Broadcast in Radio Networks in the presence of Byzantine Adversaries

Vinod Vaikuntanathan [*]

Abstract

In PODC '04, Koo [2] presented a protocol that achieves broadcast in a radio network tolerating (roughly) $\frac{1}{2}r^2$ Byzantine faults (where r is the transmission range in the radio network). We prove that the simple protocol of [2] indeed tolerates $\frac{1}{\sqrt{2}}r^2$ faults. We also consider a generalization of the model of [2] to account for missing nodes in the network, and provide a fairly general sufficient condition for broadcast.

1 Introduction

Simulating a broadcast channel among n nodes using point-to-point channels between every pair of them, tolerating malicious behavior of t nodes, is the classical setting of Byzantine Agreement (BA) [3]. It is known that, in this setting, broadcast is achievable *iff* $t < n/3$. Some of the subsequent work on this problem addressed the situation where the point-to-point channels are replaced with k-cast channels for some constant k (such as [4, 1]).

Koo [2] considered the broadcast problem in a radio network. In a radio network, each node can broadcast its message to all the nodes within a distance r (in an appropriate metric). At the end of the broadcast protocol, all the nodes in the network should have accepted m. The problem is thus to simulate global broadcast using local broadcast channels of a specific form.

Koo [2] proved that, in the ℓ_∞ metric,

- If $t < \frac{1}{2}r(r + \sqrt{r/2} + 1)$, there is an (explicit) protocol that achieves broadcast.
- If $t \geq \frac{1}{2}r(2r + 1)$, there is no protocol that achieves broadcast.

Note that the total number of neighbours of any node in the ℓ_∞ metric is $(2r+1)^2 - 1$. Thus, the above result says that, if for each node, at most a $\frac{1}{8}$ fraction of its neighbors are corrupted, then broadcast is achievable, and broadcast is not possible if more than $\frac{1}{4}$ fraction of them are corrupted.

We asymptotically improve the upper bound of [2] and prove the following.

Theorem 1. *If $t < (\frac{1}{\sqrt{2}} - \epsilon)r^2$ (for some constant $\epsilon > 0$), there is a protocol that achieves broadcast in the ℓ_∞ metric.*

Koo [2] considers the setting in which the radio network is modeled as an infinite grid and all the grid points are occupied by nodes. A natural question to ask is whether broadcast is possible even if some of the grid points are unoccupied. To this end, we consider the problem of achieving broadcast in an arbitrary graph (a multicast topology), where each node of the graph has a multicast channel to its neighbors. We obtain a sufficient condition on the structure of the multicast graph such that broadcast is achievable.

[*] E-mail: vinodv@mit.edu, MIT CSAIL, 32 Vassar Street, Cambridge, Massachusetts 02139 USA

2 Previous Work and Our Result

The model of broadcast in radio networks we work with was proposed in [2]. In this model, each integral point (x, y) of an infinite square grid (of side-length 1) represents a radio node $p(x, y)$. Each node can multicast a message to all the nodes situated within a distance of r from (x, y) (in the appropriate metric space). The message delivery is synchronous and there exists a pre-determined schedule for the nodes to send messages so that no two neighbours any node will send messages at the same time. Some set of nodes could be corrupted by an all-powerful adversary. A t-adversary is one that corrupts not more than t neighbours of any honest node. A corrupted node could act arbitrarily maliciously except that it is constrained to send messages according to the pre-fixed schedule.

The "dealer" D (which is w.l.o.g, $p(0, 0)$) multicasts a message m to its neighbours in the beginning. Broadcast is achieved if every honest node eventually receives and accepts m.

The broadcast protocol of [2] is simple: the dealer multicasts the message m to its neighbours. If a node $p(x, y)$ gets message m directly from the dealer (i.e, $p(x, y)$ is within a distance of r from the dealer), $p(x, y)$ accepts m and multicasts m to its neighbors. If a node $p(x, y)$ is not a neighbour of the dealer, then it waits till it receives the same message from more than $2t + 1$ of its neighbours, accepts m and multicasts m to its neighbours. This protocol achieves broadcast as long as at most $t < \frac{1}{2}r(r + \sqrt{r/2} + 1)$ neighbors of each node are faulty.

2.1 Proof of Theorem 1

We prove that the protocol of [2] achieves the desired fault-tolerance. Our analysis improves upon [2] by a more refined counting of the number of neighbours of a node $p(x, y)$ that have already accepted m.

Denote by $P[a \dots b, c \dots d]$ the set of all (nodes corresponding to the) points (x, y) such that $a \leq x \leq c$ and $b \leq y \leq d$. Denote by N, the number of neighbors of any node in the ℓ_∞ norm. Note that $N = 4r^2 + O(r)$. Assume (w.l.o.g) that the dealer D is at point $(0, 0)$. We prove the statement by induction on the ℓ_∞ distance n of the point (x, y) from $(0, 0)$.

Basis case: $n \leq r$. All the nodes in $P[-r \dots r, -r \dots r]$ receive and accept m. This is because all these nodes are within one-hop distance of D.

Induction Hypothesis: Assume that all nodes in $S \overset{def}{=} P[-n \dots n, -n \dots n]$ accept m. We will prove that all nodes in $P[-(n+1) \dots (n+1), -(n+1) \dots (n+1)]$ accept m eventually. We divide the nodes into sets $S_0, S_1, \dots S_k$ (for some k to be determined later). S_0 is a stack of nodes (a "triangle") with the length of the base $\beta_0 r$ (for some β_0 to be determined later), whereas $S_1, S_2, \dots, S_k$ are the "concentric regions" surrounding S_0 (Refer to figure 1). Think of S_i as the set of nodes that accept m after receiving messages from the nodes in $S \cup \bigcup_{j=0}^{i-1} S_i$. Lemma 1 shows that all nodes in S_0 accept m. Lemma 2 shows that if all nodes in $S_0 \cup S_1 \cup \dots S_i$ accept m, then all the nodes in S_{i+1} accept m. The theorem follows from these Lemmas and an appropriate setting of the parameters. (See discussion after the lemmas). If a protocol works against an adversary that corrupts an α' fraction of any node's neighbors, then it is said to have a fault-tolerance of α'.

Lemma 1. *All nodes in the region S_0 accept m. Moreover, if $\alpha' = \frac{\alpha}{8}$ is the fault-tolerance, then $\beta_0 \geq 2 - \alpha - O(\frac{1}{r})$ and $|S_0| \geq \frac{\beta_0^2}{4} r^2 - O(r)$.*

Proof. Note that a node v accepts message m after receiving m from at least $2\alpha' N + 1 = \alpha r^2 + O(r)$ neighboring nodes. The proof now follows from the following two observations:

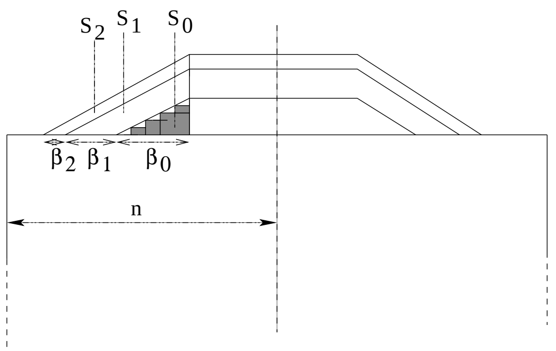

Figure 1: The sets $S_0, S_1, \ldots$ used in the proof of Theorem 1

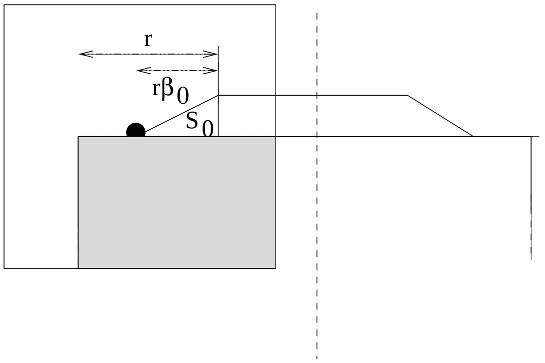

Figure 2: This construction is used to calculate β_0 in Lemma 1

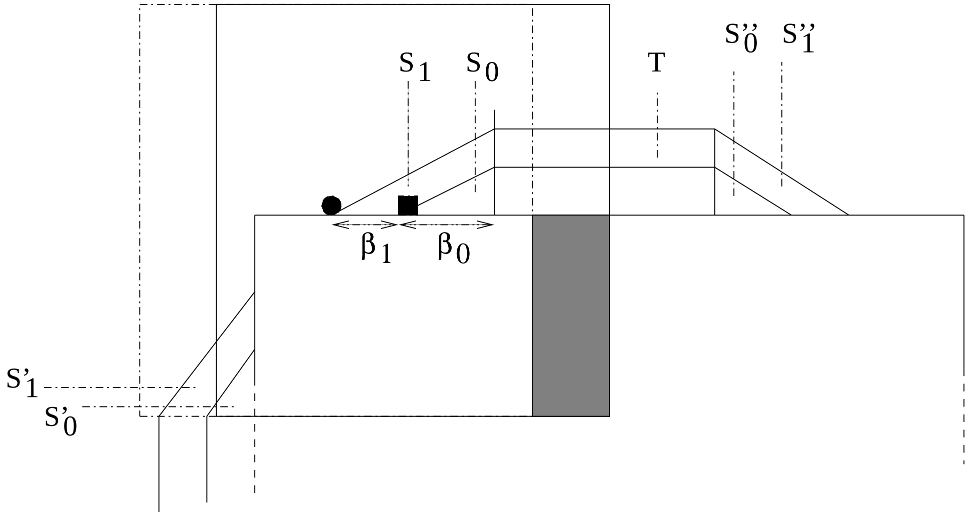

Figure 3: The grey square illustrates the neighbors of the black square in S that are not the neighbors of the black dot.

- Consider the node $P[n+1, n+1-r(1-\beta_0)] \in S_0$ (marked by a black dot in Figure 2). This node has at least $(2-\beta_0)r^2$ neighbors in S. Setting $(2-\beta_0)r^2 \geq \alpha r^2 + O(r)$, we get $\beta_0 \geq 2 - \alpha - O(\frac{1}{r})$.

- If a node (x, y) has N' neighbors in S, then a node $(x-2, y+1)$ has $N' - O(\frac{1}{r})$ neighbors in S. From this, we know that the height of the triangle is (at least) half its base (minus a constant).

Now, we count the number of nodes in S_0. Since we know that the base is $\beta_0 r$ and the height is at least $\frac{\beta_0}{2}r$ (half the base), the number of nodes is $\frac{\beta_0^2}{4}r^2 - O(r)$. □

Lemma 2. *If all nodes in $S, S_0, \ldots, S_i$ $(i \geq 0)$ accept m, then all the nodes in the S_{i+1} accept m too. Moreover, $\beta_{i+1} \geq \beta_1(\beta_0)^i$ and $|S_{i+1}| \geq \frac{1}{2}\beta_1(\beta_0)^{i+1}r^2$.*

Proof. This is proven by induction on i. From Lemma 1 and symmetry considerations, we know that all nodes in $S \cup S_0 \cup S'_0$ accept m. Now, we prove the basis case $(i = 0)$. In Figure 3, note that the black dot (the extremal node in S_1) has $\beta_1 r^2$ less neighbors in S compared to the black square (the extremal node in S_0). But, this is compensated by the nodes in $S_0 \cup S'_0$, which have already accepted m. Thus, $\beta_1 r^2 \geq 2 \times \frac{\beta_0^2}{4}r^2 - O(r)$. Thus, $\beta_1 \geq \frac{\beta_0^2}{2} - O(\frac{1}{r})$.

For induction, observe the following:

- In Figure 3, the node $v \in S_{i+1}$ (the black dot) has $\beta_{i+1}r \times r$ less neighbors in $S \cup \bigcup_{j=0}^{i-1}$ compared to node $u \in S_i$ (the cross in Figure 3). But, by induction hypothesis, we know that all the nodes in S_i have accepted m, and we can use these nodes now. These are $|S_i| \geq \beta_1 \frac{1}{2}(\beta_0)^i r^2$ nodes. Moreover, observe that we can use the nodes in the set S'_i which is symmetric to S_i along the perpendicular edge of the $n \times n$ square. (See Figure 3) and from $T \cup S''_i$ (Figure 3). These balance out when $\beta_{i+1}r^2 = 2 \times \frac{1}{2}\beta_1(\beta_0)^i r^2$. Thus, $\beta_{i+1} \geq \beta_1(\beta_0)^i$.

- From the figure, the area of S_{i+1} is the area of $\bigcup_{j=0}^{i+1} S_i$ minus the area of $\bigcup_{j=0}^{i} S_i$. $|S_{i+1}| = \frac{((\sum_{j=0}^{i}\beta_j)+\beta_{i+1})^2 r^2}{4} - \frac{(\sum_{j=0}^{i}\beta_j)^2 r^2}{4} \geq \frac{1}{4} \cdot 2\beta_{i+1}(\sum_{j=0}^{i}\beta_j)r^2 \geq \frac{1}{2}\beta_{i+1}\beta_0 r^2 \geq \frac{1}{2}\beta_1\beta_0^{i+1}r^2$. □

Now, we manage to cover all the nodes in $P[n+1, 0 \ldots n+1]$ if the sum of the bases of all the regions S_i become 1 after some point. i.e, if $\sum_{j=0}^{\infty}\beta_j > 1$. Expanding the β_j's, we get $\beta_0 + \beta_1 + \beta_1\beta_0 + \beta_1(\beta_0)^2 + \ldots = \beta_0 + \frac{\beta_1}{1-\beta_0} > 1$. We know that $\beta_0 = 2 - \alpha - O(1/r)$ (from Lemma 1) and $\beta_1 = \frac{\beta_0^2}{2}$ (from the basis case of Lemma 2). Solving the resulting quadratic gives us $\beta_0 > 2 - \sqrt{2}$ and thus, $\alpha < \sqrt{2}$. Thus, the total number of faults that we can tolerate is $\alpha' r^2 - O(r) = \frac{\alpha}{2}r^2 - O(r) = (\frac{1}{\sqrt{2}} - O(\frac{1}{r}))r^2$.

Author's Note 1: Note that we can tolerate twice this many faults, if the faults were fail-stop. i.e, $t_{\text{fail-stop}} = (\sqrt{2} - O(\frac{1}{r}))r^2$. Also, using cryptographic techniques (specifically, digital signatures), we can tolerate $(\sqrt{2} - O(\frac{1}{r}))r^2$ Byzantine faults.
Author's Note 2: We can improve the analysis of Theorem 1 to get a fault-tolerance of $(\frac{3}{4} - O(\frac{1}{r})r^2$ in the Byzantine setting. We believe that this is the maximum number of faults that can be tolerated using the simple protocol of [2].
We can extend Theorem 1 to the ℓ_2 metric, using standard techniques, giving us the following:

Corollary 1. *If $t < (\frac{1}{2\sqrt{2}} - \epsilon)r^2$ (for some constant $\epsilon > 0$), there is a protocol that achieves broadcast in the ℓ_2 metric.*

3 Simulating Global Broadcast using Local Broadcast Channels

In this section, we look at a more general form of the broadcast problem we considered in the previous section. The model of [2] (and the one we dealt with) required that all the grid points of the radio network be occupied by live radio nodes at all times. It is a natural (and very practical) question to determine the conditions under which broadcast is possible in sparse topologies of radio nodes. We take a first step in this direction, by proving a sufficient condition on the topology (which is modeled as a multicast graph) so that broadcast is possible.

More precisely, the radio network if modeled as a multicast graph $G = (V, E)$. Denote by $N(v)$ the set of all neighbours of node v in G. The node $p(v)$ corresponding to each vertex $v \in V$ has the capability to *multicast* messages to all the nodes in the set $\{p(w) : w \in N(v)\}$. An active α-adversary is one that can corrupt nodes subject to the condition that for any node u, at most an α fraction of the neighbours of u are corrupted. Any node v can initiate the broadcast of a message m by multicasting m to all its neighbours. We say that broadcast is achieved if all the honest nodes receive and accept m eventually. The goal, then, is to achieve broadcast in the presence of an α-adversary. It is easily seen that the broadcast problem in radio networks we considered in the previous section can be cast in this framework.

Below, we provide a sufficient condition on the multicast graph G so that broadcast can be achieved in the presence of an α-adversary. A directed orientation Δ of a graph G is an assignment of a direction to all the edges of the graph. The directed graph so formed is denoted $\Delta(G)$. Let $N^-_\Delta(v)$ denote the in-degree of node v in $\Delta(G)$. Below, we give a recursive definition for what it means for a graph to be orientable.

Definition 1. *A graph $G = (V, E)$ is said to be (β, S)-*orientable *for a set $S \subseteq V$, if either $S = V$ or there exists a directed orientation Δ such that*

- *There exists a node $u \notin S$, such that $|N^-_\Delta(u) \cap S| \geq \beta|N(u)|$, and*
- *G is $(\beta, S \cup \{u\})$-orientable.*

G is said to be β-orientable, if for every $v \in V$, G is $(\beta, N(v) \cup \{v\}$-orientable.

Theorem 2. *If the multicast graph is 2α-orientable, then there exists a protocol that achieves broadcast against an α-adversary.*

Proof. The simple protocol of [2] achieves broadcast in this setting. The proof is fairly easy to see, and is omitted due to lack of space. It would be interesting to come up with more natural characterizations under which broadcast is achievable. □

References

[1] L. A. Levin J. Considine and D. Metcalf. Byzantine agreement with faulty majority using bounded broadcast. In *arXiv.org e-Print archive, 2003.*

[2] Chiu-Yuen Koo. Broadcast in radio networks tolerating byzantine adversarial behavior. In *PODC '04: Proceedings of the twenty-third annual ACM symposium on Principles of distributed computing*, pages 275–282. ACM Press, 2004.

[3] L. Lamport M. Pease, R. Shostak. Reaching agreement in the presence of faults. In *Journal of the ACM (JACM), v.27 n.2*, pages 228–234, 1980.

[4] Ueli Maurer Mattias Fitzi. From partial consistency to global broadcast. In *Proceedings of the thirty-second annual ACM symposium on Theory of computing*, pages 494–503, 2000.