7-1-2010

Security in network games

Rustagi Navin

Navin Rustagi

Candidate

Computer Science

Department

This dissertation is approved, and it is acceptable in quality and form for publication on microfilm:

Approved by the Dissertation Committee:

Jared Saia, Chairperson

Thomas P. Hayes

Josep Diaz.

Accepted:

Dean, Graduate School

Date

Security in Network games

by

Navin Rustagi

B.Sc., Chennai Mathematical Institute, 2002
M.Sc., Chennai Mathematical Institute, 2004

DISSERTATION

Submitted in Partial Fulfillment of the
Requirements for the Degree of

Doctor of Philosophy
Computer Science

The University of New Mexico

Albuquerque, New Mexico

July, 2010

Dedication

To my parents, for their support, encouragement, love and and undying devotion to my education.

To my gurus at UNM and CMI, whose faith in my abilities is a huge source of inspiration for me.

Acknowledgments

I would like to thank many people without whom this dissertation would not be possible. Foremost in the list is Jared. He suggested great problems to work on and contributed to this work significantly with his advice, suggestions and insights. He was also very helpful in inculcating in me an interest for theory. His patience with my mistakes has been most helpful for the completion of this work.

I would like to thank James Aspnes and Jared Saia for their contributions to work presented in Chapter 2. I would like to thank Josep Díaz, Dieter Mitsche and Jared Saia for their contributions to work presented in Chapter 4. I would like to thank Jared for his contributions to work presented in Chapter 3. I would also like to thank Amitabh amd Muyiwa for giving several useful suggestions for figures and empirical work carried out in this dissertation. I thoroughly enjoyed collaborating with Thomas Hayes, Jared Saia and Amitabh Trehan on a paper which is not part of this dissertation and I thank them for it.

I would also like to thank Prof Deepak Kapur for showing to me the importance of solving examples and to Prof K. Narayan Kumar from my undergraduate institution for introducing me to the area of theoretical computer science. Cris and Sean, were always available for any questions which I had, or advice which I needed. George and Jeff from tech support, Lynne, Courtney and Lourdes have made my stay here hassle free and comfortable. I thank all of them for helping me out from time to time. Additionally I would like to thank Josep and Stefano for hosting me in Barcelona and Rome respectively, which was one of the best times of my student life.

There were many friends, without whom the journey would have been very boring and difficult. This list is too long to be enumerated fully. I thank Amitabh, Vaibhav, Anuj, Arnab, Rajeev, Niranjan, Manju, Tom, Japji, Animesh, Mukesh, Dhaval, Wenyun, Jack, Gaurav, Sourav, Krishna, Abhishek, Shailendra, Tamanna, Shweta, Pallavi, Srijana and Amanda for encouraging me from time to time and the several moments of enlightening discussions at Dhaka Bazaar and to Kaku and Kakima for making it possible. There are many more in this list but rest assured, you will always remain in my heart and mind.

Last but not the least I would like to thank my family for bearing with me on all difficult times and always encouraging me to pursue challenges. At this moment the contributions of my late father come to my mind, for helping me hone my abilities, both emotional and professional and always coming up with great ways to encourage me to pursue my dreams and recover my confidence after various failures. There is no ability which I have in greater measure than him, and in a way, this PhD is his. I would also like to thank my mother for being with me in Albuquerque for the most productive time of my research career. She contributed immensely with her excellent cooking and emotional support. I would also like to thank my brothers and their families for being available with their love and encouragement at all times.

Security in Network games

by

Navin Rustagi

ABSTRACT OF DISSERTATION

Submitted in Partial Fulfillment of the
Requirements for the Degree of

Doctor of Philosophy
Computer Science

The University of New Mexico

Albuquerque, New Mexico

July, 2010

Security in Network games

by

Navin Rustagi

B.Sc., Chennai Mathematical Institute, 2002

M.Sc., Chennai Mathematical Institute, 2004

Ph.D., Computer Science, University of New Mexico, 2010

Abstract

Attacks on the Internet are characterized by several alarming trends: 1) increases in frequency; 2) increases in speed; and 3) increases in severity. Modern computer worms simply propagate too quickly for human detection. Since attacks are now occurring at a speed which prevents direct human intervention, there is a need to develop automated defenses. Since the financial, social and political stakes are so high, we need defenses which are *provably good* against a worst case attacks and are not too costly to deploy. In this dissertation we present two approaches to tackle these problems.

For the first part of the dissertation we consider a game between an alert and a worm over a large network. We show, for this game, that it is possible to design an algorithm for the alerts that can prevent any worm from infecting more than a vanishingly small fraction of the nodes with high probability. Critical to our result is designing a communication network for spreading the alerts that has high expansion.

The expansion of the network is related to the gap between the 1^{st} and 2^{nd} eigenvalues of the adjacency matrix. Intuitively high expansion ensures redundant connectivity. We also present results simulating our algorithm on networks of size up to 2^{25}.

In the second part of this dissertation we consider the virus inoculation game which models the selfish behavior of the nodes involved. We present a technique for this game which makes it possible to achieve the "windfall of malice" even without the actual presence of malicious players. We also show the limitations of this technique for congestion games that are known to have a windfall of malice.

Contents

List of Figures

Chapter 1

Introduction

In recent times, it has become increasingly challenging to protect the internet from attacks. They have exhibited several trends some of which are: (i) increases in frequency: large-scale attacks are approximately doubling every year [53]; (ii) increases in speed: the recent slammer worm infected 90% of vulnerable hosts within 10 minutes [37]; and (iii) increases in severity: the slammer worm had many unforeseen consequences including failures of 911 emergency data-entry terminals, network outages, and canceled airline flights, [37, 16, 28, 25]. In addition, there has been a broadening of motivations for attack to include extortion [55, 8]; phishing [21, 57, 29]; sending anonymous spam [31, 30]; and political reasons [43, 45]. Modern computer worms simply propagate too quickly for human detection. Since worms spread at a speed which prevents direct human intervention, there is a need to develop automated defenses. Since the financial, social and political stakes are so high, we need defenses which are *provably good* against a worst case attacks.

In developing effective response mechanisms, one can follow several approaches, some of which are: develop better technology for defense, commit more resources to defenses, or collaborate more effectively with our allies, in our defenses. In this dissertation, we focus on the third approach. We studied two problems, which are motivated by the need for developing automated, fast, resource efficient, and provably secure collaborative techniques.

The rest of the chapter is organized as follows: In section 1.1 we motivate the first problem in this dissertation. We also give the model, results overview and related work for the first problem in subsection 1.1.1, 1.1.2 and 1.1.3 respectively. In section 1.2 we motivate our second problem. We present the model, results and related works for this section in subsection 1.2.1, 1.2.3 and 1.2.4 respectively.

1.1 The $Worm\ versus\ Alert$ game

Many worm containment systems are based on a *network centric* approach. In an automatic version of this approach, packet classifiers are deployed which use filter mechanisms dependent on automatically generated *content signatures* of worms. The signatures are generated by identifying common byte strings in suspicious network flows. However, automatic methods for generating content signatures fail to contain polymorphic worms, as worms can use techniques such as encryption or code obfuscation [56] to bypass any filters looking for specific byte strings.

In another approach, semantic or behavioral-based systems analyze behavioral signatures [19, 15, 62, 59]. In particular, they may determine if similar data is being sent from one machine to another [19]; if there is a suspicious sequence of systems calls that is being executed by many machines in the network [62, 15, 14]; or whether any incoming or outgoing network traffic exploits a known vulnerability [59].

Both of the above approaches do not take into account the application level vulnerabilities which worms could exploit. Also both of them are prone to false positives, which then becomes a bottleneck in the deployment of automatic worm detection systems. Work done by Costa et al. in [15], addresses these concerns to some extent by proposing a *host centric* approach. In their system, some nodes in the network run instrumented software to automatically detect a worm, determine which vulnerability the worm exploits, and then generate a short trace of the vulnerability the worm exploits. These nodes are called *detector* nodes. The trace(or proof) is called a self certifying alert(SCA's). Because a SCA is short, it is easily propagated through a network. A machine which has a SCA can generate a filter that blocks infection by analyzing the exploit which the SCA proves exists. Because the SCA focuses on the security flaw exploited by a worm, rather than the textual content of the worm, SCAs can easily be created for polymorphic worms. This approach to counter worms, certainly reduces the number of false negatives to a great extent, and has a negligible rate of false alerts.

In the Vigilante system [14] SCAs can be generated, checked and deployed efficiently. For example, it takes 18 milliseconds to generate a SCA for the Slammer worm, the resulting SCA is 457 bytes long, the time to verify this SCA is 10 milliseconds, and the time to create a filter from the verified SCA is 24 milliseconds. These times for SCA generation, verification and filter creation are on the same scale as the time it takes a worm to infect a machine. Vigilante performs similarly for two other Internet worms, Code Red and Blaster.

A crucial aspect of this approach that we focus on in this dissertation, is the algorithm for diffusion of SCA's through the network. There are certain characteristics we desire in all such systems. Firstly, we want the identity of detector nodes to be hidden from the worms, because then there is an easy worm strategy to infect the network, which is to avoid attacking any of the detector nodes. Secondly, the worms should not be able to use the alert propagation system to carry out Denial of Service Attacks on the nodes of the network. In other words, it is necessary to limit the number of SCA's which an alerted node can flood into the network, and the number of SCA's which a node can receive in a single time step. This can be achieved by making the alert propagation "polite", i.e an alerted nodes can only send SCA's to their neighbors. The distribution of alerts in the Vigilante system satisfies these constraints. The underlying overlay network used for propagating alerts in Vigilante is the Pastry [47] peer-to-peer network. It is shown empirically for this system, that a very small fraction of special detector nodes is enough to ensure that a worm infects no more than 5% of the vulnerable population. While these initial results are promising, several critical problems remain. First, Vigilante requires that the nodes participating in the overlay network all be resistant to infection. Second, Vigilante requires that the topology of the overlay network be hidden from the worm. These two assumptions may hold true for an overlay network owned and operated by a single company, but seem unlikely to hold for a large-scale open source peer-to-peer network. Finally, while the Vigilante systems performs well empirically against currently known worms, the system has no known theoretical guarantees against all worms. In our work, we make none of these two assumptions.

On the other hand, worms are becoming increasingly sophisticated. There is evidence of collaboration between online criminals [20, 53, 64]. In addition there is evidence that worms can spread arbitrarily on the internet [60]. Therefore it is reasonable to assume that the worms would be collaborating, and their propagation would not be constrained by any underlying network.

We will now define the model we studied in this dissertation, which is inspired by the discussion above.

1.1.1 Our Model

In our game initially no nodes are infected or alerted. Each node in the network is a detector node independently with fixed probability γ. The game starts with a single node becoming infected. In every round thereafter, every infected node can send out no more than β worms to other nodes in the network. The alerted node can send out alerts to no more than α neighbors. Here α and β are fixed positive integers. Nodes in the network change state according to the following four rules: 1) If a worm is received by a node that is not a detector and is not alerted, that node becomes infected; 2) If a worm is received by a node that is a detector, it is not infected, instead it becomes alerted; 3) If an alert is received by a node that is not infected, that node becomes alerted; 4) If a worm or an alert is received by a node that is already infected or already alerted, then there is no change in the state of that node. This is a synchronous game. i.e, any message sent out by any node is received by the destination node in the same time step.

An infected node can choose any nodes in the network to send worm messages. In contrast, an alerted node can send alert messages only through a previously determined, overlay network. In other words, the alert-spreading algorithm is "polite" in the sense that it does not bombard arbitrary nodes with alerts unless it knows that they are interested in receiving them. A particularly sophisticated worm may exploit the structure of the overlay network for its own purposes. An edge in this overlay network represents an agreement between two nodes to accept SCAs from each other.

Secondly, we assume that the infected nodes are intelligent, coordinated and essentially omniscient. In other words, the infected nodes know everything except for which nodes are detectors, and the alerted nodes' random coin flips i.e. they know the topology of the overlay network used by the alerts; which nodes are alerted and which are infected at any time; where alerts and worms are being sent; the overall strategy used by the alerted nodes; etc. Moreover, the worm is unconstrained in which nodes it attacks. For example, it could always try to infect nodes which have never been infected before. The alerted nodes are assumed to know nothing about which other nodes are infected or alerted, where alerts or worms are being sent, or the strategy used by the infected nodes.

For the rest of this dissertation we would call this game the *Worm versus Alert* game.

1.1.2 Contributions

We start this section with a discussion on specific questions which motivated our results. First consider the following strategy for worms: since the infected nodes collaborate and are omniscient, they come to know of any detector nodes they might have alerted. In the next time step they attack all the nodes adjacent to the alerted detector node to cut off the spread of SCA's through the network. Based on this strategy the following questions arise. Is there a strategy the alerted nodes could use, which not only will help protect against this possible strategy of the worm, but also against other perhaps more devious strategies. What are the properties of the overlay network, that would help protect the nodes?

Another natural question about this strategy is regarding the robustness of this approach to malfunctioning detector nodes. In the event of a false alert, can we limit the congestion in the network?

The work in this dissertation explores these questions in detail. Given below is a chapter wise breakup of results regarding the Worm vs Alert game.

- **Chapter 2**: Let RANDOM be the algorithm where each alerted node sends to α neighbors selected uniformly at random with replacement. In this chapter, we show that if the alerts propagate on a d-regular graph with expansion constant c, and if $\frac{\alpha}{\beta(1-\gamma)} > \frac{2d}{c}$, then RANDOM ensures, that with high probability, all but a vanishing fraction of nodes get infected as the network size grows.

Intuitively, this shows that the infected nodes can never completely surround the set of alerted nodes, because of the high expansion of the overlay network. We also show that if the overlay network has poor expansion(i.e $\beta(1-\gamma) \geq d$), then there is a strategy where the worms can infect almost all of the non-detector nodes. Next, we give empirical results that suggest that our algorithm for the propagation of alerts combined with techniques like throttling will be useful in current large-scale networks. This work appeared in *Proceedings of the Principles of Distributed Systems; 11th International Conference(OPODIS), 2007* [5].

- **Chapter 3**: False alerts are common in many worm detection systems. In this chapter, we propose a new algorithm where each alert message has a time to live(ttl) field which decides the distance to which an alert can spread. By giving an appropriate value to the ttl parameter, we ensure that a single false alert will not spread to more than polylogarithmic number of nodes. We present an alert propagation algorithm in this chapter which uses a $c_0 \log n$ regular overlay network to spread, for c_0 a constant. This algorithm has two guarantees: 1) it ensures with high probability that under certain constraints on α, β and γ, all but a small fraction of nodes will be alerted when attacked by a worm known to have a lifetime of $O(\log n)$ rounds; and 2) any false alert will not spread to more than polylogarithmic number of nodes. We complement our analysis with empirical simulations of our algorithm against a fixed worm strategy over networks of size about 2^{25}. The work in this chapter is under review.

1.1.3 Related Work

In work done in [38], Moore et al. outline three approaches for anti-worm systems: 1) preventing the attack by reducing the number of vulnerable hosts; 2) treatment based approaches, e.g develop patches and distribute them to infected nodes; and 3) containment based approaches, e.g using worm signatures etc. Since the vulnerable population will always remain due to homogeneous software in hosts, and treatment based approaches take too much time to implement, as they require distributing patches, often with human involvement, containment based approaches seem to be the most viable option.

A standard approach to containing worms, is to blacklist IP addresses which belong to infected host, but this approach has become increasingly ineffective [41, 44]. It is also prone to false alerts. Content based automatic generation of worms is a another alternative for containment of worms. There has been significant success in developing signature generating systems(Vigilante [15, 14], Earlybird [51, 52], Autograph [26], Polygraph [40] and Shield [59]).

We now describe work in distribution of alerts over an overlay network. Zhou et al. [63] propose a system for distributing alerts over a network, but their system is focused on confronting worms that can spread only through the same overlay network through which the alert is spreading. Vojnovic and Ganesh [58] and Shakkottai and Srikant [50] perform exhaustive analytical and empirical studies of the effectiveness of different types of alert dissemination. In work by Vojnovic and Ganesh, they use an hierarchical model for alert dissemination, as opposed to the flat model in Vigilante. Shattokai and Srikant make a case for using P2P systems for patch dissemination, given the exponential data dissemination capabilities of these system. However, both the above works, focuses only on worms that spread uniformly at random in the network.

Many automatic signature based systems assume that worms would be designed to spread very fast. A slow worm may exploit this vulnerability of the detection system. In [49], an approach is discussed for containing both slow and fast worms. In [61], a throttling based approach to slow down the spread of worms is discussed.

1.2 The Virus Inoculation game

In today's world, many large scale networks, are constituted of selfish components that are controlled by different authorities. In such a scenario, each node in the network is selfish and tries to reduce its cost. For the game we discuss in this section, we will assume that the players are selfish, as opposed to the game in the previous section, where players were altruistic. Unfortunately, such a scenario may lead to a large social cost[1] for the whole network and may eventually prove disastrous for each individual node as well. From a game theoretic perspective this phenomenon is often called the Tragedy of Commons effect [23]. To illustrate this effect more concretely, we first begin with some necessary definitions.

Definitions: A game consists of a set of n players, $\{1, \ldots, n\}$. For each player i there is a set of possible strategies or actions S_i. We use $s = (s_1, \ldots, s_n)$, where $s_i \in S_i$ to denote a *c*onfiguration or strategy vector of the game and $S = \times_i S_i$ is the universal set of all possible configurations. Let s_{-i} be the $n-1$ dimensional vector of strategies played by all players other than i. There is a cost function c_i for each player i which is a function from S to $\mathbb{R}$. A pure strategy Nash equilibrium of this game is a strategy vector $s = (s_1, \ldots, s_n)$, s.t $c_i(s_i, s_{-i}) \leq c_i(s'_i, s_{-i})$ for all alternate strategies s'_i of player i. If each player i picks a strategy according to a distribution on S_i, such a choice is called a mixed strategy. There is a notion of mixed strategy Nash equilibrium or simply called Nash Equilibrium, where we assume that players are trying to minimize their expected costs. Every pure strategy nash equilibrium is a mixed strategy nash equilibrium. The social cost of the game is the sum of costs of all individual players. An optimal solution for a game is when a benevolent dictator decides strategies for each player which minimizes the social cost of the game. The ratio of the social cost in the worst nash equilibria to the social cost in the optimal solution is called the Price of Anarchy [27] or PoA. Intuitively, POA measures the

[1] the sum of individual costs

Tragedy of Commons effect.

We now present a concrete example which exhibits the Tragedy of Commons effect. As an example of this effect consider the one round pollution game where there are n players, each of which decide to pollute, which costs zero or to not pollute which costs one. Each player incurs an additional cost equal to the number of players that decide to pollute. In this game it is clear that a Nash equilibria occurs when all the players decided to pollute. The social cost of the game in this case is n^2. On the other hand the best possible solution for this game is when each of players decide not to pollute. Then social cost is then n. The PoA of the pollution game is n.

In this dissertation we study a technique for reducing the PoA in a network security game. We define and motivate our game in the next section.

1.2.1 Motivating the game

How costly is it to deploy an anti-virus system over a network? In networks where there is no centralized authority and nodes are free to choose to inoculate or not, the incentive for the general population to inoculate decreases. Nodes may rely on the fact that many of their neighbors have decided to inoculate.

We study this problem in the context of a virus inoculation game. This game was first presented by, Aspnes, Chang and Yampolskiy in [4]. Since then this game has been studied extensively in the computer science community [34, 35, 33, 24, 22].

Virus Inoculation Game: This game is played out on a graph G=(V,E), which is a two dimensional torus. Each node is a player in this game. Each player has two strategies to choose from, either to inoculate or to not inoculate. The cost incurred to inoculate is 1, and there is no cost of not inoculating. After all the players have decided which actions to perform, a virus attacks one node chosen uniformly at random in G. All nodes which can be reached by this infected node, along paths consisting entirely of uninoculated nodes, are infected with the virus. Every node that gets infected pays a cost of L.

In work done by Moscibroda et al in [39], the price of anarchy for this game is shown to be $(n/L)^{1/3}$, where $n = |V|$. Since n is expected to be much larger than L, this game has a high price of anarchy. Now suppose there are some nodes which turn malicious. The malicious players are not concerned with their welfare and their aim is to always degrade the performance of this game, in this case it is too increase the social cost of this game. They can do this by communicating to non-malicious nodes that they are inoculated, even though they are not. The non-malicious players know the existence of malicious players, but do not know the identity of malicious players. Surprisingly in [39], it is shown that the presence of malicious players may decrease the social cost of the game. Intuitively, this holds since non-malicious players choose their actions based on the perceived threat from the malicious players, and so are likely to inoculate. This phenomenon is called the Windfall of Malice. In this dissertation, we show that we can achieve the windfall of malice effect without the actual presence of malicious players. To achieve this result, we use the concept of mediators described below.

1.2.2 Mediators

We first motivate the notion of a correlated equilibrium which will be necessary for defining mediators. A *correlated equilibrium* is a probability distribution over strategy vectors that ensures that no player has an incentive to deviate. In other words, players have access to a global coin toss in deciding their strategies when implementing a correlated equilibrium.

A mediator in a strategic game is a probability distribution $\mathcal{D}(\mathcal{C})$ over a finite set of different configurations $\mathcal{C}$ that implements a correlated Nash equilibrium. It publicly broadcasts $\mathcal{D}(\mathcal{C})$ together with the corresponding configurations to all strategic players. Moreover, depending on the particular game, to any strategic player it sends a private message containing the proposed strategy for that player.

As an example of a mediator, consider a mediator for the multi round version of the pollution game described earlier in this section. Let the number of rounds be finite but unknown. A valid mediator for this game, can be designed as such: ask all the players to not pollute to begin with and as soon as any one player violates these instructions, ask all the players to start polluting.

The cost of obeying the mediator is one per round for each player and n per round thereafter. So it is in the selfish interests of all the players to follow the mediators advice.

1.2.3 Contributions

We have seen in section 1.2.2 that there is a mediator for multi round games. Can a mediator be designed for single round games?

The virus inoculation game shows an improvement in social cost when there are malicious players in the network. As was described in section 1.2.1, this phenomenon is called the "Windfall of Malice". Can we design mediators in such a way that there is a threat of malicious players, without malicious players being present?

In our research, we address some of these questions. We cover the technical details of our results in Chapter 4.

Chapter 4: In this chapter, we present a mediator which decreases the social cost of the virus inoculation game. In particular we achieve a social cost that is asymptotically optimal. We also show a negative result: our technique cannot be used to decrease PoA in a certain subset of games. The work presented in this chapter was published in the *fifth Workshop on Internet & Network Economics(WINE 09)* [18].

1.2.4 Related Work

The concept of a mediator is closely related to that of a correlated equilibrium, which was introduced by Aumann in [6]. In particular, if a mediator proposes actions to the players such that it is in the best interest of each player to follow the mediators proposal, then the mediator is said to implement a correlated equilibrium. There are several recent results on correlated equilibrium and mediators. The authors in [42] give polynomial time algorithms that can optimize over correlated equilibria, via a LP approach, for a large class of multiplayer games that are "succinctly representable" in the sense that the set of possible strategy vectors over all players is polynomial. Christodoulou et al. [12] study the price of anarchy and stability in congestion games where each edge has a linear cost function with positive coefficients. They show that in such a setting, the price of anarchy for pure equilibrium is almost the same as the price of anarchy of correlated equilibrium: a difference of no more than 1.4%. Balcan et al. [9], describe techniques for moving from a high cost Nash equilibrium to a low cost Nash equilibrium via a "public service advertising campaign". They show that in many games, even if not all players follow instructions, it is possible to ensure such a move . While their result does not explicitly consider mediators, it is similar in flavor to ours in the sense that an outside third party is acting to improve social welfare. A major motivation of our use of a mediator is recent work by Abraham et al. [2], where they shows that it is possible to implement mediators just by having the players talk amongst themselves. In other words, there exists a distributed algorithm for talking among the players that enables the simulation of a mediator. Moreover, they show it is possible to achieve this in a robust manner, even with up to linear size coalitions and up to a constant fraction of adversarial players.

Several recent results study the use of mediators that may act on behalf of a player(see for example [48]). In other words, these results consider the situation where if a player decides to use the mediator, it first communicates any relevant information to the mediator and then the mediator acts for the player, without the player having the opportunity to change the mediators action.

1.3 Structure of the Document

There are four chapters in this dissertation. Chapter 2 and Chapter 3 are self contained presentations of the technical results regarding the Worm vs Alert game. Chapter 4 is a self contained presentation of our results regarding the virus inoculation game. Chapter 5 concludes and suggests areas for future work.

Chapter 2

Worm Vs Alert without Time to Live

In this chapter, we present our results about the game described in Section 1.1.1 in Chapter 1. In the algorithm used to propagate alerts in this chapter, we assume that once a node is alerted, it continues to send out alerts throughout the game. This is different from the algorithm for propagating alerts presented in the next chapter, where alerts have a time to live associated with them. For this chapter, we assume that the detection mechanism is not prone to false alerts, as even a single false alert by a detector node will spread throughout the network, thereby congesting the network.

2.1 Chapter Layout

The model of the the $Worm\ versus\ Alert$ game is the same as is given in Section 1.1.1, and so is omitted here. An overview of our results is described in 2.2. We distribute the main technical work of this chapter in two sections. In section 2.3, we present our upper bound result. In section 2.4, we show that good expansion is a desirable property for the overlay network, and if the overlay network has bad expansion then there is a strategy for the worms, which would almost infect all of the non detector nodes in the network. We then present empirical results in section 2.5 which use our algorithm for propagating alerts. We finish this chapter with conclusions in section 2.6

2.2 Results Overview

In our results, we make use of a d-regular overlay network with node expansion c. As a concrete example, a random d-regular graph has node expansion $c = d/5 - 1$ with high probability[1]. Throughout this chapter, we use the phrase with high probability (w.h.p) to mean with probability at least $1-1/n^{\epsilon}$ for some fixed $\epsilon > 0$. Let RANDOM be the algorithm that has each alerted node in each round send out alerts to α nodes selected uniformly at random without replacement from its neighbors in the overlay. Our main theoretical results are stated below as the following two theorems which are proven in Sections 2.3 and 2.4 respectively.

Theorem 2.3: If $d \geq \alpha$ and $\frac{\alpha}{\beta(1-\gamma)} > \frac{2d}{c}$, then the algorithm RANDOM ensures that, w.h.p, only $o(n)$ nodes are ever infected.

Theorem 2.6: If the overlay network has bounded degree d and $\beta(1-\gamma) > d$, then any alert algorithm in expectation will save a fraction of non-detector nodes that approaches 0 as n gets large

Our empirical results, presented in Section 2.5, show that if the overlay network is a d-regular random graph, as n grows large, the algorithm RANDOM saves an

[1] see [13] for an algorithm for sampling from random d-regular overlay networks in a distributed manner

increasingly large fraction of the nodes against a worm that spreads uniformly at random. For example, for $n = 10^6$, $d = 100$, $\beta = 1$, $\alpha = 5$ and $\gamma = .02$, we were able to save 99% of the nodes on average.

In this work we note that if a detector node generates a single false alert, it propagates throughout the network. We address this problem to some extent in Chapter 3.

2.3 Alert versus worm in an expanding overlay network

In this section, we focus on d-regular graphs for our overlay network. We show that for a suitable choice of parameters and a particular type of overlay network, we are able to save most of the nodes from getting infected with high probability. More precisely, at the end of the process only $o(n)$ nodes get infected, and all other nodes get alerted.

The essential idea is that we want the long-run growth rate of the set of alerted nodes to be higher than the rate for the infected nodes. The rate for infected nodes is easy to calculate; assuming an optimal choice of targets, each infected node infects on average an additional $\beta(1 - \gamma)$ nodes per round. The rate for alerted nodes is trickier, as alerted nodes are limited by the structure of the overlay network. But we can get a lower bound on the expected rate during the early parts of the protocol by observing that A alerted nodes will between them have at most dA neighbors, of which at least cA will not already be alerted, where c is the expansion parameter of the network. It follows that each alerted node will attempt to alert on average at least $\alpha(c/d)$ unalerted nodes at each step. In the absence of the worm, this would give the growth rate of the alerted nodes; with M infected nodes, we must subtract these from the pool of new alerted nodes (using the simplifying assumption that the worm successfully concentrates itself on the boundary of the set A). Fortunately these lost infected nodes are compensated for somewhat by the boost of $\gamma\beta M$ new alerted nodes from triggered detectors.

This overview ignores three important details. Because we want a high-probability bound, it is not enough simply to consider expected growth rates. And because the expansion factor applies only for sets with $n/2$ or fewer elements, we must consider separately the case where the set of alerted nodes is larger. To our knowledge, a differential equation will not be able to model this process, since we are dealing with an adaptive adversary and it can work against any assumption about the growth rates of the processes involved.

We handle both problems by dividing the execution into three phases. Phase I starts with a single infected node and ends when $\ln n$ worm messages have been received by nodes in the network. During this phase we ignore the spread of alerts and content ourselves with getting only the $\Theta(\gamma \ln n)$ alerted nodes that result from successful detections. Phase II starts at the end of of Phase I. During this phase we use the fact that the number of infected and alerted nodes are both $\Omega(\log n)$ to show that both the worm and the SCA propagate at close to the expected rate with high probability; the key point is that when the populations of both are large enough, Chernoff bounds apply to the increases. Phase II ends when n/d^2 nodes have been alerted by the SCA; at this point we can no longer rely on the expansion properties of the network and must resort to a different analysis. Note that there are expansion properties till the end of Phase II. For this analysis, done in Section 2.3.3, we show that in constant number of steps, we would alert n/2 nodes and then after c log(log(n)) further steps we would have only o(n) not alerted or not infected nodes. Thus we would have shown that only o(n) nodes could have been infected and $\theta(n)$ nodes have been alerted.

In the remainder of this section, all lemmas that bound a random variable's value for t rounds hold with probability greater than or equal to $1 - t/n^c$ for some fixed constant $c > 0$. Also for all the remaining lemma's in this section, $d \geq \alpha$.

2.3.1 Phase I

Let Z be the set of nodes that receive the first $\ln n$ worm messages; i.e., the set of nodes that receive worm messages in Phase I.

We write A_t for the number of nodes alerted at time t, counting from the end of Phase I; thus A_0 is the number of nodes alerted in Z.

Lemma 2.1. *At the end of Phase I, (a) the expected number of alerted nodes $\mathcal{E}[A_0]$ is at least $\gamma \ln n$; and (b) for any $c > 0$, there exists a constant $\delta \leq 1/2$, such that with probability greater than $1 - 1/n^c$, $(1 - \delta)\mathcal{E}[A_0] \leq A_0$*

Proof. For each $v \in Z$, let X_v be the indicator random variable for the event that v is alerted in Phase I and let Y_v be the event that v is a detector node. While the X_v are not necessarily independent, we do have that $X_v \geq Y_v$ for all v, and thus $A_0 = \sum_{v \in Z} X_v \geq \sum_{v \in Z} Y_v$. It follows that $\mathcal{E}[A_0] \geq \sum \mathcal{E}[Y_v] = \gamma|Z| = \gamma \ln n$. The second part is an immediate application of Chernoff bounds. ■ □

It follows that A_0 is $\Theta(\ln n)$ with high probability.

2.3.2 Analysis of Phase II

For the second phase, begin by comparing the number of infected nodes in the actual process with the number of infected nodes in an infinite graph where the SCA has no effect on the spread of the worm. The process in the latter graph has the advantage of being much easier to analyze; and, as we show, it gives an upper bound on the outcome of the original process.

Formally, let M_t be the number of infected nodes at time t in the original graph, where as before we count rounds from the start of Phase II. Let M'_t be the number of infected nodes at time t in an infinite graph under the assumptions that (a) no alert messages are ever sent out by the detector nodes, even though they are alerted by worm messages, and (b) each infected node spreads the worm to β unique, previously uninfected nodes in the network at each round. Where no confusion will result, we also use M_t and M'_t to refer to the set of nodes infected in each case.

Observe that the assumptions for M'_t only increase the number of infected nodes; so that M'_t *stochastically dominates* M_t in the sense that $\forall\ k \geq 0$, $Pr(M'_t \geq k) \geq Pr(M_t \geq k)$, no matter what strategy the worm applies in the original graph.

Let M_0 and M'_0 count the nodes infected by the end of Phase I, in their respective simulations. From Lemma 2.1, we have that $M_0 \leq |Z| - A_0 \leq \ln n$.

Lemma 2.2. *For all $t \geq 0$, the expected value of the random variable M'_t at time t is equal to $(1 + \beta(1-\gamma))^t M_0$.*

Proof. By our assumption about the number of messages sent by the infected nodes and the fraction of detector nodes, the expected number of new infected nodes is $\beta(1-\gamma)\mathcal{E}[M'_t]$, where $(1-\gamma)$ is the probability that a given node is not a detector node. Hence the recurrence relation for $\mathcal{E}[M'_t]$ is $\mathcal{E}[M'_t]{=}(1{+}\beta(1{-}\gamma))\mathcal{E}[M'_{t-1}]$. Hence $\mathcal{E}[M'_t] = (1+\beta(1-\gamma))^t M_0$.■ □

We now show that M'_t remains closely bounded around its expected value, thus giving an upper bound on the variable M_t.

Lemma 2.3. *For any $c > 0$ and fixed β and γ, there exists a constant k such that, for sufficiently large n and any t, it holds that $M'_s \leq k\mathcal{E}[M'_s]$ for all $s \leq t$*

Proof. We prove the bound for M'_t, and the bound for M_t follows from the fact that M'_t dominates M_t.

Let $S_t = \min(n - M'_{t-1}, \beta M'_{t-1})$ be the number of nodes that receive the worm message at step t, assuming that no alert messages are ever sent. Let $X(v,t) = 1$ if node v becomes infected for the first time at time t, 0 otherwise. Then for each v in S_t, $\Pr[X(v,t) = 1] = 1 - \gamma$. Define $Y_t = M'_t - M'_{t-1}$, or the number of bad nodes that have been infected at time step t. Clearly $Y_t = \Sigma_{i \in S_t} X(i,t)$, and thus (conditioned on S_t) Y_t has a binomial distribution with expectation $S_t(1-\gamma)$ and variance $S_t\gamma(1-\gamma)$.

Fix some $c' > 0$; from Chernoff's inequality, there exists a constant a such that $\Pr[Y_t > S_t(1-\gamma) + a\sqrt{S_t\gamma(1-\gamma)\log n}] < n^{-c}$.

So with probability at least $1 - n^{-c}$, we have

$$\begin{aligned} M'_t &= M'_{t-1} + Y_t \\ &\leq M'_{t-1} + S_t(1-\gamma) + a\sqrt{S_t\gamma(1-\gamma)\log n} \\ &= M'_{t-1}(1+\beta(1-\gamma)) + a\sqrt{M'_{t-1}\beta\gamma(1-\gamma)\log n}. \end{aligned} \tag{2.1}$$

Observe that the bound (2.1) is an increasing function in M'_{t-1}. It is thus maximized by maximizing M'_{t-1}, and having an upper bound on M'_{t-1} is sufficient to get a (high-probability) upper bound on M_t.

We now proceed by induction on t. Our goal is to show that, with probability at least $1 - tn^{-c}$, for all $s \leq t$,

$$M'_s \leq M_0(1+\beta(1-\gamma))^s \cdot \prod_{i=0}^{s-1}\left(1 + \frac{b}{\sqrt{(1+\beta(1-\gamma))^i}}\right), \tag{2.2}$$

where $b = \frac{a\sqrt{\beta\gamma(1-\gamma)}}{1+\beta(1-\gamma)} = O(1)$. Note that the first two factors give the expected value of M'_t from Lemma 2.2; the product arises from the error term in (2.1). The base case is $t = 0$, where $M_0 \leq M_0$ with probability 1.

Now suppose that (2.2) holds with probability at least $1 - (t-1)n^{-c}$ for $t - 1$; we wish to show that the probability that it suddenly fails for t is at most n^{-c}. First divide (2.2) by $\mathcal{E}[M'_t | M'_{t-1}] = M'_{t-1}(1 + \beta\gamma(1-\gamma))$ to get

$$\begin{aligned}
\frac{M'_t}{E[M'_t | M'_{t-1}]} &\leq 1 + \frac{a}{1+\beta(1-\gamma)} \cdot \sqrt{\frac{\beta\gamma(1-\gamma)\log n}{M'_{t-1}}} \\
&= 1 + b\sqrt{\frac{\log n}{M'_{t-1}}} \\
&\leq 1 + b\sqrt{\frac{\log n}{M_0(1+\beta(1-\gamma))^{t-1}}} \\
&= 1 + b\sqrt{\frac{1}{(1+\beta(1-\gamma))^{t-1}}},
\end{aligned}$$

where b is as in (2.2).

Now use the upper bound on M'_{t-1} from the induction hypothesis to get, with probability at least $1 - tn^{-c}$,

$$\begin{aligned}
M'_t \leq & \left(M_0(1+\beta(1-\gamma))^{t-1} \cdot \prod_{i=0}^{t-2} \left(1 + \frac{b}{\sqrt{(1+\beta(1-\gamma))^i}}\right)\right) \cdot \\
& (1+\beta(1-\gamma)) \left(1 + b\sqrt{\frac{1}{(1+\beta(1-\gamma))^{t-1}}}\right) \\
& = M_0(1+\beta(1-\gamma))^t \cdot \prod_{i=0}^{t-1} \left(1 + \frac{b}{\sqrt{(1+\beta(1-\gamma))^i}}\right)
\end{aligned}$$

as claimed.

To obtain the stated bound, take the logarithm of the correction term in (2.2) to get

$$\sum_{i=0}^{s-1} \log\left(1+\frac{b}{\sqrt{(1+\beta(1-\gamma))^i}}\right) \leq \sum_{i=0}^{s-1} \frac{b}{\sqrt{(1+\beta(1-\gamma))^i}}$$
$$\leq \frac{b}{1-\frac{1}{\sqrt{(1+\beta(1-\gamma))}}} = O(1).$$

Since the constant does not depend on s, we have $M'_s \leq M_0(1+\beta(1-\gamma))^s e^{O(1)} = d\mathcal{E}[M'_s]$ for all $s \leq t$ with probability at least $1 - tn^{-c}$. ■ □

We now turn to alerted nodes. Let A_t be the number of nodes that are in the alerted state at time t. For any set of vertices A, let $N(A)$ be the set of neighbors of nodes in A in the overlay network that are not themselves in A. Let the random variable Z_t be equal to the number of nodes in $N(A_{t-1})$ that receive an alert message at time step t.

Lemma 2.4. *For all* $t \geq 0$, $A_t \geq A_{t-1}$ + Z_t- M'_t

Proof. Out of the unalerted nodes which receive alert messages, at most M'_{t-1} nodes could be infected nodes. Hence the lower bound result holds true. ■ □

Lemma 2.5. *For all* $t \geq 0$, $E(Z_t) \geq (c\alpha/d)A_{t-1}$.

Proof. Let S_{t-1} be the set of nodes that are alerted at time $t-1$ and let $n' = |N(S_{t-1})|$. Number the nodes in $N(S_{t-1})$ from 1 to n'. Let $X_{i,t} = 1$ if the i-th such node is alerted at time step t for the first time, and 0 otherwise. Then $Z_t \geq \sum_{i=1}^{n'} X_{i,t}$. By linearity of expectation, $\mathcal{E}[Z_t] \geq \sum_{i=1}^{n'} \mathcal{E}[X_{i,t}]$. Observe that each node counted in A_{t-1} sends an alert to fixed neighbor with probability α/d; it follows that for each node i in $N(S_{t-1})$, $\Pr[X_{i,t} = 1] \geq \alpha/d$. We thus have $\mathcal{E}[Z_t] \geq n'\alpha/d \geq (c\alpha/d)A_{t-1}$, where c is the expansion factor. ■ □

Lemma 2.6. *For all* $t \geq 0$ $A_t \geq A_{t-1} + (1/2)E(Z_t) - M'_t$.

Proof. We now imagine that the alerted nodes use the following process to decide where to send out their α alert messages. They randomly permute all of their neighbors and then send out alerts to the first alpha nodes in this random permutation. Imagine further that some alerted node j determines its random permutation by assigning a random variable $X_{j,i}$ to each node i that is a neighbor of j. This random variable takes on a value uniformly at random in the real interval between 0 and 1. The nodes that the alert is sent to are thus determined by finding the α random variables among the d whose outcomes are closest to 0. For each node i and j, there is a separate such random $X_{j,i}$ and we note that these random variables are all independent. Let f be a function such that $Z_t = f(X_{1,1}, X_{1,2}, \ldots, X_{m,d})$. We note that f satisfies the Lipchitz condition, i.e $|f(X_{1,1}, X_{1,2}, \ldots, X_{l,p}, \ldots, X_{m,d}) - f(X_{1,1}, X_{1,2}, \ldots, X'_{l,p}, \ldots, X_{m,d})| \leq 1$. This is the case since a change in the outcome of a single $X_{i,j}$ will at most cause one new node to receive an alert and one old node to not receive an alert. Hence we can use Azuma's Inequality to say that $Pr(|Z_t - E(Z_t)|) \geq (1/2)E(Z_t) \leq 2e^{-\frac{(1/4)E(Z_t)^2}{2A_{t-1}d}}$. Since by the previous lemma $E(Z_t) \geq (c\alpha/d)A_{t-1}$, the right hand side is less than or equal to $2e^{-\frac{((c\alpha/d)A_{t-1})^2}{8A_{t-1}d}}$ which is $O(1/n^{k'})$ for some constant $k' > 0$ since A_{t-1} is $\theta(\ln n)$. The lemma then follows by a simple Union bound. ■ □

Let k be the multiplicative constant of the expectation, in the statement of lemma 2.3.

Lemma 2.7. *For all* $t \geq 0$, $A_t \geq (1 + (\alpha c)/(2d))A_{t-1} - k(1 + \beta(1-\gamma))^t \ln n$

Proof. From Lemma 2.5 and Lemma 2.6 we get that the number of nodes alerted at round t follows the inequality $A_t \geq A_{t-1} + (1/2)((c\alpha/d)A_{t-1}) - M'_t$. Hence $A_t \geq (1 + (\alpha c)/(2d))A_{t-1} - M'_t$. By Lemma 2.2 and Lemma 2.3 we know that M'_t is no more than k$(1 + \beta(1-\gamma))^t \ln n$ for t rounds, with probability at least 1-t/n^c. Hence replacing the upper bound value of M_t in the above expression yields the inequality

$A_t \geq (1 + (\alpha c)/(2d))A_{t-1} - k(1 + \beta(1-\gamma))^t \ln n$. ■ □

Let $p = (1 + (\alpha c)/(2d))$, $q = (1 + \beta(1-\gamma))$. Hence the recurrence relation as given in the last lemma is $A_t \geq pA_{t-1} - kq^t \ln n$.

Lemma 2.8. *For all $t \geq 0$, $A_t \geq p^t A_0 - k(q^t + pq^{t-1} + \dots p^t) \ln n$*

Proof. Proof is by induction on t. It is easy to see that the base case holds. Assume that the claim holds for all rounds less than or equal to t-1. Hence $A_t \geq p(p^{t-1}A_0 - k(q^{t-1} + \dots p^{t-1}) \ln n) - kq^t \ln n$. Expanding the algebraic expression, we get the expression in the claim. ■ □

Let $\kappa = p/q$. Then $A_t \geq p^t \ln n - p^t k(1 + 1/\kappa + \dots (1/\kappa)^t) \ln n$. Or

$$A_t \geq p^t(\ln n - k(1 + 1/\kappa + \dots (1/\kappa)^t) \ln n). \tag{2.3}$$

2.3.3 Analysis of Phase III

In this phase, we make use of a graph with two types of expansion. We show below that a random d-regular graph has the types of expansion that we need.

Theorem 2.1. *Let $d \geq 30$ and $\epsilon > 0$, then with high probability, a random d-regular graph G has the following properties*

1. *For any set S such that $\epsilon \log n \leq |S| \leq \frac{n}{d^2}$, $|N(S)| \geq |S|(\frac{d}{5} - 1)$.*
2. *For any set S such that $\frac{n}{d^2} \leq |S| \leq \frac{n}{2}$, $|N(S)| \geq \frac{|S|}{2}$.*

Proof. Recall the following procedure for constructing a graph G that is a random d-regular graph over n nodes. We create a bipartite graph with n nodes of G on the left hand side L and copies of these n nodes of G on the right hand side R. Now assume that we add edges to this graph by finding $d/2$ random perfect matchings (permutations) over n. Finally if merge each node in L with its copy in R, keeping all edges incident to either the node or its copy, we obtain a random d-regular graph over n nodes.

We now analyze the properties of a graph created according to this process. Let $S \subseteq L$ be such that $s = |S| \leq \frac{\alpha}{n}$. For fixed S and T, let $X_{S,T}$ denote the event that all edges from the set S go to the set T. Therefore probability of $X_{S,T}$ is no more than $\left(\frac{t}{n}\right)^{sd/2}$, where $t = |T|$. To see this, order the edges incident to S and note that the probability that the first of these edges falls in T is t/n . Then given that this first edge falls in T, the probability that the second edges falls in T is $\frac{t-1}{n-1} \leq \frac{t}{n}$ and so forth. Let X_s be the event that all edges from *any* set S of size s go to *any* set T of size no more cs. We can bound this probability as follows.

$$\begin{aligned} Pr(X_s) &\leq \binom{n}{s}\binom{n}{cs}\left(\frac{cs}{n}\right)^{ds/2} \\ &\leq \left(\frac{ne}{s}\right)^{s}\left(\frac{ne}{cs}\right)^{cs}\left(\frac{cs}{n}\right)^{ds/2} \\ &\leq \left[\left(\frac{s}{n}\right)^{d/2-c-1} e^{1+c}c^{d/2-c}\right]^{s} \end{aligned}$$

Simplify for the fact that $s \leq \frac{1}{d^2}n$ we have

$$\begin{aligned} Pr(X_s) &\leq \left[\left(\frac{1}{d^2}\right)^{d/2-c-1} e^{1+c}c^{d/2-c}\right]^{s} \\ &\leq \left[\left(\frac{c}{d^2}\right)^{d/2}(d^2e)^{c+1}\right]^{s} \end{aligned}$$

Setting c to be $d/5$, we get that

$$\begin{aligned} Pr(X_s) &\leq \left[\left(\frac{1}{5d}\right)^{d/2} (d^2 e)^{d/5+1}\right]^s \\ &< \left[d^{-d/2+(2/5)d+2}\right]^s \end{aligned}$$

Let $r = d^{-d/2+(2/5)d+2}$ and note that for $d \geq 21$, $r < 1$. We thus obtain that

$$\begin{aligned} \sum_{\epsilon \log n \leq s \leq n/d^2} Pr(X_s) &\leq \sum_{\epsilon \log n \leq s \leq n/d^2} r^s \\ &= O(n^{-\epsilon'}) \end{aligned}$$

Where the last line hold for some $\epsilon' > 0$ and for sufficiently large n since the summation is a decreasing geometric sum and the largest term is $r^{\epsilon \log n}$.

We next show that the second property holds w.h.p. For $n/d^2 \leq s \leq n/2$, we again get that

$$Pr(X_s) \leq \left[\left(\frac{s}{n}\right)^{d/2-c-1} e^{1+c} e^{d/2-c}\right]^s$$

Simplifying for $s \leq n/2$ we have

$$\begin{aligned} Pr(X_s) &\leq \left[\left(\frac{1}{2}\right)^{d/2-c-1} e^{1+c} c^{d/2-c}\right]^s \\ &\leq \left[\left(\frac{c}{2}\right)^{d/2} (2e)^{c+1}\right]^s \end{aligned}$$

Setting c to be $3/2$, we get that

$$Pr(X_s) \leq \left[\left(\frac{3}{4}\right)^{d/2} (2e)^{5/2} \right]^s$$

Let $r = \left(\frac{3}{4}\right)^{d/2} (2e)^{5/2}$ and note that for $d \geq 30$, $r < 1$. We thus obtain that

$$\begin{aligned} \sum_{n/d^2 \leq s \leq n/2} Pr(X_s) &\leq \sum_{n/d^2 \leq s \leq n/2} r^s \\ &= O(n^{-\epsilon'}) \end{aligned}$$

Where the last line hold for some $\epsilon' > 0$ and for sufficiently large n since the summation is a decreasing geometric sum and the largest term is r^{n/d^2}. A final union bound over shows that both the first and second property hold with high probability. □

The following theorem assumes that the overlay network has expansion properties as given in the Theorem 2.1.

Theorem 2.2. *Assume that at some point, the number of alerted nodes is at least n/d^2 and that the number of infected nodes is no more than $n^{1-\epsilon}$ for some $\epsilon > 0$. Then w.h.p, at the end of the process, all but $o(n)$ nodes will be alerted.*

Proof. We call a node a *virgin* node if it is neither alerted or infected. We will show that if initially there are at least n/d^2 alerted nodes and no more than $n^{1-\epsilon}$ infected nodes, that for some fixed constant C, after $C \ln \ln n$ rounds, there will be $o(n)$ virgin nodes. The number of infected nodes increases by no more than a $\beta + 1$ factor in each round. Thus, after $C \ln \ln n$ rounds, the number of infected nodes is no more than $(\beta+1)^{C \ln \ln n} n^{1-\epsilon} = n^{1-\epsilon'}$ for some $0 < \epsilon' < \epsilon$. Thus, if we can show there are $o(n)$ virgin nodes after $C \ln \ln n$ rounds, then it must be true that all but $o(n)$ nodes are alerted.

Our analysis will occur in two phases, first we will show that we need a constant number of rounds to have at least $n/2$ alerted nodes. Then we will show that in $\Theta(\ln \ln n)$ further rounds, the number of virgin nodes will be only $o(n)$.

We first show that the first phase will, w.h.p., take no more than a constant number of rounds. Let A be the set of alerted nodes. By the coupon collectors analysis, we expect any particular node in this set to send out an alert to all of its neighbors in less than $d \ln d$ rounds. Thus, by Markov's inequality, the probability that a particular node in this set has not sent out the alert to all its neighbors in $3d \ln d$ rounds is no more than $1/3$. Hence, by a simple application of Chernoff bounds, w.h.p., at least half of the nodes in A will send out alerts to all their neighbors in $3d \ln d$ rounds. Let A' be this set of nodes that send out alerts to all of their neighbors. Since $|A'| \leq n/2$, we know that $N(A') \geq |A'|/2$. Since every node can receive alerts from at most d unique nodes, this implies that the number of unique non-alerted nodes that receive alert messages is at least $|A'|/2d$. Moreover, since the number of infected nodes is no more than $n^{1-\epsilon'} \leq \frac{|A'|}{6d}$, the number of virgin nodes that receive alert messages is at least $\frac{|A'|}{3d} \geq \frac{|A|}{6d}$. Thus, while $|A| \leq n/2$, in every $3d \ln d$ rounds, the number of alerted nodes increases by a factor of $1 + \frac{1}{6d}$. Since $|A|$ is initially at least n/d^2 and d is a fixed constant, we can say that in $O(1)$ rounds, $|A|$ will be greater than $n/2$.

We now show that the second phase will, w.h.p., take no more than $\Theta(\ln \ln n)$ rounds. Let V be the set of virgin nodes at some round during this phase. Note that $|V| \leq n/2$ since the number of alerted nodes is now greater than $n/2$. We further assume that $|V| \geq n/\ln n$ since if this is not the case, then there are only $o(n)$ virgin nodes and the second phase is thus completed. By the expansion properties of the overlay network, we can thus say that $N(V) \geq \frac{1}{2|V|}$. Further, at least $\frac{1}{3|V|}$ of the nodes in $N(V)$ must be alerted nodes since the number of infected nodes is only $n^{1-\epsilon'}$. Again using the coupon collectors analysis, Markov's inequality and Chernoff bounds, we can say that, w.h.p., in $3d \ln d$ rounds, at least half of these alerted nodes will have sent out alerts to all their neighbors. Thus, after $3d \ln d$ rounds, at least $\frac{1}{6}|V|$ alerted nodes will send alerts to nodes in V. Since each node in V can receive alerts from at most d unique neighbors, the number of virgin nodes that receive alert messages must be at least $\frac{1}{6d}|V|$. Thus, while $|V| \geq n/\ln n$, in every $3d \ln d$ rounds, the number of virgin nodes decreases by a factor of $1-\frac{1}{6d}$. After r rounds, the number of virgin nodes will thus be no more than $\left(1-\frac{1}{6d}\right)^{\frac{r}{3d \ln d}}(n/2) \leq e^{\frac{-r}{18d^2 \ln d}}(n/2)$. This last quantity will be less than $n/\ln n$ provided that $r = (18d^2 \ln d) \ln \ln n$. Thus, we have shown that in $C \ln \ln n$ rounds, the number of virgin nodes will be $o(n)$, the number of infected nodes will be $o(n)$ and all other nodes in the network will be alerted. □

The next theorem is the main result of this section.

Theorem 2.3. *If $d \geq \alpha$ and $\frac{\alpha}{\beta(1-\gamma)} > \frac{2d}{c}$, then the algorithm RANDOM ensures that, w.h.p, only $o(n)$ nodes are ever infected.*

Proof. Since $\frac{\alpha}{\beta(1-\gamma)} > \frac{2d}{c}$, therefore $\frac{\alpha c}{2d} > \beta(1-\gamma)$. Hence $1+\frac{\alpha c}{2d} > 1+\beta(1-\gamma)$, or $p/q > 1$. From equation 2.3 it is clear that $A_t \geq p^t \ln n - 3k \ln n$. Hence $A_t \geq p^t$. Hence for $t \geq log_p n$, $A_t \geq \Omega(n)$. Hence in Phase II, the process cannot last for more that $log_p(n)$ steps. Hence from Lemma 2.3, we know that $M_{log_p(n)} \leq k(1+\beta(1-\gamma))^{log_p(n)}$ with probability greater than $1 - log_p(n)/n^c$. Hence $M_{log_p n} <$ k $q^{log_p(n)}$. Since $p > q$, clearly $M_t = o(n)$ at the end of Phase II. Further it is O($n^{1-\epsilon}$). Now, from Theorem 2.2 , we know that if we have o($n^{1-\epsilon}$) infected nodes at the end of Phase II , we would have at most o(n) infected nodes at the end of the Phase III. ■ □

2.4 Is expansion necessary?

In this section, we consider what happens in graphs with poor expansion properties. In particular, we look at the growth rate of the number of nodes at distance k from some initial point of infection, and show that if this growth rate is small, the worm successfully infects almost every node that does not detect it itself.

For the purposes of this lower bound, we adopt a simplified deterministic version of the model. We proceed in a sequence of rounds starting from the time at which the worm is first detected, and think of the graph as organized in layers V_0, V_1, $\ldots$, where V_0 contains the initial a_0 alerted and b_0 infected nodes, and each V_i is the set of nodes at distance i from this initial set.

We ignore the structure of the interconnections between layers; instead, we allow an SCA that has already alerted a_i nodes in layer V_i to alert any αa_i nodes in layer V_{i+1} in one round. Because the worm can spread without regard to the layer structure, we assume that it can attempt to infect these nodes first; a round thus consists of the worm attempting to infect nodes in layer V_{i+1} followed by the SCA attempting to alert any nodes that are left.

Let b_i be the total number of infected nodes in layer i after round i and let $B_i = \sum_{j=0}^{i} b_j$ be the total number of infected nodes after round i without regard to what layer they are in. The worm can attempt to infect up to βB_i nodes in round $i+1$; of these, $\gamma\beta B_i$ will trigger detectors.

If we similarly let a_i be the number of alerted nodes in layer V_i after round i, then the SCA can attempt to alert αa_i nodes in layer V_{i+1}. But because the worm goes first, there may not be any nodes left to alert.

The overall pattern in round $i+1$ is thus:

1. The worm attempts to infect up to βB_i nodes in layer V_{i+1}, of which $(1-\gamma)\beta B_i$ become infected and $\gamma\beta B_i$ become alerted.

2. The SCA spreads from layer V_i to layer V_{i+1}, yielding an additional $\min(\alpha a_i, |V_{i+1}| - \beta B_i)$ alerted nodes.

This gives us the recurrence

$$\begin{aligned} b_{i+1} &= (1-\gamma)\min\left(|V_{i+1}|, \beta B_i\right) \\ a_{i+1} &= \gamma\min\left(|V_{i+1}|, \beta B_i\right) + \min\left(\alpha a_i, |V_{i+1}| - \beta B_i\right) \end{aligned}$$

Theorem 2.4. *Define a_i, b_i, and V_i as above. Let $|V_0|, |V_1|, \ldots$ be such that, for all $i \geq 0$,*

$$|V_{i+1}| \leq \beta(1-\gamma)\sum_{j=0}^{i} |V_i|.$$

Let $b_0 \geq (1-\gamma)|V_0|$. Then $b_i \geq (1-\gamma)|V_i|$ for all i.

Proof. Straightforward induction on i. The base case is given. For the induction step suppose the claim holds for i. Then we have

$$\begin{aligned} b_{i+1} &= (1-\gamma)\min\left(|V_{i+1}|, \beta B_i\right) \\ &= (1-\gamma)\min\left(|V_{i+1}|, \beta\sum_{j=0}^{i} b_j\right) \\ &\geq (1-\gamma)\min\left(|V_{i+1}|, \beta(1-\gamma)\sum_{j=0}^{i} |V_j|\right) \\ &= (1-\gamma)|V_{i+1}|. \end{aligned}$$

□

In other words, if the growth rate of the graph is small enough and the initial set of alerted nodes is small enough, then the SCA has no effect beyond the original detection sites.

For a large enough graph, a higher initial growth rate or lower initial worm numbers can be compensated for in the limit. For simplicity, we consider an *infinitely large* graph that is again organized into layers $V_0, V_1, \ldots$ as above.

Theorem 2.5. *Let a_i, b_i, V_i be as in Theorem 2.4. Let $b_0 > 0$ and let*

$$\limsup_{i\to\infty} \frac{|V_{i+1}|}{\sum_{j=0}^{i} |V_i|} < (1-\gamma)\beta. \tag{2.4}$$

Suppose further that $|V_{i+1}| \geq |V_i|$ for all i. Then

$$\lim_{i\to\infty} \frac{b_i}{|V_i|} = (1-\gamma).$$

Proof. We assume that α is sufficiently large that at the end of round i, any node in layer i that is not infected is alerted. This assumption only hurts the worm, so if the assumption is violated the result only improves.

From (2.4), there exists some ϵ, i_0 such that for all $i > i_0$, $|V_{i+1}| \leq (1-\epsilon)(1-\gamma)\beta \sum_{j=0}^{i} |V_j|$. Let $r_i = B_i / \sum_{j=0}^{i} |V_j|$ and compute, for $i > i_0$,

$$\begin{aligned}
b_{i+1} &= (1-\gamma)\min\left(|V_{i+1}|, \beta B_i\right) \\
&= (1-\gamma)\min\left(|V_{i+1}|, \beta r_i \sum_{j=0}^{i} |V_i|\right) \\
&= \min\left((1-\gamma)|V_{i+1}|, r_i \beta (1-\gamma) \sum_{j=0}^{i} |V_i|\right) \\
&\geq \min\left((1-\gamma)|V_{i+1}|, \frac{r_i}{1-\epsilon}|V_{i+1}|\right) \\
&= \min\left(1-\gamma, \frac{r_i}{1-\epsilon}\right)|V_{i+1}|.
\end{aligned}$$

Unless $r_i = 1-\gamma$, we expect $b_{i+1}/|V_{i+1}|$ to be larger than r_i; in particular we have $b_{i+1}/|V_{i+1}| \geq \min((1-\gamma),(1+\epsilon)r_i)$. The new ratio r_{i+1} is a weighted average of r_i and b_{i+1}/V_{i+1}. Under the assumption that $|V_i|$ is nondecreasing, the weight on the second term is at least $1/(i+1)$. Thus we have

$$r_{i+1} \geq \frac{i}{i+1} r_i + \frac{\min(1-\gamma, \epsilon r_i)}{i+1} = \qquad r_i + \frac{\min((1-\gamma)-r_i, \epsilon r_i)}{i+1}.$$

Observe that the first term in the minimum is decreasing and the second increasing. As long as $\epsilon r_i < (1-\gamma)r_i$, we have $r_{i+1} \geq r_i \frac{\epsilon}{i+1}$. So $r_{i+k} \geq r_i \left(1 + \epsilon \sum_{j=i}^{k-1} \frac{1}{j+1}\right)$; as the series diverges, eventually r_{i+k} must be large enough that the first term takes over. But then let $s_i = (1-\gamma) - r_i$, and compute $s_{i+1} = (1-\gamma) - r_{i+1} \leq s_i - \frac{s_i}{i+1} = s_i \frac{i}{i+1}$, from which it follows via a telescoping product that $s_{i+k} \leq s_i \frac{i}{i+k}$, which goes to zero in the limit. ■ □

The proof of the following theorem follows directly from the above.

Theorem 2.6. *For a graph with bounded degree d, we have* $|V_{i+1}| \leq d \sum_{j=1}^{i} |V_j| + 1$. *So if* $(1-\gamma)\beta > d$ *we expect almost no non-detector nodes to be alerted.*

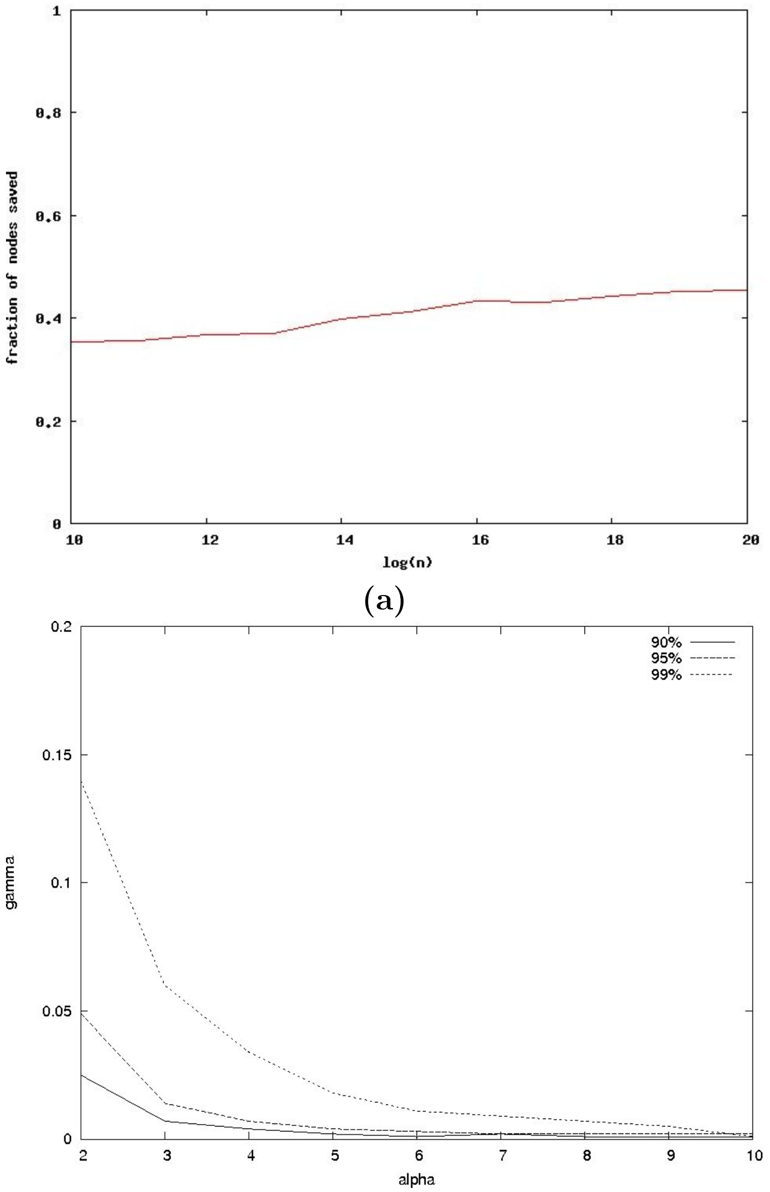
2.5 Empirical Results

We simulated the spread of a worm and an alert through a network to empirically determine the fraction of nodes saved.[2] We performed our experiment using a random d-regular graph as the overlay network and set each node in the network to be a detector node independently with probability γ. In addition, we fixed the worm strategy such that each infected node, in each round, sent out the worm to β unique nodes selected uniformly at random, and we fixed the alert strategy such that each alerted node sent out the alert to α unique nodes selected uniformly at random among its neighbors in the overlay network. We note that the worm strategy we used in these experiments is not necessarily the best possible worm strategy, but we selected this strategy for concreteness. Our d-regular random graph was created using the configuration model method proposed in [10].

In each round we iterate through the set of vertices, allowing each infected or alerted node to send the worm or alert to the appropriate number of other nodes in the network. There are several possible strategies for resolving the status of a virgin (i.e. neither alerted or infected) node that gets both a worm message and an alert message in the same round. In our previous theoretical analysis, we assumed that if a node receives just one worm message it becomes infected. However, in our experiments, we used the somewhat more relaxed and realistic assumption that the probability that the node gets infected equals the number of worm messages received divided by the total number of messages received, and that the probability the node becomes alerted is 1 minus this quantity. We note that this assumption is equivalent to assuming that the messages all arrive in the node's message queue according to some random permutation.

Figure 2.1(a) illustrates our results when $\gamma = 0.1$, $\beta = 1$, $\alpha = 1$ and $d = 10$, where

[2] All of the code necessary to replicate these experiments is available at `http://www.cs.unm.edu/~navin/worm.html`.



(a)

(b)

Figure 2.1: (a) log of the network size versus fraction of nodes saved (b) contour plot of α versus γ required to save 99%, 95% and 90% of the nodes.

we varied the value of n from 2^{10} to 2^{20}, multiplying at each step by 2. To remove noise in the simulation, each data point represents the average over 100 trials. The best result we obtained was saving only 45% of the nodes for $n = 2^{20}$. Even though this final data point is somewhat disappointing, we do observe a clear increasing trend in the fraction saved as n increases.

Given these results, it seems for current network sizes, there is not much hope for the alert when $\alpha = \beta$. We thus next considered the case where $\alpha > \beta$. In practice, this condition may hold since the alerts are traveling through a predetermined overlay network and a technique such as throttling can ensure that alert messages received through the overlay are given priority over types of messages. To explore this scenario, we conducted experiments where we fixed β at 1. We then determined necessary values of γ for each α ranging from 2 to 10, that would ensure that we save 90%, 95% and 99% of the nodes (Figure 2.1(b)). The values of n and d used in the experiment were 10^6 and 100 respectively. The results of these experiments were much more encouraging. In particular, for $\alpha = 2$, we were able to save 99% of the nodes with $\gamma = .14$. When $\alpha = 5$, we required a γ of .018 to save 99% of the nodes, and when $\alpha = 10$, we required a γ of only .001 to save 99% of the nodes. These results suggest that our algorithms for spreading alerts might be most effective in conjunction with other techniques (like throttling) that would enable the alerts to spread more quickly than the worm.

2.6 Conclusion

We have described a simple distributed algorithm for spreading alert messages through a network during a worm attack and have proven that this algorithm protects all but a vanishingly small fraction of the network provided that the alerts spread through an overlay network with sufficiently good node expansion. Our algorithm is provably good no matter what strategy the worm uses to spread through the network. We have demonstrated empirically that this algorithm works effectively against a randomly spreading worm under conditions that may be reasonable for modern computer networks. Finally, we have shown that if the overlay network has poor expansion, then the worm will likely infect almost all of the non-detector nodes in the network.

Chapter 3

Handling False Alerts in the *Worm versus Alert* Game

There has been significant success in developing signature generating systems(Vigilante [15, 14], Earlybird [51, 52], Autograph [26], Polygraph [40] and Shield [59]), but there has been little focus on designing distributed algorithms for effective deployment of worm signatures. In this chapter, we present an algorithm which can be used for distributing many types of worm signatures. In particular, this algorithm that can be used with *any* of the above signature generation systems. Moreover, our algorithm can be used in a network where many *different* signature generation systems are being used concurrently. We argue that this last property is particularly important, since a network with many different signature generation systems is more likely to catch new worms. Finally, our algorithm has provable guarantees on how quickly the signatures will be deployed, no matter what strategy the worm uses to try to infect the network (see Section 3.2).

A critical problem in creating a signature deployment algorithm is the problem of *false alerts*. Many worm signature systems sometimes falsely generate signatures for traffic that is not malicious. Propagating such a false alert through the entire network consumes significant network resources and so is extremely undesirable. There are several conceivable ways of dealing with false alerts. In this chapter, we take a dampening approach. We assume that periodically any particular detector node may misfire and generate a false alert. Our goal is to minimize how far these false alerts can propagate. In particular, we demand that every time that a particular detector node misfires and sends out a false alert, that that false alert will spread to no more than a polylogarithmic[1] number of other nodes in the network.

A weakness of our approach is that if there are faulty detector nodes in the network, we do not completely prevent those detectors from sending false alerts. However, we do constrain the network so that the false alerts are only spread to the nearby neighbors of a faulty detector. Arguably, it is easier for these nearby neighbors to track down and fix the faulty detector than for nodes that are further away in the network. A strength of our approach is that it allows for many different types of detector nodes. In particular, for any given worm, if some constant fraction of the nodes in the network are detector nodes that are able to recognize that particular worm, then our algorithm has provable guarantees for protecting the network.

3.1 Chapter Layout

The rest of the chapter is organized as follows. The model for the game is the same as is given in Section 1.1.1 in Chapter 1 and is therefore omitted here. We then give an overview of our results in section 3.2. We give detailed proofs of our theoretical results in section 3.4 with an overview of our analysis in section 3.3. We then present our empirical results in section 3.5. We end this chapter with conclusions and future

[1] Polylogarithmic means $O(\log^c n)$ for some constant c. We stress that this value is quite small, for example, it is asymptotically much less than $O(\sqrt{n})$.

work in section 3.6

3.2 Results Overview

Throughout this chapter, we use the phrase with high probability (w.h.p) to mean with probability at least $1 - 1/n^c$ for some fixed $c > 0$. Our main algorithm for propagating an alert is presented as Algorithm 1. In this algorithm all nodes and all alerts messages have a time to live(ttl) field. Intuitively, the ttl field of an alert bounds how far that alert can propagate, and the ttl of a node v determines how long v will propagate alerts. The parameter τ in Algorithm 1 is a user specified parameter giving the maximum value of any ttl field.

Our theoretical and empirical results suggest that our algorithm for spreading alerts is most effective in conjunction with techniques like throttling that enable alerts to be sent more quickly than worms, that ensure that $\alpha > \beta$. The fact that alerts spread only through the special overlay network, where they can be given priority over other messages, might facilitate a throttling approach.

3.2.1 Theoretical Results Overview

The main theorem of this chapter is stated below and proved in the section 3.4. Algorithm 1 will be used to achieve this result.

Theorem 3.1. *Let α, β, γ be fixed constants, with $\alpha, \beta \in \mathbb{Z}$ and assume that the overlay network is a $c_0 \log n$ regular graph, for some $c_0 \geq 5$, and alerts propagate according to Algorithm 1. Let $p = (1 + \frac{\alpha((c_0 \log n)/5-1)}{2c_0 \log n})$ and $q = (1 + \beta(1 - \gamma)$. Then if*

- $p \geq 2q^2$

Algorithm 1 Alert propagation for a node v

```
$ttl(v) \leftarrow 0$
for each round do
  $ttlmsgs \leftarrow$ Maximum ttl in all msgs received.
  $ttl(v) \leftarrow Max(ttlmsgs, ttl(v)) - 1$
  if $v$ is a detector node and it received a worm then
    $ttl(v) \leftarrow \tau$
  end if

  if $ttl(v) > 0$ then
    send out messages with ttl field equal to $ttl(v)$ to $\alpha$ neighbors chosen uniformly at random in the overlay network.
  end if
end for
```

- $\tau = c_2 \log\log n$ *for some fixed constant* c_2 *to be determined later.*

then

- *w.h.p in* $O(\log n)$ *steps after the start of the an infection only o(n) nodes will be infected.*

- *Each false alert propagates to at most polylogarithmic number of nodes.*

Note that with $\tau = \theta(\log\log n)$, the false alerts cannot spread to more than polylogarithmic number of nodes.

3.2.2 Empirical Results Overview

We empirically evaluated our algorithm for networks of size approximately equal to 2^{25}. We considered a worm strategy where each infected node chooses β nodes uniformly at random from the network to send a worm message in each round and the alerts follow algorithm 1. A sequence of 6 snapshots of the game as played out for 500 nodes with degree 10 and $\alpha = 4$, $\gamma = 0.15$, $\beta = 1$ and $\tau = 3$ is shown in Figure 3.1.

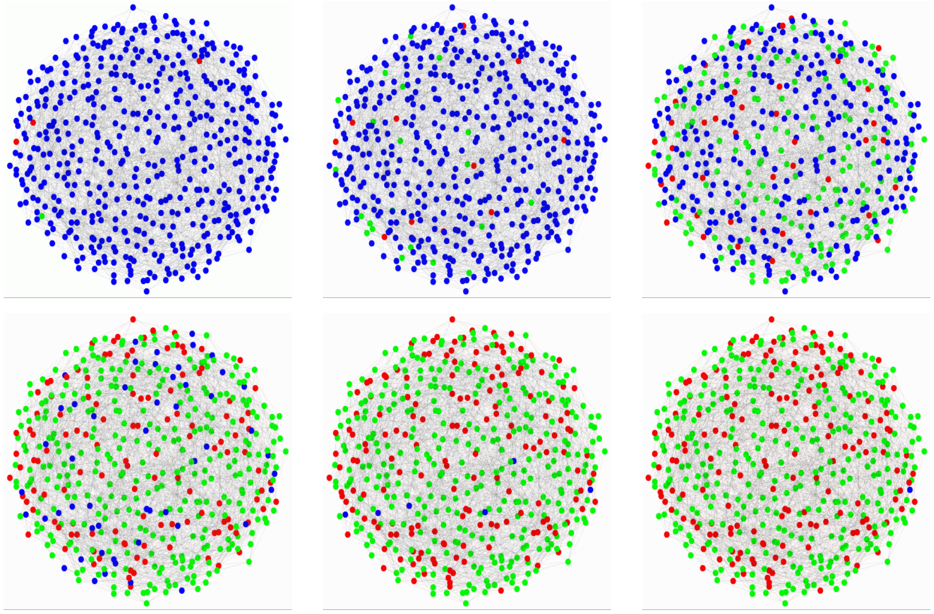


Figure 3.1: (Best viewed in color)A sequence of 6 snapshots of the game at rounds 2, 4, 7, 12, 18 and 26. Red nodes are infected nodes, green are alerted nodes and blue are neither infected nor alerted.

In this particular run of the game, the spread of alerts from a specific detector node is captured in Figure 3.2. The root of this tree is the detector node and each edge is labeled by the ttl values that the alert message carries. In the first round after being alerted the detector node is successful in alerting four neighbors. In the next round it alerts two more nodes with alert messages with ttl value two. The right most node at depth two in the tree had been alerted with ttl value two, but all the neighbors it chose to send alerts to in successive rounds were already infected, hence it could nor propagate the alerts any further.

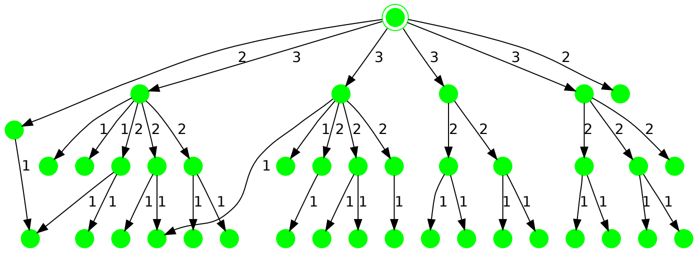


Figure 3.2: (Best viewed in color)A false alert spreading from a detector node for $\tau = 3$, $\alpha = 4$.

In section 3.5 we describe results from simulations for network sizes as large as 2^{25}. For $n = 10^7$, $\gamma = 0.02$, $\beta = 2$, $\alpha = 10$, $\tau = 4$, Algorithm 1 is able to save 93% of the nodes in the network.

3.3 Analysis Overview

In [5], we present an algorithm and constant degree overlay network, which under certain conditions w.h.p ensures that at most o(n) nodes are infected by any worm. However this past result suffers from the weakness that a single false alert can propagate through the entire network. In contrast, in this chapter, any false alerts can only spread to at most a polylogarithmic number of nodes.

Unfortunately, with the restriction imposed by the ttl mechanism on the alerts, we cannot use the lower bound on the number of alerts in [5], as any alert stops propagating once its ttl expires. However for τ rounds from the time a detector has been alerted, we can adapt the analysis for the lower bound on the growth of alerts over a $\theta(\log n)$ degree network from the corresponding lower bounds given in [5]. We divide our analysis in four phases and the number of nodes alerted, infected at the end of each phase and the duration in that phase are given in the table in Figure 3.3. In the remainder of this section, we outline our analysis of the four phases given in this table. In this table we let $q = (1 + \beta(1 - \gamma)$ be an upper bound on the expected rate of increase of infected nodes. We let $p = (1 + \frac{\alpha((c_0 \log n)/5-1)}{2c_0 \log n})$. We let $p_1 = (1 + (1 - e^{-\alpha})/4)$. Let $r = \log_{p_1} q$ The constants p and p_1 are the lower bounds on the expected rate of spread of alerts in Phase 2 and Part 1 of Phase 3 respectively.

Phase	Alerted	Infected	Duration
1	$\theta(\frac{n}{\log^{2r+2} n})$	$\theta(\frac{n}{\log^{2r+1} n})$,	$O(\log n)$
2	$\geq n/(3\lambda)$	$O\left(\frac{n}{\log^{r+1+(1-\frac{r+1}{\log_{p_1} p})r-\frac{r^2}{\log_{p_1} p}} n}\right)$	$\theta(\log \log n)$
3	$\geq n/2$	$O\left(\frac{n}{\log^{r+1+(1-\frac{r+1}{\log_{p_1} p})r-\frac{r^2}{\log_{p_1} p}-r} n}\right)$	$\theta(\log \log n)$
4	$n - o(n)$	$O\left(\frac{n}{\log^{r+1+(1-\frac{r+1}{\log_{p_1} p})r-\frac{r^2}{\log_{p_1} p}-r} n}\right)$	$\theta(1)$

Figure 3.3: Phase, # of alerted nodes at the end of the phase, # of infected nodes at the end of the phase, duration of that phase

We now outline our analysis of Phase 1. We define nodes which are neither infected nor alerted to be *virgin* nodes. We define a *small step round* to be a round when the number of virgin nodes that receive worm messages is no more than $\frac{1}{\log_{p_1} n}$ times the number of infected nodes at the end of the previous round. A round which is not a small step is called a *large step round.* It is clear(see lemma 3.1) that for a worm to take over the network in $O(\log n)$ time steps, there has to be a large step round after $\frac{n}{\log_{p_1}^{2r+1} n}$ nodes have been infected and before $\kappa_0 \frac{n}{\log_{p_1}^{2r+1} n}$ nodes have been infected, where κ_0 is a the constant used in Lemma 3.1. Phase 1 is defined to end at the end of the first large step round which occurs after $\frac{n}{\log_{p_1}^{2r+1} n}$ nodes have been infected. In our subsequent analysis we show that the number of detectors nodes alerted in the last round of Phase 1, will be successful in alerting all but $n - o(n)$ nodes in the network in τ more rounds. Since we start counting the number of alerted nodes at the end of Phase 1, we define our rounds to begin at the end of Phase 1. Let A_i and M_i be the number of alerted nodes and infected nodes at the end of round i.

Let the vertex expansion in Phase 2 be called λ. In Phase 2, we use the fact that our $c_0 \log n$ regular network has the following property w.h.p: for sets of size less than $n/(3\lambda)$, there is $\theta(\log n)$ vertex expansion. The alerted nodes uses this high expansion and the fact that $p > 2q^2$ to catch up and then overtake the number of infected nodes. We adapt Lemma 8 in [5] to get the following lower bound on the set of alerted nodes A_t at round r_t:

Lemma 3.4. W.h.p for all $t \geq 0$, s.t $A_{t-1} < n/(3\lambda)$, $A_t \geq p^t(A_0 - K)$ for some fixed constant K.

We then use the above lemma to get an upper bound on the number of rounds spent in Phase 2. We prove that, with $\theta(\log n)$ expansion, the alerts need less than τ rounds to alert $n/(3\lambda)$ nodes, whereas in the same number of rounds the worm can infect asymptotically fewer nodes.

In Phase 3, we do not have the same guarantees on vertex expansion as in Phase 2, since the number of nodes that have been alerted exceeds $n/(3\lambda)$. We now make use of the following property(Lemma 3.7) which holds w.h.p: for sets S of size less than n/3, the number of edges with one endpoint in S, and one outside S is greater than or equal to $(|S| c_0 \log n)/4$. We break our analysis of Phase 3 in 2 parts. The first part begins immediately after the end of Phase 2. We show in Lemma 3.9, that for this part the expected rate of spread of alerts is at least $p_1 \geq (1 + (1 - e^{-\alpha})/4)$. Then by an analysis similar to that of Phase 2, we show that the maximum number of rounds taken in Part 1 of Phase 3 is not enough for the infected nodes to break that o(n) barrier. Part 1 is defined to end at the first round when the number of alerted nodes is at least $n/3$. Part 2 of Phase 3 begins immediately after Part 1 ends. The main lemma of part 2 is the following

Lemma 3.13. On a random d-regular graph, if the number of infected nodes is $o(n)$ and the number of alerted nodes is greater than $n/3$, then the maximum number of rounds required for the number of alerted nodes to exceed $n/2$ is a constant w.h.p.

To prove this lemma, we show that in each round a constant fraction of the total number of nodes are being newly alerted w.h.p. Phase 3 is defined to end at the round when the number of alerted nodes is at least n/2.

At the beginning of Phase 4, there are less than $n/2$ virgin nodes, and greater than $n/2$ alerted nodes. In our analysis in Phase 4 we make use of the high edge expansion from Phase 3. We make use of the fact that once at least half of the nodes in the network have been alerted, then due to the high edge expansion from the set virgin nodes, a constant fraction of virgin nodes will become alerted in each round. Thus w.h.p, it takes constant number of steps to alert all but n-o(n) nodes.

The proof of our main theorem, i.e Theorem 3.1 is derived from Lemma 3.2, Lemma 3.6, Lemma 3.12, Lemma 3.13, and Lemma 3.14.

3.4 Proofs

3.4.1 Large Step Round

In this section we present detailed proofs of the analysis discussed in the previous section.

In the next two sections we present the detailed analysis of the first phase.

Lemma 3.1. *Let t be the first round when the number of infected nodes is at least $\frac{n}{\log_{p_1}^{2r+1} n}$.*

If the worm infects $\theta(n)$ nodes within $O(\log n)$ rounds, there must be a large step round after round t and before $\kappa_0 \frac{n}{\log_{p_1}^{2r+1} n}$ nodes are infected, for some constant κ_0.

Proof. Suppose not. Then the number of infected nodes is bounded by $\beta \frac{n}{\log_{p_1}^{2r+1} n}(1 + 1/\log_{p_1} n)^{O(\log n)-t} \leq \beta \frac{n}{\log_{p_1}^{2r+1} n}(1 + 1/\log_{p_1} n)^{O(\log n)} \leq \beta \frac{n}{\log_{p_1}^{2r+1} n} e^{\frac{O(\log n)}{\log_{p_1} n}} \leq \kappa_0 \frac{n}{\log_{p_1}^{2r+1} n}$ for some constant κ_0. This last quantity is clearly $o(n)$. □

We define Phase 1 to end at the first large step round after round t.

For the sake of simplicity, henceforth we will call the i th round after the end of phase 1 as round t_i. Let the number of nodes alerted by the end of this round be called A_i.

3.4.2 Phase 1

Lemma 3.2. *At the end of Phase 1, (a) the expected number of alerted nodes $E[A_0]$ is at least $\gamma(\frac{n}{\beta \log_{p_1}^{2r+2} n})$; and (b) for any $c > 0$, there exists a constant $\delta \leq 1/2$, such that with w.h.p, $A_0 \geq (1 - \delta)E(A_0)$.*

Proof. We note that the number of infected nodes at the beginning of the last round of phase 1 is at least $\frac{n}{\beta \log_{p_1}^{2r+1} n}$. Let Z be the nodes which are sent worm messages during the last round of Phase 1. The number of such nodes is greater than $\frac{n}{\beta \log_{p_1}^{2r+2} n}$, because the last round is a large step round. For each $v \in Z$, let Y_v be the event that v is a detector node. Thus $A_0 = \sum_{v \in Z} Y_v$. It follows by linearity of expectation that $E[A_0] \geq \sum E[Y_v] \geq \gamma |Z| = \gamma(\frac{n}{\beta \log_{p_1}^{2r+2} n})$. The second part of the lemma is an immediate application of Chernoff bounds. □

We now present the lower bound on the number of alerts which would be used in the analysis of Phase 2.

3.4.3 Lower bound on the number of Alerts

For the sake of brevity in representation we let $d{=}c_0 \log n$. These lemmas will be used for proofs in Phase 2 and Phase 3.

Let N(S) be the set of neighbors for set S.

Lemma 3.3. *Let $d \geq 30$ and $\epsilon > 0$, then with high probability, a random d-regular graph G has the following properties*

1. *For any set S such that $\epsilon \log n \leq |S| \leq \frac{n}{3\lambda}$, $|N(S)| \geq |S|(\frac{d}{5} - 1)$.*
2. *For any set S such that $\frac{n}{3\lambda} \leq |S| \leq \frac{n}{2}$, $|N(S)| \geq \frac{|S|}{2}$.*

Proof. The proof of the above lemma is very similar to the proof of Theorem 2.1 in Chapter 2, however for completeness we redo the proof with the appropriate changes.

Recall the following procedure for constructing a graph G that is a random d-regular graph over n nodes. We create a bipartite graph with n nodes of G on the left hand side L and copies of these n nodes of G on the right hand side R. Now assume that we add edges to this graph by finding $d/2$ random perfect matchings (permutations) over n. Finally if we merge each node in L with its copy in R, keeping all edges incident to either the node or its copy, we obtain a random d-regular graph over n nodes.

We now analyze the properties of a graph created according to this process. Let $|S| = s$. For fixed S and T, let $X_{S,T}$ denote the event that all edges from the set S go to the set T. Therefore Probability of $X_{S,T}$ is no more than $\left(\frac{t}{n}\right)^{sd/2}$, where $t = |T|$. To see this, order the edges incident to S and note that the probability that the first of these edges falls in T is t/n . Then given that this first edge falls in T, the probability that the second edges falls in T is $\frac{t-1}{n-1} \leq \frac{t}{n}$ and so forth. Let X_s be the event that all edges from *any* set S of size s go to *any* set T of size no more λs. We can bound this probability as follows.

$$\begin{aligned} Pr(X_s) &\leq \binom{n}{s}\binom{n}{\lambda s}\left(\frac{\lambda s}{n}\right)^{ds/2} \\ &\leq \left(\frac{ne}{s}\right)^{s}\left(\frac{ne}{\lambda s}\right)^{\lambda s}\left(\frac{\lambda s}{n}\right)^{ds/2} \\ &\leq \left[\left(\frac{s}{n}\right)^{d/2-\lambda-1} e^{1+\lambda}\lambda^{d/2-\lambda}\right]^{s} \end{aligned}$$

Simplify for the fact that $s \leq \frac{1}{3\lambda}n$ we have

$$\begin{aligned}
Pr(X_s) &\leq \left[\left(\frac{1}{3\lambda}\right)^{d/2-\lambda-1} e^{1+\lambda}\lambda^{d/2-\lambda}\right]^s \\
&\leq \left[\left(\frac{1}{3}\right)^{d/2-\lambda-1} (e)^{\lambda+1}\lambda^{d/2-\lambda-d/2+\lambda+1}\right]^s \\
&\leq \left[\left(\frac{3}{3^{d/2-\lambda}}\right) 3^{\lambda+1}\lambda\right]^s \\
&= 3\lambda e \frac{1}{3}^{d/2-2\lambda}
\end{aligned}$$

Setting $d = 5\lambda$, we get that $Pr(X_s)$ tends to zero for sufficiently large n.

We next show that the second property holds w.h.p. For $n/3\lambda \leq s \leq n/2$, we again get that

$$Pr(X_s) \leq \left[\left(\frac{s}{n}\right)^{d/2-\lambda-1} e^{1+c}e^{d/2-\lambda}\right]^s$$

Simplifying for $s \leq n/2$ we have

$$\begin{aligned}
Pr(X_s) &\leq \left[\left(\frac{1}{2}\right)^{d/2-\lambda-1} e^{1+\lambda}\lambda^{d/2-\lambda}\right]^s \\
&\leq \left[\left(\frac{\lambda}{2}\right)^{d/2} (2e)^{\lambda+1}\right]^s
\end{aligned}$$

Setting λ to be $3/2$, we get that

$$Pr(X_s) \leq \left[\left(\frac{3}{4}\right)^{d/2} (2e)^{5/2}\right]^s$$

Let $r = \left(\frac{3}{4}\right)^{d/2}(2e)^{5/2}$ and note that for $d \geq 30$, $r < 1$. We thus obtain that

$$\begin{aligned}\sum_{n/d^2 \leq s \leq n/2} Pr(X_s) &\leq \sum_{n/d^2 \leq s \leq n/2} r^s \\ &= O(n^{-\epsilon'})\end{aligned}$$

Where the last line holds for some $\epsilon' > 0$ and for sufficiently large n since the summation is a decreasing geometric sum and the largest term is r^{n/d^2}. A final union bound over shows that both the first and second property hold with high probability. □

The proof of the following lemma derives from lemma 3.3. Also this lemma is an adaptation of statement 2.3 and is stated here without proof.

Lemma 3.4. *W.h.p for all $t \geq 0$, s.t $A_{t-1} < n/(3\lambda)$, $A_t \geq p^t(A_0 - K)$ for some fixed constant K.*

Lemma 3.5. *If $r_1 > 0$ with $A_{r_1-1} < n/(3\lambda)$, then w.h.p $r_1 \leq \lceil \log_{p_1}(\frac{A_{r_1}}{A_0-K})/\log_{p_1} p \rceil$ for some constant K.*

Proof. Since $|A_{r_1-1}| < n/(3\lambda)$, we know from lemma 3.4 that w.h.p $A_{r_1} \geq p^{r_1}(A_0 - K))$ for some fixed constant K. Therefore, w.h.p $r_1 \leq \lceil \log_{p_1}(\frac{A_{r_1}}{A_0-K})/\log_{p_1} p \rceil$. □

We present the detailed analysis of Phase 2 below.

3.4.4 Phase 2

By Lemma 3.2, the number of alerted nodes at the beginning of Phase 2 w.h.p is lower bounded by $(1-\delta)\gamma\frac{n}{\beta \log_{p_1}^{2r+2} n}$. Let Phase 2 be defined to end at the first round when the number of alerted nodes is at least $n/(3\lambda)$.

Lemma 3.6. *W.h.p the number of infected nodes at the end of Phase 2 is* $O\left(\frac{n}{\log_{p_1}^{r+1+(1-\frac{r+1}{\log_{p_1} p})r-\frac{r^2}{\log_{p_1} p}} n}\right)$
.

Proof. Let r_1 be the last round in Phase 2. Note that $A_{r_1-1} < n/(3\lambda)$. By lemma 3.2, we also note that w.h.p $A_0 \geq (1-\delta)\gamma\frac{n}{\beta \log_{p_1}^{2r+2} n}$. Note that number of rounds spent in Phase 2 is r_1. Therefore by application of lemma 3.5, we get that w.h.p r_1 is less than or equal to $\log_{p_1}\left(\frac{\alpha n/(3\lambda)}{A_0-K}\right)/\log_{p_1} p$ where K is the constant in the statement of lemma 3.4. Expanding the expression for the upper bound on r_1 we get,

$$\begin{aligned}
r_1 &\leq \frac{1}{\log_{p_1} p}\left(\log_{p_1}\frac{\alpha n/(3\lambda)}{A_0-K}\right) \\
&= \frac{1}{\log_{p_1} p}\left(\log_{p_1}(\alpha n/(3\lambda)) - \log_{p_1}(A_0-K)\right) \\
&= \frac{1}{\log_{p_1} p}\left(\log_{p_1}(\alpha n/(3\lambda)) - \log_{p_1}(A_0(1-K/A_0)\right) \\
&= \frac{1}{\log_{p_1} p}\left(\log_{p_1}(\alpha n/(3\lambda)) - \log_{p_1} A_0 - \log_{p_1}(1-K/A_0)\right) \\
&\leq \frac{1}{\log_{p_1} p}((2r+1)\log_{p_1}\log_{p_1} n + C_0)
\end{aligned}$$

where C_0 is some constant. The last step follows by noting that $A_0 \geq \gamma(\frac{n}{\beta \log_{p_1}^{2r+2} n})$.

Therefore by using lemma 2.3, w.h.p the number of infected nodes is less than or equal to $q^\ell M_0 = \theta\left(\frac{n}{\log_{p_1}^{r+1+(1-\frac{r+1}{\log_{p_1} p})r-\frac{r^2}{\log_{p_1} p}} n}\right)$. □

We present the detailed analysis of Phase 3 below.

3.4.5 Phase 3

We divide the analysis of this phase into two parts. The first part begins immediately after Phase 2 has ended. The first part ends at the round when the number of alerted nodes is at least $n/3$. Let p_1 be the lower bound on the expected growth rate of the number of alerted nodes in this part of the phase. First we show in the following lemma that there is a $\theta(\log n)$ edge expansion which will be used in proofs of Phase 3 and Phase 4.

Lemma 3.7. *For any random d-regular graph and any set S with size less than $n/2$, let $\xi(S)$ be the set of edges with one endpoint in S and one outside S. Then w.h.p $\xi(S) \geq |S|d/4$.*

Proof. S has $d|S|$ edges incident on it. The expected number of incident edges with one endpoint not in S is at least $(|S|d)/2$. For each vertex i in S, let $X_{(i,i_k)}$ denote the random variable which is 1 when the kth edge incident on i falls outside S. Let $Z(S)$ denote the total number of edges which have exactly one endpoint outside S. Therefore $Z(S) = f(X_{(1,1_0)}, \ldots, X_{(i,i_k)}, \ldots, X_{(|S|,|S|_d)})$, for some function f. Note that each of the random variables are independent. Moreover the function satisfies the Lipchitz condition, i.e $|f(X_{(1,1_0)}, \ldots, X_{(i,i_k)}, \ldots, X_{(|S|,|S|_d)}) - f(X_{(1,1_0)}, \ldots, X'_{(i,i_k)}, \ldots, X_{(|S|,|S|_d)})| \leq 1$, therefore by Azuma's inequality $Pr(|Z(S) - E(Z(S))| \geq 1/2E(Z(S))) \leq 2e^{-\frac{1/4(|S|d/4)^2}{|S|d}} \leq 1/2^{\theta(n)}$. A union bound over all possible S gives the required result. □

The next two lemma's are used to compute p_1. We call an edge coming out of an alerted node as an *alerted* edge.

Lemma 3.8. *Let $f(x) = \frac{1-C^x}{x}$. For $C < 1$, and $1 \leq x \leq d$, the minimum occurs at the largest possible value of x.*

Proof. Note that $\frac{df}{dx} = -\frac{1-C^x}{x^2} - \frac{C^x \log C}{x}$. Setting the value to zero, we get that a local minima or maxima occurs only when $x_0 = \frac{1-C^{x_0}}{C^{x_0} \log \frac{1}{C}}$. Plugging the value for x_0 into f, we get that $f(x_0) = C^{x_0} \log(\frac{1}{C})$. This last quantity is minimized for x_0 as large as possible, i.e $x_0 = d$. Note that $f(1) = 1 - C$ and $f(d) = \frac{1-C^d}{d}$. Finally we show that $1 - C \geq \frac{1-C^d}{d}$.

Note that $1-C \geq \frac{1-C^d}{d}$ iff $1-C \geq \frac{(1-C)(1+C,\ldots,+C^{d-1})}{d}$ iff $d \geq 1+C+C^2+,\ldots,+C^{d-1}$. The last inequality holds since $C^i \leq 1$ for all i. □

Lemma 3.9. *W.h.p, $p_1 \geq (1 + (1 - e^{-\alpha})/4)$, as n goes to infinity.*

Proof. Let r_1 be the first round when the number of alerted nodes is at least $n/(3\lambda)$. Thus $A_{r_1} \geq n/3\lambda$. By Lemma 3.7 the number of alerted edges at the end of round r_1 is $\frac{nd}{4(3\lambda)}$. If the number of alerted edges on each virgin node is at most x, the number of nodes which have a possibility of being alerted in round $r_1 + 1$ is at least $\frac{nd}{4(3\lambda)x}$. The probability that a virgin node with at most x alerted edges gets alerted in round $r_1 + 1$ is at least $(1 - (1 - \frac{\alpha}{d})^x)$. Therefore the expected number of alerted nodes in round $r_1 + 1$ is at least $\frac{nd}{4(3\lambda)} \frac{1}{x}(1 - (1 - \frac{\alpha}{d})^x)$. Substituting $(1 - \frac{\alpha}{d})$ for C in the statement of lemma 3.8, we see that $f(x)$ is minimized at $x = d$, since d is the largest possible value for x in this context. So the number of new alerted nodes at round $r_1 + 1$ is at least $\frac{nd}{4(3\lambda)} \frac{1}{d}(1 - (1 - \frac{\alpha}{d})^d) \geq \frac{nd}{4(3\lambda)} \frac{1}{d}(1 - e^{-\alpha})$. Therefore the number of nodes in A_{r_1+1} is at least $A_{r_1}(1 + (1 - e^{-\alpha})/4)$ □

The next two lemma's are used to compute the maximum number of rounds spent in this phase.

The following lemma is an adaptation of statement 2.3 and is presented here without a proof.

Lemma 3.10. *Let r_1 be the first round when the number of alerted nodes is at least $n/(3d)$. For all $r_2 > r_1$, s.t $|A_{r_2-1}| < n/3$, w.h.p $|A_{r_2}| \geq p_1^{r_2-r_1}(n/(3d) - K')$ for* *some constant K'.*

Lemma 3.11. *Let r_1 be the first round when the number of alerted nodes is at least $n/(3\lambda)$. If $r_2 > r_1$ and $A_{r_2-1} < n/3$, then w.h.p $r_2 - r_1 \leq \lceil \log_{p_1} (\frac{A_{r_2}}{n/(3\lambda)-K'}) \rceil$ where K' is the constant in the statement of lemma 3.10.*

In the following lemma we estimate the upper bound on the number of infected nodes at the end of the first part of Phase 3.

Lemma 3.12. *Let r_1 be the first round when the number of alerted nodes is at least $n/3$. The number of infected nodes at the end of the r_1th round is w.h.p $O\left(\frac{n}{\log_{p_1}^{r+1+(1-\frac{r+1}{\log_{p_1} p})r-\frac{r^2}{\log_{p_1} p}-r} n}\right)$.*

Proof. By definition, the number of alerted nodes at the beginning of Phase 3 is at least $n/3\lambda$. By Lemma 3.11, the number of rounds taken to to reach at least $n/3$ alerted nodes is w.h.p upper bounded by $\log_{p_1} \log_{p_1} n + C_0'$, where C_0' is some constant. Then by application of lemma 2.3 the number of infected nodes is no more than $O\left(\frac{n}{\log_{p_1}^{r+1+(1-\frac{r+1}{\log_{p_1} p})r-\frac{r^2}{\log_{p_1} p}-r} n}\right)$. Note that this is o(n). □

We define the second part of the phase to begin when the first part ends. For the second part of this phase we make use of the following lemma.

Phase 3 is defined to end at the first round when at least $n/3$ nodes have been alerted. We show in the following lemma that the number of rounds spent in the second part of this phase, is w.h.p a constant.

Lemma 3.13. *On a d-regular graph, if the number of infected nodes is $o(n)$ and the number of alerted nodes is greater than $n/3$, then the maximum number of rounds required for the number of alerted nodes to exceed $n/2$ is a constant w.h.p.*

Proof. By lemma 3.7, w.h.p the number of alerted edges, incident on virgin nodes is greater than or equal to $dn/12$. Let x be the number of virgin nodes that each have greater than or equal to ϵd,$0 < \epsilon \leq 1$ alerted edges incident on them. By definition of x, $xd + (n - n/3 - x)\epsilon d \geq (dn)/12$, or $x \geq \frac{n}{(1-\epsilon)}(\frac{2\epsilon+1}{12} - \epsilon)$.

The probability that a virgin node with ϵd alerted edges gets alerted in the next time step is at least $1 - e^{-\alpha\epsilon}$. Therefore by a linearity of expectation argument the total number of nodes which receive alerts in the next time step is at least $(1 - e^{-\alpha\epsilon})\frac{n}{(1-\epsilon)}(\frac{2\epsilon+1}{12} - \epsilon)$. We can find an ϵ s.t $(1 - e^{\alpha\epsilon})\frac{n}{(1-\epsilon)}(\frac{2\epsilon+1}{12} - \epsilon)$ is greater than n/c'' for some constant $c'' > 1$.

Then by an application of Azuma's inequality, we show that the expected number of new nodes alerted is tightly bounded around the expectation w.h.p. Thus in the second part of Phase 3, $\theta(n)$ nodes are being alerted in each round. This proves that w.h.p it takes constant number of rounds for the number of alerted nodes to reach n/2 from the beginning of the second part of Phase 3. □

Corollary 1. *The number of infected nodes at the end of Phase 3 is w.h.p upper bounded by o(n).*

Proof. By lemma 3.12 the number of infected nodes at the end of part 1 of Phase 3 is $o(n)$. Lemma 3.13 says that it take $\theta(1)$ rounds to reach the end of Phase 3. The number of alerted nodes at the end of Phase 3 is w.h.p $q^{\theta(1)}o(n)$ which is $o(n)$. □

Finally we present the detailed analysis of the last phase of our analysis.

3.4.6 Phase 4

At the beginning of Phase 4, there are at least n/2 alerted nodes, o(n) infected nodes and less than n/2 virgin nodes. Let the set of virgin nodes be called V with size v.

We can assume that all the edges which go out of V, go to the set of alerted nodes, as the number of infected nodes is o(n). For the sake of analysis let us assume that the number of alerted nodes at the beginning of Phase 4 is exactly equal to n/2

Lemma 3.14. *If there are at least n/2 alerted nodes and at most o(n) infected nodes, it takes only a constant number of steps w.h.p, for the number of virgin nodes remaining to be o(n).*

Proof. We know by lemma 3.7 that the number of edges with one end point in V and one end point outside V is at least $|V|d/4$. Let x be the number of virgin nodes with greater than or equal to ϵd alerted edges, where $0 \leq \epsilon < 1$. The value of ϵ will be assigned later to suit our needs. Therefore $(v - x)\epsilon d + dx \geq dv/4$. Hence $(1 - \epsilon)x \geq v/4 - \epsilon v$, or $x \geq \frac{v}{(1-\epsilon)}(1/4 - \epsilon)$. If a virgin node has at least ϵd alerted edges incident on it, the probability that it does not receive an alert from an alerted neighbor is $(1 - \alpha/d)^{\epsilon d} \leq e^{-\alpha\epsilon}$. The probability that it does receive an alert from any of its neighbors is 1-$e^{-\alpha\epsilon}$. Please note that this analysis assumes no multi-edges . Hence, by the linearity of expectation argument, the number of new alerted nodes in each round is $\frac{v}{(1-\epsilon)}(1/4 - \epsilon)(1 - e^{-\alpha\epsilon})$, which is greater than v/c' for some constant $c' > 1$.

Then by Azuma's inequality we can show that the number of new alerted nodes is closely concentrated around its expected values, by an analysis similar to that shown in the proof of Lemma 3.13. Thus in constant number of steps, the number of alerted nodes is $n - o(n)$, whereas by an application of Lemma 2.3 the number of infected nodes remain o(n). □

3.5 Empirical Results

We simulated the spread of a worm and an alert through a network to empirically determine the fraction of nodes saved.[2]

3.5.1 Empirical Setup

For all the experiments, we fixed the worm strategy such that each infected node, in each round, sends out the worm to β unique nodes selected uniformly at random, and we fixed the alert strategy according to Algorithm 1. We note that the worm strategy we used in these experiments is not necessarily the best possible worm strategy, but we selected this strategy for concreteness. We stress that our theoretical results hold for any worm strategy.

We performed experiments on two kinds of networks. For the first network, we use a random d-regular directed graph as the overlay network. Our d-regular directed random graph was created on the lines of the configuration model proposed in [54]. We have an array of nodes id's containing d stubs each for every one of the n node id's. This nd size array is in increasing order. We take a random permutation of this array and map the corresponding elements of the first array and the permuted array to get a d-regular directed graph. We ignore self loops and multi-edges in this implementation. We call this network a *random* network

For the second network we make use of the pairwise independent hash functions described in [36]. In this model, node id i maps to $(ai+b) \bmod n$, where n is a prime and $0 < a \leq n-1$, $0 \leq b \leq n-1$. We find 50 distinct values of a chosen uniformly at random between 1 and $n-1$. We find 50 values of b chosen uniformly at random

[2] All of the code necessary to replicate these experiments is available at `http://www.cs.unm.edu/~navin/false-alerts.html`.

between 0 and $n-1$. This gives us 50 pairwise independent hash functions which map a node i to 50 other nodes in the network. We call this network a *pseudo-random* network.

The relative advantage in the implementation of the pseudo-random network over the random network is in not having to store explicitly the graph in the memory. We compute the neighbors of a node in real time when we need to access them. The implementation of the default mod function was computationally intensive. To make this operation more efficient we store $2^i \bmod p$, $0 < i < \log_2 p$ in one preprocessing step and access the stored values throughout the simulation. All multiplications between integers is reduced to multiplications between powers of two. For the rest of this section we give sizes of the network in this network model in terms of the largest power of two smaller than the prime number representing the actual size of the network. In our experiments we use hash table primes given at [1].

There are several possible strategies for resolving the status of a node that gets both a worm message and an alert message in the same round and is neither infected nor alerted before that round. In our theoretical analysis, we assumed that if a node receives just one worm message it becomes infected. However, in our experiments, we used the somewhat more relaxed and realistic assumption that the probability that the node gets infected equals the number of worm messages received divided by the total number of messages received, and that the probability the node becomes alerted is 1 minus this quantity. We note that this assumption is equivalent to assuming that the messages all arrive in the node's message queue according to some random permutation.

Our first experiment measured the fraction of nodes saved when we varied both α and n. In our second experiment we measured the fraction of nodes alerted and infected at each round of the algorithm. To further explore the role of α in our model, in our third experiment, we measured the fraction of nodes saved as α varies.

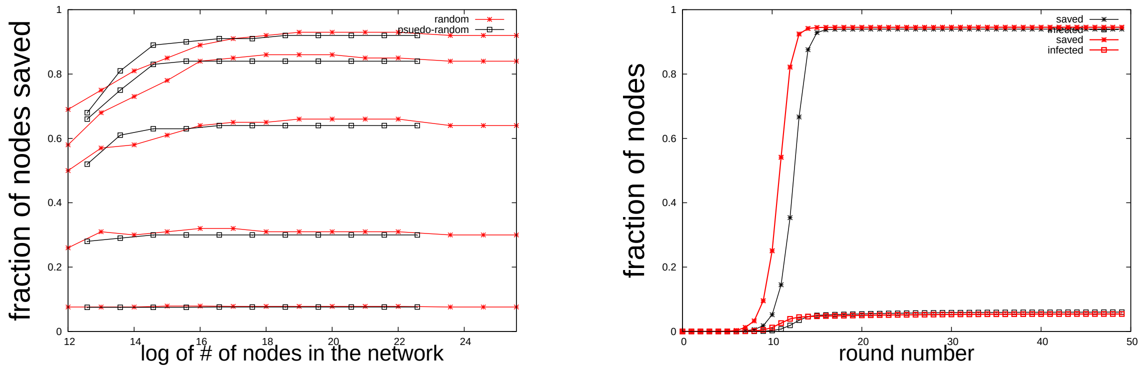
In this experiment we ensure that the τ value is always adjusted so that the number of nodes which can be alerted due to a false alert is always 10^4. So $\tau = \lfloor \log_\alpha 10^4 \rfloor$.

3.5.2 Results

To remove noise in our experiments, each data point was averaged over 100 trials.

Figure 3.4(a) shows a contour plot of log of the number of nodes in the network vs the fraction of nodes saved for the first experiment. The values of the other parameters were as follows; $\beta = 2$, $\gamma = 0.02$, $\tau = 5$ and α takes on all even values between 2 and 10. Since it is easier to carry out simulations for much larger values of n on the pseudo-random network, we vary the number of nodes for this network from $2^{12.58}$ to $2^{25.58}$. The size of the regular network varies from 2^{12} to 2^{22}. We observe that there is a very small increase in the fraction of nodes saved in the pseudo-random network as n crosses $2^{18.58}$, so in all our other experiments we have limited the network size of the pseudo-random network to the size of the random network. These results suggest that our algorithms for spreading alerts might be most effective in conjunction with other techniques (like throttling) that would enable the alerts to spread more quickly than the worm. There is a very small increase in the fraction of nodes saved as n grows larger, implying that the results may be better for very large values of n.

For the second experiment, we plotted the fraction of nodes saved/infected in each round for both kind of networks. The network sizes considered were of the order of 10^7 nodes. Here $\gamma = 0.02$, $\beta = 2$, $\alpha = 10$ and $\tau = 5$. For runs of the simulation where the number of rounds is less than the other runs, we repeat the final values to compensate for the missing rounds, while calculating the average of 100 trials. In Figure 3.4(b) we see results for the second experiment. We are able to save about 93% and 91% of the nodes for the random and pseudo-random networks



(a)

(b)

Figure 3.4: (a) contour plot of log of # of nodes vs fraction of nodes saved for $\alpha = 2, 4, 6, 8, 10$ (b)number of nodes saved/infected in each round

respectively. Notice that the value of α is five times that of β. In practice, this condition may hold since the alerts are traveling through a predetermined overlay network and a technique such as throttling can ensure that alert messages received through the overlay are given priority over other types of messages. The number of nodes saved in the pseudo-random network is less than the number of nodes saved for the random network in all of our experiments. That is expected as the expansion of the pseudo-random network may be worse than the expansion of the random network.

For our third experiment the number of nodes are fixed at $2^{23}(2^{23.58}$ for the pseudo-random network). Here $\beta = 2$ and $\gamma = 0.02$. The value of τ is always adjusted so that the number of nodes which can be alerted due to a false alert is always 10^4. So $\tau = \lfloor \log_\alpha 10^4 \rfloor$. In this experiment we find an improvement of around 20% in the fraction of nodes saved for values of α between three and seven for both the network models. The results of this experiment are given in Figure 3.5. We note that there is a decrease in the number of nodes saved when we go from $\alpha = 6$ to $\alpha = 7$. We

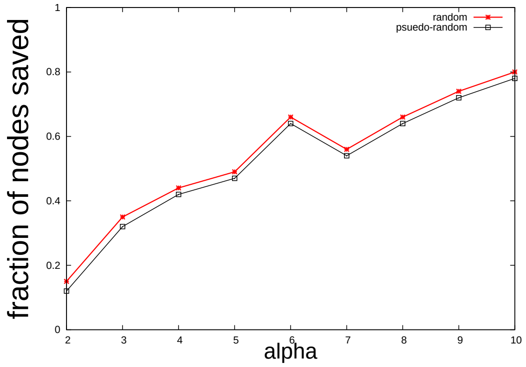


Figure 3.5: alpha vs fraction of nodes saved

believe this happens because the value of τ at $\alpha = 6$ is five and at $\alpha = 7$ it decreases to four.

3.6 Conclusion

In this chapter we have described an alert propagation algorithm which under certain conditions, w.h.p saves all but $o(n)$ nodes from a worm attack, and limits the spread of false alerts to polylogarithmic number of nodes. We make use of a $\theta(\log n)$ regular network for distributing alerts. We have demonstrated empirically that this algorithm works effectively against a randomly spreading worm under conditions that may be reasonable for modern computer networks.

Chapter 4

On the power of Mediators

In the previous chapters, we assumed that the nodes in a network were altruistic. In this chapter we present a game, in which players are selfish.

Recent results show that malicious players in a game may, counter-intuitively, improve social welfare [39, 7, 46]. For example, in [39] it is showed that for a virus inoculation game, the existence of malicious players may actually lead to better social welfare for the remaining players than if such malicious players are absent.This improvement in the social welfare with malicious players has been referred to as the "windfall of malice" [7]. The existence of the windfall of malice for some games leads to an intriguing question: Can we achieve the windfall of malice even without the actual presence of malicious players?

In this chapter, we show that the answer to the previous question is sometimes "Yes". How do we achieve the beneficial impact of malicious players without their actual presence? Our approach is to use a mediator. Informally, a mediator is a trusted third party that suggests actions to each player. The players retain free will and can ignore the mediator's suggestions. The mediator proposes actions privately to each player, but the algorithm the mediator uses to decide what to propose is public knowledge. The contributions of this chapter are the following: We introduce a general technique for designing mediators that is inspired by careful study of the "windfall of malice" effect. In our approach, the mediator makes a random choice of one of two possible configurations, where a configuration is just a set of proposed actions for each player. The first configuration is optimal: the mediator proposes a set of actions that achieves the social optimum (or very close to it). The second configuration is "fear inducing": the mediator proposes a set of actions that leads to catastrophic failure for those players who do not heed the mediators advice. The purpose of the second configuration is to ensure that the players follow the advice of the mediator when the optimal configuration is chosen. Thus, the random choice of which configuration is chosen must be hidden from the players. We show the applicability of our technique by using it to design a mediator for the virus inoculation game from [39], that achieves a social welfare that is asymptotically optimal.

We also show the limits of our technique by proving an impossibility result that shows that for a large class of games, no mediator will improve the social welfare over the best Nash equilibrium. In particular, this impossibility result holds for the congestion games that in [7] it s shown to have a windfall of malice. Thus, we show that some games with a windfall of malice effect can not be improved by the use of a mediator.

4.1 Layout of this Chapter

In Section 4.2 we give a few basic definitions which are common to all games. Then, in Section 4.3 we consider the virus inoculation game in detail. Section 4.4 and 4.5 then contain results about the network congestion games, and finally in Section 4.6 we conclude with some open problems.

4.2 Basic definitions and notation.

A *correlated equilibrium* is a probability distribution over strategy vectors that ensures that no player has incentive to deviate. We define a *configuration* for a given game to be a vector of pure strategies for that game, one for each player. We define a *mediator* for a game to be a probability distribution $\mathcal{D}(\mathcal{C})$ over a finite set of different configurations $\mathcal{C}$. The set of configurations $\mathcal{C}$ and the distribution $\mathcal{D}(\mathcal{C})$ are known to all players. However, the actual configuration chosen is unknown, and the advice the mediator gives to a particular player based on the chosen configuration is known only to that player. We say that a mediator is *valid* if all players are incentivized to follow its advice. In this case, the mediator implements a correlated equilibrium. From a distributed computing viewpoint, the major difference between a correlated equilibrium and a Nash equilibrium is that in a correlated equilibrium, players share a global coin, but in a Nash equilibrium, players only have access to private coins.

Throughout this chapter, we will only consider mediators that treat all players equally, i.e., once having decided (by a random experiment according to $\mathcal{D}(\mathcal{C})$) which is the configuration the mediator is choosing from, all players have the same probability to be proposed a particular strategy. Also, throughout the chapter we assume that the number of strategic players, n, is very large (tending to infinity). Finally, we will use the notation $a(n) \sim b(n)$ if $a(n) = b(n)(1 \pm o(1))$. We also use the notation $[n] = \{1, \ldots, n\}$.

4.3 Virus Inoculation Game

We now describe the *virus inoculation game* from [39, 4]. There are n players, each corresponding to a node in a square grid G. Each player has two choices: either to inoculate itself (at a cost of 1) or to do nothing and risk infection (which costs L).

After the decision of the nodes to inoculate or not, one node selected uniformly at random is infected with a virus. A node v that chooses not to inoculate gets infected by the virus if either the virus starts at v or the virus starts at another node v' and there is a path of not inoculated nodes connecting v and v'.

We define the *attack graph* G_a to be the graph induced on G by the set of all nodes that do not inoculate. Aspnes et al. [4] proved that in a pure Nash equilibrium every component of the attack graph has size n/L. The social welfare achieved in such an equilibria is thus $\Theta(n)$. However, Moscibroda et al. [39] proved that the minimum social cost is $\Theta(n^{2/3}L^{1/3})$ for the grid, which occurs when the components in G_a are of size $(n/L)^{2/3}$. Moreover, they show that the existence of enough Byzantine players, who can never be trusted to inoculate, ensures that the social welfare of any Nash equilibria is slightly better than $\Theta(n)$.

Based on the result from [39], we observe that the main problem in this game is that the individual players do not have enough fear of being infected. In particular, they are unable to achieve the optimal social welfare because they form connected components in G_a that are too large. Thus, we design a mediator that randomly chooses between two configurations (see Figure 4.1). The first configuration is optimal: all components in G_a are of size $(n/L)^{2/3}$. The second configuration is "fear inducing": any node that does not inoculate in this configuration has probability about $1/2$ of being infected. The only purpose of the second configuration is to ensure that the selfish players follow the advice of the mediator when the optimal configuration is chosen.

Clearly, we only want to choose the fear inducing configuration with very small probability. The critical fact that enables us to do this is the fact that for a given player, when that player is advised to inoculate, the posterior probability that the mediator is in the second configuration increases significantly over the prior probability. This is the case because so many more nodes are told to inoculate in the second configuration. Thus, players that are told to inoculate are more likely to be infected. Finally, we also note that nodes that are told not to inoculate are more likely to be in the first configuration and thus not to be attacked.

We now formally describe the mediator for this game.[1] The mediator will choose randomly between one of the following two configurations C_1 and C_2.

Configuration C_1: The mediator proposes a pattern of inoculation such that 1) all nodes that do not inoculate are in one giant component in G_a; 2) each node has equal probability of being chosen to inoculate; and 3) the probability that a fixed node inoculates is $\frac{1}{2} - \frac{1}{2\sqrt{n}}$. The mediator accomplishes this in the following manner:

1. The mediator flips a coin. If it comes up heads, it proposes that all nodes in even columns do not inoculate. If it comes up tails, it proposes that all nodes in odd columns do not inoculate.

2. The mediator chooses a random integer, x, uniformly between 1 and $\sqrt{n}$. For each of the columns that have not already been told not to inoculate, the mediator proposes that each node in that column inoculate except for the x-th node in that column.

Configuration C_2: The mediator proposes a pattern of inoculation that ensures that 1) each component in G_a is of size no more than $(\frac{n}{L})^{2/3}$; 2) each node is chosen to inoculate with equal probability; and 3) the probability that a fixed node inoculates is at most $2(L/n)^{1/3}$. It does this as follows.

1. The mediator chooses integer x uniformly at random in the range 1 to $(n/L)^{1/3}$.

[1] For ease of analysis, we assume that both $\sqrt{n}$ and $(\frac{n}{L})^{1/3}$ are integers. Also, $\sqrt{n}$ should be an integer multiple of $(\frac{n}{L})^{1/3}$ (this assumption can be removed easily without effecting our asymptotic results)

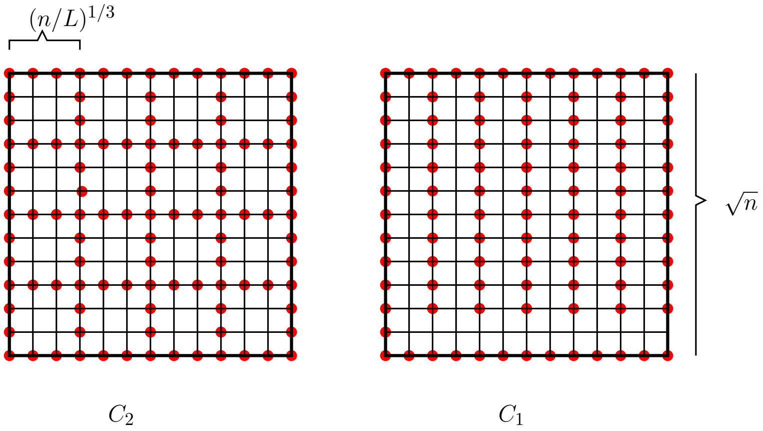


Figure 4.1: The $\sqrt{n} \times \sqrt{n}$ grid with two configurations C_1, C_2 for the virus inoculation game

2. For every node v in row r and column c, if one of the following two conditions hold, the mediator proposes v to inoculate: 1) $r \equiv xmod((n/L)^{1/3})$; or 2) $c \equiv xmod((n/L)^{1/3})$. Otherwise the mediator tells v not to inoculate.

For these two configurations C_1 and C_2 we now define the probability distribution $\mathcal{D}(\{C_1, C_2\})$ with $p_1 = cL^{-2/3}n^{-1/3}$ and $p_2 = (1 - cL^{-2/3}n^{-1/3})$, where $c > 0$ can be chosen to be any small constant satisfying $c > 2L/(L-1)$ (in particular $c = 4$ always suffices).

We can now prove the main theorem of this section which shows that $\mathcal{D}(\{C_1, C_2\})$ is asymptotically optimal.

Theorem 4.1. *$\mathcal{D}(\{C_1, C_2\})$ is a mediator with social welfare $\Theta(n^{2/3}L^{1/3})$.*

Proof. To prove the statement, we need a few definitions. Define by $\mathcal{E}_I^j$ the event that the mediator advises player j to inoculate and define by $\mathcal{E}_{\bar{I}}^j$ the event that the mediator advises player j not to inoculate. Since all players are to be treated equally by the mediator, we will omit the index j. Define also by $\mathcal{E}_A$ the event that a not inoculated node gets infected by the virus, and denote by $\mathcal{C}_A$ the infection cost of a not inoculated node. We also use the notation $\mathcal{C}_I$ to denote the cost of inoculation (clearly $\mathcal{C}_I = 1$). We first need to show that $\mathcal{D}(\{C_1, C_2\})$ indeed yields a mediator. That is, we have to verify the following conditions of a correlated Nash equilibrium:

$$\begin{aligned} \mathbf{E}\left[\mathcal{C}_A|\mathcal{E}_I\right] &\geq \mathbf{E}\left[\mathcal{C}_I|\mathcal{E}_I\right] = 1 \\ \mathbf{E}\left[\mathcal{C}_A|\mathcal{E}_{\bar{I}}\right] &\leq \mathbf{E}\left[\mathcal{C}_I|\mathcal{E}_{\bar{I}}\right] = 1, \end{aligned}$$

which is equivalent to showing that

$$\begin{aligned} \mathbf{Pr}\left(\mathcal{E}_A|\mathcal{E}_I\right) &\geq 1/L & (4.1) \\ \mathbf{Pr}\left(\mathcal{E}_A|\mathcal{E}_{\bar{I}}\right) &\leq 1/L, & (4.2) \end{aligned}$$

since for any event $\mathcal{E}$ with $\mathbf{Pr}\left(\mathcal{E}\right) > 0$, we have that $\mathbf{E}\left[\mathcal{C}_A|\mathcal{E}\right] = L\mathbf{Pr}\left(\mathcal{E}_A|\mathcal{E}\right)$. We denote furthermore by $\mathcal{E}_i$, $i = 1, 2$, the event that configuration C_i, $i = 1, 2$ is chosen. Note that $\mathbf{Pr}\left(\mathcal{E}_A|\mathcal{E}_1\right) = 1$. To prove (4.1), first observe that

$$\begin{aligned} \mathbf{Pr}\left(\mathcal{E}_1|\mathcal{E}_I\right) &= \mathbf{Pr}\left(\mathcal{E}_1, \mathcal{E}_I\right)/\mathbf{Pr}\left(\mathcal{E}_I\right) \\ &\sim \frac{p_1(1/2 - 1/(2\sqrt{n}))}{p_1(1/2 - 1/(2\sqrt{n})) + 2p_2(L/n)^{1/3}}, \end{aligned}$$

and similarly for $\mathbf{Pr}\left(\mathcal{E}_2|\mathcal{E}_I\right)$. Now, plugging in the values of p_1, p_2 and using that

$L \in o(n)$ we get [2]

$$\begin{aligned}
\mathbf{Pr}\left(\mathcal{E}_A|\mathcal{E}_I\right) &= \mathbf{Pr}\left(\mathcal{E}_A, \mathcal{E}_1|\mathcal{E}_I\right) + \mathbf{Pr}\left(\mathcal{E}_A, \mathcal{E}_2|\mathcal{E}_I\right) \\
&= \mathbf{Pr}\left(\mathcal{E}_A|\mathcal{E}_1, \mathcal{E}_I\right)\mathbf{Pr}\left(\mathcal{E}_1|\mathcal{E}_I\right) + \mathbf{Pr}\left(\mathcal{E}_A|\mathcal{E}_2, \mathcal{E}_I\right)\mathbf{Pr}\left(\mathcal{E}_2|\mathcal{E}_I\right) \\
&\geq \mathbf{Pr}\left(\mathcal{E}_1|\mathcal{E}_I\right) + \frac{1}{L^{2/3}n^{1/3}}\mathbf{Pr}\left(\mathcal{E}_2|\mathcal{E}_I\right) \\
&\sim \frac{p_1(1/2 - 1/(2\sqrt{n}))}{p_1(1/2 - 1/(2\sqrt{n})) + 2p_2(L/n)^{1/3}} \\
&+ (L^{-2/3}n^{-1/3})\frac{2p_2(L/n)^{1/3}}{p_1(1/2 - 1/(2\sqrt{n})) + 2p_2(L/n)^{1/3}} \\
&\sim \frac{(c/2)L^{-2/3}n^{-1/3} + 2L^{-1/3}n^{-2/3}}{(c/2)L^{-2/3}n^{-1/3} + 2(L/n)^{1/3}} \\
&= \frac{2cL^{2/3}n^{2/3} + 4Ln^{1/3}}{2cL^{2/3}n^{2/3} + 4L^{5/3}n^{2/3}} \\
&\sim \frac{c}{c + 2L},
\end{aligned}$$

which is greater than $1/L$ for $c > (2L)/(L-1)$. Similarly, to prove (4.2), note that

$$\begin{aligned}
\mathbf{Pr}\left(\mathcal{E}_1|\mathcal{E}_{\bar{I}}\right) &= \mathbf{Pr}\left(\mathcal{E}_1, \mathcal{E}_{\bar{I}}\right)/\mathbf{Pr}\left(\mathcal{E}_{\bar{I}}\right) \\
&\sim \frac{p_1(1/2 + 1/(2\sqrt{n}))}{p_1(1/2 + 1/(2\sqrt{n})) + p_2(1 - 2(L/n)^{1/3})},
\end{aligned}$$

[2] if $L = \theta(n)$, then any pure Nash equilibria is trivially asymptotically optimal

and analogously for $\mathbf{Pr}\left(\mathcal{E}_2|\mathcal{E}_{\bar{I}}\right)$. Hence,

$$\begin{aligned}
\mathbf{Pr}\left(\mathcal{E}_A|\mathcal{E}_{\bar{I}}\right) &= \mathbf{Pr}\left(\mathcal{E}_A,\mathcal{E}_1|\mathcal{E}_{\bar{I}}\right)+\mathbf{Pr}\left(\mathcal{E}_A,\mathcal{E}_2|\mathcal{E}_{\bar{I}}\right)\\
&= \mathbf{Pr}\left(\mathcal{E}_A|\mathcal{E}_1,\mathcal{E}_{\bar{I}}\right)\mathbf{Pr}\left(\mathcal{E}_1|\mathcal{E}_{\bar{I}}\right)+\mathbf{Pr}\left(\mathcal{E}_A|\mathcal{E}_2,\mathcal{E}_{\bar{I}}\right)\mathbf{Pr}\left(\mathcal{E}_2|\mathcal{E}_{\bar{I}}\right)\\
&\leq \mathbf{Pr}\left(\mathcal{E}_1|\mathcal{E}_{\bar{I}}\right)+\frac{1}{L^{2/3}n^{1/3}}\mathbf{Pr}\left(\mathcal{E}_2|\mathcal{E}_{\bar{I}}\right)\\
&\sim \frac{p_1(1/2+1/(2\sqrt{n}))}{p_1(1/2+1/(2\sqrt{n}))+p_2(1-2(L/n)^{1/3})}\\
&+ (L^{-2/3}n^{-1/3})\frac{p_2(1-2(L/n)^{1/3})}{p_1(1/2+1/(2\sqrt{n}))+p_2(1-2(L/n)^{1/3})}\\
&\sim \frac{(c/2)L^{-2/3}n^{-1/3}+L^{-2/3}n^{-1/3}}{(c/2)L^{-2/3}n^{-1/3}+1}\\
&\sim \frac{c+2}{2L^{2/3}n^{1/3}},
\end{aligned}$$

which is smaller than $1/L$ since $L \in o(n)$. Thus, we have shown that $\mathcal{D}(\{C_1,C_2\})$ indeed is a valid mediator in that players will follow its advice. We next compute the social cost for this mediator. Let $\mathcal{I}_1$ ($\bar{\mathcal{I}}_1$) be the set of nodes that inoculate (respectively do not inoculate) in C_1, and let $\mathcal{I}_2$ ($\bar{\mathcal{I}}_2$) be the set of nodes that inoculate (respectively do not inoculate) in C_2. Then the social cost for the mediator can be written as

$$\begin{aligned}
&p_1(|\mathcal{I}_1|+\sum_{v\in\bar{\mathcal{I}}_1} L\mathbf{Pr}\left(\mathcal{E}_A|\mathcal{E}_1,\mathcal{E}_{\bar{I}}\right))+p_2(|\mathcal{I}_2|+\sum_{v\in\bar{\mathcal{I}}_2} L\mathbf{Pr}\left(\mathcal{E}_A|\mathcal{E}_2,\mathcal{E}_{\bar{I}}\right))\\
\sim\ &\frac{c}{L^{2/3}n^{1/3}}(n/2+(n/2)L)+(2n^{2/3}L^{1/3}+nL\frac{1}{L^{2/3}n^{1/3}})\\
=\ &(3+(c/2))n^{2/3}L^{1/3}+(c/2)(n/L)^{2/3}=\Theta(n^{2/3}L^{1/3}).
\end{aligned}$$

□

4.4 Impossibility Result

In light of the results in the previous section, a natural question is: Is it possible to design a mediator that will always improve the social welfare in any game for which there is a windfall of malice? Unfortunately, the answer to this question is "No", as we show in this section. In particular, we show that the congestion games which Babaioff, Kleinberg and Papadimitriou have proven have a windfall of malice effect [7] do not admit a mediator that is able to improve the social welfare. In fact, we prove a stronger impossibility result, showing that for any non-atomic, symmetric congestion game where the cost of a path never decreases as a function of the flow through that path (of which class of games, the examples in [7] are special instances), no mediator can improve the social optimum. In the rest of this section, we first define the congestion games we consider and then prove our impossibility result for these games.

A non-atomic, symmetric *congestion game* (henceforth, simply a congestion game) is a specified by a set of $n \to \infty$ players; a set of E facilities (or edges); $A \subset 2^E$ actions (or paths); and finally, for each facility e a cost function f_e associated with that facility. A pure strategy profile $\mathcal{A} = (A_1, \dots, A_n)$ is a vector of actions, one for each player. The cost of player i for action profile $\mathcal{A}$ is given by $F_i(\mathcal{A}) = \sum_{e \in A_i} f_e(x_e(\mathcal{A}))$ where $x_e(\mathcal{A})$ is the fraction of players using e in $\mathcal{A}$. As in [7], we assume that the game is *non-atomic*: since $n \to \infty$ the contribution of a single player to the flow over a facility is negligible; and *symmetric*: all players have the same cost functions.

For an action a and a flow $x \in [0, 1]$, let $\mathcal{F}_h(a, x)$ be the maximum possible cost of following action a when the total fraction of players following this action is x, where the maximum is taken over all ways that the remaining flow of $1 - x$ can be distributed over other actions. Similarly, let $\mathcal{F}_\ell(a, x)$ be the *minimum* cost of following action a when the total fraction of players following this action is x.

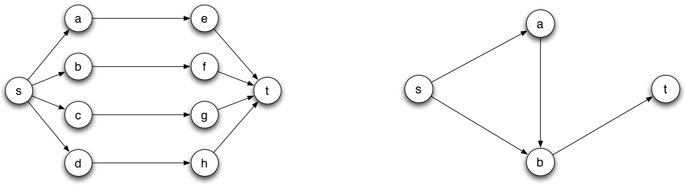


Figure 4.2: Examples where Theorem 4.2 applies

We prove the following theorem for congestion games where the cost function of every action is always non-decreasing in the fraction of players performing that action. The theorem says that for such games, coordination between the agents in order to establish a correlated equilibrium will not decrease the social cost.

Theorem 4.2. *Consider a non-atomic, anonymous congestion game. If for all* $a \in A$ *and* $0 \leq x \leq x' \leq 1$, $\mathcal{F}_h(a,x) \leq \mathcal{F}_\ell(a,x')$ *then the smallest social cost achieved by a correlated equilibrium is no less than the smallest social cost achieved by a Nash equilibrium.*

We give an overview of the proof of this theorem in subsection 4.4.1 and then the detailed proof in subsection 4.4.2

Figure 4.2 gives examples of congestion games for which Theorem 4.2 applies. In these graphs, if the costs of all edges are non-decreasing in flow, then the smallest social cost achieved by a correlated equilibria is no better than the smallest social cost achieved by a Nash equilibria. In both examples, all players must travel from the source node s to the sink node t, so the set of allowable actions are just the set of all paths from s to t. The graph on the left is a specific example of a more general class of graphs for which all paths are disjoint and edge costs are non-decreasing, for which Theorem 4.2 applies. The graph on the right is a generalization of the congestion game from [7], which they show has a positive windfall of malice for certain non-decreasing cost functions. In the next section and in Figure 4.3 described therein, examples of congestion games for which Theorem 4.2 does *not* hold are given.

4.4.1 Overview of Theorem 4.2

In this section we give a high level sketch of how we prove Theorem 4.2. We will fix a non-atomic, anonymous congestion game G with q actions, $a_1, \ldots, a_q$, and n players. We define a *configuration*, C, for such a game to be a partitioning of the set of players across the q actions. We note that the number of possible configurations is finite; in particular, q^n. We next fix a mediator, M, for this game. We assume the mediator uses ℓ different configurations $C_1, \ldots, C_\ell$; that $0 \leq x_{i,j} \leq 1$ is the fraction of the players in configuration C_j assigned to action a_i; and that $c_{i,j} \in \mathbb{R}$ is the cost in configuration C_j for action a_i. We further assume that for all $j \in [\ell]$, p_j is the probability with which the mediator M chooses C_j.

For any two actions a, a' we define the *a posteriori cost* of a given a' as the expected cost for a player of performing action a when action a' is suggested by the mediator M; formally, $\mathrm{POST}\,(a, a') = \mathbf{E}\,[\mathcal{C}_a | \mathcal{E}_{a'}]$, where $\mathcal{C}_a$ is a random variable (over the configuration chosen by the mediator) and $\mathcal{E}_{a'}$ is the event that action a' is recommended by the mediator. We define the *a priori cost* of action a as the cost of a player completely ignoring what the mediator suggests and always performing action a; formally, $\mathrm{PRI}\,(a) := \sum_{j=1}^{\ell} p_j c_{i,j}$.

The sketch behind our proof for this theorem is as follows. First, we show in Lemma 4.1 that for all actions a, if the cost of a is non-decreasing in the flow through a, then $\mathrm{POST}\,(a, a) \geq \mathrm{PRI}\,(a)$. We show this by repeated decompositions of terms in summations for the a priori and posterior costs. Next, let Y be the cost of a player listening and following the advice of the mediator, and let X be the cost of the player if she just ignores the advice of the mediator and always chooses the action a that minimized $\mathrm{PRI}\,(a)$. In Lemma 4.2 we show that it must be that $E(Y) \leq E(X)$. This lemma is shown by summing up inequality constraints on the mediator. Finally, we use these two lemmas to show the main theorem by showing that if Lemma 4.1 holds, then $E(Y) > E(X)$. The main technical challenge is the fact that we must show that $E(Y) > E(X)$ even though Lemma 4.1 does not necessarily give a strict inequality. We address this problem by a subtle case analysis in the proof of the main theorem, and by augmenting Lemma 4.1 to show that in some cases, the inequality it implies is strict.

We now present the detailed proof of Theorem 4.2

4.4.2 Proof of Theorem 4.2

Observe that the condition for all $a \in A$ and $0 \leq x \leq x' \leq 1$, $\mathcal{F}_h(a, x) \leq \mathcal{F}_\ell(a, x')$ implies that for all $i \in [m]$, $\forall j, k \in [\ell]$ we have that $x_{ij} \leq x_{ik}$ implies $c_{ij} \leq c_{ik}$, and so the conditions of the following lemma are satisfied. We begin with Lemma 4.1.

Lemma 4.1. *Given $\ell \geq 2$ configurations $C_1, \ldots, C_\ell$, with corresponding probabilities $p_r > 0$, $r \in [\ell]$. If for $i \in [m]$, $\forall j, k \in [\ell]$ we have that $x_{ij} \leq x_{ik}$ implies $c_{ij} \leq c_{ik}$, then $POST(a_i, a_i) \geq PRI(a_i)$. Moreover, if for any $i \in [q]$, not all c_{ij}, $j \in [\ell]$ are the same, then $POST(a_i, a_i) > PRI(a_i)$.*

Proof. Consider without loss of generality action a_1. During this proof we use the notation of x_i for x_{1i} and c_i for c_{1i}, $i \in [\ell]$. Assume also without loss of generality that the configurations are ordered in such a way that $x_1 \leq x_2 \leq \ldots \leq x_\ell$ and thus $c_1 \leq c_2 \leq \ldots \leq c_\ell$. Note that $\mathrm{POST}\,(a_1, a_1) = \frac{1}{\sum_{i=1}^{\ell} p_i x_i} (\sum_{i=1}^{\ell} p_i x_i c_i)$ and $\mathrm{PRI}\,(a_1) = \sum_{i=1}^{\ell} p_i c_i$. Thus we must show that:

$$\sum_{i=1}^{\ell} p_i x_i c_i \geq (\sum_{i=1}^{\ell} p_i c_i)(\sum_{i=1}^{\ell} p_i x_i).$$

If all x_i are the same, then we clearly have equality and in this case $\mathrm{POST}\,(a_1, a_1) = \mathrm{PRI}\,(a_1)$. Otherwise, we will show that this inequality is true by decomposing the x_i

terms into x_1 and ϵ_i terms, $\epsilon_i \geq 0$ (and there exists at least one j with $\epsilon_j > 0$). For any $i \in \{2, \ldots, \ell\}$ we write $x_i = x_1 + \epsilon_1 + \ldots + \epsilon_{i-1}$. Consider only the summands in the above inequality that contain the term x_1. If $x_1 = 0$ then clearly the inequality holds for such summands. If $x_1 > 0$, we get the following chain of inequalities for the summands containing x_1:

$$\begin{aligned}
\sum_{i=1}^{\ell} p_i x_1 c_i &\geq (\sum_{i=1}^{\ell} p_i c_i)(\sum_{i=1}^{\ell} p_i x_1) \\
\sum_{i=1}^{\ell} p_i c_i &\geq (\sum_{i=1}^{\ell} p_i c_i)(\sum_{i=1}^{\ell} p_i) \\
\sum_{i=1}^{\ell} p_i c_i &\geq \sum_{i=1}^{\ell} p_i c_i,
\end{aligned}$$

so this inequality holds.

Now consider the summands in the inequality containing ϵ_j for $1 \leq j \leq \ell - 1$. We get the inequality:

$$\sum_{i=j+1}^{\ell} p_i \epsilon_j c_i \geq (\sum_{i=1}^{\ell} p_i c_i)(\sum_{i=j+1}^{\ell} p_i \epsilon_j).$$

If $\epsilon_j = 0$, the inequality holds. If $\epsilon_j > 0$, for that j showing the previous inequality is equivalent to showing

$$\sum_{i=j+1}^{\ell} p_i c_i \geq (\sum_{i=1}^{\ell} p_i c_i)(\sum_{i=j+1}^{\ell} p_i).$$

To show that this inequality is true, we decompose the c_i terms into c_1 plus δ_i terms. That is, $c_i = c_1 + \delta_1 + \ldots + \delta_{i-1}$, for $i = 1, \ldots, \ell - 1$. Consider first the c_1 term. If

$c_1 = 0$, again the inequality holds trivially. If $c_1 > 0$, we get the chain of inequalities

$$\begin{aligned}\sum_{i=j+1}^{\ell} p_i c_1 &\geq (\sum_{i=1}^{\ell} p_i c_1)(\sum_{i=j+1}^{\ell} p_i)\\ \sum_{i=j+1}^{\ell} p_i &\geq (\sum_{i=1}^{\ell} p_i)(\sum_{i=j+1}^{\ell} p_i)\\ \sum_{i=j+1}^{\ell} p_i &\geq \sum_{i=j+1}^{\ell} p_i,\end{aligned}$$

which holds. Next we consider the δ_k terms for $k \leq j+1$. If $\delta_k = 0$, the inequality clearly holds for summands containing this term. If $\delta_k > 0$, we get the inequality chain:

$$\begin{aligned}\sum_{i=j+1}^{\ell} p_i \delta_k &\geq (\sum_{i=k+1}^{\ell} p_i \delta_k)(\sum_{i=j+1}^{\ell} p_i)\\ \sum_{i=j+1}^{\ell} p_i &\geq (\sum_{i=k+1}^{\ell} p_i)(\sum_{i=j+1}^{\ell} p_i)\end{aligned}$$

which also holds. In particular, since $p_1 > 0$, we have that $(\sum_{i=j+1}^{\ell} p_i) < 1$, and so if $\delta_k > 0$, the inequality is strict. Finally, we consider the δ_k terms for $k > j+1$. If $\delta_k = 0$, the inequality holds trivially. If $\delta_k > 0$ we get the inequality chain:

$$\begin{aligned}\sum_{i=k}^{\ell} p_i \delta_k &\geq (\sum_{i=k}^{\ell} p_i \delta_k)(\sum_{i=j+1}^{\ell} p_i)\\ \sum_{i=k}^{\ell} p_i &\geq (\sum_{i=k}^{\ell} p_i)(\sum_{i=j+1}^{\ell} p_i),\end{aligned}$$

which also holds.

Now, we note that if not all c_i are the same for $i \in [\ell]$, it must be the case that there exists some j such that $\delta_j > 0$, and it follows that we must also have that $\epsilon_j > 0$.

As shown above, in such a situation, we obtain a strict inequality over the summands containing the term δ_j, and so the entire inequality, $\mathrm{POST}\left(a_1, a_1\right) > \mathrm{PRI}\left(a_1\right)$ must be strict.

□

Define by $a_{pri} := \operatorname{argmin}_a \mathrm{PRI}\left(a\right)$. Given a mediator over a fixed set of configurations, let X be the random variable denoting the cost of an arbitrary player when he decides to use action a_{pri}, i.e., $\mathbf{E}\left[X\right] = \sum_{j=1}^{\ell} p_j c_{a_{pri} j}$. Denote also by Y the random variable of the cost when following the advice of the mediator, i.e., $\mathbf{E}\left[Y\right] = \sum_{i=1}^{m} \mathrm{POST}\left(a_i, a_i\right) \mathbf{Pr}\left(\mathcal{E}_i\right) = \sum_{i=1}^{m} \sum_{j=1}^{\ell} p_j x_{ij} c_{ij}$. We have the following relationship between Y and X.

Lemma 4.2. *For any mediator we have* $\mathbf{E}\left[Y\right] \leq \mathbf{E}\left[X\right]$.

Proof. Assume without loss of generality that action a_1 is the action with a_{pri}. The constraints for a correlated Nash equilibrium are that for all actions a_i and a_j, $\mathbf{E}\left[\mathcal{C}_{a_i} | \mathcal{E}_{a_i}\right] \leq \mathbf{E}\left[\mathcal{C}_{a_j} | \mathcal{E}_{a_i}\right]$. These constraints imply that

$$\forall_{i:2 \leq i \leq q} : \sum_{j=1}^{\ell} p_j x_{ij} c_{ij} \leq \sum_{j=1}^{\ell} p_j x_{ij} c_{1j}.$$

Summing all of these $q-1$ inequalities together gives the single inequality, which we can rearrange as follows to show our result:

$$\begin{aligned}
\sum_{i=2}^{q}\sum_{j=1}^{\ell} p_j x_{ij} c_{ij} &\le \sum_{i=2}^{q}\sum_{j=1}^{\ell} p_j x_{ij} c_{1j} \iff \\
\sum_{j=1}^{\ell}\sum_{i=2}^{m} p_j x_{ij} c_{ij} &\le \sum_{j=1}^{\ell} p_j c_{1j} \sum_{i=2}^{q} x_{ij} \iff \\
\sum_{j=1}^{\ell}\sum_{i=2}^{q} p_j x_{ij} c_{ij} &\le \sum_{j=1}^{\ell} p_j c_{1j}(1 - x_{1j}) \iff \\
\sum_{j=1}^{\ell}\sum_{i=1}^{q} p_j x_{ij} c_{ij} &\le \sum_{j=1}^{\ell} p_j c_{1j} \iff \\
\mathbf{E}[Y] &\le \mathbf{E}[X].
\end{aligned}$$

□

We are now ready to prove Theorem 4.2.

Proof. Denote by $a_{post} := \text{argmin}_s \text{POST}(s, s)$ the action with minimum a posteriori cost. We will consider two cases.

Case 1: Not all actions have the same a posteriori cost. Then, we have:

$$\begin{aligned}
\mathbf{E}[Y] &> \text{POST}(a_{post}, a_{post}) \\
&\ge \text{PRI}(a_{post}) \text{ by Lemma 4.1} \\
&\ge \text{PRI}(a_{pri}) = \mathbf{E}[X].
\end{aligned}$$

Case 2: All action have the same a posteriori cost. In this case, we make use of the fact that there always must be some action that does not have equal costs in each configuration. Assume not. Then the cost of each action is the same in every configuration, and so any particular configuration must be a Nash equilibrium that achieves social cost equal to the social cost of the correlated equilibrium. Thus, we let a_x be some action that does not have the same cost in all configurations. Then we have:

$$\begin{aligned}\mathbf{E}[Y] &= \text{POST}(a_x, a_x) \\ &> \text{PRI}(a_x) \text{ by Lemma 4.1} \\ &\geq \text{PRI}(a_{pri}) = \mathbf{E}[X].\end{aligned}$$

In both cases we have $\mathbf{E}[Y] > \mathbf{E}[X]$. This however contradicts Lemma 4.2, hence there can not exist a correlated equilibrium achieving social cost less than the optimal Nash equilibrium. □

4.5 The Possibility of Mediation

We end this chapter on a positive note, by describing a simple congestion game where we can show that a mediator will improve the pure nash equilibrium solution. This simple game gives additional insight into why our mediator for the virus inoculation game works.

The game we consider is a variant of the *El Farol* game [3, 17, 11, 32]. El Farol is a[3] tapas bar in Santa Fe. Every Thursday night, a population of people decide whether or not to go to the bar. If too many people go, they will have a worse time than if they stayed home, since the bar will be too crowded. In our variant of the problem, we also assume that if too few people go, they will have a worse time than if they stayed home, because the bar will be too boring. We can model this as a non-atomic, symmetric congestion game as follows. There are two facilities e_1 and e_2, and two actions $a_1 = \{e_1\}$ and $a_2 = \{e_2\}$. For all $0 \leq x \leq 1$, $f_{e_1}(x) = 1/2$ and $f_{e_2}(x) = |1 - 2x|$.

We observe that the social cost in our game is minimized when the flow over both edges is 1/2, in which case, the social cost is 1/4. This configuration, however, is not

[3] very tasty

a Nash equilibrium. Pure Nash equilibria occurs when the top flow is 1/4 or the top flow is 3/4, for a social cost of 1/2. We now describe a mediator that improves upon the social welfare of the pure nash equilibrium.

Configuration C_1: The mediator advises all players to perform action a_1.

Configuration C_2: The mediator advises half of the players to perform action a_1, and advises the other half to perform action a_2.
For these two configurations C_1 and C_2 consider now the probability distribution $\mathcal{D}(\{C_1, C_2\})$ with $p_1 = 1/3$ and $p_2 = 2/3$.

Observation 4.1. *$\mathcal{D}(\{C_1, C_2\})$ is a mediator with social welfare 1/3. Moreover, 1/3 is the optimal value that can be obtained by a mediator.*

Proof. Define by $\mathcal{E}_i^s$, $i = 1, 2$, $s = 1, \dots, n$, the event that the mediator proposes to player s to go on the i'th edge and define by $\mathcal{C}_i^s$, $i = 1, 2$, $s = 1, \dots, n$, the cost for player s of going on the i'th edge. Since the mediator treats all players equally, we will leave out the index s. Therefore, for a mediator to implement a correlated Nash equilibrium, the following inequalities must hold:

$$\mathbf{E}\,[\mathcal{C}_2 \,|\, \mathcal{E}_1] \;\; \geq \mathbf{E}\,[\mathcal{C}_1 \,|\, \mathcal{E}_1], \tag{4.3}$$

$$\mathbf{E}\,[\mathcal{C}_1 \,|\, \mathcal{E}_2] \;\; \geq \mathbf{E}\,[\mathcal{C}_2 \,|\, \mathcal{E}_2]. \tag{4.4}$$

For the particular choice of $p_1 = 1/3$ and $p_2 = 2/3$, it is easy to see that both (4.3) and (4.4) are satisfied.

Now we show that 1/3 is the optimal value that can be obtained by any mediator. Let x_1 be the flow on e_1 and x_2 be the flow on e_2. The argument is as follows: for (4.3) to be satisfied, a configuration with $x_1 \in [0, 1/4] \cup [3/4, 1]$ has to be chosen, and among all these the configuration C_1 of the previous example is the one which has minimum total cost and the same time allows for the highest probabilities for

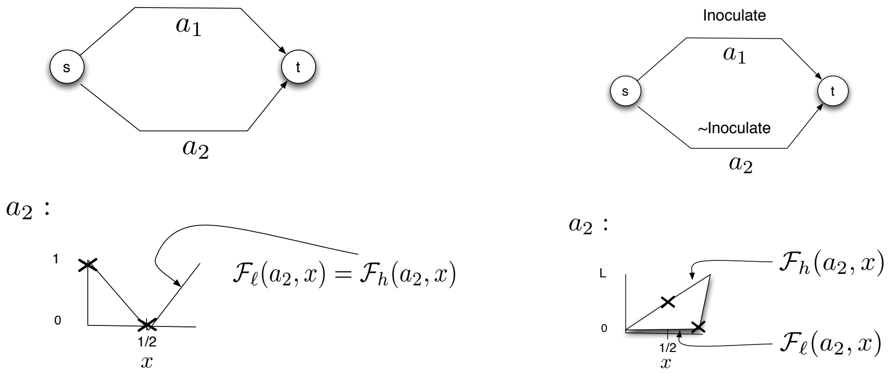


Figure 4.3: Congestion Games where mediation helps

configurations outside this interval. For the remaining values of $x_1 \in [1/4, 3/4]$, C_2 minimizes the total cost. □

Figure 4.3 illustrates the two games we have described for which mediation helps. The left subfigure portrays our variant of the El Farol game, where the cost of the top path a_1 is always 1/2 and the cost of the bottom path varies as shown in the plot below the graph. The values of $\mathcal{F}_\ell(a_2, x)$ and $\mathcal{F}_h(a_2, x)$ are equal, since in this game, when the flow through the top path is known, the cost of the bottom path is exactly determined. The two x's on the plot show the configurations used by the mediator. As implied by Theorem 4.2, for mediation to be effective, one of these x's must be below and to the right of the other on the plot. The right subfigure in Figure 4.3 portrays virus inoculation as a congestion game. The cost of the top path a_1 for this game is always 1. The cost of the bottom path, a_2, is any point in the polygon shown in the plot. We now have a polygon, rather than a line, because for a fixed number of nodes that do not inoculate, the cost of not inoculating varies depends on how the inoculated nodes are positioned on the grid. $\mathcal{F}_\ell(a_2, x)$ is the bottom border of this polygon and $\mathcal{F}_h(a_2, x)$ is the top border. Again the two x's on the plot show the configurations used by the mediator, and again it is critical that one of these x's be below and to the right of the other. For the virus inoculation problem, we needed a clever arrangement of the inoculated nodes in one of the configurations to achieve this.

4.6 Conclusion

We have shown that a mediator can improve the social welfare in some strategic games with a positive windfall of malice. We have also shown the limitation of this technique for certain games.

Chapter 5

Future Directions

This chapter is organized in two sections. In section 5.1 we describe some open problems regarding the *Worm versus Alert* game. In section 5.2 we describe some open problems, and some preliminary research on some of the problems which we have been working on.

5.1 Future work on the *Worm versus Alert* game

In our research in Chapters 3 and 4, the overlay network is a regular graph. We conjecture that our results can be generalized to many graphs in which the maximum degree a node can have is bounded by a constant d. Many other open problems remain in the *Worm versus Alert* game including: (1) tightening the upper and lower-bounds for the expansion needed in the overlay network to save almost all of the nodes; (2) developing other models for the spread of a dynamic process and its inhibitor over a network, and finding provably good strategies in these models; (3)finding out an alert spreading algorithm to handle worms which are not limited to infecting the network in $\theta(\log n)$ rounds, where false alerts spread to polylogarithmic number of nodes.

For bullet 3, consider the following strategy which the worms could use against the alert spreading algorithm. Assume that the worm has already been successful in infecting $\log n$ number of nodes in the network, without alerting any detector nodes. After that in each round the worm infects only one node. If it turns out to be a detector node, in the next round it takes over nodes in the neighborhood of this detector node to contain the spread of alerts. Let us assume that the time when the first detector node has been alerted is 0. Let Z_i be a random variable, giving the number of alerted nodes at time step i. So by assumption, $Z_0 = 1$. We can show the following

Lemma 5.1. *If* $(d\gamma < 1)$*, then* $\exists n_0$*, s.t* $Pr(Z_n = 0) = 1$ *for all* $n \geq n_0$*.*

Proof. To prove the theorem, we reason that this process is a Galton-Watson Branching Process. Each time the worm tries to take over the children of a alerted detector node, it can alerted a random number of detector nodes. The probability that each of them turn out to be a detector node is γ. Let μ represent the expected number detector nodes which are children of a detector node. We know that $\mu = d\gamma$. From [5] we know that
$E[Z_n] = \mu^n$.

Since $\mu < 1$, $E[\sum_{n=0}^{\infty} Z_n] = 1/(1-\mu) < \infty$. This implies that the probability of extinction is 1, or there exists an n_0 at which the number of detector nodes alerted will be zero. □

Note that if the precondition of this lemma holds, then only detector nodes will be alerted by this worm strategy. Assuming that τ is $\theta(\log \log n)$, the worm can resume its previous strategy of infecting only one node in each time step after TTL expires.

In our future attempts to solve this problem, we need to find an alert algorithm to counter this particular worm strategy, or prove that the current alert strategy is sufficient to counter this strategy for slow worms.

Another important direction of work, is to find algorithms for alerts which would prevent large scale infection of nodes in other kinds of networks.

5.2 Future work in designing mediators

In Chapter 4 we describe a technique using mediators to decrease the social cost of a well studied game with a high price of anarchy. Several open questions remain including the following. First, can we determine necessary and sufficient conditions for a game to allow a mediator that improves social welfare over the best Nash? In particular, can we find such conditions for general congestion games? What about arbitrary anonymous games? Second, for games where each player can choose among k actions, can we say how many configurations are needed by any mediator? Preliminary work in this direction shows that for 2 actions, sometimes more than 2 configurations are needed.

In the sections given below we present some very preliminary work on designing mediators for some problems which we have pursued.

5.2.1 Multiround Games

In this section we describe mediators for a multiround game where the number of rounds is finite but determined by a geometric random variable. Let us consider the multiround pollution game. In this game, in each round, each player can either choose to pollute and pay a cost of one or to not pollute and pay nothing. Each player pays an additional cost equal to the number of players who have decided to pollute in that round. The mediator, suggests actions to each player in each round, and comes to know of the strategies played out by the players in each round. Let there be at least n players where $n > 2$. After every round, let the probability that the game is over in the next round be p.

The optimal social cost for this game is n. We design a mediator, which is as follows: Ask all the players to not pollute,and if any one disobeys, ask every player to pollute after that in all the rounds.

The expected number of rounds the game will last is $1/p$. So the expected cost of not listening to the mediator is (1/p)(n-1). The expected cost of listening to the mediator is just 1 + (1/p). So for all p such that $(1 + (1/p)) \leq (1/p)(n-1)$, there is no reason not to listen to the mediator.

Another example of a multi-round game is the multi-round bandwidth sharing game. This game has n players. Each player has an infinite set of strategies. Let each player send x_i units of flow along a channel where $x_i = [0, 1]$. If $\Sigma_j x_j$ exceeds the channel capacity, then no player get any benefit. If $\Sigma_j x_j < 1$, then the utility of the player i is $x_i(1 - \Sigma_j x_j)$. In other words the utility of each player decreases as the flow in the channel increases.

It is also easy to see that for the multi-round bandwidth sharing game played over infinite rounds, a punitive deterministic multi-round mediator improves the optimal solution. The mediator at each round asks one player chosen uniformly at random to send 1/2 and all the other players to send 0. If ever any player decides not to follow the advice of the mediator, then in every successive round the mediator asks every player to send 1/(n+1), thus reducing the utility of this player for all successive rounds. For every rational player, the expected utility when listening to the player is $1/4\cdot$ (probability that it is chosen). The utility when not listening to the mediator is $1/(n+1)^2$. The optimal solution without a mediator is $1/(n+1)^2$ for each player in each round.

We would like to point out that a punitive multiround deterministic mediator does not help the multi-round pollution game over finite number of rounds where the number of rounds is known to every player. The basic idea is as follows. Every player would pollute in the last round, because there are no repercussions to not following the mediator in the last round. Since every player is selfish, it would thus try to minimize its cost in the last but one round by polluting. Thus every player would pollute in the last but one round as well. This analysis can be carried for every round with the result that every player pollutes in every round.

This raises the question, are there other multiround games with a finite number of rounds but the number of rounds being distributed according to distributions other than the geometric distributions, where mediators can help?

5.2.2 Non-Responsive games and do Mediators help?

In work in Chapter 4, we had the restriction of bad nodes not being players in the network. What if the bad players also have a choice of infecting nodes. In this section we predict the probabilities of infection and inoculation on some network topologies, for which there is a nash equilibria. In other words, for what probability values of infection and inoculation will we achieve a nash equilibria. We are exploring the question of using mediators in improving the social welfare in such equilibrium situations. The notation remains the same for all the network topologies covered in this section.

Complete Graph

Model: In our model we consider a complete graph on n vertices. The cost of good nodes is C_1 to inoculate and L to get infected where $C_1 \leq L$. A virus pays a cost of C_2 to create a virus minus the number of infected nodes. In this section we try to find the values of p_g and p_b for which there is a Nash equilibria with the following strategy where each good guy inoculates with probability p_g and each bad guy creates a virus with probability p_b and if it does, the virus infects a node in the network u.a.r and all nodes reachable from it which are not inoculated. There are a total of n nodes in the network. Let G be a r.v giving the utility of a good node and B be a r.v giving the utility of a bad agent. Let the expected utility of a good guy be *E(G)*, and for a virus be *E(B)*. Also let Q_i=$1/n(1+(n-1)(1-p_g)) = 1/n(n-(n-1)p_g)$.

Lemma 5.2. $E(G) = C_1 p_g + (1-p_g) \cdot p_b \cdot L \cdot Q_i$.

Proof. The probability that the virus infects this node is $p_b \cdot Q_i$, as this node is attacked with probability p_b/n and any other node among the n-1 nodes is attacked successfully with probability $p_b/n \cdot (1-p_g)$. Since it is a clique, this node could get a virus from any of the $n-1$ nodes. □

Lemma 5.3. $E(B) = C_2 p_b - p_b \cdot (1-p_g) n Q_i$.

Proof. The probability that a node is infected is $(1-p_g)p_b Q_i$, by arguments given above. There are n such nodes, so the expected number of nodes infected is $(1-p_g)p_b n Q_i$. . □

Let C(G,p) be the expected cost of a good node when $p_g = p$. Let $\Delta(G, \epsilon)$ be $C(G, p_g) - C(G, p_g + \epsilon)$. Similarly let C(B,p) be the expected cost of a node when $p_b = p$. Let $\Delta(B, \epsilon)$ be $C(B, p_b - C(B, p_b + \epsilon)$.

Fact 5.1. $\Delta(G, \epsilon)$= $\epsilon(C_1 - L p_b Q_i)$.

Fact 5.2. $\Delta(B, \epsilon) = \epsilon(C_2 - (1 - p_g)(n - (n-1)p_g)$.

Fact 5.3. *If* $C_2 = (n-1)p_g^2 - (2n-1)p_g + n$ *and* $C_1 = (1/n)Lp_b(n - (n-1)p_g)$ *then the system is in a Nash equilibria.*

Fact 5.4. *In a Nash equilibria,* $p_g = \frac{(2n-1) \pm \sqrt{4nC_2 - 4C_2 + 1}}{2(n-1)} = \theta(\frac{n - \sqrt{nC_2}}{n})$

Lemma 5.4. *In a Nash equilibria,* $p_b = \theta(\frac{C_1\sqrt{n}}{L\sqrt{C_2}})$

Proof. From fact 5.3 we get

$$\begin{aligned} p_b &= \frac{nC_1}{L(n-(n-1)p_g)} \\ &= \theta(\frac{C_1}{L(1-p_g)}) \\ &= \theta(\frac{C_1}{L(1-\frac{n-\sqrt{nC_2}}{n})})\text{(Substituting } p_g \text{ from fact 5.4)} \\ &= \theta(\frac{C_1\sqrt{n}}{L\sqrt{C_2}}) \end{aligned}$$

□

Let the social welfare of the grid be denoted by SW(Grid)

Lemma 5.5. *In a a Nash equilibria SW(Grid)=*$\theta(\sqrt{nC_2}C_1) + \theta(n - \sqrt{nC_2})$

Proof. The expected cost due to the bad nodes is $p_b(1-p_g)n(1-p_g)L$.
The expected cost due to the good nodes is np_g.
Substituting the values of P_B and P_G from fact 5.4 and 5.4, we get

$$\begin{aligned} SW(Grid) &= \theta(\frac{C_1\sqrt{n}}{L\sqrt{C_2}})(1-(\frac{n-\sqrt{nC_2}}{n}))^2 n.L + \theta(n-\sqrt{nC_2}) \\ &= \theta(\frac{C_1\sqrt{n}}{L\sqrt{C_2}})\frac{C_2}{n}nL + \theta(n-\sqrt{nC_2}) \\ &= \theta(\sqrt{nC_2}C_1) + \theta(n-\sqrt{nC_2}) \end{aligned}$$

□

Here we state the main theorem of this section.

Lemma 5.6. *As $n \to \infty$, $p_g \to 1$ and $p_b \to 1$.*

Proof. Solving for p_g in the quadratic equation in fact 5.3, we get $p_g = \frac{(2n-1) \pm \sqrt{4nc_2 - 4c_2 + 1}}{2(n-1)}$ which tends to 1 as $n \to 0$. Similarly solving for P_B we get $\frac{nC_1}{L}$ which tends to ∞ as $n \to \infty$, or $p_b = 1$. □

Corollary 2. *If $C_2 = \theta(n)$, then $p_b = \frac{C_1}{L}$.*

It is a corollary of lemma 5.4.

Lemma 5.7. *For n=1, there is a nash equilibria for $p_b = C_1/L$ and $p_g = 1 - c_2$.*

Proof. From lemma 5.2, 5.3, 5.1 and 5.2 we know that

- $E(G) = p_g C_1 + (1-p_g)p_b L$
- $E(B) = p_b C_2 - p_b(1-p_g)$
- $\Delta(G, \epsilon) = \epsilon(C_1 - p_b L)$

- $\Delta(B, \epsilon) = \epsilon(C_2 - (1 - p_g))$

Setting $\Delta(G, \epsilon)$ and $\Delta(B, \epsilon)$ equal to zero, we get $p_b = c_1/l$ and $p_g = (1 - c_2)$ respectively □

Grid

In this section, we find the values of p_g and p_b for which we achieve a Nash equilibria when the game is carried out on a Grid. The notations carry over from the previous section. The grid has n nodes, therefore the number of rows and columns in the grid is $\sqrt{n}$. The size of a connected component of uninoculated nodes is α consisting of $\sqrt{\alpha}$ rows and columns. Therefore every $(\sqrt{\alpha} + 1)^{th}$ row/column is a row/column of inoculated nodes.

Lemma 5.8. *$E(B) = p_b C_2 - p_b(1 - (\frac{1}{\sqrt{\alpha}+1})^2) \cdot \alpha$.*

Proof. Number of inoculated rows/columns $= \frac{\sqrt{n}}{\sqrt{\alpha}+1}$.
Number of nodes in the inoculated rows/columns $=\frac{\sqrt{n}}{\sqrt{\alpha}+1}\sqrt{n}$.
Number of nodes inoculated = number of nodes inoculated in rows + number of nodes inoculated in columns - the nodes which have been over counted because they were in the intersection of rows and columns, or
Number of nodes inoculated$=\frac{\sqrt{n}}{\sqrt{\alpha}+1}\sqrt{n} + \frac{\sqrt{(n)}}{\sqrt{\alpha}+1}\sqrt{n}$ - $\frac{\sqrt{n}}{\sqrt{\alpha}+1} \cdot \frac{\sqrt{n}}{\sqrt{\alpha}+1} = \frac{n}{\sqrt{\alpha}+1}(2 - \frac{1}{\sqrt{\alpha}+1})$.
Therefore the probability that a node is inoculated $= \frac{2}{\sqrt{\alpha}+1} - \frac{1}{(\sqrt{\alpha}+1)^2}$.
Therefore the probability that a node is uninoculated $= (1 - \frac{1}{\sqrt{\alpha}+1})^2$.
Since the total number of nodes in an uninoculated component is α,
$E(B) = p_b C_2 - p_b(1 - \frac{1}{\sqrt{\alpha}+1})^2 \alpha$. □

Fact 5.5. *$\Delta(B, \epsilon) = \epsilon(C_2 - (1 - \frac{1}{\sqrt{\alpha}+1})^2)\alpha$.*

Fact 5.6. *If $\Delta(B, \epsilon) = 0$, then $\alpha = \theta(C_2)$.*

Lemma 5.9. *For an uninoculated node the expected cost of choosing to not inoculate is* $p_b \frac{2\alpha L}{n}$.

Proof. If an inoculate node decided to not inoculate then it would be connecting 2 connected component of uninoculated nodes. The bad agent can infect this large connected component with probability $p_b \frac{2\alpha}{n}$. Since the cost of infection is L, the expected cost is $p_b \frac{2\alpha L}{n}$. □

Lemma 5.10. *For an uninoculated node, the expected cost to remain uninoculated is* $p_b \frac{L\alpha}{n}$.

Proof. The probability of getting infected by the bad agent is $p_b \frac{\alpha}{n}$. So the expected cost is $p_b \frac{L\alpha}{n}$. □

Lemma 5.11. *For a nash equilibria to hold,* $p_b \frac{2\alpha L}{n} > C_1 > p_b \frac{L\alpha}{n}$.

Proof. We get $p_b \frac{2\alpha L}{n} > C_1$ from lemma 5.9 and $C_1 > p_b \frac{L\alpha}{n}$ from lemma 5.10. □

Lemma 5.12. *If* $\Delta(B, \epsilon) = 0$, $p_b = \theta(\frac{C_1 L}{C_2 L})$.

Proof. From fact 5.6 we know that $\alpha = \theta(c_2)$. Substituting these values in the inequalities of lemma 5.11, we get $p_b > \frac{C_1 n}{\theta(C_2) L}$ and $p_b < \frac{C_1 n}{\theta(C_2) L}$. Therefore $p_b = \theta(\frac{C_1 n}{C_2 L})$. □

Corollary 3. *For* $\alpha = \theta(C_2)$, *there is a nash equilibria on the grid.*

Proof. This result is a corollary of lemma 5.12. □

Lemma 5.13. *If* $\alpha = \theta(C_2)$, *the social welfare of nodes on the grid is* $\theta(C_1 n)$.

Proof. The probability that the uninoculated nodes have been hit by a bad agent is $p_b(1 - (\frac{1}{\sqrt{\alpha+1}})^2)$. The total loss in that component is αL. The number of inoculated nodes is $n(\frac{2}{\sqrt{\alpha}+1} - \frac{1}{(\sqrt{\alpha}+1)^2})$. So the cost due to nodes which have inoculated is $n(\frac{2}{\sqrt{\alpha}+1} - \frac{1}{(\sqrt{\alpha}+1)^2})C_1$. Therefore the social welfare is $p_b(1-(\frac{1}{\sqrt{\alpha}+1})^2) \cdot \alpha L + n(\frac{2}{\sqrt{\alpha}+1} - \frac{1}{(\sqrt{\alpha}+1)^2})C_1$. Substituting the value of p_b from lemma 5.12 in the previous expression we get the social welfare as $\theta(C_1 n)\theta(1 - \frac{1}{\sqrt{C_2}}) + \theta(\frac{nC_1}{\sqrt{C_2}})$. In this expression $\theta(C_1 n)$ dominates. □

In conclusion we would like to add that we have not found any evidence of the mediator coming to our rescue for the virus inoculation game played in chapter 4. We also could not find a way for a mediator to help when there are m bad players who attack the network with the spoils divided equally among all of the players. An important future direction in this area would be to find games which improve the social welfare in the the case of non responsive players, and to explore the case of multiround games between non-responsive players.

5.2.3 Mediators for congestion control protocol

In this section, we decided to ascertain if the congestion control protocol as described below, has a high price of anarchy. We describe our preliminary results in some detail below.

Model: There are n agents who are competing to send a bit of information through a shared channel. Time consists of discreet time slots. If more than one agent tries to send there is a collision. A player leaves the game once it is successful in sending the bit. A strategy of a player is the probability of sending a message at that round. We assume that this game is *non-blocking*. i.e the transmission probability for every player is less than one. We assume that this game is *time independent* and *symmetric*, i.e the transmission probability in each round is dependent only on the number of remaining players in that round and not the round number, and the transmission probability is the same for all players. In our model the utility of any player decreases exponentially by a factor of $\alpha < 1$ in each successive round. The utility of sending it in the first round is 1. Let p be the nash equilibrium probability when there are k players remaining in the game. Let V_k be the expected cost for a player when there are k players remaining in the game.

Analysis: Suppose a player decides to deterministically send in every round. The expected cost in that case is

$$V(k) = (1-p)^{k-1} \cdot 1 + [1-(1-p)^{k-1}]\alpha V(k) \tag{5.1}$$

If this is a nash equilibrium it should be the case that the value of $V(k)$ when probability is 0, 1 or p should be the same and all of them should be best responses.

Also

$$V(k) = (k-1)p(1-p)^{k-2}\alpha V(k-1) + [1-(k-1)p(1-p)^{k-2}]\alpha V(k) \tag{5.2}$$

The first term on R.H.S calculates the probability of some player other than the player being considered being able to send through the channel. The second term on R.H.S considers the probability that no player is able to send.

Theorem 5.7. $\frac{1}{\alpha(k-1)} = \left[\frac{1}{1-\alpha)V(k)^{1/k-1}(1-\alpha(V(k))^{1-1/(k-1)}}\right](V(k-1)-V(k))$.

Proof. Equation 5.1 and 5.2 imply the identity

□

Let $V*(k)$ be the optimal value of $V(k)$. Therefore $V*(k)$=$1/k(\Sigma_{i=0}^{k-1}\alpha^i)$. For $\alpha = 1-1/100$ our empirical results suggested a high price of anarchy for this model.

References

[1] Good Hash Table Primes. http://planetmath.org/encyclopedia/GoodHashTablePrimes.html.

[2] Ittai Abraham, Danny Dolev, Rica Gonen, and Joe Halper. Lower bounds on implementing robust and resilient mediators. In *IACR Theory of Cryptography Conference(TCC)*, 2008.

[3] Brian Arthur. Bounded rationality and inductive behavior (the el farol problem). *American Economic Review*, 84:406–411, 1994.

[4] James Aspnes, Kevin Chang, and Aleksandr Yampolskiy. Inoculation strategies for victims of viruses and the sum-of-squares partition problem. In *ACM Symposium on Discrete Algorithms (SODA)*, 2005.

[5] James Aspnes, Navin Rustagi, and Jared Saia. Worm Vs Alert: Who wins in a battle for control of a Large Scale Network. In *Proceedings of the Principles of Distributed Systems; 11th International Conference(OPODIS)*, 2007.

[6] Robert J. Aumann. Subjectivity and correlation in randomized games. *Mathematical Economics*, 1:67–96, 1974.

[7] Moshe Babaioff, Robert Kleinberg, and Christos H. Papadimitriou. Congestion games with malicious players. In *ACM Conference on Electronic Commerce*, 2007.

[8] Stephen Baker and Brian Grow. Gambling Sites, This Is A Holdup, 2005. http://www.businessweek.com/magazine/content/04_32/b3895106_mz063.htm.

[9] Maria-Florina Balcan, Arvin Blum, and Yishay Mansour. Improved equilibria via public service advertising. In *ACM Symposium on Discrete Algorithms (SODA)*, 2009.

[10] Bela Bollobas. *Random Graphs.* Academic Press, 1985.

[11] Damien Challet, Matteo Marsili, and Gabriele Ottino. Shedding light on el farol. *Physica A: Statistical Mechanics and Its Applications*, 332:469–482, 2003.

[12] George Christodoulou and Elias Koutsoupias. On the price of anarchy and stability of correlated equilibria of linear congetion games. In *Proceedings of the European Symposium on Algorithms(ESA)*, 2005.

[13] Colin Cooper, Martin Dyer, and Catherine Greenhill. Sampling regular graphs and a peer-to-peer network. In *Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete algorithms (SODA)*, 2005.

[14] Manuel Costa, Jon Crowcroft, Miguel Castro, Anthony Rowstron, Lidong Zhou, Lintao Zhang, and Paul Barham. Vigilante: End-to-end containment of internet worms. In *Symposium on Operating System Principles (SOSP)*, 2005.

[15] Manuel Costa, Jon Crowcroft, Miguel Castro, and Antony Rowstron. Can we contain internet worms? In *Proceedings of the 3rd Workshop on Hot Topics in Networks (HotNets-III)*, 2004.

[16] Aaron Davis. Computer Worm Snarls Web, 2004. www.bayarea.com/mld/mercurynews/5034748.html.

[17] M. de Cara, O. Pla, and F. Guinea. Competition, efficiency and collective behavior in the “el farol” bar model. *The European Physics Journal B*, 10, 1998.

[18] Josep Díaz, Dieter Mitsche, Navin Rustagi, and Jared Saia. On the power of mediators. In *WINE '09: Proceedings of the 5th International Workshop on Internet and Network Economics*, pages 455–462, Berlin, Heidelberg, 2009. Springer-Verlag.

[19] Daniel R. Ellis, John G. Aiken, Kira S. Attwood, and Scott D. Tenaglia. A behavioral approach to worm detection. In *WORM '04: Proceedings of the 2004 ACM workshop on Rapid malcode*, pages 43–53, New York, NY, USA, 2004. ACM.

[20] Jason Franklin, Vern Paxon, Adrian Perrig, and Stefan Savage. An inquiry into the nature and causes of the wealth of internet miscreants. In *CCS '07: Proceedings of the 14th ACM conference on Computer and communications security*, pages 375–388, New York, NY, USA, 2007. ACM.

[21] Martin Garvey. Phishing Attacks Show Sixfold Increase This Year, June 2005. http://www.informationweek.com/story/showArticle.jhtml?articleID=164302582.

[22] Marina Gelastou, Marios Mavronicolas, Vicky Papadopoulou, Anna Philippou, and Paul Spirakis. The power of the defender. In *ICDCSW '06: Proceedings of the 26th IEEE International ConferenceWorkshops on Distributed Computing Systems*, page 37, Washington, DC, USA, 2006. IEEE Computer Society.

[23] Garret Hardin. The tragedy of the commons. *Science*, xx:1243–47, 1968.

[24] Libin Jiang, Venkat Anantharam, and Jean Walr. 1 efficiency of selfish investments in network security.

[25] Robert O'Harrow Jr. Internet Worm Unearths New Holes, 2003. www.securityfocus.com/news/2186.

[26] Hyanh-Ah Kim and Brad Karp. Autograph: Toward automated, distributed worm signature detection. In *Proceedings of the 13th Usenix Security Symposium (Security 2004)*, 2004.

[27] Elias Koutsoupias and Christos Papadimitriou. Worst-case equilibria. In *in Proceedings of the 16th Annual Symposium on Theoretical Aspects of Computer Science*, pages 404–413, 1999.

[28] Robert Lemos. Slammer Attacks May Become Way of Life for the Net, 2003. http://www.news.com/2009-1001-983540.html?tag=fd_lede2_hed.

[29] John Leyden. Phishers Tapping Botnets to Automate Attack, 2004. http://www.theregister.co.uk/2004/11/26/anti-phishing_report/.

[30] John Leyden. ISPs urged to throttle spam zombies, 2005. http://www.theregister.co.uk/2005/05/24/operation_spam_zombie/.

[31] Dan Liet. Most Spam Generated by Botnets, Says Expert, 2004. http://news.zdnet.co.uk/internet/security/0,39020375,39167561,00.htm.

[32] Hilmi Lus, Cevat Onur Aydin, Sinan Keten, Hakan Ismail Unsal, and Ali Rana Atiligan. El farol revisited. *Physica A: Statistical Mechanics and Its Applications*, 346:651–656, 2005.

[33] Marios Mavronicolas and Vicky Papadopoulou. A graph-theoretic network security game. In *Proceedings of the First International Workshop on Internet and Network Economics*, pages 969–978. SpringerVerlag, 2005.

[34] Marios Mavronicolas and Vicky Papadopoulou. A network game with attacker and protector entities. In *Proceedings of the 16th Annual International Symposium on Algorithms and Computation*, pages 05–13. Springer, 2005.

[35] Marios Mavronicolas, Vicky Papadopoulou, Anna Philippou, and Paul Spirakis. A network game with attackers and a defender. *Algorithmica*, 51(3):315–341, 2008.

[36] Michael Mitzenmacher and Eli Upfal. *Probability and Computing*, chapter 13: Pairwise Independence and Universal Hash Functions, pages 314–335. Cambridge, 2006.

[37] David Moore, Vern Paxson, Stefan Savage, Colleen Shannon, Stuart Staniford, and Nicholas Weaver. Inside the Slammer Worm. *IEEE Security and Privacy journal*, 1(4):33–39, 2003.

[38] David Moore, Colleen Shannon, Geoffrey M. Voelker, and Stefan Savage. Internet quarantine: Requirements for containing self-propagating code. 2003.

[39] Thomas Moscibroda, Stefan Schmid, and Roger Wattenhofer. When selfish meets evil: Byzantine players in a virus inoculation game. In *Principles of Distributed Computing(PODC)*, 2006.

[40] James Newsome, Brad Karp, and Dawn Song. Polygraph:automatically generating signatures for polymorphic worms. In *Proceedings of the IEEE Symposium on Security and Privacy*, 2005.

[41] Commtouch 2006 Spam Trends Report: Year of the Zombies.

[42] Christos H. Papadimitriou and Tim Roughgarden. Computing correlated equilibria in multi-player games. *J. ACM*, 55(3):1–29, 2008.

[43] Roberto Preatoni. Prophet Mohammed protest spreads on the digital ground. Hundreds of cyber attacks against Danish and western webservers spreading rage in the name of Allah, February 2006. http://213.219.122.11/en/news/read/id=205987/.

[44] Anirudh Ramachandran, Nick Feamster, and Santosh Vempala. Filtering spam with behavioral blacklisting. In *CCS '07: Proceedings of the 14th ACM conference on Computer and communications security*, pages 342–351, New York, NY, USA, 2007. ACM.

[45] Paul Roberts. Al-Jazeera hobbled by DDOS attack, 2003. http://www.infoworld.com/article/03/03/26/HNjazeera_1.html.

[46] Aaron Roth. The price of malice in linear congestion games. In *Workshop on Internet and Network Economics(WINE)*, 2008.

[47] Antony I. T. Rowstron and Peter Druschel. Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems. In *Proceedings of the IFIP/ACM International Conference on Distributed Systems Platforms Heidelberg*, pages 329–350, 2001.

[48] Ola Rozenfeld and Moshe Tennenholtz. Strong and correlated strong equilibria in monotone congestion games. In *Proceedings of the Workshop on Internet and Network Economics (WINE)*, 2006.

[49] Vyas Sekar, Yinglian Xie, Michael K. Reiter, and Hui Zhang. A multi-resolution approach forworm detection and containment. In *DSN '06: Proceedings of the International Conference on Dependable Systems and Networks*, pages 189–198, Washington, DC, USA, 2006. IEEE Computer Society.

[50] Srinivas Shakkottai and Rayadurgam Srikant. Peer to peer networks for defense against internet worms. In *Proceedings of the 2006 workshop on Interdisciplinary systems approach in performance evaluation and design of computer and communications sytems*, 2006.

[51] Sumeet Singh, Cristian Estan, George Varghese, and Stefan Savage. The early-bird system for real-time detection of unknown worms, 2003. Technical Report CS2003-0761, University of California, San Diego.

[52] Sumeet Singh, Cristian Estan, George Varghese, and Stefan Savage. Automated worm fingerprinting. In *Proceedings of the 6th ACM/USENIX Symposium on Operating System Design and Implementation (OSDI)*, 2004.

[53] Eugene Spafford. Exploring Grand Challenges in Trustworthy Computing. http://digitalenterprise.org/seminar/spafford2.html.

[54] Agelika Steger and Nicholas C. Wormald. Generating random regular graphs quickly. In *Combinatorics, Probability and Computing*, 1999.

[55] Will Sturgeon. Denial-of-service-attack victim speaks out, 2005. http://www.zdnetasia.com/insight/business/0,39044868,39233051,00.htm.

[56] Péter Ször and Peter Ferrie. Hunting for metamorphic. In *In Virus Bulletin Conference*, pages 123–144, 2001.

[57] Chris Talbot. Phishing Attacks Up More Than 200% in May, says IBM, 2005. http://www.integratedmar.com/ecl-usa/story.cfm?item=19703.

[58] Milan Vojnovic and Ayalvadi Ganesh. On the effectiveness of automatic patching. In *ACM Workshop on Rapid Malcode (WORM)*, 2005.

[59] Helen J. Wang, Chuanxiong Guo, Daniel R. Simon, and Alf Zugenmaier. Shield: Vulnerability-driven network filters for preventing known vulnerability exploits. In *Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 193–204. ACM, 2004.

[60] Martyn Williams. http://www.pcworld.com/article/109163/slammer_was_fastest_spreading_worm.

[61] Matthew M. Williamson. Throttling viruses: Restricting propagation to defeat malicious mobile code. 2002.

[62] Vinod Yegneswaran, Jonathan T. Giffin, Paul Barford, and Somesh Jha. An architecture for generating semantics-aware signatures. In *Proceedings of the 14th USENIX Security Symposium*, pages 97–112. Baltimore, MD, USA, 2005.

[63] Lidong Zhou, Lintao Zhang, Frank McSherry, Nicole Immorlica, Manuel Costa, and Steve Chien. A first look at peer-to-peer worms: Threats and defenses. In *International Symposium on Peer-to-peer Systems (IPTPS)*, 2005.

[64] Jianwei Zhuge, Thorsten Holz, Chengyu Song, Jinpeng Guo, Xinhui Han, and Wei Zou. Studying malicious websites and the underground economy on the chinese web. informatik. Technical report, on the Chinese web. Workshop on the Economics of Information Security (WEIS, 2007.