Applying Game Theory to Analyze Attacks and Defenses in Virtual Coordinate Systems

Sheila Becker†, Jeff Seibert*, David Zage*, Cristina Nita-Rotaru* and Radu State†
*Purdue University
West Lafayette, IN 47906, USA
Email: {jcseiber, zagedj, crisn}@cs.purdue.edu

†University of Luxembourg
6 rue Coudenhove-Kalergi, L-1359 Luxembourg, Luxemburg
Email: {sheila.becker, radu.state}@uni.lu

***Abstract*—Virtual coordinate systems provide an accurate and efficient service that allows hosts on the Internet to determine latency to arbitrary hosts based on information provided by a subset of participating nodes. Unfortunately, the accuracy of the service can be severely impacted by compromised nodes providing misleading information.**

We define and use a game theory framework in order to identify the best attack and defense strategies assuming that the attacker is aware of the defense mechanisms. Our approach leverages concepts derived from the Nash equilibrium to model more powerful adversaries. We consider attacks that target the latency estimation (inflation, deflation, oscillation) and defense mechanisms that combine outlier detection with control theory to deter adaptive adversaries. We apply the game theory framework to demonstrate the impact and efficiency of these attack and defense strategies using a well-known virtual coordinate system and real-life Internet data sets.

***Keywords*-virtual coordinate systems; game theory; security;**

I. Introduction

Numerous peer-to-peer (P2P) applications (*e.g.*, BitTorrent, Skype) can leverage network topology information to optimize their performance. In order to avoid the costs associated with actively monitoring all of the nodes in the network, virtual coordinate systems provide a service that allows hosts on the Internet to accurately estimate the latency between arbitrary hosts with minimal network overhead. Nodes rely on information reported by a subset of participating nodes and on latency measurements to this subset to provide low-error latency estimation at any arbitrary node in the network. The accuracy of the service and in turn the performance of any application relying on the virtual coordinate service, can be severely impacted if nodes do not behave correctly. For example, malicious nodes can lie in the reports about their own latencies, or they can influence the measurements conducted by honest nodes. As a result, such honest nodes will compute higher or lower predictions than the actual latencies, or the entire system will destabilize. Such attacks are referred to as inflation, deflation, and oscillation attacks.

Previous work has studied the consequences of attacks against virtual coordinate systems and proposed several techniques to mitigate their vulnerabilities [1]–[6]. However, these solutions are validated only through experiments and consider the effectiveness of the defense strategies under the restrictive assumption that an attacker is unaware of or fails to take into account the associated defense mechanisms. This assumption is unrealistic as attackers are often aware of the defense mechanisms employed by the system and good system design principles state that security should not rely on the secrecy of the defense algorithm [7]. Given that several attacks and several defense techniques exist, there is a need for a systematical evaluation identifying what are the best attack and defense strategies.

In this paper, we systematically study attack and defense techniques for virtual coordinate systems. We develop a game theoretical model that relies on well-known equilibrium concepts in order to assess the strategic interactions between the attacks and defenses. Game theory provides powerful tools that allow us to model an advanced adversary who knows how and what defense strategies are used and can adjust his attack strategies accordingly. This framework then allows us to draw out conclusions such as understanding what are the strengths and weaknesses of both defenses and attacks.

We conduct our work focusing on the representative virtual coordinate system Vivaldi [8], using two real-life Internet data sets. We consider a Byzantine adversary that controls a percentage of the nodes in the system and conducts three different types of attacks: inflation, deflation, and oscillation. These three attacks correspond to when a node reports large coordinates far away from the origin, small coordinates near the origin, and randomly chosen coordinates, respectively. We consider defense strategies based on outlier detection since they have been shown to provide good results under the assumption the attacker does not know the defense strategy [6]. Specifically, we assume the defender attempts to mitigate the attacks using three defense techniques based on outlier detection: spatial, temporal, and spatial-temporal. The defense techniques also use realistic system design assumptions that make them easily integrated into current virtual coordinate systems, *i.e.*, they do not rely on the triangle equality [1], do not require extra node sets and network communication [4, 5], and do not require trusted parties [2, 3].

A critical component of an outlier detection mechanism is the threshold that is used to decide if a data point is accepted by the system or is suspected of coming from a malicious node. Many outlier detection schemes use a fixed threshold, usually determined experimentally. Such an approach is inflexible, prone to errors, and may be exploited by an adversary to remain undetected. We leverage control theory to design an adaptive threshold technique to improve the threshold selection and include outlier detection mechanisms based on adaptive thresholds in our study. Our contributions include:

- We model rational attackers in virtual coordinate systems using the Nash equilibrium and irrational attackers using the quantal response equilibrium. From the defender side, we use game theory to tune our defensive mechanisms in order to mitigate the attacks.
- Using our framework, we determined that for large networks (*i.e.*, the King topology), the inflation attack has the greatest impact on the system. To defend the system, we find that spatial-temporal outlier detection is the most effective technique given lower spatial outlier thresholds (*e.g.*, ≤ 1.5) and both spatial-temporal and spatial outlier detection provide similar defense performance for higher thresholds. Furthermore, our analysis finds that, independent of the game strategy or the error metric selected, a spatial outlier threshold of 1.25 results in the best system performance, which is smaller than the value found in previous work.
- We found that the resulting strategy profiles for smaller networks (*i.e.*, the AMP topology) are not as homogeneous as those for the larger King topology, with most of the resulting strategy profiles consisting of a mixed strategy. For example, given the spatial outlier threshold of 1.75, the attacker has the greatest payoff while applying all three attacks with their given probabilities using only 10% malicious nodes. The countermeasure profile looks similar, applying each of the three defense techniques. Both the percentage of malicious nodes necessary to efficiently create the greatest negative impact and the attack and defense profiles have not previously been systematically explored.
- We found that when comparing strategies using a fixed threshold with strategies using an adaptive threshold selection for the outlier detection, the adaptive threshold is more effective in defending against attacks than a fixed threshold. Our analysis shows that when an attacker has as goal disturbing the network as much as possible, using inflation with 30% attackers is the best attack strategy. If the attacker wants to remain also undetected then oscillation and deflation attacks with 10% attackers are the best rational choice. We found that the best parameters for the adaptive threshold is to use the 75^{th} percentile of the prediction error and with a value for the constant c of 0.08 to update the threshold, where c is a system parameter that captures the importance given to the prediction error when updating the threshold.

The rest of the paper is organized as follows. We provide background information on virtual coordinate systems in Section II. We then describe the attacks and defenses we consider in our theoretical model in Section III and describe our game theory-based model in Section V. We describe our experimental results and our findings in Section VI. We overview related work in Section VII and conclude our paper in Section VIII.

II. System Model

We consider a decentralized virtual coordinate system. Decentralized virtual coordinate systems are designed to efficiently create and maintain a stable set of coordinates that accurately predict the latency between nodes without using fixed infrastructure nodes. Although each specific virtual coordinate system differs in some details, they follow a common design and operation. We selected the representative virtual coordinate system, Vivaldi, since it is a mature system, conceptually easy to understand and visualize, and has been shown to produce low error embeddings [8].

Vivaldi is a fully decentralized system which assigns each host synthetic coordinates in a multi-dimensional Euclidean coordinate space, offering a good tradeoff between performance and overhead [8, 9]. The Vivaldi algorithm is based on a spring relaxation problem in which each pair of neighbor nodes is attached by a spring and the current length of the spring is the estimated round-trip times (RTT) between the nodes. Tension on the logical springs causes the nodes to move through the coordinate space as each node attempts to minimize the difference between current spring lengths (estimated RTT) and the spring lengths at rest (actual RTT). By minimizing the tension across all of the springs in the network, the protocol minimizes the error for the system.

Initially, each node is assigned a random coordinate and establishes a reference set of peer nodes with which to exchange periodic updates. As nodes communicate with their reference set peers, they receive latency information that is used to update their coordinates. Algorithm 1 shows how a node i updates its coordinate x_i and error e_i as a result of minimizing the tension of the spring with remote node j. Node i updates its own coordinate and error based on the tuple consisting of the remote node's coordinate x_j, the remote node's relative error with respect to its coordinate, e_j (both directly reported by node j), and the latency from node i to node j, RTT_{ij} (measured by node i). First, the algorithm calculates the observation confidence w (line 1) and relative error e_s (line 2). The relative error e_s expresses the accuracy of the coordinate in comparison to the measured network latency. Next, node i updates its local error (line 4) using an exponentially-weighted moving average with the weight α (line 3). Finally, the node calculates the movement dampening factor (line 5) and updates its coordinate (line 6). Both c_e and c_c are constants acting as system parameters.

As the nodes update their coordinates and the system stabilizes, the average system error is on the order of a few percent. Once the coordinate system has stabilized, the latency (*i.e.*, RTT) between two nodes is trivially estimated by computing the Euclidean distance between their coordinates. For further details of the protocol, we refer the reader to the work by Dabek *et al*. [8].

Algorithm 1: Vivaldi Coordinate Update

Input: Remote node observation tuple ($\langle x_j,\ e_j,\ RTT_{ij}\rangle$)
Result: Updated local node coordinate and error ($x_i,\ e_i$)

$w = e_i/(e_i + e_j)$
$e_s = |\|x_i - x_j\| - RTT_{ij}|/RTT_{ij}$
$\alpha = c_e \times w$
$e_i = (\alpha \times e_s) + ((1 - \alpha) \times e_i)$
$\delta = c_c \times w$
$x_i = x_i + \delta \times (RTT_{ij} - \|x_i - x_j\|) \times u(x_i - x_j)$

The accuracy of the overall virtual coordinate system is measured by the *system prediction error* defined as:

$$Error_{pred} = |RTT_{Act} - RTT_{Est}|$$

where RTT_{Act} is the measured RTT between two nodes and RTT_{Est} is the RTT computed using the coordinates derived by the virtual coordinate system. Intuitively, the lower the system prediction error, the more accurate are the predicted RTTs.

III. Attack Strategies

In this section, we describe the attacker model and the attack strategies we consider in this work.

A. Attacker Model

We consider a Byzantine adversary model, where a bounded percentage of nodes are malicious and lie during the information exchange with other honest nodes or influence the measurement conducted by honest nodes (*i.e.*, RTT). The set of malicious nodes may collude.

We assume a malicious adversary has access to all of the data at a node that any legitimate user would have (insider access), including the cryptographic keys stored at a node. This access can be the result of the adversary bypassing the authentication mechanisms or compromising a node through other means. As malicious nodes have insider access, nodes cannot be completely trusted although they are authenticated.

In order to quantitatively compare the effect of the adversaries on the accuracy of virtual coordinate systems, we utilize the *relative error* defined as:

$$Error_{rel} = \frac{Error_{attack}}{Error_{no\_attack}}$$

where $Error_{attack}$ is the system prediction error measured when the system is under attack and $Error_{no\_attack}$ is the system prediction error when all nodes are benign and no attack takes place. A relative error greater than one indicates a degradation in accuracy and a value less than one indicates a better estimation accuracy than the baseline.

We also assume that the adversary knows how and what defense strategies are used and can adjust his attack strategies accordingly. In the context of virtual coordinate systems we assume that a malicious node may lie about its coordinate and its error in the reports sent to honest nodes, and influence the RTT measurements conducted by honest nodes by delaying the response. Below we show how an attacker can define specific attack strategies based on these malicious actions.

B. Attacks Against Accuracy of Virtual Coordinates

The correct operation of virtual coordinate systems is dependent on the assumption that the peer nodes are altruistic and respond with correct metrics to queries from any node computing its corresponding coordinates. An attacker controlling reference set nodes has the ability to influence the coordinate maintenance process by manipulating the information, such as the remote node error and coordinates, returned in response to a query. By blindly accepting this malicious information, a correct node computes incorrect coordinates.

In manipulating the information it reports as a peer node, a malicious node is able to influence a victim node to move away from its correct position by either pushing the node away from or pulling it closer to the malicious node's reported coordinates. For example, a malicious node can attract a victim node towards a random position and away from the victim's correct position by reporting a false position and having a low estimated error. In addition to manipulating reported information, a malicious node may manipulate the measurements conducted by delaying its query responses, causing victim nodes to erroneously update their coordinates to accommodate for the additional delay. Each of the attacks can be classified into one of three categories: *coordinate inflation* attacks that result in coordinate mappings farther from the correct location, *coordinate deflation* attacks that result in victim nodes having incorrect coordinates due to not performing necessary coordinate updates, and *coordinate oscillation* attacks which report varying coordinates and errors that cause disorder in the system. In the end, each of these techniques distorts the coordinate space and have a long-lasting impact on the overall system.

Inflation, deflation, and oscillation attacks can be caused by a combination of lying about error and coordinates, and delaying messages during the RTT measurement. In our analysis we consider the following specific attack strategies:

- **Inflation**: malicious nodes report a large coordinate and a small error
- **Deflation**: malicious nodes report a small coordinate and a small error
- **Oscillation**: malicious nodes report a random coordinate, a small error, and influence the honest nodes to measure a large RTT by delaying the response.

IV. Defense Strategies

The defense strategies we consider are based on outlier detection. We selected outlier detection because previous work showed experimentally that outlier detection can be an effective mechanism in improving the accuracy of virtual coordinate systems in the presence of attacks under the restrictive assumption that an attacker is unaware of the associated defense mechanisms.

An important configuration parameter for outlier detection is the threshold that is used to decide if a data point is accepted by the system or is suspected of coming from a malicious node. We first overview spatial and temporal outlier detection

defenses that use a fixed threshold, then show how control theory techniques can be leveraged to make the threshold adaptive and exemplify such a strategy for spatial outlier detection.

A. Outlier Detection

Outlier detection can be used to identify malicious behavior and take action to mitigate its effects. Instead of allowing malicious coordinate mappings to occur and then trying to detect them, statistical outlier detection reduces the likelihood of a node computing incorrect coordinates by filtering out malicious updates. Each node independently performs outlier detection before updating its coordinates in order to identify and filter out outliers in the received metrics. As the evidence of malicious activity is distributed across space and time we consider both spatial and temporal outlier detection techniques.

Spatial outlier detection identifies observations which are inconsistent with their surrounding neighbors, forcing nodes to report metrics consistent with what other peers are reporting. When a node queries a peer, it receives an observation tuple (as seen in Algorithm 1). Upon receiving the tuple, the node records the response and tracks the most recent u updates in a queue. The size of the history queue, u, is equal to the size of the reference set which allows the queue, on average, to contain one entry from each reference set nodes. Once the tuple has been received, the node first computes the centroid of the data set consisting of observation tuples from the stored u updates. The node then computes the Mahalanobis distance [10] between the received observation tuple and the centroid and accepts the update if it is less than a designated spatial outlier threshold. We note that this technique is an instance of spatial outlier detection since it examines metrics across various system nodes and not time.

Temporal outlier detection identifies inconsistencies in the metrics over time, forcing a node to report metrics consistent with what it has reported in the past. A node tracks the temporal centroid of the observation tuple from each node in its peer set and the change in the reported coordinates using incremental learning. We assume each of the reported metrics is statistically independent, necessitating the storage of just the mean, standard deviation, and sample count computed from the received query responses over time. We note that the assumption of statistical independence is reasonable, even though nodes may collude, as the temporal outlier detection is computed individually for each node. Once a query response is received from a remote node, the node performing the detection compares the received observation tuple with the corresponding temporal centroid using the "simplified Mahalanobis distance" presented by Wang and Stolfo [10] and based on a temporal threshold. The tuple is accepted if the distance to the temporal centroid is below the threshold.

Finally, *spatial-temporal outlier detection* takes advantage of both techniques by combing them using a codebook technique similar to that by Jiang and Cybenko [11]. As a node receives observation tuples, it checks each one to ensure that the tuple is not a spatial or temporal outlier. If the tuple is found to be an outlier, the update is discarded. Otherwise, it is used to update the receiver node's coordinates. For further details on the detection techniques, we refer the reader to [6].

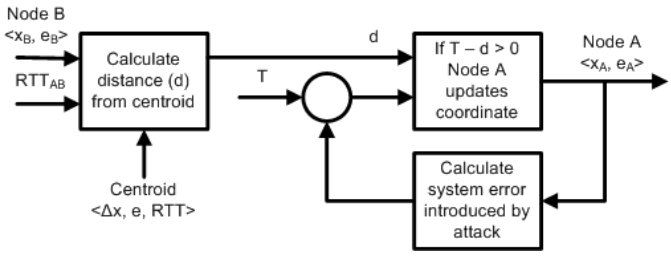


Fig. 1. Block diagram describing how we integrate control theory. For example, if updates that increase the system prediction error bypass the outlier detection then the spatial threshold is lowered.

B. Adaptive Threshold Selection

Many outlier detection schemes, including the ones proposed in [6] use a fixed threshold usually determined experimentally. Such an approach is inflexible, prone to errors, and may be exploited by an adversary to remain undetected. Below, we show how by leveraging control theory [12, 13] we design an adaptive threshold technique to improve the threshold selection.

Figure 1 shows how the adaptive threshold selection is integrated with the outlier detection mechanism. A feedback control loop is regularly updating the spatial threshold with the objective to tighten the threshold and adapt to attacks. The update of the threshold is based on the observation that more severe attacks (as per the nature of the attack and percentage of malicious nodes) will result in higher differences between the estimated RTT (predicted by the coordinates resulted from the virtual coordinate system) and the actual RTT. Specifically, at time n, the new threshold T_n is updated based on the threshold at time $n-1$, T_{n-1}, and the difference between the current prediction error $Error_{attack}(n)$ and an ideal value for the prediction error $Error_{no\_attack}(n)$ as follows:

$$T_n = T_{n-1} - c\frac{(Error_{attack}(n) - Error_{no\_attack}(n))}{RTT_{Est}(n)} \quad (1)$$

where c is a constant to define the importance of the prediction error $Error_{attack}(n)$ and $RTT_{Est}(n)$ is the current estimated RTT. The prediction error $Error_{attack}(n)$ is based on all the prediction errors calculated during each update of each single node, one can either take an averaged value or percentile values. The average value is more likely to be affected by potential malicious values that bypassed the outlier detection. Thus, different defense strategies in the case of the adaptive threshold selection involve using different percentile values in the prediction error.

We take into account that the prediction error varies over time, as before the system stabilizes nodes have high prediction error and must update their coordinates by large amounts. However, after some iterations the nodes converge to their correct coordinates which results in low prediction errors. To define an independent evaluation of the ideal value, we ran

several experimental runs without the presence of attacks and without outlier detection.

In summary, the effectiveness of an outlier detection technique that uses the adaptive threshold selection depends on the value of c, which defines the importance of the prediction error, and the percentile values on which the prediction error used in updating the threshold is computed.

V. Game Theoretical Framework

We use techniques from game theory [14] to analyze the behavior of the different types of defense mechanisms in virtual coordinate systems by investigating strategic choices made by the players. This allows us to identify the best attack strategies and corresponding defense techniques. The game model considers two players, the defender i and the attacker $-i$, and is a silent game, assuming that the players do not have any knowledge about the other player's history of actions. We consider that an attacker can perform three different types of attacks {deflation; inflation; oscillation} described in Section III-B and vary the percentage of malicious nodes to consist of 10%, 20%, or 30% of the nodes in the network so that the set of actions A for the attacker consists of nine attack strategies in total. In order to counteract the attacker, we define the set of actions D for the defender as follows: the defender has the possibility to use one of the following defense mechanisms: spatial, temporal, and spatial-temporal outlier detection using a fixed threshold. The defender can also use the adaptive threshold variant of outlier detection adapting the closed-loop for the threshold by varying the constant c $\{0.04; 0.06; 0.08; 0.1\}$ and the percentile $\{25^{th}; 50^{th}; 75^{th}\}$ of the system prediction error. We exemplify the adaptive threshold selection strategy only for the spatial outlier detection.

For every chosen action, each of the players receives a payoff, the attacker receives a payoff that measures the effectiveness of the attack, while the defender receives a payoff that measures the effectiveness of the defense. We first adopt a solution concept from game theory and define our payoff functions for both players, the defender and the attacker. We then evaluate the payoffs and identify the best strategies for each player.

A. Solution Concept

Many different game theoretical solution concepts exist, with the Nash Equilibrium [15] being one of the best known. It introduces and defines methods for the mathematical analysis of non-cooperative games in which the players, the defender and the attacker, do not communicate or cooperate. The Nash equilibrium defines the optimal strategic choices for all the players, given that no player will diverge from the equilibrium point as they cannot gain greater payoffs with any other strategy; this behavior is known as rational acting. In a game consisting of a set of n players, there exists for each player i an associated finite set of pure strategies as well as a payoff function P_i.

Calculating the Nash equilibrium results in either a pure strategy, where it is defined that a player, attacker or defender, plays constantly the same action out of the corresponding action set (A or D), or a mixed strategy S_i. A mixed strategy is a probabilistic distribution over the corresponding pure strategies, $S_i : D \rightarrow [0,1]$ and $S_{-i} : A \rightarrow [0,1]$. The player will randomly play each strategy while each strategy is selected with the associated probability. The mixed strategy profile S_i is a Nash equilibrium if for each player i there is no other strategy profile S_i' that will lead to a higher gain with respect to the strategy profile S_{-i} applied by the opponent $-i$, meaning that for player i there is no average payoff Q_i greater than the one for the strategy profiles S_i, S_{-i} : $Q_i(S_i, S_{-i}) \geq Q_i(S_i', S_{-i})$ where

$$Q_i(S_i, S_{-i}) = \sum_{q=1}^{n} S_i(d_q) \sum_{p=1}^{k} S_{-i}(a_p) * M_i(d_q, a_p)$$

In this equation M_i represents the payoff matrix of player i and $M_i(d_q, a_p)$ is the payoff for player i while choosing d_q as action while the opponent plays a_p.

The regular quantal response equilibrium (QRE) [16] is a solution concept that generalizes the Nash equilibrium by introducing an error parameter to the payoff function. This is motivated by the fact that payoff functions may be erroneous and one can not have total certainty about the payoff value. In this manner, the regular QRE provides an equilibrium with bounded rationality in contrast to the Nash equilibrium which defines all the players to be completely rational.

The error parameter λ, also called the rationality parameter, is varied until the regular QRE converges to the Nash equilibrium. Rationality in this sense means that no player has a motivation to diverge from the Nash Equilibrium as there is no other strategy where one can gain more than the ones specified in the resulting strategy profile of the Nash Equilibrium. On the opposite, irrationality means that even though the attacker can not gain greater payoff, he will chose another strategy than the one defined by the Nash Equilibrium. When $\lambda = 0$, the player is completely irrational, in this case he could, for instance, chose the strategy randomly and when $\lambda \rightarrow \infty$, the player becomes perfectly rational and follows the Nash equilibrium. We calculate the QRE by using the following equation which defines the probability of player i to choose strategy d_q out of action set D:

$$S_{id_q} = \frac{\exp \lambda \times U_{i(d_q)}(S_{-i})}{\sum_{a_p} \exp^{\lambda \times U_{i(a_p)}(S_{-i})}}$$

where $U_{i(d_q)}(S_{-i})$ describes the expected utility for player i using strategy d_q while considering other players to play with a probability distribution S_{-i}: $U_{i(d_q)} = \sum_{p=1}^{k} S_{-i}(a_p) * M(d_q, a_p)$

We use the QRE to quantify how irrational a player can be while still maintaining the same equilibrium profiles.

B. Payoff Functions

The payoff functions have to reflect the gain a player receives when playing one strategy. The definition of the payoff depends on the goal a player wants to achieve during

the game. As the attacker and the defender are opponents, their respective payoffs will reflect different goals. The attacker wants to disturb the network by distorting the coordinate space. The larger the impact on the coordinate space, the larger the gain for the attacker. Conversely, the defender wants the coordinate space to behave correctly and provide correct latency estimates. Thus, the smaller the impact of the attacks on the coordinate space, the larger the gain for the defender. A different goal for the attacker is to remain undetected as otherwise the attack is ignored due to the outlier detection. Another goal for the defender is that the coordinate space converges to a stable state and thus wants the error values to be low.

We consider two scenarios in our analysis. The first scenario considers the different outlier detection defenses when using a fixed threshold. The second scenario focuses on spatial outlier detection and analyses the different parameters that can be chosen by the defender in the adaptive threshold mechanism.

Analysis of outlier detection mechanisms with fixed thresholds. We use the system prediction error ($Error_{pred}$) and the relative error ($Error_{rel}$) as the basis for the following payoff functions:

Prediction error based payoffs:

$$P_{attacker} = \frac{Error_{pred}}{\%\text{ attackers}}, \quad P_{defender} = -Error_{pred}$$

Intuitively, $P_{attacker}$ describes the gain of the attacker in direct relation to the prediction error. If the prediction error increases, the payoff increases as well. We include the percentage of malicious nodes in the payoff function to integrate the notion of cost for the attackers since they need to invest time and effort in becoming part of the system to conduct the attacks. $P_{defender}$ describes the gain of the defender in inverse relation to the prediction error. If the prediction error increases, the gain of the defender will decrease. We do not integrate into the defender payoff function the notion of cost in terms of number of defender nodes, since the nodes are already part of the system.

Relative error based payoffs:

$$P_{attacker} = \frac{Error_{rel}}{\%\text{ attackers}}, \quad P_{defender} = \frac{1}{Error_{rel}}$$

For the attacker, $P_{attacker}$ describes the gain in direct relation to the relative error. Just as in the payoff definition for the prediction error, we include the notion of cost for the attackers by dividing the error by the percentage of malicious nodes in the system. For the defender, $P_{defender}$ describes the gain in inverse relation to the relative error.

Furthermore, we take into account different percentiles of error when calculating the payoff functions. We do not only collect median values for the error metrics, but also the 5^{th} and 95^{th} percentile errors. We define the following games:

- **Game 1**: we use the 5^{th} percentile for calculating the attacker's payoff, as the aim of the attacker is to have as large of an impact on the system as possible. At the same time, the defender wants to protect as many nodes as possible, with the goal of minimizing the impact on the system and maintaining a low 95^{th} percentile error.
- **Game 2**: we use the median error nodes for both players as this provides an average overview of the strategic situation of the system.

The payoff functions for Game 1 and Game 2 are summarized in Table I. The usage of the different payoff functions and the corresponding evaluation of the attack and defense strategies are shown in Section VI.

Analysis of spatial outlier detection with adaptive threshold selection. We use the system prediction error ($Error_{pred}$) and the evaluation of the threshold due to the closed-loop feedback control as the basis for the following payoff functions:

Prediction error based payoffs:

$$P_{attacker} = Error_{pred}, \quad P_{defender} = -Error_{pred}$$

$P_{attacker}$ describes the gain of the attacker in direct relation to the median prediction error. If the prediction error increases, the payoff increases as well. $P_{defender}$ describes the gain of the defender in negative relation to the median prediction error. If the prediction error increases, the gain of the defender will decrease.

Threshold evaluation based payoffs:

$$P_{attacker} = T_{avg}, \quad P_{defender} = -Error_{pred}$$

For the attacker, $P_{attacker}$ describes the gain in direct relation to the averaged dynamic threshold. Due to the closed-loop control the threshold changes over all updates in a simulation run, we then take the average value of the evaluation of the threshold value. From the attacker's perspective, his goal is that the threshold remains high, so the probability that his attacks remain undetected is higher than with a smaller threshold. For the defender, $P_{defender}$ describes the gain in negative relation to the median prediction error as in the previous payoff function. We do not take into consideration the threshold for defender's gain as the defender wants the prediction error to be low and that the system converges independent of what value the threshold has. We define the following games:

- **Game 3**: we use the $Error_{pred}$ as attacker payoff, as we assume that the attacker wants to have as much impact on the network as possible. For the defender we use $-Error_{pred}$ as the defender wants to have a good functioning system with a low error. In both cases, we use the median prediction error.
- **Game 4**: we use the T_{avg} as payoff for the attacker, in this case, we assume the attacker to not only want to disturb the network but also to remain undetected of the dynamic threshold, so that the attacks still have impact on the network. The defender uses again the $-Error_{pred}$ as payoff, as for the defender it is not of great importance what value the threshold has, but mainly that the system is working fine, assuming a low median prediction error.

The payoff functions for Game 3 and Game 4 are summarized in Table I. The usage of the different payoff functions

TABLE I
THE DIFFERENT GAMES AND PAYOFF FUNCTIONS OVERVIEW

	Defender payoff	Attacker payoff	Scenario
Game 1	$P_{defender\_95^{th}pred} = -95^{th}Error_{pred}$ $P_{defender\_95^{th}rel} = \frac{1}{95^{th}Error_{rel}}$	$P_{attacker\_5^{th}pred} = \frac{5^{th}Error_{pred}}{\%\text{ attackers}}$ $P_{attacker\_5^{th}rel} = \frac{5^{th}Error_{rel}}{\%\text{ attackers}}$	fixed threshold
Game 2	$P_{defender\_50^{th}pred} = -50^{th}Error_{pred}$ $P_{defender\_50^{th}rel} = \frac{1}{50^{th}Error_{rel}}$	$P_{attacker\_50^{th}pred} = \frac{50^{th}Error_{pred}}{\%\text{ attackers}}$ $P_{attacker\_50^{th}rel} = \frac{50^{th}Error_{rel}}{\%\text{ attackers}}$	
Game 3	$P_{defender} = -Error_{pred}$	$P_{attacker} = Error_{pred}$	adaptive threshold
Game 4	$P_{defender} = -Error_{pred}$	$P_{attacker} = T_{avg}$	

and the corresponding evaluation of the attack and defense strategies are shown in Section VI.

VI. EXPERIMENTAL RESULTS

In this section, we demonstrate through simulations using actual Internet topologies and quantitative analysis using game theory techniques the efficacy of different attacks at impacting the accuracy of the Vivaldi virtual coordinate system and of our defense mechanisms at preserving its ability to maintain accurate latency estimates.

A. Evaluation Methodology

In order to simulate the attack and defense strategies, we use the King [17] and AMP [18] data sets in conjunction with the p2psim simulator [19]. The King data set contains the pairwise RTT of 1740 nodes with an average RTT of 180ms and was selected since it is representative of larger scale peer-to-peer systems and has been used in validating several virtual coordinate systems. The AMP data set consists of the pairwise RTT of 90 nodes with an average RTT of 70ms and it is used to represent smaller, high speed systems (*e.g.*, a corporate network). Synthetic topologies are not considered as they do not capture important network properties inherent in real networks such as violations of the triangle inequality.

We ran simulations for each combination of attack type and defense strategy described in Section III. We ran each simulation for 200 time units, where each time unit is 500 seconds in length. Every simulation was run ten times with the reported metrics averaged over all of the runs. The nodes join in a flash-crowd scenario in which all nodes join simultaneously and are initially placed at the origin of the logical coordinate space. All nodes that join the network are physically stationary and are present for the duration of the experiment. Each node proceeds independently of other nodes and chooses a reference set of 64 nodes using the Vivaldi method where half of the nodes are selected as the closest nodes based on network latency and the rest are selected at random. All other Vivaldi parameters were initialized to the optimal values discussed by Dabek *et al*. [8].

B. King Data Set Analysis

Analysis for the different outlier detection defense mechanisms with fixed threshold. We first analyze the effect of using different spatial outlier thresholds on the Vivaldi virtual coordinate system running over the King topology. In Table IV, we can see the Nash equilibrium using *Game 1* as defined in Table I in Section V-B. From the results, we see that the inflation attack has a large impact on the system. Under this attack, we find that the most efficient defense strategy is spatial-temporal outlier detection when using lower spatial outlier thresholds (*e.g.*, ≤ 1.5). For higher thresholds, both spatial-temporal outlier detection and spatial outlier detection provide similar defense performance. We note that temporal outlier detection is ineffective as it never appears as part of one of the equilibria.

In Table V, we present the Nash equilibrium using *Game 2*. Depending on the threshold selected, either the spatial outlier detection or the spatial-temporal outlier detection defense techniques provide the best performance and are thus employed in the resulting Nash equilibrium. Based on the evaluations of both *Game 1* and *Game 2*, we conclude that the inflation attack has the greatest potential to impact the virtual coordinate system. It is interesting to note that for lower outlier thresholds (≤ 1.5), the attack is most effective for smaller percentages of malicious nodes as the effort to create larger attacks leads to diminishing returns. Only the higher threshold of 1.75 allows the inflation attack with 30% malicious nodes to be effective and appear as an equilibrium, allowing us to conclude this threshold is less effective at mitigating the effects of the attacker. Finally, similar to *Game 1*, we notice that temporal outlier detection does not appear in an equilibrium and we thus conclude that this type of outlier detection is not an effective countermeasure when used by itself.

We also analyze the best defenses against the different attacks when using a spatial outlier threshold of 1.5, as this value was suggested by previous research [6]. For the deflation attack, the optimal defense strategy is to use spatial outlier detection as it results in a pure equilibrium for both the prediction error and the relative error. The spatial-temporal outlier detection is the best defensive mechanism against the oscillation attack regarding both error metrics. Evaluations based on *Game 1* show that spatial outlier detection performs similarly. For the inflation attack, we have a different defense strategy resulting in a pure equilibrium for each of the games. For *Game 1*, spatial-temporal outlier detection represents the pure equilibrium, while in *Game 2*, spatial outlier detection represents the equilibrium. Furthermore, we assess the threshold selection for this data set, and find independent of the game or the error metric, that a threshold value of 1.25 always results in a pure equilibrium, making this the best threshold.

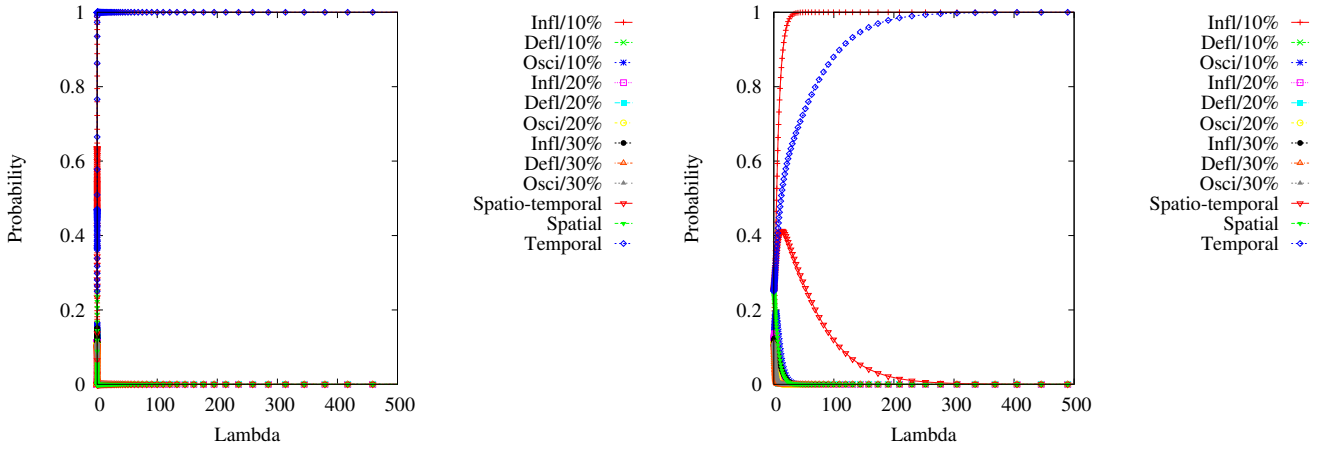
The previous results, which are based on the Nash equilibrium, assume that the players are completely rational. As this cannot be guaranteed, we use a secondary evaluation to determine how irrational the players can act while still

TABLE II
OVERVIEW OF THE GAME FOR THE DIFFERENT OUTLIER DETECTION DEFENSES WITH FIXED THRESHOLD

Player	Strategy		Payoff function	
			border value	median value
Defender	Spatial		$P_{def\_95^{th}{}_{pred}} = -95^{th}Error_{pred}$ $P_{def\_95^{th}{}_{rel}} = \frac{1}{95^{th}Error_{rel}}$	$P_{def\_50^{th}{}_{pred}} = -50^{th}Error_{pred}$ $P_{def\_50^{th}{}_{rel}} = \frac{1}{50^{th}Error_{rel}}$
	Temporal			
	Spatial-Temporal			
Attacker	10%	Inflation	$P_{att\_5^{th}{}_{pred}} = \frac{5^{th}Error_{pred}}{\%\text{ mal. nodes}}$ $P_{att\_5^{th}{}_{rel}} = \frac{5^{th}Error_{rel}}{\%\text{ mal. nodes}}$	$P_{att\_50^{th}{}_{pred}} = \frac{50^{th}Error_{pred}}{\%\text{ mal. nodes}}$ $P_{att\_50^{th}{}_{rel}} = \frac{50^{th}Error_{rel}}{\%\text{ mal. nodes}}$
		Deflation		
		Oscillation		
	20%	Inflation		
		Deflation		
		Oscillation		
	30%	Inflation		
		Deflation		
		Oscillation		

TABLE III
OVERVIEW OF THE GAME ANALYSIS FOR SPATIAL OUTLIER DETECTION WITH ADAPTIVE THRESHOLD

Player	Strategy		Payoff function	
			Error evaluation	Threshold evaluation
Defender	$c = 0$		$P_{def} = -Error_{pred}$	
	25^{th} Percentile	$c = (0.04, 0.06, 0.08, 0.1)$		
	50^{th} Percentile	$c = (0.04, 0.06, 0.08, 0.1)$		
	75^{th} Percentile	$c = (0.04, 0.06, 0.08, 0.1)$		
Attacker	10%	(Inflation, Deflation, Oscillation)	$P_{att} = Error_{pred}$	$P_{att} = T_{avg}$
	20%	(Inflation, Deflation, Oscillation)		
	30%	(Inflation, Deflation, Oscillation)		



(a) Prediction error (b) Relative error

Fig. 2. The Quantal Response Equilibrium evaluation for the King data set based on *Game 1*

maintaining the same optimal equilibrium profiles. In Figure 2, we present the regular QRE (as described in Section V-A) for the data set. The y-axis represents the probability for a strategy for a given λ. We notice that when considering the prediction error (Figure 2(a)), the QRE converges to the Nash equilibrium for $\lambda \to 0$, implying that even if the attacker is irrational, he will follow the Nash equilibrium with respect to the prediction error. Regarding the relative error, the QRE converges to the Nash equilibrium for $\lambda \approx 300$(Figure 2(b)) which means that the strategies in relation to the relative errors also converge fast to the Nash Equilibrium as $0 < \lambda < \infty$. Using this metric as the basis of the payoff function, an irrational attacker will diverge from the Nash Equilibrium, but as it becomes more rational, it quickly follows the optimal identified strategy.

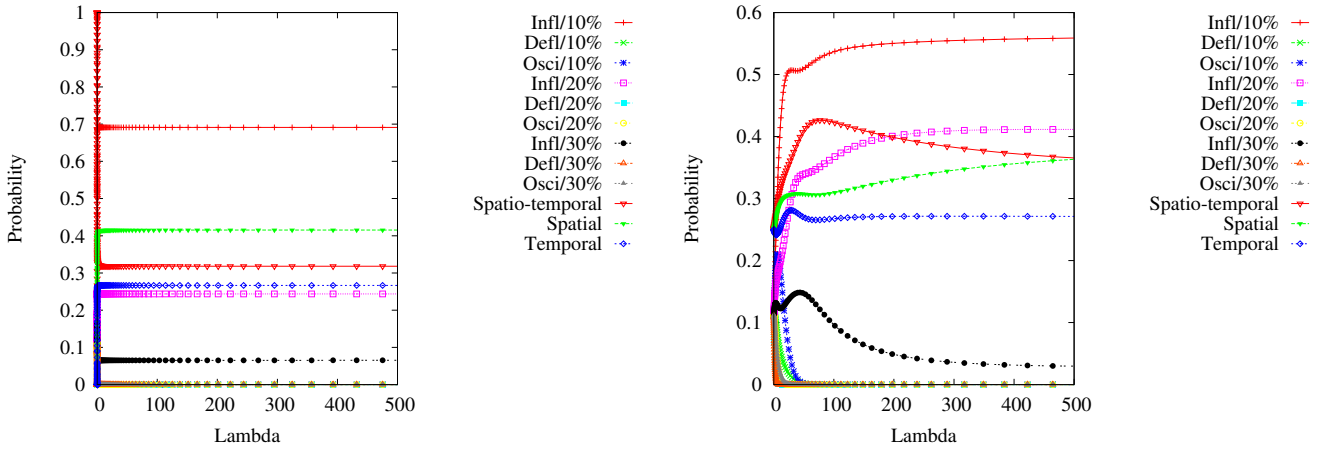
Analysis for the spatial-temporal outlier detection with adaptive spatial threshold. In the previous analysis, we observed that the best defense mechanism is to apply spatial-

TABLE IV
KING - EQUILIBRIUM POINTS BASED ON *Game 1*

Threshold	Error metric	Nash equilibrium strategy profile		
		profile	attacker	defender
1.25	pred. error	pure	Infl/10% att.	Spatial-temporal
	rel. error	pure	Infl/10% att.	Spatial-temporal
1.5	pred. error	pure	Infl/10% att.	Spatial-temporal
	rel. error	pure	Infl/10% att.	Spatial-temporal
1.75	pred. error	pure	Infl/30% att.	Spatial
	rel. error	2 pure profiles	Infl/30% att. Infl/30% att.	Spatial-temporal Spatial

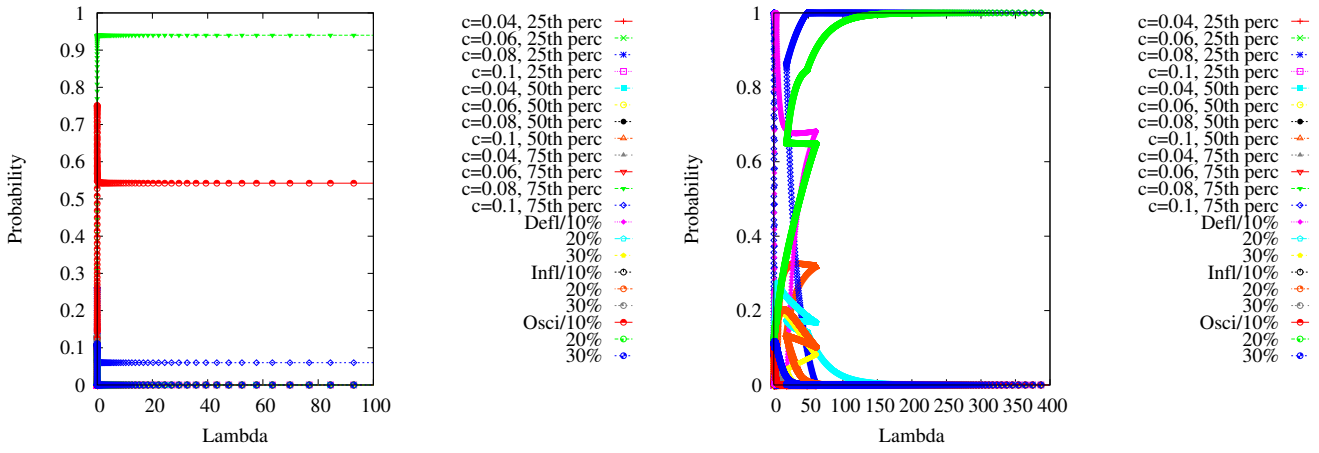
TABLE V
KING - EQUILIBRIUM POINTS BASED ON *Game 2*

Threshold	Error metric	Resulting Nash equilibrium strategy profile		
		profile	attacker	defender
1.25	pred. error	pure	Infl/10% att.	Spatial
	rel. error	pure	Infl/10% att.	Spatial
1.5	pre. error	pure	Infl/10% att.	Spatial
	rel. error	pure	Infl/10% att.	Spatial
1.75	pred. error	pure	Infl/30% att.	Spatial-temporal
	rel. error	pure	Infl/30% att. Infl/30% att.	Spatial-temporal Spatial



(a) Prediction error (b) Relative error

Fig. 3. The Quantal Response Equilibrium evaluation for the AMP data set based on *Game 1*



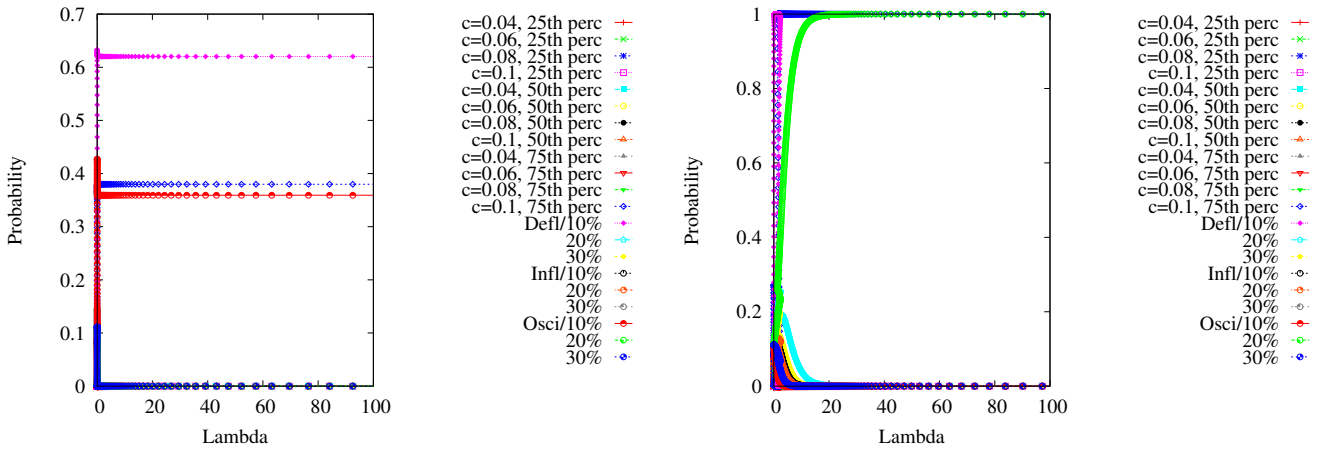
(a) *Game 3* (b) *Game 4*

Fig. 4. The Quantal Response Equilibrium evaluation for the King data set

temporal outlier detection. We now again consider spatial-temporal outlier detection, however we consider that the spatial outlier detection uses an adaptive threshold. We initialize the spatial threshold with 2. In Table VI, we can see the Nash equilibrium using *Game 3* and *Game 4* as defined in Table I in Section V-B. The different strategies are shown in Table III. We note that we also compare the strategy $c = 0$, which means that control theory is not used at all. Considering *Game 3*, where we assume the attacker wants to disturb as much as possible the correct functioning of the system with the effect that the prediction error increases significantly, the resulting best attack method is to apply inflation attack with 30% attackers with a probability of 0.54 and the oscillation attack with 30% attackers with a probability of 0.46. For the defender the best way to handle this attack method is to make use of the constant $c = 0.06$ with a probability of 0.93, $c = 0.08$ with 0.07 probability, and the 75^{th} percentile of the prediction error for updating the closed-loop feedback control described in Section IV-B.

In *Game 4*, we assume the attacker does not only intend to harm the network as much as possible but that he wants to remain undetected. With the resulting Nash Equilibrium we can see that for the attacker the best choice overall is to have only 10% attackers in the network, as otherwise the attacks become too obvious and are detected by the outlier detection. The overall defense mechanism is to use the 75^{th} percentile with $c = 0.06$ and $c = 0.08$ or to use the 50^{th} percentile with $c = 0.08$. It can be seen that there are 3 different Nash Equilibria for this game model, this means that all of these points lead to the best possible gain for the defender with respect to the attack method applied.

We again use the secondary evaluation to determine how irrational the players can act while still maintaining the same optimal equilibrium profiles, as rationality can not be guaranteed. In Figure 4, we present the regular QRE. The y-axis represents the probability for a strategy for a given λ. We notice that in *Game 3* (Figure 4(a)), the QRE converges to the Nash equilibrium for $\lambda \to 0$, implying that even if the attacker is irrational, he will follow the Nash equilibrium with respect to the prediction error. Regarding *Game 4*, the QRE converges to the Nash equilibrium for $\lambda \approx 100$ (Figure 4(b)), which is a fast convergence although λ can become ∞. Using this metric as the basis of the payoff function, an irrational attacker will diverge from the Nash Equilibrium, but as it becomes more



(a) *Game 3* (b) *Game 4*

Fig. 5. The Quantal Response Equilibrium evaluation for the AMP data set

TABLE VI
KING - EQUILIBRIUM POINTS

	Payoffs	Nash equilibrium strategy profile				
		profile	attacker		defender	
			strategy	probability	strategy	probability
Game 3	$P_{def} = -Error_{pred}$ $P_{att} = Error_{pred}$	mixed	Inflation/30% att. Oscillation/30%	0.54 0.46	c=0.06 & $75^{th} percentile$ c=0.08 & $75^{th} percentile$	0.93 0.07
Game 4	$P_{def} = -Error_{pred}$ $P_{att} = T_{avg}$	pure	Oscillation/10% att.	1	$c = 0.08$ & $75^{th} percentile$	1
		pure	Deflation/10% att.	1	$c = 0.06$ & $75^{th} percentile$	1
		mixed	Deflation/10% att. Oscillation/10% att.	0.5 0.5	$c = 0.08$ & $50^{th} percentile$	1
		pure	Oscillation /10% att.	1	$c = 0.08$ & $50^{th} percentile$	1

rational, it quickly follows the optimal identified strategy.

C. AMP Data Set Analysis

Analysis for the outlier detection defense mechanism with fixed threshold. We evaluate the AMP data set looking at both error metrics for different spatial outlier threshold selections. Table VII describes the resulting strategy profiles from following *Game 1*. We notice that for this data set, the resulting strategy profiles are not nearly as homogeneous as those for the King data set. Most of the resulting strategy profiles consist of a mixed strategy, meaning that the different strategies should be utilized with the given probability in order to be as effective as possible. For example, given the spatial outlier threshold of 1.25, the attacker has the most impact while applying the deflation attack with only 10% malicious nodes in the system with a probability of 0.55 and applying the oscillation attack with 10% malicious nodes in the system with a probability of 0.45. The countermeasures look similar, applying spatial-temporal outlier detection and temporal outlier detection with probabilities of 0.93 and 0.07 respectively. Overall, we can see that the spatial-temporal outlier detection has highest probability of being applied. Interestingly, unlike the King data set, the temporal outlier detection is often part of the equilibrium, but only with low probability. The outcomes for *Game 2* are reflected in Table VIII. For this evaluation, only the spatial-temporal outlier detection and the spatial outlier detection are considered in the equilibriums.

Next, we investigate the optimal countermeasure with respect to the different attacks. The spatial outlier detection performs best against the three attacks. Assessing the different thresholds, results show that a threshold value of 1.25 is the best choice, a threshold value of 1.5 is second best, 1.75 third, while 2 is last. Furthermore, we also evaluated the regular QRE as we did for the King data set. Similar to the previous data set, we note that with respect to the prediction error (Figure 3(a)), players could be almost completely irrational while their best strategies will still follow the Nash equilibrium, as it converges for $\lambda \to 0$. The relative error (Figure 3(b)) converges fast to the Nash equilibrium for $\lambda \approx 500$.

Analysis for spatial-temporal outlier detection with adaptive spatial threshold. We evaluate the AMP data set with spatial-temporal outlier detection and an adaptive threshold selection for the spatial threshold. We initialize the spatial threshold with 2. Table IX describes the resulting strategy profiles for the different payoffs and configuration strategies for *Game 3* and *Game 4* as defined in Table I in Section V-B. We again note that the different strategies are shown in Table III, including the strategy $c = 0$, where control theory is not used. We notice that the resulting Nash Equilibria are similar to the Nash Equilibria for the KING data set. Based on this, we can assume that independent of the data set we should apply in the closed-loop feedback control a percentile of 75^{th} percentile for the prediction error. An attacker can disturb the network the most while applying the inflation attack, but if he wants the attacks to be undetected then deflation and oscillation are the best attack choices. Furthermore, we again evaluate the regular QRE and notice that with respect to *Game 3* (Figure 5(a)) the QRE converges to the Nash equilibrium for $\lambda \to 0$, implying that even if the attacker is irrational he follows the strategy

TABLE VII
AMP - EQUILIBRIUM POINTS BASED ON *Game 1*

Threshold	Error metric	Nash equilibrium profile				
		profile	attacker		defender	
			strategy	probability	strategy	probability
1.25	pred.	mixed	Defl/10% att.	0.55	Spatial-temporal	0.93
			Osci/10% att.	0.45	Temporal	0.07
	rel.	mixed	Defl/10% att.	0.54	Spatial-temporal	0.92
			Osci/10% att.	0.46	Temporal	0.08
1.5	pred.	pure	Infl/10% att.		Spatial	
	rel.	pure	Infl/10% att.		Spatial	
1.75	pre.	mixed	Defl/10% att.	0.74	Spatial	0.08
			Infl/10% att.	0.18	Spatial-temporal	0.91
			Osci/10% att.	0.08	Temporal	0.009
	rel.	mixed	Infl/10% att.	0.29	Spatial-temporal	0.92
			Osci/10% att.	0.71	Spatial	0.08
		pure	Infl/20%att.		Spatial-temporal	
2	pred.	mixed	Infl/10% att.	0.69	Spatial-temporal	0.32
			Infl/20% att.	0.24	Spatial	0.41
			Infl/30% att.	0.07	Temporal	0.27
	rel.	mixed	Infl/10% att.	0.57	Spatial-temporal	0.33
			Infl/20% att.	0.4	Spatial	0.40
			Infl/30% att.	0.03	Temporal	0.27

TABLE VIII
AMP - EQUILIBRIUM POINTS BASED ON *Game 2*

Threshold	Error metric	Nash equilibrium profile				
		profile	attacker		defender	
			strategy	probability	strategy	probability
1.25	pred.	mixed	Defl/10% att.	0.53	Spatial-temporal	0.64
			Osci/10% att.	0.47	Spatial	0.36
	rel.	mixed	Defl/10% att.	0.5	Spatial-temporal	0.67
			Osci/10% att.	0.5	Spatial	0.33
1.5	pred.	pure	Infl/10% att.		Spatial	
	rel.	pure	Infl/10% att.		Spatial	
1.75	pred.	pure	Infl/10% att.		Spatial-temporal	
	rel.	pure	Infl/10% att.		Spatial-temporal	
2	pred.	pure	Infl/10% att.		Spatial	
	rel.	pure	Infl/10% att.		Spatial	

TABLE IX
AMP - EQUILIBRIUM POINTS BASED ON THE ANALYSIS FOR THE ADAPTIVE THRESHOLD SELECTION

	Payoffs	Nash equilibrium strategy profile				
		profile	attacker		defender	
			strategy	probability	strategy	probability
Game 3	$P_{def} = -Error_{pred}$ $P_{att} = Error_{pred}$	mixed	Oscillation/30% att.	1	$c = 0.08$ & $75^{th} percentile$	0.25
					$c = 0.1$ & $75^{th} percentile$	0.75
		pure	Oscillation/30% att.	1	$c = 0.08$ & $75^{th} percentile$	1
		pure	Inflation/30% att.	1	$c = 0.1$ & $75^{th} percentile$	1
Game 4	$P_{def} = -Error_{pred}$ $P_{att} = T_{avg}$	pure	Oscillation/10% att.	1	$c = 0.06$ & $75^{th} percentile$	1
		pure	Deflation/10% att.	1	$c = 0.08$ & $75^{th} percentile$	1
		pure	Oscillation/10% att.	1	$c = 0.08$ & $75^{th} percentile$	1

profile defined by the Nash Equilibrium. In *Game 4*, the QRE converges to the Nash equilibrium for $\lambda \approx 20$(Figure 5(b)). This converges fast as well, as $0 < \lambda < \infty$.

VII. RELATED WORK

Defense Mechanisms in Virtual Coordinate Systems. Research has previously demonstrated the susceptibility of Vivaldi to attacks [20, 21]. To address these vulnerabilities, there have been several proposed methods to maintain virtual coordinate system accuracy [1]–[5]. The PIC virtual coordinate system [1] uses a security test based on the triangle inequality in which any node that violates the triangle inequality above some margin of error is ignored and designated as malicious. However, it has been shown that RTT measurements often violate this inequality [22]–[24] and thus solutions based solely on such inequalities may degrade system performance when no attack is occurring.

Recent work by Kaafar *et al*. [2] utilizes a solution which employs a set of trusted nodes as a reference set by which to analyze all node behavior for malicious patterns and behavior. In a similar vein, the reputation-based work by Saucez *et al*. [3] uses *a priori* trusted nodes to detect malicious nodes. The key difference between these techniques and methods used in this work is that they do not necessitate the need for trusted components in the network. Such trusted components could lead to high deployment costs, for example [2] requires 10% of nodes to be trusted, thus potentially needing hundreds or thousands of trusted nodes. The work by Sherr *et al*. [4, 5] uses a verification set of nodes for a node n, where n's update is considered malicious if a percentage of the verification set perceives the error of the update to be greater than a predetermined threshold. The main differences between this technique and the methods we present is that we do not require extra node sets nor network communication which may

lead to high network overhead, and we utilize outlier detection over multiple metrics.

Game Theoretic Security Approaches. Game theory has been a mainstream research topic in the economic community since the landmark Ph.D. thesis of J. Nash [15] and the interested reader is referred to the work by Binmore [25] for a comprehensive introduction to the area. One of the first approaches for applying game theory to network security is described by McInerney *et al*. [26]. In this work, an underlying Markovian decision process and a simple one-player game are used to reason, detect, and respond to automated attack behavior in information assurance systems. Similar work on network security by Lye and Wing [27] models the interaction between an attacker and a defender as a two-player stochastic game. The explicit enumeration of states as described in the previous papers two is impossible in our context due to the large number of attacking nodes that we consider.

Applied economics concepts have also been applied to computer security to address the analysis of strategic choices that enterprises will take regarding maintenance and management under an assumed cost model [28]. For example, some of our previous work has used game theory to understand and better defend against blocking and flooding attacks against distributed hash tables used in P2P Session Initiation Protocol infrastructures [29]. However, the previous model cannot capture the varying degrees of irrationality inherent in the current environment and we incorporate the regular quantal response equilibrium [16] to accurately model the malicious behavior.

VIII. Conclusion and Future Work

In this paper, we have defined and used an analytical framework in order to analyze strategic choices and identify the best attack strategies and corresponding defense strategies used in virtual coordinate systems. We have performed experiments using two Internet topology data sets that have correspondingly different sizes and characteristics. Our results demonstrate that spatial-temporal and spatial outlier detection perform the best while temporal outlier detection is ineffective in isolation. However, temporal outlier detection is often part of the defense profile in combination with the other two techniques. From an attackers perspective, the best attack strategy to use is the inflation attack with varying percentages of malicious nodes, depending on the deployed defense technique. We have also assessed several threshold settings for the outlier detection and the spatial outlier threshold of 1.25 provides the best results for a fix threshold. We also introduced an adaptive threshold selection that was more effective than a fixed threshold, and found the best parameters to use are the 75^{th} percentile prediction error and a constant c of 0.08. Future work consists of extending these methods towards more complex games in which learning and signaling occur during system operation. For instance, benign nodes might learn defense strategies online while attackers perform sequences of simple strategies.

Acknowledgements

This work was partially supported by the National Research Fund, Luxembourg and by NSF CyberTrust 0715833-CNS. Any opinions, findings, and conclusions expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

[1] M. Costa, M. Castro, R. Rowstron, and P. Key, "Pic: practical internet coordinates for distance estimation," in *Proc. of ICDCS*, 2004.
[2] M. A. Kaafar, L. Mathy, C. B. K. Salamatian, T. Turletti, and W. Dabbous, "Securing internet coordinate embedding systems," in *Proc. of SIGCOMM*, 2007.
[3] D. Saucez, B. Donnet, and O. Bonaventure, "A reputation-based approach for securing vivaldi embedding system," *Lecture Notes in Computer Science*, vol. 4606, p. 78, 2007.
[4] M. Sherr, B. Loo, and M. Blaze, "Veracity: A fully decentralized service for securing network coordinate systems," in *Proc. of IPTPS*, 2008.
[5] M. Sherr, M. Blaze, and B. T. Loo, "Veracity: Practical secure network coordinates via vote-based agreements," in *Proc. of USENIX ATC*, 2009.
[6] D. Zage and C. Nita-Rotaru, "On the accuracy of decentralized network coordinate systems in adversarial networks," in *Proc. of CCS*, 2007.
[7] M. Bishop, *Computer Security: Art and Science*. Addison-Wesley, 2003.
[8] F. Dabek, R. Cox, F. Kaashoek, and R. Morris, "Vivaldi: a decentralized network coordinate system," in *Proc. of ACM SIGCOMM*, 2004.
[9] C. Lumezanu and N. Spring, "Playing vivaldi in hyperbolic space," in *Proc. of ACM SIGCOMM-IMC*, 2006.
[10] K. Wang and S. J. Stolfo, "Anomalous payload-based network intrusion detection," in *Proc. of RAID*, 2004.
[11] G. Jiang and G. Cybenko, "Temporal and spatial distributed event correlation for network security," in *Proc. of ACC*, 2004.
[12] C. A. Desoer, R. W. Liu, J. Murray, and R. Saeks, "Feedback system design: The fractional representation approach to analysis and synthesis," in *Decision and Control including the Symposium on Adaptive Processes, 1979 18th IEEE Conference on*, vol. 18, 1979, pp. 33 –37.
[13] J. C. Doyle, B. A. Francis, and A. R. Tannenbaum, *Feedback Control Theory*. Prentice Hall Professional Technical Reference, 1991.
[14] J. V. Neumann and O. Morgenstern, *Theory of Games and Economic Behavior*. Princeton University Press, 1944.
[15] J. Nash, "Non-cooperative games," *The Annals of Mathematics*, vol. 54, no. 2, pp. 286–295, 1951.
[16] J. Goeree, C. Holt, and T. Palfrey, "Regular quantal response equilibrium," *Experimental Economics*, vol. 8, no. 4, pp. 347–367, 2005.
[17] K. P. Gummadi, S. Saroiu, and S. D. Gribble, "King: Estimating latency between arbitrary internet end hosts," in *Proc. of ACM SIGCOMM-IMW*, 2002.
[18] "Nlanr active measurement project," http://amp.nlanr.net/.
[19] "p2psim: A simulator for peer-to-peer protocols," http://pdos.csail.mit.edu/p2psim/.
[20] M. A. Kaafar, L. Mathy, T. Turletti, and W. Dabbous, "Real attacks on virtual networks: Vivaldi out of tune," in *Proc. of LSAD*, 2006.
[21] ——, "Virtual networks under attack: Disrupting internet coordinate systems," in *Proc. of CoNext*, 2006.
[22] J. Ledlie, P. Gardner, and M. Seltzer, "Network coordinates in the wild," in *Proc. of USENIX NSDI*, 2007.
[23] E. Lua, T. Griffin, M. Pias, H. Zheng, and J. Crowcroft, "On the accuracy of embeddings for internet coordinate systems," in *Proc. of IMC*, 2005.
[24] H. Zheng, E. Lua, M. Pias, and T. Griffin, "Internet routing policies and round-trip-times," in *Proc. of PAM*, 2005.
[25] K. Binmore, *Playing for Real: A Text on Game Theory*. Oxford University Press, 2007.
[26] J. McInerney, S. Tubberud, S. Anwar, and S. Hamilton, "Friars: a feedback control system for information assurance using a Markov decision process," in *Proc. of ICCST*, 2001.
[27] K.-W. Lye and J. Wing, "Game strategies in network security," *International Journal of Information Security*, vol. 4, no. 1, pp. 71–86, 2005.
[28] J. Grossklags, N. Christin, and J. Chuang, "Secure or insure?: a game-theoretic analysis of information security games," in *Proc. of WWW*, 2008.
[29] S. Becker, R. State, and T. Engel, "Using game theory to configure p2p sip," in *Proc. of IPTComm*, 2009.