Delay-Optimal Transaction Order Fairness

Zhuo Cai, Amir K Goharshady

Delay-Optimal Transaction Order Fairness

ZHUO CAI, The Hong Kong University of Science and Technology, Hong Kong SAR, China
AMIR GOHARSHADY, University of Oxford, United Kingdom

Transaction order fairness is emerging as a critical defense against Miner Extractable Value (MEV) and censorship, problems that traditional consensus protocols exacerbate by seeking only standard consistency and liveness. This work stands among the few to rigorously address these issues through a strong, fine-grained fairness definition. Existing schemes settle for (1) coarse "block-order fairness" [Kelkar et al. 2023, 2020], which still permits intra-block manipulation, (2) "bounded unfairness" [Kiayias et al. 2024], which incurs high computational overhead, or (3) median timestamp ordering [Kursawe 2020; Zhang et al. 2020], which is susceptible to manipulation. We propose an alternative driven by *delay optimality*.

We revisit *Approximate-Order-Fairness* (AOF), which mandates that if a fraction φ of nodes receive transaction tx at least time θ before tx′, then tx must be ordered before tx′. Although prior work dismissed AOF using impossibility reductions to the ideal but impossible *receive-order-fairness* [Kelkar et al. 2020], we observe that the reduction relies on an overly restrictive assumption of logical-round-based clock functionality. In practice, especially in modern blockchain applications, we can use global physical time to circumvent the reduction. We identify a robust *feasible region* where Condorcet cycles are provably impossible: in a permissioned system of n nodes of which f are Byzantine, under a fully synchronous network with delay Δ_{sync}, for any $k \in \mathbb{Z}^+$, (φ, θ)-AOF is feasible for any $\varphi > 1 - \frac{n-f}{nk}$ and $\theta > \Delta_{\mathsf{sync}}/k$. We design protocols that achieve AOF without relying on a fixed delay parameter θ. Instead, they dynamically compute the optimal θ for each block based on the observed transaction profile, ensuring the strongest possible fairness guarantee under current network conditions. This feature also makes the protocols robust against limited asynchrony. We also provide a comprehensive comparison of AOF with existing definitions, demonstrating that AOF offers a compelling fairness alternative while maintaining practical efficiency.

CCS Concepts: • **Theory of computation → Distributed algorithms**; • **Security and privacy → Distributed systems security**.

Additional Key Words and Phrases: Blockchain, Consensus, Transaction Ordering Fairness

Full-Version Note

This manuscript is the full version of the brief announcement to appear at PODC 2026 [Cai and Goharshady 2026].

1 Introduction

Blockchain systems have evolved from simple payment networks to complex decentralized computing platforms. The central problem of blockchain consensus has traditionally focused on two properties: *consistency* (all honest nodes agree on the same log) and *liveness* (valid transactions are eventually included). This problem has a rich history in distributed computing, with protocols ranging from classical PBFT [Castro and Liskov 1999] to modern blockchain-optimized variants such as HotStuff [Yin et al. 2019], Algorand [Gilad et al. 2017], and recent DAG-based advances [Danezis et al. 2022; Guo et al. 2020; Keidar et al. 2021; Miller et al. 2016; Spiegelman et al. 2022]. However, neither property constrains the *actual order* of transactions within the log. This gap has been exploited by adversarial actors—often miners or validators—to extract value from users (Miner Extractable Value or MEV) through manipulation tactics such as front-running, back-running, and sandwich attacks [Daian et al. 2019], or to engage in censorship.

Authors' Contact Information: Zhuo Cai, zcaiam@connect.ust.hk, The Hong Kong University of Science and Technology, Hong Kong, Hong Kong SAR, China; Amir Goharshady, amir.goharshady@cs.ox.ac.uk, University of Oxford, Oxford, United Kingdom.

A variety of mechanisms have been proposed to mitigate these negative externalities. Proposer-Builder Separation (PBS) [Heimbach et al. 2023] decouples block construction from block proposal to reduce centralization risks, but effectively preserves MEV extraction by shifting it to specialized builders. To combat censorship, inclusion lists [Fox et al. 2024] (or CR-lists) force proposers to include specific transactions, yet these often require multiple slots to be effective, leaving short-term censorship vectors open. Cryptographic approaches such as threshold encryption (commit-reveal schemes) [Choudhuri et al. 2024] and economic mechanisms such as MEV burning [Mazorra et al. 2022] introduce significant latency and protocol complexity. Despite these efforts, these solutions often function as patches rather than root-cause fixes.

As a fundamental alternative, the property of *transaction order fairness* has emerged as a critical third requirement. Intuitively, fairness requires that the ledger's order reflects the order in which transactions were received by the network. Achieving this in a distributed system is non-trivial due to network latency; nodes receive transactions at different times, and malicious nodes can manipulate their reported timestamps.

The formal study of this problem was initiated by Kelkar et al. [Kelkar et al. 2020], who defined *Receive-Order-Fairness* by drawing parallels to social choice theory [Arrow 1951]. This definition requires that if a sufficient fraction of nodes receive tx before tx', then tx must be ordered before tx'. They proved that strictly satisfying this property is impossible due to the *Condorcet paradox* [de Caritat Condorcet 1785], where pairwise local orderings form a global cycle (e.g., $A \prec B \prec C \prec A$). To circumvent this, they proposed *Block-Order-Fairness* and the Aequitas protocol. They later proposed an improved implementation called Themis, which reduces communication cost and guarantees standard liveness for the same fairness notion (referred to as *Batch-Order-Fairness* in Themis) [Kelkar et al. 2023]. This relaxation allows transactions involved in a paradox to be batched into the same block. Other works have explored different relaxations: Pompē [Zhang et al. 2020] orders transactions based on aggregated timestamps but assumes weaker adversarial conditions; Blind Order Fairness [Choudhuri et al. 2022] aims to hide transaction contents to prevent value extraction; and recent optimizations such as those by Mu et al. [Mu et al. 2024] focus on efficiency.

Another approach, *Timed-Order-Fairness*, used by the Wendy protocol [Kursawe 2020], imposes an ordering constraint only when there is a strict time separation between transactions—specifically, when all honest nodes receive tx before any honest node receives tx'. While this definition avoids cycles, it is relatively weak because it leaves valid ordering constraints undefined for the vast majority of transactions whose propagation windows overlap. Ramseyer and Goel [Ramseyer and Goel 2024] further analyzed the fundamental latency trade-offs inherent in such approaches.

Most recently, Kiayias et al. [Kiayias et al. 2024] introduced *Bounded Unfairness*, quantifying fairness by the number of positions a transaction can be displaced relative to its "fair" position. They relate finding the optimal ordering to the *Directed Bandwidth* problem. While this offers a fine-grained metric, finding the optimal bound B is NP-hard. Their protocol Taxis offers approximations, but the computational complexity remains high for the optimal calibration (exponential in the size of the cycle components). Furthermore, a position-based metric does not directly capture the nature of front-running attacks: displacing a transaction by even a single position (e.g., ordering an attack transaction immediately before a victim transaction) is often sufficient for value extraction. Thus, bounding the displacement to B positions does not necessarily prevent the attack.

1.1 Our Approach: Approximate-Order-Fairness

In this work, we revisit and rigorously analyze *Approximate-Order-Fairness* (AOF) [Kelkar et al. 2020]. While this property was previously considered in the literature, it was largely dismissed as impractical or strictly weaker than Block-Order-Fairness due to impossibility reductions. We challenge this perspective by demonstrating that a significant, practical "Feasible Region" exists

where AOF provides guarantees that are more practically relevant than both Block-Order and Timed-Order fairness for preventing ordering attacks.

Current definitions suffer from a fundamental blindness: they rely on binary voting ("Did A arrive before B?") without considering the *magnitude* of the time difference. This loss of information allows a rushing adversary to manipulate the ordering of even widely separated transactions by constructing artificial Condorcet cycles (see Section 3).

Unlike Aequitas, which batches paradoxical transactions, or Wendy, which requires strict separation, AOF introduces a parameterized delay θ and minimum support threshold φ. Also unlike Taxis, which minimizes positional displacement, AOF enforces a strict pairwise ordering constraint $\mathsf{tx} \prec \mathsf{tx}'$ whenever the fairness condition is met. Combined with θ, this effectively restricts the attacker's available window for manipulation.

Our definition requires that if a φ fraction of all nodes receive tx at least θ time before tx', then tx precedes tx' in the final ledger. This approach is *delay-optimal*: it seeks to identify the minimum time gap θ required to guarantee a consistent global ordering without Condorcet cycles.

Novelty: Delay vs. Latency. It is crucial to distinguish our fairness delay θ from transaction confirmation latency. While latency measures the time until a transaction is finalized on-chain, θ serves as a filter to distinguish statistically significant orderings from network jitter.

- When θ approaches 0, our definition converges exactly to the original Receive-Order-Fairness [Kelkar et al. 2020].
- Crucially, θ is distinct from the network synchrony bound Δ_{sync}. The trade-off between the voting threshold φ and fairness delay θ is non-trivial: by increasing φ (requiring stronger consensus) and assuming a higher honest ratio h, we can achieve fairness for arbitrarily small $\theta \ll \Delta_{\mathsf{sync}}$. This allows us to enforce fine-grained ordering even for transactions arriving close together, far below the standard synchrony window. The protocol designer has the flexibility to choose (θ, φ) based on the desired security trade-off.

Physical Time vs. Logical Impossibility. Our results are enabled by a shift to a *Timed-Synchronous Execution Model* appropriate for modern blockchains. This setting is characterized by *fast computation and slow communication*: nodes operate at GHz frequencies with high-resolution local timers, while network propagation (Δ_{sync}) takes milliseconds to seconds. Prior impossibility results (specifically Theorem 4.6 in [Kelkar et al. 2020]) rely on a "padding" argument where an adversary inserts empty logical rounds to stretch the timeline, arguing that approximate fairness implies strict fairness on a "faster" network. This reduction holds in round-oblivious models but fails in our time-aware model. An adversary can insert logical rounds, but cannot pause the physical clocks of honest nodes. A node can distinguish between a transaction sequence arriving with large physical gaps (satisfying θ) and a sequence arriving in a "padded" but physically compressed window. This observational power breaks the reduction, validating AOF as a strictly weaker and achievable property in the physical world.

Case Study: Breaking Fairness Cycles. The practical power of AOF is enabling strict ordering where others fail. Consider a 3-transaction cycle (detailed in Section 6) where network jitter causes pairwise majorities to perceive $\mathsf{tx}_A \prec \mathsf{tx}_B \prec \mathsf{tx}_C \prec \mathsf{tx}_A$. Existing Block-Order schemes would batch all three into one unordered block. Bounded-Unfairness schemes might force an arbitrary linearization (e.g., $\mathsf{tx}_A, \mathsf{tx}_C, \mathsf{tx}_B$) that violates the strong signal of $\mathsf{tx}_A \prec \mathsf{tx}_B$. In contrast, AOF examines the time gaps. It identifies that the "return edge" $\mathsf{tx}_C \prec \mathsf{tx}_A$ relies on a gap smaller than θ (indistinguishable from jitter) while the other edges are strong. AOF filters out this noise, breaking the cycle and producing the correct strict linear order $\mathsf{tx}_A \prec \mathsf{tx}_B \prec \mathsf{tx}_C$.

1.2 Our Contributions

We focus on the standard synchronous network setting with dissemination delay Δ_{sync} and up to $f < n/3$ Byzantine faults. Our specific contributions are:

(1) **The Timed-Cycle Impossibility (Theorem 3.1):** We present a fundamental result showing that existing fairness definitions (Block-Order, Bounded-Unfairness) cannot guarantee the ordering of tx_A before tx_B even if *every* honest node receives tx_A minutes before tx_B. We prove that an adversary can always construct a chain of "chaff" transactions to form a Condorcet cycle, forcing the protocol to batch or reorder tx_A and tx_B. This highlights the necessity of incorporating time magnitude into the fairness definition. Furthermore, we expose the associated *Unbounded Dependency Problem*, where such cycles can extend indefinitely into the future, forcing protocols to choose between indefinite waiting or risking fairness violations. AOF resolves this dilemma via bounded temporal locality.

(2) **Feasibility Characterization of AOF:** We formalize and analyze Approximate-Order-Fairness (AOF). We refute the intuition that this property is universally impossible or strictly weaker than batching schemes. By explicitly incorporating time delay θ, AOF captures the physical constraints of the network that prior reductions ignored. We illustrate that it provides more precise ordering guarantees than Block-Order-Fairness in valid parameter regimes.

(3) **Feasibility and Impossibility Results:** We provide a rigorous analysis of the parameter space (θ, φ). We identify the *infeasible region* where Condorcet cycles can still form (e.g., when $\theta \leq \Delta_{\text{sync}}/k$ for certain φ), and the *feasible region* where AOF is strictly achievable. This establishes the fundamental limits of ordering fairness in synchronous networks.

(4) **Delay-Optimality and Advantage:** We show that our scheme achieves optimal delay for consistent ordering.
   - Compared to **Block-Order-Fairness**, AOF prevents the "cycle attack" (Contribution 1) by filtering out weak edges.
   - Compared to **Bounded Unfairness**, AOF targets time directly—the resource attackers need. In DeFi contexts such as sandwich attacks, position bounds are often insufficient: being placed five slots after the attacker can be just as harmful as being placed one slot after. AOF guarantees strict precedence (tx_A *before* tx_B). Moreover, our protocols run in polynomial time and achieve an optimal delay that is at most Δ_{sync}, and can be even smaller in a Δ_{sync}-synchronous network.

(5) **Efficient Protocols:** We present two leader-based protocols that achieve AOF, using the well-understood HotStuff architecture to reach consensus on the set of transaction reports for each block. The resulting efficiency is similar to efficiency-optimized Themis [Kelkar et al. 2023]: theoretically $O(n)$ communication per transaction using SNARK-based aggregation, and practically $O(n^2)$ without SNARKs.
   - A **Synchronous Protocol** based on Sync HotStuff [Abraham et al. 2020]. We show that under full synchrony, a malicious leader cannot exclude honest reports (requiring a $f + 1$ quorum to validly reject a report, which is impossible against an honest sender), ensuring the full honest voting power is preserved for fairness. This setting demonstrates the peak capability of AOF.
   - A **Partially Synchronous Protocol** based on standard HotStuff [Yin et al. 2019]. This variant adapts to network conditions, albeit with a potentially reduced effective honest ratio due to leader influence. In both cases, the ordering is determined by a deterministic function of the accepted reports, preventing leader manipulation where fairness does not dictate a specific order.

Scope and Limitations. We address three primary model assumptions.

First, AOF assumes access to a roughly synchronized global clock. While finding a perfect global clock is theoretically difficult, modern distributed systems (especially permissioned or high-performance blockchain clusters) routinely achieve synchronization (via NTP or GPS) with precision far finer than the transaction latency Δ_{sync}. AOF requires only that clock drift be small relative to θ, a practical constraint.

Second, the concept of AOF relies on some level of network synchrony. However, our protocol design shows that AOF is robust to minor asynchrony. Importantly, AOF degrades gracefully under instability: (1) a small fraction of nodes suffering asynchrony are effectively modeled as Byzantine faults (tolerated up to f); (2) under large network shifts where the actual network delay exceeds Δ_{sync} for the majority of nodes, AOF might not enforce the ordering constraints specified for synchronous networks, but this is still acceptable for two reasons: (a) AOF can dynamically adjust θ based on observed network conditions, allowing it to maintain some fairness guarantees even when the actual delay is unbounded; (b) during severe network instability or large-scale partition, blockchain protocols usually face a trade-off between strict safety and prioritized liveness, leaving order fairness a less important consideration.

Third, we employ a unified network model. Unlike prior works that differentiate between client-to-node (external) and node-to-node (internal) delays [Kelkar et al. 2020], we model transaction dissemination as a single process governed by Δ_{sync}. By requiring honest nodes to immediately forward received transactions to their peers, we guarantee that if any honest node receives a transaction, all honest nodes receive it within Δ_{sync}, effectively collapsing the distinction into a single comprehensive bound.

2 Preliminaries and Model

Timed-Synchronous Execution Model. Unlike classical round-based models where time advances only through discrete steps, we model the system using a continuous global physical time $\tau \in \mathbb{R}_{\geq 0}$. Each node i maintains a local clock $C_i(\tau)$ that progresses at a rate close to physical time, with bounded drift ϵ. We assume a **Modern Blockchain Setting** characterized by "Fast Computation, Slow Communication".

(1) **Global Time τ:** The physical reference time.
(2) **Local Clocks C_i:** Nodes have access to high-resolution local timers. Modern CPUs operate at GHz frequencies, allowing timestamping precision orders of magnitude finer than network latency.
(3) **Network Latency Invariant:** We assume the maximum delay is Δ_{sync}. We assume that local processing and transmission times are negligible compared to Δ_{sync}, which is reasonable given the high computational throughput of modern nodes. Thus, the dominant factor in message delivery time is network latency, not local computation or queuing delays.
(4) **Observation:** A node i receiving a transaction tx at physical time τ records a timestamp $\mathsf{ts}_i(\mathsf{tx}) = C_i(\tau)$.

This distinction between logical rounds and physical time is crucial. In classical models, an adversary can simulate a slower network by inserting "empty rounds". In our model, "silence" takes physical time, which is observable by honest nodes' local clocks.

Notations. We denote the set of all nodes by $\mathcal{S}$ with $|\mathcal{S}| = n$, the set of honest nodes by $\mathcal{H}$ with $|\mathcal{H}| = hn$, and the set of adversarial nodes by $\mathcal{A}$. We consider two settings for the honest ratio h: in the synchronous setting, we assume $h > 1/2$ following Sync HotStuff [Abraham et al. 2020]; in the partially synchronous setting, we assume $h > 2/3$ following HotStuff [Yin et al. 2019]. We use Δ_{sync} to denote the maximum network delay for guaranteed dissemination among honest nodes.

θ denotes the time-delay parameter for Approximate-Order-Fairness, and φ denotes the voting threshold.

In the synchronous setting, we enforce an additional availability constraint: if a proposal omits a report from node i, the leader must provide a cryptographic proof of exclusion (either two equivocating signatures from i or signatures from $f + 1$ nodes attesting they did not receive i's report). In the partially synchronous setting, we do not require the leader to explain missing reports.

Synchrony Assumption and Transaction Dissemination. We assume a standard synchronous network model where any message sent by an honest node is received by all honest nodes within Δ_{sync}. Crucially, we do not distinguish between an "external network" (client-to-node) and an "internal network" (node-to-node). Instead, we model transaction dissemination as a unified process: we require that upon receiving a valid transaction tx, an honest node immediately forwards it to its peers (if not already received). This ensures that if at least one honest node receives tx at time t, all other honest nodes are guaranteed to receive tx by time $t + \Delta_{\text{sync}}$. Thus, we use Δ_{sync} as the single comprehensive bound for network latency.

Partial Synchrony. While our primary analysis focuses on the synchronous setting providing the strongest fairness guarantees, we also consider partially synchronous networks for our protocol design in Section 5. We adopt the standard model [Dwork et al. 1988]: there exists an unknown Global Stabilization Time (GST) and an unknown delay bound Δ. The network behaves asynchronously before GST (messages can be delayed arbitrarily) and synchronously after GST (messages between honest nodes are delivered within Δ).

Adversary Model. We assume a static Byzantine adversary that can corrupt up to $n(1 - h)$ nodes. The honest ratio h is strictly greater than $1/2$ in the synchronous setting given by Sync HotStuff [Abraham et al. 2020], and strictly greater than $2/3$ in the partially synchronous setting aligned with HotStuff [Yin et al. 2019]. The adversary can coordinate the corrupted nodes to deviate from the protocol in any way. The adversary can also delay or reorder messages sent by honest nodes within the bounds of $[0, \Delta_{\text{sync}}]$, but cannot forge messages.

Unlike Aequitas [Kelkar et al. 2020], which relies on complex primitives such as FIFO-BC or Set-BA, we adopt a simpler report-based approach similar to Themis [Kelkar et al. 2023]. In each consensus slot (block), nodes submit their *local timestamp vector* containing timestamps for all transactions they have received but not yet finalized. The consensus protocol (e.g., HotStuff or Sync HotStuff) then agrees on a *vector of reports* (a set of signed timestamp vectors from different nodes). The validity of a block depends on **External Validity** conditions: (1) each included report must be correctly signed by the corresponding node; (2) the block must contain reports from at least $n - f$ nodes; and (3) the final transaction ordering in the block must be the deterministic result of applying the fairness function to the set of accepted reports.

2.1 Formal Fairness Definitions

We now formally define the related fairness notions. Let the final ledger ordering be a sequence $L = (\mathsf{tx}_1, \mathsf{tx}_2, \dots)$, where $\mathsf{tx} \prec_L \mathsf{tx}'$ denotes tx precedes tx′. Let $\mathsf{Block}(\mathsf{tx})$ and $\mathsf{Index}_L(\mathsf{tx})$ denote the block number and index of tx in L.

Definition 2.1 (Receive-Order-Fairness [Kelkar et al. 2020]). A protocol satisfies φ-receive-order-fairness if:

$$|\{i \in \mathcal{S} : \mathsf{ts}_i(\mathsf{tx}) < \mathsf{ts}_i(\mathsf{tx}')\}| \geq \varphi n \implies \mathsf{tx} \prec_L \mathsf{tx}' \tag{1}$$

Definition 2.2 (Block-Order-Fairness [Kelkar et al. 2020]). A protocol satisfies φ-block-order-fairness if:

$$|\{i \in \mathcal{S} : \mathsf{ts}_i(\mathsf{tx}) < \mathsf{ts}_i(\mathsf{tx}')\}| \geq \varphi n \implies \mathsf{Block}(\mathsf{tx}) \leq \mathsf{Block}(\mathsf{tx}') \tag{2}$$

Block-order-fairness is a weaker notion than receive-order-fairness, as it only requires that tx is not placed in a later block than tx′, allowing for some reordering within the same block. Receive-order-fairness requires a strict ordering of transactions based on the majority reception times. Block-order-fairness is also used in Themis [Kelkar et al. 2023] under the name "batch-order-fairness".

Definition 2.3 (Bounded Unfairness [Kiayias et al. 2024]). A protocol satisfies bounded-unfairness if:

$$|\{i \in \mathcal{S} : \mathsf{ts}_i(\mathsf{tx}) < \mathsf{ts}_i(\mathsf{tx}')\}| \geq \varphi n \implies \mathsf{Index}_L(\mathsf{tx}) - \mathsf{Index}_L(\mathsf{tx}') \leq B \tag{3}$$

Bounded unfairness is a relaxation of receive-order-fairness that allows for some bounded reordering (up to B positions) even when a majority of nodes receive tx before tx′. When $B = 0$, bounded unfairness reduces to receive-order-fairness.

Definition 2.4 (Timed-Relative-Fairness [Kursawe 2020]). A protocol satisfies timed-relative-fairness if when all honest nodes receive tx before any honest node receives tx′, then $\mathsf{tx} \prec_L \mathsf{tx}'$.

$$\max_{i \in \mathcal{H}} \mathsf{ts}_i(\mathsf{tx}) < \min_{j \in \mathcal{H}} \mathsf{ts}_j(\mathsf{tx}') \implies \mathsf{tx} \prec_L \mathsf{tx}' \tag{4}$$

Definition 2.5 (Median-Timestamp-Fairness [Zhang et al. 2020]). A protocol satisfies median-timestamp-fairness if it orders transactions based on the median of reception times: $\mathsf{tx} \prec_L \mathsf{tx}'$ if $\mathsf{median}(\{\mathsf{ts}_i(\mathsf{tx})\}_{i \in \mathcal{S}}) < \mathsf{median}(\{\mathsf{ts}_i(\mathsf{tx}')\}_{i \in \mathcal{S}})$.

Finally, we define Approximate-Order-Fairness (AOF), the focus of this work. Unlike definitions that condition blindly on reception order, AOF introduces a time delay parameter θ and a minimum support threshold φ.

Definition 2.6 (Approximate-Order-Fairness (AOF) [Kelkar et al. 2020]). A protocol satisfies φ-approximate-order-fairness if for any two transactions tx and tx′, if at least a φ fraction of all nodes receive tx at least θ time before receiving tx′, then all honest nodes deliver tx before tx′.

$$|\{i \in \mathcal{S} : \mathsf{ts}_i(\mathsf{tx}) + \theta \leq \mathsf{ts}_i(\mathsf{tx}')\}| \geq \varphi n \implies \mathsf{tx} \prec_L \mathsf{tx}' \tag{5}$$

Pruning vs. Non-Pruning. Under the Δ_{sync}-synchrony assumption, honest timestamps for the same transaction must fall within a strict time window of size Δ_{sync}. Theoretically, protocols could enforce this by *pruning* reports that deviate significantly from the cluster, treating them as adversarial. However, this approach relies heavily on perfect synchrony and requires nodes to agree on a fixed parameter Δ_{sync}, making the system brittle to temporary network instability where actual delays exceed Δ_{sync}. In contrast, a *non-pruning* approach accepts all signed reports and filters outliers solely through the fairness bounds of the ordering function. This makes the protocol robust to real-world latency fluctuations, as nodes do not need to enforce a rigid Δ_{sync} parameter during execution; Δ_{sync} serves only as a variable for feasibility analysis, not a protocol constant. Given that pruning adds significant communication overhead and fragility without yielding a substantially larger feasible parameter space, we adopt the **no-pruning** setting as our default model for the main analysis and protocol design. We relegate the theoretical discussion of the pruning setting to Section 7 for completeness.

Monotonicity of Feasible Parameters. If approximate-order-fairness can be achieved with parameters θ and φ, then it can also be achieved with parameters $\theta' \geq \theta$ and $\varphi' \geq \varphi$. This is because if $\mathsf{tx} \prec_{\theta',\varphi'} \mathsf{tx}'$, then $\mathsf{tx} \prec_{\theta,\varphi} \mathsf{tx}'$. Therefore, an ordering of transactions that satisfies approximate-order-fairness with parameters θ and φ also satisfies approximate-order-fairness with parameters θ' and φ'.

3 The "Timed-Cycle" Limitation of Existing Definitions

A stronger fairness definition should naturally provide stronger guarantees against adversarial manipulation. However, we identify a limitation in existing feasible fairness definitions (Block-Order and Bounded-Unfairness): they typically define fairness based on the *fraction* of nodes reporting relative orderings, without explicitly accounting for the *magnitude* of time differences. We formalize this limitation as the "Timed-Cycle Theorem".

3.1 No Guarantees from Purely Vote-Based Definitions

Intuitively, if all nodes receive transaction tx_A significantly earlier than tx_B (e.g., 10 minutes apart), a robust fairness notion should guarantee $\mathsf{tx}_A \prec \mathsf{tx}_B$. However, existing definitions often rely solely on pairwise majority votes. We show that, conceptually, an adversary can construct a Condorcet cycle involving tx_A and tx_B via a chain of intermediate transactions. Under strict adherence to these definitions, such a cycle could allow tx_A and tx_B to be batched or reordered, thereby technically satisfying the fairness definition even if tx_A arrived much earlier.

Theorem 3.1 (Timed-Cycle Weakness). *Consider a system with n honest nodes and network delay bound Δ_{sync}. For any majority voting threshold $\varphi \in (1/2, 1)$ and any arbitrarily large time gap $D > 0$, there exists a set of transaction arrival times (a profile) such that:*

1. *All nodes receive tx_A at least D time before tx_B. (i.e., $\forall i, \mathsf{ts}_i(\mathsf{tx}_B) - \mathsf{ts}_i(\mathsf{tx}_A) \geq D$).*
2. *There exists a set of auxiliary transactions $\{\mathsf{tx}_1, \ldots, \mathsf{tx}_m\}$ such that the pairwise majority preferences form a cycle: $\mathsf{tx}_B \prec_\varphi \mathsf{tx}_1 \prec_\varphi \mathsf{tx}_2 \prec_\varphi \cdots \prec_\varphi \mathsf{tx}_m \prec_\varphi \mathsf{tx}_A$.*

Combined with the unanimous $\mathsf{tx}_A \prec \mathsf{tx}_B$, this forms a global cycle, formally allowing standard Condorcet-based definitions to order tx_B before tx_A or leave them unordered.

Proof. The intuition is that even when $\mathsf{tx}_1 \prec_\varphi \mathsf{tx}_2$, the average timestamp of tx_1 might be larger than that of tx_2. The φ fraction of nodes that receive tx_1 before tx_2 might receive tx_1 immediately before tx_2, while the remaining $1 - \varphi$ fraction of nodes can receive tx_1 after tx_2 by a large margin. By chaining such transactions, the consistent average timestamp shift moves the window bounds in the reverse direction.

We construct a specific execution profile to demonstrate the existence of such a cycle. Let Δ_{sync} be the network delay bound. Let $k = \lceil 1/(1-\varphi) \rceil + 1$. Pick a value δ such that $0 < \delta < \Delta_{\mathsf{sync}}/k$. Let $\Delta' = \Delta_{\mathsf{sync}} - k\delta > 0$. Let $m = \lceil (D + \epsilon)/\Delta' \rceil$ for any $\epsilon > 0$.

We define the arrival times for the anchor transactions tx_A and tx_B as:

- $\forall i \in \mathcal{S} : \mathsf{ts}_i(\mathsf{tx}_A) = 0$.
- $\forall i \in \mathcal{S} : \mathsf{ts}_i(\mathsf{tx}_B) = D$.

Condition (1) is satisfied directly.

We now construct the sequence of transactions $\mathsf{tx}_1, \ldots, \mathsf{tx}_m$ such that $\mathsf{tx}_B \prec_\varphi \mathsf{tx}_1 \prec_\varphi \cdots \prec_\varphi \mathsf{tx}_{mk+1} \prec_\varphi \mathsf{tx}_A$.

Phase 1: Initial Step ($\mathsf{tx}_B \to \mathsf{tx}_1$) We partition the nodes into two groups:

- Group S_1^{late} with weight φ: receive tx_1 at $D + \delta$.
- Group S_1^{early} with weight $1 - \varphi$: receive tx_1 at $D + \delta - \Delta_{\mathsf{sync}}$.

For the pair $(\mathsf{tx}_B, \mathsf{tx}_1)$, the group S_1^{late} observes timestamp D for tx_B and $D+\delta$ for tx_1. Since the weighted size of S_1^{late} is φ, we have $\mathsf{tx}_B \prec_\varphi \mathsf{tx}_1$. The timestamp spread is exactly Δ_{sync}, ensuring validity.

Phase 2: Intra-Epoch Transitions ($\mathsf{tx}_{i-1} \to \mathsf{tx}_i$) For $1 < i \le k$, we define the arrival times as follows:

- Group S_i^{late} of weight $1-i(1-\varphi)$ receives tx_i at $D+i\delta$.
- Group S_i^{early} of weight $i(1-\varphi)$ receives tx_i at $D+i\delta-\Delta_{\mathsf{sync}}$.

Consider the transition from tx_{i-1} to tx_i. The set of nodes observing $\mathsf{tx}_{i-1} \prec \mathsf{tx}_i$ consists of:

(1) The group that remains "late" in both steps ($D+(i-1)\delta \to D+i\delta$).
(2) The group that remains "early" in both steps ($D+(i-1)\delta-\Delta_{\mathsf{sync}} \to D+i\delta-\Delta_{\mathsf{sync}}$).

The only nodes that observe a time decrease (and thus $\mathsf{tx}_i \prec \mathsf{tx}_{i-1}$) are those that switch from the "late" group of tx_{i-1} to the "early" group of tx_i. The weight of this switching group is exactly $(1-(i-1)(1-\varphi)) - (1-i(1-\varphi)) = 1-\varphi$. Consequently, the weight of nodes observing the forward order $\mathsf{tx}_{i-1} \prec \mathsf{tx}_i$ is $1-(1-\varphi)=\varphi$. Thus, $\mathsf{tx}_{i-1} \prec_\varphi \mathsf{tx}_i$ holds.

Phase 3: Epoch Reset ($\mathsf{tx}_k \to \mathsf{tx}_{k+1}$) After k steps, the "early" group has grown and the "late" group has shrunk. At the reset step, we move all nodes in the late group of round k to the left group. This time, we use a $1-\varphi$ fraction of nodes in S_k^{early} to form S_{k+1}^{late}, and all remaining nodes form S_{k+1}^{early}. The arrival times for tx_{k+1} are:

- Group S_{k+1}^{late} of weight φ: receive tx_{k+1} at $D+(k+1)\delta-\Delta_{\mathsf{sync}}$.
- Group S_{k+1}^{early} of weight $1-\varphi$: receive tx_{k+1} at $D+(k+1)\delta-2\Delta_{\mathsf{sync}}$.

Comparing tx_k (where the early group was at $D+k\delta-\Delta_{\mathsf{sync}}$) with tx_{k+1}: The group of weight φ (subset of the previous early group) sees timestamps increase from $D+k\delta-\Delta_{\mathsf{sync}}$ to $D+(k+1)\delta-\Delta_{\mathsf{sync}}$, preserving the forward order. The remaining nodes see a large time drop. Since weight φ supports the order, $\mathsf{tx}_k \prec_\varphi \mathsf{tx}_{k+1}$ holds. Critically, the timestamp bounds have shifted from $[D+\delta-\Delta_{\mathsf{sync}}, D+\delta]$ for tx_1 to $[D+(k+1)\delta-2\Delta_{\mathsf{sync}}, D+(k+1)\delta-\Delta_{\mathsf{sync}}]$ for tx_{k+1}.

Closing the Cycle By repeating this process over multiple epochs, the timestamp interval shifts left by exactly $\Delta_{\mathsf{sync}}-k\delta$ every k steps. Since $\Delta_{\mathsf{sync}} > k\delta$, this is a net decrease. After m epochs, we construct a transaction tx_{last} where all reception times are strictly less than 0. Comparing tx_{mk+1} (all times < 0) with tx_A (all times 0), all nodes unanimously agree $\mathsf{tx}_{mk+1} \prec \mathsf{tx}_A$. This completes the Condorcet cycle: $\mathsf{tx}_B \prec_\varphi \mathsf{tx}_1 \prec_\varphi \cdots \prec_\varphi \mathsf{tx}_{mk+1} \prec_\varphi \mathsf{tx}_A \prec_\varphi \mathsf{tx}_B$. □

3.2 Implications for Fairness Models

This theorem highlights a gap between the intuitive expectation of approximate-order-fairness and the formal guarantees of fairness notions that do not use timestamps:

- **Block-Order-Fairness (Aequitas):** Upon detecting the cycle $\mathsf{tx}_A \to \cdots \to \mathsf{tx}_B \to \ldots \mathsf{tx}_A$, a protocol satisfying Block-Order-Fairness is permitted to batch all involved transactions into a single block. While specific implementations might employ heuristics to break ties within such cycles and order tx_A before tx_B, the *definition* itself does not mandate this strict ordering, treating the large time gap as irrelevant once a cycle is formed.
- **Bounded-Unfairness (Taxis):** This definition requires finding a specific linearization that minimizes unfairness. However, theoretically, the definition allows for valid linearizations where tx_B could precede tx_A if the cumulative weight of the cycles favors it, or allows tx_B to be placed much closer to tx_A than the time gap suggests. Again, implementations might avoid this, but the definition does not explicitly forbid it.

The Consequences for DeFi. In Decentralized Finance (DeFi), time is generally the critical factor. Under weaker definitions, if tx_A (user trade) acts as a pivot in a cycle involving tx_B (attacker trade), the formal guarantees ensuring tx_A precedes tx_B vanish. The adversary can potentially exploit this "definition gap" by flooding the network with "chaff" transactions ($\mathsf{tx}_1 \dots \mathsf{tx}_k$) to create cycles. AOF aims to render this theoretical attack impossible by explicitly filtering out the "chaff" edges (where gaps $< \theta$) and preserving the strong $\mathsf{tx}_A \longrightarrow \mathsf{tx}_B$ edge.

3.3 The Unbounded Dependency Problem

A further critical consequence of Theorem 3.1 is the *Unbounded Dependency Problem*. The construction of the cycle $\mathsf{tx}_A \prec \mathsf{tx}_B \prec \mathsf{tx}_1 \prec \cdots \prec \mathsf{tx}_k$ can be extended indefinitely. An adversary can continue adding $\mathsf{tx}_{k+1}, \mathsf{tx}_{k+2}, \dots$ effectively "pulling" the dependency chain from the arbitrarily distant future back to the present.

Existing literature, including Aequitas [Kelkar et al. 2020] and Themis [Kelkar et al. 2023], has identified and discussed this issue of potential long-range dependencies in Condorcet cycles. Our contribution here is to provide formal reasoning about the *time span* of these dependencies relative to the fairness parameter. While implementations of existing protocols may achieve correct ordering in practice, their fairness definitions do not inherently capture or bound these time-span anomalies.

For protocols aiming to satisfy fine-grained fairness (like Aequitas or Taxis), this creates a theoretical dilemma for block production at any time t:

(1) **Wait Indefinitely:** To safely order tx_A (received at t) according to the definition, the protocol must theoretically ensure no future transaction tx_Z will form a cycle back to tx_A.
(2) **Risk Fairness Violation:** If the protocol decides to cut and finalize a block, it risks a future transaction revealing a valid cycle that retroactively complicates the ordering graph.

AOF Solution: AOF solves this by providing a "clean cut". Because we only enforce $\mathsf{tx} \prec \mathsf{tx}'$ when there is a significant gap θ, typical short-interval dependencies (close to Δ_{sync}) are filtered out. If we cut collection at time t, we can be certain that no transaction received after $t + \theta$ can possibly θ-precede any transaction received before t. This bounded dependency is crucial for practical, low-latency consensus.

4 Possibility Result

In this section, we present our main possibility result for achieving AOF in a synchronous network, ignoring communication and computation costs.

Notation - Precede Relation. For two transactions tx and tx', let $\mathcal{I}_\theta(\mathsf{tx}, \mathsf{tx}')$ denote the set of nodes among the n nodes $\mathcal{S}$ participating in the transaction-ordering scheme that receive tx at least θ before tx'.

$$\mathcal{I}_\theta(\mathsf{tx}, \mathsf{tx}') = \{i \in \mathcal{S} : \mathsf{ts}_i(\mathsf{tx}) + \theta \leq \mathsf{ts}_i(\mathsf{tx}')\}$$

We say there is a (θ, φ)-precede relation between tx and tx', denoted $\mathsf{tx} \prec_{\theta,\varphi} \mathsf{tx}'$, if $|\mathcal{I}_\theta(\mathsf{tx}, \mathsf{tx}')| \geq n\varphi$. When the context is clear, we omit the subscripts and simply write $\mathcal{I}(\mathsf{tx}, \mathsf{tx}')$ and $\mathsf{tx} \prec \mathsf{tx}'$. Using the above notation, AOF requires that for any two transactions tx and tx' such that $\mathsf{tx} \prec \mathsf{tx}'$, tx be delivered before tx'.

Protocol Design. The transaction-ordering protocols in this section do not consider communication and computation costs; the goal is to achieve AOF with the tightest possible parameters. Upon receiving a transaction, each node records the receive time and runs a Byzantine broadcast protocol for the transaction together with its timestamp. Under the assumption that the network is Δ_{sync}-synchronous and more than $2/3$ of the nodes are honest, Byzantine broadcast protocols terminate in bounded time when the broadcaster is honest. By Byzantine broadcast, if an honest

node i confirms a transaction tx with timestamp t from a node j, all other honest nodes also confirm that node j received tx at time t. Therefore, all honest nodes have the same view of when each node receives each transaction. Our transaction-ordering mechanism outputs an order of transactions based on the timestamp view of all nodes and the existing blockchain prefix, both of which are the same in the view of all honest nodes. Therefore, all honest nodes output the same new block and blockchain consensus is achieved. Such a protocol is highly inefficient in practice. In later sections, we discuss more efficient protocols for ordering transactions while satisfying AOF.

Feasibility and Precede Loops. AOF with parameters θ and φ is possible if and only if there is no sequence of transactions $\mathsf{tx}_1, \mathsf{tx}_2, \ldots, \mathsf{tx}_k$ such that $\mathsf{tx}_1 \prec \mathsf{tx}_2$, $\mathsf{tx}_2 \prec \mathsf{tx}_3$, …, $\mathsf{tx}_{k-1} \prec \mathsf{tx}_k$, and $\mathsf{tx}_k \prec \mathsf{tx}_1$. We call such a sequence a (θ, φ)-precede loop. If there is a (θ, φ)-precede loop, then AOF with parameters θ and φ is impossible because the ordering constraints in the loop contradict one another. If there is no (θ, φ)-precede loop, then AOF with parameters θ and φ can be achieved by topologically sorting the directed graph formed by all transactions as nodes and all (θ, φ)-precede relations as directed edges.

Parameter Space for φ. For AOF to be meaningful, φ should be in the range $(1-h, h]$. If $\varphi \le 1-h$, then the adversary has enough votes to impose an arbitrary order on any two transactions even though all honest nodes receive them in the opposite order. If $\varphi > h$, then even if all honest nodes receive transaction tx at least θ before transaction tx′, they do not have enough votes to impose a constraint that delivers tx before tx′ when the adversary does not cooperate.

Theorem 4.1. *(Infeasible parameters without pruning) Under a Δ_{sync}-synchronous network, in a system where an h fraction of nodes are honest, for any positive integer k such that $1 - h/k \le h$, i.e., $k \le h/(1-h)$, AOF without pruning cannot be achieved with parameters $\theta \le \Delta_{\mathsf{sync}}/k$ and $\varphi < 1 - h/(k+1)$.*

Proof. Define a function $f : \mathbb{Z}^+ \to \mathbb{R}$ as

$$f(m) = \frac{k}{k+1}h + \frac{m(k+1)-1}{m(k+1)}(1-h)$$

f is a monotonically increasing function. When $m = 1$, $f(1) = k/(k+1)$. When $m \to \infty$, $f(m) \to (k/(k+1))h + 1 - h = 1 - h/(k+1)$. For any $\varphi < 1 - h/(k+1)$, there exists a positive integer m such that $\varphi < f(m)$. We prove that $(\Delta_{\mathsf{sync}}/k, \varphi)$-AOF is infeasible by constructing a sequence of $m(k+1)$ transactions $\mathsf{tx}_0, \mathsf{tx}_1, \ldots, \mathsf{tx}_{m(k+1)-1}$ that form a $(\Delta_{\mathsf{sync}}/k, f(m))$-precede loop.

Partition honest nodes into $k+1$ groups $\mathcal{H}_0, \mathcal{H}_1, \ldots, \mathcal{H}_k$, each containing an $h/(k+1)$ fraction of all nodes. For each group $\mathcal{H}_j$ and each transaction tx_i, all nodes in $\mathcal{H}_j$ receive tx_i at time $\big((i+j) \bmod (k+1)\big) \cdot \Delta_{\mathsf{sync}}/k$. It is easy to verify that this construction does not violate the Δ_{sync}-synchrony assumption, as for each transaction tx_i the timestamps fall into a Δ_{sync}-sized time window $[0, \Delta_{\mathsf{sync}}]$. For each pair $(\mathsf{tx}_i, \mathsf{tx}_{i+1})$ (defining $\mathsf{tx}_{m(k+1)} = \mathsf{tx}_0$ for notational simplicity), k groups of honest nodes receive tx_i before tx_{i+1} by time Δ_{sync}/k.

Partition adversarial nodes into $m(k+1)$ groups $\mathcal{A}_0, \mathcal{A}_1, \ldots, \mathcal{A}_{m(k+1)-1}$, each containing a $(1-h)/(m(k+1))$ fraction of all nodes. For each group $\mathcal{A}_j$ and each transaction tx_i, all nodes in $\mathcal{A}_j$ receive tx_i at time $\big((i+j) \bmod m(k+1)\big) \cdot \Delta_{\mathsf{sync}}/k$. For each pair $(\mathsf{tx}_i, \mathsf{tx}_{i+1})$, $m(k+1)-1$ groups of adversarial nodes receive tx_i before tx_{i+1} by time Δ_{sync}/k. Overall, for each pair $(\mathsf{tx}_i, \mathsf{tx}_{i+1})$, the fraction of nodes in $\mathcal{I}(\mathsf{tx}_i, \mathsf{tx}_{i+1})$ is $\big(k/(k+1)\big) \cdot h + \big((m(k+1)-1)/(m(k+1))\big) \cdot (1-h) = f(m) > \varphi$. Therefore, $\mathsf{tx}_i \prec \mathsf{tx}_{i+1}$ for all $0 \le i \le m(k+1)-1$. These $m(k+1)$ transactions form a $(\Delta_{\mathsf{sync}}/k, \varphi)$-precede loop. □

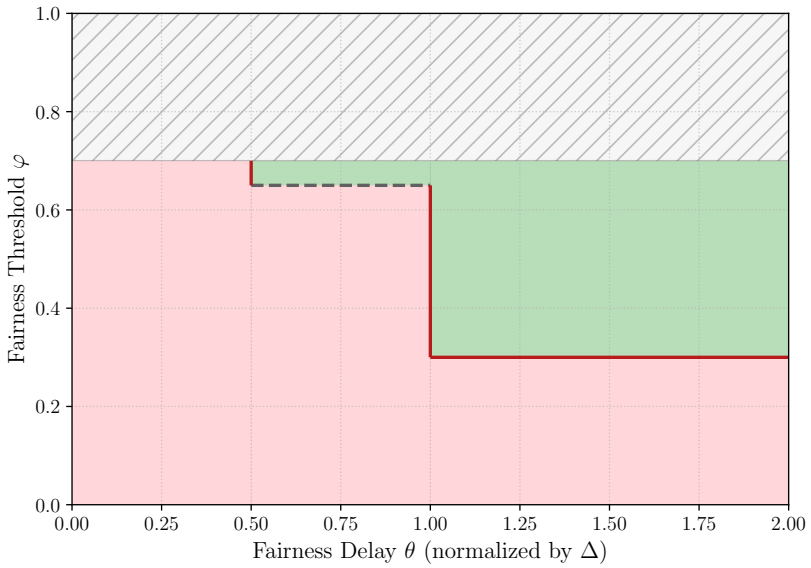

Fig. 1. Feasible and infeasible parameters for AOF without pruning bad reports (when $h = 0.7$). Green areas indicate feasible parameters, red/gray areas indicate infeasible parameters, and dashed lines indicate unknown areas.

Theorem 4.2. *(Feasible parameters without pruning) Under a Δ_{sync}-synchronous network, in a system where an h fraction of nodes are honest, for any positive integer k such that $1 - h/k \leq h$, i.e., $k \leq h/(1-h)$, AOF without pruning can be achieved with parameters $\theta > \Delta_{\text{sync}}/k$ and $\varphi > 1 - h/k$.*

Proof. We prove Theorem 4.2 by analyzing the time window in which honest nodes receive each transaction. We first prove the special case of $k = 1$. Our Δ_{sync}-synchrony assumption ensures that, for each transaction, the timestamps of all honest nodes receiving it fall into a time window of size Δ_{sync}. Assume for the sake of contradiction that there is a (θ, φ)-precede loop $\mathsf{tx}_1 \prec \mathsf{tx}_2$, $\mathsf{tx}_2 \prec \mathsf{tx}_3, \ldots, \mathsf{tx}_{l-1} \prec \mathsf{tx}_l$, and $\mathsf{tx}_l \prec \mathsf{tx}_1$. For each transaction tx_i, let $[s_i, e_i]$ be the time window in which all honest nodes receive tx_i. For simplicity, define transaction tx_{l+1} as an alias of tx_1. For each i in $1, 2, \ldots, l$, since $\mathsf{tx}_i \prec \mathsf{tx}_{i+1}$, at least $\varphi(> 1 - h)$ nodes receive tx_i at least $\theta(> \Delta_{\text{sync}})$ before tx_{i+1}, and at least one such node n_i is honest. By definition of the time windows, $\mathsf{ts}_{n_i}(\mathsf{tx}_i) \geq s_i$ and $\mathsf{ts}_{n_i}(\mathsf{tx}_{i+1}) \leq e_{i+1}$. By the precede relation, $\mathsf{ts}_{n_i}(\mathsf{tx}_i) \leq \mathsf{ts}_{n_i}(\mathsf{tx}_{i+1}) - \theta < \mathsf{ts}_{n_i}(\mathsf{tx}_{i+1}) - \Delta_{\text{sync}}$. Chaining the inequalities, we have $s_i \leq \mathsf{ts}_{n_i}(\mathsf{tx}_i) < \mathsf{ts}_{n_i}(\mathsf{tx}_{i+1}) - \Delta_{\text{sync}} \leq e_{i+1} - \Delta_{\text{sync}}$. In short, $s_i < e_{i+1} - \Delta_{\text{sync}}$. Since $e_{i+1} - s_{i+1} \leq \Delta_{\text{sync}}$, we have $s_i < s_{i+1}$ for every i. This implies $s_1 < s_{l+1} = s_1$, which is a contradiction.

Now we prove Theorem 4.2 for the case where k is an integer larger than 1, using a more fine-grained time-window analysis. Assume for the sake of contradiction that there is a (θ, φ)-precede loop $\mathsf{tx}_1 \prec \mathsf{tx}_2 \cdots \prec \mathsf{tx}_l \prec \mathsf{tx}_1$. For each precede relation $\mathsf{tx}_i \prec \mathsf{tx}_{i+1}$, more than a $1 - h/k$ fraction of nodes are in $\mathcal{I}(\mathsf{tx}_i, \mathsf{tx}_{i+1})$. This implies that less than an h/k fraction of nodes are not in $\mathcal{I}(\mathsf{tx}_i, \mathsf{tx}_{i+1})$. In particular, fewer than hn/k honest nodes are not in $\mathcal{I}(\mathsf{tx}_i, \mathsf{tx}_{i+1})$, which accounts for less than a $1/k$ fraction of all honest nodes.

Let the earliest timestamp among all honest nodes receiving tx_1 be t. By the precede relation $\mathsf{tx}_1 \prec \mathsf{tx}_2$, more than a $(k-1)/k$ fraction of honest nodes receive tx_2 at or after $t + \theta > t + \Delta_{\text{sync}}/k$.

The remaining less than $1/k$ fraction of honest nodes receive tx_2 after $t + (1/k - 1)\Delta_{\mathsf{sync}}$ due to Δ_{sync}-synchrony. The following lemma generalizes this induction.

Lemma 4.3. *Under a Δ_{sync}-synchronous network, in a system where an h fraction of nodes are honest, under approximate-order-fairness parameters $\theta > \Delta_{\mathsf{sync}}/k$ and $\varphi > 1 - h/k$, for any integer $i \in \{0, 1, \ldots, k-1\}$, if $\mathsf{tx}_a \prec \mathsf{tx}_b$ and at least a $(k-i)/k$ fraction of honest nodes receive tx_a at or after some time t, which also implies that the other at most i/k fraction of honest nodes receive tx_a at or after $t - \Delta_{\mathsf{sync}}$, then more than a $(k-i-1)/k$ fraction of honest nodes receive tx_b after $t_{i+1} = t + \theta > t + \Delta_{\mathsf{sync}}/k$. The remaining less than $(i+1)/k$ fraction of honest nodes receive tx_b after $t + (1/k - 1)\Delta_{\mathsf{sync}}$.*

Proof. Among the at least $(k-i)/k$ fraction of honest nodes that receive tx_a at or after time t, more than a $(k-i)/k - 1/k = (k-i-1)/k$ fraction of honest nodes are in $\mathcal{I}(\mathsf{tx}_a, \mathsf{tx}_b)$, and therefore receive tx_b at or after $t + \theta$. The remaining less than $(i+1)/k$ fraction of honest nodes receive tx_b at or after $t + (1/k - 1)\Delta_{\mathsf{sync}}$ due to Δ_{sync}-synchrony. □

We can apply Lemma 4.3. Given that more than $(k-1)/k$ fraction of honest nodes receive tx_2 after time $t + \Delta_{\mathsf{sync}}/k$, we can infer that more than $(k-2)/k$ fraction of honest nodes receive tx_3 after time $t + 2\Delta_{\mathsf{sync}}/k$. By repeatedly applying the lemma for k times we show that for any integer $i \in \{2, \ldots, k+1\}$, more than $(k - (i-1))/k$ fraction of honest nodes receive tx_i after $t + (i-1) \cdot \Delta_{\mathsf{sync}}/k$. In particular, for $i = k+1$, more than $0/k = 0$ fraction of honest nodes receive tx_{k+1} after $t + k \cdot \Delta_{\mathsf{sync}}/k = t + \Delta_{\mathsf{sync}}$. This implies that all honest nodes receive tx_{k+1} after t.

Recall s_i denotes the earliest time when an honest node receives tx_i, the above analysis shows that $s_1 < s_{k+1}$. The above analysis starts from tx_1, but can also start from any other tx_j in the loop, to get $s_2 < s_{k+2}, \ldots, s_{l-k} < s_l, s_{l-k+1} < s_1, \ldots, s_l < s_k$. Summing over these l inequalities, we get $s_1 + s_2 + \cdots + s_l < s_1 + s_2 + \cdots + s_l$, which is a contradiction. The proof for theorem 4.2 is complete. □

4.1 Trade-off between Granularity and Consensus Strength

The results above reveal a fundamental trade-off between the time granularity θ and the consensus strength characterized by the voting threshold φ.

- **High Robustness ($k = 1$):** When the required time separation is large ($\theta > \Delta_{\mathsf{sync}}$), AOF is feasible with a voting threshold slightly above $1 - h$. This means that even with a simple majority of honest nodes (and minimal adversarial tolerance), we can enforce ordering for clearly separated transactions. This corresponds to the standard synchronous model where Δ_{sync} is the unit of observation.
- **High Granularity ($k > 1$):** As we demand finer-grained ordering ($\theta \approx \Delta_{\mathsf{sync}}/k$), the protocol requires a significantly higher voting threshold $\varphi > 1 - h/k$. For large k, φ approaches 1, meaning almost unanimous agreement is needed to safely order transactions with small time differences. This essentially means that extraordinary claims (ordering close events) require extraordinary evidence (near-unanimity).

This flexibility allows system designers to tune the fairness parameter (θ, φ) based on network conditions and security requirements. For instance, a DeFi exchange might prioritize high granularity (large k) to prevent fine-grained front-running, accepting that this requires a very healthy network state (high φ availability). Conversely, a general-purpose chain might prioritize robustness (small k), guaranteeing ordering only for clearly separated events while maintaining liveness even under adversarial pressure. Such a parameterizable trade-off is a unique advantage of AOF compared to rigid definitions such as Timed-Order Fairness.

5 Efficient Leader-Based Protocols

We present leader-based protocols that leverage the well-understood architectures of modern BFT consensus, such as HotStuff [Yin et al. 2019], to achieve approximate-order-fairness with quadratic communication complexity ($O(n^2)$) in the straightforward implementation. As in Themis [Kelkar et al. 2023], optimal $O(n)$ communication is achievable using SNARK-based aggregation.

The core idea is to separate the **consensus on the set of reports** from the **ordering logic**.

(1) **Consensus:** The leader and committee reach consensus on a set of timestamp reports $\mathcal{R}$ to be considered for the current block.
(2) **Ordering:** The final transaction order is determined by a deterministic function $\text{Order}(\mathcal{R})$. This function applies the graph-based ordering logic (constructing the (θ, φ)-precedence graph and topologically sorting it) to the agreed-upon reports.

Crucially, because $\text{Order}(\cdot)$ is deterministic, a malicious leader cannot manipulate the transaction order freely; they can only influence the order by manipulating the input set $\mathcal{R}$ (e.g., by excluding honest reports). We present two variants: a Synchronous Protocol that prevents such exclusion, and a Partially Synchronous Protocol that trades some fairness strength for adaptability.

5.1 Deterministic Ordering Function

Before detailing the protocols, we define the common ordering function $\text{Order}(\mathcal{R})$. Given a set of reports $\mathcal{R} = \{R_1, \ldots, R_m\}$ where each R_i contains timestamps for transactions, the function proceeds as follows:

- **Choosing Transactions:** Reports may contain different subsets of transactions. We adopt a deterministic selection method. Since it depends on network properties, we describe the methods for the synchronous and partially synchronous protocols separately.
- **Pairwise Thresholds:** For every pair of transactions $(\text{tx}_a, \text{tx}_b)$, compute the available support for precedence. Let $\Delta_i(\text{tx}_a, \text{tx}_b) = \text{ts}_i(\text{tx}_b) - \text{ts}_i(\text{tx}_a)$. We calculate the $(\lceil \varphi n \rceil)$-th largest value among these differences in $\mathcal{R}$, denoted $\Theta_{a,b}$.
- **Graph Construction:** For a target delay parameter θ, construct a directed graph where an edge $\text{tx}_a \to \text{tx}_b$ exists if $\Theta_{a,b} \geq \theta$.
- **Acyclicity and Sorting:** Find the smallest $\theta \geq 0$ such that the graph is acyclic. The final order is the topological sort of this graph.

5.2 Synchronous Protocol (Sync HotStuff)

In a fully synchronous network with bound Δ_{sync}, we can achieve the strongest fairness guarantees. We build upon **Sync HotStuff** [Abraham et al. 2020], which simplifies the view-change logic and improves commit latency compared to partially synchronous variants.

5.2.1 Protocol Overview. We define a deadline for report submission in each slot (block), T_{ddl}, by which every node should send its timestamp report to every consensus node. At time $T_{ddl} + \Delta_{\text{sync}}$, for each node i, if it does not receive the report from node j, it sends a signed *absence* vote about j to the leader. At time $T_{ddl} + 2\Delta_{\text{sync}}$, the leader starts the Sync HotStuff protocol to reach consensus on a block for the slot, making a proposal that includes the block as well as the set of reports $\mathcal{R}$.

5.2.2 Transaction Selection. When we aim to achieve AOF with $\theta > \Delta_{\text{sync}}$, we simply select all transactions tx such that at least $\varphi > 1 - h$ fraction of nodes report tx before $T_{ddl} - \Delta_{\text{sync}}$. This guarantees that an honest node reports tx before $T_{ddl} - \Delta_{\text{sync}}$, so every honest node should report tx before T_{ddl}. For any report that does not include tx, we can assign a default timestamp for it as the $1 - h$'s largest timestamp among all reports for tx plus Δ_{sync}, which is guaranteed to be before T_{ddl}. If any transaction tx′ possibly precedes tx, at least φ fraction of nodes report tx′ before tx,

and at least one honest node j is in it. This also ensures that tx' is selected similar to tx. Therefore, we guarantee to select all transactions that can possibly precede tx. The same logic applies to all transactions. Our selection method ensures a "clear cut", that the selected transactions are exactly those that can possibly precede tx according to the reports.

For $\theta > \Delta_{\text{sync}}/k$ and $\varphi > 1 - h/k$, we can use a more aggressive selection method to achieve the same clean-cut effect. The intuition draws from the proof of Theorem 4.2, where for a sequence of transactions $\mathsf{tx}_1 \prec \cdots \prec \mathsf{tx}_n$, the time window of transaction tx_i gradually shifts rightward.

5.2.3 Validity Conditions. To integrate fairness into the consensus layer, we enforce strict validity conditions on the proposed blocks. A block B carrying a set of timestamp reports $\mathcal{R}$ is considered valid if and only if:

(1) **Report Integrity:** The block must contain a set of validly signed reports $\mathcal{R} = \{R_1, \ldots, R_k\}$ where each R_i is a tuple $(i, \text{view}_i, \sigma_i)$ signed by node i.
(2) **Inclusion Validity (Synchrony):** The set $\mathcal{R}$ must include reports from all honest nodes. To enforce this in the presence of faults, if a report from node j is missing ($j \notin \mathcal{R}$), the block proposal must include a proof of node j's equivocation or a *Quorum of Absence* certificate. This certificate consists of $f + 1$ signatures attesting that node j's report was not received by the proposal deadline T_{max}. Since there are at most f malicious nodes, it is impossible to form such a quorum against an honest node that broadcast its report on time.
(3) **Ordering Determinism:** The transaction ordering within block B must be the deterministic result of applying the ordering function to the set of accepted reports $\mathcal{R}$. Any deviation from this canonical ordering renders the block invalid.

5.2.4 Fairness Guarantees. Since the protocol forces the inclusion of all available honest reports, the adversary cannot artificially lower the honest ratio.

- **Parameter Preservation:** The feasible region for this protocol is identical to the theoretical bounds derived in Section 4. For any k, AOF is feasible with $\theta > \Delta_{\text{sync}}/k$ and $\varphi > 1 - h/k$.
- **Robustness:** This setting demonstrates the peak capability of AOF. A malicious leader has no power to alter the outcome beyond their own private inputs, which are already accounted for in the f Byzantine fraction.

5.3 Partially Synchronous Protocol (HotStuff)

To support networks where Δ_{sync} is unknown or unstable, we adapt the approach to standard **HotStuff** [Yin et al. 2019].

5.3.1 Validity Conditions. In the partially synchronous setting, we cannot demand the inclusion of all honest reports, as the leader cannot reliably distinguish between a Byzantine node and a slow honest node. The validity conditions are relaxed to ensure liveness:

(1) **Report Integrity:** The block must contain a set of validly signed reports $\mathcal{R}$, similar to the synchronous case.
(2) **Quorum Availability:** The set $\mathcal{R}$ must contain reports from at least a quorum of $n-f$ nodes. This ensures that the protocol can proceed even if f nodes are Byzantine or partitioned.
(3) **Ordering Determinism:** Similar to the synchronous case, the transaction ordering within the block must strictly follow the output of the fairness function $\mathcal{F}$ applied to the included set $\mathcal{R}$. Validators will recompute the ordering locally based on $\mathcal{R}$ and reject the block if the orderings do not match.

5.3.2 Effective Honest Ratio and Trade-off. In this setting, a malicious leader *can* manipulate the set $\mathcal{R}$. Specifically, the leader can wait for f malicious reports and $n - 2f$ honest reports, ignoring the remaining f honest reports (claiming they were slow).

- **Reduction in Honest Power:** The set $\mathcal{R}$ has size $M = n - f$. Within this set, only $n - 2f$ reports are guaranteed to be honest.
- **Effective Honest Ratio (h'):** The fraction of honest reports in the consensus set is $h' = \frac{n-2f}{n-f}$.

This reduction in honest participation weakens the fairness guarantee. The protocol effectively operates in a subsystem with a lower honest majority.

- **Feasibility Condition:** To achieve AOF with time parameter $\theta > \Delta_{act}/k$, where Δ_{act} denotes the actual max delay, we typically need a voting threshold φ relative to the *available* reports.

$$\varphi_{local} > 1 - \frac{h'}{k} = 1 - \frac{n-2f}{k(n-f)}$$

- **Analysis:** While weaker than the synchronous case, this result shows AOF is compatible with partial synchrony. AOF degrades gracefully: during asynchrony, no fairness is guaranteed (as timestamps diverge), but consistency and liveness are maintained. During periods of synchrony (GST), fairness is restored, albeit with parameters dictated by the worst-case leader behavior.

The transaction selection procedure used in the synchronous protocol no longer applies, which means we might lose the clean-cut property across transactions.

5.4 Comparison and Conclusion

- **Sync Protocol:** Best for high-performance, permissioned settings (e.g., DeFi rollups, consortium chains) where network bounds are reliable. It maximizes fairness granularity ($\theta \to 0$) and resilience (h).
- **Psync Protocol:** Best for general-purpose, permissionless settings (e.g., L1 blockchains) where network robustness is prioritized over maximal fairness efficiency. It offers a safe fallback, ensuring AOF does not hinder consensus liveness.

Both protocols utilize the underlying BFT consensus as a "carrier" for report aggregation, while the actual ordering logic remains a deterministic function of the aggregated data. This separation of concerns simplifies implementation and security analysis.

6 Detailed Comparison with Existing Schemes

In this section, we provide a detailed comparison of our *Approximate-Order-Fairness* (AOF) against the three primary fairness definitions in the literature: *Block-Order-Fairness* (used in Aequitas [Kelkar et al. 2020]), *Timed-Order-Fairness* (used in Wendy [Kursawe 2020]), and *Bounded-Unfairness* (used in Taxis [Kiayias et al. 2024]). We contextualize these comparisons within the standard synchronous network model where message dissemination takes at most Δ time.

To ensure a fair comparison, we focus on the intrinsic properties of the fairness definitions rather than implementation artifacts. We assume the strongest possible implementation for each fairness notion (e.g., using all-to-all broadcast to maximize information), effectively leveling the playing field regarding communication complexity. We also assume a benign environment with no malicious reporting and a fixed synchronous window Δ_{sync} for all protocols. Our comparison thus focuses on:

- **Granularity and Strength:** The ability to order fine-grained transactions and prevent manipulation.
- **Computational Complexity:** The algorithmic cost of deriving the fair order from the collected data (e.g., polynomial vs. NP-hard).

6.1 Comparison with Block-Order-Fairness

Aequitas introduced Block-Order-Fairness to circumvent the Condorcet paradox. Their definition states that if a sufficient fraction of nodes receive tx before tx′, then no honest node delivers tx in a block *after* tx′. This work also defined *Approximate-Order-Fairness* (AOF) but dismissed it under Theorem 4.6, arguing that achieving it implies achieving strict fairness on a faster network (which is impossible if the faster network allows jitter).

- **Revisiting the Impossibility (Physical vs. Logical):** We argue that the dismissal of AOF based on Theorem 4.6 in [Kelkar et al. 2020] relies on a "padding" reduction that is invalid for time-aware protocols. The theorem's proof constructs a simulation where an adversary inserts logical "round padding" (silence) to dilate time, arguing that any protocol achieving approximate fairness implies strict fairness on a faster logical network. While valid for round-oblivious protocols, this fails in a **Timed-Synchronous Execution Model** where nodes observe physical time. An adversary cannot insert physical silence without it being measured by honest local clocks. A time-aware protocol that checks for physical gaps θ can distinguish between a slow physical execution and a "padded" simulation. This breaks the simulation equivalence: the impossibility of strict fairness on the faster network does not propagate to approximate fairness on the slower physical network.
- **Granularity:** Block-Order-Fairness is coarse-grained. If tx and tx′ are involved in a cycle (or even seemingly innocent concurrent ordering), the protocol is permitted to batch them into the same block. Within a block, the ordering is arbitrary (or leader-dictated). This leaves a wide vulnerability window for attackers who can ensure their transactions are included in the same block as a victim's transaction, enabling front-running or sandwich attacks.
- **Strength:** AOF offers finer granularity. We enforce a pairwise ordering $\mathsf{tx} \prec \mathsf{tx}'$ when the condition is met, without batching. This significantly reduces the intra-block manipulation window for any pair that satisfies the time-separation condition.

6.2 Comparison with Timed-Order-Fairness

Wendy and other related works use variations of Timed-Order-Fairness (sometimes called Timed-Relative-Fairness). The typical condition is: if *every* honest node receives tx before *any* honest node receives tx′, then $\mathsf{tx} \prec \mathsf{tx}'$.

- **Generalization:** AOF is a generalization of this concept. The condition "all honest before any honest" implies a time separation of at least the network diameter Δ. In our notation, this roughly corresponds to requiring $\theta > \Delta$.
- **Coverage:** Timed-Order-Fairness is a very strong condition that is rarely met in high-load systems where transactions arrive continuously and intervals overlap. It provides no guarantees for the vast majority of transactions where receive times overlap (i.e., when tx arrives at some nodes before tx′ and vice versa). AOF handles these overlapping cases using the parameterized voting threshold φ. We can enforce order even when receive intervals overlap, provided the aggregate time difference is statistically significant (defined by our feasible (θ, φ) region).

6.3 Comparison with Bounded-Unfairness

Taxis introduces Bounded-Unfairness, minimizing the value B such that if a fairness condition is met, tx is placed no more than B positions after tx′.

- **Metric Relevance (Time vs. Positions):** Bounded-Unfairness measures error in *positions*. However, in financial contexts (e.g., decentralized exchanges), *time* is the critical resource. An attacker needs time to observe a victim's transaction and react. AOF targets this directly: if an honest user submits tx sufficiently early (θ), AOF guarantees tx is executed first, preempting reactive attacks regardless of the number of other transactions.
- **Algorithmic Complexity:** Finding the optimal ordering to minimize B is equivalent to the *Directed Bandwidth* problem, which is NP-hard. While Taxis employs heuristics or approximations (such as "Timed-Directed-Bandwidth-Fairness"), the optimal solution remains computationally intensive. In contrast, AOF achieves its guarantees using polynomial-time algorithms, specifically by constructing and topologically sorting the AOF graph.
- **Delay-Optimality:** AOF is designed to find the minimal delay θ required to order transactions consistently. In Taxis, for the synchronous setting, the bound B is shown to be related to the number of concurrent transactions within a Δ window. AOF improves upon this by proving that for specific (θ, φ) trade-offs, we can achieve $B = 0$ (strict ordering). Thus, AOF can be viewed as identifying the conditions under which Bounded Unfairness can be reduced to zero, providing the strongest possible guarantee for a given network latency.

6.4 Comparison with Median-Timestamp-Fairness

Protocols like Pompē [Zhang et al. 2020] order transactions based on the median of reported timestamps. While intuitive and computable in polynomial time, median-based ordering is effectively a single algorithm that corresponds to a specific subset of the AOF parameter space. AOF effectively generalizes this definition; while median-timestamp sorting satisfies AOF under certain strict conditions, it fails to capture fair ordering in many other valid scenarios, and can be fragile against manipulation in others.

We analyze the relationship between AOF and Median-Timestamp (MT) to map the landscape of their equivalence. In valid parameter regions (e.g., $\theta \geq 2\Delta_{\text{sync}}$), MT acts as one valid implementation of AOF. However, AOF's flexibility allows it to cover cases where MT fails.

6.4.1 Theoretical Comparison in Honest Environments. We analyze the behavior in an all-honest setting to isolate the algorithmic differences.

- **Regime A: Median Implies AOF ($\theta \geq 2\Delta_{\text{sync}}, \varphi > 0$).** When the fairness delay is very large ($\geq 2\Delta_{\text{sync}}$), the AOF condition implies that the reception intervals of tx_A and tx_B are disjoint across the entire network. Specifically, the latest possible reception of tx_A is earlier than the earliest reception of tx_B. Consequently, $\text{median}(\text{tx}_A) < \text{median}(\text{tx}_B)$ is guaranteed. Here, MT naturally solves AOF.
- **Regime B: Strong Majority ($\theta \geq \Delta_{\text{sync}}, \varphi > 1/2$).** If a strict majority ($> 1/2$) receives tx_A significantly before tx_B (by Δ_{sync}), MT also succeeds. Any node i in this majority knows that all other nodes receive tx_B no earlier than $t_i(\text{tx}_B) - \Delta_{\text{sync}}$. Since $t_i(\text{tx}_A) \leq t_i(\text{tx}_B) - \theta \leq t_i(\text{tx}_B) - \Delta_{\text{sync}}$, the median timestamp of tx_B must be strictly larger than the median of tx_A.
- **Regime C: Small Delay Failure ($\theta \leq \Delta_{\text{sync}}$).** For smaller fairness delays, MT fails to recognize valid AOF orderings. We can construct profiles where a φ-quorum observes $\text{tx}_A \prec_\theta \text{tx}_B$, yet $\text{median}(\text{tx}_A) \geq \text{median}(\text{tx}_B)$. *Construction:* Partition nodes into 3 groups. Group 1 ($\varphi/2$) receives $(\text{tx}_A : t, \text{tx}_B : t + \theta)$. Group 2 ($\varphi/2$) receives $(\text{tx}_A : t + \Delta_{\text{sync}}, \text{tx}_B : t + \Delta_{\text{sync}} + \theta)$. Group 3 ($1 - \varphi$) receives $(\text{tx}_A : t + \Delta_{\text{sync}}, \text{tx}_B : t + \theta)$. Groups 1 and 2 satisfy the

AOF condition. However, median(tx_A) is close to $t+\Delta_{\text{sync}}$ while median(tx_B) is close to $t+\theta$. Since $\theta \leq \Delta_{\text{sync}}$, we inevitably get median(tx_A) $\geq$ median(tx_B), violating the constraint.

- **Regime D: Minority Signal Failure ($\Delta_{\text{sync}} < \theta \leq 2\Delta_{\text{sync}}, \varphi < 1/2$).** AOF allows a minority signal (if φ is low) to enforce order if the time gap is large. MT allows the majority to drown out this signal. If a minority φ sees a gap θ, but the majority sees a small gap, the medians will be pulled close together, potentially reversing the order defined by AOF.

6.4.2 Fragility and Manipulation. Beyond these theoretical gaps, the reliance on a single statistic (median) makes straightforward MT protocols fragile in adversarial settings. This vulnerability has been discussed extensively in [Kelkar et al. 2023], which demonstrates how easily Condorcet cycles can be manufactured or how medians can be manipulated. We illustrate this with a specific example relevant to our parameters.

- **Example of Fragility:** Consider 2 transactions $\mathsf{tx}_A, \mathsf{tx}_B$ and 5 nodes.
  - **Nodes 1, 2, 3:** Receive tx_A at 0.9 and tx_B at 1.0 (Gap 0.1).
  - **Nodes 4, 5:** Receive tx_B at 0.0 and tx_A at 1.9 (Gap 1.9, tx_B is much earlier).

  The median timestamp for tx_A is 0.9 (from $\{0.9, 0.9, 0.9, 1.9, 1.9\}$) and for tx_B is 1.0 (from $\{0.0, 0.0, 1.0, 1.0, 1.0\}$). Pompē enforces $\mathsf{tx}_A \prec \mathsf{tx}_B$ based on the majority's slight preference, completely ignoring the large gap observed by the minority where tx_B clearly precedes tx_A.
- **Manipulation Sensitivity:** In the above profile, if just one node among the first three is malicious and changes their report, they can dictate the relation. For instance, if Node 3 shifts their report slightly, they can flip the median. Previous works like Themis [Kelkar et al. 2023] have highlighted this class of manipulation attacks against median-based schemes; our analysis confirms this fragility persists even under our generalized model.
- **AOF Robustness:** In contrast, AOF avoids enforcing a fragile ordering. With a reasonable delay threshold (e.g., $\theta > 0.1$), the votes from Nodes 1, 2, 3 for $\mathsf{tx}_A \prec \mathsf{tx}_B$ are invalid because the gap (0.1) is insufficient. AOF would thus report no consistent ordering (or impose $\mathsf{tx}_B \prec \mathsf{tx}_A$ if the threshold φ allows the minority vote of 2/5 with sufficient gap). Critically, a single malicious node in the first group cannot force an ordering of $\mathsf{tx}_A \prec \mathsf{tx}_B$ because AOF requires a valid time gap from a sufficient number of nodes, filtering out the "noise" of the 0.1 gap.

6.4.3 Summary of Median-Timestamp Comparison. In conclusion, AOF acts as a strict superset of Median-Timestamp approaches. It validates the median approach where parameters align (Regimes A/B), but correctly identifies fair orderings where median aggregates fail (Regimes C/D), while providing superior resistance to noise and manipulation.

6.5 Case Study: Breaking Fairness Cycles

To illustrate the practical advantage of AOF, consider a scenario with three transactions $\mathsf{tx}_A, \mathsf{tx}_B, \mathsf{tx}_C$ and three nodes, where network jitter creates a Condorcet cycle in raw reception times. Suppose the reception times (normalized) are:

- Node 1: $\mathsf{tx}_A(0.1), \mathsf{tx}_B(0.5), \mathsf{tx}_C(0.9)$
- Node 2: $\mathsf{tx}_B(0.1), \mathsf{tx}_C(0.5), \mathsf{tx}_A(0.6)$
- Node 3: $\mathsf{tx}_C(0.1), \mathsf{tx}_A(0.2), \mathsf{tx}_B(0.9)$

Here, a majority (2/3) perceive $\mathsf{tx}_A \prec \mathsf{tx}_B$, $\mathsf{tx}_B \prec \mathsf{tx}_C$, and $\mathsf{tx}_C \prec \mathsf{tx}_A$.

- **Block-Order-Fairness (Aequitas):** Identifies the cycle and batches all three $\{\mathsf{tx}_A, \mathsf{tx}_B, \mathsf{tx}_C\}$ into a single block with arbitrary ordering. The granularity is lost.

- **Bounded-Unfairness (Taxis):** Must linearize the cycle while minimizing position displacement, potentially outputting an order like $\mathsf{tx}_A, \mathsf{tx}_C, \mathsf{tx}_B$ which violates the strong $\mathsf{tx}_A \rightarrow \mathsf{tx}_B$ signal.
- **AOF (Ours):** With a time threshold $\theta = 0.35$, AOF examines the time gaps. The edges $\mathsf{tx}_A \rightarrow \mathsf{tx}_B$ and $\mathsf{tx}_B \rightarrow \mathsf{tx}_C$ have gaps $\geq 0.4 > \theta$ and are preserved. However, the edge $\mathsf{tx}_C \rightarrow \mathsf{tx}_A$ relies on gaps of only $0.1 < \theta$. AOF correctly identifies this "return edge" as noise indistinguishable from jitter. It filters out $\mathsf{tx}_C \rightarrow \mathsf{tx}_A$, resolving the cycle into the strict linear order $\mathsf{tx}_A \prec \mathsf{tx}_B \prec \mathsf{tx}_C$.

This example demonstrates how AOF uses time magnitude to distinguish between statistically significant orderings and spurious cycles arising from network latency.

6.6 Partial Summary of Advantages

Table 1 summarizes some of the distinctions. AOF stands out as the only definition that provides strict pairwise ordering (preventing intra-block attacks), handles overlapping intervals (improving upon Timed-Order), and runs in polynomial time (unlike optimal Bounded-Unfairness).

Fairness Definition	Ordering Type	Granularity	Computational Complexity
Receive-Order-Fairness [Kelkar et al. 2020]	Strict Pairwise	Impossible	N/A
Block-Order-Fairness [Kelkar et al. 2020]	Block-Granularity	Coarse	Polynomial
Timed-Order-Fairness [Kursawe 2020]	Strict Pairwise	Limited (Non-overlapping only)	Polynomial
Median-Timestamp-Fairness [Zhang et al. 2020]	Strict Pairwise	Medium (Fragile)	Polynomial
Bounded-Unfairness [Kiayias et al. 2024]	Relaxed Pairwise	Fine (Parameter B)	NP-Hard (Optimal)
Approximate-Order-Fairness (Ours)	**Strict Pairwise**	**Fine (Parameter θ)**	**Polynomial**

Table 1. Comparison of Transaction Order Fairness Schemes

7 Theoretical Limits of Pruning Bad Reports

In this section, we analyze a theoretical variation where nodes perform an all-to-all broadcast of transaction timestamps to enable the pruning of "bad" reports. While we adopt the non-pruning approach for our main protocol due to its robustness and simplicity, analyzing the pruning case provides insight into the theoretical limits of AOF.

We assume a protocol where nodes perform an all-to-all broadcast of transaction timestamps. For each transaction tx, let the sorted timestamps be $\mathsf{ts}_{(1)}(\mathsf{tx}) \leq \cdots \leq \mathsf{ts}_{(n)}(\mathsf{tx})$. We define two reference times:

- $T_{low}(\mathsf{tx})$: The largest time such that more than $1-h$ fraction of nodes report a time at or after it. That is, $T_{low} = \mathsf{ts}_{(n-\lfloor (1-h)n \rfloor)}(\mathsf{tx})$. Since at least one honest node must be in this set, we know $\max_{i \in \mathcal{H}} \mathsf{ts}_i(\mathsf{tx}) \geq T_{low}(\mathsf{tx})$. Combining with the synchrony bound, the earliest valid honest report must be at least $T_{low}(\mathsf{tx}) - \Delta_{\mathsf{sync}}$.
- $T_{high}(\mathsf{tx})$: The smallest time such that more than $1-h$ fraction of nodes report a time at or before it. That is, $T_{high} = \mathsf{ts}_{(\lfloor (1-h)n \rfloor + 1)}(\mathsf{tx})$. Similarly, this implies $\min_{i \in \mathcal{H}} \mathsf{ts}_i(\mathsf{tx}) \leq T_{high}(\mathsf{tx})$, so the latest valid honest report is at most $T_{high}(\mathsf{tx}) + \Delta_{\mathsf{sync}}$.

A report t is valid only if $t \in [T_{low}(\mathsf{tx}) - \Delta_{\mathsf{sync}}, T_{high}(\mathsf{tx}) + \Delta_{\mathsf{sync}}]$.

Definitions and Adversary Strategies. Throughout this section, we construct cyclic strategies to prove infeasibility. We define:

- **Mass:** The fraction of the total node population assigned to a specific reporting group.
- **Support:** The weighted fraction of transaction pairs $(\mathsf{tx}_i, \mathsf{tx}_{i+1})$ for which a node reports $t_{i+1} \geq t_i + \theta$.

We analyze three specific adversarial strategies S_1, S_2, S_3. In all cases, honest nodes form $k+1$ groups reporting $\{0, \theta, \ldots, k\theta\}$.

- **Strategy S_1 (Baseline):** Adversary reports $\{0, \ldots, k\theta\}$. Always valid. Support $\approx \frac{k}{k+1}$.
- **Strategy S_2 (Anchored):** Adversary reports $\{-0.5\theta, \ldots, (k+0.5)\theta\}$. Requires validity of -0.5θ. Support $\approx \frac{k+1}{k+2}$.
- **Strategy S_3 (Aggressive):** Adversary reports $\{-\theta, \ldots, (k+1)\theta\}$. Requires validity of $-\theta$. Support $\approx \frac{k+2}{k+3}$.

Conjecture of Tight Feasible Bounds. Based on the optimal strategies, we observe that for $h \le H_{crit}$, the adversarial support $\Phi_{opt} \ge h$, rendering fairness impossible (since we require $\varphi \le h$ for honest subset enforceability). Thus, feasible regions only exist for high h.

(1) **High h ($h > H_{crit}$), $\theta > \Delta_{\text{sync}}/(k+0.5)$**: Feasible if $\varphi > \frac{k}{k+1}$.
(2) **High h ($h > H_{crit}$), $\theta > \Delta_{\text{sync}}/(k+1)$**: Feasible if $\varphi > h\frac{k}{k+1} + (1-h)\frac{k+1}{k+2}$.

Theorem 7.1. *(Infeasible parameters - Trivial Bounds) Regardless of θ, AOF is always infeasible if $\varphi \le 1-h$, as the adversary possesses sufficient voting power to impose arbitrary constraints. Additionally, parameter settings where $\varphi > h$ are considered impossible, as honest nodes alone cannot enforce the fairness condition against a non-participating adversary.*

Theorem 7.2. *(Infeasible parameters - Disjoint Optimal Strategy) For any $\theta \le \frac{\Delta_{\text{sync}}}{k}$, AOF is infeasible for $\varphi < \Phi_{opt}$. The threshold $H_{crit} = \frac{(k+1)^2}{k^2+3k+3}$ separates the regime where fairness is theoretically impossible from the regime where it is merely constrained.*

(1) ***Low h ($h \le H_{crit}$):*** *The adversary uses strategy S_2, achieving support $\Phi_{opt} \ge h$. Since any protocol requires $\varphi \le h$, this renders fairness impossible for all valid φ.*
(2) ***High h ($h > H_{crit}$):*** *The adversary falls back to strategy S_1.*

$$\Phi_{opt} = \frac{k}{k+1}$$

Proof. Setting $\Delta_{\text{sync}} = k\theta$, the adversary chooses between S_2 (support $\frac{k+1}{k+2}$) and S_1 (support $\frac{k}{k+1}$). Strategy S_2 reports -0.5θ, requiring $T_{low} \le (k-0.5)\theta$. This requires the mass of reports strictly greater than $(k-0.5)\theta$ to be $\le 1-h$. This tail mass consists of honest nodes at $k\theta$ (fraction $\frac{h}{k+1}$) and adversary S_2 nodes at $(k+0.5)\theta$ (fraction $\frac{1-h}{k+2}$). The validity condition is:

$$\frac{h}{k+1} + \frac{1-h}{k+2} \le 1-h$$

Solving for h, we get $h(k^2+3k+3) \le (k+1)^2$, or $h \le H_{crit}$. The support of S_2 mixed with honest reports is $\Phi_{S_2} = h\frac{k}{k+1} + (1-h)\frac{k+1}{k+2}$. Algebraic verification shows that $\Phi_{S_2} \ge h$ exactly when $h \le H_{crit}$. Thus, for $h \le H_{crit}$, no feasible $\varphi \le h$ exists that exceeds the adversarial support. For $h > H_{crit}$, S_2 is invalid. The adversary is restricted to S_1, yielding $\Phi = \frac{k}{k+1} < h$. □

Theorem 7.3. *(Infeasible parameters - Continuous Optimization) For any $\theta \le \frac{\Delta_{\text{sync}}}{k+0.5}$, since $H_2 = H_{crit}$, for all $h \le H_2$, the optimal adversarial support satisfies $\Phi_{opt} \ge h$. Thus, fairness is impossible for $h \le H_2$. For $h > H_2$, the adversary is restricted to strategy S_2, with support $\Phi_{opt} = h\frac{k}{k+1} + (1-h)\frac{k+1}{k+2}$.*

Proof. The analysis from Theorem 7.2 applies here with tighter constraints on θ that allow for strategy mixing. However, since the threshold for pure S_2 validity is the same ($H_2 = H_{crit}$), and any mixing with the more aggressive S_3 only increases support, the condition $\Phi_{opt} \ge h$ holds for the entire range where S_2 or stronger strategies are valid. Specifically, we verified that for $h \le H_{crit}$, $\Phi(S_2) \ge h$. Since S_3 provides strictly higher support, the optimal mix also satisfies $\Phi \ge h$. Thus, strictly feasible regions only exist for $h > H_2$. □

8 Conclusion

In this paper, we revisited and rigorously characterized Approximate-Order-Fairness (AOF), whose feasibility was previously dismissed in prior literature. We showed that AOF provides a practical middle ground between coarse-grained block-order fairness and computationally expensive bounded unfairness. By leveraging network synchrony and introducing a time delay parameter θ, we showed that it is possible to achieve strict pairwise ordering for transactions that are sufficiently separated in time, thereby eliminating Condorcet cycles for feasible parameters.

We provided a comprehensive analysis of the feasible and infeasible parameter regions for both non-pruning and pruning protocols. We showed that pruning bad reports can improve the theoretical resilience of the system, allowing for tighter fairness guarantees with lower voting thresholds. We presented practical protocols, including an all-to-all broadcast mechanism and a more efficient leader-based approach, that implement AOF with polynomial complexity.

Future work includes extending AOF to partially synchronous networks where the bound Δ_{sync} is unknown or variable, and investigating the incentives for rational actors to adhere to fair ordering in decentralized exchanges. Additionally, implementing AOF in a high-throughput blockchain and evaluating its impact on MEV (Maximal Extractable Value) mitigation remains an important practical milestone.

Acknowledgments

This work was partially supported by the ERC Starting Grant 101222524 (SPES). Z. Cai was supported by the Hong Kong PhD Fellowship Scheme. Authors are ordered alphabetically.

References

Ittai Abraham, Dahlia Malkhi, Kartik Nayak, Ling Ren, and Alexander Spiegelman. 2020. Sync HotStuff: Simple and Practical Synchronous Consensus. In *IEEE S&P*.

Kenneth J. Arrow. 1951. *Social Choice and Individual Values*.

Zhuo Cai and Amir Goharshady. 2026. Brief Announcement: Delay-Optimal Transaction Order Fairness. In *PODC*.

Miguel Castro and Barbara Liskov. 1999. Practical Byzantine Fault Tolerance. In *OSDI*.

Arka Rai Choudhuri, Soubhik Deb, Sanjam Garg, Ari Juels, and Sreeram Kannan. 2022. Blind Order Fairness. *CoRR* abs/2203.02844 (2022).

Arka Rai Choudhuri, Sanjam Garg, Julien Piet, and Spencer Peters. 2024. Mempool Privacy via Batched Threshold Encryption: Attacks and Defenses. In *USENIX Security*.

Philip Daian, Steven Goldfeder, Tyler Kell, Yunqi Li, Xueyuan Zhao, Iddo Bentov, Ari Juels, and Ari Juels. 2019. Flash Boys 2.0: Frontrunning in Decentralized Exchanges, Miner Extractable Value, and Consensus Instability. In *IEEE S&P*.

George Danezis, Lefteris Kokoris-Kogias, Alberto Sonnino, and Alexander Spiegelman. 2022. Narwhal and Tusk: A DAG-based Mempool and Efficient BFT Consensus. In *EuroSys*.

Marie Jean Antoine Nicolas de Caritat Condorcet. 1785. *Essai sur l'application de l'analyse 'a la probabilité des décisions rendues 'a la pluralité des voix*.

Cynthia Dwork, Nancy A. Lynch, and Larry Stockmeyer. 1988. Consensus in the presence of partial synchrony. *J. ACM* (1988).

Elijah Fox, Mallesh Pai, and Max Resnick. 2024. Censorship Resistance in On-Chain Auctions. In *FC*.

Yossi Gilad, Rotem Hemo, Silvio Micali, Georgios Vlachos, and Nickolai Zeldovich. 2017. Algorand: Scaling Byzantine Agreements for Cryptocurrencies. In *SOSP*.

Bingyu Guo, Zhenliang Lu, Qiang Tang, Jing Xu, and Zhenfeng Zhang. 2020. Dumbo: Faster Asynchronous BFT Consensus. In *CCS*.

Lioba Heimbach, Lucianna Kiffer, Christof Ferreira Torres, and Christos Sodakis. 2023. Ethereum's Proposer-Builder Separation: Promises and Realities. In *ACM IMC*.

Idit Keidar, Eleftherios Kokoris-Kogias, Oded Naor, and Alexander Spiegelman. 2021. All You Need Is DAG. In *PODC*.

Mahimna Kelkar, Soubhik Deb, Sreeram Kannan, Ari Juels, and Visweshkkumar R. Krishnakkumar. 2023. Themis: Fast, Strong Order-Fairness in Byzantine Consensus. In *CCS*.

Mahimna Kelkar, Fan Zhang, Steven Goldfeder, and Ari Juels. 2020. Order-Fairness for Byzantine Consensus. In *CRYPTO*.

Aggelos Kiayias, Nikos Leonardos, and Yu Shen. 2024. Ordering Transactions with Bounded Unfairness: Definitions, Complexity and Constructions. In *EUROCRYPT*.

Klaus Kursawe. 2020. Wendy, the Good Little Fairness Widget. In *AFT*.

Bruno Mazorra, Moises Reynolds, and Vanesa Daza. 2022. Price of MEV: Towards a Game Theoretical Approach to MEV. In *ACM CCS*.

Andrew Miller, Yu Xia, Kyle Croman, Elaine Shi, and Dawn Song. 2016. The Honey Badger of BFT Protocols. In *CCS*.

Wanyi Mu, Xinyu Yang, Tjerand Silde, and Sheng Wang. 2024. Separation is Good: A Faster Order-Fairness Byzantine Consensus. In *NDSS*.

Julian Ramseyer and Ashish Goel. 2024. Speed, Fairness, and Safety: Tradeoffs in Order-Fair Consensus. In *PODC*.

Alexander Spiegelman, Alberto Sonnino, Dahlia Malkhi, and Lefteris Kokoris-Kogias. 2022. Bullshark: DAG BFT Protocols Made Practical. In *CCS*.

Maofan Yin, Dahlia Malkhi, Michael K. Reiter, Guilan G. Gueta, and Ittai Abraham. 2019. HotStuff: BFT Consensus in the Lens of Blockchain. In *PODC*.

Yunhao Zhang, Srinath Setty, Qi Chen, Lidong Zhou, and Lorenzo Alvisi. 2020. Byzantine Ordered Consensus without Byzantine Oligarchy. In *OSDI*.