

ROUTE OPTIMIZATION BASED ON THE DETECTION OF TRIANGLE INEQUALITY VIOLATIONS

Papa Ousmane Sangharé and Bamba Gueye and Ibrahima Niang

Université Cheikh Anta Diop de Dakar, Senegal
papa.sanghare@ucad.edu.sn, bamba.gueye@ucad.edu.sn, iniang@ucad.sn

ABSTRACT

During the last decade, new services networks and distributed applications have emerged. These systems are flexible insofar as they can choose their ways of communication among so much of others. However, this choice of routing is based on a large number of measurements of times (Round Time Trip (RTT)) which are sources of overload in the network. Network Coordinate Systems (NCS) allow to reduce measurements overhead by mitigating direct measurements. However, NCS encounter inaccuracies with respect to distance prediction, when the measured distances violate the principle of the triangular inequality (TIV-Triangle Inequality Violation).

Firstly, we propose a new metric, called “RPMO”, which is based on the Ratio of Prediction and the Average Oscillations of the estimated distances, to detect the potential TIVs. The obtained results show that the “RPMO” metric gives better performance compared to metrics presented in former work. Secondly, we propose to use the existence of TIVs to optimize the routing in Overlay Network. To achieve this goal, we present a new approach that enables to detect the best shortened paths offered by the existence of potential TIVs.

Keywords— Network Coordinate Systems, Triangular Inequality Violation, Overlay Routing

1. INTRODUCTION

Nowadays, Network Coordinate Systems are widely used in network applications and services on a large scale and globally distributed applications such as file sharing peer to peer [1], nearest server selection [2], online games [3] etc.

Indeed, Network Coordinate Systems (NCS) [4, 5, 6, 7] allow hosts on the network to estimate the time between them without making measurements, and thus reduce resource consumption and particularly the number of measures on demand.

The main idea of NCS is to model the Internet as a geometric space, and characterize the position of each node in the network by a set of coordinates. Therefore, the latency between two nodes in the network is thus estimated as the geometric distance between their coordinates in this geometric space. Indeed, explicit measures are no longer needed.

Nevertheless, network policies routing [8] can break down the principle of triangle inequality. These violations are the cause of distortions and prediction errors for coordinate systems [9]. Let's assume three nodes A , B , and C such that $d(A, B)$ is $36ms$, $d(B, C)$ is $16ms$ and $d(A, C)$ is $9ms$, where $d(XY)$ denotes the delay between node X and node Y . In this case, the principle of triangle inequality is violated because $d(A, B) > d(A, C) + d(C, B)$.

In such case, the triangle ABC is a TIV (*Triangle Inequality Violation*) and AB (the longest side) is a **TIV-base**. TIV-base means that it exists a potential shortcut that can be used to reduce the distance between these two nodes that form this link considered as TIV-base. Since the principle of triangle inequality should be respected in any metric space, finding “good” coordinates in order to obtain an accurate estimation of delay between each pair of nodes will be impossible. In the presence of TIVs, node's coordinates will tend to alternate between sub-estimates and over-estimates the actual distance, without ever managing to position themselves in the metric space so perfect [9, 10, 11].

In order to exploit the coordinate systems for various operations of prediction distances it is mandatory that NCS give accurate and stable coordinates. Since the presence of TIV leads to inaccuracies with respect to the prediction delays, some researchers in this field have proposed TIVs detection techniques [10, 12] to allow nodes located in the systems to avoid links that are TIV-bases. In so doing, these nodes enhance their accuracy following the distance estimates.

However, the presence of TIVs in the Internet offers an opportunity that can be exploited to improve routing distribution applications online games, file sharing, or VoIP [3]. These applications can potentially improve their performance routing, using the shortcuts provided by TIVs [13].

Firstly, we introduce our TIV detection metric called *Ratio of Average Prediction on the oscillations* (“RPMO”), which detects TIVs accurately, while solving the shortcomings of previous works [12, 10].

Secondly, since TIVs are inherent to Internet, based on the their existence, we propose to optimize overlay network routing by using the *MDGD* metric (*Metric for Detecting Good Detours*) to detect the best shortcuts offered by TIVs.

The paper is organized as follows. Section 2 describes the different Network Coordinate Systems proposed in the related work ; in addition we present the previous metrics

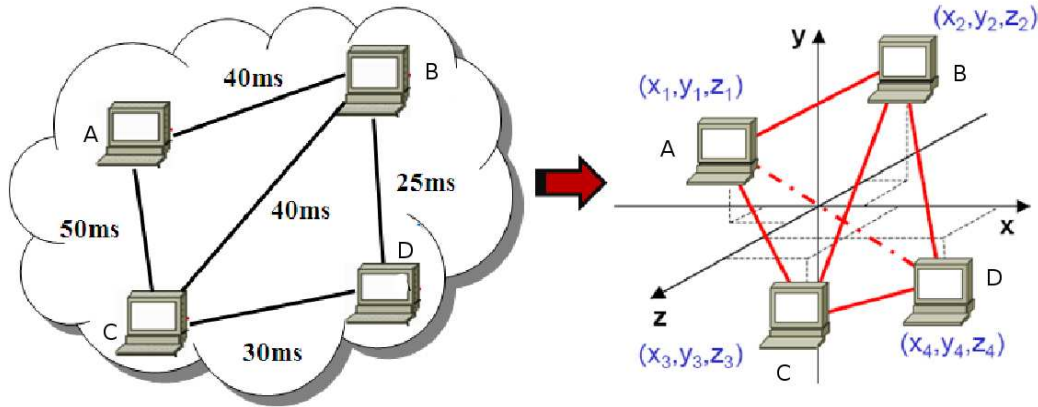


Figure 1. Geometric space model of the Internet.

used for detecting TIVs, i.e. the ratio of prediction [10] and OREE [12]. In section 3, we present and evaluate our proposed metric RPMO. In Section 4, we propose a new approach for optimizing Overlay routing by the use of triangle inequality violation in NCS. Finally, we conclude and present some research perspectives in Section 5.

2. RELATED WORK

In order to achieve the objectives of optimizing performance and scalability of network applications, several approaches for predicting network distances (propagation delay and transmission round trip “RTT”) based on coordinates have been proposed [6, 4, 5, 14, 7]. The main idea of such systems is to model the Internet as a geometric space. Consider the example shown in Figure 1, where we have four nodes (A, B, C, D) illustrated in a three-dimensional geometric space after an embedding from a given network.

Therefore, the distance between two nodes in the network is predicted as the distance between their coordinates, without making explicit measurements. In other words, if node A knows the coordinates (x, y, z) of node D , A will not need to make an explicit measure to determine the RTT towards D , instead, A computes the distance between itself and the node D in the coordinate space. The obtained distance represents the prediction of the RTT between A and D . It should be noted that until a precise location and reasonable for a node can be obtained with little overhead, much of the cost of distance measurements by sampling can be eliminated.

From the literature, Network Coordinate Systems can be splitted in two categories :

- *Centralized Coordinate Systems* : they involve a central component (a set of hosts called either Landmarks, or beacons, or Lighthouses) [6], from which other nodes calculate their own coordinates, according to the fixed infrastructure of measurements. We can give as examples *GNP* [4], *NPS* [5].
- *Distributed Coordinate Systems* : these systems general-

ize the role of landmarks in all nodes in the system, or eliminate the landmarks infrastructure. Decentralized Coordinate Systems can be seen as a peer-to-peer system. For instance we can cite *BBS* [14], *Vivaldi* [7].

2.1. Vivaldi overview

Vivaldi [7] is a decentralized coordinates system in which each node computes its own coordinates by making measurements with a small number of other nodes called its neighbors that are heterogeneous (half close and half away). Each time a node takes a measurement with one of his neighbors ; it compares the estimated delay measured by using their coordinates and modifies its position in space so as to move toward or away from its neighbor.

If a given node i wants to update its own coordinates towards a given neighbor j it needs a sample. This sample is formed by RTT_{ij} which is the RTT measured between i and j , neighbor’s coordinates x_j , and the confidence error e_j [7]. Let assume that $EST_{ij} = \|x_j - x_i\|$ represents the estimated RTT between nodes i and j based on their coordinates.

Following the algorithm proposed in [7], firstly node i computes the weight w of its sample. We have $w = \frac{e_i}{(e_i + e_j)}$. Afterwards, it uses this weight to update its local error $e_i = e_j \times w + e_i \times (1 - w)$ and then computes a value of $\delta = C_c \times w$ (with $0 < C_c < 1$). The goal of δ is to evaluate the amplitude of the displacement of the node. Finally, node i updates its coordinates as follows :

$$x_i = x_i + \delta \times (RTT_{ij} - EST_{ij}) \times u(x_i - x_j) \quad (1)$$

where $u(x_i - x_j)$ is a unit vector that indicates the direction of node i with respect to its replacement. For more details about equation 1 please refer to [7].

2.2. Metrics for detecting TIV

Previous works have proposed two metrics for detecting TIV. The first one is called “*Ratio of prediction*” [10] and the later

“OREE” [12]. The metric OREE is based on the oscillations of a given node and the relative error estimation, whereas the ratio of prediction represents the relationship between the estimated distance and the measured (actual) distance.

The authors of [10] have shown that the sides of the triangle that have a small ratio of prediction, *i.e.*, the narrowed sides according to the Euclidean space, tend to cause severe TIVs. However, the Ratio of prediction presents some issues with respect to the node’s neighbors update. Indeed, each node belonging to the network periodically chooses 32 other neighbors at random, it adds to its 32 neighbors already available. The 64 neighbors are sorted according to the value of their prediction ratio.

If the ratio of prediction of a link is very small, this implies that the link is probably underestimated due to the existence of severe TIVs [10]. Subsequently, the node removes from its list of neighbors, the 32 nodes with the smallest ratio of prediction. Quite often, these neighbors are those that are generally far from this given node.

After having removed these 32 neighbors, the given node keeps as neighbors the remaining 32 nodes (the nearest) as neighbors for the next iteration. This set of neighbors is not suitable for Vivaldi algorithm according to [7].

The metric OREE involves the variance of the estimated distances, the distance measured and the mean estimated distances. The authors of OREE [12] have shown that when OREE’s value is small the link can be considered as a TIV-base, and vice versa. This means that the probability that a link is a TIV-base increases when the value of OREE decreases.

The main drawback of OREE is that it uses a huge amount of information for detecting TIVs. In fact, we should keep node’s coordinates of previous rounds of measurement. Therefore, OREE is not scalable in large network such as Internet. It causes a considerable computing time, leading poor performance of peer-to-peer hosts that aim to determine the best path as quickly as possible (e.g. online applications gaming or VoIP [3]).

Note that, Kawahara *et al.* in [15] propose to find quality overlay routes between node pairs based on TIV optimization according to the latency and packet loss ratio metrics. It is worth noticing that they do not propose a mechanism to detect TIVs.

3. TIV’S DETECTION BASED ON RPMO METRIC

To overcome the limitation of previous works [12, 10], we propose in this section a new metric that allows us to take into account the ratio of prediction as well node’s oscillations in the network.

3.1. RPMO (Ratio of Prediction on Average Oscillations)

Our goal is to find a metric that allows us to detect TIVs accurately without altering the heterogeneous selection of neighbors according to Vivaldi, and using less computation over-

head. Our proposed metric, called RPMO, takes into account three parameters (the oscillations, the estimated distance and the actual distance) in order to detect if a link can be considered as a potential TIV-base.

$$RPMO = \frac{Estimated\ distance}{RTT} \times \frac{1}{Average\ oscillations} \quad (2)$$

By definition, a tick represents a round where a given node update its own coordinates once. An oscillation is the difference of estimated distances of two successive ticks.

For instance, let assume that d_1 is the estimated distance of AB during the first tick (tick 1), d_2 is the estimated distance of AB during the second tick (tick 2), and d_3 is the estimated distance of AB during the third tick (tick 3). Therefore, the average oscillations between these three rounds can be computed as follows :

$$Average\ oscillations = \frac{(|d_1 - d_2| + |d_2 - d_3|)}{2}$$

Therefore, the RPMO value is obtained by

$$RPMO = \frac{d_n}{RTT} \times \frac{(n-1)}{\sum_{i=1}^n |d_i - d_{i+1}|} \quad (3)$$

3.2. Experimental setup

To evaluate the RPMO metric, we used the *P2Psim* discrete-event simulator [16] which provides an implementation of Vivaldi. During our simulations, each Vivaldi node has 32 neighbors and the results are obtained for a 9-dimensional Euclidean space. The constant C_c is set at 0.25 as recommended in [7].

In order to evaluate the RPMO metric, we used three matrices delays as datasets : P2Psim King dataset (1740 nodes) [16], Meridian dataset (2500 nodes) [17] and the PlanetLab dataset (180 nodes) [18].

King and Meridian dataset are obtained following the King measurement technique [19] which is similar to ping in the sense that it estimates the latency between arbitrary end nodes using recursive DNS queries. The third matrix, which we call PlanetLab dataset, is a matrix delay constructed by performing ping measurements between 180 PlanetLab nodes [18] distributed around the world.

To study the characteristics of TIV, two criteria have been defined to indicate the severity of TIV : the *absolute severity* and the *relative severity*.

The absolute severity is computed as follows :

$$Ga = d(A, C) - (d(A, B) + d(B, C)) \quad (4)$$

The relative severity is obtained by

$$Gr = \frac{d(A, C) - (d(A, B) + d(B, C))}{d(A, B)} \quad (5)$$

These criteria reflect the potential gain that can be achieved by detecting the existing TIVs in the network. A gain equals

to $G_a = 10ms$ illustrates that instead of going through the direct path from A to B , going through the path via node C allows us to gain $10ms$.

However, a large G_a and G_r do not show only severe violations, but also a possible gain. In our work, we are interested in TIVs that meet both criteria, namely $G_a > 10ms$ and $G_r > 0.1$. Indeed, TIVs offering shortcuts that allow a gain less than $10ms$ are not very interesting.

3.3. Evaluation and Results

To study the performance of these different TIV detection metrics (RPMO, Prediction Ratio, and OREE), we take into account a comparison of their Receiver Operating Characteristic (ROC) curves. Therefore, we use the classical false/true positive/negative indicators. A *true positive* (TPR - True Positive Rate) is a TIV-base, which should therefore be suspected by the test. A *false positive* (FPR - False Positive Rate) is a non TIV-base that has been wrongly suspected by the test.

Figure 2 illustrates ROC curves obtained following different TIV detection metrics such as Ratio Prediction, OREE, and RPMO by considering the King dataset. It should be noted that the *Ratio-Pred* as depicted in Figure 2 refers to the ‘‘Ratio of Prediction’’ metric. Each point on the ROC curves (Figure 2) determines the TPR along the y-axis and the FPR along the x-axis obtained with a given detection threshold. During our simulations, we take different threshold values that range from 0.5 to 9 by step of 0.5. It should be noted that for a ROC curve, more the curve is near to the top left the corner of the graph, better is the detection.

The value 0.3 labelled in Figure 2 represents a given threshold value that gives better results among the different threshold values (0.5 to 9) that we used during our simulations. For instance, according to RPMO metric in Figure 2, a percentage detection of 59% of TIV-base with 17% of FPR corresponds to a given threshold value fixed to 0.3. This threshold value gives a better tradeoff.

In fact, the RPMO and Ratio-Pred metric have the same trend. For FPR smaller than 0.2, the RPMO metric outperforms the Ratio-Pred metric. Nevertheless, OREE is less efficient with respect to both metrics RPMO and Ratio-Pred.

Figure 3 shows the ROC curves obtained following Meridian dataset. The general trend one can observed, compared to Figure 2, is the fact that we have higher TPR detection with respect to same FPR (eg., 11%). According to RPMO metric, the threshold value that gives high TPR (88%) with low FPR (11%) is 0.3 (Figure 3).

The main reason is due to the fact that we have more links that are TIV-bases in Meridian dataset with respect to King and PlanetLab datasets. Following our three datasets, the computed values of links that are TIV-bases are estimated to 23%, 42%, and 9% for King, Meridian, and PlanetLab datasets respectively. We recall that a TIV-base is a link where it is possible to find a shortcut in the overall system.

Figure 4 also illustrates the ROC curves according to Planet-

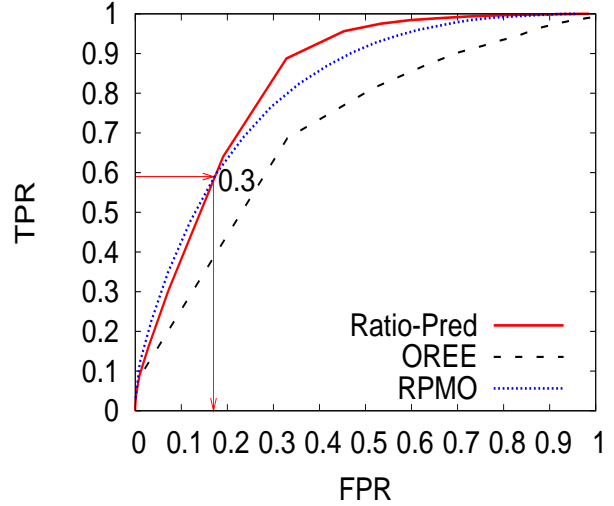


Figure 2. King dataset : Comparison between RPMO, Prediction Ratio and OREE metrics.

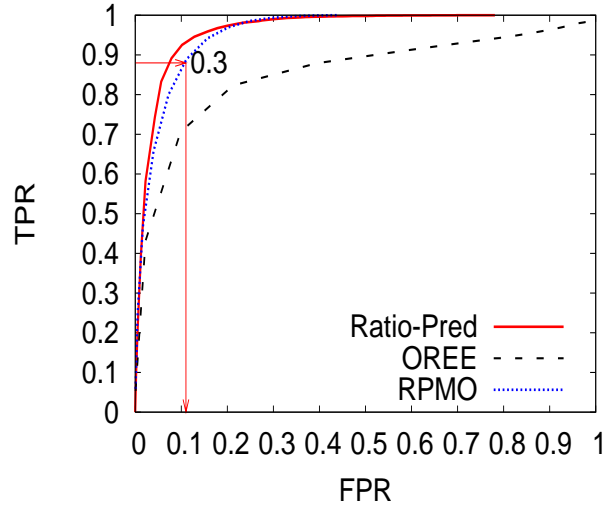


Figure 3. Meridian dataset : Comparison between RPMO, Prediction Ratio and OREE metrics.

Lab dataset with respect to our three studied metrics.

Following the PlanetLab dataset, the best threshold value with respect to RPMO metric is 0.65 with a TPR and a FPR equals to 55% and 22% respectively.

As summary, based on Figure 2 we remark that for a TPR values up to 60%, the RPMO metric is better compared to OREE and the ratio of prediction ; on the other hand for TPR values upper than 60% , the ratio of prediction becomes a little bit better than RPMO with a FPR upper than 30%. Following the Meridian dataset as illustrated on Figure 3, the gap is reduced between the ratio of prediction and RPMO. The same trend is also observed according to PlanetLab dataset (Figure 4).

It appears clearly that our TIV detection metric, called

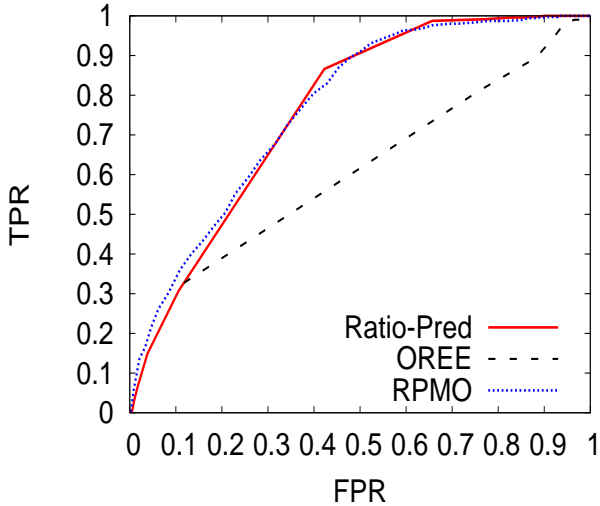


Figure 4. PlanetLab dataset : Comparison between RPMO, Prediction Ratio and OREE metrics.

RPMO, is more efficient compared to OREE metric by considering all datasets (see Figure 2, Figure 3 and Figure 4). It should be noted that with respect to the ratio of prediction metric the gap is reduced, and roughly we observe the same trend.

Nevertheless, the ratio of prediction presents several drawbacks according to the selection mechanism of node’s neighbors. The prediction ratio tends to select only nearest neighbors that is not suitable for Vivaldi algorithm [7, 6]. We argue that the metric RPMO is most suitable for detection TIV when we use a distributed coordinate system like Vivaldi.

4. OPTIMIZATION OF ROUTING IN THE OVERLAY NETWORK THROUGH TIVS DETECTION

As TIVs are inherent to Internet, they represent an opportunity that can be exploited for routing in overlay networks. In fact, multimedia, peer-to-peer file sharing, online games, distribution applications, or VoIP [3] require quality of service guarantees in term of delay. Therefore, these applications can potentially improve their performance by exploiting a TIV-based routing approach.

By definition, we recall that if the side AB of a “bad triangle” ABC (triangle where the triangle inequality is not respected) is a TIV-base, it exists a shortcut, for instance via node C , to get towards B from A instead of using the direct path (AB). In such case, applications can use the shortcut to gain more time. Our goal is to detect for each link TIV-base, for instance AB , the best C_i points that allow to gain more time from A towards B (i.e., $d(A, C_i) + d(C_i, B) < d(A, B)$).

4.1. Clustering approach

Clustering is a technique used to group elements with similar characteristics. Therefore, the idea is for each link TIV-base (eg., AB), to cluster potential nodes from a given diameter that can be considered as shortcuts with respect to the link TIV-base. In so doing, we reduce the number of shortcuts that will be evaluated in order to find the best one. As well, we can remove those shortcuts that are not clustered (*outliers*) in the set of shortcuts where we should seek good shortcuts.

To achieve this clustering, we used the “*QT_Clustering*” algorithm [20]. This algorithm has been initially proposed by Heyer et al. for genetic sequence clustering. It is based on the unique constraint of the cluster diameter, as a user-defined parameter. The cluster diameter represents the maximal distance existing among any two members of the cluster.

The main idea behind this clustering algorithm is to find the best shortcuts as well the shortcuts that are able to give the same gain of time. Indeed, these shortcuts should share the same cluster. We hope that these best shortcuts will be given by shortcuts that are not clustered (*outliers*) by the use of *QT_Clustering* algorithm.

For our simulations, we choose a diameter of $30ms$ for clustering the potential “good detour” (shortcut) obtained with respect to different links that are TIV-base. The result is shown in Figure 5.

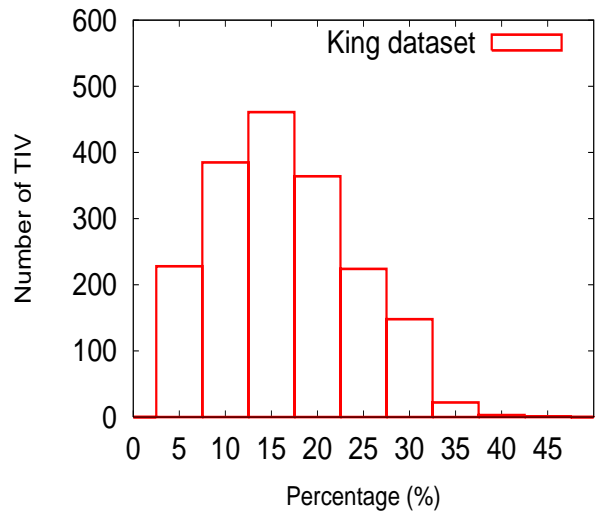


Figure 5. King dataset : Proportion of outliers (shortcuts) that represent the best shortcuts for a given TIV-base.

Figure 5 shows the percentage of shortcuts that are among the best shortcuts following a set of shortcuts obtained with respect to a given TIV-base. Note that, here the considered shortcuts are those that are not clustered by the *QT_Clustering* algorithm. The y-axis represents the number of TIV-base considered and the x-axis represents the percentage of shortcuts that are among the best shortcuts. It is worth noticing that the best shortcuts are those that enable to gain more time with respect to a given link considered as

TIV-base.

In Figure 5 we remark that the percentage of outliers that are among the best shortcuts varies between 5 and 45% with respect to all shortcuts for a given TIV-base. Note that a given box (Figure 5) can be seen as a bin where all TIV-bases give the possibility to find the same percentage of shortcuts that are among the best shortcuts. For instance, we can see that less than 10 TIV-bases have 35% of their shortcuts, considered as outliers, that are among the best shortcuts for each fixed TIV-base. Here, we recall that the considered shortcuts are those that do not belong to any cluster after having executed the QT_clustering algorithm.

The obtained results, by considering the shortcuts that are characterized as outliers, do not give high detection of best shortcuts.

Since we can find the center of each cluster, called *Cluster Head* (CH), we would like to seek if a cluster head can be considered as the best shortcut according to all other cluster heads that own the remaining clusters. In so doing, we should rank the different cluster head following the gain that they can offer as shortcut. The first one, after their ranking, is considered as the best cluster head. By definition, the “*Best Cluster*” is the one that is owned by the best cluster head. Put simply, we hope that all shortcuts that belong to this cluster offer a good shortcut.

With the QT_Clustering algorithm, each cluster has a cluster head that represents the center of the cluster. For a given TIV-base (eg., (AB)), we rank the different cluster heads with respect to the amount of time that they can offer. In other words we sort the different CH_i following their gain $1 \leq i \leq \text{number of cluster}$. Furthermore, we consider the members of the best cluster and seek their percentage among the best shortcuts that exist for this given TIV-base.

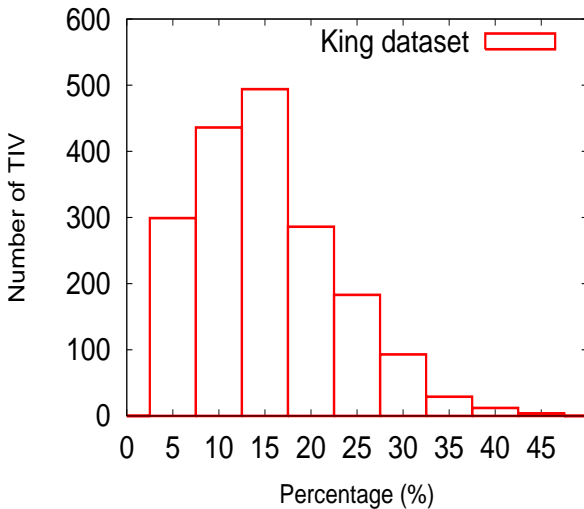


Figure 6. King dataset : Proportion of shortcuts that represent the best shortcuts following the “Best Cluster”.

The obtained results are shown on Figure 6. The y-axis represents the number of TIV-base considered and the x-axis rep-

resents the percentage of shortcuts that are among the best shortcuts with respect to the Best Cluster. This second approach, gives the same trend i.e. the percentage of shortcuts which are located in the Best Cluster and are considered as best shortcuts varies between 5 and 45%.

Based on the results illustrated in Figure 5 and Figure 6, we can conclude that the clustering approach does not allow a good detection of best shortcuts.

4.2. MDGD (Metric for Detecting Good Detours) approach

Since the previous method (clustering) does not help to find the best shortcuts, we focus on a new approach. The goal is to find a metric that enables to say whether any potential shortcut is part of the best shortcut (i.e, a shortcut with a gain of time upper than 10ms).

Therefore, we investigate the possible relation between the distance D' which is equal to $d(AB) - (d(AC) + d(CB))$, where $d(AB)$ represents the RTT between A and B . Note that D' is obtained based on actual distance (RTT delay). The *pseudo gain* for a triangle ABC represents the difference between the RTT distance ($d(AB)$) and the sum of estimated distances of links AC and CB , namely : $d(AB) - (Estimate(AC) + Estimate(CB))$.

We put the triangles in bin of 10ms based on their pseudo gain. In each bin, we calculate the minimum, the median, and the maximum distance of the distance D' of triangles present in the bin. We illustrate these three metrics in Figure 7, where on the x-axis we have the pseudo gain in milli second (ms) and on the y-axis the severity of the TIVs . The curve of the median distance D' of triangles shows that more and more that the pseudo gain increases, we are dealing with triangles TIV-bases (i.e a triangle that violates the principle of triangle inequality), which increasingly are becoming more severe (offering gains increasingly large).

It is worth noticing the negative values along the y-axis means that the triangle is not a TIV. In so doing, when we consider the minimum distance of D' , we can see that all triangles do not violate the principle of triangle inequality violation.

Furthermore, we consider these metrics in order to figure out our MDGD approach :

- The relative estimation error (Er) : $d(AB) / Estimate(AB)$
- The absolute estimation error (Ea) : $d(AB) - Estimate(AB)$
- Pseudo gain (PG) : $d(AB) - (Estimate(AC) + Estimate(CB))$

The pseudo gain can help us to find severe links that are TIV-base. Based on the metrics Er , Ea , and PG we propose the MDGD metric that allows to find the best shortcuts (gain upper than 10ms). The MDGD metric is described as follows :

$$MDGD = \frac{(Er \times Ea)}{PG} \quad (6)$$

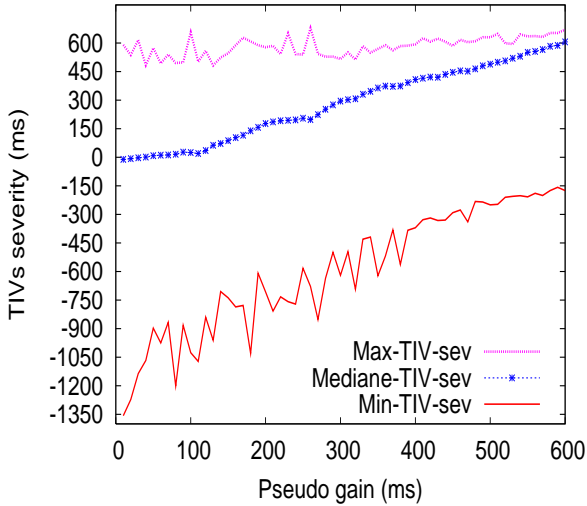


Figure 7. King dataset : Relationship between pseudo gain and TIVs severity.

4.3. Evaluation of MDGD approach

To study the effectiveness of this metric, our goal is to find the threshold value that allows to find the maximum number of “good shortcuts”. In such case, we rely on the TPR (True Positive Rate) and the FPR (False Positive Rate) according to each threshold value.

The TPR represents the percentage of shortcuts that are detected as well provide a gain of time upper than 10ms. The FPR represents the percentage of shortcuts that are wrongly detected as giving a shortcut upper than 10ms. Figure 8 illustrates the obtained results.

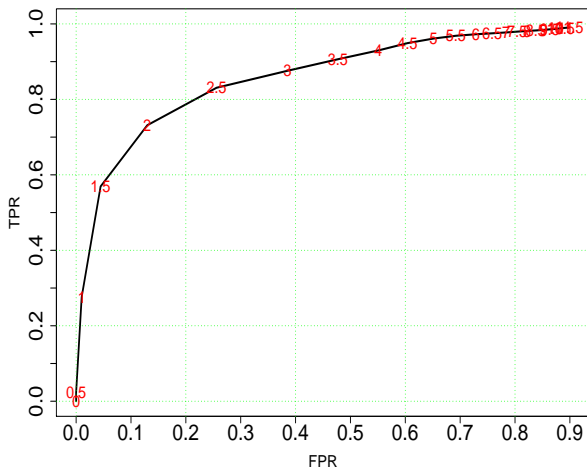


Figure 8. King dataset :ROC curve of the MDGD metric.

It should be noted that for a ROC curve, more the curve is near to the top left the corner of the graph, better is the detection. Based on Figure 8 we can notice that these following

thresholds (1.5, 2, 2.5, 3) exhibit this propriety.

To determine the best threshold that give us the best shortcuts with good accuracy, we compute the accuracy of the following thresholds 1.5, 2, 2.5, 3. By definition, the accuracy (ACC) (Table 1) represents the veracity of the classification and it is estimated as follows :

$$ACC = \frac{TP + TN}{P + N} \quad (7)$$

where TP and TN represents the number of true positive and true negative respectively. It is worth noticing that P and N represents the number of positive and negative respectively. Therefore, P expresses the total number of detours that give a gain upper than 10ms. In contrast, N represents the total number of detour that give a gain lower than 10ms.

Threshold	TPR	FPR	Accuracy (ACC)
1.5	0.57	0.04	0.68
2	0.73	0.13	0.77
2.5	0.83	0.26	0.81
3	0.88	0.38	0.80

Table 1. Evaluation of MDGD metric

Table 1 shows that a threshold value equals to 2.5 gives 83% of true positive whereas we have 26% of false positive. The threshold value equals to 2.5 gives the best accuracy.

It should be noted that it is very difficult to detect the best shortcuts with the use of clustering approach. The fact that potential shortcuts are clustered or are outliers, does not justify that they share the same characteristic. Nevertheless, with our MDGD approach, we could find the good nodes that offer shortcuts with a gain of time upper than 10ms, and with an accuracy of 81% (Table 1).

5. CONCLUSION

In this paper, we proposed a new metric called RPMO that enables Network Coordinate Systems to avoid the existence of TIV. We have shown than the RPMO outperforms OREE metric and presents the same trend with respect to the Prediction Ratio metric.

Although the TIVs are harmful to Network Coordinate Systems, they present opportunities to improve routing in overlay networks. In such case, the existence of TIV can lead to overlay networks that are TIV-aware. We can reduce consequently the delay between nodes by using the shortcut that TIVs can offer.

Therefore, we propose a metric called MDGD, to detect the best shortcuts of any triangle ABC that violate the principle of triangle inequality. The obtained results obtained show that with a threshold value equal to 2.5, MDGD, has a detection accuracy of 81%.

This result present a nice opportunity for peer-to-peer applications, online games, distributed applications, and VoIP

that require quality of service guarantees in terms of delay to maintain a certain level of performance.

Note that it is difficult to find a same RPMO's threshold value that can be applied in all studied datasets. As future work, we plan to find a metric that can enable to use a same threshold for all the used datasets. We plan also to investigate other clustering algorithms.

REFERENCES

- [1] Gnutella, "Gnutella, a distributed peer-to-peer data-sharing system," <http://www9.limewire.com/developer/gnutella/protocol/0.4.pdf>.
- [2] Sylvia Ratnasamy, Mark Handley, Richard M. Karp, and Scott Shenker, "Topologically-aware overlay construction and server selection," in *INFOCOM*, 2002.
- [3] Wookyun Kho, Salman Abdul Baset, and Henning Schulzrinne, "Skype relay calls : Measurements and experiments," in *in Proc. INFOCOM'08*, 2008.
- [4] T. S. E. Ng and Hui Zhang, "Predicting Internet network distance with coordinates-based approaches," in *Proceedings.Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies*. 2002, vol. 1, pp. 170–179, IEEE.
- [5] T. S. Eugene Ng and Hui Zhang, "A network positioning system for the internet," in *Proceedings of the annual conference on USENIX Annual Technical Conference*, Berkeley, CA, USA, 2004, ATEC '04, pp. 11–11, USENIX Association.
- [6] Benoit Donnet, Bamba Gueye, and Mohamed Ali Kaafar, "A survey on network coordinates systems, design, and security," *IEEE Communications Surveys and Tutorials*, vol. 12, no. 4, pp. 488–503, December 2010.
- [7] Frank Dabek, Russ Cox, Frans Kaashoek, and Robert Morris, "Vivaldi : a decentralized network coordinate system," *SIGCOMM Comput. Commun. Rev.*, vol. 34, pp. 15–26, August 2004.
- [8] Han Zheng, Eng Keong Lua, Marcelo Pias, and Timothy G. Griffin, "Internet routing policies and round-trip-times," in *In PAM*, 2005.
- [9] Mohamed Ali Kaafar, Bamba Gueye, Francois Cantin, Guy Leduc, and Laurent Mathy, "Towards a two-tier internet coordinate system to mitigate the impact of triangle inequality violations," in *Proceedings of the 7th international IFIP-TC6 Networking Conference, Lectures Notes in Computer Science 4982*, Singapore, May 2008, pp. 397–408.
- [10] Guohui Wang, Bo Zhang, and T. S. Eugene Ng, "Towards network triangle inequality violation aware distributed systems," in *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, New York, NY, USA, 2007, IMC '07, pp. 175–188, ACM.
- [11] Francois Cantin, Bamba Gueye, Mohamed Ali Kaafar, and Guy Leduc, "A self-organized clustering scheme for overlay networks," in *IWSOS, Lectures Notes in Computer Science 5343*, Dec. 2008, pp. 59–70.
- [12] Yongjun Liao, Mohamed Ali Kaafar, Bamba Gueye, Francois Cantin, Pierre Geurts, and Guy Leduc, "Detecting triangle inequality violations in internet coordinate systems by supervised learning," in *Proceedings of the 8th International IFIP-TC6 Networking Conference, Lectures Notes in Computer Science 5550*, Aachen, Germany, May 2009, pp. 352–363.
- [13] Cristian Lumezanu, Randy Baden, Dave Levin, Neil Spring, and Bobby Bhattacharjee, "Symbiotic relationships in internet routing overlays," in *Proceedings of the 6th USENIX symposium on Networked systems design and implementation*, Berkeley, CA, USA, 2009, pp. 467–480, USENIX Association.
- [14] Yuval Shavitt and Tomer Tankel, "Big-bang simulation for embedding network distances in euclidean space," *IEEE/ACM Trans. Netw.*, vol. 12, pp. 993–1006, December 2004.
- [15] Ryoichi Kawahara, Eng Keong Lua, Masato Uchida, Satoshi Kamei, and Hideaki Yoshino, "On the quality of triangle inequality violation aware routing overlay architecture," in *INFOCOM*, 2009, pp. 2761–2765.
- [16] P2Psim, "A simulator for peer-to-peer protocols," <http://www.pdos.lcs.mit.edu/p2psim/index.html>.
- [17] Bernard Wong, Aleksandrs Slivkins, and Emin Sirer, "Meridian : A lightweight network location service without virtual coordinates," *IN SIGCOMM*, pp. 85–96, 2005.
- [18] PlanetLab, "An open platform for developing, deploying, and accessing planetary-scale services," 2002, <http://www.planet-lab.org>.
- [19] Krishna P. Gummadi, Stefan Saroiu, and Steven D. Gribble, "King : Estimating latency between arbitrary internet end hosts," in *SIGCOMM Internet Measurement Workshop 2002*, 2002.
- [20] Laurie J. Heyer, Semyon Kruglyak, and Shibu Yooseph, "Exploring Expression Data : Identification and Analysis of Coexpressed Genes," *Genome Research*, vol. 9, no. 11, pp. 1106–1115, Nov. 1999.