# RPoA: Redefined Proof of Activity

Sina Kamali, Taha Fakharian, Mohammad Saadati, Alireza Arbabi,
Shayan Shabihi, Pouriya Tajmehrabi

September 2022

**Abstract**

Hello! is this thing working?

## 1 Introduction

The consensus protocol is the core feature of a cryptocurrency. Thus, Defining it is one of the most important tasks in designing a new network. Take Bitcoin [10] as an example; the Proof of Work [**perez2019analysis**] consensus at the heart of Bitcoin is one of the most critical factors that made it the phenomenon it is today. Bitcoin is somewhat an unflexible network at its core but has seen a series of updates in recent years [14]. Another fascinating feature of Bitcoin was its anti-inflation features, one of the most impactful of them was the impact of chain height over block rewards [4] [**aponte202151**]. Bitcoin lowers its block rewards as the height of the main chain increases, thus making the number of Bitcoins available a finite amount. Security-wise, bitcoin was resistant against up to 50% adversarial power [3], but any more than this would result in the network being insecure. This process came to be known as the 51% attack [1].

Proof of Stake is another well-known consensus used by many networks nowadays. Ethereum [**saleh2021blockchain**] [17] is the most popular chain that has PoS as the heart of its network. Ethereum is a project which attempts to build generalized technology, a technology on which all transaction-based state machine concepts may be built. There are also some other altcoins based on PoS. It is important to mention that some extensions exist on the Proof of Stake consensus [8]. Remember that, like other consensuses, some attacks on PoS [15] [11] like Reorg Attack exist. Some of these attacks have been remedied.

The Staking system is the main part of the PoS consensus. Staking locks up assets to participate and help maintain the security of that network's blockchain. In exchange for locking up assets and participating in the network validation, validators receive rewards in that cryptocurrency known as staking rewards. Our staking system is somehow similar to previous systems. The fees mentioned in this article are used as the assets in this system. This system will ensure security in the network, as well as the value of the coin.

Mainstream protocols, such as Proof of Work or Proof of Stake, still have drawbacks [13] [16]. Apart from energy consumption, other drawbacks of PoW include low throughput and decentralization. Nothing-at-stake is a major drawback of PoS, where multiple chains can be voted on by block generators because there is nothing to lose. The PoA protocol [2] is to have a decentralized cryptocurrency network whose security is based on a combination of Proof of Work and Stake. In general terms, Proof of Work-based protocols give decision-making power to entities performing computational tasks. In contrast, a Proof of Stake-based protocol gives decision-making power to entities with a stake in the system. This mechanism suffers from the issue of high energy consumption in PoW and favors the rich as in PoS [16].

This document will introduce a new protocol that tries to incorporate the best features of the said consensus protocols while introducing new features in the mix. RPoA was developed as a solution to capitalizing on mining in the blockchain networks while reducing power consumption compared to the older protocols like PoW and PoA. RPoA encourages its users to be and stay active on the network and rewards them accordingly also. Doing so results in a network where the liveliness of the network is the ultimate goal and benefits everyone.

We created this proof to try and fix some of the problems in the older consensus protocols. We aimed for accessibility while trying not to decrease security. The protocol we have created is resistant to up to 50% adversarial power and other famous attacks like the Sybil attack. We will discuss these security features thoroughly in the following sections.

In summary, our contributions include but are not limited to:

- A consensus that does not discourage new users from entering and mining
- Rewards users based on their valuable activity, rather than just the work done
- Bankrolls itself utilizing the activity of its users

Acknowledge that RPoA is a consensus protocol and thus can be used in any distributed environment that supports its needed features. Whether the said environment is a Smart Contract, standalone blockchain, or a standalone block tree, the RPoA would work flawlessly. Furthermore, many different technologies could be integrated with the RPoA for better performance and increased security(i.e., ZK-SNARKs). Also, certain features of the RPoA could be highly integrated with existing smart contracts. One such example is the REX coin.

Section 2 reviews related works. Section 3 introduces the RPoA, discusses its features, explains the mining equation, and last but not least, discusses security. Section 4 discusses possible future works and concludes the paper.

## 2 Related Works

As discussed in the previous sections, we propose a consensus that a user's mining chance is linked to their activity and functional work in the network.

Entering an established network has become more demanding with increased interest in cryptocurrency. New miners are often discouraged and leave the networks, and as a result, most networks are run by a small percentage of members.

Not much research exists about activity and valuable work as the main consensus of a decentralized network. We first inspect a previously defined proof called Proof-of-Activity [2]. As written in the paper, the PoA algorithm begins with PoW and ends with PoS. However, while mining for a block, PoA introduces a new subroutine called follow-the-satoshi. The network transforms some pseudorandom values into a satoshi (the smallest cryptocurrency unit) during this subroutine. The main problem with this PoA definition is that it leads to excessive carbon consumption, which is harmful to the environment.

Proof-of-Useful-Work has been introduced based on training a machine learning model on the blockchain. Miners get a chance to create new coins after performing honest ML training work. Clients submit tasks and pay for all training contributors. Interested parties can order, complete, and verify useful work in a distributed environment using this consensus protocol.

Recently, a new proof called Proof-of-Prestige [7] has been introduced that values the unverifiable tasks in a network. PoP introduces a new concept called Prestige, a volatile resource that, unlike coins, regenerates over time. Prestige is gained by performing "useful work" spent when benefiting from services.

We wanted to create an energy-efficient consensus protocol in which users are encouraged to enter, stay, and be active in the network to keep it alive. Also, it was important for us to design the protocol to become resistant to the most prominent attacks like Sybil and DDoS attacks. It is also important to mention that our proposed protocol supports integration into other major chain protocols.

# 3 Redefined Proof of Activity

## 3.1 Main Network Mechanisms

We propose a consensus protocol that values users based on the amount of their activity on the network. Doing so incentivizes users to stay active on the network. Thus, to track a user's activity, there must be a well-defined, measurable transaction in which a user contributes to the network in some meaningful way. This kind of transaction is called a *service transaction*. Measurability is the most crucial feature of a service transaction. One example of a *service transaction* could be a file upload transaction.

The following sections will discuss the base functionalities needed to define the protocol. More precisely, we will explain how the staking system works and how a user's activity is defined.

## 3.2 Staking System

In RPoA, users can stake their coins on the network. Staking coins does not accomplish anything of significance on this network. It merely provides the means for users to pay the upload fees without losing their money. Bare in mind that upload fees are discussed in full in a later section.

Staked coins cannot be spent or used in any way and cannot be un-staked for a certain amount of time. After which, a user can turn all of their staked coins into coins that can be traded with ease.

This staking system can bankroll the network, thus increasing the value of the network and helping it stay active and stable.

## 3.3 Activity System

### 3.3.1 Defining Activity

To define activity in the RPoA, first, we must determine what parameters a *service transaction* should have. A *service transaction* has two important factors. First, it should have measurable worth so that the system can differentiate its value based on its worth. Furthermore, it must be trackable, meaning that a *service transaction* should have a specific owner and timestamp for its creation time.

We propose an activity function computing an activity factor per *service transaction*, given the particular transaction's worth and creation time. Such a measurement has to take specific needs of the network's miners into account, As their chance of mining new blocks grows linearly with them gaining additional online activity. Lemma 1 provides further details on such a scenario.

We define the amount of a *service transaction*'s activity by defining two helping functions called $timeFactor$ and $worthFactor$. By letting $timeFactor$ be a function measuring the time passed from the creation of the transaction and $worthFactor$ a function representing the relative effect of the particular transaction's worth on the network, we define the amount of activity per *service transaction* as follows:

**Definition. 1.** *$timeFactor(stt)$ is the time factor per service transaction:*

$$timeFactor(stt) = (\frac{T}{sst + T})^r \tag{1}$$

Where $stt$ is the period passed from the creation of the *service transaction.*

**Definition. 2.** *$worthFactor(stw)$ is the worth factor per service transaction:*

$$worthFactor(stw) = (\frac{-T}{stw + \frac{T}{\beta}}) + \beta \tag{2}$$

Where $stw$ is the *service transaction*'s worth.

**Definition. 3.** *$STA(stt, stw)$ is the service transaction activity calculated per transaction:*

$$STA(stt, stw) = \alpha.timeFactor(stt).worthFactor(stw) \tag{3}$$

Here, *stt* and *stw* refer to the involved *service transaction*'s time passed from the creation and worth on the blockchain, respectively, and $\alpha$ denotes the maximum allowed activity factor assigned to a transaction at the post-time.

Note that while with $stt \to \infty$ the $timeFactor$ approaches a value of 0, with $stw \to \infty$ the $worthFactor$ approaches 1, and the cooperative effect of the two functions multiplied at $t = 0$ and $s \to \infty$ would impose a max assigned activity of 1 per transaction. The $\alpha$ variable works, so the magnitude of this maximum value could be set arbitrarily per implementation.

Also, note that in the above definition for $STA$, $\alpha$ has to be a finite value in this case, as an infinite value of $\alpha$ imposes an infinite initial activity assigned to a user at the time they make a new transaction and have it recorded on the blockchain. Such a value for $\alpha$ could be catastrophic for an RPoA network.

### 3.3.2 Cumulation of Activity

To measure the overall activity based on the user's contribution to the network, we take a summation of all the activities from the previous *service transactions* at the moment. Then, we use yielded values as a base to compute the user-specific overall mining difficulty and their dynamic fees. These use cases will be discussed in subsequent sections.

**Definition. 4.** *$activity(A_{user})$ is the total activity of a user*

$$activity(A_{user}) = 1 + \sum_{st}^{UST(A_{user})} STA(TimeOf(st), WorthOf(st)) \tag{4}$$

Here, *activity* is the user's total activity, $UST$ is the user's entire service transactions, $TimeOf$ is the time passed from the creation of a *service transaction*, $A_{user}$ is the user's address, and $WorthOf$ is the *service transaction*'s respective worth. We also add a value of 1 to the total activity, which is the base value for each user's activity by default.

## 3.4 Mining

Mining in the RPoA is done by creating a block whose hash satisfies a mining equation. Miners are responsible for mining coin transactions and *service transactions*. There is a block reward as usual, and on top of that, miner wages exist for each transaction mined to incentivize miners further to continue. Fees and miner wages are explored thoroughly in a later section.

### 3.4.1 Mining System and Mining Formula

RPoA values users based on their *activity*, meaning that it uses the *activity* of the miners as a boosting factor in its mining equation.

**Definition. 5.** *We define $H(text)$ as the hashing function we will use in our equations. We assume that $H(text)$ generates a $\theta$ bit number hash of its inputs.*

**Definition. 6.** *$D$ denotes the difficulty factor. The difficulty factor is responsible for keeping the mining frequency of the network at a constant rate (similar to the method used in bitcoin Bitcoin [2]):*

$$D = \frac{avg(TimeElapsedToMineBlock)}{ConstantDeterminedTime} \tag{5}$$

Where $avg(TimeElapsedToMineBlock)$ is the average time it takes to mine a block in the network, and $ConstantDeterminedTime$ is the desired mining rate (i.e., one block every 10 minutes).

**Definition. 7.** *The mining equation is the equation that validates a block and is defined as follows:*

$$H(header) \leq 2^{\theta-1} * D * activity(user) \tag{6}$$

Where $H(header)$ is the hash output of the block header to be mined, $D$ is the difficulty factor mentioned above, and $activity(user)$ is the cumulative activity of the mining user.

**Lemma. 1.** *Assuming UNF, the total number of previous transactions per user be distributed over $N(\frac{n_{total\_user\_transactions}}{n_{users}}, \sigma)$, the maximum activity per user is directly dependant upon the $n_{blocks}$, and inversely upon $n_{miners}$. It is also independent of the parameters of the activity function in this case. Also, based on this maximum activity per user at each time step, a fairness factor could be provided representing how fairly the mining inequality works in favor of active miners and how that could mean for the system's decentralization. Further proof is provided in Appendix A1.*

**Lemma. 2.** *Defining $dicenteralization\_level = E(\frac{total\_network\_activity}{max\_activity\_per\_miner})$ at any moment $t$, it's guaranteed that if $n_{miners}(t) \in O(n_{blocks}(t))$, the level of decentralization of the network increases over time. Further proof is provided in Appendix A2.*

## 3.5 Attacks

In this section, we discuss RPoA's security and its resilience to some famous attacks. Like many other consensus protocols, RPoA is suspectable to the 51% attack [1]. However, the power of a node cannot be defined as simple as the user's hashing power or total stake, like PoW and PoS networks, respectively. To discuss the 51% attack more thoroughly, we first must define a user's power in an RPoA-based network.

**Definition. 8.** *$P(user, t)$ is the power of a specific user at a given time:*

$$P(user, t) = \frac{hashingPower(user).activity(user)}{\sum_{u}^{allusers} hashingPower(u).activity(u)} \tag{7}$$

Although, In RPoA, the focus is on the activity of a user, hashing power still has a critical impact on one's power. While a user has a P higher than 50%, they can take control flow of the network. Thus, control the whole network [1].

The first problem we will discuss is the Nothing At Stake problem [9]. The Nothing At Stake problem is a well-known attack on the networks that incorporate a PoS consensus protocol [12]. In PoS systems, unlike in proof-of-work-based systems, it is not computationally costly for validators to add new blocks to the PoS blockchain. The nothing-at-stake problem is a theoretical security issue in PoS systems. The problem occurs anytime there is a fork in the blockchain, either because of a malicious action or accidentally when two honest validators propose blocks simultaneously. The RPoA's mining system is fundamentally different from that of PoS. In such a way that to mine a block, One needs to produce some activity and use some hashing power. Additionally, Gaining activity in the RPoA comes at a cost. As a result of this difference, the Nothing At Stake Attack doesn't apply to the RPoA. Hence, Some of the newer attacks in this area that arose from this do not apply to the RPoA either. One example of these attacks is the Avalanche Attack [11].

Next, we discuss the Sybil attack [5]. For a network to be suspectable to the Sybil attack, it has to allow the users to gain additional power by simply using a single node to simultaneously operate many active fake identities (or Sybil identities). This problem is not an issue in the PoW-based networks. Since, to use the network from several fake identities, one has to split the hashing power among the identities. Thus, their total hashing power would be lower or equivalent to their original hashing power, and as a result, they would not gain any additional power. RPoA partly uses PoW in its mining equation, but to further prevent the Sybil attack on the network, it introduces a newly defined feature called the entrance fee. The entrance fee is discussed in full in the subsequent sections. Using this feature, we can prove that a user will not have enough incentive to apply this attack to the protocol.

Lastly, we investigate the Compounding of Wealth problem [6]. The Compounding Wealth problem was first discussed as a problem in the PoS networks. This problem occurs when in a PoS network, the users with more wealth constantly get richer. Thus, gaining more power until they gain more than 51% power on the network. This problem does not apply to the RPoA explicitly, but it can be redefined as follows: "an adversary decides to place their files exclusively in their mined blocks to gain more activity. Doing so results in their activity and mining power increasing respectively and could be catastrophic for the network." We have addressed this problem utilizing another feature called *service transaction* fee. This fee requires users to pay a minimum fee to make a *service transaction*. Fees are discussed in full in the following section.

## 3.6 Upload Fees

To mitigate mentioned attacks, we define some measures to keep users from abusing the activity system in RPoA. To do so, we define fees. Users have to pay fees to be able to make *service transactions* that impact their activity. The first fee type is an entrance fee to grant users permanent access to make *service transactions* on the network. Users are also charged a dynamic fee per *service transaction*. Both of these fees are mandatory, but remember that all types of mandatory fees are staked and will not be lost. These staked values can be recovered after a certain time, thus, incentivizing the users to keep the network alive until they receive their assets back. In RPoA, users can also provide extra fees in their service transactions to encourage miners to mine their transactions faster. This kind of fee is called the *miner wage* and is not our concern.

### 3.6.1 Entrance Fee

**Definition. 9.** *$Fee_{entrance}$ generate the entrance fee of the network:*

$$Fee_{entrance}(t) = \gamma * \alpha * \sqrt{HeightOfLatestBlockAtTime(t)} \tag{8}$$

$t$ is the user registration time, $\alpha$ is the max base fee, and $\gamma$ is a constant. By using this entrance mechanism, we prevent adversaries from evading the activity factor of the *service transaction fee*. Keep in mind that by using the block height of the time of registration, entering the network becomes more challenging as time progresses, which incentivizes the users to keep the same account and thus its activity and stakes for as long as possible.

### 3.6.2 Service Transaction Fee

**Definition. 10.** *$baseFee(worth)$ is the base transaction fee of a transaction with a specific amount of worth:*

$$baseFee(worth) = \alpha \frac{worth}{MaxWorthAllowedPerBlock} \tag{9}$$

Where $\alpha$ is the max base transaction fee, *worth* is the worth of the transaction to be made, and $MaxWorthAllowedPerBlock$ is a defined *worth* capacity of a block.

**Definition. 11.** *$Fee_{transaction}$ is the function that generates the fee of a specific service transaction being made by a user:*

$$Fee_{transaction}(A_{user}, stw) = \beta * baseFee(stw).activity(A_{user}) \tag{10}$$

Where $\beta$ is a constant, *stw* is the service transaction's worth, $A_{user}$ is the user's address, $activity(A_{user})$ is the user's activity, and *baseFee* is the base transaction fee mentioned above. Based on the mentioned function, *service*

*transaction fee* is calculated based on a base fee and the user's previously gained activity. As stated earlier, a user's activity is influenced by the time interval between their *service transactions*. Hence, the more activity a user wants to achieve, the more they have to make *service transactions*, and the more they must spend. In practice, this prevents users from uploading files back-to-back

# 4 Conclusion and Future Works

In this paper, we introduced the RPoA consensus protocol. A protocol that incentivizes users to stay active on the network while not discouraging new users from joining it. One that tries to solve the power consumption problem of the PoW protocol to some extent while incorporating some of its core security measures. As mentioned, RPoA could be run by smart contracts on chains like Cardano or Ethereum, as the consensus merely needs a decentralized medium to run. Applicating the introduced consensus in this method enables us to foresee modern horizons for RPoA as the staked assets could be loaned or used in decentralized finance. The assets staked while paying the fees can be utilized significantly, reduce the non-liquidity in the network, and have other on-chain economic benefits. One possibility for the staked assets could be profit maintenance and division. As a result, activity within the chain could be translated into assets that generate daily or monthly profit(i.e., REX coin) and are tradable in the stock market /exchanges. We strongly believe that the RPoA could prove useful in many situations, including the ones mentioned.

# Acknowledgements

This work was supported and supervised by Dr. Behnam Bahrak. We want to express our gratitude to some of our colleagues who helped us in the creation of this paper: Sahar Shirmardi for her support in defining the activity equations, and Ali Ebrahimi for his assistance in developing a proof of concept based on the RPoA.

# References

[1] Fredy Andres Aponte-Novoa et al. "The 51% Attack on Blockchains: A Mining Behavior Study". In: *IEEE Access* 9 (2021), pp. 140549–140564. DOI: 10.1109/ACCESS.2021.3119291.

[2] Iddo Bentov et al. "Proof of activity: Extending bitcoin's proof of work via proof of stake [extended abstract] y". In: *ACM SIGMETRICS Performance Evaluation Review* 42.3 (2014), pp. 34–37.

[3] Usman W Chohan. "The double spending problem and cryptocurrencies". In: *Available at SSRN 3090174* (2021).

[4] Jona Derks, Jaap Gordijn, and Arjen Siegmann. "From chaining blocks to breaking even: A study on the profitability of bitcoin mining from 2012 to 2016". In: *Electronic Markets* 28.3 (2018), pp. 321–338.

[5] John R Douceur. "The sybil attack". In: *International workshop on peer-to-peer systems*. Springer. 2002, pp. 251–260.

[6] Giulia Fanti et al. "Compounding of wealth in proof-of-stake cryptocurrencies". In: *International conference on financial cryptography and data security*. Springer. 2019, pp. 42–61.

[7] Michał Król et al. "Proof-of-prestige: A useful work reward system for unverifiable tasks". In: *ACM Transactions on Internet Technology (TOIT)* 21.2 (2021), pp. 1–27.

[8] Bahareh Lashkari and Petr Musilek. "A comprehensive review of blockchain consensus mechanisms". In: *IEEE Access* 9 (2021), pp. 43620–43652.

[9] Wenting Li et al. "Securing proof-of-stake blockchain protocols". In: *Data privacy management, cryptocurrencies and blockchain technology*. Springer, 2017, pp. 297–315.

[10] Satoshi Nakamoto. "Bitcoin: A peer-to-peer electronic cash system". In: *Decentralized Business Review* (2008), p. 21260.

[11] Joachim Neu, Ertem Nusret Tas, and David Tse. "Two Attacks On Proof-of-Stake GHOST/Ethereum". In: *arXiv preprint arXiv:2203.01315* (2022).

[12] Cong T Nguyen et al. "Proof-of-stake consensus mechanisms for future blockchain networks: fundamentals, applications and opportunities". In: *IEEE Access* 7 (2019), pp. 85727–85745.

[13] Damilare Peter Oyinloye et al. "Blockchain consensus: An overview of alternative protocols". In: *Symmetry* 13.8 (2021), p. 1363.

[14] Joseph Poon and Thaddeus Dryja. *The bitcoin lightning network: Scalable off-chain instant payments*. 2016.

[15] Caspar Schwarz-Schilling et al. "Three Attacks on Proof-of-Stake Ethereum". In: *arXiv preprint arXiv:2110.10086* (2021).

[16] Yenatfanta Shifferaw and Surafel Lemma. "Limitations of proof of stake algorithm in blockchain: A review". In: *Zede Journal* 39.1 (2021), pp. 81–95.

[17] Gavin Wood et al. "Ethereum: A secure decentralised generalised transaction ledger". In: *Ethereum project yellow paper* 151.2014 (2014), pp. 1–32.

# A Proofs

## A.1 Proof of Lemma 1

First, we prove that if $UNT$ is distributed Normally having a mean of $\frac{n_{total\_user\_transactions}}{n_{users}}$, then $UNT$ is expected to grow as a linear function of $\nu = \frac{n_{blocks}}{n_{users}}$ as follows.

Observe that given the above assumptions, we have:

$$E(UNT) = \frac{n_{total\_user\_transactions}}{n_{users}}$$

Thus, assuming that we have $n_{total\_user\_transactions} = n_{blocks} * block\_size$, with a constant value for $block\_size$, we have:

$$E(UNT) = c\frac{n_{blocks}}{n_{users}}$$

Where $c$ is a constant. Thus, it is obvious that the $E(UNT)$ can be expressed as a linear function of $\nu$ with a bias of 0. Also, for the $max\_activity\_per\_user$, we have:

$$E(max\_bounded\_activity\_per\_user) = E(UNT * max\_activity\_per\_transaction)$$

From $(\text{activity}_def_eq), we have \text{max\_activity\_per\_transaction} = \alpha$, and therefore for a constant value of $\alpha$, $E(max\_bounded\_activity\_per\_user)$ is a linear function of $E(UNT)$. By linear transitivity, one can conclude $E(max\_bounded\_activity\_per\_user)$ would also be a linear function of $\nu$ in this case. As one can observe following the above-provided proofs, $E(UNT)$ is independent of the parameters provided to $(\text{activity}_def_eq) given the premise of having a Normal distribution with the discussed parameters holds.$

## A.2 Proof of Lemma 2

At any moment $t$, one can argue that the $total\_network\_activity$ is bounded by $max\_activity\_per\_user * n\_miners$. Therefore, we have:

$$total\_network\_activity \leq max\_activity\_per\_user * n\_miners$$

And setting $decentralization\_level = E(\frac{totalNetworkActivity}{max\_activity\_per\_miner})$, we would have:

$$decentralization\_level \leq E(n_{miners})$$

And so one would be able to conclude that, as expected, the level of decentralization of the network could grow based on the proportion of the network users also being active miners.