Fault-Tolerant Broadcasting in Radio Networks*

Evangelos Kranakis *‡ Danny Krizanc *‡ Andrzej Pelc †‡

Abstract

We consider broadcasting in radio networks that are subject to permanent node failures of unknown location. Nodes are spread in a region in some regular way. We consider two cases: nodes are either situated at integer points of a line or they are situated on the plane, at grid points of a square or hexagonal mesh. Nodes send messages in synchronous time-slots. Each node v has a given transmission range of the same radius R. All nodes located within this range can receive messages from v. However, a node situated in the range of two or more nodes that send messages simultaneously, cannot receive these messages and hears only noise. Faulty nodes do not receive or send any messages.

We give broadcasting algorithms whose worst-case running time has optimal order of magnitude, and we prove corresponding lower bounds. In case of nonadaptive algorithms this order of magnitude is $\Theta(D+t)$, and for adaptive algorithms it is $\Theta(D+\log(\min(R,t)))$, where t is an upper bound on the number of faults in the network and D is the diameter of the fault-free part of the network that can be reached from the source as a result of those faults.

keywords: broadcasting, fault, radio network.

---

*A preliminary version has appeared in the Proceedings of the 6th Annual European Symposium on Algorithms, ESA'98, Venice, Italy, August 1998, SVLNCS 1461, 283-294. (G. Bilardi, G.F. Italiano, A. Pietracaprina, G. Pucci, Eds.)

***School of Computer Science, Carleton University, Ottawa, Ontario, K1S 5B6, Canada. E-mail:** fkranakis,krizancg@scs.carleton.cag

†**Département d'Informatique, Université du Québec à Hull, Hull, Québec J8X 3X7, Canada. E-mail: Andrzej_Pelc@uqah.uquebec.ca**

‡Research supported in part by NSERC (Natural Sciences and Engineering Research Council of Canada) grant.

1 Introduction

Radio communication networks have recently received growing attention. This is due to the expanding applications of radio communication, such as cellular phones and wireless local area networks. Relatively low cost of infrastrucure and flexibility of radio networks make them an attractive alternative to other types of communication media.

A radio network is a collection of transmitter-receiver devices (referred to as *nodes*), located in a geographical region. Nodes send messages in synchronous time-slots. Each node v has a given transmission range. All nodes located within this range can receive messages from v. However, a node situated in the range of two or more nodes that send messages simultaneously, cannot receive these messages and hears only noise.

One of the fundamental tasks in network communication is *broadcasting*. One node of the network, called the *source*, has a piece of information which has to be transmitted to all other nodes. Remote nodes get the source message via intermediate nodes, in several hops. One of the most important performance parameters of a broadcasting scheme is the total time it uses to inform all nodes of the network.

As the size of radio networks grows, they become increasingly vulnerable to component failures. Some nodes of the network may be faulty. Such nodes do not receive or send any messages. An important feature of a broadcasting algorithm is its capacity to inform all nodes reachable from the source, as fast as possible, in spite of other node failures and without knowing the location of faults.

1.1 Previous work

In most of the research on broadcast in radio networks [1, 3, 4] the network is modeled as an undirected graph in which nodes are adjacent if they are in the range of each other. A lot of effort has been devoted to finding good upper and lower bounds on broadcast time in (fault-free) radio networks represented as arbitrary graphs. In [1] the authors constructed a family of n-node networks of radius 2, for which any broadcast requires time $\Omega(\log^2 n)$, while in [3] it was proved that broadcast can be done in time $O(D+\log^5 n)$ for any n-node network of diameter D. In [11] the authors restricted attention to communication graphs that can arise from actual geometric locations of nodes in the plane. They proved that scheduling optimal broadcasting is NP-hard even when restricted to such graphs and gave an $O(n \log n)$ algorithm to schedule optimal broadcast when nodes are situated on a line. [9, 10] are devoted to the study of asymptotically optimal broadcast and information exchange when nodes are located randomly on a line or on a ring.

Very few results are known about broadcasting in radio networks in the presence of faults – in contrast to a plethora of papers on fault-tolerant communication in wired point-to-point networks (see [8] for a survey of this literature). In [6] the issue of reliable radio broadcasting was considered but only transient faults were dealt with. In [5] the authors studied the efficiency of overcoming noise in fully connected networks under the assumption that every transmitted bit can be flipped with some probability. To our knowledge, broadcasting in multihop radio networks with permanent node failures of unknown location has never been studied.

1.2 The model

The aim of this paper is to give fast broadcasting algorithms in radio networks subject to permanent node failures of unknown location. We consider radio networks whose nodes are spread in a region in some regular way. All nodes have distinct identities. Two scenarios are considered. In the first, nodes are situated at integer points of a line and in the second they are situated at grid points of a square or hexagonal mesh. The latter layout has particularly important practical applications as it forms a basis of cellular phone networks [7]. Every node v of the network has the same *range*. This is a segment of length $2R$ in case of the line, a square of side $2R$ in case of the square mesh and a regular hexagon of side R in case of the hexagonal mesh, in all cases with center at node v. The assumption about receiving capacity of nodes is as described above for general radio networks.

We assume that at most t nodes are faulty. Faults are permanent (i.e., the fault status of a node does not change during broadcast), and faulty nodes do not send or receive any messages. The location of faults is unknown and it is assumed to be worst-case. Moreover, we do not preclude the possibility that faults may disconnect the radio network, in which case the broadcast message can reach only the fault-free connected component of the source node, called the *domain*. For any configuration of faults, consider the graph G whose vertices are fault-free nodes, and adjacent vertices are those in each other's range. The domain is then the connected component of G containing the source and the *diameter* (denoted by D throughout the paper) is the maximum length of all shortest paths between the source and nodes of the domain. As usual in the case of communication in the presence of faults, it is natural to consider two types of broadcasting algorithms.

In *nonadaptive* algorithms, all transmissions have to be scheduled in advance. Thus every node is provided with a table specifying in which time slots it should transmit. If a given node is faulty, it never transmits, and if a fault-free node is scheduled to transmit before it receives the source message, it transmits a default message. In our algorithms the prescribed periodic behavior of a node does not depend on the length of the line or the size of the mesh, or on the configuration of faults, but only on the label of the node. We do not specify the number of transmissions for each node, treating the broadcasting process as repetitive, designed for possibly many source messages. The *time* of broadcasting a message in the presence of a given configuration of at most t faults is defined as the maximum number of time units between the transmission of this message by the source and the reception of it by a node in the domain.

In *adaptive* algorithms, nodes have the ability to schedule future transmissions on the basis of their communication history. In particular, this enables to perform a preprocessing phase during which nodes learn the fault status of some other nodes from obtained messages and even from noise. Also in this case, the behavior of a node in our algorithms depends only on the label of the node. Since in adaptive algorithms nodes can send meaningful messages prior to the transmission from the source, the time of broadcasting a message in the presence of a given configuration of at most t faults is now defined as the maximum number of time units between the first transmission by any node and the reception of the source message by a node in the domain. Our adaptive algorithms consist of a preprocessing phase lasting a prescribed amount of time, and the proper broadcasting phase which is message driven: a node transmits only once, after the reception of the source message. The above definition of broadcasting time accounts for preprocessing as well as for proper broadcasting. If many messages are broadcast by the source, preprocessing can be done only once and the actual time for subsequent messages can be reduced to that of the proper broadcasting phase.

1.3 Our results

For all the above scenarios we give broadcasting algorithms whose worst-case time has optimal order of magnitude, and we prove corresponding lower bounds. In case of nonadaptive algorithms this order of magnitude is $\Theta(D+t)$, and for adaptive algorithms it is $\Theta(D+\log(\min(R,t)))$, where D is the diameter. More precisely, we give algorithms that, for any fault configuration of at most

t faults, yielding diameter D, inform the entire domain in time corresponding to the respective upper bound ($O(D+t)$ for nonadaptive and $O(D+\log(\min(R,t)))$ for adaptive algorithms). Also, for any nonadaptive (resp. adaptive) algorithm we show configurations of at most t faults yielding diameter D, for which this algorithm requires time $\Omega(D+t)$ (resp. $\Omega(D+\log(\min(R,t)))$) to inform the domain. In case of the line we show how the gap between the upper and lower bounds can be further tightened.

The difficulty of designing efficient fault-tolerant algorithms for radio communication lies in the need to overcome the contradictory impact of the power of radio broadcasting and of faults. On the one hand, scheduling simultaneous transmissions from nodes close to each other should be avoided because this will result in noise for many receiving nodes, in case both transmitting nodes are fault free. On the other hand, sparse transmission scheduling (few simultaneous transmissions from neighboring nodes) is dangerous as well: It causes communication delays in case when nodes scheduled to transmit happen to be faulty.

The paper is organized as follows. In section 2 we consider the case of the line and explain several techniques used in the case of the mesh as well. In section 3 we study the case of node layout on a (square or hexagonal) mesh pointing out the additional difficulties, as compared to the previous scenario. In both sections analysis is divided into two parts, corresponding to the nonadaptive and adaptive communication modes. Section 4 contains conclusions and open problems.

2 The line

In this section we consider the case when nodes of the radio network are situated in points $0, 1, ..., n$, on a line. We will use the notions "larger" and "smaller" with respect to this ordering. The range of every node v has the same radius R, i.e., it includes nodes $v, v+1, ..., v+R, v-1, ..., v-R$. We assume $n > R \geq 2$, since the other cases are trivial. Assume that the source is at node 0 and define the ith *segment* to be $\{(i-1)R+1, ..., iR\}$. m denotes the largest node of the domain and hence the diameter D is $\lceil m/R \rceil$. Clearly m and D depend on a particular fault configuration. Notice that in the special case when $t < R$, the domain consists of all fault-free nodes, regardless of the fault configuration. In this special case we will often get tighter results than in general. For simplicity assume that R divides n. (Modifications in the general case are straightforward.)

2.1 The nonadaptive case

Number nodes in the ith segment $a_{i1}, ..., a_{iR}$ in decreasing order. Consider the following nonadaptive broadcasting algorithm.

Algorithm Line-NA

1. The source transmits in time 1.
2. Node a_{ij} transmits in time $i+j+c(R+1)$, for $c = 0, 1, 2, ...$

In Figure 1 an example of this scheme is shown.

Lemma 2.1 *The algorithm Line-NA informs the domain in time at most* $D+t+\lceil t/R \rceil$*.*

Proof: First observe that no pair of nodes at distance at most R transmits in the same time-slot and that if node i transmits in time x then node $i-1$ transmits in time $x+1$. In the absence of faults, R new nodes are informed in every time unit. Every fault can slow information

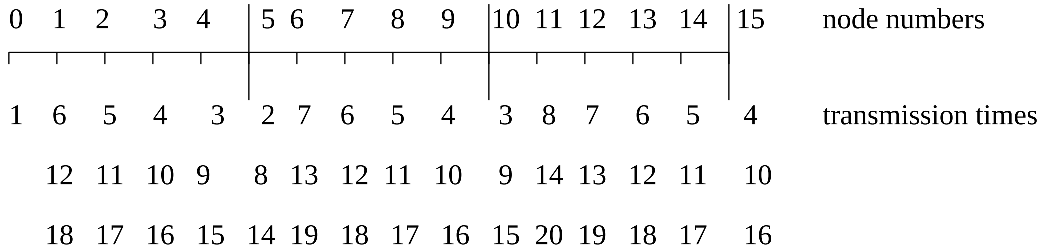

Figure 1: Algorithm Line-NA for $n = 15$ and $R = 5$.

progress in the following two ways: one time unit is lost to wait for $i-1$ (instead of the faulty i) to transmit, and the largest fault-free informed node is by one closer to the source. This implies that the worst-case effect of t faults is the following. Time $D+t$ is spent to inform nodes in the domain with numbers at most $m-t$. The remaining segment of length t is informed in time $\lceil t/R \rceil$, for a total of $D + t + \lceil t/R \rceil$. □

In order to prove the lower bound we need the following lemma.

Lemma 2.2 *Suppose that the source message is known to a set A of nodes, where $|A| \geq t$. Any nonadaptive broadcasting algorithm informing a node v outside of A must use time larger than t.*

Proof: The proof is by induction on t. For $t = 1$ suppose that some node can be informed in one time unit. If a unique node in A transmits in this time unit, the adversary makes it faulty. Otherwise, the adversary makes all nodes fault free. In both cases v cannot hear the message. Suppose that the lemma is true for t. Suppose that a nonadaptive algorithm informs v in at most $t+1$ time units in the presence of up to $t+1$ faults. If in every time unit more than one node transmits, the adversary makes all nodes fault free and no message can be heard. Otherwise, pick any time unit with only one node transmitting and make this node faulty. The rest of the scheme has to inform v in time at most t in the presence of up to t faults, which is impossible by the inductive hypothesis. □

The above lemma gives the following lower bound for $t < R$.

Lemma 2.3 *If $t < R$, every nonadaptive broadcasting algorithm must use at least time $D + t$ to inform the domain in the worst case.*

Proof: Suppose that all segments except the pre-last are entirely fault free. Informing all segments except the last requires time $D - 1$. By Lemma 2.2 informing the last segment requires time at least $t+1$, for a total of at least $D + t$. □

Notice that for $t < R$, the upper bound given by Lemma 2.1 is $D + t + 1$, which means that algorithm Line-NA is almost the best possible in this case.

For larger values of t we get the following asymptotic lower bound.

Lemma 2.4 *Every nonadaptive broadcasting algorithm must use time $D + \Omega(t)$ to inform the domain, in the worst case.*

Proof: Fix a nonadaptive broadcasting algorithm. Let $t = a(R-1)+b$, where $b < R-1$. Let the even-numbered segments be fault free and allocate $R-1$ faults to the first a odd-numbered segments, and b faults to the $(a+1)$th odd-numbered segment. The adversary can distribute faults in these segments to make the algorithm spend at least time R in the first a odd-numbered segments and at least time $b+1$ in the $(a+1)$th odd-numbered segment. (Lemma 2.2). Notice that the presence of a fault-free node in each odd-numbered segment (guaranteed by the bound $R-1$ on the number of allocated faults) implies that the domain consists of all fault-free nodes. The time spent to inform an odd-numbered segment, if the preceding (fault-free) segment is informed, is 1. There are at least $D/2$ odd-numbered segments. Hence the total time used by the algorithm is $D+t$, if $t \leq D(R-1)/2$, and $D + D(R-1)/2$, otherwise. Since $DR = n$ and $R \geq 2$, we have $D(R-1)/2 \geq DR/4 = n/4 \geq t/4$. □

Lemmas 2.1 and 2.4 imply the following result.

Theorem 2.1 *The worst-case minimum time to inform the domain on the line by a nonadaptive broadcasting algorithm is $D + \Theta(t)$. Algorithm Line-NA achieves this performance.*

2.2 The adaptive case

The flexibility given by the adaptive scenario permits to perform preprocessing after which specific fault-free nodes are elected as transmitters of information. This enables to subsequently conduct the broadcasting itself much faster than in the nonadaptive case. We will show how to do preprocessing in time logarithmic in the number of faults, thus considerably increasing the efficiency of the entire process.

We start with the description of a procedure that elects the largest fault-free node in a segment. First suppose that $t < R$ and consider a segment of length R of the line of nodes. Let A be the set consisting of $t+1$ largest nodes in this segment. Hence A contains at least one fault-free node.

The Procedure Binary Elect works as follows. All nodes in A are initialized to the status *active*. ($ACT := A$). After $\lceil \log(t+1) \rceil$ steps, exactly one node remains active (ACT has one element). This is the largest fault-free node in the segment. At any time unit $i \leq \lceil \log(t+1) \rceil$ the set ACT is divided into the larger half L and the smaller half S. All (fault-free) nodes in L transmit a message. If something is heard (a message or noise) then $ACT := L$ and nodes in S never transmit again. Otherwise (i.e., when all nodes in L are faulty), $ACT := S$.

Since all nodes are in the range of one another, this can be done distributively, and every node knows its status at every step and knows whether it should transmit or not. At the end of the procedure, when a single node remains active, this node broadcasts its identity and all other nodes in the segment learn it.

Procedure Binary Elect works in time $O(\log t)$. Clearly, an analogous procedure may be used to find the smallest fault-free node in a segment, and a multiple use of this procedure enables to find k largest fault-free nodes.

Note that in order to avoid conflicts, searching for the largest and for the smallest fault-free nodes in the same segment and searching for such nodes in adjacent segments must be performed at different times. This yields the following adaptive broadcasting algorithm. (We assume that each node sending a message appends its identity to the message, so all nodes in its range know who is the sender.)

Algorithm Line-ADA-1

1. **for all** even i **in parallel do**

   find the largest fault-free node l_i in the ith segment;

find the smallest fault-free node s_i in the ith segment;
for all odd i **in parallel do**
find the largest fault-free node l_i in the ith segment;
find the smallest fault-free node s_i in the ith segment;

2. the source tranmists the message;
if the message arrived from the source in time r
then l_1 transmits in time $r + 1$.
if the message arrived from l_i in time r
then s_{i+1} transmits in time $r + 1$.
if the message arrived from s_i in time r
then l_i transmits in time $r + 1$.

Notice that after step 1, nodes s_i and l_i know each other, and l_i and s_{i+1} know each other, for any i, if they are in the domain. (If the distance between l_i and s_{i+1} exceeded R, they could not be both in the domain because these are consecutive fault-free nodes.) This guarantees that in step 2 exactly one node transmits in any time unit.

Lemma 2.5 *If $t < R$, algorithm Line-ADA-1 informs the domain in time $2D + O(\log t)$.*

Proof: Preprocessing (step 1) takes time $O(\log t)$. In step 2 two time units are spent in every segment. □

Our lower bound (cf. Lemma 2.8) will show that the running time of algorithm Line-ADA-1 is of lowest possible order of magnitude. However, particularly for large D and small t, it may be desirable to speed up the broadcasting phase, possibly at the expense of increasing time spent in preprocessing. The aim would be to avoid spending 2 time units in every segment during broadcasting, and thus to get rid of the factor 2 in the component $2D$. Our next algorithm, based on more extensive preprocessing, achieves this goal, again assuming $t < R$.

Algorithm Line-ADA-2

1. $k := \lceil \sqrt{t/\log t} \rceil$.
for all even i **in parallel do**
for $j := 1$ **to** k **do**
find the jth largest fault-free node l^i_j in the ith segment;
find the smallest fault-free node l^i_{k+1} in the ith segment;
for all odd i **in parallel do**
for $j := 1$ **to** k **do**
find the jth largest fault-free node l^i_j in the ith segment;
find the smallest fault-free node l^i_{k+1} in the ith segment;

2. the source tranmists the message;
if a message arrived from the source in time r
then l_{11} transmits in time $r + 1$.
if a message arrived from node v belonging to the $(i-1)$th segment in time r **then**
if $v + R \geq l^i_{k+1}$ {case 1}
then l^i_p transmits in time $r + 1$,
where p is the smallest index for which $v + R \geq l^i_p$.

else {case 2}
l_1^{i-1} transmits in time $r+1$.
{second transmission from the same segment}

Notice that in case 1 all nodes $l_q^i \leq v+R$ and node l_1^{i-1} know the identity of v and hence know the smallest index p for which $v+R \geq l_p^i$. This guarantees that this l_p^i is the unique node that transmits in time $r+1$ in case 1. In case 2, node l_1^{i-1}, which knows the identity of v and of l_{k+1}^i, can compute that $v+R < l_{k+1}^i$ and thus it has the responsibility to transmit in time $r+1$. This guarantees that a unique node transmits in time $r+1$, in case 2 as well.

Lemma 2.6 *If $t < R$, algorithm Line-ADA-2 informs the domain in time $D + O(\sqrt{t \log t})$.*

Proof: Preprocessing (step 1) takes time $O(k \log t) = O(\sqrt{t \log t})$. In step 2, as long as case 1 occurs, one time unit is spent per segment. Each time when case 2 occurs, an extra time unit is spent in the $(i-1)$the segment. The number of faults encountered between a time unit when a node l_1^j transmits and the time unit when case 2 occurs in some later segment, is at least k. Hence case 2 can occur at most $\lceil t/k \rceil = O(\sqrt{t \log t})$ times. Each time one additional time unit is spent. Hence step 2 takes time $D + O(\sqrt{t \log t})$ and consequently the total time is $D + O(\sqrt{t \log t})$ as well. □

Algorithm Line-ADA-2 is almost twice faster than Line-ADA-1 for small values of range radius. For example, if R is $O(\log n)$, D has order $\Omega(n/\log n)$. Line-ADA-1 runs in time $2D + O(\log \log n)$, while the time of Line-ADA-2 is $D + O(\sqrt{\log n \log \log n})$. However, for large values of t and R, running time of Line-ADA-1 has smaller order of magnitude than Line-ADA-2. As we show below, this order of magnitude $O(D + \log t)$ is always optimal.

The lower bound for the adaptive case is based on the following lemma.

Lemma 2.7 *Suppose that the source message is known to a set A of nodes, where $|A| \geq t$. Any adaptive broadcasting algorithm informing a node v outside of A must use time larger than $\lfloor \log t \rfloor$ in the worst case.*

Proof: Let $k = \lfloor \log t \rfloor$ and consider any broadcasting algorithm working in worst-case time at most k. We describe an adversary strategy preventing message transmission to v. During the execution of the algorithm, the adversary constructs a set B of *black* nodes initialized as empty (in the beginning all nodes are *white*). If in time $i \leq k$ the set of nodes that transmit is A_i and $|A_i \setminus B| \leq \lfloor t/2^i \rfloor$ then all nodes in $A_i \setminus B$ are colored black (added to B). After k steps the adversary declares all black nodes faulty and all white nodes fault free.

It is enough to show that the number of fault-free nodes in every set A_i is different from 1. Let B_i denote the value of B before the ith step. There are two cases. If $|A_i \setminus B_i| \leq \lfloor t/2^i \rfloor$ then all nodes in A_i are faulty. If $|A_i \setminus B_i| > \lfloor t/2^i \rfloor$ then faulty nodes in $A_i \setminus B_i$ are those colored black in steps $i+1, ..., k$. The number of these nodes is at most $\lfloor t/2^{i+1} \rfloor + ... + \lfloor t/2^k \rfloor < \lfloor t/2^i \rfloor$. Consequently, there are at least 2 fault-free nodes in A_i. □

Lemma 2.7 yields the following lower bound in the case when $t < R$.

Lemma 2.8 *If $t < R$, every broadcasting algorithm must use at least time $D - 1 + \lfloor \log t \rfloor$ to inform the domain, in the worst case.*

Proof: Suppose that all faults are situated in the first segment of the domain. Time $\lfloor \log t \rfloor$ is needed to reach the second segment and additional time $D-1$ is needed to reach the last segment. □

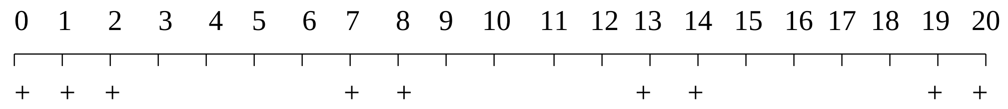

Figure 2: The fault configuration for $R = 5, x = 4$; "+" denotes a fault-free node.

Lemmas 2.5 and 2.8 imply the following theorem.

Theorem 2.2 *If $t < R$, the worst-case minimum time to inform the domain by an adaptive broadcasting algorithm is $\Theta(D + \log t)$. Algorithm Line-ADA-1 achieves this performance.* □

In the case $t \geq R$ Procedure Binary Elect can be easily modified by taking the set A to be the entire segment of length R. A segment in the domain must contain at least one fault-free node and hence Procedure Binary Elect will find the largest such node in time $\lceil \log R \rceil$. Algorithm Line-ADA-1 based on this modified procedure will work in time $2D + O(\log R)$. On the other hand, Lemma 2.7 modified by letting A to be the entire segment of length R yields the lower bound $D - 1 + \lfloor \log R \rfloor$, similarly as in Lemma 2.8. These remarks yield the following generalization of Theorem 2.2.

Theorem 2.3 *The worst-case minimum time to inform the domain on the line by an adaptive broadcasting algorithm is $\Theta(D + \log(min(R, t))$. Algorithm Line-ADA-1 achieves this performance.*
□

As we have seen, for $t < R$, it is possible to lower the summand $2D$ to D, at the expense of increasing the summand $O(\log t)$ to $O(\sqrt{t \log t})$, in the upper bound on broadcasting time. It is natural to ask if this remains possible in the general case. We conclude this section with an example showing that, for large values of t, this is no longer possible.

Example

Consider a fault configuration in which $x < R$ segments are in the domain (e.g., all nodes of the $(x+1)$th segment are faulty), and only two nodes are fault free in any of these segments. Suppose that the only fault-free nodes in the domain are: $iR + i + 1$ and $iR + i + 2$, for $i = 0, 1, ..., x - 1$ (see Figure 2).

Thus $D = x$ and it is easy to see that any broadcasting algorithm informing the entire domain must spent 2 time units in each segment. Thus the time required to inform the domain is $2D$.

3 The mesh

In this section we consider the case when nodes of the radio network are situated in grid points of a mesh. We make the presentation for the square mesh but all results remain valid for other planar subdivisions, e.g., for the hexagonal mesh, and the arguments are easy to modify.

Nodes are situated in points with coordinates $(i, j) : 1 \leq i, j \leq n$. We assume that the source is in point $(1, 1)$ (the general case is similar). For simplicity assume that R divides n and R is even. Every square $\{(i-1)R+1, ..., iR\} \times \{(j-1)R+1, ..., jR\}$, for $1 \leq i, j < n/R$ is called a *cell*. Denote by C_0 the cell containing the source. Partition every cell into 4 squares of side $R/2$, called *tiles*. Two cells (resp. tiles) are called *neighboring* if they touch each other by a side or a corner.

1	2	3	1	2	3	1	2
4	5	6	4	5	6	4	5
7	8	9	7	8	9	7	8
1	2	3	1	2	3	1	2
4	5	6	4	5	6	4	5
7	8	9	7	8	9	7	8
1	2	3	1	2	3	1	2
4	5	6	4	5	6	4	5

Figure 3: Coloring tiles.

Hence every cell (tile) has 8 neighboring cells (tiles). The range of every node is a square of side $2R$ centered at this node. Hence every two nodes in neighboring tiles are in each other's range and only nodes in the same or neighboring cells are in each other's range.

3.1 The nonadaptive case

Color all tiles in 9 colors, assigning the same color to tiles separated by two other tiles vertically, horizontally or diagonally (see Figure 3).

Let $S = R^2/4$ and fix any ordering $a_{ki} : i = 1, ..., S$ of nodes in the kth tile, for all tiles. Let $c(k)$ denote the color number of the kth tile.

Consider the following nonadaptive broadcasting algorithm.

Algorithm Mesh-NA

1. The source transmits in time 0.
2. Node a_{ki} transmits in time $9(i-1) + c(k) + 9Sr$, for $r = 0, 1, 2, ...$

That is, in time units 1,...,9, first nodes of tiles colored (respectively) 1,...,9 transmit, then in time units 10,...,18, second nodes of tiles colored (respectively) 1,...,9 transmit, etc. Notice that nodes transmitting in the same time unit are not in each other's range, and in every tile all nodes are scheduled to transmit successively in time intervals of length 9.

Lemma 3.1 *The algorithm Mesh-NA informs the domain in time $O(D+t)$, for any configuration of at most t faults.*

Proof: Fix a configuration of at most t faults. Define the following graph H on the set of all cells: Two (neighboring) cells C and C' are adjacent if there exist fault-free nodes $v \in C$ and $v' \in C'$ that are in each other's range. Clearly, the domain is included in the union of cells that form the connected component of C_0 (in H). Let L be the maximum length of the shortest path (in H) from C_0 to any vertex in this connected component. Clearly $L \leq D$. Hence it is enough to show that our algorithm informs the domain in time $O(L+t)$.

Suppose that, at some time τ, all fault-free nodes in a given cell C know the source message, and C' is a cell adjacent to C. Let q and q' denote the number of faults in cells C and C', respectively.

Claim. At time $\tau + 18(q + q' + 1)$ all fault-free nodes in C' know the source message.

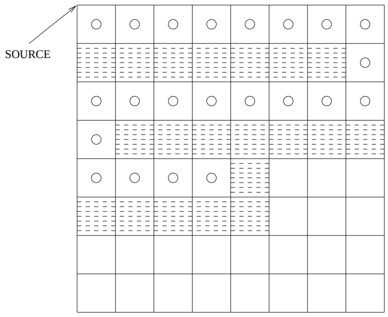

Figure 4: Fault configuration for the lower bound. Shaded cells are entirely faulty. Cells with a circle are in the domain.

First suppose that each of the 8 tiles in cells C and C' contains at least one fault-free node. After at most $9+9q$ time units, a fault-free node v in a tile T in C, touching the cell C', is scheduled to transmit. After at most another period of $9+9q'$ time units, a fault-free node v' in a tile T' in C', neighboring T, is scheduled to transmit. Nodes v and v' are in each other's range, which proves the Claim in this case.

Next suppose that one of the above mentioned 8 tiles consists entirely of faulty nodes. Thus $q+q' \geq S$. Let $w \in C$ and $w' \in C'$ be fault-free nodes in each other's range. After at most $9S$ time units node w is scheduled to transmit and after at most another period of $9S$ time units node w' is scheduled to transmit. This concludes the proof of the Claim.

The Claim shows that the source message uses time at most $18f+18$ to traverse any edge of the graph H, where f is the total number of faults in both cells incident to this edge. Hence 18 time units of delay can be charged to any edge and, additionally, 18 time units of delay can be charged to any fault. Since cells have degree 8, the same fault may be charged at most 8 times. It follows that the domain will be informed in time at most $18L+144t$. □

The following lemma establishes a lower bound matching the performance of algorithms Mesh-NA.

Lemma 3.2 *Every nonadaptive broadcasting algorithm must use time $D+\Omega(t)$ to inform the domain, in the worst case.*

Proof: For any nonadaptive broadcasting algorithm and any t, we have to show a configuration of at most t faults, yielding domain diameter D, for which this algorithm uses time $D+\Omega(t)$ to inform the domain. Fix a nonadaptive broadcasting algorithm and consider two cases.

Case 1. $t < 6R^2$.

We use $x = \min(t, R^2-2)$ faults, all located in cell C_0, as shown in the proof of Lemma 2.2. The domain consists of all fault-free nodes and $D = n/R$. By Lemma 2.2, more than x time units are needed to inform any node outside of C_0, and at least $D-1$ additional time units are needed to inform the entire domain. Hence the time used is $D+\Omega(t)$.

Case 2. $cR^2 \leq t \leq (c+1)R^2$, for $c \geq 6$.

We use approximately $t/3$ faults arranged in full cells to create a snake-shaped domain, as shown in Figure 4.

This yields $D > c/4$. In the domain we leave every second cell entirely fault free and in each of the remaining cells we use $R^2 - 1$ faults as indicated in the proof of Lemma 2.4 (fewer than $t/3$ faults are enough to do that). A similar argument shows that the time used to inform the domain is at least $D + \frac{D}{2}(R^2 - 1) = D + \Omega(t)$.

□

Lemmas 3.1 and 3.2 imply the following result.

Theorem 3.1 *The worst-case minimum time to inform the domain on the mesh by a nonadaptive broadcasting algorithm is $\Theta(D + t)$. Algorithm Mesh-NA achieves this performance.*

3.2 The adaptive case

As in the case of the line, our adaptive algorithm for the mesh consists of a preprocessing part and a proper broadcasting part. The aim of preprocessing is to elect representatives among all fault-free nodes in each cell. These representatives conduct message passing in the second phase. However, unlike in the case of the line, there is no natural choice of nodes in neighboring cells, analogous to the largest and smallest fault-free nodes in neighboring segments, guaranteed to be in each other's range. Hence, instead of Procedure Binary Elect, we use the following procedure independent of the topology of the radio network.

Let A and B be two sets of nodes of a radio network, such that every pair of nodes in A and every pair of nodes in B are in each other's range. We describe the Procedure Elect Couple that finds fault-free nodes $a \in A$ and $b \in B$, such that a and b are in each other's range, if such nodes exist. The procedure works in time $O(\log(|A| + |B|))$. First fix a binary partition of each of the sets A and B: Divide each of the sets into halves (or almost halves, in case of odd size), these halves into halves again, etc., down to singletons. For each member of the partition (except singletons), define the *first half* and the *second half*, in arbitrary order.

The basic step of Procedure Elect Couple (iterated $\lceil \log(\max(|A|, |B|)) \rceil$ times) is the following. Before this step a member set X of the binary partition of A and a member set Y of the binary partition of B are *promoted*. (They contain fault-free nodes in each other's range). This means that all fault-free nodes in A know that nodes in X have the "promoted" status; similarly for Y and B. After this step, one of the halves of X and one of the halves of Y (those containing fault-free nodes in each other's range) are promoted, the other halves are *defeated*. This basic step works in 8 time units as follows. Let X_1, X_2 and Y_1, Y_2 be the first and second halves of X and Y, respectively. We use the phrase "a nodes hears something" in the sense that a node hears either a message or noise.

1. All fault-free nodes in X_1 transmit.
2. All fault-free nodes in Y_1 that heard something in time 1, transmit. (If transmissions occur, Y_1 has been promoted).
3. All fault-free nodes in Y_2 that heard something in time 1 but nothing in time 2, transmit. (If transmissions occur, Y_2 has been promoted).
4. All fault-free nodes in X_1 that heard something in time 2 or 3, transmit. (If transmissions occur, X_1 has been promoted).
5. All fault-free nodes in X_2 that heard nothing in time 1, transmit.

6. All fault-free nodes in Y_1 that heard something in time 5, transmit. (If transmissions occur, Y_1 has been promoted).

7. All fault-free nodes in Y_2 that heard something in time 5 but nothing in time 6, transmit. (If transmissions occur, Y_2 has been promoted).

8. All fault-free nodes in X_2 that heard something in time 6 or 7, transmit. (If transmissions occur, X_2 has been promoted).

The first iteration of the above basic step is performed with $X = A$ and $Y = B$. Then, after $\lceil \log(\max(|A|, |B|)) \rceil$ iterations, exactly two fault-free nodes $a \in A$ and $b \in B$ are finaly promoted, (if fault-free nodes in each other's range existed in sets A and B). They are said to be *elected* and are called *partners*. They are in each other's range and they know each other's identities. All other nodes have "defeated" status, they know it, and never transmit again.

In order to describe our adaptive broadcasting algorithm for the mesh, consider all pairs of neighboring cells. Partition these pairs into groups in such a way that distinct pairs in the same group do not contain neighboring cells. (36 groups are sufficient for the square mesh.) To each pair assign the number (0 to 35) of the group to which this pair belongs.

Now the algorithm can be described as follows.

Algorithm Mesh-ADA

1. Run Procedure Elect Couple for all pairs of neighboring cells, in 36 distinct phases, pairs of cells from the same group in the same phase, in parallel. (If $t < R^2$, start the procedure with subsets of cells of size $t+1$, instead of entire cells.) Partners elected for a pair of neighboring cells get the number 0 to 35 corresponding to this pair. (The same node may get more than one number.)

2. The source transmits the message.
   After an elected node heard the source message for the first time, it transmits it at the first time unit $u \equiv i \bmod 36$, where i is a number assigned to this node, and stops.

Lemma 3.3 *Algorithm Mesh-ADA inform the domain in time $O(D + \log(\min(R, t))$, for any configuration of at most t faults.*

Proof: As in the nonadaptive case, define the following graph H on the set of all cells: Two (neighboring) cells C and C' are adjacent if there exist fault-free nodes $v \in C$ and $v' \in C'$ that are in each other's range. In the preprocessing part of the algorithm (step 1) partners are elected for each pair of adjacent cells. This step takes time $O(\log(\min(R, t))$. Consider two adjacent cells C and C' and suppose that at time τ all fault-free nodes in C know the message. After at most 36 time units, the partner in C, elected for the pair (C, C') transmits the message (if it had not done it before). After another period of 36 time units, its partner in C' transmits and hence, in time at most $\tau + 72$, all fault-free nodes in C' know the source message. Hence traversing every edge of the graph H takes at most 72 time units. This proves that step 2 of the algorithm takes time $O(D)$. □

The matching lower bound $D + \Omega(\log(\min(R, t)))$ can be derived from Lemma 2.7 similarly as in the case of the line. Hence we get

Theorem 3.2 *The worst-case minimum time to inform the domain on the mesh by an adaptive broadcasting algorithm is $\Theta(D + \log(\min(R, t))$. Algorithm Mesh-ADA achieves this performance.*

4 Conclusion

We presented asymptotically optimal fault-tolerant broadcasting algorithms for radio networks whose nodes are regularly situated on the line or on the mesh. The natural open problem is to generalize these results to the case of arbitrary graphs of reachability, as considered, e.g., in [1] and [3]. The lower bound from [1] shows that we cannot expect time $O(D + \log t)$ or $O(D + t)$ for arbitrary graphs. However, results from [1] and [3] leave a very small gap in the fault-free case. It would be interesting to get similarly close bounds for nonadaptive and adaptive algorithms in the presence of faults.

References

[1] N. Alon, A. Bar-Noy, N. Linial and D. Peleg, A Lower Bound for Radio Broadcast, Journal of Computer and System Sciences 43 (1991), 290-298.

[2] R. Bar-Yehuda, A. Israeli, and A. Itai, Multiple Communication in Multi-Hop Radio Networks, Proc. 8th ACM Symposium on Principles of Distributed Computing (1989), 329 - 338.

[3] I. Gaber and Y. Mansour, Broadcast in Radio Networks, Proc. 6th Ann. ACM-SIAM Symp. on Discrete Algorithms, SODA'95, 577-585.

[4] E. Kushilevitz and Y. Mansour, An $\Omega(D \log n)$ Lower Bound for Broadcast in Radio Networks, Proc. 12th Ann. ACM Symp. on Principles of Distributed Computing (1993), 65-73.

[5] E. Kushilevitz and Y. Mansour, Computation in Noisy Radio Networks, Proc. 9th Ann. ACM-SIAM Symp. on Discrete Algorithms, SODA'98, 236-243.

[6] E. Pagani and G. P. Rossi, Reliable Broadcast in Mobile Multihop Packet Networks, Proc. 3rd Annual ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM'97), (1997), 34 - 42.

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

[8] A. Pelc, Fault-Tolerant Broadcasting and Gossiping in Communication Networks, Networks 28 (1996), 143-156.

[9] K. Ravishankar, and S. Singh, Broadcasting on $[0, L]$, Discrete Applied Mathematics 53 (1994), 299 - 319.

[10] K. Ravishankar and S. Singh, Asymptotically Optimal Gossiping in Radio Networks, Discrete Applied Mathematics 61 (1995), 61 - 82.

[11] A. Sen and M. L. Huson, A New Model for Scheduling Packet Radio Networks, Proc. 15th Annual Joint Conference of the IEEE Computer and Communication Societies (IEEE INFOCOM'96) (1996), 1116 - 1124.

1. **Running Head:** Fault-Tolerant Broadcasting in Radio Networks
2. **Contact Author:**

   - Evangelos Kranakis
   - School of Computer Science, Carleton University, Ottawa, Ontario, K1S 5B6, Canada
   - kranakis@scs.carleton.ca
   - (613) 520 4330
   - (613) 520 4334 (fax)

   - I can submit final manuscript in LaTEX.