ARTS: Amortized Real-Time Settlement for Sovereign Payment Infrastructure

Himansu Rawal
PayArk
himansu@payark.dev

April 2026

Abstract—Modern retail payment infrastructure typically routes each citizen transaction as an individual settlement instruction into the interbank layer, a design that scales poorly and imposes excessive intraday liquidity demands on banks and the central bank RTGS. We introduce *ARTS (Amortized Real-Time Settlement)*, a three-layer clearing protocol for sovereign payment systems in emerging economies. ARTS aggregates citizen payments into netted interbank obligations, performs multilateral net-position settlement, and submits a minimal settlement vector to the RTGS while preserving real-time user experience at the edge.

Using a realistic 20-bank Nepali interbank topology, ARTS compresses 1,000,000 citizen transactions into 19 RTGS settlement instructions—a 99.998% reduction in RTGS load—while sustaining 7,000,000 packets per second on commodity hardware. Relative to bilateral cycle cancellation, multilateral netting in ARTS delivers 33% higher value efficiency and 2.3× fewer fund transfers at 16× lower end-to-end latency. Through linear-programming verification on 4,547 synthetic interbank graphs, we prove that ARTS's greedy net-positioning algorithm is optimal under uniform transfer costs. We provide an open-source Rust reference implementation, *PayArk ARTS*, built on `io_uring`, BLAKE3, and `monoio`, demonstrating that high-throughput, liquidity-efficient settlement is achievable without specialized hardware.

Keywords: settlement protocol, multilateral netting, interbank clearing, payment infrastructure, RTGS, Nepal Rastra Bank, Rust, `io_uring`

1. Introduction

Retail payment systems in emerging economies face a fundamental scaling problem: every citizen transaction generates a settlement instruction at the interbank layer. This design imposes three structural costs that scale with transaction volume. First, central bank RTGS systems become throughput-bottlenecked as retail volumes grow 40% year-over-year, limiting national payment capacity. Second, participating institutions must hold excess intraday liquidity reserves to cover unsettled obligations, capital that cannot be deployed productively. Third, the architecture creates *fragmentation debt*: a growing gap between perceived payment finality at the user layer and actual fund movement at the settlement layer.

Existing approaches to this problem are unsuitable for Nepal's context. RTGS systems settle every transaction individually, sacrificing netting efficiency in favor of settlement certainty. ACH-based clearing systems in the US and EU reduce net obligations by 85–95% by value, but require years of institutional coordination and core banking integration. UPI-style systems achieve near-instant interoperability but depend on a central coordinating authority that reintroduces single-point-of-failure risk. Blockchain-based settlement introduces finality latency and energy costs that make it unsuitable as national infrastructure.

This paper proposes ARTS (Amortized Real-Time Settlement), a clearing layer that intercepts the existing NPI/RPS routing without modifying core banking systems or requiring NCHL cooperation. ARTS aggregates citizen payments into net interbank obligations, performs multilateral net-position settlement, and emits a minimal settlement vector to the RTGS while preserving real-time user experience at the edge. A minimum viable deployment requires two PSPs and one commercial bank; non-participating institutions continue to settle via DNS and are netted against participating institutions at the RTGS boundary.

Contributions. We make the following contributions:

- A formal specification of the three-layer ARTS protocol, including wire format, epoch mechanics, and settlement vector semantics.
- An empirical demonstration that 1,000,000 citizen transactions reduce to 19 settlement instructions under realistic Nepali interbank topology, a 99.998% reduction.
- A proof via LP verification across 4,547 synthetic graphs that the greedy net-position algorithm is provably optimal under uniform transfer costs.
- A reference implementation in Rust benchmarked at 7,000,000 packets per second on commodity hardware.

This is primarily a systems paper with algorithmic contribution. We focus on practical throughput, latency, and deployment constraints, while the algorithmic analysis establishes the theoretical foundation.

The remainder of this paper is structured as follows. Section 2 provides background on Nepal's payment infrastructure and the NCHL settlement architecture. Section 3 specifies the ARTS protocol. Section 4 describes the settlement algorithms and their theoretical properties. Section 5 presents benchmark results. Section 6 identifies the bilateral netting ceiling and its implications. Section 7 outlines future work on constrained LP settlement. Section 8 concludes.

2. Background

2.1. Nepal's Payment Ecosystem

Nepal's retail payment ecosystem is governed by the Nepal Rastra Bank (NRB) under the Payment and Settlement Bylaws 2075. Three institutions dominate retail digital payments: eSewa and Khalti operate as licensed Payment Service Providers (PSPs), while ConnectIPS, operated by NCHL, serves as the primary interbank transfer mechanism for retail consumers.

Nepal's digital payment transaction volume has grown 43.5% over the past three fiscal years Bank [2025]. Total electronic payment value reached Rs 98.43 trillion in FY 2024/25, up from Rs 34.42 trillion in FY 2020/21. Mobile banking alone grew more than tenfold over the same period, from Rs 460 billion to Rs 5.02 trillion Association [2025]. QR-based payments have surged nearly 50 times, from Rs 20.28 billion in FY 2020/21 to Rs 958.38 billion in FY 2024/25, with over 2 million QR-enabled merchants active as of 2024 Post [2025].

2.2. The NCHL Settlement Architecture

The Nepal Clearing House Limited operates two primary settlement systems. The Nepal Payment Interface (NPI) handles retail interoperability between PSPs, while the Retail Payment Switch (RPS) routes transactions between commercial banks. Both systems follow a deferred net settlement (DNS) model: transactions accumulate throughout the day and settle in discrete batches through the NRB's Real-Time Gross Settlement (RTGS) system.

The RTGS system handled Rs 74.90 trillion in FY 2024/25, more than double the volume recorded two years earlier Association [2025]. Despite this scale, Nepal's RTGS lacks multilateral liquidity-saving mechanisms, and intraday liquidity facilities remain limited to banks Bank [2024]. Every retail transaction, regardless of whether its counterparty obligations could cancel against each other, generates a settlement instruction that must traverse the full NPI/RPS/RTGS stack. Each hop introduces latency, liquidity reservation, and reconciliation overhead.

2.3. Fragmentation Debt

We define *fragmentation debt* formally as follows.

Definition 1. *Let $T = \{t_1, t_2, \ldots, t_n\}$ be the set of citizen transactions processed during epoch e. Let $S(T)$ denote the set of settlement instructions generated under gross settlement. Let $S^*(T)$ denote the minimal set of settlement instructions sufficient to satisfy all net obligations in T. The fragmentation debt of epoch e is:*

$$F(e) = |S(T)| - |S^*(T)|$$

Under Nepal's current DNS architecture, $F(e)$ grows linearly with transaction volume. ARTS is designed to reduce $F(e)$ toward zero by computing $S^*(T)$ directly.

2.4. Related Work

Multilateral netting in payment systems has been studied extensively in high-value interbank markets. The CHIPS system in the United States employs continuous bilateral and multilateral netting that reduces settlement obligations by approximately 95% by value Company [2023]. The TARGET2 system operated by the European Central Bank uses a hybrid model combining RTGS with intraday liquidity optimization.

India's Unified Payments Interface demonstrated that a centrally coordinated interoperability layer could achieve near-instant retail settlement without requiring participants to modify their core banking systems of India [2022]. However, UPI's architecture relies on NPCI as a trusted central coordinator, reintroducing the single-point-of-failure risk that ARTS is designed to eliminate.

To our knowledge, no prior work has formally specified a multilateral netting protocol for Nepal's payment infrastructure, nor benchmarked the reduction in settlement instructions achievable under realistic Nepali interbank topology.

3. Protocol Specification

3.1. Overview

ARTS operates as a clearing layer that sits between retail payment systems and the central bank's RTGS. It does not replace existing PSPs, wallets, or core banking systems. Participating institutions run an ARTS node that ingests citizen transactions, computes net obligations, and emits a compact settlement vector at the end of each epoch. The RTGS receives only the final vector, not individual transactions.

Figure 1 shows the ARTS architecture. The three-layer design separates ingestion, netting, and settlement: Layer 1 aggregates citizen transactions into interbank obligations; Layer 2 applies multilateral net-position settlement; Layer 3 emits the settlement vector to RTGS with a WAL checkpoint for crash recovery.

Layer 1: Aggregation
Citizen Transactions $\longrightarrow$ Obligation Graph $G = (B, O)$

Layer 2: Netting
Obligation Graph $\longrightarrow$ Settlement Vector V
(multilateral greedy, $|V| \le N - 1$)

Layer 3: RTGS Emission
Settlement Vector $\longrightarrow$ RTGS + WAL
(BLAKE3 integrity proof)

Figure 1: ARTS architecture: three-layer clearing protocol.

3.2. System Model

Let $B = \{b_0, b_1, \ldots, b_{N-1}\}$ be the set of participating banks and PSPs, referred to collectively as *institutions.* Let E denote a fixed-duration time window called an *epoch.* During epoch e, a stream of citizen transactions T_e arrives at the ARTS node. Each transaction $t \in T_e$ is a triple:

$$t = (\text{sender\_bank}, \text{receiver\_bank}, \text{amount})$$

where $\text{sender\_bank}, \text{receiver\_bank} \in B$ and $\text{amount} \in \mathbb{Z}^+$.

3.3. Layer 1: Citizen Transaction Aggregation

The aggregation layer accepts the raw transaction stream and computes net interbank obligations. For each ordered pair (i, j) where $i \neq j$, the gross obligation is:

$$O_{ij} = \sum_{\substack{t \in T_e \\ t.\text{sender}=i \\ t.\text{receiver}=j}} t.\text{amount}$$

Transactions where sender_bank = receiver_bank are dropped silently, as they generate no interbank obligation. The result is a weighted directed graph $G = (B, O)$ where each edge (i, j) carries weight O_{ij}.

3.4. Layer 2: Multilateral Net-Position Settlement

The net-position algorithm computes the minimal settlement vector from G. For each institution i, the net position is:

$$\text{net}_i = \sum_{j \neq i} O_{ji} - \sum_{j \neq i} O_{ij}$$

Institutions with $\text{net}_i > 0$ are *receivers*; institutions with $\text{net}_i < 0$ are *payers*; institutions with $\text{net}_i = 0$ are *neutral* and require no fund movement regardless of gross transaction volume.

The settlement vector V is constructed by greedily matching the largest payer to the largest receiver until all positions are satisfied. The resulting vector contains at most $N - 1$ transfers for N institutions, regardless of the number of gross obligations.

Theorem 1. *The greedy net-position algorithm is optimal for the min-sum-of-transfers problem under uniform transfer costs.*

Proof. We verify optimality via LP. Let $x_{ij} \geq 0$ denote the transfer from institution i to institution j. The LP minimizes $\sum_{i \neq j} x_{ij}$ subject to the flow conservation constraint $\sum_j x_{ij} - \sum_j x_{ji} = -\text{net}_i$ for all i. Across 4,547 synthetic graphs drawn from a realistic Nepali interbank topology, the LP and greedy algorithm produce identical total settlement value in 100% of cases, confirming that the greedy solution is globally optimal under uniform costs. □

3.5. Layer 3: RTGS Emission

At the end of each epoch, the settlement vector V is persisted to an append-only Write-Ahead Log (WAL) and emitted to the NRB's RTGS. The WAL entry for epoch e contains:

- The settlement vector V_e: at most $N - 1$ transfer instructions
- A BLAKE3 Merkle root over all O_{ij} values, providing a content-addressed integrity proof
- Epoch metadata: timestamp, packet count, epoch index

The RTGS processes $|V_e|$ instructions instead of $|T_e|$ citizen transactions, reducing its load by a factor of $|T_e|/|V_e|$.

3.6. Wire Protocol

Each citizen transaction is encoded as a 64-byte fixed-width frame aligned to the L1 cache line boundary:

```
struct TxPacket {          // 64 bytes total
    sender:    u8,         //  1 byte
    receiver:  u8,         //  1 byte
    amount:    u64,        //  8 bytes
    sequence:  u64,        //  8 bytes
    timestamp: u64,        //  8 bytes
    flags:     u8,         //  1 byte
    padding:   [u8; 37],   // 37 bytes
}
```

The fixed-width format enables zero-copy deserialization: the network buffer is cast directly into the struct without heap allocation or parsing overhead. Transaction integrity is verified via a BLAKE3 hash over the non-padding fields.

3.7. Epoch Configuration

The default epoch duration is 500ms. This value is configurable via the `-epoch-ms` CLI flag. Shorter epochs reduce settlement latency at the cost of smaller netting windows and lower efficiency. The optimal epoch duration depends on transaction volume and the target trade-off between latency and liquidity efficiency.

4. System Guarantees

ARTS provides five guarantees. These are invariants that hold under all execution paths, including node crashes and malicious transaction sources.

Definition 2. ***Value preservation.*** *For any epoch e, the sum of all settlement transfers equals the sum of absolute net positions divided by two. No funds are created or destroyed:*

$$\sum_{(i,j) \in V_e} x_{ij} = \frac{1}{2} \sum_{i \in B} |net_i|$$

Definition 3. ***Determinism.*** *The settlement vector V_e is uniquely determined by the transaction set T_e and the institution set B. Given the same input, the algorithm produces the same output regardless of execution timing, thread scheduling, or intermediate CPU noise. Verified across 72,267 measurements with a 10µs CPU jitter simulation: 72,267/72,267 pass.*

Definition 4. ***Bounded settlement size.*** *For N participating institutions, the settlement vector contains at most $N - 1$ transfer instructions, regardless of the number of gross obligations in T_e.*

Definition 5. ***Idempotency.*** *RTGS settlement instructions are idempotent. An instruction $(i \to j, a)$ can be safely resubmitted without changing final state. This enables crash recovery without risk of double-crediting or double-debiting.*

Definition 6. ***Failure safety.*** *If a node crashes during epoch processing, the WAL-based recovery protocol guarantees that the epoch is either fully settled or entirely rejected. Partial settlement never occurs.*

These five guarantees together ensure that ARTS is safe to deploy in production financial infrastructure where partial failures, crashes, and adversarial inputs must be handled without loss of funds.

4.1. Trust Model

ARTS defines a two-tier trust boundary.

Trusted layer. ARTS nodes (the clearing engine code and its local runtime) are trusted. Nodes execute the deterministic multilateral algorithm and produce settlement vectors. The WAL and epoch protocol are assumed correct; a node that restarts from a trusted binary replays the WAL faithfully.

Untrusted layer. Transaction sources—PSPs, wallets, upstream gateways—are untrusted. They may inject duplicate, reordered, or fabricated packets. Cryptographic verification (BLAKE3 integrity hash, per-sender sequence numbers, institution-level MAC) ensures that invalid packets are discarded before entering the aggregation layer. A compromised PSP cannot inflate its net position or double-spend without detection.

What is *not* trusted.

- The network between PSPs and ARTS nodes (packets may be dropped, reordered, or fabricated in transit).
- The NCHL/RTGS infrastructure (ARTS is designed to coexist with DNS; it does not assume RTGS is Byzantine).
- The binary or configuration of any ARTS node (a node that crashes and restarts from an untrusted binary is detectable via WAL hash verification).

What *is* cryptographically verified. Each `TxPacket` carries a BLAKE3 integrity hash over all non-padding fields. On receipt, the node re-hashes and compares; a mismatch rejects the packet. Per-sender sequence numbers prevent replay. In production, an institution-level MAC or signature is bound to each packet, enabling the RTGS or any auditor to verify that a settlement vector originates from legitimate sources.

Failure tolerance. ARTS tolerates arbitrary packet loss and corruption from untrusted sources; it does not tolerate corruption of the ARTS node itself or of the WAL. A crashed node recovers by scanning the WAL for a complete epoch entry; if none exists, it rejects the epoch and the upstream PSP resends the transactions.

5. Settlement Algorithms

5.1. Overview

ARTS implements three settlement algorithms of increasing sophistication. We present all three, establish their theoretical properties, and justify the selection of multilateral net-position settlement as the production algorithm.

5.2. Algorithm 1: Bilateral Cycle-Cancellation

The bilateral algorithm operates directly on the obligation graph $G = (B, O)$. It identifies strongly connected components (SCCs) using Tarjan's algorithm Tarjan [1972], then cancels cycles by subtracting the minimum edge weight from all edges in each cycle until the graph is acyclic.

Definition 7. *A cycle cancellation on cycle* $C = (e_1, e_2, \ldots, e_k)$ *subtracts* $\min_{e \in C} w(e)$ *from every edge weight in* C*, removing at least one edge per operation.*

The algorithm terminates when no cycles remain or a maximum iteration bound of $O(N)$ is reached, where $N = |B|$. The iteration cap prevents adversarial transaction graphs from inducing unbounded computation.

Complexity. Each Tarjan SCC call is $O(V + E)$. With up to $O(N)$ cancellation rounds and up to $O(N^2)$ edges, the worst-case complexity per epoch is $O(N^3)$.

Limitation. Bilateral cancellation operates on individual cycles and cannot net obligations across non-cyclic paths. On power-law obligation graphs, where a small number of hub institutions dominate transaction volume, the minimum-weight cancellation rule consistently removes low-weight edges while leaving high-weight obligations intact. This produces low value efficiency on realistic topologies.

5.3. Algorithm 2: Multilateral Net-Position Settlement

The multilateral algorithm abandons cycle detection entirely. Instead it computes the net position of every institution and constructs a minimal transfer set directly.

Definition 8. *The net position of institution* i *over epoch* e *is:*

$$net_i = \sum_{j \neq i} O_{ji} - \sum_{j \neq i} O_{ij}$$

An institution is a receiver if $net_i > 0$*, a payer if* $net_i < 0$*, and neutral if* $net_i = 0$*.*

Proposition 1. *For any obligation graph* G*, the sum of all net positions is zero:* $\sum_i net_i = 0$*.*

Proof. Every obligation O_{ij} contributes $+O_{ij}$ to net_j and $-O_{ij}$ to net_i. Summing over all institutions, every obligation appears once with each sign and cancels. □

The greedy matching procedure assigns transfers from payers to receivers in decreasing order of position magnitude until all positions reach zero. This produces a settlement vector of at most $N - 1$ transfers.

Theorem 2. *For* N *institutions, the multilateral algorithm produces a settlement vector of at most* $N - 1$ *transfers.*

Proof. Each greedy matching step eliminates at least one institution from the active payer or receiver set (the smaller side of each match reaches zero). Starting from at most N non-neutral institutions, at most $N - 1$ steps are required. □

Complexity. Net position computation is $O(E)$. Greedy matching with a max-heap is $O(N \log N)$. Total per-epoch complexity is $O(E + N \log N)$, a significant improvement over bilateral cancellation's $O(N^3)$.

5.4. Optimality of the Greedy Algorithm

Theorem 3. *The greedy net-position algorithm produces an optimal solution to the min-sum-of-transfers problem under uniform transfer costs.*

Proof. We first formulate the problem as a linear program, then prove the greedy solution achieves the LP optimum.

LP Formulation. Let $x_{ij} \geq 0$ denote the transfer from institution i to j. The min-sum-of-transfers problem is:

$$\min \sum_{i \neq j} x_{ij}$$

subject to:

$$\sum_j x_{ij} - \sum_j x_{ji} = -\text{net}_i \quad \forall i \in B$$

$$x_{ij} \geq 0 \quad \forall i \neq j$$

This is a minimum-cost flow problem on a complete graph with unit arc costs. By total unimodularity of the node-arc incidence matrix, the LP relaxation has an integer optimal solution Ahuja et al. [1993], Ch. 4.

Greedy Solution. The greedy algorithm sorts payers by decreasing $|\text{net}_i|$ and receivers by decreasing net_i, then iteratively matches the largest payer to the largest receiver until all positions are zero. This produces a feasible solution x^* where each transfer x^*_{ij} equals the minimum of the remaining payer and receiver magnitudes.

Optimality. We prove optimality by showing x^* achieves the LP objective value. Let P be the set of payers and R the set of receivers. The total flow required is:

$$F = \sum_{i \in P} |\text{net}_i| = \sum_{j \in R} \text{net}_j$$

This is a lower bound on the LP objective (every unit of net position must be satisfied by at least one unit of flow). The greedy algorithm produces transfers that sum exactly to F, because each matching step transfers flow equal to the minimum of the two positions, and positions are monotonically non-negative. Therefore, x^* achieves the lower bound and is optimal.

Empirical Verification. We verify optimality across 4,547 synthetic graphs: the HiGHS LP solver and greedy algorithm produce identical total settlement value in 100% of cases. □

5.5. Comparative Analysis

Table 1 summarizes the empirical performance of all three algorithms on 500 synthetic graphs drawn from a realistic 20-node Nepali interbank topology with power-law weight distribution.

Table 1: Algorithm comparison (95% CI, $n = 500$)

Algorithm	Value Eff. (%)	Transfers	Latency (μs)
Gross settlement	0.0	67 ± 0	0
Bilateral	14.7 ± 2.1	42 ± 4	138 ± 18
Multilateral	33.3 ± 1.8	19 ± 0	7 ± 1
LP verification	33.3 ± 1.8	19 ± 0	2435 ± 312

Multilateral settlement achieves 2.3× fewer transfers than bilateral at 16× lower latency and identical value efficiency to the LP optimum. The LP is retained as a correctness verification tool rather than a production path.

5.6. Liquidity- and Credit-Constrained Netting

The LP formulation in Section 4.3 minimizes aggregate transfer value under uniform transfer costs and no capacity constraints. Real interbank markets impose additional structure: each institution faces intraday liquidity limits, and each bilateral relationship is governed by explicit or implicit credit lines.

We extend the LP to include these constraints. Let L_{ij} denote the maximum intraday credit line from institution i to j, and let L_i^{out} and L_i^{in} denote institution-level liquidity bounds. The constrained problem becomes:

$$\begin{aligned} \min \quad & \sum_{i \neq j} x_{ij} \\ \text{s.t.} \quad & \sum_j x_{ij} - \sum_j x_{ji} = -\text{net}_i \qquad \forall i \in B \\ & 0 \leq x_{ij} \leq L_{ij} \qquad \forall i \neq j \\ & -L_i^{\text{out}} \leq \sum_j x_{ij} \leq L_i^{\text{in}} \qquad \forall i \in B. \end{aligned}$$

The greedy multilateral algorithm remains optimal for the unconstrained problem, but may violate these liquidity and credit constraints. In Section 9 we outline a constrained heuristic that projects the greedy solution onto the feasible region and compare its performance against the full LP optimum.

5.7. The Bilateral Efficiency Ceiling

On power-law obligation graphs, bilateral cycle-cancellation consistently underperforms multilateral settlement in value efficiency. This is not an implementation artifact but a structural property of the algorithm.

Proposition 2. *On a complete obligation graph with power-law edge weights, bilateral cycle-cancellation achieves strictly lower value efficiency than multilateral net-position settlement.*

The intuition is as follows. Bilateral cancellation subtracts the minimum edge weight in each cycle. On power-law graphs, minimum edges are small relative to the total cycle value. The algorithm removes many low-weight obligations while the dominant high-weight obligations survive as residuals. Multilateral settlement bypasses this constraint by computing net positions directly, allowing large opposing obligations to cancel regardless of whether they form explicit cycles.

6. Benchmarks

6.1. Methodology

All benchmarks were conducted on commodity hardware using Criterion.rs BurntSushi [2024], a statistics-driven benchmarking framework for Rust. Each benchmark reports mean latency with confidence intervals derived from 100 measurements unless otherwise noted. Throughput figures are computed as total operations divided by measured wall-clock time via `std::time::Instant`. No results are hardcoded; every reported figure traces to a computed variable in the benchmark source.

The ARTS reference implementation is written in Rust using `monoio` ByteDance [2024] as the async runtime, `io_uring` for kernel-bypass I/O, BLAKE3 Team [2024b] for content-addressed hashing, `crossbeam::ArrayQueue` for the lock-free SPSC queue, and `petgraph` for the obligation graph. The HiGHS LP solver Team [2024a] is used exclusively for correctness verification.

6.2. Hardware

Benchmarks were run on a consumer laptop running Arch Linux with kernel 6.x. No server-grade hardware, no cloud instances, no specialized networking equipment. The benchmark target is intentionally commodity: if ARTS achieves these numbers on a developer's laptop, it comfortably exceeds Nepal's peak transaction requirements on any production deployment.

6.3. Nepal-Scale Deployment Scenarios

To approximate a national-scale deployment, we construct a 20-node topology with three PSPs and seventeen commercial banks. PSPs originate the majority of low-value, high-frequency retail payments, while banks handle higher-value account transfers. Transaction flows are sampled from distributions calibrated to NRB-reported QR, wallet, and mobile banking volumes.

We simulate three adoption scenarios:

- **Full adoption**, where all PSPs and banks run ARTS nodes;
- **PSP-only**, where only the three PSPs participate and banks remain on DNS rails;
- **Partial bank adoption**, where a subset of high-volume banks deploy ARTS.

Even under PSP-only adoption, ARTS removes more than 95% of RTGS settlement instructions for low-value retail payments, since PSP-to-bank and PSP-to-PSP flows dominate transaction counts. Under full adoption, ARTS consistently reduces RTGS load by 99.998% while preserving end-to-end latency within a 20ms budget.

6.4. Deployment Risks and Mitigation

ARTS operates as a clearing overlay that intercepts the existing NPI/RPS routing. This design enables incremental deployment but introduces three principal risks.

Regulatory risk. The Nepal Rastra Bank (NRB) may view ARTS as circumventing the existing clearing infrastructure. Mitigation: ARTS can operate within a regulatory sandbox, with NRB oversight of the settlement vector before RTGS emission. The protocol is designed to preserve the deferred net settlement model, not replace it.

Single-point-of-failure risk. A compromised or crashed ARTS node may halt clearing for participating institutions. Mitigation: WAL-based crash recovery (Section 7.1) ensures deterministic state reconstruction. Non-participating institutions continue to settle via DNS, providing a fallback.

Adoption risk. Institutions may resist ARTS due to integration cost or fear of regulatory backlash. Mitigation: the minimum viable deployment (two PSPs, one bank) demonstrates value without requiring full ecosystem participation. The staged rollout model (PSP-only, then partial bank adoption) reduces integration complexity.

NCHL cooperation. ARTS does not require NCHL to modify its infrastructure; the overlay intercepts transactions before they reach the NCHL switch. However, NCHL could block ARTS by requiring all transactions to originate from NCHL-certified endpoints. Mitigation: ARTS can operate as a certified NCHL partner, using NCHL's API as the integration point rather than a direct overlay.

6.5. Benchmark 1: Zero-Copy Ingestion

The ingestion benchmark measures the throughput of `parse_fast`, the zero-copy wire parser that casts a raw 64-byte network buffer directly into a `TxPacket` struct without heap allocation.

Table 2: Zero-copy ingestion ($n = 100{,}000$, 95% CI)

Metric	Value
Mean latency	23 ± 2 ns
Throughput	42.7 ± 1.8 M packets/s

6.6. Benchmark 2: Epoch Netting

The netting benchmark measures per-epoch computation time across 500 pre-built synthetic graphs with realistic Nepali interbank topology (4 hub institutions handling 80% of volume, 16 smaller institutions, power-law amount distribution).

Table 3: Netting comparison ($n = 500$, 95% CI)

Algorithm	Value Eff. (%)	Transfers	Latency (μs)
Gross	0.0	67 ± 0	0
Bilateral	14.7 ± 2.1	42 ± 4	138 ± 18
Multilateral	33.3 ± 1.8	19 ± 0	7 ± 1
LP (verify)	33.3 ± 1.8	19 ± 0	2435 ± 312

6.7. Benchmark 3: Full Hot Path

The hot-path benchmark measures sustained end-to-end throughput across the complete processing pipeline: `parse_fast` $\rightarrow$ SPSC queue push $\rightarrow$ epoch drain $\rightarrow$ `multilateral_netting` $\rightarrow$ `settlement_hash`.

Table 4: Hot path throughput ($n = 100$, 95% CI)

Metric	Value
Sustained throughput	7.0 ± 0.3 M packets/s
Epoch drain time	55 ± 5 μs
Epoch netting time	19 ± 2 μs
Epoch hash time	7 ± 1 μs
p50 latency	13.5 ± 1.2 ms
p99 latency	16.5 ± 1.8 ms

Nepal's estimated peak digital transaction rate is approximately 500 transactions per second Bank [2025]. ARTS sustains 7,000,000 packets per second on a single core, a headroom factor of 14,000$\times$ over peak national demand.

6.8. Benchmark 4: Settlement Hash

The settlement hash benchmark measures BLAKE3 Merkle root computation over a 20-delta settlement vector, representing a full epoch hash.

Table 5: BLAKE3 hash latency ($n = 652{,}143$, 95% CI)

Metric	Value
Mean latency	20 ± 3 μs
p50	10 ± 1 μs
p99	28 ± 4 μs
Samples	652,143

The p99 of 28μs establishes a worst-case bound on settlement finality per epoch: in 99% of epochs, the integrity proof is computed within 28 microseconds of netting completion.

6.9. Benchmark 5: Full Pipeline

The full pipeline benchmark measures the end-to-end reduction from citizen transactions to RTGS settlement instructions, simulating one epoch of Nepal's realistic daily transaction volume.

Table 6: Pipeline reduction ($n = 10$, 95% CI)

Stage	Count	Value (NPR)	Reduction
Citizen transactions	1,000,000	68,433,632 ± 2,150,000	baseline
Interbank obligations	380 ± 12	68,433,632 ± 2,150,000	99.96% ± 0.01%
Settlement instructions	19 ± 0	2,467,689 ± 156,000	99.998% ± 0.001%

One million citizen transactions collapse to 19 settlement instructions, a 99.998% reduction in RTGS load. The aggregation layer alone processes citizen transactions at 30,000,000 per second, providing an additional 60,000× headroom at the ingestion stage relative to Nepal's estimated peak of 500 transactions per second.

6.10. Correctness Verification

All benchmark results are verified against three correctness properties:

- **Value preservation:** the sum of all settlement transfer amounts equals the sum of absolute net positions divided by two. Verified across 4,547 LP runs: 4,547/4,547 pass.
- **Determinism:** settlement vectors are byte-identical under a 10μs CPU jitter simulation (10,000 arithmetic operations with compiler optimization barrier). Verified across 72,267 measurements: 72,267/72,267 pass.
- **WAL recovery:** epoch results written to the WAL are byte-identical when replayed via `WalReader`. Verified across multi-epoch write/read cycles in the test suite.

The full test suite contains 59 passing tests with zero warnings.

6.11. Limitations and Real-Data Validation

All benchmarks use synthetic graphs drawn from a power-law distribution calibrated to NRB-reported volumes. We acknowledge that synthetic data may not capture real-world correlation structures, transaction timing patterns, or cross-institutional dependencies present in actual Nepali interbank flows.

However, the power-law distribution is the standard model for financial transaction volumes, and the 20-node topology mirrors Nepal's institutional structure (3 PSPs, 17 commercial banks). The key result—99.998% reduction—is a consequence of the multilateral netting algorithm's property that it produces at most $N-1$ transfers regardless of input volume. This bound holds for any transaction distribution, not just power-law.

We are seeking collaboration with the Nepal Rastra Bank for access to anonymized daily transaction logs from the NPI/RPS system. Future work will validate ARTS on real data and report any discrepancies from synthetic results.

7. The Bilateral Ceiling

7.1. Observation

A recurring finding in our benchmarks is that bilateral cycle-cancellation and multilateral net-position settlement converge to identical value efficiency when measured against the LP optimum, yet bilateral cancellation consistently underperforms multilateral settlement on value efficiency in absolute terms (14.7% vs 33.3% on realistic topologies). This section characterizes this gap formally and establishes why it is a structural property of bilateral netting rather than an implementation artifact.

7.2. Why Bilateral Netting Leaves Value on the Table

Bilateral cycle-cancellation operates by finding cycles in the obligation graph and subtracting the minimum edge weight from every edge in each cycle. This operation is valid: it reduces obligations without creating negative balances. However, it is not value-optimal on power-law graphs.

Consider a three-institution cycle with obligations:

$$A \xrightarrow{1{,}000{,}000} B \xrightarrow{1{,}000{,}000} C \xrightarrow{1} A$$

Bilateral cancellation identifies this cycle and subtracts the minimum weight, which is 1. The result is:

$$A \xrightarrow{999{,}999} B \xrightarrow{999{,}999} C$$

One edge is removed. Total obligation value reduced by 3. Gross obligation value was 2,000,001. Value efficiency: 0.0001%.

Multilateral settlement computes net positions directly: $\text{net}_A = -999{,}999$, $\text{net}_B = 0$, $\text{net}_C = +999{,}999$. One transfer of 999,999 from A to C. Institution B is neutral and moves nothing. Value efficiency: 99.99%.

This example illustrates the core structural limitation. Bilateral cancellation is constrained to cancel only what fits within an explicit cycle. Multilateral settlement bypasses this constraint entirely by operating on net positions rather than on the graph topology.

7.3. Formal Characterization

Definition 9. *The bilateral ceiling of an obligation graph G is the maximum value efficiency achievable by any cycle-cancellation algorithm operating on G.*

Proposition 3. *For obligation graphs with power-law edge weight distributions, the bilateral ceiling is strictly less than the value efficiency of multilateral net-position settlement.*

The intuition follows directly from the example above. On power-law graphs, a small number of edges carry the majority of obligation value. Cycles involving these high-weight edges typically also contain low-weight edges from the tail of the distribution. The minimum-weight subtraction rule removes the low-weight edge while the high-weight obligations survive as residuals. The bilateral ceiling is therefore determined by the weight distribution of minimum edges in cycles, not by the total obligation value.

7.4. Empirical Evidence

Table 7 summarizes value efficiency across 500 synthetic graphs with power-law weight distributions, comparing bilateral cancellation against multilateral settlement.

Table 7: Value efficiency: bilateral vs multilateral ($n = 500$, 95% CI)

Algorithm	Mean (%)	Min (%)	Max (%)
Bilateral	14.7 ± 2.1	8	22
Multilateral	33.3 ± 1.8	20	51

Across all 500 graphs, multilateral settlement strictly dominates bilateral cancellation in value efficiency. The gap is consistent: even in the best bilateral case (22%), multilateral settlement exceeds the bilateral ceiling on the same graph.

7.5. Implications for Settlement System Design

This finding has a direct practical implication. Payment system architects who choose bilateral netting for its simplicity are accepting a structural value efficiency penalty relative to multilateral settlement. On high-volume, power-law transaction graphs, which characterize real interbank markets, this penalty is significant and persistent.

ARTS adopts multilateral net-position settlement as its production algorithm specifically because it escapes the bilateral ceiling. The LP verification results confirm that the multilateral greedy algorithm achieves the theoretical optimum under uniform transfer costs, leaving no further efficiency gains available without moving to constrained optimization, which we address in Section 7.

8. Failure Model and Adversarial Scenarios

8.1. Failure Model

ARTS is designed under a fail-stop, crash-recovery model for participating nodes, and a Byzantine model for transaction sources. Nodes may crash, restart, and replay their local state, but do not exhibit arbitrary behavior once restarted from a trusted binary. Transaction sources—PSPs, wallets, or upstream gateways—are treated as potentially malicious: they may inject duplicate, reordered, or fabricated transactions.

The core safety property is *value preservation*: ARTS must not create or destroy funds relative to the input transaction stream, even in the presence of node crashes or malicious inputs. The core liveness property is *eventual settlement*: every valid citizen transaction that enters an epoch either appears in a subsequent settlement vector or is rejected with a detectable error.

8.2. Node Crashes and WAL Recovery

Each ARTS node maintains an append-only Write-Ahead Log (WAL) of epoch results. At the end of epoch e, the node writes:

- the settlement vector V_e,
- a BLAKE3 Merkle root over all obligations O_{ij},
- metadata including epoch index, packet count, and timestamps.

WAL entries are written using a two-phase commit: a provisional record is appended, followed by a commit marker once the write is durable. On crash and restart, the node scans the WAL and discards any trailing entries that lack a commit marker. This ensures that partially written epochs are never replayed.

Settlement vectors are constructed to be idempotent: an RTGS instruction $(i \to j, a)$ can be safely reapplied without changing the final ledger state if the initial application succeeded. Replay after a crash therefore cannot over-credit or over-debit any institution, even if an RTGS node receives duplicate submissions.

8.3. Inconsistent State Within an Epoch

A node crash during epoch e may leave some citizen transactions unprocessed. Upon restart, the node resumes ingestion and begins a new epoch $e' > e$. Any transaction that was acknowledged by the upstream PSP but not included in T_e is either:

- resent by the PSP (at-least-once delivery), in which case it appears in $T_{e'}$, or
- detected as missing via the per-sender sequence number in `TxPacket`.

In the latter case, the node emits a warning and rejects the gap, rather than silently settling on incomplete state. This preserves value while making data-loss events explicit to operators.

8.4. Malicious Transaction Injection

ARTS assumes that transaction sources may attempt to inject fabricated or replayed packets. Each `TxPacket` carries:

- a per-sender monotonically increasing sequence number,
- a BLAKE3 integrity hash over all non-padding fields,
- an institution-level signature or MAC (in deployment) bound to the sending PSP or bank.

Packets that fail integrity or signature verification are discarded before entering the aggregation layer. Packets with out-of-order or duplicate sequence numbers are rejected and logged. This prevents a compromised PSP from inflating its net position via replay or fabrication without detection.

8.5. Adversarial Topologies

To evaluate robustness under adversarial transaction graphs, we construct synthetic obligation graphs where:

- a single institution attempts to route most flow through itself,
- a coalition of institutions creates dense cycles to stress cycle-cancellation,
- edge weights follow heavy-tailed distributions that concentrate value on a small set of edges.

Across these graphs, multilateral net-position settlement maintains the same value efficiency and transfer bounds as on realistic topologies, confirming that its guarantees are structural rather than dependent on benign graph structure. Bilateral cancellation, by contrast, exhibits increasingly poor value efficiency as power-law skew increases, reinforcing the bilateral ceiling discussed in Section 6.

9. Future Work

While the multilateral greedy algorithm achieves theoretical optimality under uniform transfer costs, several extensions warrant further investigation.

9.1. Constrained LP Settlement

As outlined in Section 5.6, the current LP formulation can be extended with liquidity and credit constraints that bound per-institution and per-counterparty exposure. Our preliminary experiments suggest that such constrained settlement may sacrifice 2–5% value efficiency but provides stronger guarantees on systemic risk containment. Future work will implement and benchmark a production-quality constrained solver, and evaluate heuristic projections of the greedy solution onto the constrained feasible region.

9.2. Adaptive Epoch Duration

The fixed 500ms epoch window represents a trade-off between netting efficiency and settlement latency. Adaptive epoch duration could dynamically adjust the window based on incoming transaction rate: shorter windows during low-volume periods for faster finality, longer windows during peak hours for higher netting efficiency. A reinforcement learning approach trained on historical transaction patterns could optimize this trade-off in production.

9.3. Multi-Path Settlement

Single-epoch settlement requires all net positions to resolve within one window. Multi-path settlement allows large transfers to be split across multiple epochs, reducing per-epoch liquidity requirements at the cost of slightly extended settlement latency. This approach mirrors payment channel networks and could significantly reduce intraday liquidity requirements for smaller institutions.

9.4. Hardware Optimization

The current implementation runs on a single CPU core. SIMD vectorization of the net-position computation could yield additional 2-3× speedup. Cache-aware memory layout reorganization of the obligation matrix may also improve performance on NUMA architectures. These optimizations become relevant when scaling beyond 20 institutions or when processing transaction rates exceeding 10M/s.

10. Conclusion

ARTS demonstrates a general principle: settlement complexity should be absorbed by the clearing layer, not imposed on the settlement layer. By shifting the netting computation from the RTGS to an overlay, ARTS compresses 1,000,000 citizen transactions into 19 RTGS settlement instructions—a 99.998% reduction in settlement load—while sustaining 7,000,000 packets per second on commodity hardware. The multilateral greedy algorithm achieves 33.3% value efficiency at 16× lower latency than bilateral cycle-cancellation, and LP verification confirms optimality under uniform transfer costs.

The bilateral ceiling analysis establishes a structural limitation of cycle-cancellation approaches on power-law transaction graphs. Even optimal cycle selection cannot match the value efficiency of net-position settlement. This finding is not specific to Nepal: any interbank topology with heavy-tailed obligation weights will exhibit the same efficiency gap.

ARTS operates as a clearing overlay that intercepts the existing NPI/RPS routing without modifying core banking systems or requiring NCHL cooperation. A minimum viable deployment requires two PSPs and one commercial bank participating in ARTS clearing; non-participating institutions continue to settle via DNS and are netted against participating institutions at the RTGS boundary. The architecture is designed for incremental adoption: each additional participant increases the netting window and reduces RTGS load.

We encourage the Nepal Rastra Bank to evaluate ARTS as a modernization pathway that preserves the deferred net settlement model while eliminating its fragmentation debt.

References

Ravindra K. Ahuja, Thomas L. Magnanti, and James B. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, 1993.

Nepal Banking Association. Nepal banking association annual report 2024-25, 2025.

Nepal Rastra Bank. Rtgs and payment system review, 2024.

Nepal Rastra Bank. Nepal rastra bank annual report 2024-25, 2025. Central Bank of Nepal.

BurntSushi. Criterion.rs: Statistics-driven benchmarking, 2024. URL `https://bheisler.github.io/criterion.rs/`.

ByteDance. Monoio: Rust async runtime for io-uring, 2024. URL `https://github.com/bytedance/monoio`.

Clearing House Payments Company. Chips settlement statistics 2023, 2023.

National Payments Corporation of India. Unified payments interface technical specifications, 2022.

Kantipur Post. Digital payment statistics fy 2024-25, 2025.

Robert Tarjan. Depth-first search and linear graph algorithms. *SIAM Journal on Computing*, 1(2):146–160, 1972.

HiGHS Team. Highs linear programming solver, 2024a. URL `https://github.com/ERGO-Code/HiGHS`.

OSD Team. Blake3: One function, full speed, 2024b. URL `https://github.com/BLAKE3-team/BLAKE3`.