Bitcoin Lightening Network: a Distributed Network for Scalability of Bitcoin*

Shahla Atapoor

Supervised by Prof. Eero Vainikko
University of Tartu, Estonia
shahla.atapoor@ut.ee

January 3, 2019

Abstract

Distributed systems are one of the most popular approaches to deal with trust issues in centralized and infrastructure systems. Bitcoin is the well-known digital currency that uses distributed networks to omit the trusted third party in traditional payment systems. In addition, unique opportunities that Bitcoin provides, there are some big concerns with this digital currency that makes it impractical in most cases. For instance, transactions in Bitcoin do not scale, and for example users have to wait at least 40 minutes (4 blocks $\times$ 10 min for each block) to get sure that their transaction are recorded in the blocks.

During last few years, there have been valuable efforts to deal with the mentioned concerns in the Bitcoin network. In order to keep the main benefits of Bitcoin network, that comes from the nature of distributed systems, solutions based on distributed networks have been more popular. In this report, we give a high level explanation of *Bitcoin Lightning Network* [PD16], which is one of the popular and promising solutions to cope with the scalability and latency of Bitcoin. The solution proposes to use a distributed network of nodes with payment channels on top of Bitcoin network to speed up transactions and increase the capacity of the network.

1 Introduction

During the last few years cryptographic currencies (a.k.a. digital coins) have become very popular and attracted lot of attention. The most famous and valuable one is Bitcoin which is introduced by Satoshi Nakamoto in 2009 [Nak08]. In digital coins, there is no need to rely on any trusted third party and transactions are broadcasted to all nodes in a distributed network and the transactions are stored by all nodes on the ledger. Blockchain is a chain of the blocks which each block is connected to the next block by putting the hash of previous block. There are bunch of transactions in each block which contains the address of origin, destination and the amount of transactions. Bitcoin is pseudonymous which means that there are the address of users not the name or explicit identified information of users Fig.1.

The Bitcoin blockchain holds incredible guarantee for distributed ledgers, however, the blockchain as a payment network, itself, cannot cover the world's trade in the near future. The blockchain is a gossip protocol whereby all state changes to the ledger are broadcast to all members. In the blockchain protocol majority of users agree on the balance of each user. Informing all the nodes about every single transaction, which occurs globally, may construct serious issues on the ability of network in handling all global trade transactions. It would be good if we could manage to have all transactions decentralized and secure.

1.1 Issues with Bitcoin

One of the big issues in Bitcoin, is the limitation on number of transactions that can be done in a particular time (e.g. in a second). It is believed that the limitation comes from the block sizes, that currently is 1 MB, therefore there are several discussions on the specification of block sizes among the Bitcoin community. More precisely, it is not possible to have more than around 7 transactions with current 1 MB block size of the Bitcoin which is not enough even for a small city with small population. The number of supported transactions in each second for another known digital coin, Ethereum, is 20. In real world, the number of transactions in each second is much bigger than 20; for example, based on some annual reports of payment companies, in 2017 PayPal and VisaNet respectively processed on average 193 and 1,667 transactions per seconds. Surprisingly, Visa company has done some testes on their system and based on their reports, their system can man-

*This report is prepared in partial fulfillment of the requirements for the course *Distributed systems Seminar* (MTAT.08.024) in fall 2018 and it is presented as a seminar at Institute of Computer Science, University of Tartu (on 11 December 2018).

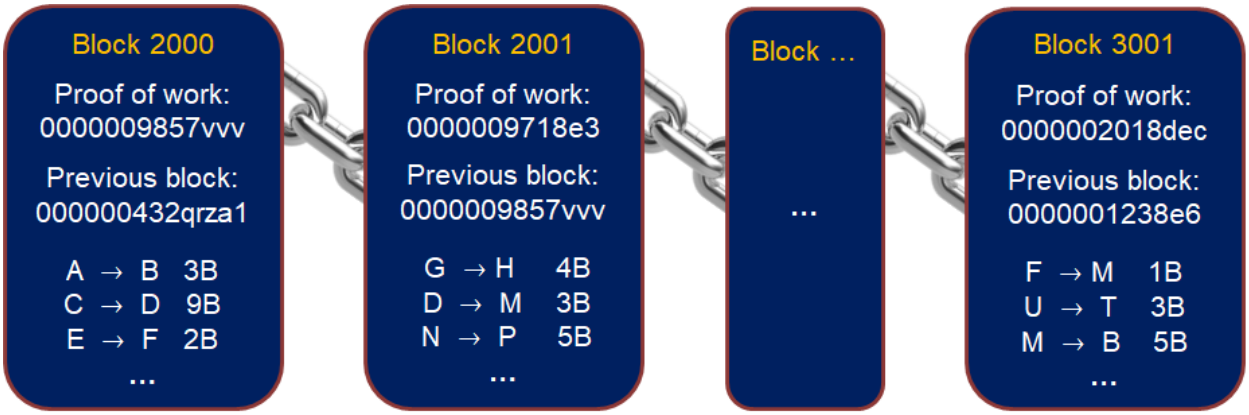

Figure 1: Bitcoin blockchain is a series of blocks, where each block contains: 1) transactions, 2) a nonce value (the answer for a proof-of-work puzzle) and 3) hash of previous block.

age more than 56,000 transactions per second [BEP] which makes it more practical than current digital coins.

On the other hand, at the moment there are several complains about Bitcoin transaction fees and waiting time for each transaction among clients and Bitcoin service providers (e.g. wallet providers). Basically, currently a common Bitcoin transaction needs to pass 6 blocks and consequently confirmations from miners before completely be sure that they have gone through, and since each block takes 10 minutes to be mined. So, in order to get sure about the status of a transaction, a user has to wait for around an hour which is not practical in majority of cases in real life. Another issue that usually users are complaining is the transactions fees, which are getting more expensive in time [BEP].

In 2017 CoinDesk researched on the increase of transaction fees and obtained that transaction fees over the distributed-ledger network have risen nearly 19-fold, from 13 cents per transaction on average in the second quarter of 2016 to $ 2.40 in the same quarter of 2017.

Proposed Solutions. In order to tackle the aforementioned issues, several solutions have been proposed and some of them are very practical.

1.2 Bitcoin with Larger Blocks

There are couple of solutions proposed regarding to the Bitcoin block size issue. A naive idea is to remove the limitation on block size, which is known as *Bitcoin Unlimited*. An another alternative, *Segregated Witness* recommends to increase the size of blocks two times. But, in the community of Bitcoin there are some disagreements and people are not sure which direction to take as yet. Among the community members, the support for *Bitcoin Unlimited* is around 36% and the support for *Segregated Witness* is about 27% [BEP]. It should be noticed that the value of support is calculated as a percentage of the total coins mined over a particular period of time, with several miners not demonstrating explicit support for both of the mentioned approaches.

Expanding block size in Bitocin causes some concerns, such as having less validators. That means, the smaller number of nodes will ensure accuracy of the blockchain, and consequently less number of companies or nodes will participate in mining that will lead to have more expensive transactions and more importantly we will get far from the main goal of the Bitcoin network, which is decentralization.

1.3 The Bitcoin Lightning Network

In order to deal with scalability and latency issues in the Bitcoin network, in another direction, *The Bitcoin Lightning Network* (www.lightning.network) is a distributed-network based approach that is proposed by Joseph Poon and Thaddeus Dryja in a white paper [PD16] in 2015. Its determination is available on Github (https://github.com/lightningnetwork/lightning-rfc).

The Bitcoin Lightning Network proposes to build a network of distributed network with its nodes hav-

ing some active micropayment channels between each other and they use the micropayment channels for direct payments without hitting the blockchain. In fact this network can be built over the Bitcoin blockchain and nodes will touch Bitcoin ledger once they needed a third trusted party. Their analysis show that the network empowers instant transactions among all the participation nodes and it can practically solve the scalability issue of the Bitcoin.

Main idea behind the Bitcoin Lightning Network. The main idea behind the Bitcoin Lightning Network is that, two nodes do not need to broadcast all their transactions to the network if they have regular repeating transactions. In other words, it is not necessary for all nodes in the Bitcoin network to get aware of all regular transactions between two particular nodes. It is much more convent that two nodes of a distributed network make a personal micropayment channel between each other, do unlimited and fast mutual transactions without touching the ledger and informing to other nodes in the system. They use blockchain as a third trusted party in the cases that they do not agree in some status or they want to close the channel.

Notice that creating a micropayment channel between two nodes, makes it possible to have overload of transactions between them and postpone distributing the points of interest for future, without danger of counterparty by default.

In the rest of the report, we introduce micropayment channel used in the Bitcoin Lightning Networks and then explain the construction of the target distributed network.

The Bitcoin Lightning Network for Scalability Issue. A micropayment channel creates a connection between two nodes, and the Bitcoin Lightning Network creates a distributed network that each node of it is a Bitcoin client and has an open (active) channel with some nodes around herself. When such a system is made, the network expresses that it is possible to play out every second inside the system as much as required. Technically speaking, in the Bitcoin Lightning Network, each node is member of the Bitcoin system and edges are channels between the clients and only the transactions with uncooperative channel counterparties are broadcast to the Bitcoin blckchain. In this manner, the constructed network that would work on Bitcoin network, properly can solve the scalability issue of the Bitcoin network.

2 Micropayment Channels

Micropayment channels allow to perform transaction between two nodes without touching the ledger. Constructing such channels is possible utilizing multi-signature accounts, smart contracts and decentralized timestamping framework. To setup an payment channel, first two parties set up a multi-signature account (which requires signatures of both parties to spend a coin from that account). This created account keeps some value of Bitcoin. The value transferred to the account then is broadcasted to the Bitcoin blockchain. A graphical portrayal of a payment direct is appeared in Fig. 2

Notice that, by determining timeframes that certain states can be broadcast and later invalidated, it is possible to create complicated smart contracts via Bitcoin transactions, which is an essential concept behind the bidirectional payment channels that both Alice and Bob can pay Bitcoin to each other.

Unidirectional Payment Channels. Unidirectional payment channel is made between two parties as a direct payment channel from only one party to another party. In unidirectional payment channels, there is a multi-signature address that requires both Alice and Bob's signature in payments from that account. A graphical representation can be seen in Fig. 2.

When Alice wants to send 1 ₿ to Bob, before any action, she gets Bob's refund signature from their joint account. So in the worth case, Alice loses her coin for a specific period of time. So basically Bob makes a nLockTime discount mark and signs and sends it to Alice (which can be spent by Alice after the specified number of blocks). Alice either can sign it now or can keep it to sign it later, she can do so once she needs it Yet, this is critical to see that Alice should keep this refund signature on her hard drive. Since she realizes that in the most pessimistic scenario she will get her coins after passing nLockTime. This graphical representation is shown in Fig. 3. At this point, once Alice got the assurance that she will get her coins after passing nLockTime, she signs and sends her coins from her single account to the Alice and Bob multi-signature account. At that point, once the coin is in their multi-signature address, Alice can focus on payment to Bob with no nLockTime. So right now, Alice signs new transaction from multi-signature deliver to Bob, and sends it to Bob. As an occurrence, Alice says that, there is 1 ₿ in the joint

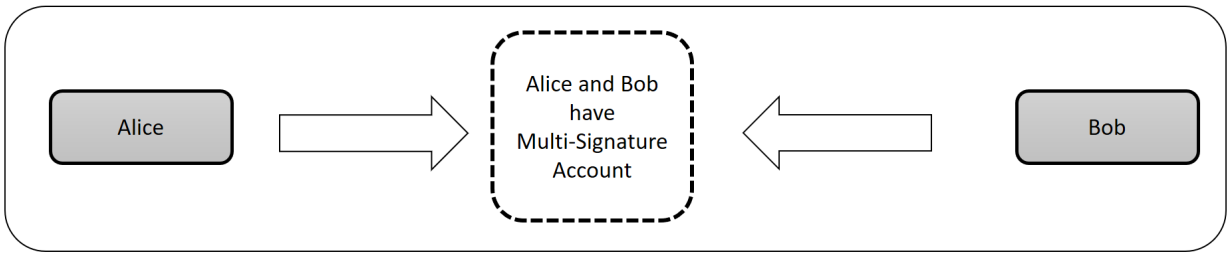

Figure 2: Alice and Bob's micropayment channel with multi-signature account.

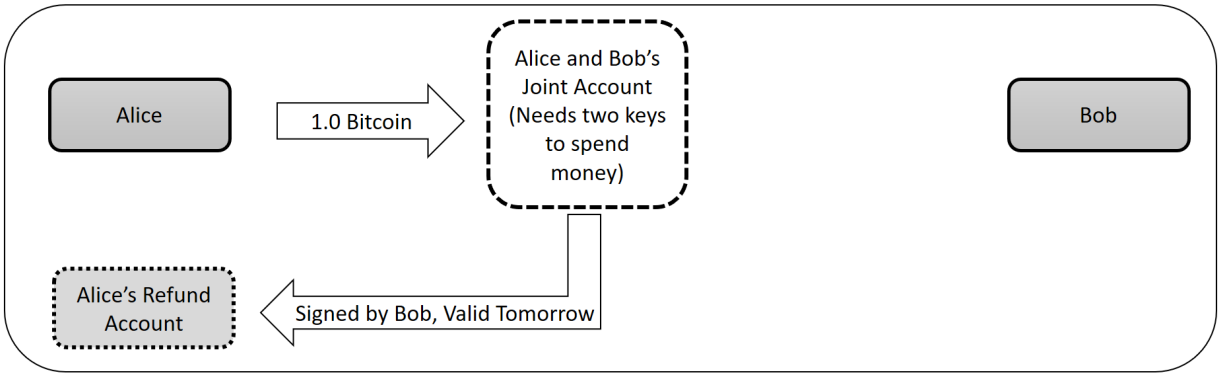

Figure 3: A micropayment channel with 2-signature account among Alice and Bob. Condition of the channel in the wake of marking the discount transaction from the shared service by Bob.

account, and I will pay 0.1 ₿ to you Bob and will remain 0.9 ₿. Now, when Bob got the transaction, it is a legitimate transaction, since Bob has not yet signed it. It implies that Alice cannot broadcast this transaction on the blockchain, but Bob can consider this as an payment. Note that since now Alice signed this transaction, Bob would be able to sign the transaction and broadcast to the ledger. Doing this will close the channel. Bob does not close the channel since he realizes that the channel will stay open until the end of nLockTime. Fig. 4 shows this progression graphically.

To use benefits of an active channel, Bob waits until the end of his refund nLockTime signature. Later when Alice needs to pay him again, they make a similar transaction and update the balance. For instance, Alice pays 0.2 ₿ for Bab and 0.8 ₿ for herself. So similar to previous case, Alice alone signs it and gives it to Bob. Then since this transaction has more Bitcoin for Bob, so essentially Bob updates the old balance, but he still keeps it unpublished to the Bolckcahin. The procedure can be repeated as many times as they want before the end of nLockTime, which makes refunding possible for Alice (Note that Bob already has signed the refund transaction, and it only needs Alice's signature to be spendable). But before getting to nLockTime, Bob can sign the final transaction and since there is no lock time, so he can broadcast to the network and since both have signed the transaction from joint account, so the network sees that 1 ₿ comes from Alice account to a multi-signature account (address) and 1 ₿ goes from the multi-signature account to two addresses (Alice and Bob's individual addresses).

Bidirectional Payment Channels. As it can be derived from the name, *bidirectional payment channels* are a extended form of unidirectional payment channels, which both nodes can send transaction to each other. The Bitcoin Lightning Network shows that infact one can change the direction of a unidirectional payment channel with small modification. Accurately, the first step is as before, which Bob signs a refund transaction from the joint multi-signature account to Alice's refund account with nLockTime, that will be valid after precise number of blocks (e.g.

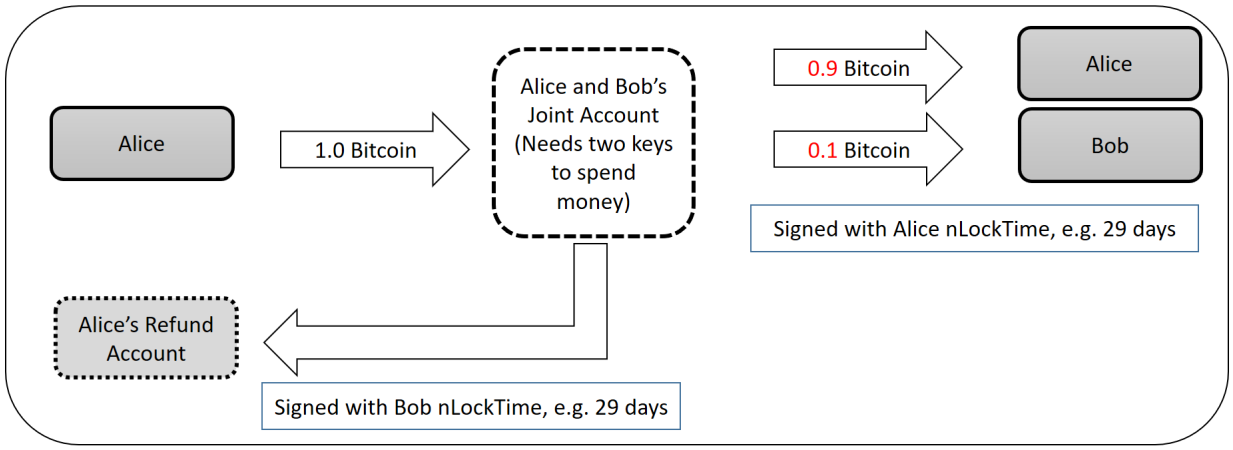

Figure 4: State of Alice and Bob's unidirectional channel after first transaction '0.9 ₿ for Alice' and '0.1 ₿ for Bob'. Note that at present the transaction just is marked by Alice.

30 days = 30 × 24 × 6 blocks). Then Alice send 1 ₿ to the joint multi-signature account. But after that, when Alice needs to spend from joint account she signs the new transaction with a lock time shorter than time lock of the refund transaction (e.g. a transaction with 29 days time lock). Fig. 5 shows the procedure graphically.

Notice that now Alice can increment this payment to Bob as many time as she wants, just she needs to keep the same time lock in new payments; See Fig. 6.

Now suppose Bob needs to pay Alice from his current available balance which is 0.2 ₿ in the example. To do so, indeed Bob can recommit to Alice and overwrite the last transaction. But he needs to sign with an nLockTime smaller than the one that Alice signed the last transaction with (e.g. in the considered example Alice signed with 29 days lock time, so Bob can sign with 28 days lock time, See Fig. 8).

Three Party Payments Assume there are three nodes Alice, Bob and Carol in the Bitcoin Lightning Network that both Alice and Carol have an active channel with Bob (As shown in Fig. 7). Alice needs to pay Carol but she does not have an open channel with Carol, so to avoid a transaction with an expensive fee she can use her active channel with Bob and Bob's active channel with Carol to transfer the coins to Carol. In this scenario, in order to get rid of trust in the system, there are two issues.

- First Bob might keep the coin of the transaction and not send it to Carol.
- Carol can claim that she never got the coins and it is not possible to verify.

The Bitcoin Lightning Network deals with the mentioned concerns with *Hash-Locked Contracts (HLC)* and *Pay to Contact* consents. As a recall, a cryptographic hash function allows an entity to easily verify that some input data maps to a given hash value, but if the input data is unknown, it is difficult to reconstruct it by knowing the stored hash value.

Using a HLC, hash functions make it possible for Alice to prove that she sent funds to Carol off-chain. Using HLCs Alice, Bob and Carl in Fig. 7 act as follows.

1. Carol picks a random number R and executes a hash function h (e.g. SHA-3) and gets $H = h(R)$. Then Carol sends H to Alice.
2. Alice uses H on her active channel to pay to Bob, but she encumbers the payment with H and this payment can only be realized if Bob can produce R. Note that in this moment R only is known for Carol.
3. Bob does the same as Alice; he uses H value and his active channel to pay to Carol, but he also encumbers the payment with H and this payment can only be realized if Carol sends R to Bob or broadcasts it to the blockchain.
4. When Bob got the R from Carol, he can send it to Alice and realist the payment.

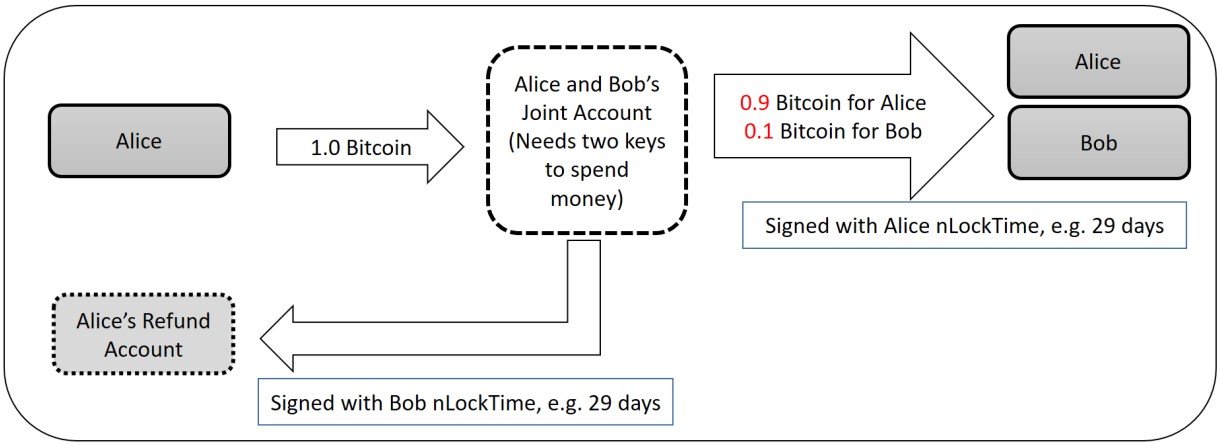

Figure 5: Province of Alice and Bob's bidirectional channel after first exchange as "0.9 Ƀ for Alice" and "0.1 Ƀ for Bob". Note that the exchange is marked with a nLockTime little than nLockTime utilized in Alice's discount exchange (e.g. 29 days).

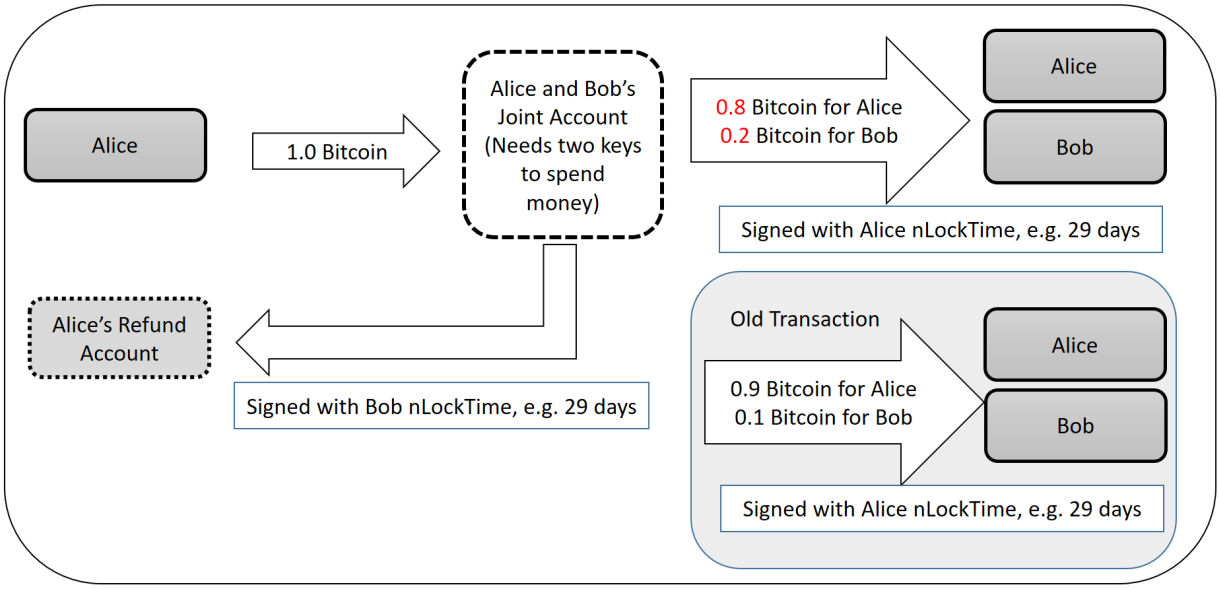

Figure 6: Province of Alice and Bob's bidirectional channel when Alice increments past transaction to the new equalization as "0.8 Ƀ for Alice" and "0.2 Ƀ for Bob". Note that the transaction is marked with indistinguishable nLockTime from first transaction (e.g. 29 days).

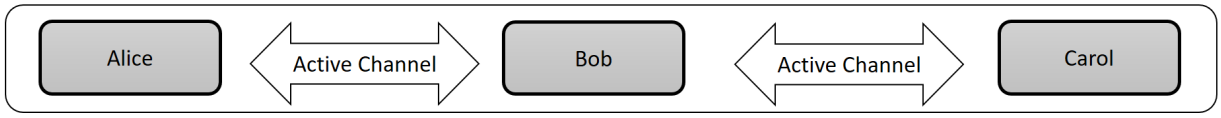

Figure 7: A three-party channel among Alice, Bob and Carol in which Alice and Carol's corner have an individual dynamic channel with Bob.

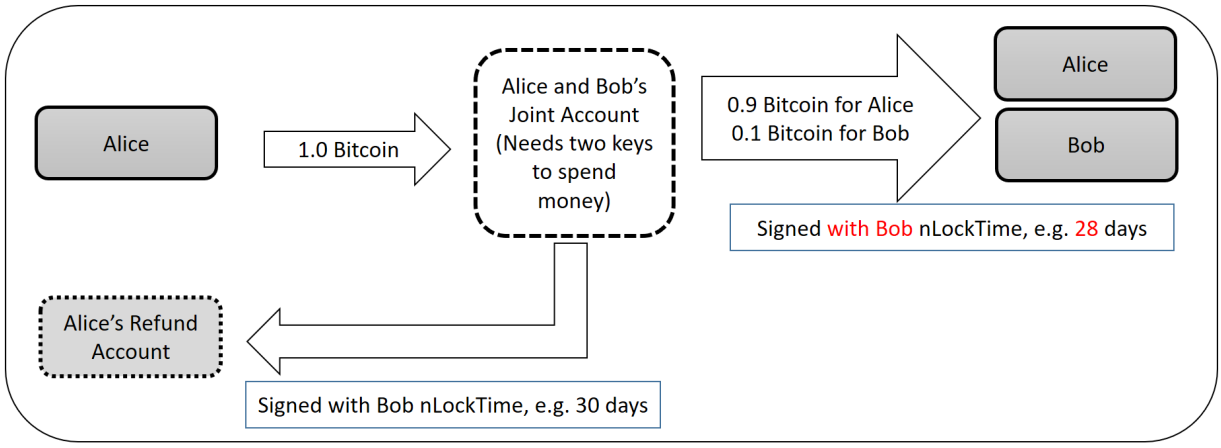

Figure 8: Territory of Alice and Bob's bidirectional channel when Bob pays to Alice 0.1 Ƀ. So the refreshed transaction would be "0.9 Ƀ for Alice" and "0.1 Ƀ for Bob". Note that the transaction is marked by Bob with a nLockTime littler than the nLockTime utilized by Alice in the last transaction(e.g. 28 days $<$ 29 days).

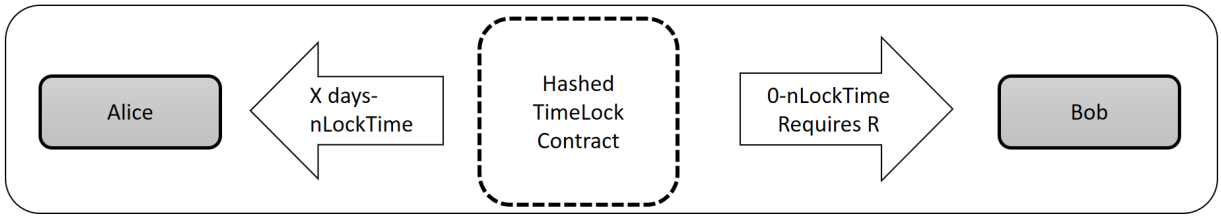

Figure 9: Contract between Alice and Bob based on Hash Time-Lock.

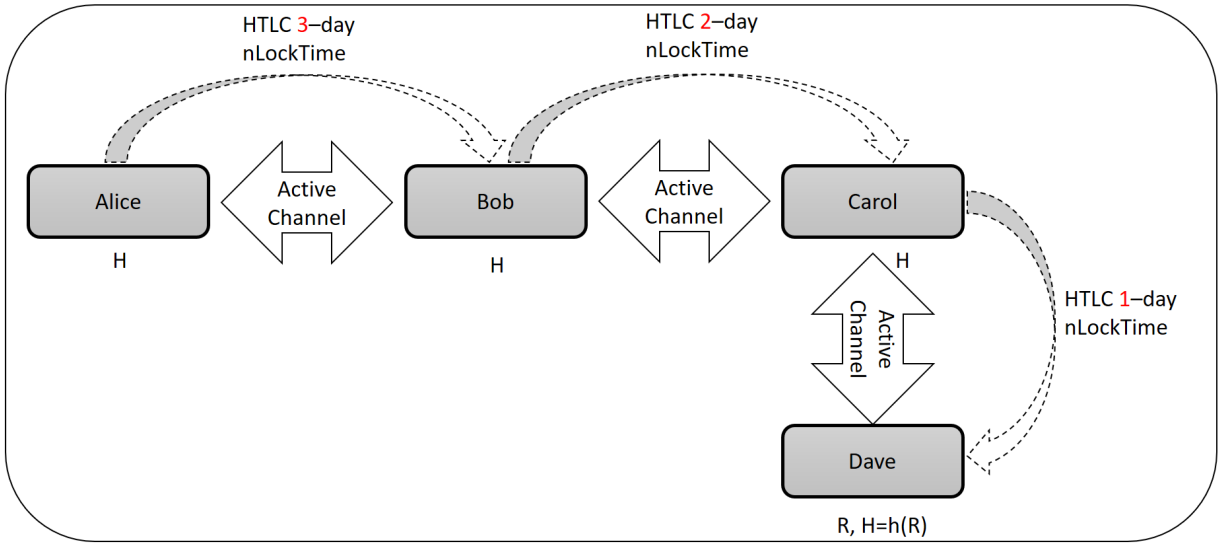

Figure 10: Transaction over BLN among Alice and Dave: Forward advances

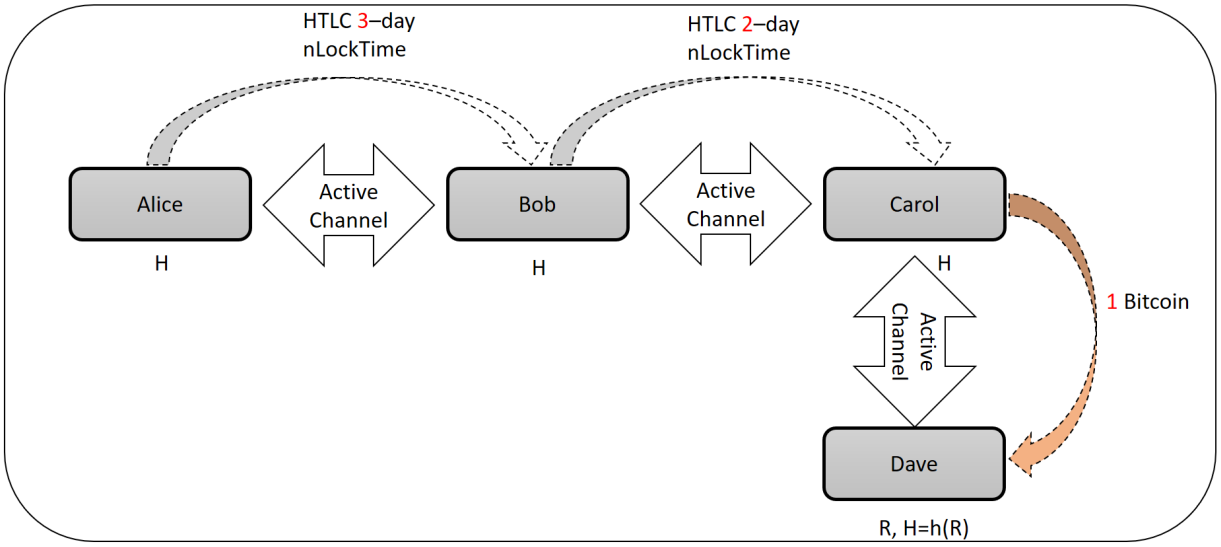

Figure 11: Transaction over BLN among Alice and Dave: Backward stage 1

By now, there is an issue with the described procedure. The issue is that if Carol refuses to disclose R, she can make some concerns for Alice and Bob's channel. Indeed, if Carol's channel expires after Alice and Bob's channel, she can steal funds by redeeming the hashlock! This issue is addressed by Hashed Time-Lock Contract (HTLC) in the Bitcoin Lightning Network and the terms of such HTLC are as,

- If Bob can send to Alice input R from the hash H within x days, Alice will pay Bob y ₿
- The above item is void after x days
- Both parties may agree to settle terms using other methods if both agree (e.g. by blockchain)
- Violation of terms incurs a maximum penalty of funds in this contract

From the terms of the smart contract one can see that Alice a Bob will make a new transaction from their channel as shown in Fig. 9.

3 The procedure of Bitcoin Lightning Network

Finally by combining the introduced payment primitives, the Bitcoin Lightning Network, makes a distributed system of transaction paths. Paths can be routed using a alternative approaches (e.g. BGP-like system), and the sender may designate a special path to the recipient. The output scripts are encumbered by a hash, that is provided by the receiver. By revealing input of hash, the receiver's counterparty will be able to pull funds along the route.

As an example, suppose that Alice wants to send 1 ₿ to Dave over the Bitcoin Lightning Network. To this aim, the procedure is as the following,

1. First Alice finds Dave through the BGP-like system
2. Dave tells Alice, "here is H, if you know R, consider your payment fulfilled", where H is hash of R; $H = h(R)$.
3. If Alice does not have a direct channel open with Dave, then she finds a route to Dave. Assume that Alice finds Bob and Carol as the intermediate nodes which can connect Alice to Dave (As appeared in Fig. 10).
4. Now, Alice and Bob make an HTLC output in the payment channel to pay Bob 1 ₿, with a 3-day nLockTime refund back to Alice.
5. Bob and Carol create an HTLC output in the payment channel to pay Carol 1 ₿, with a 2-day nLockTime refund back to Bob.
6. Carol and Dave create an HTLC output in the payment channel to pay Dave 1 ₿, with a 1-day nLockTime refund back to Carol. Note that Dave can get 1 ₿ if he discloses R to Carol.

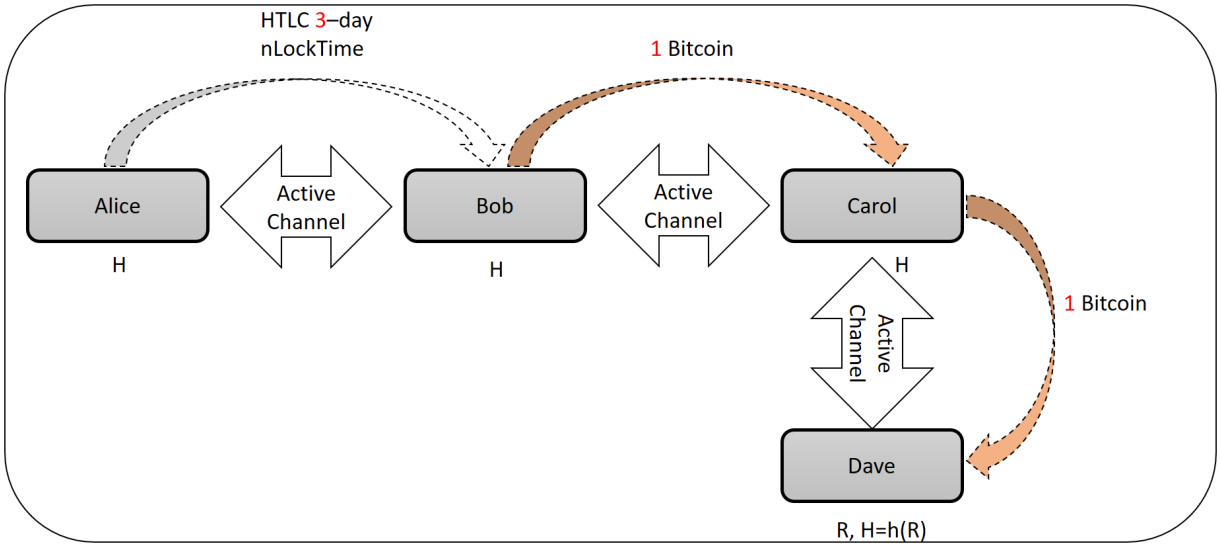

Figure 12: Transaction over BLN between Alice and Dave: Backward step 2

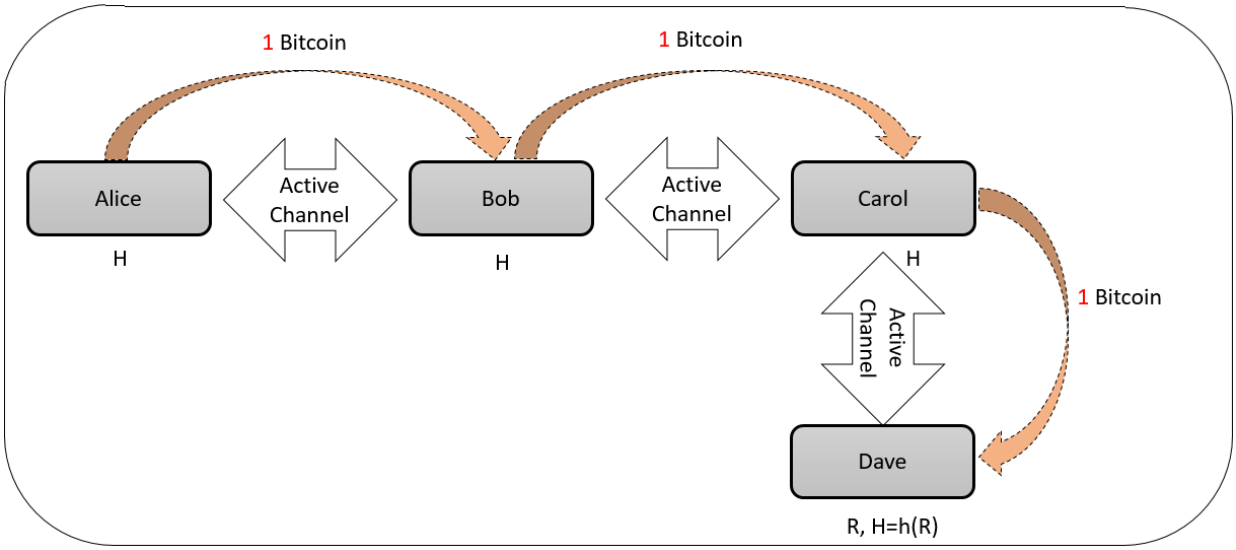

Figure 13: Transaction over BLN between Alice and Dave: Backward step 3

7. Dave discloses R to Carol within 1 day. Carol now has enough information to pull funds from Bob. Carol and Dave agree to update the balances in the channel instead of broadcasting on the blockchain (As shown in Fig. 11).
8. Carol reveals R to Bob within 2 days. Bob now has R to pull funds from Alice. Bob and Carol agree to update the balances in the channel instead of broadcasting on the blockchain (As shown in Fig. 12).
9. Bob reveals R to Alice within 3 days. Alice can prove she sent funds to Dave. Alice and Bob agree to update the balances in the channel instead of broadcasting on the blockchain (As shown in Fig. 13).

In the case nodes behave honestly, all forward and backward steps in Fig. 10-13 will be done out of Blockchain. But if one of parties does not cooperate, the next party in the channel can quickly close out the channel and broadcasts all pending transactions to the Blockchain. In that case, the honest party redeems the ₿ by revealing the HTLC output and R on

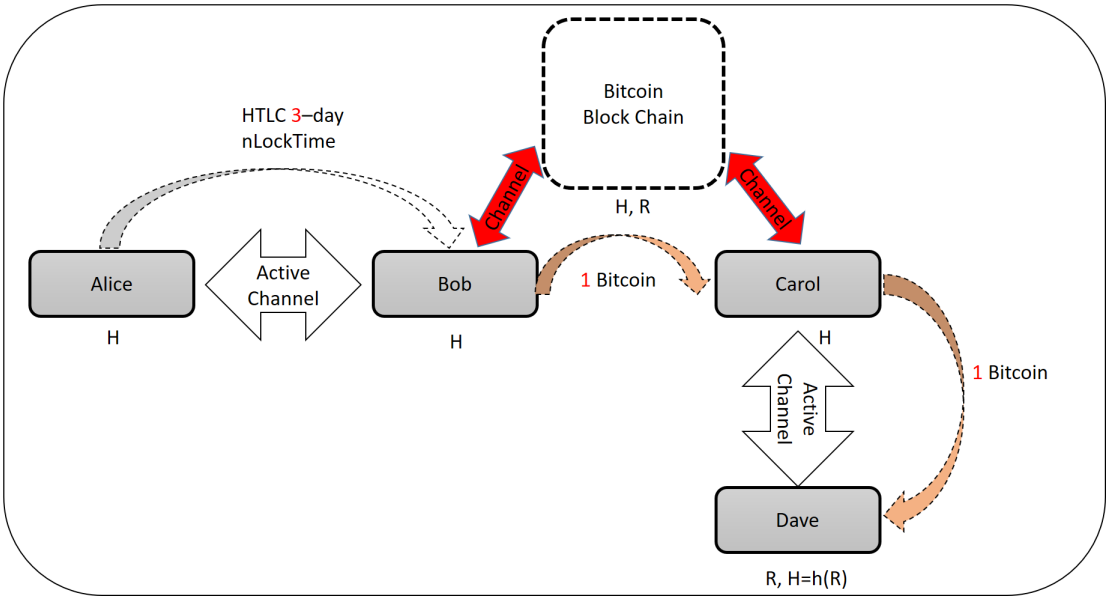

Figure 14: Broken channel in BLN

Bitcoin Blockchain. For example, in the above example, if Bob and Carol do not cooperate, Carol quickly closes out the channel and broadcasts all pending transactions in the channel onto the Blockchain. As a result, Carol redeems the 1 ₿ by disclosing the HTLC output and R on Bitcoin blockchain. This procedure is shown in Fig. 14. Then Bob observes R from the Blockchain and can pull money from Alice off-chain.

Note that if R never gets revealed, timeouts occur sequentially from the destination. The decrementing timeout ensures that everyone can receive funds contingent upon it being sent.

4 Conclusion

From a few years ago that Bitcoin got more resolving, there have been continuously various tries on issues such as scalability, latency and expensive transactions. *Bitcoin Lightning Network* is one of the practical solutions that is proposed by Joseph Poon and Thaddeus Dryja in 2015 [PD16] and aims to use a distributed network with payment channels to deal with the aforementioned concerns.

In this report, we had a short high-level explanation of the main motivation behind the Bitcoin Lightning Network, and we also investigated the primitives behind the construction of such distributed network. We observed that payment channels (including Unidirectional and Bidirectional) and HTLCs (Hash-Time Lock Contracts) play an important role in the construction of network.

We observed that transactions in Bitcoin Lightning Network are similar to the Bitcoin's transaction, just users do not touch the ledger in each transaction (they postpone broadcasting transaction to future), instead they use a micropayment channel and HTLCs between each other to perform the transaction offline, and once they had an issue that they needed a third party they use blockchain.

We discussed, currently by publishing all transactions, the Bitcoin network only can handle about 7 transactions per seconds that is very low in comparison with current real systems such as Visa or Master payments. The report discussed that using blockchain for all the payment transactions (maybe by increasing size of blocks) needs a huge power of miners to mine much more blocks and there are just a few miners who have this amount of power, meaning a fall back to the centralized systems that are not desired.

Comparing the proposed *Lightning Network*, using micropayment channels, to the real Bitcoin network that we have today, we can say that there is need to 133 MB blocks (presuming 500 bytes per transaction and 52560 blocks per year) in order to give the ability of making two channels per year with unlimited transactions inside the channel for 7 billion people.

Acknowledgment: Thanks to Karim Baghery for his helpful discussions on reading the seminal paper.

References

[BEP] Bitcoin and Ethereum vs Visa and PayPal - Transactions per second; Available on: https://mybroadband.co.za/news/banking/206742-bitcoin-and-ethereum-vs-visa-and-paypal-transactions-per-second.html.

[Nak08] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. 2008.

[PD16] Joseph Poon and Thaddeus Dryja. The bitcoin lightning network: Scalable off-chain instant payments. *See https://lightning. network/lightning-network-paper. pdf*, 2016.