Algorithmica (2007) 47: 97–117
DOI: 10.1007/s00453-006-1212-3

Algorithmica

Deterministic Communication in Radio Networks with Large Labels[1]

Leszek Gąsieniec,[2] Aris Pagourtzis,[3] Igor Potapov,[2] and Tomasz Radzik[4]

Abstract. We study *deterministic* gossiping in ad hoc radio networks with *large node labels*. The labels (identifiers) of the nodes come from a domain of size N which may be much larger than the size n of the network (the number of nodes). Most of the work on deterministic communication has been done for the model with *small labels* which assumes $N = O(n)$. A notable exception is Peleg's paper [32], where the problem of deterministic communication in ad hoc radio networks with large labels is raised and a deterministic broadcasting algorithm is proposed, which runs in $O(n^2 \log n)$ time for N polynomially large in n. The $O(n \log^2 n)$-time deterministic broadcasting algorithm for networks with small labels given by Chrobak et al. [11] implies deterministic $O(n \log N \log n)$-time broadcasting and $O(n^2 \log^2 N \log n)$-time gossiping in networks with large labels. We propose two new deterministic gossiping algorithms for ad hoc radio networks with large labels, which are the first such algorithms with subquadratic time for polynomially large N. More specifically, we propose:

- a deterministic $O(n^{3/2} \log^2 N \log n)$-time gossiping algorithm for *directed* networks; and
- a deterministic $O(n \log^2 N \log^2 n)$-time gossiping algorithm for *undirected* networks.

Key Words. Ad hoc radio networks, Communication protocols, Gossiping, Selective families of sets.

1. Introduction. Mobile *radio networks* are expected to play an important role in future commercial and military applications [35]. Such networks are particularly suitable for situations where instant infrastructure is needed and no central system administration (such as base stations in a cellular system) is available. Typical applications for this type of peer-to-peer networks include: mobile computing in remote areas; tactical communications; law enforcement operations; and disaster recovery.

Radio networks, as well as any other distributed communication systems, demand efficient implementation of communication primitives to carry out more complex communication tasks. There are two important communication primitives encountered in the process of information dissemination in networks: *broadcasting* and *gossiping*. In the *broadcasting problem* a message from a distinguished *source* node has to be sent to all

[1] A preliminary version of this paper, with weaker bounds, was presented at ESA 2002 [21]. This research was supported in part by GR/N09855 and GR/R85921 EPSRC grants. Part of this work was done while A. Pagourtzis was with the Department of Computer Science, University of Liverpool, and the Institute of Theoretical Computer Science, ETH Zürich.

[2] Department of Computer Science, University of Liverpool, Liverpool L69 7ZF, England. {leszek,igor}@csc.liv.ac.uk.
[3] School of Electrical and Computer Engineering, National Technical University of Athens, 15780 Zografou, Greece. pagour@cs.ntua.gr.
[4] Department of Computer Science, King's College London, Strand, London WC2R 2LS, England. radzik@dcs.kcl.ac.uk.

Received May 27, 2004; revised September 20, 2005. Communicated by L. Arge.
Online publication December 20, 2006.

other nodes. In the *gossiping problem* each node of the network possesses its unique message that is to be communicated to all other nodes in the network.

In this paper we consider a model of synchronous ad hoc radio networks with no faults and with messages of unbounded size. In a radio network a node v receives successfully in the current communication step a message from its neighbor w, if and only if v is in the *listening* mode and w is the only (in-)neighbor of v in the *transmitting* mode during this step. We assume that node v cannot detect *collisions*, that is, it cannot distinguish between the case when two or more of its neighbors transmit in the same step and the case when none of its neighbors transmits.

The "ad hoc" nature of the network means that initially the nodes do not know anything about the topology of the network, and this includes not knowing anything about the immediate neighbors. Each node knows however its unique label (identifier), which is drawn from the set $\{1, 2, \ldots, N\}$. If $N = O(n)$, where n is the size of the network (the number of nodes), then we say that nodes have *small labels*. If we do not assume any upper bound on N, so the domain of the node labels may be considerably larger than the size of the network, then we say that nodes have *large labels*. The main difficulty in designing communication algorithms for ad hoc radio networks is the symmetry arising when a number of nodes try to send messages to a common neighbor. Such symmetry could potentially lead to perpetual collisions. In this paper we propose deterministic gossiping algorithms for ad hoc radio networks with large node labels.

Next we discuss previous results concerning broadcasting and gossiping in ad hoc radio networks. We first consider networks with small labels, since most of the previous work has been concerned only with this case. Then we summarize what has been known prior to our work, or can be easily inferred from related results, about deterministic communication in networks with large labels. The results discussed below apply to both directed and undirected networks, unless explicitly stated that a given upper bound applies only to undirected networks, or a given lower bound applies only to directed networks (undirected, or symmetric, networks should be viewed as a special case of directed networks). The running time of an algorithm in the model which we consider is the number of (communication) steps required to complete the algorithm.

Previous results: *broadcasting*. The study of communication in ad hoc radio networks has been mostly devoted to the broadcasting problem. A natural tool for breaking the symmetry of nodes trying to transmit to a common neighbor is randomization. Most earlier work on broadcasting in radio networks indeed focused on randomized algorithms. Bar-Yehuda et al. [2] gave a randomized algorithm that achieved broadcast in expected time $O(D \log n + \log^2 n)$, where D is the diameter of the network. This is very close to the lower bound of $\Omega(D \log(n/D))$ shown by Kushilevitz and Mansour [30], and it matches this lower bound when D is $\Omega(\log n)$ and $O(n^{1-\varepsilon})$, for any $\varepsilon > 0$. Furthermore, if D is $O(\log n)$, it matches the lower bound of $\Omega(\log^2 n)$ obtained by Alon et al. [1] for constant diameter networks. The remaining gap between the lower and upper bounds has recently been closed by Czumaj and Rytter [16], who describe a randomized broadcasting algorithm which runs in optimal $O(D \log(n/D) + \log^2 n)$ time with high probability. The randomized broadcasting algorithms given in [2] and [16] do not actually use node labels, so they work on *anonymous networks* where nodes do not have any distinct labels.

In the deterministic case distinct node labels are necessary to break the symmetry, or otherwise broadcasting would not be possible even in a four-node undirected cycle. The following ROUND-ROBIN procedure is frequently used in deterministic communication algorithms for networks with small labels: for $i = 1, 2, \ldots, N$, the node with label i transmits in step i. After one application of this procedure each node knows all its neighbors, and $n-1$ applications of this procedure are sufficient to complete both broadcasting and gossiping. This gives straightforward $O(n^2)$-step deterministic broadcasting and gossiping algorithms for networks with small node labels. Bar-Yehuda et al. [2] gave a lower bound $\Omega(n)$ for deterministic broadcasting in constant diameter networks.[5] Brusci and Del Pinto [4] showed a lower bound $\Omega(D \log n)$, and then Kowalski and Pelc [27] gave an $\Omega(n \log n / \log(n/D))$ bound. The hard networks used in the proofs of the lower bounds given in [2], [4], and [27] are *undirected*. Clementi et al. [12], [15] showed that for any deterministic broadcasting algorithm there are *directed* networks on which this algorithm runs in $\Omega(n \log D)$ time.

Chlebus et al. [6] presented a broadcasting algorithm with time complexity $O(n^{11/6})$, achieving the first subquadratic upper bound. De Marco and Pelc [17] improved this upper bound to $O(n^{5/3} \log^3 n)$, and Chlebus et al. [8], using finite geometries, improved it further to $O(n^{3/2})$. Recently Chrobak et al. [11] developed a deterministic algorithm for broadcasting in ad hoc radio networks working in time $O(n \log^2 n)$. This bound was further improved to $O(n \log n \log D)$ by Kowalski and Pelc [29], and then to $O(n \log^2 D)$ by Czumaj and Rytter [16]. The $O(n \log^2 n)$-time deterministic broadcasting algorithms presented in [11], [29], and [16] are non-constructive. Indyk [25] gave an alternative constructive solution with a similar $O(n \log^{O(1)} n)$ complexity. Clementi et al. [12] presented a deterministic broadcasting algorithm for ad hoc radio networks working in time $O(D\Delta \log^2 n)$, where Δ is the upper bound on the in-degrees of nodes. Kowalski and Pelc [27] showed an $O(n \log n)$-time deterministic broadcasting algorithm for undirected networks. The lower and upper bounds for broadcasting mentioned above assume that nodes do not transmit spontaneously. That is, each node may transmit only after receiving the source message. Chlebus et al. [6] showed that if spontaneous transmissions are allowed, then broadcasting in an undirected network can be completed in $O(n)$ time.

Previous results: *gossiping*. Until recently, there has not been much known about gossiping in ad hoc radio networks. A discussion was initiated by Chrobak et al. in [11], where they proved the existence of a subquadratic $O(n^{3/2} \log^2 n)$-time deterministic gossiping algorithm in such networks. A constructive counterpart of their algorithm was proposed later by Indyk [25]. Clementi et al. [12] presented an alternative deterministic gossiping algorithm working in time $O(D\Delta^2 \log^2 n)$. Recently Gąsieniec and Lingas introduced in [20] a new gossiping paradigm leading to two deterministic algorithms with time complexities $O(n\sqrt{D} \log^2 n)$ and $O(D\Delta^{3/2} \log^3 n)$. Ying Xu [36] modified their first algorithm and obtained a better bound of $O(n\sqrt{D} + n \log^2 n)$. Gąsieniec and Lingas's $O(n\sqrt{D} \log^2 n)$-time algorithm [20] and its modification proposed by Ying Xu [36]

[5] Bar-Yehuda et al. [2] claimed that the $\Omega(n)$ bound held for a stronger model where nodes know their neighborhood. A counterexample has been recently shown by Kowalski and Pelc [26], but the $\Omega(n)$ bound does hold for the model we consider in this paper.

are nonconstructive. Their constructive versions, which can be obtained using Indyk's results [25], have running times $O(n\sqrt{D}\log^{O(1)} n)$ and $O(n\sqrt{D} + n\log^{O(1)} n)$, respectively.

In [10] Chrobak et al. showed a randomized gossiping algorithm with expected running time $O(n\log^4 n)$. This bound has been subsequently improved to $O(n\log^3 n)$ by Liu and Prabhakaran [31], and then to $O(n\log^2 n)$ by Czumaj and Rytter [16].

In the special case of undirected networks, Chlebus et al.'s deterministic $O(n)$-time broadcasting algorithm which requires spontaneous transmissions [6] can also be used to gather in the source the messages from all other nodes. Thus two applications of this algorithm complete gossiping, provided that first the nodes select (agree on) the source node. Since the selection of the source (the *leader selection* computation) can be done by $O(\log N)$ applications of a broadcasting algorithm [8], gossiping in undirected networks with small labels can be done in $O(n\log n)$ time. If $N = n$, then the node with label 1 can assume the role of the source and gossiping can be completed in $O(n)$ time. Observe that spontaneous transmissions are not an issue in the gossiping problem, since all nodes become active at the same time.

Previous results: *large node labels*. All deterministic broadcasting and gossiping algorithms mentioned above were introduced for networks with small labels. Peleg [32] raised the problem of deterministic communication in ad hoc radio networks with large labels, and proposed a deterministic broadcasting algorithm running in $O(n^2\log n)$ time for N polynomially large in n. Some deterministic broadcasting and gossiping algorithms which were introduced for networks with small labels can be easily adapted to networks with large labels, retaining their asymptotic upper bounds, if N is polynomially large in n. For example, the $O(n\log^2 n)$-time and $O(D\Delta\log^2 n)$-time broadcasting algorithms for networks with small labels given in [11] and [12], respectively, become $O(n\log N\log n)$-time and $O(D\Delta\log N\log n)$-time algorithms for networks with large labels. The $O(D\Delta^2\log^2 n)$-time and $O(D\Delta^{3/2}\log^3 n)$-time gossiping algorithms for networks with small labels presented in [12] and [20] become $O(D\Delta^2\log N\log n)$-time and $O(D\Delta^{3/2}\log^2 N\log n)$-time algorithms, respectively, for networks with large labels. On the other hand, we do not see how one could adapt to large labels the $O(n^{3/2}\log^{O(1)} n)$-time gossiping algorithms presented in [11] and [25] and the $O(n\sqrt{D}\log^2 n)$-time gossiping algorithms presented in [20] and [36], since these algorithms rely crucially on the ROUND-ROBIN procedure.

Using the following n-round process, gossiping can be completed in $O(nT_B\log N)$ time, where T_B is the time of one broadcast operation. In the ith round the node with the ith smallest label is selected in $O(T_B\log N)$ time [8], and then the selected node broadcasts its message to all other nodes. Thus using the $O(n\log N\log n)$-time broadcasting algorithm which can be derived from [11], we can obtain an $O(n^2\log^2 N\log n)$-time gossiping algorithm for networks with large labels. A little better upper bound $O(n^2\log N)$ can be obtained for undirected networks: Gossiping can be completed in $O(n\log N)$ time, similarly as in the case with small labels discussed above, provided that each node knows all its neighbors. The nodes can learn about their all neighbors in total $O(n^2\log N)$ time [12], [15].

The lower bound results presented by Clementi et al. in [15] generalized to networks with large labels give an $\Omega(n\log(N/n))$ bound on broadcasting and gossiping.

Table 1. Deterministic gossiping in radio networks: unknown topology, unbounded messages.

	Best previous results		Our results	
	Directed	Undirected	Directed	Undirected
Small labels	$O(n^{3/2}\log^2 n)$ [11] $O(n\sqrt{D}+n\log^2 n)$ [20], [36] $O(D\Delta^{3/2}\log^3 n)$ [20]	$O(n\log n)$[a]		
Large labels	$O(n^2\log^2 N\log n)$[b] $O(D\Delta^{3/2}\log^2 N\log n)$[d]	$O(n^2\log N)$[c]	$O(n^{3/2}\log^2 N\log n)$	$O(n\log^2 N\log^2 n)$

[a] Based on leader selection and the $O(n)$-time broadcasting algorithm from [6].
[b] Based on repeated leader selection and $O(n\log N\log n)$-time broadcasting which follows from [11].
[c] Based on learning about neighborhoods and the $O(n)$-time broadcasting algorithm from [6].
[d] Follows from [20].

Our results. Prior to our work, no deterministic $O(n^{2-\varepsilon}\log^{O(1)} N)$-time gossiping algorithm for ad hoc radio networks with large labels has been known, not even for the special case of undirected networks. We propose two new deterministic gossiping algorithms that avoid using the ROUND-ROBIN procedure, which is prohibitively costly for networks with large labels. Our first algorithm runs in $O(n^{3/2}\log^2 N\log n)$ time in directed networks. If N is polynomially large in n, then this upper bound on deterministic gossiping in directed networks with large labels is within an $O(\log n)$ factor from the best previous upper bound on gossiping in networks with small labels. We also show that our first algorithm can be specialized to run on undirected networks in $O(n^{4/3}\log^2 N\log n)$ time. Our second algorithm is designed for undirected networks only and runs in $O(n\log^2 N\log^2 n)$ time. This bound is almost linear, if N is polynomially large in n. Table 1 shows our new bounds and the previous best bounds on deterministic gossiping in the model which we consider. Our algorithms are nonconstructive since they are based on the existence of certain combinatorial structures but not on their construction. Using the constructions described by Indyk [25], our algorithms, similarly to Chrobak et al.'s broadcasting and gossiping algorithms [11], can be made constructive with only $O(\log^{O(1)} N)$ slowdown.

In the initial version of this work presented in [21] we showed a weaker $O(n^{5/3}\log^2 N\log n)$ bound on deterministic gossiping in directed networks. Very recently, and subsequently to our work, further refinement of our techniques has led to an improved $O(n^{4/3}\log^{O(1)} N)$ bound on gossiping in directed networks [24].

Related models of radio networks. Broadcasting and gossiping have also been considered in other models of radio networks. For example, the problems of existence and efficient construction of fast deterministic broadcasting schedules for *known* networks have been extensively studied. In this model a priori knowledge of the whole topology of the network is assumed. Chlamtac and Weinstein [5] showed that for any directed network and the source node, an $O(D\log^2 n)$-step broadcasting schedule can be constructed in polynomial time. We use such schedules as subroutines in our gossiping algorithm for undirected networks, applying them to subgraphs which have already learned about their complete topology. The existence of an $O(D\log(n/D)+\log^2 n)$-step broadcasting schedule follows from the randomized broadcasting algorithm given by Czumaj and Rytter [16], while a polynomial-time construction of an $O(D\log n+\log^2 n)$-step broadcasting schedule was given by Kowalski and Pelc [28]. Papers [19], [18], and [22] showed $O(D+\log^p n)$-step broadcasting schedules for undirected networks, for $p = 5, 4, 3$, respectively. Clementi et al. [14] studied deterministic broadcasting in known radio networks in the presence of both static as well as dynamic faults.

Another interesting setting, where broadcasting and gossiping coincide, was studied by Chlebus et al. in [7]. They presented several upper and lower bounds on the time complexity of the so-called *oblivious* (*nonadaptive*) gossiping algorithms in ad hoc radio networks.

Deterministic gossiping in radio networks has been recently studied for a model with messages of bounded size. Christersson et al. [9] considered so-called $b(n)$-*gossiping* in ad hoc radio networks, which allows each node to send only at most $b(n)$ initial messages in one step. They were interested in obtaining a good, small, bound $b(n)$ without increasing the asymptotic time of gossiping with unbounded messages. Clementi et al. [13], [15] considered *multi-broadcasting*, the problem of performing $r \geq 1$ simultaneous and independent broadcasting operations, in ad hoc radio networks with messages of size $O(\log n+\log r)$. Results on deterministic gossiping in *known* radio networks when combining messages is not allowed can be found in [23].

Ravishankar and Singh presented a distributed gossiping algorithm for networks with nodes placed uniformly randomly on a line [33] and on a ring [34]. Bar-Yehuda et al. [3] studied probabilistic protocols for k-point-to-point communication and k-broadcast in radio networks with known local neighborhoods and bounded-size messages.

2. Preliminaries. A radio network is modeled as a graph $G = (V, E)$, where the nodes in set V represent transmitters/receivers and the edges in set E represent connections (links) available in the network. The size of the network n is the number of nodes. The graph can be *directed*, representing the case where the information can be passed along an edge only in the direction of the edge, or *undirected*, representing the case where the information can flow in both directions along every edge (but only in one direction at a time). An undirected network should be viewed as a special case of a directed network where the set of edges is *symmetric*. Node w is a *neighbor* of node v, if there is a link from w to v. The set of all neighbors of a node v is called the *neighborhood* of v. We denote by $d(v)$ the degree of node v, that is, the number of neighbors of node v. To ensure that gossiping is feasible, we assume that the graph of connections is strongly connected.

The nodes in the network communicate in synchronous steps. In each step a node may choose to be in one of two modes: either the *listening mode* or the *transmitting mode*. A node in the transmitting mode sends a message along all its out-going links. A message is delivered to all the recipients in the same step when it is sent. A node v receives a message delivered along its in-coming link if and only if this is the only in-coming link to v delivering a message during this step. If two or more links are delivering messages to v during the same step, then a *collision* occurs and v does not receive any message. We assume that *collision detection* is not available.

Each node knows its unique label, drawn from the set $\{1, 2, \ldots, N\}$, and the number N, but does not know anything about the topology of the network. We assume that all nodes know also the size of the network n. It would actually suffice if the nodes knew only the (same) linear upper bound on n instead of knowing the exact value of n. This assumption could be dropped altogether and replaced with the standard doubling technique (see [8]). The communication task would be completed within the same asymptotic running time, but the nodes would not know when completion occurred (see [6] for a discussion of this issue).

Each node $v \in V$ in the network has initially its unique message m_v. A *gossiping algorithm* is a communication protocol such that at its completion each node $w \in V$ knows all messages m_v, $v \in V$. The running time of an algorithm is the number of communication steps in this algorithm. We do not count the internal computation done in the nodes. There is no bound on the size of messages transmitted in one step. For convenience we assume that when a node v transmits during the current step, then it sends a message containing the whole knowledge it has accumulated so far. It can be shown, however, that the size of each message in our algorithms can be polynomial in n and in the sizes of the initial messages m_w.

We say that a set S *hits* a set X if and only if $|S \cap X| = 1$, and that S *avoids* X if and only if $S \cap X = \emptyset$. A set S hits a set X on an element x if $S \cap X = \{x\}$. A family of sets $\mathcal{R}$ hits a set X (hits a set X on an element x) if at least one set $S \in \mathcal{R}$ hits X (hits X on x, respectively). Consider a node v and let W be the set consisting of v and all its neighbors. Node v receives a message from its neighbor w in a given step if and only if the set of the nodes transmitting in this step hits set W on w.

Families of sets as communication patterns. Throughout the paper we use fixed finite families of subset of $\{1, 2, \ldots, N\}$ as deterministic communication procedures. By applying, or running, a given family $\mathcal{R}$ of m subsets of $\{1, 2, \ldots, N\}$ we mean the following m-step communication procedure. All nodes in the network know the members of family $\mathcal{R}$ arranged in some arbitrary but fixed order $(S_1, \ldots, S_m)$. At step i, for $i = 1, 2, \ldots, m$, the nodes with labels in set S_i are in the transmitting mode while the other nodes are in the listening mode. This procedure can be applied also only to a subset $W \subseteq V$, provided that each node $v \in V$ knows whether it belongs to W: at step i the nodes with labels in $W \cap S_i$ are in the transmitting mode while the other nodes are in the listening mode. We now recall some special families of sets which are useful in designing deterministic communication algorithms in ad hoc radio networks.

Selective family. We say that a family $\mathcal{R}$ of subsets of $\{1, 2, \ldots, N\}$ is *k-selective*, for a positive integer k, if for any nonempty set $Z \subseteq \{1, 2, \ldots, N\}$ such that $|Z| \leq k$, there is a set in $\mathcal{R}$ which hits Z. In the context of radio networks, for example, running a δ-selective family on a set of nodes $V \setminus \{v\}$ ensures that node v receives a message from at least one of its neighbors, if $d(v) \leq \delta$. It is known that for each pair of positive integers N and k, there exists a k-selective family of size $O(k \log N)$, see [12].

Strongly selective family. We say that a family $\mathcal{R}$ of subsets of $\{1, 2, \ldots, N\}$ is *strongly k-selective*, for a positive integer k, if for any nonempty set $Z \subseteq \{1, 2, \ldots, N\}$ such that $|Z| \leq k$ and any $z \in Z$, there is a set in $\mathcal{R}$ which hits Z on z. Running a strongly $(\delta + 1)$-selective family on all nodes of a radio network ensures that each node v with $d(v) \leq \delta$

receives messages from all its neighbors. (Consider the set $Z = \{v, w_1, w_2, \dots, w_j\}$, where $w_1, w_2, \dots, w_j$ are the neighbors of v.) It is known that for each pair of positive integers N and k, there exists a strongly k-selective family of size $O(k^2 \log N)$, see [12].

Linearly selective family. In the context of designing efficient gossiping algorithms, selective families may be too weak while strongly selective families may be too expensive. The following definition introduces a property which lies in between these two extremes. We say that a family $\mathcal{R}$ of subsets of $\{1, 2, \dots, N\}$ is *linearly k-selective*, for a positive integer k, if for any nonempty subset $Z \subseteq \{1, 2, \dots, N\}$ such that $|Z| \leq k$, $\mathcal{R}$ hits Z on more than half of the elements in Z. Running a linearly $(\delta + 1)$-selective family on all nodes of a radio network ensures that each node v with $d(v) \leq \delta$ receives messages from at least half of its neighbors. The notion of a linearly k-selective family is stronger than the notion of a k-selective family, but we show next, using *selectors*, that the size of the former can also be $O(k \log N)$, as the size of the latter.

Selector. We say that a family $\mathcal{R}$ of subsets of $\{1, 2, \dots, N\}$ is a *k-selector*, for a positive integer k, if for any two disjoint sets $X, Y \subseteq \{1, 2, \dots, N\}$ with $k/2 \leq |X| \leq k$ and $|Y| \leq k$, there exists a set in $\mathcal{R}$ which hits X and avoids Y. It was proven in [11] that for each pair of positive integers N and k, there exists a k-selector of size $O(k \log N)$. Selectors satisfy the following property.

LEMMA 1. *Let $\mathcal{R}$ be a k-selector over $\{1, 2, \dots, N\}$ and let Z be any subset of $\{1, 2, \dots, N\}$ such that $k \leq |Z| \leq 2k$. Let Y be the set of all elements $y \in Z$ such that there exists a set in $\mathcal{R}$ which hits Z on y. Then Y contains more than half of the elements of Z.*

PROOF. The proof is by contradiction. Assume $|Y| \leq \frac{1}{2}|Z| \leq k$, and let $X = Z \setminus Y$, that is, no set in $\mathcal{R}$ hits Z on an element of X. Hence, $|X| \geq \frac{1}{2}|Z| \geq \frac{1}{2}k$. Pick any subset $W \subseteq X$ such that $|W| = \lceil \frac{1}{2}|Z| \rceil$. We have $\frac{1}{2}k \leq |W| \leq k$, and note also that $|Z \setminus W| \leq |W| \leq k$. By the definition of the k-selector, there exists a set in $\mathcal{R}$ that hits W, say on an element w, but avoids $Z \setminus W$. Thus Z is hit by a set in $\mathcal{R}$ on $w \notin Y$ ($w \in W \subseteq X = Z \setminus Y$), and this contradicts the definition of set Y. □

We now show how to use selectors to obtain linearly selective families of small size. For each $j = 0, \dots, \lceil \log k \rceil - 1$, let $\mathcal{S}_j$ be a 2^j-selector containing $O(2^j \log N)$ sets. We call the family of sets $\mathcal{S} = \bigcup_{j=0}^{\lceil \log k \rceil - 1} \mathcal{S}_j$ a *compound k-selector*. Note that the size of a compound k-selector is comparable with the size of a regular k-selector: $|\mathcal{S}| = O(k \log N)$. The definitions of a selector, a compound selector, and a linearly selective family, and Lemma 1 imply the following lemma.

LEMMA 2. *For any positive integer k, a compound k-selector is a linearly k-selective family.*

PROOF. Consider a compound k-selector $\mathcal{S}$ and any nonempty set $Z \subseteq \{1, 2, \dots, N\}$ of size $|Z| \leq k$. There is an index $j \leq \lceil \log k \rceil - 1$ such that $2^j \leq |Z| \leq 2^{j+1}$. Lemma 1 implies that the 2^j-selector $\mathcal{S}_j$, which is a part of the compound k-selector $\mathcal{S}$, contains more than $|Z|/2$ sets each of them hitting Z on a different element. □

Selection of a leader. Many gossiping algorithms use the following *leader selection* task: select one node from the nodes in a given subset $W \subseteq V$ (say, select the node from $W \subseteq V$ with the smallest label). This task can be accomplished by $O(\log N)$ applications of a broadcasting algorithm, using a process which can be viewed as a binary search over the set $\{1, 2, \ldots, N\}$ for the smallest label of a node in W; for details see [8]. Thus the selection of a leader can be done in $O(n \log^2 N \log n)$ time using the $O(n \log N \log n)$-time broadcasting algorithm which follows from [11]. In undirected networks, if only the nodes in a subset $V' \subseteq V$ participate in the computation, then the same procedure and within the same time bound selects one node in W in each connected component of the subgraph induced by V' (if a component has any node in W). In undirected networks, if each node knows its neighbors, then the selection of a leader can be done in $O(n \log N)$ time using the $O(n)$-time broadcasting algorithm given in [6].

Neighbor gossip. A core difficulty in radio gossiping with large labels lies in learning as much as possible about neighborhoods. The task of learning about the complete neighborhood (finding out the labels of all neighbors) is known as *neighbor gossip*. In undirected networks, if each node knows its neighborhood, then gossiping can be completed in $O(n \log N)$ time: select a leader in $O(n \log N)$ time and then apply twice the communication pattern of the $O(n)$-step broadcasting algorithm given in [6]. The first application is for gathering in the leader the messages from all other nodes, and the second application is for broadcasting the gathered messages from the leader to all other nodes.

Observe that one application of the ROUND-ROBIN procedure completes the neighbor gossip in N steps, but this is unacceptably slow in the case of large labels. Our gossiping algorithm for undirected networks described in Sections 4.2–4.4 can be viewed as an extension and generalization of the following $O(n \log N)$-time neighbor gossip in star networks. Let $G = (V, E)$ be a network such that $V = \{v_0, v_1, \ldots, v_{n-1}\}$, $E = \{(v_0, v_i), (v_i, v_0)\colon\ i = 1, 2, \ldots, n-1\}$, and $n \geq 3$. The main task is to inform the center node v_0 about the labels of all other nodes, since v_0 can then inform the others in a single step.

If all nodes run an n-selector of size $O(n \log N)$, then the central node receives in $O(n \log N)$ time at least half of the labels of the other nodes; see Lemma 1. The process of collecting labels in node v_0 can be iterated in the following way. Each iteration consists of an application of a selector, followed by a message from node v_0. Node v_0 tells the nodes which have already transmitted successfully not to transmit again, and tells the other nodes how many of them there are. In iteration i the x_i nodes which are still active transmit according to an x_i-selector of size $O(x_i \log N)$. Since $x_i \leq \frac{1}{2} x_{i-1}$, the total time of the whole process is $O(n \log N)$.

In general graphs we can use compound selectors to estimate the degree of each node in the network. Lemma 2 and the definition of a linearly selective family of sets imply that after application of a compound n-selector on a network of size n, if a node v has received x distinct labels of its neighbors, then $x \leq d(v) \leq 2x$.

3. Deterministic Gossiping in Directed Networks. The initial message m_v originating from node v is said to be owned by this node. Our algorithm uses a procedure for the *explicit distribution* of messages: a selected node distributes to all other nodes, by means of a broadcasting algorithm, all messages which it has received so far other than the messages which have been distributed by the previous applications of this procedure. The messages for distribution are combined into one broadcast message. The running time of this procedure is the $O(n \log N \log n)$ running time of the broadcasting procedure from [11]. We say that the messages which have already been distributed by an application of this procedure are *secure*, while the remaining messages are still *insecure*. The nodes which own insecure messages are called *active*, while the other nodes are called *dormant*. The dormant nodes do not take part in some of the communication, but they continue taking part, for example, in executions of the explicit-distribution procedure.

Our gossiping algorithm consists of four stages. Stage 1 reduces the number of active neighbors of each node to less than k. The optimal value of the parameter k will be determined later. Stage 2 reduces the lengths of active paths (paths consisting only of active nodes) to less than k. Stage 3 makes sure that all remaining insecure messages are sent to dormant nodes. Finally, in Stage 4, all nodes repeat exactly the same sequence of transmissions as in Stages 1 and 2 to distribute the remaining insecure messages to all nodes. We now describe each stage in detail.

Stage 1. This stage is the execution of the iterative procedure REDUCE(k). In one iteration of this procedure, first each node which has at least k active neighbors collects at least $k/2$ insecure messages. Then a selected node which has at least $k/2$ insecure messages distributes them to all other nodes in the network.

REDUCE(k):
loop
1. The active nodes run a compound n-selector (the dormant nodes remain in the listening mode throughout this computation). If a node (active or dormant) has at least k active neighbors, then it collects at least $k/2$ insecure messages.
2. Choose a leader node λ among the nodes which have at least $k/2$ insecure messages. If such a node does not exist, then **exit** the loop.
3. Distribute all messages collected by node λ to all nodes in the network (the explicit-distribution procedure from node λ).

end loop.

LEMMA 3. *At the end of Stage* 1, *each node in the network* (*active or dormant*) *has fewer than k active neighbors*.

PROOF. A compound n-selector is a linearly n-selective family (Lemma 2). Thus if there is a node with at least k active neighbors at the beginning of an iteration of procedure REDUCE(k), then during Step 1 of this iteration, this node collects at least $k/2$ insecure messages, and the computation continues. □

LEMMA 4. *Procedure* REDUCE(k) *runs in time* $O((n^2 \log^2 N \log n)/k)$.

PROOF. Step 1 corresponds to a single application of a compound n-selector of size $O(n \log N)$. The selection of a leader in Step 2 takes $O(n \log^2 N \log n)$ time. Step 3 corresponds to a single application of the broadcasting procedure running in time

$O(n \log N \log n)$. Thus each iteration of the loop runs in $O(n \log^2 N \log n)$ time. Each chosen leader secures at least $k/2$ messages, so there can be at most $2n/k$ iterations of the loop. Hence altogether the complexity of procedure REDUCE(k) is bounded by $O((n^2 \log^2 N \log n)/k)$. □

Stage 2. In this stage procedure SHORTEN(k) is executed, which breaks long active paths into paths of length less than k by making some active nodes dormant. The purpose of breaking down long active paths is to reduce the maximum distance from an active node to a dormant node.

SHORTEN(k):
1. The active nodes run a strongly k-selective family k times.

loop
   2. Choose a leader λ among nodes who still hold at least k insecure messages. If no such node exists, then **exit** the loop.
   3. Distribute all messages collected by node λ to all nodes of the network (the explicit-distribution procedure from node λ).

end loop.

LEMMA 5. *At the end of Stage* 2, *there is no active simple path of length* k.

PROOF. Let P be a path of length k consisting of nodes which are active at the beginning of Stage 2, and let v be the last node on P. We show that at the end of this stage, at least one node on P is dormant. At the beginning of Stage 2, each node has fewer than k active neighbors (Lemma 3). Thus during each application of the strongly k-selective family in Step 1 of procedure SHORTEN(k), each node receives successfully transmissions from all its active neighbors, so receives all insecure messages available in its neighbors. This means that all insecure messages available at the beginning of Stage 2 in the active nodes within distance k from v become available at the end of the first application of the strongly selective family in the nodes within distance $k-1$ from v. Then after the second application of the strongly selective family, all these messages become available in the nodes within distance $k-2$ from v, and so on. Therefore, after k applications of the strongly selective family, node v has all insecure messages which were initially available within distance k from v, including the $k+1$ messages owned by the nodes of path P.

If node v is chosen as the leader in Step 2 in one of the iterations of the loop, then v broadcasts all insecure messages it has, including its own message, and becomes dormant. If node v is not chosen as the leader in any iteration, then it must hold fewer than k insecure messages at the end of Stage 2. This means that at least one of the messages owned by the nodes on P is secure at the end of Stage 2, so at least one node on P must be dormant at the end of Stage 2. □

LEMMA 6. *Procedure* SHORTEN(k) *runs in time* $O(k^3 \log N + (n^2 \log^2 N \log n)/k)$.

PROOF. Since the size of the strongly k-selective family is bounded by $O(k^2 \log N)$, Step 1 takes $O(k^3 \log N)$ time. Using a similar argument as in the proof of Lemma 4, we can show that the loop takes total $O(n/k) \cdot O(n \log^2 N \log n)$ time. □

Stage 3. During Stage 3 the active nodes run a strongly k-selective family k times. At the beginning of this stage there is no simple path of length k consisting only of active nodes (Lemma 5). This means that at the beginning of this stage one of the following two cases must hold. Either there is not any dormant node in the network and the distance from any node to any other node is at most k, or there is at least one dormant node and the distance from any active node to the closest dormant node is at most k. Thus the communication accomplished during Stage 3 either completes gossiping (if there is no dormant node in the network) or sends all remaining insecure messages to dormant nodes.

LEMMA 7. *If there is at least one dormant node at the beginning of Stage* 3, *then each insecure message reaches a dormant node during this stage. Otherwise, the complete gossiping is performed during this stage. Stage* 3 *runs in* $O(k^3 \log N)$ *time.*

Stage 4. During this stage every node repeats exactly the sequence of its transmissions from Stages 1 and 2. That is, a node v is in the transmitting mode in the tth step of Stage 4 if and only if v was in the transmitting mode in the tth step of the sequence of steps of Stages 1 and 2. Let $W \subseteq V$ denote the set of the dormant nodes at the end of Stage 2. If W is empty, then the full gossiping has been completed by the end of Stage 3 (Lemma 7), so we assume now that W is not empty. At the beginning of Stage 4, each initial message m_v is in one of the nodes in W (Lemma 7; note that no new dormant nodes are created during Stage 3). The transmissions performed during Stages 1 and 2 sent the messages owned by the nodes in W to all other nodes in the network. Thus if we now repeat all those transmissions, then the messages known to the nodes in W at the beginning of Stage 4 (that is, all initial messages m_v, $v \in V$) will be sent to all nodes in the network. Stage 4 runs in time which is exactly the sum of the times taken by Stages 1 and 2.

THEOREM 1. *Deterministic gossiping in directed ad hoc radio networks with large labels can be completed in* $O(n^{3/2} \log^{7/4} N \log^{3/4} n)$ *time.*

PROOF. The total time taken by our algorithm is $O((n^2 \log^2 N \log n)/k + k^3 \log N)$, and the bound given in this theorem is obtained by taking $k = n^{1/2}(\log N \log n)^{1/4}$. □

4. Deterministic Gossiping in Undirected Networks. In undirected networks, if each node knows its neighborhood, then gossiping can be completed in $O(n \log N)$ time using the $O(n)$-time broadcasting algorithm given in [6] (as discussed in Section 2). In Section 4.1 we show how the idea of gossiping in networks with known neighborhoods can be incorporated into the framework of the algorithm described in Section 3. The obtained gossiping algorithm for undirected networks runs in $O(n^{4/3} \log^2 N \log n)$ steps. In Sections 4.2–4.4 we present our main gossiping algorithm for undirected networks. In this algorithm the nodes learn first about their neighborhoods in $O(n \log^2 N \log^2 n)$ steps, and then complete gossiping in additional $O(n \log N)$ steps.

4.1. *Specialization of the General Algorithm to Undirected Networks.* We specialize the algorithm from Section 3 to undirected networks by modifying Stages 2 and 3, where insecure messages are being sent to dormant nodes. We replace these two stages with one stage, Stage 2-3. Instead of using expensive k-strongly families $\Theta(k)$ times in Stages 2 and 3 of the original algorithm, we use them only twice in the new Stage 2-3.

Only the active nodes participate in Stage 2-3. The nodes which are dormant at the end of Stage 1 stay in the listening mode throughout Stage 2-3. At the end of Stage 1 (that is, at the completion of procedure REDUCE(k)) the network induced by the active nodes consists of connected components (possibly only one component) and the degrees of active nodes are less than k. During Stage 2-3 the main part is gossiping performed independently and simultaneously within each connected component (there are no collisions between different components). Then all insecure messages are sent to the dormant nodes. The details of Stage 2-3 are given below.

Stage 2-3.

1. Apply once a strongly k-selective family to inform each active node about its complete (active) neighborhood.
2. Perform simultaneously, in $O(n \log N)$ time, gossiping within each connected component C of the subgraph induced by the active nodes.
3. Apply again a strongly k-selective family. If there is no dormant node at the end of Stage 1, then there is only one connected component C, spanning the whole network, and the gossiping throughout the whole network is completed by the end of Step 2. If there is at least one dormant node at the end of Stage 1, then for each component C, there is a node w in C linked to a dormant node v, and each dormant node has fewer than k active neighbors. The application of a strongly k-selective family ensures that w transmits successfully to v, sending all messages originating in C. Thus at the end of this Step 3, all insecure messages are in the dormant nodes.

Observe that since Stage 2-3 does not create any new dormant nodes, then it suffices that Stage 4 repeats only the transmissions from Stage 1 to complete the gossiping.

THEOREM 2. *Deterministic gossiping in undirected ad hoc radio networks with large labels can be completed in time* $O(n^{4/3} \log^{5/3} N \log^{2/3} n)$.

PROOF. The correctness of our algorithm follows from its description and the discussion of the general algorithm given in Section 3. The algorithm runs in $O((n^2 \log^2 N \log n)/k + k^2 \log N)$ time: $O((n^2 \log^2 N \log n)/k)$ time for Stages 1 and 4, and $O(n \log N + k^2 \log N)$ time for Stage 2-3. The bound in the statement of the theorem is obtained by taking $k = n^{2/3}(\log N \log n)^{1/3}$. □

4.2. $O(n \log^2 N \log^2 n)$*-Time Deterministic Gossiping in Undirected Networks—An Overview of the Algorithm.* We use the following additional notation. For a subset $W \subseteq V$, let $\mathcal{N}(W)$ denote the neighborhood of W: $w \in \mathcal{N}(W)$ if and only if $w \in W$ or w has a neighbor in W. Let $G(W)$ denote the subgraph of G induced by W.

The initial part of the algorithm: *estimating node degrees.* In the initial part of the algorithm, all nodes run a compound n-selector in $O(n \log N)$ steps. Each node v counts the number $\bar{d}(v)$ of its distinct neighbors who have successfully transmitted to v. We view this number $\bar{d}(v)$ as the *approximate degree* of node v. We have $d(v)/2 \le \bar{d}(v) \le d(v)$

(see the last paragraph in Section 2). For $i = 0, 1, \ldots, q = \lceil \frac{1}{2} \log n \rceil$, let

$$V_i = \{v \in V\colon\ \bar{d}(v) \geq 2^i \lceil \sqrt{n}\, \rceil\}.$$

Thus for each node v: v must belong to V_i if $d(v) \geq 2^{i+1}\lceil\sqrt{n}\rceil$ (if $d(v) \geq 2^{i+1}\lceil\sqrt{n}\,\rceil$, then $\bar{d}(v) \geq 2^i\lceil\sqrt{n}\,\rceil$); v may belong to V_i, if $d(v) \in \left[2^i\lceil\sqrt{n}\,\rceil, 2^{i+1}\lceil\sqrt{n}\,\rceil - 1\right]$; and v cannot belong to V_i, if $d(v) < 2^i\lceil\sqrt{n}\,\rceil$. Note that $\emptyset = V_q \subseteq V_{q-1} \subseteq \cdots \subseteq V_0 \subseteq V$. Let $\overline{V}_i = \mathcal{N}(V_i)$, the neighborhood of V_i. For each $i = 0, 1, \ldots, q-1$ in turn, the nodes in V_i run a compound n-selector. If $v \in \overline{V}_i \backslash V_i$, then v has at least one neighbor in V_i, so it receives a message from at least one node in V_i when the nodes in V_i execute the compound selector. Thus v learns that it belongs to $\overline{V}_i$. Therefore now each node v knows for each $i = 0, 1, \ldots, q-1$ whether it belongs to $\overline{V}_i$. This initial part of the algorithm takes $O(n \log N \log n)$ time.

Stages of the algorithm: *learning the topology of* $G(\overline{V}_{q-1}), G(\overline{V}_{q-2}), \ldots, G(\overline{V}_0)$. The main part of the algorithm consists of q stages numbered from $q-1$ down to 0. During stage i, each node v in $\overline{V}_i$ learns about the complete topology of its connected component in $G(\overline{V}_i)$, utilizing the knowledge acquired in the previous stages $q-1, q-2, \ldots, i+1$. We consider now stage i, $q-1 \geq i \geq 0$, and assume inductively that at the beginning of this stage, each node in $\overline{V}_{i+1} \subseteq \overline{V}_i$ knows the whole topology of its connected component in $G(\overline{V}_{i+1})$. Since $V_q = \emptyset$, then the basis of this induction holds trivially. Only nodes in $\overline{V}_i$ participate in the computation during this stage. The computations in individual connected components of $G(\overline{V}_i)$ are performed simultaneously and independently.

Let H be the subgraph of G induced by a connected component W of the subgraph $G(\overline{V}_i)$, let $\widetilde{W} = W \cap \overline{V}_{i+1}$, and let $d = 2^i\lceil\sqrt{n}\,\rceil$. Each node in H has degree at least $d \geq \sqrt{n}$. The subset $\widetilde{W}$ of the set of nodes W contains each node from W which has degree greater than $4d$ or has a neighbor with degree greater than $4d$, and may possibly contain some other nodes as well. At the beginning of the stage, each node v in H knows d, knows whether it belongs to set $\widetilde{W}$, and if $v \in \widetilde{W}$, then v knows also the complete topology of its connected component C in the subgraph $H(\widetilde{W})$ induced by $\widetilde{W}$. (C is also a connected component of $G(\overline{V}_{i+1})$, so v has learned about the complete topology of C in previous stages.) With this initial scenario, all nodes in H learn the complete topology of H in $O(n \log^2 N \log n)$ steps using the deterministic algorithm BOUNDED_NODE_DEGREES(H) described in Section 4.3.

The final part of the algorithm. Each node with degree at least $\sqrt{n}$ and all its neighbors belong to $\overline{V}_0$. Thus when the k stages are completed, each node with degree at least $\sqrt{n}$ knows its whole neighborhood. The other nodes learn about their neighborhood in $O(n \log N)$ steps using a strongly $\lceil\sqrt{n}\rceil$-selective family. When every node in the network knows its neighborhood, the gossiping is completed in additional $O(n \log N)$ steps.

THEOREM 3. *Deterministic gossiping in an n-node undirected radio network can be completed in* $O(n \log^2 N \log^2 n)$ *time*.

PROOF. The running time of the algorithm described above is as follows. The initial part of the algorithm takes $O(n \log N \log n)$ time. The q stages take together $q \cdot$

$O(n \log^2 N \log n)$ time, and $q = O(\log n)$. The final part takes $O(n \log N)$ time for the strongly $\lceil\sqrt{n}\rceil$-selective family, and $O(n \log N)$ time for the final gossiping when the nodes know their neighborhoods. □

4.3. *Learning the Topology of a Network with Similar Node Degrees*. Let H be an undirected connected radio network with at most n nodes, and such that each node has degree at least d or has a neighbor with degree at least d. We assume that $d \geq \sqrt{n}$. Let W denote the set of nodes in H, and let $\widetilde{W}$ be a subset of W containing each node from W which has degree greater than $4d$ or has a neighbor with degree greater than $4d$. Set $\widetilde{W}$ may possibly contain some other nodes from W as well. We assume that each node v in H knows d, knows whether it belongs to set $\widetilde{W}$, and if $v \in \widetilde{W}$, it knows also the complete topology of its connected component in the subgraph $H(\widetilde{W})$ induced by $\widetilde{W}$. All nodes in H can learn the complete topology of H in $O(n \log^2 N \log n)$ steps using the deterministic algorithm BOUNDED_NODE_DEGREES(H) described below. We first explain the algorithm assuming the special case when $\widetilde{W} = \emptyset$, and then we show that the algorithm also works for the general case.

The special case: $\widetilde{W} = \emptyset$. In this case there is no node in network H with degree greater than $4d$, so each node has degree in the range $[d, 4d]$, or has degree less than d but has a neighbor with degree in the range $[d, 4d]$. Each node knows the values of d and n, but does not know anything else about the topology of H. Algorithm BOUNDED_NODE_DEGREES(H) begins with the selection of a leader λ in H in $O(n \log^2 N \log n)$ time. Let L_i denote the set of nodes in H at distance i from node λ ($L_0 = \{\lambda\}$). Set L_i is called the ith *layer* of H. Let H_s denote the subgraph of H induced by $\bigcup_{i=0}^{s} L_i$.

After the selection of the leader λ, the algorithm is a sequence of phases numbered from 0. We consider an arbitrary phase s, $s \geq 0$, and assume inductively that at the beginning of this phase each node in H_s knows the complete topology of H_s. By the end of this phase each node in H_{s+1} will know the complete topology of H_{s+1}. The phase consists of three parts. During the first part, node λ learns about the edges between layers L_s and L_{s+1}, and passes this information to all nodes in L_{s+1}. During the second part, node λ learns about the edges within layer L_{s+1}. During the final, third part of the phase, the information about the whole topology of H_{s+1} is distributed from node λ to all nodes in H_{s+1}.

We now describe the first part of phase s. It begins with the nodes in L_s (the "border" nodes of H_s) transmitting sequentially one by one. Thus the information about all edges between L_s and L_{s+1} becomes available in L_{s+1}. We want to send this whole information from layer L_{s+1} to node λ by passing it first to layer L_s and then collating it at λ. This is done in $\lfloor \log d \rfloor + 3$ iterations. Let X denote the set of those nodes in L_{s+1} which have not yet transmitted successfully to a node in L_s. At the beginning of the first iteration, $X = L_{s+1}$. During the iteration i, $i = 1, 2, \ldots$, the nodes which are still in X run a compound $\lfloor 4d/2^{i-1} + 1 \rfloor$-selector. The information received at L_s is collated at node λ using procedure COLLATE_FROM_BORDER(H_s, λ, s), is then distributed to all nodes in L_s using procedure DISTRIBUTE_TO_BORDER(H_s, λ, s), and finally is sent to all nodes in L_{s+1} by sequential transmissions from the nodes in L_s. Procedures COLLATE_FROM_BORDER(H_s, λ, s) and DISTRIBUTE_TO_BORDER(H_s, λ, s) run in $O(s+$

$|L_s|$) steps and are described in Section 4.4 below. Each node in X removes itself from X if it finds out that it has managed to transmit successfully to a node in L_s.

At the beginning of the first iteration, each node $v \in L_s$ has at most $4d$ neighbors in X. This number decreases at least by half in each iteration. Indeed, if we assume inductively that at the beginning of iteration i node v has at most $\lfloor 4d/2^{i-1} \rfloor$ neighbors in X, then the compound selector used in this iteration ensures that at least half of the neighbors of v in X send a message to v. This means that v has at most $\lfloor 4d/2^i \rfloor$ neighbors in X at the end of this iteration. Thus X becomes empty by the end of the $\lfloor \log d \rfloor + 3$ iterations. Iteration i takes $O((d/2^i)\log N + s + |L_s|)$ steps: $O((d/2^i)\log N)$ steps for the compound selector and $O(s + |L_s|)$ steps for procedures COLLATE_FROM_BORDER and DISTRIBUTE_TO_BORDER. Hence, by summing this bound over all iterations, the first part of phase s takes $O(d \log N + s \log n + |L_s| \log n)$ steps plus $|L_s|$ steps for the initial sequential transmissions from the nodes in L_s.

In the second part of phase s, node λ learns about the edges between nodes within layer L_{s+1}. First the nodes in L_{s+1} transmit sequentially. Observe that the nodes in L_{s+1} learned about each other during the first part of the phase (they learned about all edges between L_s and L_{s+1}), so they can now transmit one by one, for example in the order of the increasing node label. The information about the edges between nodes within layer L_{s+1} becomes available in L_{s+1}. Then this information is sent to λ by repeating the iterative process used in the first part of the phase. Thus the second part of the phase takes $|L_{s+1}| + O(d \log N + s \log n + |L_s| \log n)$ steps. During the final, third part of the phase, the leader node λ distributes to all nodes in H_{s+1} the information about the complete topology of H_{s+1} using the $O(s \log^2 n)$-time broadcasting schedule for known radio networks given in [5].

At the end of phase s, if H_{s+1} turns out to be the same as H_s, then $H_s = H$ and the algorithm terminates. Otherwise it proceeds to the next phase. The time taken by phase s is

$$O(d \log N + s \log^2 n + |L_s| \log n + |L_{s+1}|). \qquad (1)$$

There are at most $D + 1$ phases in total, where D is the diameter of H, so we sum the bound (1) over $s = 0, 1, \ldots, D$ to obtain the following bound on the time taken by all phases:

$$O(dD \log N + D^2 \log^2 n + n \log n). \qquad (2)$$

Lemma 8 below gives an $O(n/d)$ bound on D, and we have assumed that $d \geq \sqrt{n}$, so the bound (2) is $O(n(\log N + \log^2 n))$. Thus the $O(n \log^2 N \log n)$ bound on the time taken for the initial selection of the leader λ is also a bound on the total running time of algorithm BOUNDED_NODE_DEGREES.

LEMMA 8. *The diameter of network H is $O(n/d)$.*

PROOF. Consider a longest shortest path P in H. The diameter of H is equal to the length of this path. The sum of the degrees of the nodes on P is at most $3n$, because each node in H is a neighbor of at most three nodes on P, or otherwise there would be a shortcut for P. Thus P has at most $3n/d$ nodes with degrees at least d. Let $x_1, x_2, \ldots, x_l$ be the nodes on path P with degrees less than d, in the order as they appear along P. Each node x_i must have a neighbor z_i of degree at least d, by the definition of network H. Let Z_i be the set of neighbors of node z_i. If $l > 5n/d$, then $\sum_{i=1}^{l} |Z_i| \geq dl > 5n$, so there are six indices $1 \leq i_1 < i_2 < \cdots < i_6 \leq l$ such that the neighborhoods $Z_{i_1}, Z_{i_2}, \ldots, Z_{i_6}$ have a common node v. The path $(x_{i_1}, z_{i_1}, v, z_{i_6}, x_{i_6})$ is a shortcut for P, contradicting the assumption that P is a shortest path. Therefore we must have $l \leq 5n/d$, so the length of path P is at most $3n/d + 5n/d$. □

4.3.1. *The general case*: $\widetilde{W}$ *may be nonempty.* We show now that algorithm BOUNDED_NODE_DEGREES described above works also in the general case when set $\widetilde{W}$ may be nonempty. To see this, we re-examine phase s of the algorithm, for any $s \geq 0$, assuming inductively that at the beginning of this phase each node in H_s knows the complete topology of H_s. We show that as in the special case, during the first part of the phase, node λ learns about all edges between layers L_s and L_{s+1}, and passes this information to all nodes in L_{s+1}; and during the second part of the phase, node λ learns about all edges within layer L_{s+1}.

Consider first an arbitrary edge between a node $v \in L_s$ and a node $w \in L_{s+1}$. If the degree of node v in H is greater than $4d$, then both v and w belong to $\widetilde{W}$. Thus v knows about edge (v, w) already at the beginning of the phase, and passes this information to the leader λ during the first application of procedure COLLATE_FROM_BORDER(H_s, λ, s). If the degree of node v is not greater than $4d$, then during the first part of the phase node v receives a message from node w, learning about edge (v, w), and passes on this information to the leader λ in the same way as in the case when $\widetilde{W} = \emptyset$. Therefore in both cases, by the end of the first part of the phase, the leader λ and all nodes in L_{s+1} know about edge (v, w).

Consider now an arbitrary edge (w', w'') between two nodes in L_{s+1}. Let v' be a neighbor of w' in L_s, and let v'' be a neighbor of w'' in L_s (nodes v' and v'' are not necessarily distinct). If both v' and v'' have degrees in H greater than $4d$, then the nodes w', w'', v' and v'' belong to the same connected component of $H(\widetilde{W})$. Thus both v' and v'' know about edge (w', w'') already at the beginning of the phase, and pass this information to the leader λ. If the degree of v' or v'' is at most $4d$, then the information about edge (w', w'') is passed from w' to v' or from w'' to v'', and then forwarded to the leader λ, in the second part of the phase.

4.4. *Collating and Distributing Information Available at the Border of a Known Network.* Let H_s be an undirected connected radio network with a distinguished leader node λ and layers $L_0 = \{\lambda\}, L_1, \ldots, L_s$. We present in this section two deterministic procedures for communicating within H_s between the leader node λ and the border L_s. We assume that each node in H_s knows the leader and all inter-layer edges in H_s. Procedure COLLATE_FROM_BORDER(H_s, λ, s) given in Figure 1 sends to the leader all messages initially available at the nodes of the border L_s, completing this task in $O(s + |L_s|)$ steps.

At the beginning of iteration i of this procedure, $i = s, s-1, \ldots, 1$, the messages originating from L_s are available in the nodes of a set $L'_i \subseteq L_i$ ($L'_s = L_s$). During this iteration, these messages are forwarded to the nodes in a set $L'_{i-1} \subseteq L_{i-1}$ in the following way. Let R_i denote the set of nodes in L'_i which have not transmitted yet. At the beginning of the iteration, $R_i = L'_i$ and $L'_{i-1} = \emptyset$. While there are two nodes in R_i

COLLATE_FROM_BORDER(H_s, λ, s):

{ L_i is the set on nodes in H_s at distance i from node λ. }

1. $L'_s \leftarrow L_s$;
2. **for** $i = s$ **down to** 1 **do**
3. { All messages originating from L_s are now available in a set $L'_i \subseteq L_i$. }
4. { Pass all these messages from L'_i to some nonempty set $L'_{i-1} \subseteq L_{i-1}$. }
5. $L'_{i-1} \leftarrow \emptyset$;
6. $R_i \leftarrow L'_i$; { the nodes in L'_i which have not transmitted yet }
7. **while** there are two nodes in R_i with a common neighbor $w \in L_{i-1}$ **do**
8. all neighbors of w in R_i transmit sequentially;
9. delete from R_i the neighbors of w;
10. add w to L'_{i-1};
11. **end_while**
12. **if** $R_i \neq \emptyset$ **then**
13. let $v_1, v_2, \ldots, v_r$ be the nodes in R_i;
14. let w_j be an arbitrary neighbor of v_j in L_{i-1}, for $j = 1, 2, \ldots, r$;
15. all nodes $v_1, v_2, \ldots, v_r$ transmit in parallel;
16. $R_i \leftarrow \emptyset$;
17. add nodes $w_1, w_2, \ldots, w_r$ to L'_{i-1};
18. **end_if**
19. **end_for**

Fig. 1. Procedure COLLATE_FROM_BORDER(H_s, λ, s).

with a common neighbor $w \in L_{i-1}$, all neighbors of w in R_i transmit sequentially to deliver to node w the messages which they have, and node w is added to L'_{i-1}. When eventually the condition is reached that no two nodes in R_i have a common neighbor in L_i, then all remaining nodes $v_1, v_2, \ldots, v_r$ in R_i transmit in parallel in one step. We add to L'_{i-1} nodes $w_1, w_2, \ldots, w_r$, where w_j is an arbitrary neighbor of v_j in L_{i-1}.

LEMMA 9. *Procedure* COLLATE_FROM_BORDER(H_s, λ, s) *sends to node* λ *in* $O(s + |L_s|)$ *steps all messages initially available at the nodes in the border layer* L_s.

PROOF. To see the correctness of procedure COLLATE_FROM_BORDER(H_s, λ, s), observe that the communication during iteration i is arranged in such a way that for each node $v \in L'_i$, there is at least one node $w \in L'_{i-1}$ which successfully receives transmission from v. Thus at the end of the last iteration all messages originating from L_s are available at node λ (observe that $L'_0 = \{\lambda\}$).

During iteration i, for each node added to set L'_{i-1} in line 10 in Figure 1, at least two nodes are deleted from set R_i, so

$$|L'_{i-1}| \leq (|L'_i| - r)/2 + r = (|L'_i| + r)/2, \tag{3}$$

where r is the number of nodes in L'_i participating in the parallel transmissions in line 15. There are $|L'_i| - r$ sequential transmissions during this iteration, and using (3) we get

$$|L'_i| - r = 2|L'_i| - (|L'_i| + r) \le 2(|L'_i| - |L'_{i-1}|). \tag{4}$$

Summing up (4) for $i = s, s-1, \ldots, 1$, gives the bound of $2|L'_s| = 2|L_s|$ on the total number of sequential transmissions throughout the whole procedure. Since there are at most s steps with parallel transmissions, procedure COLLATE_FROM_BORDER uses at most $2|L_s| + s$ steps. □

Procedure DISTRIBUTE_TO_BORDER(H_s, λ, s) sends a message from λ to all nodes in L_s by reversing the steps and the directions of transmissions in procedure COLLATE_FROM_BORDER(H_s, λ, s). That is, using the notation from the description of procedure COLLATE_FROM_BORDER, during iteration i of procedure DISTRIBUTE_TO_BORDER, for $i = 1, 2, \ldots, s$, first nodes $w_1, w_2, \ldots, w_r$ transmit in parallel (to pass the message to nodes $v_1, v_2, \ldots, v_r$). Then the nodes w selected in procedure COLLATE_FROM_BORDER in the loop in lines 7–11 transmit sequentially. Procedure DISTRIBUTE_TO_BORDER (H_s, λ, s) runs in $O(s + |L_s|)$ steps.

5. Conclusions. We studied deterministic gossiping in ad hoc radio networks with large node labels. Our main results are two new deterministic gossiping algorithms for such networks: one for directed networks with running time $O(n^{3/2} \log^2 N \log n)$ and one for undirected networks with running time $O(n \log^2 N \log^2 n)$, where n is the number of nodes in the network and N is the size of the domain of the node labels. No subquadratic upper bounds were known for networks with large labels prior to our work, while the best previous bound for networks with small labels (when $N = O(n)$) was $O(n^{3/2} \log^2 n)$.

An interesting open question is to reduce further the upper bounds. The upper bound for directed networks has been very recently improved to $O(n^{4/3} \log^{O(1)} N)$ [24] by further refinement of our algorithm. However, it seems that if one wanted to aim at achieving an $O(n \log^{O(1)} N)$ bound on deterministic gossiping in directed networks, then some considerably new insight into the problem would be required.

In our model we assume that there is no bound on the length of messages which can be sent in one communication step. It would be interesting to investigate if our algorithms could be used as a basis for efficient gossiping algorithms in the more practical model of bounded size messages.

References

[1] N. Alon, A. Bar-Noy, N. Linial, and D. Peleg. A lower bound for radio broadcast. *J. Comput. System Sci.*, 43(2):290–298, 1991.

[2] R. Bar-Yehuda, O. Goldreich, and A. Itai. On the time complexity of broadcast in multi-hop radio networks: an exponential gap between determinism and randomization. *J. Comput. System Sci.*, 45(1):104–126, 1992.

[3] R. Bar-Yehuda, A. Israeli, and A. Itai. Multiple communication in multi-hop radio networks. *SIAM J. Comput.*, 22(4):875–887, 1993.

[4] D. Brusci and M. Del Pinto. Lower bounds for the broadcast problem in mobile radio networks. *Distrib. Comput.*, 10(3):129–135, 1997.

[5] I. Chlamtac and O. Weinstein. The wave expansion approach to broadcasting in multihop radio networks. *IEEE Trans. Commun.*, 39(3):426–433, 1991.

[6] B. S. Chlebus, L. Gąsieniec, A. M. Gibbons, A. Pelc, and W. Rytter. Deterministic broadcasting in unknown radio networks. In *Proc. 11th ACM–SIAM Symp. on Discrete Algorithms* (SODA 2000), pp. 861–870. ACM/SIAM, New York, 2000.

[7] B. S. Chlebus, L. Gąsieniec, A. Lingas, and A. Pagourtzis. Oblivious gossiping in ad hoc radio networks. In *Proc. 5th Int. Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications* (DIAL-M 2001), pp. 44–51. ACM, New York, 2001.

[8] B. S. Chlebus, L. Gąsieniec, A. Östlin, and J. M. Robson. Deterministic radio broadcasting. In *Proc. 27th Int. Colloq. on Automata, Languages and Programming* (ICALP 2000), LNCS vol. 1853, pp. 717–728. Springer, Berlin, 2000.

[9] M. Christersson, L. Gąsieniec, and A. Lingas. Gossiping with bounded size messages in ad hoc radio networks. In *Proc. 29th Int. Colloq. on Automata, Languages and Programming* (ICALP 2002), LNCS vol. 2380, pp. 377–389. Springer, Berlin, 2002.

[10] M. Chrobak, L. Gąsieniec, and W. Rytter. A randomized algorithm for gossiping in radio networks. In *Proc. 7th Ann. Int. on Computing and Combinatorics Conference* (COCOON 2001), LNCS vol. 2108, pp. 483–492. Springer, Berlin, 2001.

[11] M. Chrobak, L. Gąsieniec, and W. Rytter. Fast broadcasting and gossiping in radio networks. *J. Algorithms*, 43(2):177–189, 2002.

[12] A. E. F. Clementi, A. Monti, and R. Silvestri. Selective families, superimposed codes, and broadcasting in unknown radio networks. In *Proc. 12th ACM–SIAM Symp. on Discrete Algorithms* (SODA 2001), pp. 709–718. ACM/SIAM, New York, 2001.

[13] A. E. F. Clementi, A. Monti, and R. Silvestri. Distributed multi-broadcast in unknown radio networks. In *Proc. 20th Ann. ACM Symp. on Principles of Distributed Computing* (PODC 2001), pp. 255–264. ACM, New York, 2001.

[14] A. E. F. Clementi, A. Monti, and R. Silvestri. Round robin is optimal for fault-tolerant broadcasting on wireless networks. In *Proc. 9th Ann. European Symp. on Algorithms* (ESA 2001), LNCS vol. 2161, pp. 452–463. Springer, Berlin, 2001.

[15] A. E. F. Clementi, A. Monti, and R. Silvestri. Distributed broadcast in radio networks of unknown topology. *Theoret. Comput. Sci.*, 1–3(302):337–364, 2003.

[16] A. Czumaj and W. Rytter. Broadcasting algorithms in radio networks with unknown topology. In *Proc. 44th Symp. on Foundations of Computer Science* (FOCS 2003), pp. 492–501. IEEE Computer Society, Los Alamitos, CA, 2003.

[17] G. De Marco and A. Pelc. Faster broadcasting in unknown radio networks. *Inform. Process. Lett.*, 79(2):53–56, 2001.

[18] M. Elkin and G. Kortsarz. Improved schedule for radio broadcast. In *Proc. 16th Ann. ACM–SIAM Symp. on Discrete Algorithms* (SODA 2005), pp. 222–231. ACM/SIAM, New York, 2005.

[19] I. Gaber and Y. Mansour. Centralized broadcast in multihop radio networks. *J. Algorithms*, 46(1):1–20, 2003.

[20] L. Gąsieniec and A. Lingas. On adaptive deterministic gossiping in ad hoc radio networks. *Inform. Process. Lett.*, 83(2):89–93, 2002.

[21] L. Gąsieniec, A. Pagourtzis, and I. Potapov. Deterministic communication in radio networks with large labels. In *Proc. 10th Ann. European Symp. on Algorithms* (ESA 2002), LNCS vol. 2461, pp. 512–524. Springer, Berlin, 2002.

[22] L. Gąsieniec, D. Peleg, and Q. Xin. Faster communication in known topology radio networks. In *Proc. 24th Ann. ACM Symp. on Principles of Distributed Computing* (PODC 2005), pp. 129–137. ACM, New York, 2005.

[23] L. Gąsieniec and I. Potapov. Gossiping with unit messages in known radio networks. In *Proc. 2nd IFIP Int. Conf. on Theoretical Computer Science* (TCS 2002), IFIP Conference Proceedings vol. 223, pp. 193–205. Kluwer, Dordrecht, 2002.

[24] L. Gąsieniec, T. Radzik, and Q. Xin. Faster deterministic gossiping in directed ad hoc radio networks. In *Proc. 9th Scandinavian Workshop on Algorithm Theory* (SWAT 2004), LNCS vol. 3111, pp. 397–407. Springer, Berlin, 2004.

[25] P. Indyk. Explicit constructions of selectors and related combinatorial structures with applications. In *Proc. 13th ACM–SIAM Symp. on Discrete Algorithms* (SODA 2002), pp. 697–704. ACM/SIAM, New York, 2002.

[26] D. R. Kowalski and A. Pelc. Deterministic broadcasting time in radio networks of unknown topology. In *Proc. 43rd Symp. on Foundations of Computer Science* (FOCS 2002), pp. 63–72. IEEE Computer Society, Los Alamitos, 2002.

[27] D. R. Kowalski and A. Pelc. Broadcasting in undirected ad hoc radio networks. In *Proc. 22nd Ann. ACM Symp. on Principles of Distributed Computing* (PODC 2003), pp. 73–82. ACM, New York, 2003.

[28] D. R. Kowalski and A. Pelc. Centralized deterministic broadcasting in undirected multi-hop radio networks. In *Proc. APPROX-RANDOM* 2004, LNCS vol. 3122, pp. 171–182. Springer, Berlin, 2004.

[29] D. R. Kowalski and A. Pelc. Faster deterministic broadcasting in ad hoc radio networks. *SIAM J. Discrete Math.*, 18(2):332–346, 2004.

[30] E. Kushilevitz and Y. Mansour. An $\Omega(D \log(N/D))$ lower bound for broadcast in radio networks. *SIAM J. Comput.*, 27(3):702–712, 1998.

[31] D. Liu and M. Prabhakaran. On randomized broadcasting and gossiping in radio networks. In *Proc. 8th Ann. Int. Conf. Computing and Combinatorics* (COCOON 2002), LNCS vol. 2387, pp. 340–349. Springer, Berlin, 2002.

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

[33] K. Ravishankar and S. Singh. Asymptotically optimal gossiping in radio networks. *Discrete Appl. Math.*, 61(1):61–82, 1995.

[34] K. Ravishankar and S. Singh, Gossiping on a ring with radios, *Parallel Process. Lett.*, 6(1):115–126, 1996.

[35] S. Tabbane. *Handbook of Mobile Radio Networks*. Artech House, Norwood, MA, 2000.

[36] Y. Xu. An $O(n^{1.5})$ deterministic gossiping algorithm for radio networks. *Algorithmica*, 36(1):93–96, 2003.