Network-Native Computation: Foundations, Structural Leverage, External Validity, and the Limits of Prospective Governance

Driftbound Labs Technical Preprint (v1.0)

Kyle W. Luecke
Driftbound Labs (ORCID: 0009-0008-0202-7852 · admin@driftboundlabs.com)

October 2026

Abstract

Distributed computing traditionally treats computer networks as communication plumbing between arithmetic processors ("Internet as CPU"). This paper formalizes and experimentally investigates an alternative paradigm: **Network-Native Computation (NNC)**, wherein ordinary, authorized, RFC-compliant protocol transitions and persistent, network-maintained structural invariants evaluate computational distinctions through their natural dynamics without deploying custom code into network nodes. Across eight controlled micro-experiments spanning synthetic testbeds, stock production software, decentralized protocol swarms, and runtime governance controllers, we systematically test four foundational questions: (1) *Can network-native dynamics compute?* Yes, protocol transitions evaluate finite automata with 100% agreement over an RFC 6672 `DNAME` / RFC 1034 `CNAME` corpus, but representation alone confers no leverage—without pre-existing state, network boundary costs exceed local CPU cycles by $153\times$–$308\times$ (Exp01). (2) *Can maintained structural state reduce work?* Yes: reconciling sparse mutations against a continuously maintained Merkle invariant decouples work from state cardinality N to a differing frontier $|F|$, achieving $25.2\times$ wall speedup and $3{,}158\times$ wire I/O reduction at $N = 2^{18}$, $k = 1$ (Exp02). When an explicit authenticated mutation log (Δ) exists, Δ wins ($8.9\times$ over NNC); NNC provides the structural derivation $(R_t, R_{t+1}) \mapsto \widehat{\Delta}_t$ when only state roots are attested (Exp03), governed by an empirical switching surface $\Phi(N, k, d)$ in which spatial clustering acts as a $4\times$–$7.8\times$ leverage multiplier (Exp04). (3) *Does this mechanism survive unmodified external substrates?* Yes, across stock Git v2.55.0 pack negotiation ($23.8\times$ wire reduction; $|F| = 19$ clustered vs. 34 scattered objects at $k = 16$, Exp05), multi-peer IPFS/IPLD Bitswap and Kademlia DHT swarms ($13.9\times$–$48.5\times$ wire reduction, Exp06), and an audited architectural emulation of AO/Arweave checkpoint witnesses ($O(k)$ vs. $\Theta(T)$ scaling; $35.4\times$–$98.3\times$ CPU leverage and $1{,}950\times$–$2{,}000\times$ wire reduction at $T = 10{,}000$, Exp07). (4) *Can an autonomous client prospectively determine when to exploit this boundary without hindsight?* Not with bounded shallow probes: across 200 stratified held-out workloads under probe budgets $B \leq 8$, prospective controllers failed pre-specified kill conditions ($\eta_t = 1.803 > 1.25$; catastrophic misroute rate $11.0\% > 5.0\%$, Exp08). We prove the **Bounded-Probe Structural Ambiguity Principle** (Theorem 1 and Lemma 1) and combine it with **Observed Loss Asymmetry** ($L_{\text{FF}}^{\max} = 27.4\times \gg L_{\text{FB}}^{\max} = 3.4\times$) to establish the four-component accounting law $C_{\text{NNC}} = C_{\text{observe}} + C_{\text{decide}} + C_{\text{boundary}} + C_{\text{realize}}$.

1. Introduction: Beyond the Network-as-Transport Model

Modern computing architectures draw a sharp divide between *computation* and *communication*. Computation is presumed to occur within arithmetic logic units, registers, and local memory hierarchies; computer networks are relegated to passive byte-transport pipes. When distributed computation is required, standard engineering deploys remote execution runtimes—remote procedure calls, serverless containers, WebAssembly sandboxes, or programmable switch pipelines (Tennenhouse and Wetherall 1996; Bosshart et al. 2014)—that push imperative instruction streams across network boundaries.

Yet the deployed Internet and its overlay protocols are not inert plumbing. They form a continuously evolving distributed state machine governed by deterministic protocol specifications. Domain name servers evaluate hierarchical delegation graphs (Mockapetris 1987; Rose and Wijngaards 2012); content-addressed storage

swarms maintain cryptographic directed acyclic graphs (Merkle 1987; Benet 2014); version-control transports negotiate commit reachability (Chacon and Straub 2014); and replicated state machines anchor deterministic transition histories (Schneider 1990; Williams and Forward Research 2024).

This paper formalizes and stress-tests **Network-Native Computation (NNC)** around a single falsifiable question:

> **Core Research Question:** *Can ordinary, authorized, RFC-compliant network protocol transitions and persistent substrate invariants be composed into computational structures that evaluate consequential distinctions at lower physical cost than direct local execution—and can an autonomous client prospectively know when to invoke them?*

To prevent conceptual inflation, our investigation progresses through four sequential epistemic hurdles, each paired with explicit falsification gates (Table 1).

Table 1: Four-stage epistemic progression of the Network-Native Computation (NNC) experimental program.

Hurdle	Governing Question	Experimental Specimens & Verdict
I. Representation	Can unmodified protocol transitions represent formal computation?	**Exp01** (DNS `DNAME`/`CNAME` DFA): **Mixed** — Gate A **Supported** (100% agreement), Gate B **Not supported** ($153\times$–$308\times$ slower than local CPU).
II. Leverage	Can continuously maintained invariant state overcome the network boundary toll?	**Exp02** (CAS Merkle frontier), **Exp03** (Reactive DOM vs. adversarial Δ log), **Exp04** (Switching surface $\Phi(N, k, d)$): **Supported** within sparse/clustered regimes; Δ dominates when available.
III. External Validity	Does the frontier-cost structure survive independent substrates not engineered for our hypothesis?	**Exp05** (Stock Git v2.55.0), **Exp06** (Multi-peer IPFS Bitswap + DHT), **Exp07** (AO/Arweave process emulation): **Supported** across all three substrates.
IV. Prospective Control	Can a runtime controller decide prospectively between Frontier and Bulk without hindsight?	**Exp08** (200 stratified workloads, probe budget $B \leq 8$): **Not supported** ($\eta_t = 1.803 > 1.25$, catastrophic rate $11.0\% > 5.0\%$). Formalized via Theorem 1 & Lemma 1.

---

2. Formal Framework & Extended Computational Accounting

2.1 The Substrate Model

Let $x \in \mathcal{X}$ be a problem instance and $F : \mathcal{X} \to \mathcal{Y}$ a target evaluation function. A **network substrate** is a triple:

$$\mathcal{S} = (T, S, G)$$

where:

- T **(Transition Operators)**: Protocol-mandated deterministic or semi-deterministic state evolution operators (e.g., RFC 6672 suffix rewriting, Merkle hash verification, packfile `have`/`want` traversal).
- S **(Persistent State)**: Pre-existing structural state continuously maintained by the substrate (authoritative zones, cryptographic Merkle DAGs, peer routing tables, checkpoint roots).
- G **(Topology & Organization)**: Structural organization of nodes, peering links, latency distributions, and delegation hierarchies.

An NNC execution defines an encoder $E : \mathcal{X} \to \mathcal{M}_{\text{in}}$, a substrate evolution $\mathcal{S}^*$ producing terminal protocol response $m_{\text{out}} \in \mathcal{M}_{\text{out}}$, and a decoder $D : \mathcal{M}_{\text{out}} \to \mathcal{Y}$ satisfying exact functional agreement:

$$D\left(\mathcal{S}^*(E(x))\right) = F(x).$$

2.2 The Extended Four-Component Accounting Law

In informal discussions of incremental synchronization and content-addressed storage, discovering a sparse structural shortcut is frequently treated as zero-cost metadata inspection. Experiment 08 demonstrates that prospective structural discovery under uncertainty is itself an active physical process with non-negligible decision risk. We therefore define the complete physical cost of an NNC transaction as:

$$\boxed{C_{\text{NNC}} = C_{\text{observe}} + C_{\text{decide}} + C_{\text{boundary}} + C_{\text{realize}}}$$

comprising four distinct terms:

1. $C_{\textbf{observe}}$: Computational and network work expended gathering pre-reconciliation metadata and bounded structural probes (e.g., root comparisons, shallow child-hash queries, channel telemetry).
2. $C_{\textbf{decide}}$: Local CPU time consumed by the operational routing policy $\pi(x)$ to evaluate candidate cost models.
3. $C_{\textbf{boundary}}$: Physical round-trip latency, kernel socket transitions, serialization, and protocol framing overhead.
4. $C_{\textbf{realize}}$: Physical work required to traverse and retrieve the chosen state representation (fetching frontier nodes, replaying interval messages, or streaming a bulk snapshot).

2.3 Three Non-Equivalent Leverage Tiers

Let $C_{\text{ref}}(x)$ denote the cost of direct local execution. Throughout all eight experiments, we enforce a strict separation between three non-equivalent propositions:

$$\boxed{\text{Representable} \;\not\Longrightarrow\; \text{Requester-Cheaper} \;\not\Longrightarrow\; \text{Globally-Cheaper}}$$

- **Tier 1 (Semantic Equivalence)**: Exact agreement $D(\mathcal{S}^*(E(x))) = F(x)$ across all evaluated instances.
- **Tier 2 (Requester-Compute Leverage)**: $C_{\text{requester}}(x) < C_{\text{ref}}(x)$, where the local client expends fewer CPU cycles or less wall-clock time.
- **Tier 3 (Aggregate-Compute Leverage)**: $C_{\text{total}}(x) < C_{\text{ref}}(x)$, where total physical work across client, network transit, and amortized substrate state is strictly reduced.

---

3. Question I: Can Network-Native Dynamics Compute? (Exp01)

3.1 Design: RFC 6672 `DNAME` Rewriting as a Finite Automaton

Exp01 (`CONTROLLED_SYNTHETIC`) tested whether standard Domain Name System resolution (Mockapetris 1987; Rose and Wijngaards 2012) can evaluate regular languages without modifying resolver logic.

- **Target Language**: Recognition of $L = (ab)^*a$ over alphabet $\Sigma = \{a, b\}$.
- **Encoding** (E, T): A 5-state deterministic finite automaton (DFA) was compiled into static DNS zone records. An input word $w = \sigma_1\sigma_2 \ldots \sigma_m \in \Sigma^*$ is encoded in reverse label order so suffix rewriting consumes symbols left-to-right:

$$E(w) = \sigma_m.\sigma_{m-1} \ldots \sigma_1.q_0\texttt{.automaton.internal.}$$

- **Protocol Dynamics**: Under RFC 6672 (Rose and Wijngaards 2012), each `DNAME` rule rewrites suffix $\sigma_i.q_s$ to next state $q_{s'}$, stripping one symbol per resolution step. Under RFC 1034 (Mockapetris 1987), terminal state queries q_{final}`.automaton.internal.` resolve either to a synthetic `A`

record (`127.0.0.1`, decoded as `ACCEPT`) or to `NXDOMAIN` (`REJECT`). For example, on input $w =$ `ab`, `b.a.q0.automaton.internal` rewrites via `a.q0 -> q1` to `b.q1.automaton.internal`, then via `b.q1 -> q0` to `q0.automaton.internal`, terminating cleanly via RFC 1034 lookup.

3.2 Results & Diagnostic Failure

- **Gate A (Semantic Embedding — PASSED, 100% Agreement)**: Across a pre-specified corpus of 71 string instances spanning lengths $m \in [0, 10]$, standard DNS resolution reproduced the reference DFA transition table with zero errors.
- **Gate B (Computational Leverage — FAILED)**: Direct local DFA execution required $C_{\text{ref}} \approx 0.4$–$1.8$ μs, whereas DNS evaluation required $C_{\text{NNC}} \approx 123$–$329$ μs—a requester slowdown of $153.6\times$–$268.6\times$ in the frozen verification ledger (up to $308\times$ across initial microbenchmarks).

3.3 Core Negative Law: Invariant Exploitation vs. Sequential Emulation

Exp01 establishes our first boundary condition:

$$\boxed{\text{Representable} \not\Longrightarrow \text{Cheaper}}$$

Coercing stateless network transitions to mimic sequential ALU clock steps is physically uncompetitive because packet framing, name-compression parsing, and socket boundaries operate on microsecond-to-millisecond timescales whereas register transitions execute in sub-nanoseconds.

> **Principle 1 (Invariant Exploitation Principle).** *Do not force network transitions to emulate sequential CPU instructions from scratch. Computational leverage arises only when a problem is coupled to an expensive structural invariant already continuously maintained by the substrate.*

---

4. Question II: Can Maintained Structure Reduce Work? (Exp02–Exp04)

4.1 Exp02: Merkle Frontier Discovery & The Physical Balance of Forces

Exp02 (`CONTROLLED_SYNTHETIC`) evaluated state-difference discovery over a Content-Addressed Storage (CAS) Merkle hierarchy (Merkle 1987; Tamassia 2003).

- **Workload**: Two state trees G_1, G_2 of leaf cardinality $N = 2^{18} = 262{,}144$ (16.78 MB raw leaf payload) differing at $k = 1$ mutated leaf.
- **Reference Baseline**: Full-state flat scan required $C_{\text{full}} = 73.011$ ms and 16.78 MB of local I/O.
- **NNC Frontier Negotiation**: Descending only unequal cryptographic branches required $C_{\text{CAS}} = 2.896$ ms and 5.31 KB of wire transfer.

Decomposing Exp02 against an in-memory local Merkle oracle ($C_{\text{localMerkle}} = 0.0645$ ms) isolates two opposing physical ratios:

1. **Structural Representation Advantage**: $C_{\text{full}}/C_{\text{localMerkle}} = 73.011/0.0645 \approx \mathbf{1{,}132}\times$.
2. **Network Boundary Penalty**: $C_{\text{CAS}}/C_{\text{localMerkle}} = 2.896/0.0645 \approx \mathbf{44.9}\times$.

Because the structural pruning ratio exceeds the socket/serialization toll by $1{,}132/44.9 = \mathbf{25.2}\times$, maintained Merkle structure yields a net $25.2\times$ **wall-time speedup** and $3{,}158\times$ **wire I/O reduction** (`16,777,216 B / 5,313 B`).

4.2 Exp03: Reactive State & The Adversarial Delta Control (Δ Wins)

Exp03 (`CONTROLLED_SYNTHETIC`) embedded Merkle frontier reconciliation into a reactive browser DOM state benchmark ($N = 262{,}144$ nodes, 8,192 KB serialized state), comparing four synchronization strategies across $k \in [0, 64]$ mutations. At $k = 1$:

1. **Strategy 1 (Full DOM Rebuild)**: 377.3 ms wall time, 8,192 KB wire.

2. **Strategy 2 (Local Virtual-DOM Diff)**: 103.34 ms CPU, 167.09 ms wall time, $8{,}192$ KB wire.
3. **Strategy 3 (NNC Frontier Reconciliation)**: 0.180 ms client CPU, 2.59 ms wall time, 2.44 KB wire ($573.2\times$ **client CPU reduction** and $64.6\times$ **wall speedup** over Virtual-DOM; $750.2\times$ **wall speedup** at $k = 0$ identity).
4. **Strategy 4 (Adversarial Control — Explicit Mutation Log Δ_t)**: 0.005 ms CPU, 0.29 ms wall time, 0.03 KB wire ($8.9\times$ **faster than NNC**).

Principle 2 (The Reconciliation Hierarchy). *Whenever an explicit, authenticated mutation log Δ_t is continuously maintained and available to the receiver, direct log shipping (Δ_t) strictly dominates structural frontier traversal ($8.9\times$–$30\times$ faster across Exp03). NNC does not replace operation logs; rather, when only state commitments (R_t, R_{t+1}) are attested, NNC derives an effective structural diff $\widehat{\Delta}_t$ in $O(|F|)$ work rather than $\Theta(N)$.*

4.3 Exp04: The Empirical Switching Surface $\Phi(N, k, d)$

Exp04 (`CONTROLLED_SYNTHETIC`) mapped the operational crossover boundary between Frontier NNC and Bulk Sync across leaf cardinalities $N \in \{4096, 16384, 65536\}$, mutation counts $k \in [1, 2048]$, and spatial geometries $d \in \{\texttt{clustered}, \texttt{scattered}\}$ (Table 2).

Table 2: Empirical crossover density $\rho^* = k^*/N$ above which Bulk Sync overtakes Frontier NNC (Exp04).

Cardinality N	Scattered Crossover $\rho^* = k^*/N$	Clustered Crossover $\rho^* = k^*/N$	Spatial Clustering Multiplier
4,096	0.20% ($k^* = 8$)	1.56% ($k^* = 64$)	$7.8\times$
16,384	0.20% ($k^* = 32$)	1.56% ($k^* = 256$)	$7.8\times$
65,536	0.39% ($k^* = 256$)	1.56% ($k^* = 1{,}024$)	$4.0\times$

Holding $N = 65{,}536$ and $k = 1{,}024$ fixed while changing *only* spatial mutation geometry flips the faster execution regime:

- **Scattered ($k = 1{,}024$)**: Frontier $= 131.44$ ms vs. Bulk $=$ **45.45** ms $\implies$ **Bulk Sync Wins**.
- **Clustered ($k = 1{,}024$)**: Frontier $=$ **39.23** ms vs. Bulk $= 46.85$ ms $\implies$ **Frontier NNC Wins**.

Thus the viable operational envelope is governed not by raw scalar density k/N, but by the **latent structural cardinality of the differing frontier** $|F|$:

$$C_{\text{frontier}} \approx C_{\text{boundary}} + |F(X)| \cdot C_{\text{node}}.$$

---

5. Question III: Does NNC Survive Independent Substrates? (Exp05–Exp07)

Phase 2 tested external validity across three systems whose protocols were designed independently of our framework.

5.1 Exp05: Stock Git Commit-Graph Negotiation (`REAL_SOFTWARE_LOCAL`)

Exp05 executed unmodified production `git` binaries (v2.55.0) performing `git-upload-pack` and `git fetch` over repositories of $N = 1{,}000$ files organized across 20 subdirectories (Chacon and Straub 2014).

- **Sparse Baseline ($k = 1$)**: Native `have`/`want` negotiation achieved a $23.85\times$ **wire reduction** (`65,268 B` vs. `1,556,480 B` snapshot) and $2.02\times$–$2.35\times$ wall speedup.
- **Controlled Geometry Pair ($N = 1{,}000, k = 16$)**:
  - **Clustered ($k = 16$ within 1 directory)**: Git invalidated 1 commit, 1 root tree, 1 subdirectory tree, and 16 blobs—an exact frontier of $|F| = 19$ **objects** (`3,355 B` unpacked loose-object delta; `66,563 B` total pack wire; 455.3–491.5 ms fetch wall time across runs, $23.38\times$ wire leverage).

- **Scattered ($k = 16$ across 16 directories)**: Git invalidated 1 commit, 1 root tree, 16 subdirectory trees, and 16 blobs—an exact frontier of $|F| = 34$ **objects** (`22,654 B` loose-object delta, $6.75\times$ larger; `85,864 B` total pack wire; 560.7–801.5 ms fetch wall time, $18.13\times$ wire leverage).

Stock Git independently manifests the exact geometry-dependent frontier pruning identified in Exp04 without a single line of custom protocol code.

5.2 Exp06: Decentralized IPFS / IPLD Bitswap & DHT Swarms (`DISTRIBUTED_TESTBED`)

Exp06 crossed content geometry $d_{\text{content}} \in \{\texttt{clustered}, \texttt{scattered}\}$ with peer-swarm availability $d_{\text{avail}} \in \{\texttt{local}, \texttt{single_provider}, \texttt{distributed_providers}\}$ across a 6-node socket testbed implementing IPLD Merkle DAGs, Bitswap 1.2 block exchange, and Kademlia XOR routing (Benet 2014; Maymounkov and Mazières 2002).

In decentralized overlays, wall-clock realization cost depends not on $|F|$ alone, but on topological block acquisition over G_{network}:

$$\boxed{C_F = \sum_{v \in F} C_{\text{acquire}}(v, G_{\text{network}})}$$

Across all 6 cells of the 2×3 matrix—including scattered mutations retrieved via multi-hop DHT provider lookups—Frontier NNC preserved a $13.9\times$–$48.5\times$ **finite wire reduction** and $1.79\times$–$58.6\times$ **wall-time speedup** over full DAG reconstruction.

5.3 Exp07: AO / Arweave Process Dynamics (`ARCHITECTURAL_EMULATION`)

Exp07 tested whether NNC extends from static content DAGs to stateful message-driven execution (Williams et al. 2019; Williams and Forward Research 2024).

> **Epistemic Provenance Boundary:** Unlike Exp05 (stock `git` binary) and Exp06 (multi-socket peer swarm), Exp07 evaluated a deterministic in-memory architectural process emulator (`process_model.py`, $M = 1{,}000$ state slots) implementing AO transition semantics and Merkleized checkpoint commitments. It establishes algorithmic scaling laws rather than live public-testnet latency.

Comparing full genesis replay ($\Theta(T)$ CPU and wire), full state snapshots ($\Theta(M)$ wire), and the native checkpoint-interval transition witness $\mathcal{W} = (R_T, [m_{T+1}, \dots, m_{T+k}], R_{T+k})$ across history lengths $T \in \{500, 2000, 10000\}$ and intervals $k \in \{1, 4, 16, 64\}$ confirmed exact history decoupling: $\mathcal{W}$ scales as $O(k)$ **CPU and** $O(k)$ **wire**, invariant to T. At $T = 10{,}000, k = 1$, $\mathcal{W}$ transferred `80 B` vs. `160,016 B` for replay ($2{,}000.2\times$ **wire reduction**; $150.0\times$ vs. `12,000 B` state snapshot) and achieved $35.4\times$–$98.3\times$ **CPU leverage** across benchmark runs (`8.37 ms` vs. `296.01 ms` in the frozen verification ledger; `2.78 ms` vs. `273.63 ms` in the primary closure run).

---

6. Question IV: The Limits of Prospective Governance (Exp08)

Experiments 01–07 establish that maintained network invariants create cheap realization frontiers *when the workload lies inside the favorable region of* $\Phi(N, k, d)$. Experiment 08 (`CONTROLLED_SYNTHETIC`) asks the operational governance question:

> **Prospective Governance Question:** *Given only pre-reconciliation roots (R_t, R_{t+1}), channel telemetry, and a strict budget of $B \leq 8$ structural node probes—with zero hindsight into k, geometry d, or $|F|$—can an autonomous runtime controller reliably select between Frontier NNC and Bulk Transfer?*

6.1 Pre-Specified Zero-Hindsight Protocol & Kill-Condition Audit

Before evaluation, kill conditions were pre-specified in `PREREGISTRATION.md` across a stratified held-out suite of **200 non-identity workloads** spanning three ground-truth cost-ratio bands ($r = C_F/C_B < 0.67$,

$0.67 \leq r \leq 1.5$, and $r > 1.5$) up to $N = 65{,}536$. Each controller had access to: (1) Gate 0 root-equality check ($R_t == R_{t+1}$, $O(1)$), (2) tree metadata $(N, B_{\text{leaf}}, H = \log_2 N)$, (3) channel parameters $(\bar{C}_{\text{acquire}}, \overline{\text{BW}}, C_{\text{boundary}})$, and (4) an audited probe oracle raising `BudgetExceededError` if node-hash comparisons exceeded B.

Against the pre-specified thresholds, prospective governance was **not supported** (Table 3):

- **Gate 0 Identity Check**: **PASSED** (100% accuracy, mean latency 0.80 μs).
- **Controller Decision Overhead ($\rho_{\text{overhead}} \leq 2.0\%$)**: **PASSED** (0.0035%, $\approx$ 3–8 μs per decision).
- **Realized Time Efficiency ($\eta_t = \sum C_{\text{realized}} / \sum C_{\text{oracle}} \leq 1.25$)**: **FAILED** ($\eta_t = \mathbf{1.803}$ for Adaptive Controller B; $\eta_t = \mathbf{1.832}$ for Shallow Controller A).
- **Catastrophic Misroute Rate (frac$(C_{\text{realized}} > 2C_{\text{oracle}}) \leq 5.0\%$)**: **FAILED** (**11.0**% for Controller B; **10.5**% for Controller A).

Table 3: Pre-specified evaluation of six routing policies across 200 stratified non-identity workloads (Expo8).

Routing Policy	Selection Accuracy	Cumulative Realized Time	Efficiency Ratio η_t	Catastrophic Rate	p95 Regret
Always Bulk	50.0%	52,565.5 ms	1.361	15.5%	483.1 ms
Always Frontier	50.0%	166,667.0 ms	4.315	21.0%	3,061.2 ms
Fixed Threshold	53.5%	62,187.8 ms	1.610	20.5%	740.7 ms
Controller A (Shallow, $B = 2$)	62.5%	70,775.9 ms	1.832	10.5%	483.1 ms
Controller B (Adaptive, $B = 8$)	60.0%	69,642.1 ms	1.803	11.0%	472.6 ms
Time-Optimal Oracle (O_t)	100.0%	38,628.2 ms	1.000	0.0%	0.0 ms

6.2 The Bounded-Probe Structural Ambiguity Principle

The failure of Controllers A and B is not an artifact of sub-optimal threshold tuning (audited in `test_tuning.py`); it is an intrinsic consequence of observing hierarchical cryptographic commitments under sublinear query budgets (Yao 1977; Buhrman and Wolf 2002).

6.2.1 Transcript Equivalence Classes & Decision Insufficiency Let $\mathcal{T}$ be a complete binary Merkle tree over $N = 2^H$ leaves indexed by levels $\ell \in \{0, \dots, H\}$ (leaves at $\ell = 0$, root R at $\ell = H$). Under collision-resistant hashing, node digests satisfy $h_S(u) = h_{S'}(u) \iff \text{Leaves}(\mathcal{T}_u, S) = \text{Leaves}(\mathcal{T}_u, S')$.

1. **Adaptive Query Model**: An inspection algorithm $\mathcal{A}$ with budget B adaptively selects $b \leq B$ tree nodes $q_1, \dots, q_b \in \mathcal{V}(\mathcal{T})$ and observes binary equality indicators $\omega(q_i) = \mathbf{1}[h_S(q_i) \neq h_{S'}(q_i)]$.
2. **Observation Transcript & Equivalence**: On transition $X = (S, S')$, $\mathcal{A}$ records transcript $P_B(X) = ((q_1, \omega(q_1)), \dots, (q_b, \omega(q_b)))$, inducing the equivalence relation:

$$X \sim_B Y \iff P_B(X) = P_B(Y), \qquad [X]_B = \{Y : P_B(Y) = P_B(X)\}.$$

3. **Cost-Ratio Spread**: Let $C_F(X) = c_{\text{boundary}} + c_{\text{node}}|F(X)|$ and $C_B(X) = c_{\text{boundary}} + c_{\text{bulk}}N$ with $0 < c_{\text{bulk}} < c_{\text{node}}$, where $|F(X)|$ counts the differing nodes (or differing terminal leaves) traversed during frontier reconciliation. Define $r(X) = C_F(X)/C_B(X)$ and class ratio set $\mathcal{R}([X]_B) = \{r(Y) : Y \in [X]_B\}$.

   Theorem 1 (Impossibility of Deterministic Governance under Non-Degenerate Spread). *If an adaptive probe policy $\mathcal{A}$ of budget B admits an observational equivalence class*

$[X^*]_B$ *satisfying*

$$\inf \mathcal{R}([X^*]_B) < 1 < \sup \mathcal{R}([X^*]_B),$$

then $P_B(X^*)$ *is not decision-sufficient for* $\operatorname{sign}(C_F - C_B)$*. Every deterministic routing policy* $\pi : P_B \to \{\texttt{Frontier}, \texttt{Bulk}\}$ *is constant on* $[X^*]_B$ *and therefore necessarily misroutes at least one realization* $Y \in [X^*]_B$*.*

6.2.2 The Merkle Ambiguity Lemma

Lemma 1 (Merkle Ambiguity Lemma). *Let* $\mathcal{T}$ *be a complete binary Merkle tree with* $N = 2^H \geq 64$ *leaves, let* $\mathcal{A}$ *be any deterministic adaptive inspection strategy with query budget* $B \geq 1$ *satisfying* $8B^2(\log_2 N + 1) < N$*, and let cost parameters satisfy*

$$2c_{\text{node}}B(\log_2 N + 1) < c_{\text{bulk}}N < c_{\text{node}}\frac{N}{4B}.$$

Then there exists a valid observation transcript P^* *whose equivalence class* $[X^*]_B$ *contains transitions* $X_1, X_2 \in [X^*]_B$ *satisfying* $r(X_1) < 1 < r(X_2)$*.*

Proof. Set $K = 2^{\lceil \log_2(B+1) \rceil}$. Since $B \geq 1$ is an integer, $B + 1 \leq K \leq 2B$. Because $8B^2(\log_2 N + 1) < N$ implies $8B < N$, the power of two $K \leq 2B \leq N/4$ divides $N = 2^H$, partitioning $\mathcal{T}$ into K disjoint dyadic subtrees $\mathcal{T}_1, \dots, \mathcal{T}_K$, each containing $m = N/K \geq \frac{N}{2B} \geq 4$ leaves. Let $\mathcal{V}^\circ(\mathcal{T}_j) = \mathcal{V}(\mathcal{T}_j) \setminus \{\text{root}(\mathcal{T}_j)\}$ denote the strict interior and leaf nodes of $\mathcal{T}_j$ (levels 0 through $\log_2 m - 1$).

Fix a base state S and define transition $X_1 = (S, S_1')$ *prior to running* $\mathcal{A}$ by mutating solely the first leaf $v_{j,0}$ of each subtree $\mathcal{T}_j$ ($j = 1, \dots, K$). Execute deterministic $\mathcal{A}$ on X_1, yielding transcript $P^* = P_B(X_1)$ over $b \leq B$ queries $q_1, \dots, q_b$. Because the $K \geq B + 1$ node sets $\mathcal{V}^\circ(\mathcal{T}_1), \dots, \mathcal{V}^\circ(\mathcal{T}_K)$ are pairwise disjoint and $b \leq B < K$, by the pigeonhole principle at least one subtree $\mathcal{T}_u$ satisfies $\mathcal{V}^\circ(\mathcal{T}_u) \cap \{q_1, \dots, q_b\} = \emptyset$.

Now construct $X_2 = (S, S_2')$ identically to X_1 outside $\mathcal{T}_u$, while inside $\mathcal{T}_u$ mutating $m/2$ alternating leaves (including $v_{u,0}$). Under collision-free hashing, every node at or above $\{\text{root}(\mathcal{T}_1), \dots, \text{root}(\mathcal{T}_K)\}$ differs from S under *both* X_1 and X_2 (since every $\mathcal{T}_j$ contains at least one mutated leaf in both transitions), while outside $\mathcal{T}_u$ the states S_1' and S_2' coincide leaf-for-leaf. Because $\mathcal{A}$ never queried $\mathcal{V}^\circ(\mathcal{T}_u)$, induction on the query index $i = 1, \dots, b$ gives $\omega_{X_2}(q_i) = \omega_{X_1}(q_i)$ at every step; hence $P_B(X_2) = P_B(X_1) = P^*$ and $X_1, X_2 \in [X_1]_B$.

Finally, under either differing-node count or differing-leaf count, X_1 satisfies $|F(X_1)| \leq (2K - 1) + K \log_2 m \leq 2B(\log_2 N + 1)$, whereas X_2 branches across $\mathcal{T}_u$ to give $|F(X_2)| \geq m/2 \geq \frac{N}{4B}$. Applying the cost-parameter inequality yields $C_F(X_1) < C_B(X_1)$ ($r(X_1) < 1$) and $C_F(X_2) > C_B(X_2)$ ($r(X_2) > 1$). ■

By Theorem 1, every deterministic controller π operating on P_B satisfies $\pi(X_1) = \pi(X_2)$ and necessarily misroutes at least one realization in $[X_1]_B$.

6.3 Observed Loss Asymmetry Explains Why "Always Bulk" Wins

Compounding bounded-probe structural ambiguity is a severe physical asymmetry in misroute penalties across the 200-workload benchmark:

$$\boxed{L_{\text{FB}}^{\max} = 3.4\times \quad \ll \quad L_{\text{FF}}^{\max} = 27.4\times \quad \text{(Expo8 corpus)}}$$

- **False-Bulk Loss ($L_{\text{FB}} = C_{\textbf{bulk}}/C_F$)**: Mistakenly choosing Bulk when Frontier was faster incurred at most a $3.41\times$ slowdown in Band 1 ($r < 0.67$), because contiguous streaming over modern memory and socket buffers has a tightly bounded ceiling ($c_{\text{bulk}}N$).
- **False-Frontier Loss ($L_{\text{FF}} = C_F/C_{\textbf{bulk}}$)**: Mistakenly choosing Frontier on a dense scattered workload incurred up to a $27.38\times$ slowdown in Band 3 ($r > 1.5$). In Instance 43 ($N = 65{,}536$, $k = 30{,}157$), Bulk completed in 1.14 s whereas Frontier traversal stalled for 31.22 s—a single misroute penalty of 30.08 **s** (77.9% of the entire 200-workload Oracle runtime of 38.63 s).

Under symmetric cost estimation ($\widehat{C}_F < \widehat{C}_B$), occasional False-Frontier tail losses overwhelm frequent small Frontier savings, explaining why **Always Bulk ($\eta_t = 1.361$)** outperformed both uncalibrated prospective controllers ($\eta_t = 1.832$ and 1.803).

---

7. Synthesis: The Eight-Experiment Architectural Landscape

Table 4 consolidates the empirical trajectory, provenance classes, and governing laws across all eight experiments.

Table 4: Consolidated empirical ledger and provenance taxonomy across Experiments 01–08.

Exp	Provenance Class	Substrate Specimen	Headline Empirical Result	Governing Law / Principle
01	Controlled Synthetic	DNS Zone (`DNAME`/`CNAME`)	100% DFA agreement; $153.6\times$–$268.6\times$ slower than CPU	Representable $\not\Rightarrow$ Cheaper
02	Controlled Synthetic	CAS Merkle Tree ($N = 2^{18}$)	$25.2\times$ wall speedup; $3{,}158\times$ wire I/O reduction ($k = 1$)	Invariant Exploitation Principle
03	Controlled Synthetic	Reactive Browser DOM	$573.2\times$ CPU cut, $64.6\times$ wall speedup; Δ beats NNC by $8.9\times$	Reconciliation Hierarchy ($\Delta > \widehat{\Delta} >$ Bulk)
04	Controlled Synthetic	Merkle Surface Sweep	Clustering shifts ρ^* by $4.0\times$–$7.8\times$; flips winner at $k = 1024$	Switching Surface $\Phi(N, k, d)$
05	Real Software Local	Stock Git v2.55.0	$23.8\times$ wire cut ($k = 1$); $\lvert F\rvert = 19$ clustered vs. 34 scattered ($k = 16$)	External Validity of Geometry Pruning
06	Distributed Testbed	IPFS Bitswap 1.2 + DHT	$13.9\times$–$48.5\times$ wire cut; $1.79\times$–$58.6\times$ wall speedup	$C_F = \sum_{v \in F} C_{\text{acquire}}(v, G_{\text{net}})$
07	Architectural Emulation	AO / Arweave Process Model	$O(k)$ witness; $2{,}000\times$ wire cut vs. replay, $35.4\times$–$98.3\times$ CPU	History Decoupling ($T \mapsto k$)
08	Controlled Synthetic	Prospective Controller ($B \leq 8$)	Kill conditions **Not supported** ($\eta_t = 1.803$, catastrophic $= 11.0\%$)	Structural Ambiguity & Loss Asymmetry

---

8. Related Work & Adversarial Positioning

Rather than listing prior systems chronologically, we position NNC against five established research families using two physical questions: *(1) What structural invariant does the substrate already continuously maintain?* and *(2) What computation or code must the requester introduce?* (Table 5).

Table 5: Adversarial positioning of Network-Native Computation across five prior-art families.

Comparison Family	Maintained State (S)	Requester Deploys Code?	Scaling Quantity	Prospective Routing Modeled?	Relationship to NNC
1. Active / In-Network Compute (Active Nets, P4, NetCache)	Switch pipelines, SRAM match-action tables	**Yes** (custom P4/bytecode)	Line-rate packet depth	No (static deployment)	**Inverse Foil**: NNC deploys *zero* custom code to nodes; exploits stock RFC transitions.
2. Content-Addressed DAGs (Merkle, Git, IPFS/IPLD)	Cryptographic DAGs, DHT provider records	**No** (stock hash fetch)	Frontier objects $\lvert F\rvert$	No (unconditional sync)	**Direct Ancestry**: NNC models boundary tolls, $\Phi(N, k, d)$, and prospective routing limits.
3. Set Reconciliation & IVM (rsync, IBLT, Diff. Dataflow, Cassandra)	Checksums, SSTable Merkle trees, dataflow deltas	**Varies**	Set diff d or log Δ	Partial (IBLT strata; manual `rsync -W`)	**Mechanistic Baseline**: Explicit Δ wins (Exp03); NNC isolates root-only $(R_t, R_{t+1}) \mapsto \widehat{\Delta}_t$.
4. Query Complexity (Yao minimax, decision trees)	Abstract query oracle	**Yes** (adaptive queries)	Query budget B	**Yes** (lower bounds)	**Theoretical Skeleton**: Theorem 1 & Lemma 1 specialize indistinguishability to Merkle cost crossover.
5. Replicated State Machines (SMR, Raft, Arweave, AO)	Append-only logs, checkpoint roots	**No** (deterministic step)	History T or state M	No (fixed checkpoints)	**Stateful Extension**: Audited emulation (Exp07) confirms $O(k)$ witness decoupling.

8.1 Analysis of Prior-Art Families

1. **In-Network & Programmable Data Planes**: Active Networks (Tennenhouse and Wetherall 1996; Alexander et al. 1998) and modern programmable switch fabrics including P4 (Bosshart et al. 2014), NetCache (Jin et al. 2017), and NetPaxos (Dang et al. 2015) achieve throughput gains by installing operator-authored packet programs into switch ASICs or SmartNICs. NNC investigates the strict complement: *zero* code deployment into infrastructure, relying exclusively on unmodified, authorized protocol behaviors (`DNAME`/`CNAME`, `git-upload-pack`, Bitswap). Exp01 shows why naive instruction emulation over stock protocols fails ($> 150\times$ slowdown), isolating pre-existing invariant exploitation as essential.
2. **Content Addressing & Authenticated Data Structures**: Cryptographic hash trees (Merkle 1987), authenticated dictionaries (Goodrich et al. 2001; Tamassia 2003), Git object graphs (Chacon and Straub 2014), and IPFS/IPLD over Kademlia (Benet 2014; Maymounkov and Mazières 2002) form the direct intellectual ancestry of NNC. NNC does not claim to invent Merkle DAGs; rather, it measures the physical

balance between representation pruning ($\sim 1{,}132\times$) and network boundary tolls ($\sim 44.9\times$), characterizes the geometry-dependent switching surface $\Phi(N, k, d)$, and generalizes topological realization cost in peer swarms.

3. **Set Reconciliation & Incremental Computation**: Algebraic set reconciliation (Minsky et al. 2003), Invertible Bloom Lookup Tables (Goodrich and Mitzenmacher 2011; Eppstein et al. 2011), `rsync` (Tridgell and Mackerras 1996), LBFS (Muthitacharoen et al. 2001), adaptive/differential computation (Acar et al. 2002; McSherry et al. 2013), and anti-entropy repair (Demers et al. 1987; Lakshman and Malik 2010) optimize delta transmission. Exp03 explicitly incorporates an adversarial Δ-log baseline to show that maintained mutation logs dominate NNC whenever available. However, systems such as `rsync` rely on manual whole-file flags (`-W`) and Cassandra unconditionally traverses fixed-depth repair trees; none formalize the prospective decision boundary between hierarchical traversal and bulk streaming under bounded observation.
4. **Decision-Tree Query Complexity**: Indistinguishability and adversarial lower bounds are classical in query complexity and property testing (Yao 1977; Buhrman and Wolf 2002; Goldreich et al. 1998). Theorem 1 and Lemma 1 instantiate this framework specifically over cryptographic Merkle hierarchies, proving that sublinear query budgets leave untouched subtrees whose internal mutation geometry flips $\text{sign}(C_F - C_B)$, and linking this ambiguity to empirical tail-loss asymmetry ($27.4\times$ vs. $3.4\times$).
5. **State Machine Replication & Checkpoint Witnesses**: Classical state machine replication (Schneider 1990; Ongaro and Ousterhout 2014) and permanent message ledgers (Williams et al. 2019; Williams and Forward Research 2024) combine ordered logs with periodic checkpoints. Exp07 demonstrates within an audited architectural model that coupling attested checkpoint pairs (R_T, R_{T+k}) to raw message intervals yields a self-verifying $O(k)$ witness without trusting an external delta index.

8.2 Surviving Scientific Contribution

Stripping away any claim to reinventing Merkle structures or algebraic diffing leaves four sharp contributions:

1. **Boundary Separation**: Demonstrating experimentally that protocol representability is decoupled from computational leverage (Exp01 vs. Exp02) and that explicit mutation logs Δ strictly dominate root-based frontier derivation when present (Exp03).
2. **Geometry-Governed Switching Surface**: Mapping $\Phi(N, k, d)$ and verifying on unmodified Git v2.55.0 and multi-peer IPFS swarms that spatial clustering multiplies effective frontier leverage by $4\times$–$7.8\times$ (Exp04–Exp06).
3. **Prospective Governance Negative Result**: Falsifying low-budget ($B \leq 8$) zero-hindsight runtime controllers via preregistered kill conditions ($\eta_t = 1.803$, 11.0% catastrophic misroutes, Exp08).
4. **Coupled Observation–Realization Accounting**: Proving Bounded-Probe Structural Ambiguity (Theorem 1 / Lemma 1) and establishing that physical network computation is governed by the coupled pair $(C_{\text{realize}}, C_{\text{observe}} + C_{\text{decide}})$.

---

9. Open Questions & Future Research

1. **Risk-Asymmetric Confidence-Bounded Routing**: Because $L_{\text{FF}}^{\max} \gg L_{\text{FB}}^{\max}$, symmetric point-estimate routing ($\widehat{C}_F < \widehat{C}_B$) is decision-theoretically inadmissible. Future controllers should test asymmetric upper-confidence rules of the form:

$$\pi_\lambda(x) = \texttt{Frontier} \iff \widehat{C}_{\text{bulk}}(x) - \widehat{C}_F(x) > \lambda_{\text{risk}} \cdot \sqrt{\widehat{\text{Var}}[C_F \mid P_B(x)]},$$

   defaulting conservatively to Bulk Transfer whenever structural variance remains unresolved.
2. **Logarithmic & Randomized Probe Complexity**: While deterministic sublinear probes fail against worst-case equivalence classes (Lemma 1), characterizing randomized probe budgets $B(N, \epsilon, \delta)$ under natural workload priors (e.g., Zipfian directory edits) remains open.
3. **Multi-Substrate Boundary Composition**: Quantifying whether boundary tolls C_{boundary} compose additively or super-additively across chained substrates (e.g., DNS name indirection $\rightarrow$ IPFS DAG fetch $\rightarrow$ checkpointed state fold).

---

10. Conclusion

Network-Native Computation is neither a universal substitute for arithmetic processors nor a zero-cost synchronization panacea. Across eight experiments—including two deliberate negative and boundary results (Exp01 and Exp08)—our findings crystallize into a single governing principle:

> **The NNC Observation–Realization Principle:** *Continuously maintained network structure can make a consequential state distinction cheap to realize ($O(|F|)$ or $O(k)$), without making it cheap to discover prospectively when that distinction is cheap.*

When authenticated mutation logs exist, Δ wins; when only cryptographic commitments are attested, maintained invariants expose genuine multi-order-of-magnitude bandwidth and CPU reductions across CAS, Git, IPFS, and checkpointed state machines. Yet under bounded pre-reconciliation observation, hierarchical commitments conceal deep geometric ambiguity whose asymmetric tail penalties ($27.4\times$ vs. $3.4\times$) punish naive runtime routing. Rigorous accounting of computation across networks must therefore always evaluate the complete four-term physical cost:

$$C_{\text{total}} = C_{\text{observe}} + C_{\text{decide}} + C_{\text{boundary}} + C_{\text{realize}}.$$

Reproducibility & Data Availability

All eight micro-experiments, preregistered Exp08 kill conditions (`PREREGISTRATION.md`), machine-readable benchmark ledgers (`ledger.json`), and the `reproducibility/` verification suite (`manifest.json`, `verify_results.py`, and `run_all_experiments.py --smoke`) are archived in the companion repository using Python standard-library modules only (plus stock `git >= 2.40` for Exp05).

References

Acar, Umut A., Guy E. Blelloch, and Robert Harper. 2002. “Adaptive Functional Programming.” *Proceedings of the 29th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL ’02)*, 247–59. https://doi.org/10.1145/503272.503296.

Alexander, D. Scott, William A. Arbaugh, Michael W. Hicks, et al. 1998. “The SwitchWare Active Network Architecture.” *IEEE Network* 12 (3): 29–36. https://doi.org/10.1109/65.690959.

Benet, Juan. 2014. *IPFS — Content Addressed, Versioned, P2P File System*. https://arxiv.org/abs/1407.3561.

Bosshart, Pat, Dan Daly, Glen Gibb, et al. 2014. “P4: Programming Protocol-Independent Packet Processors.” *ACM SIGCOMM Computer Communication Review* 44 (3): 87–95. https://doi.org/10.1145/2656877.2656890.

Buhrman, Harry, and Ronald de Wolf. 2002. “Complexity Measures and Decision Tree Complexity: A Survey.” *Theoretical Computer Science* 288 (1): 21–43. https://doi.org/10.1016/S0304-3975(01)00144-X.

Chacon, Scott, and Ben Straub. 2014. *Pro Git*. 2nd ed. Apress. https://doi.org/10.1007/978-1-4842-0076-6.

Dang, Huynh Tu, Daniele Sciascia, Marco Canini, Fernando Pedone, and Robert Soulé. 2015. “NetPaxos: Consensus at Network Speed.” *ACM SIGCOMM Computer Communication Review* 45 (4): 5–11. https://doi.org/10.1145/2740070.2740077.

Demers, Alan, Dan Greene, Carl Hauser, et al. 1987. “Epidemic Algorithms for Replicated Database Maintenance.” *Proceedings of the 6th Annual ACM Symposium on Principles of Distributed Computing (PODC ’87)*, 1–12. https://doi.org/10.1145/41840.41841.

Eppstein, David, Michael T. Goodrich, Frank Uyeda, and George Varghese. 2011. “What’s the Difference? Efficient Set Reconciliation Without Prior Context.” *Proceedings of the ACM SIGCOMM 2011 Conference*, 218–29. https://doi.org/10.1145/2018436.2018462.

Goldreich, Oded, Shafi Goldwasser, and Dana Ron. 1998. “Property Testing and Its Connection to Learning and Approximation.” *Journal of the ACM* 45 (4): 653–750. https://doi.org/10.1145/285055.285060.

Goodrich, Michael T., and Michael Mitzenmacher. 2011. “Invertible Bloom Lookup Tables.” *Proceedings of the 49th Annual Allerton Conference on Communication, Control, and Computing*, 792–99. https://doi.org/10.1109/Allerton.2011.6120248.

Goodrich, Michael T., Roberto Tamassia, and Andrew Schwerin. 2001. “Implementation of an Authenticated Dictionary with Skip Lists and Commutative Hashing.” *Proceedings of the DARPA Information Survivability Conference and Exposition II (DISCEX ’01)* 2: 68–82. https://doi.org/10.1109/DISCEX.2001.932160.

Jin, Xin, Xiaozhou Li, Haoyu Zhang, et al. 2017. “NetCache: Balancing Key-Value Stores with Fast in-Network Caching.” *Proceedings of the 26th Symposium on Operating Systems Principles (SOSP ’17)*, 121–36. https://doi.org/10.1145/3132747.3132764.

Lakshman, Avinash, and Prashant Malik. 2010. “Cassandra: A Decentralized Structured Storage System.” *ACM SIGOPS Operating Systems Review* 44 (2): 35–40. https://doi.org/10.1145/1773912.1773922.

Maymounkov, Petar, and David Mazières. 2002. “Kademlia: A Peer-to-Peer Information System Based on the XOR Metric.” *Peer-to-Peer Systems (IPTPS 2002)*, Lecture notes in computer science, vol. 2429: 53–65. https://doi.org/10.1007/3-540-45748-8_5.

McSherry, Frank, Derek Gordon Murray, Rebecca Isaacs, and Michael Isard. 2013. “Differential Dataflow.” *Proceedings of the 6th Biennial Conference on Innovative Data Systems Research (CIDR ’13)*.

Merkle, Ralph C. 1987. “A Digital Signature Based on a Conventional Encryption Function.” *Advances in Cryptology — CRYPTO ’87*, Lecture notes in computer science, vol. 293: 369–78. https://doi.org/10.1007/3-540-48184-2_32.

Minsky, Yaron, Ari Trachtenberg, and Richard Zippel. 2003. “Set Reconciliation with Almost Optimal Communication Complexity.” *IEEE Transactions on Information Theory* 49 (9): 2213–18. https://doi.org/10.1109/TIT.2003.815784.

Mockapetris, Paul. 1987. *Domain Names — Concepts and Facilities*. Internet Engineering Task Force (IETF), RFC 1034, STD 13. https://doi.org/10.17487/RFC1034.

Muthitacharoen, Athicha, Benjie Chen, and David Mazières. 2001. “A Low-Bandwidth Network File System.” *Proceedings of the 18th ACM Symposium on Operating Systems Principles (SOSP ’01)*, 174–87. https://doi.org/10.1145/502034.502052.

Ongaro, Diego, and John Ousterhout. 2014. “In Search of an Understandable Consensus Algorithm.” *Proceedings of the 2014 USENIX Annual Technical Conference (USENIX ATC ’14)*, 305–19.

Rose, Scott, and Wouter Wijngaards. 2012. *DNAME Redirection in the DNS*. Internet Engineering Task Force (IETF), RFC 6672. https://doi.org/10.17487/RFC6672.

Schneider, Fred B. 1990. “Implementing Fault-Tolerant Services Using the State Machine Approach: A Tutorial.” *ACM Computing Surveys* 22 (4): 299–319. https://doi.org/10.1145/98163.98167.

Tamassia, Roberto. 2003. “Authenticated Data Structures.” *Algorithms — ESA 2003*, Lecture notes in computer science, vol. 2832: 2–5. https://doi.org/10.1007/978-3-540-39658-1_2.

Tennenhouse, David L., and David J. Wetherall. 1996. “Towards an Active Network Architecture.” *ACM SIGCOMM Computer Communication Review* 26 (2): 5–17. https://doi.org/10.1145/231699.231701.

Tridgell, Andrew, and Paul Mackerras. 1996. *The rsync Algorithm*. TR-CS-96-05. Australian National University.

Williams, Sam, Viktor Diordiiev, Lev Berman, and Ivan Uemlianin. 2019. *Arweave: A Protocol for Economically Sustainable Information Permanence*. Arweave Yellow Paper.

Williams, Sam, and Forward Research. 2024. *AO: The Hyper Parallel Computer — Protocol Specification*. Forward Research Technical Specification.

Yao, Andrew Chi-Chih. 1977. “Probabilistic Computations: Toward a Unified Measure of Complexity.” *Proceedings of the 18th Annual Symposium on Foundations of Computer Science (FOCS ’77)*, 222–27. https://doi.org/10.1109/SFCS.1977.24.