A3. Proposed Enhancement

Team 7

Slides by Jack Fallows & Jiacheng Liu

Video Link

https://youtu.be/RUa4VYCgudM

Group Overview

- Jack Taylor (Team Leader) → 17jmt5@queensu.ca
- Jack Fallows (Presenter) → 19jef2@queensu.ca
- Jiacheng Liu (Presenter) → 19jl193@queensu.ca
- Maninderpal Badhan → 19msb14@queensu.ca
- Gabriel Lemieux → 19gml2@queensu.ca
- Nil Shah → 20ns3@queensu.ca

I. Overview

Introduction

- Previous work established Bitcoin Core architecture
  - Conceptual architecture was first proposed
  - Concrete architecture was then derived from Bitcoin Core source code
- Now: identify problems with Bitcoin Core and propose amendment to architecture

Lightning Network

- Expedites transaction process by allowing transaction parties to form connection off blockchain
- Bitcoin network can handle seven transactions per second as-is
    - Lightning Network aims to improve **scalability**

II. Architectural Enhancement

Functional Requirements

- Lightning Network creates a P2P payment channel between parties off the blockchain
  - Goal is to create module in Bitcoin Core software that facilitates the creation and use of such a network
- Multiple submodules
  - **Network submodule:** manages communication between Lightning Network and Bitcoin P2P network
  - **Smart Contract submodule:** creates the smart contract that's required on creation of the lightning network channel
  - **Transaction submodule:** verifies authenticity of transaction
  - **Ledger submodule:** updates channel ledger, which will be added to the main Bitcoin ledger when the channel closes

Non-Functional Requirements

- **Performance**
  - Lightning Network must handle a large amount of transactions per second
- **Scalability**
  - Lightning Network allows for more transactions to be completed simultaneously by processing any intermediate transactions
- **Security**
  - Lightning Network transactions are less secure since they are not blockchain-validated
  - Implementation should allow users to close the network if they choose
- **Accuracy**
  - Funds transacted should be accurately tabulated

Non-Functional Requirements - Continued

- **Reusability/Maintainability**
  - Hotfixes and updates should not demand major refactoring
- **Availability**
  - Should be available to users 24/7 to accommodate various time zones
  - Networks should be able to remain open for long periods of time
- **Integration**
  - Each Lightning Network instance should be integrated into the Bitcoin Core system

III. SAAM Analysis

Approach 1 - New Module

- Possible solution: implement the enhancement as a new module entirely
- Numerous dependencies
    - Connection Manager
    - Transactions
    - Wallet
    - Peer Discovery
    - RPC
    - App

Approach 1 – Advantages

- Security
    - Isolating Lightning Network functionality minimizes creation of new vulnerabilities
- Maintainability
    - Module is easier to maintain when completely separate
- Integration
    - Module integration could be more simplified when new module is completely separate

Approach 1 - Disadvantages

- Scalability
    - A new module introduces more complexity
- Availability
    - Module availability is heavily reliant on its dependencies

Approach 2 - Submodule

- Could implement new functionality as submodule of Connection Manager
- New submodule would implement smart contract and update ledger
- Opening/closing the channel → HTTP Server submodule
- Similar dependencies to Approach 1
  - Transactions
  - Wallet
  - Peer Discovery
  - App

Approach 2 – Advantages

- Availability
  - Reduced dependencies minimizes points of failure
- Accuracy
  - Close coupling of Connection Manager and Lightning Network could lead to quick and accurate transaction processing

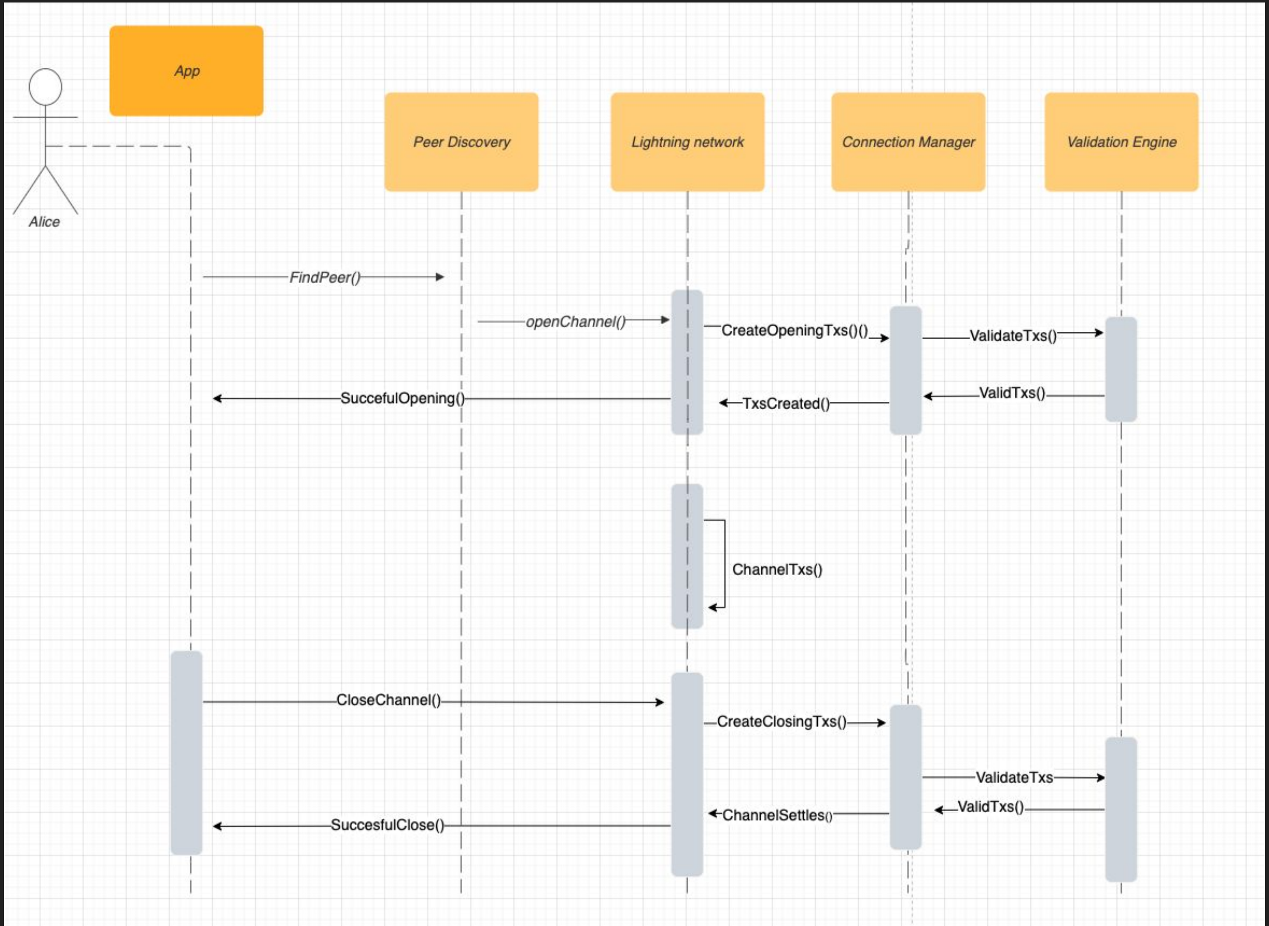
Approach 2 – Disadvantages

- Integration
  - Challenging to modify connection manager at this stage of development
- Scalability
  - While Lightning Network functionality as a whole improves scalability, making the new feature a submodule of the Connection Manager would increase coupling between the two components and make them harder to scale

Stakeholders

- Developers
- Users
- Merchants
- Miners

A user named Alice creating a channel in the lightning network with another user, making transactions inside the channel with that user and then finally closing the channel.



Use Case

IV. Reflection

Process

- Several factors were considered when developing proposal
  - Source code consulted
  - Functional and non-functional requirements of Bitcoin Core considered
- Work was delegated according to similar subteams to A2
  - Three main divisions: presentation, SAAM analysis, overview/rest of report

Lessons Learned

- Improved understanding of difficulty of new feature implementation
  - Lots of dependencies and different implementation styles to consider
- Emphasized importance of clarity of requirements of new features
  - Critical to derive most effective approach for implementation

THANK YOU FOR LISTENING