Brief Announcement: The Theory Of Network Tracing

Hrishikesh B. Acharya
Department of Computer Science
University of Texas at Austin
Austin, TX, USA
acharya@cs.utexas.edu

Mohamed G. Gouda
Department of Computer Science
University of Texas at Austin
Austin, TX, USA
gouda@cs.utexas.edu

ABSTRACT

A widely used mechanism for computing the topology of any network in the Internet is Traceroute. Using Traceroute, one simply needs to choose any two nodes in a network and then obtain the sequence of nodes that occur between these two nodes, as specified by the routing tables in these nodes. Thus, each use of Traceroute in a network produces a trace of nodes that constitute a simple path in this network. In every trace that is produced by Traceroute, each node occurs either by its unique identifier or by the anonymous identifier "*". In this paper, we introduce the first theory aimed at answering the following important question. Is there an algorithm to compute the topology of a network N from a trace set T that is produced by using Traceroute in N, assuming that each edge in N occurs in at least one trace in T, and that each node in N occurs by its unique identifier in at least one trace in T? Our theory shows that the answer to this question is "No" in general. But if N is a tree, or is an odd ring, then the answer is "Yes". On the other hand, if N is an even ring, the answer is "No", but if N is a "mostly regular" even ring, then the answer is "Yes".

Categories and Subject Descriptors

C.2.3 [**Computer-Communication Networks**]: Network Operations—*network management*

General Terms

Algorithms Theory

Keywords

traceroute, network topology, topology tracing, topology inference

1. INTRODUCTION

Traceroute is arguably the most popular mechanism for computing the topology of a network in the Internet. Executing Traceroute between any two nodes, say nodes x and y, in a network produces a sequence of nodes, called a *trace*, that corresponds to a simple path between x and y in the network. We assume that if the network has multiple simple paths between x and y, then exactly one of these paths corresponds to the trace produced from using Traceroute between x and y. Traceroute is used to compute the topology of a network N in the Internet as follows.

1. Identify the nodes that are located at the perimeter of network N. We refer to these nodes as the terminal nodes of N.

2. Execute Traceroute between every pair of distinct terminal nodes of N to produce the trace of nodes that occur between these two nodes.

3. Put all the traces, that are produced in Step 2, together to compute the topology of network N.

Unfortunately some nodes that occur in the traces produced in Step 2 appear with anonymous identifiers rather than with their unique identifiers. This feature complicates the task of putting the produced traces together to compute the topology of network N in Step 3 [1],[2], [3]. In this paper, we investigate the problem of computing a network topology from a given set of network traces where some nodes appear with anonymous identifiers. We show that the problem is unsolvable in general, and identify some interesting special cases where the problem is solvable.

2. NETWORK TRACING PROBLEM

A *network* N is a connected, undirected graph where nodes have unique identifiers. Every node in a network is designated either terminal or non-terminal. Also, every node is either regular or irregular.

A *trace* t is *generable from* a network N iff t is a sequence of nodes in N that represents a simple path between two terminal nodes in N. A regular node occurs in t by its unique identifier. An irregular node occurs in t either by its unique identifier or by the anonymous identifier $*_i$ where i is a unique integer in trace t. The first and last nodes of t occur by their unique identifiers in t.

Note that a trace t that is generable from a network N is a sequence of nodes that corresponds to a simple path in N. Thus, there are two ways to write the sequence of nodes in t. For example, t can be written as $(e, *_1, *_2, *_3, a)$, or it can be written as $(a, *_3, *_2, *_1, e)$. We regard the differences between these two ways of writing t as immaterial.

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

1. T is a set of traces, each of which is generable from N.

2. For every pair of terminal nodes x, y in N, T has at least one trace $(x, ..., y)$.

PODC'09, August 10–12, 2009, Calgary, Alberta, Canada.
ACM 978-1-60558-396-9/09/08.

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, if x and y occur in two or more traces in T, then the exact same set of nodes occur between x and y in every trace in T where both x and y occur.

Two comments concerning condition 5 in this definition, the consistency of T, are in order. First, if a trace set T has two traces of the form $(x, *_2, z)$ and (u, x, y, z), then from the consistency condition, we can conclude that node $*_2$ is in fact node y.

Second, if a trace set T has a trace of the form $(x, *_2, z)$, then from the consistency condition, T cannot have a trace of the form $(u, x, *_5, y, z)$. This is because the number of nodes between x and z in the first trace is 1, and their number in the second trace is 2, in violation of the consistency condition.

These conditions may appear extremely strong. However, note that if they are not satisfied, it is very easy to produce sets of traces which do not uniquely identify the traced network. For example, violating condition 5, we may get $\{(a, b, d), (a, c, d), (a, c, e), (a, *_1, d)\}$ - now $*_1$ can be b or c, so we have two possible networks. Our primary result is an impossibility theorem, which bounds the power of any algorithm to compute the topology of a general network from its trace set; in order to show that the result does 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 result of impossibility still holds.

The *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 network N such that T is generable from N and not from any other network."

3. SUMMARY OF RESULTS

In [4], we show that the network tracing problem is solvable for the following classes of networks: (A network is *mostly-regular* iff each node in the network has at most one irregular neighbor.)

1. tree networks
2. odd rings
3. mostly-regular even rings

We also show that the problem is not solvable for the following classes of networks:

1. networks with one (or more) irregular nodes
2. even rings
3. mostly-regular networks

From these results, it is clear that the classes of networks for which the network tracing problem is not solvable are much larger than the classes of networks for which the problem is solvable. This suggests that the network tracing problem needs to be weakened in order to make it solvable for richer classes of networks.

4. WEAK NETWORK TRACING PROBLEM

The reason that the network tracing problem is not solvable in most cases, one may argue, is that the given trace set T is required to be generable from one, and only one, network N. One may hope, then, that if this strict requirement is somewhat relaxed, then the resulting weak version of the network tracing problem becomes solvable in many cases.

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 $\{N_1, .., N_k\}$ of networks such that T is generable from each network in this set and not from any network outside this set."

The requirement that the produced set $\{N_1, .., N_k\}$ be small means, mathematically, that the cardinality k of this set is a constant, not a function of the number of unique and anonymous node identifiers in the given trace set T.

There are both practical and theoretical reasons for this requirement. From a practical point of view, the smaller the produced set of networks is, the better. From a theoretical point of view, the weak network tracing problem becomes trivially solvable if the cardinality k of the produced set is allowed to be a function of the number of unique node identifiers in the given trace set T.

Unfortunately, the following theorem shows that the weak network tracing problem is not solvable in general.

Theorem. There is no algorithm that takes as an input a trace set T that is generable from a network, and computes a set of networks $\{N_1 \dots N_k\}$ such that the following three conditions hold:

- T is generable from every network in the set $\{N_1 \dots N_k\}$.
- T is not generable from any other network.
- The value of k is not a function of the number of node identifiers in T.

5. CONCLUSION

Traceroute, while adequate for the purpose of computing the distance (measured in number of hops) between any two nodes in the Internet, is not adequate for inferring the topology of a network in the Internet.

6. REFERENCES

[1] B. Yao, R. Viswanathan, F. Chang, and D. Waddington. *Topology inference in the presence of anonymous routers.* In INFOCOM 2003, Vol. 1, pages 353-363. IEEE, 2003.

[2] X. Jin, W.-P. K. Yiu, S.-H. G. Chan, and Y. Wang. *Network topology inference based on end-to-end measurements.* IEEE Journal on Selected Areas in Communications, Vol. 24, No. 12, pages 2182-2195, 2006.

[3] M. Gunes and K. Sarac. *Resolving anonymous routers in Internet topology measurement studies.* In INFOCOM 2008, pages 1076-1084. IEEE, 2008.

[4] H. B. Acharya and M. G. Gouda. *The theory of network tracing.* Technical Report, TR-09-02. University of Texas at Austin, 2009. http://www.cs.utexas.edu/research/publications/ncstrl/ncstrl2html.cgi.