Embedding Networks using Hyperbolic Vivaldi Algorithm

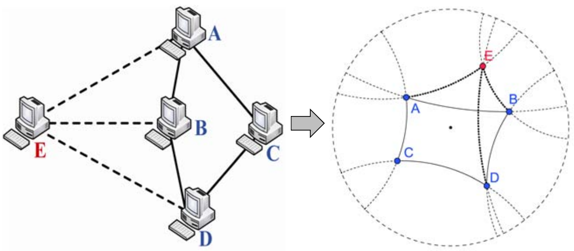
Kechao Cai, Guang Tan, Hongbo Jiang

Dept. of EIE, Huazhong University of Science and Technology, China

Introduction

Explicit measurements between every pair of nodes in the changing Internet environment are usually impractical. Although dynamic network performance properties such as bandwidth and link latencies can be accurately measured on demand, the huge number of widespread end-to-end paths that need to be considered in these distributed systems makes performing on-demand network measurements unrealizable because it is costly and time-consuming. More specifically, nodes must continuously discover the current state of network topology by exchanging information about the status of their connections to other nodes. Thus, the introduced communication overhead is regarded as one of the most serious scaling limitations of Internet today. Therefore, many researchers are finding ways to transform the all pair distance representation into a compact structure while preserving the distance close to the original ones.

Fig.1 Problem Statement



Many existing papers attempt to put the network into Euclidean spaces. However, it always results in unavoidable distortions because of the flatness of the Euclidean space. Our work is to embed the network nodes in a hyperbolic disk. We propose a hyperbolic version of the Vivaldi algorithm [1]. The algorithm helps build up a coordinate system, where nodes are mapped to the points in hyperbolic space without changing their relative distances in the real world. It allows a node to predict the round trip latencies to other nodes. The source node can compute the distances to the destination node with coordinate informed in hyperbolic space. (The whole procedure is illustrated in Fig.1.) That is, the hyperbolic distance between the source node and destination node is an accurate predictor of their reciprocal RRT latency. Thus, we get rid of plenty of explicit measurements and eliminate the communication overhead largely.

Methods

A. Generate Network. Considered the difficulties to attain the geographic locations and the network topologies of the real network, we generate the networks with a simple generative model in [2], instead. Without loss of generality, we show that the network generated share some common properties with the real network. The nodes are distributed uniformly along both radial r and angular direction θ of the hyperbolic disk with radius R. And links are formed with a simple connection probability function. Specifically, the nodes are assigned angular coordinates $\theta \in [0, 2\pi]$ with the uniform density $\rho(\theta) = 1/(2\pi)$. And the density for the radial coordinate $r \in [0, R]$ is exponential $\rho(r) = \sinh r/(\cosh R - 1)$. To form a network, every two nodes are connected if the hyperbolic distance d between them is less than R, $d = acos\,h((\cosh r_1 \cosh r_2 - \sinh r_1 \sinh r_2)\cos(\theta_1 - \theta_2))$.

Fig.2 Generated network

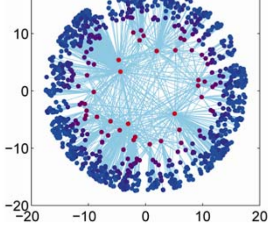


Fig.2 illustrates the generated network with a natural hierarchical structure. The color of the nodes goes from blue to red with the increase of node degrees. To examine whether the generated network reflects the real world network, we make a comparison on the power law property, which is an especially important universal property of the existing networks. The test results (omitted) show that the degree and clustering distribution are well in accord with the real networks.

B. Hyperbolic Vivaldi. Our hyperbolic Vivaldi algorithm is a simulation of the physical spring system in a hyperbolic disk. Every two nodes are connected with a hyperbolic spring. Specifically, every node maintains an estimation of its own current coordinates. Whenever two nodes communicate, they exchange their current coordinates and measure the RTT latency between them. While the computed geometric distances do not match with the latencies, the nodes starting at the origin on the hyperbolic disk should move step by step. At every step, each node will be pushed to a new position that minimizes the mismatching degree. The direction of the movement is to make the energy $E_{r\theta}$ of the spring system to the smallest. According to Hooke's law, the system energy can be expressed as:

$$E_{r\theta} = \frac{1}{2} k (RTT_{rr'} - d_h(r, r'))^2$$

The moving direction and the magnitude are the two main factors that affect the coordinates of the nodes. To reach the equilibrium position, the movement of a node should be along the opposite direction of the gradient of the energy function and the magnitude of the movement is relevant to the mismatching degree. The coordinates of the nodes are updated according to the following rule at every iteration.

$$new\_pos = cur\_pos + step \times magnitude \times direction$$

cur_pos represents the current position of the node while *new_pos* is the newly calculated position. With the gradient operator applied on the energy equation $E_{r\theta}$, we derive the formulas of direction and magnitude of movements on the hyperbolic disk. The direction of the movement is:

$$direction = u\left(\frac{\partial d_h(r, r')}{\partial r}\hat{r} + \frac{1}{r}\frac{\partial dh(r, r')}{\partial \theta}\hat{\theta}\right)$$

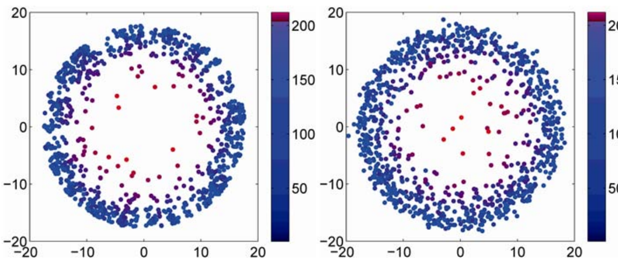
where $\hat{r}$ and $\hat{\theta}$ are the unit vectors in the polar coordinate system. u is the normalized unit vector. And the magnitude of the movement is:

$$magnitude = \frac{RTT_{rr'} - d_h(r, r')}{\sqrt{(\cosh(d_h(r, r')))^2 - 1}}$$

Results

The evaluation is performed on a generated network with 1000 nodes. The nodes are "uniformly" distributed on the hyperbolic disk with radius R=17.6. Fig.3 (a) shows the original node distribution of the generated network. Then, we select some pair distances, a reduced representation of the complete distance matrix, as the input of the hyperbolic vivaldi algorithm. All of the nodes are set to the origin on the hyperbolic disk before being processed. After several steps of moving, every node can find a position on the hyperbolic disk. We find that it needs about 200 iterative times of running for the algorithm to reach the minimum system energy. Fig.3 (b) gives the resulted node distribution of the embedded network. We see the higher degree nodes remain in the center of the disk and the lower degree node are distributed on the border of the disk. The embedded node distribution is similar to the original node distribution, which could imply a very high embedding accuracy.

Fig.3 Comparison of node distribution



(a)Original node distribution **(b)Embedded node distribution**

Conclusions

We develop a decentralized hyperbolic vivaldi algorithm which can be used to embed the generated network nodes into a hyperbolic disk. And the embedding gives a relative high accuracy with the original pair distances well preserved. Large amount of communication overhead are reduced as nodes can predict the distances in this framework with computation instead of measurements. To be more persuasive, we will embed the real network topology into the hyperbolic space. And we will make a more careful observation of the properties of the Internet topology. In our future work, we will try to reduce the routing complexity of the Internet with greedy routing mechanism. Our purpose is to design a greedy routing algorithm with high success rate and low stretch.

Bibliography

1. **F. Dabek, R. Cox, F. Kaashoek, and R. Morris. "Vivaldi: a decentralized network coordinate system". In *ACM SIGCOMM, 2004*.**
2. **F. Papadopoulos, D. Krioukov, M. Boguñá and A. Vahdat, "Greedy forwarding in dynamic scale-free networks embedded in hyperbolic metric spaces", In *IEEE INFOCOM 2010*.**