Transaction Order Fairness in Synchronous Networks

Zhuo Cai[1][0000−0001−9673−6888] and Amir Kafshdar Goharshady[2][0000−0003−1702−6584]

[1] Hong Kong University of Science and Technology, HKSAR, China
zcaiam@connect.ust.hk
[2] University of Oxford, Oxford, UK goharshady@cs.ox.ac.uk

Abstract. The abstract should briefly summarize the contents of the paper in 150–250 words.

Keywords: First keyword · Second keyword · Another keyword.

1 Introduction

Following research of [1].

1.1 System Model

Nodes/Committee Assume a committee of n nodes is responsible for processing transactions in the blockchain system.

Communication Unless otherwise stated, we assume the communication network is synchronous, i.e., there is a known upper bound Δ on the time it takes for a message to be delivered from one node to all other nodes. In other words, for each transaction request, every committee node receives it. Moreover, the first node to receive the transaction receives the request at most Δ time before the last node receives it.

Adversary Model Unless otherwise stated, we assume that a small set of nodes is controlled by an arbitrary adversary. Let h be the ratio of honest nodes, and $m = 1 - h$ be the ratio of malicious nodes. Let H be the set of honest nodes and M be the set of malicious nodes. We require that h is at least larger than $1/2$. In some cases, we require h to be larger than a higher threshold.

Notations Let $N = \{1, 2, \ldots, n\}$ denote the set of committee nodes. Let tx denote a transaction and TX denote the universe of transactions. Let $t_i : \mathsf{TX} \to \mathbb{R}$ denote the mapping from a transaction tx to the time at which node $i \in N$ receives the transaction. We assume that $t_i(\mathsf{tx})$ is defined for all nodes $i \in N$.

Timestamp Report A timestamp report for a node i, R_i, is a list of tuples $(\mathsf{tx}, t_i(\mathsf{tx}))$ for a subset of transactions that node i has received. Note that malicious nodes might report timestamps different from the actual time they received the transactions. Therefore, we use $t_i(\mathsf{tx})$ to denote the reported timestamp of transaction tx by node i, and use $\tilde{t}_i(\mathsf{tx})$ to denote the actual time of node i receiving tx.

Transaction Ordering Mechanism A transaction ordering mechanism Γ for n nodes, is a function that takes as input a set of timestamp reports from all nodes and outputs a total ordering of the transactions appearing in the timestamp reports. In application, every node should report all transactions that it has received but has not been included in previous blocks yet. Malicious nodes might not share their reports, in which case the ordering mechanism Γ use default empty reports for these nodes.

1.2 Definition

We define a new relaxation of the transaction order fairness property, which we call *(γ, δ)-transaction order fairness*, parameterized by a ratio $\gamma \in (1/2, 1)$ and a duration $\delta > 0$. We introduce more notations before presenting the definition.

More Notations Let $I^{\delta,\{R_i\}_{i\in N}}(\mathsf{tx}_1, \mathsf{tx}_2) := \{i \in N : t_i(\mathsf{tx}_1) \leq t_i(\mathsf{tx}_2) - \delta\}$ denote the set of nodes that receive transaction tx_1 at least δ time before they receive transaction tx_2. If a node does not report to receive tx_1 or tx_2, it is not considered in the set $I^{\delta,\{R_i\}_{i\in N}}(\mathsf{tx}_1, \mathsf{tx}_2)$. When it is clear from the context, we will omit δ and the set of timestamp reports $\{R_i\}_{i\in N}$ and simply write $I(\mathsf{tx}_1, \mathsf{tx}_2)$.

The definition is as follows:

Definition 1 ((γ, δ)-Transaction Order Fairness). *A transaction ordering mechanism Γ is said to satisfy* (γ, δ)-transaction order fairness *if the following conditions hold:*

- Fairness Condition: *For any two different transactions $\mathsf{tx}_1, \mathsf{tx}_2 \in \mathsf{TX}$, if $|I(\mathsf{tx}_1, \mathsf{tx}_2)| \geq \gamma n$, which we denote as a constraint relation $\mathsf{tx}_1 \prec \mathsf{tx}_2$, then Γ outputs tx_1 before tx_2.*

Lemma 1. *A (γ, δ)-transaction order fairness mechanism Γ exists, if and only if there is no list of transactions $\mathsf{tx}_1, \mathsf{tx}_2, \ldots, \mathsf{tx}_k$ such that $\mathsf{tx}_1 \prec \mathsf{tx}_2 \cdots \prec \mathsf{tx}_k \prec \mathsf{tx}_1$.*

Notation: we call a list of transactions $\mathsf{tx}_1, \mathsf{tx}_2, \ldots, \mathsf{tx}_k$ such that $\mathsf{tx}_1 \prec \mathsf{tx}_2 \cdots \prec \mathsf{tx}_k \prec \mathsf{tx}_1$ as a *precede loop*.

Proof. (1) If there exists a mechanism Γ, for any list of transactions $\mathsf{tx}_1, \mathsf{tx}_2, \ldots, \mathsf{tx}_k$ such that $\mathsf{tx}_1 \prec \mathsf{tx}_2 \cdots \prec \mathsf{tx}_k \prec \mathsf{tx}_1$, then Γ should output tx_1 before tx_2, tx_2 before tx_3, ..., and tx_k before tx_1. This is a contradiction, since the output of Γ is a total ordering of transactions. (2) Now assume there is no list of transactions

$\mathsf{tx}_1, \mathsf{tx}_2, \ldots, \mathsf{tx}_k$ such that $\mathsf{tx}_1 \prec \mathsf{tx}_2 \cdots \prec \mathsf{tx}_k \prec \mathsf{tx}_1$. We observe that for any set of transactions, one of it is not constraint to be preceded by other transactions, since otherwise every transaction is preceded by some other transaction so that a precede loop exists. We can construct a mechanism Γ by repeating the following: pick one transaction tx from the set of all unpicked transactions, such that there is no transaction tx' such that $\mathsf{tx}' \prec \mathsf{tx}$. Γ outputs transactions by the order they are picked.

Basic bounds for γ We naturally require that γ be in $(m, h]$. If γ is smaller than m, then malicious nodes can always report they receive a transaction tx_1 before another transaction tx_2 while honest nodes report the opposite. $\mathsf{tx}_1 \prec \mathsf{tx}_2$ and $\mathsf{tx}_2 \prec \mathsf{tx}_1$ can both hold, which leads to a precede loop, so that no mechanism achieves our fairness property. If γ is larger than h, if malicious nodes always report the opposite of (the majority of) honest nodes, then the number of nodes that report to receive tx_1 before tx_2 is at most hn, which is smaller than γn. No meaningful precede constraint is placed on the transactions, and the mechanism can output any order of transactions.

In some cases (small δ or asynchronous network), we require γ to be larger than $1/2$ to avoid the possibility of $\mathsf{tx}_1 \prec \mathsf{tx}_2$ and $\mathsf{tx}_2 \prec \mathsf{tx}_1$ both holding. In other cases (synchronous network and large δ), self-loop is impossible even for $\gamma \in (m, 1/2]$.

2 Results under the Synchronous Network Model

In this section, we assume that the synchronous delay is Δ, i.e., whenever an honest node receives a transaction tx at time t, all other honest nodes receive the transaction within the range $[t - \Delta, t + \Delta]$. More precisely, a transaction is received by honest nodes within a Δ-time window $[t', t' + \Delta]$, by picking t' as the earliest time for an honest node to receive it.

Pruning Due to the above assumption, a transaction ordering mechanism can prune the reports. If more than mn nodes report to receive a transaction tx at or before time t, then the mechanism can prune the reports of all nodes that report to receive tx at time larger than $t + \Delta$. This is because if an honest node receives tx by t, other nodes must have received it by $t + \Delta$. Similarly, if more than hn nodes report to receive a transaction tx at or after time t, then the mechanism can prune the reports of all nodes that report to receive tx at time before $t - \Delta$. If a node is pruned, its report is completely discarded. Pruning only excludes malicious nodes. Pruning is applied exhaustively, until no more pruning can be applied. In subsequent steps of pruning, the pruning threshold is defined as $n' - hn$ where n' is the number of remaining valid reports. This is safe because we only exclude malicious nodes in pruning.

At the end of exhaustive pruning, we have a set of valid reports. Let n' be the number of valid reports. $n - n'$ nodes are excluded, all of which are malicious nodes. The number of remaining honest nodes is hn and the number

of remaining malicious nodes is $n' - hn$. We claim that if more than $n' - hn$ nodes report to receive tx by time t, then all valid reports receive tx by time $t + \Delta$. Similarly, if more than $n' - hn$ nodes report to receive tx after time t, then all valid reports receive tx after time $t - \Delta$. The threshold $n' - hn$ translates to a ratio of $1 - (n/n')h$. $1 - (n/n')h \leq 1 - h = m$.

Missing transactions and truncation The set of transactions reported by different nodes might be different. A mechanism needs a rule to handle missing transactions. The main possible issue is missing a preceding constraint. More specifically, tx_1 and tx_2 are reported by some nodes in the current slot, but a precede relation $\mathsf{tx}_1 \prec \mathsf{tx}_2$ lacks attestation in the current slot. In the next report slot, some others nodes report tx_1 or tx_2 for the first time and reports to receive tx_1 before tx_2 by δ. In this section, we assume all nodes share a world clock and send reports at exactly pre-specified time.

We use the following general solution: transaction ordering mechanisms only consider transactions that have been received by at least hn nodes Δ time before the prescribed report time. For example, if a report is scheduled at time t, then the mechanism only considers transactions that have been received by at least hn nodes before time $t - \Delta$. We prove that transactions ordered by such a mechanism will not violate precede constraints in future slots. Proof by contradiction, if $\mathsf{tx}_1 \prec \mathsf{tx}_2$ but the mechanism (1) outputs tx_2 before tx_1, or (2) outputs tx_2 but not tx_1. In case (1), at least hn nodes received tx_1 by $t - \Delta$, by pruning all reports of nodes that received tx_1 after t are pruned. The same applies to tx_2. This implies that all valid reports of tx_1 and tx_2 are already considered. (2) Since tx_2 is output, at least hn nodes received tx_2 by $t - \Delta$, by pruning, all valid reports received tx_2 by t. If $\mathsf{tx}_1 \prec \mathsf{tx}_2$, then all valid reports received tx_1 by $t - \delta$, which should appear in the reports of the current slot.

Constructive Mechanism vs General Mechanism There are two flavors to show existence of transaction order fairness mechanisms. The first is to construct a mechanism that satisfies the fairness property, which we call a *constructive mechanism*. The second is to show that no precede loop exists, and then use a general mechanism to order the transactions.

2.1 A simple mechanism: median timestamp

We first present a simple mechanism that computes the median timestamp over nodes of every transaction, then orders the transactions by their median timestamps. If two transactions have the same median timestamp, we break the tie by hashes of the transactions.

The median timestamp mechanism uses pruning and completely discard invalid reports. The median is defined as the median among valid reports. For transactions appearing in some but not all valid reports, we compute the median timestamp by considering missing timestamps as $+\infty$. By truncation, for every transaction that the mechanism outputs, at least $hn > n/2$ timestamps are received so that its median timestamp is one of the actually reported timestamps (not $+\infty$).

2.2 $(\gamma \in (m, h], \delta \in (2\Delta, +\infty))$-fairness exists

For any sequence of transactions $\mathsf{tx}_1, \mathsf{tx}_2, \ldots, \mathsf{tx}_k$ such that $\mathsf{tx}_1 \prec \mathsf{tx}_2 \prec \cdots \prec \mathsf{tx}_k$, we show that $\mathsf{tx}_k \prec \mathsf{tx}_1$ does not hold.

Proof. For any tx_i and tx_{i+1}, $\mathsf{tx}_i \prec \mathsf{tx}_{i+1}$ implies that at least one honest node j satisfies $t_j(\mathsf{tx}_i) < t_j(\mathsf{tx}_i) - \delta$. For any other honest node j', we have $t_{j'}(\mathsf{tx}_i) \leq t_j(\mathsf{tx}_i) + \Delta$ and $t_{j'}(\mathsf{tx}_{i+1}) \geq t_j(\mathsf{tx}_{i+1}) - \Delta$. Therefore, we have $t_{j'}(\mathsf{tx}_i) < t_{j'}(\mathsf{tx}_{i+1}) - (\delta - 2\Delta) < t_{j'}(\mathsf{tx}_{i+1})$. This implies all honest nodes receive tx_i before tx_{i+1}. Therefore, all honest nodes receive tx_1 before tx_k. $I(\mathsf{tx}_k, \mathsf{tx}_1)$ is a subset of malicious nodes M. Therefore, $|I(\mathsf{tx}_k, \mathsf{tx}_1)| \leq mn < \gamma n$.

2.3 $(\gamma \in (1/2, h], \delta \in (\Delta, +\infty))$-fairness exists

We show that the median timestamp mechanism satisfies transaction order fairness for $\gamma \in (1/2, h]$ and $\delta \in (\Delta, +\infty)$. We show that whenever $\mathsf{tx}_1 \prec \mathsf{tx}_2$, the median timestamp of tx_1 is smaller than that of tx_2. But we prove it in the reverse direction, i.e., if the median timestamp of tx_1 is larger than or equal to that of tx_2, then $\mathsf{tx}_1 \prec \mathsf{tx}_2$ does not hold.

Proof. Let t_1, t_2 denote the median timestamp of tx_1 and tx_2. Let n' denote the number of valid reports. The assumption says that $t_1 \geq t_2$. More than $n'/2$ nodes report tx_1 at or after t_1. By pruning, all valid nodes report tx_1 at or after $t_1 - \Delta$. On the other hand, the set of valid nodes who report tx_2 at or before t_2, denoted S_2, consist of more than $n'/2$ nodes. The set S_2 and set $I(\mathsf{tx}_1, \mathsf{tx}_2)$ are disjoint, because for every node $j \in S_2$, $t_j(\mathsf{tx}_1) - t_j(\mathsf{tx}_2) \geq t_1 - \Delta - t_2 \geq -\Delta > -\delta$. Then the set $|I(\mathsf{tx}_1, \mathsf{tx}_2)| < n'/2 \leq n/2 < \gamma n$.

2.4 $(\gamma \in (\frac{k}{k+1}, h], \delta \in (\Delta/k, +\infty)$-fairness exists

Can we achieve fairness with smaller δ? The answer is yes, if we increase the ratio γ.

Lemma 2. *When $\gamma \in (1 - \frac{1}{k}, h]$ and $\delta > 0$, for any sequence of $j \leq k$ transactions $\mathsf{tx}_1, \mathsf{tx}_2, \ldots, \mathsf{tx}_j$ such that $\mathsf{tx}_1 \prec \mathsf{tx}_2 \prec \cdots \prec \mathsf{tx}_j$, $\mathsf{tx}_j \prec \mathsf{tx}_1$ does not hold.*

Proof. Let $S(\mathsf{tx}, \mathsf{tx}')$ denote the set of nodes that receive transaction tx before transaction tx'. It is clear that $I(\mathsf{tx}, \mathsf{tx}') \subseteq S(\mathsf{tx}, \mathsf{tx}')$. Suppose $\mathsf{tx}_j \prec \mathsf{tx}_1$ holds. Then any set from $I(\mathsf{tx}_1, \mathsf{tx}_2), \ldots, I(\mathsf{tx}_j, \mathsf{tx}_1)$ has at least γn nodes. Therefore, the intersection of these k sets has at least $j\gamma n - (j-1)n = jn(\gamma - (1 - \frac{1}{j})) > jn(1 - \frac{1}{k} - (1 - \frac{1}{j})) \geq 0$ nodes. Take one node from this non-empty intersection, it receives tx_1 before tx_j and receives tx_j before tx_1. This is a contradiction.

The above lemma implies that we can increase the ratio γ to rule out more precede loops. However, the ratio will be 1 when we want to rule out all precede loops. We need another scheme to rule out all precede loops.

Lemma 3. *If* $\gamma \in (1-\frac{h}{k}, h]$ *and* $\delta > 2D/k$*, for any sequence of* $k+1$ *transactions* $\mathsf{tx}_1, \mathsf{tx}_2, \ldots, \mathsf{tx}_{k+1}$ *such that* $\mathsf{tx}_1 \prec \mathsf{tx}_2 \prec \cdots \prec \mathsf{tx}_{k+1}$*, all honest nodes receive* tx_1 *before* tx_{k+1}*.*

Proof.

Acknowledgments.

References

1. Kelkar, M., Zhang, F., Goldfeder, S., Juels, A.: Order-fairness for Byzantine consensus. In: CRYPTO (3). vol. 12172, pp. 451–480 (2020)