Public Ledger for Auctions

Fernando Barros
up201910223

Rogério Rocha
up201805123

***Abstract*—In this document, we examine our design decisions and certain implementation aspects of a public blockchain for auction transactions, emphasizing efficiency, trust, and security. The system intends to create a decentralized ecosystem through a secure ledger that facilitates Proof-of-Work (PoW) and Delegated Proof-of-Stake (DPoS), a strong peer-to-peer (P2P) framework using Kademlia for efficient data transfer and trust strategies, along with an auction mechanism connected to the blockchain.**

***Index Terms*—Blockchain, Kademlia, Peer-to-peer, Proof-of-work, Delegated proof-of-stake, System architecture, DHT**

I. Introduction

Blockchain technology has proven to be a groundbreaking advancement in the field of digital transactions and decentralized computing. One of its most intriguing applications is the creation of a public ledger, a clear and secure system for documenting transactions among parties. This ledger guarantees integrity, traceability, and non-repudiation of records, all independently of a central authority. For this project, the public ledger utilizes blockchain principles to securely record auction transactions in a distributed and verifiable way. A suitable protocol is necessary to facilitate communication and coordination among the participating nodes. To achieve this, the Kademlia Distributed Hash Table (DHT) was chosen due to its strong peer-to-peer (P2P) features, effective search systems, and decentralized structure. Kademlia allows nodes to join and depart from the network dynamically while maintaining dependable discovery and storage of key-value pairs, which represent blockchain components and ledger records in this context. This report provides an extensive summary of the system structure, the development procedure, and the difficulties faced during the implementation phase. The following sections examine the essential elements of the project, such as the distributed ledger framework, the design of the P2P overlay network, and the incorporation of a secure and verifiable auction system within this infrastructure.

II. Architecture

At the foundation of our system is a tailored version of the Kademlia protocol developed in Java, utilizing Netty over UDP for managing peer-to-peer communication among nodes. The auction module, established on top of the networking layer, securely oversees the creation of auctions and the submission of bids using RSA-based digital signatures. It interacts with the Kademlia layer to find peers and share auction-related information without depending on internal routing mechanisms. In addition, the blockchain layer preserves an unchangeable record of every auction and bid through a Proof-of-Work consensus method. Every transaction undergoes cryptographic verification prior to its inclusion in a newly mined block. The system functions through a command-line interface that connects with all layers, enabling users to engage with the network, take part in auctions, and initiate mining. All nodes operate locally on the same machine while being tested, distinguished by individual ports.

Fig. 1. System architecture: each auction request passes through the Blockchain layer for validation and persistence, and uses the Kademlia layer for peer-to-peer communication.

III. Distributed Ledger

A. Proof-of-Work

Our system utilizes Proof-of-Work to guarantee the security and permanence of the blockchain that records every auction and bid. Nodes autonomously strive to mine blocks by tackling a resource-heavy hash puzzle, which involves locating a nonce that generates a hash beneath a specified target. This system stops interference with the ledger and guarantees that only authentic, confirmed transactions are added to the chain. Even though this implementation does not distribute mining rewards, it still ensures a decentralized and verifiable record of auction events. Nonetheless, this method presents scalability issues, since mining demands considerable computational power and time, particularly with increasing network activity.

IV. Peer-to-peer (P2P)

A. Kademlia

Kademlia is a peer-to-peer distributed hash table (DHT) that efficiently locates nodes (peers) responsible for storing data.

Node: A node consists of: (ID, IP, Port)

- Nodes are identified by 160-bit UIDs based on SHA-1. Kademlia uses the XOR metric to calculate the distance between two node IDs.
- Routing table (k-bucket): Contains information about other nodes in the network.
- Storage: Store (key,value) pairs relevant to blockchain transactions.

The main functionalities of the Kademlia protocol include the following:

- Join Network: For a new node to join the network, it must connect with a bootstrap node to obtain information about other nodes. The node iteratively finds and updates its routing table with nodes closer to its own ID, ensuring a well-connected network.
- PING: Check if a node is available.
- FIND NODE: Given a target node ID, it finds and returns the closest nodes to this target ID.
- FIND VALUE: If a node hasn't received a store for the requested key, it responds just like it would to a FIND NODE . If the node has received a store for the key, it responds with the value.
- STORE: key,value pairs are stored on the node that has the ID closest to that key.
- NOTIFY and LATEST BLOCK: To ensure that all nodes in the network are synchronized with the most recent block added to the blockchain, we created two new options to broadcast (NOTIFY) and request (LATEST BLOCK) the latest block hash. LATEST BLOCK is useful during the join network process to synchronize a new node with the latest block hash in the network. NOTIFY informs nodes about a new blockhash and determines whether the new block hash differs from the current latestBlockHash, which is an attribute in the Kademlia class that stores the hash of the latest block in the blockchain. If so, it updates latestBlockHash and notifies all nodes in the routing table, who then repeat the process.

To implement these functionalities, we used Netty for UDP communication.

B. Security Considerations in Peer Discovery

To enhance the resilience of our decentralized auction system, we utilize a distributed hash table (DHT) based on the Kademlia protocol for peer discovery and message propagation. However, as with any open peer-to-peer network, this architecture is potentially vulnerable to attacks such as Sybil and eclipse attacks.

Although some variations like S/Kademlia propose security extensions to the original protocol, we opted for a simplified model due to scope constraints. In particular, we acknowledge that eclipse attacks can be partially mitigated by reducing the ability of nodes to arbitrarily choose their identifiers. This is often achieved by binding node IDs to verifiable properties, such as public keys, and deriving the ID through a hash function. While this approach increases the cost of launching targeted attacks, it does not offer complete protection.

Sybil attacks, where a single adversary controls many fake identities, are particularly challenging to prevent without a trusted authority. Prior research suggests that the use of cryptographic signatures and identity validation mechanisms can delay such attacks and reduce their impact. However, these countermeasures were not implemented in our prototype, as our focus was on demonstrating the core functionality of the auction system and blockchain integration.

V. Auction System

A. Transaction Structure and Verification

Each transaction encapsulates critical information including the sender's and receiver's public keys, the amount being transferred, and a cryptographic signature generated using the sender's private key. This signature acts as a proof of authenticity and integrity, ensuring that the transaction has not been altered after its creation.

The signature is generated over a deterministic combination of the sender's key, receiver's key, and the transaction amount. To validate the transaction, any node in the network can verify the signature using the sender's public key. Only transactions with valid signatures are accepted and can be included in a new block.

Transactions are automatically generated using the sender's wallet instance, simplifying the submission process. All transaction data is securely serialized for efficient storage and transmission, including the reconstruction of public keys during deserialization.

This structure ensures that all transactions recorded on the blockchain are verifiable, immutable, and clearly associated with the entities involved in the value transfer.

B. Publisher System

To support the auction system, a Publisher/Subscriber system eases communication among the entities involved, including auctioneers, bidders, and subscribers.

The system allows users to create auctions, place bids, and be notified about auction updates if they are subscribers to a specific auction. Unfortunately we were not able to implement a Pub/Sub system with Google Cloud, and as such, our implementation leverages a custom peer-to-peer communication mechanism using Netty for auction-related messages, but it has a lot of limitations. Due to the direct connections between clients, it lacks the scalability and fault tolerance of a more robust Pub/Sub system like Google Cloud Pub/Sub.

We introduced two custom Options on Kademlia tailored to the auction system's needs:

- NEW AUCTION: Broadcast the auction ID of a new auction to all nodes in the network.
- AUCTION UPDATE: Notify nodes about updates to an auction. It propagates updates about bids and other changes in the auction to all relevant nodes, particularly subscribers and the node storing the auction.

VI. Conclusion

In conclusion, this report outlines the creation and execution of a public blockchain-oriented system designed for secure and decentralized auction transactions. The design incorporates fundamental blockchain concepts like

Proof-of-Work (PoW) to maintain transactional integrity, while utilizing Kademlia for robust and effective peer-to-peer communication and data distribution. Despite facing challenges with the Pub/Sub model, a different internal communication method was effectively established using Netty, allowing for dependable and nearly instantaneous message distribution throughout the network. In the future, potential improvements might aim at enhancing the scalability and reliability of the communication layer, considering the use of Delegated Proof-of-Stake (DPoS) as a different consensus method, creating a scalable Pub/Sub framework—possibly through Google Cloud Pub/Sub—and instituting sophisticated strategies to counteract Sybil and Eclipse attacks, further strengthening the system's security and trust guarantees.

REFERENCES

[1] S. Maymounkov and D. Mazieres, "Kademlia: A Peer-to-Peer Information System Based on the XOR Metric," In *Proceedings of IPTPS*, 2002. [Online]

[2] B. Ford, J. Strauss, C. Pietzuch, M. Piatek, A. Krieger, M. F. Kaashoek, and R. Morris, "Persistent Personal Names for Globally Connected Mobile Devices," In *7th USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, 2006. [Online]

[3] Riccardo Pecori, S-Kademlia: A trust and reputation method to mitigate a Sybil attack in Kademlia

[4] Satoshi Nakamoto, Bitcoin: A Peer-to-Peer Electronic Cash System

[5] David Hausheer1, Burkhard Stiller, PeerMart: The Technology for a Distributed Auction-based Market for Peer-to-Peer Services

[6] Ethan Heilman and Alison Kendler, Eclipse Attacks on Bitcoin's Peer-to-Peer Network

[7] Francis N. Nwebonyi1 and Rolando Martins1 and Manuel E. Correia, Reputation based approach for improved fairness and robustness in P2P protocols

[8] Jiangshan Yu, David Kozhaya, Jeremie Decouchant and Paulo Esteves-Verissimo, RepuCoin: Your Reputation Is Your Power