

# Robust Virtual Coordinate Systems with Byzantine Participants

David Zage

Department of Computer Science and CERIAS, Purdue University  
305 North University Street, West Lafayette, IN 47907 USA  
zagedj@cs.purdue.edu

**Abstract**—Virtual coordinate systems provide an accurate and efficient service that allows hosts on the Internet to determine the latency between arbitrary hosts without using active monitoring of all nodes in the network. Many of the proposed virtual coordinate systems were designed with the assumption that all of the nodes in the system are cooperative. However, this assumption may be violated by compromised nodes acting maliciously to degrade the accuracy of the virtual coordinate systems.

In this work, we demonstrate the vulnerability of decentralized virtual coordinate systems to insider (or Byzantine) attacks. We propose techniques to decrease the number of incorrect coordinate changes, thereby making coordinate assignment and maintenance robust to malicious attackers. We demonstrate the attacks and mitigation techniques in the context of a well-known distributed virtual coordinate system using simulations based on a representative, real-world data set of Internet latencies.

## I. INTRODUCTION

A wide range of applications taking advantage of peer-to-peer systems have emerged in recent years, including file download and distribution (e.g. BitTorrent, Emule), voice over IP (e.g. Skype), and video broadcasting (e.g. ESM, Coolstreaming). Many of these applications optimize their performance based on network topology. For example, the performance of overlay networks, the construction of multicast trees, or the selection of a replica for a file sharing application can be greatly improved by taking advantage of network locality. One basic approach to learn network locality is to actively probe all hosts in the network to determine attributes such as latency. The cost associated with active monitoring to estimate distance is non-negligible [1], being further exacerbated by the presence of multiple applications on a common network infrastructure.

In order to avoid the need for real-time measurement, virtual coordinate systems have been developed to predict latencies between arbitrary hosts in a network [2], [3], [4], [5], [6], [7], [8]. These systems allow a node to map itself to a *virtual* coordinate based on a small number of actual distance estimates from a subset of nodes. By comparing virtual coordinates, nodes can trivially estimate the latency between them. The savings in metric estimation time and traffic load indicate that virtual coordinate systems are a viable networking building block.

Two main architectures for virtual coordinate systems have emerged: landmark-based and decentralized. Landmark-based systems rely on centralized components (such as a set of landmark servers) to predict distance between any two hosts.

The set of landmarks can be pre-determined [2] or randomly selected [4]. Decentralized virtual coordinate systems, such as PIC [6], Vivaldi [5], and PCoord [8] do not rely on explicitly designated infrastructure nodes. These systems share a similar querying mechanism used by each node to assign and maintain its own coordinates. Specifically, each node maintains a reference set of randomly selected network nodes from which to periodically query one node at random for an update. Each query response is used to recalculate the requester's coordinates. Accurate coordinate estimation is achieved since coordinate updates are optimized based on system-dependent numerical error minimization techniques, such as the downhill simplex method [6].

Virtual coordinate systems have been designed with the assumption that all participating nodes correctly report their metrics, resulting in an accurate system often with an overall system error of less than ten percent [5], [8]. However, it has been shown that decentralized virtual coordinate systems are vulnerable to attacks which can have a significant impact on their accuracy [9]. As more applications rely on virtual coordinate systems, it is critical that such systems are designed to be robust to malicious attackers attempting to influence the accuracy of the distance prediction.

In this work, we focus on studying the vulnerability of coordinate assignment and maintenance to insider attacks in decentralized virtual coordinate systems. We propose techniques to enhance the accuracy of such systems with resilience to attacks without increasing the overall communication cost incurred by the system. Our solutions decrease the number of incorrect coordinate changes by introducing constraints on reported data and using outlier detection.

We provide an overview of the main attack mechanism in Section II, describe approaches to mitigate the attacks in Section III, and present initial results in Section IV. We conclude with ongoing and future work in Section V.

## II. ATTACKS AGAINST VIRTUAL COORDINATE SYSTEMS

**Attacker Model** We consider a constrained-collusion Byzantine adversary model similar to that proposed in [10], with a system size of  $N$  and a bounded percentage of malicious nodes  $f$  ( $0 \leq f < 1$ ) behaving arbitrarily. We assume a malicious adversary has access to all data at a node as any legitimate user would (insider access), including cryptographic keys stored at a node. Nodes cannot be completely trusted

although they are authenticated. We assume that data authentication and integrity mechanisms are deployed and we focus only on attacks directed at coordinate management.

**Attacks Description** In virtual coordinate systems, the coordinates of each node undergo a maintenance process in which they are updated using information gathered from querying reference set nodes. Malicious nodes can take advantage of this process to gain control over coordinate selection by lying about their observed metrics. For example, a malicious node (or set of nodes) can attack the system by choosing a remote virtual location and advertising this position with low error, thus drawing other good nodes towards the remote location. An attacker may seek to repel nodes away from specific virtual coordinates or general coordinate areas. Any attack against the coordinate system may target a particular node, subset of nodes, or region of the coordinate space. The final result of manipulating the coordinate system can include isolating subsets of nodes from the networks, rendering the coordinate system unusable due to high estimation error, and creating general disorder in the system.

### III. MITIGATING ATTACKS AGAINST VIRTUAL COORDINATE SYSTEMS

Virtual coordinate systems were designed to operate in benign environments assuming that all nodes are altruistic. The only notable exception is PIC [6], which relies on the triangular inequality to detect malicious nodes reporting incorrect latencies. The results based on synthetic networks presented in [6] show that the method does improve the accuracy of PIC in adversarial networks. However, as it was shown in [11], [12], violations of the triangle equality are very frequent for real networks resulting in inaccuracy and fragility of virtual coordinate systems even when they are deployed in non-adversarial networks.

The primary cause of the attacks is the ability of the attacker to influence the coordinate maintenance process by manipulating the metrics obtained from querying reference set nodes. By blindly accepting this potentially malicious information, a correct node may compute incorrect coordinates. We propose to prevent incorrect changes by detecting inconsistent metrics using temporal and spatial correlations among metrics in the system. By analyzing these correlations on the information received from queried reference set nodes, we identify and filter out outliers in the metrics.

In our solution, each node independently performs spatial and temporal outlier detection before changing its coordinates in order to identify potentially malicious metrics. *Spatial outlier detection* compares the recently received metrics from each of the queried nodes in a node’s reference set and forces a node to report metrics consistent with what other reference peers are currently reporting. *Temporal outlier detection* examines the consistency of the metrics received from an individual queried node over time and forces a node to report metrics consistent with what it has reported in the past. We propose to use the 3-tuple of  $\langle \text{remote error}, \text{latency}, \text{change in remote coordinates} \rangle$  to generate the spatial outlier statistics and the

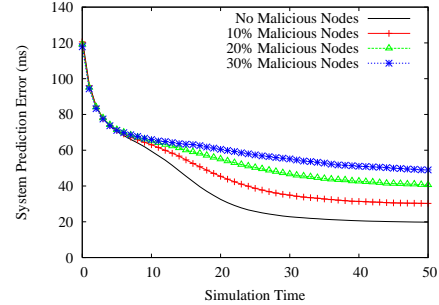


Fig. 1. Vivaldi System Prediction Error for Different Percentages of Attackers

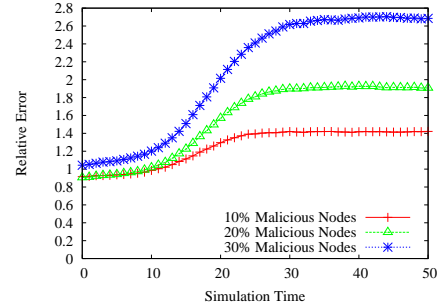


Fig. 2. Vivaldi Relative Error for Different Percentages of Attackers

5-tuple of  $\langle \text{remote error}, \text{local error}, \text{latency}, \text{change in remote coordinates}, \text{change in local coordinates} \rangle$  to generate the temporal outlier statistics. The metrics were chosen on the basis that while each of them represents a different measure of system performance, changes in one measure will result in a correlated change in other metrics. Also, our approach uses the metrics already reported in the system, thus not affecting the link stress.

Our approach uses the Mahalanobis [13] distance to detect outliers. We selected this distance function because it has been shown effective at detecting outliers with multiple attributes, scales each variable based on its standard deviation and covariance, and takes into account how the measured attributes change in relation to each other [14].

### IV. INITIAL RESULTS

We demonstrate attacks against virtual coordinate systems using the Vivaldi system with a reference set of 64 nodes and the p2psim simulator [15]. We selected Vivaldi to demonstrate the attacks because it is a mature system, uses minimization techniques to guard against *benign* error, is conceptually easy to understand and visualize, and has been shown to produce low error embeddings [5]. We use the King data set [7], which contains the pair-wise round-trip-time (RTT) of 1740 real Internet nodes with an average RTT of 180ms.

In order to quantitatively compare the effect of attacks on the accuracy of the system, we evaluate two error metrics:

*System prediction error* is defined as

$$Error_{pred} = |Act_{RTT} - Est_{RTT}| \quad (1)$$

where the  $Act_{RTT}$  is the actual measured RTT and  $Est_{RTT}$  is the predicted RTT by the virtual coordinate system. This

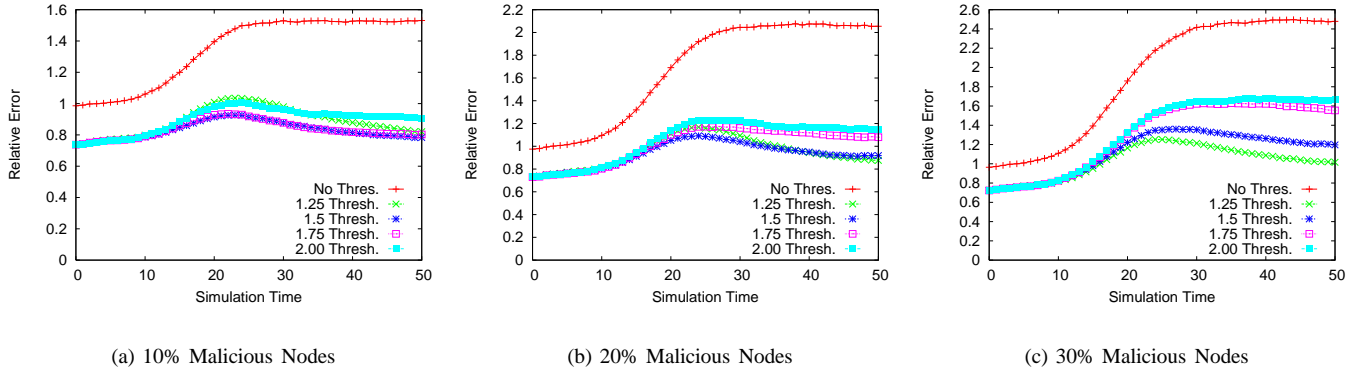


Fig. 3. Vivaldi System Error Ratio Using Various Spatial Outlier Thresholds

metric provides an intuition of how the overall system is performing. The lower the system prediction error is, the more accurate the predicted RTTs.

*Relative error* is defined as

$$Error_{rel} = \frac{Error_{attack}}{Error_{no\_attack}} \quad (2)$$

where  $Error_{attack}$  is the system prediction error measured in the presence of malicious nodes and  $Error_{no\_attack}$  is the system prediction error without malicious nodes. This metric captures the impact an attacker has on the coordinate system. 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 demonstrate the effect of malicious nodes who exploit the virtual coordinate systems by reporting random coordinates and low, non-zero error has on the overall system error in Fig. 1 and Fig. 2. Having even a small percentage of attackers incurs *double* or *triple* the estimation error when compared with the non-malicious scenario. As shown in Fig. 1, as the percentage of attackers increases, the ability of the system to accurately estimate latency significantly degrades. This trend is also evident in Fig. 2, where the system stabilizes at a much higher relative error than the baseline of one.

In order to demonstrate the effectiveness of our defense mechanisms at mitigating the effects of malicious nodes and sustaining the usability of the system, we conducted the same attacker scenario, this time using spatial outlier detection with various thresholds. We used a temporal threshold of 5.0 to allow the five features to vary by at most one standard deviation over each feature from their temporally developed mean. The value was chosen based on the formula of the simplified Mahalanobis distance as in [13]. As can be seen from Fig. 3, our solution mitigates the effects of the malicious nodes and even helps the system to stabilize at a more accurate local minimum than the initial protocol designed to tolerate benign errors. Through testing multiple spatial outlier thresholds empirically, we found that overall, a tighter threshold resulted in better system performance. We found that a spatial threshold of 1.5 worked well for different percentages of attackers.

## V. FUTURE WORK AND CONCLUSION

We study the vulnerability of decentralized virtual coordinate systems to insider attacks. We propose solutions to make coordinate management resilient to attacks without increasing the communication overhead. Initial results indicate that outlier detection is a good strategy to make the accuracy of virtual coordinate systems robust to Byzantine attacks.

Our ongoing work studies the inherent tradeoff between the false positive rate and the system error/convergence. We plan to analyze the effect of inaccurate coordinates resultant from malicious attacks and the effectiveness of our mechanisms on upper level applications which use a virtual coordinate system to estimate network measurements.

## REFERENCES

- [1] B. Zhao, J. Kubiatowicz, and A. Joseph, "Tapestry: An Infrastructure for Fault-tolerant Wide-area Location and Routing," *Computer*, vol. 74.
- [2] T. Ng and H. Zhang, "A Network Positioning System for the Internet," *Proc. USENIX Conference*, 2004.
- [3] A. Rao, S. Ratnasamy, C. Papadimitriou, S. Shenker, and I. Stoica, "Geographic routing without location information," 2003.
- [4] L. Tang and M. Crovella, "Virtual landmarks for the internet," 2003.
- [5] F. Dabek, R. Cox, F. Kaashoek, and R. Morris, "Vivaldi: a decentralized network coordinate system," in *Proceedings of SIGCOMM '04*, 2004.
- [6] M. Costa, M. Castro, R. Rowstron, and P. Key, "PIC: practical Internet coordinates for distance estimation," *Proc. of the ICDCS '04*, 2004.
- [7] K. P. Gummadi, S. Saroiu, and S. D. Gribble, "King: Estimating latency between arbitrary internet end hosts," in *Proc. of SIGCOMM-IMW*, 2002.
- [8] L. wei Lehman and S. Lerman, "A decentralized network coordinate system for robust internet distance," in *ITNG '06*, 2006.
- [9] M. A. Kaafar, L. Mathy, T. Turetli, and W. Dabbous, "Real attacks on virtual networks: Vivaldi out of tune," in *Proc. of LSAD '06*, 2006.
- [10] M. Castro, P. Druschel, A. Ganesh, A. Rowstron, and D. Wallach, "Secure routing for structured peer-to-peer overlay networks," 2004.
- [11] J. Ledlie, P. Gardner, and M. Seltzer, "Network coordinates in the wild," in *4<sup>th</sup> USENIX NSDI*, 2007.
- [12] H. Zheng, E. Lua, M. Pias, and T. Griffin, "Internet routing policies and round-trip-times," *Proc. of the Passive Active Measurement 2005*.
- [13] K. Wang and S. J. Stolfo, "Anomalous Payload-based Network Intrusion Detection," in *Proc. of RAID '04*, September 2004.
- [14] A. Walters, D. Zage, and C. Nita-Rotaru, "Mitigating Attacks Against Measurement-Based Adaptation Mechanisms in Unstructured Multicast Overlay Networks," *Proc. of the ICNP '06*, 2006.
- [15] "p2psim: A simulator for peer-to-peer protocols," <http://pdos.csail.mit.edu/p2psim/>.