

Towards a Two-Tier Internet Coordinate System to Mitigate the Impact of Triangle Inequality Violations

Mohamed Ali Kaafar¹, Bamba Gueye^{1*}, Francois Cantin^{1**}
Guy Leduc¹, Laurent Mathy²

¹ University of Liege, Belgium
Research Unit in Networking (RUN)
{ma.kaafar, cabgueye, francois.cantin, guy.leduc}@ulg.ac.be
² University of Lancaster, UK
Computing Department
laurent@comp.lancs.ac.uk

Abstract. Routing policies or path inflation can give rise to violations of the Triangle Inequality with respect to delay (RTTs) in the Internet. In network coordinate systems, such Triangle Inequality Violations (TIVs) will introduce inaccuracy, as nodes in this particular case could not be embedded into any metric space. In this paper, we consider these TIVs as an inherent and natural property of the Internet; rather than trying to remove them, we consider characterizing them and mitigating their impact on distributed coordinate systems.

In a first step, we study TIVs existing in the Internet, using different metrics in order to quantify various levels of TIVs' severity. Our results show that path lengths do have an effect on the impact of these TIVs. In particular, the shorter the link between any two nodes is, the less severe TIVs involved in are.

In a second step, we do leverage our study to reduce the impact of TIVs on coordinate systems. We focus on the particular case of the Vivaldi coordinate system and we explore how TIVs may impact its accuracy and stability. In particular, we observed correlation between the (in)stability and high effective error of nodes' coordinates with respect to their involvement in TIVs situations. We finally propose a Two-Tier architecture opposed to a flat structure of Vivaldi that do mitigate the effect of TIVs on the distances predictions.

Keywords: Internet Coordinate Systems, Performance, Experimentation, Triangle Inequality Violations.

1 Introduction

As innovative ways are being developed to harvest the enormous potential of Internet infrastructure, a new class of large-scale globally-distributed network services and applications (e.g. [1] [2], etc) have emerged. To achieve network topology-awareness, most, if not all, of these overlays rely on the notion of proximity, usually defined in

* B. Gueye is supported by the EU under the ANA FET project (FP6-IST-27489).

** F. Cantin is a Research Fellow of the Belgian Fund for Research in Industry and Agriculture (FRIA).

terms of network delays or round-trip times (RTTs), for optimal neighbor selection during overlay construction and maintenance.

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 [3]. Also, measuring and tracking proximity within a rapidly changing group requires high frequency measurements.

To palliate such problems, Internet coordinate systems [4,5] have been introduced. These systems embed latency measurements amongst samples of a node population into a metric space and associate a network coordinate vector (or coordinate in short) in this metric space with each node, with a view to enable accurate and cheap distance (i.e. latency) predictions amongst any pair of nodes in the population.

However, Internet latencies, due to routing policies or path inflation [6], do sometimes violate the triangle inequalities which must hold in a metric space. Such Triangle Inequality Violations (TIV) could be a major barrier for the accuracy of Internet coordinate systems. Suppose we have a network with 3 nodes A , B and C , where $d(A, B)$ is 1 ms , $d(B, C)$ is 2 ms , and $d(A, C)$ is 5 ms , with $d(X, Y)$ denoting the measured delay between X and Y . The triangle inequality is violated because $d(A, B) + d(B, C) < d(A, C)$. Such violations make coordinates embedding of network distances less accurate. When faced with these TIVs, coordinate systems resolve them by forcing edges to shrink or to stretch in the embedding space; this intuitively results in oscillations of the embedded coordinates, and thus causes large distance prediction errors. Indeed, ultimately Internet coordinate systems are used to estimate distances between nodes, based on their coordinates only, even and all the more so if these nodes have never exchanged a distance measurement probe. Both a reasonably stable and an accurate coordinate should then be computed.

A few works considered removal (or at least the exclusion) of the Triangle Inequality violator nodes from the system to decrease the embedding distortion [7,8]. However, we claim that sacrificing even a small fraction of nodes, is not arguable since TIVs are an inherent and natural property of the Internet; rather than trying to remove them, we consider characterizing them and mitigating their impact on distributed coordinate systems. A recent work [9] proposes also to remove from the set of neighbors, nodes that underestimate their actual distances to others. These nodes are assumed to be involved in TIVs situations, and need to be removed. This approach however restricts the neighborhood selection to the closest nodes, depriving the coordinate systems from a desirable property, namely the hybrid selection of neighbors.

In this paper, we first study the distributions of TIVs existing in the Internet, and we characterize their severity using different metrics. One of our findings is that longer edges cause more severe TIVs. That is to say, that considering shorter paths as measurement samples in coordinate systems would less likely lead to severe TIVs. Based on this insight, we do leverage our study to reduce the impact of TIVs on coordinate systems. To illustrate our results, we focus on the Vivaldi coordinate system, as a prominent representative of purely peer-to-peer (i.e. without infrastructure support) based coordinate systems. We then study the ways in which TIVs impact the Vivaldi coordinate system.

We showed that TIVs seriously impact the embedding accuracy and coordinates stability. In fact, we observed that nodes that are more involved in TIVs situations are twice less accurate. These nodes' coordinates have also been shown to have oscillations of larger amplitudes. We finally propose a Two-Tier architecture opposed to a flat structure of Vivaldi, based on the clustering of nodes. Inside these clusters, nodes compute coordinates to predict local distances, and keep predicting distances to nodes outside their clusters based on the original 'flat Vivaldi'. This hierarchical approach does mitigate the effect of TIVs on the distances predictions, and allows nodes to embed short distances with very low relative errors.

2 Analysis of triangle inequalities in the Internet

We used the *p2psim* data (1740 nodes) [10] and *Meridian* data (2500 nodes) [11] to model Internet latency based on real world measurements. These data sets are obtained following the *King* [12] measurement technique. King is a technique (similar to *ping*) that estimates the latency between arbitrary end hosts by using recursive DNS queries. Based on these delay matrices, we study through different metrics the violations of the Triangle Inequality, and characterize their severity and distribution according to path lengths.

2.1 Severity metrics

Previous studies [13,6,14] have reported characteristics of TIVs in the Internet delay space by triangulation ratio distribution and the fraction of triangles that suffer from TIVs. Let us consider a triangle ABC . By convention, AB is always the longest edge of a triangle. If $d(A, B) > (d(A, C) + d(C, B))$, then ABC is called a TIV, because the triangle inequality is violated. Note that it is enough to consider the inequality with respect to the longest edge AB of the triangle.

In this paper, we propose two basic characterizations of the *severity* of TIVs. The first one is the *relative severity* and is defined as follows:

$$G_r = \frac{d(A, B) - (d(A, C) + d(C, B))}{d(A, B)} \quad (1)$$

G_r ranges from 0 (minimum severity) to 1 (maximum severity).

Relative severity is an interesting metric, but it may be argued that for small triangles, a high relative severity may not be that critical. Therefore we also define a second metric called the *absolute severity*, which is defined as follows:

$$G_{ar} = \frac{d(A, B) - (d(A, C) + d(C, B))}{Diameter} \quad (2)$$

Note that we have normalized this metric with respect to the *Diameter* of the network, so that it also ranges from 0 (minimum severity) to 1 (maximum severity). Note that *Diameter* represents the maximal delay between any two points in the network.

In the sequel, we will refer to specific severity thresholds and select TIVs whose severities are above them. We can select all TIVs such that $G_r \geq th_r$, or $G_{ar} \geq th_{ar}$, or even when both thresholds are exceeded.

2.2 Results

We first define some notations. Let K be the total number of triangles in the two data sets. For the p2psim data set, we found $K = 854, 773, 676$, of which 105, 329, 511 (representing 12%) are TIVs, whereas $K = 2, 598, 842, 308$ and the percentage of TIVs is equal to 23.5% for Meridian data set. We divide the whole range of RTTs in our data set (from 0 to $Diameter$) into 160 (resp. 600) equal bins of $5ms$ each for p2psim data (resp. Meridian). The maximum delay between any two nodes (*i.e.* $Diameter$) in p2psim data and Meridian data are respectively $800 ms$ and $3000 ms$. Probably, the large diameter noticed in Meridian data is due to the presence of few outliers within the data set. However, only 0.03% of all pair-wise RTT measurements in Meridian data are above $1000 ms$.

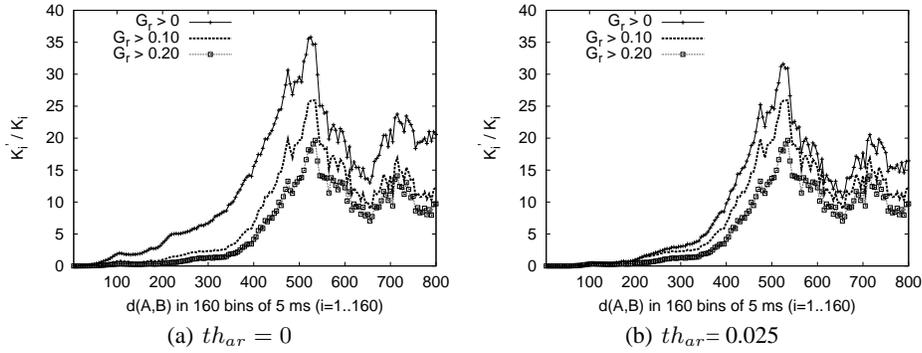


Fig. 1. Proportion of TIVs in each bin for various TIV severity levels for p2psim data set.

Let K_i be the number of triangles in the i th bin. By convention, we say that triangle ABC is in bin i if its longest edge AB is in that bin. Let K'_i be the number of TIVs in the i th bin.

We begin our analysis by showing the proportion of TIVs in each bin, namely $\frac{K'_i}{K_i}$, for different severity thresholds. Figure 1(a) (resp. Figure 2(a)) only considers relative TIV severities (as $th_{ar} = 0$). In Figure 1(b) (resp. Figure 2(a)), we filter out TIVs whose absolute severity is below $th_{ar} = 0.025$ (resp. $th_{ar} = 0.005$), which actually means below $20 ms$ (resp. $15 ms$) with respect to our diameter of $800 ms$ (resp. $3000 ms$ for Meridian). All these curves have basically the same shapes. We can see clearly that large triangles (say above $400 ms$) are more likely (severe) TIVs.

Let P_i the probability for a triangle ABC , chosen at random in the data set, to be (a) in the bin i and (b) to be a (severe) TIV, namely:

$$P_i = \frac{K'_i}{K_i} * \frac{K_i}{K} = \frac{K'_i}{K} \quad (3)$$

Obviously, P_i is simply the number of TIVs in the bin i divided by the total number of triangles. The distribution of TIVs in p2psim and Meridian data is depicted on Figure 3 and Figure 4.

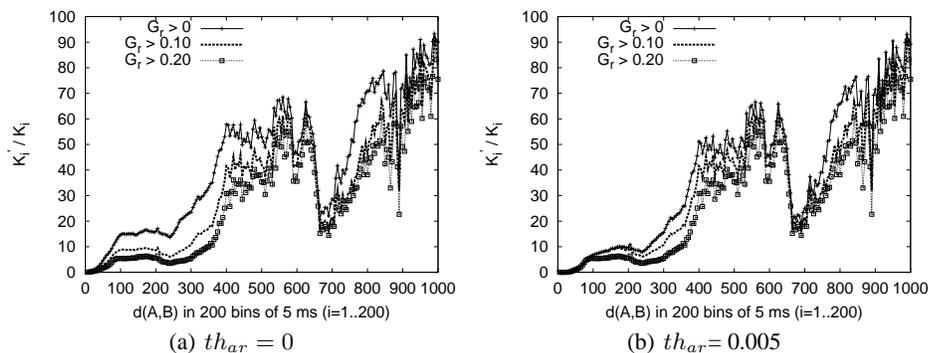


Fig. 2. Proportion of TIVs in each bin for various TIV severity levels for Meridian data set.

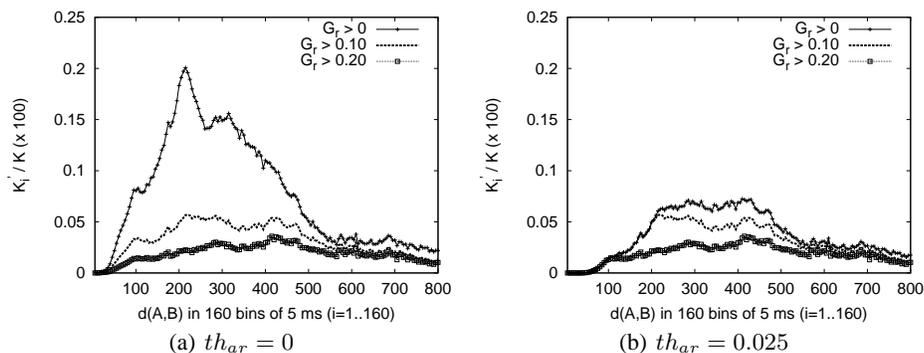


Fig. 3. Distribution of TIVs in p2psim data set for various severity levels.

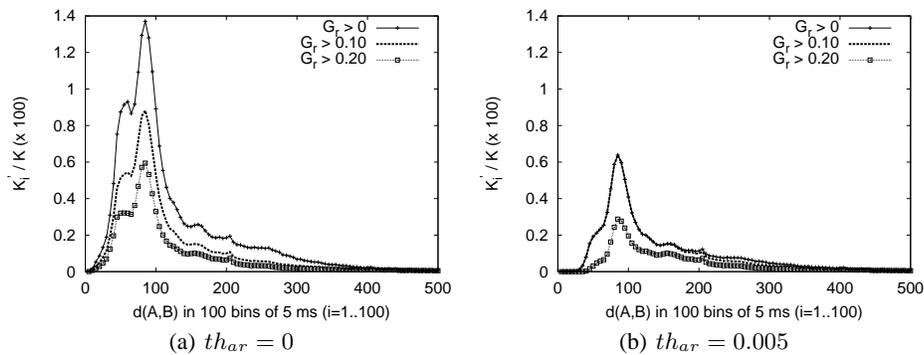


Fig. 4. Distribution of TIVs in Meridian data set for various severity levels.

As expected, Figure 3 is less conclusive than Figure 1, but it still shows that few severe TIVs are found in small triangles, that is below 100 *ms*. Similar behavior can

be observed in Figure 4 where the edges shorter than 60 *ms* cause slight violations. Moreover, the TIV severity of edges has an irregular relationship with their lengths. For instance, in Figure 4 the TIV severity has a peak for the edges around 80 – 100 *ms*.

This motivates our hierarchical approach to build a coordinate system. If we create clusters whose diameters do not exceed too much 100 *ms*, we may expect much fewer severe TIVs in each cluster, which is likely to improve the accuracy of intra-cluster coordinate systems. However, so far the impact of TIVs on the coordinates embedding remains hypothetical. In the next section, we will quantify this impact on one of the prominent P2P coordinate systems, namely Vivaldi.

3 Impact of TIVs on Vivaldi

Vivaldi [5], the focus of our present study, is based on a simulation of springs, where the position of the nodes that minimizes the potential energy of the spring also minimizes the embedding error. It is described in further details in the following section.

3.1 Vivaldi Overview

Vivaldi is fully distributed, requiring no fixed network infrastructure and no distinguished nodes. A new node computes its coordinate after collecting latency information from only a few other nodes. Basically, Vivaldi places a spring between each pair of nodes (i, j) with a rest length set to the known $RTT(i, j)$. An identical Vivaldi procedure runs on every node. Each sample provides information that allows a node to update its coordinate. The algorithm handles high error nodes by computing weights for each received sample. The sample used by each node, i is based on measurement to a node, j , its coordinates x_j and the estimated error reported by j , e_j . A relative error of this sample is then computed as follows:

$$e_s = | \| x_j - x_i \| - RTT(i, j)_{measured} | / RTT(i, j)_{measured}$$

The node then computes the sample weight balancing local and remote error : $w_i = e_i / (e_i + e_j)$, where e_i is the node's current (local) error, representing node i confidence in its own coordinate. This sample weight is used to update an adaptive timestep, δ_i defining the fraction of the way the node is allowed to move toward the perfect position for the current sample: $\delta_i = C_c \times w_i$, where C_c is a constant fraction < 1 . The node updates its local coordinates as the following:

$$x_i = x_i + \delta_i \cdot (RTT(i, j)_{measured} - \| x_i - x_j \|) \cdot u(x_i - x_j)$$

where $u(x_i - x_j)$ is a unit vector giving the direction of i 's displacement. Finally, it updates its local error as $e_i = e_s \times w_i + e_i \times (1 - w_i)$. The reader should note that after convergence of a Vivaldi system, the relative local error variation is of the order of a few percent (e.g. +/-0.05).

Vivaldi considers a few possible coordinate spaces that might better capture the underlying structure of the Internet. Coordinates embedding map into different geometric spaces, where nodes are computing their coordinates, e.g., 2D, 3D or 5D Euclidean spaces, spherical coordinates, etc.

3.2 Results

In this section, we present the results of an extensive simulation study of the Vivaldi system. For the simulation scenarios, we used the p2psim discrete-event simulator [10], which comes with an implementation of the Vivaldi system.

Unless otherwise stated, each Vivaldi node has 32 neighbors (i.e. is attached to 32 springs), half of which being chosen to be among the closest nodes. The constant fraction C_c for the adaptive timestep (see section 3.1) was set to 0.25. These values are those recommended in [5]. The system was considered stabilized when all relative errors converged to a value varying by at most 0.02 for 10 simulation ticks. We observed that Vivaldi always converged within 1800 simulation ticks, which represents a convergence time of over 8 hours (1 tick is roughly 17 seconds). Our results are obtained for a 2-dimensional euclidean space.

In order to observe the impact of TIVs on Vivaldi nodes, and in particular on the embedding performance, we could define the notion of TIVs involvement through different considerations. In this paper, we consider a node to be more or less involved into TIVs according to the number of times it belongs to a TIV. The more node C appears in bad triangles $A_iB_jC, i \neq j$, the more C is considered involved into TIVs situations.

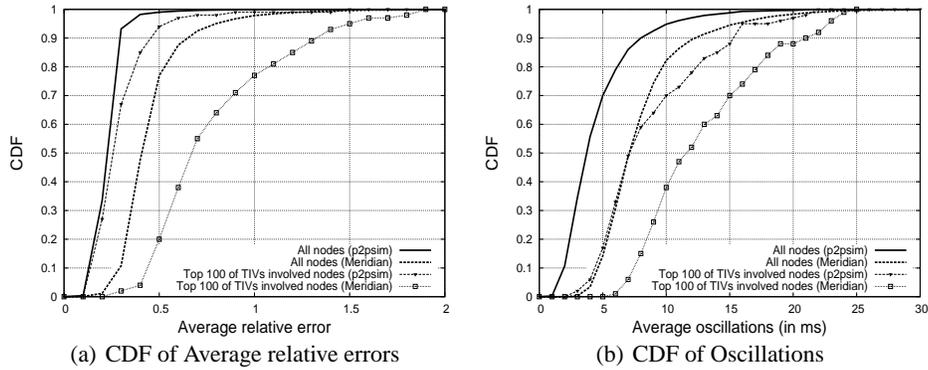


Fig. 5. Impact of TIV severity on the embedding for p2psim and Meridian data set.

In Figure 5(a), considering the p2psim and Meridian data set, we plot the Cumulative Distribution Function (CDF) of the average relative error (ARE) of the top 100 nodes involved in TIVs. We compare such distribution to the CDF of all nodes' relative errors in a flat Vivaldi system. Note that the ARE is computed for each node as the average prediction errors (along *all* the nodes) yielded by the Vivaldi system at the last tick of our simulations.

$$ARE_i = \frac{\sum_{j \neq i \in S} \frac{|(RTT(i,j) - \|x_i - x_j\|)|}{RTT(i,j)}}{|S| - 1}$$

where S is the set of all nodes in the system.

Figure 5(a) shows that, while for the distribution of errors on all the nodes of the system, more than 90% of p2psim nodes (resp. 70% Meridian nodes) have an *ARE* less than 0.3 (resp. 0.5), this percentage falls down to only 50% (resp. 20%) when considering the most involved nodes in TIVs. The coordinates computation at the level of these implicated nodes is spoiled out.

It is also worth observing the variation of coordinates in the Vivaldi system. In fact, even though the system converges in the sense that the relative errors at each node stabilizes, these errors could be so high that a great variation of the coordinate of a node barely affects the associated error. We can define such coordinates oscillation as the distance between any two consecutive coordinates. The average oscillations values are computed as the average of the oscillations during the last 500 ticks of our Vivaldi simulation. Figure 5(b) shows the CDF of these average oscillations, comparing again the distributions through all nodes and of the top 100 nodes involved in TIVs. We clearly see that the impact of TIVs can be considered as very serious with nodes involved in more TIV triangles seeing a large increase in their average oscillations values.

In light of these observations on the serious impact of TIVs on the coordinates embedding and based on our findings related to the distributions of TIVs in the Internet, the main intuition behind our proposal of a hierarchical structure of Vivaldi is to mitigate the impact of most severe TIVs. In this way, nodes would perform a more accurate embedding at least in restricted spaces (*i.e.* with small distance coverage).

4 Two-Tier Vivaldi

This section is divided into three parts. First, we present an overview of our Two-Tier architecture. Second, we define the clustering method we experimented with and, finally, we compare the results obtained with our Two-Tier Vivaldi to those obtained with a flat Vivaldi.

4.1 Overview

Recall from our section 2 that small triangles are less often (severe) TIVs. Any 3 edges with small RTTs (as observed in section 2) should be more adequate to construct a metric space without violating too much the Triangle Inequality laws. Put simply, shorter paths are more embeddable than longer paths that tend to create more severe TIVs with high absolute errors.

In this section, we exploit such property to deal with TIVs severity and their serious impact on network coordinates, by proposing a Two-Tier Vivaldi approach. The main idea is to divide the set of nodes into clusters and to run an independent Vivaldi in each cluster. Clusters are composed of a set of nodes within a given coverage distance.

Since Vivaldi instances running on each cluster are independent, nodes are collecting latency information from only a few other neighbors located within the same cluster. In this way, coordinates of nodes belonging to different clusters cannot be used to estimate the RTT between these nodes. We keep then running a Vivaldi system at a higher level (*i.e.* the whole set of nodes) in order to ensure that coordinates computed between any two nodes belonging to different clusters, 'make sense'. Coordinates computed

at the lower (resp. higher) level of clusters are called *local coordinates* (resp. *global coordinates*). Next, we describe how we set up the clusters we experimented with to illustrate our results on the Two-Tier Vivaldi structure.

4.2 Clustering method

We have first based our clusters recognition on the coordinates as observed by running a flat Vivaldi over the p2psim and Meridian data set. As already shown by Dabek et al. [5], 2-D coordinates lead to five major clusters of nodes for p2psim data. We took the three most populated as our main clusters. For Meridian data, 2-D coordinates lead to three major clusters with a set of nodes sparsely distributed. Nodes that have not been selected in any cluster would then be considered in only the higher level of the architecture, using only their global coordinates. However, this first step in our cluster selection is only based on the estimated RTTs (as predicted by the flat Vivaldi) and not on the actual RTTs between nodes.

In fact, according to their relative errors, some nodes can be misplaced by the flat Vivaldi and clusters could have their diameters larger than expected. Our second step has then consisted in a cross-checking of our preliminary clustering using the delay matrix. We proceed in a recursive way as follows: for any two nodes belonging to the same cluster, the cluster constraint is that the distance between these two nodes should be less than the cluster diameter. We then begin by verifying this cluster constraint, testing it on all the pairs of each cluster according to their actual distance provided by the delay matrix. Afterwards, we remove the top node that causes more violations of the constraint¹. We then recursively check the cluster constraint until no more pairs inside the cluster violate the cluster constraint. Following this clusters selection method, we obtained the clusters described in Figure 6 for p2psim and Meridian data set.

	Nodes	Diameter
Cluster 1	565	140 ms
Cluster 2	169	100 ms
Cluster 3	93	60 ms

(a) p2psim data

	Nodes	Diameter
Cluster 1	560	80 ms
Cluster 2	563	80 ms
Cluster 3	282	70 ms

(b) Meridian data

Fig. 6. Characteristics of the clusters

4.3 Performance evaluation of the Two-tier Vivaldi

First, we use the relative error as our main performance indicator. Again, we compute the *ARE* over all nodes to represent the accuracy of the overall system.

Figures 7(a) and 7(b) represent the CDFs of the relative error of nodes belonging to our three clusters 6(a). We clearly see that relative errors computed based on local

¹ This node is likely to have a high embedding error

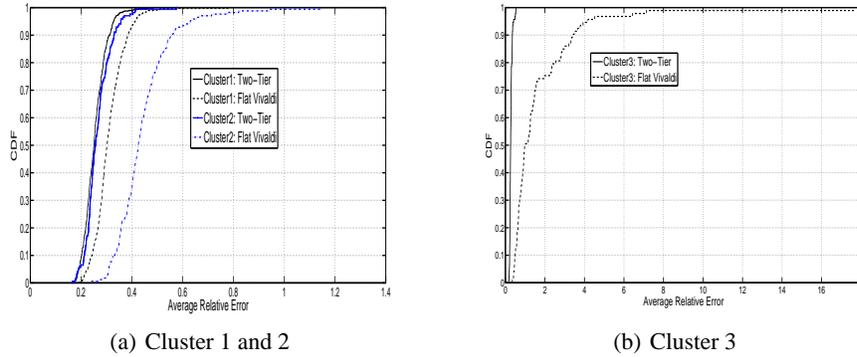


Fig. 7. Comparison of relative errors for p2psim data set: Flat versus Two-tier Vivaldi.

coordinates inside the clusters are much less than errors as computed using global coordinates (the Flat Vivaldi labeled curves). In cluster 2, for instance, more than 90% of nodes predicting their distances inside the cluster, achieve on average a relative error less than 0.3. When using the flat Vivaldi, over half of the population of the set of nodes in cluster 2, is computing coordinates with an *ARE* more than 0.5. Worse cases (for the flat Vivaldi system) are observed with respect to nodes in cluster 3, as depicted in Figure 7(b), where the flat Vivaldi system collapses with very high effective relative errors for more than 70% of the nodes. In the Two-Tier architecture, nodes are clearly performing much better. We observed the same trend for the Meridian data. By lack of space, we do not present the Meridian's figures here.

It is worth noticing here that cluster 3 is the smallest cluster in terms of Diameter. The observation of the embedding relative errors in this cluster confirm then our findings related to the effect of edges lengths on the TIVs severity and thus on their impact on the embedding. More generally, improvements inside these clusters is explained by the fact that intra cluster nodes, when computing their local coordinates select only close by nodes as their neighbors. This constraints the node-to-neighbors edges lengths and thus reduces the selection of severe TIVs likelihood. When encountering severe TIVs that cause high absolute errors, a node updates its coordinate, by jumping back and forth across its actual position. When limited to TIVs of low absolute severity, a node converges 'smoothly' towards an approximation of its correct position, then would stick to such position, and oscillate much less. In essence, it gains confidence in its local error faster (see 3.1) and performs more accurate embedding.

Limiting the neighborhood inside the cluster should then limit the high oscillations due to long and severe TIVs. In a second step, we then observed the coordinates' oscillation of nodes belonging to our three clusters. Again, we consider the average oscillations values as the average oscillations during the last 500 ticks of our simulations. We can observe that the three curves representing the CDF of the Two-Tier architecture nodes oscillations are those that are the highest in Figure 8(a) and Figure 8(b). This clearly shows that local coordinates of nodes inside our clusters oscillate with less amplitude. For instance, in Figure 8(a) more than 80% of the average oscillations are less

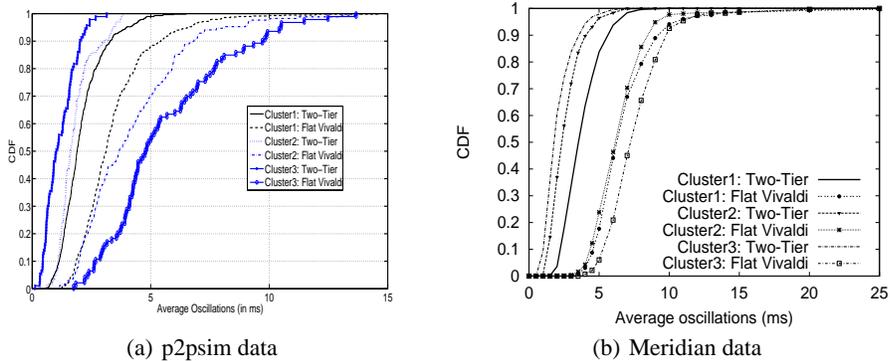


Fig. 8. CDF of coordinates oscillations for flat and Two-Tier Vivaldi.

than $3ms$ when nodes are computing local coordinates, whereas, only 40% of nodes in flat Vivaldi have, their oscillations less than this value. As can be seen, for Meridian data (Figure 8(b)), nodes oscillate over large ranges. More than 50% of nodes in flat Vivaldi have their oscillations superior to $5ms$.

5 Discussions and Conclusions

We have presented a Two-Tier architecture to mitigate the impact of potential Triangle Inequality Violations, which often occur in the Internet. We have shown that larger triangles are more likely (severe) TIVs, with respect to the distribution of RTTs in the Internet. Previous proposals of different coordinate systems focus on the geometric properties of the coordinate space and look for which space is the most convenient to embed RTTs. Our architecture does not rely on such approach, but it is instead based on clustering nodes to mitigate the impact of severe TIVs. Within their cluster, nodes use more accurate local coordinates to predict intra-cluster distances, and keep using global coordinates when predicting longer distances towards nodes belonging to foreign clusters. Knowing that coordinate systems are often used to characterize the set of close by neighbors in an overlay distribution or to select the closest download server, our approach thus succeeds where methods based on space dimensionality or properties would fail.

Although this paper focused on Vivaldi for measurements and experimentations, the Two-Tier architecture proposed is independent of the embedding protocol used. It is important to note that the deployment of our Two-Tier architecture does not equate to imposing any changes in the coordinate system process we use. Indeed, apart from running two different instantiations at the level of each cluster and at a higher level, our method does not entail any change to the operations of the embedding protocols. Our proposed method would then be general enough to be applied in the context of coordinates computed by other Internet coordinate system than Vivaldi.

Even though this paper does not address the problem of clustering techniques, and rather uses an oracle-based technique where coordinates and delay matrix are known, we note that different solutions to such issues have been proposed elsewhere (e.g. [15]).

Finally, we have also quantified the impact of TIVs on the embedding performance, considering that a node is involved in a TIV if it is within a bad triangle. However, it could also be argued that, in different embedding protocols, given a set of neighbors, the most involved node in TIVs is not necessarily affecting its non-neighbors measurements and coordinates. Other TIVs involvement definitions could be considered in order to refine the impact of TIVs on the embedding process. We can, for instance, consider that a node is involved in a TIV situation if it appears in a bad triangle *and* it considers the two other nodes of this bad triangle as its neighbors in the coordinate computation. Such new knowledge and our findings characterizing the TIVs severity could be leveraged to manage the neighbors selection in coordinate systems in order to alleviate the TIV severity at the higher-level of our Two-Tier architecture.

References

1. A. Rowstron and P. Drusche, "Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems," in *Proc. IFIP/ACM ICDS*, Heidelberg, Germany, 2001.
2. Y. h. Chu, S. G. Rao, and H. Zhang, "A case for end system multicast," in *Proc. the ACM SIGMETRICS*, Santa Clara, jun 2000.
3. S. Rewaskar and J. Kaur, "Testing the scalability of overlay routing infrastructures," in *Proc the PAM Conference*, Sophia Antipolis, France, April 2004.
4. T. S. E. Ng and H. Zhang, "Predicting Internet network distance with coordinates-based approaches," in *Proc. IEEE INFOCOM*, New York, NY, USA, June 2002.
5. F. Dabek, R. Cox, F. Kaashoek, and R. Morris, "Vivaldi: A decentralized network coordinate system," in *Proc. ACM SIGCOMM*, Portland, OR, USA, Aug. 2004.
6. H. Zheng, E. K. Lua, M. Pias, and T. Griffin, "Internet Routing Policies and Round-Trip-Times," in *Proc. the PAM Conference*, Boston, MA, USA, Apr. 2005.
7. J. Ledlie, P. Gardner, and M. I. Seltzer, "Network coordinates in the wild," in *Proc NSDI*, Cambridge, apr 2007.
8. M. Mendel Y. Bartal, N. Linial and A. Naor, "On metric ramsey-type phenomena," in *Proc. the Annual ACM Symposium on Theory of Computing (STOC)*, San Diego, CA, june 2003.
9. G. Wang, B. Zhang, and T. S. E. Ng, "Towards network triangle inequality violation aware distributed systems," in *Proc. the ACM/IMC Conference*, San Diego, CA, USA, oct 2007, pp. 175–188.
10. *A simulator for peer-to-peer protocols*, <http://www.pdos.lcs.mit.edu/p2psim/index.html>.
11. B. Wong, A. Slivkins, and E. Sirer, "Meridian: A lightweight network location service without virtual coordinates," in *Proc. the ACM SIGCOMM*, aug 2005.
12. K. P. Gummadi, S. Saroiu, and S. D. Gribble, "King: Estimating latency between arbitrary Internet end hosts," in *Proc the SIGCOMM Internet Measurement Workshop*, Marseille, France, Nov. 2002.
13. S. Savage, A. Collins, E. Hoffman, J. Snell, and T. Anderson, "The end-to-end effects of Internet path selection," in *Proc. of ACM SIGCOMM'99*, Cambridge, MA, USA, Sept. 1999.
14. S. Lee, Z. Zhang, S. Sahu, and D. Saha, "On suitability of euclidean embedding of internet hosts," *SIGMETRICS*, vol. 34, no. 1, pp. 157–168, 2006.
15. S. Min, J. Holliday, and D. Cho, "Optimal super-peer selection for large-scale p2p system," in *Proc. the ICHIT Conference*, Washington, DC, USA, nov 2006, pp. 588–593.