_u^t . We assume that nodes have *variable* transmission ranges.²

Consider a convergecast algorithm A . Let us denote by

$$E_A^t = \sum_{u \in V^t} P_u$$

the total energy spent by all nodes at time t . If A terminates at time t_f , then its total *energy consumption* is

$$C_A = \sum_{t=0}^{t_f} E_A^t.$$

We also denote by OPT the minimum-energy convergecast algorithm and by C_{OPT} its energy consumption.

In the *convergecast* problem each node has some data and the data from all nodes must be collected by a single node. The goal is to minimize the *latency* (number of time steps) required to perform convergecast while keeping the energy consumption as low as possible.

IV. DISTRIBUTED CONVERGECAST ALGORITHM

In this section we present a simple randomized distributed algorithm for convergecast, called DC . We show that our algorithm terminates in the expected time $\mathcal{O}(\log n)$ and present a matching general lower bound of $\Omega(\log n)$.

The algorithm proceeds as follows. Initially, all nodes are *active*. Then we start to collect data. The execution of the algorithm is divided into *rounds*, each of which corresponds to one time step. At the beginning of a round, we set the transmission range of each active node u to the distance to the closest active node, i.e., $\min_{v \in V^t} d(u, v)$. Then u transmits the data with a constant probability p . Thereafter, each active node w that does not detect a collision, i.e., whose message has been successfully received by all its out-neighbors, becomes *inactive*. Otherwise, the node remains active for the next round, and merges received data (if any) with its own data (which is initially its ID and rumor). Note that there is no way to distinguish between the case in which the message has been received successfully by one neighbor and collided at another neighbor and the case in which the message collided at all neighbors. Finally, the algorithm terminates when there remains only one active node. The formal description of the algorithm appears on Figure 1.

²For simplicity, we assume that the maximum transmission range of a node is unbounded.

Each *active* node u executes the following every round t :

- 1) Set the transmission range R_u^t to the distance to the closest active node.
- 2) Transmit $\text{MSG}(\text{data}, u)$ containing u 's and collected data with a constant probability p .
- 3) If a message $\text{MSG}(\text{data}, u)$ has been transmitted and there is no collision among u 's out-neighbors, enter the *inactive* mode. Otherwise, merge the received data (if any) with its own data.

Fig. 1. The Distributed Convergecast (*DC*) algorithm.

A. Analysis

The next lemma establishes the correctness of the algorithm.

Lemma IV.1: When the algorithm terminates, the last active node has data collected from all nodes.

Proof: Consider a round in which an active node u becomes inactive. We argue that the data held by u has been delivered to another node, which will be active at the beginning of the next round. Note that u must have at least one out-neighbor, say v . Since there was no collision among u 's out-neighbors, u 's transmission must have been received successfully by v . We claim that v will be active at the beginning of the next round. Observe that v did not transmit during this round, otherwise u 's transmission would have collided with v 's own transmission. Therefore, v cannot become inactive at the end of the round, which establishes the lemma. ■

Now we will analyze the performance of our algorithm. First we need some auxiliary lemmas. Intuitively, the next lemma bounds the number of collisions by establishing an upper bound on the maximum in-degree in G^t .

Claim IV.2: The in-degree of any node in G^t is at most 6.

Proof: Suppose towards a contradiction that a node w in G^t has in-degree larger than 6. We have that there are at least 7 nodes that are closer to w than to each other and there must be a triangle formed by two of them u, v , and w in which the angle $\angle u w v$ is less than $\pi/3$. Thus, the biggest angle in this triangle, say $\angle w u v$, is greater than $\pi/3$. However, in this case we obtain that $d(v, u)$ must be smaller than $d(v, w)$, which contradicts our assumption. ■

Now we will analyze the rate of data convergence. We say that a transmission of a node u at the beginning of a round is *successful* if all of its out-neighbors successfully receive u 's transmission. Note that each active node u becomes inactive at the end of the round after successful transmission. In the following lemma we show that a constant fraction of active nodes transmit successfully during a round.

Lemma IV.3: There is a constant $0 < c < 1$ such that with probability at least c , the fraction of active nodes that perform successful transmission in round t is at least c .

Proof: Consider a round t and the corresponding graph G^t . By Claim IV.2, the in-degree of each node in V^t is bounded by 6. That implies that the average out-degree among the nodes in V^t is also bounded by 6. We obtain that at least half of the nodes in V^t have out-degree of at most 12. Let us calculate the probability of successful transmission of such a node u . This probability is exactly the probability that u transmits while all its out-neighbors and the in-neighbors of its out-neighbors (other than u) remain silent. According to Claim IV.2, each of u 's out-neighbors may have at most 6 in-neighbors including u . Therefore, the probability of successful transmission for u is at

least

$$p_s = p(1-p)^{12 \cdot 5 + 12} = p(1-p)^{72}.$$

We have that for any constant $0 < p < 1$, p_s is constant and thus the expected number of nodes that do not transmit successfully during a round is at most $n \cdot (1 - p_s)$. Let $c = p_s/2$. Using Markov inequality we get that the number of nodes that do not transmit successfully during a round is at least $n \cdot (1 - p_s) \cdot (1 - c)^{-1}$ with probability at most $(1 - c)$, which yields that the event “the number of nodes which transmit successfully during a round is at least $n - n \frac{1-p_s}{1-c} \geq n \cdot c$ ” holds with probability at least c . ■

In the actual implementation of our algorithm, we will select p that maximizes the probability of successful transmission, that is $p(1-p)^{72}$. Now we are ready to show the main theorem.

Theorem IV.4: The expected running time of the *DC* algorithm is $\mathcal{O}(\log n)$ and the algorithm terminates properly.

Proof: We say that round i is *progressive* iff a fraction c of active nodes become inactive or there remains only one active node, where constant c is taken from the statement of Lemma IV.3. Note that the *termination time* τ of the *DC* algorithm is less than or equal to the time of the $\log_{1-c}(1/n)$ -th progressive round, since after $\log_{1-c}(1/n)$ progressive rounds the number of active nodes is at most

$$n \cdot (1 - c)^{\log_{1-c}(1/n)} = 1.$$

Lemma IV.3 implies that round i is progressive with probability at least c . Hence, the expected termination time of the *DC* algorithm is at most

$$E(\tau) \leq \frac{1}{c} \log_{1-c}(1/n) = \mathcal{O}(\log n).$$

The correctness of the algorithm follows by Lemma IV.1. ■

The next theorem establishes a general lower bound of $\Omega(\log n)$, which matches our upper bound up to a constant factor.

Theorem IV.5: The expected running time of any convergecast algorithm in an arbitrary network is at least $\Omega(\log n)$.

Proof: Note that each node, except the last one, must successfully transmit at least once to deliver its own data. Without loss of generality, assume that each node successfully transmits exactly once since we can accommodate all the data in the last transmission. When a node transmits, the receiving node is busy, i.e., it cannot transmit itself. Therefore, the number of nodes that have not transmitted yet is decreased by at most a factor of two during a time step, which yields the theorem. ■

V. ENERGY/LATENCY TRADE-OFF

In this section we study the trade-off between the energy and the latency of convergecast. We will also consider centralized

algorithms, since we wish to study only this specific trade-off and not the lack of coordination between nodes. We assume that there is a designated root node r that must collect the data from all nodes.

A. Minimum Energy Convergecast

First we describe a minimum-energy convergecast algorithm. The Minimum Spanning Tree (MST) algorithm proposed by Chen and Huang [7] and analyzed by Kirousis et al. [23] implicitly finds an optimal solution for the minimum-energy convergecast in polynomial time. First we construct an undirected complete graph G' on the set of nodes V , where the cost of an edge (u, v) is the energy required to establish this edge, that is $d(u, v)^2$. Then we find an MST of G' and orient all edges toward r . Finally, we construct a collision-free schedule in which the data from all nodes is collected by r and each edge is traversed exactly once. The optimality of this solution follows from the fact that OPT must have a directed spanning tree rooted at r as a sub-graph of the family of all the communication graphs created during its execution. Note that the data from each node must eventually reach r . The latency of the solution produced by the MST algorithm is at least the length of the longest path between a leaf and r – hence $\Omega(n)$ latency for some geometric topologies (unlike our DC algorithm).

B. Energy Consumption by the DC Algorithm

Now let us consider the energy consumption of our distributed convergecast DC algorithm. We assume that r is the last active node in the execution. The next claim bounds from above the energy consumption by the DC algorithm during the first round.

Claim V.1: The energy spent by the DC algorithm during the first time step is at most C_{OPT} .

Proof: Note that at time $t = 0$ for each node u we have that the transmission range of u is the distance to the closest neighbor v in V . On the other hand, in the MST algorithm u transmits to some node w , not necessarily the closest one. Hence, the transmission energy of u during the first time step of the distributed convergecast algorithm, $d(u, v)^2$, is at most the transmission energy of u during the execution of OPT , $d(u, w)^2$. The claim follows by summing the energy over all nodes. ■

Next we derive an upper bound on the energy spent by the distributed convergecast algorithm in any of the consequent rounds.

Lemma V.2: The energy spent by the DC algorithm during a round t is at most $\frac{\pi^2}{6} \cdot n \cdot C_{OPT}$.

Proof: Consider a round t and let $m \leq n$ be the number of active nodes. Enumerate the nodes in the order of non-increasing transmission range:

$$R_1^t \geq R_2^t \cdots \geq R_m^t.$$

Let Z be the sum of the transmission ranges of the nodes under OPT (we consider the family of all OPT 's communication graphs during its execution). We claim that $R_i^t \leq Z/i$. Consider the set of nodes S_i containing the first i nodes and the

root. We claim that the distance between any two nodes in S_i is at least R_i^t . If it is not the case, at least one node has its transmission range larger than the distance to the closest active node, which contradicts the definition of $R_1^t, R_2^t, \dots, R_i^t$. We obtain that $Z \geq iR_i^t$ since OPT must connect all these nodes to the root.

Hence, the energy consumption of the distributed convergecast algorithm during round t is at most

$$E_{DC}^t \leq \sum_{i=1}^m (Z/i)^2 = Z^2 \sum_{i=1}^m 1/i^2 \leq \frac{\pi^2}{6} Z^2.$$

On the other hand,

$$C_{OPT} \geq n(Z/n)^2,$$

since the transmission power is minimized when all nodes are evenly spaced. Dividing the bound for E_{DC}^t by the bound for C_{OPT} we conclude the proof of the lemma. ■

Now we establish an upper bound on the total energy of the distributed convergecast algorithm.

Theorem V.3: The total energy consumption of the DC algorithm at most $\frac{\pi^2}{6} \cdot n \cdot \log n \cdot C_{OPT}$.

The theorem follows directly from Theorem IV.4 and Lemma V.2.

C. Analysis of Line Topology

Finally, we show that a blowup of $\Omega(n)$ in the energy consumption may be necessary to achieve a convergecast latency of $\mathcal{O}(\log n)$, even in case of centralized algorithms. Consider a line topology, where the root is the last node and the distance between any two consecutive nodes is d .

It is not hard to see that the minimum energy convergecast algorithm OPT during round $i < n$ transmits from node i to node $i + 1$ the data collected from nodes $1, \dots, i$. Thus, OPT has latency $n - 1$ and energy consumption of $(n - 1) \cdot d^2$. Now we bound from below the energy required by any fast convergecast algorithm.

Theorem V.4: On a line, any convergecast algorithm that has running time $\mathcal{O}(\log n)$ requires energy $\Omega(n^2 d^2)$, where d is the distance between two consecutive nodes.

Proof: The proof of Theorem IV.5 implies that in order to achieve a latency of $\mathcal{O}(\log n)$, one has to assure that in each round a constant fraction of active nodes pass their data to adjacent active nodes and become inactive. However, in this case the transmission ranges of active nodes grow exponentially. Note that the minimum energy is attained if all active nodes are evenly spaced. We obtain that the total energy consumption of any fast convergecast algorithm is proportional to

$$\sum_{i=0}^{\log n} 2^{(\log n) - i} \cdot (2^i d)^2 = n \sum_{i=0}^{\log n} 2^i \cdot d^2 = \Omega(n^2 d^2).$$

Therefore, any fast convergecast algorithm on a line requires $\Omega(n)$ times the minimum energy. ■

VI. CONCLUSION

In this paper we consider ad hoc geometric radio networks. We present a very simple randomized distributed algorithm with the expected running time $\mathcal{O}(\log n)$ and establish a matching general lower bound of $\Omega(\log n)$. Our results show that having variable transmission ranges allows to achieve logarithmic convergecast latency, to the contrary with $\Omega(n)$ latency needed when the transmission ranges are fixed. Then we study the trade-off between the energy and the latency of convergecast.

Some interesting future research directions are to consider networks without collision detection, establish better bounds on the energy consumption of our algorithm and develop deterministic convergecast algorithms. Another open question is to find a trade-off between the latency and the power consumption for the whole range of possible latency values – here we argued only about two extreme cases, namely the latency of $\Theta(\log n)$ and $\Theta(n)$.

REFERENCES

- [1] N. Alon, A. Bar-Noy, N. Linial, D. Peleg. A lower bound for radio broadcast. *Journal of Computer and System Sciences*, 43:290–298, 1991.
- [2] V. Annamalai, S. K. S. Gupta, L. Schwiebert. On tree-based convergecasting in wireless sensor networks. *WCNC 2003 - IEEE Wireless Communications and Networking Conference*, 4(1):1942–1947, Mar. 2003.
- [3] R. Bar-Yehuda, O. Goldreich, A. Itai. On the time complexity of broadcast in radio networks: an exponential gap between determinism and randomization. *Journal of Computer and System Sciences*, 45:104–126, 1992.
- [4] M. Cagalj, J. P. Hubaux, C. Enz. Minimum-energy broadcast in all-wireless networks: NP-completeness and distribution issues. In *Proc. of the 8th Annual International Conference on Mobile Computing and Networking (MobiCom'02)*, 172–182, 2002.
- [5] G. Calinescu. Computing 2-Hop Neighborhoods in Ad Hoc Wireless Networks. In *Proc. Adhoc-Now '03*.
- [6] J. Cartigny, D. Simplot, I. Stojmenovic. Localized minimum-energy broadcasting in ad-hoc networks. In *Proc. of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'03)*, 2003.
- [7] W.T. Chen, N.F. Huang. The Strongly Connecting Problem on Multihop Packet Radio Networks. *IEEE Trans. Commun.*, 37(3):293–295, 1989.
- [8] I. Chlamtac, O. Weinstein. The wave expansion approach to broadcasting in multihop radio networks. *IEEE Trans. on Communications*, 39:426–433, 1991.
- [9] B.S. Chlebus, L. Gąsieniec, A. Lingas, A. Pagourtzis. Oblivious gossiping in ad-hoc radio networks. In *Proc. of 5th International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DIALM'2001)*, 44–51, 2001.
- [10] M. Christersson, L. Gąsieniec, A. Lingas. Gossiping with bounded size messages in ad-hoc radio networks. In *Proc. of 29th International Colloquium on Automata, Languages and Programming (ICALP'2002)*, 377–389.
- [11] M. Chrobak, L. Gąsieniec, D. Kowalski. The wake-up problem in multihop radio networks. In *Proc. of 15th ACM-SIAM Symposium on Discrete Algorithms (SODA'2004)*, 985–993, 2004.
- [12] M. Chrobak, L. Gąsieniec, W. Rytter. Fast broadcasting and gossiping in radio networks. In *Proc. of 41st Symposium on Foundations of Computer Science (FOCS'2000)*, 575–581, 2000.
- [13] A.E.F. Clementi, A. Monti, R. Silvestri. Selective families, superimposed codes, and broadcasting on unknown radio networks. In *Proc. of 12th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'2001)*, 709–718, 2001.
- [14] A. Czumaj, W. Rytter. Broadcasting algorithms in radio networks with unknown topology. In *Proc. 44th Ann. Symposium on Foundations of Computer Science (FOCS'2003)*, 492–501, 2003.
- [15] A. Dessmark, A. Pelc. Tradeoffs between knowledge and time of communication in geometric radio networks. In *Proc. of 13th Ann. ACM Symposium on Parallel Algorithms and Architectures (SPAA'2001)*, 59–66, 2001.
- [16] M. Fredigh, P. Johansson, P. Larsson. Wireless ad hoc networking - The art of networking without a network. *Ericsson Review*, 4:248–263, 2000.
- [17] I. Gaber, Y. Mansour. Broadcast in radio networks. In *Proc. of 6th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'1995)*, 577–585, 1995.
- [18] L. Gąsieniec, A. Lingas. On adaptive deterministic gossiping in ad hoc radio networks. *Information Processing Letters*, 83:89–94, 2002.
- [19] L. Gąsieniec, I. Potapov. Gossiping with unit messages in known radio networks. In *Proc. of 2nd IFIP International Conference on Theoretical Computer Science (TCS'2002)*, 193–205.
- [20] L. Gąsieniec, I. Potapov, Q. Xin. Time efficient gossiping in known radio networks. In *Proc. of 11th Colloquium on Structural Information and Communication Complexity (SIROCCO'2004)*, LNCS 3104, 173–184.
- [21] L. Gąsieniec, T. Radzik, Q. Xin. Faster deterministic gossiping in directed ad-hoc radio networks. In *Proc. of 9th Scandinavian Workshop on Algorithm Theory (SWAT'2004)*, LNCS 3111, 397–407.
- [22] J. Hightower and G. Borriello. Location Systems for Ubiquitous Computing. *IEEE Computer*, 34(8):57–66, 2001.
- [23] L.M. Kirousis, E. Kranakis, D. Krizanc, A. Pelc. Power consumption in packet radio networks. *Theoretical Computer Science*, 243:289–305, 2000.
- [24] D. Kowalski, A. Pelc. Broadcasting in undirected ad hoc radio networks. In *Proc. of 22nd ACM Symposium on Principles of Distributed Computing (PODC'2003)*, 73–82.
- [25] D. Kowalski, A. Pelc. Time complexity of radio broadcasting: adaptiveness vs. obliviousness and randomization vs. determinism. In *Proc. of 10th Colloquium on Structural Information and Communication Complexity (SIROCCO'2003)*, 195–210, 2003.
- [26] K. Krizman, T. Bieda and T. Rappaport. Wireless position location: fundamentals, implementation strategies, and source of error. In *Proc. of Veh. Tech. Conference*, 919–923, 1997.
- [27] E. Kushilevitz, Y. Mansour. An $\Omega(D \log(N/D))$ lower bound for broadcast in radio networks. *SIAM J. on Computing*, 27:702–712, 1998.
- [28] D. Liu, M. Prabhakaran. On randomized broadcasting and gossiping in radio networks. In *Proc. of 8th Annual International Conference on Computing and Combinatorics (COCOON'2002)*, LNCS 2387, 340–349, 2002.
- [29] A. Nasipuri and K.Lim. A Directionality based Location Discovery Scheme for Wireless Sensor Networks. In *Proc. of WSNA'2002*.
- [30] C. Perkins. Ad hoc networking: an introduction. In C. Perkins, ed. *Ad Hoc Networking*, 314–318, 2001.
- [31] T. S. Rappaport. *Wireless Communications: Principles and Practices*. Prentice Hall, 1996.
- [32] S. Upadhyayula, V. Annamalai, S. K. S. Gupta. A low-latency and energy-efficient algorithm for convergecast in wireless sensor networks. *GLOBECOM 2003 - IEEE Global Telecommunications Conference*, 22(1):3525–3530, Dec. 2003.
- [33] P. J. Wan, G. Calinescu, X. Y. Li, O. Frieder. Minimum-energy broadcast routing in static ad hoc wireless networks. *Wireless Networks*, 2002.
- [34] J. E. Wieselthier, G. D. Nguyen, A. Ephremides. Energy-Efficient Broadcast and Multicast Trees in Wireless Networks, *Mobile Networks and Applications (MONET)*, 7(6):481–492, 2002.