Offline Payments with Reputation-Weighted Loan Networks

Nektarios Evangelou

Offline Payments with Reputation-Weighted Loan Networks

by

Nektarios Evangelou

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Thursday June 27th 2024 at 10:30 AM.

Student number:	4978145
Programme:	Computer Science
Faculty:	Faculty of Electrical Engineering, Mathematics and Computer Science
Research Group:	Data-Intensive Systems
Project duration:	November 12th, 2023 – June 27th, 2024
Thesis committee:	Dr. Johan Pouwelse, TU Delft, Thesis Advisor Dr. Jérémie Decouchant, TU Delft, Daily Supervisor Dr. Antonis Papapantoleon, TU Delft, Committee Member Rowdy Chotkan, TU Delft, Daily Co-Supervisor

Preface

Six years ago, I graduated with a Bachelor of Applied Sciences in Software Engineering from The Hague University. However, at the end of my studies, I felt this was not enough. I did not feel satisfied, so I wanted to challenge myself to the next level and reach my ambition.

Five years ago, I got accepted and started the pre-master for Computer Science at the TU Delft. I realized it would be challenging, but I wanted to try it nonetheless.

Two years ago, after a rough start, a global pandemic, and **many** failures, I finished the pre-master and started with the master.

These past few years, I have pushed myself to the limits of my capabilities and worked hard to achieve my goal. Along the way, I have met many talented, inspiring, and unforgettable people who have given me the confidence and energy to fight through challenging times.

Today, I am finishing my thesis, and it seems surreal to think about how fast these past years have gone by and which have been full of challenges and achievements.

I would like to thank my thesis advisor, Dr. Johan Pouwelse, for the opportunity to write a thesis for the Data-Intensive Systems research group. Furthermore, I would like to express my gratitude to my daily supervisor, Dr. Jérémie Decouchant, and my daily co-supervisor, Rowdy Chotkan for the collaboration and guidance during my thesis. Moreover, I would like to thank Dr. Antonis Papapantoleon for being part of my thesis committee. I wish you all the best for the future.

Finally, I would like to express my deepest gratitude to my family for their unwavering support and love for all these years. In particular:

My parents. I cannot describe how thankful I am for your trust, love, and support, especially these past few years. You always kept faith in me and supported me whenever I was at my lowest. You both did not have the possibility to pursue an education, but your hard work and perseverance made it possible for your children to do so. Apart from making myself proud of my accomplishments, I hope I have also made you proud.

My brother. The one I always, and still, look up to. It always amazed me how intelligent you are. You always took it upon yourself to provide a helping hand during my educational endeavors, whereas you were always on your own. You always made me want to equalize you. Now, at least on an educational level, I can finally say that we are equal. Thank you for your support, and hopefully I have made you proud as well.

My partner. Thank you for your love and support all these years. We are a couple known for always studying and diving into the books; however, look at what it brought us: You are already a great dietitian, entrepreneur, and will soon become an epidemiologist and, most impressively, a medical doctor. I am proud of your accomplishments and am eager to see what the future will bring.

Nektarios Evangelou
Zoetermeer, June 2024

Contents

Preface **i**

Abstract **iii**

1 Research Paper **1**

Supplementary Material **29**

2 Extended Related Work **30**

2.1 Web of Trust 30
2.2 Network-Based Loans 31
2.3 Double-Spending Problem 32
2.4 MeritRank 32
2.5 Existing Offline Payment Solutions 34
2.5.1 Trusted Execution Environments 34
2.5.2 Rollups 35
2.5.3 Payment Channels 37
2.6 Unified Payments Interface Lite 40
2.7 Google Pay 40

3 Extended System Overview & Details **42**

3.1 Stakeholder Analysis 42
3.2 Identities in Trust-Based Networks 43
3.2.1 Strong Identities 43
3.2.2 Weak Identities 43
3.3 Game Theory Analysis 44
3.3.1 Loaning Risk Based on Reputation Score 44
3.3.2 Risk Based on Attacking 45
3.4 Exhaustive Search Algorithm 46
3.4.1 Average Maximum Transaction Amount 46
3.4.2 Probability Distribution of Successful Transaction 48
3.5 Random Walk Computational Complexity 49
3.6 Countering Reputation Manipulation in Loan Agreements 50

4 Additional Experiments **52**

4.1 Analysis for Selecting A Maximum Depth 52
4.2 Maximum Throughput and Confirmation Latency 54
4.3 Cost-Effectiveness Against VISA's Interchange Fees 55

5 Conclusion **56**

Abstract

Blockchain-based payment systems typically rely on a synchronous communication network and assume a limited workload to confirm transactions within a bounded timeframe. These assumptions render such systems inadequate in scenarios where reliable network access is not assured. Offline payment systems aim to enable users to register offline transactions and guarantee execution once network access is restored. However, these methods strongly assume that specialized hardware can be trusted to remain secure and tamper-proof, preventing double-spending and invalid transactions.

In this research, we introduce `Overdraft`, a novel offline payment system that operates without relying on trusted hardware. Instead, `Overdraft` leverages a reputation-weighted loan network. In this loan network, a user, for an agreed-upon fee, can declare that they will cover up to a certain amount for another user if their balance is insufficient for an offline transaction they have agreed upon. This system allows users to decide based on their online reputations, the fees they are willing to pay for loans, and their acceptance of offline payments from other users.

The online component of `Overdraft` uses a smart contract to maintain the loan network and execute transactions on a blockchain. Users consult their local, potentially outdated, copies of `Overdraft`'s reputation-weighted loan network before engaging in a transaction, allowing them to decide whether to accept an offline payment. Once network connectivity is regained, users can submit all their offline transactions to the system and have confidence that these will be executed, possibly by relying on the loan network in case of insufficient funds.

We implemented `Overdraft` as an Ethereum Solidity smart contract and deployed it on the Sepolia testnet. Our implementation includes an algorithm for calculating dynamic trust scores based on transaction and loan histories, ensuring scalability and security. The performance evaluation demonstrates `Overdraft`'s practicality, handling 68 Transactions Per Second (TPS) with a confirmation latency of 12 seconds on the Sepolia testnet. With advanced layer-1 or layer-2 blockchain solutions, `Overdraft`'s scalability can further increase, improving throughput, decreasing latency, and reducing fees.

Additionally, we address the challenges of double-spending and Sybil attacks by employing a Sybil-tolerant reputation mechanism within the loan network. This mechanism ensures that malicious nodes cannot easily exploit the system. Our research contributes to the theoretical foundation of trust-based offline payment systems and offers practical insights into designing a system resilient against fraud and misuse.

`Overdraft` not only proposes a solution for secure offline payments but also establishes a framework for future research and development in decentralized finance, aiming to bridge the gap between conventional financial systems and the emerging digital economy.

1

Research Paper

Taming Double-Spending in Offline Payments with Reputation-Weighted Loan Networks

Nektarios Evangelou, Rowdy Chotkan, and Jérémie Decouchant

Delft University of Technology, Delft, The Netherlands
{n.evangelou,r.m.chotkan-1,j.decouchant}@tudelft.nl

Abstract. Blockchain-based payment systems assume a synchronous communication network and a limited workload to confirm a transaction in a bounded amount of time. Therefore, these systems fall short in settings where reliable network access is not guaranteed. Offline payment systems aim to allow users to register offline transactions and guarantee that they will be executed once network access is regained. Offline payment methods strongly assume that specialized hardware can be trusted to remain secure and tamper-proof, preventing double-spending and invalid transactions.

In this work, we introduce `Overdraft`, a novel offline payment system that does not rely on trusted hardware. `Overdraft` leverages a loan network weighted by online-reputation. In this loan network, a user declares, for an agreed-upon fee, that they will pay up to a certain amount in place of another user if their balance is insufficient for an offline transaction they agreed on. Based on their online reputations, users can decide for what fees they are willing to loan money to each other and whether or not they accept offline payments from each other. For its online component, `Overdraft` uses a smart contract to maintain its loan network and execute transactions on a blockchain. We implemented `Overdraft` as an Ethereum Solidity smart contract and deployed it on the Sepolia testnet. Our performance evaluation demonstrates `Overdraft`'s practicality, handling 68 Transactions Per Second (TPS) with a confirmation latency of 12 seconds on the Sepolia testnet. `Overdraft`'s scalability can increase further with advanced layer-1 or layer-2 blockchain solutions, improving throughput, decreasing latency, and reducing fees.

Keywords: Offline Payments · Blockchain · Reputation · Network-Based Loans · Smart Contract · Sybils

1 Introduction

In the modern digital age, financial transactions have become heavily reliant on internet connectivity, a reality that overlooks scenarios where such connectivity is lacking, unreliable, or too expensive [1]–[3]. This gap highlights a need for robust offline payment solutions to overcome the discrepancies in areas that traditional internet-based systems fail to provide [4]–[10]. Although practical, traditional payment methods like cash, checks, postal orders, and bank transfers are loaded by security, speed, and scalability limitations and are insufficient for the demands of a modern, digital economy where high availability is necessary [11]–[13].

The absence of dependable offline payment mechanisms hinders financial inclusivity and exposes businesses and individuals to operational risks during network congestion or failures [14]. The challenges are most noticeable in remote regions, where the technological gap is most prominent [15]. Against this setting, it is known that the issue of double-spending in offline networks can mainly be avoided by incorporating trusted components [16], [17]. We leverage the concept of loan networks, a mechanism that introduces a trust-based system allowing for secure and scalable transactions without continuous online verification [18], [19]. The core of our approach lies in a reputation-weighted loan network, enabling network entities to guarantee (loan) tokens for one another. This loan mechanism enables a self-regulating ecosystem where trust plays a vital role as an indication and validator of the trustworthiness of the nodes in the network [20].

The importance of our research is developing a framework to tolerate the risk of double-spending, a known concern in offline transactions [21], [22], through a probabilistic evaluation grounded in trust scores and mutual loans. We will construct a theoretical framework that employs the Web of Trust model with the practicalities of offline transactions. We devise an algorithm that traverses a node's loan network to find a loaning node to cover the offline transaction amount when the node cannot pay. Intuitively, loans are valid for a pre-specified duration when published on-chain, and offline users can rely on the loans they know or learn about to accept (or deny) a payment. More specifically, a payer accepts a payment if it considers that it will (resp. will not) be online on time to claim its payment and is confident that the payer's loan network will execute the payment.

Furthermore, we explore the potential of predicting double-spending incidents in offline settings, laying the groundwork for a protocol that ensures transaction integrity, authenticity, and non-repudiation without real-time internet access.

Contributions.

- We design the first offline payment framework that leverages a reputation-weighted loan network to offer probabilistic guarantees that an offline payment will eventually be executed.
- We implement this framework by assuming the availability of a decentralized reputation scheme, such as MeritRank [23]. The online part of `Overdraft` is implemented using a Solidity Ethereum smart contract that maintains the list of active loans. In contrast, its offline part is implemented in Python and merely requires standard asymmetric cryptography.
- We provide incentives for users to participate in loan networks and demonstrate that Sybils cannot profit from the system.
- We evaluate the performance, resource consumption, and costs that using `Overdraft` incurs.

This paper is organized as follows. Section 2 surveys the related work for the subject. Section 3 provides a high-level overview of `Overdraft`. Section 4 goes into the system details, discussing how we manage reputations and calculate confidence in offline payments. Section 5 outlines our incentive model for the nodes in the network. Section 6 explains how `Overdraft` achieves Sybil tolerance. Section 7 evaluates our algorithm's computational overhead and throughput on randomized graphs. Additionally, we discuss our smart contract implementation, including its latency, throughput, and the fees for executing events. Section 8 discusses future work and optimizations, such as fee reductions, privacy, and risk contagion. Finally, Section 9 concludes this paper.

2 Related Work

2.1 Web of Trust

The Web of Trust (WoT) [24] is a decentralized approach to cryptography and digital certificate management, contrasting with centralized Certificate Authorities (CAs). In a WoT, trust is built on mutual verification among users, providing a flexible and community-driven mechanism for establishing trust. A well-known implementation is in Pretty Good Privacy (PGP) [25] for email encryption, leveraging a chain of trust formed through mutual public key signatures. However, WoT's adoption was limited by its complexity, requiring manual verification of public keys and physical meetings for security, which becomes impractical at scale.

Additionally, managing trust relationships becomes challenging as the network expands. The computational aspect of a WoT involves graph-based algorithms to calculate trust, with nodes representing users and edges indicating trust connections. Addressing the intricacies of trust and distrust within these networks requires specialized or adapted algorithms, complicating the traversal and management of WoT networks.

2.2 Existing Offline Payment Solutions

Trusted Execution Environments (TEE) provide secure areas within a device's main processor to execute code for financial transactions safely. They isolate transaction execution from the main operating system, preventing double-spending in offline payment systems by ensuring secure and atomic processing, even without constant network connectivity [9], [16], [17].

DigiTally [10] is a prototype offline payment system enabling financial transactions without mobile network coverage, using feature phones to exchange short digit strings. This system broadens financial inclusion in remote areas by leveraging the widespread availability of basic mobile phones. DigiTally securely stores transaction data on the user's SIM card, functioning as a TEE, and allows offline authentication, eliminating the need for real-time connectivity. Its transaction process is similar to mobile payment services like M-Pesa [26], ensuring ease of use and adoption.

EuroToken [7] is a proposed Central Bank Digital Currency (CBDC) designed as a digital version of the Euro. It is intended to be expandable, scalable, and secure and maintains price stability, facilitating nearly direct transfers globally and offline. Its prototype leverages the IPv8 protocol, showcasing the system's potential through Android/Kotlin and Python implementations and utilizing TrustChain [27], a block-DAG ledger technology, with adaptations to effectively address double-spending within the system.

Payment Channels are mechanisms within blockchains that facilitate high-volume and low-latency transactions off the main blockchain. By establishing these channels, transactions occur off-chain, significantly reducing the load and scalability challenges of the blockchain [5], [8], [28]–[30]. As exemplified by Bitcoin's Lightning Network (LN) [8], it addresses blockchain scalability by enabling transactions off-chain, thus reducing the load on the blockchain. These channels utilize multi-signature transactions and lock funds in a shared wallet controlled by both parties without third-party mediation.

3 System Overview

3.1 Core concepts and Operational Dynamics

`Overdraft` is designed to enable robust offline payments within a decentralized network. The system architecture leverages blockchain technology, a smart contract, and a unique reputation-weighted loan mechanism to address the challenges of various network conditions and adversarial behaviors. The primary goal is to facilitate secure and reliable transactions even when parties are offline.

`Overdraft` operates within a decentralized network encompassing many devices, including those with intermittent internet connectivity. We assume that nodes within this network can communicate directly with each other online and can enter into loan agreements validated and recorded on a blockchain appended into a single smart contract. The network is designed to support dynamic connectivity, allowing nodes to seamlessly transition between online and offline states without disrupting the overall flow of transactions.

Furthermore, `Overdraft` revolves around two main components: a blockchain with a deployed smart contract, joined by loan agreements, and the nodes participating in transactions. When online, nodes interact with the blockchain to establish loan agreements via the smart contract; these agreements are essential for facilitating offline transactions, as they specify the terms under which one node can loan to another, including the amount, duration, and conditions for repayment. Once a node makes an offline transaction and does not have the funds to cover the costs, the smart contract will traverse the loan agreements made with the node to cover it and repay the corresponding nodes.

3.2 System Actors

`Overdraft` contains different types of actors: regular nodes, malicious nodes, and central authorities. Regular nodes form the network's foundation, engaging in transactions and contributing to the system's integrity by adhering to protocols. Malicious users, however, aim to exploit the system, engaging in activities like double-spending and Sybil attacks for personal gain. Central authorities provide oversight and ensure regulatory compliance, enhancing trust and facilitating centralized system aspects.

To counteract malicious activities, `Overdraft` incorporates defense mechanisms against double-spending and Sybil attacks, including transaction cryptographic signing, a reputation system, and secure smart contract protocols. The reputation system is crucial, diminishing malicious nodes' network influence and capacity to benefit from fraudulent actions.

3.3 System Architecture

Figure 1 showcases `Overdraft`'s high-level architecture, showing how the connection is formed between the blockchain, smart contract, and nodes. A loan is registered on the blockchain at the outset, as shown in block 0. The loan is still invalid in this block, as indicated by its red color. The subsequent blocks (1, 2, 3), active as indicated by their green color, represent the ledger's ongoing chain, where loan activity and other transactions are recorded over time. The smart contract is central to `Overdraft`'s operations. It automates the enforcement of loan agreements and payments, thus managing all transactions, including those that occur while nodes are offline.

`Overdraft`'s architecture supports nodes entering into loan agreements when they are online (1) and obtaining a local copy of the blockchain. After such agreements are in place, nodes can continue to execute transactions offline (2), leveraging the trust and terms in the loan agreement. The offline transactions enable a node (User A) to pay another node (User C) without real-time blockchain connectivity. These transactions are based on the validity of the loan, which remains active from the time it is created in block 0 until it is utilized or when the predetermined active period ends. In our example, this period covers from block 1 to block 3.

Once a node (User C, who received an offline payment) reconnects to the network, the offline transactions are synchronized with the blockchain (3). This reconciliation process validates the transactions against the smart contract's terms, ensuring that all parties are compensated fairly as per the agreement.

4 System Details

4.1 Maintaining reputations

MeritRank [23] provides a decentralized mechanism that realizes a Sybil-tolerant reputation system. It employs a decentralized approach where peers within a network actively observe and evaluate each other's contributions, which are recorded in a personal ledger. This process is conceptualized within a directed feedback graph, where the peers are represented as nodes, and the weighted edges are the

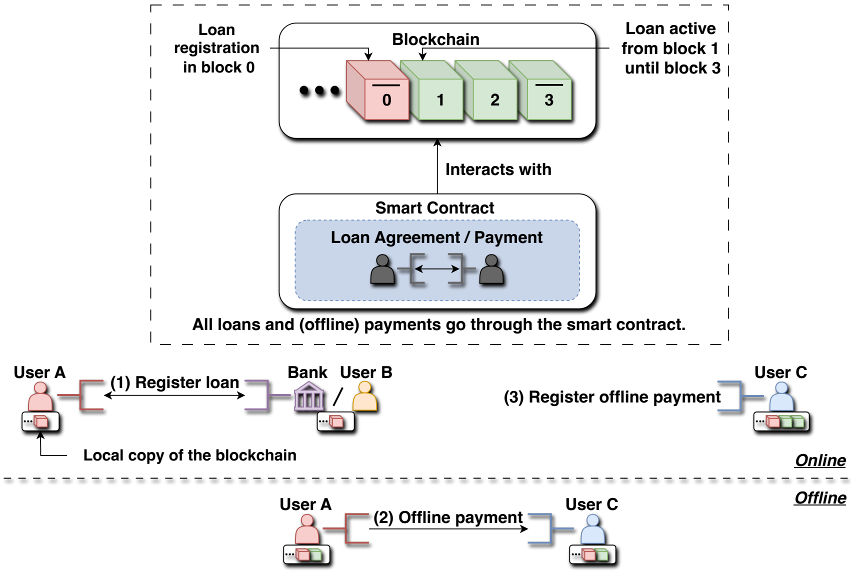

Loan registration in block 0
Blockchain
Loan active from block 1 until block 3
0
1
2
3
Interacts with
Smart Contract
Loan Agreement / Payment
All loans and (offline) payments go through the smart contract.
User A
(1) Register loan
Bank
User B
(3) Register offline payment
User C
Local copy of the blockchain
Online
Offline
User A
(2) Offline payment
User C

Fig. 1: High-level overview of `Overdraft`'s architecture displaying the core components and their interactions.

feedback accumulated over time between peers. The system assigns reputation scores to each node based on aggregated feedback, utilizing epochs to capture the dynamic nature of contributions.

`Overdraft` employs a concise and practical approach to calculating node reputation by integrating MeritRank's methodology with a decentralized ledger mechanism. This combination evaluates both the quantity and quality of node interactions, where the ledger meticulously tracks each node's successful and failed transactions. By focusing on the nature of failed transactions, our model discerns each node's risk to the network. It ensures that reputation scores are based on transaction volume and reflect nodes' actual reliability and trustworthiness, accommodating failures beyond their control and providing a transparent, secure basis for trust assessment within the network.

4.2 Interactions with the blockchain through smart contracts

In `Overdraft`, a smart contract is a foundational tool for establishing trust and facilitating transactions without immediate payment in an automated way. This contract contains multiple loan agreements binding between nodes, which is crucial for the integrity and functionality of the network. We can establish all of our agreements between two parties into a single smart contract, which helps distribute the fees for opening and maintaining loan agreements between all parties in the network. The loan agreement can be established with the following information in Table 6.

Agreement Initiation. A loan agreement is established between two parties, the loaner, and the loanee, through our smart contract on the blockchain. The agreement is created in a block with a specified opening time, allowing nodes to consider it for their confidence calculation. If the loanee accepts, the agreement is activated, triggering a token-locking mechanism for the loan amount. Rejection by the loanee nullifies the agreement and unlocks the tokens to the loaner. This agreement enables the loanee to utilize the loaned tokens, especially in offline payments when their balance is not sufficient for a payment.

The contract specifies a validity period through start and end blocks, making the agreement enforceable only within this timeframe. The loan concludes either upon repayment by the loanee or after the end block when the locked tokens and any accrued interest rate are released to the appropriate party.

Token-Locking Mechanism. An essential feature of our smart contract is the token-locking mechanism, which is crucial for the execution and enforcement of loan agreements. This mechanism ensures that tokens loaner commits are inaccessible, guaranteeing the availability of funds for the loaned transactions. For instance, if node A, holding a balance of 100 tokens, loans 50 tokens to node B, node A's effective balance remains at 100 tokens, yet only 50 are freely usable. On the contrary, node B, initially with 0 tokens, gains the ability to have the smart contract collect the 50 loaned tokens from node A for offline transactions within the network. This locking mechanism counteracts a Sybil attack, which we will discuss in Section 6.

Reconciliation with Blockchain and Agreement Closure. Upon regaining online connectivity, nodes must reconcile offline transactions with the blockchain to update it and ensure fair compensation. The smart contract reviews loan agreements to settle balances based on these transactions, addressing offline payments and compensating the receiving nodes through a random selection of loaning individuals. This random selection utilizes a hash function that combines the current block timestamp, the previous block's RANDAO beacon value, and the user's public key address. Furthermore, the extra fees required by the smart contract traversing the loan agreements must be paid by the party unable to pay.

The loan agreement is then marked inactive, triggering the smart contract to start the repayment process. Loanees must repay loaned amounts plus interest within a set period. Successful repayments adjust the parties' reputation scores according to our allocation mechanism. Figure 6 shows a visualization of our payment protocol.

4.3 Confidence in offline payments

Distribution of Transaction Amounts. Our core algorithm predicts transaction amount distributions among peers who have loaned tokens to a node. We employ a random walk simulation to navigate network paths randomly. This approach reduces the computational complexity of searching all combinations from non-paying nodes and their predecessors. It traverses each edge only once but allows for nodes to be revisited. Furthermore, it produces a distribution of the maximum transaction amounts a paying node can achieve via its loaning predecessors. The simulation halts when the random walk amount meets or exceeds the target transaction amount, avoiding unnecessary predecessor searches.

Figure 2 shows what would happen when cycles are encountered in a random walk without prohibiting going through edges multiple times. We prevent infinite loops caused by cycles by tracking visited edges. This tracking will ensure that the algorithm will terminate and enhance the accuracy of the result by avoiding repeated cycles.

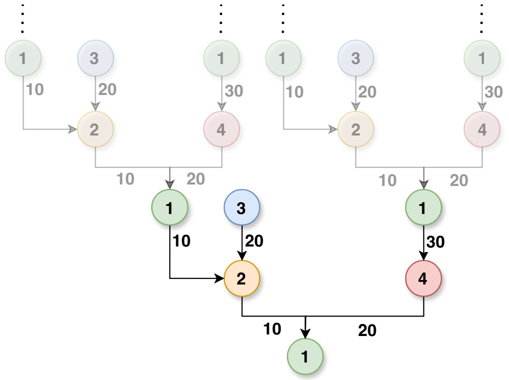


Fig. 2: An example network demonstrating how cycles lead to infinite recursion and algorithm non-termination due to repeated edge traversal.

Algorithm. Our algorithm, as outlined in Algorithm 1, begins by appending the current node's ID, determining its willingness to repay the transaction amount based on its reputation through a probabilistic model. If a node decides to pay, its loan amount is immediately returned; otherwise, the collection is set to zero, and it proceeds to evaluate the node's predecessors with two optimizations for better performance and accuracy. We incorporate a decay factor to reflect decreasing node influence with distance, inspired by trust dynamics in social and financial networks, ensuring distant nodes have less impact on the outcome [31]. Furthermore, we impose a maximum distance to prioritize closer, more reliable nodes for endorsement, effectively addressing trust dilution over distance.

The algorithm excludes visited edges to avoid cycles and iterates over a node's predecessors to avoid self-references. It accumulates loan amounts through recursion until the random walk's paths are exhausted. It also halts if the collected amount meets the transaction need or the maximum distance is reached, enhancing efficiency and network relevance.

Mathematical Notation. In the given mathematical notation, N_i represents the current node, and V_i is the loaned amount associated with the edge leading to the current node. The reputation of node N_i, denoted as $R(N_i)$, is a value within the range $[0,1] \in \mathbb{R}$. The decay parameter, D, is a fixed value also in the range $[0,1] \in \mathbb{R}$. H represents the maximum hop distance, which starts at 0 and increases with each hop to the next predecessor of the root node; this value can be fixed to any positive number. $P(N_i)$ denotes the predecessor nodes of N_i. The edge from node N_j to N_i with a loaned amount is denoted as $E(N_j, N_i)$ or simply E for ease of reading. S represents a set of visited edges to which an edge E is added after each visit. This set resets for each iteration of a random walk. The recursive formula for the amount A collected in a random walk starting from a root node N_i is defined as

$$A = \begin{cases} V_i, & \text{with probability } R(N_i) \cdot D^H \\ \sum_{N_j \in P(N_i)} A(N_j, V(E))_{E \notin S}, & \text{with probability } 1 - (R(N_i) \cdot D^H) \end{cases} \tag{1}$$

When the random walk is called on the root node N_i, the node that initializes the transaction, there is a probability $R(N_i) \cdot D^H$ that the node will be able to pay, based on its (decayed) reputation. If the node pays, the function returns the loaned amount V_i associated with the edge leading to this node, and the recursion will stop. If the node does not pay, which happens with probability $1 - (R(N_i) \cdot D^H)$, the function will recursively call itself for each predecessor node N_j of N_i. The loaned $V(E)$ associated with each outgoing edge E is passed as an argument to these recursive calls. The function then sums up the amounts returned by these recursive calls, indicating the loans of the nodes in a randomly selected path. We stop the random walk once the collected amount equals or surpasses the transaction amount.

5 Incentives

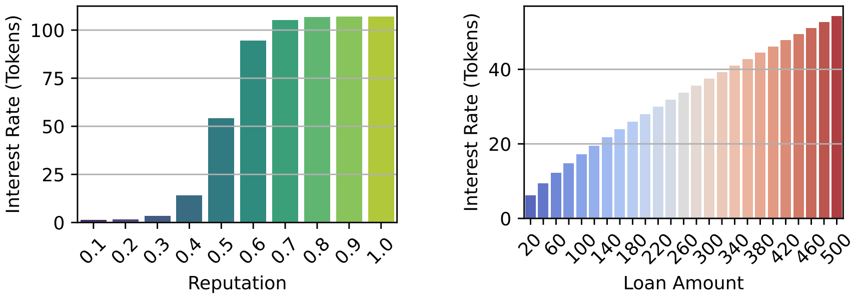
In our network, nodes can loan tokens to others, committing a portion of their resources or reputation to support another node. This loaning mechanism fosters trust and is structured similarly to a bank loan, where the person receiving the loan must repay the amount with interest. The interest rate for a loan agreement is determined by several factors, including the amount loaned, the duration the loan remains active and unused, and the loaning node's reputation. Our interest rate formula, designed to encourage participation, discourage malicious behavior, and fairly compensate for risks, can be denoted as follows:

$$I = max\left(0, ((\alpha^{\beta} \cdot \frac{1}{1 + e^{-\varsigma(R - R_0)}}) + (\frac{\gamma}{365} \cdot \delta))\right) \tag{2}$$

The interest rate (I) is determined by a complex expression that includes the loan amount (α), influencing the interest directly. The loaned percentage rate (β), acting as a power to the loan amount, adjusts the interest based on the loan size, with a current setting of 0.75 to moderate the impact of the loan's size.

Additionally, the duration (γ), set at 100 days for the following examples, the loan remains active and unused influences the accumulation of interest, which is calculated daily until the agreement's end period, based on an annual percentage rate (δ) of 5%. The lending node's reputation (R) plays a critical role, with higher reputations leading to increased interest rates due to the sigmoid function's characteristics. This is moderated by a penalizing midpoint constant (R_0) set at 0.5, which acts as a threshold for reputation, affecting the interest rate based on the lender's reputation relative to this midpoint. The sigmoid function's steepness factor (ζ), set at 20, further dictates the rate's sensitivity to changes in reputation.

The impact of reputation and the amount loaned on the interest rate is illustrated in Figure 3. The left bar chart (3a) shows how interest rates vary with reputation, highlighting the benefits of maintaining a high reputation. The α parameter was set to 500 tokens. The right bar chart (3b) demonstrates the interest rates for different loaned amounts, underlining the incentive for nodes to engage in the lending process. We set the reputation parameter R to 0.5 for a neutral reputation score.



(a) Interest rates earned for different reputations.

(b) Interest rates earned for different loaned amounts.

Fig. 3: Comparison of interest rates earned for different reputations and loaned amounts. (a) The left bar chart displays the interest rate gained with different reputation scores. (b) The right bar chart displays the interest rate gained with different loaned amounts.

6 Sybil tolerance

6.1 Reputation Scores

`Overdraft` incorporates the Sybil tolerant reputation mechanism, MeritRank, which computes reputation scores for the nodes in the network. MeritRank enhances Sybil tolerance through three essential modifications in the form of bounds. The parallel report bound addresses vulnerabilities to parallel and cycle attacks [23, Figure 4], limiting the total reputation across all involved Sybil nodes to most that of the first introduced Sybil node. The serial report bound prevents the indefinite reputation inflation from serial attacks by capping the total reputation gain from sequentially added Sybil nodes. Finally, the transitivity bound ensures reputation propagation does not exceed the minimum reputation among all nodes on a given path, thereby preventing artificial reputation amplification and reflecting genuine node trustworthiness.

Decay Strategies for Enhanced Sybil Tolerance. MeritRank implements decay strategies to enhance Sybil tolerance in networks. It uses an alpha decay parameter to limit reputation score propagation, curbing the influence of fake node chains by reducing the length of influence paths. The beta decay parameter counters structural vulnerabilities by penalizing nodes connected through bridge connections for forming separated components in the network, identified by a significant flow of random walks through a cut vertex. Finally, the gamma decay parameter combats reusing previous connection exploitation by diminishing the benefits of old attack edges, requiring continuous effort for reputation upkeep. These strategies include transitivity limitation, connectivity penalties, and epoch-based adjustments to strengthen network security.

6.2 Mitigating Sybil-Induced Loan Agreement Exploits

Mitigating Sybil attacks in loan agreements involves addressing the exploitation where Sybil nodes inflate loan totals by lending tokens to a node and then sending the exact amount of tokens to a Sybil node, repeating this process multiple times. This deception leads honest nodes to believe they have access to more tokens than are available. This attack can be seen in Figure 4a. Recognizing that these Sybil nodes typically have low reputations, they are less likely to be chosen as transaction partners, indirectly safeguarding against such schemes. The attack's root lies in the delay between initiating inflated loans and the network's recognition of these as invalid due to the original Sybil node's insufficient funds.

To counteract this, `Overdraft` restricts nodes from accepting loan tokens previously used in any agreements, coupled with a token-locking mechanism that locks the loaned amount for a set duration, hindering the Sybil nodes' ability to execute the abuse.

Despite the implementation of token-locking, the risk of double-spending persists. However, our smart contract is designed to trace through the loan tree, identifying the root transaction that led to the imbalance. Once detected, the smart contract initiates a process that goes through the loan tree to find nodes to cover the loss. It then redistributes the costs among the involved parties, ensuring a fair resolution.

6.3 Countering Reputation Manipulation in Loan Agreements

Another possible loan agreement attack is creating multiple Sybil nodes, which all loan tokens to a node for a fraction of a more considerable transaction amount. This will allow the loanee node to pick smaller nodes to pay a lower interest rate for the loaned tokens instead of one node that loans the entire amount at once. Moreover, it provides less risk as more nodes that can pay a part of the transaction amount are available instead of relying on only one node. Figure 4b shows an example of two possible scenarios. The first scenario A, which can occur, is where two correct nodes C_1 and C_2 are engaging in a loan agreement. For scenario B, we have the Sybil nodes voting for the same correct node for the same loan amount as C_2, with them having fractions of the amount and reputation. For the correct node, scenario B is the most enticing as it will have to pay a lower interest rate and have more nodes that it can rely on, meaning less risk.

However, this attack will not benefit Sybil nodes as the incentives mechanism does not incentivize lower reputations. Furthermore, the influence of a node can be seen by its reputation times its loaned amount. Suppose the loaned amount is split among multiple Sybil nodes. In that case, the risk-reward benefit will only be applicable once the cumulative influence of the Sybil nodes is equal to that of the original loaning node. Reputation systems make these attacks difficult to execute because Sybil nodes are detected and penalized with a lower reputation. This information can be turned into a Theorem 1, and its formal proof can be found in Appendix D.

Theorem 1. *In a system where a node's influence on a loan agreement is determined by the product of its reputation and its loaned amount, splitting this amount among several Sybil nodes, each with a fraction of the original node's reputation does not yield a more significant total influence, assuming a well-functioning reputation system that penalizes Sybil nodes.*

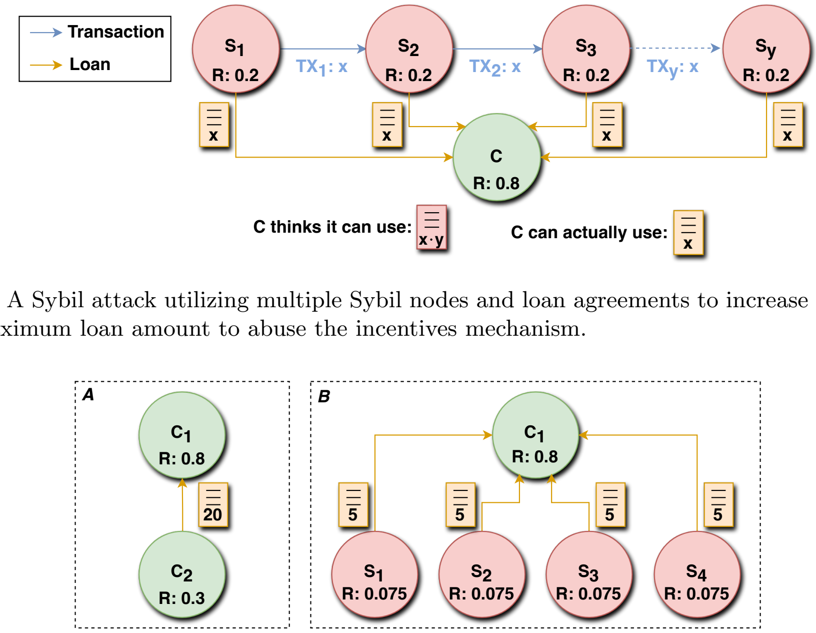
7 Performance Evaluation

7.1 Smart Contract

Contract Pseudocode We developed a smart contract using Ethereum's Solidity [32] programming language to examine and practically validate our proposed loaning system. Ethereum [33] was chosen for its widespread adoption, trivial testing tools, and extensive documentation, which collectively facilitate the creation of decentralized applications and smart contracts with custom logic and interactions. Our smart contract encapsulates the logic for handling loan agreements, handling (offline) payments, and enforcing the rules governing the interactions between nodes within the network. We provide the pseudocode of key components for this smart contract in Algorithm E.

7.2 Computational overhead of confidence algorithm with randomized graphs

We evaluate the performance of our loaning algorithm by conducting random walk simulations on randomized graphs with node counts at intervals of 10, 100, 10,000, 100,000, and 1,000,000, each connected by nine edges to mimic the



(a) A Sybil attack utilizing multiple Sybil nodes and loan agreements to increase their maximum loan amount to abuse the incentives mechanism.

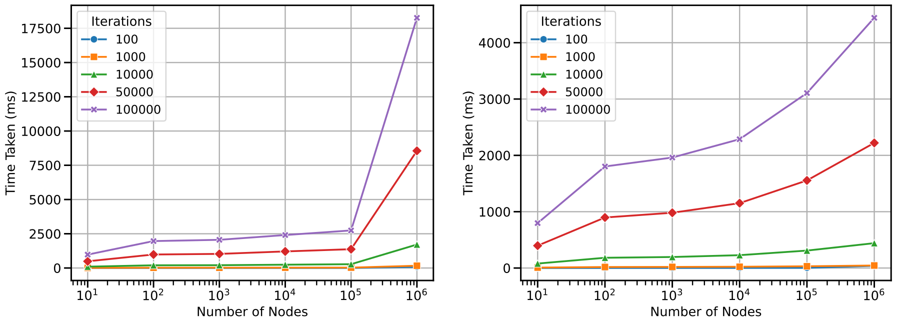
(b) A loan agreement attack utilizing multiple Sybil nodes to loan fractions of another node's loan amount. The left part (A) shows a regular loan scenario. Part (B) shows the Sybil attack with nodes splitting to fractions of a loaning node's reputation.

Fig. 4: Sybil attacks exploiting loan agreements in the network.

average degree connectivity in the LN [34]. The simulations span 100 to 100,000 intervals, testing the algorithm's accuracy across different scales.

The optimization introduces a 9-hop maximum depth, reflecting the average LN hop distance, and a 0.95 decay factor for loaning node reputation, reducing credibility with further distance from the root. The loan capacity per node is uniformly random, up to 20, increasing in increments of 10. Furthermore, the root node's reputation starts at 0.2 reputation and is transacting 100 tokens for each iteration.

We provide a performance overview of the network's different iterations and node sizes in Table 1. We evaluated our algorithm's performance with and without optimizations. In addition, we provide a graphical visualization of this evaluation, which can be found in Figure 5b. If the reputation of the root node is high, we will have less execution time for the algorithm as it does not require going through the network often to get the number of loans when unable to pay. We can see that after 50,000 iterations, the confidence interval width does not necessarily become much smaller than the 100,000 iterations. Thus, we can conclude that the accuracy of our algorithm is most optimal regarding computation time at around 50,000 iterations. We also compared the accuracy of our algorithm



(a) Without optimizations

(b) With optimizations

Fig. 5: Performance analysis of our algorithm without and with optimizations for the maximum amount loaned distribution.

using a 95% confidence interval in Table 2. We noticed a significant increase in accuracy when comparing both tables, without and with optimizations, with little increase in accuracy when surpassing 50,000 iterations.

Table 1: Combined performance evaluation of retrieving the probability distribution regarding the maximum retrievable loan amount, comparing results without optimizations (Wo. Opt.) and with optimizations (With Opt.).

Nodes	Iterations									
	100		1,000		10,000		50,000		100,000	
	Wo. Opt.	With Opt.	Wo. Opt.	With Opt.	Wo. Opt.	With Opt.	Wo. Opt.	With Opt.	Wo. Opt.	With Opt.
10	1.00	0.80	9.75	7.98	97.75	79.48	488.86	397.27	977.31	797.43
100	1.98	1.81	19.91	18.13	197.75	182.14	985.76	896.95	1965.24	1802.78
1,000	2.09	1.97	20.49	19.72	205.50	195.68	1029.86	981.18	2057.84	1961.30
10,000	2.48	2.44	24.48	22.82	240.72	228.14	1209.50	1151.75	2405.89	2286.63
100,000	2.94	3.32	28.36	31.28	277.44	309.72	1372.79	1554.86	2742.24	3106.29
1,000,000	85.40	39.89	167.04	44.84	1713.39	441.44	8554.20	2220.98	18263.57	4441.81

Table 2: Combined 95% Confidence Interval widths for our algorithm results without optimizations (Wo. Opt.) and with optimizations (With Opt.).

Nodes	100 Iter.		1,000 Iter.		10,000 Iter.		50,000 Iter.		100,000 Iter.	
	Wo. Opt.	With Opt.	Wo. Opt.	With Opt.	Wo. Opt.	With Opt.	Wo. Opt.	With Opt.	Wo. Opt.	With Opt.
10	9.47	6.99	3.46	2.26	1.08	0.75	0.49	0.33	0.34	0.23
100	16.35	13.82	5.05	4.30	1.56	1.38	0.71	0.62	0.50	0.44
1,000	11.74	11.04	4.10	3.76	1.34	1.20	0.59	0.53	0.42	0.38
10,000	18.53	13.94	5.96	4.60	1.85	1.47	0.82	0.67	0.58	0.47
100,000	22.27	24.31	7.51	6.72	2.32	2.13	1.05	0.96	0.75	0.68
1,000,000	248.74	22.39	96.17	7.80	29.42	2.45	13.20	1.10	9.27	0.78

7.3 Throughput and latency

We measured the performance of `Overdraft` in the form of Transactions Per Second (TPS) and confirmation latency (in milliseconds). TPS measures the number of transactions a system can process within a second, which can provide insight into the system's capacity and scalability. Conversely, latency measures the time elapsed from when a transaction is submitted to the network until it is confirmed, offering insights into the system's responsiveness and efficiency. A higher TPS and lower latency allow a system to handle more significant transactions, which is essential for blockchain applications competing with industry-standard digital payment applications. However, this design problem is inherently limited by blockchain technology.

As a prototype, we implemented our smart contract in a local development environment using Hardhat [35]. We then deployed this smart contract on Ethereum's Sepolia testnet and sent as many transactions as possible in one block. We took the average block confirmation time of Sepolia's testnet, which is around 12 seconds, and computed the TPS of our smart contract. Afterward, we took Ethereum's gas block limit (30,000,000) [36] and divided it by the average gas units used by the transactions multiplied by the average block confirmation time.

We compared `Overdraft`'s TPS and confirmation latency with Ethereum's in Table 3. The results show that `Overdraft`'s TPS is 4,5 times larger than Ethereum, and the confirmation latency of `Overdraft`'s is bounded by the blockchain in which it is deployed. The maximum throughput and confirmation latency could be improved by implementing and deploying our smart contract on another layer-1 (L1) or layer-2 (L2) solution, such as Algorand [37], or Polygon [38], respectively.

Table 3: Comparison of TPS and confirmation latency between Ethereum and `Overdraft`

System	**TPS**	**Confirmation latency (s)**
Ethereum [39]	15	12
`Overdraft` (Sepolia testnet)	68	12

7.4 Fees

Cost of Operations. Executing actions within the smart contract incurs fees, paid in USD, which cover the computational energy required to process the transactions on the blockchain. These fees ensure the network's security and functionality, compensating miners for their computational resources. `Overdraft` balances the cost implications to maintain an efficient, sustainable ecosystem for all participants. Our smart contract interaction costs[1] can be seen in Table 4. This table displays the smart contract operation costs when deployed on different L1 blockchains, such as Ethereum or Algorand, or an L2 solution, such as

[1] Costs shown in USD at the time of writing April 16th, 2024: 3030 USD per ETH (Ethereum), 0.17 USD per ALGO (Algorand), and 0.69 USD per MATIC (Polygon)

Polygon. We take reasonable average fee prices for each network to gain insight into the possible costs for each operation executed on the smart contract.

Cost of Bundled Operations. Bundling transactions on the smart contract would make it possible to decrease fees further. Table 5 shows the fees[1] associated with different bundled loan agreement creations. We compare Ethereum, as our smart contract was already implemented in Solidity, with Polygon, as it is Ethereum Virtual Machine (EVM) compatible, meaning the smart contract is easily deployed to the Polygon blockchain. If parties want to bundle agreement creations, they must find a way to do this outside the smart contract. However, bundling transactions would require another subprotocol, which is out of the scope of this paper.

Table 4: Comparison of fees for individual operations of `Overdraft`'s smart contract when deployed on different blockchains, expressed in USD.

Operation	**Ethereum (GWEI)**			**Algorand (ALGO)**	**Polygon (GWEI)**		
	20	**30**	**40**	**0.001**	**100**	**200**	**300**
Contract deployment	$111.17	$166.76	$222.34	$0.00017	$0.13	$0.26	$0.38
Offline payment (good case)	$2.21	$3.31	$4.41	$0.00017	$0.0025	$0.0051	$0.0076
Offline payment (avg. bad case)	$5.90	$8.84	$11.79	$0.00017	$0.0068	$0.014	$0.02
Creating a loan agreement	$21.30	$32.09	$42.79	$0.00017	$0.025	$0.049	$0.074
Closing a loan agreement	$4.08	$6.12	$8.16	$0.00017	$0.0047	$0.0094	$0.014

Table 5: Fees associated with different quantities bundled loan agreement creation of the smart contract on Ethereum and Polygon, with various network fees, expressed in USD.

Loan Agreement Creation	**Average Individual Fee**				**Total Bundle Fee**			
	Ethereum (GWEI)		**Polygon (GWEI)**		**Ethereum (GWEI)**		**Polygon (GWEI)**	
	20	**30**	**100**	**200**	**20**	**30**	**100**	**200**
5	$20.46	$30.69	$0.023	$0.046	$102.31	$153.47	$0.11	$0.23
10	$20.27	$30.40	$0.0228	$0.0455	$202.68	$304.02	$0.23	$0.46
50	$20.12	$30.17	$0.0226	$0.0452	$1,005.78	$1,508.68	$1.13	$2.26
100	$20.12	$30.17	$0.0226	$0.0452	$2,011.57	$3,017.35	$2.26	$4.52
500	$20.12	$30.17	$0.0226	$0.0452	$10,057.84	$15,086.76	$11.30	$22.60
1000	$20.12	$30.17	$0.0226	$0.0452	$20,115.68	$30,173.52	$22.60	$45.19

8 Discussion

The performance evaluation of `Overdraft` highlights its potential for scalability and efficiency through smart contract implementation and competitive transaction speeds. It provides both large-scale and personal offline transaction solutions with competitive fees. However, we noticed transaction fees may vary significantly depending on the blockchain the smart contract is deployed to, as shown in Table 4. Potential future optimizations include minimizing shared fields and employing compression techniques or integrating `Overdraft` with another L1 or L2 solution to reduce costs and increase performance further. Another approach that we leave for future work consists of using multiple smart contracts that would manage subsets of users (i.e., sharding). This strategy could increase performance and decrease costs per transaction. However, given the complexity of loan networks, this might require synchronizing smart contracts.

Furthermore, privacy remains an area for enhancement. Privacy-preserving technologies like Zero-Knowledge Proofs (ZKPs) [40] could ensure transaction confidentiality without compromising transparency and verifiability. Similar to Monero [41], which uses a variant of ZKPs for anonymity, integrating such measures in `Overdraft` would address privacy concerns effectively.

Moreover, addressing risk contagion in loan networks is crucial to maintaining network integrity. Developing dynamic risk assessment algorithms can monitor and prevent potential threats, averting widespread issues [42].

In addition to these optimizations, an improvement to the incentive formula is proposed for future work. By incorporating the loanee's node reputation into the incentive pricing, the system can ensure more reasonable incentives. This adjustment would account for the trustworthiness of individual nodes, thereby encouraging responsible behavior and enhancing the overall reliability of the network. This enhancement aligns incentives more closely with node reputation and contributes to a more balanced and just financial ecosystem within `Overdraft`.

9 Conclusion

In this work, we presented a novel framework for reputation-weighted loans that facilitates secure offline transactions by integrating network-based lending with blockchain technology. Our system, `Overdraft`, leverages trust and reputation to tolerate and minimize risks such as double-spending and enhancing security in decentralized networks. By implementing a smart contract, an effective loan management algorithm, and an incentive model, we demonstrated that `Overdraft` supports scalable and efficient offline payments with minimal computational demands. Future work will focus on advancing privacy measures, mitigating contagion risks, and improving the incentive formula. This research is pivotal in bridging conventional financial systems and the digital economy, highlighting `Overdraft`'s potential to revolutionize offline payments.

References

[1] M. X. Liu, *Stay competitive in the digital age: the future of banks.* International Monetary Fund, 2021.

[2] M. Fund, "International monetary fund," *Assisting Resource Rich Countries Mobilise and Manage Their Revenues*, 2016.

[3] S. Buteau, P. Rao, and F. Valenti, “Emerging insights from digital solutions in financial inclusion,” *CSI Transactions on ICT*, vol. 9, no. 2, pp. 105–114, 2021.

[4] Y. Chu, J. Lee, S. Kim, H. Kim, Y. Yoon, and H. Chung, “Review of offline payment function of cbdc considering security requirements,” *Applied sciences*, vol. 12, no. 9, p. 4488, 2022.

[5] J. Lind, O. Naor, I. Eyal, F. Kelbert, P. R. Pietzuch, and E. G. Sirer, “Teechain: Reducing storage costs on the blockchain with offline payment channels,” in *Proceedings of the 11th ACM International Systems and Storage Conference, SYSTOR 2018, HAIFA, Israel, June 04-07, 2018*, ACM, 2018, p. 125. DOI: 10.1145/3211890.3211904. [Online]. Available: https://doi.org/10.1145/3211890.3211904.

[6] Y. K. Gupta, G. Jeswani, and O. Pinto, “M-commerce offline payment,” *SN Comput. Sci.*, vol. 3, no. 1, p. 100, 2022. DOI: 10.1007/S42979-021-00978-X. [Online]. Available: https://doi.org/10.1007/s42979-021-00978-x.

[7] W. Blokzijl, “Eurotoken: An offline capable central bank digital currency,” 2021.

[8] J. Poon and T. Dryja, “The bitcoin lightning network: Scalable off-chain instant payments,” 2016.

[9] K. Park and S. H. Baek, “OPERA: A complete offline and anonymous digital cash transaction system with a one-time readable memory,” *IEICE Trans. Inf. Syst.*, vol. 100-D, no. 10, pp. 2348–2356, 2017. DOI: 10.1587/TRANSINF.2016INP0008. [Online]. Available: https://doi.org/10.1587/transinf.2016INP0008.

[10] K. Baqer, R. J. Anderson, L. Mutegi, J. A. Payne, and J. Sevilla, “Digitally: Piloting offline payments for phones,” in *Thirteenth Symposium on Usable Privacy and Security, SOUPS 2017, Santa Clara, CA, USA, July 12-14, 2017*, USENIX Association, 2017, pp. 131–143. [Online]. Available: https://www.usenix.org/conference/soups2017/technical-sessions/presentation/baqer.

[11] P. de Almeida, P. Fazendeiro, and P. R. Inácio, “Societal risks of the end of physical cash,” *Futures*, vol. 104, pp. 47–60, 2018.

[12] M. K. Brunnermeier, H. James, and J.-P. Landau, “The digitalization of money,” National Bureau of Economic Research, Tech. Rep., 2019.

[13] Z. Bezovski, “The future of the mobile payment as electronic payment system,” *European Journal of Business and Management*, vol. 8, no. 8, pp. 127–132, 2016.

[14] G. Uña, A. Verma, M. Bazarbash, and M. N. N. Griffin, *Fintech payments in public financial management: benefits and risks*. International Monetary Fund, 2023.

[15] G. Anakpo, Z. Xhate, and S. Mishi, “The policies, practices, and challenges of digital financial inclusion for sustainable development: The case of the developing economy,” *FinTech*, vol. 2, no. 2, pp. 327–343, 2023.

[16] J. Lind, I. Eyal, F. Kelbert, O. Naor, P. R. Pietzuch, and E. G. Sirer, "Teechain: Scalable blockchain payments using trusted execution environments," *CoRR*, vol. abs/1707.05454, 2017. arXiv: 1707.05454. [Online]. Available: http://arxiv.org/abs/1707.05454.

[17] M. Christodorescu, W. C. Gu, R. Kumaresan, *et al.*, "Towards a two-tier hierarchical infrastructure: An offline payment system for central bank digital currencies," *CoRR*, vol. abs/2012.08003, 2020. arXiv: 2012.08003. [Online]. Available: https://arxiv.org/abs/2012.08003.

[18] D. Cheng, X. Wang, Y. Zhang, and L. Zhang, "Risk guarantee prediction in networked-loans," in *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, C. Bessiere, Ed., ijcai.org, 2020, pp. 4483–4489. DOI: 10.24963/IJCAI.2020/618. [Online]. Available: https://doi.org/10.24963/ijcai.2020/618.

[19] D. Cheng, Z. Niu, and L. Zhang, "Delinquent events prediction in temporal networked-guarantee loans," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 34, no. 4, pp. 1692–1704, 2023. DOI: 10.1109/TNNLS.2020.3027346. [Online]. Available: https://doi.org/10.1109/TNNLS.2020.3027346.

[20] P. Resnick, K. Kuwabara, R. Zeckhauser, and E. Friedman, "Reputation systems," *Commun. ACM*, vol. 43, no. 12, pp. 45–48, 2000. DOI: 10.1145/355112.355122. [Online]. Available: https://doi.org/10.1145/355112.355122.

[21] G. Karame, E. Androulaki, and S. Capkun, "Double-spending fast payments in bitcoin," in *the ACM Conference on Computer and Communications Security, CCS'12, Raleigh, NC, USA, October 16-18, 2012*, T. Yu, G. Danezis, and V. D. Gligor, Eds., ACM, 2012, pp. 906–917. DOI: 10.1145/2382196.2382292. [Online]. Available: https://doi.org/10.1145/2382196.2382292.

[22] P. Everaere, I. Simplot-Ryl, and I. Traoré, "Double spending protection for e-cash based on risk management," in *Information Security - 13th International Conference, ISC 2010, Boca Raton, FL, USA, October 25-28, 2010, Revised Selected Papers*, M. Burmester, G. Tsudik, S. S. Magliveras, and I. Ilic, Eds., ser. Lecture Notes in Computer Science, vol. 6531, Springer, 2010, pp. 394–408. DOI: 10.1007/978-3-642-18178-8_33. [Online]. Available: https://doi.org/10.1007/978-3-642-18178-8%5C_33.

[23] B. Nasrulin, G. Ishmaev, and J. Pouwelse, "Meritrank: Sybil tolerant reputation for merit-based tokenomics," in *4th Conference on Blockchain Research & Applications for Innovative Networks and Services, BRAINS 2022, Paris, France, September 27-30, 2022*, IEEE, 2022, pp. 95–102. DOI: 10.1109/BRAINS55737.2022.9908685. [Online]. Available: https://doi.org/10.1109/BRAINS55737.2022.9908685.

[24] A. Ulrich, R. Holz, P. Hauck, and G. Carle, "Investigating the openpgp web of trust," in *Computer Security - ESORICS 2011 - 16th European Symposium on Research in Computer Security, Leuven, Belgium, September 12-14, 2011. Proceedings*, V. Atluri and C. Díaz, Eds., ser. Lecture Notes in Computer Science, vol. 6879, Springer, 2011, pp. 489–507. DOI:

10.1007/978-3-642-23822-2_27. [Online]. Available: https://doi.org/10.1007/978-3-642-23822-2%5C_27.

[25] S. L. Garfinkel, *PGP - pretty good privacy: encryption for everyone (2. ed.)* O'Reilly, 1995, ISBN: 978-1-56592-098-9.

[26] N. Hughes and S. Lonie, "M-pesa: Mobile money for the "unbanked" turning cellphones into 24-hour tellers in kenya," *Innovations: technology, governance, globalization*, vol. 2, no. 1-2, pp. 63–81, 2007.

[27] P. Otte, M. de Vos, and J. Pouwelse, "Trustchain: A sybil-resistant scalable blockchain," *Future Gener. Comput. Syst.*, vol. 107, pp. 770–780, 2020. DOI: 10.1016/J.FUTURE.2017.08.048. [Online]. Available: https://doi.org/10.1016/j.future.2017.08.048.

[28] S. Dziembowski, L. Eckey, S. Faust, and D. Malinowski, "PERUN: virtual payment channels over cryptographic currencies," *IACR Cryptol. ePrint Arch.*, p. 635, 2017. [Online]. Available: http://eprint.iacr.org/2017/635.

[29] M. Green and I. Miers, "Bolt: Anonymous payment channels for decentralized currencies," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017, Dallas, TX, USA, October 30 - November 03, 2017*, B. Thuraisingham, D. Evans, T. Malkin, and D. Xu, Eds., ACM, 2017, pp. 473–489. DOI: 10.1145/3133956.3134093. [Online]. Available: https://doi.org/10.1145/3133956.3134093.

[30] Z. Avarikioti, E. Kokoris-Kogias, R. Wattenhofer, and D. Zindros, "Brick: Asynchronous incentive-compatible payment channels," in *Financial Cryptography and Data Security - 25th International Conference, FC 2021, Virtual Event, March 1-5, 2021, Revised Selected Papers, Part II*, N. Borisov and C. Díaz, Eds., ser. Lecture Notes in Computer Science, vol. 12675, Springer, 2021, pp. 209–230. DOI: 10.1007/978-3-662-64331-0_11. [Online]. Available: https://doi.org/10.1007/978-3-662-64331-0%5C_11.

[31] Z. Katona, P. P. Zubcsek, and M. Sarvary, "Network effects and personal influences: The diffusion of an online social network," *Journal of marketing research*, vol. 48, no. 3, pp. 425–443, 2011.

[32] C. Dannen, *Introducing Ethereum and solidity*. Springer, 2017, vol. 1.

[33] V. Buterin *et al.*, "A next-generation smart contract and decentralized application platform," *white paper*, vol. 3, no. 37, pp. 2–1, 2014.

[34] S. Martinazzi and A. Flori, "The evolving topology of the lightning network: Centralization, efficiency, robustness, synchronization, and anonymity," *Plos one*, vol. 15, no. 1, e0225966, 2020.

[35] Nomic Foundation, *Hardhat*, version 2.22.1, Mar. 14, 2024. [Online]. Available: https://hardhat.org.

[36] Ethereum, *Gas and fees*, Mar. 2024. [Online]. Available: https://ethereum.org/en/developers/docs/gas/.

[37] S. Micali, “ALGORAND: the efficient and democratic ledger,” *CoRR*, vol. abs/1607.01341, 2016. arXiv: 1607.01341. [Online]. Available: http://arxiv.org/abs/1607.01341.

[38] M. Bjelic, S. Nailwal, A. Chaudhary, and W. Deng, *Pol: One token for all polygon chains*, Jul. 2023. [Online]. Available: https://polygon.technology/papers/pol-whitepaper.

[39] H. Guo and X. Yu, “A survey on blockchain technology and its security,” *Blockchain: research and applications*, vol. 3, no. 2, p. 100 067, 2022.

[40] U. Feige, A. Fiat, and A. Shamir, “Zero-knowledge proofs of identity,” *J. Cryptol.*, vol. 1, no. 2, pp. 77–94, 1988. DOI: 10.1007/BF02351717. [Online]. Available: https://doi.org/10.1007/BF02351717.

[41] K. M. Alonso and J. Herrera-Joancomartí, “Monero - privacy in the blockchain,” *IACR Cryptol. ePrint Arch.*, p. 535, 2018. [Online]. Available: https://eprint.iacr.org/2018/535.

[42] D. Cheng, Z. Niu, and Y. Zhang, “Contagious chain risk rating for networked-guarantee loans,” in *KDD ’20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*, R. Gupta, Y. Liu, J. Tang, and B. A. Prakash, Eds., ACM, 2020, pp. 2715–2723. DOI: 10.1145/3394486.3403322. [Online]. Available: https://doi.org/10.1145/3394486.3403322.

A Loan Agreement Details

Table 6: Loan Agreement fields with sizes and descriptions.

Field	**Size (bits)**	**Description**
Agreement ID	256	The unique identifier of the agreement between two parties.
Public keys	$160 \cdot 2$	The unique identifiers of the agreement's parties in the network.
Reputation scores	$256 \cdot 2$	The reputation scores of both parties.
Loaned amount	256	The tokens of trust and the liability the loaning node accepts. This is used to verify whether it is feasible to loan for the required amount.
Repayment time	256	The terms include the duration of blocks for repayment of the loaned amount, including the interest rate.
Dispute resolution	256	The mechanism to resolve disputes in the agreement for scenarios where there is a disagreement over agreement terms or the fulfillment.
Agreement duration	256	The amount of time an agreement will be usable.
Min. Open time	256	The minimum opening time when the agreement can be established. This value is required for the interest rate and the agreement between the parties.
Close time	256	The time when the agreement has been terminated. This value is required to calculate the final interest rate and end the agreement between the parties.
Opening fee	256	The fee to set up the agreement between two parties. Both parties need to pay the amount.
Closing fee	256	The fee to terminate the agreement between two parties. If the agreement is terminated and used in an offline payment, the loanee pays the fee.
Opening block	256	Represents the block number of the agreement opening transaction.
Closing block	256	Represents the block number of the agreement termination transaction.
Active	256	Represents whether the loan agreement is still active. If active, the amount has yet to be claimed and accrues interest rate. If inactive, the amount has been used, and the repayment time will start.

B Payment Protocol

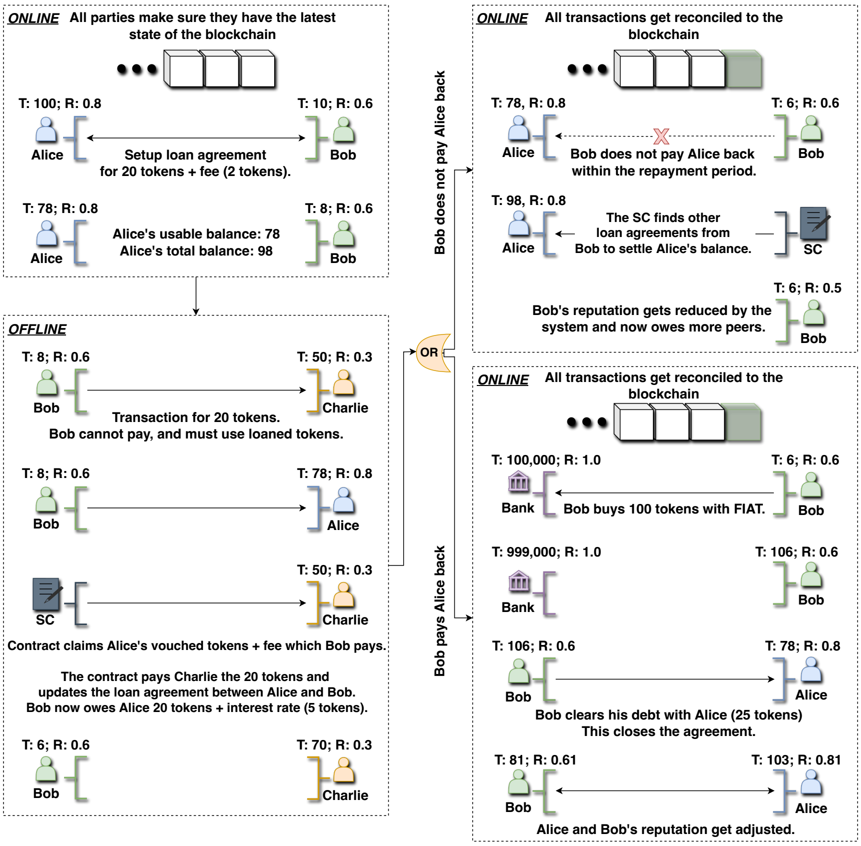

ONLINE All parties make sure they have the latest state of the blockchain
T: 100; R: 0.8
T: 10; R: 0.6
Alice
Bob
Setup loan agreement for 20 tokens + fee (2 tokens).
T: 78; R: 0.8
T: 8; R: 0.6
Alice
Bob
Alice's usable balance: 78
Alice's total balance: 98
OFFLINE
T: 8; R: 0.6
T: 50; R: 0.3
Bob
Charlie
Transaction for 20 tokens.
Bob cannot pay, and must use loaned tokens.
T: 8; R: 0.6
T: 78; R: 0.8
Bob
Alice
T: 50; R: 0.3
SC
Charlie
Contract claims Alice's vouched tokens + fee which Bob pays.
The contract pays Charlie the 20 tokens and updates the loan agreement between Alice and Bob. Bob now owes Alice 20 tokens + interest rate (5 tokens).
T: 6; R: 0.6
T: 70; R: 0.3
Bob
Charlie
OR
Bob does not pay Alice back
Bob pays Alice back
ONLINE All transactions get reconciled to the blockchain
T: 78, R: 0.8
T: 6; R: 0.6
Alice
Bob
Bob does not pay Alice back within the repayment period.
T: 98, R: 0.8
Alice
The SC finds other loan agreements from Bob to settle Alice's balance.
SC
T: 6; R: 0.5
Bob's reputation gets reduced by the system and now owes more peers.
Bob
ONLINE All transactions get reconciled to the blockchain
T: 100,000; R: 1.0
T: 6; R: 0.6
Bank
Bob buys 100 tokens with FIAT.
Bob
T: 999,000; R: 1.0
T: 106; R: 0.6
Bank
Bob
T: 106; R: 0.6
T: 78; R: 0.8
Bob
Alice
Bob clears his debt with Alice (25 tokens)
This closes the agreement.
T: 81; R: 0.61
T: 103; R: 0.81
Bob
Alice
Alice and Bob's reputation get adjusted.

Fig. 6: An overview of Overdraft's payment protocol displaying loan agreement creation, an offline payment utilizing the loan, and the repayment possibilities.

C Recursive Random Walk Algorithm

Algorithm 1 Random walk distribution loaned amounts

```
Require: node (Node), loaned_amount (integer), visited_edges (set of Edges), path (list of integer), transaction_amount (integer), decay (float), root_node (Node), max_distance (integer)
Ensure: amount (integer)
current_path ← path + [node.node_id]
if node is root_node AND len(current_path) $>$ 1 then
    return 0
end if
distance ← length(current_path) - 1
decayed_reputation ← node_reputation · (decay$^{distance}$)
is_root_node ← node.node_id == root_node.node_id
if is_root_node then
    will_pay ← random decision based on the root node's reputation
else
    will_pay ← random decision based on decayed_reputation
end if
if will_pay is True then
    return loaned_amount
else
    amount ← 0
    edges ← list of edges from node.edges not in visited_edges
    for each edge in edges do
        if amount $\geq$ transaction_amount OR distance $\geq$ max_distance then
            break
        end if
        visited_edges.add(edge)
        predecessor ← edge.from_node
        if predecessor.id is root_node.id then
            break
        end if
        amount ← amount + random_walk(predecessor, edge.loaned_amount, visited_edges, current_path, root_node)
    end for
    return amount
end if
```

D Proof: Splitting Reputation Attack

Proof. Let R be the reputation of a single node capable of loaning X, where X is a fixed value for the amount loaned. Consider ϵ as a small fixed constant representing the penalty factor for splitting the reputation into multiple nodes. Let R_1 and R_2 represent two split Sybil nodes such that each node's reputation is bounded by $R_1, R_2 \in [0, R - \epsilon]$ and their combined reputation is slightly less than R: $R_1 + R_2 < R + \epsilon$. Let Y and $X - Y$ be the amounts of these Sybil nodes loans, respectively, ensuring that the total amount remains X.

The influence of the original node is $R \cdot X$. The combined influence of the Sybil nodes is $R_1 \cdot Y + R_2 \cdot (X - Y)$. Given the penalty for splitting and that the reputation system effectively reduces the influence of Sybil nodes, we assume:

$$R \cdot X \geq R_1 \cdot Y + R_2 \cdot (X - Y)$$

The reputation system penalizes splitting by reducing the effective reputation of each split node by ϵ for each split transaction. We express this penalty in our inequality:

$$R \cdot X > (R_1 + R_2 - \epsilon) \cdot X$$

This can be rewritten to:

$$R \cdot X > R_1 \cdot (Y + (X - Y)) + R_2 \cdot (Y + (X - Y)) - \epsilon \cdot X$$

$$R \cdot X > R_1 \cdot Y + R_2 \cdot (X - Y) + [(R_1 \cdot (X - Y) + R_2 \cdot Y) - \epsilon \cdot X]$$

We need to show that:

$$R_1 \cdot (X - Y) + R_2 \cdot Y - \epsilon \cdot X > 0$$

To show that creating Sybil nodes is not profitable, we consider the case where the original node's reputation is greater than twice the penalty applied by the reputation system for splitting into Sybil nodes, i.e., for two split nodes, we need to show that $R > 2 \cdot \epsilon$.

We then examine the profitability condition for Sybil creation:

$$R_1 \cdot (X - Y) + R_2 \cdot Y > \epsilon \cdot X$$

Given that R_1 and R_2 are fractions of R, and that splitting is penalized by ϵ, the condition becomes hard to satisfy as R increases relative to ϵ. In essence, the larger the original reputation and the stricter the penalty for Sybil creation, the less likely this inequality is to hold, indicating that creating Sybil nodes becomes unprofitable:

$$\frac{R}{2} \cdot \frac{X}{2} + \frac{R}{2} \cdot \frac{X}{2} = \frac{R \cdot X}{2} > \epsilon \cdot X$$

$$R > 2 \cdot \epsilon$$

Moreover, the generalization for any number of Sybil nodes (K) where each node loans an amount X/K with a reputation R/K yields:

$$K \cdot \frac{R}{K} \cdot \frac{X}{K} > \epsilon \cdot X$$

$$R > K \cdot \epsilon$$

E Smart Contract Pseudocode

Protocol 1: Create Agreement

Data: Public keys $\mathrm{pk}_1, \mathrm{pk}_2$, loan amount L, repayment time T_r, dispute resolution D_r, agreement duration T_d, min open time T_o, opening fee F_o.
Result: Agreement creation and ID assignment.

```
1. Validate that $|\mathrm{pk}| = 2$.
2. Generate $agreementId \leftarrow \mathrm{nextAgreementId}$.
3. Create agreement with details $\{agreementId, \mathrm{pk}_1, \mathrm{pk}_2, L, T_r, D_r, T_d, T_o, F_o\}$.
4. Store agreement in $\mathrm{agreements}[agreementId]$.
5. Emit AgreementCreated(agreementId).
```

Protocol 2: Loan Tokens

Data: Sender A, recipient B, loan amount L.
Result: Tokens loaned from A to B and agreement created.

```
1. Lock $L$ tokens from $A$'s balance.
2. Create agreement with public keys $\{A, B\}$, loan amount $L$, default terms.
3. Update $\mathrm{lastLoanAgreementId}[A][B] \leftarrow agreementId$.
4. Record loan in $\mathrm{loanedAmounts}[A][B] \leftarrow L$.
5. Emit Loaned(A, B, L).
```

Protocol 3: Transfer Tokens

Data: Sender A, recipient B, amount L.
Result: Tokens transferred from A to B.

```
1. Calculate $availableBalance \leftarrow \mathrm{balanceOf}(A)$.
2. if $availableBalance < L$
   (a) Calculate $shortfall \leftarrow L - availableBalance$.
   (b) Attempt to use loaned tokens via useLoanedTokens(A, B, shortfall).
3. Update balances: $\mathrm{balances}[A] \leftarrow \mathrm{balances}[A] - L$.
4. Update balances: $\mathrm{balances}[B] \leftarrow \mathrm{balances}[B] + L$.
5. Emit Transfer(A, B, L).
```

Supplementary Material

The second part of this thesis contains supplementary material accompanying the research paper of chapter 1. The supplementary material consists of four chapters: 2 extended related work, 3 extended system overview & details, 4 additional experiments, and 5 conclusion.

2

Extended Related Work

Our global payment system increasingly relies on internet connectivity due to the popularity of various payment gateways and services. While offline payment methods like bills and coins are still globally relevant, more businesses are moving away due to the higher risks of counterfeiting and burglary. This shift poses challenges when online electronic payment systems, such as point-of-sale (POS) systems, go offline due to network outages or in remote and high-security environments where internet access is unreliable or restricted. In such situations, offline token transactions offer an essential alternative that needs to meet several critical criteria. To ensure financial inclusivity, these systems must be accessible and usable by all population segments, including those in underbanked and unbanked communities. In this chapter, we will provide background information, providing additional prerequisite knowledge for understanding `Overdraft`.

2.1. Web of Trust

The Web of Trust (WoT), the core concept of our system, serves as a decentralized model used in cryptography and digital certificate management, differing from centralized systems like Certificate Authorities (CAs) [1]. In a WoT, trust is established through mutual verification among users, creating a flexible and democratic trust mechanism. This model reduces single points of failure risks and enables personalized trust relationships. Trust in a WoT is not binary but exists on a spectrum, allowing for varied trust levels among users. Figure 2.1 shows an example WoT-based network. It displays the direct relationships between entities and their implied relationships through direct trust.

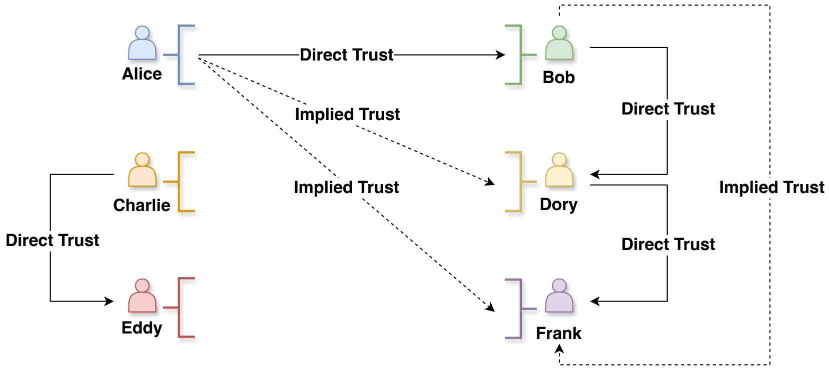


Figure 2.1: Example Web of Trust network with their direct and implied trust relationships.

The most notable application of WoT is in Pretty Good Privacy (PGP) [2], used for securing email communications. PGP combines symmetric and asymmetric encryption, with WoT as a means for users to verify public keys through mutual signatures. This decentralized method forms a chain of trust, relying on the endorsements of others within the network.

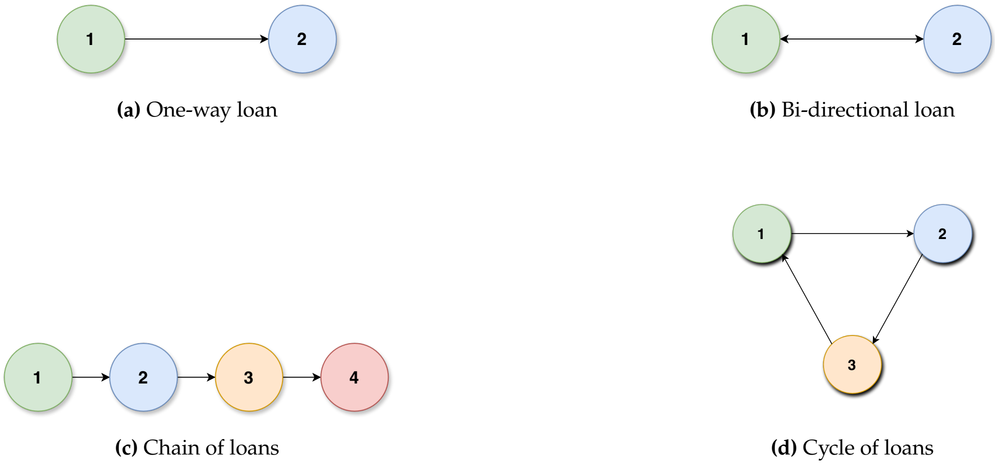
Despite its innovative design, WoT faced challenges leading to its limited adoption. Its complexity required users to verify public keys manually, often necessitating physical meetings, making it impractical for widespread use. As users grew, scalability issues arose, making trust relationships hard to manage [3]. The lack of universal adoption and security vulnerabilities, such as trust hijacking, further hindered its effectiveness.

Analyzing WoT involves examining its computational intensity and algorithms for calculating trust. Trust computations in a WoT can be seen as graph-based problems, where nodes represent users and edges signify trust relationships. Algorithms, such as Personalized PageRank [4] and network flow algorithms like MaxFlow [5], can be applied to analyze trust levels. However, these algorithms face challenges in handling interconnected paths and require adaptations to address trust and distrust dynamics. These challenges make traversing a WoT complex and require adaptations or fully customized algorithms that can fulfill all the prerequisites.

2.2. Network-Based Loans

Network-based loans are closely related to the concept of WoT and are also integral to our system; network-based loans [6] refer to the requirement of financial loans determined by the person's social connections or community relationships rather than relying on traditional evaluation methods like credit scores or collateral. This lending takes advantage of the trust and social capital within an individual's community or social network. The criteria for granting a loan could involve examining the person's relationship with network members, their activities on social media, or recommendations from peers within the network. Figure 2.2 displays the different lending relationships possible in such a network. It is possible to have one-way, bi-directional, chains, and cyclic relationships between users in the network. These types provide more coverage for possible loans in the network.

Moreover, network-based lending is particularly beneficial in regions or groups that may not have easy access to standard banking services. It aims to broaden financial inclusion by offering financial services to those who lack a formal credit history but possess strong community bonds and trust [7].



(a) One-way loan

(b) Bi-directional loan

(c) Chain of loans

(d) Cycle of loans

Figure 2.2: A visualization of the different types of lending relationships that can occur in a loan network.

2.3. Double-Spending Problem

A system with tokens or monetary value, for instance, in loan agreements in loan networks, can be exploited by a malicious user, introducing the double-spending problem. Enabling offline transactions within digital payment systems presents significant complexity, especially in mitigating double-spending risk in a Peer-to-Peer network [8]. An illustration of the double-spending issue is provided in Figure 2.3. The challenge is notably heightened when trying to facilitate these transactions without the support of an online network. Undertaking offline payments necessitates a balance between ensuring transaction availability and maintaining data integrity, where adopting an offline payment mechanism decides the extent of tolerance to network partitioning.

One approach to mitigate double-spending involves using trusted components, such as secure elements or hardware security modules (HSMs), which can provide a trusted execution environment to ensure the integrity of transaction [9, 10]. However, this method requires specialized hardware, adding complexity and cost to deploying such systems. While these trusted components can effectively prevent double-spending by securely storing and processing transaction data, their widespread implementation presents logistical and economic challenges.

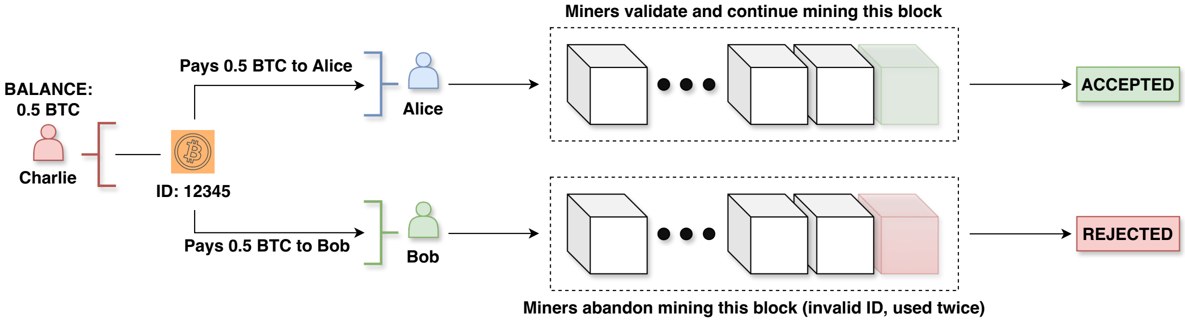

Miners validate and continue mining this block
BALANCE:
0.5 BTC
Charlie
Pays 0.5 BTC to Alice
Alice
ID: 12345
Pays 0.5 BTC to Bob
Bob
ACCEPTED
REJECTED
Miners abandon mining this block (invalid ID, used twice)

Figure 2.3: A visualization of the double-spending problem.

2.4. MeritRank

We utilize reputations in our loan network in order to quantify a user's trustworthiness. MeritRank [11] provides an innovative solution in decentralized networks, particularly in combating Sybil attacks. It employs a decentralized approach where peers within a network actively observe and evaluate each other's contributions, which are recorded in a personal ledger. This process is conceptualized within a directed feedback graph where nodes represent the users, and the weighted edges denote the feedback between users. These weights accumulate over time, reflecting the total feedback a peer has received, thus encapsulating the contributions made by peers to the community.

A key feature of MeritRank is its dynamic model, structured around epochs to capture the evolving nature of contributions and feedback. An epoch represents a complete cycle within the system, with the feedback graph for each epoch signifying the temporal aspect of reputation accumulation.

The reputation mechanism is central to MeritRank. It assigns reputation scores to each node based on the aggregated feedback from the entire network. These scores are crucial for assessing each node's contribution level relative to others, effectively quantifying their reputation within the network.

Moreover, MeritRank incorporates a gossip mechanism for communicating feedback graphs throughout the network. This mechanism ensures that global information is accessible through any fault-free gossip protocol, allowing peers to discover and update the feedback graph, thus maintaining the integrity and currency of the reputation scores.

The allocation mechanism in MeritRank leverages these reputation scores to determine the distribution of rewards. According to a predefined allocation policy, rewards are distributed automatically at the end of each epoch based on the reputation scores. This mechanism ensures that contributions are incentivized and fairly rewarded, aligning individual motivations with the overall health and security of the network.

Sybil Models

In addressing the challenge of Sybil attacks within decentralized networks, MeritRank models such attacks as strategic maneuvers by an attacker aiming to maximize rewards through minimal effort. This attack is typically achieved by creating numerous fake identities (Sybils) and establishing fake connections between them.

A Sybil attack is characterized by introducing a set of fake identities, each designed to be indistinguishable from legitimate nodes. The attacker also creates two types of edges: Sybil edges determined at the attacker's discretion and attack edges that connect the attacker's Sybil identities to honest nodes. Following the attack, the feedback graph is altered to signify the new state of the network post-attack.

MeritRank evaluates the effectiveness of Sybil attacks by considering the creation of numerous Sybil nodes and fake edges. However, to gain credibility, these Sybil identities must engage in real transactions, often with highly reputable nodes, to enhance the attack's impact. The profit from a Sybil attack is measured by the total reputation scores of all Sybil identities, excluding the original attacker.

On the other hand, the cost of a Sybil attack is determined by the reputation gained through legitimate interactions, represented by the edges created during the attack. This cost is calculated as the total reputation of all involved Sybil identities in the modified network.

Sybil tolerance measures the resilience of a reputation system against such attacks. A system is deemed Sybil tolerant if the relative benefit of executing a Sybil attack remains constant, even as the number of Sybil identities increases indefinitely. This indicates the system's effectiveness in limiting the benefits an attacker can gain from executing Sybil attacks.

Bounding Attacks

MeritRank enhances the robustness of reputation systems against Sybil attacks by introducing strategic modifications to mitigate the vulnerabilities exploited by serial, cycle, and parallel attacks.

- **Parallel Report Bound:** This modification targets vulnerabilities from parallel and cycle attacks by limiting the total reputation gain across all Sybil nodes involved in such attacks not to exceed the reputation of the first Sybil node introduced. Ensuring that the cumulative benefit of adding multiple Sybil nodes is controlled. Furthermore, the weight of each feedback edge is recalibrated based on the originating node's connectivity level, thus balancing each node's influence and countering the impact of Sybil nodes created during attacks.
- **Serial Report Bound:** This adjustment caps the total reputation gain from sequentially added Sybil nodes to a finite value, preventing serial attacks from indefinitely boosting the attacker's reputation.
- **Bounded Transitivity:** This principle restricts the reputation propagation mechanism within the network. It ensures that the reputation received by a node via any path does not exceed the lowest reputation among all nodes on that path, maintaining a truthful reflection of each node's trustworthiness.

Decay Strategies for Enhanced Sybil Tolerance

MeritRank introduces mechanisms based on decaying transitivity, connectivity, and epoch-based adjustments to mitigate the impact of serial Sybil attacks within the network. These strategies are designed to limit the advantage that attackers can gain through various manipulation tactics, ensuring a robust and Sybil-resistant reputation system.

Transitivity with α decay. For reputation scores derived from random-walk algorithms, the propagation length of these walks is limited using an alpha decay constant. This approach terminates the random walk at each step with a probability corresponding to alpha. It reduces the reach of serial attacks through extended fake node chains, as the decay effectively shortens the path lengths of influence.

Connectivity with β decay. This strategy enhances Sybil resistance by penalizing nodes that are part of a separate connected component, mainly targeting the structural vulnerabilities exploited by bridge connections between honest and Sybil nodes. The modified reputation score, $R_\beta(G, j)$, incorporates a beta decay constant for nodes connected through such bridges. Specifically, if there is a bridge between nodes i and j, the reputation score is adjusted as $(1 - \beta) \cdot R(G, j)$; otherwise, the reputation score remains $R(G, j)$.

The detection of these bridge connections relies on a cut vertex within the path from a seed node to the target node, where a significant portion of random walks pass, exceeding a predefined threshold.

Epoch with γ decay. Gamma decay applies over epochs to counteract the exploitation of outdated connections. This decay diminishes the advantage an attacker gains from maintaining old attack edges, thereby necessitating continuous effort for reputation maintenance. The reputation update formula combines a decay factor and the new contributions to the reputation score. It adjusts the score by applying a decay to the previous reputation score, reducing it by a factor of $(1 - \gamma)$, and then adding the reputation contribution from the changes in the graph during the current epoch. This approach ensures that older connections gradually lose influence while newer interactions are appropriately accounted for in the reputation score.

2.5. Existing Offline Payment Solutions

2.5.1. Trusted Execution Environments

A Trusted Execution Environment (TEE) is a secured area within a processor or module that guarantees that the code and data loaded in the processor will be protected [12].

TEEs provide a way to facilitate offline payments and mitigate double-spending attacks. Offline payments typically challenge the need to verify transactions without real-time access to a central database or network. TEEs address these challenges by securely isolating payment-processing software from the rest of the device system. This isolation allows the software within the TEE to securely store, process, and verify transaction data independently. When a payment is initiated, the TEE can securely track and update the balance of a digital wallet, ensuring that funds are only spent once. This update involves cryptographically signing the transaction, recording it, and effectively deducting the amount from the payer's wallet, preventing the funds from being double-spent. TEEs can ensure the payment process is reliable and secure for offline payments, even without network connectivity. They can do this by pre-authorizing a set number of transactions or a total monetary amount that can be securely spent offline. Once connectivity is restored, these transactions can be synchronized with a central server to update the global ledger and verify the integrity of all offline transactions.

Teechain

Teechain [13] is a payment channel and multi-hop payment protocol that facilitates efficient and secure off-chain bilateral transfers, even with just asynchronous access to the blockchain.

Unlike conventional payment channels that demand synchronous access, Teechain does not require users to monitor the blockchain to prevent potential attacks continuously. The protocol employs three primary techniques:

- **Asynchronous Blockchain Access:** Teechain uses TEEs in modern CPUs to achieve asynchronous access. These TEEs safeguard code and data in a dedicated memory area, shielding them from potential threats, even if an attacker controls the hardware. Teechain utilizes on-chain deposits as collateral, managed by the TEEs, ensuring a smooth flow of funds between various payment channels and preventing misuse of outdated channel states.
- **Payment Chain Support:** Teechain is designed to handle payments that traverse multiple channels or "hops." It guarantees that a payment is completed or all channels involved return to their original state. This safeguard ensures that no funds are misplaced, double-spent, or left pending due to failures along the payment route.
- **Fault Tolerance:** Teechain offers two fault tolerance mechanisms depending on the user's needs. For casual users, it employs hardware monotonic counters within the TEE to store state information and prevent replay attacks. For frequent transactions, like those in exchanges, Teechain introduces a form of chain replication, assuring reliability as long as one TEE remains operational. This results in Teechain delivering better performance than the Lightning Network [14].

Teechain's foundation lies in TEEs, which isolate code and data to ensure their confidentiality and integrity. The protocol can operate with any TEE, but its initial application uses Intel's Software Guard Extensions (SGX) to run code in a protected zone called an enclave. TEEs also offer remote attestation capabilities, allowing third parties to confirm that software runs within a legitimate TEE.

In Teechain's payment channel protocol, parties establish trust using remote attestation and open two-way payment channels. Before transferring funds, a blockchain deposit, managed by a Teechain TEE, is necessary. Throughout the channel's operation, the TEEs securely manage the channel's status. Fund transfers between the two parties are backed by their deposits. Payments are made via a secure interface, and channel balances are exclusively managed within the TEEs. Channels can be terminated anytime, with the TEE creating a blockchain transaction only upon closure.

2.5.2. Rollups

Rollups [15, 16] are a layer-2 (L2) scaling solution that enhances the scalability of blockchains, especially Ethereum. They execute and store most transactional data off the main Ethereum chain while periodically submitting a summary or proof of these off-chain transactions to the blockchain.

Instead of every transaction being processed directly on the mainnet, they are grouped and processed off-chain in a rollup sidechain. Only a condensed version, often cryptographic proofs or summaries, is submitted to the mainnet. By doing this, rollups increase the number of transactions handled per second while maintaining a connection to Ethereum's security.

Rollups operate with the Ethereum mainnet, leveraging the security while extending its capacity. Two primary types of rollups are:

- **Optimistic Rollups [15]** Transactions are handled off-chain; once a summary is submitted to the Ethereum mainnet, it is presumed valid. A waiting period follows, allowing any observer to contest its validity if they suspect an error or malicious activity. If a challenge arises, the specific transaction is carefully examined. It is rejected if found invalid, and penalties are imposed on the malicious actor. Figure 2.4 shows how optimistic rollup transactions are finalized on a layer-1 (L1) main blockchain.

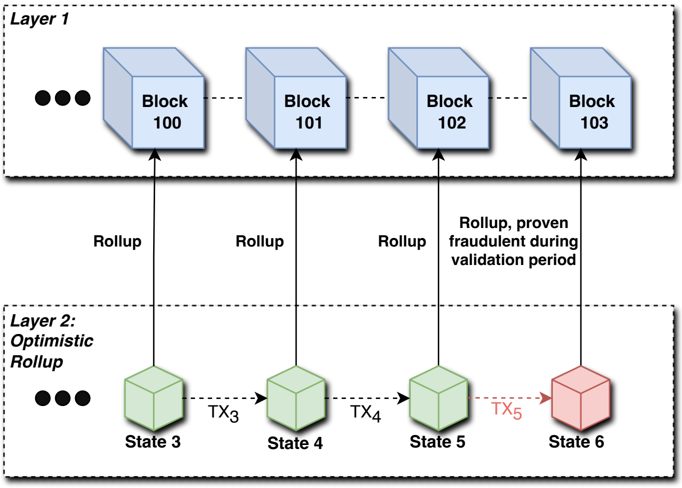


Figure 2.4: A figure depicting an L2 Optimistic Rollup batch transaction being finalized on the main blockchain. It displays successful and unsuccessful rollup finalization.

- **zk-Rollups [16]** This version utilizes zk-SNARKs (Zero-Knowledge Succinct Non-Interactive Arguments of Knowledge) to process transactions. Zk-SNARKs are cryptographic proofs that verify transactions without disclosing their details. In a zk-Rollup, transactions are combined into a single zk-SNARK, which is then submitted to the Ethereum mainnet. This transformation ensures that all transactions within the rollup are valid without needing an "optimistic" challenge mechanism. Figure 2.5 shows how zk-Rollup transactions are finalized on an L1 main blockchain.

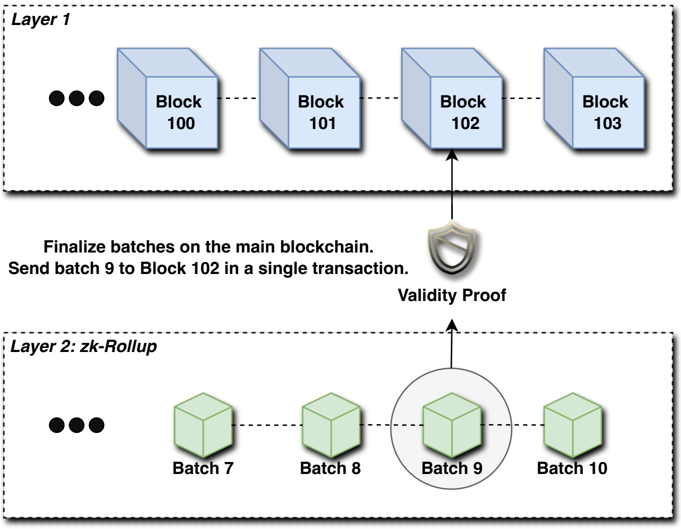


Figure 2.5: A figure depicting an L2 zk-Rollup batch transaction being finalized on the main blockchain.

Rollups support offline transactions by processing transactions off-chain before submitting the summary of all transactions to the mainnet. Participants can transact within the rollup's environment even when not connected to the Ethereum network. These offline transactions are recorded and stored within the rollup's infrastructure.

When participants or the rollup operators reconnect to the Ethereum network, they can bundle and process all the offline transactions, eventually submitting the corresponding summary or proof to the mainnet. This makes rollups a solution for scenarios where consistent online connectivity is not guaranteed. However, transactional continuity and later anchoring to the Ethereum mainnet are needed to reach finality.

Rollups provide a few ways of counteracting double-spending even when transacting offline. Online or offline transactions are processed consistently and sequentially within the rollup environment. Even if a user tries to initiate two conflicting transactions offline, only the first transaction will be considered valid once the transactions are processed and ordered within the rollup. Any subsequent conflicting transaction would be detected as a double-spend and will be invalidated.

2.5.3. Payment Channels

As blockchains became more extensive and complex, finding efficient transaction processes became crucial. The Lightning Network (LN) [14] for Bitcoin demonstrates how payment channels can tackle blockchain scalability problems and prevent double-spending. By processing transactions off the blockchain, these solutions enhance the network's efficiency and decrease the demand on the blockchain's processing capacity.

The LN utilizes a mechanism known as fund locking on the blockchain, initiated by setting up a payment channel between two entities. This channel is formed through a mutual agreement, wherein both participants allocate a specified amount of Bitcoin into a multi-signature wallet. This wallet, requiring approval from both parties to access the funds, ensures joint control. The creation of this channel is marked by a "funding transaction," a public declaration on the Bitcoin blockchain indicating the reservation of funds for the channel's use. Once the channel is established and the funds are secured, transactions can occur off the blockchain. This process enables the parties involved to conduct immediate and private fund exchanges without engaging the entire Bitcoin network for each transaction. Transactions within the channel are documented in a "commitment transaction," which reflects the current fund distribution between the participants. This documentation is kept private and only shared with the blockchain if the channel is terminated. Figure 2.6 depicts a visual overview of a payment channel.

The synchronous nature of this solution is essential. Both parties must agree to the current balance and consent to any updates. This is crucial for ensuring that the system remains secure and that both parties agree on the distribution of funds. If one party wishes to exit the channel or disputes arise, the latest agreed-upon commitment transaction can be broadcast to the blockchain. This action closes the channel and distributes the funds according to the last known balance agreed upon by both parties.

The LN reduces the number of transactions broadcast to the blockchain by allowing a connected network of payment channels to be set up. It allows parties to make payments over many channels without the need to trust the intermediate bypassing nodes or set up new payment channels. When a node fails to forward payment or refuses to perform the forwarding action, it cannot snatch the funds traveling through the network of channels [17].

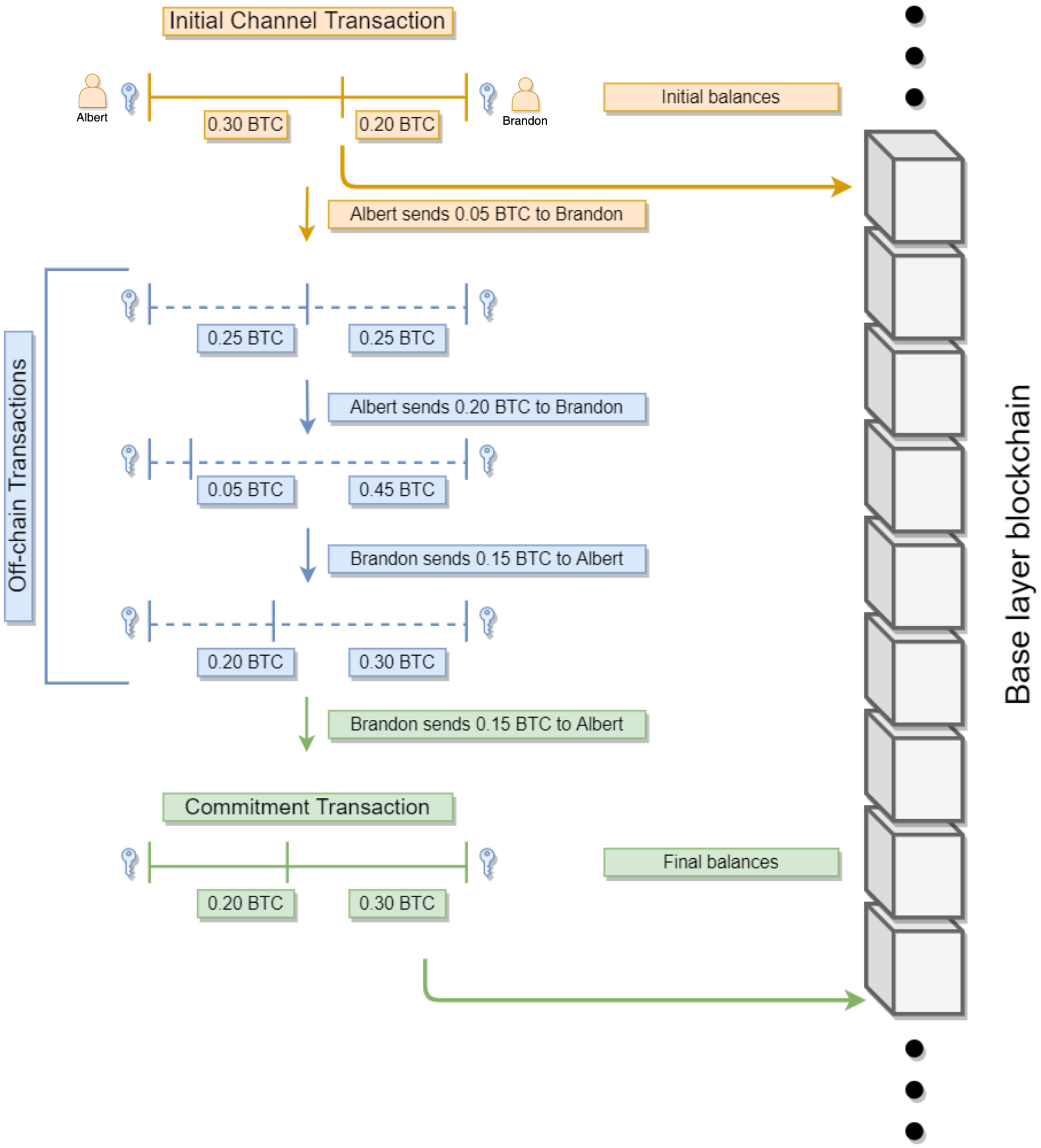


Figure 2.6: Representation of a LN payment channel between two users. Showing the funding transaction, off-chain transactions, and commitment transaction.

Lightning Network Topology

The LN is structured as a network of payment channels between parties, with the transaction details being partly public and partly private [18]. A unique ID identifies each channel in the network and includes data on the channel's capacity, the nodes involved, and detailed information about the opening and closing transactions. The channels are represented as edges in the network and nodes as vertices.

The payment channel's fields consist of many variables, such as the channel's opening and closing transactions, timestamps, block numbers, and fees. Furthermore, the channel's unique ID and capacity in satoshi (part of Bitcoin, with a maximum of 8 decimals) are provided, as are the public keys of the pair of nodes in the payment channel.

According to Lisi et al., [18], the LN is modeled as a weighted undirected multigraph. Each edge represents a channel, and each node represents a participant in the network. The graph includes functions associating each channel with its opening day, closing day, and capacity. The daily snapshots of the LN are subsets of this multigraph, providing insights into the active channels and nodes on any given day. The LN's topology is structured as a large central connected hub where most nodes can reach each other at a short hop distance. We can also see several small connected components disconnected from the central hub.

Furthermore, the average number of nodes in the network at the time of measuring was approximately ~8,000 nodes. The LN demonstrates a high degree of centrality, with a significant portion of the nodes having a small degree, around 37.7% having degree 1, and 98.82% of the nodes having a degree less or equal to 100. The distribution of nodes' degrees and their centrality measures, such as betweenness and closeness, offer insights into how likely nodes are to be part of a multi-hop payment path. A feature of the LN is the presence of 'bouquets,' structures where peripheral nodes (roses) connect to a central hub node (bouquet root). These bouquets indicate a preference for specific nodes to act as central relay points in the network.

Rebalancing

Rebalancing in off-chain payment networks is essential for maintaining these networks' efficiency and operational capability over time [19]. Payment channels facilitate transactions between parties without committing every transaction to the blockchain, reducing transaction fees and processing times. However, as transactions occur, payment channels can become unbalanced. One party might end up with most or all of the funds on their side of the channel, limiting further transactions in the opposite direction unless the channel is rebalanced or closed and reopened, which incurs high fees and delays. Thus, rebalancing is vital for sustaining the operational effectiveness of off-chain payment networks, allowing for continuous, cost-effective, and efficient transactions within the ecosystem.

Revive [19] introduces a rebalancing scheme designed to address these challenges by enabling the secure and efficient reallocation of funds within the network without necessitating on-chain transactions. This process commences with the election of a leader who coordinates the rebalancing effort. Upon receiving rebalancing requests, the leader triggers the process, during which participants confirm their intent to participate and then freeze the payment channels they wish to rebalance. They communicate their current channel states and how they wish to adjust their balances to the leader, who then calculates a set of rebalancing transactions that meet these objectives without altering the overall balance of the network. This calculation is achieved through a linear programming model that seeks to optimize the funds' distribution while ensuring the integrity and fairness of the transaction set.

Participants review and approve the proposed rebalancing transactions, ensuring consensus before proceeding. The process includes mechanisms to handle disputes or non-responsiveness, allowing challenges to be issued on-chain to revert to the last agreed-upon state. This scheme preserves the liquidity of payment channels. It enhances the overall scalability of the network by reducing the need for on-chain transactions. Figure 2.7 shows how rebalancing works in a small payment channel network with three nodes, where the balance of node 1 in the payment channel between node 1 and node 3 has been depleted.

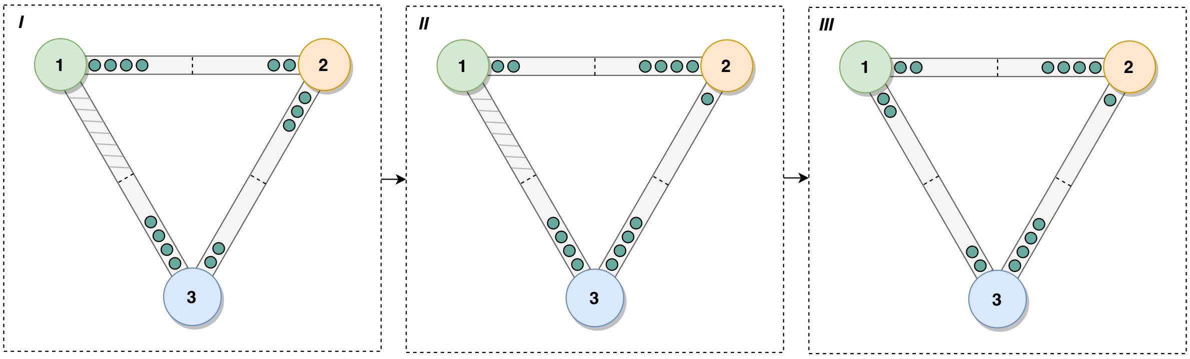


Figure 2.7: An example of how rebalancing works with three payment channels. *I*) Shows the initial status of the payment channel network. We can see that the balance of node 1 in the channel between nodes 1 and 3 is depleted. *II*) Shows the rebalancing up to the final payment channel between nodes 1 and 3. *III*) Rebalance has been completed, and all channels can be used again to transact.

2.6. Unified Payments Interface Lite

In the real world, systems that can conduct offline payments are essential for ensuring seamless financial transactions, even in areas with limited internet access. Unified Payments Interface (UPI) Lite [20], an extension of India's UPI, is a digital platform created by the National Payments Corporation of India (NPCI) [21]. Its primary purpose is to handle low-value transactions (below 500 Indian rupees), which aims to relieve the traffic from frequent small payments. This initiative addresses the need for a more efficient system in managing micro-transactions, reducing the load on banking infrastructure, especially in a country such as India with a significantly large population, spotty internet connectivity, and low-powered mobile devices.

The operational framework of UPI Lite involves users loading a certain amount of money from their bank accounts into a dedicated UPI Lite balance within their UPI application while having an internet connection. This balance is then used for executing multiple small-value transactions, and it is possible to transact even when offline. The payee can pay by scanning the merchant's QR code with their UPI Lite-powered payment app, such as PhonePE, paytm, or Google Pay. The critical advantage of UPI Lite is its ability to process transactions faster by operating outside the banks' core banking systems, thus enhancing transaction speed and reliability.

UPI Lite utilizes the same robust encryption and security protocols as the standard UPI system. While primarily reliant on internet connectivity, the design of UPI Lite holds significant potential for enhancing offline transaction capabilities. This is especially suitable in areas with limited or inconsistent internet access, ensuring the viability of digital transactions.

2.7. Google Pay

Google Pay [22] represents a transformative approach to payment systems, enabling users with Android devices to make payments. This method deviates from the conventional electronic payment systems that rely on physical cards. Utilizing near-field communication (NFC), Google Pay facilitates contactless payments between Android devices and point-of-sale (POS) systems, commonly used globally to support merchant transactions. Users can execute purchases swiftly and with enhanced security by tapping their device against an NFC-compatible POS terminal.

To safeguard these electronic transactions, Google Pay has introduced several security mechanisms. Instead of transmitting the actual bank card number during transactions, a unique token, which replaces the sensitive card information, is used. This approach significantly diminishes the potential for card detail theft. A graphical representation of Google's payment process is shown in Figure 2.8.

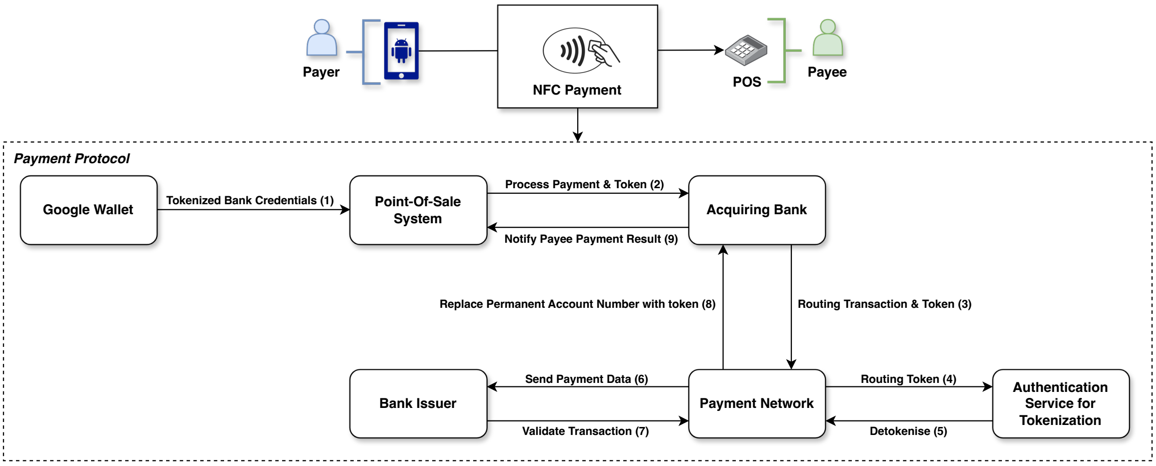

Payer
NFC Payment
POS
Payee
Payment Protocol
Google Wallet
Tokenized Bank Credentials (1)
Point-Of-Sale System
Process Payment & Token (2)
Notify Payee Payment Result (9)
Acquiring Bank
Replace Permanent Account Number with token (8)
Routing Transaction & Token (3)
Bank Issuer
Send Payment Data (6)
Validate Transaction (7)
Payment Network
Routing Token (4)
Detokenise (5)
Authentication Service for Tokenization

Figure 2.8: Visualization of Google's payment protocol.

Although Google Pay operates primarily in online environments, its use of NFC and secure transaction protocols presents an interesting parallel to offline transactions in a trust-based network. Although limited, Google Pay's ability to conduct transactions offline provides a reference point for our research. In a store-based purchase, as shown in Figure 2.8, the POS system must have an internet connection to authorize and verify the payment. While using Google Pay requires internet connectivity for most functions, NFC-enabled transactions can occur without an active internet connection, highlighting the potential for offline digital transactions.

3
Extended System Overview & Details

3.1. Stakeholder Analysis

Understanding our system's complex dynamics is essential when developing a network-based loaning system. `Overdraft` relies on the interaction of various actors, each with unique roles and interests that collectively influence the network's integrity, functionality, and sustainability. Therefore, a comprehensive stakeholder analysis is essential to inform our framework's design, governance, and operational strategies [23]. We identify our most important stakeholders and their respective roles within `Overdraft`. Given `Overdraft`'s reliance on trust and reputation, coupled with the potential for online or offline financial transactions and loans, it becomes essential to identify the parties who might affect or be affected by `Overdraft`'s operations. Our analysis will identify potential risks, opportunities for collaboration, and the requirements and incentives that must be addressed to ensure an effective, utilized, and fair system.

Honest Participants. These nodes are the backbone of our network, engaging in transactions and loan activities that utilize `Overdraft`'s trust mechanism. These nodes, which can range from individual users to entities, utilize `Overdraft` by either offering or receiving loans, essentially staking a claim on each other's trustworthiness with the ability to earn a passive interest rate on their loaned tokens. This interaction facilitates offline payment scenarios, for example, when nodes with limited or no internet connectivity do not have the funds to pay for a transaction while offline and can use the tokens loaned to them.

Malicious Participants. These nodes are realistically always part of networks where it is possible to transact on or to earn tokens, as malicious users will try to abuse `Overdraft` for their benefit. For instance, these users challenge `Overdraft`'s integrity by exploiting the loan agreements and interest rates on loaned amounts by creating multiple fake nodes. This necessitates robust security measures to the network's trust mechanism. Fortunately, through our reputation mechanisms, detecting these malicious participants will make `Overdraft` tolerable for these malicious users in a way that is not profitable for them.

Central Authorities. These nodes, such as banks or financial institutions, function as stabilizing points within the network. With their trustworthiness and financial backing, they act as central banks, providing a reliable source of tokens for loans and positively influencing the framework's financial policy. These entities may make `Overdraft` more centralized; however, users can always choose how to utilize the system independently. `Overdraft` allows users to fully transact decentralized or utilize these central authorities to loan tokens.

3.2. Identities in Trust-Based Networks

3.2.1. Strong Identities

Strong identities are essential to enhancing trust and accountability within trust-based networks. A strong identity framework can accurately and reliably link an online persona to a real-world entity, such as a person or an organization. This association is established through KYC (Know Your Customer) verification processes, such as digital certificates, biometric data, or government-issued identification documents [24]. The primary advantage of implementing strong identities in a network is facilitating trust among participants. When each node's real-world identity is known and verifiable, it creates a transparent environment where parties can engage in transactions with a higher degree of confidence.

For web of trust-based networks, strong identities significantly reduce the risk of fraudulent activities [25]. This reduction is achieved because each transaction or agreement entered into can be legally enforced, relying on the verified identity of the participants. Should a dispute arise, such as a failure to repay a loan, the loaning party possesses the necessary information to seek legal recourse, ensuring that obligations are met. Moreover, the potential for legal action is a deterrent against dishonest behavior, promoting a culture of integrity within the network.

From the perspective of agreement enforcement, strong identities simplify the process of holding parties accountable for their commitments. Whether executing a loan agreement or ensuring the repayment of loaned tokens, the clear identification of each participant streamlines dispute resolution and enforcement actions. This clarity enhances the efficiency of transactions and improves the overall security and reliability of the network.

3.2.2. Weak Identities

Conversely, a system can also have weak identities, which is the most important case for `Overdraft`. Weak identities offer different advantages and challenges within decentralized systems. A weak identity lacks a direct, verifiable link to a real-world entity [24]. Participants under weak identities might still be known within the network through pseudonyms or reputational metrics but lack the robust verification typical of strong identities. This anonymity can be seen as a double-edged sword; it provides privacy and protects the user's real-world identity but complicates trust and accountability, especially when users are either unable or have malicious intent not to pay back the loaned tokens.

Adopting weak identities is often driven by the desire for enhanced privacy and the ability to participate in a network without exposing one's real-world identity. This approach appeals to users who prioritize accessibility and privacy or wish to operate within a system free from censorship [26]. Networks often develop alternative mechanisms for establishing trust to accommodate the inherent risks associated with weak identities. In our case, we include a reputation system, where a user's historical transactions and interactions with other nodes in the network contribute to their trustworthiness score to secure transactions and mitigate the risk of default. The reputation score provides a sense of security for the users, as the higher a reputation is, the more trustworthy a node will be for paying back the loaned tokens and, thus, a successful transaction. On the other hand, the lower a node's reputation score, the more likely the node is for malicious intent.

The flexibility offered by weak identities encourages broader and more diverse participation in the network, as it lowers the barriers to entry for those unwilling or unable to undergo identity verification processes. However, managing the risks associated with weak identities requires careful approaches to ensure `Overdraft` remains resilient against fraud and abuse. `Overdraft` accomplishes this by introducing a dynamic and adaptive ecosystem where trust levels can adjust based on a participant's behavior and transaction history.

3.3. Game Theory Analysis

Since Overdraft will involve financial factors, it is essential to provide a game-theoretical model to focus more closely on the dynamics of trust, reputation, risk, and the ability to facilitate offline transactions through loans. This model will help to analyze how incentives can be structured to ensure Overdraft functions as intended, promoting cooperation while managing risk [27].

Incorporating the role of reputation into the game-theoretical model and the payoff matrix adds an essential layer of realism to the analysis. Reputation serves as a risk moderator, influencing decisions on whether or not to loan tokens to another node based on perceived trustworthiness. We will provide two perspectives of the incentives: the first perspective will be from an honest node and the risks they can take loans for nodes with low or high reputations. The second perspective will be from the attacker or dishonest node, which aims to exploit Overdraft.

3.3.1. Loaning Risk Based on Reputation Score

We can categorize nodes based on their reputation levels (high or low) and adjust the payoffs to reflect the increased risk of loaning for a node with a low reputation. We will provide a simple payoff matrix; for simplicity, we will assume two nodes, Node *A* and *B*, which include the outcomes based on the reputation of Node *B* (the loanee). Payoffs reflect the financial and reputational outcomes and the perceived risk level of the loanee's reputation.

Decisions:

- Node *A* loans tokens to a High Reputation node (*V-HR*): Node *A* decides to loan tokens to Node *B*, considering *B*'s high reputation as a sign of trustworthiness
- Node *A* loans tokens to a Low Reputation node (*L-HR*): Node *A* takes a riskier decision to loan tokens to Node *B*, whose low reputation indicates a higher risk of transaction failure
- Node *A* decides not to loan tokens to (*NV*): Node *A* decides not to loan any tokens, avoiding risk and any form of profit.

States:

- Node *B* has a high reputation (*H-ST*): Success transactions with high-reputation nodes yield low to moderate risk for both parties. In this state, Node *A* will receive their loaned amount, including the lower interest rate, back from Node *B*.
- Node *B* has a low reputation (*L-ST*): Successful transactions with low-reputation nodes yield moderate to high risk, especially for the loaner, reflecting the successful trust placed. In this state, Node *A* will receive their loaned amount, including the higher interest rate, back from Node *B*.
- Node *B* has a high reputation (*H-FT*): Failed transactions with high-reputation nodes result in lower risk, as the high reputation suggests a lower likelihood of failure. In this state, Node *A* will not receive their loaned amount back from Node *B*.
- Node *B* has a low reputation (*L-FT*): Failed transactions with low-reputation nodes result in the most risk, especially for the loaner. In this state, Node *A* will not receive their loaned amount back from Node *B*.

We use a scoring system from 0 to 3 to indicate the risk a node can receive. The 0 value indicates no risk, 1 is low risk, 2 is moderate risk, and 3 is the highest risk. This scoring system considers the immediate financial gains or losses and the change in reputation for succeeding and failing loan transactions.

	***B*: High Reputation (*H-ST*)**	***B*: Low Reputation (*L-ST*)**	***B*: High Reputation (*H-FT*)**	***B*: Low Reputation (*L-FT*)**
A*: *V-HR	(*A*: 1, *B*: 1)	Not Applicable	(*A*: 2, *B*: 2)	Not Applicable
A*: *V-LR	Not Applicable	(*A*: 2, *B*: 1)	Not Applicable	(*A*: 3, *B*: 2)
A*: *NV	(*A*: 0, *B*: 0)	(*A*: 0, *B*: 0)	(*A*: 0, *B*: 0)	(*A*: 0, *B*: 0)

Table 3.1: The following payoff matrix depicts the risk scores for different loaning scenarios for low or high reputation nodes with successful and failing transactions.

According to our payoff matrix in Table 3.1, we can now analyze the potential use and risk of `Overdraft`. The model highlights how reputation influences loaning decisions' perceived risk and potential reward. Nodes are more incentivized to loan tokens to high-reputation nodes due to the lower risk. However, `Overdraft` can offer higher rewards for loaning tokens to low-reputation nodes to compensate for the increased risk. Ultimately, it is up to the nodes to carefully assess the reputations of potential loanees and consider their risk tolerance. Loaning tokens to a low-reputation node might be worthwhile if the potential rewards are sufficiently high and the loaner has confidence in the loanee's ability to fulfill the transaction.

Finally, the matrix underscores reputation's role as a critical factor in `Overdraft`. An effective reputation management mechanism is essential to ensure accurate risk assessments and to provide a trustworthy environment.

3.3.2. Risk Based on Attacking

To construct a payoff matrix from the perspective of potential attackers within `Overdraft`, we must consider the strategies available to attackers and how `Overdraft`'s mechanisms, the reputation system, and the risk of detection affect the profitability of malicious actions. We aim to demonstrate that the expected payoffs for attacking `Overdraft` are unfavorable, thus deterring such behavior. We will have two critical factors in this matrix: the attacker (*A*), a node attempting to exploit `Overdraft`, potentially through Sybil attacks or failing to honor loaned transactions. On the other hand, we have `Overdraft` (*S*), which represents the mechanisms within our network to detect and penalize malicious behavior, including the reputation system.

Decisions:

- Node *A* attacks (*AT*): The attacker attempts to exploit `Overdraft` by not repaying the loaned amount or through other malicious actions such as smart contract attacks with Sybil nodes.
- Node *A* acts honestly (*AH*): The attacker chooses not to exploit `Overdraft` and acts as an honest node.

States:

- System *S* detects Node *A*'s attack (*D*): `Overdraft` successfully detects the attacker's malicious actions, leading to reputation penalties and possibly financial losses for the attacker.
- System *S* does not detect Node *A*'s attack (*ND*): `Overdraft` fails to detect the attacker's actions, potentially allowing the attacker to benefit from their exploit.

We use the same scoring system mentioned in the previous payoff matrix for low- and high-reputation loans. This scoring system considers the immediate financial gains or losses and the long-term impact on the attacker's ability to operate within `Overdraft` due to reputation changes.

	S: Detect (D)	**S: Not Detect (ND)**
A: Attack (AT)	(A: 3, S: 0)	(A: 1, S: 2)
A: Act Honestly (AH)	(A: 0, S: 0)	(A: 0, S: 0)

Table 3.2: The payoff matrix for the risk in different scenarios for an attacker, by either attacking `Overdraft` or acting honestly.

According to our payoff matrix in Table 3.2, we can now analyze the attacker's risk when attacking `Overdraft`. We can observe that the attacker has a significant risk when exercising their malicious actions and being detected. `Overdraft` has no risk when successfully detecting the attacker, which is attacking `Overdraft` using smart contract attacks by creating Sybil nodes. Creating Sybil nodes will result in a penalty of low reputations for these nodes. On the other hand, if the attacker tries to act honestly, `Overdraft` will not detect an attack, which results in neutral risk for both the attacker and `Overdraft`. However, suppose the attacker successfully attacks `Overdraft` without detection. In that case, it will result in a breach of trust and potential loss to honest nodes in the network, while the attacking node will only have a slight risk of being deterred from further network usage in a later stage. The honest nodes would open disputes, resulting in a decreased reputation for the attacking node, meaning other nodes will likely not transact with it.

The matrix highlights `Overdraft`'s critical role in detecting malicious actions. The high risk of being detected during these attacks makes such strategies unattractive to rational attackers, especially considering the long-term impact of reputational damage. `Overdraft` encourages all nodes, including potential attackers, to act honestly, ensuring that honest behavior consistently yields positive outcomes. The benefits of honest participation outweigh the risky gains from attacking, especially under the threat of detection. Finally, for attackers, the long-term sustainability of their presence in the network is crucial. In general, attacking `Overdraft` will risk immediate detection and penalty and their future in `Overdraft` due to reputational losses.

3.4. Exhaustive Search Algorithm

3.4.1. Average Maximum Transaction Amount

In this first iteration of our average maximum transaction amount algorithm, we started with an exhaustive search algorithm that traverses all possible paths in the network to determine the maximum transaction amounts. The maximum amount a user in the network can transact depends on two critical factors: whether the node can pay the amount for the transaction or not. The latter will require going through the network to look for the nodes that have loaned tokens to the transacting node. We consider a node's reputation, the amount they are loaning to a specific node, and the payment success probability of each node according to the guarantee they have from their predecessors.

The idea is to create an inverted tree in which the root node will be the node that initiated the transaction for a specific amount. The root node has edges going into it from its predecessors, the nodes loaning tokens to it. The predecessors can also have nested predecessors, creating an inverted tree network. However, we assumed that cycles would not be present in this inverted tree network.

For each node in the network, we have the following information:

- Let N_i denote node i in the network.
- Let $V(N_i)$ denote the loaned amount for node N_i.
- Let $R(N_i)$ denote the reputation for node N_i.

- Let $A(N_i)$ denote the amount collected by node N_i when node N_i is able to cover the transaction amount.
- Let $A(\neg(N_i))$ denote the expected amount collected node N_i can cover, given its inability to pay. It will use the loaned amount of its predecessors to provide an indicative amount that can be covered.
- Let $P(N_i)$ denote the predecessors of node N_i.

The value $A(N_i)$ can be calculated directly by the amount node N_i loans to another node:

$$A(N_i) = \begin{cases} V(N_i), & \text{if } P_{N_i} \neq \emptyset \\ 0, & \text{otherwise} \end{cases}$$

The calculation for $A(\neg(N_i))$ is more complicated as we need to traverse all predecessor nodes in the network and calculate the probability they can pay their loaned amount. This exhaustive algorithm should not be an issue for a small network; however, this will be the performance bottleneck for larger networks.

For each node N_i with predecessors $P_1, P_2, \ldots, P_k$, we calculate:

$$A(\neg(N_i)) = \sum_{i=1}^{2^k} \left(\prod_{j \in S_i} R_{P_j} \prod_{j \notin S_i} (1 - R_{P_j}) \right) \cdot \left(\sum_{j \in S_i} A(N_i)_{P_j} + \sum_{j \notin S_i} A(\neg(N_i))_{P_j} \right) \tag{3.1}$$

Here S_i represents the i-th combination of predecessors. The first product term calculates the probability of all possible i-th combinations. The more predecessors a node has, the more combinations it will require, making this algorithm a computationally intensive task. For example, if a particular node has two predecessor nodes, it has the following combinations:

- Predecessor node i is able to pay and predecessor node j is not able to pay
- Predecessor node i is not able to pay and predecessor node j is able to pay
- Predecessor node i is able to pay and predecessor node j is able to pay
- Predecessor node i is not able to pay and predecessor node j is not able to pay

The second term represents the total amount covered in the i-th combination. If a predecessor node has no other predecessor nodes and cannot pay the loaned amount, this amount will default to 0. Otherwise, it will recursively calculate the probability that its predecessors can pay their loaned amount.

Computational Complexity

The algorithm's complexity is driven by the number of predecessors each node has. For a node, N_i with k predecessors, calculating $A(\neg(N_i))$ requires evaluating all 2^k possible combinations of predecessors being able or unable to pay. Each combination involves evaluating the product of the reputations R_{P_j} and $(1 - R_{P_j})$ for the k predecessors, which is $O(k)$, and summing the amounts $A(N_i)_{P_j}$ and $A(\neg(N_i))_{P_j}$ over the subsets S_i, another $O(k)$ operation for each combination. Therefore, the complexity for each node is $O(k \cdot 2^k)$. When considering a network with n nodes, the total complexity becomes $O(n \cdot k \cdot 2^k)$. This exponential complexity concerning the number of predecessors makes the algorithm computationally intensive for nodes with many predecessors, particularly in larger networks.

3.4.2. Probability Distribution of Successful Transaction

Another closely related recursive algorithm we can use is the probability of a successful transaction in a network. We assume that there are no cycles present in the network. The algorithm is as follows:

- Let N_i denote node i in the network.
- Let $V(N_i)$ denote the loaned amount for node N_i.
- Let $R(N_i)$ denote the reputation for node N_i.
- Let T denote the transaction amount that a node N_i must pay to another node in the network.
- Let $S(N_i, T)$ denote the success probability of node N_i for a transaction amount T.
- Let $P(N_i)$ denote the predecessors of node N_i.

We require a base case for the leaf nodes when no more predecessors are present. The base case can be denoted as:

$$S(N_i, T) = \min\left(1, R(N_i) \cdot \frac{V(N_i)}{T}\right), \quad \text{where } P(N_i) = \emptyset \text{ (N_i has no predecessors).} \tag{3.2}$$

In this formula, we take the minimum of either 1, which means the probability of payment success will simply be the reputation of a node N_i, or we take the reputation of node N_i multiplied by the amount it loaned divided by the total transaction amount. This is done to directly check the probability of whether a node can pay the amount on its own.

Then, we require a recursive notation to calculate the probability of success for non-leaf nodes:

- Let C denote a combination of predecessor nodes from the root node contributing to the transaction
- Let K denote the valid combinations where the total loaned amount by successful nodes is at least T.
- Let $success(C)$ denote the set of nodes in combinations C that are successful.
- Let $fail(C)$ denote the set of nodes in combination C that fail.

A combination is considered valid if the sum of the loaned amounts from the predecessors marked in $success(C)$ in that combination is equal to or greater than the transaction amount T. This is done to ensure that only the combinations that have the potential to meet or even exceed the transaction requirement are considered in the calculation. We define the success probability of a combination C of predecessors as:

$$Q(C, T) = \left(\prod_{i \in \text{success}(C)} R(N_i) \cdot S(N_i, T)\right) \cdot \left(\prod_{j \in \text{fail}(C)} (1 - R(N_j) \cdot S(N_j, T))\right) \tag{3.3}$$

Here $Q(C,T)$ is a variable used in the final formula. This variable denotes the success probability for a specific combination C of predecessor nodes. The "success" term is the product of the reputations and individual success probabilities of the nodes in C that are successful ($success(C)$). The failure term is the product of one minus the reputation times the individual success probabilities of nodes in C that fail ($fail(C)$).

We can combine the formula above to write the formula for the overall success probability for a node N_i with predecessors:

$$
\begin{aligned}
S(N_i,T) = \min\Bigg(1, R(N_i)\cdot\frac{V(N_i)}{T} + \sum_{C\in K} Q(C,T) \\
-R(N_i)\cdot\frac{V(N_i)}{T}\cdot\sum_{C\in K} Q(C,T)\Bigg), \\
\text{where } (P(N_i)\neq\emptyset)\ (N_i \text{ has predecessors}). \qquad (3.4)
\end{aligned}
$$

Here, $S(N_i,T)$ utilizes the $Q(C,T)$ variables in its formula to calculate the overall success probability for node N_i. The overall success probability of a transaction is the sum of the direct probability, the contribution from its predecessors, and an adjustment for the overlapping probabilities. The direct probability is where node N_i completes the transaction based on its reputation and loaned amount. It assumes that node N_i can cover the transaction amount T based on its tokens. The contribution from its predecessors adds up the probabilities from all valid combinations of predecessors that can collectively support the transaction. Finally, we also adjust for the overlapping probabilities because the contribution from the predecessors is not independent, as these might also depend on their nested predecessors.

Computational Complexity

The algorithm's computational complexity depends on the number of combinations of predecessors it evaluates. For a node with k predecessors, there are 2^k possible combinations. Each combination requires evaluating the success and failure products, which takes $O(k)$ time. Thus, computing $Q(C,T)$ for all combinations involves $O(k\cdot 2^k)$ operations. As the algorithm makes a recursive call for each predecessor, the recursion depth, d, the longest path from a node to a leaf node, determines the number of recursive evaluations. Consequently, the overall time complexity of the algorithm is $O(2^k\cdot k\cdot d)$. This complexity reflects the exponential growth for the number of predecessors and the recursive depth.

3.5. Random Walk Computational Complexity

In our final maximum transaction amount algorithm, we employ a Monte Carlo simulation approach to efficiently approximate the distribution of the maximum transaction amount for a node's transaction without facing exponential computational complexity. This method involves performing a set number of random walks across the network, which allows for a practical approximation of transaction capacities. The number of iterations, denoted by I, significantly influences the reliability of the approximation; in this context, we use up to 100,000 iterations to ensure accuracy. Key parameters guiding our simulation are I, representing the number of iterations; N, indicating the number of nodes in the network; E, the count of edges; and D, which reflects the network's structure and its influence on the maximum depth of recursive calls during a walk.

The computational process is divided into two primary parts. First, this function traverses the network, potentially visiting nodes multiple times but each edge only once. Hence, the time complexity is $O(N+E)$, with space complexity primarily determined by the depth of recursion and tracked edges, capped at $O(N + D)$. On the other hand, by running the random walk function I times, the overall time complexity reaches $O(I \cdot (N + E))$. However, space complexity remains $O(N + D)$ due to the reuse of space across iterations.

This Monte Carlo method effectively reduces the problem from an unmanageable exponential complexity to a more manageable linear complexity concerning the number of iterations and the network's size and structure. This approach benefits large networks, providing accurate estimations with significantly reduced computational demand.

3.6. Countering Reputation Manipulation in Loan Agreements

In an attack scenario, a malicious user might distribute many Sybil nodes, each loaning a fraction of the total amount, with $\alpha \cdot x_i$ reputation, to increase the chances of being utilized as a loaning node. The aim is to demonstrate that the aggregate benefit, in terms of reduced risk, achieved by distributing reputation across multiple Sybil nodes is bounded by the benefit that could be obtained if the total reputation were concentrated in a single node.

We identify the risk function as a concave function. Concave functions are characterized by the property that the line segment between any two points on the function's graph lies below or on the graph [28]. In financial and various risk assessment contexts, concave functions often represent diminishing returns or increasing costs [29]. As you invest more resources, in this case, multiple nodes loaning fractions of the loaned amount, the incremental benefit decreases, or the incremental risk or cost increases at a decreasing rate. In the scenario of distributing the loaning amount across multiple nodes, a concave function implies that spreading out the loaning amount across several nodes results in a scenario where the combined assessed risk or benefit, from the perspective of the entity engaging in the Sybil attack, does not disproportionately improve.

For this reason, we conclude that the risk function will not be a convex function; convex functions often represent situations of increasing returns or costs that accelerate as investment (or, in this case, multiple nodes) increases. If the risk or benefit assessment function were convex, distributing the loaned amount across multiple nodes could lead to a situation where the combined risk or benefit is less than having a single node loan tokens to another node. This is because, under a convex function, smaller loaned amounts spread by multiple nodes are penalized less, which contradicts the desired property for assessing the risk of Sybil attacks.

We provide another proof for this agreement Sybil attack:

Proof. Given a loan agreement where nodes can loan for others based on their reputation, we hypothesize that an attacker distributes its loan amount across n Sybil nodes to reduce the risk assessment by `Overdraft`. Each Sybil node has an equal share of the attacker's total reputation, effectively making the reputation of each Sybil node $\alpha_i = \frac{\alpha}{n}$, where n is the number of Sybil nodes created by the attacker. The objective is to show that despite the distribution of the loaned amounts, `Overdraft`'s overall risk assessment, as determined by a concave function f, does not disproportionately benefit the attacker.

Jensen's Inequality provides a theorem for analyzing concave functions [30]. For a concave function f, and any real numbers $z_1, z_2, ..., z_n$ with corresponding non-negative weights $y_1, y_2, ..., y_n$ such that $\sum_{i=1}^{n} y_i = 1$, Jensen's Inequality asserts:

$$f\left(\sum_{i=1}^{n} y_i z_i\right) \geq \sum_{i=1}^{n} y_i f(z_i)$$

This inequality implies that for concave functions, the function value of the weighted average of inputs is greater than or equal to the weighted average of the function values of those inputs.

In analyzing a Sybil attack in `Overdraft`, we consider f as the risk or cost function applied by the loan agreement, which is concave concerning the reputation α. The loan amount is distributed equally across n Sybil nodes, meaning each node $\frac{1}{n}$ is loaning a fraction of the amount. Here, $z_i = \frac{\alpha}{n}$ for each i from 1 to n, and all weights $y_i = \frac{1}{n}$, reflecting the equal share of the loaned amount among the Sybil nodes. To derive the bound, we apply Jensen's Inequality to the distributed loaned amounts α_i with weights y_i. Given the equal distribution, each $y_i = \frac{1}{n}$, and the total reputation α distributed across n nodes, the inequality becomes:

$$f\left(\sum_{i=1}^{n} \frac{1}{n} \cdot \frac{\alpha}{n}\right) \geq \sum_{i=1}^{n} \frac{1}{n} f\left(\frac{\alpha}{n}\right)$$

Since $\sum_{i=1}^{n} \frac{1}{n} \cdot \frac{\alpha}{n} = n \cdot \frac{\alpha}{n^2} = \frac{\alpha}{n}$, and $\sum_{i=1}^{n} = n \cdot \frac{1}{n} = 1$, the inequality simplifies to:

$$f\left(\frac{\alpha}{n}\right) \geq \sum_{i=1}^{n} \frac{1}{n} f\left(\frac{\alpha}{n}\right)$$

However, this simplification needs to reflect Jensen's inequality correctly. The left-hand side of Jensen's Inequality should represent the function value when utilizing only one node, which in our scenario is $\frac{\alpha}{1}$ since we are considering only one node with reputation α against n distributed nodes summing to reputation α. Thus, the correct application is to show the relationship between the function value of one node and the function values from multiple Sybil nodes:

$$f(\alpha) \geq \sum_{i=1}^{n} \frac{1}{n} f\left(\frac{\alpha}{n}\right)$$

Since we know that $\sum_{i=1}^{n} \frac{1}{n} = n \cdot \frac{1}{n} = 1$ and x_i are substituted in the formula as fractions, we get our original bound:

$$f(\alpha) \geq \sum_{\substack{\alpha \cdot x_i \leq 1 \\ \sum x_i = 1}} f(\alpha \cdot x_i) \qquad \square$$

This inequality shows that the risk assessment for one node's reputation and loan amount is always greater than or equal to the aggregated assessments of distributed reputations and loaned amounts across Sybil nodes. The intuition behind this result is that the concavity of f ensures that reputation and more minor loaned amounts benefits are subject to diminishing returns; hence, splitting the loaned amount does not circumvent this property.

4

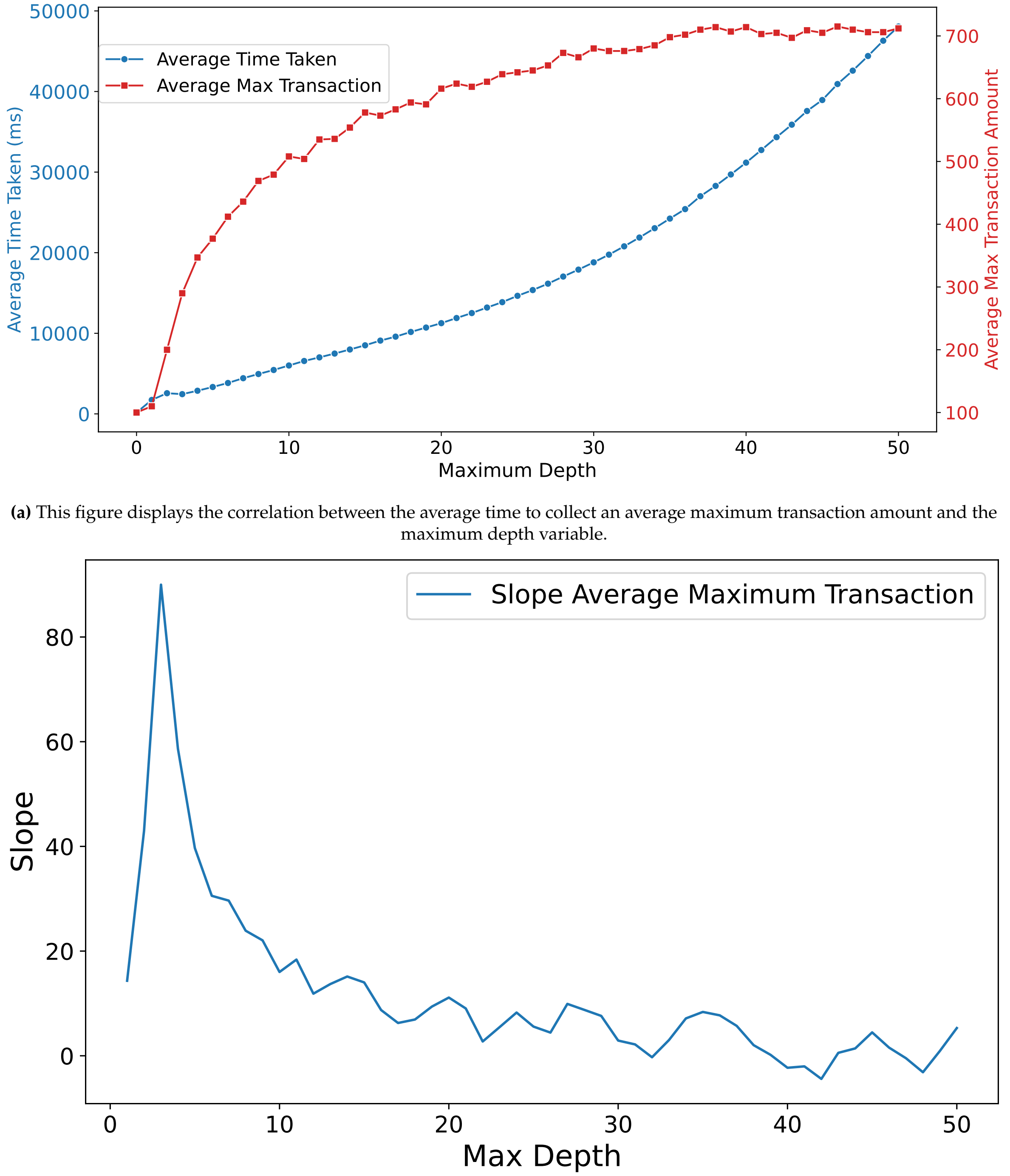
Additional Experiments

4.1. Analysis for Selecting A Maximum Depth

The performance of the optimized algorithm was evaluated by varying the maximum depth of traversal in the network, ranging from 0 to 50. Figure 4.1a displays the performance of our algorithm when choosing different maximum depth variables. It shows that deeper network traversal will require a long computing time and is not feasible for our system, where quick transaction times are required. This analysis shows an understanding of the trade-offs between the thoroughness of network exploration and the computational resources required.

The parameters used for this analysis were as follows: node count was set at 10,000, which was executed for 100,000 iterations; a constant number of maximum edges at 9 for each node; a maximum loan amount of 20; a root node with 0.2 reputation; an initial transaction amount of 100; a decay factor of 0.95 per hop from the root node; and the simulation was executed ten times for each maximum depth to get an average. The results, as depicted in Figure 4.1a, indicate a clear relationship between the maximum depth and the algorithm's performance, measured in terms of the average time taken (in milliseconds) for simulations and the average maximum transaction amount. As the maximum depth increases, there is a non-linear increase in the average time taken, which suggests a growing computational cost associated with deeper traversals in the network. This increase in time may be attributed to the additional calculations and memory overhead required to track and process a more significant number of nodes at greater depths.

Conversely, the average maximum transaction amount also increases with depth, though it exhibits a stabilizing effect as the depth approaches 20 hops. Figure 4.1b shows the slope of the average max transaction amount retrieved according to the chosen depth. It displays a conversion to around 0 after a depth of around 20 hops, which suggests that while deeper searches yield higher transaction amounts, the incremental benefit diminishes at a certain point, indicating a diminishing return on the additional computational expense.



(a) This figure displays the correlation between the average time to collect an average maximum transaction amount and the maximum depth variable.

(b) The slope of the average maximum transaction. It displays the convergence to 0 for depths larger than 20 hops away from the root node.

Figure 4.1: Comprehensive performance analysis based on transaction depth.

4.2. Maximum Throughput and Confirmation Latency

We conducted a series of tests on our smart contract to measure TPS and latency accurately. We implemented our smart contract in a local development environment using Hardhat [31] and created a series of nodes; half had a large balance, and the other had no balance. The nodes with a balance were loaned for those without a balance, and then transactions were conducted randomly. This allocation was made to accompany a variety of loaned and direct token transfers. We performed 1000 transactions between the node groups and measured the TPS and latency during execution for ten iterations. Afterward, we took the average of these iterations. The TPS was calculated based on the total number of transactions processed within a given time frame, and the latency was measured as the average time taken for transactions to be confirmed on the blockchain. We compared `Overdraft` with the TPS and latency other layer-1 (L1) blockchain-powered payment solutions such as Bitcoin [32] and Ethereum [33], and layer-2 (L2) solutions, such as Plasma [34] and Rollups [35, 36]. Our comparison results can be found in Table 4.1.

The local results of `Overdraft` being much higher than the testnet results indicate that the system's potential is best realized under controlled conditions. In a local environment, there are fewer network delays or network congestion and no contention for resources, allowing for optimal performance. Conversely, the Sepolia testnet introduces variables such as network latency, congestion, and resource competition, which are common in real-world deployments and thus impact the system's performance.

Moreover, any blockchain solution's TPS and confirmation latency are bounded by the underlying technology and infrastructure. While L1 solutions like Bitcoin and Ethereum are constrained by their consensus mechanisms and block times, L2 solutions like Arbitrum and zkSync offer higher throughput by processing transactions off-chain and then committing them to the main chain. Plasma achieves low latency by utilizing child chains, but even these advanced methods have limitations based on the specific implementation and operational environment. Since `Overdraft`'s implementation was done on Ethereum, it has the same confirmation latency as Ethereum.

System	TPS	Confirmation latency (s)
Bitcoin (L1) [37]	7	600
Ethereum (L1) [37]	15	12
Arbitrum (L2, Optimistic Rollup) [38]	102	780
zkSync (L2, ZK Rollup) [38]	165	600
Plasma (L2) [39]	175	2
`Overdraft` (run locally)	340	3
`Overdraft` (run on Sepolia testnet)	68	12

Table 4.1: Comparison of TPS and confirmation latency between L1 and L2 solutions and `Overdraft`

4.3. Cost-Effectiveness Against VISA's Interchange Fees

We compared the cost-effectiveness of `Overdraft` against one of the major global payment systems, VISA. Table 4.2 shows the cutoff point of `Overdraft`'s fees, when deployed on different blockchains with different fees[1], in comparison to VISA's interchange fee percentage of the average transaction value, for all transactions (exempt and covered) in 2022 [40].

These cutoff points show from which prices onwards `Overdraft` will be cheaper to use as a payment system than VISA for each L1 or L2 blockchain implementation. We observe that cutoff prices are low, which makes `Overdraft` more suitable for micropayments, especially when utilizing solutions with lower fees, such as Algorand or Polygon. However, using Ethereum results in prices that are too high to remain competitive due to the high fees and cost of one Ethereum, which increases the cutoff prices substantially.

VISA's interchange fee, at an average of 0.36 USD per transaction (0.76% of an average transaction value of 47.44 USD), is more cost-effective than transactions on the Ethereum blockchain. However, when comparing VISA's fee to Algorand and Polygon, we find that `Overdraft` becomes significantly more cost-effective for micropayments. For instance, Algorand's fee remains at a mere 0.02 USD even in less favorable conditions, making it an attractive option for low-value transactions. Similarly, Polygon's fees, which range from 0.33 USD to 2.63 USD depending on the GWEI used (GWEI is a unit of measurement for gas prices in the Ethereum network, representing one billionth of an ETH), remain competitive compared to VISA's interchange fees. Compared to conventional payment systems like VISA, `Overdraft` is efficient and cost-effective, especially for scenarios requiring offline transactions, as seen in Table 4.2, an area where traditional systems fall short. Despite VISA's higher throughput [41], `Overdraft`'s offline payment capabilities through loans present a compelling alternative.

Operation	**Ethereum (GWEI)**			**Algorand (ALGO)**	**Polygon (GWEI)**		
	20	**30**	**40**	**0.001**	**100**	**200**	**300**
Offline payment (good case)	$290.79	$435.53	$580.26	$0.02	$0.33	$0.67	$1.00
Offline payment (avg. bad case)	$776.32	$1163.16	$1551.32	$0.02	$0.89	$1.84	$2.63

Table 4.2: Cutoff price points in USD for transaction values where `Overdraft`'s offline payment operations are more cost-effective than VISA's average interchange fee. Note: VISA's fee refers to 0.76% as a percentage of average transaction value in 2022.

[1] Costs shown in USD at the time of writing April 16th, 2024: 3030 USD per ETH (Ethereum), 0.17 USD per ALGO (Algorand), and 0.69 USD per MATIC (Polygon)

5 Conclusion

This research introduced a novel framework for enabling reputation-weighted loans for offline payments. By leveraging the principles of network-based loans and integrating them with blockchain technology, we designed a system that allows users to conduct transactions without immediate internet connectivity, addressing the challenges of traditional online and offline payment methods. Our approach emphasizes the importance of trust and reputation within a decentralized network, providing a robust mechanism against the risk of double-spending and enhancing transaction security.

By implementing a smart contract and a sophisticated algorithm for managing loan agreements, we have shown that `Overdraft` can handle offline transactions efficiently while maintaining the integrity of the network. An incentive model ensures active participation and cooperation among nodes, fostering a healthy ecosystem. Our performance evaluation indicates that `Overdraft` is scalable and can adapt to varying-sized networks with minimal computational overhead.

Further research will continue to examine the privacy and contagion risks associated with `Overdraft`. We aim to develop advanced techniques that preserve privacy, allowing users to transact without risking their personal information. We will also address the contagion risks that arise when a few nodes' failure or malicious actions could affect the entire network. These initiatives are designed to enhance the system's defense against Sybil attacks and improve the precision of reputation metrics, ensuring the security and dependability of `Overdraft` as it evolves.

Future enhancements will focus on reducing shared fields, utilizing compression strategies, or possibly integrating `Overdraft` with alternative L1 or L2 solutions to lower expenses and boost performance. Additionally, we plan to explore using multiple smart contracts to manage different user segments (i.e., sharding), which could enhance transaction efficiency and reduce costs. However, this approach may require the synchronization of smart contracts due to the intricate nature of loan networks. Another enhancement proposed for future work is the incentive formula. By incorporating the loanee's node reputation into the incentive pricing, the system can ensure more reasonable incentives. This adjustment would account for the trustworthiness of individual nodes, thereby encouraging responsible behavior and enhancing the overall reliability of the network.

Overall, our work lays a solid foundation for the future development of offline payment systems, promising to bridge the gap between traditional financial mechanisms and the emerging digital economy.

Bibliography

[1] A. Ulrich, R. Holz, P. Hauck, and G. Carle, “Investigating the openpgp web of trust,” in *Computer Security - ESORICS 2011 - 16th European Symposium on Research in Computer Security, Leuven, Belgium, September 12-14, 2011. Proceedings*, V. Atluri and C. Dıaz, Eds., ser. Lecture Notes in Computer Science, vol. 6879, Springer, 2011, pp. 489–507. DOI: 10.1007/978-3-642-23822-2_27. [Online]. Available: https://doi.org/10.1007/978-3-642-23822-2%5C_27.

[2] S. L. Garfinkel, *PGP - pretty good privacy: encryption for everyone (2. ed.)* O’Reilly, 1995, ISBN: 978-1-56592-098-9.

[3] A. Datta, M. Hauswirth, and K. Aberer, “Beyond" web of trust": Enabling p2p e-commerce,” in *EEE International Conference on E-Commerce, 2003. CEC 2003.*, IEEE, 2003, pp. 303–312.

[4] B. Bahmani, A. Chowdhury, and A. Goel, “Fast incremental and personalized pagerank,” *Proc. VLDB Endow.*, vol. 4, no. 3, pp. 173–184, 2010. DOI: 10.14778/1929861.1929864. [Online]. Available: http://www.vldb.org/pvldb/vol4/p173-bahmani.pdf.

[5] M. Meulpolder, J. A. Pouwelse, D. H. J. Epema, and H. J. Sips, “Bartercast: A practical approach to prevent lazy freeriding in P2P networks,” in *23rd IEEE International Symposium on Parallel and Distributed Processing, IPDPS 2009, Rome, Italy, May 23-29, 2009*, IEEE, 2009, pp. 1–8. DOI: 10.1109/IPDPS.2009.5160954. [Online]. Available: https://doi.org/10.1109/IPDPS.2009.5160954.

[6] Y. Wang, Q. Zhang, and X. Yang, “Evolution of the chinese guarantee network under financial crisis and stimulus program,” *Nature Communications*, vol. 11, no. 1, p. 2693, 2020.

[7] E. Priyanto, E. L. Widarni, and S. Bawono, “The effect of internet inclusion on financial inclusion in p2p lending in indonesia based on human capital point of view,” in *Modeling Economic Growth in Contemporary Indonesia*, Emerald Publishing Limited, 2022, pp. 107–121.

[8] G. Karame, E. Androulaki, and S. Capkun, “Double-spending fast payments in bitcoin,” in *the ACM Conference on Computer and Communications Security, CCS’12, Raleigh, NC, USA, October 16-18, 2012*, T. Yu, G. Danezis, and V. D. Gligor, Eds., ACM, 2012, pp. 906–917. DOI: 10.1145/2382196.2382292. [Online]. Available: https://doi.org/10.1145/2382196.2382292.

[9] E. K. Clemons, D. C. Croson, and B. W. Weber, “Reengineering money: The mondex stored value card and beyond,” *Int. J. Electron. Commer.*, vol. 1, no. 2, pp. 5–31, 1996. DOI: 10.1080/10864415.1996.11518281. [Online]. Available: https://doi.org/10.1080/10864415.1996.11518281.

[10] J. Boly *et al.*, “The ESPRIT project CAFE - high security digital payment systems,” in *Computer Security - ESORICS 94, Third European Symposium on Research in Computer Security, Brighton, UK, November 7-9, 1994, Proceedings*, D. Gollmann, Ed., ser. Lecture Notes in Computer Science, vol. 875, Springer, 1994, pp. 217–230. DOI: 10.1007/3-540-58618-0_66. [Online]. Available: https://doi.org/10.1007/3-540-58618-0%5C_66.

[11] B. Nasrulin, G. Ishmaev, and J. Pouwelse, "Meritrank: Sybil tolerant reputation for merit-based tokenomics," in *4th Conference on Blockchain Research & Applications for Innovative Networks and Services, BRAINS 2022, Paris, France, September 27-30, 2022*, IEEE, 2022, pp. 95–102. DOI: 10.1109/BRAINS55737.2022.9908685. [Online]. Available: https://doi.org/10.1109/BRAINS55737.2022.9908685.

[12] M. Sabt, M. Achemlal, and A. Bouabdallah, "Trusted execution environment: What it is, and what it is not," in *2015 IEEE Trustcom/BigDataSE/Ispa*, IEEE, vol. 1, 2015, pp. 57–64.

[13] J. Lind, I. Eyal, F. Kelbert, O. Naor, P. R. Pietzuch, and E. G. Sirer, "Teechain: Scalable blockchain payments using trusted execution environments," *CoRR*, vol. abs/1707.05454, 2017. arXiv: 1707.05454. [Online]. Available: http://arxiv.org/abs/1707.05454.

[14] J. Poon and T. Dryja, "The bitcoin lightning network: Scalable off-chain instant payments," 2016.

[15] Ethereum, *Optimistic rollups*, Jan. 2024. [Online]. Available: https://ethereum.org/en/developers/docs/scaling/optimistic-rollups/.

[16] Ethereum, *Zero-knowledge rollups*, Mar. 2024. [Online]. Available: https://ethereum.org/en/developers/docs/scaling/zk-rollups/.

[17] *Lightning network technical summary*, https://lightning.network/lightning-network-technical-summary.pdf, (accessed 18-10-2023).

[18] A. Lisi, D. D. F. Maesa, P. Mori, and L. Ricci, "Lightnings over rose bouquets: An analysis of the topology of the bitcoin lightning network," in *IEEE 45th Annual Computers, Software, and Applications Conference, COMPSAC 2021, Madrid, Spain, July 12-16, 2021*, IEEE, 2021, pp. 324–331. DOI: 10.1109/COMPSAC51774.2021.00053. [Online]. Available: https://doi.org/10.1109/COMPSAC51774.2021.00053.

[19] R. Khalil and A. Gervais, "Revive: Rebalancing off-blockchain payment networks," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017, Dallas, TX, USA, October 30 - November 03, 2017*, B. Thuraisingham, D. Evans, T. Malkin, and D. Xu, Eds., ACM, 2017, pp. 439–453. DOI: 10.1145/3133956.3134033. [Online]. Available: https://doi.org/10.1145/3133956.3134033.

[20] A. Garg, S. Wadikhaye, and S. Maurya, "A research paper on study of mobile payment and it's security in india," *Electronic commerce research and applications*, vol. 14, pp. 265–284, 2015.

[21] R. Gochhwal, "Unified payment interface—an advancement in payment systems," *American Journal of Industrial and Business Management*, vol. 7, no. 10, pp. 1174–1191, 2017.

[22] Google, Nov. 2020. [Online]. Available: https://static.googleusercontent.com/media/pay.google.com/en//about/business/static/data/gpay_worldpay-whitepaper.pdf.

[23] S. Missonier and S. Loufrani-Fedida, "Stakeholder analysis and engagement in projects: From stakeholder relational perspective to stakeholder relational ontology," *International journal of project management*, vol. 32, no. 7, pp. 1108–1122, 2014.

[24] D. W. Arner, D. A. Zetzsche, R. P. Buckley, and J. N. Barberis, "The identity challenge in finance: From analogue identity to digitized identification to digital kyc utilities," *European business organization law review*, vol. 20, pp. 55–80, 2019.

[25] G. Caronni, "Walking the web of trust," in *9th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE 2000), 4-16 June 2000, Gaithersburg, MD, USA*, IEEE Computer Society, 2000, pp. 153–158. DOI: 10.1109/ENABL.2000.883720. [Online]. Available: https://doi.org/10.1109/ENABL.2000.883720.

[26] T. Ryberg and M. C. Larsen, "Networked identities: Understanding relationships between strong and weak ties in networked environments," *Journal of Computer Assisted Learning*, vol. 24, no. 2, pp. 103–115, 2008.

[27] L. A. Cox Jr, "Game theory and risk analysis," *Risk Analysis: An International Journal*, vol. 29, no. 8, pp. 1062–1068, 2009.

[28] S. Abramovich, G. Jameson, and G. Sinnamon, "Refining jensen's inequality," *Bulletin mathématique de la Société des Sciences Mathématiques de Roumanie*, pp. 3–14, 2004.

[29] H. U. Gerber and G. Pafum, "Utility functions: From risk theory to finance," *North American Actuarial Journal*, vol. 2, no. 3, pp. 74–91, 1998.

[30] S. P. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[31] Nomic Foundation, *Hardhat*, version 2.22.1, Mar. 14, 2024. [Online]. Available: https://hardhat.org.

[32] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," *Decentralized business review*, 2008.

[33] V. Buterin *et al.*, "A next-generation smart contract and decentralized application platform," *white paper*, vol. 3, no. 37, pp. 2–1, 2014.

[34] J. Poon and V. Buterin, "Plasma: Scalable autonomous smart contracts," *White paper*, pp. 1–47, 2017.

[35] H. A. Kalodner, S. Goldfeder, X. Chen, S. M. Weinberg, and E. W. Felten, "Arbitrum: Scalable, private smart contracts," in *27th USENIX Security Symposium, USENIX Security 2018, Baltimore, MD, USA, August 15-17, 2018*, W. Enck and A. P. Felt, Eds., USENIX Association, 2018, pp. 1353–1370. [Online]. Available: https://www.usenix.org/conference/usenixsecurity18/presentation/kalodner.

[36] Matter Labs, *Zksync*, Apr. 1, 2024. [Online]. Available: https://docs.zksync.io/build/quick-start/hello-world.html.

[37] H. Guo and X. Yu, "A survey on blockchain technology and its security," *Blockchain: research and applications*, vol. 3, no. 2, p. 100 067, 2022.

[38] R. Neiheiser, G. Inácio, L. Rech, C. Montez, M. Matos, and L. E. T. Rodrigues, "Practical limitations of ethereum's layer-2," *IEEE Access*, vol. 11, pp. 8651–8662, 2023. DOI: 10.1109/ACCESS.2023.3237897. [Online]. Available: https://doi.org/10.1109/ACCESS.2023.3237897.

[39] C. Sguanci, R. Spatafora, and A. M. Vergani, "Layer 2 blockchain scaling: A survey," *CoRR*, vol. abs/2107.10881, 2021. arXiv: 2107.10881. [Online]. Available: https://arxiv.org/abs/2107.10881.

[40] F. R. Board, *Regulation ii - average debit card interchange fee by payment card network*, Oct. 2023. [Online]. Available: https://www.federalreserve.gov/paymentsystems/regii-average-interchange-fee.htm.

[41] Visa, *Visa crypto thought leadership – a deep dive on solana*. [Online]. Available: https://usa.visa.com/solutions/crypto/deep-dive-on-solana.html#:~:text=As%20a%20global%20payments%20network,than%2065%2C000%20transactions%20per%20second..