



# Improving Success Rate of Transactions in Cryptocurrency Payment Channel Networks

RituSindhu, KartikTomar

**Abstract:** *In the recent times, the interest in Blockchain-based crypto currencies has increased rapidly. But the slow time of confirmation of transaction and the high charges per transaction prove to be a problem in the adoption of this technology for micro-payments. To solve this scalability problem, the idea of establishing payment channel networks where nodes establish payment channels between each other to perform off-chain transactions. However, this method results in a certain one-way capacity for both nodes for making transactions which means that if a node exceeds its capacity, the channel becomes useless in that direction and hence the payment fails which creates an imbalance in the network. In this paper we aim to increase the rate of success of payments by using the fact that each node is connected to multiple points in a network which can act as gateways and we can use any of these gateways to complete a transaction. By using the inbound and outbound payment traffic ratio, we also define a method for the selection of a particular gateway to carry out the transaction. We use split payment capability with this method to increase the rate of success furthermore, especially for larger transactions.*

**Keywords-** Bitcoin, Blockchain, Payment channel network, Routing, Lightning network

## I. INTRODUCTION

Crypto currency is one the most influential invention of 2010s which is used for storage and transfer of assets. The idea of cryptocurrency is storing the ownership details of different assets in a highly secure database called the Blockchain. Blockchain is also used in various other applications but Bitcoin is the first use case which is widely accepted. Any central authority to authorise and keep the record of transactions is not needed due the mechanism used in Blockchains which makes it useful for transaction in untrustworthy environment. Blockchain is available publically and is also irreversible, which makes the system secure against node failures and malicious activities. The use Blockchain removes the dependency on third-parties for transactions and solves the single point of failure problem.

The concept of virtual currency still faces two big problems: 1) high conformation time of transaction; 2) high transaction fees. It takes 10 minutes to approve a block of transaction. Also, size of a block is 4MB which limits the number of transactions in a unit time. A maximum of 7 transactions per second can be done using Bitcoin which is

very less than that of Visa or MasterCard. Some attempts have been made recently to address these issues. One of the proposed solutions is the Payment channel network concept which is also known as off-chain transaction networks. In this, we try to utilise smart contract and avoid writing every transaction on the Blockchain by recording the transactions off-chain until the accounts are reconciled. This allows us to perform a number of transactions in both the directions once a channel is created between two nodes. If many nodes come together, this off-chain channel turns into a network of payment channels. Then instead of opening a new payment channel every time, a node utilises an already existing channel to make the transaction by paying a fees for as long as the path exists between the two concerned nodes. The utilisation of existing channels reduces the cost of channel creation. The perfect example of this concept is Lightning Network. This payment network has specific node which charge a fees to users to send data over these nodes. Lightning network provides almost real-time transactions with a negligible fee. Payment channel networks like Lightning network face their own challenges because of the way the capacity of their channels is consumed. If a channel's capacity is full, a transaction cannot be completed. The channels are bi-directions, but each peer has its own one-way capacity which can be completely different from the capacity of other peers. This means, a channel's capacity in one direction can be totally consumed while the channel in the opposite direction has all the funds. Due to this, insufficient funds in one-way channels drop the chance of successful transactions drastically in the network. Recent studies done on Lightning network have shown that the chance of success of a transaction from one node to another is just 50% which is very low making it useless for end-user implementation. This issue must be resolved to increase the success rate of transaction. We propose the modification of the route selection in the payment network. The current routing method used in Lightning network enables the node operators to set the fee, select the connections and the path to complete a payment. The end-users always choose the cheapest path for the transactions. This creates an imbalance in the network. To solve this, the priority for path selection must be flow maximisation and considering the whole network for it to maintain a balance in the network. The routing method we are proposing in this paper is called inbound/outbound ratio. In this we calculate the total inbound capacity to outbound capacity of every connected gateway and choose the one with the minimum of these. This method is designed on the idea that routing nodes are balanced automatically as the transmit data in both the directions but the gateway nodes may be overload in a particular direction or may not be used at all.

Revised Manuscript Received on April 25, 2020.

\* Correspondence Author

**RituSindhu\***, Senior Professor, Computer Science and Engineering department in Galgotias University, Gautam Budhh Nagar, U.P.

**KartikTomar**, B.Tech Computer Science and Engineering department in Galgotias University, Gautam Budhh Nagar, U.P.

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an [open access](http://creativecommons.org/licenses/by-nc-nd/4.0/) article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>)

# Improving Success Rate of Transactions in Cryptocurrency Payment Channel Networks

We also use the split payment method in which a payment is divided into several smaller pieces and sent independently so the capacity variations are minor. Payment split can be done in two ways: 1)

Equal share in which the payment is equally divided among the various gateways; 2) Proportional split in which the payments are divided considering the inbound/outbound ratio of each gateway.

The implementation and testing of the proposed approaches of route selection and flow maximisation under various payment scenarios showed that by using proper gateway selection with the proportional split payment method we can improve the rate of success of transaction significantly as it maintains the balance in the network increasing its efficiency.

This paper is organised in the following manner: Section II discusses the related work in the literature; Section III explains the concepts used in payment networks and the assumptions, Section IV defines the problem statement and our approach to it. In Section V we perform the result analysis and in Section VI we conclude the paper.

## II. RELATED WORK

A number of methods have been suggested for the implementation of payment channel networks. Some of these have also already being implements like the Lightning network (LN) for Bitcoin and the Raiden for Ethereum. LN is the most widely used with more than 10 thousand nodes and 30 thousand channels. Source-routing is used for transactions in LN. Spider uses packet-switching routing technique and also uses split payment method for transactions. It uses congestion control to increase the success rate of transactions. It chooses the path for transactions specifically to maintain the balance of the network. The payments are first queued at the spider nodes and transferred only when sufficient funds are available. SpeedyMurmur is using an embedding-based routing algorithm to focus on the privacy of the transactions. Flash uses a distributed routing algorithm to counter the constantly changing balances. it distinguishes between smaller and larger payments. The smaller payments are sent randomly on paths which are already decided. But for the larger payments, it requests the nodes to find channels with sufficient funds to carry out the transaction and then it splits the payment into smaller units. Revive finds cycles in the network and makes a user to send a payment to itself through the cycle to balance the depleted channels. Flare and SilentWhisper uses landmark routing. In this only some selected Landmark nodes have the routing table for the complete network and the rest of the nodes only know how to reach these landmark nodes. In this paper, we focus on the strategies of path selection for transactions to maintain the balance of the entire network and make it more efficient. To achieve this gateway selection strategies are very important as they have a great impact on the overall success rate of the payment channels.

## III. PRELIMINARIES AND ASSUMPTIONS

### A. Blockchain and Bitcoin

Blockchain is a distributed database in which the basic data structure is called a “block”. In the case of bitcoin, a block is made up of transactions, timestamps, nonce, hash of the block and the hash of the previous block. For a block to be declared valid, the hash of the block should be smaller than a number which is decided by the total computational power of the network. A miner can find a suitable hash result by changing the nonce value in the block. When the block is declared to be valid it is distributed in the network. When other nodes are done validating that block, the next block is validated.

### B. Off-Chain Payment Channel

The average time to validate a block for Bitcoin is 10 minutes which makes it unusable for the end-user as it is not possible to use it for day-to-day spending. For example, if a person uses Bitcoin to pay for a small amount like a cup of coffee, they have to wait for about an hour for the transaction to complete and if the payment is made during congested time they might have to pay a transaction fee larger than the amount of the coffee itself which is not a desirable case for both the customer and the shop owner. To counter this problem a mechanism was developed called the “off-chain payment channel” that uses the smart contract mechanism in blockchain. In this two users agree to establish a business and sign a contract by transferring collateral to a shared multi-signature address and create a channel by publishing it on the blockchain. This contract is called the “Hash Time Locked Contract” (HTLC). A pre-image is sent to the recipient by the payer to initiate a payment. If the recipient successfully replies to the pre-image, the contract is validated and the money is transferred. The main advantage of off-chain mechanism is that the users do not need to publish every transaction on the blockchain and since there is no frequent on-chain transaction, there are no unexpectedly high transaction fees rather the fee can be nullified if the peers agree. Hence, there will be only two on-chain transactions: one to create the channel and the other to close the channel.

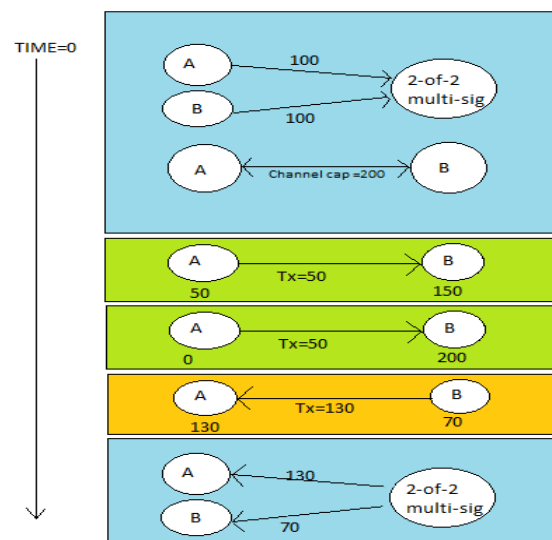


Fig. 1 Illustration of a payment channel

In such a channel, the direction of the payments matter.

Two flows from opposite direction nullify each other's capacity consumption.

This is shown in

Fig. 1. At Time=0, a channel between both the parties is created. Both put in 100 units of currency to make the capacity of the channel 200 units. After A makes two transactions consecutively of 50 units each, A cannot transfer any more money as the directional capacity of A to B is zero. B sends back 130 units to A and after closing the channel they get their respective shares from the multi-signature address.

### C. Payment Channel Networks

The off-chain payment method can be further used to create a payment channel network idea. As shown in Fig. 2, we assume that A and B have a channel, and B and C also have a channel. If in case, A wants to pay C, then he can hash-lock the amount he wants to send and forward it to C through B. As C knows the answer to the pre-image, it will get the money from B by disclosing it. The usefulness of HTLC appears here, as C discloses the answer, now B also know the answer and when B replies to the pre-image, it gets its share for helping in the transaction by A. by this method, one node can reach any other node in the network through multi-hop payments creating a payment channel network.

The payment channel network shown in Fig. 2 is made of three types of nodes: end-user node, gateway node and router node.

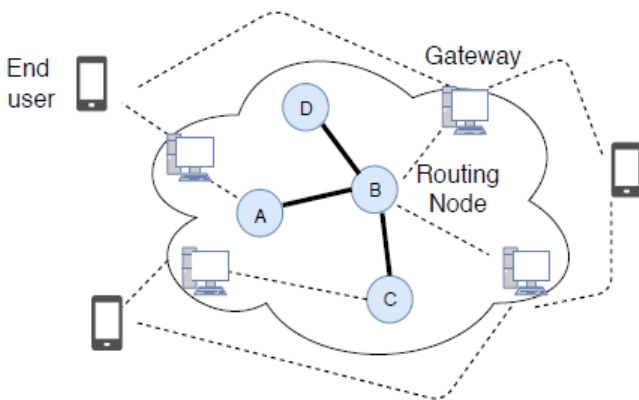


Fig. 2 Payment channel network

- 1) **End-User:** The end-user most commonly makes payments to purchase something and very rarely receives payments. The end-user connects to gateway to get access to the rest of the network and it has to maintain a minimum amount of funds in its channel with the gateway to continue doing transactions. It should open a channel with a gateway which can connect it to others continuously and cheaply and to that, the best option is to connect with a gateway which will have direct transactions.
- 2) **Gateway nodes:** These are the stores or the receivers which usually expect to receive payments and rarely send payments. These nodes also help end-users to make transactions to other nodes in the

network. Gateway nodes must have a good connectivity with routing nodes. Their primary aim must be to sell their products in cryptocurrency rather than to earn transfer fees for other transactions.

- 3) **Routing node:** These nodes are responsible for connecting different gateways in a network. Each routing node tries to have high number of connection and they are also responsible for maintaining the balance of the channels throughout the network. This makes them a hub point of the payment network.

### D. Assumptions

In this paper, we assume that the nodes present in the payment channel network are connected to each other with off-chain payment channels. The off-chain channel capacity of two nodes represents the amount of currency deposited by these two nodes in the multi-signature address. We distinguish every node as end-user node, gateway node and routing node, although any node can act as anyone of it, because we want stimulate real-world scenarios where the cash flow is mostly from a customer to gateway.

We build up our work on the assumption that the nodes in the network use an existing routing protocol to broadcast the weights to the rest of the network and to carry out a transaction. We do not put our focus on the efficiency of the routing protocol as our main aim will be to determine the gateway to start the payment process. Another assumption is that each node is aware of the complete topology of the network to calculate the path and also that the updates about the channels are broadcasted throughout the network.

We would like to note that our approach is designed to be used in any payment channel network. We use LN as our main subject of example to explain some problems and introduce some concepts but the proposed approach is not solely developed for LN.

## IV. PROPOSED APPROACH

### A. Problem Motivation and Overview

In a payment channel network, a payment is initiated by an end-user to a store through a gateway. Unless there is a direct channel between the sender and the receiver, this payment first goes through a gateway then a routing node and then arrives at the destination. This multi-hop transmission will not be completed if there are insufficient funds in the channel. If the payments are highly directional, the funds in the channel will shift to one particular side. The gateway nodes are highly prone to channel balance exhaustion.

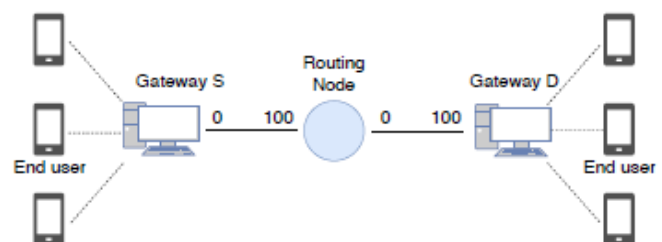
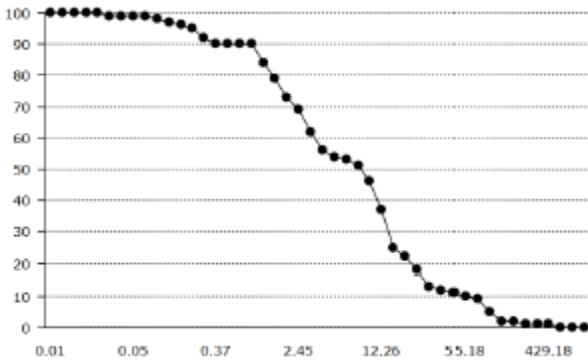


Fig. 3 Depleted channels cause disconnection



If a gateway uses its complete outgoing capacity, it cannot further operate as a transit node till it receives some payment. For example if a store is having a busy day and receives many payments, the channel between the store and its routing node will get depleted and the store will not be able to receive more payments. This is shown in Fig. 3. The gateway S has outgoing capacity of zero so the users connected to it cannot make any payments. The gateway D has consumed its full incoming capacity, so it cannot receive any more payments.

We use the fact that a payment can be initiated from various points in a payment channel network to address this issue and balance the network. We cannot change the destination of a payment but we can change the gateway through which the payment is being initiated. For example, if a node is receiving a high amount of payments, the same node should be used as a gateway to send payments and avoid using nodes with less outgoing capacity. By doing this, we can maintain the balance of a payment channel network.



**Fig. 4 Success rate vs amount of the payment**

Another problem with the Lightning network implementation is that it only supports atomic payments, i.e. a single wide enough channel is used to carry a payment which results in major unbalancing of the channel in case of larger payments as shown in the Fig. 4 which indicates that the success rate decreases sharply with the increase of the amount of payment. We suggest that the same amount split into multiple smaller parts going through several different paths may prevent such depletion in the success rate.

To summarise, we suggest that smart selection of gateways combined with split payment methods can help to improve the overall success rate of the payment channel network and also increase its efficiency.

### B. Smart Gateway Selection

We propose a method of gateway selection that effectively utilises the multiple connections between the end-user and the gateways to create a more balanced network. In this method, our main focus is to achieve and maintain the balance of the gateway channels so none of the gateways gets blocked from sending or receiving payments. Therefore, we call it inbound/outbound ratio approach. We use Alg. 1 to calculate the inbound and outbound capacity of each connected gateway and then to recognise the one with the lowest ratio. So, the inbound/outbound ratio is used as the primary factor for smart gateway selection. The links without enough transmission capacity are not considered in

path recognition as we are using single payment method. The transaction process start once the gateway is decided.

### Algorithm 1: Inbound/Outbound Ratio

```

1: Input: G= Gateway List, C= Connected Directed Graph
2: for each directed edge, (m,n), in C do
3: weigh (m,n)= bal (n,m)/{ bal (m,n) + bal (n,m)}
4: end for // weights are calculated
5: maximum = Integer.Min
6: for every gateway connection, k in G do // Finding the gateway
having the highest in/out ratio
7:   in_total=out_total=0
8:   for every channel, (u,v), of gateway k do
9:     in_total = in_total + capacity (v,u)
10:    out_total = out_total + capacity (u,v)
11:   end for
12:   ratio = in_total / out_total
13:   if ratio greater than maximum then
14:     maximum = ratio
15:   end if
16: end for
17: Path = ShortestPath (C,from = maximum, to = destination)
18: Output = Path
    
```

**Alg. 1 Calculation of inbound/outbound ratio**

### C. Split Payments

To increase the success rate of transactions, we explore various methods of split payments. We can transfer big amounts using a number of gateways by using split payment methods. Each of these methods can be applied simultaneously with the smart gateway selection methods we discussed earlier. Depending on the transaction policy, these methods may cause increased transaction charges. But these methods still prove to increase the success rate of transactions by maintaining the balance in the payment channel network.

There are various methods of splitting a payment like dividing it into very small units or equal bigger chunks and sending these parts through various gateways. In this paper we will design two different splitting models discussed below:

**Equal Share:** In this model, we divide a payment into a number of parts which are equal in size and split them among various gateways capable of carrying these amounts. Using multiple-source single-sink max-flow algorithm, we first determine that there is enough total capacity from all gateways to the destination. If the calculated capacity is not enough, the splitting of payment also does not take place. If the capacity of the channels is sufficient, then the payment is divided into multiple smaller parts depending upon the number of available connections to the end-user. The total capacity, fee or ratio of the gateways is not considered. All the gateways carry equal amounts but if the capacity of one of the channel is not enough, then the rest of the channels compensate by carrying more amount. The basic idea is to utilise all the available channels fairly rather than putting the entire load on a single channel.

**Proportional Split:** In this method, the inbound/outbound ratio of the gateways is considered for splitting the payment rather than using the number of gateways.

Similar to the Equal Share method, we still check if there is enough capacity to make the payment from the source to destination, but the division of the total payment amount is done on the basis of proportionality to the inbound/outbound ratios of the gateways. Basically, the gateway with less outbound capacity compared to its inbound capacity carries fewer amounts. This method helps in maintaining the balance of every gateway connected in the payment channel network. Note that this method is also in line with our gateway selection approach in the previous section as they both consider the same criterion. So we expect this method to further increase the success rate of transaction.

**V. PERFORMANCE EVALUATION**

In this section we present the experiment setup and results of the proposed approach.

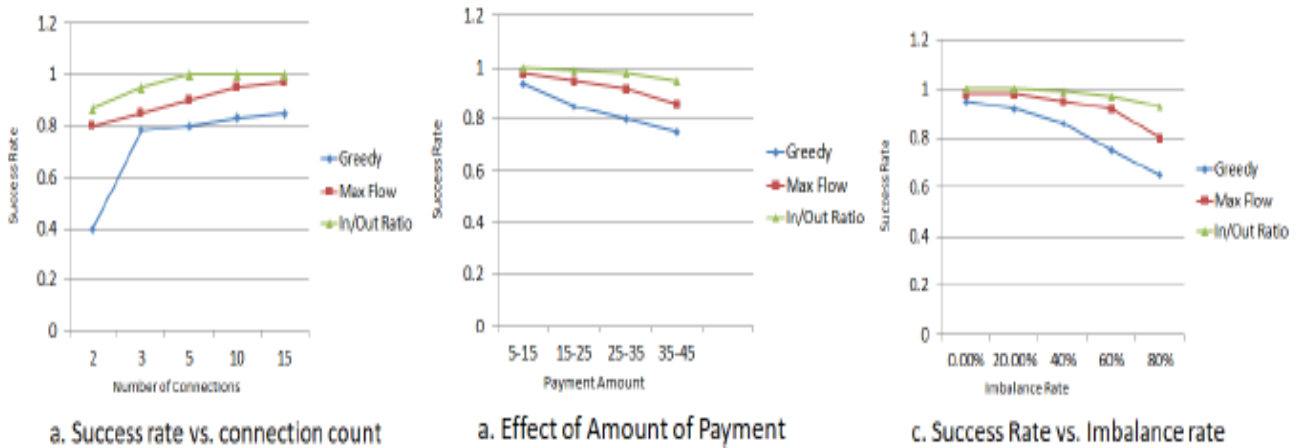
**A. Experiment Setup**

To run experiments and measure the defined metrics, we created a payment channel network simulator in Java. The

**Table I: Experiment parameters**

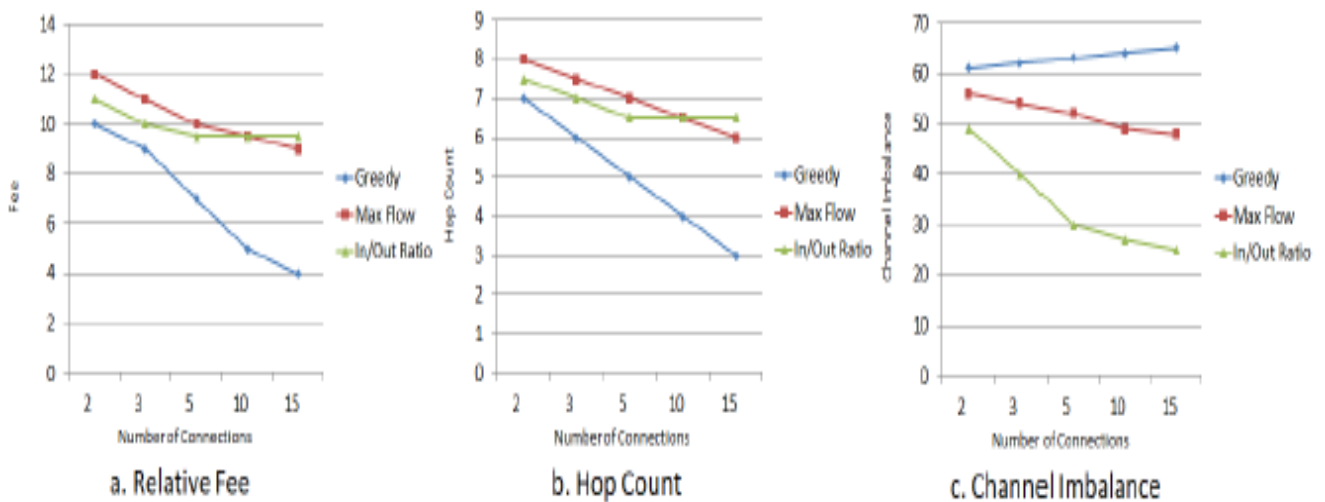
Total Number of Nodes	100
Degree	3
Channel Capacity range	50 to 150
Payment Amount Range	5 to 85
Total Number of Payments	5K

Network Configuration: The results are based on a random regular network consisting of 100 nodes. Each node has a degree of 3. The channel capacities of the link between the nodes are uniformly distributed between 50 and 150.



**Fig. 5: Experiment results for success ratio**

necessary parameters required to run the experiment are listed in Table I.



**Fig. 6: Experiment results for other factors**

Payment Files: We executed a total of 5000 transaction and each node carried out an average of 50 transactions. The amount of transaction is selected randomly during the

# Improving Success Rate of Transactions in Cryptocurrency Payment Channel Networks

experiment between a specified range of 5 and 85. Sources and destination are not necessarily the same, i.e. A can transmit to B but can also receive from C.

Payment Transfer:

The path is calculated by each node using the specified methods and the payment is sent through intermediate nodes by decrementing the amount from each channel used and increasing it in the opposite direction. In the first set of experiment, we use only single payment while in the second part we use split payment.

Experiment Run: Each payment file is run with various seeds which randomized the initial channel capacity and payment amount. Thus, the results are aggregate of 75 randomized tests on the network.

## B. Performance Metrics

We use the following metrics:

- 1) Average Success Rate: This metric measures the average number of transactions that were successful for the whole network.
- 2) Fee: This shows the total amount of fee which is to be paid to the node owners for a transaction.
- 3) Network Imbalance: This measure the difference between the capacities of a channel in the two directions. The average of the whole network is used to show the overall imbalance.
- 4) Hop Count: This measures the total number of hops during a transaction. This is important if

there is a base fee or a delay from the intermediate nodes.

## C. Benchmarks

To analyse the results of our approach, we implemented the following approaches and compared the results. In all these baseline algorithms, we assume that the changes in weights are updated throughout the network similar to our approach.

- 1) **Greedy Approach:** In this the weights of the links are considered and the one with the least weight is selected for the transaction. The user prioritises the minimisation of fee. The available capacity of the path or the imbalance problem is not considered.
- 2) **Maximum Flow:** In this, the Ford-Fulkerson algorithm is used to calculate the maximum flow from each gateway to the destination and the channel with the highest capacity is selected. The idea is to leave available funds in the channel so as to increase the chances of finding a proper path for future transactions.

## D. Experiment Results

In the first batch, we implement greedy, max-flow and in/out ratio methods using single payments and compare the results. Fig. 5 shows the success rates whereas Fig. 6 shows other metrics. These results are average of various imbalance rates with the payment amounts varying between 25 and 35.

- 1) **Success Rate Result:** Looking at the rate of success with respect to the number of connections we observe that in case of each

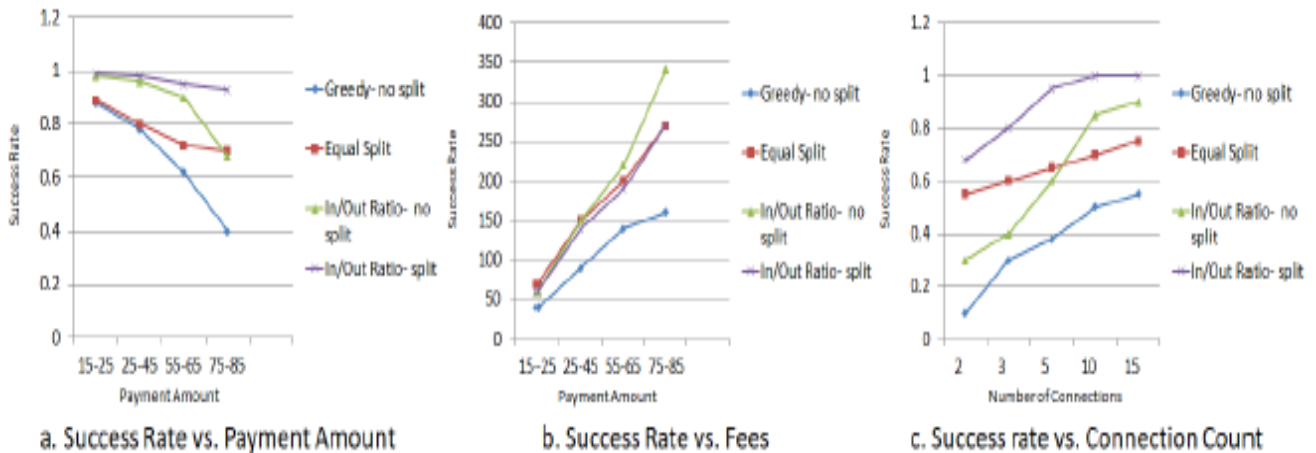


Fig. 7 Experiment results for split payments

approach the success rate increases with the increase in number of connections. Based on the results, the increase in success rate diminishes after the number of connections goes higher than 3. The performance of greedy approach is the worst but, as shown in Fig. 6, it gives the cheapest transaction cost. Its success rate is low because its main priority is to determine the cheapest path available. The max-flow method chooses the path with the highest capacity and hence its results are higher than the greedy method as choosing wider paths will leave space for upcoming transactions but it incurs higher fee per transaction as shown in Fig. 6a. The in/out ratio approach gives the best performance in terms of success rate as its focus is to keep the gateways balanced for end users. The success rate of our proposed approach almost reaches

100% with 5 connections in the current setup. Fig. 6c shows that this approach also keeps the channels balanced in both the directions.

Fig. 5b shows the impact of payment amount on the success rate of the transactions. All the approaches face a decent in the success rate with the increase in amount but our approach still outperforms both the other approaches.

By observing Fig. 5c, we see that after an imbalance rate of 40%, our approach still maintains high success rate.

- 2) **Impact on other Metrics:** In Fig. 6a, we compare the fee with increase in the number of connections for all the 3 approaches. In case of the greedy approach, the fee decreases

3) linearly as increasing a connection will help in finding the cheapest path. For our approach, it stays steady after a certain point as we try to assign payments based on gateway channel balances.

Fig. 6b shows the number of hops in the whole transaction process and we can see that the trends are similar to the fee results as they both are directly related.

In Fig. 6c, we quantitatively represent the average imbalance in the network with the increase in number of connections. We can see that the greedy method causes more imbalances in the network with increase in the number of connection whereas our method relieves it especially after 5 connections.

4) Result for Split Payments- In another experiment, we show the success rate results of split payment solutions by comparing it with the previous method for changing payment amounts and number of connections. We implement both the split methods, namely: Equal share and Proportional split. Hence, we compare 4 different methods: 1) greedy approach without splitting; 2) our approach without split; 3) equal split; 4) proportional split.

In Fig. 7a, we observe the success rate of each approach after varying the payment amount from 20 to 80. We infer that with smaller amounts, splitting does not show any significant improvement but with the increase in the payment amount the difference becomes significant for both the split methods. The proportional splitting performs the best among all the four methods.

When we look at the fees shown in Fig. 7b, the difference is not significant and the trends are similar. Note that we did not apply any base fee.

Splitting might cost more if there is a base fee in the payment channel network.

The Fig. 7c shows the success rate with respect to the number of connections with a payment amount ranging between 75 and 85. Again, the approach with the proportional split method performs the best. When the number of connections goes above 5, we see that the approach with no splitting performs better than the equal split approach because after that point there are enough connections available to provide a natural splitting effect.

## VI. CONCLUSION

We presented a smart gateway selection method for end-users in a payment network channel with an aim to increase the rate of success of transactions. To further increase this success rate, we also proposed a split payment method. We also analysed the effectiveness of this method by comparing it with two other commonly used approaches. The analysis showed that our gateway selection method and splitting method is highly effective in increasing the success rate of transactions in a payment channel network.

From this analysis, we inferred that keeping the gateways open in terms of channel balance to send and receive payments proves to be an important objective and it can be achieved by considering the inbound and outbound capabilities of a channel. We also concluded that split payment is also highly effective especially for higher payment amounts.

## REFERENCES

- 1 Nakamoto. Bitcoin: A peer-to-peer e-cash system. Technical Report, <https://bitcoin.org/bitcoin.pdf>, 2006
- 2 Ripple. <https://ripple.com/>, 2017.
- 3 M. Boone and V. Jadon. The bitcoin lightning network: Scalability of off-chain instant payments. Technical Report, <https://lightning.network/lightningnetwork-paper.pdf>, 2016.
- 4 L. Gilbert, W. Haller, N. James, F. Wlachos, and N. Zeldovich. Algorand: Scaling byzantine agreements for bitcoin. In Proc. ACM SOSP, 2014.
- 5 The Raiden Network. <https://raiden.network/>, 2018.
- 6 Mosby, T., Patanayak, V., Singh, K. and Kalyan, M., 2016. Blockchain technology: Beyond cryptocurrency. Applied Innovation, 2(3-21), p.11
- 7 Sivaraman, V., Venkataraman, S.B., Alizadeh, M., Fanti, G. and Viswanath, P., 2018. Cryptocurrency routing with spider network. arXiv preprint arXiv:1809.04308.
- 8 Rate of Transaction of Bitcoin. <https://www.blockchain.com/en/charts/transactions-per-second>, 2017.
- 9 H. Williams. Ethereum: a secure decentralised transaction book, 2013.
- 10 Historical fee chart of Bitcoin. <https://bitinfocharts.com/comparison/bitcoinmediantransactionfee.html>.
- 11 Kyle Walker, Christian Benson, Itty Vyal, Adam EfeGence, Mary Juels, Ahmed Kosba, Andrew Miller, PrateekSaxena, Elaine Shi, Emin Gun Sirer, Dawn Song, and Roger Wattenhofer. Scaling decentralisedblockchains. In FC, 2015.
- 12 Y. Sompolinsky and A. Zohar, Secure Transaction Processing with high success rate in Bitcoin, pp. 513619, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.
- 13 Narayan, A., Bonneau, H., Fitch, M., Miles, J. and John, M., 2016. Bitcoin technologies: a comprehensive introduction. Charlton University Press.

## AUTHORS PROFILE

**RituSindhu**, She is a researcher and a senior professor in the Computer Science and Engineering department in Galgotias University, Gautam Budhh Nagar, U.P. who has published more than 60 research papers in various reputed popular journals and conferences like SCOPUS, Springer etc.

**KartikTomar**, He is a final year student of B.Tech Computer Science and Engineering department in Galgotias University, GautamBudhh Nagar, U.P.