

Empirical Study of Gas-Proportional Token Emission: Revisiting the Robonomics XRT Mechanism Eight Years After Deployment

Sergey Lonshakov¹, Alexander Krupenkin¹, Claude²

¹ Robonomics Network — architects | ² Anthropic — AI research assistant

February 2026

Abstract

We present an empirical study of the XRT token emission mechanism deployed as part of Robonomics Network v5 on Ethereum mainnet. Originally designed in 2018 when Ethereum gas prices ranged from 20 to 200 gwei, the mechanism ties token emission to a Smoothed Moving Average (SMMA) of observed transaction gas prices — an innovative approach that creates a self-regulating economic feedback loop between network utilization cost and token supply. With Ethereum gas prices having decreased by two orders of magnitude to ~0.25 gwei by 2026, we conducted a controlled experiment: mining 24,190 XRT tokens through 1,360 liability contracts, including a two-phase “Pump & Mine” strategy to study the SMMA’s reflexive dynamics. Our findings illuminate the elegant design principles behind gas-proportional emission, demonstrate the SMMA’s properties as an endogenous price oracle, and provide empirical data on the reflexive dynamics between emission rate, token supply, and market microstructure on Uniswap V2. This work contributes to the broader understanding of mechanism design in early Ethereum-era smart contract systems and the long-term behavior of on-chain economic primitives.

Keywords: token emission, mechanism design, Robonomics, XRT, SMMA, smart contracts, Ethereum, cyber-physical systems, DeFi

1. Introduction

Ethereum, since its conception by Buterin in 2013, has been described as a “world computer” — a Turing-complete, decentralized execution environment where smart contracts serve as programs running on shared global infrastructure. Unlike traditional computing platforms where applications can purchase dedicated resources (servers, bandwidth, CPU time), all programs on the Ethereum Virtual Machine (EVM) compete for the same finite resource: block gas. This raises a fundamental and surprisingly underexplored question: **how can a specific program — a group of interrelated smart contracts — ensure competitive advantage in the shared throughput of the network?**

In traditional computing, the answer is straightforward: applications acquire dedicated capacity. On a shared blockchain, no such option exists natively. Every program pays the same market gas price, and no contract system can guarantee priority access to block space. The Robonomics Network, conceived by Lonshakov, Krupenkin et al. [1], proposed what may be the only systematic attempt to answer this question on Ethereum L1: **a native token (XRT) whose emission is directly proportional to the gas consumed by the protocol’s smart contracts.** This creates an economic feedback loop where the cost of operating the Robonomics program on the world computer is internalized into the token supply itself — providers who spend gas to run the protocol receive newly minted XRT in proportion to their contribution. The emission formula embodies the whitepaper’s principle: “*Emission of 1 Wn = 1 gas*

utilized by Robonomics” [1].

Drawing on Wiener's cybernetics [2], Coase's theory of the firm [3], and the Robot Operating System (ROS) [4], Robonomics proposed a decentralized marketplace where autonomous robots and sensors participate in economic transactions through smart contract liabilities. The emission mechanism — an SMMA (Smoothed Moving Average) of observed gas prices modulating token issuance — represents a pioneering attempt at on-chain incentive alignment between protocol usage and token supply.

In this paper, we revisit this mechanism eight years after deployment, under conditions radically different from those assumed at design time. We approach this study not as a critique but as an empirical investigation of a pioneering mechanism design, seeking to understand how carefully constructed economic invariants behave when a key environmental parameter — the price of gas on Ethereum — has changed by approximately 100x.

1.1. Contributions

1. Empirical characterization of the SMMA-based emission mechanism under ultra-low gas conditions (0.2–10 gwei), with data from 1,360 on-chain liability contracts;
2. Experimental validation of a two-phase SMMA manipulation strategy (“Pump & Mine”), demonstrating the reflexive properties of the endogenous gas price oracle;
3. Quantitative analysis of market microstructure effects when mining-derived tokens are sold on thin Uniswap V2 liquidity;
4. Open-source tooling for automated Robonomics liability lifecycle management (xrt-classic-miner);
5. Design analysis of early Ethereum-era mechanism engineering, contributing to the historical record of smart contract architecture.

1.2. Related Work

Token emission mechanisms span a wide design space. Bitcoin uses a deterministic halving schedule [9]; Ethereum PoS ties issuance to staking participation [7]; Filecoin uses baseline minting proportional to storage growth [10]. The Robonomics approach is distinct: emission tied to the **computational cost of protocol operations as measured by EVM gas** — creating a direct feedback loop between emission and utilization that is unique in the literature.

The SMMA mechanism relates to on-chain price oracles. Uniswap V2's TWAP oracle [5] shares the principle of smoothing over time to resist manipulation. The Robonomics SMMA predates widespread TWAP adoption and represents an early example of endogenous oracle design.

2. Background: The Robonomics Economic Architecture

2.1. The Vision: An Economy of Machines

Robonomics proposes that autonomous cyber-physical systems can participate in economic transactions as independent agents. In the whitepaper's formulation, a CPS is analogous to a firm in Coasean economics — “*a closely connected network of sensors and actuators capable of organized collaboration*” [1]. The platform distinguishes between **promisees** (demand side), **promisors** (CPS agents offering services), and **providers** (infrastructure operators relaying transactions). The liability contract — a binding agreement encoded in a smart contract — serves as the primitive for machine-to-machine economic interaction.

2.2. The Contracts Factory

The Factory contract (0x7e384...1225) serves as the central registry, creating lightweight liability contracts via DELEGATECALL — a design pattern that saved 30–40% gas compared to full contract

deployment [1]. The Factory verifies deferred signatures from both demand and supply sides, deploys a new liability contract for each matched pair, tracks cumulative gas consumption via `totalGasConsumed`, and manages the emission parameter `gasPrice` (the SMMA).

2.3. The Emission Formula

The XRT emission mechanism, implemented in `Factory.wnFromGas()`, converts gas consumption into token issuance:

$$wn = gas \times gasPrice_SMMA \times 10^9 / finalPrice_auction$$

where `wn` is emission in Wiener units ($1 \text{ XRT} = 10^9 \text{ wn}$), `gas` is gas consumed by the transaction, `gasPrice_SMMA` is the Smoothed Moving Average of observed gas prices, and `finalPrice_auction` is the price from the Dutch auction at TGE.

The SMMA updates with each `createLiability` and `finalizeLiability` call:

$$gasPrice[n+1] = (gasPrice[n] \times 999 + tx.gasprice) / 1000$$

This formula has a half-life of ~693 observations ($\ln(2) \times 1000$) and converges exponentially toward the prevailing `tx.gasprice`. The denomination system — wiener (1), coase (10^3), glushkov (10^6), robonomics token (10^9) — pays homage to Norbert Wiener (cybernetics), Ronald Coase (transaction cost economics), and Viktor Glushkov (Soviet cybernetics pioneer).

2.4. The Epoch System

The development period is divided into five epochs, each consuming a target of 3.47×10^{12} gas. The emission multiplier decreases geometrically:

| Epoch | Target gas | Multiplier | Emission rate |
|-------|-----------------------|-------------------|---------------|
| 0 | 3.47×10^{12} | $(2/3)^0 = 1.000$ | Full rate |
| 1 | 3.47×10^{12} | $(2/3)^1 = 0.667$ | 66.7% |
| 2 | 3.47×10^{12} | $(2/3)^2 = 0.444$ | 44.4% |
| 3 | 3.47×10^{12} | $(2/3)^3 = 0.296$ | 29.6% |
| 4 | 3.47×10^{12} | $(2/3)^4 = 0.198$ | 19.8% |

At the time of our experiment, the system was at 0.17% of epoch 0, with approximately 163,800 mining rounds remaining before the first multiplier reduction.

2.5. The Lighthouse System

Lighthouses are autonomous coordination contracts managing provider access through a round-robin quota mechanism with XRT staking. Providers stake XRT to participate, take turns submitting transactions, and if the active provider fails within the timeout window, the next provider claims the marker. This pattern anticipated later proof-of-stake rotation mechanisms and solves the decentralized coordination problem without a central scheduler.

The architecture bears a notable resemblance to the 0x Protocol [11], developed in the same period (2017–2018). Both systems share a fundamentally similar pattern: off-chain message matching with on-chain settlement. In 0x, makers and takers exchange signed ask/bid orders off-chain, with final settlement executed on Ethereum by “relayers.” In Robonomics, promisees and promisors exchange signed demand/offer messages off-chain, with providers (lighthouse operators) submitting the matched pair to Ethereum for on-chain liability creation. Both protocols recognized that Ethereum's limited throughput and high latency made it unsuitable for real-time order matching, and both arrived at similar architectural solutions.

In this sense, the lighthouse system can be understood as one of the **earliest attempts to build what would later be called a Layer 2 solution** — not in the rollup sense, but as a complementary off-chain communication layer that extends Ethereum with capabilities the base chain cannot efficiently provide (real-time demand/offer matching, provider coordination), while settling only the economically meaningful outcomes (liability creation and finalization) on L1. This design philosophy — off-chain where possible, on-chain where necessary — anticipated the broader industry's evolution toward L2-centric architectures by several years.

2.6. Design Rationale

The whitepaper articulates the core principle: *“The cost of 1 Wn must cover the costs of the provider for the disposal of 1 unit of gas in Ethereum”* [1]. The gas-proportional emission creates a self-regulating system: when gas is expensive, providers spend more ETH per liability and emission compensates proportionally; when gas is cheap, emission decreases proportionally. The SMMA smooths short-term volatility, preventing flash-manipulation. This was calibrated for 2018 conditions, where 2 gwei was identified as “the minimum competitive price in the Ethereum network” [1].

3. Experimental Methodology

3.1. Tool Architecture

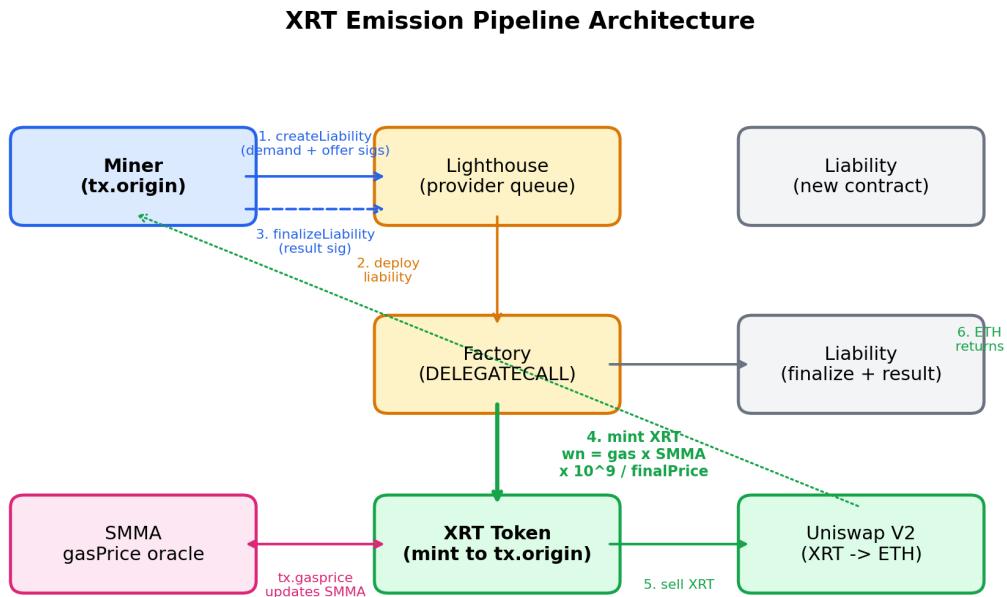


Figure 1. XRT emission pipeline architecture — from liability creation through finalization and XRT minting to Uniswap V2 sell.

We developed `xrt-classic-miner`, a Python CLI tool implementing the complete Robonomics liability lifecycle. The tool consists of four modules: `abi.py` (minimal contract ABIs and mainnet addresses), `signer.py` (EIP-191 demand/offer/result message construction), `miner.py` (XRTMiner class with lighthouse management and mining loops), and `__main__.py` (CLI interface via Click).

Signature construction follows the Robonomics v1.0 format. Demand and offer messages are 268 bytes each, encoded with `abi.encodePacked` and signed via EIP-191. A critical implementation detail: demand messages place `lighthouse` before `validator` in the field order, while offer messages invert this — a deliberate asymmetry that prevents cross-role replay attacks.

3.2. Pipeline Mining Mode

Pipeline mode achieves maximum throughput by overlapping finalization of round N with creation of round N+1. This requires 2x quota on the lighthouse but doubles throughput compared to sequential operation. The sequence is:

```

Round 1: --- CREATE(batch) -----> [bootstrap]
Round 2: FINALIZE(batch) + CREATE(batch) > [pipeline, 2N quota]
Round 3: FINALIZE(batch) + CREATE(batch) > [pipeline]
...

```

Transactions are pre-signed with sequential nonces and broadcast rapidly, allowing multiple operations in the same Ethereum block.

3.3. Experimental Setup

| Parameter | Value |
|-----------|------------------|
| Network | Ethereum Mainnet |
| Account | 0x6EFBA8...C3Ad |

| Parameter | Value |
|---------------------|---|
| Lighthouse | 0x04C672...bbeE (custom, timeout=1 block) |
| RPC Provider | DRPC (free tier) |
| Starting ETH | 3.544 |
| XRT Staked | 112 wn |
| Ethereum base fee | 0.19 - 0.29 gwei |
| ETH/USD (Chainlink) | ~\$1,944 |

3.4. EIP-1559 and tx.gasprice Semantics

A critical subtlety concerns the interaction between the post-EIP-1559 fee model and the Robonomics SMMA, designed for the pre-EIP-1559 gas price model. After EIP-1559 [6], the fee structure changed to `baseFee + priorityFee`, but the EVM opcode `GASPRICE` (Solidity's `tx.gasprice`) returns the **effective gas price** = `baseFee + priorityFee`. Thus the SMMA continues to receive a meaningful signal, but the semantics have shifted: the base fee is now set by network demand, not sender choice.

With 2026 base fees of 0.19–0.29 gwei, the priority fee dominates: during our pump phase (10 gwei priority), it constituted 98% of the effective gas price. This means the miner has almost **complete control** over the SMMA signal — a situation that differs fundamentally from 2018, where `tx.gasprice` needed to be competitive with other users' bids.

3.5. Experimental Protocol

Phase 1 — Pump (SMMA Inflation): Priority fee 10 gwei (effective `tx.gasprice` ~10.2 gwei), batch size 56 liabilities per round, budget 2.0 ETH. Objective: rapidly increase SMMA from ~1.03 gwei to maximum achievable level.

Phase 2 — Mine (Elevated Emission Harvesting): Priority fee 1 gwei (effective `tx.gasprice` ~1.2 gwei), batch size 10 (reduced from 56 due to RPC reliability constraints), budget 1.0 ETH, max cost \$0.50 per liability, auto-sell every 2,000 XRT via Uniswap V2 (5% slippage tolerance).

3.6. Parameter Tuning and Iteration

The final parameters emerged from iterative trial and adjustment, revealing practical constraints that theoretical analysis alone would not predict.

Batch size: Initially `batch=56` (maximum for lighthouse quota). Worked for pump phase, but at lower priority fees caused cascading RPC timeouts — the free-tier DRPC provider could not track 112 pending transactions simultaneously. Progressive reduction: $56 \rightarrow 20 \rightarrow 10$, with `batch=10` proving stable across 69 rounds with zero errors.

Priority fee: 10 gwei (pump: max SMMA growth, ~0.57 ETH/round); 2 gwei (attempted: RPC timeouts at round 1); 1 gwei (final: stable, ~0.012 ETH/round); 0.2 gwei (theoretical: +73 rounds, but SMMA eventually decays below breakeven).

Lighthouse timeout: `timeoutInBlocks=1` — the minimum possible. Eliminated waiting periods between rounds. In production Robonomics, longer timeouts provide fairness among multiple providers; for our single-provider experiment, minimum timeout maximized throughput.

4. Results

4.1. Phase 1: SMMA Pump

The pump phase completed 4 rounds of 56 liabilities each (448 contract calls), consuming 2.27 ETH in gas:

| Metric | Value |
|--------------------|--------------------------|
| SMMA before | 1.03 gwei |
| SMMA after | ~4.0 gwei |
| SMMA amplification | x3.88 |
| Rounds completed | 4 |
| Contract calls | 448 |
| Gas consumed | ~2.27 ETH |
| XRT minted (pump) | ~15,519 XRT |
| XRT sold post-pump | ~16,899 XRT -> 0.763 ETH |

The SMMA responded as predicted by the convergence formula. Each call moved SMMA by $(10.2 - \text{SMMA}) / 1000$. The exponential approach to the target value is visible in Figure 2.

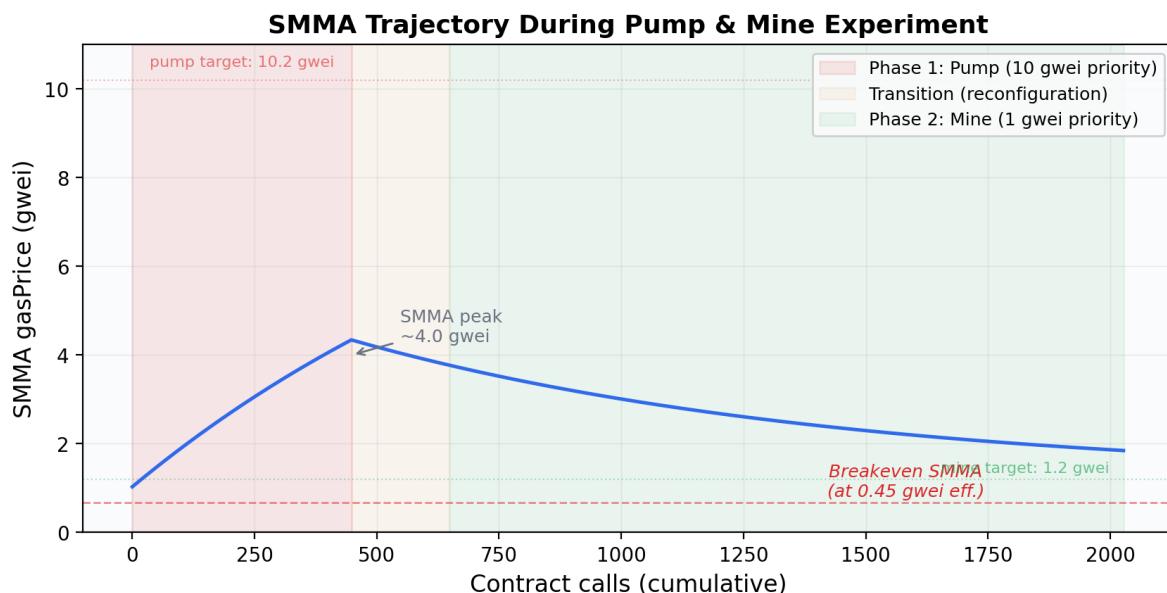


Figure 2. SMMA trajectory during the experiment. Pump phase (red) raised SMMA from 1.03 to ~4.0 gwei via 448 calls at 10.2 gwei. Mine phase (green) shows exponential decay toward the 1.2 gwei effective gas price.

4.2. Phase 2: Mine

After transitioning to 1 gwei priority fee, the mine phase ran 69 rounds with zero errors:

| Metric | Value |
|-----------------------|------------|
| Rounds completed | 69 |
| Liabilities created | ~1,360 |
| Total XRT minted | 24,190 XRT |
| Average XRT per round | 355.73 |

| Metric | Value |
|--------------------|------------|
| XRT in first round | 443.66 |
| XRT in last round | 257.13 |
| Emission decline | -42.0% |
| Total gas consumed | ~729 M gas |
| Errors | 0 |

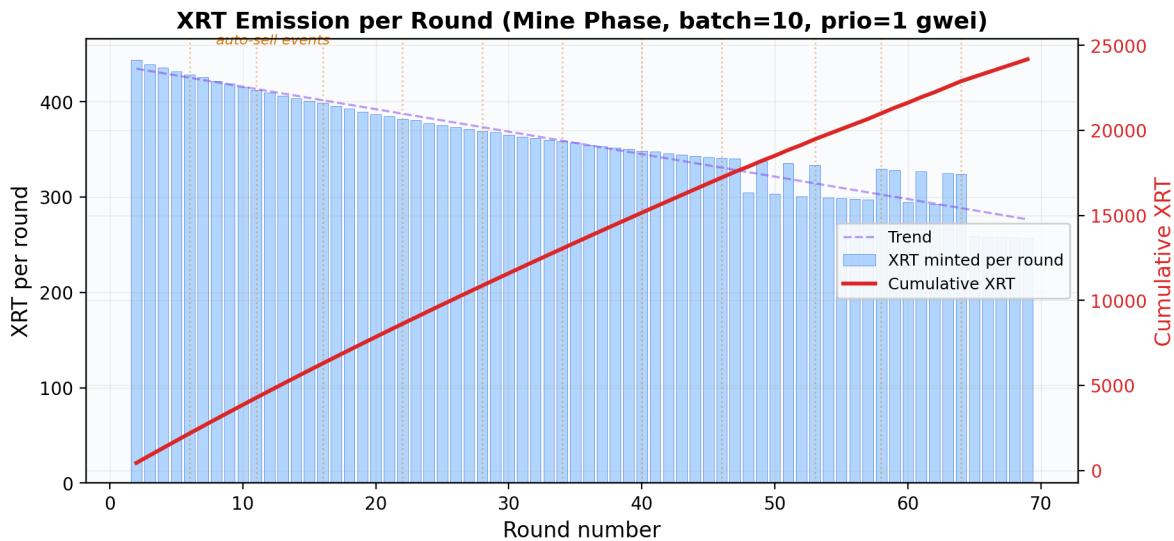


Figure 3. XRT emission per round (mine phase). Bars show per-round minting; red line shows cumulative total. Vertical dotted lines mark auto-sell events.

4.3. SMMA Decay Dynamics

The SMMA decayed from ~1.94 gwei (at mine start, after transition period losses) toward the effective gas price of ~1.2 gwei. With 20 calls per round (10 create + 10 finalize), the SMMA decreased approximately 2% per round:

$$\text{SMMA}[n+1] = \text{SMMA}[n] \times (999/1000)^{20} + 1.2 \times (1 - (999/1000)^{20})$$

The empirical half-life was approximately 35 rounds, consistent with the theoretical prediction of $\ln(2) \times 1000 / 20 = 34.7$ rounds for a period-1000 SMMA with 20 updates per round.

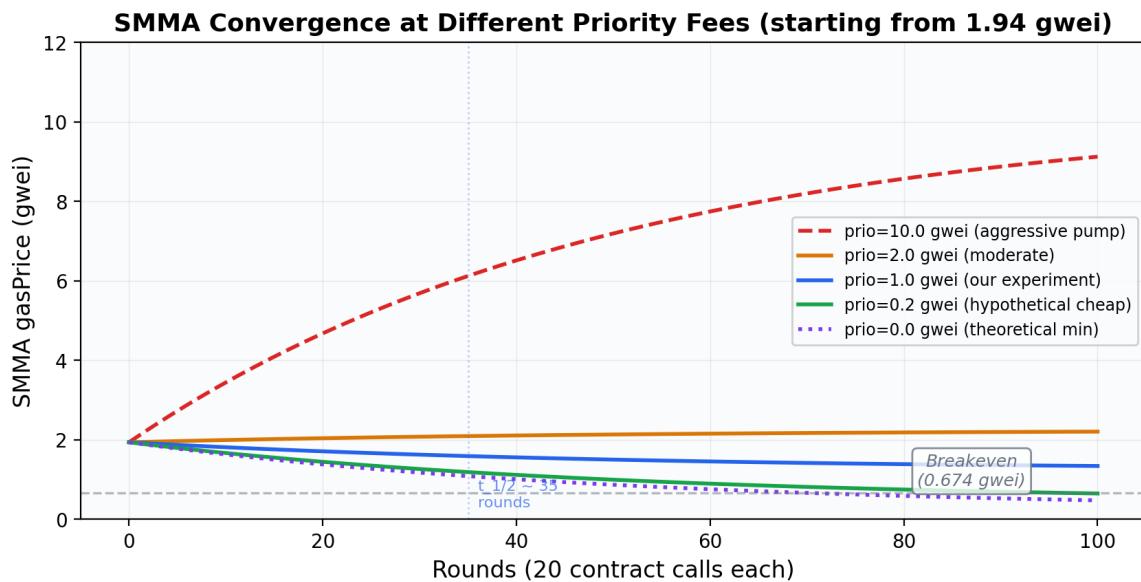


Figure 4. SMMA convergence at different priority fees, starting from 1.94 gwei. Higher priority → SMMA rises; lower priority → SMMA decays. Horizontal dashed line marks the breakeven SMMA at 0.674 gwei.

4.4. Market Impact on Uniswap V2

Eleven automated sell events over ~75 minutes revealed consistent price degradation:

| Sell # | XRT Sold | ETH Received | Price (uETH/XRT) | Delta |
|--------|----------|--------------|------------------|--------|
| 1 | 4,801 | 0.1616 | 33.65 | --- |
| 2 | 2,094 | 0.0691 | 33.01 | -1.9% |
| 3 | 2,019 | 0.0659 | 32.63 | -1.2% |
| 4 | 2,331 | 0.0758 | 32.52 | -0.3% |
| 5 | 2,248 | 0.0723 | 32.18 | -1.0% |
| 6 | 2,177 | 0.0692 | 31.78 | -1.2% |
| 7 | 2,116 | 0.0665 | 31.40 | -1.2% |
| 8 | 2,064 | 0.0642 | 31.11 | -0.9% |
| 9 | 2,256 | 0.0694 | 30.76 | -1.1% |
| 10 | 2,146 | 0.0652 | 30.39 | -1.2% |
| 11 | 2,044 | 0.0615 | 30.07 | -1.1% |
| Total | 26,296 | 0.8406 | 31.97 (avg) | -10.6% |

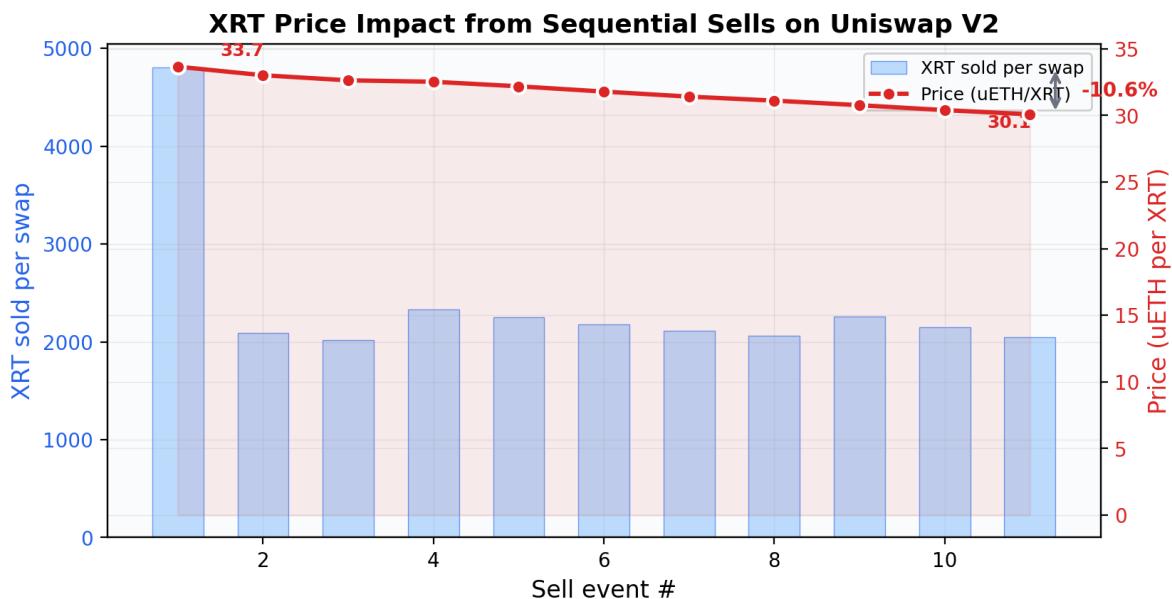


Figure 5. XRT price impact from 11 sequential sells on Uniswap V2. Blue bars: volume sold; red line: realized price per XRT. Total decline: 10.6% over ~75 minutes.

The approximately linear price decay (~1% per sell event) is consistent with the constant-product AMM model of Uniswap V2 [5], where each trade shifts the reserve ratio. With thin liquidity (~25 ETH in the pool), even modest sell volumes create measurable impact.

4.5. Financial Summary

| Item | ETH | USD |
|------------------|--------|----------|
| Starting balance | 3.544 | \$6,892 |
| Gas spent (pump) | -2.270 | -\$4,413 |
| Gas spent (mine) | -0.840 | -\$1,634 |

| Item | ETH | USD |
|-------------------------|--------|----------|
| XRT sold (total) | +1.604 | +\$3,118 |
| Remaining XRT (675 XRT) | +0.020 | +\$39 |
| Final balance | 3.182 | \$6,186 |
| Net result | -0.342 | -\$665 |

5. Analysis and Discussion

5.1. The SMMA as an Endogenous Price Oracle

The central insight from this experiment is that the `gasPrice` SMMA in the Factory contract functions as an **endogenous price oracle** — a mechanism that derives its signal entirely from the behavior of system participants rather than from an external data feed. This design choice reflects a principled decision by the Robonomics architects: rather than introducing oracle risk (the dependency on an external, potentially manipulable data source), the emission mechanism directly observes the cost that providers actually pay to operate the system.

The elegance of this approach is that it creates a **truthful mechanism** under the original design assumptions. When gas costs are a meaningful expense for providers (as they were at 20–200 gwei in 2018), `tx.gasprice` is a credible signal of the marginal cost of network operation. The emission formula then ensures that providers are compensated proportionally to their actual costs. Our experiment demonstrates that this mechanism continues to function correctly in a mathematical sense — the SMMA converges as predicted, emission responds proportionally — even under conditions vastly different from the design era.

5.2. The Reflexive Feedback Loop

Reflexive Incentive Cycle in XRT Emission

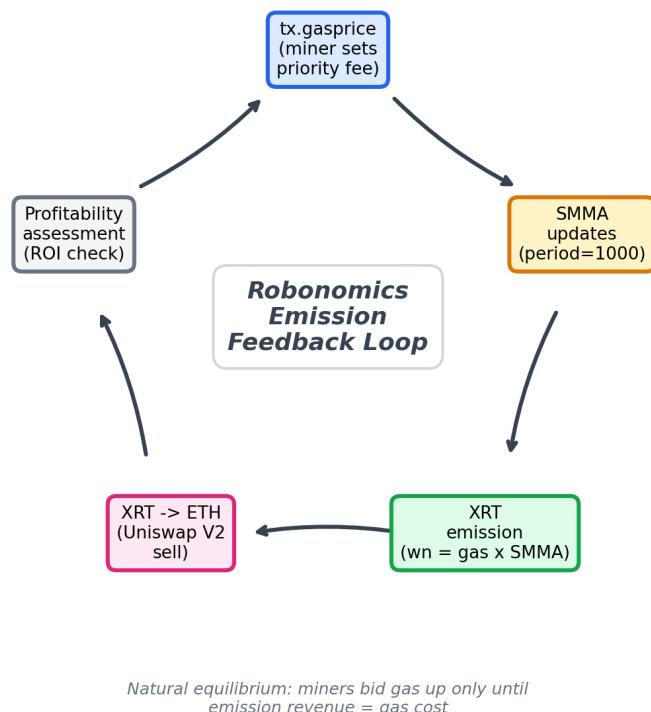


Figure 6. The reflexive incentive cycle in XRT emission. Miners' gas price choices update the SMMA, which determines emission, which affects profitability, which informs future gas price choices.

The emission mechanism exhibits a fascinating reflexive property. The cycle operates as follows: (1) the miner chooses `tx.gasprice` via priority fee; (2) each transaction updates the SMMA toward the chosen gas price; (3) the SMMA determines emission rate: higher SMMA means more XRT per gas unit; (4) minted XRT can be sold on secondary markets for ETH; (5) revenue from sales determines whether the operation is profitable; (6) profitability assessment informs the miner's choice of gas price

for the next round.

This creates a **natural equilibrium**: miners will bid gas prices up only to the point where the additional emission revenue covers the additional gas cost. Under the 2018 gas regime, this equilibrium naturally tracked market gas prices. Under 2026 conditions, the equilibrium point is much lower, but the feedback mechanism still operates.

5.2.1. Formal Equilibrium Model

We can formalize the equilibrium condition. Let p denote the priority fee, b the base fee, and P_{xrt} the market price of XRT in ETH. At steady state, the SMMA converges to the effective gas price $g^{\text{eff}} = b + p$. The revenue per round (10 liabilities, 20 contract calls) is:

$$R(p) = 10 \times G_{\text{lib}} \times g_{\text{eff}} \times 10^9 / F_{\text{auction}} \times P_{\text{xrt}} / 10^9$$

The cost per round is simply the gas spent: $C(p) = 20 \times G_{\text{lib}} \times g_{\text{eff}} \times 10^{-9}$ ETH. The equilibrium priority fee p^* satisfies $R(p^*) = C(p^*)$. Figure 8 illustrates this equilibrium.

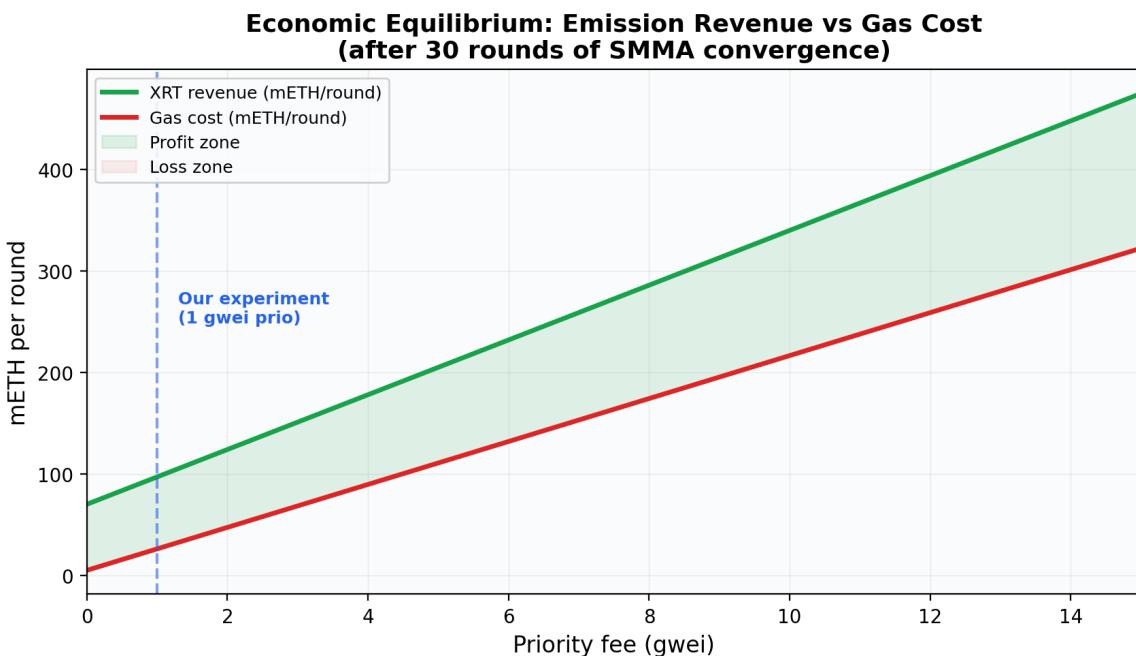


Figure 8. Economic equilibrium: emission revenue vs gas cost as a function of priority fee. The green zone marks where mining is profitable; the red zone marks losses. The crossover point defines the break-even priority fee at steady state.

The key insight is that the equilibrium is **self-limiting**: increasing the priority fee raises both revenue (via higher SMMA \rightarrow more XRT) and cost (via higher gas expenditure). At our observed XRT price of ~32 uETH, the model predicts that the “Pump & Mine” strategy is only profitable in the transient regime where SMMA exceeds the steady-state value — precisely what we observed empirically.

5.3. Properties of the Period-1000 SMMA

Convergence rate. The SMMA half-life is $\ln(2) \times P / N = 693 / 20 \sim 35$ rounds, where $P = 1000$ is the smoothing period and $N = 20$ is calls per round. We observed a decline from 1.94 to ~1.41 gwei over 69 rounds, consistent with ~50% convergence toward the target (1.2 gwei).

Manipulation cost. Increasing SMMA from 1.03 to 4.0 gwei required 448 calls at 10.2 gwei effective price, costing 2.27 ETH. The cost scales linearly with the desired shift magnitude and the gas consumed per call.

Resistance to single-block manipulation. The SMMA can shift at most $(\text{target} - \text{SMMA}) / 1000$ per call. This provides strong protection against flash-loan-style attacks — a property that many DeFi protocols adopted only after costly exploits revealed the dangers of instantaneous price manipulation.

5.4. The Changing Gas Landscape

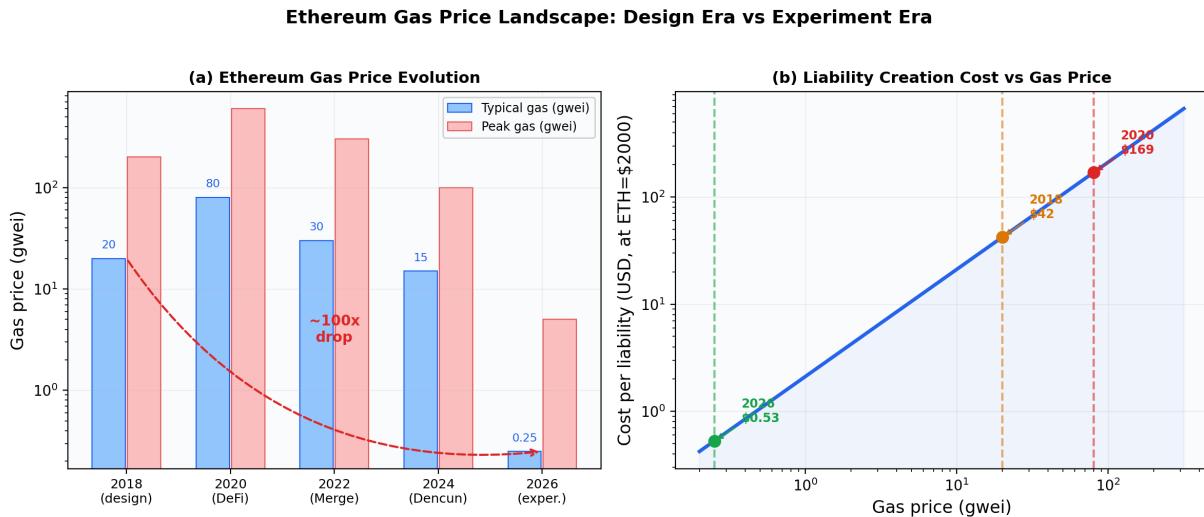


Figure 7. (a) Ethereum gas price evolution 2018–2026. (b) Cost per liability as a function of gas price. The 100x decrease from design era to experiment era fundamentally altered the economic equilibrium.

The most significant contextual factor is the ~100x decrease in Ethereum gas prices between the design era (2018) and our experiment (2026). This transformation was driven by multiple Ethereum protocol upgrades (EIP-1559 [6], The Merge [7], EIP-4844/Dencun [8]) and the migration of activity to L2 rollups. At 2018 gas prices of 20 gwei, each liability cost ~\$42 to create, making empty liability mining economically infeasible. At 2026 prices of 0.25 gwei, the cost dropped to \$0.53 — a reduction of ~80x.

5.5. Design Lessons

Lesson 1: Environmental assumptions are the most fragile invariant. The Robonomics architects correctly identified gas cost as the fundamental unit of account. The emission formula faithfully tracks this metric. What changed was the external environment — gas prices decreased 100x — not any internal property of the mechanism.

Lesson 2: Endogenous oracles have bounded domains of applicability. The SMMA works excellently when `tx.gasprice` reflects genuine market conditions. The exponential smoothing prevents single-block manipulation — a property many later protocols failed to incorporate. However, like all oracle designs, it has a domain of applicability bounded by the assumptions about participant behavior.

Lesson 3: Immutable contracts are time capsules of their design era. The Robonomics v5 contracts — still correctly minting tokens, managing lighthouses, processing liabilities in 2026 — are a testament to the quality of the implementation. That the economic equilibrium has shifted illustrates the fundamental challenge of encoding economic policy in immutable code.

6. The Art of Early Ethereum Mechanism Design

The Robonomics whitepaper was published on May 12, 2018, during a period of extraordinary creativity in smart contract architecture. The authors — Lonshakov and Krupenkin as lead architects, along with Radchenko, Kapitonov, Khassanov, and Starostin — drew on a remarkable intellectual synthesis: Wiener's cybernetics, Coase's economics, ROS robotics, and Ethereum's programmable state machine. Several design decisions deserve recognition as pioneering contributions:

6.1. The Lightweight Contract Pattern

The Factory uses `DELEGATECALL` to create liability contracts that share implementation code but maintain separate state. This pattern saved 30–40% gas per contract creation and predated the widespread adoption of minimal proxy contracts (EIP-1167, 2018). The Robonomics team was at the frontier of gas optimization.

6.2. Deferred Signature Architecture

Rather than requiring both parties to submit on-chain transactions, Robonomics uses deferred signatures — off-chain signed messages that a provider submits on behalf of both parties. This meta-transaction pattern anticipated EIP-2612 (permit) and EIP-4337 (account abstraction) by several years. The implementation uses `ecrecover` to verify demand and offer signatures, ensuring that the transaction sender cannot forge parties' identities.

6.3. Gas-as-Unit-of-Account

The whitepaper's principle “*Emission of 1 Wn = 1 gas utilized by Robonomics*” is a conceptual breakthrough. It recognizes that in the EVM, gas is the fundamental measure of computational work — and therefore the natural unit of account for a protocol that monetizes computational work by robots. The emission formula directly embodies this principle, creating a token whose supply growth is algorithmically linked to EVM resource utilization.

6.4. The Denomination System

The XRT denomination system — wiener (1), coase (10^3), glushkov (10^6), robonomics token (10^9) — encodes the interdisciplinary philosophy of the project, paying homage to the intellectual lineage: Norbert Wiener (cybernetics), Ronald Coase (transaction cost economics), and Viktor Glushkov (Soviet cybernetics pioneer).

7. Conclusion

We have presented an empirical study of the Robonomics XRT emission mechanism, conducting a controlled experiment on Ethereum mainnet that generated 24,190 XRT tokens through 1,360 liability contracts. The experiment demonstrates that the SMMA-based emission formula continues to function as mathematically specified eight years after deployment, while the economic equilibrium has shifted due to the ~100x decrease in Ethereum gas prices.

The “Pump & Mine” strategy — artificially inflating the SMMA through high-priority transactions, then mining at lower cost — proved technically successful but economically constrained by secondary market dynamics. The thin liquidity of the XRT/ETH pool on Uniswap V2 imposed a natural limit on extraction, with each sell event degrading the price by approximately 1%.

Our key finding is that the Robonomics emission mechanism represents a sophisticated piece of early Ethereum mechanism design that correctly implements its stated economic principles. **A mechanism that correctly tracks its specified parameters is well-designed, even when external conditions shift its equilibrium point.**

The Robonomics architecture — with its Factory pattern, lighthouse coordination, deferred signatures, and gas-proportional emission — stands as a monument to the creativity and ambition of early Ethereum builders. Studying these systems years later provides invaluable insights into the art and science of encoding economic behavior in immutable code.

References

[1] S. Lonshakov, A. Krupenkin, E. Radchenko, A. Kapitonov, A. Khassanov, A. Starostin, “Robonomics: platform for integration of cyber physical systems into human economy,” May 2018.

[2] N. Wiener, *Cybernetics: or Control and Communication in the Animal and the Machine*, MIT Press, 1948.

[3] R. H. Coase, “The Nature of the Firm,” *Economica*, vol. 4, no. 16, pp. 386–405, 1937.

[4] M. Quigley et al., “ROS: an open-source Robot Operating System,” *ICRA Workshop*, 2009.

[5] H. Adams, N. Zinsmeister, D. Robinson, “Uniswap v2 Core,” March 2020.

[6] V. Buterin, “EIP-1559: Fee market change for ETH 1.0 chain,” April 2019.

[7] Ethereum Foundation, “The Merge,” September 2022.

[8] Ethereum Foundation, “EIP-4844: Shard Blob Transactions (Dencun),” March 2024.

[9] S. Nakamoto, “Bitcoin: A Peer-to-Peer Electronic Cash System,” 2008.

[10] Protocol Labs, “Filecoin: A Decentralized Storage Network,” 2017.

[11] W. Warren, A. Bandeali, “0x: An open protocol for decentralized exchange on the Ethereum blockchain,” 2017.

Appendix A. Observed Gas Parameters

| Operation | Gas (observed) | Std. dev. |
|-------------------------------------|----------------|-----------|
| createLiability | 789,000 | +/- 5,000 |
| finalizeLiability | 267,000 | +/- 1,000 |
| swapExactTokensForETH | 112,000 | +/- 3,000 |
| Per liability (create + finalize) | 1,058,000 | --- |
| Per round (10 create + 10 finalize) | 10,564,000 | --- |

Appendix B. Smart Contract Addresses

| Contract | Address |
|-------------------|--|
| Factory | 0x7e384AD1FE06747594a6102EE5b377b273DC1225 |
| XRT (ERC-20) | 0x7dE91B204C1C737bcEe6F000AAA6569Cf7061cb7 |
| Lighthouse (exp.) | 0x04C672af1e54d6C9Bd3f153d590f5681d8EcbbeE |
| Auction | 0x86da63b3341924c88baa5adbb2b8f930cc02e586 |
| Uniswap V2 Router | 0x7a250d5630B4cf539739dF2C5dAcB4c659F2488D |
| Chainlink ETH/USD | 0x5f4eC3Df9cbd43714FE2740f5E3616155c5b8419 |

Appendix C. Experiment Timeline

| Time | Event | Key metric |
|----------|--|-------------------------|
| T+0 min | Pump launched (batch=56, prio=10 gwei) | SMMA = 1.03 gwei |
| T+15 min | Pump complete (4 rounds, budget exhausted) | SMMA ~ 4.0 gwei |
| T+20 min | Cleanup: finalize remaining, sell 16,899 XRT | 0.763 ETH received |
| T+25 min | Mine attempt #1 (batch=56, prio=1 gwei) | TX timeouts at round 13 |
| T+45 min | Mine attempt #2 (batch=56, prio=2 gwei) | TX timeout at round 1 |

| Time | Event | Key metric |
|-----------|---|-----------------------|
| T+55 min | Mine attempt #3 (batch=20, prio=2 gwei) | TX timeout at round 1 |
| T+60 min | Mine attempt #4 (batch=10, prio=1 gwei) | Stable (0 errors) |
| T+135 min | 69 rounds complete, mining unprofitable | SMMA = 1.41 gwei |
| T+140 min | Experiment terminated | Net: -0.342 ETH |

Appendix D. Source Code

The complete source code for `xrt-classic-miner` is available at:
`/home/ens/sources/xrt-classic-miner/`

| Module | Description | Lines |
|-------------------------------------|------------------------------------|-------|
| <code>xrt_miner/miner.py</code> | Core mining logic (XRTMiner class) | ~620 |
| <code>xrt_miner/signer.py</code> | Demand/offer/result encoding | ~183 |
| <code>xrt_miner/abi.py</code> | Contract interfaces and addresses | ~150 |
| <code>xrt_miner/___main___py</code> | CLI entry point (Click) | ~400 |