Distanced-Based Routing: Fast Anonymity for the Internet

12021/9/22

Abstract

Distance-Based Routing (DBR) is a peer-to-peer network routing protocol that aims to obfuscate all connections between computers using the protocol. DBR uses onion routing similar to TOR but the method of selecting intermediate nodes is much more flexible as they can be chosen from anywhere in the network. Unlike the Internet, DBR does not have concentrated points of failure and the self-organization property of DBR allows for flexibility and limitless scalability while still maintaining a unique way to address every node in the network. Instead of IP addresses, nodes in DBR assign themselves points in a virtual space that represent real-world latencies between computers. This allows for high optimization in routing data along the most efficient path through the network.

Introduction

Since the early days of the Internet, the main way to browse while hiding your own public IP address has been to use a proxy. This is what Virtual Private Networks (VPNs) do: they route internet traffic through other computers on the Internet to hide your IP address. However, while the destination server may not be able to know who is visiting it by correlating requests with specific IP addresses, a VPN can still match packets sent with your IP address, since all traffic is routed through it. This fundamental leakage problem with proxies is what the TOR project aims to solve. TOR (The Onion Router) creates a network of proxies through which packets are routed in such a way that no proxy can simultaneously know both the origin of a packet and its destination. TOR accomplishes this with “onion routing”: wrapping packets in layers of encryption so that each intermediate proxy can only decrypt the outermost layer and forward the packet to the next proxy. Subsequent proxies then decrypt the next layers until all the layers are unwrapped and the packet reaches its intended destination. (Onion Routing) This structure solves the privacy problem with VPNs because each intermediate node only knows at most the IP address of the sender *or* the IP address of the destination and not both at the same time. See the diagram below.

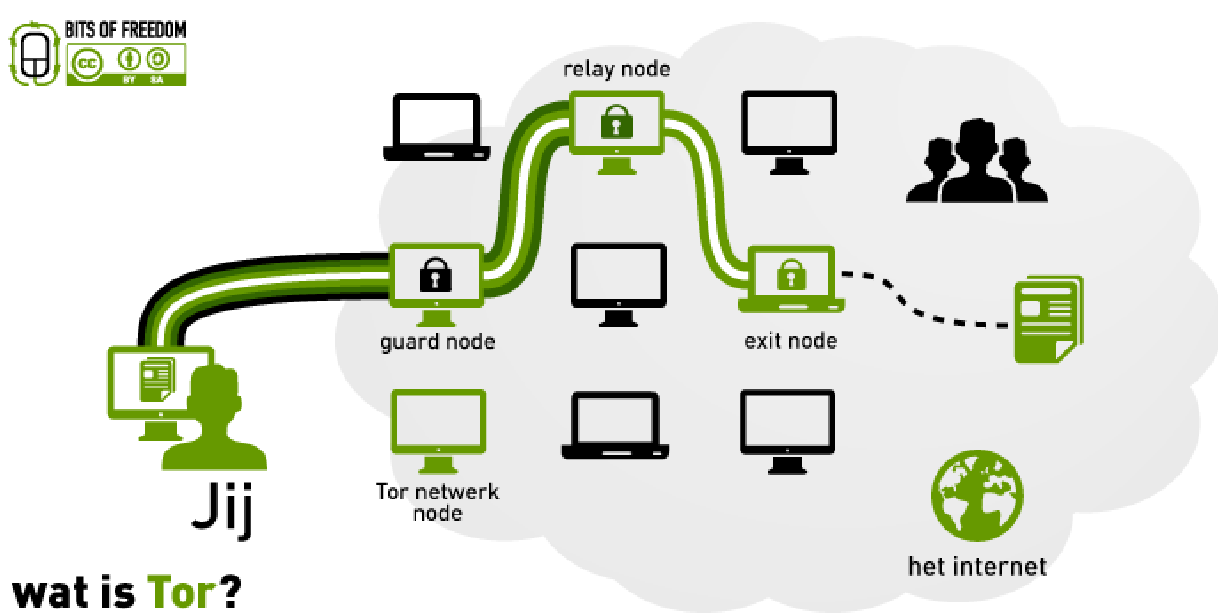

Diagram 1 (O'Sullivan)

Onion routing is excellent at obscuring connections through a network of proxies. However, TOR's implementation of proxy selection is slow. It has no concept of determining which series of proxies is most efficient to route data through. For TOR, proxies are chosen at random from a central database (Srivathsav). This means that packets routed through TOR can travel to multiple locations around the world before reaching their destination, resulting in a slow, low-fidelity connection. This paper aims to solve TOR's inefficiency by replacing the random proxy selection protocol used by TOR with an alternative strategy that is structurally aware of the time it takes for information to travel across the Internet, exponentially increasing the speed for large networks. I call this strategy "Distanced-based routing".

Distance-based routing (or DBR) aims to solve the slow nature of random proxy selection with two symbiotic strategies. The first strategy is self-organization where nodes continually search for other nearby nodes and organize themselves into the network. By knowing the virtual distance (i.e. network latency) to its closest nodes, the node can then use the second strategy of relative space calculation to compute with high precision exactly where it is in the network relative to all the other nodes. This relative distance information can then be used to plot the nodes in euclidean space (2D or 3D) replacing IP addresses with virtual distance coordinates. Using these coordinates, any node can traverse pieces of data (packets) through the other nodes in the network along the shortest path. Compared to IP addresses which have no notion of direction, these "routing coordinates" are more useful when creating a network. In current implementations of onion routing like TOR and I2P, intermediate proxies are randomly selected and can be anywhere on the planet. (I2P Developers) Using DBR, intermediate proxies are selected from computers in-between the sender and the receiver in virtual space greatly increasing network efficiency while still allowing for the obfuscation of computers participating in the network. The hypothesis for this project is that DBR will increase speed when establishing onion-routed connections compared to the method TOR uses.

Procedure

Implementing DBR in reality would be a massive undertaking requiring extensive computer resources around the world. Instead, a sophisticated simulation was programmed (https://github.com/libdither/dbr-sim) to model how the Internet works on a small scale by facilitating packet transfer between simulated nodes. Using this model, nodes can be programmed with the two different strategies of random routing and distance-based routing to compare them.

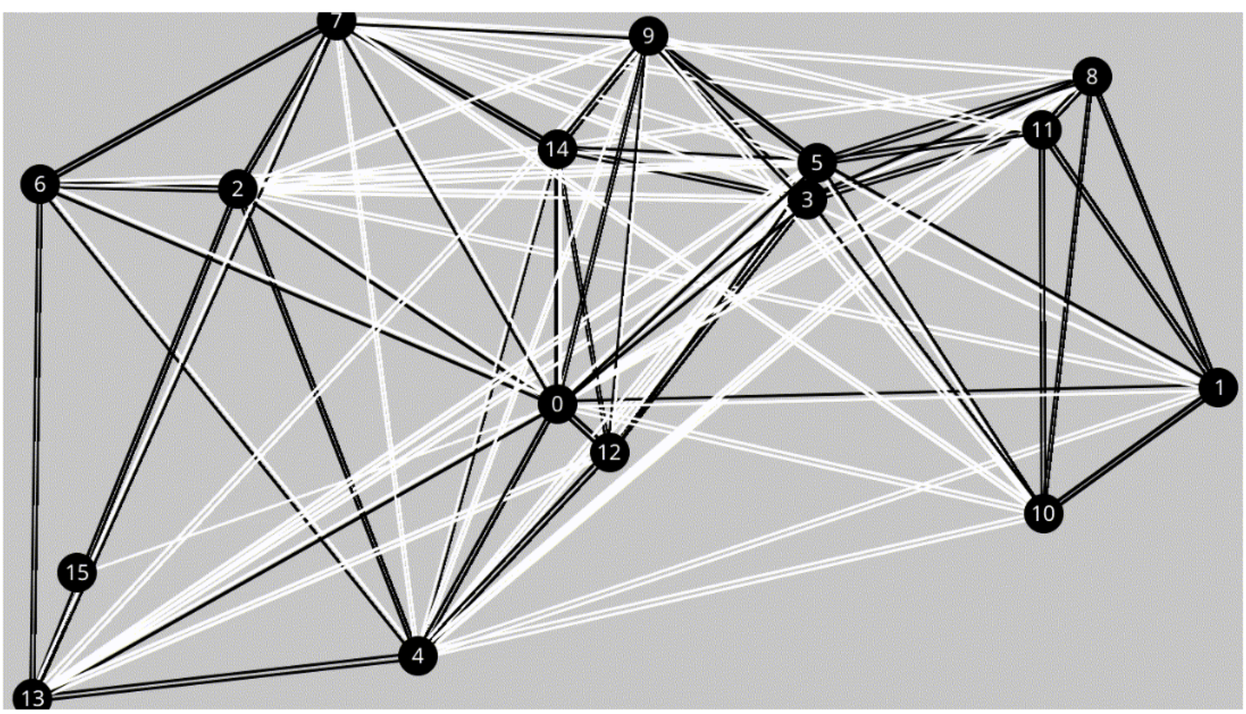

Figure 1

This is a visualization of DBR's self-organization strategy. As new nodes join the network, they recursively try to find closer and closer existing nodes. To join the network, nodes must first know the public IP address of a node already on the network. Then a direct connection can be made where a pattern of packets are passed back and forth. First, the new node and the existing node exchange information about each other such as their current routing coordinate (if they have one), how many peers they are connected to, and what each node thinks the distance between them is. Then, if the existing node has at least one other peer, the new node will send a ping request to the existing node which the existing node receives and notifies their peers that there is a new node that wants new connections. Through this mechanism new nodes on the network can discover existing nodes. As a new node discovers additional existing nodes, it will prefer to request pings from nodes closer to it virtually which will eventually result in the new node finding the closest set of nodes which it will establish as peers.

Once a new node has found enough peers closest to them, it can calculate routing coordinates. The process for this is complicated and not tested in the simulation, but it works like this: A given node has direct connections to other nodes close to it. It also knows the distance to all of its peers and their routing coordinates. Using this information the node can approximate a routing coordinate that plots it in a node space relative to all its peers, so that the measured virtual distances are as correct as possible. One possible technique that can be used for this approximation is multidimensional scaling (Wickelmaier). However, I ran into some issues with the math when trying to implement it, so I worked around it in my model simulation.

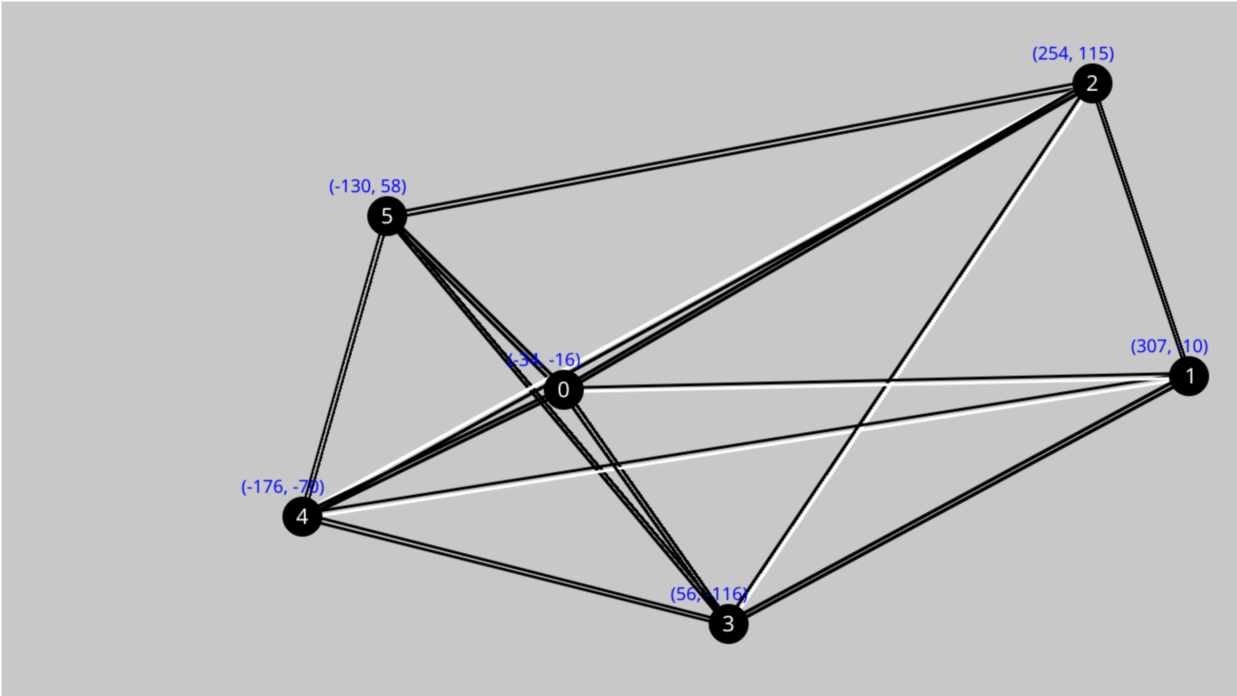

Figure 3

When the network is organized, it can do packet traversal. An example of packet traversal is presented in the above figure. In this example, Node 5 wants to establish a connection with Node 1. Since Node 5 doesn't directly connect to Node 1 (notice, no line in the figure), it will have to traverse through other nodes in the network to get there. With DBR, Node 5 knows Node 1's Routing Coordinates (marked in blue), and can calculate that the closest path to Node 1 will be through Node 2. In this case, Node 5 will attempt to connect to Node 1 through Node 2's Routing Coordinates since they are closest to Node 1. If Node 2 did not have a direct connection to Node 1 it would forward the packet to another node that is closer to Node 1 until it reaches a node that is directly connected to Node 1.

Using this protocol, packets can traverse across large spans of the network without the receiver or sender knowing each other's rough real-world location (unlike on the Internet where IP addresses can be matched with places on Earth). When coupled with onion routing, this process can be made even more secure without compromising efficiency.

Results

Does DBR work? Using the simulation, I established connections between two randomly chosen nodes using both strategies. For the first sample, I used the DBR traversal protocol. For the second sample, I chose two additional random nodes and measured the simulated latency between each one in sequence to simulate the speed that one would get using TOR. Each sample had 97 measurements, which I compiled into the two graphs below.

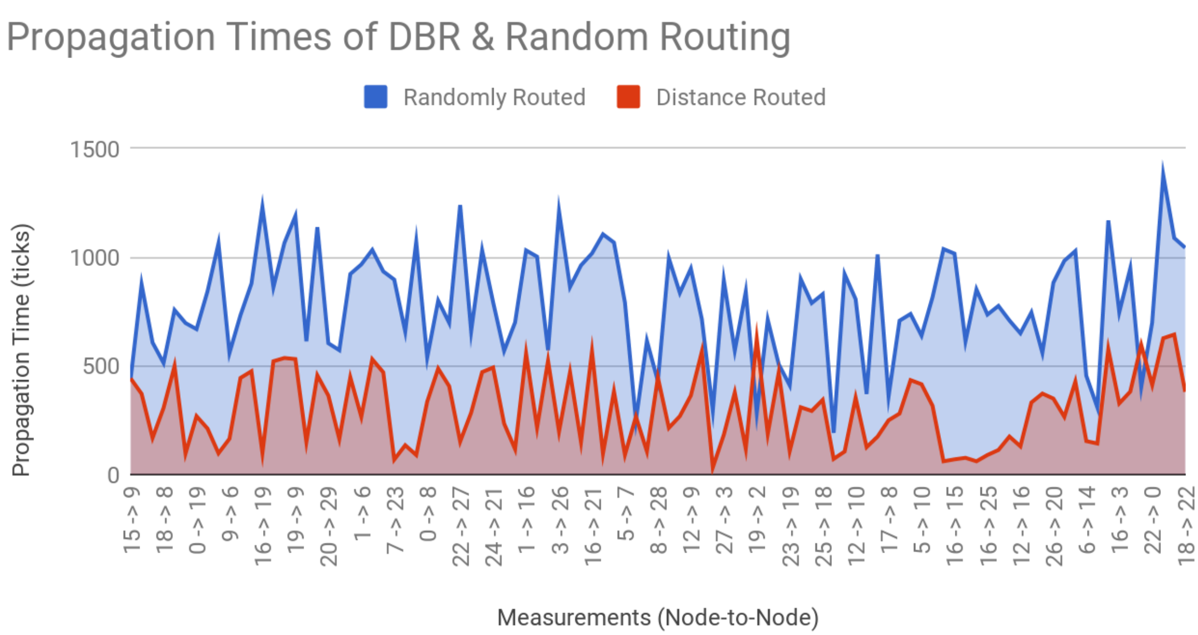

Figure 4

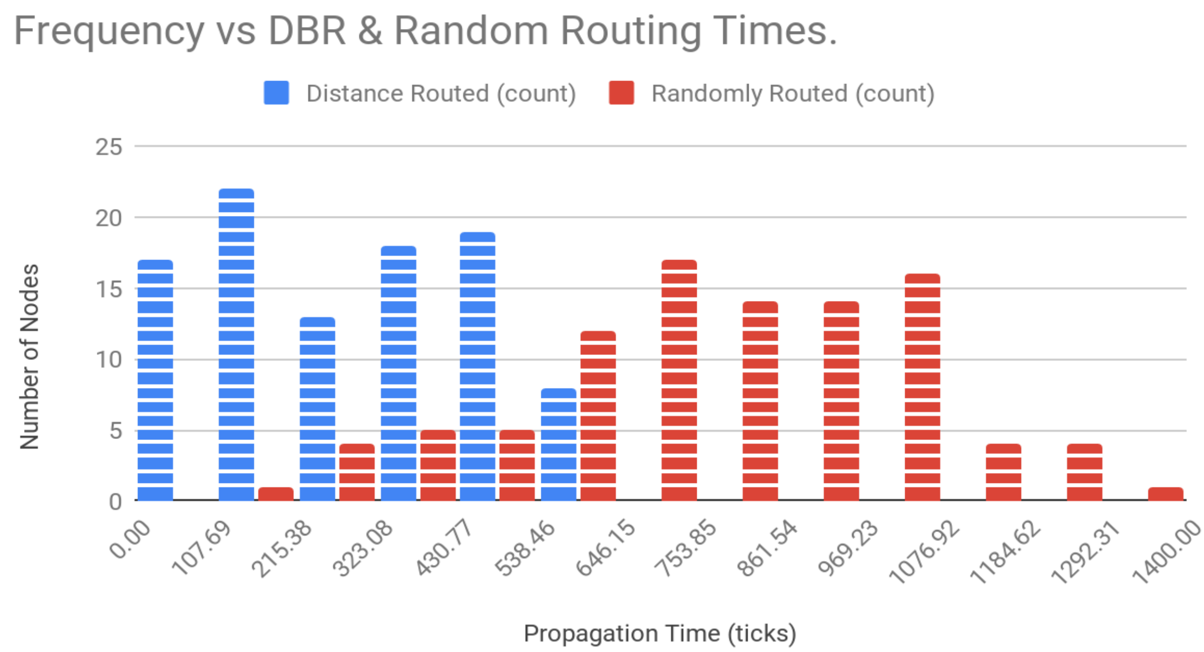

Figure 5

The histogram looks pretty telling by itself, but I ran a statistical analysis on the data anyway to quantify how significant the difference is. The p-value calculated was so low, it rounded down to zero in the calculation program! So, I'm pretty sure I can reject the null hypothesis no matter how low I set the significance threshold (α) and definitively say that DBR is in fact superior to random routing (in simulation at least).

Conclusions & The Future

DBR works well in simulation and shows considerable promise but before it can become a reality there are still a few important steps that need to be taken. First, the system for calculating virtual coordinates needs to be implemented and tested in the model. The simulation as it is now works around that problem. Second, DBR needs to be tested with real computers instead of just in a simulation to see how well it works in the real world. Once those two challenges are overcome, DBR and applications built on top of it have the potential to replace the current structure of the Internet with a more secure and private alternative. As it stands currently, the Internet is plagued by two issues: centralization and lack of privacy. DBR could become a new standard in the growing space of decentralized applications and contribute to the betterment of our Internet.

References

I2P Developers. (2021, February 17). Intro to I2P. I2P. https://geti2p.net/en/about/intro.

MaidSafe. (2018, December 11). Structuring network with XOR. https://medium.com/@maidsafe/structuring-network-with-xor-431e785b5ee7. safepodmtl. (2014).

Onion Routing - Computerphile. YouTube. https://www.youtube.com/watch?v=QRYzre4bf7I.

Srivathsav, R. (2020, June 22). TOR Nodes Explained! Medium. https://medium.com/coinmonks/tor-nodes-explained-580808c29e2d.

1.2. Xor distance and basic routing. https://www.youtube.com/watch?v=w9UObz8o8lY. Computerphile. (2017).

Wickelmaier, F. (2003, May 4). An Introduction to MDS. Retrieved April 19, 2021, from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.495.4629

Appendix

Check out and run the simulation program: https://github.com/libdither/dbr-sim.

Future developments in https://github.com/libdither/dither-sim

Data & Calculations:

https://docs.google.com/spreadsheets/d/15plXbZJtVkt-HolD320xgebIAnleJ2Bnjib9sJE-g5Y/edit#gid=323283991