Virtual Networks under Attack: Disrupting Internet Coordinate Systems

Mohamed Ali Kaafar
INRIA Sophia Antipolis, FR
mkaafar@sophia.inria.fr

Laurent Mathy*
Lancaster University, UK and LAAS-CNRS, FR
laurent@comp.lancs.ac.uk

Thierry Turletti
INRIA Sophia Antipolis, FR
turletti@sophia.inria.fr

Walid Dabbous
INRIA Sophia Antipolis, FR
dabbous@sophia.inria.fr

ABSTRACT

Internet coordinate-based systems are poised to become an important service to support overlay construction and topology-aware applications. Indeed, through network distance embedding into an appropriate geometric space, such systems allow for accurate network distance estimations with low overhead. However, coordinate systems often rely on good cooperation between nodes for correct coordination and assume that information reported by probed nodes is correct. In this paper, we identify various attacks against coordinate embedding systems and show their effectiveness on two representative positioning systems, namely Vivaldi and NPS. Our study demonstrates that these attacks can seriously disrupt the operations of these systems and therefore the virtual networks and applications relying on them for distance measurements. Through simulations of different potential scenarios where malicious nodes provide biased coordinate information and delay measurement probes, we quantify the effects of attack strategies that aim to (i) introduce disorder in the system, (ii) fool honest nodes to move far away from their correct positions and (iii) isolate particular target nodes in the system through collusion. Our findings confirm the susceptibility of the coordinate systems to such attacks.

1. INTRODUCTION

Recent years have seen the advent of Internet applications (e.g. PASTRY [1], OCEANSTORE [2], ESM [3], SKYPE [4], etc.), which are built upon and benefit from topology-aware overlays. In particular, most if not all of these applications and associated overlays rely on the notion of network proximity, usually defined in terms of network delays or round-trip times (RTTs), for optimal neighbour selection. However, proximity measurements, based on repeated pair-wise distance measurements between nodes, can prove to be very onerous in terms of measurement overheads. Indeed, the existence of several overlays simultaneously can result in significant bandwidth consumption by proximity measurements (i.e. ping storms) carried out by individual overlay nodes [5]. Also, measuring and tracking proximity within a rapidly changing group requires high frequency measurements.

To palliate this problem, network positioning systems, such as [6, 7, 8], were introduced. The thesis of such systems is that if each node can be associated with a “virtual” coordinate in an appropriate space, distance between nodes can be trivially computed without the overhead of a direct measurement. In other words, if network distances can be embedded into a coordinate space where a reasonably accurate position for each node can be established, the measurement overhead produced by this positioning can be amortized over many (un-measured) distance predictions, which drastically reduces the distance measurement sampling cost of the overall system.

Coordinate-based positioning systems achieve desirable properties, such as accuracy, robustness, stability, scalability and low overheads at the expense of slow convergence times ranging from tens of seconds to several minutes. This is several orders of magnitude slower that what is achievable with direct distance measurements between nodes and is often unacceptable for topology-aware applications whose aim is to quickly identify “best nodes”. We therefore contend that coordinate-based positioning systems are an attractive proposition only if they are deployed as a service: every node could run a coordinate system daemon at boot time which would then be capable of providing accurate coordinate estimates to applications and their overlays on request. In essence, the coordinate system could then be seen as a component of a “virtual infrastructure” that supports a wide range of overlays and applications.

But a system providing an “always-on and large scale coordinate service” would also likely be a prime target for attacks, as its disruption could result in the mis-functioning or the collapse of very many applications and overlays. Indeed, as the use of overlays and applications relying on coordinates increases, one could imagine the release of worms and other malware whose purpose is to attack the virtual infrastructure as a whole. It should also be noted that as current proposals for coordinate systems assume that the nodes partaking in the system cooperate fully and honestly with each other – that is that the information reported by probed nodes is correct – this could also make them quite vulnerable to malicious attacks. In

*L. Mathy is also supported by ENSICA, FR.

particular, insider attacks executed by (potentially colluding) legitimate users or nodes infiltrating the system could prove very effective. We believe that understanding how to secure the base of distance prediction for many applications is much more critical than detailing security of the artifacts of any particular application.

In this paper, we study just how potent this danger is for the Vivaldi [9] and NPS [10] coordinate systems. Vivaldi is a prominent representative of purely peer-to-peer-based (i.e. without infrastructure support) positioning systems, while NPS is typical of landmark-based systems. We identify three types of potential attacks against coordinate-based network positioning systems. Specifically, we study how these attacks can lead to inaccuracy of distance prediction. We demonstrate that it is easy to perform Denial of Service (DoS) attacks on such systems. We also analyze simple ways that allow malicious nodes to take control of the embedding coordinates system, as they are able to impose positions onto other honest nodes in the network, without being detected. Finally, we study how conspiracy can be achieved in these systems and how much it could affect them. The "effectiveness" of these attacks on the target systems are demonstrated through extensive simulations.

In [11], we carried out a preliminary study that showed how attackers can impact the (peer-to-peer-based) Vivaldi coordinate system, particularly in the situation where attackers are present from system creation time. In this paper, we extend our work in different ways. We first generalize attack strategies and observe their impact on both peer-to-peer-based and infrastructure-based coordinate systems. We show that infrastructure-based systems can, under some well chosen attack strategies, be as vulnerable as those based on the peer-to-peer paradigm. We consider more sophisticated attacks, that are aimed at defeating the defense mechanisms currently present in some coordinate systems, and show that these mechanisms cannot defend against all types of attacks. We also focus on studying how attackers can destabilize both the Vivaldi and NPS coordinate systems after convergence. The scenarios studied in this paper seem a better match with reality in a practical system deployment, and reflect the emergence of threats carried out by malware in the current Internet.

The rest of the paper is organized as follows. Section 2 provides a brief overview of the embedding coordinate systems. In section 3, we describe in more details the workings of the systems chosen for this study. We identify and classify attacks in Section 4. We demonstrate and study the effects of these attacks, through extensive simulations, in Section 5. Section 6 concludes the paper by putting our findings into perspective.

2. BACKGROUND

In this section, we give a brief survey of recent proposals for coordinate-based network positioning systems.

2.1 Fixed Landmark-based Coordinate Systems

These systems involve a set of landmark nodes, where other nodes compute coordinates according to measurements to these landmarks.

In Global Network Positioning (GNP) [6], the coordinates of the landmarks are first computed by minimizing the error between the measured distances and the estimated distances among the landmark nodes. An ordinary node derives its coordinates by minimizing the error between the measured distances and the estimated distances to the landmarks. GNP uses the Simplex Downhill method to compute node coordinates.

Lighthouse [7] is an extension of GNP that is intended to be more scalable. Although it has a special set of landmark nodes, a joining Lighthouse node does not have to query those global landmarks. Instead, it can query any existing set of nodes to find its coordinates relative to that set, and then transform those coordinates into coordinates relative to the global landmarks.

The Network Positioning System (NPS) [10] extends GNP into a hierarchical coordinate system, where all nodes could serve as landmarks (reference points) for other nodes.

2.2 Decentralized Internet Coordinate Systems

Practical Internet Coordinates (PIC) [8] is one of the recent decentralized coordinate systems using the Simplex Downhill to minimize an objective distance error function (sum of relative errors). It does not require explicitly designated landmarks. It uses an active node discovery protocol to find a set of nearby nodes to use to compute coordinates. Different strategies such as random nodes, closest nodes, and a hybrid of both, are proposed. PIC aims to defend the security of its coordinate system against independent malicious participants using a test based on the triangle inequality. However, [13] and [14] indicate that network RTTs commonly and persistently violate the triangle inequality. A security mechanism based on the fact that the triangle inequality systematically holds, may lead to degradation of the system performance when no malicious node is inside.

Vivaldi [9] is based on a simulation of springs, where the position of the nodes that minimizes the potential energy of the springs also minimizes the embedding error. Vivaldi defends against high-error nodes, but not malicious nodes. Finally, Big-Bang Simulation (BBS) [15] performs a similar simulation to calculate coordinates, simulating an explosion of particles under a force field.

3. NETWORK POSITIONING SYSTEMS IN OUR STUDY

In this paper, we chose to concentrate on two systems: NPS as a representative of the landmark-based approach; and Vivaldi as a representative of the infrastructure-less approach.

3.1 NPS

NPS is a hierarchical design of the GNP system. It aims to recover "gracefully" from either landmark failures, or situations where these special entities of the system and their network access links become performance bottlenecks. The main departure from GNP is that any node that has determined its position can be chosen by a membership server to be a reference point for other nodes. Actually, the membership server randomly chooses eligible nodes to become reference points when the permanent landmarks are too heavily loaded or unavailable. However, to ensure consistency, NPS imposes a hierarchical position dependency among the nodes.

Given a set of nodes, NPS partitions them into different layers. A set of 20 landmarks are placed in layer-0 (or L_0), the top layer of the hierarchy (these permanent landmarks are the fixed infrastructure used to define the bases of the Euclidean space model), and an 8-dimension Euclidean space is used for embedding. Each node in layer L_i randomly picks some nodes in layer L_{i-1} as its reference points. The relative error of the distance prediction between a pair of nodes is defined as:

$$relative\ error = \frac{|actual - predicted|}{min(actual, predicted)}$$

In [10], authors argue that a 3-layer NPS system is already very accurate and can support more than 2 billion nodes.

NPS includes a strategy for mitigating the effects of simple malicious attacks. Indeed, malicious nodes could potentially lie about their positions and/or inflate network distances by holding onto probe packets. The basic idea is to eliminate a reference point if it fits poorly in the Euclidean space compared to the other reference points. Each node, when computing its coordinates, based on measurements from different reference points, would reject the reference that provides a relative error significantly larger than the median error of all other reference nodes. Specifically, assume there are N reference points R_i, at positions P_{Ri}, and the network distances from a node H to these are D_{Ri}. After H computes a position P_H based on these reference points, for each R_i, it computes the fitting error E_{Ri} as $\frac{|distance(P_H, P_{Ri}) - D_{Ri}|}{D_{Ri}}$. Then the requesting node, H, decides whether to eliminate the reference point with the largest E_{Ri}. The criterion used by NPS is that if (1) $max_i E_{Ri} > 0.01$ and (2) $max_i E_{Ri} > C \times median_i(E_{Ri})$, where C is a sensitivity constant, then the reference point with $max_i E_{Ri}$ is filtered (i.e. H tries to replace it by another reference point for future repositioning).

3.2 Vivaldi

Vivaldi is fully distributed, requiring no fixed network infrastructure and no distinguished nodes. A new node computes its coordinates after collecting latency information from only a few other nodes. Basically, Vivaldi places a spring between pairs of nodes (i, j) with a rest length set to the known (measured) RTT (L_{ij}) between them. The current length of the spring is considered to be the distance between the nodes as estimated in the coordinate space. The potential energy of such a spring is proportional to the square of the displacement from its rest length: the sum of these energies over all springs is the error function that Vivaldi nodes try to minimize.

An identical Vivaldi procedure runs on every node. Each sample provides information that allows a node to update its coordinates. The algorithm handles high error nodes by computing weights for each received sample. Each sample used by a node i, is based on measurement to a node j, its coordinates x_j and the estimated error reported by j, e_j. The relative error of this sample is then computed as follows:

$$e_s = | \parallel x_j - x_i \parallel - RTT_{measured} \mid / \; RTT_{measured}$$

The node then computes the sample weight balancing local and remote error : $w = e_i/(e_i + e_j)$, where e_i is the node's current (local) error. This sample weight is used to compute an adaptive timestep, δ defining the fraction of the way the node is allowed to move toward the perfect position for the current sample: $\delta = C_c \times w$, where C_c is a constant fraction < 1. The node then updates its local coordinates as follows:

$$x_i = x_i + \delta \cdot (RTT_{measured} - \parallel x_i - x_j \parallel) \cdot u(x_i - x_j)$$

where $u(x_i - x_j)$ is a unit vector giving the direction of i's displacement. Finally, it updates its local error as $e_i = e_s \times w + e_i \times (1 - w)$.

Vivaldi considers a few possible coordinate spaces that might better capture the underlying structure of the Internet. Coordinates embedding maps the network distances into different geometric spaces, for instance 2D, 3D or 5D Euclidean spaces, spherical coordinates, etc. Vivaldi also introduces the *height model*, consisting of a Euclidean coordinate space augmented with a height. The Euclidean portion models a high-speed Internet core where latencies are proportional to geographic distance, and the height models the time it takes packets to travel the access link from the node to the core. In [9], authors show that the more dimensions a Euclidian space has, the more accurate the Vivaldi system is. Moreover, results show that height vectors perform better than both 2D and 3D Euclidean coordinates, as the height model is a better approximation of the hyperbolic curvature of the Internet [12].

4. THREATS AND ATTACK CLASSIFICATION

We classify attacks and identify threats that malicious nodes may seek to carry out on coordinate-based positioning systems. We consider malicious nodes that have access to the same data as legitimate users. This means that participants are not completely trusted entities, or that malicious nodes have the ability to bypass any authentication mechanisms. Malicious nodes are able to send misleading information when probed, or send manipulated information after receiving a request or affect some metrics observed by chosen targets. The main classes of attacks on positioning system behavior are:

1. Disorder: the main goal of this attack is to create chaos as a form of denial of service (DoS) attack. This results in high errors in the positioning of nodes, or the non-convergence of the algorithm. The attack consists only in maximizing the relative error of nodes in the system, either passively by not cooperating or falsifying its coordinates or by actively delaying probes.

2. Isolation: where nodes would be isolated in the coordinate space. The attack could target a particular node, in order to convince the victim that it is positioned in an remote zone of the network. The ultimate goal of such an attack can be, for instance, obliging the victim to connect to an accomplice node as the closest node in that zone, in order to perform traffic analysis or packets dropping, man in the middle attacks, etc. One way a malicious node can conduct this attack is to delay probes sent by the victim, and to falsify its own coordinates, so that the victim's computed coordinates are set to a value large enough, to be far from other nodes.

3. Repulsion: where a malicious node would convince its victims that it is positioned far them in order to reduce its attractiveness, and then, for instance, alleviate its resource consumption by not cooperating in the application progress. Ways to perform such attacks are to make its conditions (performance, position) seem worse than they actually are. This is accomplished by means of delaying measurement probes and/or by manipulating the coordinates transmitted to other nodes.

4. System Control: This attack is possible on coordinate-based systems that allow "normal" nodes to be considered as landmarks, i.e. most of the existing systems except the centralized systems. In hierarchical systems for example, such as NPS, nodes would try to get higher in the hierarchy in order to fool and influence the maximum number of correct nodes.

These classes of attacks can either be carried out by malicious nodes in an independent manner or as a conspiracy created by colluding nodes. Collusion is likely in scenarios where attack propagation happens through the now well tested means used in today's DDoS attacks (e.g. worms, etc.).

It should be noted that all attacks, be they explicitly aimed at disrupting the whole system or skew the coordinates of a single node, will often result in some distortion of the coordinate space. This is because of the possible cooperation between the nodes that will act as a catalyzer to the propagation of errors to other (non directly targeted) nodes.

5. PERFORMANCE EVALUATION

5.1 Performance Indicators

We use the relative error (defined in section 3) as our main performance indicator. We compute the average relative error over all nodes to represent the accuracy of the overall system. Since our focus is on measuring the impact of malicious nodes on the system, we also introduce the *relative error ratio* (also called "Ratio" for simplicity), which is the relative error measured in presence of malicious nodes normalized to the performance of the system without cheats used as the best case scenario (i.e. $\text{error\_ratio} = \text{error}/\text{error}_{\text{ref}}$). Obviously, a value for the error ratio above 1 indicates a degradation in accuracy.

As the worst case scenario, we also compute the relative error of a coordinate system where nodes choose their coordinates at random. In this random scenario, all nodes choose their coordinate components randomly in the interval $[-50000, 50000]$ (for each dimension of the coordinate).

5.2 Simulation Set-up

We used the "King" dataset to model Internet latencies based on real world measurements. This dataset contains the pair-wise RTTs between 1740 Internet DNS servers collected using the King method [16]. This was used to generate a topology with 1740 overlay nodes, from which we derived various group sizes by picking nodes at random (unless otherwise stated, in the simulations, the group consists of all the 1740 nodes). Each scenario was repeated 10 times with the malicious nodes selected at random within the group. We consider groups with 10%, 20%, 30%, 40%, 50% and 75% of malicious nodes. In view of the infection rates of recent worm epidemics, we believe these values to be realistic, both during and for a long time after an outbreak.

For the Vivaldi simulation scenarios, we used the p2psim discrete-event simulator [17]. Each Vivaldi node has 64 neighbours (i.e. is attached to 64 springs), 32 of which being chosen to be closer than 50 ms. The constant fraction C_c for the adaptive timestep (see section 3.2) is set to 0.25. These values are those recommended in [9]. The system is considered to have stabilized when all relative errors converge to a value varying by at most 0.02 for 10 simulation ticks. We observed that Vivaldi without malicious nodes always converged within 1800 simulation ticks, which represents a convergence time of over 8 hours (1 tick is roughly 17 seconds). Unless otherwise stated, our results are obtained for a 2-dimensional coordinate space.

For NPS, we developed our own event-driven network simulator, based on the description of the protocol in [10] and a reference implementation of the protocol[1]. Unless otherwise stated, as recommended in [10], we considered an 8-dimensional Euclidean space for the embedding. In layer-0, a set of 20 well separated permanent Landmarks are chosen. 20% of nodes are randomly chosen as reference points, in each subsequent layer. For the security mechanism of NPS, the sensitivity constant C was set to 4.

Finally, in this paper, we consider all the attacks in an "injection" context, where the malicious nodes are introduced in a system that has already converged. This is in contrast with a "genesis" attack where the malicious nodes are present from the system's creation time (which we studied in [11] for Vivaldi). The former is more realistic in a practical setting, and reflects the emergence of threats carried out by malware in the Internet.

5.3 Attacks on Vivaldi

5.3.1 *Disorder Attack*

We first discuss ways to achieve disorder attacks in Vivaldi. As it is a fully-distributed algorithm relying on cooperation of nodes in order to ensure accuracy of the computed coordinates, it seems easy to fool honest nodes. The disorder attack has no specific objective, but false coordinate computations and high positioning error. When solicited, a malicious node sends a randomly selected coordinate x_j, associated with a very low error, $e_j = 0.01$. Moreover, each node's measurement is delayed by a randomly generated value in [100..1000] ms. In this first scenario, it is not necessary to care about lie consistency, as Vivaldi uses error weights sent along with the responses to probes to adjust the adaptive timestep. Even if the measured distance $RTT_{measured}$ to malicious node j is not consistent with the coordinates x_j, the victim i would consider itself as a high error node, and would try to adjust its coordinates by a great adaptive timestep value, due to the fact that j sends a low error.

Figure 1 depicts the relative error ratio variation in function of time, for our full set of 1740 nodes, representative of the impact of the malicious nodes on the system. It is clear that enough attackers can quickly destabilize a converged system and seriously reduce the system accuracy. It is interesting to note that, in the presence of enough malicious nodes, despite the system converging in the sense that the relative errors at each node stabilize, these errors are so high that a great variation of the coordinates of a node barely affects the associated error. In other words, the coordinates of the nodes keep showing great variations and do not stabilize but the error introduced by such constant movement is stable because there is already so much chaos in the system. In essence, the system is deemed to converge because it doesn't get any better nor any worse.

Figure 2 shows the cumulative distribution of the relative error of the victims of an injected disorder attack. We clearly see that from 30% of malicious nodes the impact on the system can be considered as very serious with many nodes seeing a large increase in their relative errors. For a proportion of 50% or more malicious nodes, the system collapses with over half of the honest nodes computing coordinates that are similar or worse than if chosen randomly.

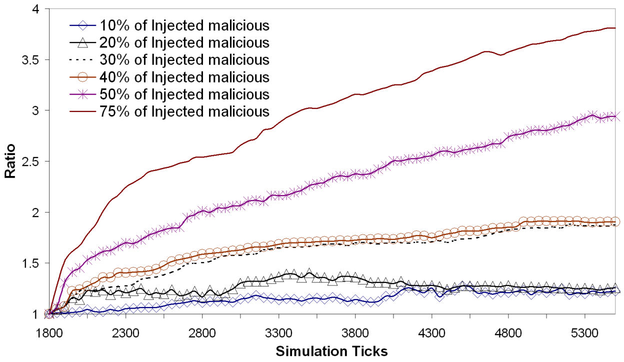

Figure 1: Injection of Disorder Attackers on Vivaldi: average relative error ratio.

Figure 3 represents the impact of the space dimension on the attack. In this figure, the average relative error of honest nodes is measured after re-convergence. We see that the more accurate the Vivaldi system is in the absence of malicious nodes, the more vulnerable it is to the disorder attack. This is because the variation of more coordinate components for a point in a larger space results in

[1] The authors would like to thank Prof. Eugene Ng for sharing his code.

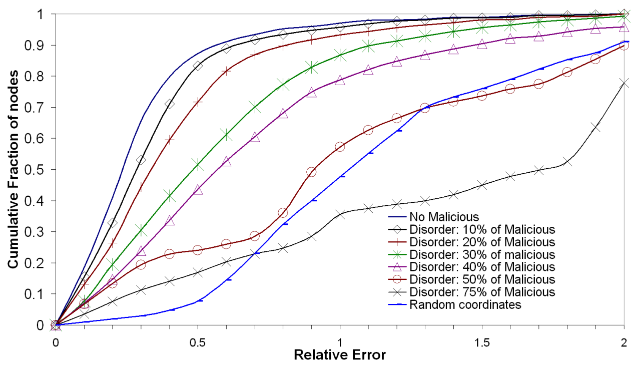

Figure 2: Injected Disorder attack on Vivaldi: CDF of relative error at simulation tick = 5000

higher displacement in that space. This observation is compounded for the 2-dimensional space augmented by a height as a variation of the height yields a greater effect on the node displacement. We also observe that in most cases, Vivaldi with half the population of malicious nodes is worse than a random coordinate system.

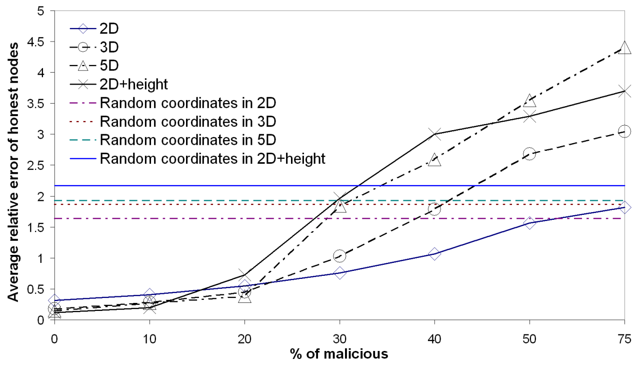

Figure 3: Injected Disorder Attack on Vivaldi: Impact of space dimensions

Figure 4 shows the impact of the attack as a function of the system size as measured a long time after the attack started. We see that a larger system is more difficult to impact for a same proportion of attackers. This is consistent with the fact that a larger Vivaldi system is more accurate, but also establishes that Vivaldi finds increased strength in a larger group. Put simply, this is because as one increases the number of springs in the system, the energy needed to disrupt it is higher. In our case, a larger group means more "good" forces to counteract and dissipate the effect of the malicious ones.

5.3.2 *Repulsion Attack*

In this scenario, malicious nodes are trying to isolate some nodes in the network, either by repulsing a set of targets away from other nodes in the coordinate space, or by repulsing all requesting nodes away from a selected target. The first attack consists in fixing coordinates where to isolate all requesting nodes, say X_{target}. It is important to notice that this value is set high enough to allow lie consistency. This means that the predicted distance after the lie should be equal to the measured distance. In fact, since we assume that a malicious node cannot shorten a distance measurement, but can however delay it, we must set the coordinates of both the victim and the malicious node to be consistent with this fact. Although for most network positioning systems, application probes are used, for generality purposes we design and test the attacks assuming ICMP ping probes. We assume here that malicious nodes know the current coordinates of their targets, $X_{Current}$, by means of previous requests for example. Malicious nodes are then able to compute the needed RTT that are consistent with the lie,

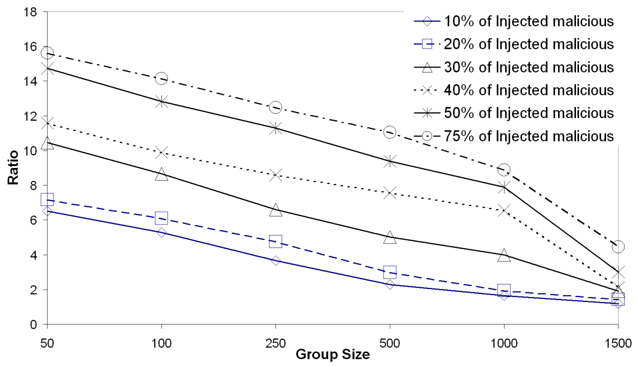

Figure 4: Injection of Disorder Attackers on Vivaldi: Impact of system size on the attack.

$$RTT = (\parallel X_{target} - X_{Current} \parallel /\delta) + \parallel X_{target} - X_{Current} \parallel$$

and to delay the measured RTT by:
$RTT_{needed} - 2 \cdot (ReceivedTimestamp - SendTimestamp)$. Each malicious node is selecting a random coordinate that is far away from the origin.

Figure 5 shows the cumulative distribution function of the measured average relative error after convergence in an repulsion attack. The gentler slope of the curves indicates that the impact of this type of attack is greater than in the case of a disorder attack (see fig. 2). This is because a repulsion attack is more structured and more consistent than a disorder attack, since the chosen target coordinate is always the same for every victim-attacker pair.

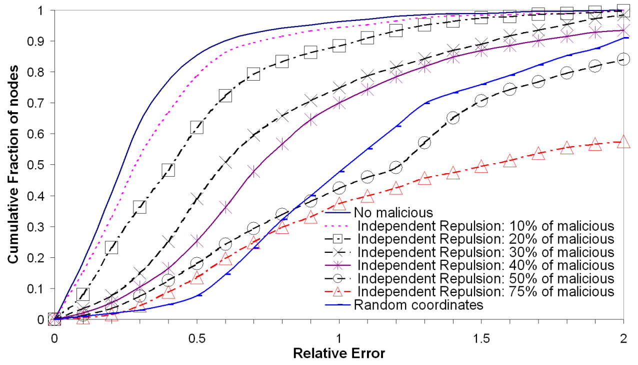

Figure 5: Injected Repulsion Attack on Vivaldi: CDF of relative error.

We study the effect of space dimension on this attack in figure 6. Again, the results confirm that the more accurate the system is without malicious nodes, the more vulnerable it is to attacks, which highlights a fundamental trade-off between accuracy and vulnerability.

So far, the repulsion attack consisted in each attacker attacking every other node. Figure 7 shows the effect of a modified repulsion attack where each attacker independently attacks a subset of

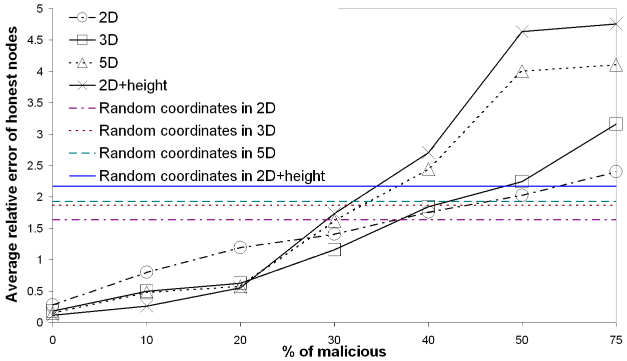

Figure 6: Injected Repulsion Attack on Vivaldi: impact of space dimensions.

the other nodes. Each attacker chooses its own target subset independently, along with their target coordinate values. However, the target subset size is fixed and equal for all attackers. We see that small subsets chosen independently result in a less effective attack and that there is no great difference in effectiveness when the set of attackers constitutes less than 30% the population. This can be explained by the fact that in such conditions the attack gets “diluted”, giving the system plenty of opportunity to correct itself through nodes that are under no, or very little, attack.

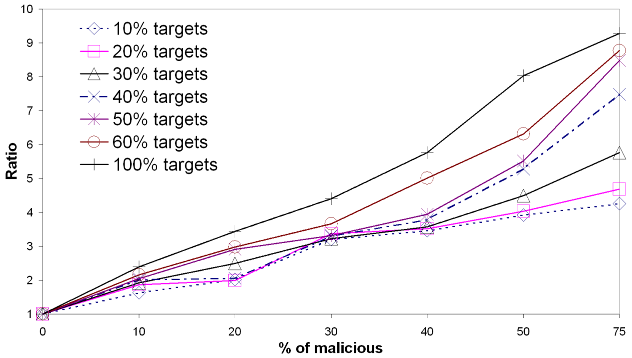

Figure 7: Injected Repulsion Attack on subsets of target nodes.

Figure 8 shows the response of a system under injection repulsion attack as a function of system size. As in the case of a disorder attack, larger systems reduce the impact of the attack. However, because a repulsion attack is much more consistent than a disorder attack, the system is less effective at countering the effects. This is why we observe higher values for the average relative error and a much gentler slope of the curve than in figure 4.

5.3.3 *Colluding Isolation Attack*

This is a repulsion attack where the attackers behave consistently in a collective way. They could, for instance, try and move all honest nodes consistently away from a same designated target node. That is, they agree on a distance from the chosen node for each victim and collectively and consistently direct victims towards their designated coordinate.

Figure 9 depicts the effects of a colluding isolation attack on the system. The salient result is that the system can quickly become worse than a random coordinate system. Indeed, from 30% of malicious nodes in the system, the accuracy becomes equal or worse than if nodes chose their coordinates at random. This clearly demonstrates that colluding attacks are very potent due to their better structure and can have a great adverse impact on overall system performance.

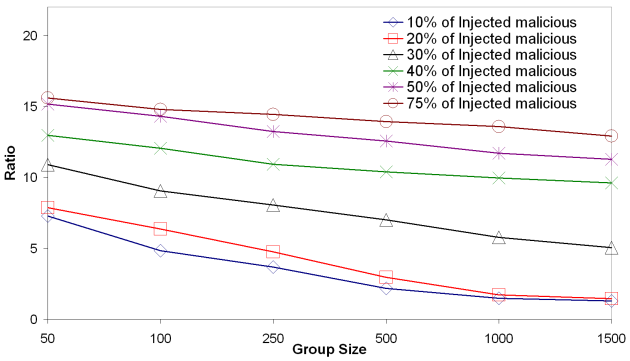

Figure 8: Injection Repulsion Attack on Vivaldi: effect of system size

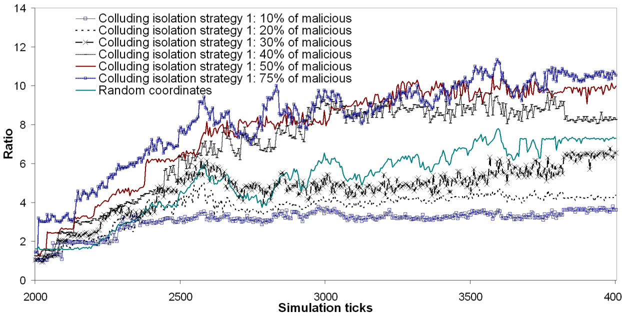

Figure 9: Colluding isolation Attack on Vivaldi: average relative error ratio

Another type of colluding isolation attack is for the attackers to set their coordinates in a remote area of the coordinate space (so that they are clustered in that area) and then to choose a victim target node and convince it that its own coordinate is within the attacker cluster. The target coordinate is set before the attack begins and agreed by all attackers.

We observe in figure 10 the variation of the relative error of the target node through time. We see that the first type of colluding isolation attack (consisting in repelling all other honest nodes from a chosen target) is more effective than trying to lure a target into a remote area of the space. Intuitively, this is because much more error is introduced in the system when more nodes are pushed away from their correct position, thus resulting in more distortion of the coordinate space with greater repercussion on the final position of the target nodes. This is indeed confirmed by the results of figure 11 that depicts the cumulative relative error for the nodes in the system under both types of colluding attacks.

5.3.4 *Combined Attacks*

In the context of system offering an always-on and large scale coordinate service, it is plausible to assume a constant and permanent low level at malicious nodes. Indeed, in the previous sections we have examined the effects of attack outbreaks. But in the wild, as has already been observed after major worm outbreaks and security

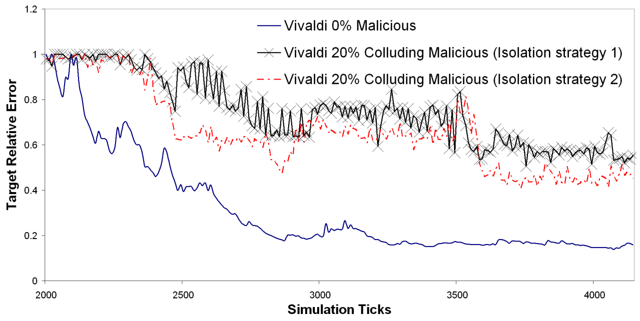

Figure 10: Colluding Isolation attack on Vivaldi: relative error of the target node

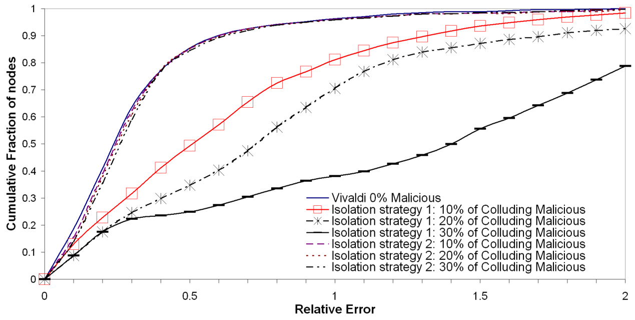

Figure 11: Colluding Isolation Attack on Vivaldi: CDF of relative errors.

warnings, once an outbreak has been contained and resolved, one can expect that some small portion of the systems are not upgraded for a very long time after the release of the necessary patches. This is especially true in the case of systems that are under many different administrative controls (as is the case for home personal computers). Figure 12 shows the impact of such low level combined attacks on Vivaldi, where colluding nodes implement strategy 1 of the colluding isolation attack. In these combined attacks, the percentage of malicious nodes of each type is the same. This figure shows that fairly low level of malicious nodes can still have a sizeable impact on the overall system performance, which, in turn, indicates that return to normality after an attack may take an extremely long time, if at all possible.

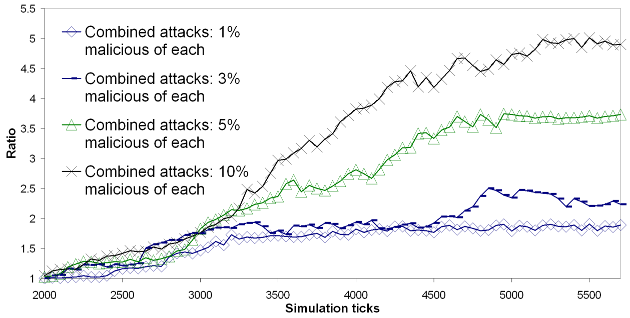

Figure 12: Combining attacks on Vivaldi: impact on convergence.

Finally, figure 13 confirms that larger systems are more resilient and recover better than smaller ones.

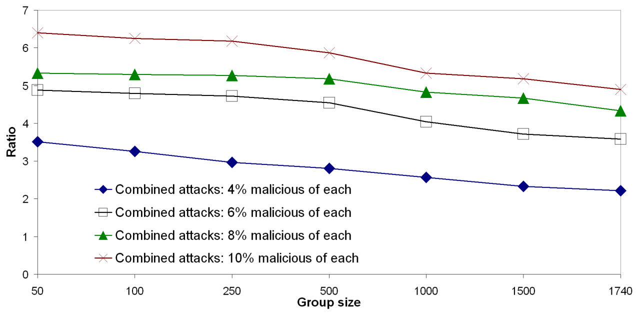

Figure 13: Combined attacks on Vivaldi: effect of system size.

5.4 Attacks on NPS

We experimented with the NPS system in both a secure and non secure version. Unless stated otherwise, the security mechanism is switched on. Note also that we consider the ideal, hypothetical case where the landmarks are highly secure machines that never cheat. The results we present in the following sections can therefore be considered as best case scenarios from a security point-of-view, as the impact of attacks could be much more severe should our security of landmark hypothesis not hold.

5.4.1 Injection of Independent Disorder Attackers

In this first attack, when malicious nodes are chosen as reference points by the membership server (or when an already active reference point gets infected by malware), they perform simple attack that consists in transmitting the correct coordinates of the (malicious) reference point to the victim, and delaying measurement probes without caring about lie consistency. Figure 14 depicts the average relative error variation in function of time, while injecting after convergence of the system, a percentage of malicious nodes. When the malicious reference node detection mechanism is off, we notice the sharp climb in relative error when 20% of malicious nodes join the system. The accuracy of NPS is destroyed when cheating nodes get introduced in layer-1 of the measurement hierarchy. On the other hand, the malicious reference node detection mechanism is shown to be highly effective in combating such a malicious population of up to more than 30% of the overall population. However, a population of 40% or more malicious nodes

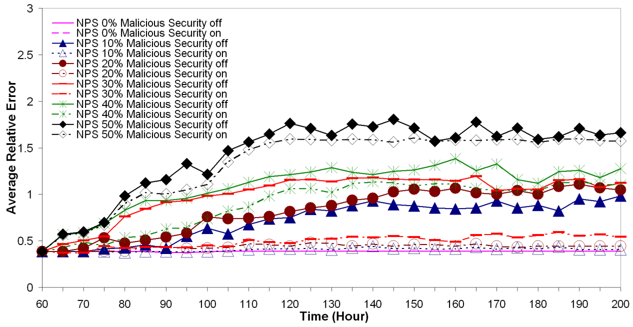

Figure 14: Injection in NPS of Independent Disorder attackers (No prevention): average relative error.

in the system defeats the NPS security mechanism. This can be explained by the fact that the security mechanism relies on simple statistical properties of the observed errors (i.e the median) to filter out perceived outliers. In the presence of enough malicious nodes serving as reference points, the computation of the median itself gets skewed sufficiently that malicious behaviour is assimilated to normal behaviour. The cumulative distribution function of the measured average relative error shown in figure 15 confirms previous results. The gentler slope, and heavy tail feature, of the 40% and 50% curves when security is on indicates the impact of the attack when enough malicious nodes are introduced in the system. We observe that when introducing 40% of malicious nodes, only 50% of honest nodes would re-converge to a relative error less than 0.5.

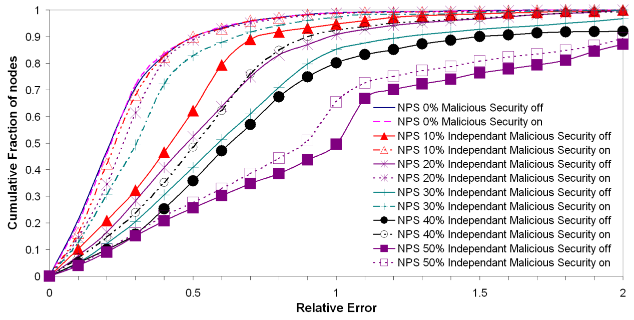

Figure 15: Injection in NPS of Independent Disorder attackers: CDF.

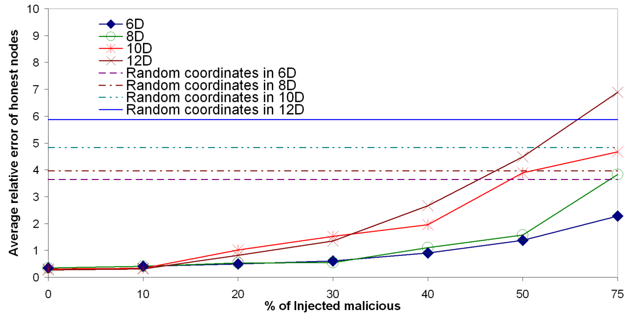

Figure 16: Injection of Independent Disorder attackers: Impact of dimensionality.

Figure 16 shows the effect of space dimension when NPS is subjected to a simple disorder attack. Just as in the Vivaldi case, this experiment proves again that the more accurate the system is without malicious nodes, the more vulnerable to attacks it is. In particular, we observe that with more dimensions used in the coordinate space, the NPS system is much more vulnerable to a smaller portion of malicious nodes. We observe that systems running with 6 and 8 dimensions still can prevent against a minority of malicious nodes, whereas a simple attack can destabilize a 10 or 12-dimensions NPS system more easily. In the later cases, when malicious nodes only constitute 20% of the population, the relative error climbs to more than 1. From 50% of malicious nodes injected in the system, the accuracy becomes equal or worse than if nodes chose their coordinates at random. This is explained by the fact that the more dimensions are used, the more "chances" malicious nodes get to become reference nodes, creating greater confusion among the honest nodes that depend on them in the layers below. Moreover, as in the Vivaldi case, more dimensions result in greater displacement in the coordinate space for the victim.

5.4.2 *Injection of Naive Anti-Detection Disorder Attackers*

In this section, we consider an attack whose primary strategy is to try and defeat the NPS security mechanism. To this end, attackers will lie consistently about their position and inflate network distances by that corresponding amount, while paying particular attention that the relative error computed by the victim is lower than 0.01. Doing so essentially negates the very first condition checked to detect malicious nodes (see section 3.1), in effect shutting down detection of the attackers.

First, we consider that malicious nodes know their targets' coordinates with a probability $p = 1/2$. We discuss next the effect of coordinate information on the efficiency of the attack. The target coordinate information allows first to better estimate the distance between the target and the attacker and second to compute the direction defined in the coordinate space by the nodes themselves. When not available, the malicious node sets a random direction and estimates the distance between itself and the target as $ReceivedTimestamp - SendTimestamp$.

As illustrated in figure 17, the attack consists in delaying the victims' probes by $\| P'_{Ri} - P_{Ri} \| = d'$ such that $\| P'_{Ri} - P_{Ri} \| \gg d$ and then send coordinates P''_{Ri} such that
$\| P''_{Ri} - P_{Ri} \| < 0.01 \| P'_{Ri} - P_{Ri} \|$.

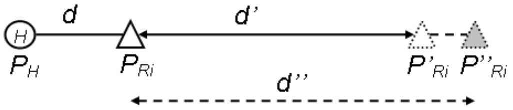

Figure 17: Anti-Detection NPS attack

It is easily shown that $E_{Ri} < 0.01 \Rightarrow d'' > \frac{\alpha+1.99}{0.01} \cdot d$ with $\alpha\, d = d'' - d'$.

To make the attack harder and make the security mechanism of NPS behave in more realistic way, we add a probe threshold condition to each probe, such that a probe would be considered by the requesting node as suspicious if the RTT it measured was above that threshold. Such probes are then discarded. In the following simulations, the probe threshold is set to 5 seconds. In a first scenario, we consider malicious nodes that ignore this probe threshold, yielding a so-called naive anti-detection disorder attack.

In figure 18, we observe the average relative error variation after injection of malicious nodes in a converged NPS system. We see that this attack has a bigger impact on the whole system than the simple disorder attack (see figure 14), causing greater average relative errors. We also observe that the attack is very effective at defeating the security mechanism, with the security-protected relative errors only trailing marginally the errors observed when no security mechanism but the probe threshold is employed. This is despite the attacker guessing half of the time, and could therefore appear surprising. However, the reader should note that the NPS security mechanism discards at most one malicious reference point at each positioning (i.e. the one yielding the greater error), giving the malicious nodes potentially several reprieves on bad guesses.

As for the Vivaldi system, we note that in presence of only a minority of malicious nodes, despite the system converging in the sense that the relative errors at each node stabilize, these errors are so high that a great variation of the coordinates does not affect the

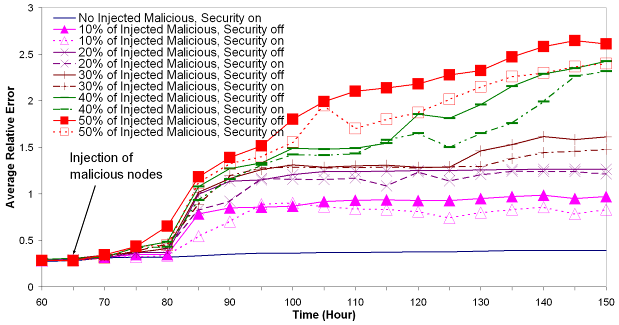

Figure 18: Injection in NPS of Anti-detection naive attackers: impact on convergence.

associated error.

We measured the impact of dimensionality and group size on the effectiveness of this attack and found the now expected results that higher precision (i.e. higher dimensionality) was more affected while larger groups present a better immunity.

More interesting in this attack is the effect the knowledge of the attacker has on its effectiveness. In figure 19, we show the relative error ratio for various probabilities that the attacker knows a victims's coordinates prior to striking. We see that in the presence of a small malicious population, full knowledge of victims' coordinate can almost triple the effectiveness of the attack compared to the pure guess work case. However, as the population of malicious nodes grows, the benefits of more knowledge diminish. This confirms again that, regardless of the sophistication of the attack, the NPS security system soon gets overwhelmed when the population of malicious node exhibits a certain critical mass. As figure 20 shows by representing the ratio of malicious nodes filtered to the overall number of filtered nodes by the security mechanism, this critical mass is about 20% (about half the needed population of malicious node compared to the simple disorder attack). Furthermore, this figure also confirms that, as more and more malicious nodes are able to operate in all impunity, the errors they introduce in the positioning of honest nodes result in higher false positive rates with the security mechanism filtering out more and more (mis-positioned) honest reference points. But because at most one reference point gets filtered per positioning, these false positives actually create some extra protection for the malicious ones.

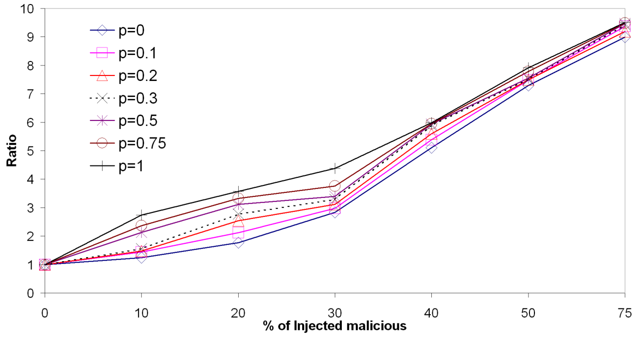

Figure 19: Injection in NPS of Anti-detection naive attackers: effect of victims coordinates knowledge.

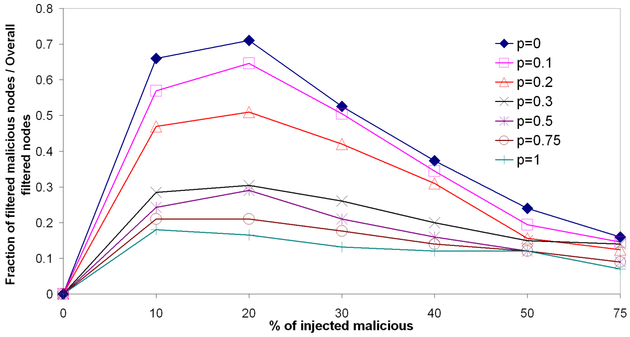

Figure 20: Injection in NPS of Anti-detection naive attackers: effect of victims coordinates knowledge on the ratio of filtered malicious nodes over the overall filtered nodes.

5.4.3 Injection of Sophisticated Anti-Detection Disorder Attackers

We now present a modification of the previous attack where the malicious nodes make an attempt to not only defeat the NPS security mechanism but also avoid detection by the probe threshold mechanism. To do so, an attacker will only interfere with the positioning process of nodes known, or believed, to be nearby. Indeed, if we recall the discussion in section 5.4.2, with a probe threshold of 5 s and $\alpha = 2$, then $d'' + d < 5s \Rightarrow d < 25ms$ in order to avoid detection by the NPS security system, d being the real distance between an attacker and its victim. As this attack is bound to be less detectable by the security mechanisms than the previous one which already yielded small differences between the ”security on” and ”security off” cases, only results in the presence of these security mechanisms are presented here. Unless stated otherwise, the attackers guess the position of their victims half of the time.

Figure 21 shows the cumulative distribution function of the relative errors in a system under sophisticated anti-detection disorder attack. Clearly, this attack is devastating on the overall accuracy of the coordinate system, despite the attackers being more selective of their victims. This is because, even though the errors introduced by each attacker are smaller than in the naive case (nodes that are closer can only be ”pushed” less aggressively if the attacker is to avoid detection), these errors are allowed to permeate unchallenged through the system, propagating more widely through the undetected mis-positioning of honest nodes. We observed that in the system without malicious nodes, the mean relative error converged towards a value of about 0.4. Here we see that as little as 10% of attackers leaves over 60% of the overall population worse off than the average node in a clean system. We also observed that compared with the more naive version of this attack (figure 18), the sophisticated version induces higher average errors.

Again, better accuracy (i.e. higher dimensionality) and smaller group sizes were observed to be more sensitive to the attack.

Figure 22 shows the impact of the attacker's knowledge on the attack. By going from pure guessing to full knowledge (i.e. attacking only victims whose coordinates are known), an attacker can reduce by half its chances of being caught. We also see that the intrinsically more cautious strategy of this attack dramatically reduces the chances of an attacker being detected compared with the naive attack case (figure 20), especially when malicious nodes represent a smaller proportion of the population and operate with little knowledge of the exact coordinates of their victims. Indeed, for the case where the attackers never know exactly the coordinate of their victims, figure 22 shows that over 75% of all detections are false

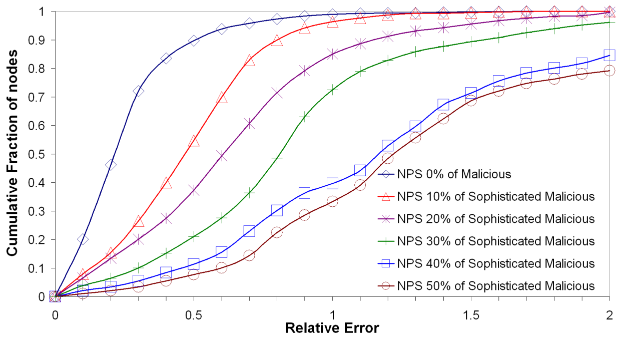

Figure 21: Injected Anti-detection Sophisticated attacks on NPS: CDF.

positives for attackers populations of 10% and over of the group.

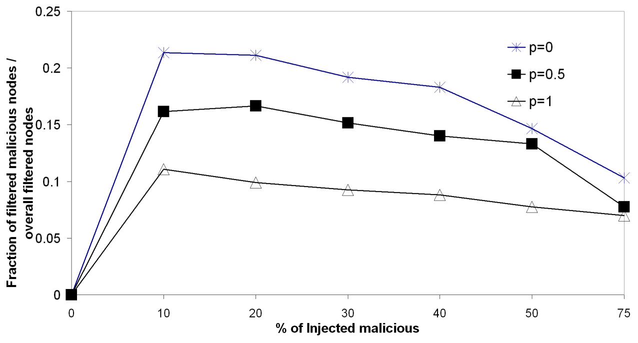

Figure 22: Injected Anti-detection Sophisticated attacks on NPS: effect of victims coordinates knowledge on the ratio of filtered malicious nodes over the overall filtered nodes.

5.4.4 *Injection of Colluding Isolation Attackers*

In a colluding isolation attacks, the malicious nodes cooperate with each other and behave in a correct and honest way until enough of them become reference points at the same layer. Once at least a minimum number of malicious reference points has been reached (in our simulation this number is set to 5), these attackers identify a common set of victims. When involved in the positioning of any other nodes, the attackers do not cheat; while when dealing with a target node, they agree to pretend they are all clustered into a remote (far away) part of the coordinate space and carry out a naive anti-detection attack on the victim. The goal of this attack is to push the victims into a remote location at the "opposite" of where the attackers pretend to be, thus isolation the victims from all the other nodes (in the coordinate space). The other main idea behind this attack is that by acting in a consistent way as a group, the attackers can maybe avoid detection by influencing the value of the median relative error (condition 2 of the NPS security mechanism – see 3.1). Also, as already mentioned, even if detected, at most one attacker would be filtered at each positioning, giving the others more opportunities to act.

We consider 2 scenarios for this attack. The first scenario consists in experimenting with a 3-layer NPS system, i.e. a system with the landmarks in layer-0, 20% of nodes serving as reference points in layer-1, and the rest of the nodes in layer-2. The second scenario is aimed at observing the propagation of errors through different layers and uses a 4-layer NPS system, with 2 layers (layer-1 and layer-2) containing 20% of the nodes acting as reference points.

Figures 23 and 24 show the cumulative distribution function of the relative errors in a 3-layer and 4-layer NPS system (respectively) under this colluding isolation attack. We observe a striking difference of impact depending on the structure of the NPS system. Indeed, the overall accuracy of a 3-layer system is much less affected than the accuracy of a 4-layer system. On the one hand, it is worth remembering that, in the 3-layer system, non victim nodes do not see any degradation of the accuracy of their positions (compared to a clean system), because they observe an honest behaviour from the attackers. This means that the overall degradation in accuracy is caused by the mis-positioning of the victims only. Hence, the perceived little impact of the attack depicted in figure 23 actually tends to indicates that the attack is very effective on the victim.

On the other hand, in a 4-layer system, some of the victims may be unwittingly selected by the membership server to act as layer-2 reference points. The position errors inflicted on these nodes is then propagated through the rest of the system, resulting is an amplification of the errors from layer to layer. This is demonstrated in figure 25 that shows the average relative error of layer-2 and layer-3 nodes in clean 3-layer and 4-layer systems respectively, as well as the average relative error observed by layer-2 targets and layer-3 nodes in corrupted systems with a population of 20% of malicious nodes. From this figure, it is clear that the impact of layer-1 cheats on layer-2 victims is independent of the system structure (the curves are similar), layer-3 nodes of an attacked 4-layer system experience the worse mis-positionning. This propagation and amplification of the errors in this 4-layer system can be seen as a system-control attack (see section 4).

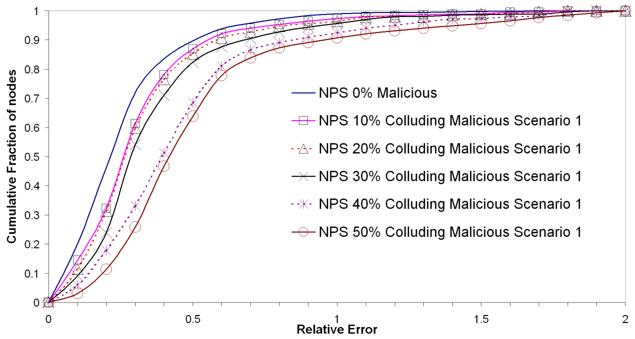

Figure 23: Injection of colluding Isolation attack on NPS in scenario 1: CDF of relative errors.

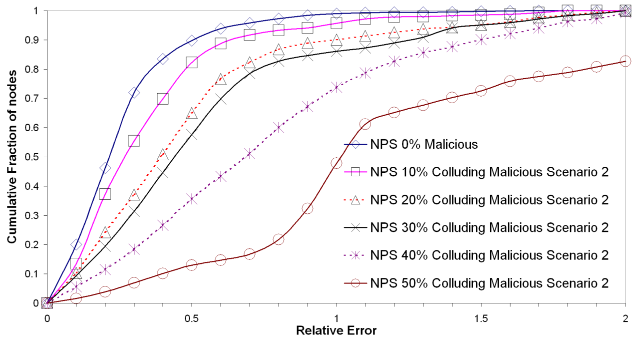

Figure 24: Injection of colluding Isolation attack on NPS in scenario 2: CDF of relative errors.

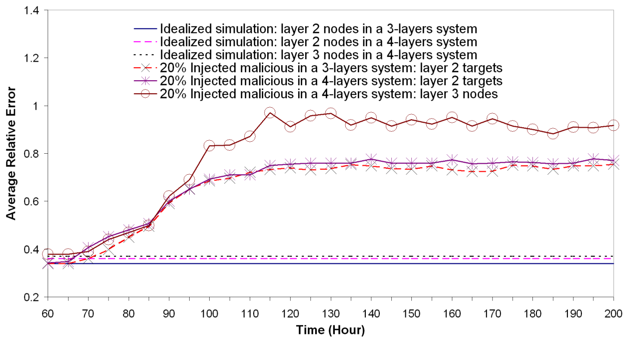

Figure 25: Injection of colluding Isolation attack on NPS: Propagation of errors.

Finally, as in the Vivaldi case, we measured the impact of several small population of attackers which concurrently carry out all the previous attacks. This is reminiscent of a situation where some nodes are still misbehaving for some time following the release of patches and updates after a major outbreak of malware. Again, we see that attacks can have long lasting consequences on the operation of the coordinate system.

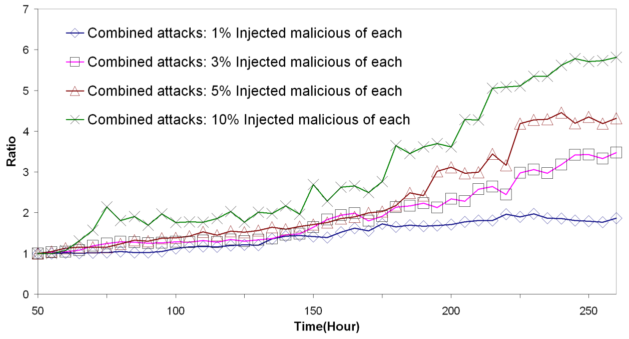

Figure 26: Injection of combined attacks on NPS (Independent disorder, Anti-Detection Sophisticated disorder and colluding isolation attackers): Impact on convergence.

6. CONCLUSIONS

In this paper, we have studied various types of attacks on two prominent coordinate system proposals. One of our salient findings is that larger systems are consistently more resilient than smaller ones. Given the observation in [9] and [10] that larger systems are more accurate and the well known fact that larger systems converge slower at start-up time, there seems to be a compelling case for large-scale coordinate systems to be built as a virtual infrastructure service component. The paradox is of course that always-on, large-scale systems supporting many different applications will always attract more attacks than systems with a smaller reach, while the large size of the system itself would act as a particularly good terrain to create especially virulent propagation of the attack.

Our results also show that there is an intrinsic trade-off to be made between accuracy and vulnerability. Indeed, we have shown that the more accurate the system for a given system size, the more susceptible it was to a same proportionate level of attack.

Also, we have shown that while an attack is in full swing, the performance of the coordinate systems (and of the applications it supports) can easily degrade below that of a systems where coordinates are chosen randomly, whilst the aftermath of an attack could have very long lasting effects on the system due to a small number of remaining malicious nodes.

We have also shown that infrastructure-based systems can, under some well chosen attack strategies, be as vulnerable as those based on the peer-to-peer paradigm. Furthermore, the security mechanisms that have been proposed to date to defend against malicious nodes are clearly rather primitive and still in their infancy and definitely cannot defend against all types of attacks.

In our future work, we will concentrate on designing generic defense and security mechanisms to protect coordinate-based systems from large-scale malicious attacks. This work will be guided by the understanding of attack mechanisms and of their consequences on the coordinate systems gained from the study presented in this paper.

Acknowledgment

This work has been partially supported by the European Union under the E-Next network of Excellence Project FP6-506869.

7. REFERENCES

[1] A. Rowstron and P. Druschel, *Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems*, in Proceedings of IFIP/ACM International Conference on Distributed Systems Platforms, Heidelberg, Germany, November, 2001.

[2] J. Kubiatowicz et al., *OceanStore: An Architecture for Global-Scale Persistent Storage*, in Proceedings of the International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS 2000), Cambridge, November 2000.

[3] Y. h. Chu, S. G. Rao and H. Zhang, *A case for end system multicast*, In Proceedings of the ACM International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS), Santa Clara, June 2000.

[4] `www.skype.com`

[5] S. Rewaskar and J. Kaur, *Testing the Scalability of Overlay Routing Infrastructures*, in Proceedings of of the Passive Active Measurement (PAM) Workshop, Sophia Antipolis, France, April 2004.

[6] T. E. Ng, and H. Zhang, *Predicting internet network distance with coordinates-based approaches*, in Proceedings of the IEEE INFOCOM, New York, June 2002.

[7] M. Pias, et al., *Lighthouses for Scalable Distributed Location*, in Proceedings of International Workshop on Peer-to-Peer Systems (IPTPS0, Berkeley, February 2003.

[8] M. Costa, et al., *Practical Internet coordinates for distance estimation*, in Proceedings of the IEEE International Conference on Distributed Computing Systems (ICDCS), Tokyo, March 2004.

[9] F. Dabek, R. Cox, F. Kaashoek and R. Morris, *Vivaldi: A decentralized network coordinate system*. In Proceedings of the ACM SIGCOMM, Portland, Oregon, August 2004.

[10] T. E. Ng and H. Zhang, *A Network Positioning System for the Internet*, in Proceedings of the USENIX annual technical conference, Boston, June 2004.

[11] M. A. Kaafar, L. Mathy, T. Turletti and W. Dabbous, *Real attacks on virtual networks: Vivaldi out of tune*, In Proceedings of the SIGCOMM workshop on Large Scale Attack Defense LSAD 2006,p129-146, PISA, September 2006.

[12] Y. Shavitt and T. Tankel, *The curvature of the Internet and its usage for overlay construction and distance estimation*, In Proceedinsg of the IEEE INFOCOM, Hong Kong, March 2004.

[13] E. K. Lua et al., *On the accuracy of Embeddings for Internet Coordinate Systems*, in Proceedings of Internet Measurement Conference (IMC), Berkeley, October 2005.

[14] H. Zheng et al., *Internet Routing Policies and Roun-Trip Times*. In Proceedings of the Passive Active Measurement (PAM), Boston, March 2005.

[15] Y. Shavitt and T. Tankel, *Big-bang simulation for embedding network distances in euclidean space*, in Proceedings of the IEEE INFOCOM, San Francisco, April 2003.

[16] K. P. Gummadi, S. Saroiu, and S. D. Gribble, *King: Estimating Latency between Arbitrary Internet End Hosts*, in Proceedings of SIGCOMM Internet Mesasurement Workshop (IMW), Pittsburgh November 2002.

[17] A simulator for peer-to-peer protocols. `http://www.pdos.lcs.mit.edu/p2psim/index.html`