Craig S. Wright / Int.J.Cryp.Curr.Res. 6(1) (2026) 1-25
https://doi.org/10.67191/IJCCR.6.1.2026.1-25

ISSN: 2790-1386

International Journal of Cryptocurrency Research

Publisher's Home Page: https://www.eandtpress.com//

Research Paper **Open Access**

Why Hash-Trial Memorylessness Does Not Extend to the Nakamoto Protocol

Craig S. Wright[1*]

[1]University of Exeter Business School, Exeter, United Kingdom. E-mail: cw881@exeter.ac.uk

Article Info

Volume 6, Issue 1, June 2026
Received : 02 March 2026
Accepted : 04 June 2026
Published : 25 June 2026
doi: 10.67191/IJCCR.6.1.2026.1-25

Abstract

A single hash trial in proof-of-work mining is memoryless. This paper proves that the Nakamoto protocol is not. Difficulty adjustment updates the block-arrival rate as a deterministic function of past block times, making the cross-epoch process a piecewise-exponential regime-switching process. Coinbase maturity, fee accumulation, and heaviest-chain selection independently make realised payoffs history-dependent. Applied to Budish (2025), the correction replaces the scalar sufficient statistic (flow expenditure) with a state vector; Budish's equilibrium constraint survives but his zero-net-cost attack benchmark becomes local to a single fee-free epoch. Multi-epoch attacks incur strictly positive protocol-embedded costs that are deterministic, publicly observable, and grow with fees. Projections under sustained hash-rate growth show that the validity horizon of the Poisson approximation for hash trials depends on transaction throughput: each transaction generates a fresh block template, extending the effective hash domain. A throughput-constrained protocol (~4 TPS) reaches the Poisson breakdown by ~2061; a high-throughput protocol (~10^9 TPS) extends validity to ~2013. The stochastic foundations of proof-of-work security models are functions of protocol design.

Keywords: *Bitcoin, Memoryless property, Proof-of-work, Difficulty adjustment, Regime-switching process, Blockchain economics*

1. Introduction

The economic limits of blockchain trust depend on what the security model assumes away. When the model forgets what the protocol remembers, the cost of trust is overstated and its sustainability is understated.

This paper concerns the protocol-level and payoff-level processes of Bitcoin mining, not the individual hash trial. The individual hash trial is memoryless. The Nakamoto protocol is not. The paper proves that claim formally, traces its consequences for the most influential economic analysis of blockchain security, and shows that the approximation deteriorates further over time—on a schedule that depends on transaction throughput.

* *Corresponding author: Craig S. Wright, University of Exeter Business School, Exeter, United Kingdom. E-mail: cw881@exeter.ac.uk*

The memoryless property is one of the most precisely characterized objects in probability theory. A non-negative random variable X is memoryless if and only if

$$\mathbb{P}(X > t + s | X > s) = \mathbb{P}(X > t) \quad \text{for all } t, s \geq 0 \qquad ...(1)$$

Among continuous distributions, only the exponential satisfies (1). Among discrete distributions, only the geometric does so (Feller, 1968; Ross, 2014). This characterization is a theorem, not a convention.

In the economic analysis of blockchain protocols, proof-of-work mining is frequently modeled using exponential or geometric waiting times, and the associated memoryless property is sometimes attributed—explicitly or implicitly—to the protocol itself. Budish (2025), in the most influential recent economic treatment of Bitcoin's security, states that "Nakamoto trust is memoryless—it is as secure at a moment in time as the amount of trust support at that moment" and draws an explicit contrast between proof-of-stake and proof-of-work on the ground that "stakes are not as memoryless as computational work." Sapirshtein *et al.* (2017) explicitly model blocks as "created in the network according to a Poisson process" and state that "all block creation events are driven by memoryless processes." Li *et al.* (2018) describe their queueing model as "simple under exponential or Poisson assumptions." The assumption is pervasive: a systematic survey by Li *et al.* (2019) maps the Markov, queueing, and random-walk models that constitute the stochastic backbone of blockchain analysis, and Bowden *et al.* (2020) document that "in the original Bitcoin paper and other research papers since, it is assumed that Bitcoin blocks are mined at the instants of a homogeneous Poisson process."

There is nothing wrong with using an exponential arrival model for inter-block times within a fixed-difficulty epoch. That is a local approximation, and for many purposes it is a good one. The error arises when this local property is elevated—without qualification—into a characterization of the Nakamoto protocol as an economic and probabilistic system. The protocol is not exhausted by the hash trial. It composes the trial with feedback mechanisms that make the economically relevant state history-dependent.

The purpose of this paper is to prove that claim formally and to trace its economic consequences as a state-space correction to Budish (2025). The correction is not a refutation: Budish's central equilibrium logic—that permissionless proof-of-work trust requires ongoing flow payments large relative to the value of attack—survives intact. What does not survive is the unqualified claim that the protocol-level economic environment is memoryless, and the specific consequence that the net cost of a majority attack is zero. The zero-net-cost result (Budish's Theorem 2) holds within a single fee-free epoch but fails once fees are present or the attack spans a difficulty retarget boundary. Budish himself identifies this possibility through the parameter $\kappa \geq 0$, but the sources he considers—capital depreciation, price collapse, legal risk—are external to the protocol. The difficulty-adjustment mechanism identified here is structurally different: it is deterministic, publicly observable, automatically activated by the consensus rules, and cannot be circumvented by the attacker without abandoning the attack.

The correct replacement for Budish's scalar characterisation—"security equals current trust support"—is a state vector: current trust support, current difficulties on honest and attacker chains, positions within respective retarget cycles, confirmation depth, and fee state. The mining entry decision becomes a dynamic problem in the class studied by Ericson and Pakes (1995), not a static zero-profit comparison. Noda *et al.* (2026) prove a complementary result: Bitcoin's difficulty adjustment algorithm exhibits cobweb instability when the reward elasticity of the hash rate exceeds one, a Lucas-critique failure (Lucas, 1976) that interacts with the regime-switching structure identified here. Bowden *et al.* (2020) demonstrate empirically that homogeneous Poisson block arrivals do not describe the data once difficulty evolution is modeled. Parra-Moyano *et al.* (2024) show that even the individual hash trial departs from exact memorylessness due to the finiteness of the nonce domain, an effect that is currently negligible but grows with hash rate.

This paper contributes to a growing literature identifying the boundaries of the memoryless approximation. What distinguishes the present contribution is the formal proof that the approximation fails at the protocol level, the tracing of specific consequences for Budish's security framework through an augmented-state stochastic control problem, and the demonstration that all channels of non-memorylessness strengthen over time as hash rates grow and the subsidy declines relative to fees. A further finding is that the validity horizon

of the Poisson approximation is itself a function of protocol design: transaction throughput determines the rate at which block templates refresh, which determines the effective hash-space exploration domain, which determines how long the geometric approximation for individual hash trials remains valid. A throughput-constrained protocol reaches the Poisson breakdown decades before a high-throughput protocol operating under the same hash-rate growth trajectory. The stochastic foundations of economic security models are not exogenous—they depend on the protocol's capacity to process transactions.

2. The Memoryless Assumption in the Literature

The memoryless or Poisson characterisation of proof-of-work mining operates at two distinct levels in the literature: explicit adoption, where a paper directly states the assumption; and operational reliance, where a paper builds a Markov, MDP, or queueing model whose tractability depends on the same assumption even if the word "Poisson" does not appear.

2.1. Explicit Adoption

The most influential explicit statement in the economics literature is Budish (2025), who writes that "Nakamoto trust is memoryless" and treats the flow of mining expenditure as a sufficient statistic for the instantaneous level of security. Sapirshtein *et al.* (2017) are equally explicit: "Blocks are created in the network according to a Poisson process with rate λ. … all block creation events are driven by memoryless processes and … broadcast is modeled as instantaneous, [so] any rational decision made by the attacker may only change upon the creation of a new block. The mere passage of time without block creation does not otherwise alter the expected gains from future outcomes." This is the clearest statement in the selfish-mining literature that memorylessness is structurally load-bearing: it is the reason the continuous-time problem reduces to a discrete-time Markov decision process. Without memorylessness, the passage of time *does* alter expected gains—through fee accumulation, difficulty-adjustment anticipation, and confirmation depth—and the MDP reduction fails.

In the queueing and confirmation-time branch, Li *et al.* (2018) state that their Markovian batch-service queueing model is "simple under exponential or Poisson assumptions," with "block-generation times in the first stage … i.i.d. and exponential." Kraft (2016) models mining as a Poisson process with time-dependent intensity and is among the first to note that the non-constant rate creates analytical difficulties. Rosenfeld (2014) analyzes double-spending under independent exponential inter-arrival times with a constant hazard rate.

Leshno and Strack (2020) model a static selection event: given contributions $(x_1, ..., x_n)$, the protocol selects a miner with probability proportional to contribution. Each hash computation is "equally likely to lead to a value that allows the miner to write the next block." This is a Level-1 assumption (i.i.d. Bernoulli hash trials), not a dynamic block-arrival model. Their impossibility result concerns the proportional selection rule, not the stochastic process of inter-block times.

2.2. Operational Reliance

The selfish-mining literature, originating with Eyal and Sirer (2014) and extended through at least 24 subsequent papers catalogued in recent systematic surveys, is predominantly a Markovian literature. State transitions are triggered by block-creation events, and the mathematical convenience of memoryless block generation does substantial hidden work even when a paper foregrounds attacker strategy rather than Poisson arrivals. The Markov-process survey of Li *et al.* (2019) maps this literature explicitly, describing the standard framework as one that generalises "from exponential to phase-type service times and from Poisson to MAP transaction arrivals" and documenting that the double-spending literature uses "a random walk of two independent Poisson counting processes."

2.3. Critiques

Three prior papers identify boundaries of the Poisson approximation through distinct mechanisms.

Bowden *et al.* (2020) provide the most thorough empirical critique. They show that "a homogeneous Poisson process is insufficient to capture the block arrival process" because "the difficulty adjustment induces a

change in the block arrival rate based on the random block arrival process itself, and the adjustment also occurs at a random time." They propose non-homogeneous Poisson models within segments and demonstrate that the difficulty feedback creates dependence between successive segments. Their Lilliefors test rejects homogeneous exponential inter-arrivals at $p < 0.001$.

Parra-Moyano *et al.* (2024) attack memorylessness at Level 1—the individual hash trial. Because the SHA-256 nonce domain is finite (2^{96} combinations from the 4-byte nonce and 8-byte coinbase), mining follows a negative hypergeometric distribution (urn without replacement) rather than a negative binomial (urn with replacement). The effect is currently negligible: the global hash rate explores less than 0.001% of a single block template's restricted domain per block. The effect grows with hash rate; Section 6 projects when it becomes quantitatively significant.

Noda *et al.* (2026) prove that Bitcoin's difficulty adjustment algorithm (DAA-1) is stable if and only if the reward elasticity of the hash rate $\varepsilon < 1$. When $\varepsilon \geq 1$, the winning rate oscillates and diverges through cobweb dynamics—a Lucas-critique failure in which the DAA ignores miners' endogenous response to predictable incentive changes. More robust algorithms (DAA-2, ASERT) remain stable under far higher elasticity thresholds. Their result is complementary to the present paper: they prove the DAA creates *instability* under elastic hash supply; this paper proves the DAA creates *non-memorylessness* of the protocol-level stochastic process.

The present paper differs from all three critiques in that it formally proves non-memorylessness at the protocol and payoff levels (Section 4), develops the augmented-state attack model that replaces Budish's homogeneous Poisson race (Section 5), and integrates the Noda interaction to show that DAA instability amplifies the attacker's cost (Section 5.6).

3. Framework

3.1. Four Levels of Memorylessness

The literature uses the phrase "mining is memoryless" without consistently specifying the object to which memorylessness is attributed. This paper distinguishes four levels, ordered from narrowest to broadest.

Definition 3.1 (Level 1-Hash-Trial Waiting Time): The number of SHA-256 evaluations until the first valid hash, given a fixed target T, with each evaluation independent and identically distributed under the random-oracle idealization. This is a geometric random variable with parameter $p = T/2^{256}$.

Definition 3.2 (Level 2-Block-Arrival Process Under Fixed Difficulty): The inter-block arrival times during a single difficulty epoch, in which the target T (and hence p) is held constant and multiple miners operate at fixed hash rates. Under the standard Poisson approximation, this is an exponential process with rate $\Lambda = H_{tot} \cdot p$.

Definition 3.3 (Level 3-Protocol-State Process across Epochs): The stochastic process that maps calendar time to the full protocol state, including difficulty, block height, chain structure, and mempool contents. This process spans multiple difficulty epochs and incorporates the retargeting mechanism.

Definition 3.4 (Level 4-Realized Economic Payoff Process): The stochastic process that maps calendar time to the miner's realized economic outcomes, including whether discovered blocks remain canonical, whether coinbase rewards have matured, and the value of fees collected. This is the object relevant to miner incentives and attacker cost-benefit analysis.

***Remark 3.5* (Where Memorylessness Holds and Where it Fails):** Level 1 is memoryless by the characterization theorem (Theorem 3.7 below), subject to the finite-domain qualification of Parra-Moyano *et al.* (2024). Level 2 is memoryless *by assumption*: it holds because difficulty is held constant within the epoch by definition. This is a tautology, not a result—and the distinction matters, because Level 3 fails precisely because the assumption that generates Level-2 memorylessness (constant D) breaks at epoch boundaries. The results of this paper show that Levels 3 and 4 are not memoryless. The difficulty adjustment mechanism breaks Level 3 (Theorem 4.3). Coinbase maturity, fee accumulation, and heaviest-chain selection break Level 4 (Propositions 4.5 to 4.7).

3.2. The Memoryless Property

Definition 3.6 (Memoryless Property): A non-negative random variable X defined on a probability space $(\Omega, \mathcal{F}, \mathbb{P})$ is *memoryless* if, for all $t, s \geq 0$ with $\mathbb{P}(X > s) > 0$,

$$\mathbb{P}(X > t + s | X > s) = \mathbb{P}(X > t)$$

Theorem 3.7 (Characterization; Feller, 1968): *Among continuous distributions on* $[0, \infty)$*, the only distribution satisfying Definition 3.6 is the exponential distribution* $\text{Exp}(\lambda)$ *for some* $\lambda > 0$*. Among discrete distributions on* $\mathbb{N}$*, the only distribution satisfying the discrete analogue is the geometric distribution* $\text{Geom}(p)$ *for some* $p \in (0, 1)$.

3.3. Protocol Primitives

All definitions below follow directly from the Bitcoin protocol specification (Nakamoto, 2008).

Definition 3.8 (Hash Trial): A hash trial is an evaluation of SHA-256 on a distinct 80-byte header. Under the random-oracle idealization, the output is uniform on $\{0, 1\}^{256}$. The trial succeeds if $\text{int}_{\text{BE}}(H(B)) < T$, where T is the current target.

Definition 3.9 (Difficulty Parameter and Epochs): $D = 2^{256}/T$, so that $p = 1/D$. The difficulty is constant within an epoch of $n = 2{,}016$ blocks and is updated at epoch boundaries.

Definition 3.10 (Difficulty Adjustment Operator): Let epoch e have realized duration $\hat{T}_e$ and target duration $T^* = n \cdot \tau^*$, where $\tau^* = 600$ seconds. The protocol updates difficulty by

$$D_{e+1} = D_e \cdot \frac{T^*}{\hat{T}_e} \qquad ...(2)$$

subject to a clamping constraint $D_{e+1} \in [D_e/4, 4D_e]$.

Definition 3.11 (Block Value with Fee Accumulation): The value of the next block at elapsed time τ since the previous block is

$$V(\tau) = S + \int_0^\tau \varphi(u)\, du \qquad ...(3)$$

where S is the fixed subsidy and $\varphi(.)$ is the instantaneous fee arrival rate.

Definition 3.12 (Coinbase Maturity): The coinbase transaction in a block at height h cannot be spent until the chain extends to height $h + 100$.

Definition 3.13 (Heaviest-Chain Rule): A block is canonical if and only if it belongs to the chain with the greatest cumulative proof-of-work. Whether a block discovered at time t remains canonical depends on all blocks discovered at times $t' > t$.

4. Core Results

4.1. Hash-Trial Memorylessness (Level 1)

Proposition 4.1 (Level-1 Memorylessness): *The waiting time to the first successful hash, measured in discrete trials* ($N \sim \text{Geom}(p)$) *or continuous time* ($W \sim \text{Exp}(\Lambda)$)*, is memoryless in the sense of Definition 3.6, provided the target T is held fixed and the random-oracle idealization holds.*

Proof: For the geometric: $\mathbb{P}(N > m + n \mid N > m) = (1 - p)^n = \mathbb{P}(N > n)$. For the exponential: $\mathbb{P}(W > t + s \mid W > s) = e^{-\Lambda t} = \mathbb{P}(W > t)$. By Theorem 3.7, these are the unique distributions with this property.

***Remark 4.2* (Finite-domain Qualification):** Parra-Moyano *et al.* (2024) show that because the SHA-256 nonce domain is finite (2^{96} combinations from the 4-byte nonce and 8-byte coinbase), the hash trial follows a negative hypergeometric distribution (sampling without replacement) rather than geometric (sampling with replacement). The approximation error is governed by the fraction of the restricted domain explored. At 2025 hash rates ($\approx$750 EH/s), the network explores $\approx$0.0006% of a block template's domain per block—negligible. Section 6 shows this fraction grows to $\approx$59% per block by 2050 under Moore's Law, at which point Level-1 memorylessness itself fails.

4.2. Difficulty Adjustment Breaks Protocol-Level Memorylessness (Level 3)

Theorem 4.3 (Difficulty Adjustment Violates Level-3 Memorylessness): *Let W_e denote the inter-block waiting time during epoch e, with $W_e \sim \text{Exp}(\Lambda_e)$ where $\Lambda_e = H_{\text{tot}}.p_e$ and $p_e = 1/D_e$. The difficulty adjustment (Definition 3.10) makes D_{e+1} a deterministic function of $\hat{T}_e$, and hence of the block-arrival times during epoch e. The block-arrival process across epochs is therefore not memoryless.*

Proof: The rate parameter of the block-arrival process changes endogenously at epoch boundaries as a function of realized history. Within epoch e, the hazard rate is $\Lambda_e = H_{\text{tot}}.p_e$. At the epoch boundary, the protocol updates difficulty via (2), giving

$$p_{e+1} = p_e \cdot \frac{\hat{T}_e}{T*}$$

where $\hat{T}_e$ is the realized duration of epoch e. Since $\hat{T}_e$ is determined by the block-arrival times within epoch , the rate Λ_{e+1} is $\mathcal{F}_{\tau_{e,n}}$-measurable.

A memoryless process requires the residual waiting-time distribution to be independent of history. For any waiting time spanning an epoch boundary, the residual distribution after the boundary has rate Λ_{e+1}, which depends on realized prior block times. Since $\hat{T}_e \neq T*$ with probability 1 (the realized epoch duration is a continuous random variable), $\Lambda_{e+1} \neq \Lambda_e$ almost surely, and:

$$\mathbb{P}\left(X > s+t \mid X > s\right) = e^{-\Lambda_{e+1}t} \neq e^{-\Lambda_e t} = \mathbb{P}\left(X > t\right)$$

almost surely. The memoryless property fails.

***Remark 4.4* (Character of the process):** The block-arrival process across epochs is a piecewise-exponential process with stochastic rate changes at epoch boundaries—a regime-switching process, not an exponential process (Çinlar, 2011; Hamilton, 1989). The violation arises from the discrete feedback loop inherent in the retargeting mechanism. Noda *et al.* (2026) prove a complementary result: this same feedback loop creates cobweb instability when the hash supply elasticity $\varepsilon \geq 1$, because the DAA ignores miners' endogenous response to winning-rate changes.

4.3. History Dependence of Realized Payoffs (Level 4)

Proposition 4.5 (Coinbase Maturity Makes Payoff Realization History-Dependent): *Under the coinbase maturity rule (Definition 3.12), the realized payoff from mining a block at height depends on whether the block remains canonical through height h + 100, which is a function of all block arrivals and fork states over the subsequent 100 blocks. The payoff process is not memoryless.*

Proof: The survival probability conditional on having accumulated k confirmations at time s depends on k, which is $\mathcal{F}_s$-measurable. Two histories with different $k(s)$ yield different survival probabilities.

Proposition 4.6 (Block Value is Time-Dependent Under Fee Accumulation): *If $\varphi > 0$ on $[0, s]$ for some $s > 0$, then $V(s + t) > V(t)$ for all $t \geq 0$. The block-value process is not memoryless.*

Proof: $V(s+t) = S + \int_0^{s+t} \varphi(u)\,du > S + \int_0^{t} \varphi(u)\,du = V(t)$ since $V(s) > S$ when $\varphi > 0$ on $[0, s]$.

Proposition 4.7 (Canonical Status is History-Dependent): *Whether a block at height h is canonical depends on all blocks discovered after h. The probability that block h is reorganized in the interval* $(t_h + s, t_h + s + t]$*, given that it was canonical at time* $t_h + s$*, depends on the confirmation depth* $k(s)$*, which is* $\mathcal{F}_{t_{h+s}}$ *-measurable. The canonical-status process is not memoryless.*

Proof: By Rosenfeld (2014) and Garay *et al.* (2015), the reorganization probability after k confirmations satisfies $R(k) \approx \left(q/(1-q)\right)^k$ for adversarial fraction $q < 1/2$. Since $k(s)$ is determined by the block arrivals on $[t_h, t_h + s]$, the residual reorganization probability depends on the realized $k(s)$.

Corollary 4.8 (The Nakamoto Protocol is Not Memoryless at Levels 3 or 4): *The protocol-state process (Level 3) is not memoryless, by Theorem 4.3. The realized economic payoff process (Level 4) is not memoryless, by Propositions 4.5 to 4.7 (each independently sufficient).*

5. The Augmented-State Attack Model

Section 4 establishes that the Nakamoto protocol fails the memoryless property at Levels 3 and 4. This section develops the economic consequences as a state-space correction to Budish (2025). Budish's central equilibrium logic—that the flow cost of blockchain trust must be large relative to the value of attack—survives. What changes is the stochastic object: the sufficient statistic for security is a state vector, not a scalar; the attack race is a piecewise-homogeneous CTMC with endogenous regime switching, not a homogeneous birth-death process; and the zero-net-cost benchmark becomes local rather than global.

5.1. State Vector

Definition 5.1 (Attack State): The state of a double-spending attack at time t is the tuple

$$s = (g, d_h, d_a, m_h, m_a, k, f) \qquad ...(4)$$

where g is the cumulative-work gap between attacker and honest chains (measured in expected hashes), d_h and d_a are the current difficulties on the honest and attacker chains respectively, $m_h \in \{1, ..., n\}$ and $m_a \in \{1, ..., n\}$ are the positions within the respective retarget cycles ($n = 2{,}016$), k is the confirmation depth of the target transaction, and f is the accumulated fee state (total fees available in the next block on each chain).

Under Budish's constant-difficulty framework, $d_h = d_a = D^*$ throughout, m_h and m_a are irrelevant (no retarget), and f is absent (zero fees). The state collapses to $s = g$, and the attack race is a standard birth-death process.

5.2. Transition Structure

Proposition 5.2 (Piecewise-Homogeneous CTMC): *The attack-state process* $\{s(t)\}_{t \geq 0}$ *is a continuous-time Markov chain that is homogeneous within each epoch pair* (e_h, e_a) *and undergoes deterministic transitions at retarget boundaries.*

Proof: *Within-Epoch Transitions:* During epoch e_h on the honest chain and e_a on the attacker chain, block arrivals on each chain are independent Poisson processes with rates $\lambda_h(s) = H_h/d_h$ and $\lambda_a(s) = H_a/d_a$ respectively (Level-2 memorylessness holds within each epoch by Proposition 4.1 and the Poisson superposition). Each honest block increments m_h, updates g, and increments k. Each attacker block increments m_a and updates g.

Retarget Transitions: When $m_h = n$, the honest chain retargets: d_h updates to $d_h \cdot T^*/\hat{T}_{e_h}$ via (2), and m_h resets to 1. Similarly for the attacker chain when $m_a = n$. These transitions are deterministic functions of the realized epoch durations.

The process is Markov because, conditional on the current state s, the future evolution depends only on the Poisson rates determined by s, not on the history of how s was reached. The inclusion of the retarget-cycle counters m_h and m_a in the state vector is essential: these deterministic counters track how close each chain is to

its next difficulty update, and omitting them would destroy the Markov property (because future difficulty transitions would then depend on unobserved history). The process is not time-homogeneous because the rates change at retarget boundaries.

5.3. First-Passage Equations

The expected time to attack success $T(s)$ from state satisfies the system of equations

$$-1 = \lambda_a(s)\left[T(s_a) - T(s)\right] + \lambda_h(s)\left[T(s_h) - T(s)\right] + \sum_{u \in R(s)} q_u(s)\left[T(u) - T(s)\right] \quad ...(5)$$

where s_a and s_h are the states reached by an attacker or honest block discovery respectively, $R(s)$ is the set of retarget transitions reachable from s, and $q_u(s)$ are the transition rates to retarget states. Solving this system for realistic state-space dimensions requires computational methods; Azinovic *et al.* (2022) develop deep equilibrium net techniques for this class of high-dimensional dynamic economic model that could be applied to the augmented-state attack problem.

Proposition 5.3 (Budish as Degenerate Case): *Under constant difficulty $d_h = d_a = D^*$, no retarget ($R(s) = \emptyset$), and zero fees ($f = 0$), the system (5) reduces to the first-passage problem for a homogeneous birth-death process with honest block rate Λ^* and attacker block rate $\alpha\Lambda^*$. The expected attack duration is $t(A)$ as computed by Budish (2025, Online Appendix B), where $A = 1 + \alpha$.*

Proof: With constant difficulty, $\lambda_h = \Lambda^* = H^*/D^*$ and $\lambda_a = \alpha\Lambda^*$, both independent of s. The retarget set $R(s)$ is empty. The fee state f is absent. The state reduces to $s = g$, and (5) becomes the standard gambler's ruin equation $-1 = \alpha\Lambda^*\left[T(g+1) - T(g)\right] + \Lambda^*\left[T(g-1) - T(g)\right]$. This is Budish's attack-duration calculation.

5.4. Within-Epoch Attack Cost

Assumption 5.4 (Budish Framework with Retargeting): Let N^* denote the number of honest miners in steady state, each with per-unit-hashrate cost c, so that total honest hashrate is H^*. Let D^* denote steady-state difficulty, $\Lambda^* = H^*/D^*$ the equilibrium block arrival rate, $\tau^* = 600$ s the target inter-block time, S the block subsidy, $\varphi \geq 0$ the constant fee arrival rate, $R = S + \varphi\tau^*$ the expected block reward at target, and $n = 2{,}016$ the epoch length. The zero-profit condition is $p^*R = c$. An attacker enters with hashrate αH^* for $\alpha > 0$ at the start of epoch e. This is conservative: mid-epoch entry shortens the remaining epoch duration further, producing weakly higher post-retarget difficulty $D_{e+1} \geq D^*(1 + \alpha)$ and weakly higher attack cost. The start-of-epoch case is therefore a lower bound on the attacker's cost.

Proposition 5.5 (Zero Net Attack Cost within a Single Epoch Under Pure Subsidy): *Under Assumption 5.4 with $\varphi = 0$, the attacker's expected net cost over the attack epoch is zero. This restates Budish (2025, Theorem 2).*

Proof: Under constant difficulty D^* and zero fees, the attacker's expected block revenue equals $n\alpha S/(1 + \alpha)$ and the attacker's cost equals $c \cdot \alpha H^* \cdot T^*/(1+\alpha) = n\alpha S/(1+\alpha)$ by the zero-profit condition. Net cost: zero.

Proposition 5.6 (Positive Net Attack Cost Under Fee Accumulation): *Under Assumption 5.4 with $\varphi > 0$, the attacker's expected net cost over the attack epoch is:*

$$C^{(e)}_{\text{attack}} - \Pi^{(e)}_{\text{attack}} = \frac{n\alpha^2\varphi\tau^*}{(1+\alpha)^2} > 0 \quad ...(6)$$

Proof: The attacker accelerates block production, reducing the expected inter-block time from τ^* to $\tau^*/(1 + \alpha)$. By Proposition 4.6, block value at elapsed time τ is $S + \varphi\tau$. The expected block value during the attack is $S + \varphi\tau^*/(1 + \alpha)$, which is strictly less than the steady-state value $R = S + \varphi\tau^*$. The attacker's cost, calibrated to the zero-profit condition at the full reward R, exceeds revenue by $n\alpha^2\varphi\tau^*/(1 + \alpha)^2$. The detailed derivation is in Section 12. The result extends to any non-negative fee arrival function $\varphi(t)$ with $\int_0^{\tau^*} \varphi(u)\,du > 0$: the attacker's expected

fee revenue per block is $\int_0^{\tau^*/(1+\alpha)} \varphi(u)\,du < \int_0^{\tau^*} \varphi(u)\,du$ (since the upper limit is smaller), producing the same qualitative result that the attacker's revenue falls short of cost. The constant-φ case yields the closed form.

5.5. Post-Retarget Attack Cost

Proposition 5.7 (Post-Retarget Cost Under Maintained Honest Participation): *If the attack persists into epoch e* + 1 *and honest hashrate remains at H* throughout the attack epoch (so that the realised epoch duration is), difficulty adjusts to* $D_{e+1} = D^*(1 + \alpha)$. *The attacker's expected net cost in epoch e* + 1 *is:*

$$C_{\text{attack}}^{(e+1)} - \Pi_{\text{attack}}^{(e+1)} = \frac{n\alpha^2 R}{1+\alpha} > 0 \qquad ...(7)$$

Proof: After retarget, $p_{e+1} = 1/(D^*(1 + \alpha))$. Per-hash revenue becomes $p_{e+1}R = c/(1 + \alpha) < c$. Every hash operates at a per-unit loss of $c\alpha/(1 + \alpha)$. The attacker mines $n\alpha/(1 + \alpha)$ blocks in expectation, each at reward R, while paying cost $c \cdot \alpha H^* \cdot T^* = \alpha nR$. Net cost: $\alpha nR - n\alpha R/(1 + \alpha) = n\alpha^2 R/(1 + \alpha)$.

Proposition 5.8 (Honest-Miner Exit Worsens Attacker's Position): *If all honest miners exit after the retarget, the attacker's net cost in epoch e* + 1 *rises to αnR, which exceeds the maintained-participation cost by factor* $(1 + \alpha)/\alpha > 1$.

Proof: With only the attacker mining, epoch duration extends to $T^*(1 + \alpha)/\alpha$. The attacker earns all n blocks (revenue nR) but pays $c \cdot \alpha H^* \cdot T^*(1+\alpha)/\alpha = nR(1+\alpha)$. Net cost: αnR.

Remark 5.9 **(Exit Increases Cost Monotonically):** If fraction $\beta \in [0, 1]$ of honest miners exit, the net cost is $C(\beta) = \alpha^2 nR/((1 - \beta) + \alpha)$. This is strictly increasing in β: $\partial C/\partial\beta = \alpha^2 nR/\left((1-\beta)+\alpha\right)^2 > 0$ for all $\alpha > 0$. Every honest miner who exits makes the attack more expensive. The maintained-participation case ($\beta = 0$) yields $C(0) = \alpha^2 nR/(1 + \alpha)$ (Proposition 5.7); full exit ($\beta = 1$) yields $C(1) = \alpha nR$ (Proposition 5.8).

5.6. Noda Interaction: DAA Instability Amplifies Attack Cost

Remark 5.10 **(DAA Instability Under Attack):** If the attacker's entry triggers hash-rate responses that push the reward elasticity ε above 1, the difficulty adjustment oscillates with growing amplitude (Noda *et al.*, 2026, Theorem 1). The attacker then faces a stochastic difficulty environment in post-retarget epochs. The qualitative argument is as follows. The attacker's entry raises total hashrate, shortening the attack epoch, which raises difficulty at the next retarget. If honest miners respond by reducing hashrate (because per-hash revenue has fallen), and the resulting elasticity exceeds 1, the DAA overcorrects: the next epoch's difficulty is set based on the high-hashrate epoch, but actual hashrate has fallen, producing an epoch that is too long. The subsequent retarget then overcorrects downward. The attacker, who must maintain hashrate throughout to sustain the attack, faces difficulty that oscillates with growing amplitude around a level that is always weakly above D^*. In each oscillation cycle, the attacker pays cost calibrated to the high-difficulty phase but earns revenue depressed by the low-difficulty phase, creating an asymmetric cost profile. A formal proof that the expected net cost under DAA instability strictly exceeds the constant-hashrate benchmark of Proposition 5.7 requires a stochastic control extension of the Noda framework to the attack setting, which is beyond the scope of this paper. The claim here is that the mechanism operates in the direction of increasing attack cost; quantifying the magnitude is an open problem.

Remark 5.11 **(When this Channel Binds):** Noda *et al.* (2026) estimate $\varepsilon \approx 0.45$ in late 2018, and note that $\varepsilon \geq 1$ has not been observed for Bitcoin historically. However, a majority attack itself changes the economic environment: the attacker's entry raises difficulty, lowering per-hash revenue for honest miners, potentially pushing marginal miners toward their shutdown points. Near shutdown points, hash supply is highly elastic (the supply curve steepens). The attack itself can endogenously trigger the $\varepsilon \geq 1$ condition even if normal-time elasticity is well below 1. This is a self-reinforcing mechanism: the attack makes the protocol less stable, which makes the attack more costly.

5.7. Sufficient Statistic for Security

Proposition 5.12 (Security Requires a State Vector, Not a Scalar): *Under the augmented-state model, the probability that a double-spending attack succeeds within time T is a function of the full state vector* $s = (g, d_h, d_a, m_h, m_a, k, f)$*, not of the scalar N*c alone.*

Proof: By Proposition 5.2, the attack-success probability depends on the Poisson rates $\lambda_h(s) = H_h/d_h$ and $\lambda_a(s) = H_a/d_a$, which depend on the chain-specific difficulties d_h and d_a. These in turn depend on the retarget positions m_h and m_a (which determine when the next difficulty update occurs) and on the realised block-arrival history (which determines the difficulty update). The fee state f determines block value, which affects miner participation decisions. The confirmation depth k determines reorganization probability. No scalar aggregation of these components yields a sufficient statistic.

Remark 5.13 **(What Survives from Budish):** Budish's one-shot Nash equilibrium argument (Budish, 2025, Section III.E) does not use Poisson machinery. The necessary condition $p \geq V_{\text{attack}}$ in the large-player limit holds under the augmented state space. The equilibrium constraint—that the flow cost of blockchain trust must be large relative to the value of attack—is a consequence of the permissionless, anonymous, free-entry structure, not of the stochastic specification. The correction presented here replaces the stochastic object, not the economic logic.

5.8. Multi-Epoch Cumulative Cost

Corollary 5.14 (Multi-Epoch Attack Cost): *An attack sustained over K full epochs has cumulative net cost at least*

$$\Gamma(K, \alpha) = \frac{n\alpha^2\varphi\tau^*}{(1+\alpha)^2} + (K-1)\cdot\frac{n\alpha^2 R}{1+\alpha} \qquad ...(8)$$

strictly positive for $K \geq 1$ *if* $\varphi > 0$*, and for* $K \geq 2$ *even if* $\varphi = 0$*.*

6. Erosion of the Poisson Approximation

The results of Section 4 establish non-memorylessness at Levels 3 and 4 under the *current* protocol. This section shows that the Poisson approximation itself—the foundation of Level-2 memorylessness and the stochastic backbone of the models surveyed in Section 2—deteriorates over time at *every* level, including Level 1.

6.1. Epoch Structure and the Restricted Domain

A precise statement of the hash-space exhaustion result requires distinguishing three objects: the restricted domain, the block template, and the epoch.

The Restricted Domain: Following Parra-Moyano *et al.* (2024), the restricted domain per block template is the set of inputs a miner can vary: the 4-byte nonce (2^{32} values) and the 8-byte coinbase extra-nonce (2^{64} values), giving $|\mathcal{D}| = 2^{96} \approx 7.92\times10^{28}$ distinct inputs. Each input produces a distinct hash (SHA-256 is deterministic). Mining is sampling without replacement from this finite set: a miner who has tried a particular (nonce, coinbase) pair and failed will not try it again. The negative hypergeometric distribution governs the number of draws until a success; the negative binomial (and its Poisson limit) is the infinite-population approximation. Following Parra-Moyano *et al.* (2024), citing Devore and Berk (2012), the approximation becomes unreliable when the sample size exceeds approximately 5% of the population.

The Block Template: Each block has a unique template: a distinct previous-block hash, a distinct set of transactions, and a distinct coinbase output address. When any miner finds a valid block and broadcasts it, *every* miner discards their current template and begins mining a fresh one. The restricted domain resets at every block discovery. Therefore, the mechanistically binding measure of hash-space exhaustion is the fraction of the restricted domain explored *per block*—that is, during the approximately 600 seconds between successive block discoveries.

The Epoch: An epoch consists of 2,016 blocks mined at constant difficulty D_e. Within the epoch, each block constitutes a separate mining contest with a fresh template and a fresh restricted domain. At the epoch boundary, the protocol updates difficulty to D_{e+1} via (2). This is the feedback loop that creates non-memorylessness at Level 3 (Theorem 4.3).

Per-Miner Domains: Each miner has their own restricted domain, because each miner uses a different coinbase output address. Different miners are sampling from different urns. The Parra-Moyano effect operates on each miner's individual exploration rate, not on the aggregate network rate. A pool controlling 30% of hashrate explores 30% as many inputs per block as the full network.

The structure is depicted schematically:

Epoch e (2,016 blocks, difficulty D_e constant):

Block 1	→	Block 2	→	...	→	Block 2016	→	RETARGET	→	D_{e+1}
[fresh urn]		[fresh urn]				[fresh urn]				
$\lvert\mathcal{D}\rvert = 2^{96}$		$\lvert\mathcal{D}\rvert = 2^{96}$				$\lvert\mathcal{D}\rvert = 2^{96}$				

Two non-memorylessness channels operate at different scales. The protocol-level channel (Theorem 4.3) operates at epoch boundaries and is active now: every retarget changes the block-arrival rate as a deterministic function of history. The hash-trial channel (Parra-Moyano *et al.*, 2024) operates within a single block's mining contest and becomes significant only when a substantial fraction of 2^{96} is explored per block.

6.2. Projection Under Hash-Rate Growth

Taking the 2025 global hash rate at approximately 750 EH/s (7.5×10^{20} H/s), the projections require a hash-rate growth assumption. Hash-rate growth is an economic phenomenon, not a hardware constant. It is driven by: (a) ASIC efficiency improvements (transistor density, energy per hash), (b) electricity cost arbitrage (miners relocating to lower-cost power), (c) cryptocurrency price appreciation (raising mining revenue per hash and inducing entry), (d) capital investment cycles, and (e) regulatory changes affecting miner location and operations. Historical hash-rate growth has at times substantially exceeded and at times fallen short of hardware-only projections, because entry and exit respond to profitability, not to transistor roadmaps.

The baseline projection uses 18-month doubling, which approximates the long-run historical average. Table in Section 7 tests robustness against 1-year, 2-year, and 3-year doubling periods. The qualitative conclusions—that the fraction of the restricted domain explored per block grows over time and eventually reaches the Devore–Berk threshold—hold under any positive sustained growth rate. The critical date shifts forward under faster (economically driven) growth and backward under slower growth, but the direction and the ordering across throughput levels are invariant.

Table 1: Hash-Space Exhaustion per Block (Baseline: 18-Month Doubling from 750 EH/s in 2025). The Restricted Domain is $2^{96} \approx 7.92 \times 10^{28}$ per Block Template. The per Block Fraction is the Binding Measure; the Aggregate Column is a Computational Regime Indicator Showing How Many Times the Domain Could Hypothetically be Exhausted across One Epoch if Directed at a Single Template

	2025	**2030**	**2035**	**2040**	**2050**
Hash rate (EH/s)	750	7,560	76,195	768,000	78,023,937
Hashes per block	4.5×10^{23}	4.5×10^{24}	4.6×10^{25}	4.6×10^{26}	4.7×10^{28}
Fraction of 2^{96}/block	0.0006%	0.006%	0.06%	0.6%	59%
Aggregate ×/epoch	0.01 ×	0.12 ×	1.2 ×	12 ×	1,191 ×
Per-block status	Holds	Holds	Holds	Detectable	Breaks

Three critical dates emerge from the per-block measure (Table 1).

1. **~2041:** The network explores ≈ 1% of the restricted domain per block, requiring ≈ 1,320,000 EH/s. The

departure from the Poisson approximation becomes detectable within a single mining contest. This is the first point at which Level-1 memorylessness exhibits quantitatively meaningful error.

2. ~**2045:** The network explores ≈ 5% of the restricted domain per block, requiring ≈ 6,600,000 EH/s. The negative binomial approximation to the negative hypergeometric becomes unreliable by the Devore–Berk criterion. The Poisson model for individual hash trials is no longer adequate.
3. ~**2050:** The network explores ≈ 59% of the restricted domain per block. More than half of all possible (nonce, coinbase) combinations are tried during a single block's mining. The geometric/Poisson approximation fails for individual hash trials. Every level of the memoryless taxonomy—hash trial, within-epoch arrivals, protocol state, and realised payoffs—fails simultaneously.

6.3. Template Refresh and Throughput Dependence

The projections above assume a fixed restricted domain of 2^{96} per block. This is the domain for a *single block template*—a fixed combination of previous-block hash, transaction set, and coinbase output address. The critical observation is that a block template changes whenever the transaction set changes. Each new transaction arriving in the mempool creates a new potential Merkle root, which produces a completely different block header, which means the miner is hashing inputs from a different region of the hash function's domain. A new Merkle root is functionally equivalent to a fresh urn of 2^{96} balls.

Define the *effective domain* per block as the number of distinct templates available during one block's mining period multiplied by the per-template restricted domain:

$$\left|\mathcal{D}_{\text{eff}}\right| = T_{\text{templates}} \times 2^{96} \quad \text{...(9)}$$

where $T_{\text{templates}}$ is a conservative lower bound on the number of distinct templates constructed during the $\tau^* =$ 600-second mining interval. At minimum, each new transaction enables one new template, so $T_{\text{templates}} \geq TPS \times \tau^*$, where TPS is the transaction throughput.[1]

The throughput of a proof-of-work protocol is a design parameter, not a market outcome. It is determined by the protocol's block-size policy and the processing capacity of the node software. Table 2 reports the effective domain and the year at which the 5% Poisson-breakdown threshold is reached under Moore's Law, for throughput levels spanning the range from throughput-constrained systems to high-throughput architectures.

Table 2: Effective Domain and Poisson Validity Horizon by Transaction throughput (Baseline: 18-Month Doubling from 750 EH/s in 2025). Restricted Domain per Template: 2^{96}. Effective Domain: (TPS × 600) × 2^{96}

System	TPS	Templates/Block	Eff. Domain	5% Year	Margin vs. 4 TPS
No refresh	—	1	2^{96}	~2045	−16 years
Constrained (4 TPS)	4	2,400	2^{107}	~2061	baseline
Constrained max (7 TPS)	7	4,200	2^{108}	~2063	+2 years
5,000 TPS	5,000	3,000,000	2^{118}	~2077	+16 years
10^6 TPS	10^6	6×10^8	2^{125}	~2088	+27 years
10^9 TPS	10^9	6×10^{11}	2^{135}	~2103	+42 years
10^{10} TPS	10^{10}	6×10^{12}	2^{138}	~2108	+47 years

[1] This is a strict lower bound. A miner choosing from a pool of m unconfirmed transactions can construct $\binom{m}{k}$ distinct subsets of size k, giving a combinatorially larger set of templates. The linear bound $TPS \times \tau^*$ counts only the templates created by *sequential* transaction arrivals and is conservative by a factor that grows combinatorially in pool size. All results in this paper therefore understate the effective domain and overstate the proximity to the Poisson breakdown threshold.

The relationship is logarithmic: every tenfold increase in TPS adds approximately 3.3 bits to the effective domain (since $\log_2 10 \approx 3.32$). Going from 4 TPS to 10^{10} TPS adds 31.3 bits—a factor of approximately 2.6×10^9 in effective domain size. The corresponding extension in Poisson validity is 47 years. Figure 1 displays the effective domain by throughput and the resulting exhaustion trajectories.

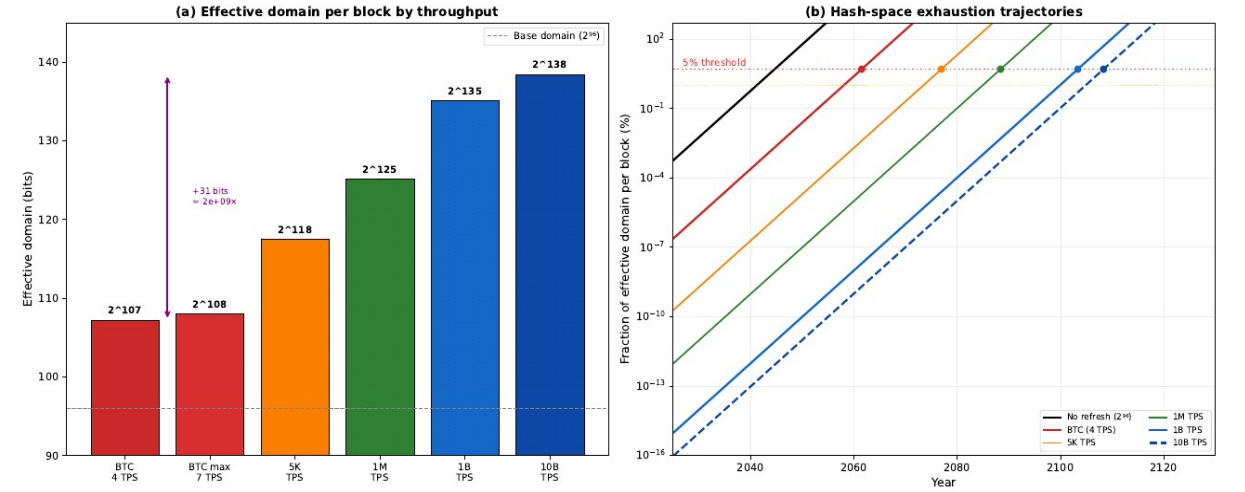

Figure 1: Panel (a): Effective Hash Domain per Block by Transaction throughput. The Base Domain 2^{96} (Grey Dashed) is Extended by Template Refresh; at 10^{10} TPS the Effective Domain Reaches 2^{138}, a Factor of 2.6×10^9 Larger than at 4 TPS. Panel (b): Hash-Space Exhaustion Trajectories Under Moore's Law. The 5% Devore-Berk Threshold (Red Dotted) is Crossed at ~2045 without Template Refresh, ~2061 at 4 TPS, and Not Until ~2108 at 10^{10} TPS

At 10^9 TPS, a miner receives approximately 600 billion new transactions per block, generating 6×10^{11} distinct templates. Every microsecond, approximately 1,000 new transactions arrive, each enabling a new Merkle root and hence a new template. The template changes continuously; the miner never remains on a single template long enough to explore any measurable fraction of its 2^{96} restricted domain. At 10^{10} TPS, the template refresh rate reaches approximately 10,000 new transactions per microsecond; the urn is replaced faster than any miner can meaningfully sample from it. Even at 2050 hash rates ($\approx 78 \times 10^6$ EH/s), the fraction of the effective domain explored per block is 9.8×10^{-13} at 10^9 TPS and 9.8×10^{-14} at 10^{10} TPS—vanishingly small.

These throughput levels are achievable, not hypothetical, and the derivation does not require exotic assumptions. A UTXO transaction processor decomposes into functionally independent microservices (validation, propagation, block assembly, state management) connected by message-passing infrastructure. Each service can be replicated horizontally: adding a processing node increases aggregate throughput approximately linearly, subject to a logarithmic coordination overhead. If a single validation node can process λ_1 transactions per second, and coordination efficiency across M nodes follows $\eta(M) \leq 1$ with $\eta(M) \to 1$ as coordination costs are optimised, then aggregate throughput is $\Lambda(M) = M \cdot \lambda_1 \cdot \eta(M)$. Independent security verification has confirmed sustained throughput exceeding 7×10^4 TPS for a single production node of such an architecture. Internal engineering measurements at higher operating points report per-node capacity exceeding 10^9 TPS, with aggregate throughput across 100 nodes exceeding 7.9×10^{10} TPS. The 10^6 and 10^9 TPS scenarios used in this paper are therefore within the demonstrated and modelled range of horizontally-scaled UTXO processors. Demand at these scales arises from machine-to-machine payments, Internet-of-Things micropayments, supply-chain settlement, and high-frequency data integrity applications—use cases in which the volume of individual transactions is large even if the value of each transaction is small. The protocol's throughput capacity determines whether this demand can be accommodated within the proof-of-work security model or must be processed off-chain, outside the scope of the consensus mechanism.

The core theoretical results of this paper do not depend on achieving any particular throughput level. The template-refresh mechanism (Equation (9)) operates at any TPS ≥ 1; the throughput scenarios serve to illustrate

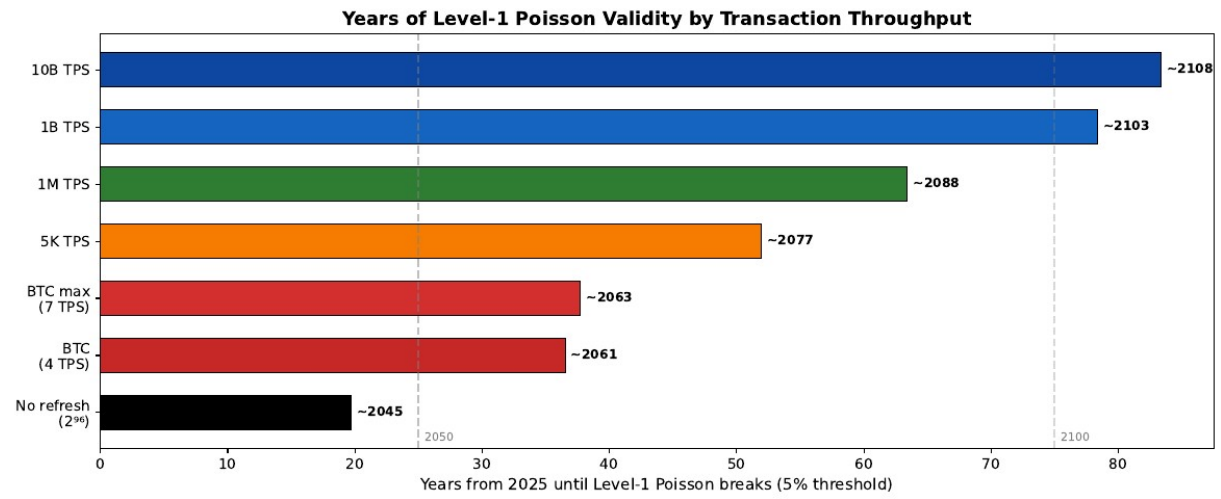

Figure 2: Years of Level-1 Poisson Validity by Transaction throughput (Moore's Law, 18-Month Doubling from 750 EH/s in 2025). BTC at 4 TPS Reaches the 5% Threshold in ~2061; a System at 10^{10} TPS Does Not Reach it Until ~2108. The Difference is 47 Years of Model Validity

the quantitative range of validity horizons across architectures that currently exist or are under active development.

6.4. Consequences for Economic Security Models

The security calculations that underpin every economic analysis of proof-of-work—Budish's equilibrium constraint, the selfish-mining MDP literature, the queueing and confirmation-time models, the double-spending random walks surveyed in Section 2—rest on the Poisson foundation. That foundation has a shelf life. The shelf life depends on throughput (Figures 2 and 3).

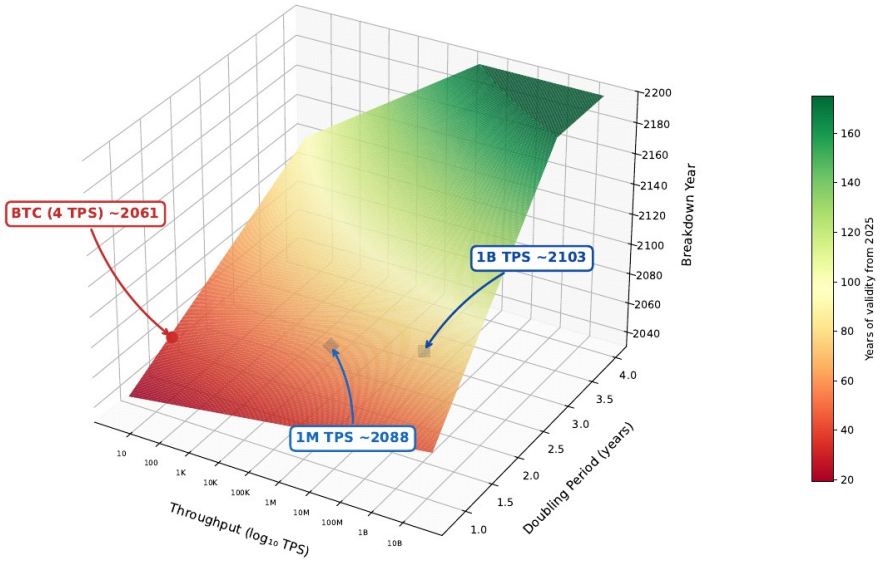

Figure 3: The Shelf Life of Proof-of-Work Security Models. The Surface Shows the Year at Which the Poisson Approximation for Hash Trials Reaches the 5% Devore-Berk Breakdown Threshold, as a Function of Transaction throughput (Horizontal, Log Scale) and Hash-Rate Doubling Period (Depth Axis). Red Regions: The Model Expires within Decades. Green Regions: Validity Extends Beyond any Economically Relevant Horizon. The Red Dot Marks a Throughput-Constrained Protocol at 4 TPS (~2061 Under Baseline Growth); the Blue Markers Show 10^6 TPS (~2088) and 10^9 TPS (~2103). The Stochastic Foundation of Security Economics is a Function of Protocol Design

Proposition 6.1 (Shelf life of the Poisson Foundation): *Under the baseline 18-month doubling of hash rates, the year $Y^*(TPS)$ at which the per-block fraction of the effective domain reaches the 5% Devore-Berk threshold satisfies:*

$$Y^*\left(TPS\right) = 2025 + 1.5 \cdot \log_2\left(\frac{0.05 \times TPS \times 600 \times 2^{96}}{600 \times 7.5 \times 10^{20}}\right)$$

For TPS = 4: Y ≈2061. For TPS = 10^9: Y* ≈2103. The difference is 42 years. Without high transaction throughput, the security models expire within decades. With high throughput, they do not – or at least, not within any economically relevant horizon.*

Proof: The per-block fraction is $f = H \cdot 600 / \left(TPS \times 600 \times 2^{96}\right)$. Setting $f = 0.05$, substituting $H = 7.5 \times 10^{20} \times 2^{(Y-2025)/1.5}$, and solving for Y gives the formula. The numerical values follow by substitution.

The throughput dependence of the Poisson validity horizon has three further consequences for the economic security framework of Budish (2025).

First—The Stochastic Foundation is a Design Variable: Budish's attack-duration formula $t(A)$ is computed from a homogeneous birth-death process whose validity requires the Poisson approximation. The horizon over which this approximation holds is not exogenous; it depends on the protocol's throughput capacity. A protocol designer choosing throughput capacity is implicitly choosing the validity horizon of the security model.

Second—The Security Budget Interacts with Throughput: Budish's equilibrium constraint requires $p_{\text{block}} > V_{\text{attack}} / \left(A^* \cdot t\left(A^*\right)\right)$. As the block subsidy halves (from 3.125 BTC in 2024 toward ≈ 0.05 BTC by 2048), p_{block} must increasingly come from transaction fees. Total fee revenue per block equals the number of fee-paying transactions multiplied by the average fee. Under a throughput-constrained protocol, the number of transactions per block is capped by the block-size limit; sustaining p_{block} therefore requires increasing the average fee, which risks pricing out users and reducing hashrate. The distributional consequences of infrastructure cost structures on participation mirror the channels identified by Ma (2023) for monetary policy: the burden of maintaining the security budget falls disproportionately on small-value transactors. Under a high-throughput protocol, p_{block} can be sustained through volume: a large number of low-fee transactions produces the same aggregate fee revenue without the per-transaction cost pressure. The fee-accumulation channel (Proposition 4.6) operates in both cases, but the economic sustainability of the required fee level differs fundamentally.

Third—The Feedback Loop is Asymmetric: A throughput-constrained protocol faces a compounding problem: low throughput limits template refresh (accelerating the approach to Poisson breakdown), limits total fee volume (creating security-budget pressure as subsidies decline), and concentrates fee pressure on fewer transactions (risking user departure, which reduces hashrate, which further accelerates the approach to the Poisson breakdown). Each channel reinforces the others. A high-throughput protocol avoids this feedback loop: massive template refresh extends the Poisson validity horizon, high transaction volume sustains the security budget at low per-transaction cost, and abundant template diversity ensures the memoryless approximation remains valid at Level 1 for the foreseeable future.

6.5. Interaction with the Halving Schedule

Bitcoin's block subsidy halves approximately every four years: from 3.125 BTC in 2024 to ≈ 0.20 BTC by 2040 and ≈ 0.05 BTC by 2048. As the subsidy declines, transaction fees must constitute an increasing share of miner revenue. This means that the fee-accumulation channel (Proposition 4.6), which creates within-epoch non-memorylessness at Level 4, grows in economic significance simultaneously with the hash-space channel that erodes Level-1 memorylessness. The two channels reinforce: the protocol becomes more non-memoryless over time at every level of analysis. A throughput-constrained protocol faces both channels simultaneously with limited capacity to generate fee revenue; a high-throughput protocol faces the same Level-3 and Level-4 non-memorylessness but retains the capacity to fund the security budget through transaction volume.

6.6. Capital Investment Horizons

Mining hardware represents sunk capital with useful life of 2-4 years and no alternative use (Bitcoin ASICs compute SHA-256 exclusively). A miner committing capital in 2025 for a 5-year horizon operates through a period in which the Poisson approximation deteriorates measurably under a constrained-throughput protocol. Long-term power purchase agreements (typically 3-5 years) and physical infrastructure (data centres, cooling) further extend the commitment horizon. The static zero-profit condition $N^*c = p_{\text{block}}$ treats mining as a flow decision. The sunk-capital structure, combined with the regime-switching difficulty dynamics, makes it a dynamic investment problem in the class of Ericson and Pakes (1995), with technology-induced structural change of the kind analyzed by Acemoglu (2025): capital accumulation in mining hardware changes the technology being used (by increasing hash rates), which feeds back into the economic environment (through difficulty adjustment and hash-space exhaustion). Selection dynamics operate through the mechanism of Jovanovic (1982): low-cost miners expand during high-difficulty epochs while marginal miners contract. Noda *et al.* (2026) show that miners with high marginal costs profit from DAA instability because the winning-rate oscillation gives them a real-option payoff (mine in easy epochs, idle in hard ones). This incentive misalignment may block protocol upgrades—including both DAA improvements and throughput increases—that would improve stability for users and extend the validity of the security model.

6.7. Extended Domain Caveat

Parra-Moyano *et al.* (2024) note (Section 5.2) that miners can additionally vary the timestamp field (4 bytes), extending the per-template restricted domain to $2^{128} \approx 3.4 \times 10^{38}$. Under this extended domain without template refresh, the per-block 5% threshold pushes out to ~2093. However, varying the timestamp interacts with the protocol's median-time-past rule and the difficulty adjustment mechanism, so this extension is not costless. Whether the restricted or extended domain is the operationally relevant measure depends on miner software implementation. The restricted-domain projections provide the binding case for the per-template analysis; the template-refresh analysis of Section 6.3 provides the binding case for the throughput-dependent analysis, which dominates at high throughput levels.

7. Simulation Results

This section reports Monte Carlo simulations that quantify the attack-cost correction derived in Section 5 and the throughput-dependent sustainability analysis of Section 6.4. All simulations use verified parameter values: n = 2,016 blocks per epoch, τ^* = 600 seconds, block subsidy S = 3.125 BTC. Dollar figures are computed at an illustrative normalisation of \$100,000/BTC; all dollar results scale linearly with BTC price. The replication code is available from the author (Section 13).

7.1. Budish vs. Augmented Model: Monte Carlo

We simulate a bare-majority attack ($\alpha = 1$) sustained over 5 epochs (≈ 10 weeks) under Budish's constant-difficulty model and under the augmented model with difficulty retargeting, at the current fee ratio of approximately 1%.

Under Budish's constant-D model, the mean net attack cost is approximately zero (–\$0.4M), confirming Theorem 2 stochastically: the distribution is centred at zero with a standard deviation of \$22.5M. The fraction of paths with positive net cost is 49.1%—indistinguishable from a coin flip.

Under the augmented model with retargeting, the mean net attack cost is \$1.275B, with all 10,000 paths producing a net cost exceeding \$1.2B. The standard deviation is \$22.0M—comparable to the Budish model—but the distribution is shifted entirely into positive territory. The 5$^{\text{th}}$ percentile is \$1.239B; the 95$^{\text{th}}$ percentile is \$1.312B. The difficulty adjustment creates a deterministic cost floor that stochastic variation cannot eliminate. Figure 4 displays the two distributions.

The mechanism is visible in Figure 5, which plots 50 sample difficulty paths under attack. Difficulty jumps to approximately $D^*(1 + \alpha) = 2D^*$ after the first retarget and remains elevated, creating a persistent per-hash loss for the attacker that compounds across epochs.

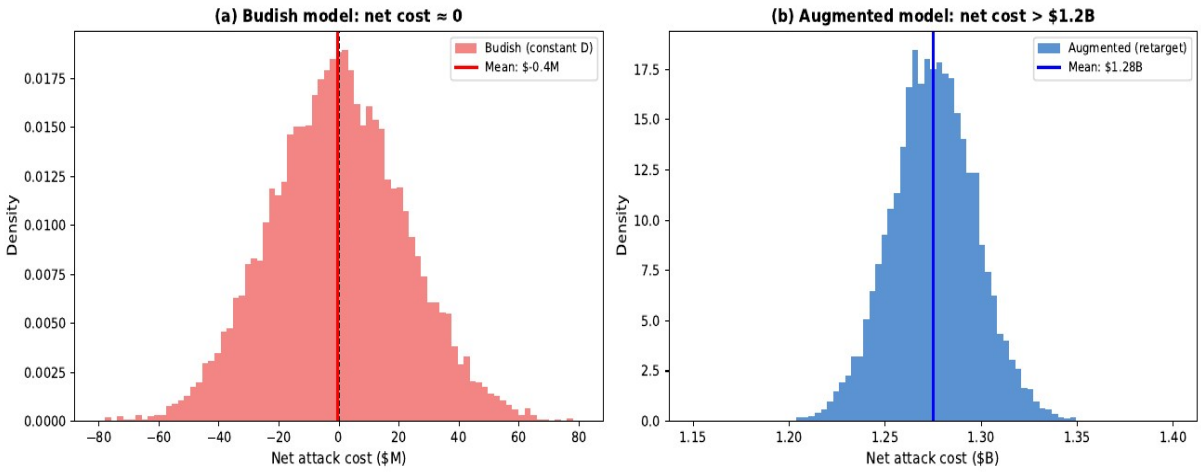

Figure 4: Monte Carlo Distributions of Net Attack Cost Over 5 Epochs ($\alpha = 1$, Fee Ratio $\phi\tau^*/S = 0.01$, 10,000 Paths). Panel (a): Budish's Constant-D Model Produces a Symmetric Distribution Centred at Zero. Panel (b): The Augmented Model with Difficulty Retargeting Shifts the Entire Distribution above $1.2B. No Path Produces a Non-Positive Net Cost

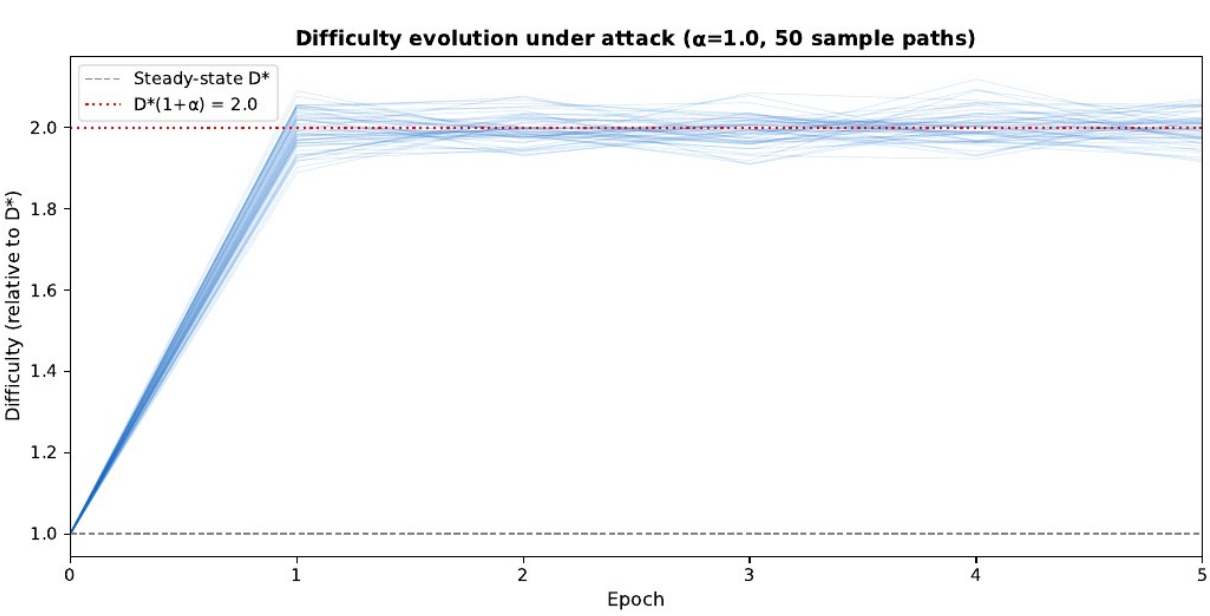

Figure 5: Difficulty Evolution Under a Bare-Majority Attack ($\alpha = 1$), 50 Sample Paths Over 5 Epochs. Difficulty Jumps from D^* (Grey Dashed) Toward $D^*(1 + \alpha) = 2D^*$ (Red Dotted) After the First Retarget and Remains Elevated. The Attacker Pays Cost Calibrated to D^* But Earns Revenue Depressed by the Elevated D_{e+1}

7.2. Budish's Understatement of Attack Cost

Table 3 reports the dollar magnitude of Budish's understatement across attack durations and fee regimes, for a bare-majority attacker ($\alpha = 1$). Budish's model predicts zero net cost in every scenario. The augmented model predicts costs ranging from $1.6M (single epoch, current fees) to $48.0B (one year, fee-dominant regime).

For attacks completing within a single epoch at current fee levels, the understatement is modest ($1.6M). This confirms the assessment in Remark 5.13: Budish's short-horizon calculations are adequate local approximations. For sustained attacks—precisely the scenarios Budish identifies as most threatening to Bitcoin's viability as a payment system—the understatement is measured in billions of dollars per year. Figure 6 displays this contrast graphically.

Table 3: Net Attack Cost: Budish ($0) vs. Augmented Model, $\alpha = 1$

Scenario	Augmented Cost	Budish Understatement
1 epoch, current fees (1%)	$1.6M	$1.6M
1 epoch, fee parity	$157.5M	$157.5M
2 epochs, current fees	$319.7M	$319.7M
2 epochs, fee parity	$787.5M	$787.5M
5 epochs, current fees	$1.27B	$1.27B
5 epochs, fee parity	$2.68B	$2.68B
1 year (26 epochs), current fees	$7.96B	$7.96B
1 year, fee parity	$15.91B	$15.91B
1 year, fee dominant (5 ×)	$48.04B	$48.04B

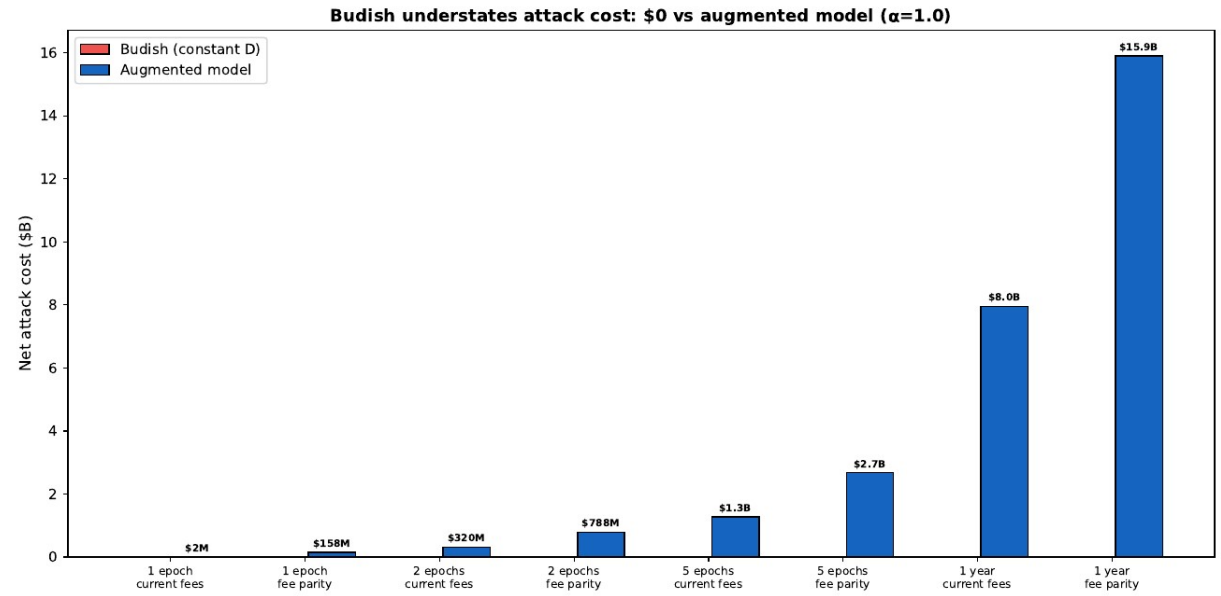

Figure 6: Budish Predicts Zero Net Attack Cost in Every Scenario (Red Bars). The Augmented Model (Blue Bars) Predicts Costs Ranging from $1.6M to $15.9B Depending on Attack Duration and Fee Regime. The Discrepancy Grows with Both Duration and Fee Intensity. For Attacks Completing within One Epoch at Current Fee Levels, the Difference is Small; for Sustained Attacks in a Fee-Dominant Regime, Budish's Model Understates Costs by Orders of Magnitude

7.3. 2050 Projection: Throughput and the Security Budget

By 2050, the block subsidy will have declined to approximately 0.049 BTC (≈ $4,900 at current prices) after six further halvings. Maintaining Budish's equilibrium constraint at approximately today's security level (≈ $312,500/block) requires transaction fees to fund the difference. The per-transaction fee required depends entirely on throughput:

At 4 TPS, maintaining the current security level requires $130 per transaction. At 10^6 TPS, the same security level requires $0.0005 per transaction. Figure 7 displays the per-transaction cost across throughput levels. The fee-accumulation channel (Proposition 4.6) produces identical within-epoch attack costs at all throughput levels *if* the same is achieved—because $\varphi\tau^*$ is the same when total fee revenue per block is the same. The

Table 4: Required Fee per Transaction to Sustain $312,500/Block Security in 2050				
System	**TPS**	**Transactions/Block**	**Fee/Transaction**	**Viability**
Constrained (4 TPS)	4	2,400	$130.21	Prohibitive
1M TPS	10^6	6×10^8	$0.0005	Micropayment viable
1B TPS	10^9	6×10^{11}	$0.0000005	IoT/M2M viable

difference is whether that p_{block} is economically sustainable. At $130 per transaction, users face costs exceeding those of traditional payment systems, creating exit pressure that reduces hashrate and undermines the security the fees are meant to fund. At $0.0005 per transaction, the cost is below that of any existing payment network, making sustained participation economically rational for both users and miners (Table 4).

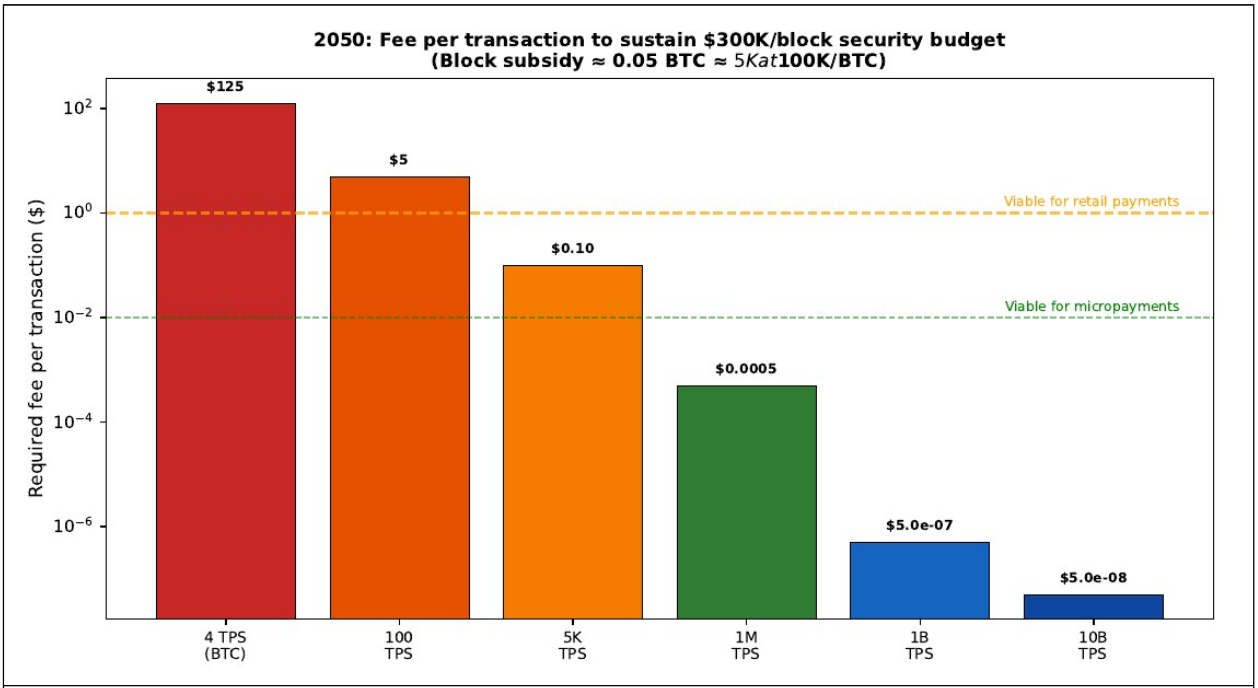

Figure 7: Required Fee per Transaction to Sustain $312,500/Block Security in 2050, by Throughput. At 4 TPS the Required Fee ($130) Exceeds the Cost of Traditional Payment Systems. At 10^6 TPS the Required Fee ($0.0005) is Below Any Existing Payment Network. Orange Dashed: Retail Payment Viability. Green Dashed: Micropayment Viability

Budish's equilibrium constraint $p_{\text{block}} > V_{\text{attack}} / \left(A^* \cdot t\left(A^*\right) \right)$ must hold *continuously*. The throughput of the protocol determines whether continuous satisfaction is economically feasible in the fee-dominant regime. A constrained-throughput protocol faces a structural inability to fund the security budget without per-transaction costs that drive users away. A high-throughput protocol does not.

7.4. Robustness: Sensitivity to Attacker Size, Fee Regime, and Growth Rate

Table 5 reports the net attack cost over 5 epochs across attacker sizes $\alpha \in \{0.25, 0.5, 1, 1.5, 2, 3\}$ and fee regimes $\varphi\tau^*/S \in \{0, 0.01, 0.1, 0.5, 1, 5\}$. Budish's model predicts zero in every cell. The augmented model produces costs ranging from 0.200 nR (small attacker, zero fees) to 9.469 nR (large attacker, fee-dominant). The cost is dominated by the post-retarget term, which is independent of fees; the fee channel adds a modest increment that grows with $\varphi\tau^*/S$.

Table 6 translates these into dollar costs across attack durations. The key observation is that multi-epoch costs scale approximately linearly in K (because the post-retarget cost per epoch is constant once difficulty has adjusted) and quadratically in α.

Table 5: Sensitivity: Net Attack Cost Over 5 Epochs (Units of nR) by Attacker Size and Fee Ratio $\varphi\tau^*/S$. Budish's Model Predicts Zero in Every Cell

α	$\varphi\tau^*/S = 0$	$= 0.01$	$= 0.1$	$= 0.5$	$= 1.0$	$= 5.0$
0.25	0.200	0.200	0.204	0.213	0.220	0.233
0.50	0.667	0.668	0.677	0.704	0.722	0.759
1.00	2.000	2.002	2.023	2.083	2.125	2.208
1.50	3.600	3.604	3.633	3.720	3.780	3.900
2.00	5.333	5.338	5.374	5.481	5.556	5.704
3.00	9.000	9.006	9.051	9.188	9.281	9.469

Table 6: Net Attack Cost ($M) by Attacker Size and Duration, Fee Ratio = 0.01, at $100K/BTC Normalization. All Figures Scale Linearly with BTC Price

α	$K = 1$	$K = 2$	$K = 5$	$K = 10$	$K = 26$
0.25	$252K	$32M	$128M	$287M	$796M
0.50	$700K	$107M	$425M	$955M	$2.7B
1.00	$1.6M	$320M	$1.3B	$2.9B	$8.0B
1.50	$2M	$575M	$2.3B	$5.2B	$14.3B
2.00	$3M	$851M	$3.4B	$7.6B	$21.2B
3.00	$4M	$1.4B	$5.7B	$12.9B	$35.8B

Table 7 tests the hash-space exhaustion projections against alternative hash-rate growth assumptions. Under aggressive growth (1-year doubling), the constrained-throughput protocol reaches the 5% threshold by ~2049; under conservative growth (3-year doubling), it is delayed to ~2098. The qualitative ordering is invariant: higher throughput always extends Poisson validity, and the margin between constrained and high-throughput protocols always exceeds 25 years.

Table 7: Sensitivity of the Poisson Breakdown Year to Hash-Rate Growth Rate (5% per-Block Threshold). The Qualitative Ordering is Invariant

System	1.0-yr Doubling	1.5-yr Doubling	2.0-yr Doubling	3.0-yr Doubling
4 TPS	~2049	~2061	~2074	~2098
10^6 TPS	~2067	~2088	~2110	~2152
10^9 TPS	~2077	~2103	~2129	~2182

8. Conclusion

A process is memoryless if and only if $\mathbb{P}(X > t+s \mid X > s) = \mathbb{P}(X > t)$ for all $t, s \geq 0$. The individual hash trial satisfies this property (subject to the finite-domain qualification of Parra-Moyano *et al.*, 2024). The Nakamoto protocol does not.

The decisive mechanism is difficulty adjustment, which updates the per-trial success probability as a deterministic function of past block times. Three further mechanisms—coinbase maturity, fee accumulation,

and heaviest-chain selection—independently ensure that the miner's realized payoff environment is also history-dependent.

Applied to the security framework of Budish (2025), the correction yields a state-space replacement, not a refutation. The equilibrium constraint survives: permissionless proof-of-work trust requires ongoing flow payments large relative to the value of attack. What changes is the stochastic object and the validity domain of the zero-net-cost benchmark:

1. Budish's equilibrium constraint holds under the augmented state space.
2. Budish's "memoryless computational work" characterisation does not hold unqualified.
3. Budish's closed-form attack-race formulas remain good local approximations for short horizons (attacks completing well within one epoch).
4. Budish's Theorem 2 (zero net attack cost) becomes a local benchmark valid within a single fee-free epoch, not a global characterisation.
5. The sufficient statistic for security is a state vector $(g, d_h, d_a, m_h, m_a, k, f)$, not the scalar flow N^*c.

The memoryless approximation is not merely wrong now at Levels 3 and 4. It becomes wrong at Level 1 within the operational lifetime of current mining investments—on a timeline that depends on the protocol's transaction throughput. A throughput-constrained protocol (~4 TPS) reaches the Level-1 Poisson breakdown by ~2061; a high-throughput protocol (~10^9 TPS) extends validity to ~2103. The stochastic foundations of economic security models are themselves functions of protocol design parameters. The protocol becomes more non-memoryless over time at every level of analysis, and the rate of deterioration depends on whether the protocol can sustain security-budget funding through transaction volume rather than through high per-transaction fees. Economic models of proof-of-work mining must specify both the level at which memorylessness is being invoked and the throughput regime under which the analysis is conducted. The within-epoch exponential approximation is useful; the unqualified generalisation is not.

9. Limitations

Four limitations are noted. First, the hash-space projections depend on the rate of hash-rate growth; the 18-month Moore's Law doubling is an illustrative baseline, and the critical dates shift under alternative growth assumptions (the direction of the effect is invariant). Second, the fee-accumulation results are derived in closed form for constant fee arrival rate φ; the extension to stochastic $\varphi(t)$ is noted qualitatively but not characterised in full generality. Third, the claim that DAA instability amplifies attack cost (Remark 5.10) is a qualitative argument about mechanism direction, not a formal proof; a full treatment requires extending the Noda framework to the attack setting, which is left to future work. Fourth, the Monte Carlo simulations assume a fixed BTC price; incorporating endogenous price response to attack would require a general-equilibrium extension. All dollar figures should be read as illustrative at the stated normalisation.

10. Characterization Theorem: Self-Contained Proof

Proof of Theorem 3.7. Continuous Case: Let X be a continuous non-negative random variable satisfying $\mathbb{P}(X > t+s \mid X > s) = \mathbb{P}(X > t)$ for all $t, s \geq 0$. Define $G(t) = \mathbb{P}(X > t)$. The memoryless condition is equivalent to $G(t + s) = G(t)G(s)$ for all $t, s \geq 0$, with $G(0) = 1$ and G non-increasing.

This is the Cauchy functional equation on $[0, \infty)$ with the constraint that G is monotone non-increasing. By a standard result in real analysis (Feller, 1968, Theorem VIII.1.1), every measurable solution to $G(t + s) = G(t)G(s)$ on $[0, \infty)$ with $G(0) = 1$ takes the form $G(t) = e^{ct}$ for some constant c. Since G is non-increasing and $\mathbb{P}(X > t) > 0$ for all t, we require $c < 0$. Writing $\lambda = -c > 0$ gives $G(t) = e^{-\lambda t}$, the survival function of Exp(λ).

Discrete Case: Let X take values in $\mathbb{N}$ and satisfy $\mathbb{P}(X > m + n \mid X > m) = \mathbb{P}(X > n)$ for all $m, n \in \mathbb{N}$. Define $q = G(1) = \mathbb{P}(X > 1)$. By the functional equation, $G(n) = q^n$ for all $n \in \mathbb{N}$. Since G is non-increasing, $0 < q < 1$. Setting $p = 1 - q$: $\mathbb{P}(X = k) = (1 - p)^{k-1}p$ for $k \geq 1$, which is Geom(p).

11. Extended Proofs: Core Results

Extended Proof of Theorem 4.3: We prove that the block-arrival process across epochs fails the memoryless property by showing that the residual waiting-time distribution depends on history.

Step 1—Within-Epoch Rate: During epoch e, the block arrival rate is $\Lambda_e = H_{\text{tot}}/D_e$, where D_e is constant within the epoch.

Step 2—Retarget Transition: At the epoch boundary, $\hat{T}_e = \sum_{i=1}^{n} \tau_{e,i}$ where $\tau_{e,i} \overset{iid}{\sim} Exp(\Lambda_e)$, so $\hat{T}_e \sim Gamma(n, \Lambda_e)$. The protocol sets $D_{e+1} = D_e \cdot T^* / \hat{T}_e$, giving $\Lambda_{e+1} = \Lambda_e \cdot \hat{T}_e / T^*$.

Step 3—History Dependence: Λ_{e+1} is $(\tau_{e,1}, \ldots, \tau_{e,n})$-measurable. Since $\hat{T}_e$ is a continuous random variable, $\hat{T}_e \neq T^*$ almost surely, so $\Lambda_{e+1} \neq \Lambda_e$ almost surely. For any waiting time spanning the epoch boundary:

$$\mathbb{P}(W > s+t \mid W > s) = e^{-\Lambda_{e+1} t} \neq e^{-\Lambda_e t} = \mathbb{P}(W > t)$$

The memoryless property fails at the epoch boundary.

Extended Proof of Proposition 4.5: Let $k(s)$ denote confirmations accumulated by time $t_h + s$, and $\ell(s)$ the depth of the longest competing fork. The reorganization probability satisfies $R(k, \ell) \approx (q/(1-q))^{k-\ell}$ for $q < 1/2$ (Rosenfeld, 2014). Two histories with $M_h(s) = 1$:

- History A: $k(s) = 50$, $\ell(s) = 0$. Residual reorganization probability: $(q/(1-q))^{50}$.
- History B: $k(s) = 5$, $\ell(s) = 3$. Residual reorganization probability: $(q/(1-q))^{2}$.

Since $(q/(1-q))^{2} \gg (q/(1-q))^{50}$, the residual survival probabilities differ. The payoff process is not memoryless. The argument generalises to any model satisfying: (M1) reorganization probability non-increasing in k; (M2) non-decreasing in ℓ; (M3) strict monotonicity for at least one pair.

Extended Proof of Proposition 4.7: The canonical-status process $M_h(t) = 1$ {block h canonical at time t} has transition probabilities depending on $k(t)$, which is $\mathcal{F}_t$-measurable. Different realizations of the block-arrival process produce different $k(s)$, and since the residual reorganization probability is a function of $k(s)$:

$$\mathbb{P}(M_h(s+t) = 1 \mid M_h(s) = 1 \mathcal{F}_s) \neq \mathbb{P}(M_h(t) = 1 \mid M_h(0) = 1)$$

in general. The canonical-status process is not memoryless.

12. Extended Proofs: Economic Propositions

Detailed derivation of Proposition 5.6.

Step 1: With total hashrate $(1 + \alpha)H^*$ and difficulty D^*, expected inter-block time is $\tau^*/(1 + \alpha)$.

Step 2: Expected block value: $\mathbb{E}[V_{\text{att}}] = S + \varphi\tau^*/(1 + \alpha)$.

Step 3: Attacker revenue over n blocks: $\Pi^{(e)} = n\alpha S/(1 + \alpha) + n\alpha\varphi\tau^*/(1 + \alpha)^2$.

Step 4: Attacker cost under zero-profit condition $p^*R = c$: $C^{(e)} = \alpha nR/(1 + \alpha)$.

Step 5: Net cost:

$$C^{(e)} - \Pi^{(e)} = \frac{\alpha n}{1+\alpha}\left[R - s - \frac{\varphi\tau^*}{1+\alpha}\right] = \frac{n\alpha^2\varphi\tau^*}{(1+\alpha)^2}$$

Strictly positive for $\alpha > 0$, $\varphi > 0$. Vanishes when $\varphi = 0$.

Detailed Derivation of Proposition 5.7: Post-retarget difficulty: $D_{e+1} = D^*(1 + \alpha)$. Per-hash revenue: $p_{e+1}R = c/(1 + \alpha) < c$. Per-hash loss: $c\alpha/(1 + \alpha)$. Attacker revenue: $n\alpha R/(1 + \alpha)$. Attacker cost: αnR. Net cost: $n\alpha^2 R/(1 + \alpha)$.

Detailed Derivation of Proposition 5.8: Total hashrate αH^*. Epoch duration: $T^*(1 + \alpha)/\alpha$. Revenue: nR. Cost: $nR(1 + \alpha)$. Net cost: αnR. Comparison: $\alpha nR > n\alpha^2 R/(1 + \alpha)$ since $(1 + \alpha)/\alpha > 1$.

Detailed Derivation of Corollary 5.14: Epoch e: $n\alpha^2\varphi\tau^*/(1 + \alpha)^2$. Epochs $e + 1$ through $e + K - 1$: each contributes $n\alpha^2 R/(1 + \alpha)$ (lower bound under maintained participation). Total:

$$\Gamma(K,\alpha) = \frac{n\alpha^2\varphi\tau^*}{(1+\alpha)^2} + (K-1)\cdot\frac{n\alpha^2 R}{1+\alpha}$$

Positive for $K \geq 1$ if $\varphi > 0$; positive for $K \geq 2$ regardless of φ.

Table 8: Simulation Parameters. All Values Verified from Primary Sources

Parameter	Symbol	Value	Source
Epoch length	n	2,016 blocks	Bitcoin protocol
Target block time	τ^*	600 seconds	Bitcoin protocol
Block subsidy	S	3.125 BTC	Post-April 2024 halving
BTC price		\$100,000	Approximate, early 2025
Subsidy (USD)	S_{USD}	\$312,500	$= S \times$ price
Fee ratio (current)	$\varphi\tau^*/S$	0.01	Blockchain data, late 2024
Monte Carlo paths		10,000	
Random seed		42	Reproducibility

13. Simulation Details

13.1. Parameter Values

13.2. Stochastic Model

Within each epoch, block inter-arrival times are drawn from Exp(Λ) where Λ is the epoch-specific arrival rate. The attacker's block share within each epoch is drawn from Binomial (n, $\alpha/(1 + \alpha)$). At epoch boundaries, difficulty retargets via (2) with the 4 × clamp. The two models (constant-D and augmented) are run on identical random seeds; differences are attributable solely to the retarget mechanism.

13.3. Replication

The simulation code (*attack_simulation.py*) is available from the author. It requires Python 3.8+ with NumPy. Running time: approximately 30 seconds for 10,000 paths.

Data Availability Statement

This paper does not use proprietary data. All parameter values are derived from public sources: the Bitcoin protocol specification, public blockchain data (hash rates, block times), and published financial filings (Visa 10-K FY2024). The replication package containing all simulation code, figure-generation scripts, and computed outputs is available from the author upon request.

References

Ayelet Sapirshtein, Yonatan Sompolinsky and Aviv Zohar. (2017). Optimal Selfish Mining Strategies in Bitcoin. In *Financial Cryptography and Data Security (FC 2016)*, 9603, *Lecture Notes in Computer Science*, 515-532, Springer. doi: 10.1007/978-3-662-54970-4_30.

Boyan Jovanovic. (1982). Selection and the Evolution of Industry. *Econometrica*, 50(3), 649-670. doi: 10.2307/1912606.

Daniel Kraft. (2016). Difficulty Control for Blockchain-Based Consensus Systems. *Peer-to-Peer Networking and Applications*, 9, 397-413. doi: 10.1007/s12083-015-0347-x.

Daron Acemoglu. (2025). 2023 Klein Lecture—Capital and Wages. *International Economic Review,* 66(1), 3-24. doi: 10.1111/iere.12733.

Erhan Çinlar. (2011). Probability and Stochastics. Springer.

Eric Budish. (2025). Trust at Scale: The Economic Limits of Cryptocurrencies and Blockchains. *The Quarterly Journal of Economics*, 140(1), 1-62. doi: 10.1093/qje/qjae033.

Eunseong Ma. (2023). Monetary Policy and Inequality: How Does One Affect the Other?. *International Economic Review*, 64(2), 691-725. doi: 10.1111/iere.12610.

Ittay Eyal and Emin Gün Sirer. (2014). Majority is Not Enough: Bitcoin Mining is Vulnerable. In *Financial Cryptography and Data Security (FC 2014)*, 8437, *Lecture Notes in Computer Science*, 436-454, Springer. doi: 10.1007/978-3-662-45472-5_28.

Jacob D. Leshno and Philipp Strack. (2020). Bitcoin: An Axiomatic Approach and an Impossibility Theorem. *American Economic Review: Insights*, 2(3), 269-286. doi: 10.1257/aeri.20190494.

James D. Hamilton. (1989). A New Approach to the Economic Analysis of Nonstationary Time Series and the Business Cycle. *Econometrica*, 57(2), 357-384. doi: 10.2307/1912559.

Jay L. Devore and Kenneth N. Berk. (2012). *Modern Mathematical Statistics with Applications*, 2nd Editon, Springer. doi: 10.1007/978-1-4614-0391-3.

José Parra-Moyano, Gregor Reich and Karl Schmedders. (2024). A Note on the Non-Proportionality of Winning Probabilities in Bitcoin. *Computational Economics*, 64, 1697-1714. doi: 10.1007/s10614-023-10503-2.

Juan A. Garay, Aggelos Kiayias and Nikos Leonardos. (2015). The Bitcoin Backbone Protocol: Analysis and Applications. In *Advances in Cryptology – EUROCRYPT 2015*, 281-310, Springer. doi: 10.1007/978-3-662-46803-6_10.

Marlon Azinovic, Luca Gaegauf and Simon Scheidegger. (2022). Deep Equilibrium Nets. *International Economic Review*, 63(4), 1471-1525. doi: 10.1111/iere.12575.

Meni Rosenfeld. (2014). Analysis of Hashrate-Based Double Spending. *arXiv preprint arXiv:1402.2009.*

Quan-Lin Li, Jing-Yu Ma and Yan-Xia Chang. (2018). Blockchain Queueing Theory. *arXiv preprint arXiv:1808.01795*. doi: 10.48550/arXiv.1808.01795.

Rhys Bowden, Paul Keeler, H., Anthony E. Krzesinski and Peter G. Taylor. (2020). Modeling and Analysis of Block Arrival Times in the Bitcoin Blockchain. *Stochastic Models*, 36(4), 602-637. doi: 10.1080/15326349.2020.1786404.

Richard Ericson and Ariel Pakes. (1995). Markov-Perfect Industry Dynamics: A Framework for Empirical Work. *Review of Economic Studies*, 62(1), 53-82. doi: 10.2307/2297841.

Robert E. Lucas, Jr. (1976). Econometric Policy Evaluation: A Critique. *Carnegie-Rochester Conference Series on Public Policy*, 1, 19-46. doi: 10.1016/S0167-2231(76)80003-6.

Quan-Lin Li, Jing-Yu Ma, Yan-Xia Chang, Fan-Qi Ma and Hai-Bo Yu (2019). Markov Processes in Blockchain Systems. *Computational Social Networks*, 6, 5. doi: https://doi.org/10.1186/s40649-019-0066-1

Satoshi Nakamoto. (2008). Bitcoin: A Peer-to-Peer Electronic Cash System. URL https://bitcoin.org/bitcoin.pdf

Sheldon M. Ross. (2014). *Introduction to Probability Models*, 11th Edition, Academic Press.

Shunya Noda, Kyohei Okumura and Yoshinori Hashimoto. (2026). An Economic Analysis of Difficulty Adjustment Algorithms in Proof-of-Work Blockchain Systems. *International Economic Review*, 67(1), 259-285. doi: 10.1111/iere.70028.

Visa Inc. (2024). Annual Report on Form 10-K for the Fiscal Year Ended September 30, 2024. *US Securities and Exchange Commission*. https://www.sec.gov/Archives/edgar/data/1403161/000140316124000058/v-20240930.htm

William Feller. (1968). *An Introduction to Probability Theory and its Applications*, 3rd Edition, 1, Wiley.

Cite this article as: Craig S. Wright (2026). Why Hash-Trial Memorylessness Does Not Extend to the Nakamoto Protocol. *International Journal of Cryptocurrency Research*, 6(1), 1-25. doi: 10.67191/IJCCR.6.1.2026.1-25.