Maximizing Revenue in Lightning Network

Final Project - Introduction to Cryptocurrencies (67513)

Alon Netser Daniel Afrimi Ariel Elnekave

August 31, 2020

Outline

The Lightning Network

- Suggested as a solution to Bitcoin's long-known scalability issues.
- Move the majority of transactions off-chain, in a trustless fashion.
- Solves the problem of a limited transaction rate.
- Lowers the number of interactions with the blockchain.
- **TODO cool lightning-graph picture**

Incentives in the Lightning Network

- Support transactions between participants without direct channels, using *multi-hop routing*.
- Incentivize participating in other transactions by allowing intermediate nodes to require fees for transferring the money forwards to the next node in the route.
- **TODO Multi-hop transaction image**

Our Goal: Maximize The Profit From The Fees

- Establish channels in strategic positions to make profit.
- The main challenge is deciding which channels to create, and how much money to lock in them.
- Need to be attractive for other parties to route through them.
- We discuss different trade-offs faced by the policies and analyze them using simulations.

The Cost Of Establishing a Channel

- Establishing a channel is a costly procedure.
  It requires:
  - "Locking" some amount of bitcoin (called the *capacity* of the channel).
  - Pay fees to the miners for including the channel's creation transaction in their mined block.
- We treat the locked money as an *investment*.
- The miner's fee is treated as a fee for a mediator to handle our investment.
- **TODO miner's fee image**

Hijacking Routes in Payment Channel Networks

- Tochner et al. examines an attack in which malicious nodes join the network, establishing new channels in strategic locations, maximizing the number of routes that go through.
  This enables a denial-of-service attack.
- **TODO plot from Saar's paper**
- We use similar methods but to another end:
  Maximize the revenue from transaction fees,
  instead of maximizing the amount of routes passing through us.
- Unfortunately, we did not find much research regarding making profit from the fees in the Lightning Network.

Reinforcement Learning

- An agent operating in an environment which is the Lightning Network.
- The agent observes a state s and decides to perform some action a which results in a new state s' and a reward r.
- The agent is not aware of the distribution from which the new state and reward are generated
- The state should include the structure of the graph describing the Lightning Network, which includes the connections between the nodes and the description of each channel.
- **TODO RL image**

Difficulties in Reinforcement Learning

- Unfortunately, getting data for this problem is hard due to privacy reasons.
- In real life, the distribution of the transactions is unknown to the agent.
- This includes which two parties will participate in the next transaction, and how much money will be transferred.
- In order to fully simulate the environment one needs to connect to the Lightning Network with an actual node and monitor its income from the fees.
- We leave it for future work, as this is costly.

Our Setting - A Sub-Graph of The Lightning Network

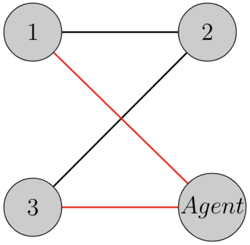
- An optimization problem: Given the Lightning Network's graph (at some time-step) and some fixed distribution over transactions, maximize the reward received from the fees.
- We model the problem as a simulator and an agent communicating between them.
- We took a dump of the Lightning Network from May 2020 and used a sub-graph of the full graph for our experiment.

Baseline - Random Agent

- This one is the simplest algorithm - establish channels with nodes selected uniformly at random.
- Used mainly as a baseline for other more sophisticated ones.

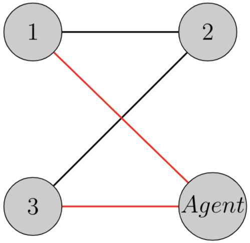
Greedy Agent

- We defined three methods for scoring the nodes, each defines a corresponding greedy algorithm:
  - **Total-Capacity**: Each node's score is its total capacity, i.e. the sum of the capacities in all of the channels it's participating in.
  - **Graph-Degree**: Each node's score is its degree in the multi-graph.
  - **Routeness**: Each node's score is the number of routes it might participate in, when some two nodes in the graph will make a transaction.



Lightning++ Agent

- The motivation for this algorithm is taken from kmeans++ clustering algorithm.
- Add randomness to our agents, so instead of selecting greedily the best node, define a distribution over the nodes where each node probability is according to its score.
- **TODO plot the distribution using visualize lpp script**



Results

- TODO
- TODO
- TODO

Conclusion and Future Work

- TODO
- TODO
- TODO