Ordering Transactions with Bounded Unfairness: Definitions, Complexity and Constructions

https://eprint.iacr.org/2023/1253

EUROCRYPT 2024

[1]University of Edinburgh [2]IOG [3]National and Kapodistrian University of Athens

Aggelos Kiayias[1,2], Nikos Leonardos[3], Yu Shen[1]

1. **Introduction**

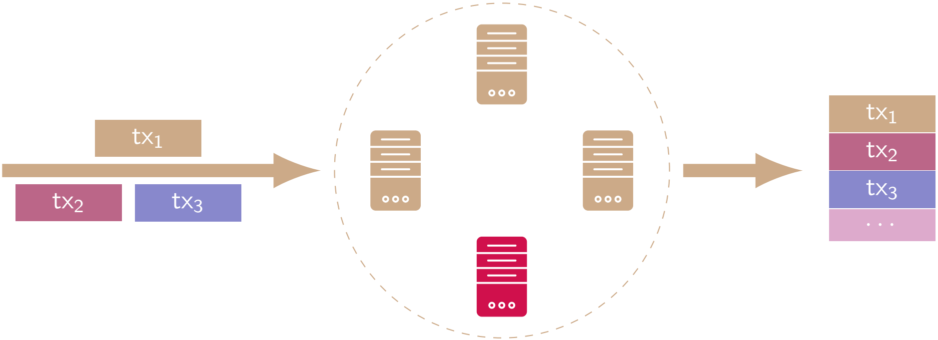
2. Directed-Bandwidth Order Fairness

3. Protocol Overview

4. Fairness vs. Liveness

5. Takeaways

State Machine Replication



State Machine Replication (Cont'd)

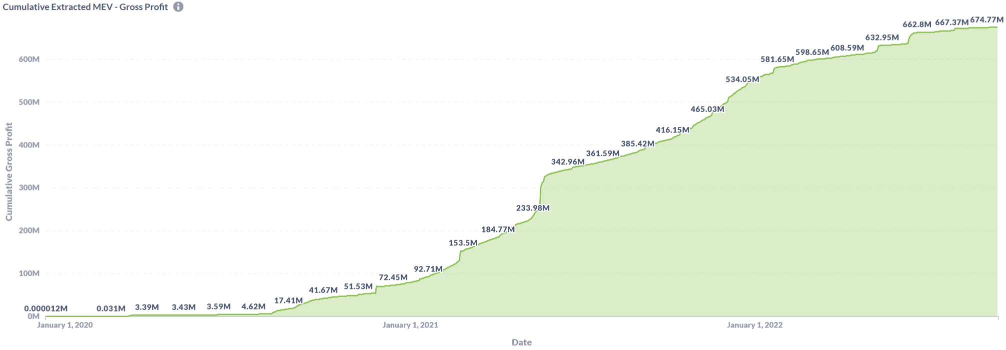
- ✓ **Consistency:** Honest parties output the same log (prefix).
- ✓ **Liveness:** New transactions are processed timely.

State Machine Replication (Cont'd)

- ✓ **Consistency:** Honest parties output the same log (prefix).
- ✓ **Liveness:** New transactions are processed timely.
- ✗ **Order-fairness:** Emulate the following behavior: A central server processes the commands it receives **sequentially**, in a **First-Come-First-Served** manner.
  - Re-ordering attacks: front-running, sandwich attacks, etc..

Maximal Extractable Value (MEV)

- Total Extracted MEV (Dec 2019 - Sep 2022) : $675,623,114.[1]



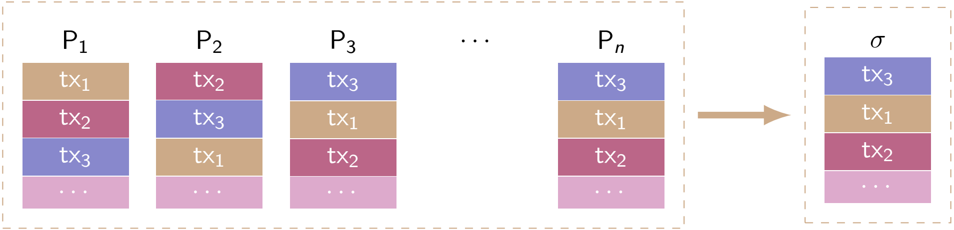
[1] Source: https://explore.flashbots.net/

Sender/Receiver Order Fairness

- **Sender order fairness:** Order transactions based on the time that they are sent.

Sender/Receiver Order Fairness

- **Sender order fairness:** Order transactions based on the time that they are sent.
- **Receiver order fairness:** Order transactions based on the time that they are received by protocol participants.



Condorcet Cycles

- A natural definition of fair order: $\mathsf{tx} \prec^{1/2+\epsilon} \mathsf{tx}' \implies \sigma(\mathsf{tx}) < \sigma(\mathsf{tx}')$.

Condorcet Cycles

- A natural definition of fair order: $\mathrm{tx} \prec^{1/2+\epsilon} \mathrm{tx}' \implies \sigma(\mathrm{tx}) < \sigma(\mathrm{tx}')$.

✗ **Impossible to achieve!** — Condorcet cycles.

Condorcet Cycles

- A natural definition of fair order: $\mathsf{tx} \prec^{1/2+\epsilon} \mathsf{tx}' \implies \sigma(\mathsf{tx}) < \sigma(\mathsf{tx}')$.

× **Impossible to achieve!** — Condorcet cycles.

P_1	P_2	P_3
tx_1	tx_2	tx_3
tx_2	tx_3	tx_1
tx_3	tx_1	tx_2

- $\mathsf{tx}_1 \prec \mathsf{tx}_2, \mathsf{tx}_2 \prec \mathsf{tx}_3, \mathsf{tx}_3 \prec \mathsf{tx}_1$

Condorcet Cycles

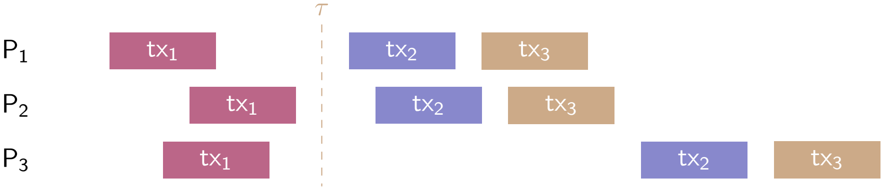
- A natural definition of fair order: $\mathsf{tx} \prec^{1/2+\epsilon} \mathsf{tx}' \implies \sigma(\mathsf{tx}) < \sigma(\mathsf{tx}')$.
- × **Impossible to achieve!** — Condorcet cycles.

P_1	P_2	P_3
tx_1	tx_2	tx_3
tx_2	tx_3	tx_1
tx_3	tx_1	tx_2

- $\mathsf{tx}_1 \prec \mathsf{tx}_2, \mathsf{tx}_2 \prec \mathsf{tx}_3, \mathsf{tx}_3 \prec \mathsf{tx}_1$
- Cycles can be chained to arbitrary size.

Timed Order Fairness

- Order $\mathsf{tx} \prec \mathsf{tx}'$ if their receiving time are sufficiently separated by a time τ. [Zha+20; Kur20]

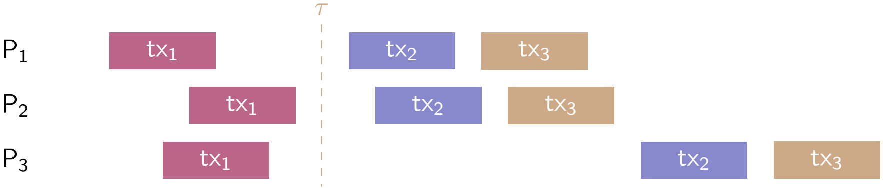


[Zha+20] Yunhao Zhang, Srinath T. V. Setty, Qi Chen, Lidong Zhou, and Lorenzo Alvisi. "Byzantine Ordered Consensus without Byzantine Oligarchy".

[Kur20] Klaus Kursawe. "Wendy, the Good Little Fairness Widget: Achieving Order Fairness for Blockchains".

Timed Order Fairness

- Order $\mathsf{tx} \prec \mathsf{tx}'$ if their receiving time are sufficiently separated by a time τ. [Zha+20; Kur20]



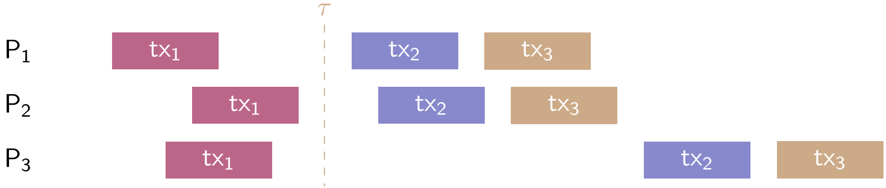
- We have $\mathsf{tx}_1 \prec \mathsf{tx}_2$ above; yet the order of tx_2 and tx_3 remains unspecified, even if honest parties unanimously saw tx_2 before tx_3.

[Zha+20] Yunhao Zhang, Srinath T. V. Setty, Qi Chen, Lidong Zhou, and Lorenzo Alvisi. "Byzantine Ordered Consensus without Byzantine Oligarchy".

[Kur20] Klaus Kursawe. "Wendy, the Good Little Fairness Widget: Achieving Order Fairness for Blockchains".

Timed Order Fairness

- Order $\mathrm{tx} \prec \mathrm{tx}'$ if their receiving time are sufficiently separated by a time τ. [Zha+20; Kur20]



- We have $\mathrm{tx}_1 \prec \mathrm{tx}_2$ above; yet the order of tx_2 and tx_3 remains unspecified, even if honest parties unanimously saw tx_2 before tx_3.

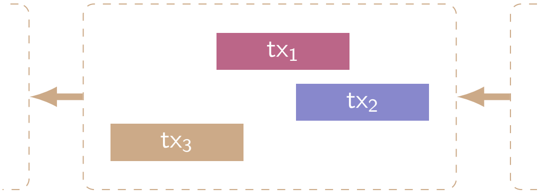
✗ Give up on ordering many transactions when their dissemination windows overlap.

[Zha+20] Yunhao Zhang, Srinath T. V. Setty, Qi Chen, Lidong Zhou, and Lorenzo Alvisi. "Byzantine Ordered Consensus without Byzantine Oligarchy".

[Kur20] Klaus Kursawe. "Wendy, the Good Little Fairness Widget: Achieving Order Fairness for Blockchains".

Batch Order Fairness

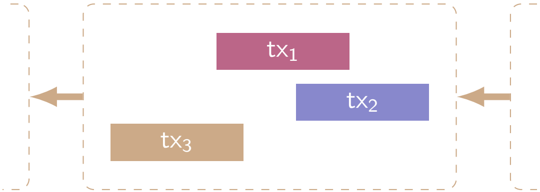
- If Condorcet cycle exists, order them in a “batch”. [Kel+20]



[Kel+20] Mahimna Kelkar, Fan Zhang, Steven Goldfeder, and Ari Juels. “Order-Fairness for Byzantine Consensus”.

Batch Order Fairness

- If Condorcet cycle exists, order them in a "batch". [Kel+20]

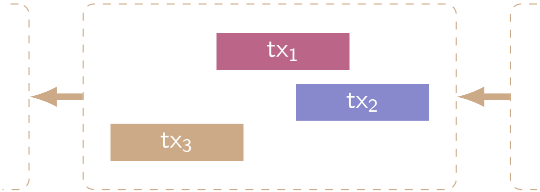


✓ No dependence on shared notion of time.

[Kel+20] Mahimna Kelkar, Fan Zhang, Steven Goldfeder, and Ari Juels. "Order-Fairness for Byzantine Consensus".

Batch Order Fairness

- If Condorcet cycle exists, order them in a "batch". [Kel+20]



✓ No dependence on shared notion of time.

✗ Give up on assigning unique index to each transaction.

[Kel+20] Mahimna Kelkar, Fan Zhang, Steven Goldfeder, and Ari Juels. "Order-Fairness for Byzantine Consensus".

Bounded Unfairness

- **Question:** Since it is **unavoidable** to order transactions **unfairly**, can we minimize the number of transactions between any pair of transactions that violate fair order?

Bounded Unfairness

- **Question:** Since it is **unavoidable** to order transactions **unfairly**, can we minimize the number of transactions between any pair of transactions that violate fair order?
- In the context of state machine replication this translates to an important guarantee: Can we minimize the number of **unfair state updates** occurring prior to any given transaction?

Bounded Unfairness

- **Question:** Since it is **unavoidable** to order transactions **unfairly**, can we minimize the number of transactions between any pair of transactions that violate fair order?
- In the context of state machine replication this translates to an important guarantee: Can we minimize the number of **unfair state updates** occurring prior to any given transaction?
  - For example, in DeFi bounded unfairness can minimize the "unfair slippage" when multiple users interact with an AMM.

Bounded Unfairness (Cont'd)

- Ideally, $\mathsf{tx} \prec^{1/2+\epsilon} \mathsf{tx}' \implies \sigma(\mathsf{tx}) < \sigma(\mathsf{tx}')$.

Bounded Unfairness (Cont'd)

- Ideally, $\mathsf{tx} \prec^{1/2+\epsilon} \mathsf{tx}' \implies \sigma(\mathsf{tx}) < \sigma(\mathsf{tx}')$.
- When it is forced to order $\mathsf{tx}, \mathsf{tx}'$ unfairly, tx' should not be ordered too earlier than tx.

Bounded Unfairness (Cont'd)

- Ideally, $\mathsf{tx} \prec^{1/2+\epsilon} \mathsf{tx}' \implies \sigma(\mathsf{tx}) < \sigma(\mathsf{tx}')$.
- When it is forced to order $\mathsf{tx}, \mathsf{tx}'$ unfairly, tx' should not be ordered too earlier than tx.

Definition ((φ, B)-fair-order). *Profile σ is a (φ, B)-fair-order on $\mathsf{P}_1, \ldots, \mathsf{P}_n$ if for all $\mathsf{tx}, \mathsf{tx}'$ such that $\mathsf{tx} \prec^{\varphi} \mathsf{tx}'$, it holds that $\sigma(\mathsf{tx}) - \sigma(\mathsf{tx}') \leq B$ where B is a function of $\mathsf{P}_1, \ldots, \mathsf{P}_n, \varphi, \mathsf{tx}$ and tx'.*

1. Introduction

2. **Directed-Bandwidth Order Fairness**

3. Protocol Overview

4. Fairness vs. Liveness

5. Takeaways

(φ, B)-Order-Fairness

Definition ((φ, B)-fair-order). *Profile σ is a (φ, B)-fair-order on $\mathsf{P}_1, \ldots, \mathsf{P}_n$ if for all $\mathsf{tx}, \mathsf{tx}'$ such that $\mathsf{tx} \prec^{\varphi} \mathsf{tx}'$, it holds that $\sigma(\mathsf{tx}) - \sigma(\mathsf{tx}') \leq B$ where B is a function of $\mathsf{P}_1, \ldots, \mathsf{P}_n, \varphi, \mathsf{tx}$ and tx'.*

(φ, B)-Order-Fairness

Definition ((φ, B)-fair-order). *Profile σ is a (φ, B)-fair-order on $\mathsf{P}_1, \ldots, \mathsf{P}_n$ if for all $\mathsf{tx}, \mathsf{tx}'$ such that $\mathsf{tx} \prec^{\varphi} \mathsf{tx}'$, it holds that $\sigma(\mathsf{tx}) - \sigma(\mathsf{tx}') \leq B$ where B is a function of $\mathsf{P}_1, \ldots, \mathsf{P}_n, \varphi, \mathsf{tx}$ and tx'.*

- (φ, B)-order-fairness is **unrealizable** if B is too “small” on some transaction pairs.
  - E.g., $B = 0$ for $\mathsf{tx}, \mathsf{tx}'$ in a Condorcet cycle.

(φ, B)-Order-Fairness

Definition ((φ, B)-fair-order). *Profile σ is a (φ, B)-fair-order on $\mathsf{P}_1, \ldots, \mathsf{P}_n$ if for all $\mathsf{tx}, \mathsf{tx}'$ such that $\mathsf{tx} \prec^{\varphi} \mathsf{tx}'$, it holds that $\sigma(\mathsf{tx}) - \sigma(\mathsf{tx}') \leq B$ where B is a function of $\mathsf{P}_1, \ldots, \mathsf{P}_n, \varphi, \mathsf{tx}$ and tx'.*

- (φ, B)-order-fairness is **unrealizable** if B is too "small" on some transaction pairs.
  - E.g., $B = 0$ for $\mathsf{tx}, \mathsf{tx}'$ in a Condorcet cycle.
- (φ, B)-order-fairness is **trivial** if B is too "large" on some transaction pairs.
  - E.g., $B = |\sigma| - 1$ for some $\sigma, \mathsf{tx}, \mathsf{tx}'$.

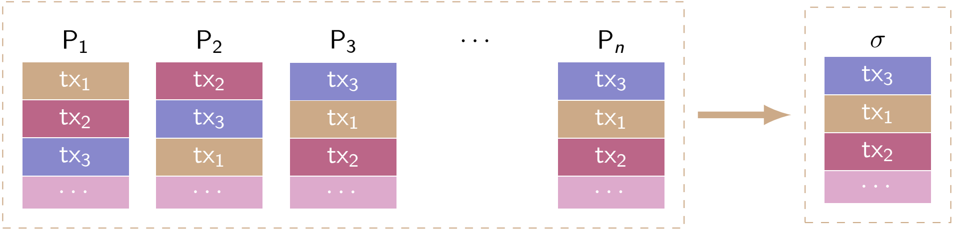
(φ, B)-Order-Fairness

Definition ((φ, B)-fair-order). *Profile σ is a (φ, B)-fair-order on $\mathsf{P}_1, \ldots, \mathsf{P}_n$ if for all $\mathsf{tx}, \mathsf{tx}'$ such that $\mathsf{tx} \prec^{\varphi} \mathsf{tx}'$, it holds that $\sigma(\mathsf{tx}) - \sigma(\mathsf{tx}') \leq B$ where B is a function of $\mathsf{P}_1, \ldots, \mathsf{P}_n, \varphi, \mathsf{tx}$ and tx'.*

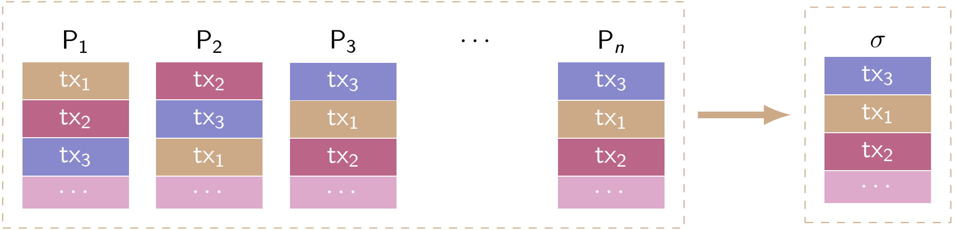
- (φ, B)-order-fairness is **unrealizable** if B is too "small" on some transaction pairs.
  - E.g., $B = 0$ for $\mathsf{tx}, \mathsf{tx}'$ in a Condorcet cycle.
- (φ, B)-order-fairness is **trivial** if B is too "large" on some transaction pairs.
  - E.g., $B = |\sigma| - 1$ for some $\sigma, \mathsf{tx}, \mathsf{tx}'$.

Question: *How to explicitly define the smallest possible function B?*

Transaction Dependency Graphs

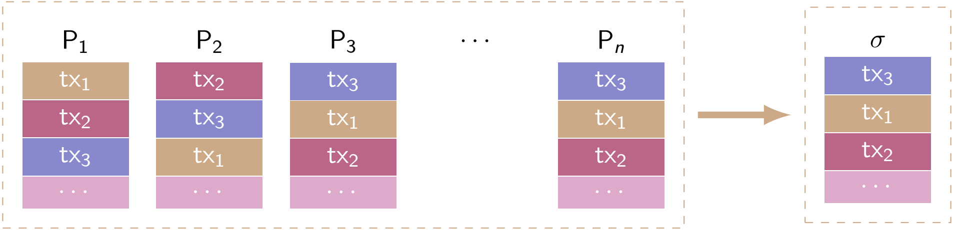


Transaction Dependency Graphs



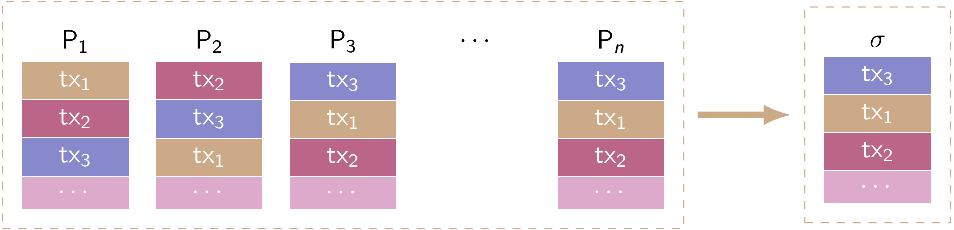
- Honest parties' transaction log can be converted to a dependency graph $G(\mathcal{P}, \varphi)$.
  - For each transaction, add a node; for $\mathsf{tx} \prec^{\varphi} \mathsf{tx}'$, add an edge $\mathsf{tx} \to \mathsf{tx}'$.

Transaction Dependency Graphs



- Honest parties' transaction log can be converted to a dependency graph $G(\mathcal{P}, \varphi)$.
  - For each transaction, add a node; for $\mathsf{tx} \prec^{\varphi} \mathsf{tx}'$, add an edge $\mathsf{tx} \to \mathsf{tx}'$.
- Condorcet cycles = Strongly connected components.

Transaction Dependency Graphs



- Honest parties' transaction log can be converted to a dependency graph $G(\mathcal{P}, \varphi)$.
  - For each transaction, add a node; for $\mathsf{tx} \prec^{\varphi} \mathsf{tx}'$, add an edge $\mathsf{tx} \to \mathsf{tx}'$.
- Condorcet cycles $=$ Strongly connected components.
- Large $\varphi \implies$ large cycles.

Theorem. *$G(\mathcal{P}, \varphi)$ does not contain a cycle of size $k < 1/(1-\varphi)$.*

Directed Bandwidth Problem

- Dependency graphs are oriented graphs.

Directed Bandwidth Problem

- Dependency graphs are oriented graphs.
- For a vertex ordering on $G(\mathcal{P}, \varphi)$, we only care about **backward edges**.
  - Backward edges imply that transactions were ordered "unfairly."

Directed Bandwidth Problem

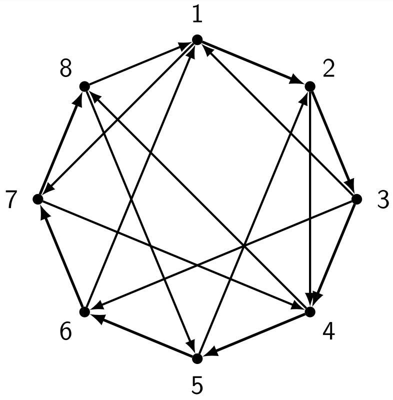
- Dependency graphs are oriented graphs.
- For a vertex ordering on $G(\mathcal{P}, \varphi)$, we only care about **backward edges**.
  - Backward edges imply that transactions were ordered "unfairly."

Definition (Directed Bandwidth). *Given a directed graph* $G = (V, E)$, DirectedBandwidth *asks to find a vertex ordering* σ^* *such that* $\mathtt{DBW}(\sigma^*, G) = \min_\sigma \mathtt{DBW}(\sigma, G)$ *where*

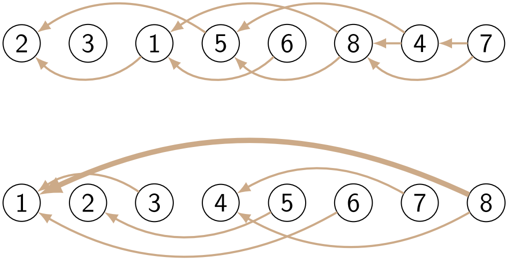
$$\mathtt{DBW}(\sigma, G) = \max_{\substack{(u,v)\in E,\\ \sigma(u)>\sigma(v)}} \sigma(u) - \sigma(v).$$

The directed bandwidth of a graph G *is* $\mathtt{DBW}(G) = \mathtt{DBW}(\sigma^*, G)$.

Directed Bandwidth Problem (Cont'd)



Dependency graph $G_{\mathcal{P},\varphi}$



Bandwidth-optimal vertex ordering (above)
and a "bad" ordering (below)

Directed Bandwidth Problem (Cont'd)

Theorem. DIRECTEDBANDWIDTH *is* NP*-hard and* NP*-hard to approximate within any constant ratio over oriented graphs.*

[Jai+19] Pallavi Jain, Lawqueen Kanesh, William Lochet, Saket Saurabh, and Roohani Sharma. "Exact and Approximate Digraph Bandwidth".

Directed Bandwidth Problem (Cont'd)

Theorem. DirectedBandwidth *is* NP-*hard and* NP-*hard to approximate within any constant ratio over oriented graphs.*

Theorem. *Let* $\mathbb{G}_n$ *denote the set of all oriented graphs with* n *vertices. It holds that*

$$n - 4\log n < \max_{G \in \mathbb{G}_n} \mathtt{DBW}(G) < n - \log n/2.$$

[Jai+19] Pallavi Jain, Lawqueen Kanesh, William Lochet, Saket Saurabh, and Roohani Sharma. "Exact and Approximate Digraph Bandwidth".

Directed Bandwidth Problem (Cont'd)

Theorem. DirectedBandwidth *is* NP*-hard and* NP*-hard to approximate within any constant ratio over oriented graphs.*

Theorem. *Let* $\mathbb{G}_n$ *denote the set of all oriented graphs with* n *vertices. It holds that*

$$n - 4\log n < \max_{G \in \mathbb{G}_n} \mathtt{DBW}(G) < n - \log n/2.$$

- DirectedBandwidth can be solved trivially in $\mathcal{O}^*(n!)$ time.
- DirectedBandwidth can be solved in $\mathcal{O}^*(2^{|E|} \cdot 3^{|V|})$ time. [Jai+19]

[Jai+19] Pallavi Jain, Lawqueen Kanesh, William Lochet, Saket Saurabh, and Roohani Sharma. "Exact and Approximate Digraph Bandwidth".

Directed-Bandwidth Order Fairness

Definition ($(\varphi, \mathtt{DBW})$-fair-order). *Profile σ is a $(\varphi, \mathtt{DBW})$-fair-order on $\mathcal{P} = \mathsf{P}_1, \ldots, \mathsf{P}_n$ if for all $\mathsf{tx}, \mathsf{tx}'$ such that $\mathsf{tx} \prec^{\varphi} \mathsf{tx}'$, it holds that*

$$\sigma(\mathsf{tx}) - \sigma(\mathsf{tx}') \leq \mathtt{DBW}\big(\mathsf{SCC}(G(\mathcal{P}, \varphi), \mathsf{tx}, \mathsf{tx}')\big),$$

where $\mathsf{SCC}(G, \mathsf{tx}, \mathsf{tx}')$ is a function that outputs an SCC in G that contains both $\mathsf{tx}, \mathsf{tx}'$ if it exists, and a null graph otherwise.

Directed-Bandwidth Order Fairness

Definition (($\varphi, \mathtt{DBW}$)-fair-order). *Profile σ is a $(\varphi, \mathtt{DBW})$-fair-order on $\mathcal{P} = \mathsf{P}_1, \ldots, \mathsf{P}_n$ if for all $\mathsf{tx}, \mathsf{tx}'$ such that $\mathsf{tx} \prec^{\varphi} \mathsf{tx}'$, it holds that*

$$\sigma(\mathsf{tx}) - \sigma(\mathsf{tx}') \leq \mathtt{DBW}\big(\mathsf{SCC}(G(\mathcal{P}, \varphi), \mathsf{tx}, \mathsf{tx}')\big),$$

where $\mathsf{SCC}(G, \mathsf{tx}, \mathsf{tx}')$ is a function that outputs an SCC in G that contains both $\mathsf{tx}, \mathsf{tx}'$ if it exists, and a null graph otherwise.

- $\mathsf{tx} \prec^{\varphi} \mathsf{tx}'$ and $\mathsf{tx}, \mathsf{tx}'$ are not in the same cycle $\implies \sigma(\mathsf{tx}) < \sigma(\mathsf{tx}')$.

Directed-Bandwidth Order Fairness

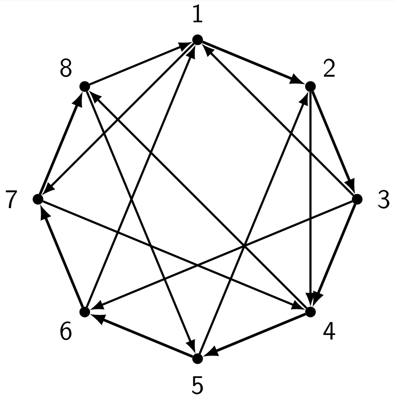
Definition ((φ, $\mathtt{DBW}$)-fair-order). *Profile σ is a $(\varphi, \mathtt{DBW})$-fair-order on $\mathcal{P} = \mathsf{P}_1, \ldots, \mathsf{P}_n$ if for all $\mathsf{tx}, \mathsf{tx}'$ such that $\mathsf{tx} \prec^{\varphi} \mathsf{tx}'$, it holds that*

$$\sigma(\mathsf{tx}) - \sigma(\mathsf{tx}') \leq \mathtt{DBW}\big(\mathsf{SCC}(G(\mathcal{P}, \varphi), \mathsf{tx}, \mathsf{tx}')\big),$$

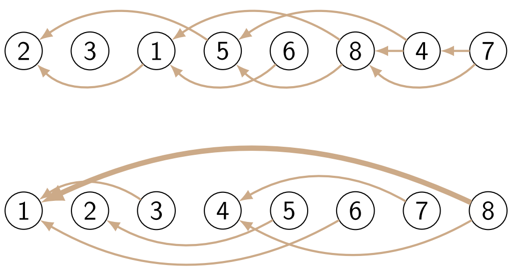
where $\mathsf{SCC}(G, \mathsf{tx}, \mathsf{tx}')$ is a function that outputs an SCC in G that contains both $\mathsf{tx}, \mathsf{tx}'$ if it exists, and a null graph otherwise.

- $\mathsf{tx} \prec^{\varphi} \mathsf{tx}'$ and $\mathsf{tx}, \mathsf{tx}'$ are not in the same cycle $\implies \sigma(\mathsf{tx}) < \sigma(\mathsf{tx}')$.
- $\mathsf{tx} \prec^{\varphi} \mathsf{tx}'$ and $\mathsf{tx}, \mathsf{tx}'$ are in the same cycle $C \implies \sigma(\mathsf{tx}) - \sigma(\mathsf{tx}') < \mathtt{DBW}(C)$.

Directed-Bandwidth Order Fairness (Cont'd)



Dependency graph $G_{\mathcal{P},\varphi}$



Serialization with bounded unfairness (above) and a "bad" serialization (below)

Directed-Bandwidth Order Fairness (Cont'd)

- $(\varphi, \texttt{DBW})$-Fair-Order is the best possible bounded unfairness that is feasible.

Directed-Bandwidth Order Fairness (Cont'd)

- $(\varphi, \texttt{DBW})$-Fair-Order is the best possible bounded unfairness that is feasible.

Theorem. *Suppose that a protocol implements (φ, B)-fair-order for a function B. Then for all $\mathcal{P}$ there are $\mathrm{tx}, \mathrm{tx}'$ with $\mathrm{tx} \prec^{\varphi} \mathrm{tx}'$, such that B satisfies $B(\mathcal{P}, \varphi, \mathrm{tx}, \mathrm{tx}') \geq \texttt{DBW}(\mathrm{SCC}(G(\mathcal{P}, \varphi), \mathrm{tx}, \mathrm{tx}'))$.*

1. Introduction

2. Directed-Bandwidth Order Fairness

3. Protocol Overview

4. Fairness vs. Liveness

5. Takeaways

Protocol Design

1. Parties use 2×1 PoW to mine blocks and **transaction profiles** and include valid profiles in the blockchain.
2. Build transaction dependency graph using majority preferences in profiles.
3. Run DirectedBandwidth algorithm on SCCs and serialize transactions.

1. Introduction

2. Directed-Bandwidth Order Fairness

3. Protocol Overview

4. Fairness vs. Liveness

5. Takeaways

Fairness vs. Livenss

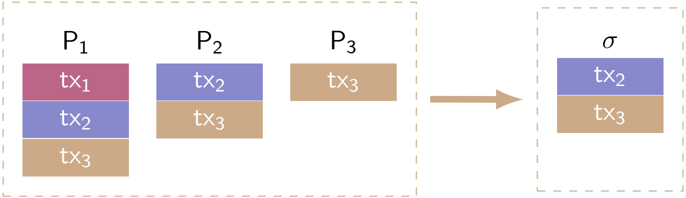
- In SMR problem, parties process an ever-growing transaction log.

Fairness vs. Livenss

- In SMR problem, parties process an ever-growing transaction log.
- Infinite condorcet cycles $\implies$ Failure of liveness.

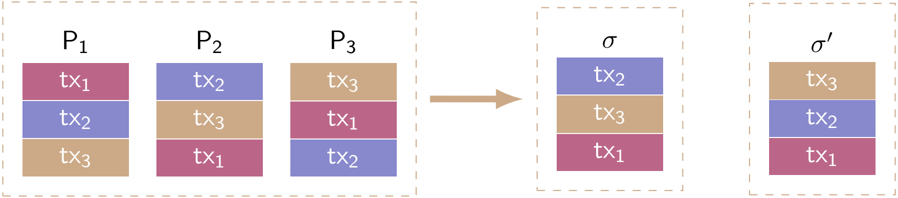
Fairness vs. Livenss

- In SMR problem, parties process an ever-growing transaction log.
- Infinite condorcet cycles $\implies$ Failure of liveness.



Fairness vs. Liveness

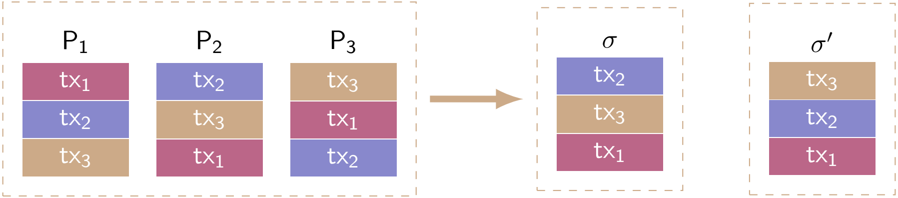
- In SMR problem, parties process an ever-growing transaction log.
- Infinite condorcet cycles $\Longrightarrow$ Failure of liveness.



Theorem. *Suppose the transaction dissemination is asynchronous, there is no protocol that can achieve consistency, liveness and* $(\varphi, \texttt{DBW})$*-order-fairness.*

Fairness vs. Liveness

- In SMR problem, parties process an ever-growing transaction log.
- Infinite condorcet cycles $\implies$ Failure of liveness.



Theorem. *Suppose the transaction dissemination is asynchronous, there is no protocol that can achieve consistency, liveness and* $(\varphi, \mathtt{DBW})$*-order-fairness.*

- We can hope to achieve "weak liveness."

1. Introduction

2. Directed-Bandwidth Order Fairness

3. Protocol Overview

4. Fairness vs. Liveness

5. Takeaways

Takeaways

- **Definitions:** We define fair transaction serialization with **bounded unfairness**.
- **Complexity:** We analyze the complexity of achieving bounded unfairness.
- **Constructions:** We design a new permissionless blockchain protocol that achieves consistency, weak liveness and bounded unfairness (`DBW`).
  - We also relax bounded unfairness into a "timed" version that enables the protocol to offer consistency, liveness and bounded unfairness.

Thank You

Thank You

https://eprint.iacr.org/2023/1253

References

[Jai+19] Pallavi Jain, Lawqueen Kanesh, William Lochet, Saket Saurabh, and Roohani Sharma. "Exact and Approximate Digraph Bandwidth". In: **39th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2019, December 11-13, 2019, Bombay, India**. Ed. by Arkadev Chattopadhyay and Paul Gastin. Vol. 150. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019, 18:1–18:15.

[Kel+20] Mahimna Kelkar, Fan Zhang, Steven Goldfeder, and Ari Juels. "Order-Fairness for Byzantine Consensus". In: **Advances in Cryptology - CRYPTO 2020 - 40th Annual International Cryptology Conference, CRYPTO 2020, Santa Barbara, CA, USA, August 17-21, 2020, Proceedings, Part III**. Ed. by Daniele Micciancio and Thomas Ristenpart. Vol. 12172. Lecture Notes in Computer Science. Springer, 2020, pp. 451–480.

[Kur20] Klaus Kursawe. "Wendy, the Good Little Fairness Widget: Achieving Order Fairness for Blockchains". In: **AFT '20: 2nd ACM Conference on Advances in Financial Technologies, New York, NY, USA, October 21-23, 2020**. ACM, 2020, pp. 25–36.

[Zha+20] Yunhao Zhang, Srinath T. V. Setty, Qi Chen, Lidong Zhou, and Lorenzo Alvisi. "Byzantine Ordered Consensus without Byzantine Oligarchy". In: **14th USENIX Symposium on Operating Systems Design and Implementation, OSDI 2020, Virtual Event, November 4-6, 2020**. USENIX Association, 2020, pp. 633–649.