Time Efficient Radio Broadcasting in Planar Graphs*

Fredrik Manne[1], Qin Xin[2,†,*]
[1]Department of Informatics, The University of Bergen, Bergen, Norway. Email: fredikm@ii.uib.no
[2]Simula Research Laboratory, Lysaker, Norway. Email: xin@simula.no

***Abstract*— We study the communication primitive of broadcasting (one-to-all communication) in known topology radio networks, i.e., where for each primitive the schedule of transmissions is precomputed based on full knowledge about the size and the topology of the network. We show that radio broadcasting can be completed in $D + O(\log n)$ time units in planar graphs of size n, and with diameter D. This improves the currently best known $D + O(\log^3 n)$ time schedule proposed by Elkin and Kortsarz in [16] [SODA'05], and $3D$ time schedule due to Gąsieniec, Peleg and Xin in [23] [PODC'05]. In this paper, we also explore broadcasting in radio networks in the presence of edge failures.**

***Index Terms*— Broadcasting, centralized radio networks, fault-tolerance, gossiping, planar graphs.**

I. Introduction

We consider the following radio network model: an undirected connected graph $G = (V, E)$, where V represents the set of nodes of the network and E contains unordered pairs of distinct nodes, such that $(v, w) \in E$ iff the transmissions of node v can directly reach node w and vice versa (the reachability of transmissions is assumed to be a symmetric relation). In this case, we say that the nodes v and w are *neighbours* in G. Note that in a radio network, a message transmitted by a node is always sent to all of its neighbors.

The *degree* of a node w is its number of neighbours. We use Δ to denote the *maximum degree* of the network, i.e., the maximum degree of any node in the network. The *size of the network* is the number of nodes $n = |V|$.

Communication in the network is synchronous and consists of a sequence of communication steps. In each step, a 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 adjacent to v successfully receives this message iff in this step w is listening and v is the only transmitting node among w's neighbors. If node w is adjacent to a transmitting node but it is not listening, or it is adjacent to more than one transmitting node, then a *collision* occurs and w does not retrieve any message in this step.

The two classical problems of information dissemination in computer networks are the *broadcasting* problem and the *gossiping* problem. The broadcasting problem requires distributing a particular 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 the aim is to distribute all messages to all nodes. For both problems, one generally considers as the efficiency criterion the minimization of the time needed to complete the task.

In the model considered here, the running time of a communication schedule is determined by the number of time steps required to complete the communication task. This means that we do not account for any internal computation within individual nodes. Moreover, no limit is placed on the length of a message which one node can transmit in one step. In particular, this assumption plays an important role in the case of the gossiping problem, where it is then assumed that in each step when a node transmits, it transmits all the messages it has collected by that time. (i.e., the ones received and its own one.)

Our schemes rely on the assumption that the communication algorithm can use complete information about the network topology. Such topology-based communication algorithms are useful whenever the underlying radio network has a fairly stable topology/infrastructure. As long as no changes occur in the network topology during the execution of the algorithm, the tasks of broadcasting and gossiping will be completed successfully. In this paper, we also consider reliability issues. Furthermore, we show that it is possible to increase the level of fault-tolerance in our algorithms, at the expense of some small extra time consumption to deal with a limited number of edge failures. We defer this issue to Section III.

A. Our results

We provide a new (efficiently computable) deterministic schedule that uses $D+O(\log n)$ time units to complete the broadcasting task in any planar graph of size n, and with diameter D. This significantly improves on the previously known best schedule, i.e., the $D + O(\log^3 n)$ schedule from [16]. Remarkably, our new broadcasting scheme also improves the $3D$-time schedule in [23] for

The preliminary version of this paper was appeared in the Proceedings of 4th Annual Conference on Wireless on Demand Network Systems and Services, WONS'07, January 2007, Obergurgl, Austria.

*This work was supported in part by the Research Council of Norway through the SPECTRUM project.

†Corresponding author (fax: 0047 67828201).

*The work was also supported in part by the Research Council of Norway through the project on Resilient Wireless Networks at Simula Research Laboratory.

the graphs with large diameter. In this paper, we also propose a $2D+O(\log n)$-time algorithm for fault-tolerant radio broadcasting in planar graphs, in the presence of a constant number of edge failures.

B. Related work

The work on communication in known topology radio networks was initiated in the context of the broadcasting problem. In [12], Chlamtac and Weinstein prove that the broadcasting task can be completed in time $O(D\log^2 n)$ for every n-vertex radio network of diameter D. An $\Omega(\log^2 n)$ time lower bound was proved for the family of graphs of radius 2 by Alon *et al* [1]. In [16], Elkin and Kortsarz give an efficient deterministic construction of a broadcasting schedule of length $D + O(\log^4 n)$ together with a $D+O(\log^3 n)$ schedule for planar graphs. Recently, Gąsieniec, Peleg and Xin [23] showed that a $D + O(\log^3 n)$ schedule exists for the broadcast task, that works in *any* radio network. In the same paper, the authors also provide an optimal randomized broadcasting schedule of length $D+O(\log^2 n)$ and a new broadcasting schedule using fewer than $3D$ time slots on planar graphs. Very recently, a $24D + O(\log^2 n)$ time deterministic broadcasting schedule for any radio network was proposed by Kowalski and Pelc in [28]. This is asymptotically optimal unless $NP \subseteq BPTIME(n^{\mathcal{O}(\log\log n)})$ [28]. Nonetheless, for large D, a $D + O(\frac{\log^3 n}{\log\log n})$ time broadcasting scheme outperforms the one in [28], which was proposed by Cicalese, Manne and Xin in [10] very recently. Efficient radio broadcasting algorithms for several special types of network topologies can be found in Diks *et al.* [13]. For general networks, however, it is known that the computation of an optimal (radio) broadcast schedule is NP-hard, even if the underlying graph is embedded in the plane [8], [34].

Many authors also studied deterministic distributed broadcasting in ad-hoc radio networks, in which every node knows only its own label, using the model of directed graphs, see for instance [4]–[7], [9], [11], [14], [26]. Increasingly faster broadcasting algorithms working on arbitrary n-node (directed) radio networks were constructed, the currently fastest being the $O(n\log^2 D)$-time algorithm from [11]. (Here D is the diameter of the network, i.e, the longest distance from the source to any other node). On the other hand, in [9] a lower bound of $\Omega(n\log D)$ on the time required to perform broadcasting was proved for directed n-node networks of radius D.

Very few results [15], [25], [33] are known about radio broadcasting in presence of node (edge) failures in contrast to plethora of papers on fault-tolerant communication in wired P2P networks (For a survey see [33]).

Radio gossiping in networks with known topology was first studied in the context of radio communication with messages of limited size, by Gąsieniec and Potapov in [21]. They also proposed several optimal or close to optimal $O(n)$-time gossiping procedures for various standard network topologies, including lines, rings, stars and free trees. For general topology radio network a $O(n\log^2 n)$ gossiping scheme is provided and it is proved that there exists a radio network topology in which the gossiping (with unit size messages) requires $\Omega(n\log n)$ time. In [30], Manne, and Xin show the optimality of this bound by providing an $O(n\log n)$-time gossiping schedule with unit size messages in any radio network. The first work on radio gossiping in known topology networks with arbitrarily large messages is [22], where several optimal gossiping schedules are shown for a wide range of radio network topologies. For arbitrary topology radio networks, an $O(D + \Delta\log n)$ schedule was given by Gąsieniec, Peleg, and Xin in [23]. Very recently, Cicalese, Manne, and Xin [10] provided a new (efficiently computable) deterministic schedule that uses $O(D + \frac{\Delta\log n}{\log\Delta - \log\log n})$ time units to complete the gossiping task in any radio network of maximum degree $\Delta = \Omega(\log n)$. Later in [31], Manne and Xin further improve the gossiping time to $O(D+\frac{\Delta\log n}{\log\Delta})$ in any radio network of maximum degree $\Delta = \Omega(\log^{\frac{c}{c-1}} n)$, for any constant $c > 1$, which is an optimal schedule in the sense that there exists a radio network topology, specifically a Δ-regular tree, in which the radio gossiping cannot be completed in less than $\Omega(D + \frac{\Delta\log n}{\log\Delta})$ units of time.

So far, the gossiping problem has mostly been studied in the context of ad-hoc radio networks, where the topology of connections is unknown to the nodes. In this model, Chrobak *et al.* [7] proposed a fully distributed deterministic algorithm that completes the gossiping task in time $O(n^{3/2}\log^3 n)$. For small values of the diameter D, the gossiping time was later improved by Gąsieniec and Lingas [18] to $O(nD^{1/2}\log^3 n)$. Another interesting $O(n^{3/2})$-time algorithm, a tuned version of the gossiping algorithm from [7], can be found in [37]. A very recent $O(n^{4/3}\log^{10/3} n)-$time gossiping algorithm has been proposed by Gąsieniec, Radzik, and Xin in [24]. A study of deterministic gossiping in ad-hoc radio networks, with messages of limited size, can be found in [17]. The gossiping problem in *ad-hoc* radio networks also attracted studies based on efficient randomized algorithms. In [7], Chrobak et al. proposed an $O(n\log^4 n)$-time gossiping procedure. This time was later reduced to $O(n\log^3 n)$ [29], and very recently to $O(n\log^2 n)$ [11].

II. BROADCASTING IN PLANAR GRAPHS WITH KNOWN TOPOLOGY

In this section we present the idea of a deterministic algorithm that generates a schedule for completing the broadcasting task in a planar graph in time $D+O(\log n)$. Our schedule is based on the notion of a *gathering spanning tree* as given in [23].

A. Preliminaries

We first recall the following recursive ranking procedure of nodes in a rooted tree (see [23]). Leaves have rank 1. Next consider a node v and the set Q of its children and let r_{max} be the maximum rank of the nodes in Q. If there is a unique node in Q of rank r_{max} then set the rank of v to r_{max}, otherwise set the rank of v to $r_{max} + 1$.

Lemma 1: The largest rank in a tree of size n is bounded by $\lceil \log n \rceil$ (see [23]).

Given any graph G with source node s, the nodes are partitioned into consecutive layers $L_i = \{v \mid dist(s, v) = i\}$, for $i = 0, .., D$ where D is the diameter of G. A *gathering spanning tree GST* of G is a BFS spanning tree T of G rooted at s, such that T is ranked as above and that also satisfies the following condition: every node in L_{j+1} of rank i is at most adjacent to one node in L_j also of rank i, and thus if all the nodes of rank i in L_j transmit at the same time then the messages will be received by the nodes in L_{j+1} of rank i successfully without any collisions.

The following lemma was shown in [23].

Lemma 2: There exists a polynomial time construction of a GST in any graph G.

Figure 1 shows how a gathering spanning tree can be constructed from a graph G.

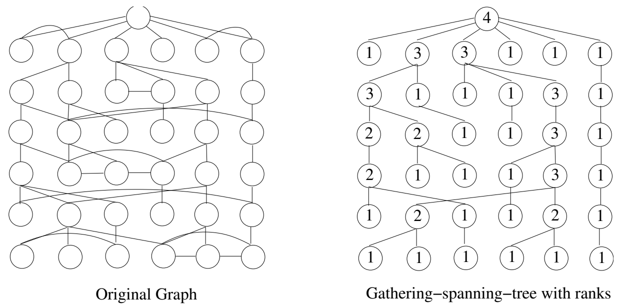

Figure 1. Creating a gathering spanning tree.

In the following we assume a ranked *GST* in G given by the parent relation on each node. For clarity of presentation, we use the same definitions as in [23].

Definition 3: the *rank sets*: $R_i = \{v \mid \mathtt{rank}(v) = i\}$, where $1 \le i \le r_{max} \le \lceil \log n \rceil$.

Definition 4: the *fast transmission set*: $F_i^k = \{v \mid v \in L_k \cap R_i$ and $parent(v) \in R_i\}$. We also define $F_i = \bigcup_{k=1}^{D} F_i^k$ and $F = \bigcup_{i=1}^{r_{max}} F_i$.

Definition 5: the *slow transmission set*: $S_i^k = \{v \mid v \in L_k \cap R_i$ and $parent(v) \in R_j, j > i\}$. We also define $S_i = \bigcup_{k=1}^{D} S_i^k$ and $S = \bigcup_{i=1}^{r_{max}} S_i$.

Lemma 6: In a planar graph, all nodes in set S_{i+1} of a GST can receive the source message from their parents in L_i of the GST in three time units.

Proof. In [23], it states that all nodes in one partition can be informed by another partition in a bipartite planar graph (in this case two consecutive BFS layers) in three time units. The solution is based on the use of the *3-step subschedule Procedure PB*. (See an example in Figure 2.) Note that during this process the source message m_s received by a node $v \in S_{i+1}$ may be delivered by some other nodes in L_i of GST rather than exactly by its parent in the GST.

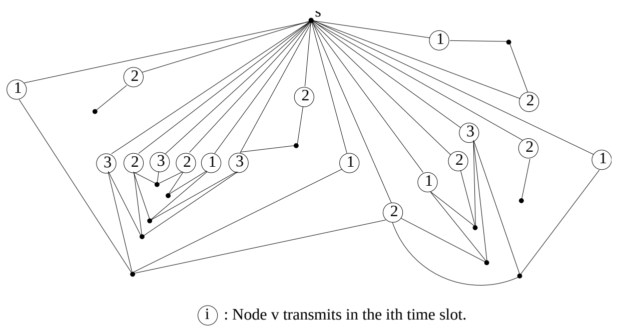

Figure 2. An example of Procedure PB. (All solid black nodes in L_2 will receive the source message from the white nodes in L_1 after three time slots. Note that all white nodes have already received the source message from s in the previous time slot.)

B. Deterministic construction of a $D + O(\log n)$ broadcasting schedule

The deterministic algorithm uses the ranked gathering spanning tree GST, on this occasion rooted in the *source node* s. The algorithm uses *fast* and *slow* transmissions that partition the set of nodes to the sets F and S, where the broadcast message is disseminated from the root s (using parent-child connections) towards the leaves of the tree.

Let us start with an overview of the broadcast process from the point of view of a copy of the message that was eventually received at some leaf a of the tree. Note that this message does not necessarily have to follow the unique shortest path $p(a)$ leading from the root of the tree to a. In fact, there are many paths on which the message could be forwarded. For the sake of the time complexity analysis, however, we fix our attention on the path $p(a)$ and argue about the potential progress of the message along this path.

Conceptually, the path $p(a)$ is broken down into segments

$$p(a) = \langle p_1^F(a), p_1^S(a), p_2^F(a), p_2^S(a), \ldots, p_q^F(a), p_q^S(a)\rangle ,$$

where each $p_i^F(a)$ is a segment consisting of fast transmission edges (i.e., edges leading from $parent(v)$ to v, both of the same rank) and each $p_i^S(a)$ is a single edge (u, w) where u is a node on layer L_k for some k, w is a node on layer L_{k+1} and $\mathtt{rank}(u) > \mathtt{rank}(w)$. We refer to such edges (u, w) as slow transmission edges (see Figure 3 for an example). (Note that some of the $p_i^F(a)$ segments may be empty.)

Again, we stress that in reality, the message need not follow this path. Nevertheless, we may consider the

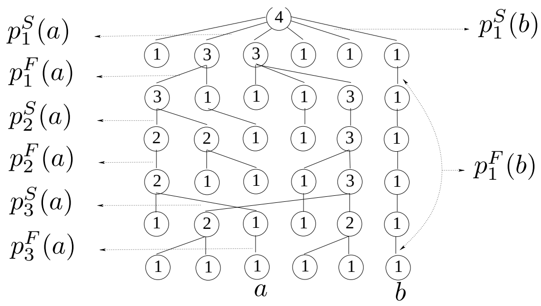

Figure 3. An example of the path partition.

"progress" of the message along this path, by measuring the delay from the time the message is already available at some node v on the path $p(a)$ to the time the message has already reached the following node w on the path (though not necessarily via a transmission from v). Hence conceptually, the message progress can be viewed as traversing the path $p(a)$ by alternating (flipping) between chains of fast transmission edges connecting nodes of the same rank and slow transmission steps over edges, connecting higher ranked nodes to lower ranked nodes.

Next we describe the schedule governing these transmissions. During the broadcasting process the nodes in the tree use the following pattern of transmissions. Let $r_{max} \leq \lceil \log n \rceil$ be the largest rank in the tree. Consider a node v of rank j, where $1 \leq j \leq r_{max}$ on BFS layer L_i with a child w also of rank j at the next BFS layer L_{i+1}. Then v is set to perform a fast transmission to w in time steps t satisfying $t \equiv i + 6(r_{max} - j) \bmod 6r_{max}$. Observe that in real terms, v will perform such a fast transmission exactly once, on the first appropriate time slot after it receives the message for the first time. The slow transmissions on BFS layer L_i are performed in time steps t satisfying $t \equiv i + 3 \bmod 6$. Note that this pattern of transmissions separates the fast and the slow transmissions at any BFS layer by three units of time. Thus there are no collisions between the fast and the slow transmissions at the same BFS layer. The pattern also ensures that at any time step, transmissions are performed on BFS layers at distances that are multiples of 3 apart. Thus there will be no conflicts between transmissions coming from different BFS layers.

Note also that once the broadcast message arrives at the first node v of a fast segment $p_i^F(a)$ of the route with a particular rank j, it may have to wait for at most 6 (e.g. $O(1)$) time steps if the parent of v in the GST has rank $j+1$, otherwise it has to wait at most $6(rank(parent(v)) - j)$, but then, when finally transmitted to the next BFS layer, it will be forwarded through the fast segment $p_i^F(a)$ without further delays.

Once reaching the end node u of the fast segment $p_i^F(a)$, the message has to be transmitted from some node on u's BFS layer to the next node w on $p(a)$, which is of lower rank, using a slow transmissions mechanism procedure PB which was discussed in Lemma 6. The slow transmission mechanism is run repeatedly in a periodic manner at every BFS layer of the tree. In particular, at any BFS layer, the steps of the slow transmission procedure PB are performed in every 6th step of the broadcasting schedule.

Hence, suppose the broadcast message traversing towards any destination a in the tree has reached a node u of BFS layer L_j on its path $p(a)$, such that the next edge (u, w) on the path is a slow transmission edge. It is possible that neither u nor any other neighbor of w on BFS layer L_j participates in the current activation of procedure PB on L_j (possibly because neither of those nodes had the message at the last time the procedure was activated). Nevertheless, u will participate in the next activation of procedure PB on BFS layer L_j, which will be started within at most $O(1)$ time (namely, the time required for the current activation to terminate). Moreover, it is guaranteed that by the time that activation of procedure PB terminates, w will have the message (although it may get it from any of its neighbors in L_j, and not necessarily directly from u). Hence this entire stage can be thought of as a slow transmission operation on the edge (u, w), taking a total of at most $O(1)$ time steps.

In the view of these observations, the total time required for the broadcast message to reach a leaf a in the tree can be bounded as follows. Let D_i, for $1 \leq i \leq r_{max}$, denote the length of $p_i^F(a)$, the ith fast segment of the route $p(a)$ used by the broadcast message to reach a. Thus the time required for the broadcast message to reach a is bounded by $O(1) + D_1 + O(1) + D_2 + \ldots + O(1) + D_{r_{max}} \leq D + O(\log n)$ for the fast transmissions plus $r_{max} \cdot O(1) = O(\log n)$ for the slow transmissions, yielding a total of $D + O(\log n)$. Thus we have the following theorem.

Theorem 7: There exists a deterministic polynomial time algorithm that constructs, for any planar graph of size n and diameter D, a broadcasting schedule of length $D + O(\log n)$.

C. An example of the broadcasting schedule

In this section, we give an example for our broadcasting schedule.

(i) Construction of a gathering spanning tree. (See Figure 4.)

(ii) The transmission pattern (fast: $\xrightarrow{f}$; slow: $\xrightarrow{s}$).

Steps	Transmissions
(1)	$1 \xrightarrow{s} 2 \parallel 1 \xrightarrow{s} 3 \parallel 1 \xrightarrow{s} 4 \parallel$ $1 \xrightarrow{s} 5 \parallel 1 \xrightarrow{s} 6 \parallel 1 \xrightarrow{s} 7 \parallel$
(2)	$3 \xrightarrow{f} 8 \parallel 4 \xrightarrow{f} 12 \parallel$
(3)	$12 \xrightarrow{f} 18 \parallel$
(4)	$18 \xrightarrow{f} 24 \parallel 3 \xrightarrow{s} 9 \parallel 4 \xrightarrow{s} 10 \parallel 4 \xrightarrow{s} 11 \parallel$
(5)	$8 \xrightarrow{s} 14 \parallel 8 \xrightarrow{s} 15 \parallel$
(6)	$15 \xrightarrow{s} 21 \parallel 15 \xrightarrow{s} 22 \parallel 18 \xrightarrow{s} 23 \parallel$
(7)	$24 \xrightarrow{s} 27 \parallel 24 \xrightarrow{s} 30 \parallel$

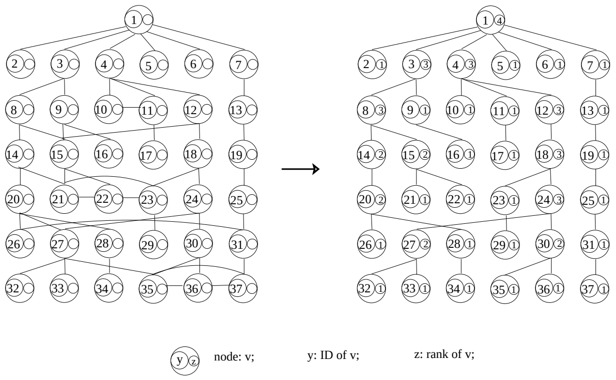

Figure 4. Construct a gathering spanning tree from the original planar graph.

(8) $27 \xrightarrow{s} 32 \;||\; 27 \xrightarrow{s} 33 \;||\; 30 \xrightarrow{s} 36 \;||$

(9) $14 \xrightarrow{f} 20 \;||$

(10) $20 \xrightarrow{s} 26 \;||\; 20 \xrightarrow{s} 28 \;||\; 7 \xrightarrow{f} 13 \;||$

(11) $9 \xrightarrow{f} 16 \;||\; 11 \xrightarrow{f} 17 \;||\; 13 \xrightarrow{f} 19 \;||$
$30 \xrightarrow{s} 35 \;||$

(12) $19 \xrightarrow{f} 25 \;||$

(13) $23 \xrightarrow{f} 29 \;||\; 25 \xrightarrow{f} 31 \;||$

(14) $28 \xrightarrow{f} 34 \;||\; 31 \xrightarrow{f} 37 \;||$

III. FAULT-TOLERANT BROADCASTING

In this section, we show how a fault-tolerant protocol for the broadcasting task in any planar graph could be achieved in time $2D + O(\log n)$. This has almost same time complexity as the fault-free broadcasting schedule we proposed in the previous section.

A. The model

We now assume that all nodes have distinct identities, and each node has full knowledge about the topology of the network. The network now contains a constant number of edge failures but the nodes do not know which edges are at fault. Furthermore, we assume that there are no further edge failures during the execution of our fault-tolerant algorithm. To make the broadcasting problem feasible, we have to assume that the fault edges would not disconnect the graph.

Under these assumptions, the following lemma holds.

Lemma 8: Let D and D' denote the diameters of original graph and the fault-tolerant graph respectively, then $D' = D + O(1)$.

B. The broadcasting protocol

Our fault-tolerant broadcasting protocol consists of three consecutive and disjoint stages labeled DETECTION, CONVERGECAST, and BROADCAST respectively. In the DETECTION stage, all the fault edges that are used in the fault-free broadcasting schedule in Section II are detected. In the CONVERGECAST stage, all detected information about edge failures from the previous stage are collected at the source node s. Finally, in the BROADCAST stage, the combined information of the source message, updated knowledge of the topology of the network, and a fault-free broadcasting schedule which computed at s is distributed to all nodes from s.

Let c denote the number of edge failures. We use the notation $T_D(n, D, c)$, $T_C(n, D, c)$, and $T_B(n, D, c)$, to denote the number of rounds used by DETECTION, CONVERGECAST, and BROADCAST respectively. It is then clear that our fault-tolerant broadcasting protocol solves the broadcasting problem in time $T_D(n, D, c) + T_C(n, D, c) + T_B(n, D, c)$. We formulate this in the following result.

Lemma 9: If the protocols DETECTION, CONVERGECAST, and BROADCAST used in the fault-tolerant broadcasting protocol complete their assigned task, then the fault-tolerant broadcasting protocol completes the broadcasting task in time $T_D(n, D, c) + T_C(n, D, c) + T_B(n, D, c)$.

C. The DETECTION stage

During this stage, all fault edges that are used in the fault-free broadcasting schedule in Section II are marked.

All nodes first compute the ranked *gathering spanning tree* GST rooted at s. The communication process is now split into consecutive blocks of 6 time units each. The first 3 units of each block are used for fast transmissions from the set F, and the remaining 3 units are reserved for slow transmissions from the set S. We use 3 units of time for each type of transmission in order to prevent collisions between neighbouring BFS layers, which is the same approach as in [10], [23], [30].

Recall that we can inform all children of the nodes in S_j^k in GST within 3 time units due to Lemma 6.

We compute for each node $v \in S_j^k$ at layer k the number of a step $1 \le s(v) \le 3$ in which node v can transmit without interruption from other nodes in S_j^k also in layer k. Let v be a node with rank j in the GST.

Depending on if v belongs to the set F, or to the set S, it will transmit in the time block $t(v)$ given by:

$$t(v) = \begin{cases} 3 \cdot (j-1) + 1 & \text{if } v \in F \\ 3 \cdot (j-1) + s(v) & \text{if } v \in S \end{cases}$$

We observe that any node v at layer k in the GST is required to wait at most $O(\log n)$ time units to transmit. Further, due to the properties of the GST and Lemma 6, any node w will wait for at most $O(\log n)$ time units to receive the message successfully from their parents in the fault-free GST. Also note that w will know the exact time when it is supposed to receive the message. Furthermore, if the message failed to get to the node w from its parent v by the scheduled time, then w will response to report this failure to the source node s in the CONVERGECAST stage. We call such a node w, as a corresponding node. Moreover, the above definition of

$t(v)$ results in the following lemma.

Lemma 10: All edge failures in the GST can be detected by the corresponding nodes in time $O(\log n)$. Thus $T_D(n, D, c) = O(\log n)$.

D. The CONVERGECAST stage

In this stage, all information computed from previous stage related to the edge failures will be collected at the source node s.

1) Resolving competition: The main difficulty that occurs in radio communication is the presence of collisions. It has been shown before, see, e.g., [7], [9], [20], [24], that the most efficient tools designed for collision resolution are based on combinatorial structures possessing a *selectivity property*. We say that a set R hits a set Z on element z, if $R \cap Z = \{z\}$, and a family of sets $\mathcal{F}$ hits a set Z on element z, if $R \cap Z = \{z\}$ for at least one $R \in \mathcal{F}$. In [9] a family of subsets of the set $\{0, 1, \ldots, N-1\} \equiv [N]$ is defined that hits each subset of $[N]$ of size at most $k \leq N$ on all of its k elements. This family of subsets is referred to as being *strongly $k-$selective*. It is also shown that there exists such a family of size $O(k^2 \log N) = O(k^2 \log n)$, which is also referred to as a strong $k-$selector. The work presented in [7] defines a family of subsets of the set $\{0, 1, \ldots, N-1\} \equiv [N]$ that hits each subset of $[N]$ of size at most k on at least $k/2$ distinct elements, where $N \geq k \geq 1$. This family is referred to as a *$k-$selector* and such a family of size $O(k \log N) = O(k \log n)$ is shown to exist.

In the following we show how to cope with collisions that occur during the competition process through the use of selective families and selectors.

2) Promoting messages in bipartite graphs: Assume that we have a connected bipartite graph B in which nodes are partitioned into two sets U and L. While, in general, nodes in U and L are aware of the presence of each other, we assume here that each node $v \in L$ is associated with exactly one of its neighbors $u \in U$ (labeled as the *parent* of v) and that this relation is known to both of them. Note that a node in U can be the parent of several nodes in L. Due to the edge failures observed by the corresponding nodes, the corresponding nodes in L must to choose a new parent in U. In what follows we show how to move at most c messages which indicated the edge failures that are available at the corresponding nodes of L, to the parent nodes in U in time $O(c \log n)$, if the edge between a corresponding node v and the parent v chose is fault-free, where the constant c is upper bound of the edge failures in the network.

It is known that a communication mechanism based on the selector idea allows a fraction (e.g., a half) of the c competing nodes in L to deliver their messages to their parents in U in time $O(c \log n)$ [7]. Let $S(c)$ represent the collision resolution mechanism based on selectors. Note that $S(c)$, if applied in undirected networks, can be supported by an *acknowledgement of delivery* mechanism in which each transmission from the participating nodes in L is alternated with an acknowledgement message coming from the parent node $u \in U$. If during the execution of $S(c)$ a transmission from v towards u is successful, i.e., one of u neighbors succeeds in delivering its message, the acknowledgement issued by u and returned to v confirms the successful transmission; otherwise the acknowledgement is null. Let $\mathbf{S}(c)$ be the mechanism with this acknowledgement feature added to $S(c)$. In other words, the use of $\mathbf{S}(c)$ allows us to exclude all nodes in L that have managed to deliver their message to their parent in U during the execution of $\mathbf{S}(c)$ from further transmissions. Note that the duration of $\mathbf{S}(c)$ is $O(c \log n)$, see [7].

Let $S^*(i)$ be the communication mechanism based on the concatenation (superposition) of i selectors $S(2^i), S(2^{i-1}), \ldots, S(2^1)$. We call this a *descending selector*. The descending selector extended by the acknowledgement mechanism, i.e., the concatenation of $\mathbf{S}(2^i), \mathbf{S}(2^{i-1}), \ldots, \mathbf{S}(2^1)$, forms *a promoter* and it is denoted by $\mathbf{S}^*(c)$. Note that the duration of $\mathbf{S}^*(c)$ is $O(c \log n)$.

Lemma 11: The message from a corresponding node v from one partition of a bipartite graph can be sent to its parent u in another partition in time $O(c \log n)$, if the edge (v, u) is fault-free.

Proof. The proof is done by induction, and is based on the fact that after the execution of each $\mathbf{S}(2^j)$, for $j = \lceil \log c \rceil, \ldots, 1$, the number of competing nodes in L is bounded by 2^{j-1}.

We run the *descending selector* c times to guarantee that each corresponding node in L will choose a fault-free parent $u \in U$ properly, if there exists such a u. Note that a corresponding node will propose a new potential parent node if it failed to get one in previous rounds.

Corollary 12: All messages from at most c corresponding nodes can be collected from one partition of a bipartite graph to another partition in time $O(c^2 \log n)$, if there exists at least one fault-free edge between the node $u \in U$ and each corresponding node in L.

In the case that there does not exist any fault-free edge between a corresponding node v in L and any node $u \in U$, the node v arbitrarily picks one of its neighboring nodes in the highest BFS layer as it parent. Due to Lemma 8, the time complexity will be same as for the former case. From now on, we only analyse the former case.

3) 1-reduction approach in bipartite graphs: The 1-reduction approach is used to collect all the messages from the c corresponding nodes in L to at most $c-1$ nodes in U, when the collision occurs.

When each corresponding node $v \in L$ promots the message to its parent $u \in U$, this relation is known to both of them. Furthermore, each $v \in L$ is associated with exactly one node $u \in U$. This also means that the edge between a corresponding node v and its parent u is fault-free. In the 1-reduction approach, we will use these fault-free edges for communication.

Our 1-reduction approach consists of three consecutive phases. The first phase is to inform the corresponding nodes about the fault-free edges between it and other parent nodes. This could be done by running a *strong selector* (of size $O(c^2 \log n)$) on the parent nodes. In the second phase, a corresponding node v will choose a new parent u such that (v, u) is a fault-free edge, and the identity of u is smaller than its current parent, where u is a parent node for other corresponding nodes. This can be performed by execution of a *descending selector* to inform the new parent of v. It is clear that there exists at least one parent nodes with two children from the corresponding nodes. The third phase is to withdraw the relationship between the corresponding node v and its previous parent. Similarly, it could be also achieved by execution of a *descending selector*.

Due to Lemma 11 and the properties of the *strong selector* we employed, the following lemma holds.

Lemma 13: After one execution of the 1-reduction approach at least two messages from competing nodes will be grouped together by one of their parents in time $O(c^2 \log n)$, where c is the upper bound of the number of edge failures in the network.

In our further considerations, the sets U and L will correspond to two adjacent BFS levels, upper and lower respectively, in a gathering spanning tree of G.

The messages from the corresponding nodes will promot to the source node c along the path on the modified gathering spanning tree using the fault-free edges layer by layer from bottom to top.

We observe that any corresponding node v in the modified GST requires at most $D + O(1)$ collision-free transmissions, c executions of the 1-reduction approach to deliver its message to the source node s. This results in the following lemma.

Lemma 14: All messages about the edge failures can be collected at the source node s in time $D+O(c^2 \log n)$.

Corollary 15: $T_C(n, D, c) = D + O(c^2 \log n) = D + O(\log n)$.

E. The BROADCAST stage

When the source node s has received all information about the edge failures, s could update the knowledge of the topology of the network, and compute a fault-free broadcasting schedule. Finally, the compound message of the source message, the updated topology information of the network, and the fault-free broadcasting schedule is broadcast by the source node s to the other nodes in the network according to the offline fault-free broadcasting schedule. This can be achieved in time $D + O(\log n)$ due to Theorem 7.

Lemma 16: $T_B(n, D, c) = D + O(\log n)$.

Combining Lemma 9, Lemma 10, Corollary 15, and Lemma 16, we get the desired result.

Theorem 17: The fault-tolerant broadcasting problem with constant number of edge failures for any planar graph of size n and diameter D, can be solved in time $D + O(\log n)$.

IV. CONCLUSION

We have proposed an efficient (polynomial time) construction of a deterministic schedule that performs radio broadcasting in time $D + O(\log n)$ in planar graphs of size n, and with diameter D. The evident open problem is whether there exists a deterministic broadcasting schedule of time $D+O(1)$ for planar graphs. In this paper, we also considered reliability issues. Fault-tolerant broadcasting with a large number of edge failures is left as an intriguing open problem.

REFERENCES

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

[2] S. Banerjee, and S. Khuller, A Clustering Scheme for Hierarchical Control in Multi-hop Wireless Networks, In Proc. *INFOCOM 2001*, pp 1028-1037.

[3] 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, *Journal of Computer and System Sciences*, 45 (1992), pp. 104–126.

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

[5] B. Chlebus, L. Gąsieniec, A. Gibbons, A. Pelc, and W. Rytter, Deterministic broadcasting in unknown radio networks, Distributed Computing 15 (2002), 27-38.

[6] B. Chlebus, L. Gąsieniec, A. Ostlin, and M. Robson, Deterministic Radio Broadcasting, Proc.*27th Int. Colloq. on Automata, Languages and Programming*, ICALP'00, pp 717-728.

[7] M. Chrobak, L. Gąsieniec, and W. Rytter, Fast Broadcasting and Gossiping in Radio Networks, *Journal of Algorithms* 43(2), 2002, pp. 177-189.

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

[9] A.E.F. Clementi, A. Monti, and R. Silvestri, Selective families, superimposed codes, and broadcasting on unknown radio networks, Proc. 12th Ann. ACM-SIAM Symposium on Discrete Algorithms (SODA'2001), 709-718.

[10] F. Cicalese, F. Manne, and Q. Xin. Faster Centralized Communication in Radio Networks. *Proc. 17th International Symposium on Algorithms and Computation*, LNCS 4288, pp. 339-348.

[11] A. Czumaj and W. Rytter, Broadcasting algorithms in radio networks with unknown topology, Proc. 44th Ann. Symp. on Foundations of Computer Science, (FOCS'2003), 492-501.

[12] I. Chlamtac and O. Weinstein. The wave expansion approach to broadcasting in multihop radio networks. *IEEE Trans. on Communications* 39, (1991), pp. 426-433.
[13] K. Diks, E. Kranakis, D. Krizanc, and A. Pelc. The impact of information on broadcasting time in linear radio networks. *Theoretical Computer Science*, 287, (2002), pp. 449-471.
[14] G. De Marco and A. Pelc, Faster broadcasting in unknown radio networks, Information Processing Letters 79 (2001), 53-56.
[15] Y. Desmedt, Y. Wang, R. Safavi-Naini, and H. Wang. Radio Networks with Reliable Communication. *11th Annual International Conference on Computing and Combinatoric*, COCOON'05: 156-166.
[16] M. Elkin and G. Kortsarz. Improved broadcast schedule for radio networks. *Proc. 16th ACM-SIAM Symp. on Discrete Algorithms*, SODA 2005, pp. 222-231.
[17] M. Christersson, L. Gąsieniec, and A. Lingas, Gossiping with bounded size messages in ad-hoc radio networks, *29th International Colloquium on Automata, Languages and Programming*, ICALP'02, pp. 377-389.
[18] L. Gąsieniec and A. Lingas, On adaptive deterministic gossiping in ad hoc radio networks, *Information Processing Letters* 2(83), 2002, pp. 89-94.
[19] I. Gaber and Y. Mansour. Broadcast in radio networks. *Proc. 6th ACM-SIAM Symp. on Discrete Algorithms*, 1995, pp. 577-585.
[20] L. Gąsieniec, E. Kranakis, A. Pelc, and Q. Xin. Deterministic M2M multicast in radio networks. *Proc. 31st ICALP*, 2004, LNCS 3142, pp. 670-682.
[21] L. Gąsieniec and I. Potapov, Gossiping with unit messages in known radio networks. *Proc. 2nd IFIP Int. Conference on Theoretical Computer Science*, 2002, pp. 193-205.
[22] L. Gąsieniec, I. Potapov, and Q. Xin. Efficient gossiping in known radio networks. *Proc. 11th SIROCCO*, 2004, LNCS 3104, pp. 173-184.
[23] L. Gąsieniec, D. Peleg, and Q. Xin. Faster communication in known topology radio networks. *Proc. 24th Annual ACM SIGACT-SIGOPS PODC*, 2005, pp. 129-137.
[24] L. Gąsieniec, T. Radzik, and Q. Xin. Faster deterministic gossiping in ad-hoc radio networks. *Proc. 9th Scandinavian Workshop on Algorithm Theory*, 2004, LNCS 3111, pp. 397-407.
[25] E. Kranakis, D. Krizanc, and A. Pelc. Fault-Tolerant Broadcasting in Radio Networks. *J. Algorithms*, 39(1): 47-67 (2001).
[26] D. Kowalski and A. Pelc, Faster deterministic broadcasting in ad hoc radio networks, Proc. 20th Ann. Symp. on Theor. Aspects of Comp. Sci. (STACS'2003), LNCS 2607, 109-120.
[27] D. Kowalski and A. Pelc, Broadcasting in undirected ad hoc radio networks, In Proc. *22nd ACM Symposium on Principles of Distributed Computing*, PODC'03, pp 73-82.
[28] D. Kowalski and A. Pelc. Optimal deterministic broadcasting in known topology radio networks. *Distributed Computing*, 19(3): 185-195 (2007).
[29] D. Liu and M. Prabhakaran, On Randomized Broadcasting and Gossiping in Radio Networks, Proc. *8th Annual International Conference on Computing and Combinatorics*, COCOON'02, pp 340-349.
[30] F. Manne and Q. Xin. Optimal gossiping with unit size messages in known radio networks. *Proc. 3rd Workshop on Combinatorial and Algorithmic Aspects of Networking*, LNCS 4235, pp. 125-134.
[31] F. Manne and Q. Xin. Optimal centralized gossiping in radio networks. Manuscript, 2007.
[32] P.J. Slater, E.J.Cockayne, and S.T. Hedetniemi, Information Dissemination in Trees, *SIAM Journal on Computing*, vol. 10, pp. 892–701, 1981.
[33] A. Pelc. Fault-tolerant broadcasting and gossiping in communication networks. *Networks*, 28(3): 143-156 (1996).
[34] A. Sen and M.L. Huson. A new model for scheduling packet radio networks. *Proc. 15th Joint Conf. of IEEE Computer and Communication Societies*, 1996, pp. 1116-1124.
[35] A.N. Strahler. Hypsometric (area-altitude) analysis of erosional topology. *Bull. Geol. Soc. Amer.* 63, (1952), pp. 117–1142.
[36] X.G. Viennot. A Strahler bijection between Dyck paths and planar trees. *Discrete Mathematics* 246, (2002), pp. 317–329.
[37] Y. Xu, An $O(n^{1.5})$ deterministic gossiping algorithm for radio networks, *Algorithmica*, 36(1):93–96, 2003.

Fredrik Manne obtained his Ph.D from the University of Bergen, Norway. Currently, he is a professor in the Department of Informatics at the University of Bergen. His main interests are in parallel and distributed computing.

Qin Xin graduated with his Ph.D from the University of Liverpool in December, 2004. Currently, he is working as a postdoc at Simula Research Laboratory, Norway. Prior to joining Simula, he spent two years for the postdoc position at University of Bergen, Norway. His main research focus is on algorithms for various problems related to wireless communication networks.