The Weak Network Tracing Problem

H.B. Acharya[1] and M.G. Gouda[1,2]

[1] The University of Texas at Austin, USA
[2] The National Science Foundation, USA
{acharya,gouda}@cs.utexas.edu

Abstract. Computing the topology of a network in the Internet is a problem that has attracted considerable research interest. The usual method is to employ Traceroute, which produces sequences of nodes that occur along the routes from one node (source) to another (destination). In every trace thus produced, a node occurs by either its unique identifier, or by the anonymous identifier $''*''$. We have earlier proved that there exists no algorithm that can take a set of traces produced by running Traceroute on network N and compute one topology which is guaranteed to be the topology of N. This paper proves a much stronger result: no algorithm can produce a small set of topologies that is guaranteed to contain the topology of N, as the size of the solution set is exponentially large. This result holds even when every edge occurs in a trace, all the unique identifiers of all the nodes are known, and the number of nodes that are irregular (anonymous in some traces) is given. On the basis of this strong result, we suggest that efforts to exactly reconstruct network topology should focus on special cases where the solution set is small.

1 Introduction

A wide variety of networked applications can be optimized for performance using information about the underlying network, i.e. the topology of the actual connectivity graph showing the paths along which packets are routed. One example of such an approach is P4P [11], which enables P2P networks to optimize traffic within each ISP and reduce cross-ISP traffic. Considerable research has been devoted to the problem of reconstructing the topology of a network in the Internet [9].

The usual mechanism for generating the topology of a network is by the use of Traceroute [5]. Traceroute is executed on a node, called the source, by specifying the address of a destination node. This execution produces a sequence of identifiers, called a *trace*, corresponding to the route taken by packets traveling from the source to the destination. A trace set T is generated by repeatedly executing Traceroute over a network N, varying the *terminal* nodes, i.e. the source and destination.

In theory, given that T contains traces covering every node and every edge, it is possible to reconstruct the network exactly. However, in practice, there arise problems: incomplete coverage, anonymity (nodes may refuse to state their unique identifiers), and aliasing (nodes may have multiple unique identifiers).

K. Kant et al. (Eds.): ICDCN 2010, LNCS 5935, pp. 184–194, 2010.

The situation is further complicated by load balancing, which may cause incorrect traces; tools such as Paris Traceroute [10] attempt to correct this problem.

This paper deals with the node anonymity problem. An anonymous node in a trace may or may not be identical to any other anonymous or named node. Consequently, there may be multiple topologies for the computed network, all of which can generate the observed trace set. Previous work, as discussed in Section 5, employs heuristics to compute a topology with a "small" number of anonymous nodes. We consider the extreme case where no nodes are consistently anonymous; a node may be anonymous in some trace, but there exists at least one trace in trace set T where it states its unique identifier. Such nodes are called *irregular* nodes. In our earlier paper [1] we have proved that even in this special case, given a trace set it is not in general possible to compute a unique network topology, or even a constant number of network topologies. This paper proves a much stronger negative result: there is no general algorithm that takes as input a trace set and computes a solution set (of all possible network topologies) of polynomial size. This result is true even if m, the number of anonymous nodes, is known a priori; the size of the solution set is worst-case exponential in m. Further, we show that the result holds even under multiple strong assumptions (stable and symmetric routing, unique identifiers, and complete coverage). This proves that the problem remains intractable to the strongest known network tracing techniques, such as Paris Traceroute and inference of missing links [8].

In the next section, we formally define terms such as network, trace and trace set, so as to be able to develop our mathematical treatment of the problem.

2 Networks, Traces and Network Tracing

In this section, we present formal definitions of the terms used in the paper. We also explain why we assume several strong conditions. Finally, we provide a formal statement of the problem studied.

2.1 Term Definitions

A network N is a connected graph where nodes have unique identifiers. Nodes are either *regular* or *irregular*, and either *terminal* or *non-terminal*. (These terms are used below.)

A *trace* is a sequence of node identifiers.

A trace t is said to be *generable from* a network N iff the following four conditions are satisfied:

(a) t represents a simple path in N.
(b) The first and last identifiers in t are the unique identifiers of terminal nodes in N.
(c) Each regular node "a"in N appears as "a" in t.
(d) Each irregular node "$a/*$"in N appears as either "a" or "$*_i$" in t, where i is a unique integer in t.

A *trace set* T is *generable from* a network N iff the following five conditions are satisfied:

1. Every trace in T is generable from N.
2. For every pair of terminal nodes x, y in N, T has at least one trace between x and y.
3. Every edge in N occurs in at least one trace in T.
4. The unique identifier of every node in N occurs in at least one trace in T.
5. T is *consistent*: for every two distinct nodes x and y, exactly the same nodes must occur between x and y in every trace in T where both x and y occur.

These conditions, and our reasons for assuming them, are discussed in detail below.

2.2 Generable Trace Sets

The five conditions, that we impose on a trace set T to be generable from a network N, may appear too strong. However, as we illustrate below, if any one of these conditions is not satisfied by a trace set T, then trace set T becomes generable from an exponentially large number of networks.

As an example, we exhibit a trace set T_1 that is generable from any one of $m!$ networks when Condition 5 is not satisfied by T_1, but Conditions 1 through 4 are.

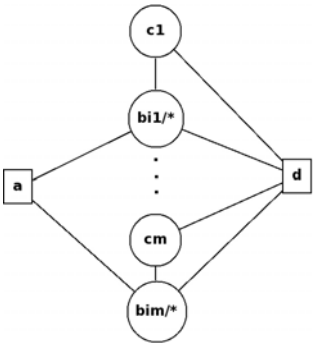
$$T_1 = \{(a, b_1, d), (a, b_2, d), \ldots (a, b_m, d),$$
$$(a, *_1, c_1, d), (a, *_2, c_2, d), \ldots (a, *_m, c_m, d)\}$$

The terminal nodes are a and d, while the nodes $b_1, ..b_m$ and $c_1, ..c_m$ are non-terminal nodes. As a and d appear with their unique identifiers in every trace, they are definitely regular nodes.

It is straightforward to show that the trace set T_1 can be generated from every member of the family of networks shown in Figure 1. This family consists of $m!$ networks, since $\{b_{i1}, b_{i2}, .., b_{im}\} = \{b_1, b_2, .., b_m\}$ but there are $m!$ permutations that satisfy this condition.

Note that there is no way to decide which unique identifier corresponds to any of the anonymous identifiers $*_1$, $*_2$ .. $*_m$. If we only assume that there are no loops in a route, then each of the anonymous nodes may still correspond to several unique identifiers. For example, $*_1$ may correspond to any one of the identifiers $b_1, b_2..b_m$; $*_2$ may correspond to any of the remaining identifiers, and so on. Hence the number of topologies is $m!$.

Our results are negative, and bound the power of any algorithm to take as input a trace set generable from network N and compute the topology of N. In order to show that we do not depend on conditions like inconsistent routing, which may or may not be true, we assume the worst case, develop our theory assuming that all these conditions are met, and prove that our results are still valid.



(a) Network N_0

Fig. 1. A family of topologies for T_1

2.3 The Weak Network Tracing Problem

We can now state a formal definition of the problem studied in this paper.

The *weak network tracing problem* can be stated as follows: "Design an algorithm that takes as input a trace set T, that is generable from a network, and produces a small set $S = \{N_1, .., N_k\}$ of networks such that T is generable from each network in this set and not from any network outside this set." We have previously proved [2] that the cardinality of S is not bounded by a constant. The current paper proves that the problem is indeed unsolvable - the size of the solution set S is, in general, exponential in the number of irregular nodes (and therefore not small). (From this, it is trivial to see that the original network tracing problem, which is similar to the weak problem but requires that the solution set S have exactly one network, is unsolvable.)

3 The Impossibility of Weak Network Tracing

In this section, we begin by considering the special case of a network with exactly one irregular node, and show how to construct a pathological trace set. This construction is then extended to show how the number of possible network topologies grows with the number of irregular nodes. The formal analysis, proving that the solution set is exponential in size for this trace set, is given in Section 4. (Note that we count all topologies which only differ in which nodes are marked as irregular, as a single topology.)

3.1 Weak Tracing with One Irregular Node

Consider the simplest possible trace with an anonymous node, $(a, *_1, b)$. This trace cannot be the only trace in a trace set generable from any network, because by consistent routing neither a nor b lie on the path connecting a and b, so the unique identifier of at least one node - $*_1$ - is unknown. This violates condition 4 (from Section 2).

We add two new traces (a, x_1, y_1) and (b, x_1, y_1). This introduces the new node y_1 and identifies $*_1$ to be x_1, as it is the only node one hop from a (and also the only node one hop from b). Note that it was essential to add two new traces rather than one, as y_1 is a new terminal, and so by condition 2 there is at least one trace connecting y_1 to every other terminal - a and b.

We can repeat this step an arbitrarily large number of times, adding the new traces (a, x_i, y_i) and (b, x_i, y_i) for $i = 1, 2..k$. In order to satisfy Condition 2, we also add the traces (y_i, y_j) for all $i \neq j$. For brevity, we name the whole operation (adding the two traces, and adding traces to maintain a completely connected graph among the y_i) 'Op1'. Note that k, the number of times we choose to execute Op1, is a positive integer.

For any value of k, the trace set is generable from a network; further, given $k > 1$, it is not possible to state which x_i corresponds to $*_1$.

All the topologies generated thus far are identical, only differing in the placement of the irregular node. (For the case where $k = 3$, this topology is shown in network N_1 of Figure 2. In the figure, we show $*_1 = x_1$.) Now we take the critical step : we replace

(a, x_1, y_1) with $(a, *_1, y_1)$,
and (b, x_1, y_1) with $(b, *_1, y_1)$.

This step will be referred to as 'Op2'.

The trace set is no longer generable from a network, as Condition 4 (all identifiers must occur in a trace set) is now violated. We replace

$(a, *_1, b)$ with (a, x_1, b),

so the trace set is generable from a network again. This step is called 'Op3'.

The importance of this change is that, now, we no longer know which node connects y_1 to a or b. Setting $*_1 = x_i$ produces a different topology for every i. (The different topologies are not even necessarily isomorphic.) Note that we could have selected any one x_i (instead of x_1) and interchanged its place with $*_1$ in the trace set; the results would have been the same.

As an example, we present the special case when $k = 3$ in Figure 2. The trace set

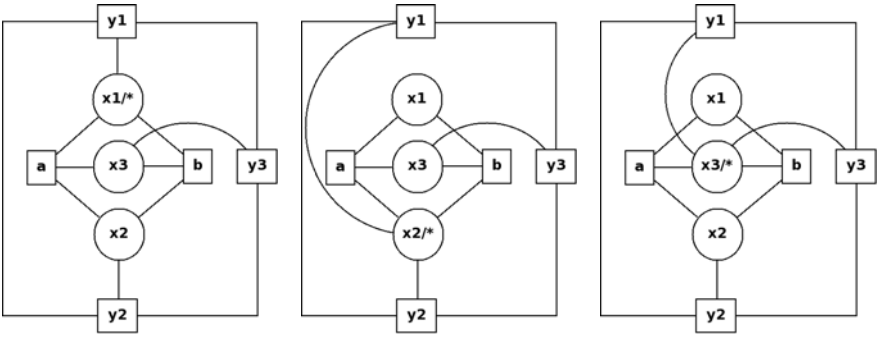
$$\begin{aligned} T_2 = \{&(a, x_1, b), (a, *_1, y_1), (a, x_2, y_2), (a, x_3, y_3), \\ &(b, *_2, y_1), (b, x_2, y_2), (b, x_3, y_3), \\ &(y_1, y_2), (y_1, y_3), (y_2, y_3)\} \end{aligned}$$

is generable from any of the three networks N_1, N_2 and N_3 in the figure. Note that all these networks have exactly one irregular node; hence, even if the number of irregular nodes is known, it does not help to distinguish between them.

3.2 Weak Tracing in a General Network

We now extend the previous construction to the case of a general network with $m > 0$ irregular nodes.

First, we observe that simply repeating the previous construction and allowing the presence of m irregular nodes will result in the same number of topologies



(a) Network N_1 (b) Network N_2 (c) Network N_3

Fig. 2. Networks generable from T_2

as allowing only one. The reason is that, in our trace set, the only traces with the identifier x_i are (a_i, x_i, y_i) and (b_i, x_i, y_i). Both these traces identify which x-node y_i is connected to. When we apply Op2 to replace the identifier x_i with an anonymous identifier, the resultant trace set is no longer generable from a network. To make the trace set generable from a network again, we must apply Op3 after Op2, and specify a path (a, x_i, b). Unfortunately, we cannot apply Op3 more than once; once we specify a path (a, x_1, b), Condition 5 - consistent routing - prevents us from specifying any other path (a, x_i, b) where $i \neq 1$. Hence we cannot replace any other x_i with an anonymous identifier.

In order to overcome this problem, we introduce more a and b nodes. We rename the original terminals a and b to a_1 and b_1, and add new pairs of terminals a_2 and b_2 , .., a_m and b_m. Each of these terminals is connected to x_1 .. x_k, i.e. the exact same connections as the original a and b. To achieve this in the trace set, we add the traces (a_i, x_i, b_i) and (a_i, x_i, y_i) for all i.

To satisfy Condition 2, which states that there must exist at least one trace between every pair of terminals, there must also be traces connecting every a_i to a_j and b_j, $j \neq i$. There are many ways we can write such traces; for this construction,we simply add the traces (a_i, x_1, a_j) and (a_i, x_1, b_j) for all values of i and j. We call the entire operation involved in adding one new pair of terminals 'Op4', so Op4 is executed m times.

We can now perform Op2 multiple times, choosing a node x_i every time. In each execution, we replace all traces $(\square, x_i, y_i)$ with $(\square, *_j, y_i)$ where $\square$ is any identifier. Note that we no longer need to explicitly perform Op3. The reason is that, in this construction, we have already added traces of the form (a_i, x_i, b_i). Even though the traces (a_i, x_i, y_i) and b_i, x_i, y_i are lost, the remaining trace (a_i, x_i, b_i) contains the identifier x_i. As all five conditions are still satisfied, the trace set remains generable from a network after Op2 is executed.

Note that there are two upper bounds on the number of times that Op2 can be executed - k, the number of nodes x_i, and m, the number of irregular nodes. As our goal is to demonstrate the existence of a trace set that is generable

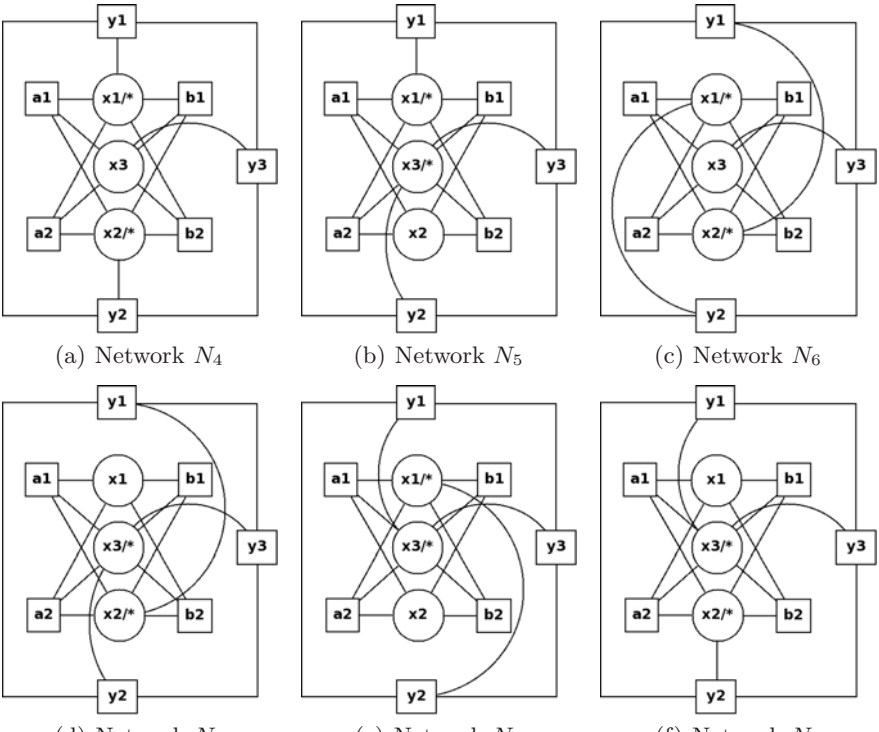


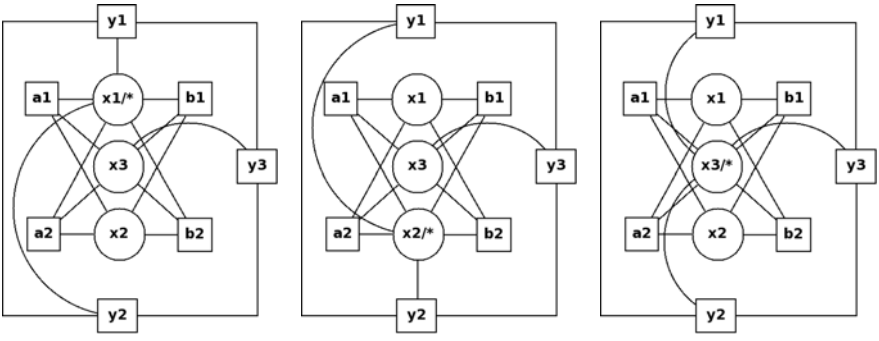
Fig. 3. Networks from T_3 with two irregular nodes

from each of an exponential number of topologies, we are free to specify any (satisfiable) conditions for this set. We now add the condition that $m \leq k$. In our construction, we may choose any m of the k available x_i-nodes as irregular nodes, so there are $\binom{k}{m}$ distinct trace sets that can be constructed by our method.

We now present as an example a trace set generated with the parameters $m = 2, k = 3$. For our example, the x_i identifiers that we replace with anonymous identifiers are x_1 and x_2. The trace set

$$\begin{aligned} T_3 = \{&(a_1, x_1, b_1), (a_1, x_1, a_2), (a_1, *_1, y_1), (a_1, *_2, y_2), (a_1, x_3, y_3), (a_1, x_1, b_2) \\ &(a_2, x_1, b_1), (a_2, x_2, b_2), (a_2, *_3, y_1), (a_2, *_4, y_2), (a_2, x_3, y_3) \\ &(b_1, *_5, y_1), (b_1, x_1, b_2), (b_1, *_6, y_2), (b_1, x_3, y_3), \\ &(b_2, *_7, y_1), (b_2, *_8, y_2), (b_2, x_3, y_3), \\ &(y_1, y_2), (y_1, y_3), (y_2, y_3)\} \end{aligned}$$

is generable from any of the networks N_4, N_5, N_6, N_7, N_8, and N_9 in Figure 3. Note that all these networks have exactly two irregular nodes. If the number of irregular nodes was not known, the networks N_{10}, N_{11} and N_{12} in Figure 4 would also be members of the solution set.



(a) Network N_{10} (b) Network N_{11} (c) Network N_{12}

Fig. 4. Networks from T_3 with one irregular node

4 Size of the Solution Set

In this section, we use the trace set constructed in Section 3 and derive a formal proof that the number of candidate topologies computed from any such trace set is exponential.

Multiple topologies are possible because, for m of the y_i nodes, the only available information is that they are known to be connected to some irregular node. It is not possible to determine exactly which x_i-nodes they are connected to.

By construction, there are k x_i-nodes.

If the number of irregular nodes is not known, then from the trace set, for each of m y_i-nodes there are k possible x_i-nodes to choose to connect to, and the total number of valid topologies is at least k^m. (The number is actually larger because there is no reason to assume that each y_i-node connects to exactly one x_i-node. For example, given the traces $(a_5, *_i, y_8)$ and $(b_5, *_j, y_8)$, we do not know that $*_i = *_j$. However, if we consider only the cases when every y_i-node connects to exactly one x_i-node, we see there are k^m cases. As this is the size of a subset of the solution set, k^m is a lower bound on the number of topologies, and is clearly exponential.)

Now we deal with the case when the number of irregular nodes in a topology is exactly m. In this case, when a y_i-node "chooses" which x_i-node to connect to, it cannot choose an x_i-node that has already been chosen by another y_i-node. Hence the number of topologies is

$$ {}^kP_m = \frac{k!}{(k-m)!} $$

Hence the number of possible topologies is

$$ \begin{aligned} \frac{k!}{(k-m)!} &= (k-m+1).(k-m+2)..(k) \\ &\geq 1.2..m \\ &= m! \\ &\geq 2^{m-1} \end{aligned} $$

This proves that, even if we restrict the solution set to topologies with exactly m irregular nodes, the size of the solution set is exponential. We are now in a position to state the following theorem.

Theorem 1. *There is no algorithm which, given a trace set which is generable from a network, produces an "small" solution set containing all network topologies from which the trace set is generable.*

Proof. By our construction, there exist trace sets such that the size of the solution set is exponential in the number of irregular nodes (as shown above).

(Note that we can strengthen this theorem further and add that the problem remains intractable even when the number of irregular nodes is given.)

We will now show how these results relate to our earlier work in [2].

We express k is terms of m and n, where n is the total number of nodes. Observe that in the construction, we start with two unique identifiers(a and b), remove two identifiers (again a and b), and add new identifiers by executing Op2 (k times) and Op4(m times). Also note that each execution of Op2 or Op4 adds two new identifiers, and as each node has exactly one unique identifier, the number of unique identifiers is also n.

Using the notation $\#(a_i)$ to represent the number of a_i nodes etc. we have

$$\begin{aligned} &\#(a_i) = \#(b_i) = k \\ &\#(x_i) = \#(m_i) = m \\ \Rightarrow &k + k + m + m = n \\ \Rightarrow &k = \frac{n - 2m}{2} \end{aligned}$$

We can now state the following theorem:

Theorem 2. *Given a trace set generable from a network, and known to have exactly one irregular node, the size of the solution set (ie. the number of network topologies that the trace set is generable from) is linear in the number of unique identifiers.*

Proof. We substitute $m = 1$ into the equations derived above. The size of the solution set is given by

$$\begin{aligned} \frac{k!}{(k-1)!} &= k \\ &= \frac{n - 2m}{2} \\ &= \frac{n - 2}{2} \end{aligned}$$

which is clearly $O(n)$.

This is obviously a stronger version of Theorem 7 in [2], which states that the size of the solution set is not upper-bounded by any constant.

5 Related Work

Anonymous router resolution is an established problem in topology mapping studies. Unfortunately, many authors simply avoid dealing with the problem. For example, in [5], the authors stop traces as soon as they encounter an anonymous router. In [4], authors handle anonymous routers by replacing them either with arcs connecting known routers, or with random fresh unique identifiers. It is trivial to see that both approaches produce inaccurate maps. The "sandwich" approach used in [3], merges a chain of anonymous nodes, "sandwiched" between the same pair of known nodes, with each other - thereby losing resolution.

The theoretical study of the router resolution problem was started by Yao et al., who formulate it as an optimization problem [12]. Their goal is to build the smallest possible topology by combining anonymous nodes with each other under two constraints : trace preservation and distance preservation. They prove that the optimum topology inference under these conditions is NP-complete, then propose an $O(n^5)$ heuristic.

All further study of the problem has been based on heuristics. Jin et al. propose two approaches in [9]. The first, an ISOMAP based dimensionality reduction approach, uses link delays or node connectivity as attributes in the dimensionality reduction process. This is an $O(n^3)$ algorithm. (Their approach has been attacked as they ignore the difficulty of estimating individual link delays from round trip delays in path traces [6].) The second, a simple $O(n^2)$ neighbor matching heuristic, suffers from accuracy problems: it has high rates of both false positives and false negatives. Gunes et al. propose their own heuristics in [7], and show performance strictly better than $O(n^3)$ for five heuristics they apply in succession.

In our work, we try to return to a rigorous theoretical approach, and identify a problem complementary to that studied by Yao; in their study, the authors assume that all anonymous nodes must remain consistently anonymous, while in [1] we explore what happens if no nodes are truly anonymous, only irregular. In [2], we show that the (strong) network tracing problem is unsolvable in the general case, but provide special cases of networks, such as trees and odd rings, where it becomes solvable. We also begin our study of the weak network tracing problem. This paper reports our results from studying weak network tracing, strengthens our earlier results, and proves (by counterexample) that there exists no general algorithm to produce a complete but "small" solution set.

6 Conclusion

This paper concludes our study of the general weak network tracing problem, and proves that no solution exists. On this basis, we suggest that research in network tracing focus on special cases that are known to be tractable; for instance, trees and odd rings. We anticipate the scope for a rich body of future work that studies which topologies are tractable to compute from their trace sets. Further, as our theory is developed under strong constraints such as complete coverage

and consistent routing, it would be interesting to study whether these topologies remain easy to compute as these constraints are relaxed.

As our work is graph-theoretic in nature and does not make use of any domain-specific information, we anticipate that it may also be of interest in other problem domains. For example, in social networks which allow an observer to see that A and B are friends-of-friends but not who their mutual friend is, the mutual friend is an irregular node. In such a case, we can use the results in this paper to state that, in general, plotting the social network from "chains of friends" (traces) is a provably hard problem.

References

1. Acharya, H.B., Gouda, M.G.: Brief announcement: The theory of network tracing. In: Principles of Distributed Computing (May 2009) (Accepted)
2. Acharya, H.B., Gouda, M.G.: The theory of network tracing. In: 11th International Symposium on Stabilization, Safety, and Security of Distributed Systems (June 2009) (Submitted for review)
3. Bilir, S., Sarac, K., Korkmaz, T.: Intersection characteristics of end-to-end internet paths and trees. In: 13th IEEE International Conference on Network Protocols, pp. 378–390 (November 2005)
4. Broido, A., Claffy, K.C.: Internet topology: connectivity of ip graphs. Scalability and Traffic Control in IP Networks 4526(1), 172–187 (2001)
5. Cheswick, B., Burch, H., Branigan, S.: Mapping and visualizing the internet. In: Proceedings of the USENIX Annual Technical Conference, pp. 1–12. USENIX Association, Berkeley (2000)
6. Feldman, D., Shavitt, Y.: Automatic large scale generation of internet pop level maps. In: IEEE Global Telecommunications Conference (GLOBECOM), December 4, pp. 1–6 (2008)
7. Gunes, M., Sarac, K.: Resolving anonymous routers in internet topology measurement studies. In: INFOCOM 2008. The 27th Conference on Computer Communications. IEEE, pp. 1076–1084 (April 2008)
8. Gunes, M.H., Sarac, K.: Inferring subnets in router-level topology collection studies. In: Proceedings of the 7th ACM SIGCOMM conference on Internet measurement, pp. 203–208. ACM Press, New York (2007)
9. Jin, X., Yiu, W.-P.K., Chan, S.-H.G., Wang, Y.: Network topology inference based on end-to-end measurements. IEEE Journal on Selected Areas in Communications 24(12), 2182–2195 (2006)
10. Viger, F., Augustin, B., Cuvellier, X., Magnien, C., Latapy, M., Friedman, T., Teixeira, R.: Detection, understanding, and prevention of traceroute measurement artifacts. Computer Networks 52(5), 998–1018 (2008)
11. Xie, H., Yang, Y.R., Krishnamurthy, A., Liu, Y.G., Silberschatz, A.: P4p: provider portal for applications. SIGCOMM Computer Communications Review 38(4), 351–362 (2008)
12. Yao, B., Viswanathan, R., Chang, F., Waddington, D.: Topology inference in the presence of anonymous routers. In: Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies. IEEE, March-3 April 2003, vol. 1, pp. 353–363 (2003)