RPoA: Redefined Proof of Activity

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

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 critical tasks in designing a new network since most features of a blockchain network are tied closely to the consensus protocol, including decentralization, security, throughput, and so forth. Thus, A suitable consensus protocol can guarantee blockchain systems' fault tolerance and security.

Launched in 2009, Bitcoin [11] continues to grow rapidly. At its core, Bitcoin uses *Proof of Work* (PoW) [15] as its main consensus protocol. Bitcoin is inflation resistant, which is a direct result of block rewards decaying over time with respect to the block height [6], thus making the total amount of Bitcoins available finite. Other impressive features of Bitcoin include decentralization, provable security against most prominent attack vectors, integration with other on-chain and off-chain protocols, etc. Regarding security, Bitcoin is resistant against $> 50\%$ adversarial power commonly known as the infamous 51% attack [5] [2]. Recently, with the integration of the Lightning Network, multi-signature transactions, and Smart Contracts into Bitcoin's PoW protocol, it was expanded further.

Recently *Proof of Stake* (PoS) protocols have emerged as an energy-efficient alternative to PoW [**posat**]. Ethereum [18] used to run on PoW, but has recently transitioned to PoS, for its lower power consumption, better throughput, and more flexibility. Ethereum introduces a technology called *smart contracts*, a technology upon which all transaction-based state machines could be developed. Smart contracts let users develop decentralized applications, also known as DAPPs and DAOs. Such features let developers also build alternative blockchain-based systems on Ethereum's smart contracts. Over the years, PoS has been proved to be susceptible to many forms of alternative attacks, including the Nothing-at-Stake attack, the Reorg and Liveness attacks [16], the Avalanche attack [12].

The Staking system is a significant part of the PoS consensus protocol. Staking is used to freeze assets on the network. One use case of staking is in the PoS systems, where decision-making power is given to entities with a stake in the system. The term "validator" refers to the mentioned entities in most PoS networks. Validators collect transactions from a shared pool, cumulate them into new blocks, and provide them to the network.

Mainstream protocols such as PoW or PoS still have drawbacks [14] [17]. Apart from energy consumption, other drawbacks of PoW [1] include low throughput. Nothing-at-Stake is a major venerability of PoS, where the same stake can be used to grind on the many blocks. *Proof of Activity* (PoA) [3] is another protocol that combines PoW and PoS into a more robust and secure protocol. In PoA, the mining process begins similarly to PoW, with various miners trying to outpace each other based on their computation power to find a new block. When a new block is mined, the system switches to PoS, where a group of validators is pseudo-randomly selected to validate the block. Although PoA provides more security and decentralization than its discussed counterparts, it still suffers from issues such as high energy consumption and a high mining latency.

We introduce RPoA, a newly defined protocol that combines the best features of PoW, PoS, and PoA. RPoA gives additional mining power to users based on their accumulated activity. We define *activity* as a factor that measures the amount of service a user provides to the network. RPoA provides most of the features mentioned above while adding extra features such as reduced energy consumption, lowered hardware requirements for entry-level mining, and the same level of security as its predecessors. Furthermore, RPoA encourages users to stay active on the network, thus keeping it alive, which propels users not to leave and stay active and make more contributions to the network.

Our contributions: We propose a system that defines *activity* as a measure of user's dedication to the network, provides the same level of security with much lower power consumption, better supports dynamic availability, features less mining latency, correctly supports non-private blockchains, and requires less computational resources from new miners.

As a consensus protocol, RPoA could be suitably used within any compatible decentralized framework, potentially based on smart contracts, blockchains, or block trees. Furthermore, many different technologies including zk-SNARKs [4] and cryptographic accumulators could be integrated with RPoA for better performance and improved security.

The rest of the paper is structured as follows. Section 2 reviews related works. Section 3 introduces RPoA, discusses its features, explains the mining equation, and discusses security. Finally, section 4 discusses possible future works and concludes the paper.

2 Related Work

Not much research exists on activity and valuable work being considered the main consensus of a decentralized network. One of the major protocols that came out with a similar idea is *Proof of Useful Work* (PoUW), which encourages miners to train machine learning models in exchange for a reward. More precisely, PoUW has users pay fees for submitting their models to the network and brings miners to train such models later and serve as "useful entities" on the network. Additionally, *Proof of Prestige* (PoP) [9] has been introduced that values unverifiable tasks in the network. PoP introduces a new concept called *Prestige*, a volatile resource that regenerates over time. Prestige is gained by performing *useful work* and is spent while deriving benefits from the provided services. Other similar protocols include consensus protocols in *File-Coin*, *Golem*, decentralized mining pools like *SmartPool*, and networks that reward entities bridging on-, and off-chain protocols.

RPoA seeks to address issues of similar protocols and attempts to amend some of their major characteristics to perform better in many ways. In contrast to many other protocols, RPoA's *activity* function could be defined in several ways as applicable to each application domain and thus provides better compatibility with the currently existing chains. Counterexamples include *Proof of Useful Work*File-Coin*SmartPool*, which only define domain-specific *value* functions. In addition, compared to PoA and PoW, our proposed approach offers much lower energy consumption. Therefore, it is more scalable and environmentally friendly due to the net effect of *activity* in lowering the mining difficulty. Furthermore, our work offers more simplicity and is easier to implement than PoA and PoP, which incorporate more complicated and costly algorithms PoPPoA. RPoA also supports integration with most external systems such as 2 to 3 DeFi-based systems and other previously discussed protocols, a concept that makes RPoA flexibly applicable to be used in real-world networks.

While we address some major drawbacks of most current systems, like disregard for users' valuable work and vast energy inefficiency, we also do not devote ourselves to other principal features a consensus protocol must support. Our design additionally focuses on much easier implementation, fairer rewards by using Geometric Rewarding, better integration with currently available on- and off-chain networks, and incentivizing users to better contribute to the network over time. It also displays resistance against all prominent attack vectors, one of the main features every mainstream consensus protocol should have.

3 Redefined Proof of Activity

3.1 Main Network Mechanisms

The following sections will discuss the base subsystems for the protocol to run, including the staking and activity subsystems.

3.2 Staking System

Staking freezes assets for a set period to help bankroll the network. RPoA has users stake service fees as refundable assets on the network, which keeps the network funded. It also indirectly incentivizes users to stay more active, which helps further grow the network for an increased worth of their assets when unfrozen. In other words, there are no non-refundable service fees in the RPoA; Every paid service fee is refundable and released after a certain period.

The target network's characteristics decide how staked assets are recorded on the corresponding blockchain. For example, UTXO-based networks could assign certain transactions for staking the required fees and have nodes verify that all fees are already staked before recording *service transactions.* In contrast, networks based on smart contracts could use contracts between the two parties to help ensure the essential fees are paid in advance. Other systems could also verify staked fees similarly and thus be compatible with RPoA.

3.3 Activity System

3.3.1 Defining Activity

RPoA prioritizes users based on the amount of their activity on the network. Doing so incentivizes users to stay active on the network. Thus, to track users' activities, at least one specific transaction type with measurable worth should be defined. This type of transaction will be referred to as *service transaction* in the rest of this paper. These types of transactions should be defined network-specifically, some examples of which include *file-upload transactions* in file-sharing networks and *gamble transactions* in gambling networks.

As previously discussed, service transactions assign activities to users. Therefore, these transactions must have some essential characteristics. First, a *worth* function should be defined, measuring each transaction's relative worth in the system. In RPoA, the *activity* function computes this relative worth value per service transaction at any particular time. Second, there must be a way to track such transactions and validate every prior *service transaction* a user has made. To do so, RPoA records *service transactions* on the blockchain to be later verified by the active nodes.

The *activity* function computes the relative weight of every single *service transaction.* Let us define two helper functions, $timeFactor$ and $worthFactor$, that respectively weigh the effects of elapsed time and relative worth of each *service transaction.* We first give a formal definition for these functions and later define *activity* as computed concerning these factors.

To put a declining effect of *activity* allocated to each *service transaction*, the $timeFactor$ must decay over time. To put this element in place, we use an exponentially decaying function plateauing near zero parameterized $T \in R$ and $r \in R$

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

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

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 α denotes the maximum allowed activity factor assigned to a transaction at the post-time.

While with $stt \to \infty$ the $timeFactor$ approaches a value of 0, with $stw \to \infty$ the $worthFactor$ approaches β. Thus, the cooperative effect of the two functions multiplied at $stt \to 0$ and $stw \to W$ would impose a max assigned activity of α per transaction. The α variable works, so the magnitude of this maximum value could be set arbitrarily per implementation.

3.3.2 Cumulation of Activity

So far, we have only defined activity per *service transaction*, yet there is still a need to define the total activity for each user. There are several ways to calculate the total activity at a point in time: to take a summation or exponential average over their entire activity history. Here we opt for the former, but the latter option could also be considered depending on the characteristics of the target network. Taking this into consideration, we define the *activity* function as follows:

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 default base value for each user's activity.

3.4 Mining

Mining is the process in which a miner proposes new blocks whose hashes satisfy a criterion. More precisely, a block, in this sense, is generated via the accumulation of a series of transactions taken from a common pool. Miners are rewarded in two ways for mining each block: There is a block reward in place, and on top of that, users can put extra tips called *miner wages* in transactions similar to Bitcoin to encourage miners to prioritize them.

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 θ 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 [3]):*

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

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. *Let $\zeta(t)$ be the total number of previous transactions per user, $n_u(t)$ be denote the number of network users, and $\Omega(t)$ denote total network activity at time t. If ζ be normally 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. Further proof is provided in Appendix 1.*

Lemma. 2. *Let $\Lambda(t)$ and $\Omega(t)$ denote total network activity and the expected maximum activity per user at time t. Let $d(t)$ denote the decentralization level of the network at time point t. Provably in this case, the upper bound on $d(t)$ is directly dependent upon the number of network users. Further proof is provided Appendix 1.*

3.5 Security

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 [2]. 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 [2].

The first problem we will discuss is the Nothing At Stake problem [10]. The Nothing At Stake problem is a well-known attack on the networks that incorporate a PoS consensus protocol [13]. 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 [12].

Next, we discuss the Sybil attack [7]. 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 [8]. 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, α is the max base fee, and γ 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 α 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 β 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. Applying 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 [19]. 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)[**rex**] 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 and Shamim Nasiri 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] Lennart Ante and Ingo Fiedler. “Bitcoin’s energy consumption and social costs in relation to its capacity as a settlement layer”. In: *Available at SSRN 3910778* (2021).

[2] 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.

[3] 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.

[4] Nir Bitansky et al. “From extractable collision resistance to succinct non-interactive arguments of knowledge, and back again”. In: *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*. 2012, pp. 326–349.

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

[6] 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.

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

[8] 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.

[9] 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.

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

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

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

[13] 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.

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

[15] Cristina Pérez-Solà et al. “Analysis of the SegWit adoption in Bitcoin”. In: *URL: https://deic-web. uab. cat/˜ guille/publications/papers/2018. recsi. segwit. pdf (visited on 06/13/2020)* (2019).

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

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

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

[19] Dirk A Zetzsche, Douglas W Arner, and Ross P Buckley. “Decentralized finance”. In: *Journal of Financial Regulation* 6.2 (2020), pp. 172–203.

A Proofs

A.1 Proof of Theorem 1

In this proof let $\nu = \frac{n_b}{n_u}$. If ζ is distributed over $N(\frac{n_f}{n_u}, \sigma)$, following equation AP-a1 we get

First, we prove that if ζ is distributed over $N(\frac{n_f}{n_u}, \sigma)$, then it is expected to grow as a linear function of ν as defined in equa...

$$\nu = \frac{n_b}{n_u} \tag{11}$$

$$(UNT) = \frac{n_{total\_user\_transactions}}{n_{users}} \tag{12}$$

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 ν 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 (activity$_{d}ef_{e}q), wehave$max_activity_per_transaction $= \alpha$, and therefore for a constant value of α, $E(max\_bounded\_activity\_per\_user)$ 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 ν in this case. As one can observe following the above-provided proofs, $E(UNT)$ is independent of the parameters provided to (4) given the premise of having a Normal distribution with the discussed parameters holds.

A.2 Proof of Theorem 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.