Redefining Blockchain Resource Pricing Through Decentralized Cloud Economics

BlockApex

November 2024

1 Abstract

Ethereum's high transaction fees stem from its limited blockspace and redundancy in validation, which create economic and operational challenges. This research aims to establish an alternative approach for Nerochain, treating blockspace as a decentralized cloud service where pricing is based on a cost-plus model. NERO's prepaid blockspace model adjusts dynamically based on demand, similar to cloud service pricing. By incorporating economic theories on resource pricing, this research aims to formalize NERO's approach and examines its impact on transaction costs and validator incentives.

2 Introduction

Ethereum's high transaction costs result from replicating each transaction across multiple nodes, increasing compute, bandwidth, and storage requirements. This fee-based model compensates validators but leads to unpredictable, volatile costs for users. NERO redefines blockspace as a dynamic, prepaid service, aligning blockchain costs with the cloud pricing model.

3 Economic Framework

The pricing model can be formalized as:

$$TotalFee = BaseFee + PriorityFee$$

Base Fee ($C_{hardware}$) represents the foundational cost of core blockchain operations, including execution, propagation, and replication, directly tied to the hardware resources required for these tasks. Priority Fee ($M_{dynamic}$), which is analogous to the Management Fee in centralized systems, reflects the dynamic scaling of resources to meet real-time application demand, optimizing network latency and throughput. The Priority Fee captures the economic impact of on-demand resource allocation, a critical component in adapting to fluctuating network activity. This enables a flexible infrastructure where validator nodes dynamically scale their computational resources. Formally, this adjustment can be represented by:

$$M_{dynamic} = \alpha \cdot f(D)$$

where α represents the elasticity coefficient for scaling resources, and D is the variable application demand.

4 Prepaid Blockspace and Validator Incentives

NERO's prepaid blockspace model diverges from traditional staking by allowing developers to prepay for resources with NERO tokens. This Reverse Proof of Stake (R-PoS) front-loads network investment, creating stable revenue for validators and reducing fee spikes. For a validator count N, the total prepaid blockspace cost Cprepaid for a set transaction volume Q over time T is:

$$C_{prepaid} = Q \cdot \frac{1}{N} \cdot (C_{transaction} + Premium)$$

This predictable revenue model promotes economic efficiency by incentivizing validators to optimize operations around stable, prepaid transactions.

5 Impact on Transaction Economics

NERO's elastic blockspace model effectively reduces social costs by introducing a pay-as-you-go structure. By allowing app developers to prepay for blockspace, demand becomes more predictable, which in turn reduces the variability in transaction costs for end-users. Unlike Ethereum's fee-based pricing—where high fees are necessitated by limited blockspace—NERO employs demand elasticity principles. This approach, governed by the law of diminishing marginal costs, helps maintain lower fees as validator resources scale. In practice, NERO's model ensures that the total cost for users (even with replication across 21 validators) remains below Ethereum's fee threshold, represented as:

$$\sum_{i=1}^{N} C_{user} \leq C_{Ethereum}$$

This controlled pricing supports scalable, cost-efficient transactions while maintaining network security, making the model an economically sustainable alternative to traditional blockchain fee structures.

6 Notes

6.1 Reverse Proof of Stake and App-Driven Blockspace Demand

6.1.1 Core Idea

- App developers prepay for blockspace by buying NERO tokens.
- Prepaid blockspace gives developers control, empowering developers to manage demand and revenue creating an ecosystem where users actively reimburse.

6.1.2 Incentive Shift

- Instead of traditional staking, app builders front-load their investment in network resources, profit motives align with efficient blockspace usage and transaction optimization.

6.1.3 Impact

- This sets up an efficient pay-as-you-go model where the demand for blockspace is closely tied to application usage rather than relying solely on fluctuating transaction fees.

6.2 Dynamic Blockspace Allocation and Elastic Validator Infrastructure

6.2.1 Elastic Infrastructure

Validator nodes can scale their physical resources (e.g., CPU, memory) as application demand grows, leading to an elastic blockspace that adapts based on prepaid usage.

6.2.2 Technical Solution

Validators can expand vertically, scaling their hardware to process increased transaction volume. This approach requires robust, protocol-level support for:

- Efficient transaction prioritization without solely prioritizing fees (similar to Avalanche's reputation-driven selection).
- Parallel processing capabilities, leveraging FuelVM's PSI model (parallelization, state minimization, and interoperability).

Outcome: Validators gain rewards that scale with demand, ensuring they are not resource-constrained by fixed blockspace but instead can optimize their infrastructure to support real-time needs.

6.3 Role of FuelVM and Parallel Execution for Infrastructure Optimization

6.3.1 FuelVM

Parallel execution allows validators to process transactions simultaneously.This ensures that hardware scaling by validators is matched by software efficiency, removing bottlenecks from single-threaded operations.

6.3.2 Problem Solving

This setup addresses the mismatch between high-performing hardware and traditional single-threaded VM models. FuelVM enables NERO to execute transactions in parallel, which is essential for the high throughput necessary to support fat applications and complex, computation-heavy DApps.

6.3.3 Enhanced Throughput

Validators using FuelVM can optimize state access lists and transaction dependencies in real-time, leading to vastly increased compute throughput.

6.4 Specialized Blockspace Partitioning: Hotspace for High-Priority and Socially Necessary Transactions

6.4.1 Hotspace Concept

Allocate a section of blockspace (hotspace) specifically for high-priority actions like MEV arbitrage, market-balancing transactions, and bug fixes. This area would operate as an auction space where prioritized transactions can gain expedited processing.

6.4.2 Utility and Fairness

This partition reduces contention for general blockspace by separating transactions that are essential for network health or social good.

6.4.3 Implementation

Validators can dynamically allocate resources to this partition, ensuring that high-priority transactions have dedicated processing power, which supports network stability during high-demand periods.

6.5 Redefining Blockspace 2.0 as a Decentralized Cloud Service

6.5.1 Blockchain as a Cloud Database

Position NERO as a decentralized cloud service where developers can calculate transaction costs like those of a database operation on AWS. This model, which we'll test by simulating an AWS instance running a geth client, would strip down blockchain operations to base-level costs—execution, propagation, and replication—separating NERO from "managed" blockchains with embedded gas market models.

6.5.2 Managed Services

NERO can eventually introduce managed services that optimize blockspace usage and offer value-added services (e.g., analytics, API gateways, priority queues), incentivizing validators by enabling them to charge for additional services on top of basic blockspace access.

6.5.3 Simulation and Hypothesis Testing

Simulate blockspace costs on an AWS node to test if a transaction executed on a decentralized platform can be competitive with traditional cloud services. This approach could validate cost-effectiveness compared to full mainnet deployment costs, with replication and base-level computation as primary cost drivers.

6.6 Next Steps: Detailed Analysis Points

6.6.1 Quantitative Modeling of Validator Economics under Elastic Blockspace

How does validator income scale as blockspace demand fluctuates? We need a model to calculate expected validator income under varying levels of hardware scaling, demand variability, and app-driven transaction frequency.

6.6.2 Testing FuelVM's Impact on Throughput and Parallel Execution

Perform theoretical and, if feasible, empirical testing of FuelVM's throughput gains. Establish benchmarks against traditional VM models (e.g., EVM) to quantify throughput improvements in NERO's elastic blockspace.

6.6.3 Dynamic Blockspace and Hotspace Performance Under Load

Simulate transaction prioritization and performance under varying load conditions, particularly focusing on the hotspace auction for critical transactions and the elastic scalability of the general blockspace. Examine the potential impact of this structure on latency, especially under high-demand scenarios.

6.6.4 Cost Comparison Simulation on AWS (or Similar Cloud Infrastructure)

Set up a controlled test environment on AWS to simulate blockspace operations, measuring direct compute costs, storage, and network usage. Use these results to define baseline costs and validate the hypothesis of blockspace cost optimization.