Visualization of Blockchain Consensus Degradation

Luca Ambrosini*†, Matija Piškorec†‡, Claudio J. Tessone†§
* Scuola Universitaria Professionale della Svizzera Italiana, Lugano, Switzerland
† Blockchain and Distributed Ledger Technologies Group, University of Zurich, Zurich, Switzerland
‡ Ruđer Bošković Institute, Zagreb, Croatia
§ UZH Blockchain Center, University of Zurich, Zurich, Switzerland

2022 IEEE International Conference on Blockchain and Cryptocurrency (ICBC) | 978-1-6654-9538-7/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICBC54727.2022.9805498

***Abstract*—In this paper we present a prototype system for easy testing and visualization of the Bitcoin consensus protocol. The system consists of a modified version of the Bitcoin Core client which is intended to run on Raspberry PI machines, modified so that the latency in the block production and propagation can be manually introduced, which simulates network communication issues in the real Bitcoin network. Built-in LCD displays on each machine visualize current state of their local blockchains which allows easy visual observation of the consensus in the network. As machines are running fully capable Bitcoin Core clients, as well as mining clients for production of new blocks in the Proof of Work (PoW) consensus protocol, it is possible to investigate different configuration scenarios and when these lead to the break of the consensus.**

***Index Terms*—blockchain, Proof of Work, consensus mechanism, Bitcoin, cryptocurrency, Raspberry PI**

I. Introduction

Blockchain protocols use consensus algorithms to determine which blocks compose the chain so that, amongst all the possible local blockchains, only one is considered valid by the majority of participants. Bitcoin [1] - the first practical implementation of a blockchain protocol, uses Proof of Work (PoW) consensus mechanism where blockchain with the highest amount of computational work is considered to be the valid chain. Bitcoin nodes propagate information on new blocks in a peer-to-peer fashion - each node is connected to a number of other nodes in a bidirectional way from whom it receives new blocks, validates them, and, if valid, propagates to all other peers. With the current Bitcoin parameter setup (blocks of 1 MB produced every 10 minutes) the hardware required to be able to receive, validate and propagate the blocks are low enough that even a machine with low computational resources and limited connection can stay on top of the blockchain. Changes of these parameters influence the ability of a node to keep up - a larger block size or faster block creation time requires faster connection and better hardware (CPU, RAM, DISK). Changes in Bitcoin parameters could then deteriorate consensus which manifests by the co-existence of multiple blockchains in the network.

In the following paper, we introduce a demo setup in which we can show the effect of changing the parameters on the consensus in the network. To do so, we artificially introduce *latency* in the process of verifying a node which represents a time required to receive, process and transmit a block. We implement our system using several Raspberry Pi's (RPi) [3] - low cost machines with enough computational resources to run a Bitcoin Core client as well as a Bitcoin miner, allowing them to operate as a fully capable Bitcoin network nodes. Raspberry PI's are equipped with LCD displays which visualize the state of the local blockchain for each node, which allows to easily visualize the state of consensus in the network as a whole. This makes our system ideal for education and demonstration purposes, allowing for easy visual investigation of how changing different parameters on a full Bitcoin node client can influence the consensus on the Bitcoin network. The system consists of the open source code and documentation on how to install and configure it on any number of RPi's. Basic knowledge of the RPi assembly and configuration is needed by the end user. However, users should have sufficient knowledge of the Bitcoin Core and how to configure it in order to replicate various use cases we outline in this paper, as well as to do their own experiments.

LA, MP, CJT acknowledge financial support from UZH Lehrkredit "Operable Platforms for Experimenting with Cryptocurrency"

978-1-6654-9538-7/22/$31.00 ©2022 IEEE

II. Description of the system

In our experiment we are using a modified version [2] of the Bitcoin Core 0.20 client, to which we have added the option to specify the *latency* parameter - an amount of time that the node should wait before starting to process the block. Latency can also be introduced after a specific block height to simulate degradation of the node capabilities after a specific event. In practice, the block propagation now works as follows: (1) a node receives a block from another node, (2) the node sleeps for the specified latency, (3) the node then validates the block, and, if valid, (4) propagates the block. It's important to notice that the latency is there even when a block is invalid, as in a real case scenario. Additional modifications to Bitcoin Core have been performed to enable data visualization, but they do not impact the performance of the node or its existing functionalities.

In our demo case, each node is running on a RPi with 4-cores and 4 Gb RAM. Bitcoin Core is locally compiled on the RPi and only the required dependencies are installed. On each RPi we have mounted a 3.5" LCD screen that shows the local visualization of the blockchain - two most recent blocks with their respective block heights, block hashes and distinct colors which is unique for each block. Each RPi

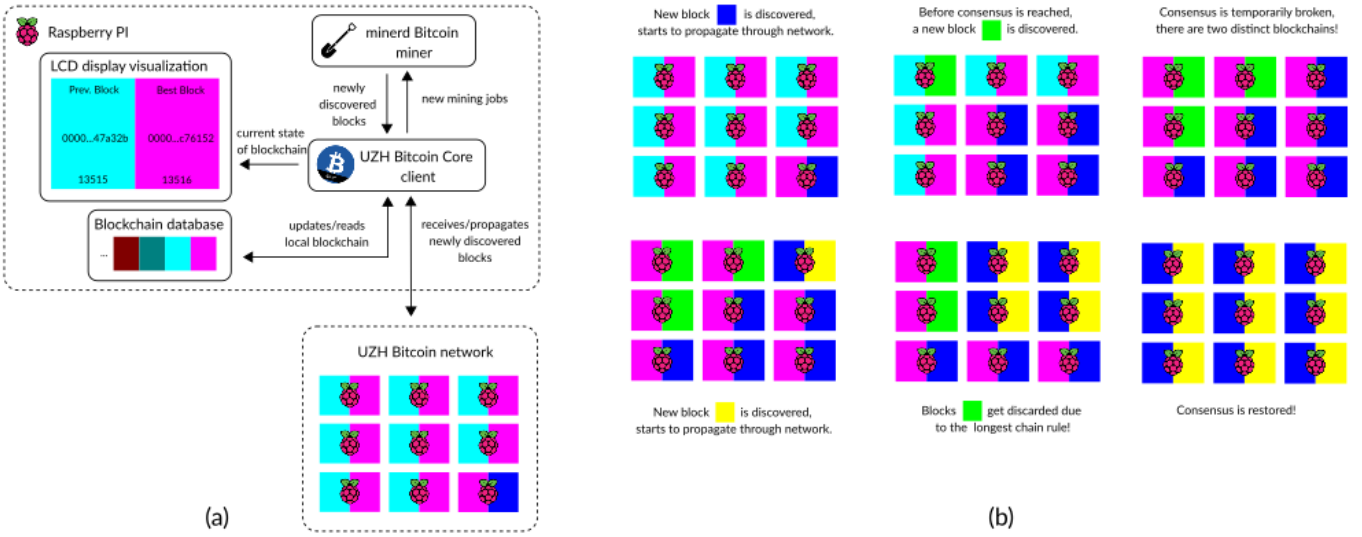

Fig. 1. A schematic of the system architecture (a) and an example of the consensus visualization for 9 Raspberry PI's prototypes (b). The example in (b) demonstrates how color-coded visualization of the last two blocks in each local blockchain makes it easy to visualize consensus degradation in practice. Broken consensus happens when multiple miners construct different valid blocks at approximately same time, and start propagating them through the network. Nodes in different parts of network might then temporarily end up with inconsistent version of the blockchain - containing different blocks at the tip of their blockchains, until consensus is restored.

can also produce blocks through `minerd` (a PoW mining node) [5] installed locally. Fig. 1 shows the architecture of the system and an example visualization of how consensus in the network is broken and again established after a while. The LCD visualization interface shows the last and the second-last block details as block height and block hash. Each block also has an unique color which enables us to easily see differences in local blockchains between different nodes. When a new block is received it is shown as the last block and a timer starts at the bottom of the screen. When the timer goes to 0 the block will be propagated to all the peers. Each RPi is also running a full Bitcoin explorer [4], enabling us to see every detail of the blockchain through a web interface.

In the showcase configuration, all the RPi's are in the same local network so that we can assume that there are no bottlenecks at the network level. They are all configured to use a DHCP and receive their IP address at boot from the local DHCP router, which is configured to give static IP addresses to all the RPi's based on their MAC address. In this way we know the IP of each RPi in advance and we can use it as the unique identifier of the nodes during network reconfiguration. Nodes then download its specific `bitcoind` configuration from a remote server. This enables us to remotely define the latency to be introduced during block propagation. Each node then downloads the list of peers to which it should be connected. We have turned off peer discovery in `bitcoind` so that our nodes will only be connected to specified peers. Last, we can specify if the node is a block producer or not. In this way the topology is defined on the remote server, enabling us to run multiple experiments without having to change the physical topology.

III. Use Cases

An example use case is detailed in example (b) of Fig. 1. By choosing a specific network topology we can aid in visual understanding of block propagation, for example by choosing a grid topology where nodes are connected only to their immediate neighbors in a grid. LCD displays on RPi's show two most recent blocks in each node's local blockchain, color-coded so that corresponding blocks can be easily visually recognized across different nodes. In this way, when a new valid block appears and starts propagating through the network it will be easily recognized on LCD displays of all nodes that incorporated it into their local blockchains. We can also change other settings - positioning miners, changing the relative CPU power devoted to mining and setting latency, in order to showcase scenarios in which consensus degrades and we can easily see it by the colors of the LCD displays on the RPi's. This prototype allows us to present in a practical manner the consensus in PoW blockchains in a classroom or in outreach activities.

References

[1] Satoshi Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System", [Online], Available: https://bitcoin.org/en/bitcoin-paper. [Accessed 24th January 2022]

[2] UZH Bitcoin Github repository, [Online], Available: https://gitlab.uzh.ch/luca.ambrosini/uzhbitcoin. [Accessed 24th January 2022]

[3] Raspberry PI 4 technical specifications, [Online], Available: https://www.raspberrypi.com/products/raspberry-pi-4-model-b/specifications. [Accessed 28th January 2022]

[4] Bitcoin Explorer, [Online], Available: https://github.com/janoside/btc-rpc-explorer. [Accessed 31th January 2022]

[5] minerd - CPU miner for Bitcoin and Litecoin, [Online], Available: https://www.gsp.com/cgi-bin/man.cgi?topic=MINERD. [Accessed 31th January 2022]