CISC 322/326
Assignment 3: Architectural Enhancement of Bitcoin Core
April 12, 2023

Manny Cassar	m.cassar@queensu.ca
Sawyer Proud	19sqp@queensu.ca
Cameron Krupa	19cmer@queensu.ca
Marcus Tantakoun	20mt1@queensu.ca
Duncan Scanga	19dms7@queensu.ca
Eric Jin	e.jin@queensu.ca

Table of Contents

Abstract

The Architectural Enhancement Report examines the proposed enhancement of the Bitcoin Core: the addition of the Lightning Network feature. We suggest two possible ways to implement such a feature within the Bitcoin Core. We then proceed to perform a SAAM architectural analysis, which involves assessing the impact of each implementation on various stakeholders and non-functional requirements. Once we have settled on a particular implementation, we go on to discuss the possible effects that its implementation will have on the overall system and present two relevant use cases. We then discuss the plans for testing the two enhancement implementations. Finally, the report ends with a discussion of the possible risks of implementing this enhancement and a brief discussion of the lessons learned.

Introduction

The implementation of Bitcoin and cryptocurrencies is crucial to fulfilling the promise of a no-trust, anonymous, decentralized form of transaction. The increasing scale of Bitcoin and its reference software implementation Bitcoin Core has dramatically increased the expense and pace of using the network. This results in a poor user experience and prohibits the technology from replacing instant transactions currently available through intermediaries like Visa and Mastercard. Getting over this hurdle is a priority for developers. To tackle these issues, we propose the Lightning Network, which is another layer built on top of Bitcoin. Ideally, it can be easily integrated into the Bitcoin protocol and allow for instant micropayments with low transaction fees.

In our first report (A1), we analyzed the conceptual architecture of Bitcoin Core. Our analysis was mainly based on the project's documentation, online papers, and forums describing its workings. At this point, we relied on the developers' idealized conceptual framework and had not yet dug into the actual source code. From this, we could understand the system's basic structure, its subcomponents, and how they work together.

In our second report (A2), we explored the concrete architecture of Bitcoin Core through an analysis of its structure from the open-source code repository in great detail. Using the software tool Understand, we analyzed Bitcoin Core’s software dependency tree to revise our first-suggested conceptual architecture and highlight any modifications made. We then conducted a reflexion analysis by examining the absences, convergences, and divergences between our conceptual and concrete architectures. Based on the derivation process described above, we arrived at the following concrete architecture:

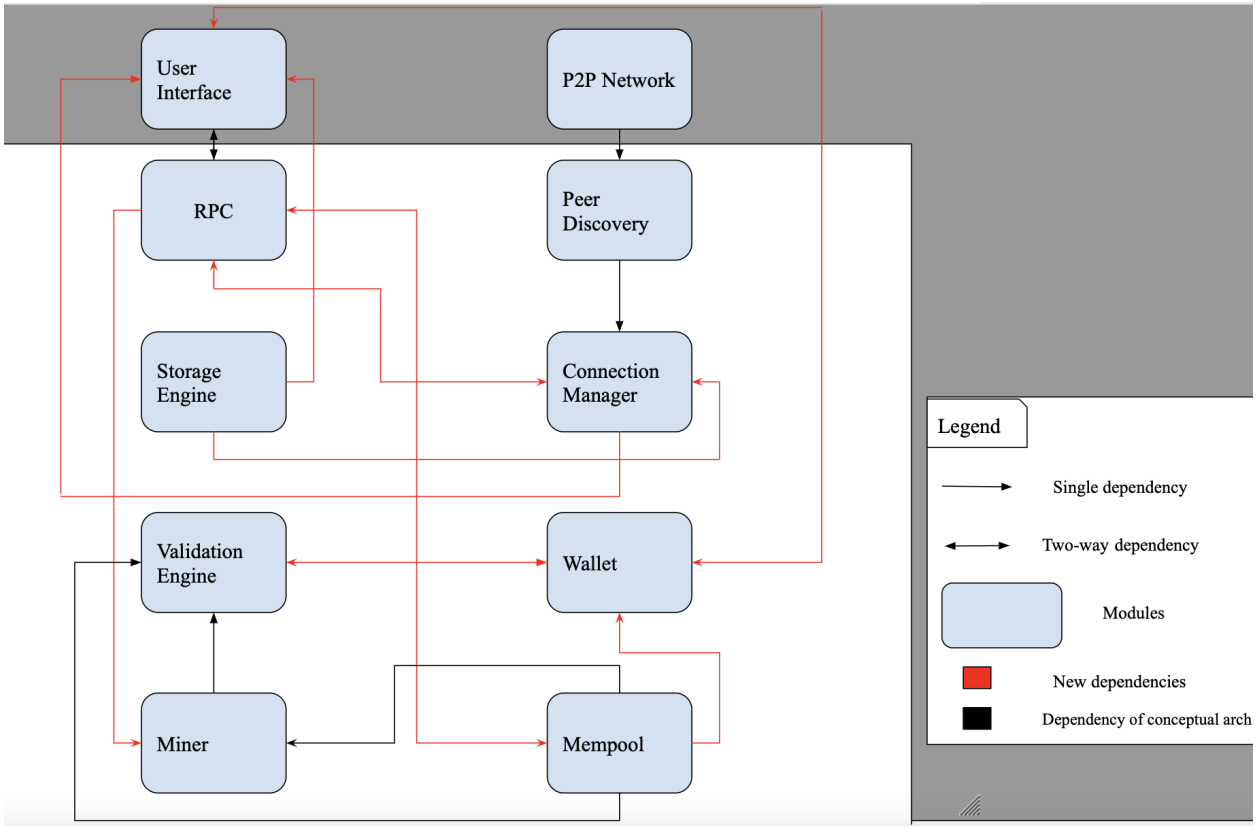

Figure 1: Concrete Dependency Graph

In this report (A3), we wish to provide an enhancement on Bitcoin's Software architecture, with its current scalability problem in mind. First, we outline and compare two possible implementations of the Lightning layer. Then, we discuss how this technology will affect Bitcoin's users, miners, business owners, and developers. We also utilize two use cases and their corresponding sequence diagrams to show how these stakeholders can take advantage of the Lightning technology. Next, we suggest a method for testing the Lightning Layer's useability and effectiveness and discuss potential associated risks. Finally, we conclude our findings and present the lessons learned throughout the architectural enhancement process.

Possible Implementations

Option 1: New layer

A possible implementation of the lightning layer could involve an alternate interface to communicate with a network of new wallets. Servers on this new interface layer would process transactions between these new wallets and periodically update accumulations of these lightning transactions to the Bitcoin layer, updating the blockchain. These new wallets would have to be Lightning compatible. Once you have bitcoins in the Lightning wallet managed by the Lightning client, you open and deposit funds into a "channel" with another Lightning system user. This

channel serves as an intermediary between the two parties where transactions can happen immediately without leaving the coins in a limbo state that requires a miner to verify the transaction. In this way, the Lightning layer facilitates a peer-to-peer transaction while remaining decentralized. Additionally, these Lightning clients can implement their own features without affecting the security or functionality of the blockchain or Bitcoin network. Examples of features that may be desirable are cold storage, two-factor authentication, or funds recovery through a secret phrase. Overall, this external Lighting wallet and client implementation improve the transaction process by increasing speed, concurrency, and decreasing fees while also opening the door for additional features.

Option 2: Rust LDK

The second option for a Lightning Network feature that will enhance Bitcoin core is minimizing the dependencies in the system to allow a flexible implementation of Bitcoin core. In this implementation, we would have simple APIs, language binding, demo apps, and other features that will produce a more efficient system in Bitcoin core through a Lightning Network. Rust lightning would provide a more secure memory database and develop a more secure software system in Bitcoin core. LDK would allow Bitcoin Core to simplify its libraries and tools, making a quicker transaction process. The LDK is very useful for simple and complex problems in the Bitcoin core system, such as using blocks to perform the payment process. The network would be more secure using Rust lightning, yet easier to use with LDK, which will ultimately enhance the performance and reliability of the system. Since Rust Lightning can develop applications and services, Bitcoin Core will be provided with a much more convenient system. Altogether, the system's functionality will improve and create a more flexible implementation.

SAAM Analysis

Based on the two possible implementations of the enhancement, we will now consider how these affect the stakeholders. To evaluate this enhancement's impact on the stakeholders, we will list the major stakeholders and then identify the most important non-functional requirements regarding the enhancement.

Bitcoin Core Developers

Since we are proposing a new network in which transactions will be made, the developers responsible for implementing this change and maintaining the system will be stakeholders in the proposed enhancement. Developers will be interested in the difficulty of adapting the system and the required maintenance.

Non-Functional Requirements:

- *Compatibility:* The new feature must be implemented without major changes to the system's architecture and compatible with the current libraries.

- *Maintainability/Security:* The new feature must not introduce any new security vulnerabilities or bugs requiring further maintenance.

Bitcoin Miners

Since Miners validate the on-chain transactions, transactions across the Lightning Network will not require Miners. Furthermore, the Miners will be used to open and close the channel between two users but will not need to be involved in the transactions that occur in this channel. Similarly, the benefits of the Lightning Network are its scalability and ability to process small transactions quickly. Since Miners take a percentage fee of the transactions, smaller fees may result in smaller revenue for Miners.

Non-Functional Requirements:

- *Development Cost/Feasibility:* The new feature must be viable for Miners to continue making similar revenue. In practice, this means that the Miners cannot have their revenue reduced by more than 25% on average.
- *Compatibility:* The new feature must be able to work alongside the current implementation of the Miners so as not to create new problems.

Business Owner

With this new enhancement, the scalability of the Bitcoin Core architecture will be increased, which may be interesting to certain business owners. More businesses may choose to accept Bitcoin as a payment. However, business owners are not trained to use Bitcoin, so the new enhancement should not greatly change how they send and receive transactions.

Non-Functional Requirements:

- *Ease of Use:* The proposed enhancement must not increase the level of understanding of the system. In practice, the current users should be able to continue using the system the same way, but the transactions will be performed on the Lightning Network.
- *Security:* The improved system cannot create new vulnerabilities that could compromise the system's integrity.

Bitcoin Core Users

The enhancement proposed would mean users could send smaller transactions faster, which may open more possibilities for potential use cases for users. Users will not want to see a decrease in security or reliability due to the new enhancement. The enhancement may add functionality for users to send more frequent smaller transactions, but this will only be meaningful if the transaction time is decreased by the enhancement.

Non-Functional Requirements:

- *Performance:* On average, the time to send a transaction between users must be smaller or the same as the current transaction time while using the Lightning Network.
- *Reliability/Security:* The implementation must be as reliable as the current system, with no increased risk of transaction failures or security vulnerabilities.

Evaluation of the Two Implementations

The first implementation of a Lightning Network is an alternative interface to communicate with a new network of wallets. The Bitcoin Core developers would have a compatible system as all this implementation is doing is adding transactions by updating the blockchain. This process would be as secure as the previous one since we are enhancing the use of new wallets by simply adding them to the blockchain so that the system will stay as secure as before. Bitcoin miners would be provided with improved revenue since new wallets will be added in a more efficient manner, making the process quicker and a better experience for the consumers. The lighting system will be compatible for miners since the only change is how we add the new wallets to the blockchain. Business owners will love the new system as acquiring new consumers will be much easier since the cost will be reduced, and adding them to the system will be easier than ever. The security for business owners will stay the same as they will be provided with a successful network and know that their new clients will have a stable and secure transaction network. To close it off, bitcoin core users will have a much easier experience in joining the Bitcoin network with a system that is reliable for the safety of their currencies.

The second implementation of a Lightning Network is rust lightning and LDK. For Bitcoin developers, the enhanced system would be compatible with the current system as we are adding security and functionality to the network. The system will maintain security as rust lightning allows an enhancement to the memory and software security. Bitcoin miners' costs will not be reduced as the system will be more efficient, allowing more users to transact bitcoins through the network. Since the system is just adding security and efficiency, the system will be compatible without the use of miners, cutting out the middleman. Business owners are able to use the system with ease as nothing will change for the users, but more so, how the system works to transact bitcoins. There are no new vulnerabilities that will be added to the system since rust creates a better environment that is not only more secure but efficient. Lastly, Bitcoin Core users can send transactions faster, which will increase the benefit of using Bitcoin Core.

Determination of Chosen Option

The chosen option for the enhancement of Bitcoin core's network is the alternative interface to communicate with new networks of wallets. This is most beneficial to the system since the developers will have to implement a more efficient manner of adding to the blockchain. The lightning wallet will transact between lightning wallets and not require an extra verification step.

Effects of the Enhancement

Having decided on a specific implementation of the lightning feature, we now summarize the possible effects that our implementation will have on the rest of the system. For the most part, the effects should not be too impactful to the overall P2P architecture. It should be possible to add the Lightning Network component without affecting many of the other components – as users can still use the standard method of Bitcoin transactions.

Effects on the Current Architecture

- *Wallet:* the Wallet component may need to modify itself to create and manage LN-specific transactions. Specifically, it will need to be able to interface with the LN nodes. This might require a new UI and API to allow users to create and manage lightning channels, monitor their states, and receive/view lightning payments.
- *Validation Engine:* VE is responsible for ensuring incoming transactions are valid according to consensus rules; however, the implementation of Lightning Networks may need to be modified to support the validation of LN-specific transactions (i.e. opening and closing transaction channels).
    - When a user opens/closes a channel, a special transaction is broadcasted on the Bitcoin network that funds the channel. Thus, the VE may need to recognize and validate according to the rules set by the LN protocol.
    - Need to enforce the maximum amount of funds transferred through a single LN channel and the maximum time that a channel can remain open before needing to be closed and settled on the Bitcoin network.
- *Mempool:* alongside handling LN transactions, the mempool component will need to work closely together with the VE to ensure these transactions are valid and can be included in the blockchain.
    - Might also need to be able to handle increased transaction volumes from the LN (since LN transactions can be conducted at a higher rate), which may require additional optimization and scalability measures.
- *Peer Discovery:* peer discovery component is responsible for discovering and connecting to other nodes on the Bitcoin network. If the Lightning Network is implemented, it may require a change to the process to enable the discovery and connection to these LN nodes – access to opening channels and initiating payment with another party.

Effects on Scalability

Bitcoin Scalability Problem: the Lightning Network can improve the scalability of the Bitcoin network by reducing the number of transactions that need to be processed on a blockchain. This can reduce the overall strain on the network, as well as increase throughput performance on components that regulate these blockchains (i.e., storage and validation engine).

Effects on Performance

The Lightning Network can greatly improve the performance of the overall system, but it also introduces new performance considerations, such as the need to maintain multiple channels and the need to monitor these channels for fraud/attempts to cheat using a backup service like Watchtowers.

***Effects on* Security**

The Lightning Network introduces new security considerations related to the opening and closing of payment channels. The system must implement mechanisms for dispute resolution and channel monitoring to prevent fraud or attacks by malicious actors. They may require changes to the existing transaction validation and consensus mechanisms (validation engine component). Additionally, these Lightning Networks rely on secure communication channels to prevent attacks (i.e. eavesdropping, man-in-the-middle attacks, etc.). This can be achieved by implementing additional encryption and authentication mechanisms – which may require changes to the existing network stack and communication protocols Bitcoin Core uses.

Use Cases

Use Case 1: Cooperative Transaction of Bitcoin on the Lightning Network

The first use case assumes that both users (a customer and recipient) already have a lighting-enabled wallet attached to their Bitcoin Core client, which is installed, has funds, and is running on their local machine. This use case is meant to outline the process of a normal, successful transaction using the Lightning Layer.

The process begins with the recipient initiating a payment request (invoice) in their Lightning-enabled Wallet client. To notify the customer, the Lightning Layer finds the most efficient path between the customer and recipient nodes to send the message. The customer then opens their Lightning Wallet, selects to send a payment to the recipient, and enters its corresponding information, such as the payment amount and the recipient's Lightning ID or payment request.

After this, the nodes open a payment channel along the path. This channel serves as a temporary, two-way connection between the recipient and customer nodes which doesn't require the Bitcoin blockchain. The customer lighting node then sends the payment down the channel after the customer confirms it. The payment through the channel is encrypted so that it is secure and private. Once the payment is received, they verify it and then send a deposit verification message on the path. If other payments are to be made, they can now occur, and information about this accumulation of transactions is stored in the channel.

Once the customer or recipient is done making transactions, the payment channel is closed, and the payment channel's transaction history is summarized into one transaction. Finally, this single transaction is settled on the Bitcoin blockchain.

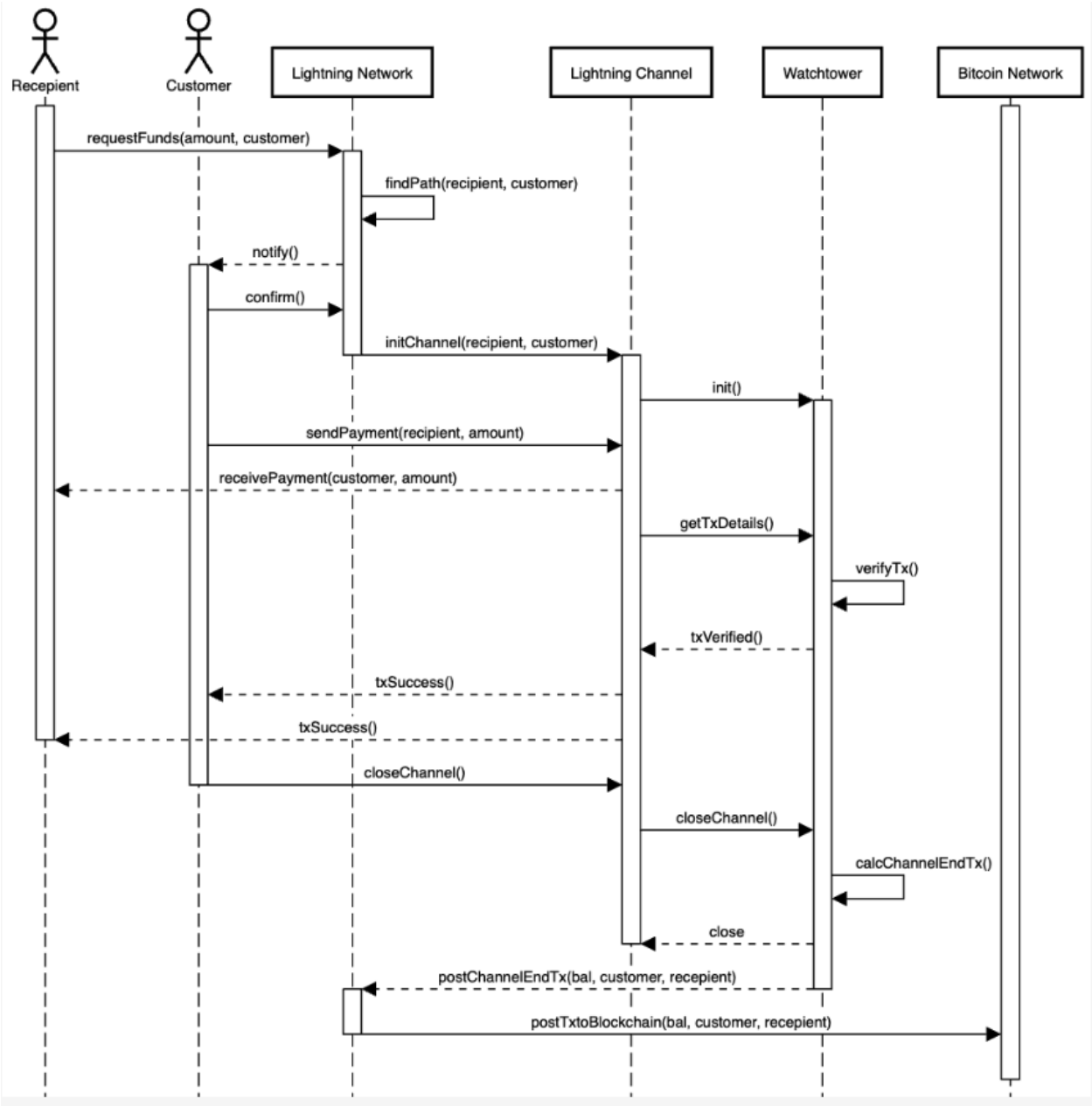

Figure 2: Sequence Diagram of a Cooperative Transaction of Bitcoin on the Lightning Network

Use Case 2: Uncooperative Transaction of Bitcoin on the Lightning Network

The second use case assumes that both users already have an account and all necessary software installed on their local machine. This use case is meant to outline the process where one of the two participants engages in uncooperative behavior, whether they are cheating or by accident (closing the payment channel early and broadcasting an old version of a transaction). The process begins when a customer opens a payment channel with a recipient and they have

both contributed some funds to it. To initiate the payment, the customer creates a payment transaction that updates the payment channel balance to reflect the payment. It then sends this transaction to the channel, which verifies and signs it, and sends it to the recipient.

The payment is now complete, but the transaction has not yet been broadcasted to the Bitcoin blockchain. Instead, both parties keep the updated payment channel state between themselves and continue to transact. However, at some point, the recipient decides to close the payment channel early and broadcast an old version of the payment transaction that reflects a lower balance. In the case of this uncooperative behavior, a Watchtower can act as a backup service that monitors the payment channel and ensures that both parties behave correctly. The Watchtower broadcasts the most recent transaction on behalf of the cheating party (customer), effectively invalidating the recipient's attempt to cheat, as the most recent transaction shows the updated payment channel state compared to the old version.

Because the customer has broadcast the most recent transaction, they are now entitled to all funds in the payment channel. This is accomplished by the cheated party creating a penalty transaction and broadcasting it to the Bitcoin network. Once the penalty transaction is confirmed on the Bitcoin network, the cheated party’s funds are returned to them.

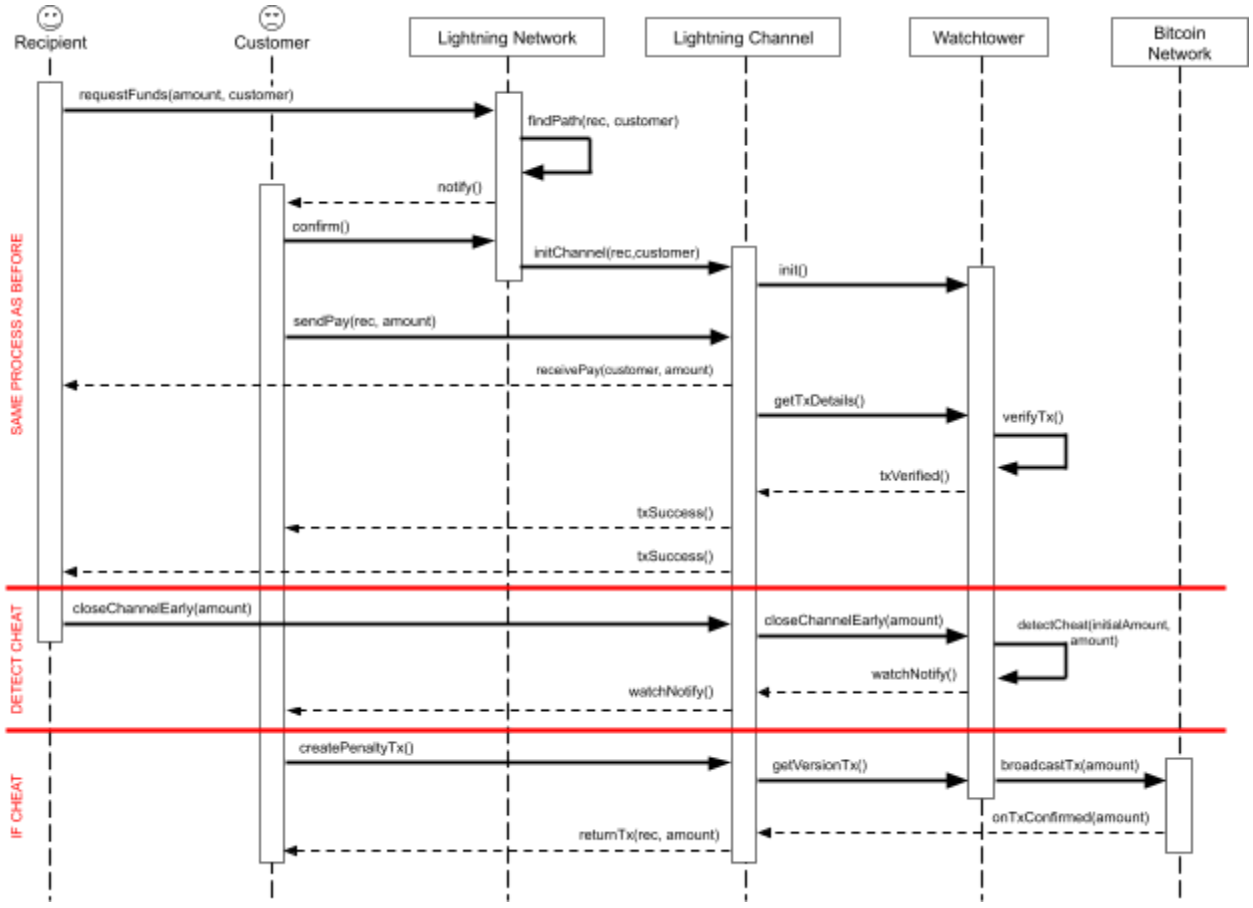

Figure 3: Sequence Diagram of an Uncooperative Transaction

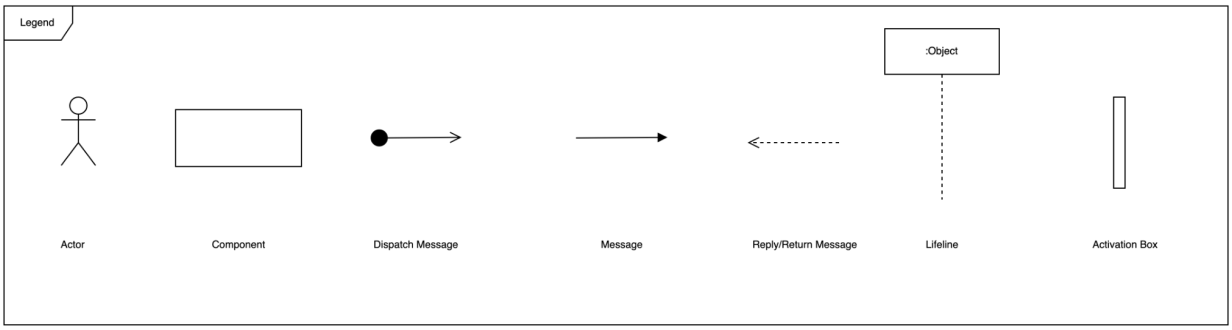

Figure 4: Legend for Sequence Diagrams

Plans for Testing

The Bitcoin Core provides testing tools for developers to test their applications with reduced risks and limitations. For development, it's safer and cheaper to use Bitcoin's test network (testnet), where the satoshis spent have no real-world value. Testnet also relaxes some restrictions (such as standard transaction checks) so developers can test functions that might currently be disabled by default on mainnet. For situations where interaction with random peers and blocks is unnecessary or unwanted, Bitcoin Core's regression test mode (regtest mode) lets developers instantly create a brand-new private blockchain with the same basic rules as testnet but with the ability to choose when to create new blocks, so developers have complete control over the environment. Conduct regression will ensure that the proposed enhancement does not introduce any new defects in previously working features.

For the new layer implementation, an alternate interface will be added to communicate with a network of new wallets. To test this implementation, we will perform unit testing on the new interface layer code to ensure that the new interface layer code is functioning correctly. The second test is to perform integration testing of new wallets with the Lightning client. We will test the compatibility of new wallets with the Lightning client to ensure that these wallets can communicate with the network and initiate transactions. We will then perform end-to-end testing of Lightning channel functionality to ensure that Lightning channel functionality is working as expected. We will verify that transactions can happen immediately and that funds can be deposited and withdrawn securely. The next step is to perform security testing of new wallets, and we will conduct penetration testing and vulnerability assessment of the new wallets to ensure that they are secure and cannot be compromised. Lastly, we will perform performance testing to ensure that the Lightning Network can handle a large number of transactions and scale as needed. They will also verify that fees are reduced and concurrency is increased.

The testing plan for the Rust IDK implementation is mostly the same as the new layer implementation. First, we will conduct unit tests to ensure that the Rust LDK code is functioning correctly. Secondly, we will test the compatibility of Rust LDK with Bitcoin core to ensure that they can communicate with each other and perform transactions. Next is End-to-end testing of

Rust LDK functionality; we will perform end-to-end testing to ensure that Rust LDK functionality is working as expected. They will verify that the transaction process is quicker and the system is more efficient. Furthermore, Security testing is to be performed; we will conduct penetration testing and vulnerability assessment of Rust LDK to ensure that it is secure and cannot be compromised. Lastly is the Performance testing; we will perform performance testing to ensure that Bitcoin core can handle a large number of transactions and scale as needed; we will also verify that the network is more secure and reliable with Rust LDK.

After the tests, we will document the results, including any issues found and their resolution. We will subsequently iterate on the test plan and retest as necessary until the proposed enhancement is fully integrated into Bitcoin Core and deemed ready for release based on the previously mentioned NFRs.

Potential Risks

Undoubtedly, adding a Lightning Network to Bitcoin Core has several benefits as Bitcoin faces challenges with transaction speed and scalability as its user base continues to grow. Implementing a Lighting Network would allow for faster and cheaper transactions while increasing scalability. While there are clear benefits to adding a Lightning Network to Bitcoin Core, such as reducing transaction fees and improving the user experience, there are also potential risks that must be considered. These risks include concerns about fees, security, and fraud. It is important to weigh the potential benefits against the risks before implementing such a change to the Bitcoin network.

Malicious Attacks

Similarly to a DDos attack, a malicious attacker can congest the network using the Lighting Network in order to delay transaction times between users significantly. This is done by creating multiple channels and forcing them all to close at once, which overloads the blockchain causing the delay. This attack can be used to steal funds from vulnerable users then, as they will be unable to withdraw their funds during this time.

Fraud

When a transaction is done on the Lighting Network, it can be done in one of two ways, as seen in the above use cases, either cooperative or uncooperative. When a cooperative transaction is done, both parties must agree that the transaction is finished, which then closes the channel. However, an uncooperative transaction allows only one of the users to close the channel, which could be used to commit fraud and scams on other users. An example would be if user A began a transaction with user B, but when user B sent bitcoin to user A instead of sending back whatever user B wanted, user A simply closed the channel stealing from user B.

Fees

Lower fees for transactions for most would be seen as a positive aspect of the implementation of the Lighting Network. However, because the fees are lower, that means that miners receive less back for mining which in turn could potentially see a loss in users as mining could become less sustainable in the long term.

Conclusion

In conclusion, Bitcoin Core transactions are becoming more expensive and slower, which has a negative impact on the user experience and prevents it from replacing instant transactions available through other applications. To address this issue, we recommend implementing a Lighting Network which adds another layer on top of Bitcoin, allowing for instant micropayments with low transaction fees. In this report, we cover two potential Lightning layer implementations, its potential effects on different stakeholders and components, two use cases, a testing strategy for its efficacy, and associated risks. Overall, the software architecture of Bitcoin is improved in this paper, increasing its scalability and performance.

Lessons Learned

Our research into a possible architectural enhancement of Bitcoin Core found that implmentaion of the Lighting Network could provide significant improvements all around with minimal downsides for Bitcoin Core. For example, the performance and scalability would be greatly enhanced without any significant change to its compatibility and security. Furthermore, its implementation improves the transaction process by increasing speed, concurrency, and decreasing fees. However, as our research showed us all these positive aspects do come with negatives such as its potential risk with malicious attacks and possible fraud and even loss of bitcoin for miners as the fees that they make money off of are diminished. Other than learning about its pros and cons could affect Bitcoin core, we also learned the SAAM analysis which showed us how to consider how the stakeholders will be affected by this enhancement instead of just looking at whether the enhancement will affect on a software level.

Data Dictionary

Term	**Definition**
Blockchain	Collection of Blocks stacked on top of each other
Blocks	Set of Bitcoin Transactions
Consensus	The name used by the source code for the ‘Validation Engine’ component

Term	Definition
Node	Computer connected to other computers, following rules and sharing information
Stakeholder	An individual, team or organization with interests in a system
Understand	Software tool used for code analysis

Naming Conventions

Acronym	Full Context
A1	Assignment 1 (Conceptual Architecture)
A2	Assignment 2 (Concrete Architecture)
A3	Assignment 3 (Architectural Enhancement)
API	Application Program Interface
CLI	Command Line Interface
LDK	Lightning Dev Kit
LN	Lightning Network
NFR	Non-Functional Requirement
OS	Operating System
P2P	Peer-to-Peer
UI	User Interface

References

"Bitcoin Core." *Bitcoin.org*, https://bitcoin.org/en/bitcoin-core/. Accessed 5 April 2023.

Howell, James. "What is Lightning Network in Bitcoin and How Does It Work?" *101 Blockchains*, 3 February 2023, https://101blockchains.com/lightning-network-in-bitcoin/. Accessed 5 April 2023.

Paszke, Antoni. "Layer 2." *Binance Academy*, https://academy.binance.com/en/glossary/layer-2. Accessed 5 April 2023.

Sharma, Rakesh, and Amilcar Chavarria. "Bitcoin's Lightning Network: 3 Possible Problems." *Investopedia*, https://www.investopedia.com/tech/bitcoin-lightning-network-problems/. Accessed 5 April 2023.