

Buying Time: Latency Racing vs. Bidding for Transaction Ordering

Akaki Mamageishvili ✉

Offchain Labs, Zürich, Switzerland

Mahimna Kelkar¹ ✉

Cornell University, New York, NY, USA

Jan Christoph Schlegel ✉

City, University of London, UK

Edward W. Felten ✉

Offchain Labs, Washington, D.C., USA

Abstract

We design **TimeBoost**: a practical transaction ordering policy for rollout sequencers that takes into account both transaction timestamps and bids; it works by creating a score from timestamps and bids, and orders transactions based on this score.

TimeBoost is transaction-data-independent (i.e., can work with encrypted transactions) and supports low transaction finalization times similar to a first-come first-serve (FCFS or pure-latency) ordering policy. At the same time, it avoids the inefficient latency competition created by an FCFS policy. It further satisfies useful economic properties of first-price auctions that come with a pure-bidding policy. We show through rigorous economic analyses how **TimeBoost** allows players to compete on arbitrage opportunities in a way that results in better guarantees compared to both pure-latency and pure-bidding approaches.

2012 ACM Subject Classification Theory of computation → Algorithmic game theory

Keywords and phrases Transaction ordering, First-come-first-serve, First-price auctions

Digital Object Identifier 10.4230/LIPIcs.AFT.2023.23

Related Version *Full Version*: <https://arxiv.org/abs/2306.02179>

Acknowledgements We are grateful to Lee Bousfield, Chris Buckland, Potuz Heluani, Raul Jordan, Mallesh Pai, Ron Siegel, Terence Tsao as well as participants at the Swiss National Bank Technology and Finance Seminar for interesting discussions and valuable feedback.

1 Introduction

Transaction ordering is critically important for financial systems – the order in which user transactions are executed can directly impact the profits made by users. This motivates the study of designing transaction ordering policies with useful properties.

In this work, we focus on ordering policies for *centralized sequencers* – meaning that a single *sequencer* receives transactions from users and publishes an ordered sequence to be used for execution. A transaction ordering policy here specifies how the resulting output sequence depends on the contents and arrival times of transactions at the sequencer. Our work provides rigorous economic analyses to justify the utility of our proposed policy.

¹ This work was completed in the author’s role at Offchain Labs.



Why consider a centralized sequencer? In addition to the centralized sequencer setting being a potentially simpler model to study as a first step, there are two other main reasons why we choose to do so in this work:

1. *Existing use-cases are already centralized.* Decentralized blockchains such as Ethereum are still *ephemerally centralized* with respect to ordering – for a given block, similar to a centralized sequencer, only a single miner/validator is in complete control of the inclusion and ordering of transactions within the block. Similarly, current layer-2 “rollup” protocols (such as Arbitrum and Optimism) also employ a centralized sequencer to order transactions in a batch posted to the underlying Ethereum base-chain.
2. *Ordering policies are mostly orthogonal to the problem of sequencer decentralization.*

While decentralizing the sequencer is an important active research direction, we note that a suitable transaction ordering policy can be chosen orthogonally to the method of sequencer decentralization. In particular, the decentralized protocol can first be used to agree on single *pre-ordering* or *scoring* of transactions, following which a specific ordering policy can be applied. In other words, the output of the decentralized protocol can be thought of simulating the input of a virtual centralized sequencer on which the ordering policy gets applied.

An example of this is seen in the recent line of works on fair-ordering [3,8,9,11,20] – they can be thought of as a decentralized implementation of a first-come-first-serve ordering policy which combines local transaction orderings from many nodes.

Furthermore, while current centralized sequencer implementations are semi-trusted in that they receive transactions in plaintext and are expected not to deviate from the specified ordering policy or insert transactions of their own, we note that transaction data can be hidden from the sequencer by using threshold decryption by a committee (i.e., the sequencer only sees encrypted transactions and orders them, only after which a committee decrypts the plaintext) or trusted hardware (such as Intel SGX). Through these techniques, the adversarial behavior of the sequencer can be substantially restricted.

The study of ordering policies is important even when the sequencer is trusted (or is suitably constrained as mentioned above) due to the presence of other profit-seeking entities in the system. For instance, after the sequencer publishes state after execution of previous transaction(s), arbitrage opportunities can be created; players in the system will compete with each other to take advantage of these opportunities. Similar situations can also arise due to state updates from external systems.

1.1 Existing Ordering Policies

Ordering policies used on blockchains today fall roughly into three categories described below.

First-come first-serve (FCFS). One natural ordering policy is the first-come, first-serve (FCFS) rule. Here, transactions are sequenced in the same order that they were received from users. There are several advantages to FCFS: to begin, it is simple to implement and seems intuitively fair – after all, it is a commonly used policy even for real-world interactions. FCFS also minimizes transaction latency: transactions can be continuously sequenced as they arrive, and do not need to conform to the discrete granularity of blocks. The sequencer in the layer-2 rollup Arbitrum employs an FCFS policy.

One major disadvantage of FCFS however, is that creates *latency competition* in the sense that entities are incentivized to position themselves as close to the sequencer as possible in order to be the first to react to any new market information. This is a well known and studied

problem within traditional financial systems. Indeed, high frequency trading (HFT) firms invest millions of dollars into low-latency infrastructure that can operate sub-microsecond or even finer scales; their trading accounts for roughly half of all trading volume [13]. This inclination to latency investment is highly inefficient since the investment happens externally to the system (as opposed to bidding; see below) and therefore cannot be used beneficially within the system. Recent works [1, 17] have also shown the potential for similar strategic manipulation within a pure FCFS protocol in the decentralized setting.

One crucial point to emphasize here is that this latency competition in FCFS *does not disappear even if transaction data is hidden* (e.g., transactions are encrypted). This is because any state changes (from the sequencer or even from external systems) can trigger a profit opportunity wherein it is beneficial to have the quickest access to the sequencer. As a specific example, an update on the trading price of a token can create an arbitrage opportunity whose profit will go only to the player who can submit its transaction to the sequencer first². This kind of latency-based arbitrage has already been seen in Arbitrum, which implements a centralized FCFS sequencer.

Per-block transaction bidding. A second natural policy is to group transactions into blocks, then order transactions within a block based on their *bid*. Specifically, each transaction is submitted along with a fee or bid; the sequencer now collects all transactions submitted within some time interval and sequences them by the descending order of their bids. This essentially simulates a first-price all-pay auction [10] (i.e., players bid independently; the highest bid wins but all players need to pay their bid amount) to take advantage of a particular arbitrage opportunity. Since players submit their bids independently, the bidding policy can work as expected even when transactions are encrypted (since state or market updates create arbitrage opportunities).

One advantage of a bidding policy (compared to FCFS) is that the payment is internal to the system and therefore can be utilized within it to e.g., subsidize protocol operation costs.

When the block-time is large (e.g., 12s as in Ethereum), it is expected that for almost all arbitrage opportunities, all interested players can post their bid within the time interval in an attempt to take advantage of the opportunity. However, when the block-time is small (this is typically the case in layer-2 protocols to increase scalability), perhaps surprisingly, having a connection with lower latency can provide a substantial advantage. This is because when the market update happens close to end of the block time, only players with a faster connection will be able get their transaction included in the block; consequently, they may be able to take advantage of the arbitrage opportunity with a smaller (or even a zero) bid.

Looking ahead, our TimeBoost policy (which combines both arrival times and bidding) will enable arbitrageurs to prefer bidding even when block times are small, thereby allowing the protocol to capture this value rather than it being lost to external latency infrastructure.

Block or MEV auctions. A third widely-used policy auctions off the complete rights to choose and order transactions within a block. Here, the sequencer does not order transactions itself but rather accepts block proposals from external players (often called block *builders*) and chooses the proposal from the builder who pays the most. These auctions initially arose from the realization that significant profit (often referred to as maximal (previously miner) extractable value or MEV [2, 6]) can be extracted by manipulating the ordering of user

² Another approach if the sequencer broadcasts state information in a random order to clients is to create many dummy client copies, thereby increasing the chances that some copy gets the feed faster.

transactions. In the past two years, through companies such as Flashbots and Bloxroute, an MEV marketplace has been created on Ethereum *outside of the protocol* to connect block proposers (entities in charge of proposing or sequencing a block) to block builders (players who find MEV opportunities and order user transactions to take advantage of them) – the result has been the extraction of hundreds of millions of dollars in profit from user transactions [15, 18].

While some MEV (such as arbitrage, which provides incentives for price discovery) is benign and can be done without the knowledge of user transactions, other forms of MEV extraction crucially rely on the transaction data. Recent works [18, 19] have shown such MEV to be significantly detrimental to users. The emergence of such MEV extraction has largely been attributed to the rationality of block proposers as well as the lack of regulation. For example, in traditional financial systems, it is often illegal or at the very least heavily constrained to profit from the knowledge of user transactions (for instance, payment-for-order-flow (PFOF): the selling of user transaction data is illegal in the UK, and, while legal in the US, still requires users to be provided with guarantees of “best execution”).

A design goal for our work is therefore to design ordering policies that are data-independent, i.e., they do not use transaction data for ordering. This will allow them to be used even when transactions are encrypted at the time of sequencing.

1.2 Our contributions

TimeBoost: An ordering policy that combines FCFS and bidding. We propose TimeBoost, an ordering policy that combines both FCFS-style timestamps and first-price auction style bids. Below, we describe several natural goals that went into our design.

1. **Data independence.** The policy should not utilize the transaction data for ordering. This is a natural goal in order to support encrypted transactions and prevent data-dependent MEV attacks on transaction ordering.
2. **Low finalization time.** The policy should be able to sequence transactions within a short time g (the specific parameter can be set according to the application). This is important to improve the user experience with the system since transactions will be sequenced within time g after they are received.
3. **Independence of irrelevant transactions.** The ordering between two given transactions should not depend on the presence of other transactions. This is useful to prevent an adversary from inserting irrelevant transactions that results in flipping the ordering between two target transactions. Importantly, this property also ensures that a transaction submitter’s strategy need only consider transactions that are relevant to the party’s goals – for example if Alice is trying to capture a particular arbitrage opportunity, she need only worry about other transactions affecting that opportunity.
4. **Inclination to spending via bids instead of latency infrastructure.** As mentioned before, investments into latency infrastructure are highly inefficient from the system standpoint since the value spent cannot be utilized effectively by the system. Therefore, a natural goal is to disincentivize latency investment and instead incentivize players to bid for their transactions. Looking ahead, perhaps surprisingly, we find that the pure bidding policy results in a larger latency competition than our TimeBoost policy which combines bidding with FCFS style timestamps.

TimeBoost details. Intuitively, TimeBoost works by assigning *scores* to transactions based on both their arrival times and their bid. The final ordering is taken to be descending in the transaction scores. More specifically, for a transaction with arrival time t and bid b ,

TimeBoost assigns it the score $S(\text{tx}) = \pi(b) - t$ where intuitively π represents a function for “buying time” – by increasing the transaction bid, users can reduce their effective timestamp (or equivalently, increase their score). Section 3 describes how to choose the function π .

Importantly, there is a limit to how much time can be “bought” through the bid – in particular, no transaction can outbid a transaction received some g time earlier. Such a property is required to ensure the quick finalization of user transactions. At the same time, transactions received less than g time before can always be outbid; this means that arbitrageurs always have g time to compete for any arbitrage opportunity as opposed to a pure bidding policy and will therefore prefer bidding over latency infrastructure investments.

We also show that TimeBoost satisfies all the useful economic properties of first-price all-pay auctions. Further, we show that players spend exactly the same amount in total with TimeBoost, as they would spend if only latency investment was allowed, except that most of the investment is done through bidding and therefore can be captured within the protocol for e.g., lowering user fees or for protocol development.

2 Ordering Policies

2.1 Preliminaries

A transaction tx that arrives at the sequencer can be characterized by a tuple (data, t, b) where data represents the transaction data, t denotes the arrival time, and b denotes the transaction bid (note that when transactions are of different sizes, b can be instead be considered to be a bid per unit size). Let \mathcal{T} denote the set of all possible transactions; in principle this can be infinite or even uncountable (e.g., if arrival times are in \mathbb{R}^+) and our results do hold for these cases. For practical use-cases, typically, arrival times can be assumed to be in \mathbb{Q}^+ and bids can be assumed to be in $\mathbb{N}^{\geq 0}$.

An ordering policy now defines how a sequencer orders a finite set \mathcal{T}' of transactions that it has received. A formal definition is given below:

► **Definition 1** ((Data-Independent) Ordering Policy). *An ordering policy (or algorithm) \mathbb{P} takes as input a finite subset $\mathcal{T}' \subseteq \mathcal{T}$ of transactions and outputs a linear ordering $\mathbb{P}(\mathcal{T}')$. For $\text{tx} \in \mathcal{T}'$, let $\mathbb{P}(\mathcal{T}', \text{tx})$ denote the position of transaction tx in the ordering $\mathbb{P}(\mathcal{T}')$. In other words, given \mathcal{T}' and $\text{tx}_a, \text{tx}_b \in \mathcal{T}'$, \mathbb{P} outputs tx_a before tx_b if $\mathbb{P}(\mathcal{T}', \text{tx}_a) < \mathbb{P}(\mathcal{T}', \text{tx}_b)$.*

A policy is further called data-independent if it does not make use of the transaction data (i.e., it only uses the arrival time and the bid).

Since we want our ordering policies to not be based on the transaction content, we only consider data-independent policies for the rest of the paper. For simplicity, we can therefore represent a transaction tx simply by the tuple $(\text{tx}.t, \text{tx}.b)$. Furthermore, since ties can be broken by some chosen technique, without loss of generality, we can also assume $(\text{tx}.t, \text{tx}.b)$ tuples are unique. While the tie-breaking can be dependent on e.g., transaction ciphertext or metadata, this does not affect our analysis and therefore can be safely ignored for the purpose of our paper.

2.2 Independence of Irrelevant Transactions (IIT)

A useful property for our ordering policy to have is to prevent the ordering decision between transactions tx_a and tx_b to change depending on what other transactions are being ordered; in other words, the ordering decision should not depend on irrelevant transactions. Intuitively, this is done to ensure that an adversary cannot create dummy transactions in order to flip

the ordering decision between two transactions, and so that a party's bidding strategy can ignore transactions irrelevant to that party. We define this property of independence of irrelevant transactions (IIT) below.

► **Definition 2** (Independence of Irrelevant Transactions). *We say that a policy \mathbb{P} satisfies independence of irrelevant transactions (IIT) if for any pair of transactions $\mathbf{tx}_a, \mathbf{tx}_b$ and any pair of finite subsets $\mathcal{T}_1, \mathcal{T}_2 \subset \mathcal{T}$, the following holds:*

$$\begin{aligned} & \mathbb{P}(\{\mathbf{tx}_a, \mathbf{tx}_b\} \cup \mathcal{T}_1, \mathbf{tx}_a) < \mathbb{P}(\{\mathbf{tx}_a, \mathbf{tx}_b\} \cup \mathcal{T}_1, \mathbf{tx}_b) \\ \Leftrightarrow & \mathbb{P}(\{\mathbf{tx}_a, \mathbf{tx}_b\} \cup \mathcal{T}_2, \mathbf{tx}_a) < \mathbb{P}(\{\mathbf{tx}_a, \mathbf{tx}_b\} \cup \mathcal{T}_2, \mathbf{tx}_b). \end{aligned}$$

2.3 IIT Implies a Score-Based Policy

We now show that the IIT property implies that a score-based policy needs to be used – that is, also needs to be independent of the set \mathcal{T}' being ordered.

Intuitively, a score-based policy works as follows: for transaction \mathbf{tx} , it assigns a score $S(\mathbf{tx})$ based only on the arrival time $\mathbf{tx}.t$ and the bid $\mathbf{tx}.b$. Here too, scoring ties can be broken in a pre-specified manner. The output sequence is then taken to the descending order of transaction scores. Score-based policies are formally defined below:

► **Definition 3** (Score-based policy). *A score is a function $S : \mathcal{T} \rightarrow \mathbb{R}$ that assigns to each possible transaction $\mathbf{tx} \in \mathcal{T}$ a score $S(\mathbf{tx})$. An ordering policy \mathbb{P} is called score-based if there exists a score function S such that \mathbb{P} sorts transactions according to S . In other words, there exists S such that for any $\mathcal{T}' \subseteq \mathcal{T}$ and $\mathbf{tx}_a, \mathbf{tx}_b \in \mathcal{T}'$, it holds that $\mathbb{P}(\mathcal{T}', \mathbf{tx}_a) < \mathbb{P}(\mathcal{T}', \mathbf{tx}_b)$ if and only if $S(\mathbf{tx}_a) > S(\mathbf{tx}_b)$.*

For finite \mathcal{T} , we can directly show that IIT implies score-based policies. To show the result for infinite sets, we need to employ the following set-theoretic axiom (defined below) by Cantor [4]. Similar definitions have also been used in the context of utility theory [7]

► **Property 4** (Cantor's Axiom [4]). *We say that a pair $(\mathbb{P}, \mathcal{T})$ satisfies Cantor's axiom if there exists a countable set $\mathcal{T}' \subseteq \mathcal{T}$ such that for any pair of transactions $\mathbf{tx}_a, \mathbf{tx}_b \in \mathcal{T}$ there exists an instance of \mathbb{P} in which some transaction in \mathcal{T}' is ordered between \mathbf{tx}_a and \mathbf{tx}_b .*

Formally there is a finite set $\mathcal{T}'' \subset \mathcal{T}$ with $\mathbf{tx}_a, \mathbf{tx}_b \in \mathcal{T}''$ and a $\mathbf{tx}_c \in \mathcal{T}' \cap \mathcal{T}''$ (possibly $\mathbf{tx}_c = \mathbf{tx}_a$ or $\mathbf{tx}_c = \mathbf{tx}_b$) such that

$$\mathbb{P}(\mathcal{T}'', \mathbf{tx}_a) \leq \mathbb{P}(\mathcal{T}'', \mathbf{tx}_c) \leq \mathbb{P}(\mathcal{T}'', \mathbf{tx}_b),$$

or

$$\mathbb{P}(\mathcal{T}'', \mathbf{tx}_b) \leq \mathbb{P}(\mathcal{T}'', \mathbf{tx}_c) \leq \mathbb{P}(\mathcal{T}'', \mathbf{tx}_a).$$

We can now establish the following correspondence between IIT and score-based policies.

► **Theorem 5** (IIT \Leftrightarrow Score-Based). *Let \mathcal{T} denote the set of all transactions. The following hold for any ordering policy \mathbb{P} :*

1. *If \mathcal{T} is countable, then \mathbb{P} satisfies IIT if and only if it is score-based.*
2. *If \mathcal{T} is uncountable and $(\mathbb{P}, \mathcal{T})$ satisfies Cantor's axiom, then \mathbb{P} satisfies IIT if and only if it is score-based.*

Proof. It is straightforward to see that a score-based algorithm satisfies the independence of irrelevant transactions (since the score of a transaction depends only on itself and not other transactions).

For the opposite direction, we first prove the second part of the theorem (the uncountable case). We define an order \prec over \mathcal{T} where

$$\mathbf{tx}_a \prec \mathbf{tx}_b :\Leftrightarrow \mathbb{P}(\{\mathbf{tx}_a, \mathbf{tx}_b\}, \mathbf{tx}_a) < \mathbb{P}(\{\mathbf{tx}_a, \mathbf{tx}_b\}, \mathbf{tx}_b).$$

Since $\mathbb{P}(\{\mathbf{tx}_a, \mathbf{tx}_b\})$ is a well-defined for any two transactions $\mathbf{tx}_a, \mathbf{tx}_b \in \mathcal{T}$, the order \prec is complete and anti-symmetric. By independence and since $\mathbb{P}(\{\mathbf{tx}_a, \mathbf{tx}_b, \mathbf{tx}_c\})$ is a well-defined order for any three transactions $\mathbf{tx}_a, \mathbf{tx}_b, \mathbf{tx}_c \in \mathcal{T}$ we have

$$\begin{aligned} & \mathbf{tx}_a \prec \mathbf{tx}_b \prec \mathbf{tx}_c \\ \Rightarrow & (\mathbb{P}(\{\mathbf{tx}_a, \mathbf{tx}_b\}, \mathbf{tx}_a) < \mathbb{P}(\{\mathbf{tx}_a, \mathbf{tx}_b\}, \mathbf{tx}_b) \text{ and } \mathbb{P}(\{\mathbf{tx}_b, \mathbf{tx}_c\}, \mathbf{tx}_b) < \mathbb{P}(\{\mathbf{tx}_b, \mathbf{tx}_c\}, \mathbf{tx}_c)) \\ \Rightarrow & \mathbb{P}(\{\mathbf{tx}_a, \mathbf{tx}_b, \mathbf{tx}_c\}, \mathbf{tx}_a) < \mathbb{P}(\{\mathbf{tx}_a, \mathbf{tx}_b, \mathbf{tx}_c\}, \mathbf{tx}_b) < \mathbb{P}(\{\mathbf{tx}_a, \mathbf{tx}_b, \mathbf{tx}_c\}, \mathbf{tx}_c) \\ \Rightarrow & \mathbb{P}(\{\mathbf{tx}_a, \mathbf{tx}_c\}, \mathbf{tx}_a) < \mathbb{P}(\{\mathbf{tx}_a, \mathbf{tx}_c\}, \mathbf{tx}_c) \\ \Rightarrow & \mathbf{tx}_a \prec \mathbf{tx}_c \end{aligned}$$

Therefore, \prec is transitive. We let $\mathbf{tx}_a \preceq \mathbf{tx}_b$ iff $\mathbf{tx}_a \prec \mathbf{tx}_b$ or $\mathbf{tx}_a = \mathbf{tx}_b$.

The Cantor axiom and independence imply that there is a countable $\mathcal{T}' \subset \mathcal{T}$ so that the order \prec satisfies that for any $\mathbf{tx}_a, \mathbf{tx}_b \in \mathcal{T}$ there is a $\mathbf{tx}_c \in \mathcal{T}'$ such that

$$\mathbf{tx}_a \prec \mathbf{tx}_b \Rightarrow \mathbf{tx}_a \preceq \mathbf{tx}_c \preceq \mathbf{tx}_b$$

By Theorem 1.1 in [5], this, in turn, implies that there is a numerical representation of the order \prec which is a score $S : \mathcal{T} \rightarrow \mathbb{R}$ such that for any two transactions $\mathbf{tx}_a, \mathbf{tx}_b \in \mathcal{T}$ we have $\mathbf{tx}_a \prec \mathbf{tx}_b$ if and only if $S(\mathbf{tx}_a) > S(\mathbf{tx}_b)$.

For the first part of the theorem, note that the previous argument also works for a countable \mathcal{T} and in that case we can choose $\mathcal{T}' = \mathcal{T}$ where the Cantor axiom is now trivially satisfied. \blacktriangleleft

► **Remark 6.** The above result extends to the case where the policy creates a weak ordering (which can be made strict through a tie-breaking procedure) rather than a strict ordering of transactions. In that case, Definitions 2 and 3 are adapted to weak orders, and we get a score that might assign the same value to two different transactions. The relaxation to weak orders is useful for the case that the set of transactions is uncountable and not a subset of the real numbers (e.g. if $\mathcal{T} = \mathbb{R}_+^2$). In that case, the Cantor axiom is impossible to satisfy for strict orders but satisfiable for weak orders.

Discussion. We note that in our context, assuming \mathcal{T} is countable or even finite is safe, as there is a finite smallest time increment for timestamps and a finite smallest bid increment. Moreover, the ordering policy deals with ordering transactions in a finite time interval and bids will be upper-bounded by the maximum value in the system (e.g., the maximum number of tokens). However, for the subsequent economic analysis, it will be more convenient to work with the continuum where differences in time stamps and bids can be arbitrarily small.

Having proven that score-based algorithms are essentially the only ones satisfying the independence of irrelevant transactions property, we turn to selecting the most natural one among them. Note that FCFS is the scoring function that corresponds to scoring transactions by their timestamp only while scoring transactions only by bids corresponds to the first-price auction solution. In the next section, we show how our scoring policy TimeBoost corresponds to a simple mixture of these two strategies.

3 TimeBoost Description

We now formally define the TimeBoost ordering policy in this section. As mentioned before, we want TimeBoost to satisfy the independence of irrelevant transactions property (i.e., it needs to be a scoring function based on Theorem 5) and also provide low confirmation-latency for transactions. Therefore, we will only allow TimeBoost to consider transactions within a time g interval; this granularity g can be set suitably based on the particular usecase.

Basic model. Suppose there are n transactions in the g time interval, labeled with $\text{tx}_1, \text{tx}_2, \dots, \text{tx}_n$, and sorted by increasing arrival time. Each transaction tx_i is characterized by a pair of a timestamp or arrival time, denoted by t_i , and a bid, denoted by $b_i \geq 0$. Formally, we view a transaction as a tuple of non-negative reals, $\text{tx}_i = (t_i, b_i) \in \mathbb{R}^+ \times \mathbb{R}^{\geq 0}$.

TimeBoost scoring function. Intuitively, for the TimeBoost scoring function, we propose to allow users to “buy time” using their transaction bid; in other words, transactions will be sorted by increasing timestamps (as in FCFS) but now users are allowed to decrease their effective timestamp (i.e., increase their score) through bids.

Formally, the score of a transaction $\text{tx}_i = (t_i, b_i)$ is computed as follows:

$$S(t_i, b_i) = \pi(b_i) - t_i. \quad (1)$$

where $\pi(b_i)$ denotes the priority or advantage gained by bidding b_i . Transactions are now chosen in descending order of their scores.

Choosing a bidding function π . To choose the bidding function π for TimeBoost, we start by defining several natural properties that should be satisfied.

1. $\pi(0) = 0$. This normalization implies that paying 0 bid gives no additional advantage.
2. $\pi'(b) > 0$ for all $b \in \mathbb{R}^+$ where π' denotes the first derivative of π with respect to the bid. This implies that the priority increases with the bid, which gives incentive to bid more for a higher priority.
3. $\lim_{b \rightarrow \infty} \pi(b) = g$. This implies that no transaction can outbid a transaction which arrived g time earlier (but any time advantage of less than g can be outbid). Through this, we can guarantee that the transaction ordering can be finalized within time g .
4. $\pi''(b) < 0$ for all $b \in \mathbb{R}^+$ where π'' denotes the second derivative of π with respect to the bid. This means that priority is concave, or equivalently, the cost of producing the (bidding) signal is convex. This is generally necessary to obtain the interior solution of the equilibrium condition.

The simplest bidding function satisfying the above constraints is the function:

$$\pi(b_i) := \frac{gb_i}{b_i + c} \quad (2)$$

where c is some constant. We will use this as the bidding function for TimeBoost. In the next section, we provide an economic analysis for TimeBoost. For this, we will assume that $c = 1$.

Complexity. For any incoming transaction $\text{tx} = (t, b)$, the sequencer can finalize tx after a delay of $g - \pi(b_i)$. This is because after this point, no later transaction can outbid tx . If transactions arrive at rate r , the space complexity of the sequencing algorithm is $\Theta(r)$ and the computational cost per transaction is $\Theta(\log r)$, assuming pending transactions are stored in a priority queue, ranked by score.

3.1 TimeBoost Economic Analysis Overview

We now describe the model for analyzing the economics for our TimeBoost ordering policy. The next two sections will describe this analysis in detail.

Basic model. Consider an arbitrage opportunity that occurs at some time (w.l.g., this can be taken as time 0). Users (from now on referred to as players) need to now take a decision on (1) how to send their transaction to the sequencer; this corresponds to the investment in latency; and (2) how much extra to bid for their transaction to get higher priority. We will analyze a simple economic model of this decision problem.

Assume that it costs user i the amount $c_i(t)$ to get its transaction received by the sequencer t time after the arbitrage opportunity arises. The only requirement on $c_i(t)$, for now, is that it is decreasing in increasing t . When the arbitrage opportunity arises, a player i has a valuation v_i to have its transaction included for execution the earliest, among those transactions contending for the same opportunity.

Analysis organization. We begin with an analysis with two players in Section 4. Within this, we consider different models based on when the latency investment needs to occur. Broadly, we consider two models for latency investment: ex-ante (Section 4.1) and ex-post (Section 4.2). Ex-ante means that the latency investment needs to happen *before* learning the arbitrage opportunity while ex-post means that the latency investment can occur after learning about the arbitrage opportunity.

In Section 5, we generalize our results to many competing players.

4 Analysis of TimeBoost with 2 Players

As a starting point, assume that there are two players with valuations v_1 and v_2 , distributed as per the cumulative distribution functions (CDFs) F_1 and F_2 . That is, the probability that the valuation of player i is less or equal to x is equal to $F_i(x)$.

For each valuation v , the player may choose their specific latency investment. We can model this as a function $t_i : V \rightarrow \mathbb{R}$, such that, $t_i(v)$ is the latency / time chosen by a player i with valuation v . For simplicity, assume that the cost functions and value distributions are the same: $c_i(t) = c(t)$ and $F_i = F$. Throughout the paper, we assume that $F : [0, 1] \rightarrow [0, 1]$ is a uniform distribution with $F_i(x) = x$ iff $x \in [0, 1]$, for $i \in \{1, 2\}$, when final numerical values are derived. Obtaining numerical values for different distribution functions is very similar to that of uniform distribution, but we choose a uniform for simplicity of exposition. However, most of the computations are done for general distribution functions.

We now consider two different assumptions regarding the investment in latency improvement. In the first model (ex-ante), we assume that the players need to invest in their latency infrastructure in advance: they acquire or rent servers close to the sequencer prior to knowing the value of the arbitrage opportunities they are competing for. In the second (ex-post) model, we assume that the players are able to invest in the latency after they learn about their valuation of the arbitrage. This corresponds to the case where the arbitrage opportunity itself takes some time to be realized³. In this case, the transaction sender can schedule its transaction through the third-party service, which guarantees the delivery of the transaction

³ Example of such an opportunity is a 12-second delay on the Ethereum network for a transaction to be scheduled.

23:10 Buying Time: Latency Racing vs. Bidding for Transaction Ordering

within some time interval, once the arbitrage opportunity is realized. In both cases, bidding is naturally assumed to be an interim decision, and in fact, one of the biggest advantages, as the valuation is already learned.

4.1 Ex-Ante Latency Investment

In this model, players learn their valuations only after they have already invested in latency infrastructure. If players can only compete through latency, the interaction between them becomes a static game. We study equilibria solutions of these games. A similar setting is considered in [16]. The results obtained in the following two subsections are concrete cases of folk results in the microeconomic theory; however, we include their proofs for completeness.

4.1.1 Only latency investment

As a simple first step, we start by analyzing the game where only latency investment is allowed. Let x_i be the amount invested in latency by player i (so that he obtains a delay of $t_i(x_i)$). Let V_i denote the valuation random variable of player i . Then, player i has the following ex-ante payoff:

$$\text{Payoff}_i = \begin{cases} E[V_i] - x_i & \text{if player invests strictly more than the other player} \\ \frac{1}{2}E[V_i] - x_i & \text{if he invests an equal amount (assuming random tie-breaking)} \\ -x_i & \text{otherwise} \end{cases}$$

First, we note that there is no pure strategy Nash equilibrium solution of the game, in which player strategy sets consist of \mathbb{R}^+ . It is easy to show this by case distinction on valuations: there are simple deviating strategies in each case. Next, we focus on the mixed equilibrium solution and obtain the following result.

► **Proposition 7.** *There is a symmetric equilibrium in mixed strategies where each player i chooses x_i uniformly at random on the interval $(0, E[V_i])$.*

Proof. By construction, the payoff of player j of playing $x_j \leq E[V_j]$ against the uniform strategy on $(0, E[V_i])$ is

$$F(x_j)E[V_j] - x_j = \frac{x_j}{E[V_i]}E[V_j] - x_j = 0.$$

Choosing a strategy $x_j > E[V_j]$ gives a negative pay-off. Therefore, each $0 \leq x_j \leq E[V_j]$ is the best response of player j , and mixing uniformly among them is also the best response. ◀

The above-described equilibrium is unique up to a change of strategy on a null set and in any mixed equilibrium, both players obtain the same payoffs as in this equilibrium. Note that the result is independent of the latency cost function. The only property used is that if a player invests more than the other player in the latency technology, its transaction is scheduled earlier.

4.1.2 Budget constraints

We now model the fact that players may not have access to an arbitrary amount of money they need to invest to improve their latency, but are instead are constrained by a budget. Let B_i denote the budget of player i , meaning that player i cannot spend more than B_i . We consider an asymmetric case where one (weak) player has a budget $B_1 < E[V_i]$ and the other

(strong) player has a larger budget with $B_2 > B_1$. First, note that similar to the previous section with unlimited access to money, there is no pure strategy Nash equilibrium. Therefore, we switch to mixed strategy equilibrium. Let F_i denote the probability distribution of playing different strategies.

► **Proposition 8.** *There exists a mixed Nash equilibrium solution in the game in which the weak player receives a payoff of 0 and the strong player receives a payoff of $E[V_i] - B_1$.*

Proof. The following strategy profile in which the first player plays according to the following (mixed) strategy:

$$F_1(x) = \begin{cases} \frac{x}{E[V_i]} + \frac{E[V_i] - B_1}{E[V_i]}, & x \in (0, B_1], \\ \frac{E[V_i] - B_1}{E[V_i]}, & x = 0, \\ 1, & x > B_1, \end{cases}$$

the second player plays according to

$$F_2(x) = \begin{cases} \frac{x}{E[V_i]}, & x \in [0, B_1), \\ 1, & x \geq B_1, \end{cases}$$

is a mixed strategy equilibrium. The first, weak player obtains an expected payoff 0 for any choice of $0 \leq x_1 < B_1$. The second, strong player obtains an expected payoff of $E[V_i] - B_1$ for any choice $0 < x_2 \leq B_1$. Choosing $x_2 > B_1$ is wasteful for the second player and will not occur in equilibrium. Thus, both players are indifferent between all pure strategies in support of F_1 resp. F_2 and for player 2 choosing an action outside of the support of F_2 is dominated. The mixed strategies form a Nash equilibrium. ◀

Similarly to the unconstrained case, the above-described equilibrium is unique up to a change of strategy on a null set and in any mixed equilibrium, both players obtain the same payoffs as in this equilibrium. Also similarly to the previous section, the result is independent of the latency cost function. The only property to derive this result is that if a player invests more than the other player in the latency technology, its transaction is faster (has a lower timestamp).

4.1.3 Ex-ante Latency with Interim Bidding

We now analyze the model where both latency and bidding are allowed but the latency is ex-ante. That is, investment in latency happens before players learn their valuations but after learning their valuation players can use bidding to improve the transaction score.

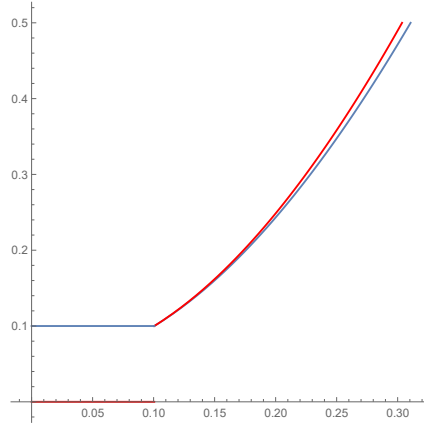
We consider a version where players learn the other players' latency investment decisions before bidding. This models the fact that players will typically play the game repeatedly and can therefore observe latency levels of each other.

In the following let $x = (x_1, x_2)$ be the latency investment levels chosen by the two bidders and let $\Delta := t_2(x_2) - t_1(x_1)$ be the corresponding difference in latency. W.l.o.g. assume $\Delta \geq 0$. First, we consider the case that $\Delta = 0$:

► **Proposition 9.** *There is a completely separating equilibrium of the bidding game when both bidders have made the same ex-ante investment.*

Proof. Given in Section 4.1.4 ◀

23:12 Buying Time: Latency Racing vs. Bidding for Transaction Ordering



■ **Figure 1** Example of equilibrium signaling functions for $g = 10$ and $\Delta = 0.1$. Timestamps are normalized so that $t_2 = 0$. The blue function is the equilibrium signal $\pi_1(v) - t_1$ for bidder 1 as a function of the valuation. The red function is the equilibrium signal $\pi_2(v) - t_2$ for bidder 2 as a function of the valuation.

Next, we consider the case that $\Delta \neq 0$. For the case of different ex-ante investment we get partially separating equilibria where bidders do not bid for low valuations and bid for high valuations. The bidding strategies are asymmetric in general. However, for sufficiently large g the equilibrium becomes approximately symmetric and approximately efficient. See Figure 1 for a graphical illustration.

► **Proposition 10.** *There is an equilibrium of the bidding game which is separating conditional on bidding: There is a threshold $\sqrt{\frac{\Delta}{g-\Delta}}$, such that a bidder does not bid if his valuation is below the threshold and bids if his valuation is above the threshold. Conditional on bidding, the high latency bidder i produces a higher signal than the low latency bidder j for equal valuations: $\pi_i(v) - t_i > \pi_j(v) - t_j$, for $v > \sqrt{\frac{\Delta}{g-\Delta}}$.*

Proof. Given in Section 4.1.4 ◀

The equilibrium analysis in Propositions 9 and 10 indicates how efficient our transaction ordering policy is as a function of the latency investment of bidders. If bidders have the same latency we have a standard all pay auction which yields a fully efficient outcome. If there is a difference in latency we have no bidding for low valuation bidders and approximately equal signals produced for equal valuations for high valuation bidders. Conditional on entry, low latency bidders underbid and high latency bidders overbid relative to the standard all pay strategies. Efficiency depends on the latency difference and the g parameter. If g is chosen sufficiently large the auction is approximately efficient. A too low g can be detected by low bidding activity. Hence our transaction policy can strike a balance between fairness, low latency and efficiency if properly parameterized.

4.1.4 Proofs

Proof of Proposition 9. We want to determine bidding signals $\pi_1(v_1, \Delta)$ and $\pi_2(v_2, \Delta)$, which are functions of valuations and the difference in latency. For a given Δ denote the inverse of $\pi_1(\cdot, \Delta)$ and $\pi_2(\cdot, \Delta)$ by $\tilde{v}_1(\cdot, \Delta)$ and $\tilde{v}_2(\cdot, \Delta)$. Then bidder 1 solves at the interim stage

$$\max_{\pi \geq 0} Pr[\pi - t_1(x_1) \geq \pi_2(v_2, x) - t_2(x_2)]v_1 - \frac{\pi}{g - \pi} = F(\tilde{v}_2(\pi + \Delta, \Delta))v_1 - \frac{\pi}{g - \pi},$$

We obtain the first order condition:

$$f(\tilde{v}_2(\pi + \Delta, \Delta))v_1 \frac{\partial \tilde{v}_2(\pi + \Delta, \Delta)}{\partial \pi} = \frac{g}{(g - \pi)^2}$$

For the uniform distribution, this simplifies to:

$$v_1 \frac{\partial \tilde{v}_2(\pi + \Delta, \Delta)}{\partial \pi} = \frac{g}{(g - \pi)^2}.$$

Similarly, for bidder 2 we obtain

$$v_2 \frac{\partial \tilde{v}_1(\pi - \Delta, \Delta)}{\partial \pi} = \frac{g}{(g - \pi)^2}.$$

The two equations give a system of differential equations that need to be solved for π_1 and π_2 or alternatively for \tilde{v}_1 and \tilde{v}_2 . Alternatively, we can write the system as:

$$\tilde{v}_1(\pi, \Delta) \frac{\partial \tilde{v}_2(\pi + \Delta, \Delta)}{\partial \pi} = \frac{g}{(g - \pi)^2}. \quad (3)$$

$$\tilde{v}_2(\pi, \Delta) \frac{\partial \tilde{v}_1(\pi - \Delta, \Delta)}{\partial \pi} = \frac{g}{(g - \pi)^2}. \quad (4)$$

The solution to (3) and (4) in case of equal investment (so that $\Delta = 0$) and a symmetric equilibrium is given by the following formula:

$$\tilde{v}_1(\pi, 0) = \tilde{v}_2(\pi, 0) = \sqrt{2 \int_0^\pi \frac{g}{(g - \pi)^2} d\pi} = \sqrt{\frac{2\pi}{g - \pi}}. \quad (5)$$

We solve for the signal as a function of the valuation:

$$v^2 = \frac{2\pi}{g - \pi} \Leftrightarrow \pi = \frac{gv^2}{2 + v^2}. \quad \blacktriangleleft$$

Proof of Proposition 10. When $x_1 \neq x_2$, we can first sum up (3) and (4) to obtain a differential equation for the expected payoff $v_1 v_2$:

$$\frac{d(v_1(\pi)v_2(\pi + \Delta))}{d\pi} = \frac{g}{(g - \pi)^2} + \frac{g}{(g - \pi - \Delta)^2}. \quad (6)$$

Integrating both sides of the differential equation above gives the solution:

$$v_1(\pi)v_2(\pi + \Delta) = \frac{\pi}{(g - \pi)} + \frac{\pi + \Delta}{g - \pi - \Delta} + K. \quad (7)$$

To determine the constant we need to determine boundary conditions. For bidder 1, at the threshold where he is indifferent between bidding and not bidding, we have $\pi_1 = 0$ and for bidder 2, at the threshold where he is indifferent between bidding and not bidding, he needs to overcome the handicap, we have $\pi_2 = \Delta$. At the threshold bidder 2 should make the same profit as from pooling,

$$v_1(0)v_2(\Delta) = \frac{\Delta}{g - \Delta} \Rightarrow K = 0.$$

23:14 Buying Time: Latency Racing vs. Bidding for Transaction Ordering

Combining (7) and (4) we obtain a separable differential equation:

$$\frac{dv_1(\pi, \Delta)}{v_1(\pi, \Delta)} = d\pi \frac{g}{(g - \pi - \Delta)^2} \left(\frac{\pi}{g - \pi} + \frac{\pi + \Delta}{g - \pi - \Delta} \right)^{-1}. \quad (8)$$

Combining (7) and (3) we obtain another separable differential equation:

$$\frac{dv_2(\pi + \Delta, \Delta)}{v_2(\pi + \Delta, \Delta)} = d\pi \frac{g}{(g - \pi)^2} \left(\frac{\pi}{g - \pi} + \frac{\pi + \Delta}{g - \pi - \Delta} \right)^{-1}. \quad (9)$$

Integrating both parts of the equation (8) solves the (logarithm of) the value as a function of the bid:

$$\ln(v_1(\pi)) - \ln(v_1(0)) = \int_0^\pi \frac{g}{(g - \pi - \Delta)^2} \left(\frac{\pi}{g - \pi} + \frac{\pi + \Delta}{g - \pi - \Delta} \right)^{-1} d\pi.$$

Similarly, integrating both parts of the equation (9) solves the (logarithm of) the value as a function of the bid:

$$\ln(v_2(\pi + \Delta)) - \ln(v_2(\Delta)) = \int_0^\pi \frac{g}{(g - \pi)^2} \left(\frac{\pi}{g - \pi} + \frac{\pi + \Delta}{g - \pi - \Delta} \right)^{-1} d\pi.$$

To determine the marginal valuations $v_1(0)$ and $v_2(\Delta)$ at which the two bidders start bidding, note that the support of $\pi_i - t_i$ and that of $\pi_j - t_j$ need to coincide for valuations where we have separation of types. Therefore, $v_1(0) = v_2(\Delta)$. Since $v_1(0)v_2(\Delta) = \frac{\Delta}{g - \Delta}$ it follows that $v_1(0) = v_2(\Delta) = \sqrt{\frac{\Delta}{g - \Delta}}$. This is the threshold where bidders start bidding. It follows that for $\Delta \neq 0$

$$v_1(\pi) = \sqrt{\frac{\Delta}{g - \Delta}} \exp \left(\int_0^\pi \frac{g}{(g - \pi - \Delta)^2} \left(\frac{\pi}{g - \pi} + \frac{\pi + \Delta}{g - \pi - \Delta} \right)^{-1} d\pi \right)$$

and

$$v_2(\pi) = \sqrt{\frac{\Delta}{g - \Delta}} \exp \left(\int_0^{\pi - \Delta} \frac{g}{(g - \pi)^2} \left(\frac{\pi}{g - \pi} + \frac{\pi + \Delta}{g - \pi - \Delta} \right)^{-1} d\pi \right).$$

To compare the equilibrium signals $\pi_1(v) - t_1$ and $\pi_2(v) - t_2$ for $v > \sqrt{\frac{\Delta}{g - \Delta}}$, we need to compare $\pi_1(v) + \Delta$ to $\pi_2(v)$.

From the expressions for the valuations as a function of the bid, we can observe (observe that $\frac{g}{(g - \pi - \Delta)^2} \geq \frac{g}{(g - \Delta)^2}$) that

$$v_1(\pi) > v_2(\pi + \Delta),$$

for $\pi > 0$. It follows that

$$\pi_1(v) \leq \pi_2(v) - \Delta,$$

for $v \geq \sqrt{\frac{\Delta}{g - \Delta}}$. ◀

4.2 Ex-Post Latency with Bidding

We now analyze the ex-post model with bidding; here both the latency investment, and the bid can be made after the valuation is observed. First, we start with only the latency investment decision. The expected utility of player i is equal to:

$$\Pr[t(v_i) < t(v_j)]v_i - c(t(v_i)),$$

where $j \in \{1, 2\} \setminus i$.

We can look at this from a dual perspective: by $v(t)$ we define the inverse of $t(v)$. This is the so-called *Revelation Principle*. Instead of some function of the type, we report our type directly. Then, the optimization problem becomes:

$$\max_v \Pr[v \geq v_2]v_1 - c(t(v)). \quad (10)$$

By replacing the probability with $F(v)$, we get that it is equivalent to

$$\max_v F(v)v_1 - c(t(v)).$$

By the first order condition, we get:

$$v_1 f(v) - c'(t(v))t'(v)|_{v=v_1} = 0,$$

where f is a density function of the valuation distribution F . By plugging in $v = v_1$, it is equal to:

$$v_1 f(v_1) - c'(t(v_1))t'(v_1). \quad (11)$$

For the uniform distribution and cost function $c = \frac{1}{t}$, first order condition gives the following differential equation:

$$v_1 + \frac{t'(v_1)}{t^2(v_1)} = 0. \quad (12)$$

Solving this equation gives $t(v) = \frac{2}{c_1 + v^2}$. By the boundary condition that 0 valuation type should wait infinitely (or equivalently pay 0 in the latency), we obtain the value of the constant in the solution: $c_1 = 0$. Therefore, cost incurred is equal to $\frac{1}{t} = \frac{v^2}{2}$. On average each player pays:

$$\int_0^1 \frac{v^2}{2} f(v) dv \Big|_0^1 = \frac{1}{6},$$

for better latency, before learning their types. The cost of producing score $s = \frac{gm}{m+1} - t$ is:

$$c(s) := m + \frac{1}{t}. \quad (13)$$

We decompose total expenditure into 2 parts, for bidding and for time, by representing m and $c(t(v))$ as functions of v and taking integrals:

$$b(g) := \int_0^1 m(v) f(v) dv \text{ and } \int_0^1 \frac{1}{t(v)} f(v) dv.$$

► **Proposition 11.** *The limit of $b(g)$ when g tends to infinity is equal to $\frac{1}{6}$. $b(g)$ is an increasing function in g , for g large enough.*

23:16 Buying Time: Latency Racing vs. Bidding for Transaction Ordering

Proof. Given in Section 4.2.1 ◀

The proposition implies that by taking large enough g , the system extracts almost all value invested in the latency through bidding. Starting from some threshold value on g , extraction increases with increasing g .

We can verify whether the constructed equilibrium is unique by checking the conditions given in [12].

► **Example 12.** We can calculate a few values of $b(g)$. In particular, $b(1000) \approx 0.1294$, meaning a player pays approximately 77% of the total expenditure in bids, and $b(10000) \approx 0.1537$, meaning a player pays approximately 92% of the total expenditure in bids.

Note that in the proof of the proposition 11, the total investment in both latency and bidding, $c(v)$, is the same value $\frac{v^2}{2}$, as in the case of only investing in the latency. We show that this is not a coincidence. In general, assume that there is an arbitrary signaling technology described by an increasing, differentiable cost function $C(s)$. The following result shows the revenue equivalence of ex-post bidding:

► **Proposition 13.** *Both players spend the same amount on average for any cost function C .*

Proof. Given in Section 4.2.1 ◀

The amount spent depends only on the value belief distribution function.

4.2.1 Proofs

Proof of Proposition 11. The optimization problem of the player in the equilibrium is to minimize cost, subject to the score equation constraint. By plugging in $t = \frac{gm}{m+1} - s$, we obtain the minimization problem:

$$\min_m \left(m + \frac{m+1}{gm - s(m+1)} =: x(m) \right).$$

The first order condition on $x(m)$ gives:

$$\frac{dx(m)}{dm} = 1 + \frac{gm - s(m+1) - (m+1)(g-s)}{(gm - s(m+1))^2} = 1 - \frac{g}{(gm - s(m+1))^2} = 0, \quad (14)$$

gives that the value of m that minimizes the cost function. The solutions of the last equation are $gm - sm - s = \sqrt{g}$ equivalent to $m = \frac{s+\sqrt{g}}{g-s}$ and $gm - sm - s = -\sqrt{g}$ equivalent to $m = \frac{s-\sqrt{g}}{g-s}$, or the boundary condition $m = 0$. For $m = 0$, the value $x(0) = -\frac{1}{s}$, while for $m = \frac{s+\sqrt{g}}{g-s}$, the value

$$x\left(\frac{s+\sqrt{g}}{g-s}\right) = \frac{s+\sqrt{g}}{g-s} + \frac{\frac{s+\sqrt{g}}{g-s} + 1}{g\frac{s+\sqrt{g}}{g-s} - s\left(\frac{s+\sqrt{g}}{g-s} + 1\right)} = \frac{1 + 2\sqrt{g} + s}{g-s}.$$

Accordingly, the marginal cost of producing signal s is:

$$c'(s) = \begin{cases} \frac{(1+\sqrt{g})^2}{(g-s)^2}, & \text{if } s > -\sqrt{g}, \\ \frac{1}{s^2}, & \text{if } s \leq -\sqrt{g}. \end{cases}$$

We solve a similar differential equation as (11), just with different marginal cost function c' , and instead of time function t , we have a score function s of valuation v . The differential equation becomes:

$$vf(v) - c'(s)s'(v) = 0. \quad (15)$$

We need to solve for the $s(v)$ function. For types v with $\frac{2}{v^2} \geq \sqrt{g}$ who only use latency we have the same solution as before

$$s(v) = -\frac{2}{v^2}.$$

The marginal type who is indifferent between using only latency and using a combination of the two technologies is given by

$$u = \sqrt{\frac{2}{\sqrt{g}}}.$$

which gives the boundary condition $s(u) = -\sqrt{g}$ for the differential equation describing the behavior of types who choose a signal $s \geq -\sqrt{g}$:

$$v = \frac{(1 + \sqrt{g})^2}{(g - s)^2} s'(v).$$

We obtain the solution

$$s(v) = (4c_1g^{3/2} + 2c_1g^2 + 2c_1g + g(v^2 - 2) - 4\sqrt{g} - 2)/(2c_1g + 4c_1\sqrt{g} + 2c_1 + v^2). \quad (16)$$

The value of the constant c is obtained from the boundary condition that a zero-value player does not invest and it is equal to

$$c_1 = \frac{1}{(1 + \sqrt{g})^2}.$$

Therefore, plugging in the constant value in the solution (16) and simplifying it gives:

$$s(v) = \frac{gv^2 - 4\sqrt{g} - 2}{v^2 + 2}.$$

Plugging this into the formula of $c(s)$, gives the cost value as a function of valuation v :

$$c(v) = \frac{1 + 2\sqrt{g} + \frac{gv^2 - 4\sqrt{g} - 2}{v^2 + 2}}{g - \frac{gv^2 - 4\sqrt{g} - 2}{v^2 + 2}} = \frac{v^2}{2}.$$

Separate expenditure in the bidding is calculated by the following formula:

$$\begin{aligned} b(g) &= \int_u^1 m(v)f(v)dv = \int_{\sqrt{\frac{2}{\sqrt{g}}}}^1 \frac{\frac{gv^2 - 4\sqrt{g} - 2}{v^2 + 2} + \sqrt{g}}{g - \frac{gv^2 - 4\sqrt{g} - 2}{v^2 + 2}} dv = \\ &= \int_{\sqrt{\frac{2}{\sqrt{g}}}}^1 \frac{v^2(g + \sqrt{g}) - 4\sqrt{g} - 2}{2g - 4\sqrt{g} - 2} dv = \\ &= \frac{1}{2g + 4\sqrt{g} + 2} \left(\frac{g + \sqrt{g}}{3} \left(1 - \frac{2}{\sqrt{g}} \sqrt{\frac{2}{\sqrt{g}}}\right) - (4\sqrt{g} + 2) \left(1 - \sqrt{\frac{2}{\sqrt{g}}}\right) \right). \end{aligned}$$

The dominant term in the nominator above is g and also in the denominator, it is $6g$. Therefore, $\lim_{g \rightarrow \infty} b(g) = \frac{1}{6}$. ◀

23:18 Buying Time: Latency Racing vs. Bidding for Transaction Ordering

Proof of Proposition 13. We are interested in the equilibrium signaling strategy $s(v)$. Suppose that this strategy is increasing (so no pooling of types) and differentiable. Then, we can define a differentiable function

$$\tilde{C}(v) := C(s(v)).$$

To figure out what $\tilde{C}(v)$ is, we have to consider an optimization problem with the first player:

$$\max_v Pr[v \geq v_2]v_1 - C(s(v)) = Pr[v \geq v_2]v_1 - \tilde{C}(v).$$

Taking first order conditions with respect to v gives:

$$v_1 f(v) - \tilde{C}'(v)|_{v=v_1} = 0,$$

that is,

$$v_1 f(v_1) = \tilde{C}'(v_1).$$

For the uniform distribution:

$$v_1 = \tilde{C}'(v_1).$$

Using the boundary condition $\tilde{C}(0) = 0$ and integrating we get

$$\tilde{C}(v_1) = v_1^2/2.$$

More generally:

$$\tilde{C}(v_1) = \int_{-\infty}^{v_1} v f(v) dv. \quad \blacktriangleleft$$

5 Analysis of TimeBoost with n players

In this section, we consider n players with the same valuation distribution as in the 2 players case. The optimization problem is now the following:

$$\max_v Pr[v \geq \max\{v_2, \dots, v_n\}]v_1 - c(t(v)),$$

similarly to (10). By replacing the probability with cumulative distribution, this is equivalent to:

$$\max_v F_{n-1}(v)v_1 - c(t(v)),$$

where $F_{n-1}(x)$ is a cumulative distribution function of the random variable $X := \max\{X_1, \dots, X_{n-1}\}$. By independence we have

$$F_{n-1}(x) = F(x)^{n-1}.$$

The first-order condition and plugging in $v = v_1$ gives the following differential equation, similar to (11):

$$f_{n-1}(v_1)v_1 - c'(t(v_1))t'(v_1) = 0,$$

where $f_{n-1}(v_1) = (n-1)v_1^{n-2}$ is a density function of maximum among $n-1$ uniformly distributed random variables. The differential equation w.r.t. $t(v)$ becomes:

$$(n-1)v_1^{n-1} + \frac{t'(v_1)}{t^2(v_1)} = 0.$$

Solving the equation gives $t(v) = \frac{n}{c+(n-1)v^n}$. The same boundary condition ensures that $c = 0$, that is, $t(v) = \frac{n}{(n-1)v^n}$. Each player pays:

$$\frac{n-1}{n} \int_0^1 v^n dv = \frac{n-1}{n} \frac{v^{n+1}}{n+1} \Big|_0^1 = \frac{n-1}{n(n+1)}.$$

Together, the players pay $\frac{n-1}{n+1}$, that converges to 1 as n converges to infinity. Note that the first place in the transaction order is given to the maximum-value player. The average valuation of the maximum value player is $\frac{n}{n+1}$, order statistics. This value also converges to 1 as n tends to infinity.

The analysis is the same as in the case of 2 players, until the differential equation that solves score function s . Instead of (15), for n players we solve:

$$(n-1)vv^{n-1} - c'(s)s'(v) = 0. \quad (17)$$

For types v with $\frac{n}{(n-1)v^2} \geq \sqrt{g}$, who only use latency, we have the same solution as before

$$s(v) = -\frac{n}{(n-1)v^2}.$$

Marginal type investing in bidding is:

$$u = \sqrt{\frac{n}{(n-1)\sqrt{v}}}.$$

Plugging in functional forms of c and s in (17) gives the same limit results as in Proposition 11. Next, we show a revenue equivalence for n players. The argument is similar to 2 players' case. Assume that there is an arbitrary signaling technology described by an increasing, differentiable cost function $C(s)$.

► **Proposition 14.** *All n players spend the same amount on average for any cost function C .*

Proof. We are interested in the equilibrium signaling strategy $s(v)$. Suppose that this strategy is increasing (so no pooling of types) and differentiable. Then, we can define a differentiable function

$$\tilde{C}(v) := C(s(v)).$$

To figure out what $\tilde{C}(v)$ is, we have to consider an optimization problem of the first player:

$$\max_v Pr[v \geq \max\{v_2, \dots, v_n\}]v_1 - \tilde{C}(v) = F(v)^{n-1}v_1 - \tilde{C}(v).$$

Taking first order conditions with respect to v :

$$[(n-1)v_1f(v)F(v)^{n-2} - \tilde{C}'(v)] \Big|_{v=v_1} = 0,$$

For the uniform distribution, we get:

$$(n-1)v_1^{n-1} = \tilde{C}'(v_1).$$

23:20 Buying Time: Latency Racing vs. Bidding for Transaction Ordering

Using the boundary condition $\tilde{C}(0) = 0$ and integrating we get

$$\tilde{C}(v_1) = \frac{(n-1)v_1^n}{n}.$$

More generally:

$$\tilde{C}(v_1) = \int_{-\infty}^{v_1} (n-1)vf(v)F(v)^{n-2}dv. \quad \blacktriangleleft$$

6 Comparison of TimeBoost with a Pure Bidding Policy

We now compare TimeBoost to what to a pure bidding policy. Recall that for the bidding policy, all transactions sent in fixed time intervals of length g are collected, and sorted in decreasing order of their bids. This effectively simulates a first-price all-pay auction for each interval. We note this can be thought of as a quantized version of TimeBoost, because it produces the same sequence that would be produced by first rounding off each transaction's arrival timestamp to the nearest multiple of g and then applying TimeBoost.

Generically speaking, a first-price auction where only the winning bidder pays and first-price all-pay auctions are both *payoff equivalent* for Bayesian-Nash incentive compatible mechanisms, (see e.g., [14]). In our setting, the following result holds for each individual arbitrage opportunity.

► **Proposition 15** (see [14]). *The expected payoff of the bidding game where the only the highest bidder pays their bid is equal to the expected payoff in the bidding game where the highest bidder wins but all players pay their bids, independently of valuation distributions.*

For simplicity, to compare TimeBoost with a pure bidding policy, we consider two players. It is straightforward to generalize to more parties. For a given arbitrage opportunity, two cases arise as described below depending on whether transactions can be submitted within the same g -time interval as the arbitrage opportunity or not:

1. Both players can submit their transactions within the same g interval. For the pure bidding policy, if both players can get their transaction submitted inside the same g -time interval as the arbitrage opportunity, then they will both compete for it. It is easy to see that when the valuations of the two parties are the same, the bidding strategy for the pure-bidding policy vs the ex-ante latency with bidding policy will be the same. In other words, in this scenario, TimeBoost maintains the economic properties of the first-price auction pure-bidding policy.
2. Only one player can get its transaction within the same g interval. If only one player can get its transaction inside the same g -time interval as the arbitrage opportunity, then in the pure-bidding policy, that player can pay a 0 bid and still take advantage of it. In contrast, since TimeBoost does not require discrete boundaries, both players will always have g time to submit their transactions (recall that bidding can be used to get priority over any transaction received up to g time earlier). This means that even for a reasonably small g (say 0.5 sec), both parties will always be able to compete for the opportunity. In equilibrium, this results in bids equal to value of the arbitrage.

Analysis for the second case. Suppose the first party (denoted by A) can reach the sequencer in s_1 time, and the second party (denoted by B) can reach in s_2 time, with $s_1 < s_2$. Then, with the pure-bidding policy, A can wait until $g - s_1$ seconds pass since the beginning of a new block creation, and send its transaction to the sequencer at exactly $g - s_1$, while B has to send its transaction by time $g - s_2$ in order to be included in the same block.

Assuming that arbitrage opportunities are uniformly distributed over the g -interval, this means that, with probability $\frac{g-s_1-(g-s_2)}{g} = \frac{s_2-s_1}{g}$, B has no chance to win the race against A, even if it values the arbitrage opportunity much more than A. When g is large (e.g., on Ethereum with 12 sec block times), this latency advantage is not a big issue, as A would only have an advantage with probability $(s_2 - s_1)/12$. In contrast, for faster blockchains, or layer-2 rollups which have shorter block-times to achieve scalability, this latency advantage can be significantly more important in the pure-bidding policy vs in TimeBoost. For instance, when $g = 0.5\text{sec}$, A's latency advantage is 24 times greater than what it was in Ethereum. This means that compared to TimeBoost, a pure-bidding strategy will either result in substantial latency competition (when g is small) or will not be able to provide low transaction finalization time (since g will be large).

7 Discussion on Sequencer Decentralization

We now briefly discuss how TimeBoost can be supported by a decentralized sequencer – i.e., a committee of ℓ sequencers (of which at most some f can be dishonest). We only provide possible implementations here; a formal rigorous analysis is outside the scope of this paper.

In the decentralized setting, transactions to be sequenced are now submitted by users to all sequencers instead of just one. Note that as before, threshold decryption techniques can be used for transaction privacy before ordering.

The most natural way to support TimeBoost in a decentralized setting is to have a protocol for sequencers to agree on both the timestamp and the bid of transactions. After this is done, the TimeBoost policy can simply be applied on the consensus output of the decentralized committee to obtain the final ordering. Agreeing on the bid is easy since we can have the same bid be submitted to all sequencers for a given transaction. Agreeing on the timestamp is a more challenging problem since the same transaction can arrive at different nodes. While it adds significant complexity, one potential technique here is to employ a fair-ordering protocol (this can be as simple as e.g., computing the median timestamp [11, 20] or support more complicated techniques as in [3, 8, 9]). We leave the formal analysis of such a decentralized TimeBoost implementation to future work.

8 Conclusion

We designed TimeBoost: a policy for transaction ordering that takes into account both transaction arrival times and bids. We showed that any ordering scheme that guarantees the independence of different latency races is a generalized scoring rule. By choosing a suitably designed mixture of timestamps and bids, we showed the economic efficiency of the system: transaction senders spend most of their resources on bidding instead of latency improvement, which can later be used by the protocol for improvement and development.

References

- 1 Kushal Babel and Lucas Baker. Strategic peer selection using transaction value and latency. In *DeFi @ CCS*, pages 9–14, 2022.
- 2 Kushal Babel, Philip Daian, Mahimna Kelkar, and Ari Juels. Clockwork finance: Automated analysis of economic security in smart contracts. In *IEEE S&P*, pages 2499–2516, 2023.
- 3 Christian Cachin, Jovana Micic, and Nathalie Steinhauer. Quick order fairness. In *FC*, 2022.
- 4 Georg Cantor. Beiträge zur begründung der transfiniten mengenlehre. *Mathematische Annalen*, pages 481–512, 1895.

23:22 Buying Time: Latency Racing vs. Bidding for Transaction Ordering

- 5 Christopher P Chambers and Federico Echenique. *Revealed preference theory*, volume 56. Cambridge University Press, 2016.
- 6 Philip Daian, Steven Goldfeder, Tyler Kell, Yunqi Li, Xueyuan Zhao, Iddo Bentov, Lorenz Breidenbach, and Ari Juels. Flash boys 2.0: Frontrunning in decentralized exchanges, miner extractable value, and consensus instability. In *IEEE S&P*, pages 585–602, 2020.
- 7 Gerard Debreu. Representation of a preference ordering by a numerical function. *Decision processes*, 3:159–165, 1954.
- 8 Mahimna Kelkar, Soubhik Deb, Sishan Long, Ari Juels, and Sreeram Kannan. Themis: Fast, strong order-fairness in byzantine consensus. *IACR Cryptol. ePrint Arch.*, page 1465, 2021. URL: <https://eprint.iacr.org/2021/1465>.
- 9 Mahimna Kelkar, Fan Zhang, Steven Goldfeder, and Ari Juels. Order-fairness for Byzantine consensus. In *CRYPTO*, pages 451–480, 2020.
- 10 Vijay Krishna. *Auction Theory*. Academic Press, 2002.
- 11 Klaus Kursawe. Wendy, the good little fairness widget: Achieving order fairness for blockchains. In *ACM AFT*, pages 25–36, 2020.
- 12 Eric Maskin and John Riley. Uniqueness of equilibrium in sealed high-bid auctions. *Games and Economic Behavior*, 45(2):395–409, 2003.
- 13 Ciamac C. Moallemi and Mehmet Saglam. OR forum - the cost of latency in high-frequency trading. *Oper. Res.*, 61(5):1070–1086, 2013.
- 14 Roger B. Myerson. Optimal auction design. *Math. Oper. Res.*, 6(1):58–73, 1981.
- 15 Kaihua Qin, Liyi Zhou, and Arthur Gervais. Quantifying blockchain extractable value: How dark is the forest? In *IEEE S&P*, pages 198–214, 2022.
- 16 Ron Siegel. All-pay contests. *Econometrica*, 77:71–92, 2009.
- 17 Weizhao Tang, Lucianna Kiffer, Giulia Fanti, and Ari Juels. Strategic latency reduction in blockchain peer-to-peer networks. *Proc. ACM Meas. Anal. Comput. Syst.*, 7(2):32:1–32:33, 2023.
- 18 Anton Wahrstätter, Liyi Zhou, Kaihua Qin, Davor Svetinovic, and Arthur Gervais. Time to bribe: Measuring block construction market, 2023. [arXiv:2305.16468](https://arxiv.org/abs/2305.16468).
- 19 Sen Yang, Fan Zhang, Ken Huang, Xi Chen, Youwei Yang, and Feng Zhu. Sok: Mev countermeasures: Theory and practice, 2022. [arXiv:2212.05111](https://arxiv.org/abs/2212.05111).
- 20 Yunhao Zhang, Srinath Setty, Qi Chen, Lidong Zhou, and Lorenzo Alvisi. Byzantine ordered consensus without Byzantine oligarchy. In *OSDI*, pages 633–649, 2020.