Reducing TIV Interference in Network Coordinate Systems

Zhuo Chen[1], Yibo Zhu[1], Yang Chen[2], Beixing Deng[1] and Xing Li[1]
[1] Department of Electronic Engineering, Tsinghua University, Beijing, China
[2] Institute of Computer Science, University of Goettingen, Goettingen, Germany
E-mail: {chenzhuo08,zyb07}@mails.tsinghua.edu.cn,yang.chen@cs.uni-goettingen.de

ABSTRACT

The network coordinate (NC) systems, which are scalable in estimating the latencies between network hosts, are significantly limited by the Triangle Inequality Violations (TIVs). Previously, two-layer *Vivaldi* systems like Pharos were proposed to remedy the impact of TIV. In our recent research, we built a new system, Tarantula, which deals with a larger portion of existing TIVs. Our simulation demonstrates that this new system achieves a higher estimation accuracy than the two-layer systems do.

1. INTRODUCTION

The NC system provides a mechanic for predicting latencies between two hosts with high accuracy and low overheads. Using this approach, we can estimate latencies between N^2 pairs of nodes in real network with only $O(N)$ measurements. So far, NC systems are widely used in many P2P services, such as multi-player online games and Azureus BitTorrent. However, as most existing systems like Vivaldi [3] are based on a Euclidean distance model, the Triangle Inequality Violations (TIVs) have been a major cause of inaccuracy in estimation. Therefore, several studies have been made to attenuate the impact of TIVs, among which is the two-layer NC systems like Pharos [2]. However, having categorized the TIVs into three categories [1], we find that the two-layer structure can only optimize the first category TIV. What's more, for every triangle with first category TIV, only one edge of it can be estimated more accurately.

In this paper, we first observe the proportion of three categories of TIVs in different data sets, and then propose a new system, Tarantula, by which two out of the three categories of TIVs are characterized accurately.

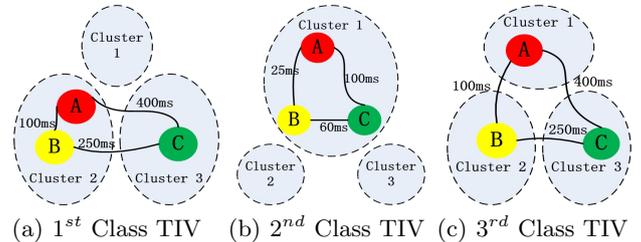(a) 1st Class TIV (b) 2nd Class TIV (c) 3rd Class TIV

Figure 1: Different Categories of TIVs

Table 1: Percentage of Different TIV Categories

Data sets	1st category	2nd category	3rd category
PlanetLab	70%	12%	18%
King	67%	16%	17%

Compared with the two-layer systems, Tarantula has proved to estimate latencies with more accuracy, especially on inter-cluster links.

2. MOTIVATION

In [1], when dividing the whole network into several clusters, the TIVs can be categorized into three categories (Figure 1): 1) two of the three nodes lie in the same cluster while the other one does not; 2) all of the three nodes belong to the same cluster 3) each node belongs to a different cluster.

We observe the percentage of different kinds of TIVs existing in several data sets when dividing the space into three clusters (table 1). The cluster dividing strategy used here is to let every host join the cluster with the closest anchor [1], which has proved to achieve the best performance according to our experiments so far. Table 1 shows that the first category TIV accounts for the majority of TIVs while the second and third categories of TIVs are not negligible. Using a two-layer structure, the intra-cluster link within a triangle of first category TIV can be estimated without interference of that TIV. The main idea of this design is to appropriately divide the whole space into some subsets. In a TIV triangle, If one or two of the three links lie in one subset while the
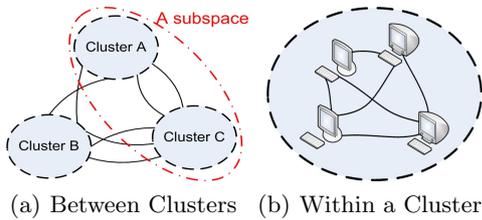
(a) Between Clusters   (b) Within a Cluster

Figure 2: Structure of Tarantula

other(s) do not, we can estimate the links within the subset without any interference by this TIV. However, for the inter-cluster links in a first category TIV and two other categories of TIVs, the two-layer systems have little advantage over original Vivaldi.

3. DESIGN

Our new structure, Tarantula, divides the whole space into k clusters, while every two of the k clusters constitute a subspace(Figure 2). Each node then belongs to $k-1$ subspaces. Moreover, in every subspace, we will not keep the intra-cluster links, as Figure 2(a) shows. For now, we set $k = 3$ so that the measuring cost is comparable with the two-layer systems.

We run a Vivaldi system on each cluster or subspace. We then estimate the intra-cluster latency using the Vivaldi system ran on that cluster, and predict the inter-cluster latency using the Vivaldi system ran on the subspace which consists of the two clusters. In this case, the links in a triangle with either first or third category TIV can be appropriately divided. For instance, A, B and C are the nodes in a triangle with first category TIV, while A and B are lying in the same cluster but C is not (Figure 1(a)). In Tarantula, the latencies between AC and BC are estimated using the coordinates of the subspace containing cluster 2 and cluster 3, which will not be affected by this TIV since link AB does not belong to this subspace. Moreover, the case of estimating delays of links in the third category TIV is similar. In conclusion, we have eliminated the impact of these two categories of TIVs, which, according to table 1, accounts for as much as 85% of total TIVs.

4. EVALUATION

Our experiments were carried out in the MATLAB simulation environment, while we measured the data set from the PlanetLab test bed consisting of 226 nodes.

We compared the accuracy of Tarantula with original Vivaldi and the two-layer Vivaldi system in terms of relative error (RE) [3]. Both Tarantula and two-layer systems have three clusters. The number of neigbors in a Vivaldi system within Vivaldi, two-layer Vivaldi, and Tarantula was set to 48, 24 and 16, respectively, so the overheads of maintaining neighbors are equal among all
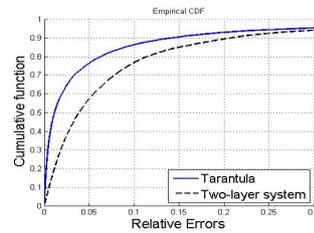

Figure 3: Inter-cluster REs

Table 2: Performance of Different Systems

	Tarantula	Two-layer system	Vivaldi
EPRE	0.119	0.137	0.166
Median RE	0.025	0.043	0.045

the three systems. The coordinate dimension was set to 7. Since our improvement mainly lay in the inter-cluster links—whether those belonging to a third category TIV or those inter-cluster edges within a first category TIV, we compared the estimation of latencies between inter-cluster nodes above all.

The REs of inter-cluster latencies of different systems are depicted in Figure 3 in the format of CDF. The figure shows that the eighty percentage relative error (EPRE) [4] of Tarantula is 44.8% smaller than the two-layer model. As it comes to overall estimation accuracy, Tarantula still greatly outperforms the other systems, since the intra-cluster estimation of our system is comparable with the two-layer system. Table 2 shows the overall performance of different systems using EPRE and median RE.

5. CONCLUSION

By dividing the whole network space into several subspaces, our new system, Tarantula can greatly reduce the negative impact of the first and third category TIVs, thus achieving a higher estimation accuracy. We will deploy this system in the real network for future work.

6. REFERENCES

[1] Y. Zhu, Y. Chen, Z. Zhang, X. Fu, et al. Taming the Triangle Inequality Violations with Network Coordinate System on Real Internet. To appear in Proc. of ACM ReArch, 2010.

[2] Y. Chen, Y. Xiong, X. Shi, J. Zhu, et al. Pharos: Accurate and Decentralised Network Coordinate System. IET Communications, 2009, 3(4):539-548.

[3] F. Dabek, R. Cox, and F. Kaashoek. Vivaldi: A Decentralized Network Coordinate System. In Proc. of ACM SIGCOMM, 2004.

[4] Y. Mao, L. Saul, and J. M. Smith. IDES: An Internet Distance Estimation Service for Large Network. IEEE Journal on Selected Areas in Communications (JSAC), 2006.