Overlay Routing using Coordinate Systems

François Cantin*, Bamba Gueye, Dali Kaafar, Guy Leduc
University of Liège, Belgium
Research Unit in Networking (RUN)
{francois.cantin,cabgueye,ma.kaafar,guy.leduc}@ulg.ac.be

ABSTRACT

We address the problem of finding indirect overlay paths that reduce the latency between pairs of nodes in an overlay. To this end we propose to rely on an Internet Coordinate System (ICS), namely Vivaldi, to estimate RTTs and help find these interesting detours. We define two initial criteria to illustrate our approach and assess their true/false positive rates.

1. INTRODUCTION

Overlay routing is a well-know technique used to circumvent shortcomings of the underlying routing based on deployed interdomain and intradomain protocols. This technique has been used to provide multicast routing or QoS routing among others.

In this paper we explore the possibility of using an Internet Coordinate System (ICS), namely Vivaldi [1], to estimate RTTs in a scalable manner (i.e., without too much measurement overhead) and use this knowledge to improve overlay routing delays. This problem boils down to finding low latency paths in the overlay by using other overlay nodes as relays. More formally, given two overlay nodes A and B, our problem consists in finding a node C, such that the delay along path ACB is smaller than the delay along the direct path AB resulting from the underlay routing.

In section 2, we will first introduce briefly the concept of an ICS and the problem an ICS faces in the Internet due to the presence of Triangular Inequality Violations (TIVs). We will also explain that finding an interesting detour is equivalent to finding such a TIV. In section 3 we propose two criteria to narrow the search for TIVs and assess their performance on two datasets in section 4. We finally conclude and discuss further work in section 5.

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

ACM CoNEXT 2008 Student Workshop, December 9, 2008, Madrid, SPAIN

2. ICS AND TIV

Internet coordinate 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, such that the distance between two node coordinates gives an estimation of the delay between these nodes. For example, in the ICS called Vivaldi [1], each node computes its own coordinate by doing measurements with only a few (generally 32 or 64) other nodes chosen in the network (its *neighbors*).

Since they give an estimation for the RTT existing between any node pair AB in the network (even if it has never been measured), using an ICS seem to be interesting to find nodes C such that the RTT along the path ACB is smaller than the RTT along the path AB. However, the coordinates of an ICS cannot be used directly for this purpose. Indeed, it is well known [2] that Triangle Inequality Violations (TIVs) are a major problem for ICS because they can't be represented in the embedding space. Suppose we have a network with 3 nodes A, B and C, where $d(A,B) = 5$ ms, $d(A,C) = 2$ ms, and $d(C,B) = 1$ ms, with $d(X,Y)$ denoting the measured RTT between X and Y. The triangle inequality is violated because $d(A,C) + d(C,B) < d(A,B)$ and we say that the node pair AB is a *TIV base*. As TIVs are not representable in the embedding space, when faced with such TIVs, the ICS resolves the problem by forcing edges to shrink or to stretch in the embedding space.

The problem is that an interesting C node for a path AB is, by definition, such that ACB is a TIV. Consequently, it is impossible to detect interesting relays for a path AB by using only the estimations provided by an ICS. In this paper, we show the first detection results that we obtained by combining estimated RTTs and measured RTTs to detect the C nodes that are shortcuts for a node pair AB.

3. FINDING THE SHORTCUTS

Concerning the detection technique, our main constraint is that we want to find existing C nodes for a node pair AB without sending pings to every potential C node. So, we have only limited information: the estimated RTTs for all the node pairs and the measured RTTs for the subset of AB pairs connecting Vivaldi neighbors. In the sequel, we consider that the RTT of the node pair AB for which we are searching for C nodes is known, because we can always measure the RTT between A and B if needed.

A first possibility is to compare $RTT(A, B)$ to the estimated distance of the path AC_iB, where $1 \leq i \leq N-2$ and N denotes the number of nodes existing in the overlay. With this criterion, called *estimation* detection, if $EST(A, C_i) + EST(C_i, B) < RTT(A, B)$, where $EST(X, Y)$ denotes the estimated RTT between the nodes X and Y, then C_i is considered as a shortcut for AB. As estimated distances can be subject to estimation errors in spaces in which there are lots of TIVs, we also define a second method based on the measured distances between nodes.

Since each node knows the RTT to all its Vivaldi neighbors, for a node C_i we search among A's (resp. B's) neighbors the node, say C' (resp. C''), that is the closest to C_i by using the coordinates. If $RTT(A, C') + RTT(B, C'') < RTT(A, B)$ then C_i is considered as a shortcut. We called this method *approximation* detection.

4. EVALUATION

To test these detection criteria, we used the ICS called Vivaldi [1] simulated with the p2psim simulator. In our simulations, each node chooses 32 neighbors and computes its coordinates in a 9D euclidean space. We tested our detection techniques on two real data sets: the "*P2psim*" data set, which contains the measured RTTs between 1740 Internet DNS servers, and the "*Meridian*" data set, which contains the measured RTTs between 2500 nodes. Considering the P2psim data set (resp. Meridian), we found that 42% (resp. 83%) of node pairs are TIV bases.

To characterize the performance of our detection criteria, we use the classical false/true positive indicators. For a node pair AB, a positive (resp. negative) is a node C which is a shortcut (resp. which is not a shortcut). A true positive is a positive that has been correctly detected as a shortcut and a false positive is a negative that has been detected as a shortcut. Note that if a node C is a shortcut (resp. is not a shortcut) for k node pairs, it will be counted k times in the total number of positives (resp. negatives). The *false positive rate* (FPR) is the proportion of C nodes that have been wrongly reported as positives by the test and the *true positive rate* (TPR) is the proportion of C nodes that have been rightly reported as shortcuts by the test.

We applied our C node detection criteria to all node pairs of the two data sets. For the estimation detection criterion, with the P2psim (resp. Meridian) data set, we obtained a TPR of 84.6% (resp. 64.2%) and a FPR of 2.3% (resp. 10.2%); for the approximation detection criterion we obtained a TPR of 84% (resp. 74.4%) and a FPR of 9.2% (resp. 24.8%).

The estimation detection technique gives very good results on the P2psim dataset. These results are less satisfactory on the Meridian dataset. This is probably because this dataset contains lots of TIVs and, so, the estimated RTTs are less accurate. Moreover, with the Meridian data set the approximation detection technique gives a better TPR than the estimation detection method. However, with both datasets, the FPR obtained with the approximation detection criterion is really important compared to the estimation criterion. This is probably because we run Vivaldi with only 32 neighbors. So, there are many cases in which it is impossible to find C' and C'' near C_i. It could be interesting to run Vivaldi with more than 32 neighbors (for exemple 64) and see if we obtain better results with this criterion. Another solution could be to use a hybrid criterion: if it is impossible to find a C' (resp. C'') which is near C, we can switch back to the estimation detection criterion and use $EST(A, C)$ (resp. $EST(C, B)$) instead of $RTT(A, C')$ (resp. $RTT(C'', B)$).

5. CONCLUSION

The first results obtained with these two simple detection criteria are encouraging but we will try to obtain a higher TPR combined with a lower FPR by using more sophisticated techniques. However, searching for an existing C for a node pair AB could be quite costly: it is necessary to collect the coordinates of all the network's nodes and to compute all the estimated delays using these coordinates. This work must be done before searching for the C nodes. To avoid unnecessary work, we are currently working on a TIV base detector: we want to be able to detect the TIV bases by observing the behavior of the ICS. With such a tool, we will only search the existing C nodes for a node pair AB if this node pair is suspected to be a TIV base.

6. REFERENCES

[1] F. Dabek, R. Cox, F. Kaashoek, and R. Morris. Vivaldi: A decentralized network coordinate system. In *Proc. ACM SIGCOMM*, Portland, OR, USA, Aug. 2004.

[2] M. A. Kaafar, B. Gueye, F. Cantin, G. Leduc, and L. Mathy. Towards a two-tier internet coordinate system to mitigate the impact of triangle inequality violations. In *Proc. IFIP Networking Conference*, LNCS 4982, pages 397–408, Singapore, May 2008.