A Reputation-Based Approach for Securing Vivaldi Embedding System

Damien Saucez, Benoit Donnet, Olivier Bonaventure

Université Catholique de Louvain, CSE Department

Abstract. Many large-scale Internet applications optimize their overlay network to reduce latencies. Embedding coordinate systems like Vivaldi are valuable tools for this new range of applications since they propose light-weight algorithms that permit to estimate the latency between any pair of nodes without having to contact them first. It has been recently demonstrated that coordinate systems in general and Vivaldi in particular are sensible to attacks. Typically, nodes can lie about their coordinate and distort the coordinate space. In this paper, we propose a formal reputation model to detect misbehaving nodes and propose a reputation adaptation of Vivaldi called RVivaldi. We evaluate the performance of RVivaldi using the King dataset and show that RVivaldi is less sensitive to malicious nodes than Vivaldi.

1 Introduction

During the last few years, many different application-level overlays have been proposed to support new range of applications from file sharing to Voice over IP (VoIP) and, more recently, IPTV. Most of these applications rely on the network delay or round-trip times (RTTs) to ensure quality of service (QoS). To limit resources consumption of the proximity measures, Internet coordinate systems have been proposed to allow hosts to estimate delays without doing direct measurements [1–3]. Every node of an Internet coordinate system computes its coordinates into a geometric space such that the distance from itself to any host predicts the latency – called *distance* – to that node. Ledlie et al. [4] have shown coordinate systems are valuable tools for distributed systems depending on the topology of the network. However, due to their slow convergence, coordinate systems must be deployed as always-on services available for higher level applications.

Content distribution and file sharing systems can benefit from network coordinates in order to select a number of replicated servers to fetch a data item from. *Azureus*, for instance, was the first large-scale real world application to use a coordinate system. Open-source widely spread systems like Azureus may interest attackers. One can imagine modifying Azureus client to alter the coordinate space and disrupt the whole service or controlling all the traffic to achieve a denial-of-service (DoS) attack.

Large-scale always-on services are prime target for attackers as disruption may result in a mis-functioning of many applications or overlay. Kaafar et al. have recently demonstrated that coordinate systems are sensible to attacks [5].

Kaafar et al. [5] have proposed to separate attacks on coordinate systems in different categories. To summarize, one can say that there are two kinds of attacks. The first one is performed when an honest node asks coordinates to a malicious one. The malicious node replies with false coordinates resulting in a bad latency prediction. Secondly, attackers disrupt the coordinates computation process itself resulting in a deformation of the space of both honest and malicious nodes (i.e., the predicted distances of the entire system are altered).

In this paper, we propose an extension to Vivaldi, called *Reputation-based Vivaldi* (RVivaldi), that ensures the security of Vivaldi. The key idea of RVivaldi is to add two new types of entities in the system: The *RCA*, a certificating agent, and the *surveyors* that estimate the reputation of the classic nodes. We propose a formal model of RVivaldi and validate it using the King data set used in [3]. We show that RVivaldi leads to a better accuracy of the coordinates than Vivaldi in presence of malicious nodes.

The remainder of this paper is organized as follows: Sec. 2 gives a brief overview of Vivaldi, the embedding system on which this paper is based; Sec. 3 presents our reputation model for embedding systems and its application to Vivaldi; Sec. 4 evaluates our solution. Finally, Sec. 5 summarizes this paper and discusses further works.

2 Vivaldi

Vivaldi [3] does not require a fixed network infrastructure and make no distinctions between nodes. A Vivaldi node collects distance information for a couple of neighbors and computes its new coordinates with the collected measures. The idea is that node i is represented as a unitary mass connected to each neighbor j by a spring with the rest length set to the measured RTT (d_{ij}). The actual length of the spring is the distance ($\hat{d}_{ij}$) predicted by the coordinate space. A spring always tries to have an actual length equals to its rest length. Thus if $\hat{d}_{ij}$ is smaller than the measured RTT, the spring will push the two masses attached to it. On the contrary, if the spring is too long, it will pull the masses and reduce its actual length. The coordinates in Vivaldi are updated following this principle. If we note $\overrightarrow{x}_i$ the coordinates of i and $\overrightarrow{x}_j$ the coordinates of j, the new coordinates are computed as follows:

$$\overrightarrow{x}_i = \overrightarrow{x}_i + \delta \cdot \left(d_{ij} - \hat{d}_{ij}\right) \cdot u\left(\overrightarrow{x}_i - \overrightarrow{x}_j\right). \tag{1}$$

which must be understood as the displacement of the mass by a small part of the displacement induced by the spring applying the Hooke's law. δ, the adaptative timestep, defines the fraction of the way the node is allowed to move towards the perfect position for the current information. The timestep depends on the local errors of the two nodes (e_i and e_j) and reduces the displacement if the error is important. The timestep is defined by $\delta = c_s \cdot \omega$ where c_s is a tuning constant and $\omega = e_i/(e_i + e_j)$. When a node has computed its new coordinates, it computes its local error $e_i = e_s \cdot \omega + e_i \cdot (1 - \omega)$ where e_s is the *relative error* defined by Eqn. 2. $u\left(\overrightarrow{x}_i - \overrightarrow{x}_j\right)$ gives the direction of the displacement of i and is normalized to 1.

$$e_s = |d_{ij} - \hat{d}_{ij}|/d_{ij}. \tag{2}$$

Eqn. 1 is the core of Vivaldi since it allows nodes to discover their coordinates.

3 Reputation-Based Vivaldi

3.1 A Reputation Model for Embedding Systems

The *reputation* of an entity A is the combination of trusts of all other entities towards A and the *trust* is a subjective expectation that an agent has about another's future behavior based on the history of their encounters [6]. These definitions suggest that reputation is global and objective while trust is local and subjective. The trust is built on the *experiences* the agent observed about A.

In traditional coordinate systems, any node A updates its coordinates based on the coordinates of one of its neighbors and the distance to it. In our new approach, the new coordinates also depend on the reputation of the neighbors. When A updates its coordinates based on measurements with neighbor B, it first contacts B to retrieve its coordinates and reputation. A then computes its coordinates as a function of its own coordinates, B's coordinates and B's reputation. Then, A contacts a special certification agent, the *Reputation Computation Agent* (RCA) to update its own reputation. This RCA is similar to the RCA proposed by [7]. The RCA is used to construct a reliable reputation for any node in the embedded system. For this, we follow the recently proposed approach by Kaafar et al. [8] and introduce new entities in the system: The *surveyors*. A few surveyors are attached to each node in the system. Surveyors are well chosen nodes that perform experiences measurements and trust estimation on other nodes. Next, the RCA computes its own trust to A's surveyors. Finally, the RCA computes the new reputation of A with all these parameters. The RCA introduces scalability issue so that a solution must be found to allow replication of this entity [9].

We now propose a more formal approach to the notions of experience, trust and reputation.

Experience Model. At time t, an experience is an observation of a node A about some behavior of another node B. This observation is evaluated as follows.

$$\xi(A,B,t) = 1 - \frac{\left|\hat{d}(A,B,t) - d(A,B,t)\right|}{\max\left(d(A,B,t), \hat{d}(A,B,t)\right)}. \tag{3}$$

Where $\hat{d}(A,B,t)$ is the estimated distance between A and B and $d(A,B,t)$ is the real distance. This metric is derived from the relative error (Eqn. 2).

The relative error gives information about the accuracy of the predicted distances. The lower the relatives errors are, the accurate the coordinates are. The experience converts the relative error in the bounded interval $[0, 1]$. The experience is maximum for a perfect estimation and decreases with the augmentation of the *prediction error*.

Trust Model. The trust A has in B is an expectation of the future behavior of a node based on the previous experiences A had in B. However, the experience that we defined before depends on external elements and is inherently not absolutely reliable. We use the concept of uncertain probabilities proposed by Jϕsang [10] to model this doubt. These uncertain probabilities introduce the concepts of *belief* (b), *disbelief* (d) and *uncertainty* (u). The belief is the probability that the affirmation is true while the disbelief is the probability that this affirmation is false. The uncertainty quantifies the doubt associated to the affirmation. These three concepts together compose an *opinion* which is a tuple $\omega = (b, d, u)$ [11]. Belief, disbelief and uncertainty are linked together by the *belief function additivity* which states that $b + d + u = 1$.

Conceptually, the trust must limit the risk of nodes using multiple identities. It incites therefore nodes to remain in the system for a long time. However, the trust must be reactive enough to adapt to sudden changes in the topology [7]. These requirements can be achieved by using the concept of *trustworthiness*. The trustworthiness $\tau(A, B, t)$ of A in B at time t is an exponentially averaged sum of the experiences [12] multiplied by an ageing factor:

$$\tau(A, B, t) = a(t) \cdot \gamma \cdot \left(\sum_{i=0}^{h} (1 - \gamma)^i \cdot \xi(A, B, t - i) \right). \tag{4}$$

where $a(t)$ is the ageing factor $(a(0) = 0)$, γ is a weighting constant and h is the number of previous experiences that must be taken into account. The exponentially averaged sum of the experiences gives more importance to the most recent experiences [12]. The ageing factor increases with the seniority and limits the trustworthiness of recent nodes (similar to the loss factor proposed in [13]). It is defined as follows:

$$a(t) = c_a + (1 - c_a) \cdot a(t - 1). \tag{5}$$

Where c_a is the age bonus coefficient such that $0 < c_a < 1$ and $a(0) = 0$. c_a controls the gain of the age for the trust computation. The value of c_a is a tradeoff between wisdom and convergence time. A low value of c_a implies a slow convergence to 1, meaning that only old nodes may completely benefit from the experiences. On the contrary, a large value quickly increases the ageing factor to 1 allowing recent nodes to use their entire experience rapidly.

The *untrustworthiness*, $\bar{\tau}(A, B, t)$, is the complement to 1 of the trustworthiness. The *doubt* $\varepsilon(A, B, t)$ A has in B at time t is the variation of the experiences with the time. This variation is estimated with the variance of the last h experiences:

$$\varepsilon(A,B,t) = \sigma\left(\bigcup_{i\in\{0..h\}} \xi\left(A,B,t-i\right)\right). \tag{6}$$

The model of uncertain probabilities offers strong perspectives to the reputation in general. The trust $\omega(A,B,t) = (b_B^A(t), d_B^A(t), u_B^A(t))$ the node A has in B at time t has the following bijection with trustworthiness, untrustworthiness and doubt:

$$\begin{array}{rl} b_B^A(t) &= \frac{\tau(A,B,t)}{(\tau(A,B,t)+\bar{\tau}(A,B,t)+\varepsilon(A,B,t))} \\ d_B^A(t) &= \frac{\bar{\tau}(A,B,t)}{(\tau(A,B,t)+\bar{\tau}(A,B,t)+\varepsilon(A,B,t))} \\ u_B^A(t) &= \frac{\varepsilon(A,B,t)}{(\tau(A,B,t)+\bar{\tau}(A,B,t)+\varepsilon(A,B,t))}. \end{array} \tag{7}$$

Reputation Model. The reputation of an entity at a particular time must be unique and must be a function of the trust that all nodes have in it. However, for scalability reasons, it is impossible to construct a fully-meshed reputation model where each node cooperates with all others to exchange trust information. We therefore propose a *pseudo-reputation* model in which only a few nodes cooperate to determine the reputation.

The uncertain probabilities model proposes two evidential operators [10, 11]: *discounting* ($\otimes$) and *consensus* ($\oplus$). The first one can be seen as an operator of transitivity and the second as an operator of averaging. Each node has a set of well-chosen surveyors assigned to it [8]. The surveyors are normal nodes in the system. A surveyor measures the experiences of its assigned nodes. When the reputation of node A has to be updated, the RCA computes its trust in the A's surveyors and combines these trusts with the trust the surveyors have in A. This process is formalized as follows:

$$\hat{\omega}_A^{RCA} = \bigoplus_{\{H_n\in\mathcal{S}_A\}} \tilde{\omega}_{H_n}^{RCA} \otimes \hat{\omega}_A^{H_n}. \tag{8}$$

Where $\tilde{\omega}_{H_n}^{RCA}$ is the opinion the RCA has in H_n, the nth surveyor of A. In this opinion, the experience is not computed with Eqn. 3 but with Eqn. 9. This particular experience is introduced to avoid the RCA to have to compute its own coordinates. Indeed, if the RCA had coordinates, it would be easy for an attacker to alter the coordinates of the RCA and invalidate the reputation model.

$$\xi(RCA, H_n, t) = 1 - \frac{\sqrt{\sigma\left(\overrightarrow{v}_A^{H_n}(t)/n^2\right)}}{\#\overrightarrow{v}_A^{H_n}(t)}. \tag{9}$$

$\overrightarrow{v}_A^{H_n}(t)$ is the variation history. $\#\overrightarrow{x}$ is the size of the variation history vector and n is a normalization factor computed by Eqn. 11. The variation history vector is the history at time t of the last h variations of coordinates that the

RCA has observed for node A (see Eqn. 10). The intuition behind the division by $\#\overrightarrow{v}_A^{H_n}(t)$ is that a large variance in a small set is more abnormal than a similar variance in a large set. The normalization is used to bound the experience within $[0, 1]$.

$$\overrightarrow{v}_A^{H_n}(t) = \langle \|\overrightarrow{c}_{t-h} - \overrightarrow{c}_{t-h+1}\|, \ldots, \|\overrightarrow{c}_{t-1} - \overrightarrow{c}_t\| \rangle \, . \tag{10}$$

$$n = argmax \left(\bigcup_{\{H_n \in \mathcal{S}_A\}} \overrightarrow{v}_A^{H_n}(t) \right) . \tag{11}$$

We define the scalar reputation $\hat{\varrho}_A$ of a node A based on the opinion $\hat{\omega}_A^{RCA}$ in the equation 12.

$$\hat{\varrho}_A = \hat{b}_A^{RCA} \cdot (1 - \hat{u}_A^{RCA}). \tag{12}$$

The scalar reputation is the belief in A weighted by the doubt that persists on that affirmation. A receives the scalar reputation as a time-limited ticket. Hence, the RCA is never contacted to retrieve coordinates and does not become a bottleneck [7]. The ticket is digitally signed by the RCA to avoid tampering.

More details about the ageing factor and reputation are given in [9].

3.2 Application to Vivaldi

It is possible to improve the robustness of Vivaldi by introducing the notion of reputation in Eqn. 1 of Vivaldi which computes the new coordinates of B based on the knowledge B has in A. This modification is presented in Eqn. 13.

$$\overrightarrow{x}_B = \overrightarrow{x}_B + (\hat{\varrho}_{\mathbf{B}} \cdot \delta) \cdot \left(d_{BA} - \hat{d}_{BA} \right) \cdot u\left(\overrightarrow{x}_B - \overrightarrow{x}_A\right) . \tag{13}$$

Vivaldi has been proposed for environments without attackers and works well in that case. The idea is to keep Vivaldi when the neighbor is reliable and to limit the modification of coordinates if the neighbor is not reliable. When the reputation is at its maximum (i.e., $\hat{\varrho}_B = 1$), the modification is the same as traditional Vivaldi. On the contrary, when the reputation is at its worst (i.e., $\hat{\varrho}_B = 0$), the coordinates are not modified.

4 Evaluation

We validate our proposition using the King data set, as performed in the original Vivaldi experimentation [3]. This data set gives a matrix of RTTs between 1740 nodes spread around the world. Our simulator considers 32 neighbors randomly chosen among the entire set of 1740 nodes. The attackers (i.e., the malicious nodes) are fixed at the beginning of the simulation. Attackers reply with random coordinates each time a node asks coordinates. Such an attack is called a *random coordinates attack*. Note that the reputation is protected such that a malicious

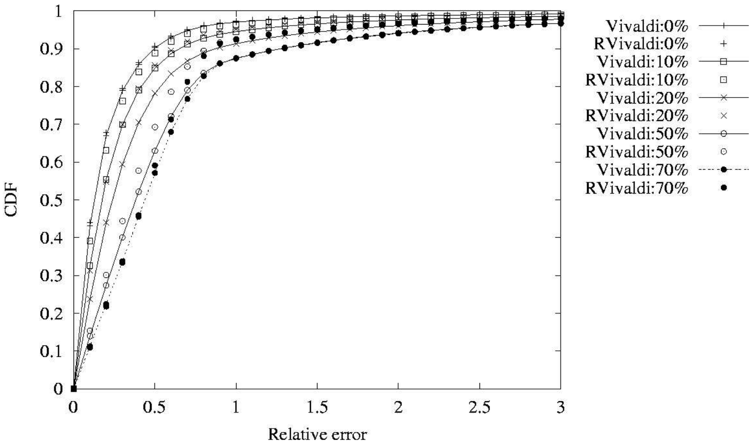

Fig. 1. Random coordinate attack: CDF of relative error at simulation tick = 4000

node cannot modify its own reputation. Only the RCA is able to modify a reputation of the nodes and the RCA is perfectly reliable.

We consider the following proportion of malicious nodes among the entire nodes set: 0% (i.e., there is no attack), 10%, 20%, 50% and 70%. We compare the performance of classic Vivaldi with RVivaldi, as described in Sec. 3.2. The constants used during the simulation are $c_c = 0.25, c_s = 0.25, h = 10, \gamma = 0.5$ and $c_a = 0.01$. The coordinate space is the classic 3-dimensions Euclidean space.

The relative errors (Eqn. 2) are good estimators of the accuracy of the coordinates. Fig. 1 shows the cumulative distribution of the relative error of the victims of a random coordinates attack. The vertical axis is the cumulative fraction of nodes and horizontal axis is the value of the relative error. In absence of attackers, RVivaldi does not provide better results than Vivaldi. However, when some malicious nodes are present, RVivaldi outperforms Vivaldi, whatever the relative errors. For example, in presence of 20% of attackers, RVivaldi is as accurate as Vivaldi in presence of only 10% of malicious nodes. RVivaldi mainly outperforms Vivaldi for relative errors between 0.4 and 1.5. Regarding lower or higher relative error, RVivaldi and Vivaldi converge in the same manner. This error can be explained by the error introduced by the coordinate system itself. A more precise evaluation of RVivaldi can be found in [9].

These results confirm that adding reputation to coordinate systems permit to reduce the incidence of attacks on the accuracy of the whole system.

5 Conclusion

Coordinate systems, such as Vivaldi, might be used in various applications where the notion of proximity, expressed as network delay or RTT, is used. However, it has been recently shown that such a system is sensible to attacks. Indeed, a malicious node can lie about its coordinates and, as a consequence, deform the coordinate space. In this paper, we proposed a formal reputation model for coordinate systems. The reputation gives an estimator of the probability a node lies about its coordinates based on the previous experiences with this node.

In addition, we applied this model to Vivaldi and proposed RVivaldi, a reputation adaptation of Vivaldi. We validated RVivaldi using the King data set and showed that adding reputation to Vivaldi improves the accuracy of coordinates in presence of malicious nodes.

In the near future, we aim at validating RVivaldi in a real environment, such as PlanetLab. We further aim at confronting RVivaldi with all the attacks proposed by Kaafar et al. [5]. Moreover, a solution must be proposed to secure the surveyors and the RCA and to avoid it to be a single point of failure.

Acknowledgements

This work was partially supported by the European-founded 034819 OneLab project. Authors would like also to thank Laurent Mathy for his suggestions.

References

1. Francis, P., Jamin, S., Paxson, V., Zhang, L., Gruniewicz, D.F., Jin, Y.: An architecture for a global internet host distance estimator service. In: Proc. IEEE INFOCOM. (1999)
2. Ng, T.S.E., Zhang, H.: A network positioning system for the internet. In: Proc. USENIX Annual Technical Conference. (2004)
3. Dabek, F., Cox, R., Kaashoek, K., Morris, R.: Vivaldi, a decentralized network coordinated system. In: Proc. ACM SIGCOMM. (2004)
4. Ledlie, J., Gardner, P., Seltzer, M.: Network coordinates in the wild. In: Proc. Symposium on Networked Systems Design and Implementation (NSDI). (2007)
5. Kaafar, M., Mathy, L., Turletti, T., Dabbous, W.: Virtual networks under attack: Disrupting internet coordinate systems. In: Proc. ACM CoNEXT. (2006)
6. Kinateder, M., Rothermel, K.: Architecture and algorithms for a distributed reputation system. In: Proc. 1st Conference on Trust Management. (2003)
7. Gupta, M., Judge, P., Ammar, M.: A reputation system for peer-to-peer networks. In: Proc. 13th ACM International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV). (2003)
8. Kaafar, M., Mathy, L., Barakat, C., Salamatian, K., Turletti, T., Dabbous, W.: Securing internet coordinate embedding systems. In: Proc. ACM SIGCOMM. (2008)
9. Saucez, D.: Securing network coordinate systems. Master's thesis, Université Catholique de Louvain (UCL), Belgium (2007)
10. Jϕsang, A.: A logic for uncertain probabilities. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems **9**(3) (2001) 279–311
11. Twigg, A.: A subjective approach to routing in p2p and ad hoc networks. In: Proc. 1st Conference on Trust Management (iTrust). (2003)
12. Yu, B., Singh, M., Sycara, K.: Developping trust in large-scale peer-to-peer systems. In: Proc. 1st IEEE Symposium on Multi-Agent Security and Survivability. (2004)
13. de Launois, C., Uhlig, S., Bonaventure, O.: Scalable route selection for IPv6 multihomed sites. In: Proc. IFIP Networking. (2005)