Reliable Broadcast in Radio Networks: The Bounded Collision Case

Chiu-Yuen Koo
Dept. of Computer Science
University of Maryland
cykoo@cs.umd.edu

Vartika Bhandari*
Dept. of Computer Science, and
Coordinated Science Laboratory
University of Illinois at Urbana-Champaign
vbhandar@uiuc.edu

Jonathan Katz†
Dept. of Computer Science
University of Maryland
jkatz@cs.umd.edu

Nitin H. Vaidya
Dept. of Electrical and Computer Eng., and
Coordinated Science Laboratory
University of Illinois at Urbana-Champaign
nhv@uiuc.edu

ABSTRACT

We study the problem of achieving global broadcast in a radio network where a node can multicast messages to all of its *neighbors* (that is, nodes within some given distance r), and up to t nodes in any single neighborhood may be corrupted. Previous work assumes that corrupted nodes can neither cause collisions nor spoof addresses of honest nodes. In this work, we eliminate these assumptions and allow each faulty node to cause a (known) bounded number of collisions and spoof the addresses of arbitrary other nodes. We show that the maximum tolerable t in this case is identical to the maximum tolerable t when collisions and address spoofing are not allowed. Thus, by causing collisions and spoofing addresses an adversary may be able to degrade the efficiency of achieving broadcast, but it cannot affect the feasibility of this task.

Categories and Subject Descriptors

C.2.4 [**Computer-Communication Networks**]: Distributed Systems; C.4 [**Performance of Systems**]: Fault Tolerance

General Terms

Algorithms, Reliability

Keywords

Byzantine failure, Broadcast, Fault tolerance, Radio networks

1. INTRODUCTION

Advancements in wireless technology make possible the deployment of large-scale networks in which the sole means of communication is via wireless (radio) transmission. Since reliable broadcast can serve as a building block for many applications in these environments, it is of interest to establish the conditions under which it can be achieved. Classical results for the feasibility of broadcast/Byzantine agreement (e.g., [11]) do not immediately apply in the context of radio networks because the communication model is different: Traditional work assumes the existence of point-to-point channels between each pair of nodes in the network, whereas in radio networks a more reasonable model is one in which a message transmitted by a node u is received by all nodes within some distance r of u. (We refer to a node within distance r of u as a *neighbor* of u). This has two consequences: (1) some nodes (specifically, those at distance greater than r from each other) cannot communicate directly, and (2) each message is actually sent on a "local broadcast" channel of sorts. Note that it is not *a priori* clear whether this represents a weaker or a stronger communication model than the classical one.

Achieving broadcast in radio networks has been studied in both fault-free and faulty settings. (See Section 1.2 for a detailed discussion.) Previous work in the Byzantine setting assumes, as in standard treatments of Byzantine faults, that faulty nodes can send arbitrary messages to other nodes (subject to the limitations of the communication model). However, previous work assumed that faulty nodes could not cause *collisions* in messages sent by honest nodes. A collision at node u occurs when two neighbors v_1, v_2 of u

*This work was supported by a Vodafone Graduate Fellowship.

†This work was supported by NSF Trusted Computing grants #0310499 and #0310751, NSF CAREER award #0447075, and US-Israel Binational Science Foundation grant #2004240.

PODC'06, July 22-26, 2006, Denver, Colorado, USA.

transmit at the same time. In this case, there is no guarantee as to what message(s) u will receive. So if v_1 is honest but v_2 is faulty, a collision caused by v_2 interferes with messages transmitted by v_1. Note that collision is a receiver-side phenomenon, and thus the same transmission (by v_1, say) may be successfully received by some neighbors of the sender, but may not be received correctly by other neighbors due to collision. This will lead to inconsistent views among non-faulty nodes even if the sender of the message is honest. While the absence of collisions is a reasonable assumption in wired, point-to-point networks, it is not a valid assumption in radio networks where such collisions represent a real concern.

Additionally, previous work assumed that faulty nodes could not "spoof" the address of another node, where "spoofing" means to send a message and claim that this message is being sent by someone else. Again, the assumption that nodes cannot spoof the addresses of other nodes may be reasonable in wired networks, or networks where some prior infrastructure (such as a PKI or shared symmetric keys) exists, but is no longer valid when all messages are sent and received on the same channel and spoofing becomes easy.

1.1 This Work

This work shows how the assumptions mentioned above can be avoided. That is, we show broadcast protocols that are resilient to (a certain fraction of) Byzantine failures even if faulty nodes can cause collisions or spoof addresses of other nodes. Clearly, however, it is impossible to provide full guarantees if the actions of faulty nodes are unlimited in all respects; for example, a faulty node might continually cause collisions for all messages received by a particular (honest) node, thereby preventing this honest node from obtaining any useful information from the protocol! On the other hand, since transmitting data requires power and this is a bounded resource (this is especially true for radio networks, which are often composed of small, battery-operated nodes), it is reasonable to assume a known upper bound on the number of messages that faulty nodes can send. As a consequence, this gives a bound on the number of collisions any faulty node can cause as well as a bound on the number of spoofing attempts by any faulty node.

As in previous work, we will assume that there are at most t faulty nodes in any single neighborhood (as in [2], [12], the neighborhood of a node u, referred to here, comprises all nodes within distance r of u including u itself, and there can not be more than t faulty nodes in this set, for any node u). Our main result is to show that the same threshold t of faulty nodes can be tolerated in our setting, where collisions and address spoofing are allowed, as in previous work (which did not allow for collisions or address spoofing). Thus, allowing faulty nodes the extra capabilities considered here does not affect the *feasibility* of obtaining broadcast (though, as we will see, it may well affect the *efficiency* of obtaining broadcast).

1.2 Related Work

Reliable broadcast in radio networks has been considered in much previous work [9, 8, 2, 13, 14]. In [9], the focus was on obtaining time-efficient broadcast algorithms in finite networks comprising nodes located in a regular grid pattern, with crash-stop failures. Byzantine faults were addressed by Koo [8]. As we have mentioned, Koo assumes that no neighborhood has more than t faults, and that the adversary cannot cause collisions nor carry out address spoofing. Feasibility results and impossibility results were shown for a network of nodes located on an infinite grid and having transmission radius r. In particular, it was shown that reliable broadcast is not achievable for $t \geq \frac{1}{2}r(2r+1)$ in the L_∞ norm. These results were improved in [2, 14]. The work of Bhandari and Vaidya [2] established the exact threshold for the L_∞ norm by presenting a protocol that achieves broadcast for any $t < \frac{1}{2}r(2r+1)$. That work also provides approximate bounds for the L_2 metric; these bounds indicate that the threshold fraction of faulty nodes must lie in a similar range as in the L_∞ norm. Bhandari and Vaidya also quantified the per-neighborhood fault threshold for the case of crash-stop failures.

Subsequent work by these authors [3] showed a sufficient condition for reliable broadcast in general graphs, and a simpler protocol for a grid network (as compared to the protocol given in [2]) was presented.

Reliable broadcast in an arbitrary graph was also considered in [12]. Upper and lower bounds for feasibility of reliable broadcast were presented based on graph-theoretic parameters. However, no exact thresholds were established. It was also shown that there exist certain graphs in which algorithms that have knowledge of the topology succeed, while those lacking this knowledge fail.

Random transient failures were considered in [13]. Here, each node fails at each step with some constant probability p. Tight bounds on p for which broadcast can be achieved were obtained. Random permanent failures in a grid network were considered in [4], and necessary and sufficient conditions on the required transmission range as a function of the probability of node failure were derived.

The problem of achieving consensus in a wireless network was studied in [5, 7] in a slightly different model. We refer the reader there for details.

Finally, we mention the work of Considine, et al. [6] which considers the problem of realizing broadcast in a model where "multicast" channels of bounded size are available to the parties (in addition to pairwise channels). Roughly speaking, their model assumes the existence of a multicast channel among every subset of players of some size b; they show that in this case a fraction of malicious nodes greater than the standard $n/3$ [11] can be tolerated. Our work differs from their in that we do not assume a "multicast" channel among every set of players of size b. Instead a message sent by a node is received only by players physically located in some neighborhood of radius r. Nevertheless, both our work and theirs are motivated by the consideration of alternate communication topologies.

2. DEFINITIONS AND NOTATION

We consider the same *radio network* model as in earlier work [8, 2, 14]. Nodes are located at the integer grid points of an infinite grid (i.e., each grid unit is a 1×1 square). A node is uniquely identified by its grid location $(x, y) \in \mathbb{Z}^2$.

Assuming absence of collisions, if a node performs a *local broadcast* of a message m then all nodes within distance r (in the appropriate metric) will receive the message. This distance r is known as the *transmission radius*. The set of nodes within this radius is termed as the neighborhood of (x, y) and is denoted as $\mathsf{nbd}(x, y)$. Other nodes in $\mathsf{nbd}(x, y)$ are known as the *neighbors* of (x, y). We denote by $\mathsf{nbd}_2(x, y)$

the set of nodes that are at most two hops away from (x, y); i.e., $\mathsf{nbd}_2(x, y) \stackrel{\text{def}}{=} \cup_{(x',y')\in\mathsf{nbd}(x,y)}\mathsf{nbd}(x', y')$. We also let $\tilde{r} = \lfloor r \rfloor$. We primarily present results in the L_∞ metric, where the distance between points (x_1, y_1) and (x_2, y_2) is given by $\max\{|x_1 - x_2|, |y_1 - y_2|\}$. However, our algorithms are also applicable in the L_2 (also known as the "Euclidean") metric, where the distance between points as before is given by $\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$. This issue is briefly discussed in Section 6.

We consider the locally-bounded adversarial model [8] where no single neighborhood contains more than t faults. As in [8, 2, 14], we assume there is a pre-determined TDMA schedule such that if all nodes follow the schedule then no collision will occur. Unlike previous work, in this paper a faulty node may cause *message collision* and/or *spoof* the identity of another node for some bounded number of times (on the other hand, a non-faulty node always follows the schedule). Let n_c and n_s be the corresponding bounds on the number of message collisions and instances of address spoofing, respectively, that a faulty node can perform. Both n_c and n_s are assumed to be known in advance by all nodes.

When two nodes i and j perform a local broadcast at the same time, a message collision occurs at the nodes in $\mathsf{nbd}(i) \cap \mathsf{nbd}(j)$ and those nodes do not receive the messages broadcast by either i or j. If nodes are equipped with *collision detectors*, then nodes in $\mathsf{nbd}(i) \cap \mathsf{nbd}(j)$ detect that a message collision has occurred and can substitute default messages instead. In the absence of a collision detector, there is no guarantee on what messages are received by nodes in $\mathsf{nbd}(i) \cap \mathsf{nbd}(j)$, and our protocols will be proven resilient under a worst-case assumption regarding what is received. We note that most prior work (e.g., [13]) implicitly assumes the former scenario when collisions occur.

A faulty node i is able to spoof a non-faulty node j when it is the turn of j to broadcast a message (according to the underlying TDMA schedule) but j has no messages to send (according to the prescribed protocol). In this scenario, i can impersonate j by broadcasting a message m with the sender identity falsely set to j. If this happens, then nodes in $\mathsf{nbd}(i) \cap \mathsf{nbd}(j)$ receive m and treat m as originating from j. We remark that address spoofing can be reduced to message collision by having node j always broadcast something (e.g., a fixed dummy message) when it is his turn instead of remaining silent. However, this approach is communication inefficient, and our solution takes a different approach.

We now formally state the requirements of the broadcast problem. There is a distinguished node s known as the *source* that holds an initial input $\mathcal{M}$. A protocol is said to achieve *broadcast* if the following conditions hold:

1. Completeness: every non-faulty node i eventually outputs a value v_i.
2. Agreement: for any two non-faulty nodes i and j, $v_i = v_j$.
3. Correctness: if s is non-faulty, then $v_i = \mathcal{M}$ for any non-faulty node i.

When referring to a particular broadcast instance, the source of the broadcast is assumed to have coordinates $(0, 0)$ without loss of generality, and other nodes are identified by their coordinates relative to the source. Note that even if the source is faulty, there will be a single message $\mathcal{M}$ that is locally broadcast to the neighborhood of s as the first step of the protocol. Therefore, we will take this as $\mathcal{M}$ when s is faulty.

3. OUR RESULTS

We state and prove the following:

THEOREM 1. *In the L_∞ metric, if $t < \frac{1}{2}\tilde{r}(2\tilde{r} + 1)$ then there exists a protocol that achieves broadcast as long as there is a bound on the number of collisions caused and spoofed messages sent by each faulty node.*

We present a constructive proof by showing a protocol in Section 5 that achieves broadcast under the stated conditions.

The above bound is tight since it is shown in [8] that:

THEOREM 2. *[8, Theorem 9] In the L_∞ metric, if $t \geq \frac{1}{2}\tilde{r}(2\tilde{r}+1)$, broadcast is impossible even if the adversary cannot cause collisions nor carry out address spoofing.*

The techniques we describe in this paper are also applicable to the L_2 metric, up to the bound for tolerable number of faults per neighborhood (as established in [2]). Hence we obtain that:

THEOREM 3. *In the L_2 metric, if $t < 0.23\pi\tilde{r}^2$ then there exists a protocol that achieves broadcast as long as there is a bound on the number of collisions caused and spoofed messages sent by each faulty node.*

Recall that the $t < 0.23\pi\tilde{r}^2$ bound from [2] is approximate, but is increasingly accurate for large r.

4. PRELIMINARIES

Our solution uses some known results in the literature, summarized in Sections 4.1 and 4.2.

4.1 Broadcast Without Collisions or Address Spoofing

We review the broadcast algorithm $\mathcal{B}_{\texttt{no collision}}$ described in [3] that achieves broadcast up to the maximum tolerable value of t, assuming no of collisions or address spoofing. As mentioned in Section 1.2, the protocol presented in [3] is simpler than the protocol described in [2] (which can also tolerate the optimal fault threshold).

1. **(Broadcast in $\mathsf{nbd}(s)$):** The source s performs a local broadcast of the message $\mathcal{M}$. Each neighbor i of s outputs the first value it receives from s and then performs a one-time local broadcast of COMMITTED$(i, \mathcal{M})$.
2. **(Broadcast in the rest of the network):** Every node j (including the source and the neighbors of the source) follows the procedure below:
   - On receipt of a COMMITTED(i, v) message from neighbor i, record the message and broadcast a HEARD(j, i, v) message.
   - On receipt of a HEARD(i, k, v) message from neighbor i, record the message (but do not propagate it further).
   - *(Deciding on output):* All j that are not neighbors of s continually check the following: if there exists

a node q, a value v, and j has recorded $t+1$ messages $m_1, m_2, \ldots, m_{t+1}$ such that (i) for $1 \leq i \leq t+1$, message m_i is of the form COMMITTED(a_i, v) or HEARD(a_i, a_i', v); and (ii) $a_1, \ldots, a_1', \ldots$ are all distinct neighbors of q, then j outputs the value v.

THEOREM 4. *[3, Theorem 2] Assuming no collisions and no address spoofing, $\mathcal{B}_{\text{no collision}}$ achieves broadcast in the L_∞ metric whenever $t < \frac{1}{2}\tilde{r}(2\tilde{r}+1)$.*

The following is implicit in the proof of [3, Theorem 2]:

CLAIM 1. *Assuming the L_∞ metric, no collisions, no address spoofing, and $t < \frac{1}{2}\tilde{r}(2\tilde{r}+1)$, then there exists a constant T (dependent on $t, \tilde{r}$) such that if the nodes start executing $\mathcal{B}_{\text{no collision}}$ at time 0, all non-faulty nodes in $\mathsf{nbd}_2(s)$ output $\mathcal{M}$ by time T.*

In other words, even if nodes execute $\mathcal{B}_{\text{no collision}}$ only for a period of time T, all non-faulty nodes in $\mathsf{nbd}_2(s)$ will still output $\mathcal{M}$.

In fact, the following stronger version of Theorem 4 and Claim 1 holds (and this is again implicit in the proof of [3, Theorem 2]):

CLAIM 2. *Theorem 4 and Claim 1 remain true even if the following holds: when a faulty node $i \neq s$ performs a local broadcast, the neighbors of i can receive* different *messages, subject to the choice of the adversary.*

4.2 Broadcast in Point-to-Point Networks

In a fully connected point-to-point network, there is an authenticated channel connecting each pair of nodes. Address spoofing and collisions are not possible; however, an adversary can observe the messages sent between non-faulty nodes. An execution of a *synchronous* protocol takes place in a sequence of rounds. In each round, nodes send messages to each other depending on the messages they have received in the previous rounds. An adversary is said to be *rushing* if it can see the messages sent to faulty nodes in the current round before it decides the outgoing messages of faulty nodes for that round. Let n be the total number of nodes. The following result is well-known (see, e.g. [11, 1]):

THEOREM 5. *There exists a fixed-round, synchronous protocol $\mathcal{B}_{p2p}$ that achieves broadcast in a fully connected point-to-point network in the presence of a rushing adversary corrupting $f < \frac{1}{3}n$ nodes. Moreover, $\mathcal{B}_{p2p}$ has the following property: within the same round, a non-faulty node always sends the same message to all other nodes.*

5. OUR PROPOSED ALGORITHM

Following Claim 2, broadcast can be achieved if we obtain a protocol based on $\mathcal{B}_{\text{no collision}}$ such that:

1. **(Broadcast in $\mathsf{nbd}(s)$):** In step 1, neighbors of the source agree on a common message (as the message originated at the source) before they propagate it to other nodes in the network.

2. **(Broadcast in the rest of the network):** In step 2, whenever a non-faulty node performs a local broadcast of message m, its neighbor nodes are able to receive m correctly despite the possible occurrence of collisions caused by faulty nodes in the vicinity. If address spoofing is possible, a node will accept a message m from a neighbor only if it is convinced that m originated from that neighbor.

Condition (1), above, is required to handle situations where a faulty source can collude with other faulty nodes, and use collisions to send conflicting values to different neighbors (such a scenario was discussed in [8]). To this effect, we develop an agreement protocol among nodes in $\mathsf{nbd}(s)$. We use a primitive called *weak broadcast* as a building block in this agreement protocol. Weak broadcast is defines as follows:

DEFINITION 1. *We say that a node i performs a* **weak broadcast** *of a message m to a set of nodes $\mathcal{S}$ within time T if the following conditions hold:*

1. *A non-faulty node $j \in \mathcal{S}$ outputs m_j within time T.*

2. *If i is non-faulty, then $m_j = m$ for all non-faulty nodes $j \in \mathcal{S}$.*

Note that if i is faulty, then two non-faulty nodes may output two different messages.

In Section 5.2, we show how to construct protocols for weak broadcast and, subsequently, broadcast. But first, in section 5.1, we will show how to achieve agreement among the neighbors of the source, assuming each node in $\mathsf{nbd}(s)$ is capable of performing a weak broadcast to all other nodes in $\mathsf{nbd}(s)$ within time T.

5.1 Agreement among Neighbors of the Source

We transform the broadcast protocol $\mathcal{B}_{p2p}$ (cf. Theorem 5) for $n = |\mathsf{nbd}(s)|$ nodes and working in the point-to-point model to a broadcast protocol $\mathcal{B}_{\mathsf{nbd}(s)\_p2p}$ for the set $\mathsf{nbd}(s)$ and working in the radio network model.

$\mathcal{B}_{\mathsf{nbd}(s)\_p2p}$ simulates $\mathcal{B}_{p2p}$ round by round. Suppose in a given round of $\mathcal{B}_{p2p}$, node i is instructed to send the message m_i to other nodes. To simulate one round of execution in $\mathcal{B}_{p2p}$, the nodes run the following subroutine sequentially:

> for each node $i \in \mathsf{nbd}(s)$, node i does a weak broadcast of the message m_i to all nodes in $\mathsf{nbd}(s)$.

Note that the weak broadcast may be viewed as establishing a *virtual* point-to-point link between pairs of nodes in $\mathsf{nbd}(s)$. Thus, it is ensured that if i is non-faulty, all other nodes receive the same value from i. If i is faulty, receipt of conflicting values is acceptable, as i is capable of sending different values to different nodes in the point-to-point model (cf. Claim 2).

Finally, node i outputs whatever it is directed to output by $\mathcal{B}_{p2p}$. We note that if the round complexity of $\mathcal{B}_{p2p}$ is R, then $\mathcal{B}_{\mathsf{nbd}(s)\_p2p}$ takes time $RT|\mathsf{nbd}(s)|$.

CLAIM 3. *If $t < \frac{1}{2}\tilde{r}(2\tilde{r}+1)$, then $\mathcal{B}_{\mathsf{nbd}(s)\_p2p}$ ensures that all neighbors of the source output the same message m'. If the source is non-faulty, then $m' = \mathcal{M}$.*

PROOF. $n = |\mathsf{nbd}(s)| = (2\tilde{r}+1)(2\tilde{r}+1)$. If $t < \frac{1}{2}\tilde{r}(2\tilde{r}+1)$, then $t < \frac{1}{4}|\mathsf{nbd}(s)|$. The claim then follows from the fact that $\mathcal{B}_{p2p}$ can tolerate a rushing adversary corrupting fewer than $\frac{1}{3}n$ nodes. □

5.2 Weak Broadcast and Reliable Broadcast

Depending on different assumptions (i.e., whether faulty nodes are allowed to spoof the identity of other nodes, whether honest nodes are equipped with collision detectors, etc.), we show how to obtain a weak broadcast protocol and then a protocol for reliable broadcast in the entire network. The most general case is that faulty nodes are allowed to do address spoofing and nodes are not equipped with collision detectors. However, we provide constructions for other cases to serve as a warmup.

5.2.1 No Address Spoofing; Collision Detectors

In this subsection, we assume $n_s = 0$. A non-faulty node may fail to receive a message from another non-faulty node due to message collision; however, this can happen at most tn_c number of times. We observe that if an honest node i repeats the local broadcast of message m for a total of tn_c+1 times, then any neighbor of i will receive at least one copy of m successfully.

Based on the broadcast algorithm $\mathcal{B}_{\texttt{no collision}}$, we construct an algorithm $\mathcal{B}_{repeat}$ where a node i will execute the same instructions as in $\mathcal{B}_{\texttt{no collision}}$ except that:

- If i is instructed to perform a local broadcast of message m in $\mathcal{B}_{\texttt{no collision}}$, then i performs a local broadcast of message m $tn_c + 1$ times.
- If i is instructed to carry out an action after receipt of a message m from j in $\mathcal{B}_{\texttt{no collision}}$, then i carries out the corresponding action only when it receives m from j for the first time.

For the sake of completeness, we include the protocol description for $\mathcal{B}_{repeat}$:

1. The source s' performs $tn_c + 1$ local broadcasts of the message $\mathcal{M}$. Each neighbor i of s' outputs the first value v it heard from s'.
2. When it outputs a value v, each neighbor of s' sends $tn_c + 1$ local broadcasts of COMMITTED(i, v).
3. Every node j (including the source and the neighbors of the source) follows the procedure below:
   - On receipt of a COMMITTED(i, v) message from a neighbor i for the first time, record the message and perform $tn_c + 1$ local broadcasts of the message HEARD(j, i, v).
   - On receipt of a HEARD(i, k, v) message from a neighbor i for the first time, record the message (but do not re-propagate).
   - Output the value v and perform $tn_c + 1$ local broadcasts of COMMITTED(j, v) if: not already committed to a value, and there exists a node q and $t + 1$ recorded messages $m_1, m_2, \ldots, m_{t+1}$ such that (1) either m_i = COMMITTED(a_i, v) or m_i=HEARD$(a_i, a_{i'}, v)$ (for all i), and (2) $a_i, a_{i'}$ are all distinct neighbors of q.

The source s' mentioned above is the source in the protocol $\mathcal{B}_{repeat}$ (used as a building block), and is not to be confused with the source of the overall broadcast. As can be seen, $\mathcal{B}_{repeat}$ primarily differs from $\mathcal{B}_{\texttt{no collision}}$ in that messages are repeated sufficiently-many times so that they will eventually be received even if there are collisions. We have:

CLAIM 4. *Assume the L_∞ metric, and $t < \frac{1}{2}\tilde{r}(2\tilde{r}+1)$. Then there exists a constant T (depending on r) such that the following holds: If non-faulty node i is the source of a weak broadcast and all nodes execute $\mathcal{B}_{repeat}$ for time T, then all non-faulty nodes in $\mathsf{nbd}_2(i)$ will output m.*

PROOF. This follows from Claim 2. □

5.2.1.1 Achieving Weak Broadcast.

Note that in $\mathcal{B}_{repeat}$, if the node i is faulty then a non-faulty node may not output a value. However, it is easy to construct a weak broadcast protocol.

CLAIM 5. *Assume the L_∞ metric, and $t < \frac{1}{2}\tilde{r}(2\tilde{r}+1)$. Then for any node s there exists a protocol $\mathcal{B}_{\texttt{weak broadcast}}$ that allows a node $i \in \mathsf{nbd}(s)$ (which can be potentially faulty) to perform a weak broadcast to $\mathsf{nbd}(s)$.*

PROOF. In $\mathcal{B}_{\texttt{weak broadcast}}$ (with i as source of the weak broadcast), nodes execute $\mathcal{B}_{repeat}$ for a period of time T. After time T, if a node has not yet been able to output a value (which means that i is faulty), then a node outputs a default value. Also, note that $\mathsf{nbd}(s) \subseteq \bigcup_{i \in \mathsf{nbd}(s)} \mathsf{nbd}_2(i)$. The rest of the claim follows from Claim 4. In fact, it can be observed that even if only nodes in $\mathsf{nbd}_2(s)$ participate in B_{repeat}, this suffices for our purposes, as we only require weak broadcast amongst nodes in $\mathsf{nbd}(s)$.

We further illustrate the weak broadcast process in Figures 1 and 2. Figure 1 (reproduced from [3]) illustrates how the broadcast propagates in the absence of collisions. In particular it shows how — given that $\mathsf{nbd}(a, b)$ has output the correct broadcast value — nodes in

$$\mathsf{nbd}(a-1, b) \cup \mathsf{nbd}(a+1, b) \cup \mathsf{nbd}(a, b-1) \cup \mathsf{nbd}(a, b+1)$$

are also able to do so, and this step only requires participation of nodes within $\mathsf{nbd}_2(a, b)$. Figure 2 illustrates how a suitably modified form of $\mathcal{B}_{\texttt{no collision}}$ (that is, $\mathcal{B}_{repeat}$) ensures weak broadcast by a non-faulty node in $nbd(s)$ to all other nodes in $\mathsf{nbd}(s)$. Consider the extreme case in which the source of the weak broadcast is node i located at $(-\tilde{r}, -\tilde{r})$, and envision step-by-step propagation of node i's broadcast value amongst $\mathsf{nbd}(s)$. Nodes in $\mathsf{nbd}(i)$ can simply output the first value they hear from i. Thereafter, if $\mathcal{B}_{repeat}$ is run by all nodes in $\mathsf{nbd}_2(s)$, then by a similar inductive argument as that used for $\mathcal{B}_{\texttt{no collision}}$ in [3], one can see that the correct value will propagate first to the region A (shaded in horizontal lines), and then to region B (shaded in vertical lines). It thus follows that on using $\mathcal{B}_{\texttt{weak broadcast}}$ (which is basically $\mathcal{B}_{repeat}$ along with the default rule), all nodes in $\mathsf{nbd}(s)$ (including the extremal node j at $(\tilde{r}, \tilde{r})$) will output the correct value within a finite time period T. □

5.2.1.2 Achieving Reliable Broadcast.

Following Claim 5, every node in $nbd(s)$ can perform a weak broadcast to $\mathsf{nbd}(s)$. Thus, we have the primitive required to run protocol $\mathcal{B}_{\mathsf{nbd}(s)\_p2p}$. We can now obtain a broadcast protocol resilient to a bounded number of collisions. This protocol $\mathcal{B}_{reliable\ broadcast}$ is a modified version of $\mathcal{B}_{repeat}$, where the first step of $\mathcal{B}_{repeat}$ is changed to the following:

> The nodes execute the protocol $\mathcal{B}_{\mathsf{nbd}(s)\_p2p}$ with the source using the message $\mathcal{M}$ as the input. Let

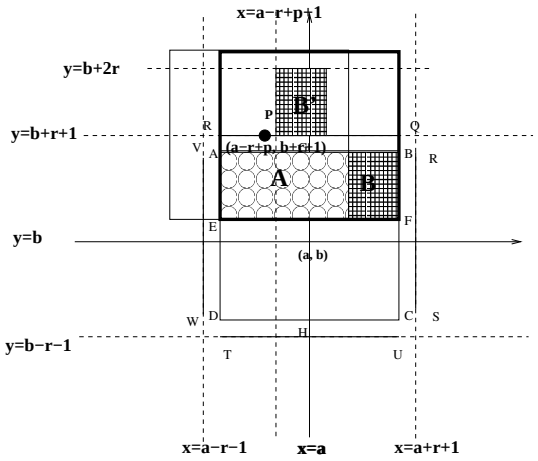

Figure 1: Propagation in the absence of collisions/spoofing (taken from [3]).

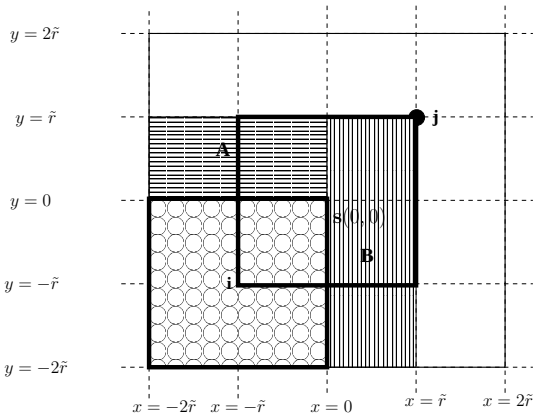

Figure 2: Using $\mathcal{B}_{repeat}$ to achieve weak broadcast in $\mathsf{nbd}(s)$.

m_i be the output of node i in $\mathcal{B}_{\mathsf{nbd}(s)\_p2p}$. Each neighbor i of s outputs m_i.

With this change, we obtain a protocol achieving broadcast.

The above protocol can also be used in the absence of a collision detector, and in the presence of address spoofing, after minor modifications. We discuss various such scenarios in the subsequent sections.

5.2.2 No Address Spoofing; No Collision Detectors

The construction is similar to the previous subsection except that in the transformation of $\mathcal{B}_{\texttt{no collision}}$ into $\mathcal{B}_{repeat}$:

- If node i is instructed to perform a local broadcast of message m in $\mathcal{B}_{\texttt{no collision}}$, then in $\mathcal{B}_{repeat}$ node i performs $2tn_c + 1$ local broadcasts of message m.
- If node i is instructed to carry out an action after receipt of a message m from j in $\mathcal{B}_{\texttt{no collision}}$, then in $\mathcal{B}_{repeat}$ node i carries out the corresponding action only when it receives $tn_c + 1$ copies of m from j.

Note that if a non-faulty node i performs a local broadcast of message m for a total of $2tn_c + 1$ times, then a neighbor of i will receive at least $tn_c + 1$ legitimate copies of m. On the other hand, if a node i receives $tn_c + 1$ copies of m from j, then i can conclude that m has not been corrupted due to message collisions.

5.2.3 Address Spoofing; Collision Detectors

In the transformation of $\mathcal{B}_{\texttt{no collision}}$ into $\mathcal{B}_{repeat}$, we now do the following:

- If node i is instructed to performs a local broadcast of message m in $\mathcal{B}_{\texttt{no collision}}$, then in $\mathcal{B}_{repeat}$ node i performs $t(n_c+n_s)+1$ local broadcasts of message m.
- If node i is instructed to carry out an action after receipt of a message m from j in $\mathcal{B}_{\texttt{no collision}}$, then in $\mathcal{B}_{repeat}$ node i carries out the corresponding action only when it receives $tn_s + 1$ copies of m from j.

If a non-faulty node i performs a local broadcast of message m for a total of $t(n_c+n_s)+1$ times, then a neighbor of i will receive at least $tn_s + 1$ legitimate copies of m. On the other hand, if a node i receives $tn_s + 1$ copies of m claimed to be originated from j, then i can conclude that m indeed originated from j.

5.2.4 Address Spoofing; No Collision Detectors

In the transformation of $\mathcal{B}_{\texttt{no collision}}$ into $\mathcal{B}_{repeat}$, we now do the following:

- If node i is instructed to perform a local broadcast of message m in $\mathcal{B}_{\texttt{no collision}}$, then in $\mathcal{B}_{repeat}$ node i performs $t(2n_c+n_s)+1$ local broadcasts of message m.
- If node i is instructed to carry out an action after receipt of a message m from j in $\mathcal{B}_{\texttt{no collision}}$, then in $\mathcal{B}_{repeat}$ node i carries out the corresponding action only when it receives $t(n_c+n_s)+1$ copies of m from j.

If a non-faulty node i performs a local broadcast of message m for a total of $t(2n_c + n_s) + 1$ times, then a neighbor of i will receive at least $t(n_c+n_s)+1$ legitimate copies of m. On the other hand, if a node i receives $t(n_c + n_s) + 1$ copies of m claimed to be originated from j, then i can conclude that m indeed originated from j.

6. DISCUSSION AND FUTURE WORK

The techniques described in this paper are quite generic. We believe that, in general, these techniques can be used to transform any broadcast protocol designed under the assumptions that collisions and address spoofing do *not* occur to one that is resilient even when collisions and address spoofing *do* occur (and achieving the same tolerable threshold t). However, we were unable to formally prove a general result of this sort.

In our protocols, the communication overhead per non-faulty node grows as $\Omega(t(n_c + n_s))$. In fact, the overhead at non-faulty nodes $i \notin \mathsf{nbd}_2(s)$ gets multiplied by a factor of $\Theta(t(n_c + n_s))$. Though the overhead at non-faulty nodes in $\mathsf{nbd}_2(s)$ is greater due to the need to run an agreement protocol, the fact that the network is infinite ensures that the *average* overhead per node grows by a factor of only $\Theta(t(n_c + n_s))$. It is also to be noted that if the adversary performs the maximum number of disruptive actions permitted, the average cost of causing disruptions is $\Theta(n_c+n_s)$

per faulty node. Thus, in our algorithms, non-faulty nodes are required to send more messages than faulty nodes are assumed able to send! Independent of whether this represents a reasonable state of affairs or not, it certainly allows an adversary to mount a denial of service attack that drains the power of non-faulty nodes and reduces their operational lifetime relative to faulty nodes. Ideally, one would desire a protocol that achieves broadcast with the same energy-drain at non-faulty nodes as at faulty nodes, so that non-faulty nodes expend more energy only when the adversary expends more energy. Our algorithms do not achieve this since they are *proactive*, requiring players to repeatedly send messages sufficiently-many times to overcome any collisions (or instances of address spoofing) that may occur.

It would be of interest to determine whether a *reactive* protocol might perform better in the above regard. If collision detectors are available, n_c, n_s are known, and t is small but not known *a priori*, it may be possible to devise a reactive algorithm of this sort. We briefly mention one key idea on which such an approach may potentially be based. Nodes that detect a collision transmit a *collision alarm* that is propagated for two hops. Nodes should not act on a received message immediately. Instead they should maintain tentative state, and wait for a pre-specified number of rounds. If, during this period, they receive a *collision alarm* then they should flush the tentative state. Receipt of an alarm is also indication to the sender that the message needs to be retransmitted. Since a resource-constrained adversary cannot cause an unbounded number of collisions or false alarms, an algorithm proceeding along these lines should eventually succeed in achieving broadcast.

The above assumes that the values of n_c and n_s are known *a priori*. It is also relevant to consider the possibility of achieving broadcast even if these values are not known in advance.

7. CONCLUSION

In this paper we have demonstrated that the maximum number of tolerable faults per-neighborhood in a grid radio network with bounded collisions and address spoofing is the same as the maximum number of tolerable faults when collisions and address spoofing are assumed not to occur. We have presented broadcast protocols that achieve broadcast up to the maximum tolerable fault threshold. Further work is needed in order to obtain more efficient protocols, or to rule out any such improvements by proving appropriate lower bounds.

8. REFERENCES

[1] P. Berman and J. A. Garay. Asymptotically optimal distributed consensus (extended abstract). ICALP '89.

[2] V. Bhandari and N. H. Vaidya. On reliable broadcast in a radio network. PODC 2005.

[3] V. Bhandari and N. H. Vaidya. On reliable broadcast in a radio network: A simplified characterization. Technical Report, CSL, UIUC, May 2005. Available at http://www.crhc.uiuc.edu/wireless/papers/bcast-addendum.pdf

[4] V. Bhandari and N. H. Vaidya. Reliable broadcast in a wireless grid network with probabilistic failures. Technical Report, CSL, UIUC, Oct. 2005 (A revised version is under preparation).

[5] G. Chockler, M. Demirbas, S. Gilbert, C. Newport, and T. Nolte. Consensus and collision detectors in wireless ad hoc networks. PODC 2005.

[6] J. Considine, M. Fitzi, M. Franklin, L.A. Levin, U. Maurer, and D. Metcalf. Byzantine agreement given partial broadcast. *J. Cryptology* 18(3): 191–217, 2005.

[7] S. Gilbert, R. Guerraoui, and C. Newport. Of malicious motes and suspicious sensors. Technical Report MIT-CSAIL-TR-2006-026, CSAIL, Massachusetts Institute of Technology, April 2006.

[8] C.-Y. Koo. Broadcast in radio networks tolerating Byzantine adversarial behavior. PODC 2004.

[9] E. Kranakis, D. Krizanc, and A. Pelc. Fault-tolerant broadcasting in radio networks. *J. Algorithms* 39(1): 47–67, 2001.

[10] E. Kreyszig. *Advanced Engineering Mathematics*, 7th edition. John Wiley & Sons, 1993.

[11] M. Pease, R. Shostak, and L. Lamport. Reaching agreement in the presence of faults. *J. ACM* 27(2): 228–234, 1980.

[12] A. Pelc and D. Peleg. Broadcasting with locally bounded Byzantine faults. *Information Processing Letters* 93(3): 109–115, 2005.

[13] A. Pelc and D. Peleg. Feasibility and complexity of broadcasting with random transmission failures. PODC 2005.

[14] V. Vaikuntanathan. Brief announcement: broadcast in radio networks in the presence of Byzantine adversaries. PODC 2005.