Using Graph Theoretical Methods and Traceroute to Visually Represent Hidden Networks

Jordan Sahs

University of Nebraska at Omaha, Omaha, NE.

Abstract

Within the scope of a Wide Area Network (WAN), a large geographical communication network in which a collection of networking devices communicate data to each other, an example being the spanning communication network, known as the Internet, around continents. Within WANs exists a collection of Routers that transfer network packets to other devices. An issue pertinent to WANs is their immeasurable size and density, as we are not sure of the amount, or the scope, of all the devices that exists within the network. By tracing the routes and transits of data that traverses within the WAN, we can identify routers and create both the paths and weights between devices that are communicating. However, there is the issue of hidden routers who transfer data but do not identify themselves to identification requests like Traceroute, and the undocumented edges between Routers. Like a blackbox function that outputs data in a way that we do not know the interior mechanics of the function, we do not know all the internal components that manage the traffic within the WAN. Finding out is called the Anonymous Routing Blackbox Problem, and we will use labelled graphs, vertex and edge coloring, and pathfinding to derive solutions.

1) Introduction

Anonymous Routers are defined to be routers who remain undiscovered when tracing communication networking routes with applications like Traceroute. Traceroute is a network diagnostics tool in which a series of ping requests are sent through a network across a series of devices until they reach the intended domain. The requests identify all devices in-between the domain, usually by an Internet Protocol (IP) Address, and the originating device of the Traceroute requests. We will be using graphs, specifically a few kinds of graphs called trees, cycles, and paths, with graph coloring to document this behavior.

Transmission mediums between devices (Ethernet, wireless, coaxial, etc) are denoted as edges. Black colored edges denote confirmed connections where anonymous devices do not exist in-between both vertices, and red edges denote where it is possible for hidden devices to exist. Vertices denote devices, with routers being a subcategory of devices. However, by using a labelled graph, and a controlled source vertex with a sink vertex into which the network packets will start and end on, and intermixing the graph with two contrasting colors will help make the distinctions apparent.

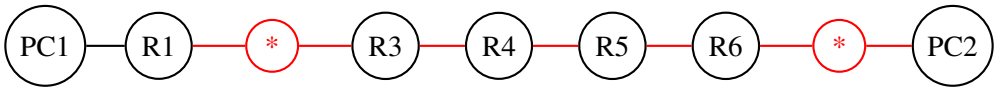

Figure 1: Concept path graph of Anonymous Routing behaviors.

Figure 1 is an example of a path graph and that is marked-up with coloring and labels that we will define as an Anonymized Graph. Red denotes unconfirmed routers and connections, while asterisks denote anonymous routers. Traceroute, Address Resolution Protocol (ARP) Tables, IP address comparison, and M.A.C. address comparisons will be used, and explained, to confirm connections that will further model these constructed anonymized graphs. Lastly, in this paper we will be using the term PC (Personal Computer) to denote a user controlled node, but note that is an arbitrary label and the device does not have to be a personal computer.

2) Single Node Population via a Second Domain

Using a single node, populate an anonymized graph from a WAN as follows:

To start, assume there is a connection from the source node to an outside device of the WAN for out-of-network communication. This device is not controlled or monitored. Traceroute the domain as denoted in Figure 2 (with R2 being the assumed domain):

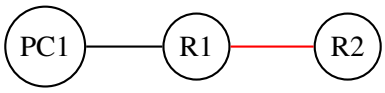

Figure 2: Concept path graph of start of Single Node Population.

Notice that the connecting edge from the controlled source PC (PC1) to the assumed domain (R1) is guaranteed, as this is the first possible outbound network connection in PC1's environment and thus is assumed to have no hidden elements. From here, PC1 must request an ARP table (a data table within routers that contains IP and M.A.C. addresses of devices it frequently communicates with) from R1 so that PC1 can populate a graph by adding nodes based on the devices from the ARP table's data as such:

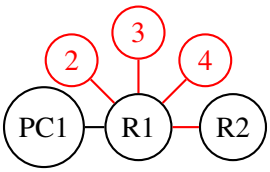

Figure 3: First Iteration tree graph of Single Node Population via a Second Domain.

Notice that each ARP discovered device, and each of its connections, are red as PC1 does not know if the devices do or do not exist. Thus, PC1 must ping devices two through five and individually confirm their existences in the network. The notation for these devices are arbitrary, we can use whatever labeling we like to denote non-PC and non-routing devices.

If a another router is discovered and confirmed, its ARP table must be matched with the data from existing devices in the graph so that multiple connections between devices can be found. An example:

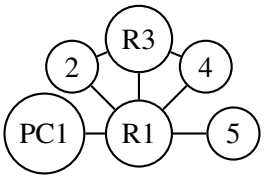

Figure 4: Completed Iteration of Single Node Populated Graph with two Routers.

This process repeats until all possible devices are found in the WAN, or the program's discovery limit is reached. We will explore the situation in which we do not assume a second domain in the next section.

Steps of Single Node Population via a Second Domain:

(1) Request an ARP Table from the assumed out-of-network connection.
(2) Ping each ARP'd device to confirm existence.
(3) Identify the category of each confirmed device (Ie. is it a router, switch, or none of those?)
(4) Match ARP tables of found Routers
(5) Repeat (2) through (4) until discovery limit is reached or no more devices can be found.

3) Single Node Population via Network Scanning

Starting from the controlled PC, scan all available networking ports for all available devices and denote found routers as such:

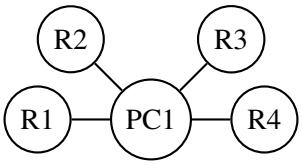

Figure 5: First Iteration of Single Node Population via Network Scanning.

For each found router, request ARP tables so that we can populate the graph further. Continue populating until there are no devices left to be found, or until the predefined iteration count limit is reached.

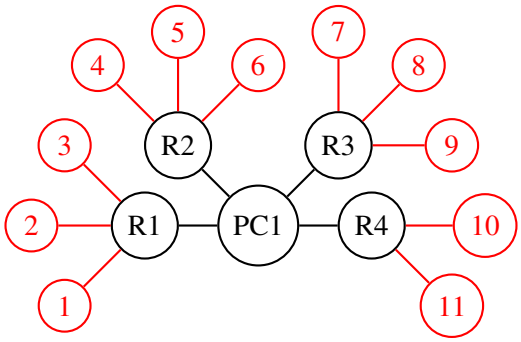

Figure 6: First ARP'd Iteration of Single Node Population via Network Scanning.

Like we have done before, ping and retrieve the ARP of each device in the anonymized graph to confirm devices' existence such that for the third iteration, we can populate the graph further. This is demonstrated in Figure 7.

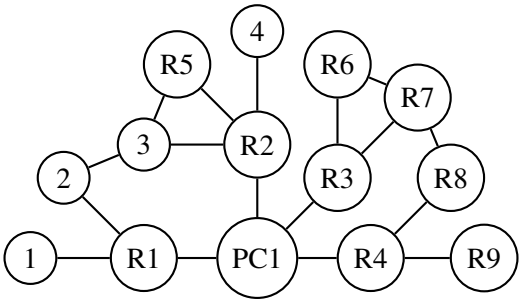

Figure 7: 2nd ARP'd Iteration of Single Node Population via Network Scanning.

Steps of Single Node Population via Network Scanning:

(1) Scan all available networking ports/connections available from the source node.
(2) Request an ARP Table from each found router.
(3) Ping each ARP'd device to confirm existence.
(4) Identify the purpose of each confirmed device.
(5) Match ARP tables of confirmed found Routers
(6) Repeat (2) through (5) until discovery limit is reached or no more devices can be found.

4) Double Node Population

Using two controlled nodes, populate an anonymized graph from a WAN as follows:

To start, pick which initial node is to be a source vertex and the other is then by default a sink vertex for Traceroute requests. For any predefined, multiple amounts of delay, assume there is a high chance that an anonymous router must exist there. Denote these ”dead” pings as an anonymous router and denote them with a red asterisk nodes on the graph. Any Router that replies back to the Traceroute request with their IP address is confirmed to exist. An example of a first iteration of a Double Node Populated graph:

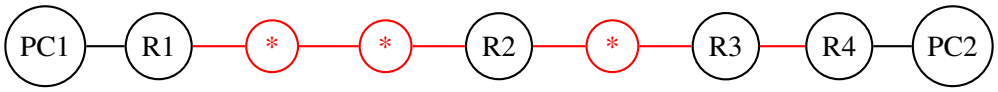

Figure 8: First Iteration example of a Double Node Populated Graph.

Next, starting from the chosen source node, request an ARP table from each confirmed Router and populate the graph with the ARP’d devices as such:

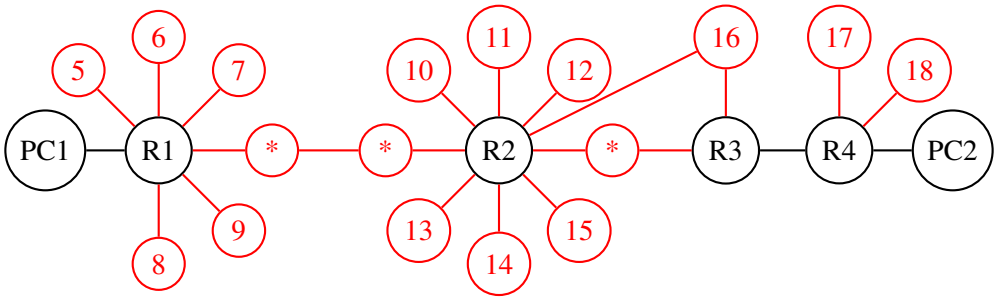

Figure 9: ARP Populated First Iteration.

In Figure 8, take notice of the edges that exist in-between R2 and R3: when we collect all ARP’d devices we must match all identical devices that overlap between the two routers by comparing the Media Access Control (MAC) address of each device (which is manufactured into, and is specifically only for that device.) While it may be assumed that node 16 is possibly the asterisk connection in-between R2 and R3, we must ping node 16 and check if it rejects our ping request. If it does, it is then the anonymous router and is thus the router in-between R2 and R3.

Steps of Double Node Population:

(1) Traceroute from source PC vertex to another PC vertex and populate the graph.
(2) Request ARP Tables of each found Router and populate the graph.
(3) Ping each ARP’d device to confirm existence.
(4) Identify the purpose of each confirmed device.
(5) Request and match ARP tables of confirmed found Routers.
(6) Repeat (2) through (5) until discovery limit is reached or no more devices can be found.

5) N Node Population

Assume the total number of available nodes that you control (PCs) is $n \geq 2$.

Doing as we did for Double Node Population, we assign one node to be the source of all Traceroute requests and we have all other initial nodes be the sinks for all the Traceroute requests. This is demonstrated in Figure 9 with PC1 being the source of all Traceroute requests and all other PCs are the sinks for the requests.

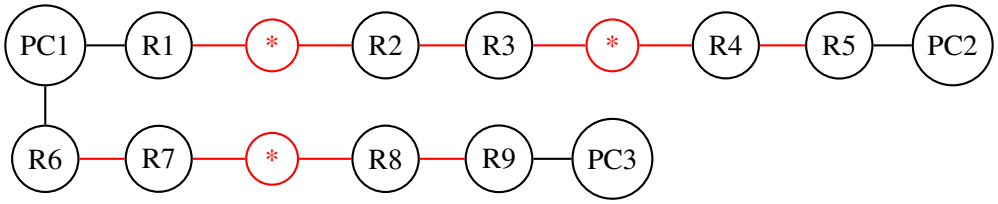

Figure 10: First Iteration example of a N Node Populated Graph (with $n = 3$).

Like what has been done before with both Single and Double Node Population, we can further populate Figure 10 with ARP table collection matching as shown in Figure 11. This process can be abstracted to take as many controlled nodes as possible. Next, we will now explore other population techniques with which we can populate these anonymized graphs of WANs further in the next section.

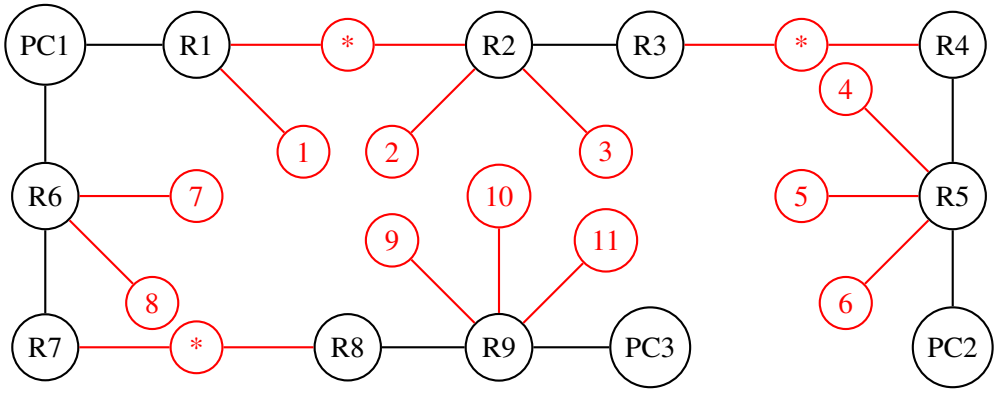

Figure 11: ARP Populated First Iteration of a N Node Graph (with $n = 3$).

Steps of N Node Population:

(1) Traceroute from each controlled vertex to all other vertices and populate the graph.

(2) Request ARP Tables of each found Router and populate the graph.

(3) Ping each ARP'd device to confirm existence.

(4) Request and match ARP tables of found Routers.

(5) Repeat (2) through (4) until discovery limit maxed or no more devices can be found.

6) Multi-Traceroute Pathing

Let us take another look at N Node Populations where we instead kept requesting Traceroutes instead of the one. In the context N Node Population with $n = 2$, we can populate the graph further by more Traceroute operations before we start the ARP table collection stage. Figure 12 depicts three Traceroute operations that are an example of this behavior.

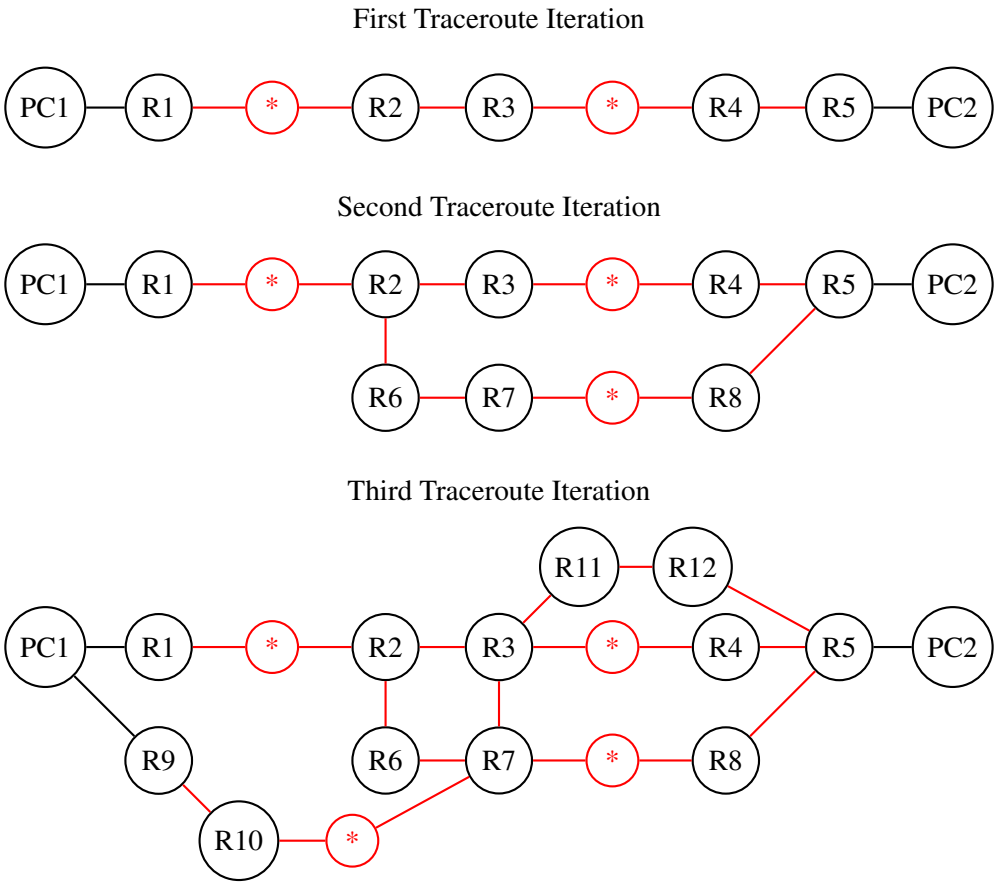

Figure 12: Double Node Population Graph with three iterations of Traceroute.

The additional routes are determined by the availability of routers in transit for the Traceroute data. These aspects are determined by pathfinding behaviors and protocols inherently built-in to routing architecture, and the latency of the transmission mediums at the time the Traceroute request was sent. This allows us to find additional routers by waiting additional time after the first Traceroute request so that traffic congestion forces our requests to path-find another route to the destination device.

Steps for implementing Multi-Traceroute before populations:

(1) If $n = 2$, then Traceroute the sink, controlled node until traceroute iteration limit is reached.

(2) If $n > 2$, then Traceroute each sink, controlled nodes until traceroute iteration limit is reached.

7) Adding Weights and Pathfinding Anonymized Graphs

Figure 13: Microsoft's Windows 10 Traceroute.

In Figure 13, we see the output of Traceroute in Microsoft's Windows 10 Command Line prompt. Denote that the source device is not listed on this results page, but this figure shows that from the source device it took 10 hops to reach the domain. In other words, the Traceroute request pings visited nine devices and then reached the domain. This can be denoted as a Single Node Populated graph since we did not have control over the domain device for Figure 13. Lastly, the three millisecond sections after the hop counter denotes the average round-trip-time each Traceroute packet took to get to that router back to the source device. Windows Traceroute sends three requests at a time, so that is why there are three columns. We can take the average of all three trips to determine the latency of that route as such:

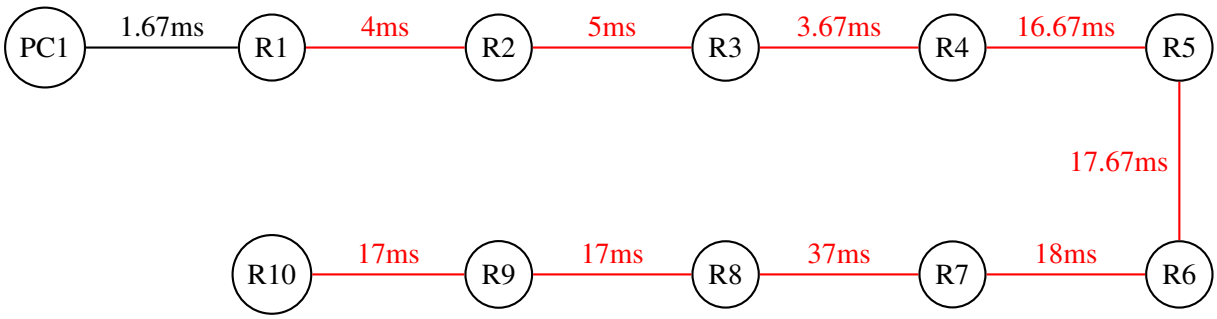

Figure 14: Anonymized Graph of Figure 13.

Pathfinding a route to any node in Figure 14 is simple, as it is already a path graph of 11 nodes. For larger graphs like Figures 11 and 12, we can add weights as we iterate through the construction of the anonymized graph. The applications for this ranges from optimizing packet traffic to avoid network congestion, monitoring traffic behaviors such that we can influence network traffic going through certain routes to take alternative paths, and finding optimal pathways such that the source and destination devices can communicate to each other on the most efficient path.

8) Conclusion

We have shown how to populate WANs with Single, Double, and N source nodes. In addition, we have explored the process of expanding the anonymized graphs with the multi-Traceroute technique. Throughout the process of solving the Anonymous Blackbox Problem, it was shown that we can add weights to the graphs such that we can optimize optimal paths to each node in the anonymized graph. Moving forward with these concepts and techniques, we can even possibly demonstrate that anonymized graphs can be used to geographically map all devices within a network contained in certain physical region. This paper explored and solved a particular problem of network scanning, and we have demonstrated that we can find anonymous routers and document a Wide Area Network in process.

Provided below is Pseudocode that can be used to construct a software application that can document, and locate, anonymous routers in a communication networks using the techniques shown in this paper. Do note that this paper assumed the generally understood definition of Traceroute, but know that the Traceroute software is dependent on how the developer constructs its methodology of pinging domains, and its data collection along the way. Not all Traceroute software is the same, and thus the developer may be better off designing a Traceroute of their own.

Pseudocode

Let n denote the number of starting nodes in the WAN with $n \neq 0$.

If $n = 1$ (Single Node Populating), then:
- From the single source device collect its own ARP table.

Else if $n \geq 2$ (N Node Populating), then:
- From source device, Traceroute all other sink devices and markdown all found Routers.

Populate the graph with found devices.

While number of iterations or WAN is populated and complete:
- Collect ARP tables of all found Routers
- Populate the graph with found devices.
- Exclude ARP'd Routers from next iterations.
- Ping all ARP'd found devices to confirm existences.
- Make Traceroute requests that take different paths in the WAN.

End Pseudo Code.

9) References

Acharya H.B., Gouda M.G. (2009) A Theory of Network Tracing. In: Guerraoui R., Petit F. (eds) Stabilization, Safety, and Security of Distributed Systems. SSS 2009. Lecture Notes in Computer Science, vol 5873. Springer, Berlin, Heidelberg, https://doi.org/10.1007/978-3-642-05118-0_5

Almog, Anat & Goldberger, Jacob & Shavitt, Yuval. (2009). Unifying Unknown Nodes in the Internet Graph Using Semisupervised Spectral Clustering. Proceedings - IEEE International Conference on Data Mining Workshops, ICDM Workshops 2008. 174 - 183. 10.1109/ICDMW.2008.12.

B. Holbert, S. Tati, S. Silvestri, T. F. La Porta and A. Swami, 'Network Topology Inference With Partial Information,' in IEEE Transactions on Network and Service Management, vol. 12, no. 3, pp. 406-419, Sept. 2015, doi: 10.1109/TNSM.2015.2451032.

Du, Jiang & Li, Yu. (2013). A Solution for Anonymous Routers Discovery Based on Source-Routing Traceroute. Advanced Materials Research. 765-767. 10.2991/isccca.2013.166.

Marchetta, Pietro & Montieri, Antonio & Persico, Valerio & Pescapè, Antonio & Cunha, Italo & Katz-Bassett, Ethan. (2016). How and how much traceroute confuses our understanding of network paths. 1-7. 10.1109/LANMAN.2016.7548847.

M. H. Gunes and K. Sarac, 'Resolving Anonymous Routers in Internet Topology Measurement Studies,' IEEE INFOCOM 2008 - The 27th Conference on Computer Communications, Phoenix, AZ, 2008, pp. 1076-1084, doi: 10.1109/INFOCOM.2008.162.