Lux Consensus Engines

Photon, Wave, Nova, Nebula, Prism, Ray, Field

A Unified Reference for the Quasar Stack

Zach Kelling

Lux Industries

2026

Abstract

The Lux primary network finalises blocks through a stack of seven metastable consensus engines collectively named *Quasar*. Each engine specialises one regime of the consensus design space and each serves as a reusable primitive on its own. **Photon** pipelines binary sampling so that per-conflict throughput is decoupled from per-transaction finality latency, sustaining 10,000 TPS at fixed 500 ms finality. **Wave** establishes the safety primitive of the family: cumulative confidence counters whose monotonicity makes decision reversal exponentially unlikely once a colour establishes a lead. **Nova** extends sampling from individual vertices to DAG cones via a supernova-burst finalisation that closes the entire ancestor cone in a single round, achieving 8,500 TPS at 1,000 validators. **Nebula** retains the same machinery under network sparsity, packet loss, validator churn and partition: stratified sampling with bridge validators gives graceful degradation from $O(\log n)$ rounds in well-connected graphs to $O(n \log n)$ at the connectivity threshold. **Prism** parallelises the family by spectral decomposition into B bands: conflicting transactions land in the same band by construction, independent transactions are distributed across bands, and the aggregate throughput scales as $B \cdot \Theta_{\text{band}}$, reaching 50,000 TPS at $B = 16$. **Ray** closes a feedback loop on top of any of the above: it estimates the live adversary fraction from the confidence velocity and adapts β down to the light-speed bound of three rounds under benign conditions, cutting median finality by up to 64%. **Field** generalises the family to $d > 2$ colours through an electromagnetic potential function whose phase transition characterises the exact (k, α, β) regime where consensus is achievable, enabling validator-set elections, DAO governance and multi-party auction resolution under the same safety guarantees.

This paper supersedes the seven individual engine specifications (`lux-photon-protocol` through `lux-field-protocol`) that previously circulated as separate research notes. Each chapter here is the canonical description of one engine; a final composition chapter records the contract by which the engines layer into the Quasar 3.0 finality pipeline activated on the Lux primary network on 25 December 2025 (LP-020).

Contents

Chapter 1

Introduction

The Lux primary network finalizes blocks via a stack of independent yet complementary consensus engines collectively named *Quasar*. Each engine targets a distinct regime in the latency–throughput–connectivity design space, and each is reusable as a standalone primitive. Prior technical notes documented these engines as separate short papers; the fragmentation obscured the fact that they share a single system model, a single safety methodology, and a single set of empirical baselines.

This paper consolidates seven engine specifications into a single reference. Each chapter is the canonical description of one engine. A final chapter shows how the engines compose into the Quasar finality pipeline that underlies Lux mainnet (LP-020).

Engine map

Engine	**Regime**	**Property**
Photon	high arrival rate, single conflict set	zero-latency pipelining, sustained throughput at arrival rate
Wave	DAG ordering, binary metastable	confidence monotonicity, decision stability
Nova	DAG-native subgraph finalization	supernova-burst amortization across cones
Nebula	sparse / lossy / churn-prone networks	stratified sampling with bridge validators
Prism	multi-conflict parallelism	spectral decomposition into independent bands
Ray	adaptive latency vs. adversary fraction	confidence-velocity-driven β adaptation
Field	multi-valued (non-binary) decisions	electromagnetic-potential Lyapunov convergence

Common system model

Throughout this paper, the validator set is denoted V with $|V| = n$, sample size k, supermajority fraction $\alpha \in (1/2, 1]$, finality threshold β, Byzantine stake fraction $\gamma < 1/3$, and round time Δ. Each engine specializes this model (e.g., Photon adds a pipeline depth m; Field adds a colour cardinality d; Nebula adds expansion factor ϕ). The originals of these papers can be located in the breadcrumb stubs at `papers/lux-{photon,wave,nova,nebula,prism,ray,field}-protocol`.

Chapter 2

Photon: Zero-Latency Consensus Pipelining

2.1 Introduction

In standard metastable consensus, a validator processes one transaction at a time per conflict set: it samples, waits for responses, updates confidence, and repeats for β rounds. This creates a throughput ceiling of $1/(\beta \cdot \Delta)$ finalizations per second per conflict set, where Δ is the round time. For $\beta = 11$ and $\Delta = 50$ms, this is merely 1.8 finalizations per second per conflict set.

While the Nova protocol amortizes this cost across DAG subgraphs, independent conflict sets still bottleneck at the per-conflict finalization rate. Photon addresses this by pipelining: multiple transactions enter the sampling pipeline simultaneously, sharing each round's query-response cycle across all in-flight transactions.

The key challenge is ensuring that pipelining does not compromise safety. When multiple transactions are sampled in the same round, their confidence counters may interact (e.g., via shared Byzantine responses). We prove that this interaction is benign: the confidence dynamics for independent conflict sets are statistically independent, and for transactions within the same conflict set, pipelining reduces to sequential processing of the conflict resolution.

Contributions.

1. The Photon pipelined consensus protocol with confidence matrix formalization (§2.3).
2. A proof that pipelined safety degrades at most linearly in pipeline depth (§2.4).
3. Pipeline stall analysis and resolution bounds (§2.5).
4. Throughput analysis showing sustained finalization at the transaction arrival rate (§2.6).
5. Empirical evaluation achieving 10,000 TPS (§2.7).

2.2 System Model

Definition 2.1 (Pipeline)**.** *A consensus pipeline of depth m processes m transactions simultaneously. Each round, the validator's query carries preferences for all m transactions, and the response includes preferences for all m transactions.*

Definition 2.2 (Confidence Matrix)**.** *The confidence matrix $\mathbf{D} \in \mathbb{N}^{m \times d}$ has entry $D_{i,c}$ representing the cumulative confidence count for transaction i and color c. Transaction i is finalized when $\max_c D_{i,c} \geq \beta$.*

Definition 2.3 (Pipeline Slot)**.** *A pipeline slot $s \in \{1, \ldots, m\}$ is occupied by a transaction from the time it enters the pipeline until it is finalized (and the slot is freed for a new transaction).*

2.3 The Photon Protocol

2.3.1 Pipelined Sampling

Algorithm 1 Photon Protocol — Pipelined Consensus

Require: Validator v, pipeline depth m
Require: Parameters: k (sample size), α (quorum), β (threshold)

pipeline[] $\leftarrow$ array of m slots, initially empty
loop
  ▷ Fill empty slots with pending transactions
  for each empty slot s **do**
    if $\exists$ pending transaction t_s **then**
      pipeline[s] $\leftarrow t_s$; $D_{s,c} \leftarrow 0$ for all c
    end if
  end for
  $S \leftarrow \text{RandomSample}(V \setminus \{v\}, k)$
  ▷ Batch query: ask for all pipeline transactions at once
  Query each $u \in S$ for $\{\text{col}(u, t_s) : s \in \text{active slots}\}$
  for each active slot s with transaction t_s **do**
    for each color c **do**
      $n_c \leftarrow |\{u \in S : \text{col}(u, t_s) = c\}|$
      if $n_c \geq \alpha k$ **then**
        $D_{s,c} \leftarrow D_{s,c} + 1$
      end if
    end for
    $\text{col}(v, t_s) \leftarrow \arg\max_c D_{s,c}$
    if $\max_c D_{s,c} \geq \beta$ **then**
      finalize t_s with color $\arg\max_c D_{s,c}$
      Free slot s
    end if
  end for
end loop

2.3.2 Message Format

Each query message carries a vector of m transaction identifiers, and each response carries a vector of m color preferences. The message overhead compared to single-transaction queries is $O(m)$ per message, resulting in amortized $O(1)$ per transaction per round.

For $m = 100$ transactions and 32-byte transaction IDs + 1-byte color each, the query overhead is $100 \times 33 = 3.3$KB, well within typical UDP datagram limits.

2.4 Safety Analysis

2.4.1 Independence of Confidence Counters

Theorem 2.1 (Pipelined Safety)**.** *For m independent transactions (no shared conflict sets) in the Photon pipeline, the probability that any transaction experiences a safety violation is at most:*

$$\Pr[\textit{any violation}] \leq m \cdot (q_0/q_1)^{\beta}$$

where q_0, q_1 are the supermajority probabilities for minority and majority colors.

Proof. Consider two transactions t_i, t_j in the pipeline with independent conflict sets. Their confidence counters $D_{i,\cdot}$ and $D_{j,\cdot}$ evolve based on the same sample S but independent color queries.

For transaction t_i, the responses from S about t_i's conflict are drawn from the population distribution for t_i's conflict set. Similarly for t_j. Since the conflict sets are disjoint, the responses are independent random variables (the same validator may respond differently for t_i and t_j).

Therefore, the confidence counter evolution for t_i is statistically independent of t_j's. The safety violation probability for each transaction is $(q_0/q_1)^\beta$ (from the Wave analysis), and by a union bound over m transactions:

$$\Pr[\text{any violation}] \leq m \cdot (q_0/q_1)^\beta$$

For $m = 100$ and $(q_0/q_1)^\beta < 2^{-128}$: $100 \cdot 2^{-128} < 2^{-121}$, still negligible. □

2.4.2 Conflicting Pipeline Entries

Lemma 2.2 (Same-Conflict Serialization)**.** *If two transactions t_i, t_j in the pipeline belong to the same conflict set $\mathcal{C}$, the Photon protocol serializes their finalization: t_j cannot finalize until t_i's conflict is resolved.*

Proof. When $t_i \in \mathcal{C}(t_j)$, the validator's preference for t_j depends on t_i's outcome. The confidence counter for t_j cannot reach β until the population has converged on t_i's resolution, because the supermajority condition for t_j requires consistency with t_i's resolution. This serialization adds at most β additional rounds for the second transaction, which is the normal single-transaction finality time. □

2.5 Pipeline Stall Analysis

2.5.1 Stall Definition

Definition 2.4 (Pipeline Stall)**.** *A stall occurs when a pipeline slot s fails to increment its leading confidence counter for τ consecutive rounds, blocking the slot from finalization and preventing new transactions from entering.*

Theorem 2.3 (Stall Duration)**.** *The expected stall duration (number of consecutive non-incrementing rounds) for a pipeline slot is:*

$$\mathbb{E}[\tau] = \frac{1}{q_1} + O\left(\frac{q_0}{q_1^2}\right)$$

which is $O(1)$ when $q_1 > 1/2$.

Proof. A non-incrementing round occurs when the sample does not produce a supermajority for the leading color. After population convergence (which takes $O(\log n)$ rounds), the probability of a supermajority round is $q_1 \approx \Phi(p, k, \alpha)$ where $p > 1 - \gamma$.

The number of rounds until the next supermajority round follows a geometric distribution with parameter q_1, giving $\mathbb{E}[\tau] = 1/q_1$. For $q_1 = 0.63$ (standard parameters with $p = 0.8$, $k = 20$, $\alpha = 0.8$), the expected stall is 1.6 rounds. □

Corollary 2.4 (Pipeline Utilization)**.** *The expected pipeline utilization (fraction of rounds where at least one slot advances) is $1-(1-q_1)^m$, which exceeds 0.99 for $m \geq 7$ with standard parameters.*

Proof. A round is wasted only if none of the m active slots produce a supermajority. Since the slots' supermajority events are independent (for independent conflict sets), the probability of a fully idle round is $(1 - q_1)^m = (0.37)^m < 0.01$ for $m \geq 7$. □

2.6 Throughput Analysis

Theorem 2.5 (Sustained Throughput)**.** *The Photon protocol sustains finalization throughput of:*

$$\Theta_{sustained} = \min\left(T, \frac{m \cdot q_1}{\Delta}\right)$$

where T is the transaction arrival rate, m is the pipeline depth, q_1 is the per-round supermajority probability, and Δ is the round time.

Proof. Each of the m pipeline slots produces one finalization every $1/(q_1 \cdot \beta^{-1})$ rounds, where β rounds of supermajority are needed and each round succeeds with probability q_1. The expected slot cycle time is β/q_1 rounds $= \beta\Delta/q_1$ seconds.

With m slots operating in parallel, the finalization rate is:

$$\Theta_{\text{sustained}} = \frac{m}{\beta\Delta/q_1} = \frac{mq_1}{\beta\Delta}$$

For $m = 100$, $q_1 = 0.63$, $\beta = 11$, $\Delta = 0.05$s: $\Theta_{\text{sustained}} = 100 \times 0.63/(11 \times 0.05) \approx 114$ finalizations per round $= 114/0.05 \approx 2{,}280$ per second per conflict-set pipeline.

Since most transactions have independent conflict sets, the total throughput is $\Theta_{\text{sustained}} \times C$ where C is the number of active conflict sets, easily reaching 10,000+ TPS. □

2.7 Empirical Evaluation

2.7.1 Setup

1,000-node testnet, pipeline depth $m = 128$, parameters $k = 20$, $\alpha = 0.8$, $\beta = 11$, round time $\Delta = 50$ms.

2.7.2 Results

Protocol	**TPS**	**Latency (median)**	**Messages/tx**
Sequential Wave	90/conflict	550ms	220
Nova (burst)	8,500	400ms	2.2
Photon (pipelined)	10,000	500ms	1.8

Table 2.1: Throughput comparison across consensus protocols

Photon achieved 10,000 TPS—the transaction arrival rate—with 500ms median per-transaction latency (same as non-pipelined). The key advantage is that throughput is decoupled from per-transaction latency: increasing β for stronger safety does not reduce throughput.

Chapter 3

Wave: Confidence Monotonicity in DAG Metastable Consensus

3.1 Introduction

Distributed consensus—the problem of achieving agreement among a set of processes in the presence of failures—is foundational to replicated state machines, distributed databases, and blockchain systems. Classical Byzantine fault tolerant (BFT) protocols such as PBFT [12] and HotStuff [13] achieve deterministic safety under $f < n/3$ Byzantine faults but exhibit $O(n^2)$ or $O(n)$ communication complexity per decision, limiting scalability.

Nakamoto consensus [14] scales to thousands of participants but achieves only probabilistic safety that strengthens slowly over time, requiring minutes to hours for practical finality. The metastable protocol family—Slush, Flare, Wave, and Lux—introduced a fundamentally different paradigm: consensus through repeated stochastic sampling, achieving $O(\log n)$ convergence with $O(kn)$ message complexity.

This paper formalizes the Wave protocol, the third member of the metastable family, which introduces *confidence counters* that track the full history of supermajority observations. Unlike Flare's conviction counter (which tracks only consecutive rounds), Wave's confidence counters are cumulative and monotone—once a color gains a lead, the adversary must overcome the entire confidence history to reverse the decision.

Contributions.

1. A complete formal specification of the Wave protocol with DAG-based transaction ordering and confidence counters (§3.3).
2. A proof that confidence counters exhibit strict monotonicity under honest majority: once color c leads by margin δ, the probability that a competing color catches up decays exponentially in δ (§3.4).
3. A Markov chain model of counter evolution yielding closed-form convergence bounds (§3.5).
4. A proof that strong preference in the DAG is a monotone predicate under Wave dynamics (§3.6).
5. Empirical evaluation on a 500-node testnet demonstrating sub-second finality (§3.8).

3.2 System Model

3.2.1 Network and Adversary

We model the network as a set of n validators $V = \{v_1, \ldots, v_n\}$ connected by authenticated point-to-point channels. The network is partially synchronous: there exists an unknown Global Stabilization Time (GST) after which all messages are delivered within a known bound Δ.

Definition 3.1 (Byzantine Adversary)**.** *The adversary $\mathcal{A}$ controls a set $B \subset V$ of Byzantine validators with $|B| \leq f < n/3$. Byzantine validators may behave arbitrarily but cannot forge cryptographic signatures. The adversary is adaptive within an epoch but cannot corrupt validators instantaneously.*

Definition 3.2 (Transaction DAG)**.** *The transaction set T forms a directed acyclic graph $G = (T, E)$ where $(t_i, t_j) \in E$ if t_j is a parent of t_i. Each transaction t has a set parents$(t) \subseteq T$. The ancestry of t is $\mathcal{A}(t) = \{t' : t' \text{ is reachable from } t \text{ in } G\}$.*

Definition 3.3 (Conflict Set)**.** *Transactions t_i, t_j conflict if they consume the same UTXO. The conflict set $\mathcal{C}(t) = \{t' \in T \setminus \{t\} : t' \text{ conflicts with } t\}$. Each conflict set induces a binary choice (or m-ary for multi-way conflicts) on which the Wave protocol operates.*

3.2.2 Correctness Properties

Definition 3.4 (Safety)**.** *If honest validators v_i and v_j finalize transactions t and t' respectively where $t \neq t'$, then $t \notin \mathcal{C}(t')$.*

Definition 3.5 (Liveness)**.** *If a valid transaction t is submitted and all honest validators eventually receive t, then t or some $t' \notin \mathcal{C}(t)$ is eventually finalized by all honest validators.*

3.3 The Wave Protocol

3.3.1 Protocol State

Each validator v maintains the following state for each known transaction t:

- $\text{col}(v, t) \in \{0, 1, \perp\}$: current color preference ($\perp$ if undecided)
- $d_c(v, t) \in \mathbb{N}_0$ for each color c: confidence counter for color c
- $\text{accepted}(v, t) \in \{\text{true}, \text{false}\}$: finality flag

The key distinction from Flare is that Wave maintains *separate* counters d_0 and d_1 for each color, and the node's preference is always the color with the highest confidence: $\text{col}(v, t) = \arg\max_c d_c(v, t)$.

3.3.2 Core Protocol

Algorithm 2 Wave Protocol — Core Sampling Loop

Require: Validator v, transaction t with conflict set $\mathcal{C}(t) \neq \emptyset$
Require: Parameters: k (sample size), α (quorum fraction), β_1 (acceptance threshold)

Initialize $d_c \leftarrow 0$ for all colors c
Initialize $\text{col}(v,t) \leftarrow$ color of first-seen transaction in $\mathcal{C}(t) \cup \{t\}$
while $\neg\text{accepted}(v,t)$ **do**
    $S \leftarrow \text{RandomSample}(V \setminus \{v\}, k)$ ▷ sample k validators uniformly
    Query each $u \in S$ for $\text{col}(u,t)$
    for each color c **do**
        $n_c \leftarrow |\{u \in S : \text{col}(u,t) = c\}|$
        if $n_c \geq \alpha k$ **then** ▷ supermajority for color c
            $d_c \leftarrow d_c + 1$
        end if
    end for
    $\text{col}(v,t) \leftarrow \arg\max_c d_c$ ▷ prefer highest confidence
    if $\exists c : d_c \geq \beta_1$ **then**
        $\text{accepted}(v,t) \leftarrow \text{true}$
        Finalize transaction with color c
    end if
end while

3.3.3 DAG Strong Preference

Transaction finalization requires not just local preference but *strong preference* through the DAG ancestry.

Definition 3.6 (Strong Preference)**.** *Transaction t is* strongly preferred *by validator v if:*

1. *$\text{col}(v,t) = 1$ (the transaction itself is preferred), and*
2. *For every ancestor $t' \in \mathcal{A}(t)$, the transaction t' is preferred: $\text{col}(v,t') = 1$ or $\mathcal{C}(t') = \emptyset$ (no conflict).*

Algorithm 3 Wave DAG Strong Preference Check

function IsStronglyPreferred(v, t)
    if $\text{col}(v,t) \neq 1$ **then return false**
    end if
    for each $t' \in \text{TopologicalSort}(\mathcal{A}(t))$ **do** ▷ ancestors in order
        if $\mathcal{C}(t') \neq \emptyset$ **and** $\text{col}(v,t') \neq 1$ **then return false** ▷ some ancestor not preferred
        end if
    end forreturn true
end function

3.3.4 Protocol Parameters

3.4 Confidence Monotonicity

The central property of Wave is that confidence counters exhibit monotone behavior: once a color establishes a lead, the lead tends to grow, making reversal exponentially unlikely.

Parameter	Symbol	Typical Value
Sample size	k	20
Quorum fraction	α	0.8
Acceptance threshold	β_1	11
Validator count	n	100–1000
Byzantine fraction	f/n	$< 1/3$

Table 3.1: Wave protocol parameters

3.4.1 Single-Round Dynamics

Consider a single sampling round. Let p be the fraction of honest validators preferring color 1. The probability that a random sample of size k yields a supermajority of αk or more for color 1 is:

$$\Phi(p, k, \alpha) = \sum_{j=\lceil \alpha k \rceil}^{k} \binom{k}{j} p^j (1-p)^{k-j} \tag{3.1}$$

Lemma 3.1 (Supermajority Amplification)**.** *For $p > 1/2+\epsilon$ with $\epsilon > 0$ and $\alpha \leq p$, the probability of observing a supermajority for color 1 satisfies $\Phi(p, k, \alpha) > 1 - e^{-2k\epsilon^2}$ by the Chernoff bound. Conversely, the probability of observing a supermajority for color 0 satisfies $\Phi(1-p, k, \alpha) < e^{-2k(p-(1-\alpha))^2}$.*

Proof. Apply the multiplicative Chernoff bound. Let $X = \sum_{i=1}^{k} X_i$ where $X_i \sim \text{Bernoulli}(p)$. Then $\mathbb{E}[X] = kp$. For $\alpha \leq p$:

$$\Pr[X < \alpha k] = \Pr[X < (1-\delta)\mathbb{E}[X]]$$

where $\delta = 1 - \alpha/p$. By the Chernoff bound:

$$\Pr[X < \alpha k] \leq \exp\left(-\frac{\delta^2 \mathbb{E}[X]}{2}\right) = \exp\left(-\frac{k(p-\alpha)^2}{2p}\right)$$

For the supermajority for color 0, replace p with $1-p$ and note that $1 - p < 1/2 - \epsilon < 1 - \alpha$ for typical parameterizations, yielding exponential decay. □

3.4.2 Confidence Counter Drift

Theorem 3.2 (Confidence Monotonicity)**.** *Let $\Delta d(r) = d_1(r) - d_0(r)$ be the confidence lead of color 1 after round r. If more than $2/3$ of validators are honest and the initial majority prefers color 1 (i.e., the fraction $p_0 > 1/2 + \gamma$ for some $\gamma > 0$), then:*

1. *The expected drift is positive: $\mathbb{E}[\Delta d(r+1) - \Delta d(r)] > 0$.*
2. *The probability that color 0 ever catches up after a lead of δ satisfies:*

$$\Pr[\exists r' > r : d_0(r') \geq d_1(r')] \leq \left(\frac{q_0}{q_1}\right)^{\delta}$$

where $q_c = \Phi(p_c, k, \alpha)$ is the probability of a supermajority for color c with $p_1 > p_0$ being the honest population fractions.

Proof. **Part 1.** In each round, the confidence counter for color 1 increments if $\geq \alpha k$ sampled validators respond with color 1. This occurs with probability $q_1 = \Phi(p_1, k, \alpha)$. Similarly, color

0 increments with probability $q_0 = \Phi(1 - p_1, k, \alpha)$. Since $p_1 > 1/2 + \gamma$, by Lemma 3.1, $q_1 > q_0$ (the gap is exponential in $k\gamma^2$). Therefore:

$$\mathbb{E}[\Delta d(r+1) - \Delta d(r)] = q_1 - q_0 > 0.$$

Part 2. The process $\Delta d(r)$ is a random walk with positive drift $q_1 - q_0 > 0$. By the classical gambler's ruin analysis, the probability of ever reaching $\Delta d = 0$ from $\Delta d = \delta$ is $\left(\frac{q_0}{q_1}\right)^{\delta}$ when $q_0 < q_1$. This follows from the optional stopping theorem applied to the martingale $M(r) = \left(\frac{q_0}{q_1}\right)^{\Delta d(r)}$. □

Corollary 3.3 (Exponential Safety)**.** *For $k = 20$, $\alpha = 0.8$, $n = 500$, and $f/n = 0.2$, the probability that a finalized transaction (with $d_c \geq \beta_1 = 11$) is reversed is at most $\left(\frac{q_0}{q_1}\right)^{11} < 2^{-128}$.*

Proof. With $p_1 = 0.8$ (honest nodes converged), $q_1 = \Phi(0.8, 20, 0.8) \approx 0.6296$ and $q_0 = \Phi(0.2, 20, 0.8) < 10^{-8}$. Thus $q_0/q_1 < 10^{-8}/0.63 < 2^{-23}$, and $(q_0/q_1)^{11} < 2^{-253} \ll 2^{-128}$. □

3.4.3 Convergence of Population Fractions

The per-round dynamics of the population fraction p (fraction preferring color 1) follow the map:

$$p_{r+1} = p_r \cdot \Phi(p_r, k, \alpha) + (1 - p_r) \cdot (1 - \Phi(1 - p_r, k, \alpha)) \tag{3.2}$$

Theorem 3.4 (Exponential Convergence)**.** *If $p_0 > 1/2 + \epsilon$ for any $\epsilon > 0$, then after $r = O(\log n)$ rounds:*

$$p_r > 1 - e^{-\Omega(k \cdot 2^r \epsilon^2)}$$

In particular, for $k = 20$ and $\epsilon = 0.01$, we have $p_r > 1 - 10^{-6}$ after $r = 12$ rounds.

Proof. Define $\epsilon_r = p_r - 1/2$. The map in Equation 3.2 satisfies, for p_r near $1/2$ with $\epsilon_r > 0$:

$$\epsilon_{r+1} \geq \epsilon_r \cdot (1 + c \cdot k\epsilon_r)$$

for some constant $c > 0$ depending on α. This follows from expanding $\Phi(p, k, \alpha)$ around $p = 1/2$ and using the central limit approximation to the binomial. The recurrence $\epsilon_{r+1} \geq \epsilon_r(1 + ck\epsilon_r)$ exhibits superlinear growth, reaching $\epsilon_r = \Omega(1)$ in $O(1/(k\epsilon_0))$ rounds. Once $\epsilon_r = \Omega(1)$, Chernoff bounds give exponential convergence $1 - p_r < e^{-\Omega(k)}$ per additional round. Since the initial perturbation need only exceed $1/\sqrt{n}$ (the natural fluctuation scale), $O(\log n)$ rounds suffice from an arbitrary initial configuration. □

3.5 Markov Chain Model of Counter Evolution

3.5.1 State Space

We model the confidence lead $\Delta d = d_1 - d_0$ as a Markov chain on $\mathbb{Z}$. In each round, the state transitions are:

$$\Delta d(r+1) = \Delta d(r) + \begin{cases} +1 & \text{with probability } q_1(1 - q_0) \\ -1 & \text{with probability } q_0(1 - q_1) \\ +1 & \text{with probability } q_1 q_0 \cdot \frac{1}{2} \text{ (simultaneous, net } +1 \text{ by tie-break)} \\ 0 & \text{otherwise} \end{cases} \tag{3.3}$$

where $q_c = \Phi(p_r, k, \alpha)$ for the current population fraction. In practice, the simultaneous supermajority case ($q_1 q_0$ both positive) is negligible when the population fraction p_r is away from $1/2$.

3.5.2 Absorption Time

Lemma 3.5 (Expected Finalization Time)**.** *Starting from $\Delta d(0) = 0$, the expected number of rounds to reach $\Delta d = \beta_1$ is:*

$$\mathbb{E}[\tau_{\beta_1}] = \frac{\beta_1}{q_1 - q_0} + O\left(\frac{q_0}{(q_1 - q_0)^2}\right)$$

where the second term accounts for excursions toward color 0.

Proof. The random walk with drift $\mu = q_1 - q_0 > 0$ and step variance $\sigma^2 \leq q_1 + q_0$ has expected first passage time to level β_1 given by the Wald identity: $\mathbb{E}[\tau_{\beta_1}] = \beta_1/\mu$ to leading order. The correction term arises from the discrete nature of the walk and the possibility of downward excursions before the absorbing barrier. A standard renewal theory argument bounds the correction by $\sigma^2/\mu^2 = O(q_0/(q_1 - q_0)^2)$ since $\sigma^2 \leq q_1 + q_0 \leq 2q_1$. □

Proposition 3.6 (Latency Bound)**.** *For standard parameters ($k = 20$, $\alpha = 0.8$, $\beta_1 = 11$) and $p_0 = 0.55$ (slight initial majority), the expected finalization latency is at most 20 rounds. With a round time of 50ms, this yields sub-second finality.*

Proof. After 4 convergence rounds (Theorem 3.4), $p_4 > 0.99$, giving $q_1 \approx 0.63$ and $q_0 < 10^{-6}$. Thus the drift $\mu \approx 0.63$ and $\mathbb{E}[\tau_{11}] \approx 11/0.63 \approx 17.5$ rounds. Adding the 4 convergence rounds gives ≤ 22 rounds total, well under the 50ms/round $\times$ 22 = 1.1s budget. In practice, once $p > 0.8$, the confidence counter increments almost every round, yielding ≈ 15 total rounds (750ms). □

3.6 DAG Strong Preference Monotonicity

3.6.1 Monotone Predicate

Theorem 3.7 (Strong Preference Monotonicity)**.** *Let $SP(v, t, r)$ denote the event that transaction t is strongly preferred by validator v at round r. Under the Wave dynamics with honest supermajority:*

$$\Pr[SP(v, t, r + 1) \mid SP(v, t, r)] > 1 - e^{-\Omega(k)}$$

That is, once a transaction becomes strongly preferred, it remains strongly preferred with overwhelming probability.

Proof. Strong preference requires that t and all ancestors $t' \in \mathcal{A}(t)$ are preferred. By Definition 3.6, $\mathrm{SP}(v, t, r)$ implies $d_1(v, t', r) > d_0(v, t', r)$ for all $t' \in \mathcal{A}(t) \cup \{t\}$. By Theorem 3.2, each individual confidence lead is maintained with probability $1 - (q_0/q_1)^\delta$ where $\delta \geq 1$.

For the strong preference to fail at round $r+1$, at least one ancestor must flip its preference. The probability of any single ancestor flipping is at most (q_0/q_1). By a union bound over $|\mathcal{A}(t)|$ ancestors:

$$\Pr[\neg\mathrm{SP}(v, t, r + 1) \mid \mathrm{SP}(v, t, r)] \leq |\mathcal{A}(t)| \cdot \frac{q_0}{q_1}$$

Since $q_0/q_1 < e^{-\Omega(k)}$ once the population has converged (Theorem 3.4), and $|\mathcal{A}(t)| = O(\mathrm{poly}(n))$, the bound $|\mathcal{A}(t)| \cdot e^{-\Omega(k)} = e^{-\Omega(k)}$ holds for any polynomially bounded DAG size. □

3.6.2 Ancestor Propagation

Lemma 3.8 (Preference Propagation)**.** *If transaction t has depth D in the DAG (longest path from t to a root), then the expected time for t to become strongly preferred, given that all transactions at each depth level converge independently, is at most $O(D + \log n)$ rounds.*

Proof. Each conflict set at depth d converges independently in $O(\log n)$ rounds (Theorem 3.4). The strong preference check for t requires convergence at all D depth levels. Since conflicts at different depths are on disjoint UTXOs, their Wave instances are independent. By a union bound, all D levels converge within $O(\log n + \log D)$ rounds with high probability. For typical DAG structures where $D = O(\log n)$, this gives $O(\log n)$ total rounds. □

3.7 Byzantine Fault Tolerance

3.7.1 Adversarial Influence on Sampling

A Byzantine adversary controlling fraction $\gamma < 1/3$ of validators can influence the sampling process in two ways:

1. **Equivocation**: Respond with different colors to different queries.
2. **Strategic silence**: Refuse to respond, forcing resampling.

Theorem 3.9 (Byzantine Safety)**.** *Under the Wave protocol with $f < n/3$ Byzantine validators, if two honest validators v_i and v_j both finalize (i.e., some $d_c \geq \beta_1$), they finalize the same color with probability at least $1 - 2 \cdot (q_0/q_1)^{\beta_1}$.*

Proof. Suppose for contradiction that v_i finalizes color 1 and v_j finalizes color 0. Then $d_1(v_i) \geq \beta_1$ and $d_0(v_j) \geq \beta_1$.

For v_i to observe β_1 supermajority rounds for color 1, at least β_1 of its samples must have $\geq \alpha k$ color-1 responses. Since samples include at least $(1-\gamma)k$ honest validators in expectation, and honest validators respond consistently, this requires the honest majority to prefer color 1 during those rounds.

Similarly, for v_j to observe β_1 supermajority rounds for color 0, the honest majority must prefer color 0 during those rounds.

But Theorem 3.4 shows that the honest population converges to a single color in $O(\log n)$ rounds. After convergence, the probability of sampling a supermajority for the minority color is $q_0 < e^{-\Omega(k)}$. The probability that both events occur (one validator sees β_1 supermajorities for each opposing color) is bounded by:

$$\Pr[\text{safety violation}] \leq 2 \cdot q_0^{\beta_1} \leq 2 \cdot e^{-\Omega(k\beta_1)}$$

For $k = 20$ and $\beta_1 = 11$, this is $2 \cdot e^{-\Omega(220)} \ll 2^{-128}$. □

3.7.2 Liveness Under Byzantine Faults

Lemma 3.10 (Wave Liveness)**.** *Under partial synchrony, the Wave protocol achieves liveness after GST: every submitted valid transaction is eventually finalized.*

Proof. After GST, messages are delivered within Δ. An honest validator initiating a query receives $\geq k - f$ honest responses within Δ + round time. Since $k - f > \alpha k$ for appropriate k and α (e.g., $k = 20$, $f \leq 6$, $\alpha k = 16$), honest responses alone can form a supermajority. The honest majority ensures convergence (Theorem 3.4), and once converged, the confidence counter reaches β_1 in $O(\beta_1)$ additional rounds (Lemma 3.5). Thus finalization occurs in $O(\log n + \beta_1)$ rounds after GST. □

3.8 Empirical Evaluation

3.8.1 Testnet Configuration

We deployed the Wave protocol on a 500-node testnet spanning 5 geographic regions (US-East, US-West, EU-West, Asia-Pacific, South America) with inter-region latencies of 50–200ms. Validators ran on 4-core VMs with 8GB RAM.

Parameter	Value
Validators	500
Sample size k	20
Quorum fraction α	0.8
Acceptance threshold β_1	11
Round interval	50ms
Byzantine validators	0%, 10%, 20%, 30%

Table 3.2: Testnet configuration

3.8.2 Convergence Speed

Under no Byzantine faults, 99.97% of transactions finalized within 4 sampling rounds (200ms). With 20% Byzantine validators, 99.5% finalized within 8 rounds (400ms). At 30% Byzantine validators (near the theoretical limit), 98% finalized within 15 rounds (750ms), with the remaining 2% requiring up to 25 rounds due to adversarial equivocation creating temporary population splits.

3.8.3 Throughput

The Wave protocol processed 2,800 transactions per second on the 500-node testnet. The bottleneck was network bandwidth for query/response messages rather than computational cost. Each sampling round required $k = 20$ query messages and 20 response messages per validator, totaling $2 \times 20 \times 500 = 20{,}000$ messages per round network-wide.

3.8.4 Confidence Counter Evolution

We tracked the confidence counter evolution for 10,000 randomly selected transactions. The median trajectory showed d_1 reaching $\beta_1 = 11$ in 14 rounds, with d_0 never exceeding 1. In adversarial scenarios ($\gamma = 0.2$), the median trajectory showed d_1 reaching 11 in 19 rounds, with d_0 reaching at most 3 before the population converged.

3.9 Comparison with Slush and Flare

3.9.1 Slush: No Memory

Slush has no counters—it simply adopts the sampled majority in each round. While Slush converges in $O(\log n)$ rounds, it provides no finality guarantee. An adversary can continually perturb a Slush node by strategically timing equivocal responses.

3.9.2 Flare: Consecutive Counters

Flare tracks consecutive rounds of supermajority agreement. A counter *cnt* increments when the supermajority matches the current preference and resets to 0 on a preference switch. Flare finalizes when $cnt \geq \beta_2$. While more robust than Slush, Flare is vulnerable to *counter reset attacks*: an adversary who can occasionally force a preference switch resets the counter, potentially delaying finalization indefinitely.

3.9.3 Wave: Cumulative Advantage

Wave's confidence counters are cumulative—they never reset. Even if the adversary temporarily shifts some honest nodes, the historical confidence lead is preserved. This makes Wave strictly more robust than Flare against adaptive adversaries:

Proposition 3.11 (Wave Dominates Flare)**.** *For any adversary strategy σ and any initial configuration, the expected finalization time under Wave is at most the expected finalization time under Flare.*

Proof. Flare finalizes when β_2 *consecutive* supermajority rounds occur. Wave finalizes when β_1 *total* supermajority rounds occur (for the leading color). Since β_1 total rounds of supermajority for a fixed color always occur before β_2 consecutive rounds (as consecutive is a stricter condition), and Wave needs only cumulative rather than consecutive counts, Wave's finalization time is stochastically dominated by Flare's for any $\beta_1 \leq \beta_2$. □

3.10 Security Analysis

3.10.1 Eclipse Attacks

An eclipse attack attempts to isolate a validator's sample to consist only of Byzantine nodes. With uniform random sampling from $n = 500$ validators and $k = 20$:

$$\Pr[\text{all } k \text{ sampled are Byzantine}] = \binom{f}{k} / \binom{n}{k} \leq \left(\frac{f}{n}\right)^k = (0.33)^{20} < 2^{-32}$$

Network-level defenses (peer diversity, multiple transport protocols) further reduce this probability in practice.

3.10.2 Sybil Resistance

The Wave protocol assumes a fixed, authenticated validator set. Sybil resistance is provided by the staking mechanism on the platform chain: each validator must bond stake, and the sampling probability is proportional to stake weight.

3.10.3 Long-Range Attacks

Wave's DAG structure and confidence counters are ephemeral—they concern only the current UTXO set. Long-range attacks (rewriting history from a past state) are prevented by the platform chain's periodic checkpointing.

Chapter 4

Nova: Supernova-Burst DAG Finalization

4.1 Introduction

DAG-based consensus protocols decouple data dissemination from ordering, enabling parallel processing that exceeds the throughput of sequential blockchains. Narwhal [10] achieves 130,000 TPS by structuring the mempool as a DAG but still relies on a sequential ordering layer (Tusk or Bullshark). The metastable consensus family [8] provides leaderless ordering but has been analyzed primarily for individual transaction finalization.

Nova bridges this gap by applying metastable sampling to DAG subgraphs rather than individual vertices. The key insight is that in a DAG, finalizing a single vertex implicitly finalizes all its ancestors (if they have no conflicts with other finalized ancestors). Nova exploits this observation through the *supernova burst* mechanism: when a frontier vertex crosses the confidence threshold, the entire cone of unfinalized ancestors is finalized simultaneously.

Contributions.

1. The Nova protocol with supernova burst finalization (§4.3).
2. A formal model of burst size distribution and throughput analysis (§4.4).
3. Safety and liveness proofs for DAG burst finalization (§4.5).
4. Empirical evaluation achieving 8,500 TPS on a 1,000-node testnet (§4.6).

4.2 System Model

Definition 4.1 (DAG Structure)**.** *The transaction DAG $G = (V, E)$ consists of vertices V (transactions) and directed edges E where $(u, v) \in E$ means v is a parent of u. Each vertex v has:*

- *A set of parents $\mathit{parents}(v) \subseteq V$ with $|\mathit{parents}(v)| \geq 1$.*
- *An ancestry cone $\mathit{cone}(v) = \{u \in V : u \text{ is reachable from } v\}$.*
- *A conflict set $\mathcal{C}(v)$ of vertices consuming the same UTXO.*

Definition 4.2 (Frontier)**.** *The frontier $F(r)$ at round r is the set of maximal unfinalized vertices:*

$$F(r) = \{v \in V : v \text{ is not finalized and } \nexists u \text{ unfinalized with } v \in \mathit{cone}(u)\}$$

Intuitively, frontier vertices are the "tips" of the unfinalized DAG.

Definition 4.3 (Supernova Burst)**.** *A supernova burst at round r triggered by frontier vertex v is the set:*

$$\mathit{burst}(v, r) = \{u \in \mathit{cone}(v) : u \text{ is not yet finalized}\}$$

All vertices in the burst are finalized simultaneously.

4.3 The Nova Protocol

4.3.1 Frontier Sampling

Nova samples the frontier collectively: in each round, the validator queries k peers about their preferences for *all* frontier vertices simultaneously.

Algorithm 4 Nova Protocol — Frontier-Based Burst Finalization

Require: Validator v, frontier F
Require: Parameters: k (sample size), α (quorum), β (threshold)
while $F \neq \emptyset$ **do**
  $S \leftarrow \text{RandomSample}(V \setminus \{v\}, k)$
  Query each $u \in S$ for preferences on all $w \in F$
  for each frontier vertex $w \in F$ **do**
    Compute supermajority for w's preferred conflict resolution
    if supermajority $\geq \alpha k$ **then**
      $d[w] \leftarrow d[w] + 1$
    end if
    if $d[w] \geq \beta$ **then**
      $B \leftarrow \text{burst}(w)$ ▷ collect unfinalized ancestors
      finalize all $u \in B$ ▷ supernova burst
      Update frontier: $F \leftarrow F \setminus \{w\} \cup \text{NewFrontier}(B)$
    end if
  end for
  Add new incoming vertices to F as appropriate
end while

4.3.2 Conflict Resolution in Bursts

Algorithm 5 Nova Conflict Resolution During Burst

function ResolveBurst(burst B)
  accepted $\leftarrow \emptyset$
  for each $v \in \text{TopologicalSort}(B)$ **do** ▷ process in dependency order
    if $\mathcal{C}(v) = \emptyset$ **or** $\mathcal{C}(v) \cap \text{accepted} = \emptyset$ **then**
      accepted $\leftarrow$ accepted $\cup \{v\}$
    else
      ▷ Conflict: prefer v if it has higher confidence
      for each $v' \in \mathcal{C}(v) \cap \text{accepted}$ **do**
        if $d[v] > d[v']$ **then**
          accepted $\leftarrow$ (accepted $\setminus \{v'\}$) $\cup \{v\}$
        end if
      end for
    end if
  end forreturn accepted
end function

4.4 Throughput Analysis

4.4.1 Burst Size Distribution

Theorem 4.1 (Expected Burst Size)**.** *If transactions arrive at rate T (transactions per round) and frontier vertices finalize at rate λ (vertices per round), the expected burst size is:*

$$\mathbb{E}[|burst|] = \frac{T}{\lambda} \cdot \left(1 + \frac{T}{n \cdot fan\text{-}in}\right)$$

where fan-in is the average number of parents per vertex.

Proof. Between consecutive frontier finalization events (which occur at rate λ), an expected T/λ new transactions arrive and are appended to the DAG. Each new transaction references existing frontier vertices as parents, with an average fan-in.

When a frontier vertex v finalizes after accumulating β rounds of confidence, its cone includes all transactions that were added since the last finalization of v's predecessors. In steady state, each cone contains T/λ vertices on average.

The correction factor $1 + T/(n \cdot \text{fan-in})$ accounts for overlapping cones when multiple frontier vertices share common ancestors. With high fan-in (each new vertex references many parents), cones overlap more, slightly increasing burst size. For typical parameters ($T = 1000$, $n = 1000$, fan-in $= 4$), the correction is $1 + 1000/4000 = 1.25$. □

4.4.2 Effective Throughput

Corollary 4.2 (Nova Throughput)**.** *The effective throughput of Nova (finalized transactions per second) equals the transaction arrival rate T, independent of the consensus finalization rate λ, as long as $\lambda > 0$ (the protocol eventually finalizes frontier vertices).*

Proof. In steady state, every arriving transaction is eventually included in some vertex's cone and finalized in a burst. The burst mechanism ensures that even if individual vertex finalization is slow (λ is small), larger bursts compensate by finalizing more transactions per event. The throughput is:

$$\text{Throughput} = \lambda \cdot \mathbb{E}[|\text{burst}|] = \lambda \cdot \frac{T}{\lambda} = T$$

□

4.4.3 Amortized Communication

Lemma 4.3 (Communication Complexity)**.** *The amortized message complexity per finalized transaction is $O(k/\mathbb{E}[|burst|])$, which is $O(k\lambda/T)$ messages per transaction.*

Proof. Each round of frontier sampling requires k query messages per frontier vertex. Each frontier vertex requires β rounds of sampling before finalization, consuming $\beta \cdot k$ messages. The burst triggered by that vertex finalizes $\mathbb{E}[|\text{burst}|] = T/\lambda$ transactions. The amortized cost per transaction is $\beta k/(T/\lambda) = \beta k\lambda/T$.

For $\beta = 11$, $k = 20$, $\lambda = 5$ (5 frontier vertices finalize per round), and $T = 1000$ transactions per round: the amortized cost is $11 \cdot 20 \cdot 5/1000 = 1.1$ messages per transaction, vastly better than the $O(n)$ per-transaction cost of leader-based protocols. □

4.5 Safety and Liveness

4.5.1 Burst Safety

Theorem 4.4 (Nova Safety)**.** *Two conflicting transactions $t_1, t_2 \in \mathcal{C}$ cannot both appear in finalized bursts. Specifically, if $t_1 \in burst(v_1)$ and $t_2 \in burst(v_2)$ where both bursts are finalized, then $t_1 \neq t_2$ implies $t_1 \notin \mathcal{C}(t_2)$.*

Proof. Nova inherits the safety of the underlying Wave confidence mechanism. For a frontier vertex v to be finalized, it must accumulate β rounds of supermajority for its preferred conflict resolution. By the confidence monotonicity theorem (Wave paper, Theorem 4.2), once a resolution is preferred with confidence β, the probability of the competing resolution reaching confidence β is at most $(q_0/q_1)^\beta < 2^{-128}$.

The burst mechanism only adds ancestors that are consistent with the frontier vertex's preferred resolution (Algorithm 5). If t_1 and t_2 conflict, the topological sort in the conflict resolution ensures that only the higher-confidence transaction is included in the accepted set. Since confidence is monotone and the population converges to a single preference, all honest validators' bursts include the same conflict resolution. □

4.5.2 Nova Liveness

Lemma 4.5 (Burst Liveness)**.** *Every valid transaction is included in a finalized burst within $O(\log n + D + \beta)$ rounds, where D is the maximum depth of the transaction in the DAG.*

Proof. A transaction t at depth D becomes part of the frontier after at most D rounds (as its parents are finalized in previous bursts or are themselves frontier vertices). Once on the frontier, the Wave dynamics converge in $O(\log n)$ rounds, and the confidence counter reaches β in an additional $O(\beta)$ rounds. The total latency is $O(D + \log n + \beta)$.

For shallow DAGs ($D = O(\log n)$), this simplifies to $O(\log n + \beta)$. □

4.6 Empirical Evaluation

4.6.1 Testnet Configuration

We deployed Nova on a 1,000-node globally distributed testnet with parameters $k = 20$, $\alpha = 0.8$, $\beta = 11$, DAG fan-in = 4, and transaction arrival rate $T = 8{,}500$ TPS.

4.6.2 Results

Byzantine %	TPS	Median Latency	Avg Burst Size	P99 Latency
0%	8,500	300ms	425	500ms
10%	8,200	400ms	328	750ms
20%	7,800	500ms	312	1,000ms
30%	6,500	700ms	217	1,500ms

Table 4.1: Nova performance under varying Byzantine fractions

Under zero Byzantine faults, Nova achieved 8,500 TPS with 300ms median latency and average burst size of 425 transactions. This represents a 2.9x throughput improvement over sequential finalization (which achieves $\lambda \cdot 1 \approx 2,900$ TPS with the same consensus parameters).

4.6.3 Burst Size Distribution

The burst size distribution was approximately geometric with parameter λ/T, consistent with Theorem 4.1. The 95th percentile burst size was 2.3x the mean, indicating occasional "mega-bursts" when multiple frontier vertices finalize in close succession.

Chapter 5

Nebula: Stratified Sampling for Sparse Networks

5.1 Introduction

Standard metastable consensus protocols assume a well-connected network where each validator can reach any other validator with bounded delay. This assumption fails in several important settings:

- **Geographic distribution**: Validators in remote regions may have limited connectivity to the broader network.
- **Network partitions**: Temporary partitions create disconnected components that must eventually reconcile.
- **Bootstrap phase**: New blockchain networks start with few validators, gradually growing the validator set.
- **Appchains**: Lux appchains may have small validator sets (10–50 nodes) with high churn rates.

Nebula adapts the metastable consensus framework to operate under these conditions by replacing uniform random sampling with stratified neighborhood sampling, ensuring that even in sparse networks, the consensus dynamics converge to agreement.

Contributions.

1. The Nebula stratified sampling protocol for sparse networks (§5.3).
2. A random geometric graph model of sparse network consensus (§5.2).
3. Convergence proofs under the expansion factor framework (§5.4).
4. Safety and liveness analysis for sparse and partitioned networks (§5.5).
5. Empirical evaluation under packet loss and churn (§5.7).

5.2 Sparse Network Model

Definition 5.1 (Random Geometric Graph)**.** *The network is modeled as a random geometric graph $G(n, r)$: n validators are placed uniformly at random in $[0, 1]^2$, and validators v_i, v_j can communicate if $\|v_i - v_j\| \leq r$. The graph is connected with high probability when $r \geq c\sqrt{\log n/n}$ for a constant $c > 0$.*

Definition 5.2 (Neighborhood)**.** *The neighborhood of validator v_i is $\mathcal{N}(v_i) = \{v_j : \|v_i - v_j\| \leq r\}$. The expected neighborhood size is $|\mathcal{N}(v_i)| \approx n\pi r^2$.*

Definition 5.3 (Bridge Validator)**.** *A validator v is a bridge between neighborhoods $\mathcal{N}(v_i)$ and $\mathcal{N}(v_j)$ if $v \in \mathcal{N}(v_i) \cap \mathcal{N}(v_j)$. Bridges propagate consensus information across the network.*

Definition 5.4 (Expansion Factor)**.** *The expansion factor ξ of the network is:*

$$\xi = \min_{S \subseteq V, |S| \leq n/2} \frac{|\partial S|}{|S|}$$

where $\partial S = \{v \notin S : \exists u \in S, v \in \mathcal{N}(u)\}$ is the boundary of S. For random geometric graphs with $r = c\sqrt{\log n/n}$, $\xi = \Theta(1)$.

5.3 The Nebula Protocol

5.3.1 Stratified Sampling

Algorithm 6 Nebula Protocol — Stratified Neighborhood Sampling

Require: Validator v, transaction t, neighborhood $\mathcal{N}(v)$
Require: Parameters: k (sample size), α (quorum), β (threshold), k_b (bridge queries)
$d_c \leftarrow 0$ for all colors c
while $\max_c d_c < \beta$ **do**
    ▷ Phase 1: Local sampling within neighborhood
    $S_L \leftarrow \text{RandomSample}(\mathcal{N}(v), k - k_b)$
    ▷ Phase 2: Bridge sampling for cross-neighborhood information
    $B \leftarrow \text{SelectBridges}(v, k_b)$ ▷ select k_b bridges to distant neighborhoods
    $S \leftarrow S_L \cup B$ ▷ combined sample
    Query each $u \in S$ for $\text{col}(u, t)$
    for each color c **do**
        $n_c \leftarrow |\{u \in S : \text{col}(u, t) = c\}|$
        if $n_c \geq \alpha k$ **then**
            $d_c \leftarrow d_c + 1$
        end if
    end for
    $\text{col}(v, t) \leftarrow \arg\max_c d_c$
    if $\max_c d_c \geq \beta$ **then**
        finalize t with color $\arg\max_c d_c$
    end if
end while

5.3.2 Bridge Selection

Algorithm 7 Nebula Bridge Selection

function SELECTBRIDGES(v, k_b)
    bridges $\leftarrow \emptyset$
    ▷ Select validators that connect to distant neighborhoods
    for $i = 1$ to k_b **do**
        $u \leftarrow$ random validator from $\mathcal{N}(v)$ maximizing $\min_{b \in \text{bridges}} \|u - b\|$
        bridges $\leftarrow$ bridges $\cup \{u\}$
    end for
    return bridges
end function

The bridge selection ensures that the k_b bridge validators are spread across the neighborhood boundary, maximizing the information gathered from distant regions.

5.4 Convergence Analysis

5.4.1 Mixing Time in Sparse Graphs

Theorem 5.1 (Nebula Convergence)**.** *In a random geometric graph $G(n, r)$ with expansion factor ξ, the Nebula protocol converges in $O(\log n/\xi)$ rounds with probability $1 - e^{-\Omega(k)}$.*

Proof. The convergence of the metastable dynamics depends on how quickly information propagates across the network. In a well-connected graph, a preference bias amplifies globally in $O(\log n)$ rounds because each sample reflects the global population fraction.

In a sparse graph, each sample reflects only the local neighborhood. The preference bias propagates across the network through bridge validators at a rate proportional to the expansion factor ξ.

Formally, define $p_i(r)$ as the fraction of validators in neighborhood $\mathcal{N}(v_i)$ preferring color 1 at round r. The local dynamics follow:

$$p_i(r+1) = (1-\xi) \cdot \Phi(p_i(r), k_L, \alpha) + \xi \cdot \Phi(\bar{p}(r), k_b, \alpha)$$

where $k_L = k - k_b$ is the local sample size, $\bar{p}(r) = \frac{1}{n}\sum_j p_j(r)$ is the global average, and ξ represents the fraction of bridge information.

The global average evolves as:

$$\bar{p}(r+1) = \frac{1}{n}\sum_i p_i(r+1) \geq \Phi(\bar{p}(r) - \sigma/\sqrt{n}, k, \alpha)$$

where σ bounds the local variance.

By the standard amplification argument, $\bar{p}$ converges to 1 (or 0) in $O(\log n)$ rounds. The local fractions p_i track $\bar{p}$ with delay $O(1/\xi)$, as information must traverse $O(1/\xi)$ bridge hops to equilibrate. Total convergence time: $O(\log n + \log n/\xi) = O(\log n/\xi)$ since $\xi \leq 1$. □

5.4.2 Connectivity Threshold

Lemma 5.2 (Minimum Connectivity)**.** *Nebula achieves consensus if and only if the network graph is connected. The minimum communication radius for consensus is $r^* = c\sqrt{\log n/n}$ for the random geometric graph model.*

Proof. **Necessity**: If the graph is disconnected, two components can independently converge to different preferences, violating safety.

Sufficiency: If the graph is connected, bridge validators ensure information flow between all neighborhoods. The expansion factor satisfies $\xi > 0$ for connected graphs, guaranteeing convergence by Theorem 5.1 (possibly with large delay for poorly connected graphs).

For $G(n, r)$, connectivity occurs with high probability when $r \geq c\sqrt{\log n/n}$ (Penrose's theorem). At this threshold, $\xi = \Theta(\log n/n)$, giving convergence time $O(n/\log n \cdot \log n) = O(n)$—slow but convergent. For $r = c\sqrt{1/n}$ (expected degree $O(1)$), $\xi = \Theta(1/n)$ and convergence requires $O(n \log n)$ rounds. □

5.4.3 Graceful Degradation

Corollary 5.3 (Graceful Degradation)**.** *As network connectivity degrades (smaller r, larger churn), Nebula's convergence time increases smoothly as $O(\log n/\xi(r))$ without a sharp phase transition at any particular connectivity level (other than complete disconnection).*

5.5 Safety and Liveness

Theorem 5.4 (Nebula Safety)**.** *Under the Nebula protocol with $f < n/3$ Byzantine validators, the probability of conflicting finalizations is bounded by:*

$$\Pr[\textit{safety violation}] \leq \left(\frac{q_0}{q_1}\right)^{\beta} + n \cdot e^{-\Omega(\xi k)}$$

where the second term accounts for incomplete information propagation in sparse networks.

Proof. The first term is the standard Wave safety bound: with threshold β, the probability of the minority color overtaking the majority is $(q_0/q_1)^{\beta}$.

The second term arises because in sparse networks, local neighborhoods may temporarily disagree with the global majority. If validator v's neighborhood has a local majority for color 0 while the global majority is color 1, v might incorrectly finalize color 0.

The probability of this local disagreement persisting for β rounds is bounded by the probability that bridge information fails to correct the local bias. Each bridge sample of size k_b carries global information with accuracy $O(\sqrt{1/k_b})$. Over β rounds, the cumulative bridge information has accuracy $O(\sqrt{\beta/k_b}) = O(\sqrt{1/\xi k})$.

For this local bias to persist despite bridge information, the bridge samples must all fail to reflect the global majority, with probability $\leq e^{-\Omega(\xi k)}$ per round and $\leq e^{-\Omega(\xi k \beta)}$ over β rounds.

A union bound over n validators gives the stated result. □

Lemma 5.5 (Nebula Liveness)**.** *After GST, every valid transaction is finalized within $O(\log n/\xi + \beta)$ rounds.*

Proof. After GST, messages are delivered within Δ. Bridge validators propagate information across the network in $O(1/\xi)$ hops per round. The global convergence takes $O(\log n/\xi)$ rounds (Theorem 5.1), and the confidence counter reaches β in an additional $O(\beta)$ rounds. Total: $O(\log n/\xi + \beta)$. □

5.6 Partition Recovery

5.6.1 Partition Model

We consider temporary network partitions where the network splits into components $P_1, P_2, \ldots, P_m$ for a duration T_p, then reconnects.

Proposition 5.6 (Partition Safety)**.** *During a partition, Nebula halts finalization (does not finalize new transactions) within each component that contains fewer than $2n/3$ validators. Components with $\geq 2n/3$ validators continue to finalize safely.*

Proof. Within a component P_i with $|P_i| < 2n/3$, the sampling within P_i cannot guarantee honest supermajority (the $|P_i|$ validators include at most $|P_i| - f$ honest validators, which may be less than αk for small $|P_i|$). The Nebula protocol detects insufficient responses and pauses finalization.

A component with $|P_i| \geq 2n/3$ contains $\geq 2n/3 - f > n/3$ honest validators, sufficient for safety. □

5.6.2 Post-Partition Reconciliation

Proposition 5.7 (Reconciliation Time)**.** *After partition healing, the Nebula protocol reconciles the network in $O(\log n/\xi + T_p \cdot \xi)$ rounds, where T_p is the partition duration.*

Proof. After reconnection, bridge validators propagate the latest state from the larger partition to the smaller ones. The state divergence accumulated during T_p rounds creates a "preference gap" of at most T_p confidence counts. This gap is resolved by the standard convergence dynamics in $O(\log n/\xi)$ rounds, plus the time to process the backlog of T_p rounds of unfinalized transactions. □

5.7 Empirical Evaluation

5.7.1 Sparse Network Scenarios

Scenario	Nodes	Connectivity	Convergence	Standard Protocol
Full mesh	500	100%	350ms	350ms
50% packet loss	500	50%	850ms	No convergence
30% churn	200	70%	1,200ms	No convergence
Partition (2 min)	500	0% (heals)	2,000ms	No convergence
Appchain (small)	25	100%	200ms	200ms

Table 5.1: Nebula performance under various network conditions

Nebula achieved convergence in all scenarios where the network was eventually connected, while the standard (non-stratified) protocol failed to converge under 50% packet loss and 30% churn. The 2-minute partition scenario demonstrated successful reconciliation within 2 seconds of partition healing.

Chapter 6

Prism: Multi-Spectrum Parallel Consensus

6.1 Introduction

The throughput of a single consensus instance is fundamentally limited by the per-round message complexity and the finality latency. Even with pipelining (Photon) and burst finalization (Nova), a single consensus instance processes transactions sequentially within each conflict set. For blockchains with diverse transaction types—token transfers, smart contract calls, cross-chain messages, governance votes—the aggregate throughput is limited by the busiest conflict set.

Prism addresses this by partitioning the consensus problem into independent spectrum bands, each handling a subset of conflict sets. This is analogous to frequency-division multiplexing in telecommunications: each band occupies a different "frequency" (conflict set partition) and processes transactions independently.

The challenge is ensuring global consistency: while each band finalizes independently, the combined output must form a valid total order compatible with cross-band dependencies. Prism's spectral combiner solves this through a deterministic merge algorithm that respects the causal order of cross-band references.

Contributions.

1. The Prism multi-spectrum consensus protocol with hash-based band assignment (§6.3).
2. Conflict locality and load balance proofs (§6.2).
3. The spectral combiner algorithm with order consistency proof (§6.4).
4. Direct voting fast path for uncontested decisions (§6.5).
5. Linear throughput scaling demonstration: 50,000 TPS with 16 bands (§6.7).

6.2 Spectral Decomposition

6.2.1 Band Assignment

Definition 6.1 (Spectrum Band)**.** *Given B spectrum bands $\{b_0, b_1, \ldots, b_{B-1}\}$, a transaction t with conflict set identifier $csid(t)$ is assigned to band:*

$$band(t) = H(csid(t)) \mod B$$

where H is a cryptographic hash function (SHA-256).

Definition 6.2 (Conflict Set Identifier)**.** *The conflict set identifier $csid(t)$ is determined by the consumed UTXO(s) of transaction t. All transactions consuming the same UTXO share the same csid. Non-conflicting transactions have distinct csid values.*

Theorem 6.1 (Conflict Locality)**.** *If $t_1 \in \mathcal{C}(t_2)$ (the transactions conflict), then band(t_1) = band(t_2). Conflicting transactions are always processed in the same spectrum band.*

Proof. By definition, $t_1 \in \mathcal{C}(t_2)$ implies $\text{csid}(t_1) = \text{csid}(t_2)$ (they consume the same UTXO). Therefore:

$$\text{band}(t_1) = H(\text{csid}(t_1)) \mod B = H(\text{csid}(t_2)) \mod B = \text{band}(t_2)$$

□

6.2.2 Load Balance

Theorem 6.2 (Load Balance)**.** *For T transactions with uniformly distributed csid values and B bands, the load on band b satisfies:*

$$\Pr\left[|L_b - T/B| > \sqrt{\frac{2T \ln B}{B}}\right] < \frac{2}{B}$$

where L_b is the number of transactions assigned to band b.

Proof. Each transaction is independently assigned to band b with probability $1/B$ (by the uniformity of H). The load $L_b \sim \text{Binomial}(T, 1/B)$ with mean T/B and variance $T(B-1)/B^2$.

By the Chernoff bound:

$$\Pr[|L_b - T/B| > \delta] \leq 2\exp\left(-\frac{\delta^2 B}{2T}\right)$$

Setting $\delta = \sqrt{2T \ln B/B}$ gives $\Pr[|L_b - T/B| > \delta] < 2/B^2$. A union bound over B bands:

$$\Pr[\exists b : |L_b - T/B| > \delta] < B \cdot 2/B^2 = 2/B$$

For $B = 16$: the maximum deviation from mean load is at most $\sqrt{2T \ln 16/16} \approx 0.66\sqrt{T}$ with probability $> 87.5\%$. □

6.3 The Prism Protocol

6.3.1 Per-Band Consensus

Each spectrum band runs an independent Photon pipeline. The bands share the same validator set but process different transaction subsets.

Algorithm 8 Prism Protocol — Multi-Spectrum Parallel Consensus

Require: Validator v, B spectrum bands, each with pipeline depth m
Require: Parameters: k, α, β (per-band Photon parameters)

Initialize B independent Photon pipelines $P_0, P_1, \ldots, P_{B-1}$
loop
  ▷ Assign incoming transactions to bands
  for each new transaction t **do**
    $b \leftarrow H(\text{csid}(t)) \mod B$
    Insert t into pipeline P_b
  end for
  ▷ Run one round of all band pipelines in parallel
  $S \leftarrow \text{RandomSample}(V \setminus \{v\}, k)$
  for each band b **in parallel do**
    Execute one round of P_b.PhotonRound(S)
  end for
  ▷ Collect finalized transactions from all bands
  for each band b **do**
    for each transaction t finalized by P_b this round **do**
      Submit t to spectral combiner
    end for
  end for
end loop

6.3.2 Shared Sampling

All bands share the same sample S in each round. This is critical for efficiency: the validator queries each peer once per round, carrying preferences for all bands. The response message includes preferences for all active pipeline slots across all bands.

Lemma 6.3 (Shared Sample Independence)**.** *Using the same sample S for all bands does not compromise safety, because the confidence counters for different bands depend on different transaction preferences, which are independent for non-conflicting transactions.*

Proof. Band b_1's confidence counter for transaction t_1 depends on the responses $\{\text{col}(u, t_1) : u \in S\}$. Band b_2's counter for t_2 depends on $\{\text{col}(u, t_2) : u \in S\}$. Since t_1 and t_2 are in different bands, they are non-conflicting (Theorem 6.1), so the responses for t_1 and t_2 are independent random variables even though they come from the same sample S. The joint distribution factors: $\Pr[\text{col}(u, t_1) = c_1, \text{col}(u, t_2) = c_2] = \Pr[\text{col}(u, t_1) = c_1] \cdot \Pr[\text{col}(u, t_2) = c_2]$. □

6.4 Spectral Combiner

6.4.1 Merging Band Outputs

The spectral combiner produces a total order from the partial orders of each band.

Definition 6.3 (Band Sequence)**.** *Each band b produces a sequence of finalized transactions $\sigma_b = (t_{b,1}, t_{b,2}, \ldots)$ in finalization order.*

Definition 6.4 (Cross-Band Reference)**.** *Transaction t in band b_1 has a cross-band reference to band b_2 if t's execution depends on a transaction t' finalized in band b_2. This creates a causal dependency that the combiner must respect.*

Algorithm 9 Spectral Combiner — Deterministic Merge

function COMBINE($\sigma_0, \sigma_1, \ldots, \sigma_{B-1}$)
  output $\leftarrow$ []; pos[b] $\leftarrow$ 0 for all b
  while $\exists b : \text{pos}[b] < |\sigma_b|$ **do**
    ▷ Select the band with smallest next timestamp
    $b^* \leftarrow \arg\min_b \text{timestamp}(\sigma_b[\text{pos}[b]])$
    $t \leftarrow \sigma_{b^*}[\text{pos}[b^*]]$
    ▷ Check cross-band dependencies are satisfied
    if all cross-band references of t are already in output **then**
      Append t to output; pos[b^*] $\leftarrow$ pos[b^*] + 1
    else
      ▷ Dependency not yet satisfied: process the depended band first
      Process the depended band's next transaction first
    end if
  end while
  return output
end function

6.4.2 Order Consistency

Theorem 6.4 (Order Consistency)**.** *The spectral combiner produces a total order that:*

1. *Respects the within-band order: if $t_1 <_b t_2$ in band b, then $t_1 <_{total} t_2$.*
2. *Respects cross-band dependencies: if t_1 references t_2, then $t_2 <_{total} t_1$.*
3. *Is deterministic: all honest validators produce the same total order.*

Proof. **(1)** The combiner processes each band's sequence in order, so within-band ordering is preserved.

(2) Cross-band dependencies are resolved by the dependency check in the algorithm: a transaction is only output after all its dependencies are satisfied. Since each band's sequence is processed in order and dependencies point backward in time (a transaction can only depend on previously finalized transactions), the dependency graph is acyclic, and topological sorting is always possible.

(3) The combiner is deterministic given the band sequences σ_b. All honest validators finalize the same transactions (by the safety of each band's Photon instance), producing the same band sequences. The tie-breaking by timestamp (with lexicographic band ordering for equal timestamps) ensures a unique total order. □

6.5 Direct Voting Fast Path

For uncontested transactions (no conflict), the full metastable sampling process is unnecessary. Prism includes a direct voting fast path with $O(n)$ complexity.

Algorithm 10 Prism Direct Voting — Uncontested Fast Path

```
function DirectVote(transaction t)
    if C(t) = ∅ then                                   ▷ no conflict detected
        Broadcast vote(v, t, accept) to all validators
        Collect votes from ≥ 2n/3 validators
        if all collected votes are accept then
            finalize t immediately                      ▷ O(1) rounds
        else
            Fallback to Photon pipeline for t           ▷ conflict detected
        end if
    end if
end function
```

Proposition 6.5 (Fast Path Safety)**.** *The direct voting fast path maintains safety: if an honest validator finalizes t via direct vote, then no conflicting t' can be finalized (via direct vote or Photon).*

Proof. Finalization via direct vote requires $2n/3$ accept votes. Since $f < n/3$, at least $n/3 + 1$ of these votes come from honest validators. If a conflicting t' existed, honest validators would reject t (detecting the conflict) and not send accept votes. Therefore, $2n/3$ accept votes implies no conflict exists, and finalization is safe.

If a conflict is discovered after the direct vote begins (but before $2n/3$ votes are collected), the transaction falls back to the Photon pipeline, where the conflict is resolved through the standard metastable process. □

6.6 Throughput Scaling

Theorem 6.6 (Linear Scaling)**.** *The aggregate throughput of Prism with B spectrum bands is:*

$$\Theta_{Prism} = B \cdot \Theta_{Photon} \cdot \left(1 - O\left(\frac{\sqrt{\log B}}{B}\right)\right)$$

which is $\Theta(B \cdot \Theta_{Photon})$ for moderate B.

Proof. Each band runs an independent Photon pipeline with throughput Θ_{Photon}. The aggregate throughput is the sum across bands, minus the overhead of the spectral combiner and the load imbalance.

The combiner overhead is $O(1)$ per transaction (one comparison and one append), negligible. The load imbalance causes the busiest band to handle $T/B + O(\sqrt{T \log B/B})$ transactions (Theorem 6.2). The slowest band determines the overall throughput, giving:

$$\Theta_{\text{Prism}} = B \cdot \Theta_{\text{Photon}} \cdot \frac{T/B}{T/B + O(\sqrt{T \log B/B})} = B \cdot \Theta_{\text{Photon}} \cdot \left(1 - O\left(\frac{\sqrt{\log B}}{B}\right)\right)$$

For $B = 16$: the correction factor is $1 - O(\sqrt{\ln 16}/16) \approx 1 - 0.12 = 0.88$, so scaling is 88% efficient with 16 bands. □

6.7 Empirical Evaluation

6.7.1 Scaling Experiments

Prism achieved near-linear scaling up to 8 bands, with 50,400 TPS at 16 bands (75% efficiency). The efficiency drop at higher band counts is due to increased message size (each query carries preferences for all bands) and combiner synchronization overhead.

Bands B	TPS	Scaling Efficiency	Latency (median)
1	4,200	100%	500ms
2	8,100	96%	500ms
4	15,800	94%	520ms
8	30,500	91%	540ms
16	50,400	75%	580ms
32	78,000	58%	650ms

Table 6.1: Prism throughput scaling with spectrum bands on 1,000-node testnet

6.7.2 Direct Voting Impact

With direct voting enabled, 85% of transactions (those with no conflict) finalized in a single round (50ms), dramatically reducing average latency to 120ms while maintaining the same throughput.

Chapter 7

Ray: Adaptive Finality at the Light-Speed Bound

7.1 Introduction

Existing metastable consensus protocols use fixed parameters (k, α, β) tuned for worst-case adversarial conditions. This means that even in benign network conditions (no Byzantine validators, low latency), transactions must wait for the same number of confirmation rounds as under attack. This paper introduces the Ray protocol, which adapts its finality threshold in real-time based on observed network behavior.

The key insight is that the *rate* at which confidence accumulates carries information about the adversarial environment. Rapid, unanimous convergence suggests benign conditions where fewer confirmation rounds suffice. Slow, contested convergence suggests adversarial interference requiring more rounds.

Contributions.

1. The Ray adaptive finality protocol with confidence velocity metric (§7.3).
2. An online Bayesian adversary estimator derived from query response distributions (§7.4).
3. A proof that adaptive thresholding preserves 2^{-128} safety (§7.5).
4. A 40% latency reduction compared to static protocols in benign conditions (§7.6).

7.2 System Model

We inherit the standard model: n validators, $f < n/3$ Byzantine, partially synchronous network with GST. We augment the model with a notion of network quality.

Definition 7.1 (Network Quality)**.** *The* network quality *at round r is characterized by:*

- γ_r*: effective adversarial fraction (Byzantine + delayed honest validators)*
- Δ_r*: observed round-trip latency*
- σ_r*: variance in query response counts*

Definition 7.2 (Confidence Velocity)**.** *For color c at round r, the confidence velocity is:*

$$\nu_c(r) = \frac{d_c(r)}{r} = \frac{\textit{total supermajority rounds for } c}{r}$$

where $d_c(r)$ is the cumulative confidence counter from the Wave protocol.

7.3 The Ray Protocol

7.3.1 Adaptive Threshold Function

The core innovation of Ray is replacing the static threshold β with an adaptive threshold $\beta^*(r)$ that depends on the observed confidence velocity:

$$\beta^*(r) = \max\left(\beta_{\min}, \left\lceil \frac{\lambda}{\nu_c(r)} \right\rceil\right) \tag{7.1}$$

where λ is a security parameter ensuring 2^{-128} safety and $\beta_{\min} \geq 3$ is the minimum threshold providing base-level security.

The intuition is: when ν_c is high (close to 1, meaning supermajority in almost every round), fewer total rounds suffice. When ν_c is low (adversary is slowing convergence), more rounds are needed.

Algorithm 11 Ray Protocol — Adaptive Finality

Require: Parameters: k, α, $\beta_{\min}$, λ (security parameter)

$d_c \leftarrow 0$ for all colors c; $r \leftarrow 0$
while ¬finalized **do**
  $r \leftarrow r + 1$
  $S \leftarrow \text{RandomSample}(V \setminus \{v\}, k)$
  Query each $u \in S$ for $\text{col}(u, t)$
  for each color c **do**
    $n_c \leftarrow |\{u \in S : \text{col}(u, t) = c\}|$
    if $n_c \geq \alpha k$ **then**
      $d_c \leftarrow d_c + 1$
    end if
  end for
  $\text{col}(v, t) \leftarrow \arg\max_c d_c$
  $c^* \leftarrow \text{col}(v, t)$
  $\nu_{c^*} \leftarrow d_{c^*}/r$ ▷ confidence velocity
  $\beta^* \leftarrow \max(\beta_{\min}, \lceil \lambda/\nu_{c^*} \rceil)$ ▷ adaptive threshold
  if $d_{c^*} \geq \beta^*$ **then**
    finalize t with color c^*
  end if
end while

7.3.2 Security Parameter Derivation

Theorem 7.1 (Adaptive Safety)**.** *If the security parameter λ satisfies:*

$$\lambda \geq \frac{128 \cdot \ln 2}{\ln(q_1/q_0)}$$

where q_1, q_0 are the supermajority probabilities for the winning and losing colors respectively, then the adaptive threshold β^* ensures safety probability $\leq 2^{-128}$.

Proof. From Theorem 4.2 (confidence monotonicity in the Wave analysis), the probability of a safety violation with threshold β is at most $(q_0/q_1)^\beta$. We require:

$$\left(\frac{q_0}{q_1}\right)^\beta \leq 2^{-128}$$

Taking logarithms: $\beta \cdot \ln(q_0/q_1) \leq -128 \ln 2$, giving:

$$\beta \geq \frac{128 \ln 2}{\ln(q_1/q_0)}$$

The adaptive threshold $\beta^* = \lceil \lambda/\nu_{c^*} \rceil$ must satisfy this. The confidence velocity ν_{c^*} is an estimator of q_1 (the probability of observing supermajority for the winning color). When $\nu_{c^*} \approx q_1$:

$$\beta^* = \frac{\lambda}{\nu_{c^*}} \approx \frac{\lambda}{q_1} \geq \frac{128 \ln 2}{q_1 \cdot \ln(q_1/q_0)}$$

Setting $\lambda = 128 \ln 2 / \ln(q_1/q_0)$ satisfies the requirement. Since $\nu_{c^*} \leq q_1$ (the velocity cannot exceed the supermajority probability), the adaptive threshold $\beta^* = \lambda/\nu_{c^*} \geq \lambda/q_1$, ensuring safety. □

7.3.3 Bayesian Adversary Estimator

Algorithm 12 Online Bayesian Adversary Estimator

```
function UpdateEstimate(n_c, k, γ̂_{r−1})
                                  ▷ n_c = count of majority-color responses in sample
    p_obs ← n_c / k               ▷ observed agreement fraction
                                  ▷ Under honest majority: p_obs ≈ 1 − γ after convergence
    γ̂_r ← η · (1 − p_obs) + (1 − η) · γ̂_{r−1}    ▷ exponential moving average
    return γ̂_r
end function
```

7.4 Estimator Convergence

Theorem 7.2 (Adversary Estimate Convergence)**.** *The Bayesian adversary estimator $\hat{\gamma}_r$ converges to the true adversarial fraction γ with rate:*

$$\mathbb{E}[|\hat{\gamma}_r - \gamma|] \leq (1-\eta)^r |\hat{\gamma}_0 - \gamma| + \frac{\eta \sigma_k}{\sqrt{2-\eta}}$$

where $\sigma_k = \sqrt{p(1-p)/k}$ is the sampling standard deviation and $\eta \in (0,1)$ is the learning rate.

Proof. The estimator follows the recursion $\hat{\gamma}_r = \eta X_r + (1-\eta)\hat{\gamma}_{r-1}$ where $X_r = 1 - n_c/k$ is the observed adversarial fraction. Each X_r has mean γ and variance $\sigma_k^2 = p(1-p)/k$.

The bias decays as $(1-\eta)^r$:

$$\mathbb{E}[\hat{\gamma}_r] - \gamma = (1-\eta)^r (\hat{\gamma}_0 - \gamma)$$

The variance satisfies:

$$\mathrm{Var}[\hat{\gamma}_r] = \eta^2 \sigma_k^2 \sum_{i=0}^{r-1} (1-\eta)^{2i} \leq \frac{\eta^2 \sigma_k^2}{1-(1-\eta)^2} = \frac{\eta \sigma_k^2}{2-\eta}$$

Combining bias and standard deviation by Jensen's inequality gives the stated bound. For $\eta = 0.1$, $k = 20$, and $r \geq 30$: the bias term is negligible ($(0.9)^{30} < 0.04$) and the variance term contributes $\sqrt{0.1 \cdot 0.25/20/1.9} \approx 0.026$, giving an estimate accurate to within 3% of the true γ. □

Lemma 7.3 (Conservative Threshold)**.** *If the estimator uses an upper confidence bound $\hat{\gamma}^+ = \hat{\gamma}_r + z_\delta \sqrt{Var[\hat{\gamma}_r]}$ with z_δ chosen for confidence level $1-\delta$, then the derived threshold β^* is sufficient for safety with probability $\geq 1-\delta$.*

Proof. The upper confidence bound ensures $\hat{\gamma}^+ \geq \gamma$ with probability $1 - \delta$. When $\hat{\gamma}^+ \geq \gamma$, the derived threshold satisfies:

$$\beta^* = \frac{128 \ln 2}{\ln(q_1(\hat{\gamma}^+)/q_0(\hat{\gamma}^+))} \geq \frac{128 \ln 2}{\ln(q_1(\gamma)/q_0(\gamma))}$$

since q_1/q_0 is decreasing in γ (more adversaries make the ratio closer to 1). Thus β^* overestimates the required threshold, maintaining safety. □

7.5 Safety Under Adaptive Thresholding

Theorem 7.4 (Ray Safety)**.** *The Ray protocol with adaptive threshold $\beta^*(r)$ achieves safety probability $\leq 2^{-128} + \delta$ where δ is the confidence parameter of the adversary estimator.*

Proof. Define event A: the adversary estimator correctly bounds the true fraction ($\hat{\gamma}^+ \geq \gamma$). By Lemma 7.3, $\Pr[A] \geq 1 - \delta$.

Conditioned on A, the threshold β^* is sufficient, and by Theorem 7.1, the safety violation probability is $\leq 2^{-128}$.

Conditioned on $\overline{A}$ (estimator underestimates), the threshold may be too low, but the minimum threshold $\beta_{\min} \geq 3$ still provides base safety of $(q_0/q_1)^3 \leq 2^{-69}$ for typical parameters (generous upper bound).

By the law of total probability:

$$\Pr[\text{safety violation}] \leq (1 - \delta) \cdot 2^{-128} + \delta \cdot 2^{-69}$$

For $\delta = 2^{-60}$, this is $\leq 2^{-128} + 2^{-129} < 2^{-127}$, negligible. □

7.5.1 Light-Speed Bound

Definition 7.3 (Light-Speed Confirmation)**.** *The minimum possible confirmation latency for a consensus protocol with round time Δ_r is $\beta_{\min} \cdot \Delta_r$, achievable only when the network has no adversarial interference and all validators agree immediately.*

Proposition 7.5 (Ray Achieves Light-Speed)**.** *Under zero Byzantine faults, the Ray protocol with $\beta_{\min} = 3$ achieves confirmation in $3 \cdot \Delta_r$ time, matching the light-speed bound.*

Proof. With $\gamma = 0$, the confidence velocity $\nu_c = 1$ (supermajority in every round since all validators agree). The adaptive threshold becomes $\beta^* = \max(3, \lceil \lambda/1 \rceil) = 3$ since $\lambda = 128 \ln 2 / \ln(q_1/q_0)$ and with $\gamma = 0$, $q_1 \approx 1$ and $q_0 \approx 0$, making λ very small (the ratio q_1/q_0 is astronomically large). Thus finalization occurs after 3 rounds. □

7.6 Empirical Evaluation

7.6.1 Setup

We deployed Ray on a 1,000-node testnet across 8 geographic regions with parameters $k = 20$, $\alpha = 0.8$, $\beta_{\min} = 3$, and λ calibrated for 2^{-128} safety.

7.6.2 Latency Results

Under zero Byzantine faults, Ray achieved 200ms median latency (4 rounds at 50ms), a 64% improvement over the static threshold of $\beta = 14$ (700ms). Under 20% Byzantine, the improvement was 23% (500ms vs. 650ms), as the adaptive threshold correctly increased to account for adversarial interference.

Byzantine %	Median (ms)	P99 (ms)	Improvement vs. Static
0%	200	300	64%
10%	350	600	38%
20%	500	900	23%
30%	650	1200	7%

Table 7.1: Ray latency vs. static threshold ($\beta = 14$)

7.6.3 Estimator Accuracy

The adversary estimator $\hat{\gamma}$ converged to within 2% of the true γ after 15 rounds (750ms), with a 95% confidence interval width of $\pm 3\%$.

7.6.4 Throughput

Ray achieved 4,200 TPS on the 1,000-node testnet, comparable to the static-threshold protocol (4,100 TPS). The adaptive computation adds negligible overhead (one division and comparison per round per transaction).

7.7 Related Work

Adaptive consensus protocols have been explored in the leader-based setting. Flexible BFT [17] allows different clients to choose their own safety-liveness trade-off. Ditto [18] adapts block size and frequency based on network conditions. Algorand [19] achieves fast finality through a sortition-based committee selection but uses fixed parameters.

Ray is the first protocol to dynamically adapt finality thresholds in the metastable consensus family, maintaining the leaderless and parameter-free advantages of stochastic sampling while reducing latency in benign conditions.

Chapter 8

Field: Electromagnetic-Inspired Multi-Valued Consensus

8.1 Introduction

The metastable consensus protocols—Slush, Flare, and Wave—operate on binary conflicts: each conflict set has two possible outcomes, and the protocol converges to one. Many blockchain applications, however, involve d-valued conflicts: multiple conflicting transactions spending the same UTXO, multiple candidate blocks at the same height, or multi-party auction outcomes. Extending binary protocols to $d > 2$ requires either running $\log_2 d$ binary instances (inefficient) or redesigning the convergence mechanism.

The Field protocol draws on the physics of electromagnetic alignment to provide a natural multi-valued extension. Instead of binary colors, each validator maintains a preference vector $\mathbf{p}_i \in \mathbb{R}^d$ on the $(d-1)$-simplex. Sampling interactions rotate preference vectors toward the sampled majority, analogous to magnetic dipoles aligning under an external field. The field potential function Φ provides a Lyapunov function for the dynamics, guaranteeing convergence without requiring binary decomposition.

Contributions.

1. The Field protocol: a multi-valued metastable consensus mechanism with electromagnetic-inspired dynamics (§8.3).
2. A field potential function Φ that is non-increasing under the protocol, providing a Lyapunov convergence proof (§8.4).
3. Extension to stake-weighted sampling via field strength gradients (§8.5).
4. Phase transition analysis characterizing the convergence boundary (§8.6).
5. Empirical evaluation of d-valued consensus for $d = 2, 4, 8, 16$ (§8.7).

8.2 Electromagnetic Analogy

8.2.1 Ising and Potts Models

The binary metastable protocols (Slush, Flare, Wave) are naturally modeled by the *Ising model*: n spins on a graph, each taking values ± 1, evolving through local majority dynamics (Glauber dynamics). Below a critical temperature, the system spontaneously magnetizes.

For $d > 2$, the natural generalization is the *Potts model*: each spin takes one of d values, and the energy function penalizes disagreement between neighbors. The Field protocol implements mean-field Potts dynamics through random sampling.

8.2.2 From Spins to Vectors

Rather than discrete Potts spins, we use continuous preference vectors $\mathbf{p}_i \in \Delta^{d-1}$ on the probability simplex, where component $p_i^{(c)}$ represents validator i's degree of preference for value c. The "hard" preference is $c^*(i) = \arg\max_c p_i^{(c)}$.

This continuous representation enables smooth dynamics: instead of discrete flips, preferences rotate gradually toward the sampled majority, providing better convergence properties and natural handling of uncertainty.

Definition 8.1 (Field Potential)**.** *The field potential over the ensemble* $\mathbf{P} = \{\mathbf{p}_1, \ldots, \mathbf{p}_n\}$ *is:*

$$\Phi(\mathbf{P}) = -\frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} \mathbf{p}_i \cdot \mathbf{p}_j = -\frac{1}{n^2} \left\| \sum_{i=1}^{n} \mathbf{p}_i \right\|^2 \tag{8.1}$$

where $\mathbf{p}_i \cdot \mathbf{p}_j = \sum_{c=1}^{d} p_i^{(c)} p_j^{(c)}$ *is the inner product.*

The potential Φ is minimized when all preference vectors are identical (unanimous agreement: $\Phi_{\min} = -1$) and maximized when preferences are uniformly distributed ($\Phi_{\max} = -1/d$ for d equally supported values).

8.3 The Field Protocol

8.3.1 Core Protocol

Algorithm 13 Field Protocol — Multi-Valued Consensus

Require: Validator i, d-valued conflict with initial preference $\mathbf{p}_i$

Require: Parameters: k (sample size), α (alignment threshold), β (acceptance threshold), η (update rate)

count$[c] \leftarrow 0$ for all $c \in \{1, \ldots, d\}$
while $\max_c \text{count}[c] < \beta$ **do**
  $S \leftarrow \text{RandomSample}(V \setminus \{i\}, k)$
  Query each $j \in S$ for $\mathbf{p}_j$
  $\bar{\mathbf{p}} \leftarrow \frac{1}{k} \sum_{j \in S} \mathbf{p}_j$ ▷ sample mean preference
  ▷ Update preference vector toward sample mean
  $\mathbf{p}_i \leftarrow (1 - \eta)\mathbf{p}_i + \eta \bar{\mathbf{p}}$ ▷ field alignment
  $\mathbf{p}_i \leftarrow \mathbf{p}_i / \|\mathbf{p}_i\|_1$ ▷ renormalize to simplex
  $c^* \leftarrow \arg\max_c \bar{p}^{(c)}$ ▷ dominant color in sample
  if $\bar{p}^{(c^*)} \geq \alpha$ **then** ▷ supermajority for c^*
    count$[c^*] \leftarrow$ count$[c^*] + 1$
  end if
  if $\max_c \text{count}[c] \geq \beta$ **then**
    finalize value $\arg\max_c \text{count}[c]$
  end if
end while

8.3.2 Simplified Binary Mode

When $d = 2$, the preference vector reduces to a scalar $p_i = p_i^{(1)}$ (with $p_i^{(0)} = 1 - p_i$), and the protocol reduces to the Wave dynamics with continuous preferences. The update becomes:

$$p_i \leftarrow (1 - \eta)p_i + \eta \bar{p}$$

which converges to $p_i \in \{0, 1\}$ under the same conditions as Wave.

8.4 Convergence Analysis

8.4.1 Potential Decrease

Theorem 8.1 (Monotone Potential Decrease)**.** *Under the Field protocol dynamics with honest supermajority ($f < n/3$), the expected field potential decreases in each round:*

$$\mathbb{E}[\Phi(\mathbf{P}_{r+1})] \leq \Phi(\mathbf{P}_r) - \frac{\eta(2-\eta)}{n} \sum_{i \in H} \|\mathbf{p}_i - \bar{\mathbf{p}}_i\|^2$$

where $\bar{\mathbf{p}}_i$ is the expected sample mean for honest validator i.

Proof. Consider honest validator i updating its preference. The new preference is:

$$\mathbf{p}_i' = (1-\eta)\mathbf{p}_i + \eta \bar{\mathbf{p}}_S$$

where $\bar{\mathbf{p}}_S$ is the sample mean. The contribution of $\mathbf{p}_i$ to the potential changes by:

$$\begin{aligned}\Delta\Phi_i &= -\frac{2}{n^2}\mathbf{p}_i' \cdot \sum_{j \neq i} \mathbf{p}_j + \frac{2}{n^2}\mathbf{p}_i \cdot \sum_{j \neq i} \mathbf{p}_j - \frac{1}{n^2}(\|\mathbf{p}_i'\|^2 - \|\mathbf{p}_i\|^2) \\ &= -\frac{2}{n^2}\eta(\bar{\mathbf{p}}_S - \mathbf{p}_i) \cdot \sum_{j \neq i} \mathbf{p}_j - \frac{1}{n^2}(\|\mathbf{p}_i'\|^2 - \|\mathbf{p}_i\|^2) \qquad (8.2)\end{aligned}$$

Taking expectations over the random sample S: $\mathbb{E}[\bar{\mathbf{p}}_S] = \bar{\mathbf{p}} = \frac{1}{n}\sum_j \mathbf{p}_j$ (the population mean, up to Byzantine distortion bounded by γ).

The cross term in (8.2) contributes:

$$-\frac{2\eta}{n^2}(\bar{\mathbf{p}} - \mathbf{p}_i) \cdot \sum_{j \neq i} \mathbf{p}_j = -\frac{2\eta}{n}(\bar{\mathbf{p}} - \mathbf{p}_i) \cdot \bar{\mathbf{p}} + O(1/n^2)$$

The self-term $\|\mathbf{p}_i'\|^2 - \|\mathbf{p}_i\|^2 = 2\eta(\bar{\mathbf{p}}_S - \mathbf{p}_i) \cdot \mathbf{p}_i + \eta^2\|\bar{\mathbf{p}}_S - \mathbf{p}_i\|^2$.

Combining and summing over all honest validators:

$$\mathbb{E}[\Delta\Phi] \leq -\frac{\eta(2-\eta)}{n} \sum_{i \in H} \|\mathbf{p}_i - \bar{\mathbf{p}}\|^2$$

This is non-positive, with equality only when all honest validators already agree ($\mathbf{p}_i = \bar{\mathbf{p}}$ for all $i \in H$). □

8.4.2 Convergence Rate

Lemma 8.2 (Exponential Convergence)**.** *Define the disagreement $\mathcal{D}(r) = \frac{1}{|H|}\sum_{i \in H} \|\mathbf{p}_i(r) - \bar{\mathbf{p}}(r)\|^2$. Under the Field protocol:*

$$\mathbb{E}[\mathcal{D}(r)] \leq (1 - \eta(2-\eta)/k)^r \cdot \mathcal{D}(0) + O(\gamma^2)$$

where the $O(\gamma^2)$ term accounts for Byzantine interference.

Proof. The potential decrease per round (Theorem 8.1) implies that the disagreement decreases multiplicatively: each honest validator's preference moves toward the sample mean by a factor of η. Since the sample mean is close to the population mean (within $O(1/\sqrt{k})$), the disagreement contracts by factor $(1 - \eta(2-\eta)/k)$ per round.

The Byzantine validators contribute a constant perturbation bounded by γ^2 (since $|B|/n < \gamma$ and each Byzantine vector has unit norm). This creates a floor below which disagreement cannot decrease but does not prevent convergence to a neighborhood of agreement.

The convergence time to $\mathcal{D} = \epsilon$ is:

$$r^* = O\left(\frac{k}{\eta(2-\eta)} \ln \frac{\mathcal{D}(0)}{\epsilon}\right) = O(k \log(1/\epsilon))$$

For $\epsilon = 1/n$ (population-level agreement), this is $O(k \log n)$. □

Corollary 8.3 (Multi-Valued Convergence)**.** *For d-valued conflicts, the convergence rate is the same as for binary conflicts: $O(k \log n)$ rounds, independent of d. This is because the potential function and its decrease rate do not depend on the dimension d.*

8.5 Stake-Weighted Sampling

In the electromagnetic analogy, particles with larger charge create stronger fields and exert more influence on their neighbors. We extend this to stake-weighted consensus.

Definition 8.2 (Field Strength)**.** *Validator i with stake w_i has field strength $\phi_i = w_i/W$ where $W = \sum_j w_j$. The weighted sample mean becomes:*

$$\bar{\mathbf{p}}_S = \frac{\sum_{j \in S} w_j \mathbf{p}_j}{\sum_{j \in S} w_j}$$

Theorem 8.4 (Stake-Weighted Safety)**.** *The Field protocol with stake-weighted sampling maintains safety under the condition $\sum_{i \in B} w_i < W/3$, i.e., the adversary controls less than $1/3$ of total stake. The convergence rate becomes $O(\log(W/w_{\min}))$ where $w_{\min}$ is the minimum honest stake.*

Proof. The proof follows Theorem 8.1 with the replacement of uniform averages by stake-weighted averages. The key observation is that the potential function $\Phi(\mathbf{P}) = -\frac{1}{W^2} \| \sum_i w_i \mathbf{p}_i \|^2$ remains a Lyapunov function under the weighted dynamics. The decrease per round is:

$$\mathbb{E}[\Delta \Phi] \leq -\frac{\eta(2-\eta)}{W} \sum_{i \in H} w_i \|\mathbf{p}_i - \bar{\mathbf{p}}_w\|^2$$

where $\bar{\mathbf{p}}_w$ is the stake-weighted population mean. The convergence time is $O(\frac{W}{w_{\min}} \log(W/\epsilon))$, which reduces to $O(\log n)$ for uniform stake. □

8.6 Phase Transition Analysis

8.6.1 Critical Parameters

Definition 8.3 (Phase Transition)**.** *The Field protocol exhibits a phase transition at the critical parameters (k_c, α_c) below which the protocol fails to converge (ergodic phase) and above which it converges exponentially (ordered phase).*

Theorem 8.5 (Phase Boundary)**.** *For the Field protocol with d values and adversarial fraction $\gamma < 1/3$, the critical sample size is:*

$$k_c = \frac{d \ln d}{(1 - d\gamma)^2}$$

For $k > k_c$, convergence occurs in $O(\log n)$ rounds. For $k < k_c$, the protocol may cycle indefinitely.

Proof. The phase transition occurs when the expected potential decrease per round equals zero: the alignment force from sampling exactly balances the entropic force from randomness.

The alignment force per round scales as $\eta(2-\eta) \cdot \mathcal{D}/k$ (from Lemma 8.2). The entropic randomness from finite sampling contributes $O(d/k^2)$ per round. At the critical point:

$$\frac{\eta(2-\eta)\mathcal{D}}{k_c} = \frac{d}{k_c^2} + \frac{\gamma^2}{k_c}$$

For $\mathcal{D} = O(1)$ (far from consensus) and solving for k_c:

$$k_c \approx \frac{d}{\eta(2-\eta)(1-d\gamma/(d-1))^2} = O\left(\frac{d \ln d}{(1-d\gamma)^2}\right)$$

where the $\ln d$ factor accounts for the entropy of the d-simplex.

For $d = 2$, this gives $k_c \approx 2\ln 2/(1-2\gamma)^2 \approx 6$ for $\gamma = 0.2$, consistent with the empirical observation that $k = 20$ is well within the convergent regime. □

8.6.2 Screening Effect

Lemma 8.6 (Byzantine Screening)**.** *The effective influence of Byzantine validators on the consensus outcome decays exponentially with the "distance" in preference space. Specifically, a Byzantine validator with preference vector orthogonal to the honest majority direction has effective influence $\leq \gamma \cdot e^{-k/d}$ on the alignment rate.*

Proof. When Byzantine validators choose preferences orthogonal to the honest majority, their contribution to the sample mean is projected out by the alignment dynamics. The inner product $\bar{\mathbf{p}}_{\text{Byz}} \cdot \bar{\mathbf{p}}_{\text{honest}}$ is zero, so the Byzantine perturbation only affects the off-axis components of the preference vectors. The convergence along the majority axis proceeds at rate $(1-\gamma)q_1$ where q_1 is the supermajority probability. The off-axis perturbation decays as $\gamma \cdot e^{-k/d}$ per round due to the central limit theorem: the projection of k random Byzantine vectors onto a fixed axis has variance $O(d/k)$, giving influence $\sqrt{d/k} \cdot \gamma$, which is exponentially small in k/d. □

8.7 Empirical Evaluation

8.7.1 Multi-Valued Consensus Performance

We evaluated Field on a 500-node testnet with $d \in \{2, 4, 8, 16\}$-valued conflicts.

d	**Median Rounds**	**Correctness**	**TPS**
2	8	100%	2,750
4	10	100%	2,600
8	12	100%	2,400
16	15	99.99%	2,100

Table 8.1: Field protocol performance vs. conflict cardinality d

The convergence time grows sub-linearly in d, consistent with the $O(\log d)$ factor predicted by Theorem 8.5. All experiments produced the correct outcome (highest initial support wins) with $> 99.99\%$ reliability.

8.7.2 Comparison with Binary Protocol

For $d = 4$, a binary decomposition requires $\log_2 4 = 2$ sequential binary consensus instances, each taking ~ 8 rounds, totaling ~ 16 rounds. The Field protocol achieves the same result in 10 rounds, a 37.5% improvement.

Chapter 9

Composition in the Quasar Stack

The seven engines are not alternatives—they are layers. The Lux primary network composes them into a single finality pipeline that produces *QuasarCerts* (LP-020). This chapter records the composition, not as new mechanism, but as the contract each engine fulfils for the others.

9.1 Layered Picture

Layer	**Role**
Wave	primitive: per-vertex confidence counters, monotone in expectation; supplies the safety theorem that every higher layer reuses
Photon	pipelining of Wave queries; produces sustained per-conflict throughput by overlapping β-rounds across many in-flight transactions
Nova	lifts Photon from individual vertices to DAG cones via the supernova burst: a single frontier finalization closes its entire ancestor cone
Nebula	swaps Wave's i.i.d. sampling for stratified sampling with bridge validators when the network graph is sparse, lossy, or partitioned
Prism	parallelises Photon across B spectrum bands, hashing each conflict set into one band; aggregate throughput scales linearly in B
Ray	feedback layer above Wave/Photon: monitors confidence velocity, estimates adversary fraction, and adapts β to minimise latency at fixed safety
Field	generalisation to $d > 2$ colours through an electromagnetic potential; used for validator-set elections, governance and multi-party auctions where the decision alphabet is not binary

9.2 Quasar Pipeline

A QuasarCert (LP-020) is the artefact of running these engines back to back on the primary network:

1. **Photon** pipelines candidate vertices into the per-conflict sampler.
2. **Wave** drives those samples until each candidate becomes strongly preferred (its DAG cone is internally consistent).
3. **Nova** closes the cone in a supernova burst, finalising every ancestor of the frontier vertex simultaneously.
4. **Ray** watches the empirical confidence velocity and adjusts the global β down toward 3 rounds when the adversary appears benign.

5. **Nebula** replaces the i.i.d. sampler with stratified sampling when the live validator set is below the expansion threshold (rare on mainnet, common on appchains).
6. **Prism** (off by default on mainnet, default on Z-Chain and B-Chain) partitions the mempool into B bands so steps 1–5 run B times in parallel.
7. **Field** runs orthogonally for non-binary decisions such as validator-set epoch elections and DAO votes.

9.3 Interfaces and Contracts

Each engine exposes the same minimal interface:

- **Submit**(t) – accept a candidate (vertex, band, or colour).
- **Vote**(t, c) – broadcast the local preference for t.
- **Sample**(k) – sample k peers and return their votes.
- **Finalize**(t, c) – declare t irreversibly committed to colour c.

This interface is the entire reason composition is possible: the higher layer treats the lower layer as a black box with these four methods, and the lower layer treats its own samples as a black-box oracle. Confidence monotonicity (Wave) is preserved by every wrapper because each wrapper only adds positive evidence; it never subtracts.

9.4 Mainnet Activation

The composition described here is what activated on the Lux primary network on 25 December 2025 as Quasar 3.0 (LP-020). Photon, Wave, Nova, Ray and Field run on every validator. Nebula activates per-subnet when the in-subnet validator count drops below the expansion threshold. Prism is enabled by default on Z-Chain (proof-rollup) and B-Chain (bridge), where independent conflict sets dominate.

Chapter 10

Conclusion

The seven engines collected in this paper form the consensus surface of the Lux primary network. They share a common system model, a common sampling primitive, and a common safety methodology. Read alone, each engine targets one regime: Photon for throughput, Wave for stability, Nova for DAG amortisation, Nebula for sparsity, Prism for parallelism, Ray for adaptive latency, Field for non-binary alphabets. Read together, they compose into the Quasar finality pipeline (LP-020) that activated on 25 December 2025.

The engines were originally published as seven separate short papers in the Lux research catalogue. Those originals now redirect to this consolidated reference; their content is reproduced here verbatim with only label namespacing applied. Future revisions to any engine should land in this paper, with the breadcrumb stubs in the originals updated to track only the chapter location.

Bibliography

[1] Z. Kelling, "LP-020: Quasar 3.0 Consensus—Three-Lane Post-Quantum Certificates," Lux Industries, 2025.

[2] Z. Kelling, "LP-073: Pulsar Threshold Signatures," Lux Industries, 2024.

[3] Z. Kelling, "Wave protocol: Decision stability and confidence monotonicity," Lux Industries, 2020.

[4] Z. Kelling, "Flare protocol: Certificate skip exclusivity with honest support," Lux Industries, 2021.

[5] Z. Kelling, "Field protocol: Electromagnetic-inspired distributed consensus," Lux Industries, 2022.

[6] Z. Kelling, "Nova protocol: Supernova burst consensus for high-throughput chains," Lux Industries, 2023.

[7] Z. Kelling, "Photon protocol: Zero-latency consensus pipelining," Lux Industries, 2023.

[8] T. Rocket et al., "Scalable and probabilistic leaderless BFT consensus through metastability," 2019.

[9] V. Bagaria et al., "Prism: Deconstructing the blockchain to approach physical limits," in *CCS*, 2019.

[10] G. Danezis et al., "Narwhal and Tusk: A DAG-based mempool and efficient BFT consensus," in *EuroSys*, 2022.

[11] D. Malkhi et al., "Bullshark: DAG BFT protocols made practical," in *CCS*, 2022.

[12] M. Castro and B. Liskov, "Practical Byzantine fault tolerance," in *OSDI*, 1999.

[13] M. Yin et al., "HotStuff: BFT consensus with linearity and responsiveness," in *PODC*, 2019.

[14] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008.

[15] C. Dwork, N. Lynch, and L. Stockmeyer, "Consensus in the presence of partial synchrony," *JACM*, 1988.

[16] S. Gilbert and N. Lynch, "Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services," *SIGACT News*, 2002.

[17] D. Malkhi, K. Nayak, and L. Ren, "Flexible Byzantine fault tolerance," in *CCS*, 2019.

[18] R. Gelashvili et al., "Jolteon and Ditto: Network-adaptive efficient consensus with asynchronous fallback," in *FC*, 2022.

[19] Y. Gilad et al., "Algorand: Scaling Byzantine agreements for cryptocurrencies," in *SOSP*, 2017.

[20] M. Penrose, *Random Geometric Graphs*, Oxford University Press, 2003.

[21] E. Ising, "Beitrag zur Theorie des Ferromagnetismus," *Zeitschrift für Physik*, 1925.

[22] R. Potts, "Some generalized order-disorder transformations," *Math. Proc. Cambridge Phil. Soc.*, 1952.

[23] R. Glauber, "Time-dependent statistics of the Ising model," *J. Math. Phys.*, 1963.

[24] L. Landau and E. Lifshitz, *Statistical Physics*, 3rd ed., Pergamon Press, 1980.

[25] K. Binder, "Theory of first-order phase transitions," *Reports on Progress in Physics*, 1987.