

# Weekly Update

Ben Weintraub  
Northeastern University

October 30, 2019

## Dataset Generation

This majority of my time these past couple weeks has been spent writing scripts to generate topologies and transaction sets. This involved generating network graphs, sampling transactions from a distribution, calculating link weights for the graph, and converting that data into the file format that GTNA (our simulator) needs for ingestion.

### Generating Graphs

I am using a Python library called Networkx to create a network topology. Networkx is a well-documented and actively developed library for analyzing graphs and network topologies. Networkx can generate various types of connected topologies while accepting some tunable input parameters. For this project, I have written wrapper functions to generate four types of graphs, these are: Watts-Strogatz (small-world), Barabási-Albert (scale-free), Erdős-Rényi (random), and a hybrid graph that I have tuned to be a mixture of the previous three. This hybrid is an attempt to meet the measurements of the Lightning Network from [1] (though it falls a little bit short, further tuning is needed).

The graph I generated has 100000 nodes.

### Generating Transactions

Generating transactions has been the most difficult part of the dataset. The two current models that have been implemented were built up iteratively. Both models rely on sampling from distribution, but they do so in different ways. In both cases the transaction values are sampled from a Pareto distribution with the tuning parameter  $\alpha$  set to 1.16 which creates the 80-20 powerlaw we were going for.

The area where the two models differ is in how they select sources and destinations. In the first model, which we'll call *random*, sources and destinations are selected from a uniform random (without replacement) distribution over the set of all nodes in the graph. This will generate *a* transaction set, but not one that reflects any of the subtleties of how sources and destinations interact. My

first attempt to ameliorate this is by selecting sources and destinations from independent Parato distributions. This captures the idea that some nodes are much more likely to be sources, and other nodes are much more likely to be destinations.

Using the second model, I generated a dataset with 1M transactions.

## Generating Link Credit

To assign credit values to the links in the graph, I used the full-knowledge weight assignment algorithm from the previous write up. This relies on an deterministic routing algorithm, so for this I used Networkx’s built-in shortest path algorithm.

## Results

To use the results, I have created a few functions for converting the Networkx data into GTNA format. For scripting and usability, all the input parameters are read of a configuration file.

The total time it took to generate the dataset with link weights was 24 minutes on our powerful cluster, or 30 minutes on my laptop.

## References

- [1] Elias Rohrer, Julian Malliaris, and Florian Tschorsch. Discharged payment channels: Quantifying the lightning network’s resilience to topology-based attacks. *arXiv preprint arXiv:1904.10253*, 2019.