Bi-directional payment channels over the Bitcoin cryptocurrency

Initial report

David Lozano Jarque

Computer Science Engineering degree

2016 - 2017 course

ETSE UAB

Introduction

Cryptocurrencies

In a few words, a cryptocurrency is a currency that uses digital assets in combination with cryptography to secure and verify its transactions, and regulate the generation of units of currency independently from a central bank.

Unlike physical currencies, cryptocurrencies do not rely on official authorities or institutions to get its value worth, instead, the users of the cryptocurrency give value to the currency itself by being a part of it.

How a cryptocurrency works

Functionalities

To create a currency, we need to be able to handle accounts, create and manage the creation of units of currency, know the accounts' balances and let users create transactions between their accounts. This is done with a software that will run every user of the cryptocurrency and will communicate with other users to create a decentralized network where the cryptocurrency operates.

Currency units generation

To generate units of currency, the software is coded with an algorithm that allows users to create units of currency under certain conditions that require consensus between all users to validate the currency unit value. Commonly the conditions for the currency unit generation are some work required for the cryptocurrency to be safe. Therefore there's an incentive for the nodes to work for the network safety. This action of currency units generation by nodes is often called *mining*.

Identities

In most current cryptocurrencies, users are identified with their public key-pair (or their hash in Bitcoin) so everyone can create an account without being part of the network simply by generating a public / private key pair.

Transactions

Once the account is created, transactions are created locally with the software and afterwards broadcasted to the network so other users know about the transaction. When the transaction is added to the blockchain, the transaction is valid and therefore accepted as valid by the nodes on the network. To select which user can add transactions to the blockchain, consensus algorithms are used.

Consensus

As said, some actions require consensus like adding transactions to the blockchain. Consensus algorithms are needed to take decisions like this one and also to ensure the generation of currency and transactions is valid and all nodes agree on the status of the cryptocurrency.

Components of a cryptocurrency

Technically, a cryptocurrency as is known nowadays involves the following components:

- **P2P network of users**
  To use the cryptocurrency. Users in the network are commonly called nodes
- **Communication protocol**
  End-to-end P2P protocol to communicate nodes between them
- **Cryptographic algorithms**
  And specifically, hashing functions to create the blockchain and digital signatures to verify the transactions validity
- **Blockchain: a distributed ledger**
  Public ledger of transactions shared among all cryptocurrency users secured using a blockchain. All valid transactions and currency unit generation must appear in the blockchain so nodes accept them as valid. Each node of the network has a copy of it stored.
- **Consensus decision-making**
  The main problem to have consensus about is that addition of new transactions to the ledger (usually grouped in a structure called block of transactions) needs consensus about who can create and append new blocks to the ledger that are valid and accepted by most nodes, so the node retrieves some kind of incentive. Techniques like *proof-of-work* or *proof-of-stake* are used to do this. The consensus of the status of the cryptocurrency is then automatically done as the blockchain every node has stored contains it and will be the same for all nodes.

The Bitcoin cryptocurrency

Bitcoin is the most known cryptocurrency and the first decentralized cryptocurrency since its appearance in 2009, when an anonymous programmer whose nickname was *Satoshi Nakamoto* published a white paper in a cryptography mailing list and later on, the implementation of the cryptocurrency.

Bitcoin components

In the case of the Bitcoin cryptocurrency, the implementation of their components are:

- **P2P network of users**
  The bitcoin network operates over the Internet using several networks to connect their users. There are two main networks currently:
  - *mainNet*: what we could call the production network, where bitcoin officially operates. Uses the port 8333
  - *testNet*: a network for testing purposes. All new features are tested here before being implemented in the *mainNet*. Uses the port 18333
    The messages are sent between peers through TCP connections depending on the network used.
- **Communication protocol**
  Bitcoin uses low-level binary messages to communicate between peers
- **Crytpographic algorithms**
  SHA-256 and RIPEMD-160 are used as hashing functions around all the implementation, commonly used together `(RIPE-MD(SHA-256(value)))`.
  ECDSA is used for signatures.
- **Blockchain**
  Bitcoin blockchain consists of a linked list with hash pointers of blocks that contain multiple transactions in a structure called *merkle tree* that is basically a hash tree. All valid transactions must appear in the blockchain. Blocks are also used to generate units of currency.
- **Consensus decision-making**
  In Bitcoin, *proof-of-work* is used as the technique to select the node that will emit a block containing a group of transactions. Nodes receive transactions that other nodes or the node itself has created and they are broadcasted to the network so eventually all nodes know about the transaction. Nodes that create blocks to include transactions in the blockchain will take that transaction and if it's valid, include it in a block so it will appear in the blockchain.
  To select the node that can create blocks, *proof-of-work* is a technique that riddles the nodes that want to add blocks to the blockchain to perform hashes of the block until the block hash is under certain value. The first node that has

a valid block to continue the blockchain and completes the riddle succesfully will broadcast it and the rest of nodes will accept them as valid and start mining the following block of transactions.
The value is dynamically changed so the difficulty of the riddle is modified and blocks get emitted approximately after 10 minutes.

Problems of the cryptocurrency

Speed

The problem as stated is that blocks of transactions are emitted every 10 minutes approximately, therefore, knowing the maximum size of the block and normal size of transactions, we obtain that the throughput of the cryptocurrency is one transaction per second. Traditional credit card companies process more than 10.000 per second.

Size

Also, all transactions are saved in the blockchain to be valid so even if we could process more transactions, nodes would have to buy more storage to store the blockchain as its size would increase 10TB per year, rather than the current rate of 20GB per year.

Payment channels

A solution for the stated problems is to create payment channels between nodes that don't operate on the blockchain, although their transactions could end up in the blockchain and appear as valid. This way, we can achieve higher throughputs as the transactions are sent between payment channel peers immediately. To implement payment channels, game theory is applied so new transactions that reflect the status of the payment channel make older transactions unprofitable so they are dropped and just the last transaction is kept, solving the size problem too.

To implement payment channels, Bitcoin scripting used in transactions to send and receive funds are used to create *smart contracts*.

Goals

The goal of the project is to implement a real a bi-directional payment channel using the idea stated article *A Fast and Scalable Payment Network with Bitcoin Duplex Micropayment Channels* that can be found in references section. The payment channel must work and be able to ensure the good behavior of all the parties in the channel with the use of *smart contracts*

Methodology

To perform the project, a research about the Bitcoin cryptocurrency low-level implementation to see what *smart contracts* has to be made to know the limits of the scripting language and therefore check if the implementation of the payment channel can be performed.

Later, we'll create the transactions to operate the channel. To do so, we'll create a software that generates real transactions that a payment channel would use.

As an optional step, automate the software so a final user without any knowledge about payment channels rather than the basic concept and definition can use it to create a payment channel with another peer running the software.

Software development methodology

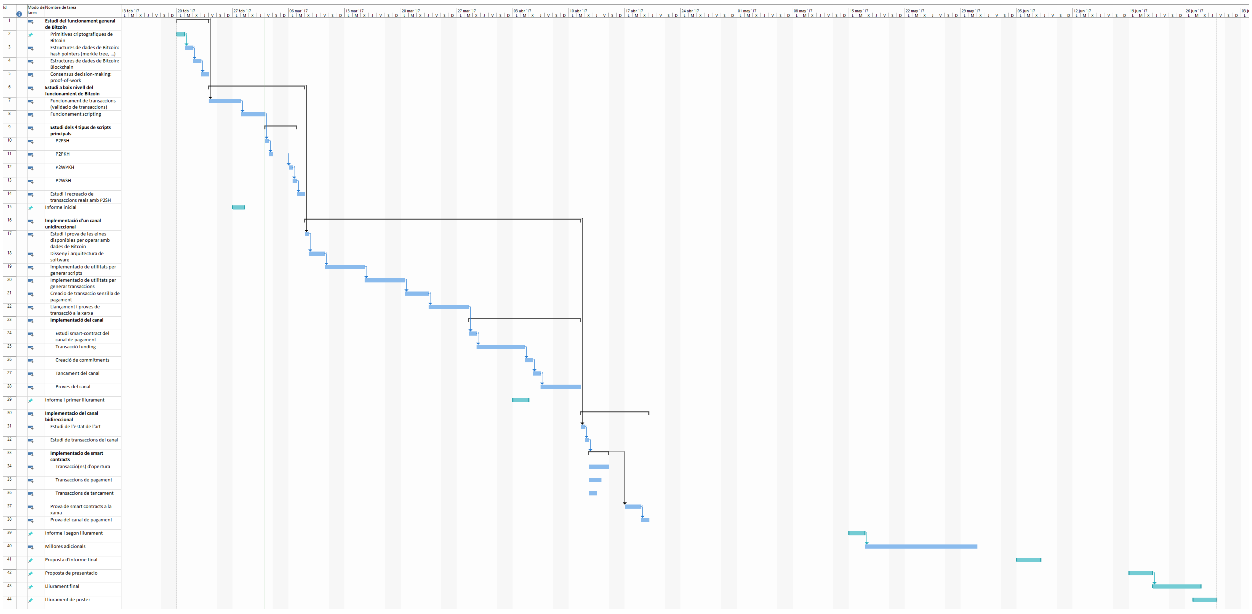
To develop the software we'll use *Git* control versioning using a private repository in *GitHub*:

https://github.com/ccebrecos/btc-payment-channels

The software development methodology will be an agile methodology with cycles of design, implementation and test of one or two weeks.

Planning

To plan the project, we have used *Microsoft Project* to list all the tasks we'll perform to achieve the project goal and finally create a Gantt diagram to see the project estimated schedule.



Id		Modo de tarea	Nombre de tarea
1			**Estudi del funcionament general de Bitcoin**
2			Primitives criptografiques de Bitcoin
3			Estructures de dades de Bitcoin: hash pointers (merkle tree, ...)
4			Estructures de dades de Bitcoin: Blockchain
5			Consensus decision-making: proof-of-work
6			**Estudi a baix nivell del funcionamient de Bitcoin**
7			Funcionament de transaccions (validacio de transaccions)
8			Funcionament scripting
9			**Estudi dels 4 tipus de scripts principals**
10			P2PSH
11			P2PKH
12			P2WPKH
13			P2WSH
14			Estudi i recreacio de transaccions reals amb P2SH
15			Informe inicial
16			**Implementació d'un canal unidireccional**
17			Estudi i prova de les eines disponibles per operar amb dades de Bitcoin
18			Disseny i arquitectura de software
19			Implementacio de utilitats per generar scripts
20			Implementacio de utilitats per generar transaccions
21			Creacio de transaccio senzilla de pagament
22			Llançament i proves de transacció a la xarxa
23			**Implementació del canal**
24			Estudi smart-contract del canal de pagament
25			Transacció funding
26			Creació de commitments
27			Tancament del canal
28			Proves del canal
29			Informe i primer lliurament
30			**Implementacio del canal bidireccional**
31			Estudi de l'estat de l'art
32			Estudi de transaccions del canal
33			**Implementacio de smart contracts**
34			Transacció(ns) d'opertura
35			Transaccions de pagament
36			Transaccions de tancament
37			Prova de smart contracts a la xarxa
38			Prova del canal de pagament
39			Informe i segon lliurament
40			Millores adicionals
41			Proposta d'informe final
42			Proposta de presentacio
43			Lliurament final
44			Lliurament de poster

Visit http://bit.ly/2mqxb2V for a high-resolution version

Agile development

The group of tasks number 30 that consists of the software development defines the first cycle of development, that will be repeated until the project deadline while adding new features in each cycle until the goal is completed.

If the goal is completed before the deadline, next development cycles will add features to the software such as the listed in the *Goals* previous section.

References

- [1] "cryptocurrency - definition of cryptocurrency in English | Oxford Dictionaries", Oxford Dictionaries | English. [Online]. Available: https://en.oxforddictionaries.com/definition/cryptocurrency. [Accessed: 28- Feb- 2017].
- [2] "Bitcoin P2P e-cash paper", Metzdowd.com, 2009. [Online]. Available: http://www.metzdowd.com/pipermail/cryptography/2008-October/014810.html. [Accessed: 28- Feb- 2017].
- [3] "Network - Bitcoin Wiki", En.bitcoin.it, 2017. [Online]. Available: https://en.bitcoin.it/wiki/Network. [Accessed: 02- Mar- 2017].
- [4] "bitcoin/bitcoin git repository", GitHub, 2017. [Online]. Available: https://github.com/bitcoin/bitcoin. [Accessed: 27- Feb- 2017].
- [5] "Protocol documentation - Bitcoin Wiki", En.bitcoin.it, 2017. [Online]. Available: https://en.bitcoin.it/wiki/Protocol_documentation. [Accessed: 04-

Mar- 2017].

- [6] C. Decker and R. Wattenhofer, A Fast and Scalable Payment Network with Bitcoin Duplex Micropayment Channels, Distributed Computing Group, ETH Zurich, 2017.