considerations

June 14, 2024

1 Proposal 1: Shortest Path Fixed Charge NFP - Lightning Network payment graph

This proposal consist into solving a Shortest Path Fixed Charge NFP problem in the context of a state-channel based payment protocol, where payments between partecipants flow based on the shortes and cheapest path between the two nodes. This specific problem is a mixture of the shortest path problem with the classical NFP problem, whereby the cost of taking a specific path is the sum of the fixed fees along the path plus the sum of the variable fee rates (that are proportional to the amount of the payment flowing across the edge).

The dataset to tackle the problem has already been retrieved with some tools of the LND suite. Minor fixes and addons can be executed with the mempool.space APIs, targeting specific nodes and channels.

The network that results from the problem setup is a network where the vertexes are the nodes in the Lightning Network, whereas the edges are the payment channels between the nodes. Each payment channel has a fixed amount of liquidity, a fixed base-fee rate and a variable fee rate. The variable fee rate is based on the amount that flows in the channel. The goal is to find the shortest path between two nodes that minimizes the total fees paid, considering both the fixed and variable fees.

```
"" Nodes in the graph
grep -o 'pub_key' data/network_graph_2024_06_12.json | wc -l
" 12462
```

```
"" Channels in the graph
grep -o 'channel_id' data/network_graph_2024_06_12.json | wc -l
" 34868
```

1.1 Other possible constrains

1.1.1 Liquidity balance

In real world implementations of LN nodes, the pathfinding algorithm used also considers the probability of the payment going through at the first attempt based on the latest tryout attempts that involved that channel. The real LN protocol includes an another complex dynamic constrain that cannot be defined a-priori: the constrain is that liquidity in payment channels can be asymmetrically allocated in each channels such that a channel betweeen peer A and B has 90% of the liquidity in A and 10% in B. This means that at a that specific time, only 10% of the payment channel capacity can flow from B to A and 90% can flow from A to B. This constrain can be manually implemented by allocating randomly the liquidity in the channels considering some previous research carried on by Rene Pickhartt, in which the scientists discovered that the distribution of liquidity in LN payment channels is mostly skewed.

1.1.2 Multi-party Payments

In the context of a state-channel based payment protocol, a payment between A and B can also be executed by breaking up the amount into smaller chuncks and executing multiple smaller payments in a parallel way. This element can enrich the problem formulation.

2 Proposal 2: Knapsack problem - coin selection problem for UTXO based accounting schemes (onchain bitcoin, GNU Taler, cashu)

3 Proposal 3: Knapsack problem - block template creation for mining based blockchain protocols (Bitcoin mining)