Enhancing Blockchain Security Through Game Theory: A Comprehensive Approach

CSE4037- Reinforcement Learning

School of Computer Science and Engineering

By

21BCE8418	Sanjana Mitra
21BCE8629	G Divya

2023-2024

TABLE OF CONTENTS

Chapter No.	Title	Page No.
	Abstract	3
	List of Tables	4
	List of Figures	5
	List of Abbreviations	6
1	Introduction 1.1 Objective of the Project 1.2 Need of the Work 1.3 Scope and Motivation 1.4 Organization of the Report	7-11
2	Literature Survey 2.1 Basics 2.2 Preceding works	12-19
3	Existing Method and Disadvantages	20-22
4	Project Flow/Framework of the Proposed System	23-26
5	Hardware and Software Requirements 2pg	27-29
6	Proposed System 12pg 6.1 System Model 6.2 Functional Architecture 6.3 Slice Selection Algorithm 6.4 Evaluation	30-38
7	Conclusion and Future Work 3pg	39-41
	Modules	42-43
	References	44

Abstract

This paper explores the integration of game theory principles into blockchain technology to ensure safe and secure transactions and mining. The objective is to mitigate potential vulnerabilities and enhance the robustness of blockchain networks. The problem statement addresses the need for improved security mechanisms to counter emerging threats in decentralized systems. The scope encompasses various applications of game theory in blockchain security, motivated by the growing importance of safeguarding digital assets. The literature survey delves into existing research on game theory and blockchain security. Preceding work and drawbacks are analysed to identify gaps and opportunities for improvement. The tentative proposed method outlines a strategic framework leveraging game theory principles for enhanced security. The project flow and framework detail the implementation strategy for integrating game theory into blockchain systems. Hardware and software requirements are discussed to support the implementation process. Modules are elucidated, highlighting the key components of the proposed system. References are provided for further exploration of the topic.

Blockchain technology, with its decentralized and distributed nature, relies on the strategic interactions of various participants. Game theory emerges as a powerful tool for analysing these interactions, allowing us to understand the decision-making processes and potential outcomes within a blockchain network.

- **Modelling strategic behaviour:** Game theory frameworks model the interactions between different players in a blockchain network, such as miners, validators, and users. These models capture the strategic choices each player makes to maximize their individual utility, considering the actions of others.

- **Designing incentive mechanisms:** By understanding the motivations and rewards of different players, game theory aids in designing effective incentive mechanisms. These mechanisms aim to encourage desired behaviours, such as honest validation and participation, while discouraging malicious activities like attacks.

- **Analysing security and stability:** Game theory allows for the analysis of potential vulnerabilities and security risks in blockchain systems. By simulating various scenarios and predicting the behaviour of players under different conditions, developers can identify and address potential security issues.

- **Optimizing resource allocation:** Game theory models can be used to optimize the allocation of resources within a blockchain network. This includes analysing factors like mining power, stake distribution, and network fees to ensure efficient operation and fair distribution of rewards.

List Of Tables:

Table 1:

Feature	Network Slicing	NS Brokering
Function	Creates and manages logical networks	Selects and manages resources in existing networks
Focus	Resource allocation and isolation	Matching demand with optimal resources
Outcome	Customized network slices	Efficient resource utilization and cost optimization

Table 2:

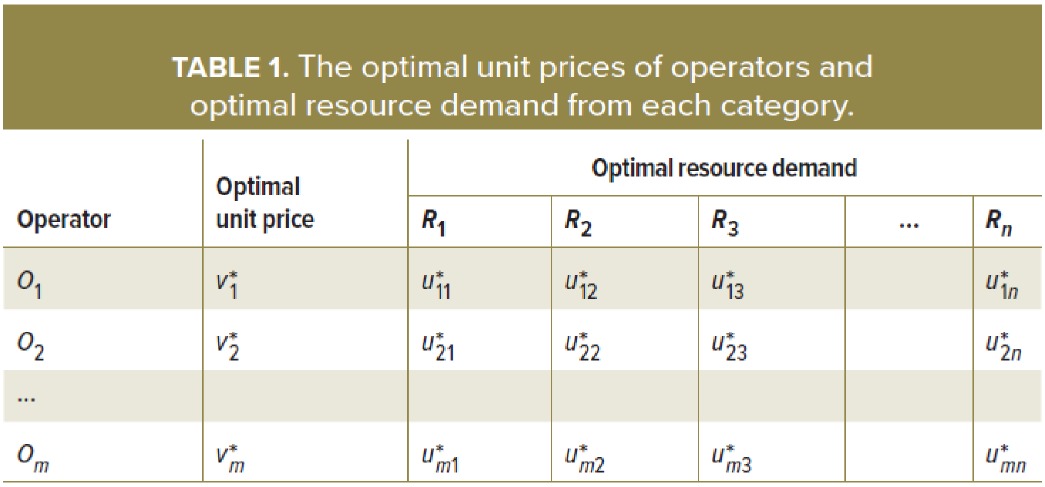


TABLE 1. The optimal unit prices of operators and optimal resource demand from each category.

Operator	Optimal unit price	Optimal resource demand				
		R_1	R_2	R_3	...	R_n
O_1	v_1^*	u_{11}^*	u_{12}^*	u_{13}^*		u_{1n}^*
O_2	v_2^*	u_{21}^*	u_{22}^*	u_{23}^*		u_{2n}^*
...						
O_m	v_m^*	u_{m1}^*	u_{m2}^*	u_{m3}^*		u_{mn}^*

List Of Figures:

Fig 1:

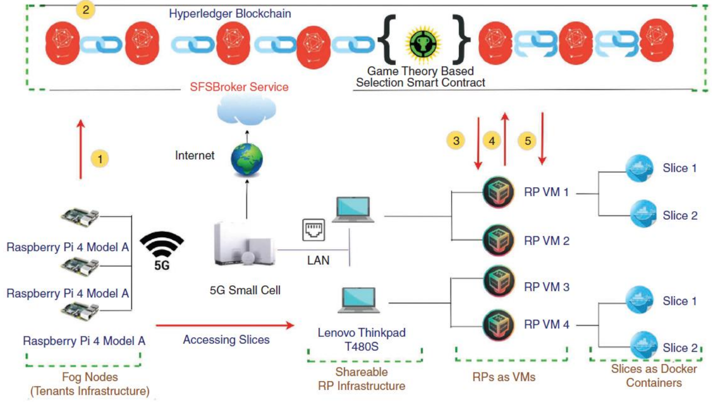


Fig 2:

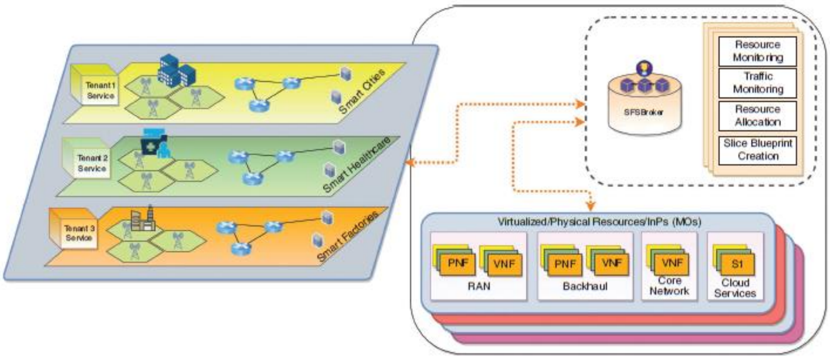
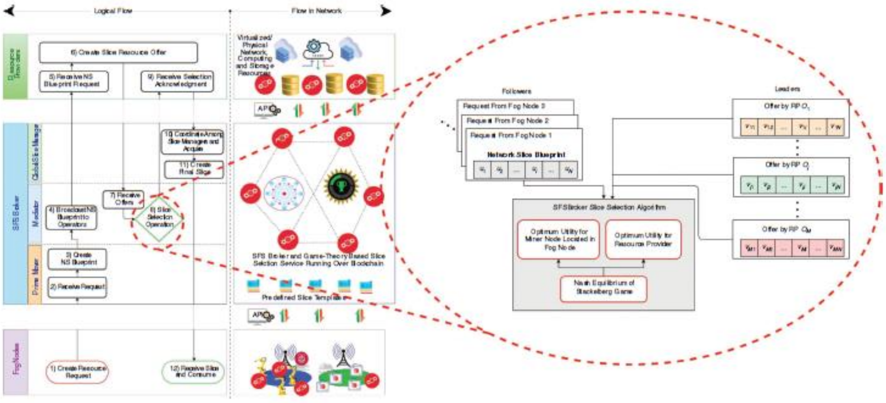


Fig 3:



List of Abbreviations:

NS - Network Security

SFSBroker - Secured and Federated Slice Broker

PoW - Proof of Work

PoS - Proof of Stake

ESS - Evolutionarily Stable Strategy

RP - Resource Provider

E2E- End to End

Introduction

Objective of the Project:

The primary aim of this project is to enhance the security of blockchain networks by incorporating game theory principles. By leveraging strategic decision-making models, we seek to address the inherent vulnerabilities in decentralized systems and ensure the integrity of transactions and mining processes.

Blockchain mining involves fierce competition among powerful computers to solve complex puzzles for new transaction blocks. The winner "mines" the block, adding it to the chain and earning cryptocurrency rewards. This process secures the network by making tampering computationally expensive and incentivizes participation through rewards, maintaining a robust and decentralized system.

Proof of work (PoW) and proof of stake (PoS) are two main methods for securing blockchain networks and verifying transactions. Here's a quick breakdown of the key differences:

Proof of Work (PoW): Imagine miners competing in a lottery.

- Miners use powerful computers to solve complex puzzles to validate transactions and add new blocks to the blockchain.
- The first miner to solve the puzzle wins the right to add the block and earn cryptocurrency rewards.
- This process consumes significant energy due to the heavy computations involved. (Think of all the lottery tickets being printed!)

Proof of Stake (PoS): Imagine securing a place in line based on how much money you have.

- Validators lock up some of their cryptocurrency holdings (their stake) to participate in transaction validation.
- The chance of a validator being chosen to add a new block is proportional to the size of their stake. (The more money you have, the closer you are to the front of the line).
- This method uses significantly less energy compared to PoW.

Both PoW and PoS incentivize honest behaviour by penalizing those who try to cheat. In PoW, miners waste computing power if they try to add invalid blocks. In PoS, validators lose part of their stake for dishonest actions.

The core of game theory revolves around analysing strategic decision-making in situations with multiple participants. Here's a breakdown of the basics:

- **Players:** These are the individuals or groups making decisions that affect each other.
- **Strategies:** Each player has a set of choices they can make.
- **Payoffs:** These are the outcomes or rewards each player receives based on the combination of strategies chosen by all players.

- **Equilibrium:** This is a situation where no player has an incentive to change their strategy because it's the best response to the strategies of others. A famous example is the Nash Equilibrium.

Imagine two friends deciding on a movie (a game):

- Players: You and your friend
- Strategies: You can choose comedy, horror, or drama (and vice versa for your friend)
- Payoffs: You enjoy comedies most, hate horror, and drama is okay. Your friend loves horror, dislikes comedies, and is fine with drama.
- Equilibrium: You might both pick drama because it avoids the strong dislikes of horror and comedies (assuming you value your friend's enjoyment too).

Game theory helps us understand complex interactions and predict potential outcomes by analysing these elements.

Game theory plays a crucial role in designing secure and efficient mining mechanisms for blockchain ledgers.

Understanding Miner Behaviour:

- Game theory helps model the strategic decision-making of miners in a blockchain network.
- Miners act strategically, trying to maximize their rewards (e.g., cryptocurrency) while minimizing costs (e.g., computing power).

Designing Incentive Mechanisms:

- By understanding miner motivations, game theory allows for the design of incentive mechanisms that encourage cooperation and honest participation.
- These mechanisms can penalize malicious behaviour like attempting to tamper with the ledger or launching attacks.

Maintaining Network Security:

- Game theory helps analyse potential security vulnerabilities arising from strategic manipulation by miners.
- By simulating different scenarios and predicting miner behaviour under attack conditions, developers can identify and address potential security weaknesses.

Here's a specific example:

Imagine a Proof of Work (PoW) blockchain where miners compete to solve complex puzzles.

- Game theory can help determine the optimal difficulty level for the puzzles.
- If the puzzles are too easy, miners might collude and manipulate the network.
- If they're too hard, honest mining becomes inefficient, discouraging participation.

Overall, game theory provides a framework to analyse miner interactions, design effective incentives, and ultimately create a secure and robust blockchain system for maintaining the ledger.

Problem Statement:

As blockchain technology becomes increasingly pervasive, it faces growing threats from malicious actors seeking to exploit vulnerabilities in the system. Traditional security measures may not suffice in the face of evolving threats, necessitating innovative approaches such as integrating game theory to bolster blockchain security.

As blockchain technology scales and holds ever-increasing value, the potential rewards for malicious actors also rise. Traditional security measures like cryptography are crucial, but they may not be enough to stay ahead of constantly evolving threats. This is where **game theory** emerges as a powerful tool for fortifying blockchain security.

1. Understanding the "Game":

- Game theory provides a framework to model the strategic interactions between different participants in a blockchain network, including miners, validators, and users.
- By analysing these interactions, we can understand the incentives that drive their behaviour, both honest and malicious.

2. Predicting and Mitigating Attacks:

- By simulating various scenarios and predicting how different actors might react in different situations, we can **identify potential vulnerabilities** in the system.
- This allows developers to anticipate and **design mechanisms to disincentivize malicious behaviour** and make attacks less profitable for attackers.

3. Designing Robust Incentive Systems:

- Game theory helps in designing **effective incentive mechanisms** that encourage **cooperative and honest behaviour** among participants.
- These mechanisms can be tailored to **punish malicious actors** who attempt to disrupt the network or steal funds, further discouraging such behaviour.

Examples of Game Theoretic Applications:

- **Proof of Stake (PoS) consensus mechanisms:** In PoS, the chance of validating a block is linked to the amount of cryptocurrency staked. This discourages malicious behaviour as losing their stake is a significant penalty.
- **Sybil attacks:** Game theory can help design systems that make it **costly for attackers to create fake identities** (Sybils) to manipulate the network.

By integrating game theory, blockchain security can evolve beyond reactive measures to become more **proactive and adaptable**. By understanding the "game" and anticipating the moves of malicious actors, we can design robust systems that are **resilient** against evolving threats, safeguarding the future of this transformative technology.

A ledger, also known as a **general ledger** or **distributed ledger**, is a **chronological record of all transactions** that have taken place on a blockchain network. It essentially acts as a **shared database** that is constantly growing and **cannot be altered or deleted**.

Here are some key characteristics of a ledger in blockchain:

- **Distributed:** Unlike traditional ledgers maintained by a single entity, a blockchain ledger is distributed across a network of computers. This means that every participant in the network has a copy of the ledger, making it **tamper-proof** and **resistant to censorship**.
- **Immutable:** Once a transaction is added to the ledger, it cannot be changed or deleted. This is ensured by the use of **cryptographic hashing** and **consensus mechanisms** like Proof of Work (PoW) or Proof of Stake (PoS).
- **Transparent:** All participants in the network can view the entire transaction history on the ledger, promoting **transparency and trust** within the system.
- **Secure:** Cryptography and consensus mechanisms ensure the **security and integrity** of the ledger, making it resistant to various cyberattacks.

The ledger plays a vital role in several functions of a blockchain network:

- **Recording Transactions:** It records all transactions that occur on the network, such as transfers of digital assets, execution of smart contracts, and other interactions.
- **Maintaining Consensus:** The ledger helps maintain consensus among all participants in the network about the current state of the system (e.g., who owns what digital assets).
- **Verification and Validation:** When a new transaction is submitted, participants can use the ledger to verify its validity and ensure it complies with the network's rules.

The ledger is maintained through cryptographic hashing, where each block references the hash of the previous block, creating an immutable chain. However, game theory plays a crucial role in **incentivizing honest behaviour** among participants who maintain the ledger, indirectly contributing to its security and accuracy. Here's how:

- **Modelling Miner/Validator Strategies:** Game theory helps model the strategic decision-making of miners or validators in a blockchain network. These actors are responsible for adding new blocks containing validated transactions to the ledger.
- **Encouraging Honest Block Validation:** By understanding their motivations (e.g., earning rewards), game theory allows for the design of incentive mechanisms that reward honest validation and discourage attempts to tamper with the ledger.
- **Penalizing Malicious Behaviour:** These mechanisms can involve penalties for adding invalid blocks or attempting to manipulate the ledger history. Game theory helps design these penalties to be strategically significant, ensuring the cost of cheating outweighs any potential gains.

Imagine miners competing to add new blocks to the ledger (like a race).

- Game theory helps set the "rules of the race" (incentive mechanisms).
- Honest miners who follow the rules and validate transactions fairly are rewarded (win the race).
- Cheating miners who try to add invalid blocks are penalized (disqualified from the race).

Examples of Game Theory in Ledger Maintenance:

- **Proof of Work (PoW) Difficulty Adjustment:** Game theory helps determine the optimal difficulty level for the puzzles miners need to solve to add blocks. This discourages miners from colluding and manipulating the ledger.
- **Proof of Stake (PoS) Stake Slashing:** In PoS, validators lose part of their stake for dishonest actions. Game theory helps design the penalty structure to make cheating a strategically unattractive option.

Overall, game theory doesn't directly maintain the ledger, but it helps ensure its integrity by promoting honest behaviour among those responsible for adding new blocks. This, in turn, contributes to a secure and accurate ledger.

Scope and Motivation:

The scope of this project encompasses the application of game theory principles to various aspects of blockchain security, including consensus mechanisms, transaction validation, and incentive structures for miners. Motivated by the critical need to safeguard digital assets and maintain trust in decentralized systems, this research aims to contribute to the ongoing efforts to fortify blockchain networks against attacks and manipulation.

Literature Survey:

The literature survey explores existing research on the intersection of game theory and blockchain security. It delves into seminal works that have laid the foundation for applying game-theoretic concepts to decentralized systems. Key topics include strategic decision-making, Nash equilibrium, and mechanism design in the context of blockchain consensus and incentive mechanisms. Additionally, recent advancements and emerging trends in the field are examined to identify opportunities for further exploration and innovation.

Game theory has emerged as a powerful analytical framework for understanding the strategic interactions and incentives at play in blockchain networks. As these decentralized systems rely on distributed consensus among untrusted nodes, it is crucial to model how rational participants will behave and respond to different situations and rules. This comprehensive survey reviews the extensive applications of game theory across various aspects of blockchain design and management.

At its core, blockchain technology provides a transparent, immutable and decentralized public ledger maintained through consensus protocols like proof-of-work (PoW) or proof-of-stake (PoS). However, ensuring nodes follow the prescribed rules requires careful incentive engineering and understanding potential vulnerabilities. Game theory models have been widely applied to analyse security threats such as selfish mining attacks, where miners withhold blocks for higher rewards. Non-cooperative games elucidate the conditions under which such attacks are profitable versus following honest mining. Stochastic games capture more dynamic scenarios where miners can adapt strategies over time based on the evolving blockchain state. Similar models investigate the risk and incentives around majority attacks, where coalitions of miners attempt to gain control over the network.

Beyond security issues, game theory provides insights into optimal mining strategy management by individual miners and pools. This includes determining computational power allocation levels using non-cooperative games that maximize miners' utilities based on factors like electricity costs. Stochastic games model miners' dynamic entry and exit decisions based on fluctuating mining populations over time. Other models examine strategies around fork chain selection when the blockchain diverges. For mining pools, games analyse reward allocation mechanisms, pool fee setting, and block withholding attacks between pools.

The rise of blockchain technology has introduced a fascinating economic landscape, where miners play a crucial role in securing the network and earning rewards. However, navigating this landscape requires strategic thinking, as individual miners and mining pools constantly seek the optimal approach to maximize their profits. This delves into the strategies employed by both individual miners and mining pools to ensure their continued success in the ever-evolving world of blockchain mining.

For individual miners, the primary decision revolves around **hardware selection and resource allocation**. The computational power of their mining rigs directly impacts their hash rate, which is the number of hashes they can generate per second. Miners must carefully consider the cost of acquiring and maintaining powerful hardware against the potential rewards from successfully mining a block. Careful calculations are needed to ensure their operations remain profitable, especially as the difficulty of mining often increases over time.

Another critical factor for individual miners is **pool selection**. Joining a mining pool allows them to combine their hash rate with others, increasing their collective chance of discovering a block and sharing the rewards. However, pool fees and payout structures can vary significantly. Understanding these variations is crucial for miners to choose a pool that aligns with their risk tolerance and profitability goals. Some pools offer a **pay-per-share (PPS)** model, where miners receive a fixed reward for each valid share they submit, regardless of whether the pool finds a block. Other pools utilize a **pay-per-last-N-share (PPLNS)** model, where rewards are distributed based on the miner's contribution to the pool's most recent block discoveries.

For mining pools, the focus shifts towards **optimizing pool operations and attracting miners**. Pool operators strive to maintain a **healthy hash rate** by offering competitive payout structures and ensuring efficient operations. They may also employ **hash rate rental services** to dynamically adjust their pool's overall hash rate based on market conditions.

Game theory plays a vital role in both individual and pool-level strategies. By understanding the motivations and actions of other miners and pools, individuals can make informed decisions about resource allocation and pool selection. Similarly, pool operators leverage game theory to design incentive structures that encourage miners to join and remain active within their pool.

The optimal mining strategy is a dynamic equation that constantly adapts to factors like network difficulty, cryptocurrency price fluctuations, and the emergence of new mining technologies. Individual miners must continuously evaluate their hardware, pool selection, and overall operational efficiency. Mining pools, on the other hand, focus on attracting and retaining miners by offering lucrative payout structures and efficient operations. Through strategic decision-making and a keen understanding of the ever-shifting mining landscape, both individual miners and mining pools can ensure their continued success in securing the blockchain and reaping the rewards of this innovative technology.

Looking beyond core blockchain operations, game theory is increasingly applied to reason about incentives and behaviours in applications built atop blockchain platforms. This includes energy trading markets secured by blockchain smart contracts, as well as mobile edge computing scenarios leveraging blockchain's decentralized framework. As blockchain technology expands into new domains, game theoretical models will be essential for aligning incentives and developing robust decentralized systems and applications.

Traditional energy trading often involves a central authority and intermediaries, leading to slow, opaque transactions with high costs. Smart contracts offer a solution by facilitating **automated, transparent, and decentralized energy trading** on the blockchain.

Imagine a home with solar panels (a prosumer) has excess electricity. A smart contract, pre-programmed with the selling price, is deployed on the blockchain. This contract automatically matches the prosumer with nearby buyers and facilitates the real-time transfer of energy and payment.

Smart contract-based energy trading boasts several advantages:

- **Efficiency:** Automates transactions, reducing reliance on intermediaries.
- **Transparency:** Transactions are recorded immutably on the blockchain, ensuring transparency.

- **Decentralization:** Eliminates the need for a central authority, empowering prosumers to participate directly.
- **Cost Reduction:** Lower transaction fees and potentially more competitive pricing.
- **Sustainability:** Incentivizes renewable energy generation and consumption through peer-to-peer trading.

However, challenges remain:

- **Scalability:** Blockchain technology needs to scale to handle a high volume of transactions.
- **Regulation:** Regulatory frameworks need to adapt to this new approach.
- **Grid Integration:** Integrating smart contracts with existing infrastructure requires further development.

Despite the challenges, smart contracts hold immense potential to revolutionize energy trading, paving the way for a more efficient, transparent, and sustainable energy future.

In summary, this comprehensively demonstrates the power of game theory for modelling strategic decision-making in blockchain ecosystems. From mitigating attacks to optimizing mining strategies to emerging decentralized applications, game theoretical analysis provides crucial insights into understanding incentive structures, potential vulnerabilities, and pathways to reinforce intended system behaviours. As blockchain technology continues to evolve, game theory will remain an indispensable tool for fortifying security, promoting cooperation, and realizing blockchain's transformative potential across industries.

Blockchain ecosystems are complex environments where various participants, like miners, validators, and users, interact strategically. Predicting their behaviour and designing robust systems requires a powerful tool: **game theory**.

Game theory allows us to model these strategic interactions. It treats the ecosystem as a "game" with defined players, actions, and potential outcomes (payoffs). By analysing these elements, we can understand the motivations of each participant. For example, miners strive to maximize their block rewards, while validators aim to secure the network and earn their designated fees.

This understanding allows us to predict how participants might react under various circumstances. Imagine a scenario where the difficulty of mining increases. Game theory helps us predict how miners might respond by upgrading their hardware or joining mining pools to stay competitive. This knowledge is crucial for developers to design secure and efficient blockchain systems.

Furthermore, game theory empowers the creation of incentive mechanisms that promote desired behaviours. By understanding the "game" and the players' motivations, we can design rewards and penalties that encourage honest participation and discourage malicious activities. For instance, a blockchain might adjust mining difficulty automatically based on the total hash rate, ensuring a balance between security and efficiency.

In essence, game theory acts as a guiding light for navigating the strategic complexities of blockchain ecosystems. It facilitates the prediction of participant behaviour, the design of robust systems, and the creation of effective incentives for a secure and thriving blockchain environment.

Reviewing different reward mechanisms used by prominent PoS blockchains like Algorand, Avalanche, Cardano, Cosmos, Ethereum 2.0, and Polkadot. It notes that these ad-hoc designs differ considerably in terms of who gets rewarded, whether rewards are proportional to stake, and whether penalties are imposed for malicious behaviour.

Proof-of-Stake (PoS) blockchains have emerged as a secure and energy-efficient alternative to Proof-of-Work (PoW) for securing blockchain networks. However, PoS blockchains differ in their specific implementation details, particularly regarding their **reward mechanisms**. Here's a breakdown of the reward structures employed by some prominent PoS blockchains:

1. Cardano (ADA):

- **Delegated Proof-of-Stake (DPoS):** Cardano utilizes a DPoS system where users delegate their stake-to-stake pools. Stake pools then compete for the right to validate blocks, with rewards distributed proportionally to the amount of ADA staked in each pool.
- **Focus:** Decentralization and community involvement are key focuses. Users have more control over their stake by choosing a pool that aligns with their values.

2. Polkadot (DOT):

- **Nominated Proof-of-Stake (NPoS):** Polkadot uses a NPoS system where users nominate validators. Validators with the most nominations have a higher chance of being selected to validate blocks and earn rewards.
- **Focus:** Scalability and interoperability are emphasized. NPoS allows for a more efficient selection process for validators while maintaining decentralization.

3. Cosmos (ATOM):

- **Tendermint BFT:** Cosmos utilizes Tendermint BFT, a Byzantine Fault Tolerance (BFT) consensus mechanism. Validators are chosen based on their stake and good behaviour. Rewards are distributed proportionally to the validator's stake.
- **Focus:** Interoperability and modularity are central to Cosmos. Tendermint BFT allows for a high degree of scalability and security.

4. Algorand (ALGO):

- **Pure Proof-of-Stake (PPoS):** Algorand employs a unique PPoS system where all token holders participate in block selection. A random selection process chooses a committee of users to vote on the next block. Rewards are distributed to committee participants and those who help them reach consensus.
- **Focus:** Scalability and participation are prioritized. PPoS allows for very fast transaction processing and encourages broad participation in the consensus process.

5. Ethereum 2.0 (ETH):

- **Validator Selection based on Stake:** Similar to Cosmos, validators in Ethereum 2.0 are chosen based on their stake. Rewards are distributed proportionally to the validator's stake.
- **Focus:** Security and future scalability are key considerations. Ethereum 2.0 leverages a well-established developer community and a large user base to ensure network security.

6. Avalanche (AVAX):

- **Avalanche Consensus Protocol (ACP):** Avalanche employs a complex protocol involving multiple blockchains working together. Validators are chosen based on their stake across these chains. Rewards are distributed proportionally to the validator's stake.
- **Focus:** Scalability and high transaction throughput are prioritized. ACP allows for very fast transaction processing and high network capacity.

Key Considerations:

- **Decentralization vs. Efficiency:** Some reward mechanisms prioritize decentralization, encouraging wider participation (Cardano, Polkadot). Others focus on efficiency, allowing for faster block selection (Algorand, Avalanche).
- **Security:** All the mentioned blockchains ensure security through stake slashing, where validators lose part of their stake for malicious behaviour.

The choice of reward mechanism significantly impacts a PoS blockchain's characteristics. Understanding these differences allows developers to design blockchains tailored to specific needs, fostering innovation and growth within the blockchain ecosystem.

To better understand reward mechanism design, the model block validation in a PoS blockchain as a game between rational, self-interested validators who can act honestly (approve valid blocks) or maliciously (approve invalid blocks).

They first consider a universal reward case where all validators get rewarded regardless of their behaviour. Game theory analysis shows that the rational strategy here is to act maliciously to maximize rewards, leading to cartels and potential network insecurity.

To address the free-rider problem, they update the rewards to only incentivize validators who actually participate and vote. However, this still leads to validators approving all blocks indiscriminately to get rewarded.

To tackle the nothing-at-stake issue, the authors introduce penalties for validators who deviate from the honest majority. Drawing from prospect theory on loss aversion, they argue penalties are more effective than incentives in shaping human behaviour.

Using evolutionary game theory, they analyse whether the honest strategy emerges as an evolutionarily stable strategy (ESS) that cannot be invaded by mutant malicious strategies over time. They prove that with sufficient penalties, honest behaviour becomes an ESS as long as the initial population has an honest majority during the network's genesis.

In Proof-of-Stake (PoS) blockchains, security relies on the assumption that most participants will act honestly.

Why Honesty Can Be an ESS:

- **Stake Slashing:** Most PoS systems penalize malicious behaviour like forging blocks or disrupting the network by slashing the validator's stake. This creates a significant disincentive for dishonesty, as losing stake translates to losing potential rewards and jeopardizing the validator's reputation.
- **Long-Term Gains:** Dishonest behaviour can lead to short-term benefits, but it also risks undermining the entire network's security and value. If the network becomes unreliable, the value of the cryptocurrency (and the validator's stake) plummets, hurting everyone. A secure and healthy network, on the other hand, fosters long-term growth and profitability for honest validators.
- **Nash Equilibrium:** Game theory's concept of Nash Equilibrium suggests that if everyone is acting honestly, then no individual validator has an incentive to deviate. A rational validator would not risk their stake for potentially meager gains when the system is functioning well.

Challenges to Honesty as an ESS:

- **Short-Term Greed:** The potential for immediate gains from malicious activity might be tempting for some validators, especially if they believe they can get away with it.
- **Collusion:** A group of malicious validators could potentially collude to manipulate the network for their benefit. This can be a significant threat if the stake distribution is too concentrated among a few entities.
- **New Attack Vectors:** As blockchain technology evolves, new vulnerabilities and attack vectors might emerge. If these exploit weaknesses in the PoS protocol, they could disrupt the stability of honest behaviour as an ESS.

How PoS Systems Promote Honesty:

- **Stake Size Requirements:** Many PoS systems require a minimum stake amount to participate in validation. This discourages individuals from using small amounts of stake for malicious purposes, as the potential rewards wouldn't justify the risk of losing the entire stake.
- **Reputation Systems:** Some blockchains incorporate reputation systems that track validator behaviour. A history of malicious actions could damage a validator's reputation and make it difficult for them to participate in future consensus rounds.
- **Ongoing Protocol Development:** Blockchain developers constantly work on improving PoS protocols to address emerging security threats and strengthen the incentives for honest behaviour.

While there are challenges, PoS blockchains incorporate various mechanisms to make honesty an ESS. The combination of stake slashing, long-term benefits from a secure network, and ongoing protocol development creates a strong environment where honest participation is the most rational strategy for validators. However, continuous vigilance and adaptation are crucial to ensure that PoS systems remain secure and promote honest behaviour in the long run.

Through simulations, they confirm that high penalties quickly lead the entire network to converge to the honest ESS, while low/no penalties allow an initial malicious minority to eventually take over. Their findings highlight the crucial role of penalties in maintaining blockchain integrity.

In summary, the key contributions are formulating the validator game, designing reward matrices to incentivize honest behaviour, using evolutionary game theory to analyse strategy dynamics, and establishing the importance of penalties for sustainable PoS blockchains. **Designing sustainable Proof-of-Stake (PoS) blockchains** hinges on **incentivizing honest behaviour** among validators. This delves into three key elements: **reward matrices, evolutionary game theory, and penalties**, exploring their roles in achieving this goal.

1. Reward Matrices:

- These matrices define the **payoffs** (rewards or penalties) received by validators based on their **actions** (honest vs. malicious) and the **actions of others** (cooperative vs. non-cooperative).
- Designing effective reward matrices involves several considerations:
  - **Rewarding honest participation:** This can be achieved through fixed rewards for validating blocks or transaction fees proportional to stake size.
  - **Penalizing malicious behaviour:** Mechanisms like **stake slashing** (partial or complete loss of stake) discourage dishonesty. The severity of the penalty can be determined based on the severity of the offense.
  - **Encouraging long-term commitment:** Reward structures that favour validators with longer lock-up periods for their stake can incentivize long-term commitment to the network's health.

2. Evolutionary Game Theory:

- This branch of game theory analyses **strategic behaviour** in **populations** that evolve over time based on their **fitness** (success in the game).
- Applying evolutionary game theory to PoS helps us understand how **strategies spread and persist** within the validator population.
- By simulating scenarios with different reward matrices, we can **predict** whether honest behaviour will emerge as an **evolutionarily stable strategy (ESS)**:
  - If honest validators consistently earn higher rewards (or suffer less penalties) than malicious ones, honesty becomes the dominant strategy in the long run.
  - Conversely, if malicious behaviour offers even temporary advantages, it can disrupt the system and undermine its sustainability.

3. Importance of Penalties:

- Even with well-designed reward matrices, **penalties are crucial** for deterring malicious behaviour.
- Effective penalties should be:
    - **Significant enough** to outweigh any potential gains from cheating.
    - **Fairly and transparently applied** to maintain trust within the network.
    - **Adaptable** to address emerging threats and exploit attempts.

Combining these elements:

By strategically designing reward matrices, applying evolutionary game theory for analysis, and implementing effective penalties, blockchain developers can foster an environment where honest behaviour becomes the **dominant and sustainable** strategy for validators in a PoS system. This ultimately leads to a **secure, reliable, and thriving blockchain ecosystem**.

Additional Considerations:

- **Stake distribution:** A **concentrated stake** can increase the risk of collusion and manipulation. Mechanisms to encourage wider participation and a more **even stake distribution** can mitigate this risk.
- **Reputation systems:** Integrating reputation systems can further incentivize honest behaviour by allowing validators to build trust and gain advantages based on their positive track record.
- **Ongoing monitoring and adaptation:** Continuously monitoring the network for new vulnerabilities and adapting the reward structure and penalties based on observed behaviour are crucial for maintaining the long-term sustainability of a PoS blockchain.

Developing a sustainable PoS blockchain requires a **multifaceted approach** that considers reward structures, game theory analysis, and effective penalties. By carefully implementing these elements, blockchain developers can create a secure and robust system that incentivizes honest participation and fosters the long-term growth of the network.

Existing Methos and Disadvantages:

This section critically evaluates preceding efforts to enhance blockchain security using game theory approaches. While existing research has demonstrated the potential of game-theoretic models to address security challenges, certain drawbacks and limitations persist. These include scalability issues, complexity in implementation, and the need for robust incentive structures to align the interests of participants. By analysing the strengths and weaknesses of prior approaches, this study aims to propose a comprehensive framework that overcomes existing limitations and enhances the resilience of blockchain networks.

Blockchain technology holds immense promise for revolutionizing various sectors, but it faces several challenges that hinder its widespread adoption. This explores three critical hurdles: **scalability issues, implementation complexity, and the need for robust incentive structures**.

1. Scalability Issues:

- At the heart of scalability challenges lies the fundamental design trade-off in blockchains: **decentralization, security, and scalability**. Traditional blockchains prioritize security and decentralization, but this often comes at the cost of slow transaction processing times and limited throughput.
- This is particularly evident in **Proof-of-Work (PoW)** blockchains like Bitcoin, where transaction verification requires significant computational power, leading to slow processing times. This bottleneck hinders the ability to handle the growing demand for blockchain applications.

2. Complexity in Implementation:

- Developing and deploying blockchain applications can be a complex and resource-intensive process.
- The underlying technology involves cryptography, distributed computing, and complex consensus mechanisms, requiring a deep understanding of these fields.
- This complexity poses a barrier for mainstream adoption, as it discourages developers and businesses from venturing into the blockchain space due to the significant learning curve and development costs.

3. Aligning Interests with Robust Incentive Structures:

- Ensuring long-term sustainability requires well-designed incentive structures that **align the interests of different participants** in the blockchain ecosystem.
- In PoW blockchains, miners are incentivized to validate transactions through block rewards. However, this mechanism can lead to issues like energy inefficiency and potential centralization among large mining pools.
- In **Proof-of-Stake (PoS)** blockchains, the focus shifts to staking, where participants lock up their cryptocurrency holdings to participate in the consensus process. While PoS offers advantages like improved scalability and energy efficiency, designing

effective reward structures in PoS systems remains a challenge. These structures need to incentivize honest participation, discourage malicious behaviour, and ensure the long-term security and stability of the network.

Finding the Equilibrium:

Addressing these challenges is crucial for achieving widespread blockchain adoption. Developers and researchers are actively exploring various **scaling solutions** like layer-2 scaling, sharding, and alternative consensus mechanisms to **improve transaction processing speed and throughput**. Additionally, **simplifying the development process** through user-friendly tools and standardized frameworks can attract more developers and businesses to the blockchain space.

The importance of robust incentive structures cannot be overstated. By carefully balancing rewards and penalties, developers can encourage desired behaviours, promote long-term network security, and create a **sustainable ecosystem** where all participants are incentivized to contribute to the network's growth and success.

While challenges persist, the potential benefits of blockchain technology are undeniable. Through continuous research, development, and collaboration, we can overcome these hurdles and unlock the full potential of blockchain technology, paving the way for a more secure, transparent, and efficient future.

Blockchain technology has emerged as a transformative force across various industries, but it still faces limitations that hinder its widespread adoption and long-term sustainability. This essay proposes a **comprehensive framework** that addresses these limitations and enhances the **resilience** of blockchain networks.

The Framework:

1. Scalability Solutions:

- **Layer-2 scaling:** Implement solutions like sidechains and state channels to offload transaction processing from the main chain, improving throughput without compromising decentralization.
- **Sharding:** Divide the blockchain into smaller partitions (shards), allowing parallel processing of transactions and significantly scaling transaction capacity.
- **Directed Acyclic Graphs (DAGs):** Explore alternative consensus mechanisms like DAGs, which offer faster transaction processing and potentially higher scalability compared to traditional blockchains.

2. Enhanced Security Measures:

- **Formal verification:** Employ formal verification techniques to mathematically prove the correctness and security of smart contracts, mitigating the risk of vulnerabilities and exploits.
- **Post-quantum cryptography:** Implement post-quantum cryptography algorithms to prepare for the potential threat of quantum computers breaking current cryptographic standards.

- **Byzantine Fault Tolerance (BFT) consensus mechanisms:** Utilize BFT-based consensus mechanisms to tolerate Byzantine failures, ensuring the network remains functional even in the presence of malicious actors.

3. Robust Incentive Structures:

- **Dynamic reward adjustments:** Design mechanisms that automatically adjust rewards based on network activity and participation levels, ensuring fair distribution and incentivizing continuous network growth.
- **Stake slashing with slashing delays:** Implement stake slashing penalties for malicious behaviour, but introduce a delay mechanism before the slash is executed. This allows for potential identification and correction of accidental mistakes while maintaining deterrence against intentional attacks.
- **Reputation systems:** Integrate reputation systems that track the behavior of participants and reward those who consistently contribute positively to the network, further incentivizing honest participation.

4. Interoperability and Interchain Communication:

- **Standardized protocols:** Develop standardized protocols for communication and interaction between different blockchains, fostering interoperability and enabling seamless exchange of data and assets across diverse blockchain ecosystems.
- **Cross-chain bridges:** Implement secure and reliable cross-chain bridges to facilitate the transfer of digital assets and data between different blockchains, breaking down silos and promoting collaboration within the broader blockchain landscape.

Benefits of the Framework:

- **Improved Scalability:** This framework addresses scalability limitations by employing various solutions, paving the way for handling an increased volume of transactions without compromising decentralization.
- **Enhanced Security:** By incorporating advanced security measures, the framework mitigates potential vulnerabilities and strengthens the network's resistance against cyberattacks.
- **Sustainable Incentive Structures:** Robust incentive structures encourage desired behaviours, promote long-term network health, and ensure the alignment of interests among various participants.
- **Increased Interoperability:** Standardized protocols and cross-chain bridges foster collaboration and data exchange across blockchains, unlocking the full potential of the interconnected blockchain ecosystem.

By implementing this comprehensive framework, we can overcome the existing limitations of blockchain technology and build **resilient networks** that are **scalable, secure, and sustainable**. This will pave the way for the widespread adoption of blockchain technology and unlock its transformative potential across various sectors, shaping a more secure, transparent, and efficient future.

Project Flow/ Framework of the Proposed System:

The proposed method outlines a strategic framework for integrating game theory into blockchain systems to enhance security. Building upon the insights gained from the literature survey and analysis of preceding work, this approach incorporates elements of game theory such as strategic decision-making, incentive compatibility, and mechanism design. By designing robust consensus protocols and incentive mechanisms, we aim to create a more secure and resilient blockchain ecosystem capable of withstanding various threats and attacks.

Maintaining interoperability between a vast number of business vertical tenants in the 6G era will be crucial for several reasons:

1. Proliferation of diverse applications: 6G is expected to support a wider range of applications than ever before, catering to various industries and domains like:

- **Manufacturing:** Industrial automation, real-time monitoring and control of complex systems.
- **Healthcare:** Remote surgery, real-time patient monitoring, advanced medical imaging.
- **Transportation:** Connected and autonomous vehicles, intelligent traffic management.
- **Entertainment:** Ultra-high-definition content streaming, immersive virtual reality experiences.

Each of these applications will have specific requirements and communication protocols, making interoperability paramount.

2. Multi-stakeholder ecosystem: The 6G network is envisioned as a platform for collaboration and innovation, bringing together numerous stakeholders such as:

- **Network operators:** Providing the underlying infrastructure and core network services.
- **Service providers:** Developing and offering value-added services to tenants.
- **Vertical industries:** Deploying their own applications and leveraging network capabilities.

Interoperability ensures seamless communication and data exchange between these diverse entities, fostering collaboration and innovation across different domains.

3. Dynamic service provisioning: 6G networks are expected to be highly dynamic, allowing tenants to scale their services up or down rapidly based on demand. This requires seamless switching between different network resources and services, which is only possible with robust interoperability standards.

Challenges and Solutions:

Maintaining interoperability in such a complex environment presents significant challenges:

- **Standardization:** Defining and agreeing upon common protocols and interfaces across different vendors and industries.
- **Security:** Ensuring secure communication and data exchange among diverse tenants with varying security needs.
- **Management complexity:** Effectively managing a massive number of interconnected services and tenants.

Several potential solutions are being explored to address these challenges:

- **Open-source networking technologies:** Promoting open standards and interoperable platforms for network infrastructure and services.
- **Software-defined networking (SDN) and network function virtualization (NFV):** Enabling dynamic configuration and management of network resources, facilitating interoperability across diverse environments.
- **Standardization efforts:** Collaborative efforts by industry bodies and research communities to develop and implement unified interoperability standards for 6G networks.

By addressing these challenges and fostering an open and collaborative environment, 6G can unlock its full potential as a platform for innovation and collaboration across diverse industries and domains.

Network slicing and NS brokering are distinct concepts within the realm of network management, particularly in the context of 5G and beyond (B5G) networks. Here's a breakdown of their key differences:

Network Slicing:

- **What it does:** Creates **customized logical networks** on top of a shared physical infrastructure. Think of it like carving out virtual slices from a physical pie, each slice tailored to specific needs.
- **Focus: Resource allocation and isolation** for various applications or services.
- **Example:** A network can be sliced to provide a dedicated, low-latency network for autonomous vehicles while simultaneously supporting a high-bandwidth network for streaming services, all sharing the same physical infrastructure.

NS Brokering:

- **What it does:** Facilitates the **selection and management of network resources** based on specific requirements.
- **Focus: Matching tenant demands with the most suitable resources** from different providers, considering factors like **price, quality of service (QoS), and resource availability**.
- **Example:** An NS broker can help a company find the most cost-effective network slice with the desired bandwidth and latency for their cloud gaming application, potentially choosing from offerings of multiple infrastructure providers.

Key Differences:

Feature	Network Slicing	NS Brokering
Function	Creates and manages logical networks	Selects and manages resources in existing networks
Focus	Resource allocation and isolation	Matching demand with optimal resources
Outcome	Customized network slices	Efficient resource utilization and cost optimization

In simpler terms:

- **Network slicing** is like creating dedicated lanes on a highway for different types of vehicles.
- **NS brokering** is like choosing the most efficient lane based on your individual needs and priorities.

By working together, **network slicing and NS brokering** can help achieve efficient and adaptable network resource management, catering to diverse application demands in a dynamic and cost-effective manner.

System Model Components:

- **Fog Nodes:** These are geographically distributed computing resources located at the network edge, closer to the data sources and user devices. They provide processing, storage, and communication capabilities to support latency-sensitive applications.
- **Resource Provider (RP):** This entity manages a pool of resources at the fog nodes, such as compute power, storage, and network bandwidth. The RP can be a cloud service provider, a network operator, or even an individual user contributing their own fog node resources.
- **SFS Broker:** This is a service that acts as an intermediary between tenants (applications) and RPs. It plays a crucial role in facilitating communication, resource discovery, and service provisioning.

Flow Diagram:

1. **Tenant Request:** A tenant application running at the network edge (potentially on a fog node) sends a request to the SFS Broker. This request specifies the type of resources required (e.g., CPU, storage, bandwidth) and the desired quality of service (QoS) parameters (e.g., latency, throughput).

2. **Resource Discovery:** The SFS Broker receives the request and initiates a search process. It interacts with RPs to discover suitable fog nodes that can meet the tenant's requirements. This may involve factors like resource availability, pricing, and proximity to the tenant's location.

3. **Negotiation and Selection:** The SFS Broker negotiates with RPs on behalf of the tenant, considering factors like cost, performance guarantees, and service level agreements (SLAs). Based on the negotiation results, the SFS Broker selects the most suitable RP and fog node for the tenant's needs.

4. **Service Provisioning:** The SFS Broker establishes communication between the tenant application and the selected fog node. The RP at the chosen fog node allocates the requested resources and provides the necessary service interfaces for the tenant to interact with.

5. **Service Monitoring and Management:** The SFS Broker continuously monitors the performance of the allocated resources and the overall service delivery. It can also manage dynamic changes in resource requirements or handle situations where resources become unavailable on the chosen fog node. This may involve renegotiating with RPs or migrating the service to a different fog node.

Benefits of this Architecture:

- **Efficient Resource Utilization:** By facilitating communication and negotiation between tenants and RPs, the SFS Broker optimizes resource allocation and avoids underutilization of fog node resources.

- **Reduced Latency:** Processing data closer to the edge with fog nodes minimizes data transfer distances and network delays, leading to improved application performance.

- **Scalability and Flexibility:** The architecture can accommodate a growing number of tenants and applications by dynamically provisioning resources based on demand.

- **Cost-Effectiveness:** By enabling efficient resource utilization and potentially leveraging multiple RPs, the SFS Broker can help tenants achieve cost-effective service provisioning.

Hardware and Software Requirements:

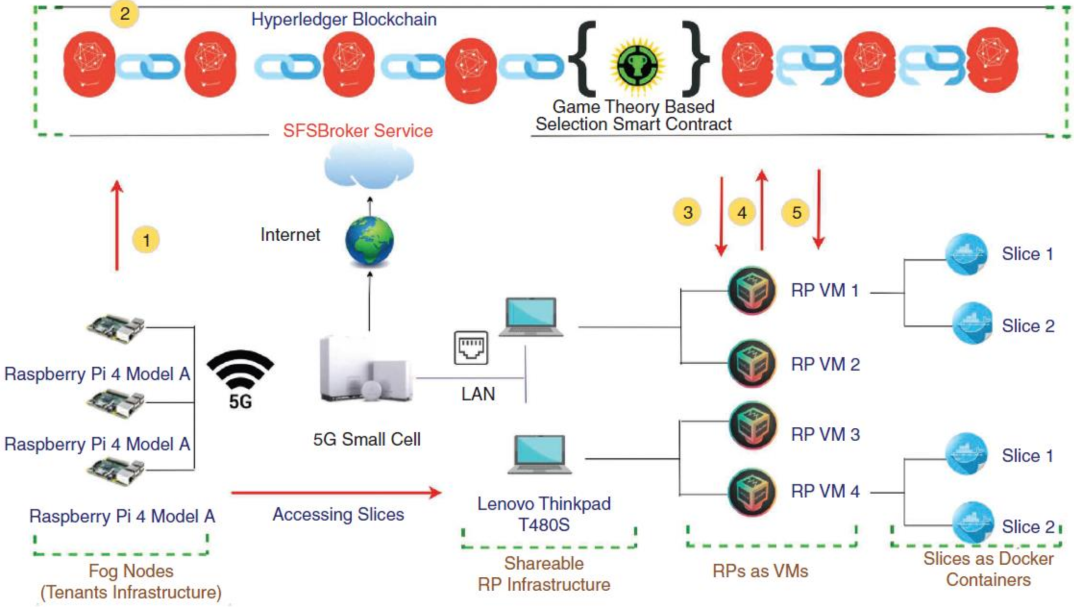
Software:

Hyperledger Fabric is an open-source framework for developing permissioned blockchain applications. It's hosted by The Linux Foundation and aims to provide a secure, flexible, and scalable platform for businesses and organizations to build and deploy distributed ledger applications.

Docker is an open-source platform for **developing, deploying, and managing containerized applications**. It allows developers to package their code and all its dependencies (libraries, system tools, settings) into standardized units called **containers**. These containers are lightweight and portable, meaning they can run consistently on any system that has Docker installed, regardless of the underlying operating system.

Hardware:

- Raspberry Pi 4 Boards for Fog Nodes
- 5G Network Cell
- Laptops for RP Infrastructure
- Virtual Machines for RPs
- Docker for ledger maintenance



Fog Node to SFSBroker Interaction using Blockchain:

This process describes how fog nodes request network slice (NS) provisioning through an SFSBroker service that leverages blockchain for secure and transparent management.

Step 1: Fog Node Request:

1. **NS Blueprint:** The fog node initiates the process by sending an NS blueprint request to the SFSBroker service via an API. This blueprint specifies the required resources for the network slice, categorized into N categories (e.g., CPU, memory, bandwidth).

Step 2: SFSBroker Verification:

1. **Retrieve Blueprint:** The SFSBroker retrieves the NS blueprint from the tenant's request.
2. **Smart Contract Validation:** A smart contract on the blockchain verifies the request parameters against the previously recorded transaction details (likely containing information about the tenant's quota or authorization).

Step 3: Request Publication and RP Responses:

1. **Publish Request:** The SFSBroker publishes the verified NS request on the blockchain, making it visible to all relevant Resource Providers (RPs).
2. **RP Offer Formulation:** RPs receive the request and individually formulate offers based on their resource availability and pricing models.

Step 4: Offer Evaluation and Selection:

1. **Offer Response:** RPs submit their offers within a designated timeframe. These offers are then recorded as transactions on the blockchain.
2. **Validation and Selection:** The SFSBroker retrieves the corresponding NS blueprint details from the blockchain ledger and compares them with the received offers.
3. **Smart Contract Selection:** A selection algorithm, detailed elsewhere (potentially in the "Performance Evaluation" section), is executed by the smart contract to choose the best RP offer based on defined criteria (e.g., cost, performance, reliability).

Step 5: Slice Deployment and Acknowledgement:

1. **Slice Formation:** The SFSBroker formulates the network slice based on the chosen offer and sends acknowledgement messages to both the selected RP and the requesting fog node.
2. **Slice Instantiation:** The chosen RP then proceeds with instantiating the network slice, provisioning the requested resources according to the agreed-upon terms.

This process leverages blockchain technology to ensure:

- **Transparency:** All transactions and interactions are recorded immutably on the blockchain, providing an auditable history.
- **Security:** The smart contract enforces the defined rules and validates the transactions, minimizing the risk of unauthorized modifications.
- **Decentralization:** The selection process is decentralized, preventing any single entity from manipulating the outcome.

While the process utilizes some functionalities that align with Hyperledger Fabric's capabilities, we use Hyper Ledger Fabric as the blockchain platform.

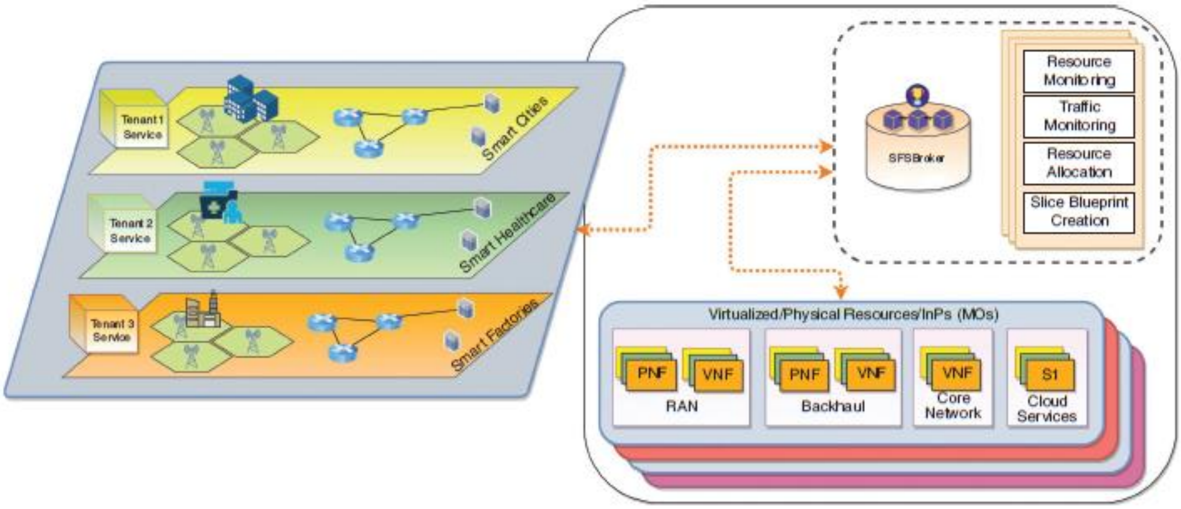
Similarities between the scenario and Hyperledger Fabric:

- **Smart Contract Involvement:** The scenario mentions a smart contract performing tasks like parameter validation and offer selection. Hyperledger Fabric supports the development and deployment of smart contracts to automate specific actions within the network.
- **Permissioned Network:** The process seems to involve authorized participants (fog nodes, RPs) interacting with the blockchain. Hyperledger Fabric is designed for permissioned networks where participants are identified and pre-approved.
- **Focus on Security and Transparency:** The scenario emphasizes secure and transparent transactions, which are core principles of Hyperledger Fabric.

Proposed System:

System Model:

It would be essential to preserve interoperability between the numerous business verticals in the 6G future.
The figure depicts a comprehensive scenario in which various tenants, or use cases, are utilizing a shared resource pool to access services.

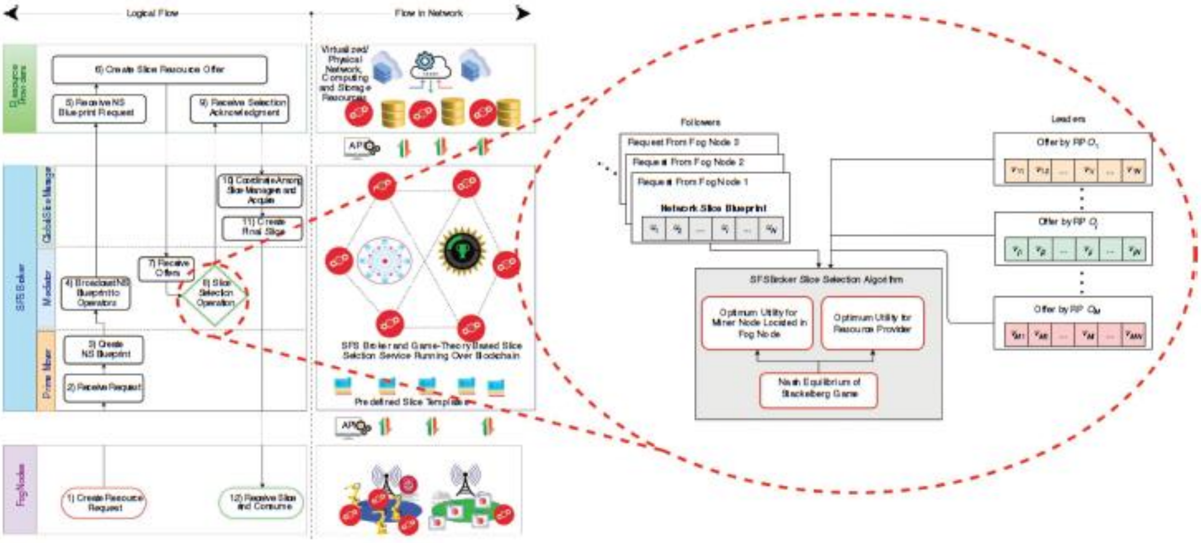


Virtualized resources, physical resources, and computing and communication infrastructure are all included in RPs.
The consumers receive these resources in the form of network services (NSs), where storage, computational infrastructure, RAN, and core network are possible candidates to share with the consumers based on demand.

SFSBroker serves as an international middleman between two ends to expedite the distribution of network addresses (NSs) to tenants that are procured from infrastructure suppliers.
In order to deliver a well-rounded and instantaneous service, the brokering mechanism must possess comprehensive knowledge regarding the supply and demand conditions of both service providers and consumers.
Tenant slice requests are received by SFSbroker, who then distributes them to RPs, chooses the best slice offer from among RPs' offerings, monitors traffic, and works with orchestration services. Large-scale service requests from tenants should be handled by this mechanism with guaranteed security (i.e., guaranteed authentication, availability, privacy, trust, and access control).

Functional Architecture:

The given figure shows the architectural framework and flow diagram for the SFS Broker mechanism.


Logical Flow
Flow in Network
Resource Providers
6) Create Slice Resource Offer
5) Receive NS Blueprint Request
9) Receive Selection Acknowledgment
Virtualized/ Physical Network, Computing and Storage Resources
API
SFS Broker
Global Slice Manager
Mediator
Prime Mover
10) Coordinate Among Slice Managers and Acquire
11) Create Final Slice
7) Receive Offers
8) Slice Selection Operation
4) Broadcast NS Blueprint to Operators
3) Create NS Blueprint
2) Receive Request
SFS Broker and Game-Theory Based Slice Selection Service Running Over Blockchain
Predefined Slice Templates
Fog Nodes
1) Create Resource Request
12) Receive Slice and Consume
Followers
Request From Fog Node 3
Request From Fog Node 2
Request From Fog Node 1
Network Slice Blueprint
Leaders
Offer by RP O1
Offer by RP Oj
Offer by RP OM
SFSBroker Slice Selection Algorithm
Optimum Utility for Miner Node Located in Fog Node
Optimum Utility for Resource Provider
Nash Equilibrium of Stackelberg Game

Fog Nodes: The consumer end of the suggested solution, which is deployed in various use cases, is represented by the fog nodes and directly communicates with the IoT tenants. Each fog node in the multitenant scenario serves one or more IoT tenant clusters for certain use cases.

RP: RPs are regarded as the entities that offer networking and computational resources, or infrastructure, to the consumers on a multioperator platform. Cloud computing infrastructure, storage, network services, and mobile data connectivity are a few examples of these RPs. This can include a diverse range of service providers, including MNOs, cloud service providers, and local (micro) network operators.

SFS Broker. SFSBroker is represented by the middle layer in the diagram. It is implemented as a blockchain network that runs as a service in the central cloud and serves as a middleman between fog nodes and RPs. The three submodules that make up the SFSBroker are the global slice manager, mediator, and prime mover. The prime mover is in charge of responding to requests for resources and drafting the NS blueprint. Mediator performs the slice selection process and broadcasts NS blueprints to RPs.

An algorithm modeled using the Stackelberg game is used to select the best matching RPs offer (or build a new slice with several RPs) for a particular NS blueprint.17 Global Slice Manager uses the fog node to manage the last slice offer to IoT tenants.

SFSBroker uses a consortium blockchain to be deployed as a decentralized entity and employs a modular architecture for improved scalability. The flow of SFSBroker has 12 phases, as Figure illustrates:

This study suggests an algorithm based on the Stackelberg game model to choose the best NS given two types of input: RP resource offers and IoT tenant demands. A smart contract has been written to represent the selection method based on the game model. A single point of failure can be eliminated, the selection service can be moved from the cloud to the edge, and improved operational transparency can be achieved with an immutable transaction ledger, among other benefits of such an implementation.

Step 1: When an Internet of Things tenant submits a service request, a fog node initiates an instance of the process. The fog node generates a resource request in response , incorporates it into a transaction, digitally signs it, and transmits the request—or transaction, as it is known on the blockchain—to SFSBroker. Since fog nodes serve as IoT nodes' gateways to SFSBroker, they make the request on behalf of the tenants.

Step 2: After receiving the request, the prime mover saves the validated request in the blockchain and checks the digital signature to confirm the legitimacy of the requesting fog node.

Step 3: Subsequently, the mediator module receives the NS blueprint that the prime mover created based on the quantitative demand (for different preset categories of resources) in the received request.

Step 4: The mediator module merely broadcasts the NS blueprint requests to any RP that is accessible. Writing to the blockchain allows for broadcasting, granting access to the NS blueprints to all legitimate RPs. The mediator module additionally initiates a timer t, one for each NS blueprint request, at the moment of broadcasting.

Step 5: RPs analyzing the NS blueprint request to ensure its viability. This means that every RP objectively contrasts the quantity of resources requested with the amount of unoccupied resources that are accessible.

Step 6: Next, the willing and able RPs to lease their resources (based on demand) formulate offers with terms and conditions, incorporate them into digitally signed transactions, and forward them to the SFSBroker mediation module.

Step 7: Once agreement is reached by approving offer values, the mediator module checks each and every one of the incoming offers before storing them on the blockchain. When timer t expires for a particular NS blueprint request, the window of opportunity for receiving bids from RPs closes.

Step 8: Moreover, it initiates the application of a selection algorithm to the received offers. After establishing an agreement, the mediator uses the results of the selection algorithm to determine the ideal offer and, based on that offer, writes one or more acknowledgements to the winning RP(s) at their blockchain address(es).

Step 9: Following its decision to provide a portion or the entirety of an NS, the winning RP (virtually) slices the resource(s) and notifies SFSBroker's global slice manager module.

Step 10 and 11: The global slice manager works with the slice management of the winning RP or RPs to arrange the acquisition of the constituent resources, the creation of the final federated NS, and its delivery to the fog node.

Step 12: The federated NS is finally received by the fog node. It should be noted that all correspondence between RPs, SFSBroker, and fog nodes is digitally signed and captured in immutable transactions. Additionally, the stakeholders can trust that the best bids are chosen in a decentralized manner through the use of blockchain-based SFSBroker.

From the standpoint of the customer, the best deal is crucial, and from the RP's, maximum profit is crucial. The slice selection process of SFSBroker must take into account the perspectives of fog nodes and RPs. To optimize their utilities, RPs and fog nodes both continuously modify their tactics. We explain in the selection method how an exclusive winner being one RP is just a special scenario where the winning RP can offer the best offer for every resource category in the specified NS. Nonetheless, an output of the same selection algorithm might produce an ideal offer that combines resources from several suppliers to establish a federated network system with the right modifications.

Slice Selection Algorithm:

The total number of winning RPs may not exceed the entire number of unique resource categories.

We assume that n number of resource (or network function) categories are used in the creation of a certain NS blueprint. The quantity of resource demand for the i^{th} resource, where $i \in \{1, 2,..., n\}$, is indicated by the variable u_i in a resource request sent by a certain fog node. O_j represents the j^{th} RP out of a total of m RPs, where j is any of the following: 1, 2,..., m}. Assume that RP (or operator) O_j determines the unit price of the i^{th} resource using the pricing strategy $\{v_j = [v_{ji}\ i \in N : 0 < v_{ji} < \bar{v}]\}$where v_{ji} represents the proposed price and $\bar{v}$ represents the maximum price. Moreover, c is taken as the common and constant cost resulting from the general operation and maintenance cost.

The selection method should identify the best predicted utility (reward) that each RP offers for a certain NS blueprint, as shown in the picture. In this case, we look at one NS blueprint that was developed in response to a resource request made by a miner node that was housed in a fog node. Additionally, the estimated utility needs to be calculated for each resource category that the mining node requests.

As a result, O_j RP's predicted utility (reward) can be written as

$$P_j = \sum_{i=1}^{N} u_i v_{ij} - \sum_{i=1}^{N} c u_i$$

In addition to that, we define a utility function P_i expected utility (reward) for R_i resource category requested by the miner node located at fog node (based on the offer given by O_j):

$$P_i = P \times \frac{u_i}{\sum_{i=1}^{N} u_i} - v_{ji} \times u_i$$

As previously said, the selection algorithm first determines the overall service demand of fog nodes and sets the offer rates to maximize profits for RPs after receiving all of the offers from RPs.

The miner nodes situated in fog nodes, on the other hand, must optimize the payout for each resource requirement. Thus, the selection method will use (1) and (2) as stated in Yao et al.7 to design the optimization issue of miners while taking into account the price strategies of RPs.

The Stackelberg game is represented mathematically by two sides, with the RPs acting as leaders and the fog nodes (miner nodes) acting as followers. The selection algorithm is in charge of providing updates to the fog nodes and RPs regarding their ability to continuously modify their strategies in order to optimize their utilities. The goal of the stackelberg game is to locate the Nash equilibrium, in which no player intends to change course after taking into account the decision made by its opponent.

According to Yao et al. (7), there is a Nash equilibrium and the utility functions are strictly concave. According to Yao et al.7, a reinforcement learning method is utilized to determine the NE.

In the first part of the selection, the algorithm should be run for each resource request and compute the optimal values for the operator price and required resource amount for each resource category as shown in Table 1. By referring to the values in Table 1, the NS is formed in such a way as to minimize the total price and match the resource availability with the operators.

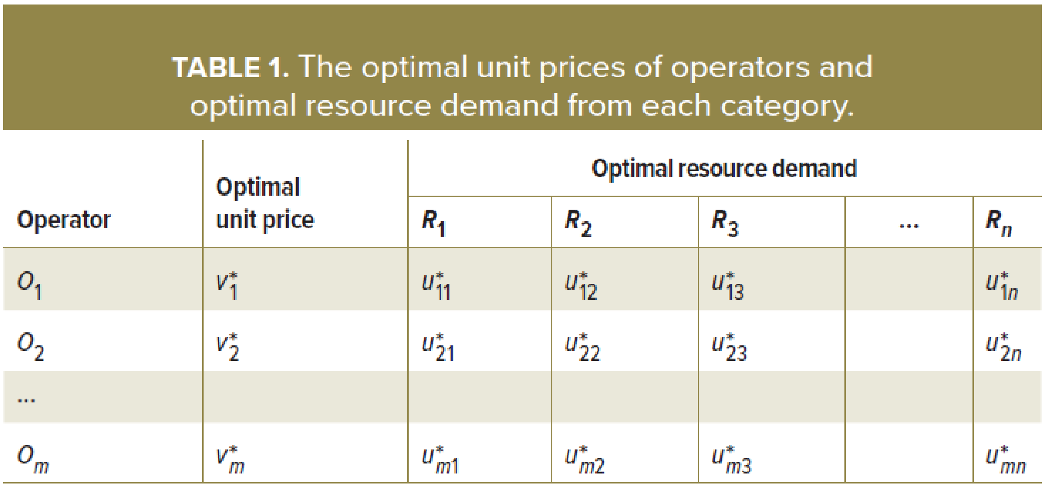
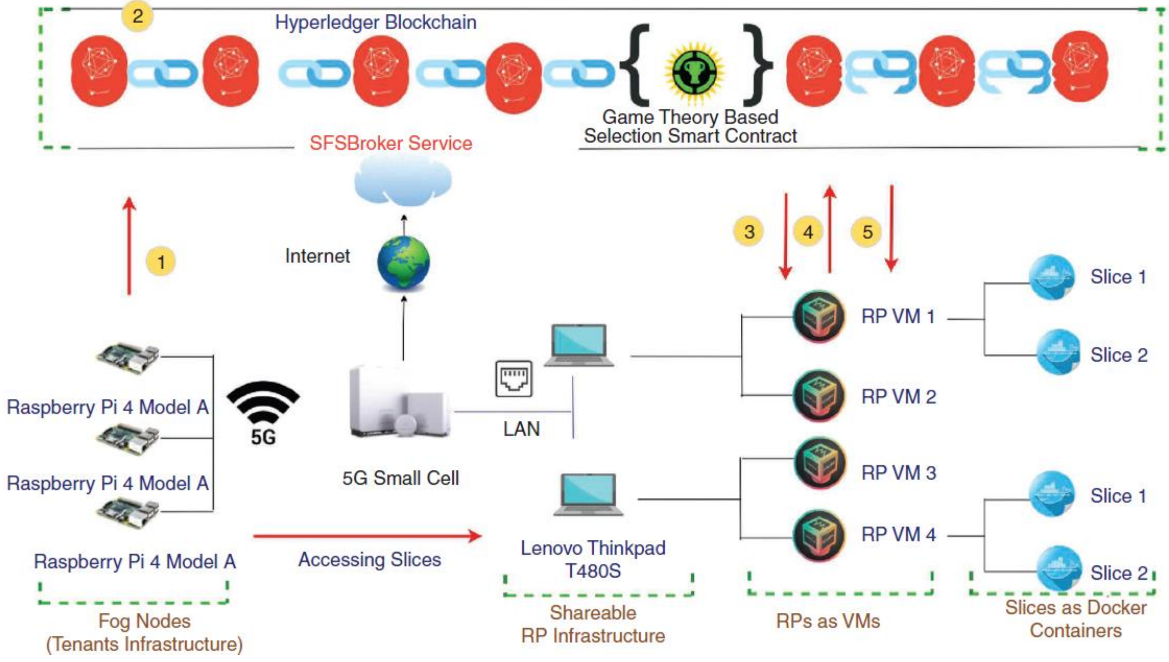


TABLE 1. The optimal unit prices of operators and optimal resource demand from each category.

Operator	Optimal unit price	Optimal resource demand				
		R_1	R_2	R_3	…	R_n
O_1	v_1^*	u_{11}^*	u_{12}^*	u_{13}^*		u_{1n}^*
O_2	v_2^*	u_{21}^*	u_{22}^*	u_{23}^*		u_{2n}^*
…						
O_m	v_m^*	u_{m1}^*	u_{m2}^*	u_{m3}^*		u_{mn}^*

Evaluation:

The proposed solution includes four main service components: fog nodes, SFSBroker, 5G infrastructure, and blockchain service. The implementation setup developed to perform a proof of concept of the SFSBroker is illustrated in in the given figure.



Raspberry Pies are used as fog nodes, while Ubuntu 18.04 virtual computers running on a Lenovo Thinkpad T480S running Windows 10 (64-bit) with 16 GB RAM are used to emulate the sharing RP infrastructure. The Hyperledger was installed on a cloud instance with a 2.33 GHz Intel Xeon CPU and 16 GB of RAM that was accessible via public IP.
IoT tenants, or fog nodes, are deployed with 5G dongles (HUAWESPAN E3372) and Raspberry Pi 4 Model A devices. The testbed's various components were connected via the 5G test network (5GTN)18.

The University of Oulu and the VTT Technical Research Center of Finland have set up an experimental 5G network called the 5G test network, which is utilized for 5G-related activities.
5GTN facilitates high-speed access for cloud resources, edge computing capabilities, and 5G new radio (5GNR) connectivity. The Internet of Things components in our experiment are linked to the 5GTN, and we utilized the fast Internet connection provided by the 5GTN backhaul to link them to the cloud layer.

Docker containerization is used to mimic NS instances. A pre-made Docker image is used to emulate the NS blueprint. The Docker container that is now operating and initialized with the various resource categories requested by the fog nodes is used to imitate the instantiated

network system.
As seen in Figure , the Docker containers with designated resources (such memory and storage) that operate on the virtual machines (VMs) mimic how NS uses RP resources. In order to do the evaluation, we'll assume that each resource request's associated service is operating in a separate port on the instantiated Docker container and that customers can access all of the services via those ports.
But the simulation makes sure that the chosen slice has actually been instantiated.

The implementation setup demonstrates a near realistic transaction simulation for the proposed architecture:

- Blockchain: A five-node Hyperledger Fabric19 1.4.4 instance with a raft consensus setup and Java smart contract execution is used to implement blockchain.
-

  SFS Broker: The global slice manager, mediator, and prime mover of SFSBroker are all implemented as smart contracts.

- RPs: In order to keep this initial prototype simple, we will treat all RP kinds and MNOs in the same ground as equal entities that are able to supply networking or computational resources. As a result, for the remainder of the section, both RP and MNO are used. Virtual machines (VMs) with preallocated computing resources serve as MNO representations. A subset of the VM resources that the tenants will have access to is called a slice.

- Connectivity: Connectivity among fog nodes, blockchain, and RPs is established using Message Queuing Telemetry Transport (MQTT). Hyperledger software development kit integrates with MQTT library to push the resource requests and offers to the blockchain.

The implementation steps of SFSBroker reflected in Figure are:

- Step 1: Fog nodes use the application programming interface (API) to send an NS blue print request (i.e., with 1... N resource categories) to the SFSBroker service.

- Step 2: The tenants provide SFSBroker with the NS blueprint, and the smart contract verifies the parameters against the committed blocked transaction in the request.

- Step 3: The NS request is posted to the blockchain via SFSBroker. After receiving a request, RPs formulate specific offers.

- Step 4: RPs answer by making proposals. Responses are received by SFSBroker within the allotted time frame. The offers that are received are committed to the blockchain as transactions. The ledger is accessed to retrieve the relevant NS blueprint, and the ledger transaction is used to validate the parameters. The smart contract chooses the best RP offer by executing the selection method mentioned in the "Performance Evaluation" section.

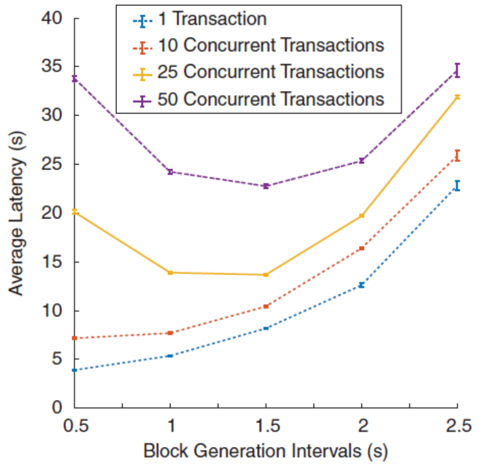
- Step 5: SFSBroker creates the slice and responds to the fog node and RPs regarding the best offer. The slice is instantiated by the selected RP.

Conclusion and Future Work:

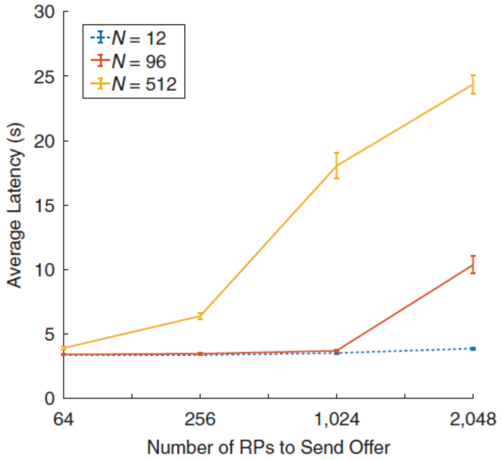
Performance Evaluation:

Block time variation and E2E slice generation latency. The multitenant situation and the assessment of E2E slice formation delay (steps 1–10 in Figure 2) are illustrated in this experiment.

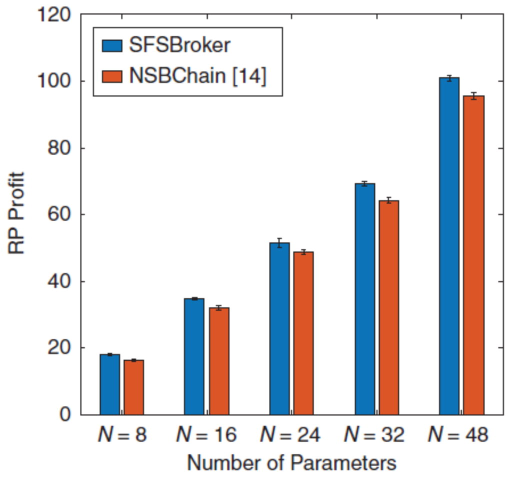
Variable block generation time intervals are used to measure E2E latency in Hyperledger. The fog node starts a series of concurrent transactions for a specific defined block generation period (that is, for each test 1, 10, 25, 50). For a particular block time and concurrent transaction, the E2E latency is measured for each slice using 100 trials. From there, the confidence intervals, or 95% of them, are calculated [Figure (a)].



(a)



(b)



(c)

The performance evaluation results in the implementation and simulation: (a) The E2E latency of slice selection with different block-mining time configurations. (b) The E2E latency of slice selection for multiple RPs and different parameter counts. (c) A simulation result comparison of SFSBroker and NSBChain.

The experiment results show that the reduction in the block generation interval is not directly advantageous in terms of latency.

We note that before the intermediary steps of all the transactions in the batch are finished, the blocks are mined. Transactions from each batch are spread across several blocks in these situations. The delay caused by distributing transactions across several blocks has an impact on the total batch completion time when calculating the completion latency. In other scenarios, the block production time increases the E2E slicing formation latency.

Latency in slice selection: The latency of the game theory-based selection method for various inputs was the main focus of this test. Here, we increased the amount of parameters N in the slice request to add more than one RP. The block-mining time for the experiment was set at 1,000 milliseconds. Each test is run 100 times for each RP and parameter setting, with one transaction submitted to the SFSBroker every trial. The latency is measured for steps 8 and 9 in this configuration; these steps are shown in Figure. This entails deciding which offer is best for a tenant request, committing the transaction to the ledger using the chosen MNO offer, and endorsing the ledger transaction. The latency grows as the number of parameters (N) does.

The graphs in Figure (b) show that the algorithm's selection latency is directly impacted by an increase in the number of parameters (N) and RPs. The system, even with 2,048 RPs and 512 parameters, can choose the best offer in less than 30 seconds. We raised the number of RPs in the experiment to 2,048 because we needed to assess the performance on various input scales. Scaling up to 2,048 RPs [Figure 4(b)] mimics the scenarios where local 5G operators supply the services as RPs. We evaluated RPs and MNOs at the same ground level.

Comparison with previous works:

Using Matlab, we examined how SFSBroker and NSBChain14 algorithms behaved. In each experiment, the number of parameters (N) varies while the number of RPs (M) remains constant. Every experiment has 100 trials, and on each trial, the RPs earnings are computed. In every trial, the resources that customers requested, the expenses incurred to fulfill those demands, and the profits were all determined at random. We made the assumption that the total of the randomly generated cost and profit values for each trial would be the final price that the customer would be given for each slice request.

The resource offers from the RPs and the customer resource demand are the inputs used in each algorithm. The findings displayed in Figure 4(c) indicate that SFSBroker's RP earnings are greater than NSBChain's. Rather than choosing the lowest offer, SFSBroker selects based on the profit factor of RPs and the lowest pricing. As a result, SFSBroker offers greater equity than NSBChain to both customers and RPs throughout the slice selection procedure.

In this study, we presented an NS brokering mechanism (SFSBroker) for applications in multioperator multitenant situations, which are anticipated to be present in 6G networks, using blockchain technology. With SFSBroker, the optimal tenant-operator match that guarantees the best utilities to the consumer and service provider is found by modeling the situation as a Stackelberg game, where Nash equilibrium can be reaching. There are specifics on the implementation setup and the functional architecture given. Additionally, the E2E slice creation latency and slice selection delay are used to assess SFSBroker's performance. The findings demonstrated that when block production time increases, so does the E2E slicing creation latency. Furthermore, the slice selection delay is significantly impacted by increasing numbers of N and RPs.

Modules:

Key characteristics of Hyperledger Fabric:

Permissioned Network: Unlike public blockchains where anyone can participate, Hyperledger Fabric operates on a permissioned network. This means that only pre-approved participants, identified and authenticated by the network, can join and transact. This feature is particularly attractive for businesses looking for a more controlled and secure environment for their transactions.

Modular Architecture: Hyperledger Fabric is built on a modular architecture, allowing developers to choose and combine different components based on their specific needs. This flexibility makes it suitable for a wide range of use cases across different industries.

Pluggable Consensus Mechanism: Hyperledger Fabric is unique in its use of a pluggable consensus mechanism. This means that the consensus algorithm used to validate transactions can be chosen or replaced based on the application's specific requirements. This allows for customization and optimization of the network's performance and security.

Private by Design: Hyperledger Fabric prioritizes privacy by default. Transactions typically occur only between authorized participants, and the ledger itself doesn't store all data publicly. This ensures that sensitive information remains confidential and only accessible to relevant parties.

Focus on Security: As a permissioned blockchain platform, Hyperledger Fabric incorporates various security features to protect against unauthorized access and tampering. This includes cryptographic mechanisms for data encryption and digital signatures, as well as access control mechanisms to limit who can perform specific actions on the network.

Applications: Hyperledger Fabric is being used across various industries for a diverse range of applications, including:

- **Supply chain management:** Tracking the movement and origin of goods throughout the supply chain.
- **Trade finance:** Streamlining trade finance processes and reducing fraud risks.
- **Healthcare:** Securely sharing patient data and managing electronic health records.
- **Voting systems:** Enhancing the security and transparency of voting processes.

Overall, Hyperledger Fabric provides a robust and versatile framework for businesses and organizations to explore the potential of blockchain technology in a secure and controlled environment.

Key benefits of using Docker:

- **Faster development and deployment:** Docker helps streamline the process of building, testing, and deploying applications by providing a consistent environment across development, testing, and production stages.

- **Improved application isolation:** Each container runs in isolation from other containers on the same system, ensuring that applications don't interfere with each other or the host system.
- **Increased portability:** Docker containers are self-contained and portable across different environments, making it easier to deploy applications on different platforms.
- **Efficient resource utilization:** Containers are lightweight and share the underlying operating system kernel, making them more resource-efficient compared to virtual machines.

Here are some key components of the Docker ecosystem:

- **Docker Engine:** This is the core software that installs and runs containers on your system. It allows you to build, run, and manage containers.
- **Docker Hub:** This is a public registry of pre-built Docker images, allowing developers to share and discover containerized applications.
- **Docker Compose:** This is a tool that helps define and manage multi-container applications with a single YAML file. It simplifies the process of deploying applications that consist of multiple interacting services.

Docker is widely used in various industries and applications, including:

- **Software development:** Streamlining development workflows and simplifying deployments.
- **DevOps practices:** Enabling continuous integration and continuous delivery (CI/CD) pipelines.
- **Cloud computing:** Providing a portable way to deploy and manage applications across different cloud platforms.
- **Data science and machine learning:** Simplifying the creation and deployment of data science environments.

Overall, Docker offers a powerful and versatile platform for developers and IT professionals to build, deploy, and manage containerized applications efficiently and effectively.

References:

https://ieeexplore.ieee.org/document/10058751/

https://www.sciencedirect.com/science/article/pii/S0142061523001680

https://ieeexplore.ieee.org/document/9282820

https://ieeexplore.ieee.org/document/9797894

https://ieeexplore.ieee.org/document/10109049

https://ieeexplore.ieee.org/document/8684838