A Game-Theoretic Analysis of Shard-Based Permissionless Blockchains

Mohammad Hossein Manshaei , Murtuza Jadliwala , Nindya Maiti and Mahdi Fooladgar [IEEE'18]

K Ugender

Indian Institute of Science

Outline

- Introduction
- Prerequisites
- Protocols
- Game Model
- Numerical Analysis
- Conclusion

Objective

- The objective of this paper is to introduce and discuss the research on shard-based consensus protocols for public blockchains, with a specific focus on understanding the strategic behavior of rational processors within committees.

- We will explore how game theory models can be used to analyze processor behavior and propose novel incentive mechanisms to foster cooperation and prevent free-riding in shard-based consensus protocols.

- The paper aims to highlight the importance of these findings in enhancing the scalability and overall performance of blockchain networks.

Introduction

- The Blockchain is an immutable distributed database that records time-sequenced transactions, which are grouped into blocks.

- The first blockchain protocol was introduced in 2009 by Satoshi Nakamoto, the creator of Bitcoin.

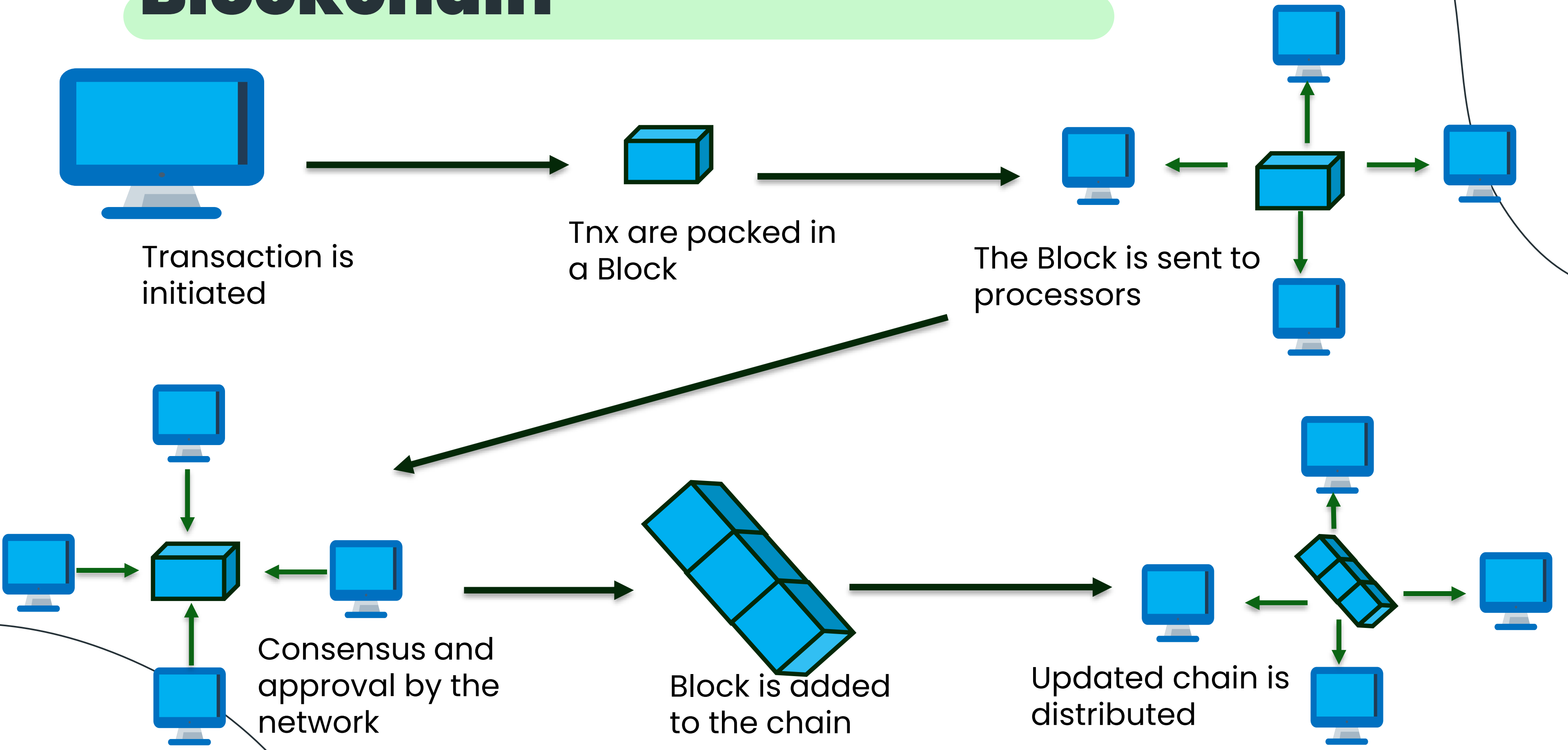
- The blockchain protocol relies on a Consensus Algorithm, often referred to as Nakamoto consensus, to reach agreement on the state of the blockchain. This consensus accommodates potentially malicious participants.

- Despite its tremendous popularity, one significant shortcoming of Bitcoin's consensus protocol is its low transaction throughput and poor scalability.

- There have been significant efforts towards improving the transaction throughputs, for example, BIP and Bitcoin-NG for Bitcoin and Raiden for Ethereum.

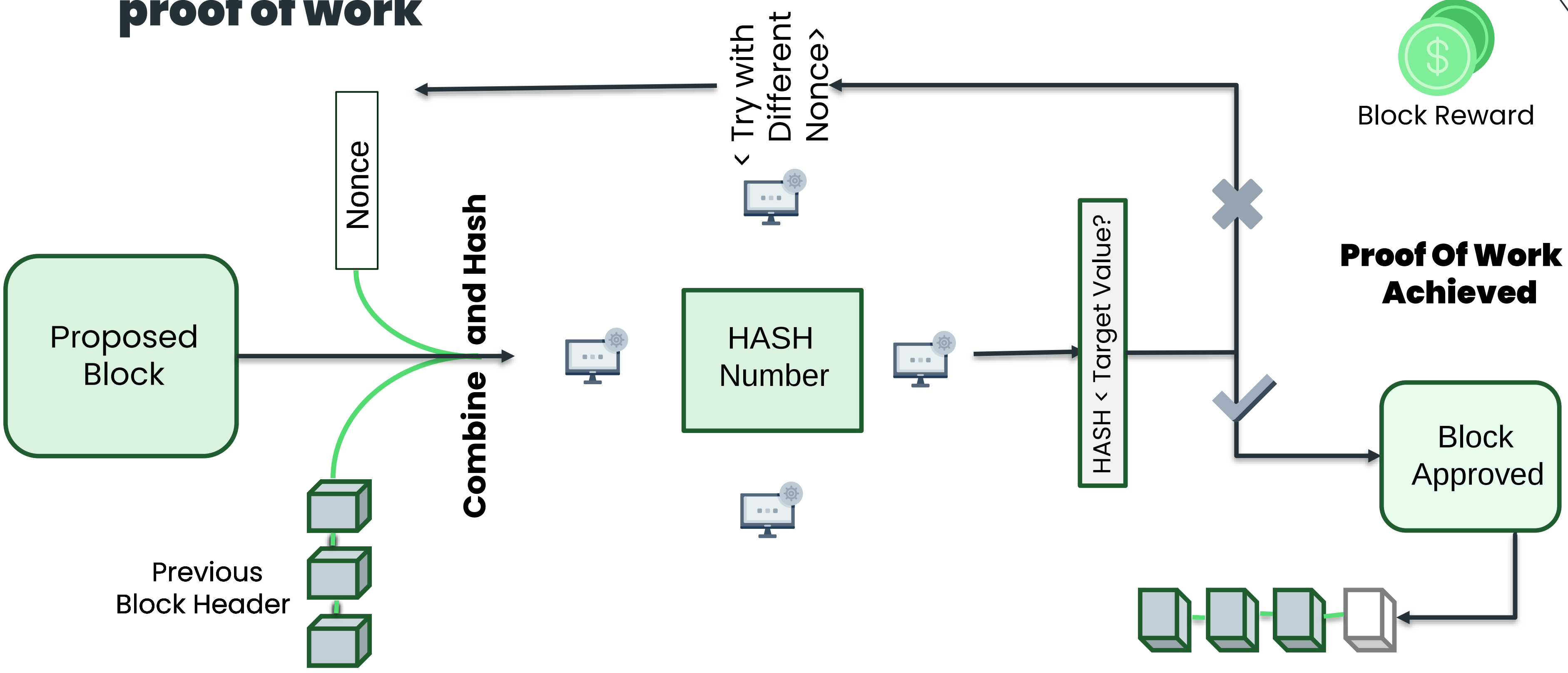
- One key outcome of this line of research is Sharding

Blockchain


Transaction is initiated
Tnx are packed in a Block
The Block is sent to processors
Consensus and approval by the network
Block is added to the chain
Updated chain is distributed

Consensus

proof of work

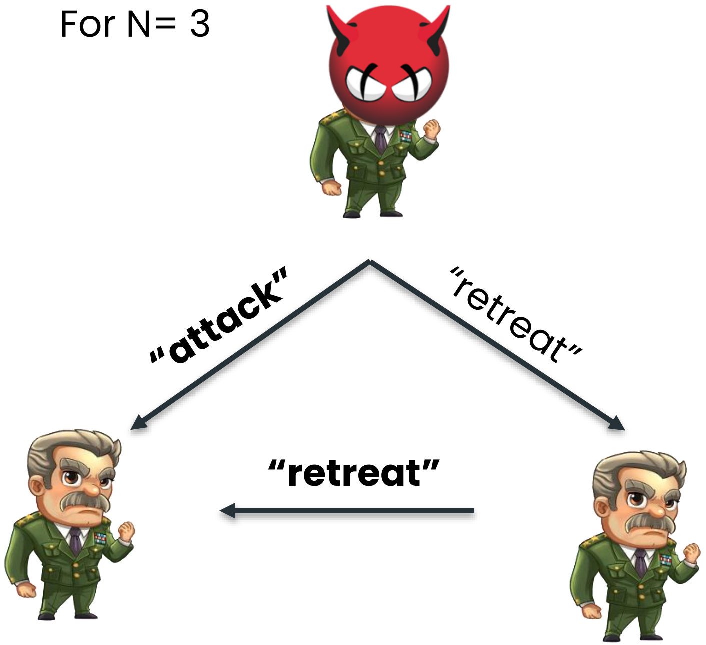


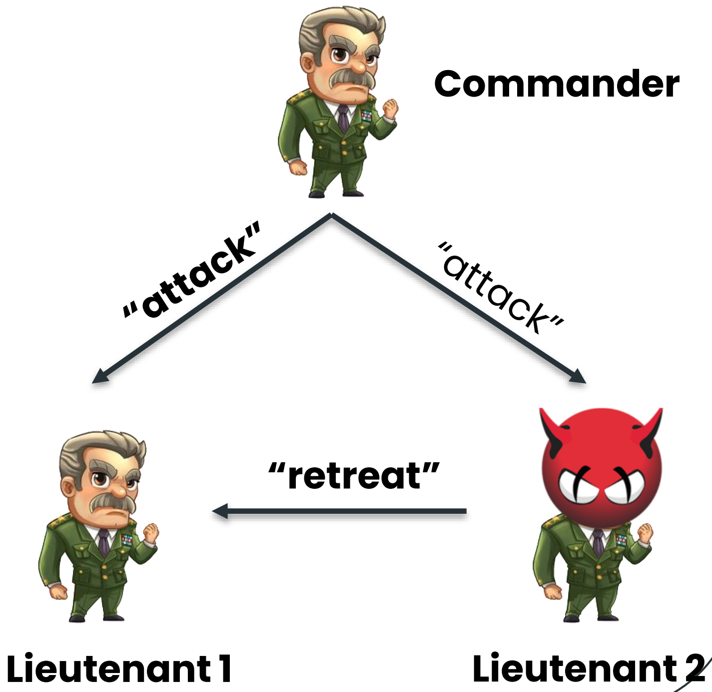
Byzantine Fault-Tolerant Consensus

[The Byzantine Generals Problem (acm.org)]

BFT Consensus

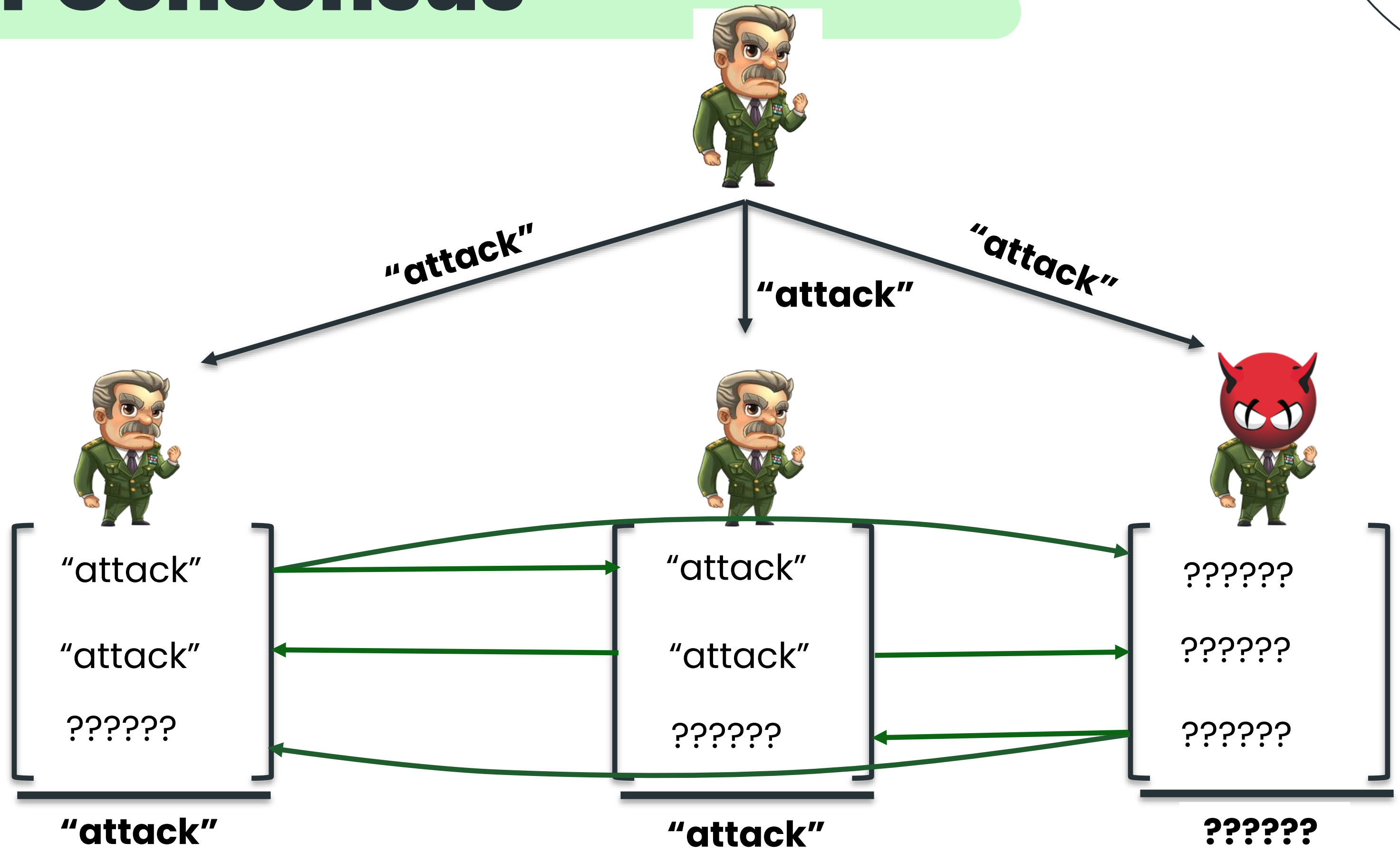
For N= 3





No solution for **Traitors** $> \frac{N}{3}$

BFT Consensus



Protocols

1

SHARD-BASED CONSENSUS PROTOCOL

We First define Shard-Based consensus protocol and analyze cost imposed on processors

2

SHARD-BASED BLOCKCHAIN GAME

we present the game-theoretic aspects of a shard-based blockchain protocol with multiple processors in an honest but selfish environment.

3

INCENTIVE-COMPATIBLE REWARD SHARING

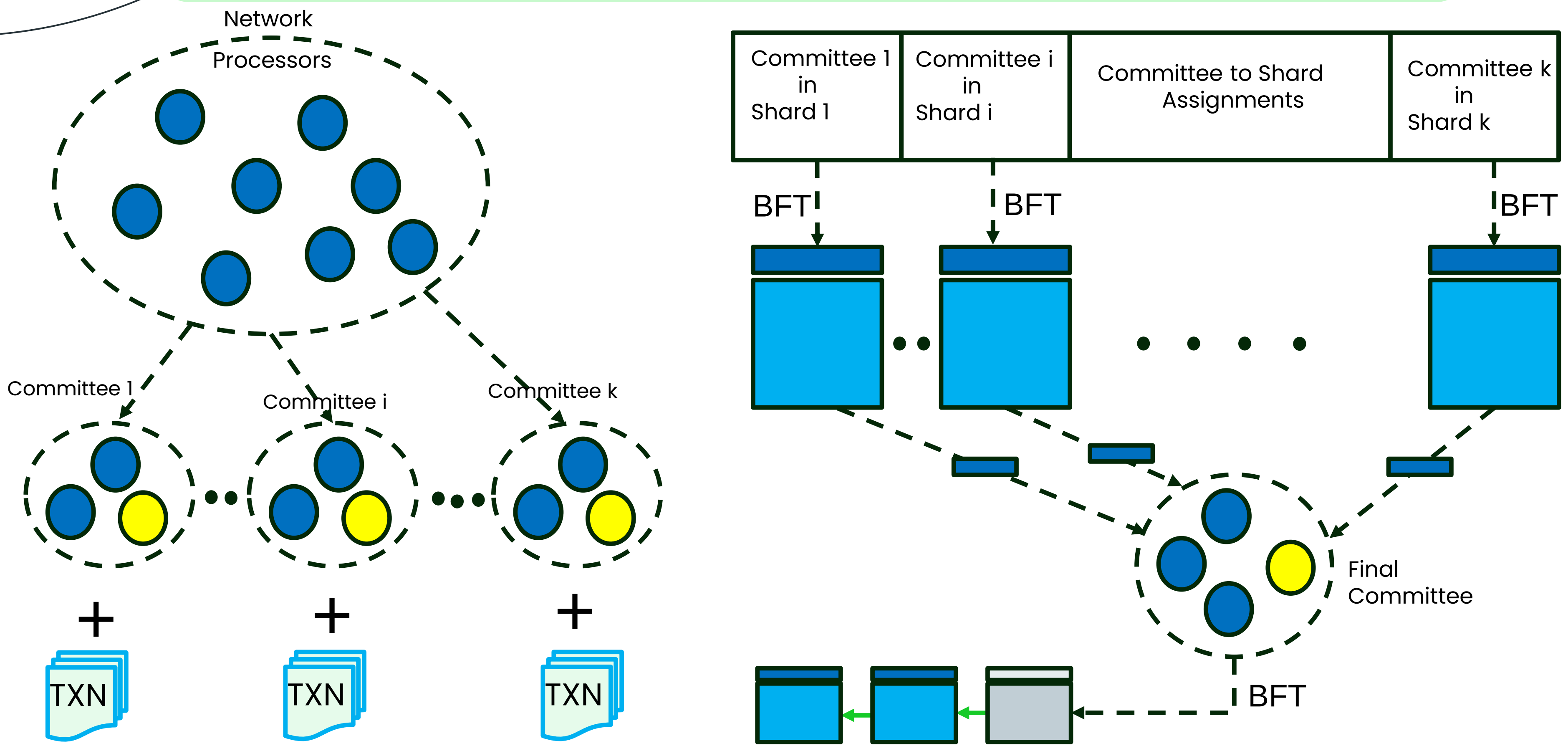
Our next goal is to extend the current shard-based consensus protocols by considering the strategic behavior of rational processors

4

NUMERICAL ANALYSIS

To validate our proposed incentive-compatible protocol, we'll compare it with uniform and fair reward sharing protocols in shard-based blockchains.

SHARD-BASED CONSENSUS PROTOCOL



Elastico Protocol

1. **Committee Formation:** Processors establish publicly verifiable identities through PoW puzzles. Processors are assigned to committees, and each committee processes a distinct shard.

2. **Overlay Setup:** Processors communicate to discover identities within their committee, resulting in a fully-connected overlay network for each committee.

3. **Intra-Committee Consensus:** Processors in committees run a standard PBFT to agree on a set of transactions. Each committee sends its consensus set of transactions (shard Bi) to a final committee for inclusion in the new block B.

4. **Final Consensus:** The final committee merges consensus shards (Bi) to create a final block B. Each processor validates the shard signatures and computes a union.

5. **Randomness Generation:** The final committee then generates random strings and broadcasts them to the network.

organization phase

Committee Participation phase

Processors Cost

The cost borne by the processors in each epoch is characterized as follows:

1. ***mandatory cost*:** The cost borne by the processors in the first phase of the protocol
   Let's assume this cost is c^m
   This cost depends on the difficulty of PoW
2. ***Optional cost*:** The cost borne by the processors in the second phase of the protocol
   Let's assume this cost is c^o this cost has two components
   i. *Fixed Component* c^f
   ii. *Transaction Dependent component* c^v

The average cost bore by the processor P_i is given by c^t

$$c_i^o = c_i^f + |x_i^j| c^v$$

$$\boldsymbol{c_i^t = c^m + c_i^o}$$

Game Model

Game Theory allows us to model the shard-based blockchain game as a static game as all processors must choose their strategy simultaneously.

This modeling decision also keeps our analysis tractable, while conforming to a simple model of processor rationality.

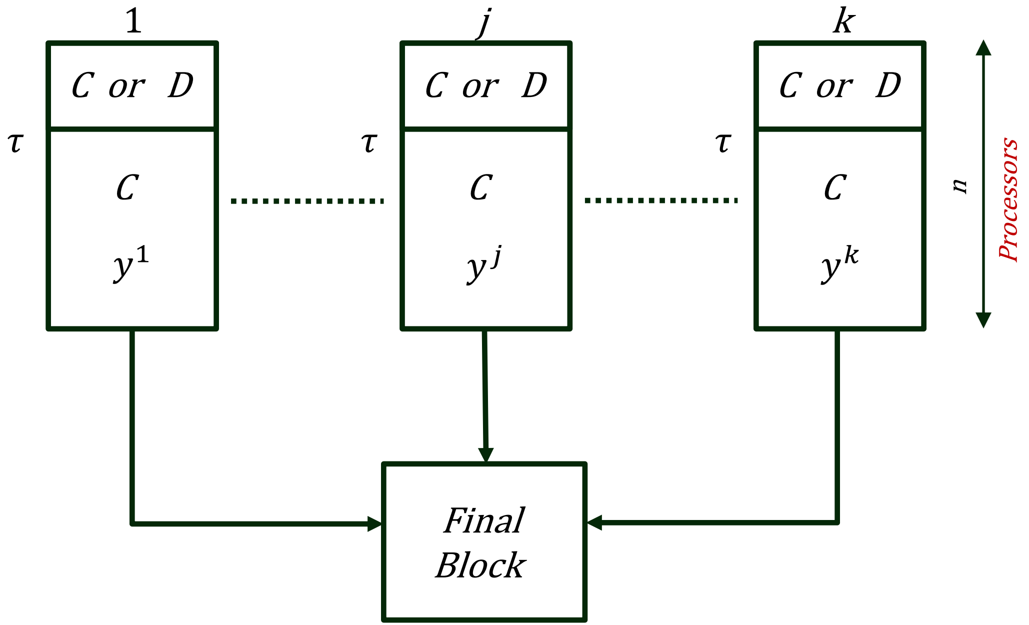
$$G = (P, S, U)$$

$P = \{ P_i \}_{i=1}^{N}$

$S = \{ C, D \}$

$u_i^j(C) = b_i - c_i^t$

$u_i^j(D) = b_i - c_i^m$



Game Analysis

Definition 1: In a Nash equilibrium strategy profile, none of the players can unilaterally change its strategy to increase its utility.

Let's consider two processors and analyse the game

	Cooperate	**Defect**
Cooperate	$(b - c^t, b - c^t)$	$(b - c^t, b - c^m)$
Defect	$(b - c^m, b - c^t)$	$(b - c^m, b - c^m)$

This game is as good as Prisoners Dilemma.

Public Good Game

Hamburger introduced the N-player version of the Prisoner's Dilemma game, known as the Public Good Game (PGG), more than 20 years after the original definition.

The PGG is Defined as Follows :

1. In PGG each player has two strategies (C, D)
2. Players can cooperate and pay a contribution α or defect
3. Then all the contributions are summed up and multiplied by $\gamma > 1$
4. Finally, the total reword is distributed among the players uniformly

Now let's analyse the utilities of the players if n out of N players cooperate

$$u(C) = \frac{\alpha\gamma n}{N} - \alpha$$

$$u(D) = \frac{\alpha\gamma n}{N}$$

Game G as a PGG

In our shard-based blockchain game G, it is demonstrated that G behaves as a PPG. In other words, if all processors initially defect, the system fails to create new blocks and remains in the same state.

Theorem 1 : *In each epoch of a shard-based blockchain game G with N processors, if rewards are equally shared among all processors, then G reduces to a public goods game.*

Theorem 2 : *In each epoch of a shard-based blockchain game G with N processors, if rewards are equally shared among all processors, we cannot establish All Cooperation strategy profile as a Nash equilibrium.*

Theorem 3 : *Let* $C_j^{l_j}$ *and* $D_j^{n-l_j}$ *denote the sets of* l_j *cooperating processors and* $n - l_j$ *defecting processors inside each shard* j *with* n *processors. If* $L = \sum_{j=1}^{k} l_j$ *is the total number of cooperative processors,* (C^L, D^{N-L}) *represents Nash equilibrium profile in each epoch of the game G, if and only if* $l_j = \tau$ *in all shards* j*, where* $C^L = \bigcup_j C_j^{l_j}$ *and* $D^{N-L} = \bigcup_j D_j^{n-l_j}$ *.*

Fair Reward Sharing

The Game model is extended to include a fair reward sharing, where only processors that cooperated with others within shard are rewarded.

- Payoff of cooperative processors in set $C_j^{l_j}$: $u_i^j(C) = \frac{BR}{kl_j} + \frac{r|y^j|}{l_j} - \left(c^m + c^f + \left|x_i^j\right| c^v\right)$
- Payoff of defective processors is calculated as : $u_i^j(D) = -c^m$

Theorem 4: *Let $C_j^{l_j}$ and $D_j^{n-l_j}$ denote the sets of l_j cooperating processors and $n - l_j$ defecting processors inside each shard j with n processors, respectively. (C^L, D^{N-L}) represents a Nash equilibrium profile in each epoch of game G^F, if the following conditions are satisfied:*

1. *In all shards j, $l_j \geq \tau$.*
2. *If for a given processor P_i in shard j, $x_i^j = y^j$, then the number of transactions $|x_i^j|$ must be greater than* $\theta_c^1 = \frac{c^f - \frac{BR}{kl_j}}{\frac{r}{l_j} - c^v}$.
3. *If for a given processor P_i in shard j, $x_i^j \neq y^j$, then number of transactions $\left|x_i^j\right|$ must be smaller than* $\theta_c^2 = \frac{\frac{BR}{kl_j} + \frac{r|y^j|}{l_j} - c^f}{c^v}$.

Incentive-Compatible Reward Sharing

The fair reward sharing game model and its analysis offer valuable insights into designing incentive-compatible shard-based consensus protocols.

However, there are two key challenges that must be addressed before applying game-theoretic results, specifically from Theorem 4, to design such an incentive-compatible protocol.

1. How to enforce, and who will enforce, cooperation in the distributed computing environment of the protocol?

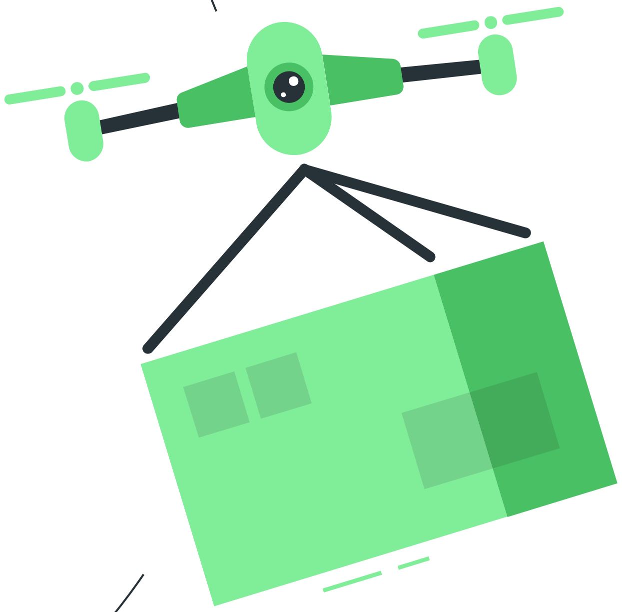
2. How can one determine the optimal strategy for a processor prior to the consensus taking place?

Incentive-Compatible Reward Sharing

- ✓ **First Challenge:** To ensure cooperation, a coordinator in each shard can guide processors on whether to cooperate in the upcoming epoch. Coordinators may be randomly selected from within the shard or a centralized trusted entity. They announce cooperation/defection decisions for each processor based on received information and enforce compliance through rewards and punishments, following the fair reward sharing strategy.

- ✓ **Second Challenge:** To efficiently obtain transaction information from processors, each processor can share a HASH of their current transaction set x_i^j with the coordinator, and determine the optimal strategy

Incentive-Compatible Protocol



Incentive-Compatible Protocol

procedure Initialization and Committee Creation

$ID, Shard \leftarrow ComputeID(epochRandomness, IP, PK)$

$x_i \leftarrow ShardTransactions(Shard)$

end procedure

Incentive-Compatible Protocol

procedure Cooperaive/Defective Processor Selection
    P_i sends $H(x_i^j)$ to *Coordinator*
    if *Coordinator* **then**
        Receive $H(x_i^j)$s
        $l_j \leftarrow$ Maximum number of processors with
            common transactions
        if $l_j < \tau$ **then**
            return $All - D$
        else
            Prepare the list of l_j processors $C_j^{l_j}$
            Calculate θ_c^1 and θ_c^2 from *Theorem 4*
            return θ_c^1, θ_c^2, and $C_j^{l_j}$
        end if
    end if
end procedure

Incentive-Compatible Protocol

procedure Shard Participation (Consensus)
  if $P_i \in C_j^{l_j}$ and $|x_i^j| \le \theta_c^1$ **then**
    return Defect
  else if $P_i \notin C_j^{l_j}$ and $|x_i^j| \ge \theta_c^2$ **then**
    return Defect
  end if
  Verify transactions and create a set of verified transactions y^j by all remaining cooperative processors
  Consensus on verified transactions
  Sign BFT agreement result
  return Signature, Agreed block's header
end procedure

Incentive-Compatible Protocol

procedure Verification, Reward, and Punishment

Verify whether $P_i \in C^L$ have cooperated in each shard

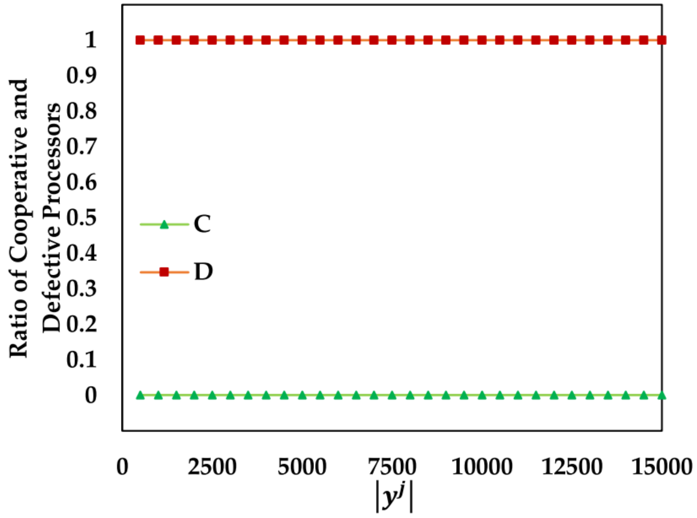
Distribute rewards among cooperative P_i

end procedure

Numerical Analysis

Number of Transactions

Effect of Varying Transaction Numbers: We analyze the impact of varying the average number of transactions $|x_i^j|$ in the range of 500 to 15,000. The corresponding ratios of cooperative and defective processors are as follows

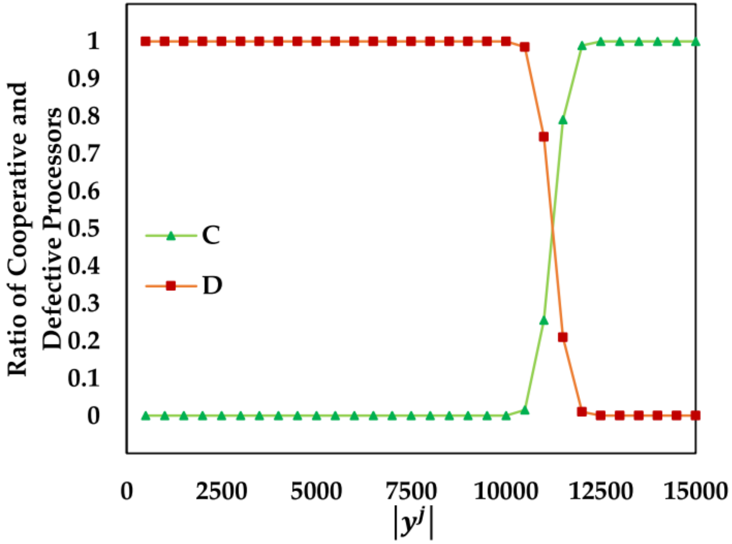


Uniform Reword Sharing

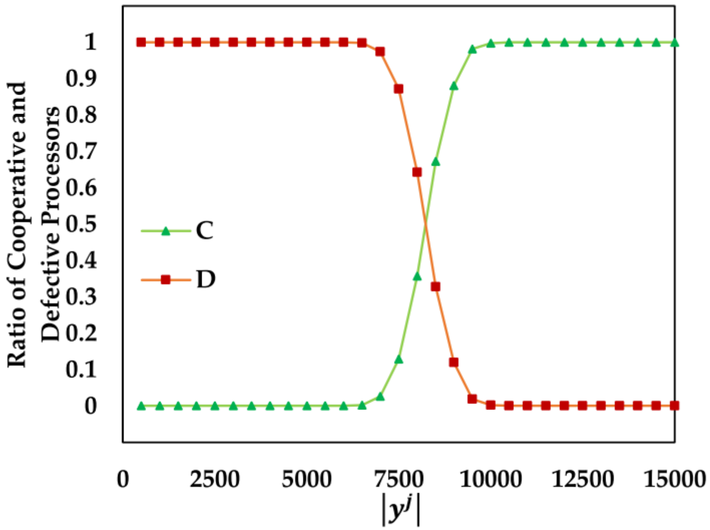
BR	1000
c^m	10
c^f	6
c^v	0.0005
r	0.1
$P(x_i^j \neq y^j)$	15%
N	$\approx$ 3000
n	$\approx$ 100
$\lvert y^j \rvert$	$\approx$ 500-15000

Simulation Parameters

Number of Transactions



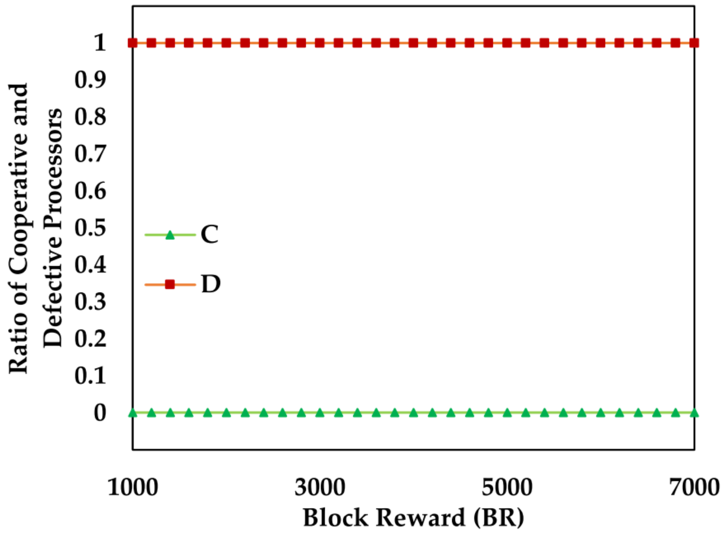
Fair Reword Sharing



Incentive-Compatible Reword Sharing

Block Rewards

Effect of Varying Block Reward: We examine the impact of varying the block reward (BR) within the range of 1,000 to 7,000, and the corresponding ratios of cooperative and defective processors are illustrated

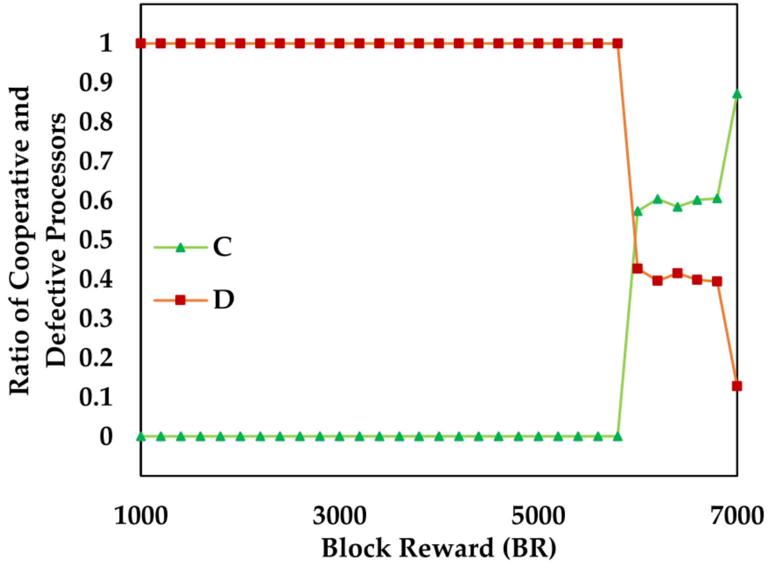


Uniform Reword Sharing

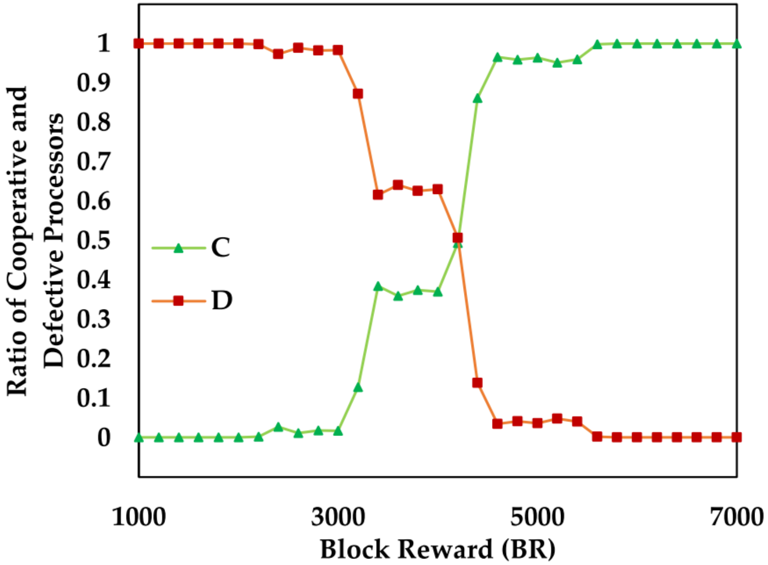
BR	1000-7000
c^m	10
c^f	6
c^v	0.001
r	0.1
$P(x_i^j \neq y^j)$	15%
N	≈ 1000
n	≈ 100
$\lvert y^j \rvert$	≈ 10000

Simulation Parameters

Block Rewards



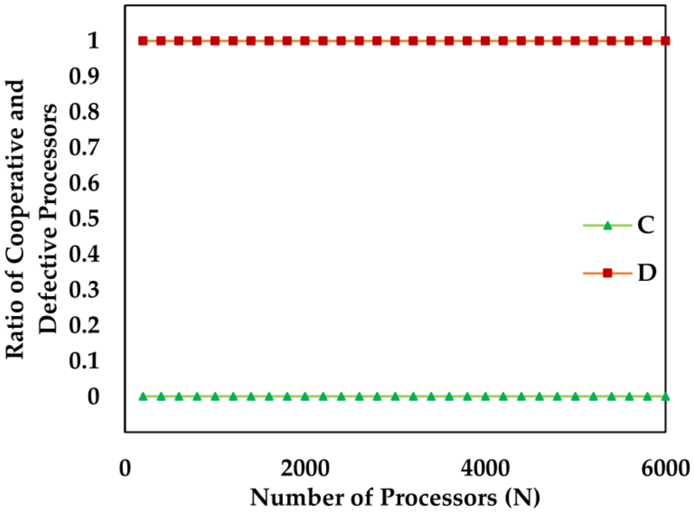
Fair Reword Sharing



Incentive-Compatible Reword Sharing

Size of The Network

Impact of Processor Count: The number of processors in the network during a given epoch significantly influences individual processor strategies. When a small reward is distributed among large number of cooperative processors, it may not cover other participation costs (e.g., c^f). This effect is observed, with N varying from 100 to 6,000.

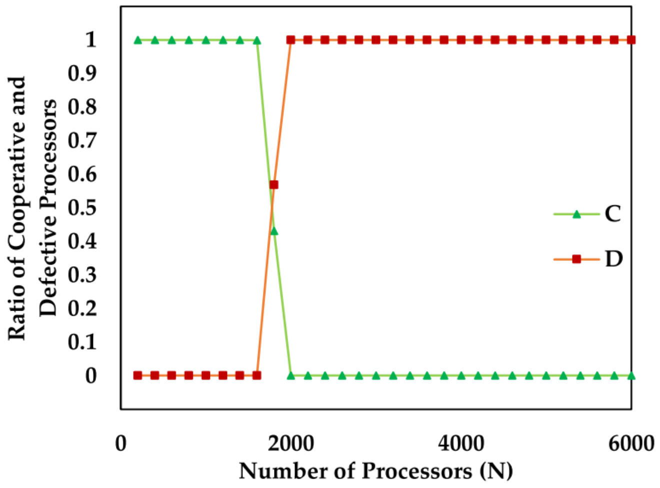


Uniform Reword Sharing

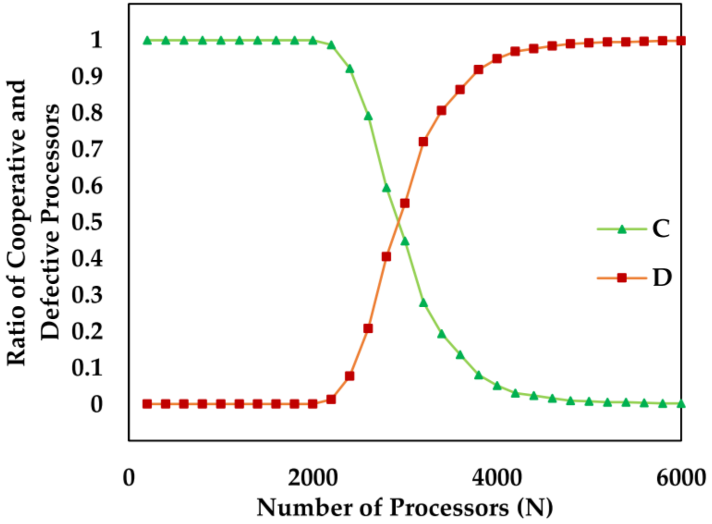
BR	10000
c^m	10
c^f	6
c^v	0.001
r	0.1
$P(x_i^j \neq y^j)$	15%
N	$\approx$ 100-6000
n	$\approx$ 100
$\lvert y^j \rvert$	$\approx$ 10000

Simulation Parameters

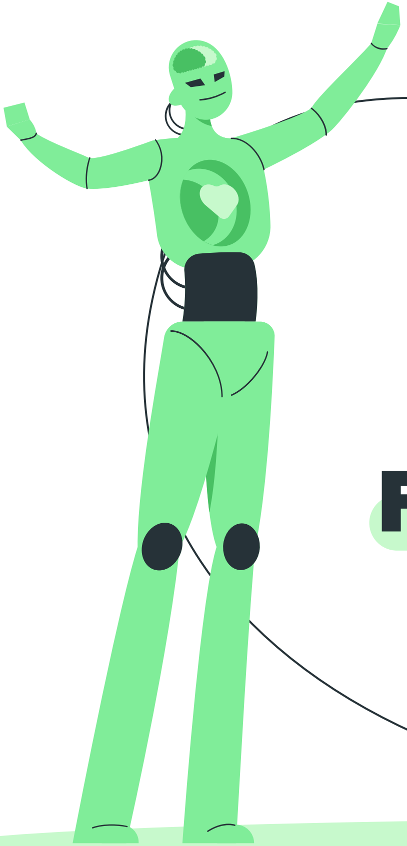
Size of The Network



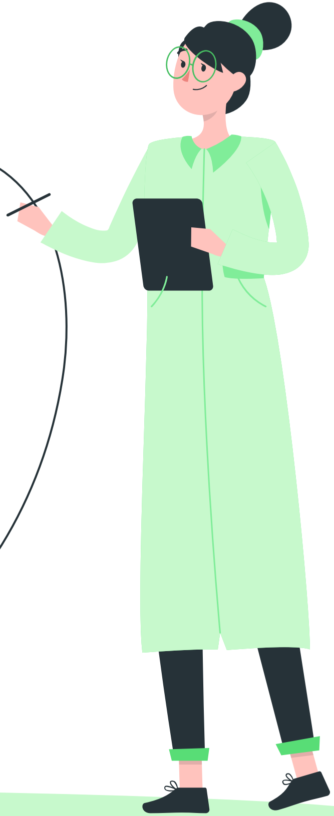
Fair Reword Sharing



Incentive-Compatible Reword Sharing



Limitations and Future Directions



Limitations

objective in this work was to create practical incentive mechanisms for encouraging cooperation in shard-based blockchains. The results presented above, both analytical and empirical, demonstrate how our proposed reward sharing mechanism successfully encourages cooperation and discourages free-riding processors.

- **Inter-Shard Communication:** Due to the absence of communication between committees, cooperative processors in a shard where consensus is reached may suffer when another committee fails to reach consensus, resulting in no block addition to the blockchain.
- **Inclusion of Malicious Processors:** In reality, malicious processors might exist, with the sole intention of disrupting the blockchain network. These malicious entities may engage in misbehavior at various protocol stages, such as providing false $H(x_i^j)$.
- **Parametric Values :** The parameters used for the numerical analysis may or may not reflect the values in a real shard-based blockchain network.

Future Directions

- Investigate the impact of inter-shard communication on processor cooperation and blockchain consensus in shard-based systems.

- Extend the analysis to include the presence of malicious processors to understand the dynamics and strategies in the presence of adversarial entities.

- Explore the effects of varying parameters dynamically over time, reflecting the changing conditions in real-world blockchain networks.

Questions

Conclusion

- ✓ We introduced a system model capturing the primary operational parameters in contemporary shard-based blockchain protocols.
- ✓ We evaluated the strategic behavior of processors in these protocols using concepts from game theory, modeling shard-based blockchain protocols as n-player non-cooperative games under various reward sharing scenarios.
- ✓ We obtained the Nash equilibria (NE) strategy profiles for each scenario.
- ✓ Based on analytical results, we designed an incentive mechanism for shard-based blockchain protocols to ensure processor cooperation by guaranteeing optimal incentive distribution.
- ✓ Our numerical analysis confirmed that the proposed reward sharing mechanism outperforms uniform reward sharing and provides stronger incentives for cooperation when the block reward or number of transactions is small.

Thanks!