Decentralised Exchange Protocols

Imosuke Takakuni[1]

10 April 2025

ABSTRACT

Cryptocurrencies were conceived to break away from traditional finance and the dominance of central institutions. Yet in practice, building viable, scalable, and fair marketplaces has proven harder than expected. This paper identifies three core limitations—throughput, leverage, and fairness—that continue to constrain decentralized trading. We outline how these have historically pushed crypto back toward familiar TradFi tools and propose a new protocol that aims to restore the original vision: permissionless, fast, and fair markets with no centralized choke points. Our approach leverages network latency and cryptographic fragmentation to protect against front-running and manipulation, allowing for real-time, censorship-resistant execution at scale.

From Dream to Deployment

The crypto dream began with visions of a golden era—one without banks, borders, or intermediaries. But that dream eventually came to in a cold sweat, tangled in the plumbing of traditional finance (TradFi). Institutions—often with far less oversight and experience than their TradFi equivalents—have led the industry through several self-inflicted winters. Exchange collapses and outright fraud are common enough to warrant their own textbook.

1. "Imosuke Takakuni" is a pseudonym. Contact: takakuni@tensorcash.org

Still, there's a logic to how we got here. If you hold an asset or a currency, odds are you either want to spend it—or at least know what it's worth. That means users want price discovery, liquidity, and, eventually, a way to buy groceries. And this is where the ideal meets the inconvenient: most crypto assets can't function in isolation. They need bridges to the real world.

Now we have the Binance debit card: a gateway from your crypto exchange account to your neighborhood checkout aisle. It works by settling in dollars over Visa channels. You don't own crypto, just an IOU in an offshore entity—converted, for a fee, into something the store understands. The irony writes itself. Today's Binance card could just as easily become yesterday's FTX card.

TABLE OF CONTENTS

I. Three Foundational Limitations 2
II. Introduction 2
III. Our Proposal: A Fair, Fast, and Fully Decentralized Exchange Protocol 3
IV. Speed of Light and Geographical Fairness – a Simplified Example 6
V. Market Microstructure and Incentives for Price Discovery 7
VI. Technical Overview 8
VII. Network Stack, Feasibility, and Network Throughput 9
VIII. Security and Attack Vectors 11

I. Three Foundational Limitations

Let's unpack the core limitations that brought us here:

1. **Transaction Throughput**
   Blockchains rely on decentralized consensus. This inherently takes time and imposes strict constraints on achievable transactions per second.
2. **Anonymous Leverage**
   Marketplaces require liquidity providers and arbitrageurs. These actors generally don't hold long-term positions—they borrow on margin. But extending or receiving credit in an anonymous environment is as oxymoronic as it sounds.
3. **Fairness Without Central Authority**
   Even if the first two are addressed, achieving a fair and efficient marketplace is much easier with a central exchange—ideally impartial—that manages order flow and enforces trading rules.

These are non-trivial problems. While some progress has been made, it's no surprise that the crypto world has borrowed liberally from TradFi to keep things moving.

II. Introduction

We suggest a framework to relax these constraints and bring the crypto dream closer to reality.

The three key issues—transaction throughput, non-recourse leverage, and market fairness—are discussed in increasing order of difficulty. In this piece, we focus mainly on decentralized marketplace design, the hardest and least-satisfactorily addressed challenge. The other two have largely known solutions; what remains is adoption, often hindered by the lack of a robust marketplace supporting it.

1. **Throughput:**

The issue of transaction throughput is well-known. The Bitcoin blockchain can handle approximately 7-10 transactions per second (TPS), with finality taking up to 60 minutes. In contrast, Visa averages around 1,700 TPS, with a theoretical capacity of up to 65,000 TPS. Centralized exchanges like Binance routinely process over thousands orders per second, with matching engine latencies measured in microseconds, underscoring the massive performance gap between decentralized and traditional systems.

To improve this, newer protocols promise faster speeds—but usually at the cost of some security guarantees. Most of these are structured as Layer 2 solutions: think of Layer 1 as your vault, and Layer 2 as your checking account—less secure[2], but much more usable for day-to-day needs. Examples like the Lightning Network (for micropayments) are already in use and inching toward broader adoption.

2. **Financing:**

Let's be clear: "on-chain" financing cannot be reconciled with traditional finance. The counterparty is unknown, and legal recourse is unrealistic. That said, it does offer one major advantage: no settlement or custodian risk. Collateral and exchanges are embedded into the protocol at the moment of transaction.

The practical solution here is simple: overcollateralized lending—possibly with collateral-price margin requirements. This converts "jump-to-default" risk into plain asset price risk. Protocols like Aave already implement this.

But adoption is hampered by complexity. It's orders of magnitude easier for a retail trader to wire some dollars from a bank account to a centralized exchange and trade with 10x margin than to juggle stablecoins and recursive leverage strategies. Institutional investors face similar frictions—especially when they're pampered with prime brokerage support.

3. **The Hardest Problem: Fair Liquidity Formation:**

Liquidity emergence is where decentralization runs into the buzzsaw of reality. In TradFi, exchanges are held accountable by regulation and policy: they must not take the other side of a client trade, leak order information, or reorder transactions arbitrarily. Spoofers can be banned. Flooders can be prosecuted.

But in a decentralized environment, where entry is permissionless and actors are anonymous, none of these guarantees exist. There is no one to trust, and no way to ensure the same entity isn't playing both referee and participant.

Yes, we can enforce transparency. Yes, we can codify matching rules. But the real issue is sequencing. If a bad actor can rearrange or selectively exclude transactions, they can manipulate markets at will—buying before a price spike or cancelling just in time to dodge a wave.

This kind of manipulation isn't hypothetical. It's rational. It's expected. And it has a name: MEV (Maximum Extractable Value). Every decentralized exchange today suffers from it. Some try to mitigate it with batch auctions or encrypted transactions, but these add latency, complexity, and uncertainty. And who wants to trade at an uncertain price when their neighbor can execute faster and cleaner on a centralized exchange?

III. Our Proposal: A Fair, Fast, and Fully Decentralized Exchange Protocol

We propose a Byzantine Fault Tolerant (BFT) protocol for fully decentralized and permissionless trading, capable of scaling to thousands of transactions per second. The system works with millisecond-precision ordering over public networks.

2. Some recent Blockchains have moved to Layer 1 consensus protocols that sacrifice robustness for throughput, e.g. by using a fixed set of rotating validators, replacing Proof of Work with Proof of History.

Core Concept: We use global network latencies and cryptographic fragmentation as protection against premature reveal and transaction reordering.

How it works:

- Each user establishes a direct connections to a rotating, unpredictable subset of 10 out of 100+ geographically dispersed validators.
- Each transaction is encrypted, the decryption key is split into 10 different fragments sent to each validator, and at least 6 fragments are needed for decryption.
- Validators can't see the full transaction initially, but are required to timestamp, sequence, and sign the fragments.
- After a short delay, each validator is required to “gossips” its message and timestamp to the network.

Crucially, the transaction becomes visible to the broader network only after enough fragments have been reassembled via gossip propagation. Validators, not knowing the full message content, have no incentive to delay or reorder it. Withholding isn’t a Nash equilibrium—honest validators will already be circulating the data.

Once enough transactions are in play:

- A randomly chosen validator normalizes for clock drift and network noise,
- Generates a fair ordering,
- Proposes a block,
- And achieves finality via supermajority confirmation.

Meanwhile, the gossip network acts as a real-time market data feed—letting traders observe flow even before it’s formally committed to the blockchain. Settlement then occurs on a Layer 2 system, with eventual anchoring to the main chain.

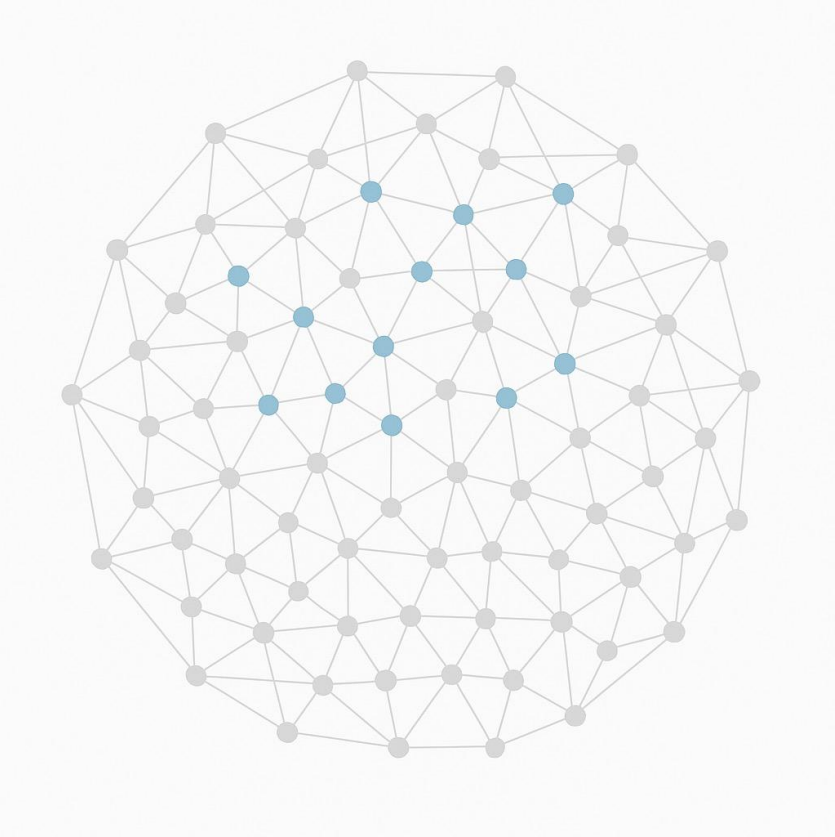

Fig.1: Trader initiates a transaction broadcasting to randomized validators

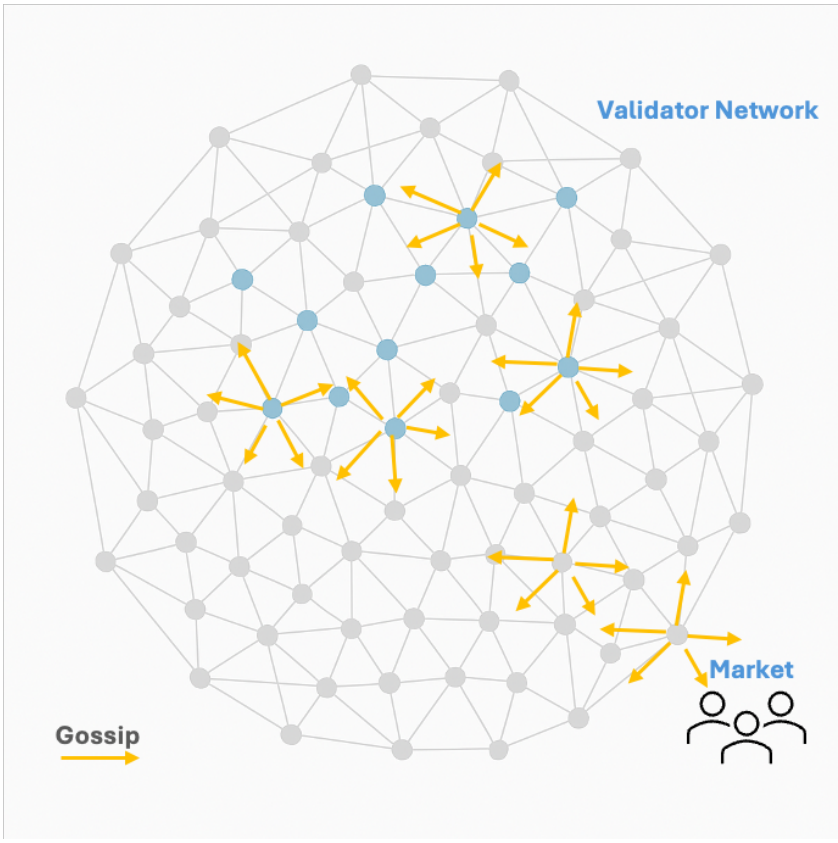

Fig 2: Gossip propagates, transaction is reassembled and analyzed by the market

This is a Sybil-resistant, Byzantine Fault Tolerant, and front-running-proof trading protocol, allowing:

- Fair ordering of transactions,
- Real-time visibility for informed traders,
- Instantaneous execution without batch auctions, and
- A robust market environment where no single actor can cheat the sequence.

This system could serve not just as a “follower” for centralized exchanges price action, but potentially as a “leader”—setting prices with confidence and integrity. The system is entirely decentralised with no central actor able to influence a fair matching process, individual traders can listen to the gossip and inform their decisions. An off-chain ecosystem can develop for transaction visualisation, historical data and order entry to support the permissionless protocol.

IV. Speed of Light and Geographical Fairness – a Simplified Example

Consider a global trading setup with two synchronized validators—one in Germany, one in Australia. A trader in Dubai and another in Argentina attempt to take liquidity at the same time. A market maker (MM), co-located at both validator sites, provides this liquidity and maintains a low-latency microwave link between the two hubs.

The question is: if both traders act simultaneously, who gets the fill? And can the MM use its infrastructure to avoid honoring the quote once it sees activity at one site?

Traders send orders over public internet routes, while the MM uses a dedicated microwave link. The table below gives realistic one-way latency estimates:

Route	Connection Type	Latency (ms)
Dubai → Germany	Internet	~70
Dubai → Australia	Internet	~150
Argentina → Germany	Internet	~160
Argentina → Australia	Internet	~180
Germany ↔ Australia	Microwave	~53

To evaluate fairness, we consider the average arrival time of each trader's order at both validators from the equal sending timestamp. We assume both nodes are honest and include transactions, and timestamping is based on local receipt.

- Dubai: (70 + 150) / 2 = 110 ms
- Argentina: (160 + 180) / 2 = 170 ms

Dubai’s order, on average, arrives 60 ms earlier, giving it a clear edge in obtaining the liquidity. This demonstrates how geographic positioning and network topology influence fairness, even in synchronized distributed systems.

Can the Market Maker Front-Run or Cancel? The MM might try to detect a fill at one hub and cancel liquidity but this is constrained by the ~53 ms microwave latency between hubs, this is needed to reconstruct the two pieces of the transaction[3]. By the time a cancel signal can travel and be acted upon, the second validator has already received and timestamped the trader’s order.

In short, while the MM benefits from faster inter-hub coordination, it cannot outpace the propagation of external orders to both validators. Speed-of-light latency imposes a hard lower bound on reaction time, preventing strategic cancellation or front-running. Using average arrival time offers a fair metric in dual-validator systems. In this setup, Dubai has the advantage. But even a globally optimized market maker cannot circumvent validator ordering—physical limits on signal propagation uphold the integrity of transaction sequencing.

V. Market Microstructure and Incentives for Price Discovery

One of the most critical aspects of any market is how its microstructure incentivizes—or fails to incentivize—price discovery. Latency of information dissemination is typically the dominant factor. It shapes how participants respond to new information and fundamentally impacts the incentives faced by both liquidity providers and end users.

Take the case of U.S. cash equities. These markets have evolved into a competitive yet fragmented ecosystem of geographically dispersed exchanges. The microstructure has naturally encouraged the development of costly inter-exchange connectivity. Informed participants—usually market makers—invest billions in infrastructure to gain a competitive edge, allowing them to digest and act on information from any exchange as early as possible, whether by removing or taking liquidity. As a result, even end users have been pushed to invest heavily in optimizing their order routing—often synchronously—just to remain competitive[4].

Futures markets, despite being single-venue, are not immune to latency challenges. In some venues, information dissemination is rigorously synchronized across participants. This creates a “winner-takes-most” dynamic, where the race isn’t just to acquire the information, but to be the first to process and act on it[5]. Other venues take a less structured approach—information spreads more organically, influenced by the specifics of the matching engine or the quirks of the network stack. These asymmetries offer valuable optimization opportunities, particularly for those with insider knowledge or privileged relationships.

If these are necessary evils or just suboptimal designs remains an open question and, whether this constant cat-and-mouse game truly enhances market efficiency is a matter of debate. Many investment banks and dealers would argue otherwise, suggesting instead to transact over-the-counter, charging a spread commensurate with the perceived sophistication of their counterparty.

3. The earliest reaction time to the Dubai transaction in Germany would be 150+53=203ms, i.e. the time to pick up the message in Australia and send it to Germany for reconstruction. In Australia one could receive the information at 70+53=123ms, but would still need to wait for the second fragment at 150. Even assuming costless processing time the earliest response would be (203+150)/2 = 176.5ms, i.e. 66ms later than the Dubai trader. Even if we were to add a third friendly validator available to the MM in Australia, the honest timestamp would still be 150ms, failing a simple averaging aggregation and also a “first decodable receipt” paradigm in real world situation.
4. An inexperienced trader wants to buy 10,000 shares of Apple Inc., which is actively traded on both NASDAQ and BATS. They first send a marketable buy order for 6,000 shares to NASDAQ, where it executes at $172.40, and then—just 150 milliseconds later—submit a second order for the remaining 4,000 shares to BATS. However, a well-connected market maker monitoring both venues anticipates the trader’s intent and reacts faster, sweeping the BATS liquidity at $172.41 before the second order arrives. The trader’s delayed order ends up filling at $172.45 due to diminished liquidity, costing an extra $160 and highlighting how latency and fragmented execution can penalize slower participants.
5. Ttraders use the *corrupted packet strategy* to react faster than full trade decoding allows. As soon as a trade packet is partially read, cancels or aggressive orders are formed and launched immediately via hardware acceleration—*every time*, regardless of certainty. If later decoding shows the signal was a false alarm, the order is deliberately corrupted in-flight—via a bad checksum or malformed header—so it’s rejected by the exchange. If the signal proves valid, the packet goes through untouched. The edge comes from always preparing action, but only allowing it to complete when warranted.

At its core, the common denominator remains: the speed and determinism with which market participants disseminate, digest, and respond to information. Well-resourced players will continue to optimize for these factors until the marginal dollar no longer justifies further investment. With this context in mind, it is worth examining how a proposed decentralized market might differ—what it offers, what optimizations it enables, and how it might affect price discovery.

For the purpose of discussion, let's assume a well-functioning network equilibrium, where clocks are synchronized and validator participation is honest. From a latency-sensitive trader's perspective, achieving the earliest timestamp and transaction priority would require broadcasting to at least six of ten validators as quickly as possible. If the validator set were static and their topology known in advance, a high-budget strategy could involve colocation next to each validator, with private links connecting them faster than public networks. However, the permissionless and rotating nature of validator sets (changing per user every block) makes such an approach both costly and fragile, highly susceptible to randomness.

By contrast, a naive, latency-insensitive price taker needn't worry about network topology or synchronization. Their transaction is protected from front-running by the fundamental constraint of the speed of light. This is a stark contrast to U.S. equities example, where aggressive orders must be routed synchronously across multiple venues to avoid signaling intent too early—thus preventing liquidity from vanishing before the full order is executed. The decentralized setup is materially more efficient for such actors.

Market data dissemination follows a gossip-based model. The timing with which one receives market information depends on the network topology and the randomness of validator assignment to the original aggressor. For instance, hearing about a significant market event depends on where one's gossiping endpoints sit relative to the randomly chosen validators broadcasting the event on behalf of the initiating user. Owning a robust, globally connected network certainly helps—but it offers no deterministic guarantee of first access to new data. For the broader participant base, fairness is found in randomness.

A final but important consideration is anonymity. In any decentralized market, the ledger must remain publicly auditable. As a result, a trader's activity, positioning, and historical footprint are uniquely traceable. This is an unavoidable consequence of transparency. However, mitigations exist—account rotation and transaction masking strategies can reduce traceability, though they introduce complexity in optimizing order placement.

The proposed decentralized market structure creates a novel and arguably healthier balance of incentives:

- It **rewards intelligent aggregation and analysis** of price data without necessitating an arms race in infrastructure spending.
- It **discourages excessive investment in colocation and proprietary networks**, lowering barriers to entry.
- It **protects liquidity takers** from the need to engage in complex and costly execution strategies.

The result is a system where competition is still fierce—but arguably more aligned with price discovery than with privileged access. That, in itself, is a compelling evolution.

VI. Technical Overview

Protocol runs as a side-chain to a Layer 1 "main net":

- Clients commit funds via a self custodial locking. These funds are controlled exclusively by the client and become visible on Layer 2 protocol only after six Layer 1 confirmations.
- The side chain periodically commits back to Layer 1 using compact zero-knowledge proofs summarizing all transactions, ensuring auditability without disclosing transaction details.

- Finality on Layer 1 is intentionally deferred, allowing the system to absorb and correct temporary inconsistencies or inclusion lags without breaking consensus.

Validator access is gated through a dual proof-of-work mechanism. Validators must solve a calibrated challenge to be admitted and maintain their status for K blocks. The difficulty is tuned to ensure that the validator set remains within 100–200 participants, anchoring participation in real-world resource expenditure and providing Sybil resistance. Each validator is uniquely identified within every block.

The network operates over the public internet via a peer-to-peer protocol. Validators establish secure QUIC/TCP connections for low-latency point-to-point order routing and participate in gossip-based message propagation through a pub-sub system (gossiplib) to ensure robustness. Transactions are signed by users from unique, possibly obfuscated addresses, and encrypted using a 6-of-10 Shamir secret sharing scheme. At least six validators must receive and share fragments before any transaction is reconstructible.

Each user is assigned a random set of 10 validators for each block, based on the hash of their address and the previous block hash. Validators who receive fragments must timestamp them, attach a monotonic sequence ID, and gossip them with signatures after a 25ms intentional delay to allow batching and integrity tracking.

Block production is triggered by either the arrival of 11,000 transactions or 60 seconds from the last block timestamp. Transactions younger than 1 second are excluded to allow for fair propagation. A validator is deterministically chosen to propose the block and collect clearing or exchange fees. If the chosen validator fails to act within the timeout, a backup is appointed.

During block assembly:

- Each transaction is validated for structural correctness, provenance (including proper routing), and compliance with Shamir reconstruction requirements.
- Gossip integrity is verified via signatures, sequence ordering, and timestamps.
- Violating users incur additional transaction fees, while non-compliant validators are ejected from the set.

A lightweight regression is run over valid messages to account for latency discrepancies between validators and neutralize clock drift, eliminating the need for global time sync. Outliers and clearly manipulated data (e.g., negative latency) are excluded.

Once validated, transactions are ordered deterministically, and matching logic is executed to allocate orders. A compact zero-knowledge proof is produced and submitted to the validator set. If approved by a supermajority, the block is accepted. If rejected, the process rotates to the next eligible proposer.

Although finality may be delayed on both Layer 2 and Layer 1, the entire block assembly process is fully deterministic and can be reproduced offline by any validator or market participant. The system remains statistically robust under mild assumptions about gossip propagation, without requiring synchrony.

VII. Network Stack, Feasibility, and Network Throughput

This section outlines the technical feasibility, protocol recommendations, and resilience considerations for a blockchain system where 100,000 users interact directly over QUIC/TCP with a random subset of 10 out of 150 validators, and where all 100,150 nodes (users + validators) participate in gossip listening, while only validators are allowed to post gossip

1. **Network stack and protocols**

The preferred transport protocol is QUIC due to its efficient, connectionless design and built-in TLS 1.3 encryption. It supports multiplexed streams and mitigates connection churn when users interact with multiple validators. TCP remains a viable fallback but requires pooled, persistent connections with strict connection reuse. In both cases, encryption is mandatory to ensure confidentiality and authentication. For the application layer, Gossipsub provides a scalable, deduplicated, and mesh-based pub-sub model. Its adoption in existing large-scale blockchain systems demonstrates its viability under high-throughput conditions.

2. **Transaction flow and throughput**

Each transaction originates from a user and is sent directly to a random subset of 10 validators. At 1 transaction per user per second, this results in 100,000 transactions per second overall. Each validator receives approximately 6,667 transactions per second on average. Every received transaction is then gossiped by each receiving validator, producing roughly 1,000,000 signed gossip messages per second network-wide. Assuming full transaction content is included in the gossip payload, and with an average transaction size of 500 bytes including metadata (transaction data, validator signature, timestamp, and sequence ID), this results in a total gossip bandwidth of approximately 500 MB/s. Distributed evenly across all participants, each node would need to process around 5 MB/s of gossip traffic. Per-node bandwidth requirements are within the capabilities of modern networking equipment and cloud infrastructure.

3. **Network balancing and efficiency**

Users maintain connection pools to a rotating validator subset to reduce connection overhead. Validators implement non-blocking receive queues to accommodate incoming transactions. Gossip emissions are batched every 50 to 100 milliseconds, reducing redundant messages and improving network efficiency. This balancing ensures validators can operate under heavy input load without degradation in propagation speed.

4. **Flooding, abuse resistance, and adversarial resilience**

Transaction flooding is mitigated through per-IP rate limiting and by enforcing that transactions are only processed when acknowledged by authorized validators. Gossip flooding is controlled by Gossipsub's internal scoring and deduplication. Man-in-the-middle attacks are mitigated through TLS encryption and validator public key pinning. Sybil resistance is enforced at the layer-2 blockchain level: both validators and users must undergo admission procedures that include economic or resource-based commitments, such as staking or prior on-chain identity registration. Replay protection is handled by timestamping and unique nonces within each transaction receipt.

5. **Feasibility and scale**

The system design supports direct interaction at high scale. Validator hardware requirements remain reasonable due to load balancing and batched gossip. The gossip layer supports millions of messages per second without centralized coordination, and passive listeners can operate in low-resource modes. The increased bandwidth from full transaction propagation is manageable with proper infrastructure and scalable networking. The architecture ensures secure origination, rapid propagation, and a low attack surface, while maintaining strong decentralization properties.

This network model offers robust scalability, efficiency under stress, and cryptographic accountability across all propagation and validation phases. Its direct transaction origination model and gossip-based redundancy allow for trust-minimized operation even at global user scale.

VIII. Security and Attack Vectors

We go through more formally a number of possible attack vectors from a subset of the validators nodes, assuming a minority are Byzantine and colluding or impersonating traders for manipulative purposes.

Attack Type	Description	Mitigation
Colocation and Timestamp Manipulation	Traders colocated near validators try to gain speed advantage and manipulate timestamps.	Honest validators timestamp upon physical arrival. Signed attestations from multiple validators and regression-based ordering detect inconsistencies. Tamper-evident signatures and gossip comparison reveal manipulation attempts.
Forward-Dated Timestamps and Reordering	Validators delay fragments to push transactions later in the order.	Random assignment of fragments and requirement for multiple honest recipients lead to outlier rejection. Forward-dating fragments of withheld adverse transactions offers no advantage in the protocol aggregation and only delays the attacker's own information gathering.
Partial Adversarial Control and Impersonation	An attacker tries to control part of the validator set to impersonate users or submit fake transactions.	Validator selection is random and per-user/transaction. Attackers must control the exact subset and evade gossip-based detection. Timestamp tracking, sequence validation, and convergence make impersonation or backdating highly unlikely.
Fragment Withholding	A user submits only part of a transaction to test market behavior or stall execution.	Transactions need a full set of fragments to be valid. Incomplete ones are dropped and may incur penalties like forfeited deposits. Timestamp is based on completed fragments last receipt, so withholding provides no benefit.
Validator Withholding and Selective Gossip	A validator refuses to share a transaction fragment or delays gossip.	As long as the minimum threshold (e.g., 6 of 10 fragments) is met, the transaction proceeds. Non-cooperative validators are penalized or excluded. The system tolerates partial failure while maintaining performance and integrity.
Sybil Registrations and Identity Flooding	Attackers create many fake validators or users to overwhelm the system.	Entry requires deposits or previous on-chain activity. Lightweight proof-of-work or staking prevents mass registration. Transport-layer rules also prevent excess connections or messaging rates.
Eclipse and Network Isolation	Attempting to isolate a group of honest validators from the broader network.	Validators regularly update and randomize peer connections. Broad and diverse connections make total isolation statistically unlikely unless most of the network is compromised.
Equivocation and Conflicting Messages	Validators send inconsistent information (e.g., different timestamps for the same transaction).	Each message is signed and sequenced. Conflicts are exposed during block proposal. Offenders can be slashed or removed. Tampering leaves clear, detectable evidence.
Denial of Service and Spam	Flooding the network with fake transactions or connections to slow it down or disable it.	Validators and users must meet entry requirements (e.g., stake, registration). Network protocols enforce rate limits. TCP/QUIC congestion control absorbs excess load. These filters block attackers without adding friction for legitimate users.
Identity Based Front Running	Collusive Validator front run based on originator identity.	Rotating set makes this difficult to enforce without super-majority ownership. Gossip based front-running without transaction content knowledge is theoretically possible but easily preventable by user id rotation
Censorship by Block Proposers	A block proposer ignores transactions they dislike.	The network rotates block proposers and requires approval by the wider validator set. If any validator detects that a well-gossiped transaction is missing, they can reject the block or trigger a re-election.