ANT ROUTING ALGORITHM FOR THE LIGHTNING NETWORK

CYRIL GRUNSPAN AND RICARDO PÉREZ-MARCO

Abstract. We propose a decentralized routing algorithm that can be implemented in Bitcoin Lightning Network. All nodes in the network contribute equally to path searching. The algorithm is inspired from ant path searching algorithms.

1. Introduction

Bitcoin's network is a decentralized peer-to-peer payment network [6] that allows programmable transactions. Transactions are not instantaneous, and the current protocol does not scale to thousands of transactions per second as it will be necessary to scale to a worldwide payment system. The Lightning Network is a second layer payment network with these properties. Security of the Lightning Network relies on the Bitcoin network. We want to build such an extension preserving decentralization and confidentiality of transactions.

Bitcoin scripting language (and proposed extensions) allows to open one way or bidirectional payment channels ([2], [5], [7], [1]) so that two individuals can perform peer-to-peer almost instantaneous, anonymous and secure off-chain transactions. Only the Initial Commitment Transaction and the final Settlement Transaction need to be registered in Bitcoin's blockchain.

Hashed Timelocked Contracts (HTLC) can be used to compose payment channels, therefore allowing the emergence of a second layer network for instantaneous and more anonymous payments: The Lightning Network.

The Lightning Network, LN from now on, was described in [7]. Its ambition is to scale to an instantaneous decentralized worlwide payment network with minimal fees, and is currently being tested and implemented.

In the original white paper [7] the routing problem is only addressed in section 8.4 where only some tips are given about possible routing algorithms. The belief is that routing tables are necessary for large operators as for current Internet routing algorithms as BGP or Cjdns protocols. A proposal for implementation was presented

Key words and phrases. Bitcoin, blockchain, Lightning network, routing, ant.

in [8]. We refer to this article for historical background on the routing problem and ideas proposed and discussed among developers. In [8] an algorithm called "Flare" is proposed that uses routing tables and beacon nodes that have a richer information about the geometry of the LN network.

The existence of a group of beacon nodes (even when selected randomly and renewed regularly) presents a threat to decentralisation. Indeed, even the global knowledge of the geometry of the network can be a vector of attack.

The main challenge is to implement a resilient, anti-fragile, secure, anonymous, scalable and decentralized routing algorithm.

A necessary condition for perfect decentralisation is to have all nodes performing the same tasks and having access to the same information. Therefore we want to avoid "beacon nodes" as well as any privileged role of a node.

The main inspiration for our algorithm comes from ant behavior. Ant colonies exhibit a superior performance in their "food finding" algorithms. Mathematically speaking, they employ a "balayage technique" of the geometry of their natural habitat in order to collect food. Only the best performers of these algorithms have survived natural selection over millions of years. This is an important reason to consider seriously these algorithms from a mathematical and geometrical point of view. An important body of research has been conducted both by biologists, but also by other scientists and mathematicians. For general background we refer the reader to [3][1]. One of the distinctive idea of ant algorithms is the use of pheromones in order to leave a "trace" of their passage and marking the food collecting routes. A natural routing algorithm is at work. The idea of "leaving a trace" appears as well in greek mythology in the myth of Ariadne's thread in order to solve in a practical efficient way the "path exiting problem" in a labyrinth.

We present in this article a routing algorithm where we implement similar ideas in order to achieve maximal decentralization. We present here only a first draft of the algorithm. Further numerical simulations are necessary to evaluate the performance and scalability of the algorithm.

An interesting feature of the algorithm is that it has potential learning capabilities: It can adjust to the dynamic geometry of the network and can improve its performance over time (see 8).

[1] Even in section 6 we can find a discussion on "ant routing algorithms" that, although more complicate, have the same flavor and share some ideas with the algorithm presented in this article.

2. Preliminaries on the Lightning Network.

As described in the white paper [7] the Ligthning Network (LN) is an aggregation of payment channels that can be composed. For the purpose of the present article we don't care about the specifics of the payment channels (if they are bidirectional or not only one-way channels, etc). The payment channel network is assumed to be richly and randomly connected, as for the Bitcoin network.

We assume that on top of the LN there is a separate, but richer, fast communication network. This means that all individuals (nodes) with an open payment channel between them do have a communication channel, but individuals without a direct payment channel can have a communication channel. This will be the case between Alice and Bob when both are part of the LN and want to make a transaction from Alice to Bob. The route finding algorithm operates at the level of the communication network. The speed of communications is one of the main bottlenecks for the speed of the LN.

What we need to know from the LN is essentially that payment channels can be composed, once we know a payment path. Also payment channels support a maximal volume per transaction. This is a particular feature of the LN and it is discussed in Section 5. We postpone the analysis of possible misbehaviors and attacks on the LN, that deserves by itself a lengthly and careful discussion.

3. Simple ant routing algorithm.

Alice wants to pay Bob a certain amount of Bitcoins through the LN network by composing already existent payment channels. We can assume that the network is composed of bidirectional channels (although this is not necessary as discussed in Section 7) and we can even have a mixture of unidirectional or bidirectional channels). We assume that a communication channels are open between neighboring nodes. In this section we describe how the network finds a path between Alice and Bob. This is only, for now, the simplest geometric problem. In this section we don't address the limitation by the volume of the channels (you can assume that all channel are of a volume larger than the payment), nor we address the fee question (assume free transmission). Later on we consider channel payment volume limitations and fees (see Sections 6 and 5).

(1) Alice and Bob agree on a large random number. For example, Alice and Bob choose a random 128 bit numbers, $R(A)$ and $R(B)$ and exchange them in a secured way.

(2) Alice concatenates the bit 0, and the hash[2] $R = h(R(A)^\frown R(B))$ to get a pheromone seed $S(A) = 0^\frown R$ and communicates $S(A)$ to its immediate neighbors in the LN with whom she has an open payment channel.
(3) Bob concatenates the bit 1, $R(A)$ and $R(B)$ to get a pheromone seed $S(B) = 1^\frown R$ and communicates it to its neighbors in the LN with whom he has an open payment channel.
(4) Alice waits from an answer from its neighbors indicating her that a path has been found by the network.
(5) Bob waits to have news from Alice that a path has been found.

If S is a pheromone seed, we denote S' the "derived seed" without the appended first bit, that is, the hash R. (thus $S = 0^\frown S'$ or $S = 1^\frown S'$). If $S = 0^\frown S'$ (resp. $S = 1^\frown S'$) we denote by $\bar{S}$ the "conjugate" seed $\bar{S} = 1^\frown S'$ (resp. $\bar{S} = 0^\frown S'$).

The nodes perform the following tasks (on top of a possible payment task if they are Alice or Bob).

(1) Each nodes reserves a fast access memory space for the routing tasks. We refer to this as the "mempool" of the node.
(2) Each node keeps in memory a numbered list of neighbors in the LN together with the relevant information about its payment channel(s) opened with them. Also about historical performance of payments through these neighbors.
(3) When a node receives a pheromone seed S, it checks if S' is not a derived seed of a seed already stored in the mempool.
(4) If S' is not found, then it stores S in the mempool together with the information about the neighbor that has communicated S (the "transmitter neighbor"). Then it broadcasts S to the other neighbors.
(5) If S' is found, then it checks if S is stored.
    (a) If S is stored it adds the information about the new transmitter neighbor.
    (b) If S is not stored it means that $\bar{S}$ is stored, so a matching occurs.
(6) When a matching occurs, the node concatenates the bit 0 to S (resp. $\bar{S}$) and constructs a "matched seed" $S_m = 0^\frown S$ (resp. $\bar{S}_m = 0^\frown \bar{S}$ and sends it to the neighbors from which it received S (resp. $\bar{S}$). Note that "matched seeds" are one bit longer than pheromone seed. The node keeps track of the neighbors having transmitted the unmatched seed.
(7) When a node receives a matched seed S_m it broadcasts it back to the neighbors that send to him the unmatched seeds and keeps track of them.

[2] For example h can be the commonly used SHA256 in Bitcoin protocol. We take a hash in order to preserve a shared secret by Alice and Bob that they can used to prove that they are the originators of the transaction and can serve extensions of the algorithm.

Following this procedure, Alice will receive back several matched seeds that will correspond to different possible payment paths. She chooses one, say S_m, and concatenates a further 0 bit creating a confirmed seed $S_c = 0 ^\frown S_m$ that she sends back to the neighbor that send her the matched seed. Confirmed seeds are one bit longer than matched seeds, and two bits longer than pheromone seeds. The nodes broadcast back the confirmed seed to the corresponding neighbor from which he received the matched seed until they reach the node that did the match. This node continues the broadcasting back of the confirmed seed until it reaches Bob. Once Alice receives from Bob the confirmation of the payment path, she starts the conditional payment chain as described in [7] and the transaction will be completed.

Once the payment have been done, the nodes erase the data corresponding to the confirmed seed. Also, after some threshold time τ, the nodes drop all data (matched and unmatched) older than τ. If no path is found after this thereshold time, the "path finding" request of Alice and Bob is erased from the network. In that way the mempool keeps a controlled size. Each node can decide its own threshold time.

4. Properties of the algorithm.

Some of the properties of the algorithm are listed below:

(1) Anonimity: Intermediary nodes have no information about Alice and Bob. They only have information about the two neighbors in the payment path.
(2) Anonimity: No records of payments are kept.
(3) Anonimity: No node that is not on the payment path knows about the payment.
(4) Anonimity: No global information about the geometry of the network is necessary. No routing tables are needed. We avoid all the computation load of updating routing tables. Nobody needs to share information about their neighbors.
(5) Decentralization: All nodes perform the same function and follow the same rules. In particular, no beacon nodes, and again no routing tables.
(6) Scalability: Pending of numerical simulations, the routing algorithm should be able to handle thousands of transactions per second.
(7) Instantaneous: Depends on the speed of communication that appears to be the first bottleneck.

As drawback we can cite the intensive computational communication and processing which is typical from a totally decentralized protocol. As described in Section 8, nodes can improve the efficienty and limit the workload by algorithmically selecting preferred neighbors from historical data that they gather.

5. Channel volume compatibility.

The amount of the transaction should be compatible with the maximal volume of each payment channel (in the intended payment direction). Thus, some of the paths found could be incompatible with the payment amount.

In order to ensure that the path found is volume compatible, Alice and Bob add an "amount field" to their pheromone seed. Then they broadcast it as before. Nodes in the network only broadcast the pheromone seed to neighbors which whom they have a volume compatible open payment channel. The rest of the procedure is the same and we get the subset of paths that are "volume compatible" with the transaction.

Disclosure of the amounts is not suitable for the sake of preservation anonymity of the transaction. One may think to obfuscation implementations (as for the mimblewimble protocol), but for the LN this is not as important as for standard bitcoin transactions since Alice and Bob may issue multiple transactions dividing the total amount into micropayments.

6. Fee considerations.

The incentive for nodes to participate in payment paths in the LN are fees for transaction relay. Each node freely determines the fee it will take to participate in a payment path. It will deduct the fee amount from the transacted amount. There are different approaches for the fee treatment. We describe one that is simpler and straightforward, but leaks some information about the length of the payment path (since it is correlated to the fee amount). Other approaches obfuscate the fee amount to the intermediary nodes, but involve a longer procedure and will be presented in future versions.

Fee algorithm. In Alice and Bob's pheromone seed a "maximum fee field" and a "current fee field" are added. Alice and Bob initialize the "current fee field" by 0 and the "maximum fee field" by the maximum amount that Alice is disposed to pay (that she communicates to Bob). Each time a pheromone seed is broadcast by a network node, it checks that the current amount and its fee is smaller than the maximal fee amount, it adds its fee to the amount in the current fee field. It keeps track in the mempool of the neighbor nodes it has communicated the pheromone seed as well as the corresponding fee amounts.

When the matching node receives Alice and Bob pheromone seeds, it checks that the sum of the amounts in both "current fee fields" is smaller than the maximum amount fee deducted from its own fee. Only under that provision the matching occurs. The total fee becomes then the sum of the current fee amounts of both pheromone seeds plus the matching node fee. Then in the matched seed this total fee amount is included in the “current fee field”. When nodes broadcast the matched seed they do not change the “current fee field“. The subsequent nodes that relay the matched seed relay respecting the fee they did indicate prior. If some node intents to increase the fee, it will be noticed by other nodes, an anomaly will be detected, and the matched seed will be not be further relayed.

Alice will then received a list of proposed matched seeds with associated fee amounts. She can then select the lowest fee path (if she wishes).

Obviously, in this setup the matching node can take advantage of his position and set a maximal fee in order to match the maximal fee. This could be a reward to be a matching node, but it is not clear that it has any advantage in trying to maximize its profit since after all there is a competition between the paths discovered and Alice will probably select the paths with lowest fee, thus maximizing the fee will result in lower probability for being selected.

7. Hybrid channels.

So far, the only property we have used of payment channels is its transitivity, i.e. that they can be composed. The algorithm described works as well for unidirectional channels, and one may even imagine an hybrid LN with swaps between different blockchains, as long as they can be composed. The algorithm is independent of the nature of the payment channels as long as they have the usual properties for composition of payment channels.

8. Self-improvement of the algorithm.

One typical feature of ant path finding algorithms is the self-reinforcement of paths by the intensity of the pheromone trace. This intensity increases with the number of ants taking the path.

We can propose a similar reinforcement mechanism in order to boost the performance of the network. This can be done with each node assigning a performance benchmark to each neighbor. In the neighbor tables with information about its payment channels, it can store historical information, as for example how many payments have been completed, or what is the total historical volume having circulated through that channel, etc Also short term and long term ”neighbor performance“ is an important data.

Then each node can selectively broadcast the pheromone seeds according to the numerical criteria he wishes to implement taking into account its neighbor’s numerical data. In particular, the comparison between short term and long term data reflects the dynamical changes of the network and allow the nodes to adjust to the new geometry.

The nodes may also use random algorithms to select their preferred neighbors for broadcasting pheromone seeds. A Pareto type distribution in terms of an historical activity index will emulate closely the ant algorithms (see [4] for example).

References

[1] C. Decker, R. Russell, O. Osuntokun. *eltoo:A simple layer2 protocol for Bitcoin*, 2018.

[2] C. Decker, R. Wattenhofer. *A fast and scalable payment network with bitcoin duplex micropayment channels*, Symposium on Self-Stabilizing Systems, 2015.

[3] M. Dorigo, T. Stützle. *Ant colony optimization*, MIT Prees, 2004.

[4] M. Vela-Préz, M.A. Fontelos,J.J.L. Velzquez. *Ant foraging and geodesic paths in labyrinths: Analytical and computational results*, Journal of Theoretical Biology 320, p.100112, 2013

[5] M. Hearn, J. Spilman. *Bitcoin contracts*, `en. bitcoin.it/wiki/Contracts`, 2015.

[6] S. Nakamoto. *Bitcoin: a peer-to-peer electronic cash system*, `Bitcoin.org`, 2008.

[7] J. Poon, T. Dryja. *Lightning network*, 2015.

[8] P. Prihodko, A. Ostrovskiy, O. Osuntokun, M. Sahno, S. Zhigulin. *Flare: An Approach to Routing in Lightning Network*, version 1.0, 2016.

Cyril Grunspan
Léonard de Vinci Pôle Univ, Research Center, Labex Réfi
Paris, France,

E-mail address: `cyril.grunspan@devinci.fr`

Ricardo Pérez-Marco
CNRS, IMJ-PRG, Univ. Paris 7, Labex Réfi
Paris, France

E-mail address: `ricardo.perez.marco@gmail.com`

Author's Bitcoin Beer Address (ABBA)[3]:
1KrqVxqQFyUY9WuWcR5EHGVvhCS841LPLn

[3] Send some bitcoins to support our research at the pub.