Multicast Capacity Through Perfect Domination

Michael Dinitz  
Johns Hopkins University

Naomi Ephraim*  
Cornell University

February 15, 2017

Abstract

The capacity of wireless networks is a classic and important topic of study. Informally, the capacity of a network is simply the total amount of information which it can transfer. In the context of models of wireless radio networks, this has usually meant the total number of point-to-point messages which can be sent or received in one time step. This definition has seen intensive study in recent years, particularly with respect to more accurate models of radio networks such as the SINR model. This paper is motivated by an obvious fact: radio antennae are (at least traditionally) omnidirectional, and hence point-to-point connections are not necessarily the best definition of capacity. To fix this, we introduce a new definition of capacity as the maximum number of messages which can be received in one round, and show that this is related to a new optimization problem we call the Maximum Perfect Dominated Set (MaxPDS) problem. Using this relationship we give tight upper and lower bounds for approximating the capacity. We also analyze this notion of capacity under game-theoretic constraints, giving tight bounds on both the Price of Anarchy and the Price of Stability.

*This work was partially done while at Johns Hopkins University.
1 Introduction

A fundamental quantity of a wireless network is its capacity, which informally is just the maximum amount of data which it can transfer. There is a large literature on analyzing and computing the capacity of wireless networks under various modeling assumptions, including models of how interference works and assumptions on how nodes are distributed in space. The last decade has witnessed a flurry of activity in this area, particularly for worst-case (rather than random) node distributions, motivated by the ability to apply ideas from multiple areas of theoretical computer science (approximation algorithms and algorithmic game theory in particular) to these problems.

We continue that line of work in this paper, but with a new, and arguably more natural, definition of capacity. Much of the research in the last decade (see, e.g., [7, 8, 1, 11, 9, 12], has used a point-to-point definition of capacity: given a collection \((s_i, t_i)\) pairs of nodes, and some model of interference, the capacity is the maximum number of pairs which can simultaneously successfully transmit a message. This is sometimes motivated by its utility in scheduling: if we are trying to support many unicast demands in a wireless network, a natural thing to do is make as much progress as possible in each time step, i.e., maximize the number of successful transmissions. For this reason, the problem of computing the maximum capacity is also sometimes called the One-Shot Scheduling problem [1, 8].

But while well-motivated by scheduling, this is not obviously the right definition of capacity. For example, suppose we are in a classical radio network where we are given a communication graph and interference is destructive: \(u\) will receive a message from \(v\) if \(v\) sends a message, \(u\) does not send a message, and no other neighbor of \(u\) sends a message. Suppose that we are given a star topology with \(r\) as the center and leaves \(x_1, \ldots, x_n\), and that there is a demand from \(r\) to each leaf. What is the capacity of this network? Traditionally, the answer would be 1: only one of the unicast links can be successful, since \(r\) can only send one message at a time. On the other hand, if \(r\) really only has a single message which it is trying to send to all of its neighbors, then all of these demands can be satisfied in a single round, so the capacity is \(n\).

In other words, recently popular notions of capacity do not take into account the ability to multicast or broadcast, since they assume that there is a different message for each unicast demand. But one of the defining features of traditional wireless networks is that antennas are omnidirectional; this is one of the main differences between wireless and wireline networks. So if our goal is to understand the capacity of a network, it might be reasonable to measure this as the total number of messages which can be successfully received in a single time step, without taking demands into account at all. After all, this would be the true limit on the single-step “usefulness” of the network.

In this paper we study this notion of capacity in radio networks. We first show that it is equivalent to a new optimization problem we call the Maximum Perfect Dominated Set (MaxPDS) problem, and then using this connection we give tight upper and lower bounds on its approximability. We also study it in a distributed context by following the lead of previous work on distributed network capacity [1, 2] and looking at a natural game-theoretic formulation in which each transmitter acts a a self-interested agent, and proving bounds on the price of anarchy and the price of stability.

1.1 Modeling and MaxPDS

We consider the classical radio network model. In this model there is a communication graph \(G = (V, E)\), and each node in \(V\) can act as either a transmitter or a receiver. In a given unit of
time (we make the standard assumption of synchronous rounds), each node can either broadcast a message to all of its neighbors, or choose to not broadcast and thus act as a receiver. Interference is modeled by requiring a single message arriving at each receiver, or else the messages interfere and cannot be decoded. In other words, a vertex $i$ can successfully decode a message from a neighbor $j$ if and only if $i$ is not broadcasting (and so is acting as a receiver), $j$ is broadcasting, and no other neighbor of $i$ is broadcasting. If multiple neighbors if $i$ are broadcasting then their messages all interfere with each other at $i$, and so $i$ would not receive any message.

In this model, the equivalent of the unicast notion of “capacity” used in recent work would be a maximum matching (or maybe a maximum matching subject to being a subset of some set of demands), which can clearly be computed in polynomial time. But this may be significantly smaller than the number of nodes which successfully hear a message, as the star example shows. So we will instead adopt a multicast notion of capacity:

**Definition 1.** The multicast capacity of a wireless network $G = (V, E)$ is the maximum number of nodes which can simultaneously receive a message.

It is straightforward to relate this to reasonably well-studied notions in graph theory. In particular, since each node successfully receives a message if and only if it does not broadcast and exactly one of its neighbors does broadcast, the multicast capacity is quite similar to the definition of a perfect dominating set [15, 16, 13].

**Definition 2.** Given a graph $G = (V, E)$, a set $S \subseteq V$ is a perfect dominating set if every vertex in $V \setminus S$ is adjacent to exactly one vertex in $S$.

We say that a node is perfectly dominated by $S$ if exactly one neighbor is in $S$. Note that a perfect dominating set always exists since we can set $S = V$, in which case trivially every node of $V \setminus S = \emptyset$ is perfectly dominated by $S$ (a node is perfectly dominated by $S$ if exactly one neighbor is in $S$). This motivated [15, 16] and others to study the Minimum Perfect Dominating Set problem, in which the goal is to find a perfect dominating set of minimum size (analogous to the Minimum Dominating Set problem, in which we only require domination rather than perfect domination). But note that if $S$ is the set of nodes transmitting, the set of non-transmitters which are perfectly dominated is exactly the set of nodes who successfully receive a message. Thus the multicast capacity of a network is equal to the maximum number of nodes which can be perfectly dominated at the same time. Hence computing the multicast capacity is equivalent to solving the following problem:

**Definition 3.** Given a graph $G = (V, E)$, the Maximum Perfect Dominated Set Problem (MaxPDS) is to find a set $S \subseteq V$ which maximizes the number of perfectly dominated vertices.

Note that the solution to this may not be a perfect dominating set. Rather, it may dominate some vertices multiple times and may not dominate some at all in order to perfectly dominate the maximum number of vertices. This problem has not been considered or defined before (to the best of our knowledge).

### 1.2 Our Results and Outline

Our first results are matching upper and lower bounds for MaxPDS, i.e., for the problem of maximizing multicast capacity. The precise lower bound we obtain depends on the hardness assumption that we use, but all are essentially polylogarithmic.
**Theorem 1.** MaxPDS cannot be approximated to better than a polylogarithmic factor. More precisely:

- Let $\varepsilon > 0$ be an arbitrary small constant, and suppose that $\text{NP} \not\subseteq \text{BPTIME}(2^{n^\varepsilon})$. Then there is no polynomial time algorithm which approximates MaxPDS to within $O(\log^\sigma n)$ for some constant $\sigma = \sigma(\varepsilon)$.

- Under Feige’s Random 3SAT Hypothesis \cite{6}, no polynomial time algorithms approximates MaxPDS to within $O(\log^{1/3-\sigma} n)$ for arbitrarily small constant $\sigma > 0$.

- Under the assumption that the Balanced Bipartite Independent Set Problem (BBIS) cannot be approximated better than $O(n^\varepsilon)$ for some constant $\varepsilon > 0$ (Hypothesis 3.22 of \cite{3}), there is no polynomial time algorithm which approximates MaxPDS to within $o(\log n)$.

We complement these lower bounds with an essentially matching upper bound.

**Theorem 2.** There is a polynomial time $O(\log n)$-approximation algorithm for MaxPDS.

The lower bound is described in Section 2.2, and the upper bound is in Section 2.3. Both are obtained in a similar way: a connection to another problem known as the Unique Coverage Problem UCP. We discuss UCP in more detail in Section 2.1 but informally it is a variation of Maximum Coverage with a similar uniqueness requirement as in MaxPDS (an element only counts as covered if it is contained in exactly one chosen set). Upper and lower bounds for UCP are known \cite{3}, so we derive our lower bounds by reducing from UCP to MaxPDS (in particular, the different lower bounds and their hardness assumptions are all direct from equivalent bounds and assumptions for UCP). For the upper bound, for technical reasons we do not give a black-box reduction to UCP, but instead give an algorithm which is directly inspired by the upper bound for UCP from \cite{3}.

The bulk of this paper is devoted to the next two results, which are about a natural game-theoretic version of MaxPDS / multicast capacity which we call the multicast capacity game. Informally, this is a game in which the nodes are players, and the utility of each node is 0 if it does not transmit, and otherwise is the number of neighbors which successfully heard the message minus the number who did not. In other words, each node gets a benefit from successfully transmitting its message to a neighbor, but pays a price for an unsuccessful transmission. We follow the lead of previous work on unicast capacity, in which a similar game is analyzed \cite{1,4,2}. The motivation is twofold. First, the game is a reasonable (though obviously not perfect) model of what incentives might be like for transmitters (at least in certain situations). But perhaps more interestingly, proving bounds on the quality of the equilibria gives a bound on any distributed algorithm which converges to such an equilibrium.

Since Nash equilibria are the standard solution concepts in game theory and algorithmic game theory, we focus on them here. In particular, we will bound the Price of Anarchy (the optimal number of successful receptions divided by the expected number of successful receptions in the worst Nash) and the Price of Stability (the same but with respect to the best Nash). Note that, like in the unicast game of \cite{1,4,2} but unlike in most games considered by the AGT community, the quality of a solution is not just the social welfare (sum of utilities) or some notion of fairness, but is instead a quantity (number of received messages) which is not directly optimized by any player. We provide nearly matching bounds on these quantities.

**Theorem 3.** The Price of Anarchy of any instance of the multicast capacity game is at most $O(\sqrt{n})$. 

3
Theorem 4. There is an instance of the multicast capacity game in which the Price of Stability is $\Omega\left(\frac{\sqrt{n}}{\log(n)}\right)$.

We prove Theorem 3 in Section 3.1 and Theorem 4 in Section 3.2. Note that the combination of these two bounds means that we have an extremely good understand of the value of the Nash equilibria (in particular, stronger than if we just had nearly-matching upper and lower bounds on the price of anarchy). The first bound tells us that in every instance, every single Nash equilibrium is within $O(\sqrt{n})$ from the optimum, while the second tells us that there are instances in which all Nash equilibria are at least $\Omega\left(\frac{\sqrt{n}}{\log(n)}\right)$ from optimum.

1.3 Related Work

As discussed earlier, this paper follows a fascinating line of work in the last decade on computing the capacity of wireless networks. There has been a particular focus on the SINR or physical model, in which we explicitly reason about the signal strength and interference at each receiver. However, there has also been significant work directly on graph-based models (e.g., [4]) and on the relationship between graph models and the SINR model [12] (which shows in particular that graphs can do a surprisingly good job of representing the physical model, motivating continued study of graph models). Typically in these graph models each link is represented by a node (rather than an edge) and two nodes are adjacent if they interfere, in which case the unicast capacity is equal to the maximum independent set. Typically authors assume (e.g. [1, 4, 12]) that the graph has some geometric structure (such as being a unit-disc graph) which makes computing maximum independent sets (at least approximately) an easier task.

From the perspective of computing the capacity, the most directly related work (and much of the inspiration for this paper) are [1] and [8], which to a large extent introduced the unicast capacity problem for worst-case inputs and gave the first approximation bounds. These bounds were improved in a series of papers, most notably including a constant-factor approximation [14], and have been generalized to even more general models and metrics, e.g. [9, 11].

Much of this paper focuses on analyzing a natural game-theoretic version of multicast capacity. This is directly inspired by a line of work on a related game for unicast capacity, initiated by [1] and continued in [4, 2]. These papers study various equilibria for the unicast capacity game (Nash equilibria in [1], coarse correlated equilibria in [4, 2]) and prove what are essentially price of anarchy bounds (upper bounds on the gap between the optimal capacity and the capacity at equilibrium). This game was also considered in [5], which showed the existence of Braess’s Paradox in the game (improvements in technology can result in worse performance) but bounded the damage it could cause. This paper is equivalent to [1] in that it is only the beginning of the study of the multicast capacity game; analyzing more complicated notions of equilibrium and studying Braess’s paradox are interesting future work.

1.4 Notation

Throughout this paper, we use the following notation and conventions. Given any graph $G = (V, E)$, unless otherwise stated, we refer to undirected graphs with $|V| = n$. Additionally, for any vertex $v \in V$, we define $N(v)$ as the open neighborhood of $v$, that is, $N(v) = \{u \in V : \{v, u\} \in E\}$.

All approximation ratios in this paper are written such they are at least 1, so for maximization problems such as MaxPDS they are given as the ratio of the optimal solution to the
solution constructed by the algorithm. This convention also extends to the price of anarchy or stability for a utility-maximization game.

2 Hardness and Approximations

In this section, we present a hardness of approximation result for the Maximum Perfect Dominated Set Problem as well as an approximation algorithm. We begin by defining the Unique Coverage Problem (UCP), which will be useful for both the upper and lower bounds.

2.1 The Unique Coverage Problem

The Unique Coverage Problem (UCP) was introduced by Demaine et al. in [3], where they gave both upper and lower bounds. In particular, it is defined as follows.

Definition 4. Given a universe $U$ of elements and a collection $S$ of subsets of $U$, the unique coverage problem (UCP) is to find a subcollection $S' \subseteq S$ of subsets which maximizes the number of elements that are uniquely covered, i.e., are in exactly one set of $S$.

The unique coverage problem is a variation on the maximum coverage problem with an added uniqueness requirement. In UCP, a solution attempts to maximize the number of elements covered by exactly one set, rather than the number of elements covered by at least one set. This is similar to the requirement we need for MaxPDS, which is that we are maximizing the number of perfectly dominated vertices, not the number of vertices dominated by at least one vertex.

Demaine et al. [3] proved the equivalent of Theorem 1 for UCP (all bounds and assumptions are exactly the same, just for UCP rather than MaxPDS) and an $O(\log n)$-approximation for UCP. Because of the similarity between UCP and MaxPDS, we base both our upper and lower approximability bounds on UCP.

2.2 Hardness of Approximation

In this section, we provide a polynomial time approximation-preserving reduction from UCP to MaxPDS, thus implying that the same hardness assumptions used to show a lower bound for UCP also hold for showing the hardness of approximating MaxPDS.

Theorem 5. Assuming UCP cannot be approximated to within $O(\log^c(n))$ for some constant $c$ satisfying Theorem 1, then MaxPDS is hard to approximate to within $O(\log^c(n))$.

Proof. Consider an instance of UCP with a universe $U$ of elements and a collection $S$ of subsets of $U$. For specified parameters $\alpha', \beta'$, given a subcollection $S' \subset S$, we define the following two cases.

1. $S'$ is a Yes-instance of UCP if the number of elements uniquely covered is at least $\alpha'$.
2. $S'$ is a No-instance of UCP if the number of elements uniquely covered is less than $\beta'$.

Given an instance of this problem, construct an undirected bipartite graph $G' = (V', E')$ such that $V'$ consists of a vertex $s_i$ for each set $S_i \in S$ and a vertex $x_j$ for each element $e_j \in U$. Let $(s_i, x_j) \in E'$ if $e_j \in S_i$. Let $A$ denote the set of vertices $s_i$ corresponding to sets in $S$, and let $B$ denote the vertices corresponding the elements in $U$. 

5
Construct a new bipartite graph \( G = (V, E) \) such that \( V \) consists of \( A \) and \( k \) copies of \( B \), denoted \( B_1, B_2, \ldots, B_k \). Let \( V \) have an additional vertex \( v \) that is adjacent to all vertices in \( A \). Let \( E \) consist of \( k \) copies of \( E' \), one for each bipartite subgraph over \((A, B_i)\) for all \( i \in [k] \).

Consider some solution \( S' \) to the UCP instance. Define \( D = \{s_i : S_i \in S'\} \cup \{v\} \). If \( S' \) is a Yes-instance of UCP, then the number of vertices perfectly dominated by \( D \) is \( \alpha \geq \alpha'k \), because in each of the \( B_i \), there are at least \( \alpha' \) perfectly dominated vertices. On the other hand, if \( S' \) is a No-instance of UCP, then there are only \( \beta < |S| + k\beta' \) vertices perfectly dominated by \( D \), because \( \{s_i : S_i \in S'\} \) perfectly dominates less than \( k\beta' \) of the vertices in the \( B_i \) and \( v \) perfectly dominates the \( |S| \) vertices in \( A \).

Now, set \( k = |S| \). Then \( \alpha \geq \alpha' |S| = \alpha'k \) and \( \beta < |S| + |S|/\beta' = k + k\beta' = k(\beta' + 1) \). Therefore, the approximation ratio for \( \text{MaxPDS} \) in this setting is \( \frac{\alpha'}{\beta'} > \frac{\alpha'k}{\beta'k + 1} = \frac{\alpha'}{\beta' + 1} \geq \frac{\alpha'}{2p} \) when \( \beta' \geq 1 \), which is trivially true. Since all we have done is create \( |S| \) repetitions of \( B \), this can be done in polynomial time.

Therefore, this reduction begins with an instance of UCP with an approximation ratio of \( \frac{\alpha'}{2p} \) and transforms the problem into an instance of \( \text{MaxPDS} \) with an approximation ratio of \( \frac{\alpha'}{2p} \). Let \( n' \) be the size of the input to this reduction, and let \( n \) be the size of the resulting instance of \( \text{MaxPDS} \). By assumption, \( \frac{\alpha'}{2p} = \Omega(\log^c(n')) \). Therefore, we want to show that \( \frac{n'}{2p} = \Omega(\log^c(n)) \). We start with \( n' = |S| + |E'| \) and end with \( n = |S| + k|E'| \). Then \( n = |S| + k|E'| = |S| + |S||E'| = |S|(1 + |E'|) \leq 2|S||E'| < 2(n')^2 \), and hence \( \log^c(n) \leq \log^c(2(n')^2) \leq 4\log^c(n') \), implying that \( \log^c(n') \geq \frac{1}{4} \log^c(n) \). Therefore, \( \frac{n}{2p} \geq \frac{\alpha'}{2p} \geq \frac{1}{4} \log^c(n') \geq \frac{1}{4^c+1} \log^c(n) = \Omega(\log^c(n)) \) as desired, thus showing that \( \text{MaxPDS} \) is hard to approximate to within \( O(\log^c(n)) \).

This reduction from UCP to \( \text{MaxPDS} \) shows that \( \text{MaxPDS} \) is hard to approximate to within \( O(\log^c(n)) \) under any hardness assumption for which UCP is hard to approximate to within \( O(\log^c(n)) \). In particular, this holds for the three different hardness assumptions used to show the hardness of approximating UCP, thus proving Theorem 1.

2.3 Approximation Algorithm

In this section we present an \( O(\log(n)) \)-approximation algorithm for \( \text{MaxPDS} \), i.e., a proof of Theorem 2. In [3], Demaine et al. show an \( O(\log(n)) \)-approximation algorithm for UCP. Because of the similarities between UCP and \( \text{MaxPDS} \), the obvious approach of \( \text{MaxPDS} \) would be to give a black-box reduction to UCP. For example, the following reduction would be very natural: given a \( \text{MaxPDS} \) instance \( G = (V, E) \), create a UCP instance such that for every vertex \( i \in V \), there exists a set \( S_i \) containing elements corresponding to \( N(i) \cup \{i\} \). However, in the UCP instance, a set \( S_i \) can uniquely cover element \( i \), while in the \( \text{MaxPDS} \) instance, if a vertex \( i \) is in the set, it cannot be perfectly dominated by definition. In other words, an algorithm for UCP might get “credit” for uniquely covering \( i \) with set \( S_i \), even though the equivalent solution to \( \text{MaxPDS} \) would not get credit for perfectly dominating \( i \). A similar issue arises if we leave \( i \) out of \( S_i \), since in that case a different set could get credit for uniquely covering \( i \) even if \( S_i \) is included, which would correspond to “perfectly dominating” \( i \) even if \( i \) in the \( \text{MaxPDS} \) solution.

To get around this difficulty, we simply directly give an \( O(\log n) \)-approximation algorithm for \( \text{MaxPDS} \). Both the algorithm and the analysis are straightforward adaptations of the algorithm for UCP from [3], so we defer them to Appendix A.
3 Game Theory

Given our understanding of maximizing the multicast capacity in a graph, we now analyze the problem in a distributed setting as a game with self-interested players. We define a natural game for this setting, where a player $i$ has incentive to broadcast if most of $N(i)$ would receive $i$'s transmission, but does not have an incentive if $i$ would mostly be broadcasting to neighbors that aren’t listening.

Formally, the multicast capacity game is defined as follows. Let $S = \{0, 1\}^n$ be the strategy space, where for each player $i \in [n]$, for each $s \in S$, $s_i = 1$ when $i$ chooses to broadcast and is 0 otherwise. Let $c_i(s)$ denote the number of neighbors of $i$ which are broadcasting under $s$, i.e., $c_i(s) = |\{j : j \in N(i) \land s_j = 1\}|$. Then, given $s \in S$, define $A_i(s) = \{j \in N(i) : c_j(s) = 1 \land s_j = 0\}$ to be the neighbors of $i$ receiving exactly one message under $s$, and $B_i(s) = \{j \in N(i) : c_j(s) \geq 2 \lor s_j = 1\}$ to be the neighbors of $i$ either receiving at least two messages under $s$ or broadcasting, meaning that $i$ cannot succeed on these neighbors. Note that $|A_i(s)| + |B_i(s)| = |N(i)|$ for all $i \in V$. The utility for player $i$ is $u_i : S \rightarrow \mathbb{Z}$, defined as

$$u_i(s) = \begin{cases} |A_i(s)| - |B_i(s)| & \text{if } s_i = 1 \\ 0 & \text{if } s_i = 0 \end{cases}$$

This game intuitively models the fact that each node would like to send its message to its neighbors, and gets a benefit proportional to the number of successes but with a penalty for failures (possibly due to either the cost of wasting the transmission power, or more altruistically, a payment for the interference caused).

This definition of the multicast capacity game gives us a way to analyze the relationship between the quality of equilibria in a distributed setting and the optimal solution to the graph theory problem in question. Similarly, in [3], the independent set game is discussed, which is just the graph-theoretic unicast capacity game. The utilities are similar to those in our game: player $i$ receives utility 1 for broadcasting while none of $N(i)$ is broadcasting, -1 for broadcasting if a neighbor is broadcasting, and 0 for being quiet (the motivation is that each node is a link, and an edge between two links means that they cannot simultaneously succeed). While similarly motivated, it turns out that the independent set game and our multicast capacity game are quite different in terms of their equilibria.

A pure Nash equilibrium (PNE) is a strategy vector $s \in S$ in which no player has any incentive to deviate. Slightly more formally, $s$ is a pure Nash equilibrium if $u_i(s_{-i}, s_i) \leq u_i(s)$ for all players $i$, where $s_{-i}, s_i$ is a vector formed by replacing the $i$’th coordinate of $s$ with $s_i$. We can generalize this by allowing probabilities, and in particular allowing each player to have a probability distribution over its possible strategies (i.e., a distribution over $\{0, 1\}$). Such a collection of distributions is a mixed Nash equilibrium (MNE), or just a Nash equilibrium, if for every player the expected utility (when $s$ is from the product distribution of the player distributions) cannot be increased by changing its own distribution.

For each strategy profile $s \in S$, we define $V(s)$ as the number of successful receptions throughout the network (note that $V(s)$ is not simply the social welfare, i.e., the sum of the utilities, and hence differs from much of modern algorithmic game theory). Hence the optimal solution, in the sense of $\text{MAXPDS}$ and the previous section, is simply $\text{OPT} = \max_{s \in S} V(s)$.

In this section we analyze this game. We first note in Appendix [4] that it is not clear whether this game always admits a pure Nash equilibrium, since the natural guess that maximal perfect
dominating sets are equilibria is incorrect (in the independent set game any maximal independent set is a pure Nash). Fortunately, as we all know, mixed Nash equilibria do always exist, and so in Sections 3.1 and 3.2 we analyze general mixed Nash equilibria and their distance from optimality.

### 3.1 Price of Anarchy

The Price of Anarchy of a game is a measurement which compares the equilibrium with the lowest value to \( \text{OPT} \).

**Definition 5 (Price of Anarchy (PoA)).** Let \( N \) be the set of product distributions \( \sigma \) over \( S \) corresponding to Nash Equilibria (pure or mixed). Then the Price of Anarchy is defined as
\[
\max_{\sigma \in \mathcal{S}} \min_{\sigma' \sim \sigma} \mathbb{E} V(\tau),
\]
which is the ratio of \( \text{OPT} \) to the Nash Equilibrium with the lowest value.

We now prove Theorem 4 an upper bound of \( O(\sqrt{n}) \) on the price of anarchy. We give an outline of the proof, but all omitted details can be found in Appendix 2.

Let \( G = (V, E) \) with \( V = [i] \) be an instance of the Maximum Perfect Dominated Set game. Assume that \( G \) is connected, because any vertex \( i \) with no neighbors cannot contribute to the value of a Nash Equilibrium or the value of \( \text{OPT} \), and can therefore be deleted. Let \( \sigma \) be a product distribution over the set of strategies for each player and suppose that \( \sigma \) is a MNE.

We define the following probabilities with respect to \( \sigma \). Given any vertex \( i \), let \( p_i \) be the probability with which \( i \) broadcasts. Then, for each \( j \in N(i) \), define \( \alpha_{ij} \) as the probability that, if \( i \) chooses to broadcast, \( j \) would successfully hear the transmission from \( i \). Formally, \( \alpha_{ij} = (1 - p_j) \prod_{i' \in N(j) \setminus \{i\}} (1 - p_{i'}) \). Additionally, let \( S_i \) be the expected number of vertices that receive only a transmission from \( i \) given that \( i \) chooses to broadcast, which is \( S_i = \sum_{j \in N(i)} \alpha_{ij} \). Similarly, let \( F_i = \sum_{j \in N(i)} (1 - \alpha_{ij}) \) be the expected number of failures at neighbors of \( i \) if \( i \) chooses to broadcast.

Then, we can define the following quantities. Let \( B \) be the expected number of broadcasters; that is, \( B = \sum_{i \in [n]} p_i \). Let \( S \) be the expected number of successes, meaning the vertices that are receiving exactly one transmission and are not broadcasting. This is the value which we are trying to bound. To express \( S \) in terms of \( \alpha_{ij} \), consider the following. Given any vertex \( i \) with neighbors \( j_1, \ldots, j_m \in N(i) \), let \( X_{jk} \) denote the event that \( i \) successfully receives a message from \( j_k \) for any \( k \in [m] \). Because \( i \) can only be a success for one of the \( j_k \), then \( X_{j_1}, X_{j_2}, \ldots, X_{j_m} \) are disjoint. Each of these events occurs with probability \( p_j \alpha_{ji} \). Hence, \( \Pr [i \text{ is a success}] = \sum_{j \in N(i)} p_j \alpha_{ji} \). Therefore, we can write \( S \) as \( S = \sum_{i \in [n]} \sum_{j \in N(i)} p_j \alpha_{ji} \).

Let \( F \) be the expected number of failures due to collisions, that is, vertices that are not broadcasting but are also receiving at least two transmissions. Formally,
\[
F = \sum_{i \in [n]} (1 - p_i) \left( 1 - \prod_{j \in N(i)} (1 - p_j) - \sum_{j \in N(i)} p_j \prod_{j' \not\in N(i)} (1 - p_{j'}) \right),
\]
\[
= \sum_{i \in [n]} (1 - p_i) \left( 1 - \prod_{j \in N(i)} (1 - p_j) - \frac{1}{1 - p_i} \sum_{j \in N(i)} p_j \alpha_{ji} \right).
\]

Finally, let \( A \) be the expected number of vertices that do not broadcast and do not receive any transmissions, which is \( A = \sum_{i \in [n]} (1 - p_i) \prod_{j \in N(i)} (1 - p_j) \).

We now give some lemmas which will let us relate these quantities.
Lemma 6. \( B + S + F + A = n \).

Lemma 7. \( S_i \geq F_i \) for any vertex \( i \) with \( p_i > 0 \).

Lemma 8. \( S_i = \sum_{j \in N(i)} \alpha_{ij} \geq \frac{1}{2} \) for any vertex \( i \) with \( p_i > 0 \).

The previous two lemmas will let us relate \( B \) and \( F \) to \( S \).

Lemma 9. \( B \leq 2S \).

Lemma 10. \( F \leq S \).

Relating \( A \) to \( S \) is significantly more difficult, and requires reasoning separately about vertices which contribute significantly by themselves to \( A \) and vertices which do not contribute much individually. Dividing up vertices in this way lets us reason more combinatorially about degrees between various sets, allowing us to eventually prove the following.

Lemma 11. \( A \leq .9n + 2000S^2 \).

We are now ready to complete the proof of Theorem 3.

Proof of Theorem 3. By Lemmas 9, 10, and 11,
\[
n = B + S + F + A \leq B + S + F + .9n + 2000S^2
\leq 4S + .9n + 2000S^2
\]
so \( n \leq 40S + 2000S^2 = O(S^2) \). Therefore \(|\text{OPT}| \leq n = O(S^2)\), implying that the Price of Anarchy is at most \( O(\sqrt{n}) \). \( \square \)

3.2 Price of Stability

The Price of Stability of a game is a measurement which compares the equilibrium with the highest value to \( \text{OPT} \). Intuitively, it gives a bound on how “good” an equilibrium can be, when compared to \( \text{OPT} \).

Definition 6 (Price of Stability (PoS)). Let \( N \) be the set of product distributions \( \sigma \) over \( S \) corresponding to Nash Equilibria (pure or mixed). Then the Price of Stability is defined as
\[
\max_{\sigma \in N} \frac{V(s)}{\sup_{\sigma \in N} |V(\sigma)|},
\]
which is the ratio of \( \text{OPT} \) to the Nash Equilibrium with the highest value.

We now prove Theorem 4, a lower bound of \( \Omega(\sqrt{n}/\log n) \) on the Price of Stability.

Proof of Theorem 4. Let \( G = (V, E) \) be a graph composed of \( n = q + 3\sqrt{q} + 2 \) vertices for some parameter \( q \), such that \( V = A \cup B \) and \( B = \bigcup_{i \in [\sqrt{q}+2]} B_i \). Let \( A \) be a clique on \( \sqrt{q} + 2 \) vertices, and for each \( v_i \in A \), let \( B_i \) be an independent set of size \( \sqrt{q} \) such that \( v_i \) is adjacent to each vertex in \( B_i \). We will show that in this graph any MNE has value of at most \( O(\sqrt{q} \log(q)) \), whereas \( \text{OPT} \) in this graph has a value of at least \( q + 2\sqrt{q} \). Let \( \sigma \) be a MNE, and let \( p_i \) be the probability with which \( i \) broadcasts.
For any $i \in A$, suppose that $\sum_{j \in A, j \neq i} p_j > \log(2\sqrt{q} + 2)$. Then, the probability that $i$ succeeds at broadcasting to $A$ is equivalent to the probability that no other node $j \in A$ broadcasts. Since $\sigma$ is a product distribution, this probability is

$$\prod_{j \in A, j \neq i} (1 - p_j) \leq \prod_{j \in A, j \neq i} e^{-p_j} = e^{-\sum_{j \in A, j \neq i} p_j} < e^{-\log(2\sqrt{q} + 2)} = \frac{1}{2\sqrt{q} + 2}.$$  

Then, the expected utility of broadcasting with probability $p_i$ for vertex $i$ would be

$$E_{s \sim \sigma} [u_i(s)] = p_i (\Pr \{\text{No other } j \in A \text{ broadcasts}\} \cdot (2\sqrt{q} + 1) - \Pr \{\exists j \in A, j \neq i, \text{ that broadcasts}\})$$

$$< p_i \left( \frac{1}{2\sqrt{q} + 2} (2\sqrt{q} + 1) + \frac{1}{2\sqrt{q} + 2} - 1 \right) = p_i \left( \frac{1}{2\sqrt{q} + 2} (2\sqrt{q} + 2) - 1 \right) = 0.$$  

This is maximized when $p_i = 0$ so vertex $i$ has no incentive to broadcast. Therefore, in any MNE, the expected number of broadcasters in $A$ can be at most $\log(2\sqrt{q} + 2)$. If any vertices in $B$ broadcast, they can add at most $\sqrt{q} + 2$ successes, all of which are in $A$. Therefore,

$$E_{s \sim \sigma} [V(s)] \leq \sqrt{q} + 2 + \sqrt{q} \log(2\sqrt{q} + 2)$$

$$= O(\sqrt{q} \log(q)) = O(\sqrt{n} \log(n)).$$

Now, consider $\text{OPT}$. When all vertices in $A$ broadcast with probability 1, the number of successes is $q + 2\sqrt{q}$ so $\text{OPT} \geq q = \Omega(n)$. Therefore, the Price of Stability in this graph is at least $\Omega(\sqrt{n} \log(n))$.  

4 Conclusion

In this paper, we analyzed the multicast capacity of a network as both an optimization problem on a graph and a game in a distributed network. We introduced the Maximum Perfect Dominated Set problem as the equivalent of maximizing multicast capacity in a graph, and showed both upper and lower bounds for the approximability of the problem. We also defined the multicast capacity game and gave complementary bounds on the Price of Anarchy and the Price of Stability.

We hope that this is only the beginning of analyzing the multicast capacity of wireless networks. Many interesting open questions remain, paralleling the work on unicast capacity. For example, what if we consider restricted classes of graphs, such as unit-disc graphs, which are typically used to model wireless networks? Does MaxPDS become easier, and are equilibria in the multicast capacity game closer to optimum? Or what if we consider version of equilibria such as coarse correlated equilibria which well-known learning algorithms (namely, no-regret algorithms) are known to converge to? In [4, 2] these equilibria were used to analyze simple distributed algorithms for unicast capacity maximization – can something similar be done here? And for all of these questions, what happens if we work in the SINR model rather than the graph model?
References

[1] M. Andrews and M. Dinitz. Maximizing capacity in arbitrary wireless networks in the sinr model: Complexity and game theory. In Proceedings of IEEE INFOCOM, pages 1332–1340. IEEE, 2009.

[2] E. I. Asgeirsson and P. Mitra. On a game theoretic approach to capacity maximization in wireless networks. In Proceedings of IEEE INFOCOM, pages 3029–3037. IEEE, 2011.

[3] E. Demaine, U. Feige, M. Hajiaghayi, and M. Salavatipour. Combination can be hard: Approximability of the unique coverage problem. SIAM Journal on Computing, 38(4):1464–1483, 2008.

[4] M. Dinitz. Distributed algorithms for approximating wireless network capacity. In Proceedings of IEEE INFOCOM, pages 1–9, March 2010.

[5] M. Dinitz and M. Parter. Braess’s paradox in wireless networks: The danger of improved technology. In Proceedings of the 27th International Symposium on Distributed Computing (DISC), pages 477–491. Springer, 2013.

[6] U. Feige. Relations between average case complexity and approximation complexity. In Proceedings of the Thiry-fourth Annual ACM Symposium on Theory of Computing, STOC ’02, pages 534–543, New York, NY, USA, 2002. ACM.

[7] O. Goussevskaia, M. M. Halldórsson, and R. Wattenhofer. Algorithms for wireless capacity. IEEE/ACM Trans. Netw., 22(3):745–755, 2014.

[8] O. Goussevskaia, R. Wattenhofer, M. M. Halldórsson, and E. Welzl. Capacity of arbitrary wireless networks. In Proceedings of IEEE INFOCOM, pages 1872–1880. IEEE, 2009.

[9] M. M. Halldórsson and P. Mitra. Wireless capacity with oblivious power in general metrics. In Proceedings of the Twenty-second Annual ACM-SIAM Symposium on Discrete Algorithms, SODA ’11, pages 1538–1548, Philadelphia, PA, USA, 2011. Society for Industrial and Applied Mathematics.

[10] M. M. Halldórsson and P. Mitra. Wireless connectivity and capacity. In Proceedings of the Twenty-third Annual ACM-SIAM Symposium on Discrete Algorithms, SODA ’12, pages 516–526, Philadelphia, PA, USA, 2012. Society for Industrial and Applied Mathematics.

[11] M. M. Halldórsson and P. Mitra. Wireless capacity with arbitrary gain matrix. Theor. Comput. Sci., 553:57–63, 2014.

[12] M. M. Halldorsson and T. Tonoyan. How well can graphs represent wireless interference? In Proceedings of the Forty-seventh Annual ACM Symposium on Theory of Computing, STOC ’15, pages 635–644, New York, NY, USA, 2015. ACM.

[13] T. W. Haynes, S. T. Hedetniemi, and P. J. Slater. Fundamentals of domination in graphs, volume 208 of Monographs and Textbooks in Pure and Applied Mathematics. Marcel Dekker, Inc., New York, 1998.
[14] T. Kesselheim. A constant-factor approximation for wireless capacity maximization with power control in the SINR model. In D. Randall, editor, Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), pages 1549–1559. SIAM, 2011.

[15] P. M. Weichsel. Dominating sets in n-cubes. J. Graph Theory, 18(5):479–488, Aug. 1994.

[16] C.-C. Yen and R. Lee. The weighted perfect domination problem. Information Processing Letters, 35(6):295 – 299, 1990.
A Proofs from Section 2

A.1 Proof of Theorem 2

Let $G = (V, E)$ be an instance of MAXPDS with $|V| = n$. For any set $S$ of vertices, let $f(S)$ denote the number of perfectly dominated vertices by $S$. Let ALG be an initially empty set and let OPT denote the optimal set of dominating vertices in the above instance.

Partition the vertices into $\log(n)$ groups $G_i$ such that $v \in G_i$ if $2^i \leq d(v) < 2^{i+1}$. Then, there must exist a group $i^*$ such that $|G_{i^*}| \geq \frac{1}{\log(n)} \cdot n$. Additionally, since $f(OPT) \leq n$, then

$$|G_{i^*}| \geq \frac{1}{\log(n)} \cdot n \geq \frac{1}{\log(n)} f(OPT).$$

Our solution ALG is now constructed by randomly adding each vertex $i$ to ALG independently with probability $\frac{1}{2^{i^*}}$ when $i^* > 0$, and with probability $\frac{1}{2}$ when $i = 0$.

Let $S \subset V$ be the vertices that are perfectly dominated by ALG. For any vertex $v \in G_{i^*}$, let $d = d(v) \in [2^i, 2^{i+1})$. Then, the probability that $v$ is perfectly dominated by ALG is the probability that exactly one of $N(v)$ is in ALG and the remaining vertices in $N(v)$ are not in ALG. Since each vertex is chosen to be in ALG independently, then when $i > 0$,

$$\Pr [v \in S] = \left(d \cdot \frac{1}{2^{i^*}}\right) \left(1 - \frac{1}{2^{i^*}}\right)^{d-1} \geq \left(1 - \frac{1}{2^{i^*}}\right)^{2^{i^*+1} - 1} \geq \left(1 - \frac{1}{2^{i^*}}\right)^{2^{i^*+1}} \geq \frac{1}{e^4}.$$

When $i = 0$, then $d = 1$ and

$$\Pr [v \in S] = \left(1 \cdot \frac{1}{2}\right) \left(1 - \frac{1}{2}\right)^{d-1} = \frac{1}{2} \cdot \left(\frac{1}{2}\right)^0 = \frac{1}{2}.$$

Therefore,

$$\mathbb{E}[f(ALG)] = \sum_{v \in V} \Pr [v \in S] \geq \sum_{v \in G_{i^*}} \Pr [v \in S] \geq \min \left\{ \frac{1}{e^4}, \frac{1}{2} \right\} |G_{i^*}| \geq \frac{1}{e^4 \log(n)} f(OPT).$$

Therefore, $\frac{\mathbb{E}[f(ALG)]}{f(OPT)} = O(\log(n))$ as desired.

Note that while the above algorithm is randomized, it is straightforward to derandomize in polynomial time using the standard method of conditional expectation.

B Proofs from Section 3

B.1 Nash Equilibria and Maximal Perfect Dominating Sets

When analyzing a game such as the multicast capacity game, we would like to understand when pure Nash equilibria exist. Because the optimal solution to an instance of this game is a maximum perfect dominated set, a natural idea to find a potential equilibrium is to consider a maximal perfect dominated set, which is a set of vertices such that the addition or deletion of any vertex to or from that set does not increase the number of perfectly dominated vertices. For example, in the independent set game discussed above, it is easy to see that the utilities are defined such that every maximal independent set constitutes a pure Nash equilibrium. While that may provide hope
of showing the existence of a PNE in our game by analyzing the maximal perfect dominating sets, in this section we show that there is not necessarily overlap between the set of PNEs and the set of maximal perfect dominating sets.

Our main result for this section is the following.

**Theorem 12.** There exists an instance of the multicast capacity game for which there exists at least one PNE and there is no intersection between the set of PNEs and the set of maximal perfect dominating sets.

**Proof.** Consider the following example. Let \( G = (V, E) \) be a graph shown in Figure 1. Let \( C_1 \) and \( C_2 \) each be independent sets of size 10, where \( a_i \) is adjacent to all vertices in \( C_i \) for \( i \in \{1, 2\} \).

![Figure 1: Example for Theorem 12](image)

We now enumerate the set of PNEs in this graph. Suppose that there exists a PNE where at least one of the \( a_i \) does not broadcast. Since the majority of \( N(a_i) \) is in \( C_i \), it must be the case that most of \( C_i \) is transmitting. However, this would not be a PNE, because if more than one node in \( C_i \) is broadcasting, they all have incentive to stop broadcasting. Therefore, there does not exist a PNE where either \( a_1 \) or \( a_2 \) is not broadcasting. Now, consider \( a_1 \cup a_2 \) as the set of broadcasters. This is already a PNE, because no other nodes have incentive to broadcast. Furthermore, one can see that this is the only PNE. Note that this is not a maximal perfect dominating set, because if \( b_1 \) were to broadcast, \( u \) would be added to the number of successes, and in fact the resulting set \( \{a_i \cup a_2 \cup b_1\} \) is a maximal perfect dominating set. Therefore, there is no intersection between the set of PNEs and the set of maximal perfect dominating sets in this graph.

**B.2 Proofs from Section 3.1**

**Proof of Lemma 6.** Consider the sum \( B + S + F + A \). Each of \( B, S, F, A \) is a sum over all \( i \in [n] \). Therefore, taking the \( i \)th term from each, the contribution this term to \( B + S + F + A \) is

\[
P_i + \sum_{j \in N(i)} p_j \alpha_{ji} + (1 - p_i) \left( 1 - \prod_{j \in N(i)} (1 - p_j) - \frac{1}{1 - p_i} \sum_{j \in N(i)} p_j \alpha_{ji} \right) + (1 - p_i) \prod_{j \in N(i)} (1 - p_j)
\]

\[
= p_i + \sum_{j \in N(i)} p_j \alpha_{ji} + 1 - p_i - (1 - p_i) \prod_{j \in N(i)} (1 - p_j) - \sum_{j \in N(i)} p_j \alpha_{ji} + (1 - p_i) \prod_{j \in N(i)} (1 - p_j)
\]

\[
= p_i + 1 - p_i - (1 - p_i) \prod_{j \in N(i)} (1 - p_j) + (1 - p_i) \prod_{j \in N(i)} (1 - p_j)
\]

\[
= p_i + 1 - p_i = 1.
\]

Therefore, \( B + S + F + A = \sum_{i \in [n]} 1 = n \).
\textbf{Proof of Lemma 7.} Let \( i \in [n] \) with \( p_i > 0 \). By definition,
\[
\mathbb{E}_{\sigma \sim \sigma} [u_i(s)] = p_i(S_i - F_i).
\]
Because \( \sigma \) is a MNE, \( \mathbb{E}_{\sigma \sim \sigma} [u_i(s)] \geq 0 \) so \( p_i(S_i - F_i) \geq 0 \). Therefore, since \( p_i > 0 \), it must be the case that \( S_i - F_i \geq 0 \), so \( S_i \geq F_i \) as desired. \( \Box \)

\textbf{Proof of Lemma 8.} Suppose that \( p_i > 0 \). Because we are not considering any isolated vertices, \( S_i + F_i = \sum_{j \in N(i)} \alpha_{ij} + \sum_{j \in N(i)} (1 - \alpha_{ij}) = |N(i)| \geq 1 \).
By Lemma 7, \( S_i \geq F_i \). Therefore, since \( S_i \) accounts for more than half of the above sum, \( S_i \geq \frac{1}{2} \). \( \Box \)

\textbf{Proof of Lemma 9.} Let \( i \) be a vertex that is broadcasting with \( p_i > 0 \). By Lemma 8 we have that \( \sum_{j \in N(i)} \alpha_{ij} \geq \frac{1}{2} \). Therefore,
\[
B = \sum_{i \in [n]} p_i = \sum_{i : p_i > 0} p_i = 2 \sum_{i : p_i > 0} \frac{1}{2} p_i \leq 2 \sum_{i : p_i > 0} \sum_{j \in N(i)} p_i \alpha_{ij} = 2 \sum_{i \in [n]} \sum_{j \in N(i)} p_i \alpha_{ij} = 2 S.
\]

\textbf{Proof of Lemma 10.} Let \( i \) be any vertex and let \( j_1, j_2, \ldots, j_m \in N(i) \). Let \( X_{i1}^i, X_{i2}^i, \ldots, X_{im}^i \) denote the events that \( i \) is a failure for \( j_k \) for each \( k \), meaning the event that \( j_k \) attempts to broadcast to \( i \) but \( i \), while not broadcasting, receives at least one other transmission. The probability of such an event \( X_k^i \) occurring is \( p_j(1 - \alpha_{jki}) \). Let \( X^i \) denote the event that there exists a \( j \in N(i) \) such that \( j \) attempts to broadcast to \( i \) and fails. Then, by a union bound we have that
\[
\Pr [X^i] = \Pr [X_{i1}^i \cup X_{i2}^i \cup \ldots \cup X_{im}^i] \leq \sum_{j_k \in N(i)} \Pr [X_{j_k}^i] = \sum_{j_k \in N(i)} p_j(1 - \alpha_{jki}).
\]
Therefore,
\[
F = \sum_{i \in [n]} \Pr [X^i] \leq \sum_{i \in [n]} \sum_{j \in N(i)} p_j(1 - \alpha_{jki}) = \sum_{i \in [n]} \sum_{j \in N(i)} (1 - \alpha_{jki}) = \sum_{i \in [n]} p_i F_i.
\]
Also, note that
\[
S = \sum_{i \in [n]} \sum_{j \in N(i)} p_j \alpha_{jki} = \sum_{i \in [n]} \sum_{j \in N(i)} \alpha_{ij} = \sum_{i \in [n]} p_i S_i.
\]
Then, we have that
\[
F \leq \sum_{i \in [n]} p_i F_i = \sum_{i : p_i > 0} p_i F_i \leq \sum_{i : p_i > 0} \sum_{i \in [n]} p_i S_i = \sum_{i \in [n]} p_i S_i = S
\]
by Lemma 7 so \( F \leq S \) as desired. \( \Box \)
Proof of Lemma 11. Let $\beta_i$ be the probability that $i$ does not broadcast and does not receive any messages. Let $X = \{i \in [n] : \beta_i > .9\}$ be the set of vertices who do not broadcast or receive messages with probability greater than .9. Let $Y = V \setminus X$. For any vertex $i$, let $d^Y_i = |N(i) \cap Y|$ and let $d^X_i = |N(i) \cap X|$. For any set of vertices $U$ and vertex $i$, let $S_i^U$ be the expected number of successes $i$ receives on $U \cap N(i)$ given that $i$ chooses to broadcast and similarly let $F_i^U$ be the expected number of failures $i$ receives on $U \cap N(i)$ should $i$ choose to broadcast.

For any $j \in X$ and $i \in N(j)$, since $\beta_j > .9$, we have that 
\[ .9 < \beta_j = (1 - p_j) \prod_{i' \in N(j)} (1 - p_{i'}) \leq (1 - p_j) \prod_{i' \notin N(j)} (1 - p_{i'}) = \alpha_{ij}. \]

Therefore, for any $i$, 
\[ S_i^X = \sum_{j \in N(i) \cap X} a_{ij} > \sum_{j \in N(i) \cap X} .9 = .9d_i^X \]
and 
\[ F_i^X = \sum_{j \in N(i) \cap X} (1 - a_{ij}) = d_i^X - \sum_{j \in N(i) \cap X} \alpha_{ij} \leq d_i^X - .9d_i^X = .1d_i^X. \]

Additionally, we trivially know that $S_i^Y \geq 0$ and $F_i^Y \leq d_i^Y$. Therefore, for any $i$, 
\[ E_{s \sim \sigma} [u_i(s)] = p_i(S_i - F_i) \]
\[ = p_i(S_i^X - F_i^X + S_i^Y - F_i^Y) \]
\[ \geq p_i(.9d_i^X - .1d_i^X + 0d_i^Y - d_i^Y) \]
\[ = p_i(8d_i^X - d_i^Y). \]
We know that for all vertices, because we are in a MNE, $E_{s \sim \sigma} [u_i(s)] \geq 0$ and furthermore $p_i$ is set to maximize the above quantity. We will use this to bound $d_i^X$ for all $i$.

Suppose $i \in X$. Then it must be that $d_i^Y > 0$, because otherwise $E_{s \sim \sigma} [u_i(s)] = p_i.8d_i^X$ and would be maximized when $p_i = 1$. Since $i \in X$, $.9 < \beta_i \leq (1-p_i)$ so $p_i < .1$, which is a contradiction. Therefore, for all $i \in X$, $d_i^Y > 0$ so $i$ has at least one neighbor in $Y$.

Now, let $i \in Y$. First, suppose that $p_i = 0$. Then, if $.8d_i^X - d_i^Y > 0$, $E_{s \sim \sigma} [u_i(s)]$ would be maximized when $p_i = 1$. Therefore, it must be the case that $.8d_i^X - d_i^Y \leq 0$ so $.8d_i^X \leq d_i^Y$. For the second case, suppose $p_i > 0$. If $.8d_i^X - d_i^Y > 0$, then $E_{s \sim \sigma} [u_i(s)]$ would be maximized when $p_i = 1$. In that case, for $j \in N(i) \cap X$, $.9 < \beta_j \leq \prod_{i' \in N(j)} (1 - p_{i'}) < (1 - p_i)$ so $p_i < .1$, which is a contradiction. Therefore, in either case, we have that $.8d_i^X \geq d_i^Y$.

We can now bound the size of $X$. Since all vertices in $X$ have a neighbor in $Y$, we can bound $X$ by counting the neighbors of $Y$ in $X$. Therefore, 
\[ |X| \leq \sum_{i \in Y} d_i^X \leq \sum_{i \in Y} \frac{1}{8} d_i^Y \leq \frac{5}{4} \sum_{i \in Y} |Y| \leq \frac{5}{4} |Y|^2. \]

Now, consider $|Y|$. Each node $i \in Y$ has $\beta_i \leq .9$, so by Lemma 8, the contribution of vertex $i$ to $B + S + F$ is at least .1. Therefore, $B + S + F \geq .1 |Y|$, so 
\[ |X| \leq \frac{5}{4} |Y|^2 \leq \frac{5 \cdot 10^2}{4} (B + S + F)^2 \leq 125(4S)^2 = 2000S^2. \]

Finally, consider $A$, which is the sum over all nodes $i$ of $\beta_i$. Each vertex $i \in X$ has $\beta_i \leq 1$ and each vertex $j \in Y$ has $\beta_j \leq .9$, and there are at most $n$ nodes in $Y$. Therefore, $A \leq .9n + |X| \leq .9n + 2000S^2$. 

\[ \square \]