Faster Deterministic Gossiping in Directed Ad Hoc Radio Networks

Leszek Gąsieniec[1], Tomasz Radzik[2], and Qin Xin[1]

[1] Department of Computer Science, University of Liverpool, Liverpool L69 7ZF, UK
{leszek,qinxin}@csc.liv.ac.uk
[2] Department of Computer Science, King's College London, London WC2R 2LS, UK
radzik@dcs.kcl.ac.uk

Abstract. We study the *gossiping problem* in directed ad-hoc radio networks. Our main result is a deterministic algorithm that solves this problem in an n-node network in time $O(n^{4/3} \log^4 n)$. The algorithm allows the labels (identifiers) of the nodes to be polynomially large in n, and is based on a novel way of using *selective families*. The previous best general (*i.e.*, dependent only on n) deterministic upper bounds were $O(n^{5/3} \log^3 n)$ for networks with polynomially large node labels [1], and $O(n^{3/2} \log^2 n)$ for networks with linearly large node labels [2,3,4].

1 Introduction

The two classical problems of information dissemination in computer networks are *broadcasting* and *gossiping*. In the broadcasting problem, we want to distribute a message from a distinguished *source* node to all other nodes in the network. In the gossiping problem, each node v in the network initially holds a message m_v, and we want to distribute all messages m_v to all nodes in the network. For both problems, an important performance measure is the time needed to complete the required task.

We consider the following model of a radio network. A network is a directed, strongly-connected graph $G = (V, E)$, where V represents the set of nodes of the network, and E contains an (ordered) pair of distinct nodes $(v, w) \in V \times V$ iff node v can directly send a message to node w. If $(v, w) \in E$, then we say that w is a *neighbour* of v and v is an *in-neighbour* of w. The total number of the in-neighbours of a node w is its *in-degree*, and the maximum in-degree of a node is called the *max-indegree* of the network. The *size of the network* is the number of nodes $n = |V|$. Each node $v \in V$ is labelled by a distinct positive integer. The set of nodes directly reachable from a node $v \in V$ is the *range* of v. One of the radio network properties is that a message transmitted by a node is always sent to all nodes within its range.

The communication in the network is synchronous and consists of a sequence of (communication) steps. During one step, each node v either transmits or listens. If v transmits, then the transmitted message reaches each of its neighbours by the end of this step. However, a node w in the range of v successfully receives

T. Hagerup and J. Katajainen (Eds.): SWAT 2004, LNCS 3111, pp. 397–407, 2004.

this message iff in this step w is listening and v is the only transmitting node which has w in its range. If node w is in the range of a transmitting node but is not listening, or is in the range of more that one transmitting node, then a *collision* occurs and w does not retrieve any message in this step. Moreover, the "noise of collision" is indistinguishable from the "background noise" experienced by a node which is not in the range of any transmitting node (that is, the nodes do not have a collision detection mechanism). Dealing with collisions is one of the main challenges in efficient radio communication. A commonly used tool for coping with collisions is the concept of *selective families of sets* [5,6,2,1]. In Section 2 we recall this concept and introduce a novel, more powerful way of using it.

The (communication) time of an algorithm is the number of communication steps required to complete the algorithm. That is, we do not account for any internal computation within individual nodes. Another abstraction of the model is no limit on the length of a message which one node can transmit in one step. Actually, to simplify the presentation of algorithms, we assume that if a node transmits in the current step, it transmits its whole knowledge.

The algorithms we present in this paper are for *ad-hoc radio networks*: the topology of connections is unknown in advance. At the beginning of an algorithm each node knows only its label, its initial message, and the global bound N on the node labels. We assume that $N = O(n^p)$ for some constant p.

1.1 Previous Work

The broadcasting problem has attracted considerably more attention than the gossiping problem. For networks with linearly bounded labels ($N = O(n)$), the trivial $O(n^2)$ upper bound on broadcasting was first improved by Chlebus et al. [7] to $O(n^{11/6})$. The subsequent improvements included an $\tilde{O}(n^{5/3})$[1] time algorithm proposed by De Marco and Pelc [8], an $O(n^{3/2})$ time algorithm proposed by Chlebus et al. [5], and an $O(n \log^2 n)$ time algorithm developed by Chrobak, Gąsieniec and Rytter [2]. Clementi, Monti and Silvestri [6] presented a deterministic broadcasting algorithm for *ad-hoc* radio networks which works in time $\tilde{O}(D\Delta)$, where D is the diameter of the network (the number of edges on the longest shortest path) and Δ is the maximum in-degree of a node. The $O(n \log^2 n)$ and $\tilde{O}(D\Delta)$ algorithms, presented, respectively, in [2] and [6], can be easily adapted to work within the same asymptotic times for polynomially bounded node labels. Brusci and Del Pinto [9] showed that for any deterministic algorithm $\mathcal{A}$ for broadcasting in ad-hoc radio networks, there are networks on which $\mathcal{A}$ requires $\Omega(n \log n)$ time.

The first sub-quadratic deterministic algorithm for the gossiping problem in ad-hoc radio networks was the $\tilde{O}(n^{3/2})$ time algorithm proposed by Chrobak *et al.* [2]. Subsequently Xu [4] improved this bound by a polylogarithmic factor obtaining a $O(n^{3/2})$ bound. For small values of diameter D, the gossiping time

[1] Notation $\tilde{O}(f(n))$ denotes a function in $O(f(n) \log^c n)$ for a constant c. In all cases when we use this notation in this paper, constant c is at most 4.

was later improved by Gąsieniec and Lingas [3] to $\tilde{O}(nD^{1/2})$. The gossiping algorithms presented in [2,4,3] assume that the node labels are linear in n and we do not see how they could be extended to the case where node labels are polynomially large. Clementi, Monti and Silvestri [6] presented a $\tilde{O}(D\Delta^2)$-time deterministic gossiping algorithm, and subsequently Gąsieniec and Lingas [3] showed an $\tilde{O}(D\Delta^{3/2})$ algorithm. Both these algorithms work for polynomially large node labels. Prior to this paper, the best general (dependent only on n) bound on a deterministic algorithm for gossiping in ad-hoc networks with polynomially large node labels was $\tilde{O}(n^{5/3})$ due to Gąsieniec, Pagourtzis and Potapov [1].

A study on deterministic gossiping in unknown radio networks with messages of limited size can be found in [10]. The gossiping problem in *ad-hoc* radio networks attracted also studies on efficient randomised algorithms. Chrobak *et al.* [2] proposed an $O(n\log^4 n)$ time randomise gossiping algorithm. This time was later reduced to $O(n\log^3 n)$ in [11], and then to $O(n\log^2 n)$ in [12]. This shows a relatively large gap between the best known deterministic and randomised algorithms for gossiping.

We also mention some results for communication in the model where the network topology is known to all nodes in advance. Gaber and Mansour [13] showed that in such a model the broadcasting task can be completed in time $O(D+\log^5 n)$. Diks *et al.* [14] proposed efficient radio broadcasting algorithms for (various) particular types of network topologies. The gossiping problem was not studied in the context of known radio networks until very recent work of Gąsieniec and Potapov [15]. One can find there a study on the gossiping problem in known radio networks, where each node transmission is limited to unit messages. In this model they proposed several optimal and almost optimal $O(n)$-time gossiping algorithms for various standard network topologies, including lines, rings, stars and free trees. They also proved that there exists a radio network topology in which the gossiping (with unit messages) requires $\Omega(n\log n)$ time.

1.2 Our Results

In this paper we present a deterministic algorithm that solves the gossiping problem in directed *ad-hoc* radio networks with polynomially large node labels in time $O(n^{4/3}\log^{10/3} n)$. This is the fastest currently known deterministic radio gossiping algorithm in graphs with an arbitrary topology. The previous best algorithm for this task requires $\tilde{O}(n^{5/3})$ time [1]. Our algorithm improves also the previous best upper bound $\tilde{O}(n^{3/2})$ for gossiping in ad-hoc networks with node labels only linearly large [2]. The algorithm is based on an extension of the concept of strongly selective families [5,6] from star-like sub-graphs into sub-graphs with a more general topology. We also show a simple $O(n\Delta\log^2 n)$-time deterministic gossiping algorithm, which improves Gąsieniec and Lingas' [3] upper bound of $\tilde{O}(\min\{nD^{1/2}, D\Delta^{3/2}\})$ algorithm, if $\Delta = O(D^{1/2-\epsilon})$ and $D\Delta^{1/2} = \Omega(n^{1+\epsilon})$ for some constant $\epsilon > 0$.

The paper is organised as follows. In Section 2 we recall basic definitions on selectivity, selective families, and selectors. We also introduce a new notion of a *path selector* which forms a crucial part in our main gossiping algorithm. In Section 3 we present our two algorithms. In Section 4 we briefly summarise the paper and mention some directions for further research in deterministic radio gossiping.

2 Selectivity and Tools

In this section we recall the mechanics of selectivity. In particular, we refer to the definition of *selectors* which form a backbone of a new combinatorial structure of an *path selector*, a tool that is used in our new radio gossiping algorithm.

The *neighbourhood of a node* v is the set $NB(v)$ consisting of node v and all its in-neighbours. For any simple path $P = <v_0, v_1, \ldots, v_q>$, the length of P, denoted by *length*(P), is the number of edges on P. The *neighbourhood of* P is defined as the union $NB(P)$ of the neighbourhoods of nodes $v_1, \ldots v_q$. Observe that node v_0 belongs to $NB(P)$ since it belongs to the neighbourhood of node v_1, but an in-neighbour of node v_0 does not belong to $NB(P)$, unless it is also an in-neighbour of an other node on P. If $(u, v) \in E$ and u is the only node in $NB(v)$ transmitting in the current step, then v receives this transmission. For path P, if v_i is a node on P other than the first node v_0, $(u, v_i) \in E$, and u is the only node in $NB(P)$ transmitting in the current step, then v_i receives this transmission.

2.1 Selectors

We say that a set R hits a set Z on element z, if $R \cap Z = \{z\}$, and a family $\mathcal{F}$ of sets hits a set Z on element z, if $R \cap Z = \{z\}$ for at least one $R \in \mathcal{F}$.

De Bonis *et al.* [16] introduced a definition of a family of subsets of set $[N] \equiv \{0, 1, \ldots, N-1\}$ which hits each subset of $[N]$ of size at most k on at least m distinct elements, where N, k and m are parameters, $N \geq k \geq m \geq 1$. They proved existence of such a family of size $O((k^2/(k-m+1)) \log N) = O((k^2/(k-m+1)) \log n)$. For convenience of our presentation, we prefer the following slight modification of this definition, obtained by using the parameter $r = k - m$ instead of the parameter m. For integers N and k, and a real number r such that $N \geq k \geq r \geq 0$, a family $\mathcal{F}$ of subsets of $[N]$ is a (N, k, r)-*selector*, if for any subset $Z \subseteq [N]$ of size at most k, the number of all elements z of Z, such that, $\mathcal{F}$ *does not* hit Z on z is at most r. That is,

$$|\{z \in Z : \text{ for each } R \in \mathcal{F},\ R \cap Z \neq \{z\}\ \}| \ \leq\ r.$$

In terms of this definition, De Bonis *et al.* [16] showed existence of a (N, k, r)-selector of size $T(N, k, r) = O((k^2/(r+1)) \log N)$. In particular, there exists a $(N, k, 0)$-selector of size $O(k^2 \log N)$ – such a *"strong" selector* hits each set $Z \subseteq [N]$ of size at most k on each of its elements; and a (N, k, ck)-selector of size $O(k \log N) = O(k \log n)$, for any constant $0 < c < 1$ – such a *"weak" selector* guarantees only that it hits each set $Z \subseteq [N]$ of size at most k at least on a constant fraction of its elements.

2.2 Path Selectors

An *interleaved sequence* $\mathbf{S}(A_0, A_1, \ldots, A_{p-1})$ of finite sets A_i is an infinite sequence $(\alpha_0, \alpha_1, \ldots)$ obtained by arbitrarily ordering the elements of each set A_i into a sequence $(a_{i,0}, a_{i,1}, \ldots, a_{i,q_i-1})$, where $q_i = |A_i|$, and setting $\alpha_{i+jp} = a_{i,j \bmod q_i}$, for all integers $j \geq 0$ and $0 \leq i \leq p-1$. That is, subsequence $(\alpha_i, \alpha_{i+p}, \alpha_{i+2p}, \ldots)$ is a periodic sequence based on ordered elements of set A_i, for each $i = 0, 1, \ldots, p-1$.

A *path selector* is an interleaved sequence of properly chosen selectors. Let $\mathcal{F}_i$ be an $(N, k, k/2^{i+1})$-selector of size $T(N, k, k/2^{i+1})$, for $i = 0, 1, \ldots, \lceil \log k \rceil$. Also let

$$T_{N,k} = (\lceil \log k \rceil + 1) \sum_{i=0}^{\lceil \log k \rceil} \frac{k}{2^i} T(N, k, \frac{k}{2^{i+1}})$$

$$= (\lceil \log k \rceil + 1) \sum_{i=0}^{\lceil \log k \rceil} \frac{k}{2^i} O(2^i k \log N) = O(k^2 \log N \log^2 k).$$

Definition 1. *The prefix of an interleaved sequence* $\mathbf{S}(\mathcal{F}_0, \mathcal{F}_1, \ldots, \mathcal{F}_{\lceil \log k \rceil})$ *of length* $T_{N,k}$ *forms a path selector* $S_{N,k}$.

Note that according to its definition the length of the path selector $S_{N,k}$ is $O(k^2 \log N \log^2 k) = O(k^2 \log^3 n)$.

2.3 Extended Selectivity

In previous work the selectivity properties of selectors were used in the context of star-like sub-graphs, where a number (bounded by certain value k) of nodes is competing to communicate to one distinguished node c, the *centre* of the star. For example, a single application of a strong $(N, k, 0)$-selector allows each of the competing nodes to transmit successfully to the centre c in some step. To apply a family $\mathcal{F}$ of subsets of $[N]$ means first to arrange the sets of $\mathcal{F}$ in a sequence $F_1, F_2, \ldots, F_{|\mathcal{F}|}$. Then in step i, the nodes with labels in F_i transmit, while the other nodes listen. In this paper we extend the notion of selectivity to graphs with larger eccentricity by proving the following lemma.

Lemma 1. *Let* $P = <v_0, .., v_{k'} = c>$ *be a directed simple path, s.t.,* $|NB(P)| \leq k$. *A single application of path selector* $S_{N,k}$ *allows node* v_0 *to deliver its own message* m_0 *along path* P *to its endpoint* c.

Proof. Let $Z(-1)$ be the set of the labels of the nodes in $NB(P)$. For $i = 0, .., \lceil \log k \rceil$, let $Z(i) \subseteq Z(i-1)$ be the set of the labels in $Z(i-1)$ that are not hit by selector $\mathcal{F}_i$. Note that set $Z(i-1) - Z(i)$ contains all labels that are not hit by any $\mathcal{F}_j$, for $j = 0, .., i-1$, but are hit by $\mathcal{F}_i$.

According to the definition of selectors $\mathcal{F}_i$, the cardinality of $Z(i-1)$ is at most $k/2^i$, so $|Z(i-1) - Z(i)| \leq k/2^i$ too. Let $X_i = \{v_0, .., v_{k'-1}\} \cap (Z(i-1) - Z(i))$, for $i = 0, .., \lceil \log k \rceil$. Then $|X_i| \leq k/2^i$ too. Observe that $Z(\lceil \log k \rceil)$ is empty, since selector $\mathcal{F}_{\lceil \log k \rceil}$ hits every label in $NB(P)$, so $\bigcup_{i=0}^{\lceil \log k \rceil}(Z(i-1)-Z(i)) = NB(P)$ and $\bigcup_{i=0}^{\lceil \log k \rceil} X_i = \{v_0, .., v_{k'-1}\}$.

Note that the copy of message m_0 starting fom node v_0 and proceeding along path P will use selector $\mathcal{F}_i$ to progress from nodes in X_i, that is, at most $k/2^i$ times. Thus the total time spent by message m_0 in all sets X_i (including time multiplexing used for selector interleaving) is:

$$(\lceil \log k \rceil + 1) \sum_{i=0}^{\lceil \log k \rceil} \frac{k}{2^i} T(N, k, \frac{k}{2^{i+1}}) = T_{N,k}$$

Corollary 1. *Let $P = <v_0, .., v_{k'} = c>$ be a directed simple path, s.t., $|NB(P)| \leq k$. A single application of the path selector $S_{N,k}$, which takes $O(k^2 \log^3 n)$ times, allows all nodes in P to deliver their own messages along P to the endpoint c.*

We remark that prior to our paper, the best available upper bound on completing the communication task referred to in the above corollary was the $\tilde{O}(k^3)$ bound of k successive applications of a strong $(N, k, 0)$-selector.

3 Faster Deterministic Gossiping

The algorithms presented in this section use procedure Broadcast(v), which distributes from v all messages known to v (that is, the message originating at v and all messages received by v so far) to all other nodes in the network. We say that a message is *secured* if it has already been communicated to all nodes in the network by an application of the procedure Broadcast(v). Otherwise we say that the message is still *active*. A *dormant node* is a node whose original message is already secured. And an *active node* is a node which is not yet dormant. An *active path* is a simple path such that all nodes on this path other than the last one are active. The last node of an active path may be active or dormant.

Our algorithms use a *quasi-gossiping* principle. The *quasi-gossiping* procedure guarantees that on its completion every not yet secured message is communicated to at least one dormant node. Observe that full gossiping can be completed by an application of a quasi-gossiping procedure followed by further execution of all transmissions in this quasi-gossiping procedure in exactly the same order as in the first run.

Another important component in our algorithms is the following procedure Disperse(x), which is mainly responsible for distribution of large enough (containing at least x) combined active messages.

Disperse(x):
  repeat
    select a node v which has at least x active messages;
    if such a node v exists
      then Broadcast(v);
      else return.

Lemma 2. *On the completion of procedure* DISPERSE(x)*, each node contains less than x active messages. If an algorithm executes procedure* DISPERSE(x) *r times, then the total running time of these executions is $O(((n/x)+r)n\log^3 n)$.*

Proof. The first part of the Lemma is immediate.

The time complexity of the procedure BROADCAST() is bounded by $O(n\log^2 n)$ [2]. Selection of a node containing at least x active messages is done in time $O(n\log^3 n)$ by binary search combined with the broadcast procedure; for details see [2]. Thus each iteration takes $O(n\log^3 n)$ time. For each call to procedure DISPERSE(x), each iteration other than the last one secures at least x active messages, so the total number of iterations, over all calls, is at most $(n/x)+r$.

3.1 Gossiping in Time $\tilde{O}(n\Delta)$

We assume here that max-indegree of the network is bounded by Δ. The gossiping algorithm works in three phases reflecting the principle of quasi-gossiping. Phase I is based on application of a strong $(N,\Delta,0)$-selector $k=\lceil(n\log n)/\Delta\rceil$ times. Phase II is a single application of procedure DISPERSE(k). Phase III repeats all transmissions from Phases I and II.

GOSSIP1(n,Δ):
  Let $k=\lceil(n\log n)/\Delta\rceil$;
  Phase I (move all messages along paths of length k):
    apply $(N,\Delta,0)$-selector k times;
  Phase II (make at least every k-th node of an active path dormant):
    DISPERSE(k);
  Phase III: repeat all transmissions from Phases I and II.

Theorem 1. *The algorithm* GOSSIP1(n,Δ) *performs radio gossiping in any ad-hoc network of size n and max-indegree at most Δ in time $O(n\Delta\log^2 n)$.*

Proof. Recall that the size of $(N,\Delta,0)$-selector is $O(\Delta^2\log n)$. Thus the running time of Phase I is $k\cdot O(\Delta^2\log n)=O(n\Delta\log^2 n)$. Lemma 2 implies that the running time of Phase II is $O((n/k)n\log^3 n)=O(n\Delta\log^2 n)$. Hence the total running time is $O(n\Delta\log^2 n)$. It remains to prove that the algorithm always completes gossiping. We prove this by showing that phases I and II always complete quasi-gossiping.

If there is no simple path of length k coming out of a node v, then on the completion of Phase I, the message m_v is known to all other nodes in the network. If there is a simple path P coming out of v of length k, then at the end of Phase I, m_v is known to all nodes on P. Note that on the completion of Phase II, at least one node on P is no longer active. Otherwise the last node on P would contain at least q active messages, which is not possible after execution of the procedure DISPERSE(k). Thus m_v must be known to at least one dormant node.

3.2 Gossiping in Time $\tilde{O}(n^{4/3})$

In this section we present our main deterministic radio gossiping algorithm, which works in graphs with an arbitrary topology. The framework of the algorithm is to keep transmitting active messages so that individual nodes keep accumulate more and more "local" messages, and to apply periodically procedure Disperse(x) with appropriately chosen values of the threshold parameter x. The control of the "local" accumulation of active messages is initially done using weak selectors but at some point switches to path selectors.

The algorithm consists of three phases and follows the quasi-gossiping principle: Phases I and II complete quasi-gossiping, while Phase III is an exact repetition of all transmissions done in Phases I and II. Phase I is the same as the initial phase of the gossiping algorithm proposed by Gąsieniec *et al.* in [1]. Repeatedly apply a weak $(N, q, q/4)$-selector followed by procedure Disperse($q/4$), for q geometrically decreasing from n to k. The value of the parameter k will be set later. At the end of the i-th iteration, the size of the active neighbourhood of each node (that is, the number of active nodes in the neighbourhood of each node) is less than $n/2^i$, and at the end of the last iteration, the size of the active neighbourhood of each node is less than k. In Phase II we iterate a logarithmic number of times the path selector $\mathcal{S}_{N,k}$ followed by procedure Disperse($k/2$), to reduce the active neighbourhoods of active paths. We show below (Lemma 6) that this computation results in delivery of all active messages to dormant nodes. The pseudo-code of the gossiping algorithm follows. From now one, the neighbourhood of a node or path refers to the active neighbourhood.

Gossip2(n):
    Phase I (reduction of neighbourhoods of nodes):
        $q \leftarrow n$,
        while $q \geq k$ **do**
            the active nodes transmit according to a $(N, q, q/4)$-selector,
            Disperse($q/4$),
            $q \leftarrow q/2$;
    Phase II (reduction of neighbourhoods of active paths):
        repeat $\lceil \log k \rceil + 1$ times:
            the active nodes transmit according to the path selector $\mathcal{S}_{N,k}$,
            Disperse($k/2$);
    Phase III, repeat all communication from Phases I and II.

Lemma 3. *At the end of Phase I, the size of the neighbourhood of each node is less than k.*

Proof. A simple inductive argument using the definition of a selector shows that at the beginning of each iteration of the loop in Phase I, the size of the neighbourhood of each node is less than q.

Lemma 4. *In Phase II, if at the beginning of an iteration the size of the neighbourhood of each active path of length at most l is less than k, then at the end of this iteration the size of the neighbourhood of each active path of length at most $2l$ is less than k.*

Proof. Consider an iteration of the loop in Phase II and assume that at the beginning of this iteration the size of the neighbourhood of each active path of length at most l is less than k. Let P be a path of length at most $2l$ which is active at the end of this iteration.

If $\mathit{length}(P) \leq l$, then the size of the neighbourhood of P is already less than k at the beginning of the iteration, and it can only decrease further.

If $\mathit{length}(P) > l$, then let P_1 and P_2 be paths of length at most l each whose concatenation is path P. The size of the neighbourhood of path P_i, $i = 1, 2$, at the beginning of the iteration is less than k. Therefore Corollary 1 and Lemma 2 imply that less than $k/2$ active nodes are left in the neighbourhood of P_i at the end of the iteration. The neighbourhood of P is the union of the neighbourhoods of P_1 and P_2, so its size at the end of the iteration must be less than k.

Lemma 5. *In Phase II, at the beginning of the last iteration, the size of the neighbourhood of each active path is less than k.*

Proof. Lemma 3 implies that at the beginning of Phase II, the size of the neighbourhood of each active path of length 1 is less than k. This fact and Lemma 4 imply that at the end of iteration i, the size of the neighbourhood of each active path of length at most 2^i is less than k. Thus at the end of iteration $\lceil \log k \rceil$, each active path of length at most k has neighbourhood of size less than k. We cannot have an active path of length greater than k at the end of this iteration, because a subpath of length k of such a path would have neighbourhood of size at least k (the size of the neighbourhood of a path cannot be less than the length of this path). Hence at the beginning of the last iteration, the size of the neighbourhood of each active path is less than k.

Lemma 6. *At the end of Phase II, either the full gossiping is already completed or each active message is in a dormant node.*

Proof. If there is at least one dormant node in the network at the beginning of the last iteration in Phase II, then Lemma 5 implies that at this point of the computation, for each active node v and each active path from v to a dormant node u, the size of the neighbourhood of this path is less than k. Lemma 1 implies that the last iteration in Phase II sends the message from v to u.

If there is no dormant node in the network at the beginning of the last iteration in Phase II, then Lemmas 5 and 1 imply that the last iteration in Phase II sends the message from each node to all other nodes, completing the full gossiping. (Actually, one can show that in this case the full gossiping is completed already by the end of the first iteration in Phase II.)

Theorem 2. *The algorithm* Gossip2(n) *performs radio gossiping in ad-hoc networks of size n and arbitrary topology in time $O(n^{4/3}\log^{10/3} n)$.*

Proof. Since Phases I and II complete quasi-gossiping (Lemma 6), then the whole algorithm completes gossiping.

The running time of Phase I is $O(n \log n)$ for all applications of weak selectors, plus $O((n^2/k)\log^3 n)$ for all applications of procedure Disperse (see Lemma 2). The running time of Phase II is $O(k^2 \log^3 n \log k)$ time for all applications of the path selector, plus $O((n/k+\log k)n \log^3 n)$ time for all applications of procedure Disperse (see Lemma 2). Thus the total running time of the algorithm is $O((n^2/k + k^2 \log k + n \log k)\log^3 n)$, which is $O(n^{4/3}\log^{10/3} n)$ for $k = (n^{2/3})/\log^{1/3} n$.

4 Conclusion

In Section 3 we presented two new radio gossiping algorithms. The algorithm Gossip1(n, Δ) is designed for graphs with max-indegree bounded by Δ. With the running time $O(n\Delta \log^2 n)$, this algorithm performs best when the diameter of the network is large (close to n) and the max-degree is relatively small ($o(n^{1/3})$). The algorithm Gossip2(n) is designed for graphs with an arbitrary topology. With the running time $O(n^{4/3}\log^{10/3} n)$, this algorithm is currently the best (up to our knowledge) known deterministic radio gossiping algorithm in this case.

An obvious open problem is to close further the gap between the best currently known randomised $O(n \log^2 n)$−time gossiping, given in [12], and our new deterministic $O(n^{4/3}\log^{10/3} n)$−time gossiping procedure. It seems that to improve the deterministic upper bound one would need to introduce new, more adaptive gossiping paradigms. An implication of our main algorithm is that now the upper bounds for deterministic gossiping is the same for polynomially large node labels as for linearly large labels. One might gain some further insight into the time complexity of the gossiping problem by looking for the cases when the linear node labels enable faster algorithms than the polynomially large labels.

References

1. Gąsieniec, L., Pagourtzis, A., Potapov, I.: Deterministic communication in radio networks with large labels. In: Proc. 10th European Symposium on Algorithms (ESA'02). Volume 2461 of Lecture Notes in Computer Science, Springer Verlag (2002) 512–524
2. Chrobak, M., Gąsieniec, L., Rytter, W.: Fast broadcasting and gossiping in radio networks. Journal of Algorithms **43** (2002) 177–189
3. Gąsieniec, L., Lingas, A.: On adaptive deterministic gossiping in ad hoc radio networks. Information Processing Letters **83** (2002) 89–94
4. Xu, Y.: An $o(n^{1.5})$ deterministic gossiping algorithm for radio networks. Algorithmica **36** (2003) 93–96

5. Chlebus, B., Gąsieniec, L., Ostlin, A., Robson, M.: Deterministic radio broadcasting. In: Proc. 27th International Colloquium on Automata, Languages and Programming (ICALP'00). Volume 1853 of Lecture Notes in Computer Science, Springer Verlag (2000) 717–728
6. Clementi, A., Monti, A., Silvestri, R.: Selective families, superimposed codes, and broadcasting on unknown radio networks. In: Proc. 12th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'01), ACM/SIAM (2001) 709–718
7. Chlebus, B., Gąsieniec, L., Gibbons, A., Pelc, A., Rytter, W.: Deterministic broadcasting in unknown radio networks. In: Proc. 11th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'00), ACM/SIAM (2000) 861–870
8. Marco, G.D., Pelc, A.: Faster broadcasting in unknown radio networks. Information Processing Letters **79** (2001) 53–56
9. Brusci, D., Pinto, M.D.: Lower bounds for the broadcast problem in mobile radio networks. Distributed Computing **10** (1997) 129–135
10. Christersson, M., Gąsieniec, L., Lingas, A.: Gossiping with bounded size messages in ad-hoc radio networks. In: Proc. 29th International Colloquium on Automata, Languages and Programming (ICALP'02). Volume 2380 of Lecture Notes in Computer Science, Springer Verlag (2002) 377–389
11. Liu, D., Prabhakaran, M.: On randomized broadcasting and gossiping in radio networks. In: Proc. 8th Annual International Conference on Computing and Combinatorics (COCOON'02). Volume 2387 of Lecture Notes in Computer Science, Springer Verlag (2002) 340–349
12. Czumaj, A., Rytter, W.: Broadcasting algorithms in radio networks with unknown topology. In: Proc. 44th Symposium on Foundations of Computer Science (FOCS'03), IEEE Computer Society (2003) 492–501
13. Gaber, I., Mansour, Y.: Broadcast in radio networks. In: Proc. 6th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'95), ACM/SIAM (1995) 577–585
14. Diks, K., Kranakis, E., Pelc, A.: The impact of knowledge on broadcasting time in radio networks. In: Proc. 7th European Symposium on Algorithms (ESA'99). Volume 1643 of Lecture Notes in Computer Science, Springer Verlag (1999) 41–52
15. Gąsieniec, L., Potapov, I.: Gossiping with unit messages in known radio networks. In: Proc. 2nd IFIP International Conference on Theoretical Computer Science (TCS'02). Volume 223 of IFIP Conference Proceedings, Kluwer (2002) 193–205
16. Bonis, A.D., Gąsieniec, L., Vaccaro, U.: Generalized framework for selectors with applications in optimal group testing. In: Proc. 30th International Colloquium on Automata, Languages and Programming (ICALP'03). Volume 2719 of Lecture Notes in Computer Science, Springer Verlag (2003) 81–96