A logic distance-based method for deploying probing sources in the topology discovery

Xin Zou, Zhongliang Qiao, Gang Zhou*, Ke Xu
State Key Lab. of Software Development Environment, Beihang University, Beijing 100191, P.R.CHINA
{zouxin, qiaozl, gzhou, kexu}@nlsde.buaa.edu.cn

Abstract—**Internet topology plays a vital role in studying network's internal structure and properties. Currently traceroute-based topology discovery is the main approach to map the network. However, the deployments of probing sources are usually quite costly and complex. Even if the total numbers of sources are the same, the overall coverage of the sampled network may vary significantly for different sources. As a result, it is of great importance for a topology discovery project to select a limited set of probing sources to detect more nodes and links. The aim of this paper is to investigate how to select a fixed set of probing sources to maximize the coverage of the sampled network. We propose a novel logic distance-based method to make source placement decisions. Also we evaluate our approach and compare it with other known methods on real network topology and generated topologies.**

I. Introduction

The study of Internet topology is important for a better understanding on the network's internal structures and properties, which could help us improve the performance of network protocols and applications. Traceroute-based measuring is the main approach to map the Internet. Namely, a limited set of sources perform traceroute-like probes to a large set of destinations, then merge these traces into one graph. This approach has been adopted by many famous topology discovery systems[1][2][3][4].

Traceroute-based topology discovery has two major drawbacks. Firstly it is often incomplete and secondly it bring heavy burden on networks. The reason of the first drawback is that a single probe source only discovers a tree of nodes with the source as the root node and so *cross links* [5] could not be discovered. This problem is partially solved by placing multiple probe sources. For instance, the *skitter* [1][6] project places more than 30 sources around the world. We should note that the overall coverage of the network may differ greatly on different selections of sources, even if the total number of sources is fixed. Furthermore, the placement of sources is usually quite costly. So selecting a limited number of sources to obtain a more complete map is of great significance for topology discovery.

Specifically, the problem that we study in this paper is: if we could only deploy N probing sources, then how to select N sources among a set of M potential sources ($N \leq M$) to maximize the number of nodes and links discovered. This is a *NP-complete* problem if the whole network topology is known. This problem becomes even harder if we know little information about the network topology. For simplicity, most topology discovery projects only try to place their source probes geographically widely distributed in the Internet.

The main contribution of this paper is to present an approximate method to solve this problem without knowing the network topology. In this study, we introduce a metric the logic distance (*SDIS*) to denote the distance between two sources. We show that *SDIS* could easily be measured at a low cost: it only needs a small number of traceroute probes. So the selecting problem can be solved by knowing only a little information of the network, which makes our method more applicable compared to other methods. We observe that for a fixed set of sources, if the sum of *SDIS*es between every two sources is large, then the probability to detect more nodes and links is very high. Based on this fact, this problem can be converted into a constrained optimization problem. And we also present a greedy algorithm to solve this problem. The algorithm could provide a result that is close to optimal. Finally, we describe the detailed procedure of our approach and verify its effectiveness.

The rest of the paper is organized as follows. In Section II we discuss some related work and we describe the data sets used for analysis in Section III. Section IV establishes the basic definitions. We discuss the properties related to the logic distance in Section V. In Section VI we present our method on the placement of probing sources. And we evaluate and compare it with other methods in Section VII. In the end, we summarize, conclude and discuss the application and drawbacks of our method in Section VIII.

II. Related work

Barford et al [7] quantified the marginal utility of network topology measurements. After analyzing the *skitter* data set, they pointed out that the marginal utility of adding sources quickly drops from the perspective of interface, node and link. They also observed that low marginal utility does not imply the overall coverage of the network is high.

Guillaume et al [8] proposed an algorithm for the placement of Internet instrumentation in the context of the *IDMaps* project [9]. Their research mainly focuses on how to place a fixed set of measurement mirrors on generated topologies to minimize the maximum distance between a node and

This work was supported in part by National 973 Program of China (Grant No.2005CB321901)

* Gang Zhou is also with National Digital Switching System Engineering & Technological Research Center, Zheng Zhou, He Nan, 450002, P.R.CHINA

the nearest server. They measured the distance between two nodes on the Internet by hop count, round-trip time, minimum bandwidth, etc. Unfortunately this algorithm cannot be applied to traceroute-like topology probes.

Dall' Asta et al [10] pointed out that the probability of node and link detection is related to the *betweenness* centrality of each element and the deployment of probing sources. They reckoned that placing sources and destinations preferentially on low *betweenness* nodes may increase overall coverage of the sampled graph. Since it is very difficult to determine the *betweenness* of each node without knowing the whole graph, the authors also recommended that the sources could be placed on low-connectivity nodes for simplicity. However the conjecture has never been verified on any real deployed system.

Han et al [11] investigated the relationship between the placement of traceroute sources and their sampled result. The authors classified the nodes and links that cannot be detected by traceroute probes into two categories: *cross link* and *equivalent shortest paths*. Besides, they proposed an iterative method to select sources, the key idea is: when adding a new probing source, the intersections of the probing results of the new source and each existing source should be as small as possible. They also evaluated their approach on several generated graphs. However, their method is based on the premise that all nodes on the graph can be selected as sources, which is impossible in the real network topology discovery.

However, there are distinct differences between our method and others mentioned above. Firstly, our method does not require to know the overall topology of network before deploying sources. Secondly, we select sources from a limited set of potential probing sources. Not all the nodes of the network are required to be potential sources. Thirdly, we introduce a new metric to quantify the distance between two nodes by measurements instead of geographical positions.

III. Data sets

We employ the following two data sets to carry on our research: the *Dolphin2* data set and *Rocketfuel* data set.

The *Dolphin2* data set refers to the public IPv6 backbone network traceroute raw trace data collected by the authors. A single host was used to probe traceroute packets with source routing for more than two weeks. Source routing support is mostly enabled in IPv6 routers, and it allows one to discover paths between arbitrary two nodes of the network. We randomly selected 280 source routing enabled routers as intermediate routes, each route traced to 1802 IPv6 prefixes dumped from IPv6 BGP routing tables. We can simply regard this data set as the results of 280 probing sources tracing to 1802 destinations.

The *Rocketfuel* data set refers to the IPv4 network raw trace data collected by the *Rocketfuel* project [12] on 12/20/2002. There are 38 geographically widely distributed probing sources deployed on Planetlab [13]. Each source performs traceroute to all of the 125,000 prefixes in the BGP routing tables of RouteViews [14]. This is one of the most

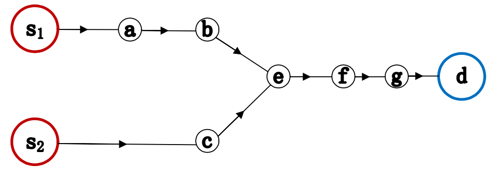

Fig. 1. $SDIS(s_1, s_2, d)$

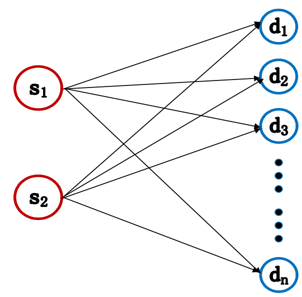

Fig. 2. $SDIS(s_1, s_2, DEST_n)$

complete traceroute data sets open to the public. Additionally, because five nodes of them could not get valid traceroute results for a large portion of destinations, we only use 33 nodes.

IV. Preliminary definitions

Definition 1

$SOURCE_n$ is a set of probing sources, the size of which is n.

$DEST_m$ is a set of destinations, the size of which is m.

Definition 2

We use $I(s,d)$ to denote a set of interfaces discovered when probing source s performs traceroute to destination d. It should be noted that this set does not include s, d and anonymous routes.

For instance, in Fig. 1, $I(s_1,d) = \{a,b,e,f,g\}$ and $I(s_2,d) = \{c,e,f,g\}$.

Definition 3

We use *SDIS* to denote the logic distance between two probing sources. For probing sources s_1, s_2 and a single destination d, *SDIS* is defined as:

$$SDIS(s_1, s_2, \{d\}) = \frac{|I(s_1,d) \bigtriangleup I(s_2,d)|}{|I(s_1,d)| + |I(s_2,d)|}{}^{1}$$

In Fig. 1, $|I(s_1,d)| = 5$, $|I(s_2,d)| = 4$, $|I(s_1,d) \bigtriangleup I(s_2,d)| = 3$, so $SDIS(s_1,s_2,\{d\}) = 3/9 = 1/3$.

For probing source s_1, s_2 and $DEST_n$, $SDIS$ is defined as:

$$SDIS(s_1, s_2, DEST_n) = \frac{\sum_{d \in DEST} SDIS(s_1, s_2, \{d\})}{n}$$

We can conclude from the definition of *SDIS*: (1)Unreachable destinations and anonymous routers are not involved in the computing of *SDIS*. Generally, most destination addresses are generated from prefixes in the BGP routing tables. There

[1] $A \bigtriangleup B = (A - B) \bigcup (B - A)$

is a high probability that these destination addresses do not exist. Besides, anonymous routers are also quite common in topology discovery. Hu et al [15] also introduce a metric *RSIM* to quantify route similarity, which is somewhat similar to *SDIS*. While *RSIM* requires that there should be no anonymous routers in the paths and all the destinations are reachable, otherwise it cannot be measured. This constraint limits its wide use. (2)*SDIS* is related to destinations. In Section V, we show that the value of $SDIS(s_1, s_2, DEST_n)$ varies little corresponding to different destination sets. As a result, we can easily measure the logic distance between two probing sources by tracing to a small number of destinations. So we can simplify $SDIS(s_1, s_2, DEST_n)$ as $SDIS(s_1, s_2)$.

Definition 4

We define $dissimilarity(SOURCE_n)$ as the *dissimilarity* of a set of probing sources:

$$dissimilarity(SOURCE_n) = \frac{\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} SDIS(s_i, s_j)}{\frac{n(n-1)}{2}}$$

We can obtain $n(n-1)/2$ pairs of sources by combining every two elements of $SOURCE_n$. So the $dissimilarity$ of $SOURCE_n$ is the average logic distance of each source-pair.

V. PROPERTIES

We investigate two properties relevant to *SDIS* in this Section. Destination insensitivity property means we can simply choose a small set of destinations to calculate *SDIS* of two probing sources instead of a large set of destinations. The correlation between the *dissimilarity* of a source set and the number of interfaces/links discovered allows us to approximately convert probing sources selecting problem into a constrained optimization problem.

A. Destination Insensitivity

The definition of *SDIS* shows that $SDIS(s_1, s_2, DEST_n)$ is related to the set of destinations. Many topology discovery projects have a very large destination data set, for instance, *skitter* has more than 970,000 destinations. If we have to perform traceroute to all these destinations only for calculating $SDIS(s_1, s_2)$, it is too costly not only on the time of traceroute, but also on the burden of network brought by these probes. So it is of great significance to find a costless way to obtain *SDIS*. We conjecture that if the logic distance between two sources varies little corresponding to different destination sets, then we can obtain *SDIS* only by probing to a small set of destinations.

In order to verify this conjecture, we employed the *Rocketfuel* data sets to make some off-line analysis firstly. We denote $DEST_{125,000}$ contains all the destinations of the *Rocketfuel* data set. $SDIS(s_1, s_2, DEST_{125,000})$ is used as a standard value to compare with $SDIS(s_1, s_2, DEST_X)$, where X is a much smaller value than 125,000. We selected all 33 potential probing sources in the *Rocketfuel* data set and got 528 source pairs ($= 33 * 32/2$) by combining every two elements of these sources.

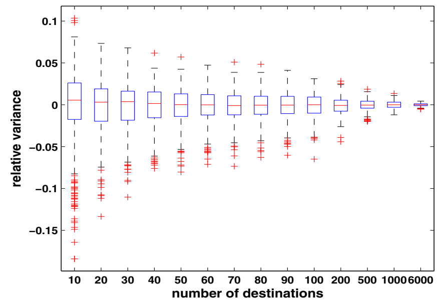

Fig. 3. Dispersion of *relative variances* for different destination sets

The *relative variance* of $SDIS(s_1, s_2, DEST_X)$ and $SDIS(s_1, s_2, DEST_{125,000})$ is defined as

$$\frac{SDIS(s_1, s_2, DEST_X) - SDIS(s_1, s_2, DEST_{125,000})}{SDIS(s_1, s_2, DEST_{125,000})}$$

For a given X, we calculated each *relative variance* of all the 528 source pairs corresponding to a randomly selected destination set $DEST_X$. If these values are small enough, then we can conclude that the variances of destinations have little impact on the logic distance between two sources.

In Fig. 3, box plot is used to indicate the dispersion degree of the *relative variances* for each $DEST_X$. The figure depicts the dispersion of 528 *relative variances* with respect to different size of destination sets. Note that x-axis is the size of $DEST_X$. The three line of a box represent lower quartile, median and upper quartile. We can see from the figure that the data dispersions become narrower when the number of destinations increases. When X = 100, more than 97% (501/528) of the data lie between -0.03 and 0.03. It indicates that $SDIS(s_1, s_2)$ is approximately close to $SDIS(s_1, s_2, DEST_{100})$. So we can randomly choose 100 destinations, perform traceroute from s_1 and s_2 respectively and then compute $SDIS(s_1, s_2, DEST_{100})$ to obtain $SDIS(s_1, s_2)$.

We also carried out a similar experiment based on the *Dolphin2* data set, and this property has also been observed. More validations are needed to determine whether $DEST_{100}$ is precise enough to compute *SDIS* for other data sets with a larger destination set. But we believe $DEST_{100}$ can be applied to most topology discovery projects.

B. Correlation between the dissimilarity and the results detected

In this subsection we study the correlation between the *dissimilarity* of a set of sources and the number of nodes/links discovered. We randomly selected 1000 different $SOURCE_{30}$ from the *Dolphin2* data set and computed the *dissimilarity* of each source set. In Fig. 4, we use scatter plot to demonstrate how the number of nodes and links distribute when the value of $dissimilarity(SOURCE_{30})$ grows.

We can conclude from Fig. 4 that the probability of detecting more nodes and links is becoming higher when

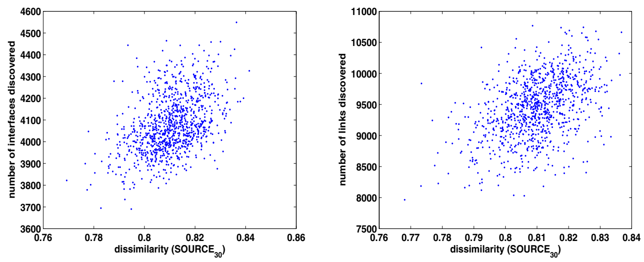

Fig. 4. Correlation between the *dissimilarity* and the number of interfaces/links

$dissimilarity(SOURCE_{30})$ grows. We have also observed the same property when the size of source set varies. We are inspired by this property: maybe we can try to obtain a higher *dissimilarity* so as to obtain a higher coverage of sampled network. That means, if we want to place N sources, we should try to maximize $dissimilarity(SOURCE_N)$.

In the rest of this paper, we present a method based on this property and validate it.

VI. Source Placing Method

A. Problems description

First we present how to convert probing source selecting problem into a constrained optimization problem.

- *Original problem:* A topology discovery project would like to place N probing sources. Then the problem is how to select N probing sources from M ($N \leq M$) potential probing sources to maximize the total number of nodes and links discovered.
- *Converted problem:* $SOURCE_M$ is the possible probing source set. Try to select a subset $SOURCE_N$ ($N \leq M$) to maximize $dissimilarity(SOURCE_N)$.

  A more formalized definition is:

$$maximize \quad f(k_1, k_2, \cdots, k_n) = \sum_{i=1}^{M-1} \sum_{j=i+1}^{M} SDIS(s_i, s_j) k_i k_j$$

$$subject\ to \qquad \sum_{i=1}^{M} k_i = N, \ \ k_t = 0 \ or \ 1 \ (t = 1, 2, 3, \cdots, M)$$

B. The method

Based on the experiments and analysis previously, we first present our method for this probing source selecting problem. The detailed procedure is:

1) Randomly select 100 destinations to construct $DEST_{100}$.
2) Every potential probing source in $SOURCE_M$ performs traceroute to all the destinations in $DEST_{100}$. Record these raw traces results.
3) Compute the logic distance of each source-pair in $SOURCE_M$. The result is a $SDIS(s_i, s_j)$ set ($s_i, s_j \in SOURCE_M, i \neq j$), which is termed as *dissimilarity* set of $SOURCE_M$. The size of this set is $M \cdot (M-1)/2$.
4) Select a subset $SOURCE_N$ from $SOURCE_M$ to maximize $dissimilarity(SOURCE_N)$.

The step 4 of the method brings forward a *NP-complete* problem. Note that we do not focus on investigating an optimal algorithm to solve this problem in this study. However, a greedy algorithm will be presented in next subsection, which could give an approximate result for this problem.

C. Greedy algorithm

The key idea of this algorithm is: when adding a new source into the set, the *dissimilarity* of currently selected source set should be the biggest compared to adding the other sources. Fig. 5 is the pseudo code.

Input:
the number of selected sources N
potential source set $SOURCE_M$
dissimilarity set of $SOURCE_M$
Output:
RESULT
Steps:
RESULT ← ∅
find the maximal element $SDIS(s_p, s_q)$ *in the dissimilarity set*
$SOURCE_M \leftarrow SOURCE_M - \{s_p, s_q\}$
$RESULT \leftarrow RESULT \cup \{s_p, s_q\}$
while $|RESULT| < N$ **do**
  for each S **in** $SOURCE_M$ **do**
    if $dissimilarity(\{RESULT, S\}) > Max$
      NEW ← *S*
      Max ← $dissimilarity(\{RESULT, S\})$
  end for
  $SOURCE_M \leftarrow SOURCE_M - \{NEW\}$
  $RESULT \leftarrow RESULT \cup \{NEW\}$
  Max ← *0*
end while

Fig. 5. Greedy algorithm for solving problem in step 4

The complexity of this algorithm is $O(MN)$. We also compared the *dissimilarity* of $SOURCE_N$ which is selected by the greedy algorithm and the simulated annealing algorithm respectively(see Table I). It shows that the results of these two algorithms are quite close to each other. Nevertheless, whether a more appropriate algorithm exists still needs more study. Though in general a greedy algorithm cannot give the best answer to our problem, it usually obtains a pretty good result that is approximately close to the best answer. And usually greedy algorithms are substantially faster than other algorithms, such as exact algorithms.

TABLE I
Comparison of two algorithms on $dissimilarity(SOURCE_N)$

N	10	50	90	130	170	210
greedy	0.928	0.877	0.859	0.850	0.839	0.829
simulated annealing	0.929	0.877	0.860	0.849	0.838	0.829

VII. EXPERIMENTS

In this section we perform some experiments to evaluate the effect of our method. But it should be noted that not all topology discovery data sets are appropriate for this experiment. So firstly we discuss the characteristics of appropriate data sets, and then we evaluate our method and compare it with other methods.

A. Characteristic of data set

The data set employed should at least satisfy one characteristic: compared with the size of the network measured, the number of potential probing sources is not small (e.g. network with 10,000 nodes and 100 potential source nodes). Otherwise there is a high probability that these sources are evenly distributed among the network. In this case, although the probing source sets are different, the network coverage may vary little. For instance, we randomly select 1000 groups of $SOURCE_{10}$ from the *Rocketfuel* data set and count the number of nodes for each group. We find that the maximal value is 4.6% bigger than the minimal value. Similarly, we randomly select 1000 groups of $SOURCE_{10}$ from the *Dolphin2* data set, the maximal value is 23.8% bigger than the minimal value. So the *Dolphin2* data set is more suitable for our evaluations and validations. We also generated graphs with 10,000 nodes and 500 potential source nodes for validations.

Note that the latest reports say *CAIDA* discovered 17791 ASes from IPv4 Internet and 489 ASes from IPv6 Internet [16]. Our IPv6 topology discovery project [17] also shows that the backbone of IPv6 network only consists of about ten thousand routers, which is much smaller than the backbone of IPv4 network. Therefore 33 sources in IPv4 network (e.g. *Rocketfuel*) have a higher probability to be evenly distributed than 280 sources in IPv6 network (e.g. *Dolphin2*). So it is not surprising that the range of the *Dolphin2* data set (23.8%) is wider than that of the *Rocketfuel* data set (4.6%).

B. Evaluations

We carried out experiments on the *Dolphin2* data set and generated topologies to validate our method. Note that our method does not require that all observations from M possible sources are in hands before we start to select N sources. We use these data sets just to verify and evaluate the effectiveness of our method.

Firstly,we used the raw traces of the *Dolphin2* data set to compute the logic distances. So it is almost equal to employing our method on real topology discovery system. There are 280 probing sources in the *Dolphin2* data set. We randomly selected $X(1 \le X \le 280)$ sources from the 280 sources and counted the number of interfaces and links discovered respectively. We did this 100 times and computed the average value, which is $RANDOM_X$. We also selected $X(1 \le X \le 280)$ sources using our methods, the number is $GREEDY_X$. The *relative variance* is $(GREEDY_X - RANDOM_X)/RANDOM_X$. Fig. 6 plots the *relative variances* corresponding to different number of sources selected.

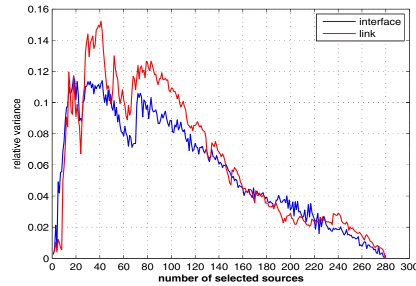

Fig. 6. Compare with random selection method on the *Dolphin2* data set

We can see from Fig. 6 that the trend of the two curves is ascending when X is less than 40. Then the trend is descending and reaches zero when all the sources are selected $(X = 280)$. The average values of the two curves are 0.057 and 0.064 respectively. That means we could detect 5.7% more interfaces and 6.4% more links on average than randomly selected.

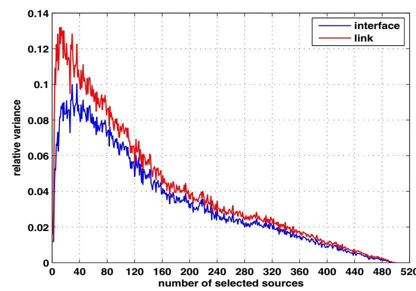

Fig. 7. Compare with random selection method based on the BA model

Secondly, we generated some graphs(10,000 nodes) based on several models and randomly choose 500 potential source nodes and 2000 destination nodes from them. And we recorded the shortest paths between sources and destinations to simulate traceroute on these graphs. Then we performed experiments similar to what we did on the *Dolphin2* data set. Fig. 7 plots the results on generated graph based on BA model[18]. The average values of the two curves are 0.034 and 0.041 respectively. We also observed very similar results on graph based on the PLRG model[19] and even on random graphs. So it can be concluded that our method is also effective on these generated graphs.

C. Comparisons with other methods

We also compared our method with others known to us on the *Dolphin2* data set. Similar to the above experiments, we selected $SOURCE_N$ from $SOURCE_M(N \leqslant M)$ and counted the number of interfaces and links discovered. These methods are:

1) *Selection by degree*: Dall' Asta et al [10] suppose that deploying sources on low degree nodes may improve overall topology coverage. We ranked the sources in

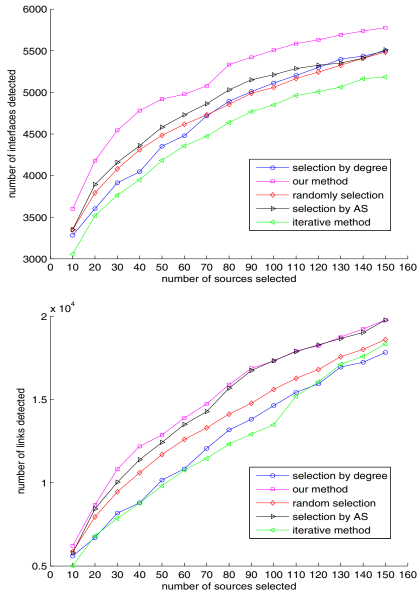

Fig. 8. Comparisons of five methods

$SOURCE_M$ by node degree, and selected the top N lowest degree sources.

2) *Selection by AS*: We queryed whois server [20] to determine the AS number of all probing sources in $SOURCE_M$. And then we tried to make all selected sources evenly spread to different ASes. Generally, different ASes always locate at different geographical positions. So this method is very similar to selection by geography.
3) *Iterative method*: We used the revised iterative method proposed by Han [21]. This revised method allows that not all the nodes are possible probing sources.
4) *Random selection*: For each N, we selected 100 different $SOURCE_N$, and counted the average number of interfaces and links.

The comparison results are plotted in Fig. 8 and the x-axis is N. We can conclude from these results: (1) Our method is better than others, especially on the number of interfaces detected. (2) The method "selection by AS", though discover less interfaces than our method, almost detects as much links as our method when N is above 80. We find that two probing sources in different ASes also can be very similar. For instance, sources 2001:40d0::122 (AS20912) and 2001:1418:1:400::22 (AS12779) belong to different ASes and organizations, but more than 90% of the interfaces and links they discovered are the same. (3) The iterative method is not so effective here by reason that the limitations on potential sources restrict the effectiveness of this method.

VIII. Conclusion and Future Work

In this paper, we present a logic distance-based method on how to select a given number of traceroute-like topology discovery probing sources. And then we validate it on our data set and on some generated topologies. The results demonstrate that our method is more effective than random selection method as well as other known methods.

We should point out the fact that our approach is effective on our data set does not imply that it also works well on others. Our approach is more suitable to be applied on a relatively small network with a large number of potential probing sources (e.g. IPv6 backbone network). Besides, measurement errors (e.g. caused by abnormal network behaviors) may also bring great impacts on determining the logic distances. For instance, five probing sources behave abnormally in the *Rocketfuel* data set. All the factors mentioned above may cause ineffectiveness of our method. Despite these drawbacks, it is still very useful for some topology discovery projects, especially for the kinds using source routing or tunnels [22][23].

As for future work, we would like to validate our approach on more data sets. And this work will be applied in our *Dolphin* system [17][22].

References

[1] "CAIDA Skitter tool," http://www.caida.org/tools/measurement/skitter/.
[2] R. Govindan and H. Tangmunarunkit, "Heuristics for Internet map discovery," in *INFOCOM 2000.*, 2000, pp. 1371–1380.
[3] D. G. Waddington, F. Chang, R. Viswanathan, and B. Yao, "Topology discovery for public IPv6 networks," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 33, pp. 59–68, 2003.
[4] N. Spring, R. Mahajan, D. Wetherall, and T. Anderson, "Measuring ISP topologies with Rocketfuel," *IEEE/ACM Trans. Netw.*, vol. 12, 2004.
[5] R. Siamwalla, R. Sharma, and S. Keshav, "Discovering Internet topology," Tech. Rep.
[6] B. Huffaker, D. Plummer, D. Moore, and k Claffy, "Topology discovery by active probing," *Applications and the Internet Workshops, IEEE/IPSJ International Symposium on*, vol. 0, p. 90, 2002.
[7] P. Barford, A. Bestavros, J. Byers, and M. Crovella, "On the marginal utility of network topology measurements," in *Proceedings of the 1st ACM SIGCOMM Workshop on Internet Measurement*, 2001, pp. 5–17.
[8] G. Jean-Loup, L. Matthieu, and M. Damien, "Relevance of massively distributed explorations of the Internet topology: qualitative results," *Comput. Netw.*, vol. 50, pp. 3197–3224, 2006.
[9] S. Jamin, C. Jin, Y. Jin, D. Raz, Y. Shavitt, and L. Zhang, "On the placement of Internet instrumentation," in *INFOCOM*, 2000.
[10] L. Dall'Asta, I. Alvarez-Hamelin, A. Barrat, A. Vazquez, and A. Vespignani, "Exploring networks with traceroute-like probes: theory and simulations," *Theoretical Computer Science*, vol. 355, p. 6, 2006.
[11] W. Han and K. Xu, "A method for placing traceroute-like topology discovery instrumentation," in *ICCS 2008. 11th International Conference on Communication Systems. Proceedings. IEEE*, 2008.
[12] "Rocketfuel," http://www.cs.washington.edu/research/networking/rocketfuel.
[13] "Planetlab," http://www.planet-lab.org.
[14] "Route views project," http://www.antc.uoregon.edu/routeviews/.
[15] N. Hu and P. Steenkiste, "Quantifying Internet end-to-end route similarity," in *Passive and Active Measurement Conference*, 2006.
[16] "IPv4 and IPv6 Internet topology at a macroscopic scale," Tech. Rep., http://www.caida.org/research/topology/as_core_network.
[17] "IPv6 backbone network topology," http://ipv6.nlsde.buaa.edu.cn.
[18] B. A.L. and A. R., "Emergence of scaling in random networks," *Science*, pp. 509–512, October 1999.
[19] W. Aiello, F. Chung, and L. Lu, "A random graph model for massive graphs," in *STOC '00*. ACM, 2000, pp. 171–180.
[20] "RISwhois," http://www.ripe.net/ris/riswhois.html.
[21] W. Han, "Efficient automatic topology discovery for IPv6 Internet, Beihang University," Master's thesis, 2008.
[22] X. Lang, G. Zhou, C. Gong, and W. Han, "Dolphin: The measurement system for the next generation Internet," in *Communications, Internet, and Information Technology*, 2005.
[23] Z. Liu, J. Luo, and Q. Wang, "Large scale topology discovery for public IPv6 networks," in *IEEE ICN '08: Proceedings of the Seventh International Conference on Networking*, 2008.