AS Path Inference: From Complex Network Perspective

Narisu Tao, Xu Chen and Xiaoming Fu,
Institute of Computer Science
University of Göttingen, Göttingen, Germany
Email: {narisu.tao,xu.chen,fu}@informatik.uni-goettingen.de

***Abstract*—AS-level end-to-end paths are of great value for ISPs and a variety of network applications. Although tools like traceroute may reveal AS paths, they require the permission to access source hosts and introduce additional probing traffic, which is not feasible in many applications. In contrast, AS path inference based on BGP control plane data and AS relationship information is a more practical and cost-effective approach. However, this approach suffers from a limited accuracy and high traffic, especially when AS paths are long.**

In this paper, we bring a new angle to the AS path inference problem by exploiting the metrical tree-likeness or low hyperbolicity of the Internet, part of the complex network properties of the Internet. We show that such property can generate a new constraint that narrows down the searching space of possible AS paths to a much smaller size. Based on this observation, we propose two new AS path inference algorithms, namely HyperPath and Valley-free HyperPath. With intensive evaluations on AS paths from real-world BGP Routing Information Bases, we show that the proposed new algorithms can achieve superior performance, in particular, when AS paths are long paths. We demonstrate that our algorithms can significantly reduce inter-AS traffic for P2P applications with an improved AS path prediction accuracy.

I. Introduction

As a network of networks, the Internet infrastructure consists of tens of thousands of networks or Autonomous Systems (ASes). Each AS, as a part of the Internet, is owned and administered by the same organization and adheres to a single and clearly defined routing policy. AS Number (ASN) is a globally unique identifier for every AS [1]. AS path is a series of ASNs, representing the route taken by data packets sent from one AS to a certain network and originally exchanged by neighboring ASes to avoid loops in inter-domain routing.

The knowledge of the actual AS path between arbitrary pairs of end hosts directly reflects the topological property of the connection. Therefore it is essential for network operators and researchers to detect and diagnose problems, study routing protocol behavior, characterize end-to-end paths through the Internet and optimize network performance [2]. Moreover, many network applications can benefit from being aware of AS paths. For example, it has been shown that most bottleneck links are more likely to appear in the access network or on the links between ISPs, rather than in the backbones of the ISPs [3]. Therefore, preferring the peers or servers with a shorter AS path can reduce chances of having bottlenecks in the path and, in turn, improve performance of applications (e.g., P2P), reduce the inter-domain traffic and lower cost for ISPs. With this motivation, J. Li and K. Sollins have proposed a structured P2P network, in which AS hop counts are used to filter out unlikely candidates [4]. This proposed system significantly reduces network traffic while maintaining fast lookups. As another example, AS path information has been leveraged for improving QoS of the VoIP service (e.g., Skype) [5]. In addition, AS path information has also been used for network delay estimation [6], cache deployment in Content Delivery Networks (CDNs) [7] and assessment of Internet routing resilience to failures and attacks [8], [9].

ISBN 978-3-901882-68-5

Although AS paths are of great value for many network applications, how to obtain such information is still a challenging issue. Collecting the BGP routing tables directly is impractical, since the number of ASes that support public direct access is very limited. To the best of our knowledge, only hundreds (out of totally around 47,000) ASes on the Internet can support remote access and routing information viewing [10]–[13]. Another way to obtain AS paths is active probing (e.g.,traceroute, iPlane [14] and iPlane Nano [15]). However, besides the direct access requirement, these active probing approaches have to deal with other issues, such as mapping between IP address to ASN, blocking from ISPs and additional overload to the infrastructure. A more practically-relevant and cost-effective approach is to estimate the AS paths by inference techniques based on BGP control plane data and AS relationship information [2], [16]. However, traditional inference-based approaches suffer from limited accuracy, especially when AS paths are long.

In this paper, we study the AS path inference problem from a complex network's point of view. In particular, we focus on exploring a key and intrinsic geometrical characteristic of complex networks, namely hyperbolicity or metrical tree-likeness. Roughly speaking, hyperbolicity measures the extent to which a graph resembles a tree from the metric's point of view. The key rationale for considering hyperbolicity for the AS path inference problem is that an AS system can be regarded as a complex network (i.e., a network of networks) and many complex networks (e.g., web graphs, collaboration networks, social networks and biological networks) have been empirically shown to have a low hyperbolicity or be metrically tree-like.

In this paper, we leverage the property of hyperbolicity to design an efficient AS path inference scheme. To this end, we

address the following main challenges:

- AS path inference problem is complicated by the fact that information collected from the current routing system is highly incomplete [17].
- Hyperbolicity is only studied under the shortest path distance metric of graph models of communication networks [18]–[21]. However, due to the policy-based inter-domain routing, actual AS path is not necessarily the shortest path and usually longer than the shortest path [22]. With the actual AS path hop count as the distance function, whether the AS-level Internet still exhibits metrical tree-likeness and to which extent it follows remain open questions.
- If the actual AS paths respect the underlying geometry of the Internet, how can we leverage this fact to improve AS path inference technique?

To tackle the above-mentioned challenges, we first conduct intensive empirical study with AS paths extracted from BGP control plane data to understand the extent to which actual AS paths exhibit metrical tree-likeness. Then we propose HyperPath and Valley-free HyperPath, two novel AS path inference algorithms which consider the impact of underlying geometric structure on the actual AS paths. To show the performance of the new methods, we implement two state-of-the-art benchmark methods, namely AS relationships based inference method [2] and KnownPath method [16], and compare them with the new algorithms. Experiments with ground truth AS paths show that our methods can be highly competitive when AS path is short and achieve significant performance gain when AS path is long with much less computation time and information. Moreover, while the benchmark techniques based on valley-free property frequently fail to work when actual AS paths are with 6 hops or more, the new inference algorithms can still achieve impressive prediction accuracy. We also show that the improvement of AS path prediction accuracy by our methods can reduce inter-AS traffic on BitTorrent network [23].

The remainder of the paper is organized as follows. In section II, we introduce related work. In section III, we introduce the concept of δ-hyperbolicity of graphs and illustrate with synthetic network models. In section IV, we conduct empirical study to understand the impact of underlying geometry of the Internet on AS path and propose new algorithms. In section V we evaluate the new methods and conclude the paper in section VI.

II. RELATED WORK

Active probing from an end host in a source AS can reveal the AS path. By running traceroute, a series of IP addresses of router interfaces would be obtained. Mapping these IP addresses into ASNs can give us a raw AS path. After removing the repeatedly occurring ASNs, the AS path can be finally generated. Although active probing can deliver accurate AS path, it can be problematic in practice. Firstly, traceroute can be blocked by ISPs for security consideration. Secondly, the mapping from IP address to AS number is not always accurate. Thirdly, it introduces additional measurement overhead into the infrastructure. Finally, the biggest problem is that it requires direct access to the end host in the source AS, which is usually hard to achieve.

To deal with the lack to direct access of active probing method, a plethora of different techniques have been proposed for inferring AS path. The most straightforward way is to run the shortest path algorithm, such as Dijkstra's algorithm, on the AS topology generated from BGP routing information as an approximation [4]. However, due to the inflation of AS paths, this method cannot provide high accuracy [22].

Later, AS relationships were introduced to design better AS path inference methods [2], [16], [24]. Specifically, AS relationships between two connected ASes can be classified into the following three types: customer to provider (c2p), peer to peer (p2p) or sibling to sibling (s2s). In a c2p relationship, the customer pays the provider to obtain transit service through the provider's network, while, in a p2p relationship, it is assumed that two peering ASes share the deployment and maintenance cost for the connecting link. Siblings are peering ASes that generally have a mutual transit agreement, i.e., merging ISPs.

Gao et al. [25] pointed out that patterns of AS path should follow, the so called, valley-free property. The valley-free property stems from the fact that ASes don't want to be used as a transit. Gao et al. characterize a path as downhill (uphill) if it only contains p2c or s2s links (c2p or s2s links) and any valid (valley-free) path must match one of the following patterns [25]:

- An uphill path;
- A downhill path;
- An uphill segment followed by a downhill segment;
- An uphill segment followed by a p2p link;
- A p2p link followed by a downhill segment;
- An uphill segment followed by a p2p link, followed by a downhill segment.

Mao et al. proposed one of the first methods to infer arbitrary AS path from the BGP routing tables [2]. Their method filter out the AS paths violating the valley-free property and choose the shortest AS path from the remaining AS paths. Later, Qiu and Gao [16] proposed the KnownPath algorithm for AS path inference. The key idea of the KnownPath method is to exploit AS paths that have appeared in BGP routing tables. This method has been cited by many recent research papers as one of the state-of-the-art method for AS path inference based on BGP control plane data [6], [26], [27]. One weak point of inference based on AS relationships information is that AS relationships can contain errors. In fact, inference about AS relationships itself is an active research problem [25], [28], [29].

In this paper, we enrich the AS path inference techniques by introducing a new kind of constraint or filtering mechanism, which is similar to the role the valley-free property plays. The new constraint mainly narrows down the candidate AS path set to a much smaller size. Without this filtering of possible AS path sets, originally, we have to go through every possible path

connecting source AS and destination prefix, which can be $O(|V|^2)$ in AS relationships based inference method, where V is the number of ASes in an AS topology. Therefore, the new constraint can enable speed-up in inference time. In addition, even though the new methods infer with much less information input, they can be highly competitive and even outperform the benchmark methods. Finally new inference methods can be implemented in a distributed manner, which is not easy for the state-of-the-art methods.

The hyperbolic space has been used for distance embedding and greedy routing for communication networks [18]–[21]. These methods are based on a given topology and a shortest path length distance function. The main difference between our methods and the existing studies is that we investigate and leverage the hyperbolicity of a metric space where the distance function is the actual AS hop count, rather than the shortest path distance.

III. δ-HYPERBOLICITY: TREE-LIKENESS FROM METRIC POINT OF VIEW

To facilitate discussions, in this part, we first give a brief introduction to the definition of hyperbolicity.

A. Definition

The notion of δ-hyperbolicity comes from the field of geometric group theory and the geometry of negatively curved metric spaces [30], [31]. Intuitively speaking, hyperbolicity of a graph/network can be viewed as a measure of how close a graph is to a tree from a metric point of view.

There are two definitions of hyperbolicity, which are equivalent to each other up to a multiplicative constant. In this paper, we use the 4-point δ-hyperbolicity definition by Gromov [30].

Definition 1. *[30] Let $\delta \geq 0$. A metric space (X, d), where X is the set of points and d is the distance measure, is called δ-hyperbolic if and only if given quadruplet $x,y,u,v \in X$ satisfying that $d(x,y)+d(u,v) \geq d(x,u)+d(y,v) \geq d(x,v)+d(y,u)$, the following condition holds:*

$$(d(x,y) + d(u,v)) - (d(x,u) + d(y,v)) \leq 2 * \delta \qquad (1)$$

For a graph $G = (V, E)$, we can regard it as a metric space where $X = V$ and d is the graph distance (e.g., shortest path distance) between two vertices u and v in the graph G. Then, the hyperbolicity δ of the graph G is typically defined as the minimum value of δ and the metric space (V, d) based on graph G is δ-hyperbolic.

A key property of hyperbolicity is that it can characterize the metrical tree-likeness of a graph. Generally, the lower the hyperbolicity of a graph is, the more likely it is metrically to a tree. For example, trees are exactly 0-hyperbolic. A cycle of length $2k$ is $\frac{k}{2}$-hyperbolic, which is the largest hyperbolicity a finite graph with $2k$ vertexes can have. It has been empirically shown that many real-world graphs/networks, such as collaborative graphs, email networks, biological networks, web graphs, p2p networks and social networks, have low hyperbolicity [32], [33].

B. Low hyperbolicity of scale-free networks

In this section, we will demonstrate the metrical tree-likeness of scale-free networks by using the measure of hyperbolicity through numerical evaluation. Note that this is useful for our study later since the AS topology is also scale-free as shown in section IV. The scale-free networks are generated according to the H^2 model [34], which is one of the latest scale-free network generation models. The generated networks by the H^2 model exhibit many similar properties of real-world complex networks. The H^2 model requires input parameters, such as the node number (N), the average node degree (d), the exponent of the power law distribution of the node degrees (γ) and the temperature (T).

We generate three synthetic scale-free networks (i.e., S_1, S_2, S_3) according to the parameters in Table I. Note that we choose γ as 2.1, which is the exponent of the power distribution of node degrees in the AS topology observed in our numerical study.

To gain more useful insight of the network structures, we compute the δ-hyperbolicity value distribution of the Largest Connected Component (LCC) of each network. $O(N^4)$ number of quadruplets has to be exhaustively iterated to obtain a complete δ-hyperbolicity value distribution. It is computationally prohibitive to obtain such a complete distribution when networks are of tens of thousands of nodes. Therefore, we randomly sample 100 million quadruplets to approximate the distribution when a network has more than one thousand nodes.

TABLE I: Synthetic scale-free networks.

ID	$\|V\|$	$E(d)$	γ	T	$\|V\|$ in LCC	$\|E\|$ in LCC
S_1	100	60	2.1	0	100	293
S_2	1,000	25	2.1	0	875	3,435
S_3	10,000	30	2.1	0.8	8,952	31,058

TABLE II: Distribution of the δ-hyperbolicity value of quadruplets in different graphs

δ \ ID	(S_1, d)	(S_2, d)	(S_3, d)	(T, d_2)
0	0.838	0.932	0.724	0.460
0.5	0.162	0.068	0.275	0.430
1	1.64E-06	1.88E-06	0.002	0.093
1.5	-	-	2.20E-07	0.015
2	-	-	-	0.002
2.5	-	-	-	1.41E-04
3	-	-	-	1.75E-05
3.5	-	-	-	6.73E-07
4	-	-	-	5.34E-09
$\% \leq 1$	1.000	1.000	0.999	0.983

Table II shows that scale-free networks (i.e., (S_1, d), (S_2, d), (S_3, d) in Table II) are metrical tree-like and almost every quadruplet has a δ-hyperbolicity smaller than or equal to one.

IV. HYPERPATH METHOD FOR AS PATH INFERENCE

As mentioned in section III-A, the graph hyperbolicity is typically defined under the shortest path distance metric. But due to AS path inflation [22], the actual AS path is usually

not the shortest one. In this case, whether the space (T, d_2), where T is ASes set and d_2 is actual AS hop count distance, is hyperbolic or not is not explored yet.

To understand to which extent (T, d_2) exhibits metrical tree-likeness, in this section, we conduct a data-driven analysis on AS paths obtained from real-world BGP control plane data.

A. Data collection and analysis

To facilitate the data analysis, we need a large survey of ground truth AS paths set. To obtain this set, we use a collection of BGP tables (collected on 08:00 AM UTC on August 29, 2013) obtained from the RouteViews [11] and RIPE [12] repositories. Although we only consider one snapshot data in this study, a brand new snapshot on BGP tables is available in every two hours and an additional update is available in every fifteen minutes [11], [12]. From the BGP routing tables, we can extract AS paths. Each AS path is a path from a source AS, via a set of intermediate ASes, to a destination IP prefix. For example, the AS path from the AS680 (German National Research and Education Network) to the IP prefix of $65.169.169.0/24$ in U.S. is $AS680 \rightarrow AS6939 \rightarrow AS6598 \rightarrow AS25612$. Note that the IP prefixes $65.169.169.0/24$ belongs to the $AS25612$.

The full dataset is collected from 389 unique monitors; it consists of over 60 million AS paths and contains at least $646,567$ unique destination prefixes. The AS topology obtained from the AS paths data includes $48,133$ ASes and $164,883$ links. The degree distribution of the AS topology is given in figure 1, which is scale-free and follows a power law distribution.

Since part of the monitor-to-prefix paths is missing, we hence filter out the monitors with few known AS paths to IP prefixes, leading to 70 out of 389 monitors selected. All of these 70 monitors can simultaneously reach $30,000$ distinctive IP prefixes, which are from more than $7,000$ different ASes.

Moreover, by accounting the paths originated from one of the vantage ASes and ending with prefixes only appeared in one individual AS, the final ground truth AS paths set contains 2,446,644 AS paths.

Note that, to get the AS hop count, we don't treat the multiple occurrences of the same AS as multiple hops. In other words, the AS hop count is equal to the number of the distinctive ASes in the AS path minus one.

Using the dataset above, we compute the δ-hyperbolicity value distribution based on a sample set of hundreds of millions of quadruplets in the largest connected component of AS topology graph (T, d_2). The result is given in the last column of table II. We can see that (T, d_2) is indeed metrically tree-like with most quadruplets having δ value smaller than or equal to one.

B. Algorithms

Motivated by the observation that AS topology (T, d_2) is metrically tree-like (i.e., low hyperbolicity), we then propose AS path inference algorithms accordingly. To proceed, we first introduce the following definitions.

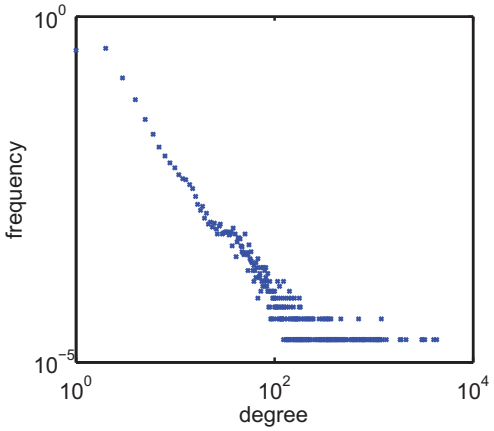

Fig. 1: Power law distribution of node degrees in the AS topology.

Definition 2. *We denote the shortest distance between two points $x, y \in X$ by $|x - y|$. If $x \in X$ and $A \subseteq X$ then*

$$dist(x, A) = \inf\{|x - y|\colon y \in A\}. \tag{2}$$

Definition 3. *For $\epsilon > 0$ the open ϵ-neighborhood $N_\epsilon(A)$ of a set $A \subseteq X$ is*

$$N_\epsilon(A) = \{x \in X : dist(x, A) < \epsilon\} \tag{3}$$

According to the property of the δ-hyperbolicity [31], all triangles in the space are δ-thin, i.e. for all $x, y, z \in X$ and segments $[x, y]$,$[x, z]$ and $[y, z]$, we have

$$[x, y] \subseteq N_\delta([x, z]) \cup N_\delta([y, z]). \tag{4}$$

For the AS topology space (T, d_2), this property implies that, given two AS paths rooted from the same origin to two different destinations ASes, the ground truth AS path between two destinations ASes should be in δ-neighborhood of these two paths. Based on the property above, we then propose an AS path inference algorithm. The key idea is to construct an AS path that is within the δ-neighborhood of the AS path we want to know.

To construct such an AS path, let's first look at AS paths obtained from BGP control plane data. There are hundreds of vantage ASes and each has AS paths from itself to hundreds of thousands of IP prefixes. The entire AS paths originated from every vantage AS can make up a sub-graph of the AS topology. This sub-graph can include loops, so it is not a spanning tree of the original graph. But, still, every pair of AS paths from the same vantage AS n_v to two different IP prefixes prefix_1 and prefix_2 always split at a certain node which we call a branching point, denoted by n_b. Note that, while the two paths may have several branching points, we only consider the first one. Assuming that two paths are $p = n_v \rightarrow \ldots \rightarrow n_b \ldots \rightarrow n_1 \rightarrow \text{prefix}_1$ and $q = n_v \rightarrow \ldots \rightarrow n_b \ldots \rightarrow n_2 \rightarrow \text{prefix}_2$, we define the following function to construct a path to approximate the ground truth AS path:

$$\phi_{n_v}(p, q) = n_1 \rightarrow \ldots \rightarrow n_b \rightarrow \ldots \rightarrow n_2. \tag{5}$$

Fig. 2: Two paths example.

Figure 2 shows a simple example, where the vantage AS is AS10026 and the paths to two different IP prefixes are p = AS10026 $\rightarrow$ AS174 $\rightarrow$ AS39792 $\rightarrow$ $37.140.192.0/22$ and q = AS10026 $\rightarrow$ AS174 $\rightarrow$ AS2914 $\rightarrow$ AS8151 $\rightarrow$ $189.245.128.0/19$. AS174 is the branching point and $\phi_{\text{AS10026}}(p, q)$ = AS39792 $\rightarrow$ AS174 $\rightarrow$ AS2914 $\rightarrow$ AS8151.

In practice, we can have k pairs of AS paths(p_i, q_i) that are originated from multiple vantage ASes $n_{v_i}, i = 1, \ldots, k$ to IP prefix$_1$ and IP prefix$_2$. In this case, suppose that each $\phi_{n_{v_i}}(p_i, q_i), i = 1, \ldots, k$ hits ground truth AS path with a probability P_k independently, the probability that all $\phi_{n_{v_i}}, i = 1, \ldots, k$ fail to hit the AS path would be as the following:

$$P_\emptyset = \prod_{i=1}^{k}(1 - P_k) \tag{6}$$

$P_\emptyset$ decreases exponentially as the number of vantage ASes increase. One straightforward way to incorporate estimation from multiple vantage ASes would be to choose the $\arg\min_{\phi_{n_{v_i}}} |\phi_{n_{v_i}}|, i \in [1, \ldots, k]$ as the estimation. The match rate of this method is equal to the probability that at least one of the $\phi_{n_{v_i}}$ hits the AS path, which is $1 - P_\emptyset$. Based on this simple idea, HyperPath algorithm is given in Algorithm 1.

For the HyperPath algorithm, we do not require AS relationship information. When AS relationship information is taken into account, we develop the Valley-free HyperPath algorithm. It is an extension of the HyperPath algorithm by integrating the valley-free property and is given in Algorithm 2. The idea is to consider two constraints (i.e., valley-free property and low hyperbolicity of the Internet) together to filter possible AS paths. When the valley-free property fails to work, we return the AS path that only considers low hyperbolicity in the inference process.

Algorithm 1 HyperPath Algorithm

INPUT: k pairs of AS paths $(p_i, q_i), i = 1, \ldots, k$. p_i can reach prefix$_1$ and q_i can reach prefix$_2$. Both paths are originated from vantage AS n_i.
OUTPUT: Inferred AS path $\hat{p}$ between prefix$_1$ and prefix$_2$.

$\hat{b} = +\infty$; $\hat{p} = \emptyset$
for $i = 1$ to k **do**
  path$= \phi_{n_i}(p_i, q_i)$
  if $\hat{b} \geq HopCount$(path) **then**
    $\hat{b} = HopCount$(path)
    $\hat{p}$ = path
  end if
end for
return $\hat{p}$

Algorithm 2 Valley-free HyperPath Algorithm

INPUT: k pairs of AS paths $(p_i, q_i), i = 1, \ldots, k$. p_i can reach prefix$_1$ and q_i can reach prefix$_2$. Both paths are originated from vantage AS n_i; AS relationships information on each edge appeared in the AS paths.
OUTPUT: Inferred AS path $\hat{p}$ between prefix$_1$ and prefix$_2$.

$\hat{b_1} = +\infty$; $\hat{b_2} = +\infty$; $\hat{p_1} = \emptyset$; $\hat{p_2} = \emptyset$
for $i = 1$ to k **do**
  path $= \phi_{n_i}(p_i, q_i)$
  if $isValidPath$(path) **and** $\hat{b_1} \geq HopCount$(path) **then**
    $\hat{b_1} = HopCount$(path)
    $\hat{p_1}$ = path
  end if
  if $\hat{b_2} \geq HopCount$(path) **then**
    $\hat{b_2} = HopCount$(path)
    $\hat{p_2}$ = path
  end if
end for
if $\hat{b_1} \neq +\infty$ **then**
  $\hat{p} = \hat{p_1}$
else
  $\hat{p} = \hat{p_2}$
end if
return $\hat{p}$

C. Discussion

Comparisons between the different AS path inference methods from complexity and information requirement aspects are given in Table III. It shows that our proposed algorithms require less information and demand lower computation complexity. Note that, in Table III, $|V|$ and $|E|$ are the total numbers of nodes and links in the AS topology respectively.

Because the HyperPath algorithm and the Valley-free HyperPath algorithms only consider dozens of constructed paths recorded by the vantage ASes, the computational complexity of both algorithms are $O(K)$. Here K is the number of vantage ASes (around a few hundreds), which is much smaller than the number of all ASes in the AS topology. If two end hosts store the AS paths set from the vantage ASes to the networks they are sitting in locally, our methods make it possible for them to infer the AS path connecting them by exchanging the AS paths sets. However, the benchmark methods need to build entire AS topology locally to do inference and, therefore, have to iterate through a much bigger search space. As a result, these methods require higher computational complexity and demand more information. Moreover, our methods are able to infer certain individual AS path between two end hosts at a time. In contrary, to infer certain individual AS path, one of the benchmark methods, KnownPath method, has to infer all AS paths from one node to all other nodes in the graph, even when they are not required.

TABLE III: Comparisons between the different AS path inference methods.

	Baseline method	HyperPath method	AS relationships based method	KnownPath method	Valley-free HyperPath method
Time complexity from one node to all nodes	$O(\|E\|log(\|V\|))$	$O(\|V\|K)$	$O(\|V\|^3)$	$O(\|V\|\|E\|)$	$O(\|V\|K)$
Time complexity from one node to another node	$O(\|E\|log(\|V\|))$	$O(K)$	$O(\|V\|^2)$	$O(\|V\|\|E\|)$	$O(K)$
AS topology information required	global	local	global	global	local
AS relationships information required	no	no	yes	yes	yes

V. Evaluation

In this section, we evaluate the performance of the two proposed methods with realistic AS paths data. We will use two state-of-the-art methods (i.e., AS relationships based inference algorithm and KnownPath algorithm) as the benchmark. In addition, we also implement the no policy method (shortest path heuristic) as the baseline method. Experiment set-up details and evaluation result will be discussed after the introduction of the benchmark methods.

A. Benchmark methods

1) AS relationships based inference algorithm: This algorithm is one of the pioneers and the most cited work on AS path inference algorithm [2]. The key idea of this method is to filter out the paths that don't satisfy the valley-free property and to find the shortest AS path from the remaining valid AS paths set. The algorithm is given in Algorithm 3.

2) KnownPath algorithm: As an extension of the AS relationships based inference algorithm, besides using the valley-free property, this algorithm improves the inference accuracy by further integrating the AS paths that are already observed from the vantage ASes. The algorithm is detailed in Algorithm 4, in which rib_in(u)[p] is a path set that contains all the feasible paths from AS u to a specific IP prefix p learned from u's neighbors. The baseASset contains the ASes that have the assured paths from themselves to the prefix p.

3) No policy baseline method: We also implement the no policy method as the baseline method. In no policy method, the actual AS path is approximated by the shortest AS path in the AS topology obtained from BGP control plane data.

B. Experiment set-up

Algorithm Input:

- AS paths: we use the 2.4 million ground truth AS paths to do the evaluation, which has been introduced in section IV. Besides that, we have also AS paths from 70 vantage ASes to feed our proposed algorithms.
- AS topology: To mimic the case where we don't have access to the routing table of the ASes, we build the AS topology out of the BGP control plane data of the 69 ASes, excluding the one we are interested in. These AS topologies are the only input required by the no policy method. In comparison, our methods don't need to construct the entire topology locally. For the AS relationships based inference method and knownPath method, the AS relationships information on links in the AS topology is also required.
- AS relationship: We use the AS relationships data from Caida's Inferred AS Relationships Dataset [35]. This data is of as high as 97% accuracy [28]. Although it is possible to generate the AS relationships data of the same day when we collected BGP data, we use the AS relationships data on 1st of September 2013 in our study, simply because Caida's AS relationships data is only available on the first day of each month. The date on which AS relationships data is generated is two days later than the date on which BGP control plane data is collected. But we still use this AS relationships data in our study, assuming that most of the AS relationships would not change dramatically and remain almost the same within several days.

To achieve a fair comparison, we organize the experiments in two categories by considering the cases with and without AS relationships information.

- Comparisons without considering AS relationship: In this part, we compare against HyperPath with no policy baseline method. Both methods don't require AS relationship information to do estimation.
- Comparisons with considering AS relationship: In this part, we compare against Valley-free HyperPath with the benchmark methods. In addition to AS path information, all of them take the AS relationships information into account.

C. Estimation Accuracy

Similar to many studies in the literature [2], [16], we evaluate the methods' performance based on the hop count number of AS paths and present the prediction accuracy in the form of confusion matrices.

- Comparisons without considering AS relationships: Table IV shows the prediction accuracy of both the HyperPath method and the no policy baseline method. From Table IV, we observe that the HyperPath method achieves similar performance as the baseline method when AS paths are short (e.g., hop counts are smaller than or equal to 2). HyperPath method achieves significant performance improvement over the baseline method when AS paths are long (e.g., hop counts are greater than 2). For example, when AS path hop counts are greater than 3, the

Algorithm 3 AS relationships based inference algorithm [2]

INPUT: A pair of source and destination ASes(s, d), AS topology T and AS relationships information R of each link in the AS topology.

OUTPUT: An inference path $\hat{p}$ between source and destination ASes (s, d)

find all shortest uphill paths rooted from source node s as $S = \{s \rightarrow \ldots \rightarrow p_i | i \in \{1, \ldots, K\}\}$; and find all shortest uphill paths rooted from destination node d as $D = \{d \rightarrow \ldots \rightarrow q_i | i \in \{1, \ldots, L\}\}$. We denote the end points of the uphill paths rooted from s as $P = \{p_i | i \in \{1, \ldots, K\}\}$ and the end points of the uphill paths rooted from d as $Q = \{q_i | i \in \{1, \ldots, L\}\}$

//cost without peer to peer link
if $P \cap Q \neq \emptyset$ **then**
  $\text{cost}_0 = min_k(dist(s, k) + dist(d, k))$
else
  $\text{cost}_0 = -1$
end if

//cost with peer to peer link
if true $\in \{rp(p, q) | p \in P, q \in Q, p \neq q\}$ **then**
  $\text{cost}_1 = min_{p,q}(dist(s, p) + dist(d, q) + 1)$
else
  $\text{cost}_1 = -1$
end if

Here we assume for two ASes x and y, function $rp(x, y) = \begin{cases} \text{true} & \text{if } (x, y) \text{ is a p2p link;} \\ \text{false} & \text{otherwise.} \end{cases}$

if $\text{cost}_0 == -1$ **and** $\text{cost}_1 == -1$ **then**
  return null.
else if $\text{cost}_0 == -1$ **then**
  return uphill(s,p), (p,q), reverse(uphill(d,q)).
else if $\text{cost}_1 == -1$ **then**
  return uphill(s,k), reverse(uphill(d,k)).
else if $\text{cost}_0 <= \text{cost}_1$ **then**
  return uphill(s,k), reverse(uphill(d,k)).
else
  return uphill(s,p), (p,q), reverse(uphill(d,q)).
end if

HyperPath method possesses more than 50% prediction accuracy, while the baseline method only has an accuracy of 27%.

- Comparisons with considering AS relationships: Table V shows the prediction accuracy of AS relationships based method, KnownPath method and Valley-free HyperPath method. Similar to the case where AS relationship information is not considered, from table V, we also observe similar performance among the different methods when AS paths are short (e.g., hop counts are smaller than or equal to 3). The valley-free HyperPath method, however, achieves significant performance improvement over the benchmark methods when AS paths are long (e.g., hop counts are greater than 3). For example, when AS path hop counts are greater than 6, the Valley-free HyperPath method achieves more than 70% prediction accuracy, while the accuracy of the benchmark methods drops significantly, with the accuracy of less than 17%. Moreover, as reported in [2], [16], we also observe (see the columns of "N/A" in Table V) that benchmark methods could fail to return estimation values, in particular, when AS paths are long. By contrast, the Valley-free HyperPath method is more robust and doesn't suffer from this issue.

Algorithm 4 KnownPath algorithm [16]

INPUT: A destination IP prefix p, baseASset, AS topology T and AS relationships information R of each link in the AS topology.

OUTPUT: Inferred paths from all ASes to the IP prefix p.

queue $\leftarrow \emptyset$
for $v \in$ baseASset **do**
  append v to queue
  path(v)[p] $\leftarrow$sure path of v
  SORT(rib_in(v)[p])
end for
while queue.length > 0 **do**
  $u \leftarrow$ POP(queue, 0)
  for $w \in$ peers(u) **do**
    $P_u \leftarrow$ rib_in(u)[p][0]
    if $w \notin$ baseASset and $(w) + P_u$ is a valid path **then**
      tmppath $\leftarrow$ rib_in(w)[p][0]
      rib_in(w)[p] $\leftarrow$ rib_in(w)[p] $\cup$ $\{(w) + P_u\}$
      SORT(rib_in(w)[p])
      if tmppath == path(w)[p][0] **and** $w \notin$ queue **then**
        append w to queue
      end if
    end if
  end for
end while
return $\{\text{rib\_in}(v) | \forall v \in V\}$

D. Application: inter-domain traffic reduction for BitTorrent P2P system

To demonstrate the usefulness of our AS path inference methods for the inter-domain traffic reduction, we simulate the BitTorrent, which is one of the most popular unstructured peer to peer overlay network applications.

We use the same dataset in the previous section for the underlying network condition. Specifically, we assume that a certain BitTorrent client is located in one of the 93 different IP prefixes. Then the client has to select k number of peers out of a pool of 100 peers. The 100 peers are randomly located in $26,308$ different IP prefixes. We change the selected peers number k from 10, 20 to 90. To evaluate how much inter-domain traffic is reduced by each peer selection method, we

TABLE IV: Confusion matrices of the prediction performance of the baseline method and the HyperPath method.

(a) **No policy method**.

%		Predicted hop count							
		1	2	3	4	5	6	7	8
Actual hop count	1	**50.7**	49.1	0	0	0	0	0	0
	2	2.2	**83.4**	14.4	0	0	0	0	0
	3	0.9	30.2	**67.1**	1.9	0	0	0	0
	4	0.5	13.2	58.3	**26.9**	1.2	0	0	0
	5	1.1	7.9	27.0	38.9	**24.8**	0.3	0	0
	6	0	2.7	16.3	17.9	31.9	**26.1**	5.2	0
	7	0	1.2	1.2	3.5	1.5	67.3	**25.4**	0
	8	0	0	30.7	11.4	17.6	1.1	39.2	**0**

(b) **HyperPath method**.

%		Predicted hop count							
		1	2	3	4	5	6	7	8
Actual hop count	1	**48.1**	51.6	0.3	0	0	0	0	0
	2	0.1	**78.0**	21.8	0	0	0	0	0
	3	0	10.3	**84.0**	5.7	0	0	0	0
	4	0	4.0	32.7	**60.1**	3.2	0	0	0
	5	0	0.7	10.2	30.0	**57.5**	1.6	0	0
	6	0	0.1	3.1	7.4	25.2	**58.6**	5.6	0
	7	0	0	1.3	1.5	1.1	2.3	**92.3**	1.4
	8	0	0	1.1	1.1	12.0	6.9	0.6	**78.3**

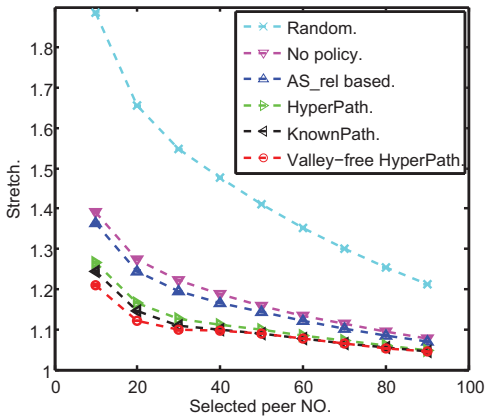

Fig. 3: inter-domain traffic reduction on unstructured overlay networks.

measure the performance on traffic reduction with the stretch metric, which is defined as the following:

$$\text{stretch} = \frac{\sum_{i=1}^{k} d_{x'_i}}{\sum_{i=1}^{k} d_{x_i}} \tag{7}$$

where $x'_i, i = 1, \ldots, k$ is the IDs of the selected peers based on the inferred information and $x_i, i = 1, \ldots, k$ is the IDs of the true best-performing peers in the pool. d denotes the actual AS hop count from the node to the peer. The stretch metric can reflects the ratio of the inter-ASes traffic introduced by the peer selection strategy based on estimation to the inter-ASes traffic introduced by idealized peer selection strategy. We simulate 50 times. For each time, we conduct more than 10,000 rounds of peer selection.

Figure 3 shows the stretch metric with confidence interval with different k for all methods. We can see that Valley-free HyperPath method outperforms all the other methods. For instance, when $k = 10$, the random peer selection method introduces 89% additional inter-domain traffic, compared with the ideal case. If we do peer selection with the help of inference methods without considering AS relationships information, no policy method introduces 39% additional traffic and HyperPath introduces 27% additional traffic. If we take AS relationships information into account, AS relationships based method, KnownPath method and Valley-free HyperPath method introduce 36%, 25% and 21% additional traffic respectively.

This experiment shows that, even though without considering AS relationships information, traffic reduction by HyperPath method is competitive with that by KnownPath method. When we further take into account AS relationships information in our algorithm design, the proposed Valley-free HyperPath method can achieve better performance than the KnownPath method. Please note that, as shown in Table III, KnownPath method also requires additional information on AS topology and higher computational complexity.

TABLE V: Confusion matrices of the prediction performance of the AS relationships based method, the KnownPath method, and the Valley-free HyperPath method.

(a) **AS relationships based method**.

%		Predicted hop count								
		1	2	3	4	5	6	7	8	N/A
Actual hop count	1	**50.7**	43.7	5.4	0.1	0	0	0	0	0
	2	1.8	**81.6**	15.8	0.7	0	0	0	0	0.1
	3	0.9	18.8	**77.1**	3.0	0.1	0	0	0	0.2
	4	0.5	8.9	41.3	**47.6**	1.5	0	0	0	0.2
	5	1.1	6.1	19.5	38.8	**31.6**	0.3	0	0	2.7
	6	0	2.5	14.2	17.5	16.5	**2.9**	0	0	46.5
	7	0	1.2	0.7	3.6	1.1	0.1	**0**	0	93.4
	8	0	0	2.3	0.6	17.6	1.1	0	**0**	78.4

(b) **KnownPath method**.

%		Predicted hop count								
		1	2	3	4	5	6	7	8	N/A
Actual hop count	1	**50.4**	41.3	7.9	0.1	0	0	0	0	0.3
	2	0.3	**78.5**	19.0	1.9	0.1	0	0	0	0.3
	3	0.3	3.7	**88.6**	5.8	0.3	0	0	0	1.3
	4	0.1	2.4	14.6	**74.9**	3.4	0.5	0	0	4.2
	5	0.4	0.2	6.0	16.6	**67.2**	3.2	0.2	0	5.8
	6	0	0.4	3.2	2.8	10.4	**55.2**	0.6	0	27.4
	7	0	0.2	1.1	2.3	0	0.9	**16.4**	0	79.1
	8	0	0	1.1	1.1	12.5	0	4.6	**8.0**	72.7

(c) **Valley-free HyperPath method**.

%		Predicted hop count								
		1	2	3	4	5	6	7	8	N/A
Actual hop count	1	**48.1**	42.5	9.2	0.2	0	0	0	0	0
	2	0.1	**75.1**	22.6	2.2	0	0	0	0	0
	3	0	3.2	**88.8**	7.7	0.3	0	0	0	0
	4	0	2.4	14.9	**77.7**	4.4	0.5	0	0	0
	5	0	0.3	6.4	18.4	**70.2**	4.3	0.3	0	0
	6	0	0.1	2.3	2.9	17.0	**70.9**	6.7	0	0
	7	0	0	1.3	1.5	0.3	2.3	**93.2**	1.5	0
	8	0	0	1.1	1.1	11.9	0	5.7	**79.6**	0

VI. CONCLUSION

In this paper, we revisit the AS path inference problem from the complex network perspective. A brand new constraint is proposed based on the fact the AS paths respect the underlying

geometric structure of the Internet. Resulting two new AS path inference algorithms, HyperPath and Valley-free HyperPath, have $O(K)$ complexity to infer certain end-to-end AS path and can run locally. Intensive evaluation on the ground truth AS paths shows that HyperPath method can not only outperform no policy method, it can be superior to AS relationships based method by being blind to the AS relationships information. The Valley-free HyperPath method outperforms both AS relationships based method and KnownPath method. Moreover, two new algorithms are immune to the fail-to-detect problem, by which the benchmark methods are always haunted. We also simulate BitTorrent P2P applications to show the potential of our methods on inter-domain Internet traffic reduction.

ACKNOWLEDGEMENT

The authors would like to thank Sergey Gorinsky and anonymous reviews for their valuable comments. Thanks to the German Academic Exchange Service (DAAD) for its generous support. This work is also partially supported by the funding from Alexander von Humboldt Foundation.

REFERENCES

[1] J. Hawkinson and T. Bates, "RFC 1930 (the internet engineering task force, fremont, 1996)."

[2] Z. M. Mao, L. Qiu, J. Wang, and Y. Zhang, "On as-level path inference," in *Proceedings of the 2005 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, ser. SIGMETRICS '05. New York, NY, USA: ACM, 2005, pp. 339–349. [Online]. Available: http://doi.acm.org/10.1145/1064212.1064257

[3] A. Akella, S. Seshan, and A. Shaikh, "An empirical evaluation of wide-area internet bottlenecks," *SIGMETRICS Perform. Eval. Rev.*, vol. 31, no. 1, pp. 316–317, Jun. 2003. [Online]. Available: http://doi.acm.org/10.1145/885651.781075

[4] J. Li and K. Sollins, "Exploiting autonomous system information in structured peer-to-peer networks," in *In ICCCN*. IEEE CS Press, 2004, pp. 403–408.

[5] S. Ren, L. Guo, and X. Zhang, "Asap: An as-aware peer-relay protocol for high quality voip," in *Proceedings of the 26th IEEE International Conference on Distributed Computing Systems*, ser. ICDCS '06. Washington, DC, USA: IEEE Computer Society, 2006, pp. 70–. [Online]. Available: http://dx.doi.org/10.1109/ICDCS.2006.18

[6] D. K. Lee, K. Jang, C. Lee, G. Iannaccone, and S. Moon, "Path stitching: Internet-wide path and delay estimation from existing measurements," in *Proceedings of the 29th Conference on Information Communications*, ser. INFOCOM'10. Piscataway, NJ, USA: IEEE Press, 2010, pp. 201–205. [Online]. Available: http://dl.acm.org/citation.cfm?id=1833515.1833556

[7] S. Hasan, S. Gorinsky, C. Dovrolis, and R. Sitaraman, "Trade-offs in optimizing the cache deployments of cdns," in *INFOCOM, 2014 Proceedings IEEE*, April 2014, pp. 460–468.

[8] J. Wu, Y. Zhang, Z. M. Mao, and K. G. Shin, "Internet routing resilience to failures: Analysis and implications," in *Proceedings of the 2007 ACM CoNEXT Conference*, ser. CoNEXT '07. New York, NY, USA: ACM, 2007, pp. 25:1–25:12. [Online]. Available: http://doi.acm.org/10.1145/1364654.1364687

[9] D. Dolev, S. Jamin, O. Mokryn, and Y. Shavitt, "Internet resiliency to attacks and failures under bgp policy routing," *Comput. Netw.*, vol. 50, no. 16, pp. 3183–3196, Nov. 2006. [Online]. Available: http://dx.doi.org/10.1016/j.comnet.2005.11.010

[10] Bgp ipv4/ipv6 looking glass servers. [Online]. Available: http://www.bgp4.as/looking-glasses

[11] "University of oregon route views project." [Online]. Available: http://www.routeviews.org

[12] "Ripe's routing information service raw data project." [Online]. Available: http://www.ripe.net/data-tools/stats/ris/ris-raw-data

[13] T. Bates, P. Smith, and G. Huston. Cidr report. [Online]. Available: http://www.cidr-report.org/as2.0

[14] H. V. Madhyastha, T. Isdal, M. Piatek, C. Dixon, T. Anderson, A. Krishnamurthy, and A. Venkataramani, "iplane: An information plane for distributed services," in *Proceedings of the 7th Symposium on Operating Systems Design and Implementation*, ser. OSDI '06. Berkeley, CA, USA: USENIX Association, 2006, pp. 367–380. [Online]. Available: http://dl.acm.org/citation.cfm?id=1298455.1298490

[15] H. V. Madhyastha, E. Katz-Bassett, T. Anderson, A. Krishnamurthy, and A. Venkataramani, "iplane nano: Path prediction for peer-to-peer applications," in *Proceedings of the 6th USENIX Symposium on Networked Systems Design and Implementation*, ser. NSDI'09. Berkeley, CA, USA: USENIX Association, 2009, pp. 137–152. [Online]. Available: http://dl.acm.org/citation.cfm?id=1558977.1558987

[16] J. Qiu and L. Gao, "As path inference by exploiting known as paths," In Proceedings of IEEE GLOBECOM, Tech. Rep., 2005.

[17] M. Roughan, W. Willinger, O. Maennel, D. Perouli, and R. Bush, "10 lessons from 10 years of measuring and modeling the internet's autonomous systems," *Selected Areas in Communications, IEEE Journal on*, vol. 29, no. 9, pp. 1810–1821, October 2011.

[18] Y. Shavitt and T. Tankel, "On the curvature of the internet and its usage for overlay construction and distance estimation," 2004.

[19] R. Kleinberg, "Geographic routing using hyperbolic space," in *in Proc. of INFOCOM*, 2007, pp. 1902–1909.

[20] A. Cvetkovski and M. Crovella, "Hyperbolic embedding and routing for dynamic graphs," in *INFOCOM 2009, IEEE*, April 2009, pp. 1647–1655.

[21] D. Krioukov, F. Papadopoulos, M. Bogu, and A. Vahdat, "Efficient Navigation in Scale-Free Networks Embedded in Hyperbolic Metric Spaces," arXiv cond-mat.stat-mech/0805.1266, Tech. Rep., May 2008.

[22] L. Gao and F. Wang, "The extent of as path inflation by routing policies," 2002.

[23] "Vuze bittorrent." [Online]. Available: http://www.vuze.com/

[24] B. Quoitin and S. Uhlig, "Modeling the routing of an autonomous system with c-bgp," *Netwrk. Mag. of Global Internetwkg.*, vol. 19, no. 6, pp. 12–19, Nov. 2005. [Online]. Available: http://dx.doi.org/10.1109/MNET.2005.1541716

[25] L. Gao, "On inferring autonomous system relationships in the internet," *IEEE/ACM Trans. Netw.*, vol. 9, no. 6, pp. 733–745, Dec. 2001. [Online]. Available: http://dx.doi.org/10.1109/90.974527

[26] J. Karlin, S. Forrest, and J. Rexford, "Nation-state routing: Censorship, wiretapping, and bgp," arxiv preprint arxiv:0903.3218," 2009.

[27] A. J. N. B. M. C. Joshua Juen, Anupam Das, "Defending tor from network adversaries: A case study of network path prediction," 2014.

[28] M. Luckie, B. Huffaker, A. Dhamdhere, V. Giotsas, and k. claffy, "As relationships, customer cones, and validation," in *Proceedings of the 2013 Conference on Internet Measurement Conference*, ser. IMC '13. New York, NY, USA: ACM, 2013, pp. 243–256. [Online]. Available: http://doi.acm.org/10.1145/2504730.2504735

[29] X. Dimitropoulos, D. Krioukov, M. Fomenkov, B. Huffaker, Y. Hyun, k. claffy, and G. Riley, "As relationships: Inference and validation," *ACM SIGCOMM Computer Communication Review (CCR)*, vol. 37, no. 1, pp. 29–40, Jan 2007.

[30] M. Gromov, pp. 75–263, 1987.

[31] M. R. Bridson and A. Häfliger, 2009.

[32] A. Adcock, B. Sullivan, and M. Mahoney, "Tree-like structure in large social and information networks," in *Data Mining (ICDM), 2013 IEEE 13th International Conference on*, Dec 2013, pp. 1–10.

[33] F. F. D. Muad Abu-Ata, "Metric tree-like structures in real-life networks: an empirical study," arxiv preprint arxiv:1402.3364," 2014.

[34] D. Krioukov, F. Papadopoulos, M. Kitsak, A. Vahdat, and M. Boguñá, "Hyperbolic geometry of complex networks," *Phys. Rev. E*, vol. 82, p. 036106, Sep 2010. [Online]. Available: http://link.aps.org/doi/10.1103/PhysRevE.82.036106

[35] Inferred as relationships dataset. [Online]. Available: http://www.caida.org/data/request˙user˙info˙forms/as˙relationships.xml