Fast Broadcasting and Gossiping in Radio Networks

Marek Chrobak* Leszek Gąsieniec† Wojciech Rytter ‡

Abstract

We establish an $O(n \log^2 n)$ upper bound on the time for deterministic distributed broadcasting in multi-hop radio networks with unknown topology. This nearly matches the known lower bound of $\Omega(n \log n)$. The fastest previously known algorithm for this problem works in time $O(n^{3/2})$. Using our broadcasting algorithm, we develop an $O(n^{3/2} \log^2 n)$ algorithm for gossiping in the same network model.

1. Introduction

A radio network can be viewed as a directed graph, whose nodes represent distributed processors and edges represent the relation of one node being in the range of another. All processors work synchronously and have unique identifiers; however, they do not know the network topology and they do not have the capability to detect message collisions.

The radio network abstraction captures the features of distributed communication networks with multi-access channels, with minimal assumptions on the channel model and processors' knowledge. Directed edges model unidirectional links, including situations in which one of two adjacent transmitters is more powerful than the other. In particular, there is no feedback (see, for example, [21]). In some applications, message collisions may be difficult to distinguish from the noise that is normally present on the channel, justifying the need for protocols that do not depend on the reliability of the collision detection mechanism (see [14, 13]). Some network configurations are subject to frequent changes. In other networks, topologies could be unstable or dynamic; for example, when mobile users are present. In such situations, algorithms that do not assume any specific topology are more desirable.

We consider two problems of disseminating information in radio networks: broadcasting and gossiping. In the *broadcasting problem*, one distinguished source node has a message that needs to be sent to all other nodes. In the *gossiping problem*, also known as the *total information exchange*, each node is initially given a different message that needs to be distributed to all other nodes.

One natural tool for dealing with uncertainty and conflicts in a distributed setting is randomization and, indeed, most earlier work on broadcasting in radio networks focussed on randomized algorithms. Bar-Yehuda *et al* [4] gave a randomized algorithm that achieves broadcast in expected time $O(D \log n + \log^2 n)$, where D denotes the network diameter. This is very close to the lower bound of $\Omega(D \log(n/D))$, by Kushilevitz and Mansour [19], and it matches this lower bound when $D = O(n^{1-\epsilon})$, for any $\epsilon > 0$. Further, if D is a constant, it also matches the lower bound of $\Omega(\log^2 n)$ for constant diameter networks, obtained by Alon *et al* [1].

In the deterministic case, Bar-Yehuda *et al* [4] gave an $\Omega(n)$ lower bound for constant diameter networks. For general networks, the best currently known lower bound of $\Omega(n \log n)$ was obtained by Bruschi and M. del Pinto [6] and, independently, by Chlebus *et al* [7]. In [7], the authors also present a broadcast algorithm with time complexity $O(n^{11/6})$ – the first sub-quadratic upper bound. This upper bound was later improved to $O(n^{5/3} \log^3 n)$ by De Marco and Pelc [12]. Recently, Chlebus *et al* [8] developed several broadcasting algorithms, including one with time complexity $O(n^{3/2})$, which is presently the best upper bound known. In an independent work, Peleg [23] gave an $O(n^{3/2}\sqrt{\log n})$ upper bound, using a probabilistic construction.

Our results. In this paper we prove that there is a radio broadcast algorithm with time complexity $O(n \log^2 n)$. This considerably improves the best known upper bound from [8] and it nearly matches the lower bound of $\Omega(n \log n)$ from [6, 7]. The proof is non-constructive. Using a probabilistic argument, we prove the existence of fam-

*Department of Computer Science, University of California, Riverside, CA 92521. Email: marek@cs.ucr.edu. Research done while the author was visiting University of Liverpool. Research supported by grant EPSRC GR/N09077.

†Department of Computer Science, University of Liverpool, Liverpool L69 7ZF, UK. Email: leszek@csc.liv.ac.uk. Research supported by grants EPSRC GR/N09077 and NUF-NAL (The Nuffield Foundation Awards to Newly Appointed Lecturers).

‡Instytut Informatyki, Uniwersytet Warszawski, Banacha 2, 02–097, Warszawa, Poland, and Department of Computer Science, University of Liverpool, Liverpool L69 7ZF, UK. Email: rytter@csc.liv.ac.uk. Research supported by grants EPSRC GR/N09077 and KBN 8T11C03915.

ilies of sets that we call *selectors*. Our algorithm is deterministic and uses these sets to coordinate the broadcast.

Next, in Section 5, we consider the problem of gossiping. Using our broadcasting algorithm, we develop an algorithm for gossiping with time complexity $O(n^{3/2} \log^2 n)$ which, to our knowledge, is the first sub-quadratic upper bound for this problem. Since this algorithm is based on our broadcasting algorithm, this upper bound is non-constructive. However, using our technique, any constructive algorithm for broadcasting with complexity $B(n)$ will automatically give an $O(n\sqrt{B(n)} \log n)$-time, constructive gossiping algorithm. In particular, using the broadcasting algorithm from [8], we get a constructive algorithm with complexity $O(n^{7/4} \log n)$.

In Section 6 we consider complete layered networks, in which all nodes in one layer are connected to all nodes in the next layer. We prove that for such layered networks both broadcasting and gossiping can be done in time $O(n \log n)$, which is optimal because of the lower bound from [6, 7].

Other related work. One major obstacle we need to deal with in radio networks is uncertainty – the processors do not have information about the network topology, and thus they also do not know, at a given time slot, whether they compete with other nodes for the channel, nor whether their past transmissions were successful. That knowledge of topology helps was shown by Gaber and Mansour [15], who gave a centralized broadcasting algorithm with running time $O(D + \log^5 n)$. Diks *et al* [10] gave efficient broadcasting algorithms for special types of known networks. It is also known that computing an optimal broadcast schedule for a given network is NP-hard, even for points in the plane, where the graph is induced by node ranges, see [9, 25].

The issue of processor synchronization in radio networks was studied in [16], where the authors study the differences between two levels of synchronization in a distributed broadcast system: global synchronization, where all processors have access to a global clock, and local synchronization, where the processors have local clocks ticking at the same rate but they lack initial time agreement.

2. Preliminaries

Radio networks. A detailed, formal model of a radio network was given in references [4, 8]. We describe briefly the features necessary for the proof of an upper bound.

A *radio network* is defined as a directed graph. Throughout the paper, by n we denote the number of nodes in this graph. If there is an edge from u to v, then we say that v is an *out-neighbor* of u and u is an *in-neighbor* of v. Each node is assigned a unique identifier from the set $\{1, 2, \ldots, n\}$. In the broadcast problem, one node, for example node 1, is distinguished as the *source node*. Initially, the nodes do not possess any other information. In particular, they do not know the network topology.

In the broadcasting problem, we assume that all nodes in the network are reachable from the source node. In the gossiping problem, we assume that the graph is strongly connected.

The time is divided into discrete time steps. All nodes start simultaneously, have access to a common clock, and work synchronously. (As noted by Peleg [23], the assumption about having access to a common clock is not necessary.) A broadcasting algorithm is a protocol that for each identifier *id* and for each time step t, given all past messages received by *id*, specifies whether *id* will transmit a message at time t, and if so, it also specifies the message. A message M transmitted at time t from a node u is sent instantly to all its out-neighbors. However, an out-neighbor v of u receives M at time step t only if no collision occurred, that is, if the other in-neighbors of v do not transmit at time t at all. Further, collisions cannot be distinguished from background noise. If v does not receive any message at time t, it knows that either none of its in-neighbors transmitted at time t, or that at least two did, but it does not know which of these two events occurred.

The running time of a broadcasting algorithm is the smallest t such that for any network topology, and any assignment of identifiers to the nodes, all nodes receive the source message no later than at step t. For gossiping, we require that each node receives each other node's message by time t.

Simplifying assumptions. For clarity of presentation, we will present our algorithms as if the nodes knew n, the size of the network. This assumption can be eliminated by a standard doubling technique (see [7, 8]) that works as follows: We organize the computation into phases, and we modify a given algorithm so that in phase i only nodes with labels at most 2^i participate in the algorithm. This does not change the asymptotic running time.

Further, we will also assume throughout the paper that n is a power of 2. For other n, the processors can execute the algorithm for the nearest power of 2 larger than n, without changing the asymptotic running time.

Algorithm ROUNDROBIN**.** A simple ROUNDROBIN algorithm avoids conflicts by global time-division multiplexing. It works in n rounds, where in each round the nodes transmit one by one, in order of their identifiers: $1, 2, \ldots, n$. ROUNDROBIN solves both the broadcasting and gossiping problems in time $O(n^2)$. There are no conflicts, so each message will reach all other nodes after at most n rounds. We will use ROUNDROBIN as a procedure in our algorithms.

3. Selectors

All sets considered in this section are subsets of $\{1, \ldots, n\}$. We say that a set S *hits* a set X iff $|S \cap X| = 1$, and that S *avoids* X iff $S \cap X = \emptyset$. Given a positive integer w, a family $\overline{S}$ of sets is called a w*-selector* if it satisfies the following property:

$(*)$ For any two disjoint sets X, Y with $w/2 \le |X| \le w$ and $|Y| \le w$ there exists a set in $\overline{S}$ which hits X and avoids Y.

Lemma 1 *For each n and each positive integer $w \le n$ there exists a w-selector $\overline{S}$ with $m = O(w \log n)$ sets.*

Proof: We use a probabilistic argument. Fix an integer m whose value will be determined later. For $i = 0, 1, \ldots, m-1$, let S_i be a random set obtained by independently including each element from $\{1, \ldots, n\}$ with probability $1/(w+1)$. We take $\overline{S} = \{S_0, \ldots, S_{m-1}\}$.

Fix i, and fix two disjoint sets X and Y with $|X| = x$ and $|Y| = y$, where $w/2 \le x \le w$ and $y \le w$. Then

$$
\begin{aligned}
&\Pr[\, S_i \text{ hits } X \text{ and avoids } Y \,] \\
&\quad = x\left(\frac{1}{w+1}\right)\left(1 - \frac{1}{w+1}\right)^{x-1}\left(1 - \frac{1}{w+1}\right)^{y} \\
&\quad = \frac{x}{w}\left(1 - \frac{1}{w+1}\right)^{x+y} \\
&\quad \ge \frac{1}{2}\left(1 - \frac{1}{w+1}\right)^{2(w+1)} \\
&\quad \ge \frac{1}{32}.
\end{aligned}
$$

The last inequality follows from the fact that the expression in the previous line is minimized for $w = 1$.

By the above estimate and the independence of the sets S_i, the probability that every S_i either does not hit X or does not avoid Y is at most $(31/32)^m$. Thus estimating the probability of a sum of events (over all pairs X, Y) by the sum of their probabilities, we get

$$
\begin{aligned}
&\Pr\big[\, \overline{S} \text{ is not a } w\text{-selector} \,\big] \\
&\quad \le \sum_{x=w/2}^{w} \binom{n}{x} \sum_{y=0}^{w} \binom{n}{y} (31/32)^m \\
&\quad \le w^2 n^{2w} (31/32)^m \\
&\quad \le n^{4w} (31/32)^m \\
&\quad \le 2^{4w \log n - m \log(32/31)} \\
&\quad < 1
\end{aligned}
$$

for $m = \frac{4}{\log(32/31)} w \log n + 1$. Therefore, for this m, some $\overline{S}$ will be a w-selector. $\Box$

When $w = O(1)$, the results from [7] imply that the upper bound on the selector size in Lemma 1 is tight. However, this bound is not tight for $w = \omega(n/\log n)$, since then we can take $\overline{S}$ to be the family of all singleton sets $\{1\}$, $\{2\}$, ..., $\{n\}$, so the size of $\overline{S}$ will be $n = o(w \log n)$.

4. The Broadcasting Algorithm

We now describe our broadcasting algorithm DoBroadcast. The algorithm is almost completely oblivious. It is specified as a sequence of *transmission sets*. At each time step t, the nodes that transmit the message are those that belong to the t-th transmission set and have already received the message.

For each $j = 0, \ldots, \log n$ let $\overline{S}_j = \big(S_{j,0}, S_{j,1}, \ldots, S_{j,m_j-1}\big)$ be a 2^j-selector with $m_j = O(2^j \log n)$ sets. Lemma 1 guarantees that such families $\overline{S}_j$ exist.

Algorithm DoBroadcast**.** The algorithm consists of *stages*, with each stage having $\log n + 1 = O(\log n)$ steps. The transmission set at the jth step of stage s is $S_{j, s \bmod m_j}$.

Theorem 1 *Algorithm* DoBroadcast *completes broadcasting in time $O(n \log^2 n)$.*

Proof: The proof is by amortized analysis, as in [8]. A node is called *dormant* if it has not yet received the source message. If a node has received the source message, then it is called a *frontier node* if it has a dormant out-neighbor, otherwise it is called *inner*. The state transitions are

dormant $\to$ frontier $\to$ inner

with the first transition occurring when a node receives a source message, and the second when all its out-neighbors receive the source message. It is possible that both transitions occur at the same time step. It is easy to see that if a node transmits as the only frontier node then it becomes inner.

Each time a node changes its status we say that *progress* has been made. Each change dormant $\to$ frontier or frontier $\to$ inner contributes a unit to the measure of progress. When progress reaches $2n - 1$, all the nodes are inner, and broadcasting is complete.

Fix some stage s. To prove the theorem, it is sufficient to show that we can find $s' \ge s$ such that in each stage $s, s+1, \ldots, s'$ the amortized progress is $\Omega(1/\log n)$. For if it is so, the number of stages sufficient to make progress $(2n-1)$ – and thus accomplish broadcasting – is $O(n \log n)$. Since every stage has $O(\log n)$ steps, the total time is $O(n \log^2 n)$.

Let F be the set of frontier nodes at the beginning of stage s, and let f be such that $2^{f-1} \le |F| < 2^f$. For each $j = 1, \ldots, f$, let Y_j be the set of nodes that received the message for the first time in stages $s, s+1, \ldots, s + m_j - 1$ (but were dormant when stage s started).

We have two sub-cases:
Case 1: There is j for which $|Y_j| \geq 2^j$.

In this case, after $m_j = O(2^j \log n)$ stages the progress is at least 2^j, so the amortized progress for each stage $s, s+1, \ldots, s+m_j-1$, is $\Omega(1/\log n)$.

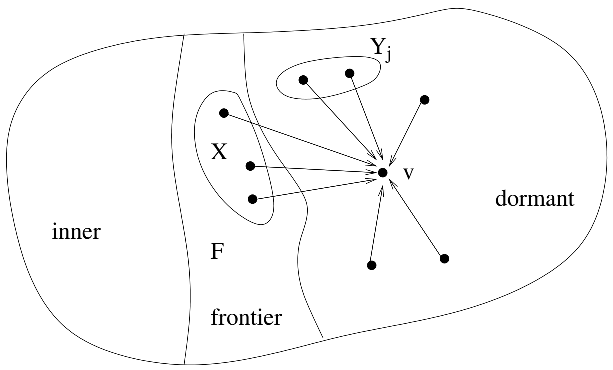

Figure 1. Illustration of Case 2 in the proof of Theorem 1.

Case 2: For each j we have $|Y_j| \leq 2^j$.

We show that in this case all nodes in F will become inner after m_f stages.

Fix any node v that is dormant when stage s starts, and whose set X of in-neighbors in F is not empty (See Figure 1.) Pick j such that $2^{j-1} \leq |X| < 2^j$. Since $2^j/2 \leq |X| \leq 2^j$ and $|Y_j| \leq 2^j$, by property $(*)$, family $\overline{S}_j$ contains a set $S_{j,i}$ that hits X and avoids Y_j. This set $S_{j,i}$ will occur in one of the stages $s, s+1, \ldots, s+m_j-1$.

All in-neighbors of v are either in X or are dormant when stage s starts. When we use $S_{j,i}$ for transmission then:

(i) exactly one in-neighbor of v in X will transmit because $|S_{j,i} \cap X| = 1$,

(ii) the nodes from Y_j will not interfere because $S_{j,i} \cap Y_j = \emptyset$,

(iii) the nodes that were dormant at the beginning of stage s and are not in Y_j remain dormant when $S_{j,i}$ is issued, so they will not transmit.

Therefore v will receive the message. We conclude that all nodes in F will become inner after m_f stages, so the amortized progress in each stage $s, s+1, \ldots, s+m_f-1$ is $\Omega(1/\log n)$. $\Box$

5. The Gossiping Algorithm

A simple, but important observation is that gossiping is not simply n simultaneous broadcasts, because nodes can collect many messages and encapsulate them all in one big message that can be transmitted in one step. Our algorithm takes advantage of this feature.

The essential idea of the algorithm is to reduce the problem to broadcasting. We assume that we are given a deterministic algorithm for broadcasting with running time $B(n)$.

We first show how to solve the following auxiliary task: Assume that each node v stores an integer $r_v \in \{0, \ldots, n\}$. The task is to have all nodes agree on some node $v = v_{\max}$ for which r_v is maximum.

The key idea is to use binary search combined with broadcasting. We also use the concept of *informing by silence*: if a node does not receive a message for a specified period of time then it can interpret this fact as a special information and act according to that.

Procedure FINDMAX.
We first find $r_{\max} = \max_v r_v$, using binary search. At each step, all nodes know that $r_{\max}$ is between a and b, where $a \leq b$. Initially $a = 0$ and $b = n$. If $a = b$, then $r_{\max} = a$, and the computation of $r_{\max}$ is complete.
For $a < b$, we proceed as follows. Let $c = \lceil (a+b)/2 \rceil$. Each node v for which $c \leq r_v \leq b$ sends the message $[c, b]$ to all other nodes. Since all these nodes send the same message, this is no different than broadcasting from a single node, so we can use just one application of the broadcasting algorithm. Then, after $B(n)$ steps, either all nodes will receive the message $[c, b]$, in which case they know that the maximum is between c and b, or (informing by silence) all nodes will not receive anything for time $B(n)$, in which case they know the maximum is between a and $c-1$. Depending on the outcome, either all nodes update their $[a, b]$ intervals (containing maximum) to $[\lceil (a+b)/2 \rceil, b]$ or all of them update it to $[a, \lceil (a+b)/2 - 1 \rceil]$.
Given $r_{\max}$, we determine $v_{\max}$ using binary search again. Let $r'_v = v$ if $r_v = r_{\max}$ and $r'_v = 0$ otherwise. Using the procedure from the previous paragraph applied to the values r'_v, we find the largest v for which $r_v = r_{\max}$, and we set $v_{\max} = v$.

Algorithm DOGOSSIP**.**
In the algorithm, each node v stores a set $K(v)$ of messages. Initially $K(v)$ consists of the single message that originates from v. Whenever v receives a message, this message is automatically added to $K(v)$. We can also delete some elements from $K(v)$.
The algorithm has two phases:

Phase 1:
Perform $\sqrt{B(n)} \log n$ rounds of ROUNDROBIN.
Phase 2:
repeat
Use FINDMAX to find a node $v_{\max}$ such that

$|K(v_{\max})| = \max_v |K(v)|$.
Distribute from $v_{\max}$ message $K(v_{\max})$ to all other nodes in the network.
For each node v, set $K(v) \leftarrow K(v) - K(v_{\max})$.
until $\max_v |K(v)| = 0$

Lemma 2 *If, in* DoGossip*, we use a broadcasting algorithm with time complexity $B(n)$, then* DoGossip *completes gossiping in time $O(n\sqrt{B(n)}\log n)$.*

Proof: Phase 1, obviously, runs in time $O(n\sqrt{B(n)}\log n)$, so it is sufficient to analyze Phase 2. After Phase 1, the sets $K(v)$ cover the whole set of messages, and thus Phase 2 can be viewed as the greedy algorithm for the set cover problem. Our analysis is similar to the one from [3] for dominating sets in large degree graphs.
Let $\alpha = \sqrt{B(n)}\log n/n$. Consider one iteration of the repeat loop in Phase 2. Suppose that there are m distinct messages when this iteration starts. Since initially, in Phase 1, each message was distributed to at least αn nodes, the number of message copies in the sets $K(v)$ is now at least αmn. Consequently, $\max_v |K(v)| \geq \alpha m$. Therefore, at the next iteration, the number of different messages will be reduced to at most $m - \max_v |K(v)| \leq (1-\alpha)m$. We start with $m = n$, so after i iterations we have at most $n(1-\alpha)^i$ messages left. Thus no messages will be left after $O(\frac{1}{\alpha}\cdot\log n) = O(n/\sqrt{B(n)})$ steps.

We conclude that the repeat loop is executed $O(n/\sqrt{B(n)})$ times. Procedure FindMax executes broadcasting $\log n$ times, so it runs in time $O(B(n)\log n)$. Therefore the total cost of Phase 2 is $O(n\sqrt{B(n)}\log n)$. □

Note: M. Robson [24] observed that it is possible to eliminate the binary search in Phase 2 by computing only an approximate maximum in each iteration. By further rebalancing the two phases, one can then reduce the running time by a sub-logarithmic factor.

If we use DoBroadcast from Section 4 to perform broadcasting in DoGossip, we obtain the running time of $O(n^{3/2}\log^2 n)$. The resulting algorithm is not explicit, as it depends on a probabilistic construction. (We use the term "explicit" in a somewhat informal manner. One can define "explicit" as efficiently computable, for example in polynomial time.) However, using the $O(n^{3/2})$-time broadcasting algorithm from [8], we get the running time of $O(n^{7/4}\log n)$. We summarize these results in the following theorem.

Theorem 2 *(a) There is a deterministic algorithm for gossiping in radio networks that runs in time $O(n^{3/2}\log^2 n)$.*
(b) There is an explicit deterministic algorithm for gossiping in radio networks that runs in time $O(n^{7/4}\log n)$.

6. Complete Layered Networks

The lower bound proofs for broadcasting in [7, 19] use complete layered networks, in which all nodes in one layer are connected to all nodes in the next layer. The idea behind the lower bounds is that all nodes in a layer have exactly the same information at each time step. Using this property and a counting argument, one can build a graph layer by layer, by choosing the nodes in each consecutive layer so that they collide for as long as possible.

Our next result shows that there is an $O(n\log n)$-time broadcasting algorithm for such layered networks. Therefore similar methods cannot work for proving a lower bound on broadcasting higher than $\Omega(n\log n)$.

Formally, a *complete layered network* is a graph consisting of layers $L_0,\ldots,L_{m-1}$, in which we have edges from each node in layer L_i to each node in layer L_{i+1} for $i < m$. Layer L_0 contains only the source node s.

Theorem 3 *There is a deterministic $O(n\log n)$-time algorithm for brodcasting in complete layered radio networks.*

Proof: The proof is similar to that of Theorem 1 (in fact, it is substantially simpler, since we do not need to use the sets Y_j to eliminate interferences), so we omit the details.

For $j = 0,\ldots,\log n$, let $\overline{S}_j$ be a 2^j-selector with $O(2^j\log n)$ sets. Take $\overline{T} = \overline{S}_0\overline{S}_1\ldots\overline{S}_{\log n}$. Then $\overline{T}$ consists of $O(n\log n)$ sets. The algorithm is simple: once a node receives the source message, it begins to transmit according to the sequence $\overline{T}$.

If a given layer has size w, then it will successfully transmit the message to the next layer after $O(w\log n)$ steps, by the definition of $\overline{T}$ and by Lemma 1. Amortizing, the total running time will be $O(n\log n)$. □

Next, we show that we can achieve the same running time even for gossiping. Since gossiping requires the network to be strongly connected, we slightly modify the definition of complete layered networks. Define a *complete cyclic layered network* to be a graph consisting of layers $L_0,\ldots,L_{m-1}$, in which we have edges from each node in layer L_i to each node in layer L_{i+1} for $i < m$ and, similarly, from layer L_{m-1} to L_0.

Theorem 4 *There is a deterministic $O(n\log n)$-time algorithm for gossiping in complete cyclic layered radio networks.*

Proof: In the first stage the algorithm executes one round of RoundRobin. After this stage, each node knows all the messages from the previous layer. In the next stage, one node, say node 1, initiates a broadcast procedure similar to the one described in the proof of Theorem 3. However, instead of forwarding one fixed source message, the nodes attach to the received message all the messages from the previous layer that were collected in the ROUNDROBIN stage. Assume, without loss of generality, that node 1 is in layer L_0. After the broadcast is complete, the nodes in layer L_{m-1} know all messages in the network. Using the information about the network (that can also be disseminated in the broadcast), they also know that they are in the last layer. So they now initiate one more broadcast that will distribute all messages to all nodes in the network. □

7. Final Comments

Several open problems remain. Our $O(n \log^2 n)$ upper bound for broadcasting is not constructive, and it would be interesting to find an explicit construction of a broadcasting protocol with this running time. By Lemma 2, this would also give a fast, constructive algorithm for gossiping. In the construction in [8], the transmission sets are lines in a finite geometry. The problem of constructing w-selectors is similar, in flavor, to some problems in non-adaptive group testing [11] (with the nodes being the soldiers to be tested for syphillis), multi-access communications [26, 17], coding, and some other problems in combinatorial designs, but we are not aware of any construction that can be applied directly to obtain w-selectors of desired size.

The exact complexity of broadcasting remains an open problem, although the gap between the lower and upper bounds is now only a factor of $\log n$. The complexity of gossiping remains open as well. Our results from Section 6 indicate that, if any of these problems requires more than $O(n \log n)$ time, to prove such a lower bound we will need to find techniques different from those used in the $\Omega(n \log n)$ bound on broadcasting.

Acknowledgements. We would like to thank the anonymous referees for valuable comments that helped us improve the presentation.

References

[1] N. Alon, A. Bar-Noy, N. Linial and D. Peleg, A lower bound for radio broadcast, *Journal of Computer and System Sciences* 43 (1991) 290–298.

[2] N. Alon, A. Bar-Noy, N. Linial and D. Peleg, Single round simulation of radio networks, *Journal of Algorithms* 13 (1992) 188–210.

[3] N. Alon, J. Spencer and P. Erdös, *The Probabilistic Method*, John Wiley & Sons, 1992.

[4] R. Bar-Yehuda, O. Goldreich, and 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 (1992) 104–126.

[5] R. Bar-Yehuda, A. Israeli, and A. Itai, Multiple communication in multi-hop radio networks, *SIAM Journal on Computing* 22 (1993) 875–887.

[6] D.Bruschi and M. del Pinto, Lower bounds for the broadcast problem in mobile radio networks, *Distributed Computing* 10 (1997) pp. 129-135

[7] B.S. Chlebus, L. Gąsieniec, A.M. Gibbons, A. Pelc, and W. Rytter, Deterministic broadcasting in unknown radio networks, in *Proc. 11th Ann. ACM-SIAM Symp. on Discrete Algorithms*, San Francisco, California, 2000, pp. 861–870.

[8] M. Chlebus, L. Gąsieniec, A. Östlin and J.M. Robson, Deterministic broadcasting in radio networks, *Proc. 27th International Colloquium on Automata, Languages and Programming,* ICALP'2000, to appear.

[9] I. Chlamtac and S. Kutten, On broadcasting in radio networks - problem analysis and protocol design, *IEEE Transactions on Communications* 33 (1985) 1240–1246.

[10] K.Diks, E.Kranakis, D.Krizanc, A.Pelc, The impact of knowledge on broadcasting time in radio networks, In *Proc. 7th European Symp. on Algorithms* (1999) 41-52

[11] D-Z. Du and F.K. Hwang, *Combinatorial Group Testing and its Applications*, World Scientific, 1993.

[12] G. De Marco and A. Pelc, Faster broadcasting in unknown radio networks, unpublished manuscript, 1999.

[13] A. Ephremides and B. Hajek, Information theory and communication networks: an unconsummated union, *IEEE Trans. on Inf. Theory*, IT-44 (1998) 2416-2434.

[14] R. Gallager, A perspective on multiaccess communications, *IEEE Trans. on Inf. Theory*, IT-31 (1985) 124-142.

[15] I. Gaber and Y. Mansour, Broadcast in radio networks, in *Proc. 6th Ann. ACM-SIAM Symp. on Discrete Algorithms*, 1995, pp. 577–585.

[16] L. Gąsieniec, A. Pelc and D. Peleg, Radio communication in locally synchronous broadcast systems, Proc. 19th Ann. ACM Symposium on Principles of Distributed Computing (PODC'2000), Portland, Oregon, U.S.A., July 2000, ACM Press, pp. 113-122.

[17] J. Komlos and A.G. Greenberg, An asymptotically nonadaptive algorithm for conflict resolution in multiple-access channels, *IEEE Trans. on Inf. Theory*, IT-31 (1985) 302-306.

[18] E. Kranakis, D. Krizanc and A. Pelc, Fault-tolerant broadcasting in radio networks, in *Proc. 6th European Symposium on Algorithms*, Venice, Italy, 1998, Springer LNCS 1461, pp. 283–294.

[19] E. Kushilevitz and Y. Mansour, An $\Omega(D \lg(N/D))$ lower bound for broadcast in radio networks, *SIAM Journal on Computing* 27 (1998) 702–712.

[20] E. Kushilevitz and Y. Mansour, Computation in noisy radio networks, in *Proc. 9th Ann. ACM-SIAM Symp. on Discrete Algorithms*, 1998, pp. 236–243.

[21] J.L. Massey and P. Mathys, The collision channel without feedback, *IEEE Trans. on Inf. Theory*, IT-31 (1985) 192-204.

[22] K. Pahlavan and A. Levesque, *Wireless Information Networks*, Wiley-Interscience, New York, 1995.

[23] D. Peleg, Deterministic radio broadcast with no topological knowledge, manuscript, 2000.

[24] M. Robson, private communication, 2000.

[25] A. Sen and M. L. Huson, A new model for scheduling packet radio networks, in *Proc. 15th Ann. Joint Conference of the IEEE Computer and Communication Societies*, 1996, pp. 1116–1124.

[26] J.K. Wolf, Born again group testing: multi-access communications, *IEEE Trans. on Inf. Theory*, IT-31 (1985) 185-191.