A Balanced Routing Algorithm for Blockchain Offline Channels Using Flocking

Subhasis Thakur(✉) and John G. Breslin

National University of Ireland, Galway, Ireland
{subhasis.thakur,john.breslin}@nuigalway.ie

Abstract. Offline channels have the potential to mitigate the scalability problem of blockchains. A Path-Based fund Transfer (PBT) uses a path in the channel network. PBTs can make the channel network imbalanced, i.e., funds in a few channels become very low and funds in other channels become very high. Imbalanced channel network may make PBT infeasible. Hence we need a routing algorithm for PBT that keeps the channel network balanced. In existing solutions for this problem have privacy problem as the channels have to reveal their balances in order to find suitable routes for PBTs. In this paper, we mitigate this problem as we propose a flocking based algorithm for PBTs that keeps the channel network balanced.

Keywords: Offline channels · Bitcoin lightning network · Flocking

1 Introduction

Public and proof of work based blockchains have a scalability problem. For example, the Bitcoin network can process 7 transactions per second, Ethereum network can process 15 transactions per second while Mastercard can process 50000 transactions per section. Offline channels can improve the scalability of blockchains. Offline channels such as Bitcoin lightning network aims to reduce the number of transaction to be recorded in the blockchain by allowing offline transactions among pairwise parties. An offline channel among two parties aims to create only two transactions in the blockchain. One transaction is created during opening the channel and another transaction is created to close the channel. The blockchain does not require to remain informed about transactions between these two events. A Path-Based fund Transfer (PBT) in channel network uses a path in the channel network to transfer fund between two peers who do not have a mutual channel.

There are few problems with the blockchain offline channels that prevent its practical deployment for micropayments. For greater longevity of the channel network (maximum usage of channel network without recording transactions into the blockchain), PBTs must be coordinated to maintain the balance of a channel

network. Balancing the channel network means maintaining uniform usage of channels. For example: Let the pair of peers A and B (and the pair of peers B and C) have bidirectional channels between them. If the path $A \rightarrow B \rightarrow C$ is used for fund transfer between A and C then, channels $A \rightarrow B$ and $B \rightarrow C$ will lose fund and mirrored channels ($B \rightarrow A$ and $C \rightarrow B$) will gain funds. The repeated usage of the paths $A \rightarrow B$ and $B \rightarrow C$ will drastically reduce their value and eventually they will no longer support PBT. The longevity of a channel network can be improved by coordinating PBTs. The current methods for such coordination collect PBT information to predict the best paths for future PBTs that maintain a balanced channel network. The problem with such an approach is the privacy of the peers. Peers must reveal their fund transfer information to maintain the balance of the channel network. In this paper, we propose a routing algorithm that preserves the privacy of peers as they do not need to reveal PBT information to maintain balanced channel network. The paper is organized as follows: In Sect. 2 we mention related literature and in Sect. 3 we describe the basics of offline channels. In Sect. 4 we present our routing algorithm. We evaluate our solution in Sect. 5 and we conclude the paper in Sect. 6.

2 Related Literature

Bitcoin lightning network was proposed in [9] which allows peers to create and transfer funds among themselves without frequently updating the blockchain. Similar networks are proposed for Ethereum [1] and credit networks [7]. A routing algorithm for Bitcoin lightning network was proposed in [10]. A fast routing protocol was proposed in [2]. A method for anonymous payment to improve privacy in PBT was developed in [4]. [5] proposed a decentralised routing algorithm for the channel network. [8] proposed privacy preserving routing protocol for PBT that can handle concurrency. [6] proposed a balanced routing algorithm for the channel network. [4] proposed privacy preserving schemes for fund transfer in the offline channel network. [3] proposed a privacy-preserving routing method for fund transfer in the channel network. The contributions in this paper advance the state of the art as follows: (a) Majority of routing algorithms for PBT in channels do not consider the problem of balancing the channel network. (b) Privacy-preserving solutions in these routing algorithms are not designed for maintaining privacy for balancing the channel network, rather these algorithms ensure the privacy of the sender and the receiver of a PBT. (c) [6] proposed a routing protocol that balances the network. It requires finding cycles in the channel network in order to make fund transfers to keep the channels balanced. Our proposed method does not need any fund transfer among the channels to keep it balanced. Also, this routing algorithm ignores the privacy of peers as they have to reveal the PBT information.

3 Offline Channels

In this paper, we use the offline channel network construction for Bitcoin lightning network [9]. The basic protocol for using an offline channel is as follows:

(1) Say Alice and Bob want to create a channel between them with balances 10 tokens (each contributes 5 tokens). (2) Alice and Bob create two pairs of lock (hash) and key (random string). They exchange the locks. (3) Bob creates a 'confirmation transaction' as follows: (3.a) There is a multi-signature address between them which requires a signature from both to transfer fund from it. We will call this address M_1. (3.b) Bob creates transactions from M_1 which states that Bob will get 5 tokens and remaining 5 tokens will go to another multisignature address between them. We will call this address M_2. (3.c) The 5 tokens in M_2 will be given to Alice after 10 days or Bob can claim it if it can produce the key to the lock of Alice. (4) Bob signs this transaction and sends it to Alice who can use it to get tokens from the channel by signing it and publishing it to the blockchain network. (5) Alice produces mirrored confirmation transaction and sends it to Bob. The confirmation transactions ensure that both parties can recover from if they fund the channel between them. (6) Now Alice and Bob transfer fund in the multi-signature address by creating transactions in the blockchain and hence the channel becomes operational. (7) Both parties should exchange keys and create new confirmation transaction to update the channel. (8) If any party announces a confirmation transaction then the channel closes.

The protocol for path based transfer in channel network is as follows: Say Alice wants to send fund to Carol via Bob. (1) Carol will create a lock and a key. (2) In the multi-signature address between Carol and Bob, a contract will be created as follows: (2.a) Bob will send 5 tokens to this address. (2.b) Bob will get these tokens back after 9 days if Carol does not claim it. (2.c) Carol can claim it anytime if it can produce the key to the lock. (3) Similarly, another contract will be created between Alice and Bob as follows: (3.a) Alice will send 5 tokens to this address. (3.b) Alice will get these tokens back after 10 days if Bob does not claim it. (3.c) Bob can claim it anytime if it can produce a key to the lock. (4) Thus Carol reveals the key to Bob as it collects the fund, which Bob uses to get refunded from Alice.

4 Flocking Based Routing Protocol

In this routing protocol, we preserve the privacy of channels by not gathering information on exact channel balances. Each channel is assigned a coordinate in two-dimensional space. In such space, channel distance between two peers is measured as the Euclidean length of the edge between them in the X-Y plane if they have a channel. We use the length of a channel to 'replace' value of a channel in order to preserve the privacy of the peers. A solution to keep the channels balanced is: increase the usage of channels with high value and decrease the usage of channels with low funds. In a similar approach, our routing algorithm prefers shortest paths (measured as the total Euclidean distance of a path) for PBTs. Our main innovation is how to keep updating distances of channels in order to reflect their usage. We solve this by using the flocking algorithm. Flocking algorithm mimicries how flocks of animals perform coordinated group traversal without crashing with each other only using local information about the position

and velocity of animals in close proximity. We interpret flocking behavior to keep the channels balanced as follows:

(1) If a channel is used in a PBT then it moves in a specific direction according to the details of the PBT. The channels move in such a direction that mutual distances among the channels used in a PBT increase and the same for the complementary path decrease.
(2) Triggered by the movement of channels in a PBT, surrounding channels adjust their positions according to rules of flocking. Flocking behavior ensures equilibrium among channel distances. For example, as shown in Fig. 1 (right), a PBT uses path $V1$ to $V2$. Hence the distance between $V1$ and $V2$ is increased to make it less likely that it will be used again (less likely to be in any shortest path). But as $V1$ is moved to a new coordinate, the distance between $V1$ and $V3$ also increased. In order to compensate such increase in distance $V3$ will follow the flocking procedure and find a new coordinate to reduce their distance .

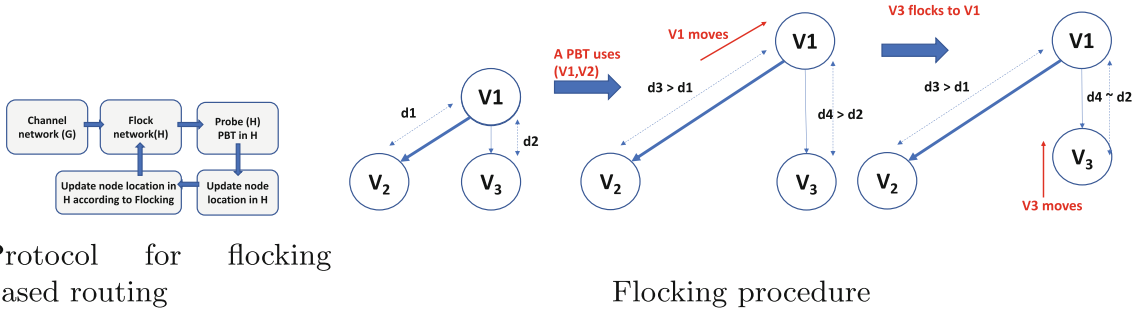


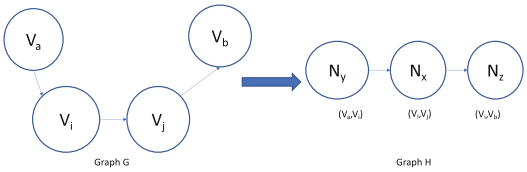
Fig. 1. Flocking based routing

The overview of the flocking based routing is shown in Fig. 1 (left). First, we generate the Flock graph from the channel network by creating a node every edge (a channel) in the channel graph. In the next section, we will assign coordinates to each node in the flock graph and describe the probing and PBT method for the Flock graph. After the execution of a set of PBTs, the flock graph is updated as the position of nodes are changed due to the execution of a PBT and due to flocking behavior. After updating the channel network with new coordinates for nodes in the Flock graph we execute the next set of PBTs.

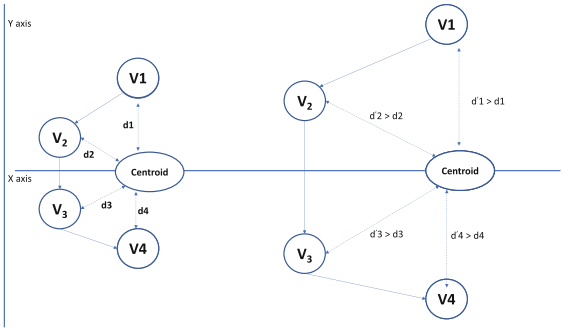
4.1 Creation of Flock Graph H

Let $G = (V, E)$ be a directed graph representing the channel network with n peers (V) and m channels (E). $W(E) \mapsto \mathbb{R}^+$ be the value of the channels. We assume that the channels are bi-directional. Hence if $(V_i, V_j) \in E$ then there exists a complementary channel $(V_j, V_i) \in E$. We create a directed graph $H = (N, L)$ with nodes N and edges L from G as follows:

(1) For each edge $(V_i, V_j) \in E$ we create a node N_x in H. (2) Let $F \subset E$ be the set of edges incident on V_i. For each edge $(V_a, V_i) \in F$, we create an edge (N_y, N_x) in H where N_y is the vertex corresponding to the edge (V_a, V_i). (3) Let $F' \subset E$ be the set of outgoing edges from V_j. For each edge $(V_j, V_b) \in F$, we create an edge (N_z, N_x) in H where N_z is the vertex corresponding to the edge (V_j, V_b). (4) Vertex N_x's weight will be the same as the weight of the edge (V_i, V_j). (5) Each node N_x will be assigned a coordinate in the X-Y plane. Length of an edge in H will be determined by the Euclidean distance. $D(L_x)$ will denote the length of edge $L_x \in L$. (6) A path in H will denote the set of channels used in a PBT. The complementary path is the path in H corresponding to opposite channels in G.



Creation of Flock graph H.



Nodes v_1, v_2, v_3, v_4 move away from the centroid after completion of the PBT $v1 \rightarrow v2 \rightarrow v3 \rightarrow v4$.

Fig. 2. Flocking procedure

4.2 Probing and PBT

Let V_x wants to transfer x tokens to V_y using a PBT. V_x uses the following steps:

1. Let $F \subset E$ be all outgoing edges from V_x and $F' \subset E$ be all incoming edges to V_y in the graph G.
2. Let $N^F \subset N$ be all nodes in the graph H corresponding to channels F and $N^{F'} \subset N$ be all nodes in the graph H corresponding to channels F'.
3. V_x can start the transfer from any node in the set N^F and end the transfer in any node in the set $N^{F'}$.
4. Thus V_x has to probe paths from the set N^F to the set $N^{F'}$.

Probing these paths are executed as follows:

1. Let $Cost$ be a $A \times B$ matrix where $|N^F| = A$ and $|N^{F'}| = B$. $Cost[i, j]$ will denote the distance starting from $N^F[i]$ to $N^{F'}[j]$.
2. $Cost[i, j]$ is calculated as the length of the shortest path in the graph H. Note that distances between the nodes in H are changing every time a new transaction is executed.
3. After computing $Cost$ matrix, V_x probs the paths starting from the lowest $Cost$ if it has enough balance to support its PBT.

4.3 Node Location Update in H

Coordinates of nodes in H are updated for every PBT. For each PBT $v1 \rightarrow v2 \rightarrow v3 \rightarrow v4$, we find the centroid of the v_1, v_2, v_3, and v_4. All nodes move further away from the centroid as shown in Fig. 2 (right). Similarly, nodes in the complementary path of the path $v1 \rightarrow v2 \rightarrow v3 \rightarrow v4$, move towards their centroid. In a general PBT algorithm, the sender of a PBT can inform all peers in the PBT to change their locations and in a multi-hop PBT, the sender can send such location change information to all peers by introducing small noise in the new location information.

4.4 Node Location Update in H Using Flocking

There are two steps in flocking [11] based node location change. Each node categories its surrounding nodes into two groups (1) Repulsion zone and (2) Align zone. Nodes in the repulsion zone are too close to a node and it wants to move away from them and nodes in align zone are neither too far from it nor too close to be in repulsion zone. Each node finds the mean angle of all nodes in its repulsion zone. The angle of a node is its heading calculated from the point $(0, 0)$ in the given X-Y plane. The node finds the opposite of this mean angle by adding 180°. Let's call this angle $Angle_1$. Similarly, the node finds the mean angle for all nodes in its align zone. Let's call this angle $Angle_2$. The position of the node (x_i, y_i) is changed as follows: $x_i = x_i + .2(Cos(Angle_1) + Cos(Angle_2)), y_i = y_i + .2(Sin(Angle_1) + Sin(Angle_2))$.

5 Evaluation

We use Bitcoin lightning network data. There are 2800 nodes and approximately 22000 edges. We create three subgraphs from this dataset excluding channels with a very high degree. We extract three subgraphs from the lightning network data with the number of nodes 400, 600 and 800 respectively. The average degree of nodes in these graphs is 11. The average channel balance is 7 and the average value of PBTs is .5. We simulate the PMTs and balanced routing algorithm in R. In each experiment, we execute 1000 PBTs. We compare the performance of our routing algorithm with a routing procedure that always uses the shortest path from the sender to the receiver. Majority of routing algorithms for PBT aims to develop a fast routing algorithm. Hence we use the shortest paths between the parties as to the outcome of such routing algorithms. We use three parameters to evaluate the performance of our routing algorithm as (a) standard deviation of channel values (high value of the standard deviation indicates that channels are highly imbalanced) (b) number of successful PBTs and (c) path length of PBTs (it indicates the cost transfer). The outcome of these experiments is shown in Fig. 3. First three plots show the standard deviation of the channels. It clearly shows that standard deviation of channels is increasing over the time for shortest path based routing and the same remains almost constant for our balanced routing algorithm. The last figure shows (a) number of completed PBTs for balanced

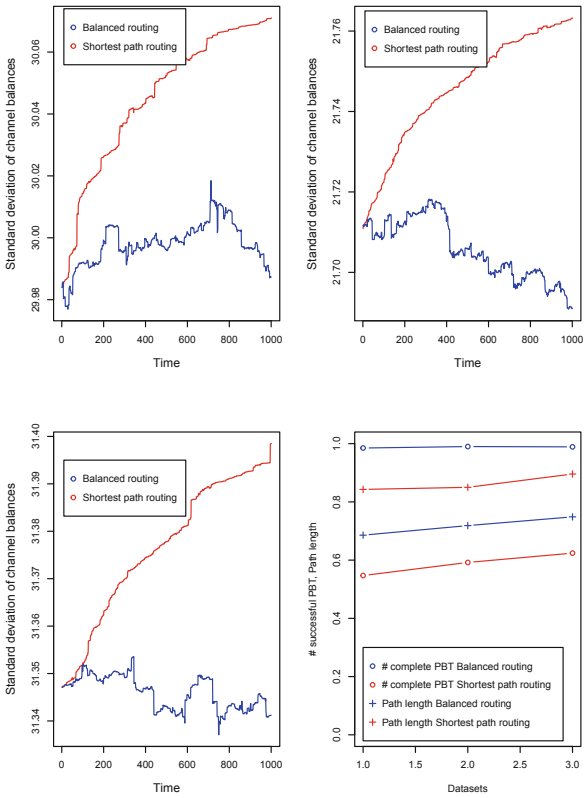


Fig. 3. Results: First three plots show the standard deviation of channels remains low for balanced routing. The last figure shows (a) number of completed PBTs for balanced routing is higher than the same for shortest path routing and (b) Average path length for PBTs is higher for balanced routing.

routing is higher than the same for shortest path routing and (b) Average path length for PBTs is higher for balanced routing. In the proposed routing protocol it is possible that a peer broadcasts wrong coordinates intentionally. But it is difficult to find appropriate coordinate to mislead the peers as by moving away from a set of peers, a peer may move closer to another set of peers. Also, we may use the reputation of peers to deter such behavior.

6 Conclusion

In this paper, we proposed a routing algorithm that can maintain a balanced channel network. Our routing algorithm preserves the privacy of the peers as they do not need to reveal exact channel balances. Using experimental evaluation we have shown that our algorithm maintains a balanced channel network and outperforms other routing algorithms which use the shortest path as the route for PBT.

Acknowledgement. This publication has emanated from research supported in part by a research grant from Science Foundation Ireland (SFI) and the Department of Agriculture, Food and the Marine on behalf of the Government of Ireland under Grant

Number SFI 16/RC/3835 (VistaMilk), co-funded by the European Regional Development Fund and the financial support of Science Foundation Ireland (SFI) under Grant Number SFI/12/RC/2289.

References

1. Raiden network. http://raiden.network/
2. Decker, C., Wattenhofer, R.: A fast and scalable payment network with bitcoin duplex micropayment channels. In: Pelc, A., Schwarzmann, A.A. (eds.) Stabilization, Safety, and Security of Distributed Systems, pp. 3–18. Springer, Cham (2015)
3. Malavolta, G., Moreno-Sanchez, P., Schneidewind, C., Kate, A., Maffei, M.: Anonymous multi-hop locks for blockchain scalability and interoperability. In: NDSS (2019)
4. Green, M., Miers, I.: Bolt: anonymous payment channels for decentralized currencies. In: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. CCS 2017, pp. 473–489. ACM, New York (2017). https://doi.org/10.1145/3133956.3134093
5. Grunspan, C., Pérez-Marco, R.: Ant routing algorithm for the lightning network CoRR abs/1807.00151 (2018). arXiv:abs/1807.00151
6. Khalil, R., Gervais, A.: Revive: rebalancing off-blockchain payment networks. In: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. CCS 2017, pp. 439–453. ACM, New York (2017). https://doi.org/10.1145/3133956.3134033
7. Malavolta, G., Moreno-Sanchez, P., Kate, A., Maffei, M.: Silentwhispers: enforcing security and privacy in decentralized credit networks. IACR Cryptology ePrint Archive 2016/1054 (2016)
8. Malavolta, G., Moreno-Sanchez, P., Kate, A., Maffei, M., Ravi, S.: Concurrency and privacy with payment-channel networks. In: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. CCS 2017, pp. 455–471. ACM, New York (2017). https://doi.org/10.1145/3133956.3134096
9. Poon, J., Dryja, T.: The Bitcoin Lightning Network:Scalable Off-Chain Instant Payments. https://lightning.network/lightning-network-paper.pdf
10. Prihodko, P., Zhigulin, S., Sahno, M., Ostrovskiy, A., Osuntokun, O.: Flare : an approach to routing in lightning network white paper (2016)
11. Reynolds, C.W.: Flocks, herds and schools: a distributed behavioral model. In: Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques. SIGGRAPH 1987, pp. 25–34. ACM, New York (1987). https://doi.org/10.1145/37401.37406