

Mapping the Internet

Arman Danesh and Ljiljana Trajkovic
Simon Fraser University
Burnaby, BC, Canada
{adanesh, ljilja}@cs.sfu.ca

Stuart H. Rubin
SPAWAR Systems Center
San Diego, CA, USA
srubin@spawar.navy.mil

Michael H. Smith
University of California
Berkeley, CA, USA
mhs@mining.ubc.ca

Abstract

Discovery of a network topology is a challenging task. Available algorithms that rely on simple hop-limited, *traceroute*-style probes give different performance in terms of the completeness (fuzziness) of the resulting map, the speed of mapping, and the efficiency of mapping. In this paper, the authors provide a brief overview of the types of mapping abstractions that have been used and review available techniques for generating maps of the Internet's infrastructure. A small study conducted in order to compare two of these techniques is described. Results of this study indicate that informed random address probing offers more complete network maps quickly and more efficiently. They also suggest that probing from multiple sources and amalgamating the results may improve the completeness of maps.

1. Internet mapping abstractions

Maps are the basis by which a society discovers and navigates its world [1]. They serve as a fuzzy metaphor: an abstraction of the world from a point of view that is generally accepted by the society in which the map is produced. Different societies may have different points of view, thus Malamud considers maps to be "consensual hallucinations." Even today's maps that are scientifically and theoretically valid representations of the real physical world still form an abstraction and, therefore, can be considered a hallucination; maps are not reality but rather allow us to interpret aspects of reality we cannot directly perceive with our senses. Maps are fuzzy in nature, with varying degrees of detail, uncertainty, and vagueness, yet still remain understandable (to varying degrees) to different users. This is the reason why maps remain important and useful.

To do a complete study of maps is beyond the scope of this paper. However, in this paper, the authors focus on techniques currently being used (mostly non-fuzzy in nature) so that the reader can better understand the issues and challenges of this field, and perhaps, use formal fuzzy techniques to solve more complex mapping problems.

With this in mind, in the context of the Internet, there are three main abstractions currently being used for mapping the network: geographic, conceptual, and infrastructural. Geographic maps, e.g., those produced by MapNet [2] or NetBoy from NDG Software [3], arise out of the need to study the network from a geographical point of view and can be used to identify sources of Internet delays and congestion and to correlate them to geographic, climatic, or other causes.

Conceptual maps have become important with the emergence of the World Wide Web. The structure of the Web is derived from its content rather than its infrastructure. Accordingly, the task of mapping the Internet in terms of the information structure is important for effective navigation of the concepts represented on the network. Meta-information is at the core of Internet mapping [1] and today's search engines and catalogues such as Yahoo! and Google are examples of attempts to produce conceptual maps of the Internet using techniques such as eigenvector analysis [4] and hyperbolic trees [5].

Infrastructural mapping is the focus of the project we describe in this paper. Infrastructural maps are undirected graphs where nodes represent routers and edges represent the links between routers. There are numerous examples how to generate infrastructural maps, including the work of Cheswick and Burch [6] and the CAIDA Skitter project [7].

It is interesting to note that these different abstractions produce notably different pictures of the Internet. When looked at in terms of infrastructure, the width of the Internet is at most 256 hops [8]; however, research has shown that the average distance between any two randomly-selected pages (a conceptual abstraction) on the World Wide Web is only 19 links [9].

Infrastructural mapping of TCP/IP networks has two main components:

- Discovery of the network topology
- Rendering of a visual graph of the data.

We are concerned with the first issue in this paper. Specifically, we investigate the available techniques to map the Internet or other large TCP/IP networks, and their comparison in terms of accuracy and efficiency.

2. Techniques for network discovery

No perfect technique is available. Fuzzy techniques do hold the promise of more sophisticated tools someday. However, currently, all the techniques now available for network discovery rely on hop-limited probes of the type used by the Unix *traceroute* utility or the Windows NT *tracert.exe* tool. *Traceroute*-style network probes follow the path that packets take from a source node to a destination node.

This technique relies on two key principals:

- Internet Protocol (IP) packets have an 8-bit Time-To-Live (TTL) header field. As a packet passes through routers on the Internet, each router decreases the TTL value by one until it reaches zero. When a router receives a packet with a TTL value of zero, it drops the packet instead of forwarding it.
- When a router drops a packet, it sends an Internet Control Message Protocol (ICMP) error message to the source node where the packet originated indicating that the packet exceeded its maximum transit time [8].

By combining these principles, *traceroute* works as follows:

1. 40-byte User Datagram Protocol (UDP) packets are sent to the target node of the probe with a TTL value of 1.
2. The first router to receive the packet drops it and returns an error to the source node.
3. The source node uses the origination point of the error to report the first router in the path to the target.
4. Another 40-byte UDP packet is sent to the target node with a TTL value of 2.
5. The second router to receive the packet drops it and returns an error to the source node.
6. The source node uses the origination point of the error to report the second router in the path to the target.
7. This cycle continues, each time with the TTL value increasing by one, until the target node responds or the maximum number of hops is reached without successfully reaching the target. If the packet reaches the target node, the target node will return an error to the source node because the packets

are destined for obscure ports that are normally not used on TCP/IP systems. This error is used by the source node to identify successful contact with the target.

Two broad categories of techniques are available for mapping the network by using *traceroute*: the basic algorithm and an intelligent heuristic. Both techniques can be used to either attempt to map the network from a single node, or to map the network from multiple nodes and amalgamate the results.

The basic method of discovering the topology of a TCP/IP network is to attempt to probe every possible IP address with *traceroute*, and to record every router reported and the adjacencies of the reported routers. Because of time limitations, most implementations use some criteria to select a subset of valid IP addresses to probe in order to obtain router data [6]. Generally, these techniques obtain routing information from a database or from selected hosts that are used as the targets of the *traceroute*-style probes.

An alternative to the basic mapping techniques is to use an intelligent heuristic [6], [10]. An intelligent heuristic for map discovery known as informed random address probing was proposed by Govindan and Tangmunarunkit [10]. This technique does not require a database of targets for exploring the network topology. Instead, it uses a heuristic to decide how to choose targets for probing. It is designed to map the network from a single source location without an initial database of target nodes for probing. The basic heuristic is as follows:

1. Whenever a response from a router is received, its network prefix is assumed to contain addressable nodes.
2. For each probe, a prefix is selected from the pool of prefixes and a target address for that prefix is randomly selected and probed.
3. If, after a preset amount of time, the pool of available prefixes has not grown, then a new prefix is added to the pool by selecting a prefix from the pool and selecting a neighboring prefix.

The heuristic uses a lottery-scheduling algorithm to select each prefix for probing from the pool of prefixes and biases selection towards recently created prefixes known to be densely populated with addressable nodes. Informed random address probing [10] creates more complete maps than it is possible with more basic techniques.

3. Methodology

The project described in this paper is designed to make a comparison of three approaches to Internet topology discovery based on the following criteria:

- **Completeness:** Which technique produces the largest map for a comparable scan of the network? The size of the map is determined by the number of nodes (routers) and edges (router connections) in the map.
- **Speed:** How long do the algorithms take to complete similar scans of the network?
- **Efficiency:** Which technique generates its map with the least redundant discovery of nodes in the map?

In addition, another question is being studied: Does increasing the maximum TTL value of *traceroute* probes improve the quality of resulting maps in terms of the completeness and does it affect efficiency and speed? It may not be necessary to probe the network with the maximum TTL value, but instead limit probes to smaller values. It is likely that there is a point of diminishing return after which increasing the TTL has minimal effect on the size of the final map even though it may significantly decrease the speed and efficiency of the mapping process. Thus, to study these issues, two algorithms have been implemented:

- The basic algorithm, using evenly spaced IP addresses from the IP address space as targets for probes.
- A variation of the informed random address probing heuristic that randomly selects prefixes for probing instead of using the lottery ticket selection algorithm. The implementation added a new prefix to the pool if the pool did not grow through 16 consecutive target probes.

The algorithms were implemented in Perl on Windows NT, and used the standard Windows NT *tracert.exe* utility to conduct the probes. An open-source Java-based graph drawing tool named VGJ was used to generate visual representations of the mapping results as illustrated in Figure 1 [11]. (VGJ uses a simple markup language called the Graph Markup Language (GML) to define nodes and edges for a graph.) Data were collected in four files for each mapping:

- A log file that shows all of the activity occurring during the mapping session.
- A file containing a list of all discovered routers.
- A file containing a list of all discovered router connections.

- A file containing the necessary GML tags to render the map using VGJ.

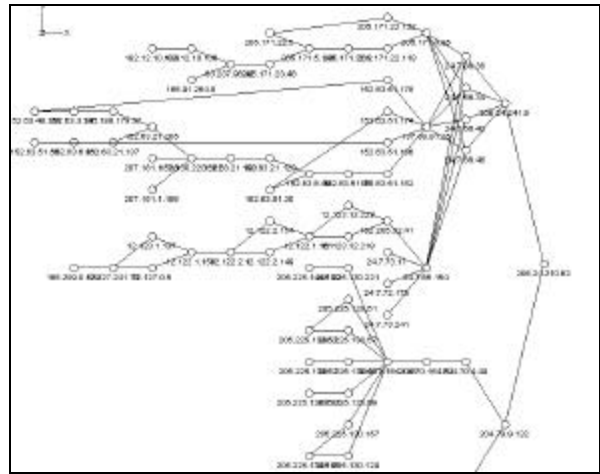


Figure 1: A map rendered using VGJ.

Three types of tests have been conducted with these implementations:

- The basic algorithm was used to probe 256 evenly spaced IP addresses. The probes were run from a computer connected to the Rogers @Home cable Internet service with TTL values of 8, 15, and 30 and were compared to study the effect of TTL values on the completeness, speed, and efficiency of the mapping.
- The basic algorithm was used to probe 256 evenly-spaced IP address with a TTL of 15 from a computer connected to the Rogers @Home cable Internet service and from a computer on the Internet Gateway network in downtown Vancouver. These data were amalgamated and compared with the data from the scan from the Internet Gateway network to study the impact of scanning from multiple nodes on the completeness, speed, and efficiency of the mapping.
- The informed random address probing heuristic was used to probe 256 random addresses with a TTL of 15 from a computer connected to the Internet Gateway network. The results were compared with the basic algorithm with a TTL value of 15. This analysis compared the approaches in terms of the completeness, speed, and efficiency of the mapping.

While probes of 256 IP addresses with a limit of TTL values of 30 will not generate comprehensive or conclusive Internet maps, it is expected that the results

will be complete enough to be valid for comparison purposes. While it would have been useful to perform mapping with a broader range of IP address targets or with deeper probes (even up to the maximum TTL of 255), limited resources prevented this. Even with the limited nature of the probes being performed, the longest took in excess of 12 hours and without access to dedicated machines, which could be left unattended for the purpose of running these tests, the length of each testing session had to be necessarily constrained.

4. Test results

Results of all scans were analyzed using a Perl script that extracted data from the output files of each scan described in the previous section.

This statistics output contains the following results:

- *Number of nodes discovered*: the number of routers uncovered in the mapping session.
- *Number of edges discovered*: the number of router connections uncovered in the mapping session.
- *Number of targets probed*: The number of target nodes probed with *traceroute*.
- *Number of hosts probed*: The number of routers listed in the *traceroute* results.
- *Number of nodes probed X time(s)*: The number of routers listed the specified number of times in the *traceroute* results.

Several interesting results emerged from analysis of the data obtained in the study. They are discussed in the remainder of this section.

4.1 Comparing the effect of TTL values

Table 1 outlines the basic results of probing 256 evenly spaced IP addresses using the basic algorithm with TTL values of 8, 15, and 30. All probing was done from a host connected to the Rogers @Home cable Internet network.

Analysis of this data indicates the following:

- There appears to be a minimum threshold for TTL value in order to obtain useful results. When the TTL value is 8, the scan is effectively useless (only 15 routers discovered) which suggests that all routers probed are on the Rogers @Home network or close to it. The probes do not go deep enough to get a broad picture of the Internet. By almost doubling the TTL value to 15, the effect is a five-fold increase in the number of nodes discovered.

- There appears to be a point of diminishing returns somewhere between TTL values of 15 and 30. Doubling the TTL value from 15 to 30 does not double the number of nodes discovered (which only increase from 75 to 100), nor does it double the number of edges discovered (which increases from 91 to 125). In addition, the speed results indicate that the TTL value of 15 produces the quickest scan in terms of the number of nodes discovered per minute. This indicates that as TTL values increase, the mapping process spends more time waiting for timeouts to occur because the relatively high TTL value forces most probes to probe further than the target.

The statistics for the amount of redundant discovery taking place indicate several results:

- TTL values of 15 and 30 appear to produce similar levels of efficiency. For instance, the largest number of nodes was discovered 16 times for TTL values of 15 (30 nodes) and 30 (31 nodes).
- For TTL values of 15, the largest percentage (20%) of nodes was successfully discovered non-redundantly (discovered only once). For TTL values 30, only 13% are discovered non-redundantly.

Behavior is anomalous for TTL values of 8: no routers are discovered non-redundantly. At a minimum, all routers are discovered 8 times.

	TTL=8	TTL=15	TTL=30
Number of nodes discovered	15	75	100
Number of edges discovered	22	91	125
Number of targets probed	256	256	256
Number of hosts probed	895	1782	2187
Time to complete probe	3:13	5:49	12:23
Nodes discovered per minute	0.078	0.215	0.135

Table 1: Effects of TTL values on the basic algorithm.

4.2 Comparing probes from different locations

Table 2 outlines the basic results of probing 256 evenly spaced IP addresses using the basic algorithm with a TTL value of 15. These probes were run from a single host connected to Internet Gateway and from a single host on the Rogers @Home network.

Analysis of this data indicates the following:

- Results are similar in terms of the number of nodes and connections discovered. This indicates that the source location of probe does not significantly affect the degree of discovery.
- The speed is significantly different both in terms of total time to complete the probe and the number of nodes discovered per minute. This may reflect differences in connection speed, system speed, or differences between locations on the Internet.

The statistics for the amount of redundant discovery taking place indicate that the Internet Gateway probe is more efficient, discovering more than twice as many routers non-redundantly (14 versus 6). This likely reflects differences in how close the two systems are to their networks' respective backbone connections.

	Internet Gateway	Rogers @Home
Number of nodes discovered	72	75
Number of edges discovered	82	91
Number of targets probed	256	256
Number of hosts probed	2785	1782
Time to complete probe	3:27	5:49
Nodes discovered per minute	0.348	0.215

Table 2: Effects of source location on the basic algorithm.

4.3 Comparing single-source and multi-source mapping

Table 3 outlines the results of probing 256 evenly spaced IP addresses using the basic algorithm with a TTL value of 15. These probes were run from a single host connected to the Internet Gateway network. The results are amalgamated with the results of the same probe run from two distributed nodes on the Internet (connected to Rogers @Home and the Internet Gateway network in Vancouver).

Analysis of this data indicates the following:

- Probing from two sources has a significant effect on the number of nodes discovered (an increase of 76%) and the number of connections discovered (an increase of 90%).
- The speed of discovery in terms of the number of nodes discovered per minute increases by 46%.

The statistics for the amount of redundant discovery taking place indicate that efficiency decreases when probing from multiple sources:

- When probing from a single source, 68% of nodes were discovered redundantly 5 or more times. When probing from multiple sources, this number increases to 83% of nodes being discovered redundantly 5 or more times.
- When probing from a single source, 19% of nodes were non-redundantly discovered. When probing from multiple sources, this number drops to only 6%.

	Single source	Multi-source
Number of nodes discovered	72	127
Number of edges discovered	82	156
Number of targets probed	256	256
Number of hosts probed	2785	4568
Time to complete probe	3:27	5:49
Nodes discovered per minute	0.348	0.510

Table 3: Comparison of single-source and multi-source mappings on the basic algorithm.

4.4 Comparing the basic algorithm and informed random address probing

Table 4 outlines the basic results of probing 256 evenly spaced IP addresses using the basic algorithm with a TTL value of 15. These probes were run from a single host connected to the Internet Gateway network and were compared with the results of probing 256 addresses with a TTL value of 15 using informed random address probing from the same host.

Analysis of this data indicates the following:

- There is a significant improvement in the number of nodes and connections discovered when switching from the basic algorithm to informed random address probing. There is a 67% and 55% increase, respectively.
- The increase in the number of nodes and connections discovered is similar to the increase when switching from single-source mapping to multiple-source mapping with the basic algorithm. However, the speed increase (from 0.348 to 0.538 nodes discovered per minute) in the case of informed random address probing was reduced when switching to multiple-source mapping.

The statistics for the amount of redundant discovery taking place indicate a significant increase in efficiency when switching to informed random address probing. Specifically, when probing with the basic algorithm, 68% of nodes were discovered redundantly 5 or more times. When probing with informed random address probing, this number drops to 41%.

	Basic algorithm	Informed random
Number of nodes discovered	72	120
Number of edges discovered	82	127
Number of targets probed	256	256
Number of hosts probed	2785	2109
Time to complete probe	3:27	3:43
Nodes discovered per minute	0.348	0.538

Table 4: Comparison of the basic and informed random address probing algorithms.

5. Discussion and future work

The results outlined in Section 4 lead to several observations and avenues for future work:

- The selection of the appropriate maximum TTL value for probes is an important factor in speeding up the mapping process while maintaining efficiency. The authors feel that fuzzy techniques hold great promise in potentially being able to select the appropriate TTL values, but further research is needed. Also, further research needs to be done regarding the depth of probing (as determined by TTL values) to see if this result is anomalous or consistent regardless of the set of IP addresses used as targets or the algorithm used for mapping.
- When the TTL value increases above 15 towards 30, informed random address probing does not show as noticeable decrease in completeness and efficiency as the basic algorithm. Further experiments are needed to study the effect of TTL values on this algorithm.
- Probing from multiple sources appears to offer diminishing returns. Doubling the number of sources does not double the completeness or speed of mapping. However, because multiple source probing does appear to improve completeness, it would be interesting to investigate the results of performing informed

random address probing from more than two distant locations.

- Informed random address probing offers better completeness, speed and efficiency than the basic algorithm from both single and multiple sources.
- Test runs with informed random address probing revealed that the algorithm for selecting targets to probe affected map completeness. At times, probing fewer targets produced more complete maps than probing more targets. This indicates that the random selection of prefixes to probe and the random selection of targets within a prefix network lead to high variability in results. Future studies may consider prefix selection algorithms that include random, lottery ticket [2], round-robin, knowledge-based, and fuzzy selection algorithms. When heuristics are employed to make a computational decision there are tradeoffs to be considered. One must consider the value of what is being decided vs. the cost/benefit of using a more informed heuristic. In the final analysis, heuristics need to be computationally fast and economical of storage usage. This simple heuristic needs to be kept in mind for design purposes.

6. Conclusion

Maps, many fuzzy in nature, are difficult to construct. In order for fuzzy researchers to understand mapping better, in this paper we describe a small study and compare two current techniques (basic algorithm and informed random address probing) to discover TCP/IP network topologies. The results of this study indicate that informed random address probing offers better completeness and efficiency than the basic algorithm. It is hoped that the case study of these two techniques helps researchers better develop and apply fuzzy techniques to complex real-world mapping problems.

7. References

- [1] C. Malamud, "A shared reality," <http://mappa.mundi.net/cartography/Maps/>.
- [2] K. Claffy, and B. Huffaker, Macroscopic Internet visualization and measurement, <http://www.caida.org/tools/visualization/mapnet/>.
- [3] NetBoy, <http://www.ndgsoftware.com/>.
- [4] Google, <http://www.google.com/>.
- [5] Inxight: making information makes sense, <http://www.inxight.com/>.
- [6] B. Cheswick and H. Burch, "Internet mapping project," <http://www.cs.bell-labs.com/who/ches/map/>.
- [7] Skitter, <http://www.caida.org/tools/measurement/skitter/>.

- [8] J. Rickard, "Mapping the Internet with traceroute," <http://boardwatch.internet.com/mag/96/dec/bwm38.html>
- [9] R. Albert, H. Jeong, and A. Barabasi, "Diameter of the World-Wide Web," in *Nature*, No. 401, 9 Sept. 1999, pp 130-131. Macmillan Publishers Ltd.
- [10] R. Govindan and H. Tangmunarunkit, "Heuristics for Internet map discovery," <ftp://ftp.usc.edu/pub/csinfo/tech-reports/papers/99-717.ps.Z>.
- [11] Drawing graphs with VGJ, http://www.eng.auburn.edu/departement/cse/research/graph_drawing/graph_drawing.html.