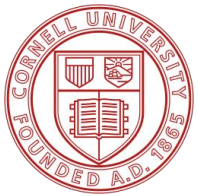
Equal Opportunity: A Correctness Condition for Ordered Consensus

Yunhao Zhang*[1], **Haobin Ni***[2], Soumya Basu[3], Shir Cohen[1], Maofan Yin[4], Lorenzo Alvisi[1], Robbert van Renesse[1], Qi Chen[5], Lidong Zhou[5]

*Equal contribution





OpenReserve Holdings

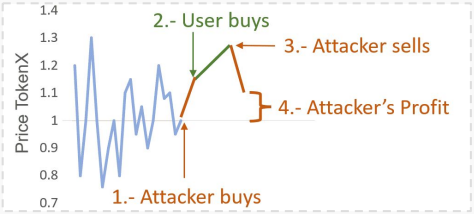


Microsoft Research

Order! Order!

Classic BFT Consensus (e.g., PBFT)

- Output order only guarantees to be **total**
- A Byzantine leader can dictate ordering



However, ordering can be critical!

- Ordering of financial transactions affects their gain
- **Front-running** and **sandwich attacks** rely on specific ordering to profit

$7.2 million

per month

is paid to Ethereum validators to guarantee certain transactions remained in the first quartile of the block

[Mizrach & Yoshida, 25]

Order! Order!

Ordered Consensus (e.g., Pompē, Themis)

- Output order is determined by servers' observations on transactions
- Byzantine influence on ordering is **restricted**

However, servers' observations are **not necessarily fair**

- Protocols favor clients with fast network to majority of the nodes
- A marginal advantage leads to huge gain due to determinism
  - Result in "**latency arms race**"

I *love* the smell of

overran transactions

in the morning!



Can we build a system that is

not pay to win?



Our Contributions

1. Formal definitions for specifying **fairness** in ordering for BFT-SMR
2. Theoretically **optimal trade-off** curve between fairness and latency
3. New BFT-SMR system artifacts providing fairness using **randomness**
4. Empirical evaluation showing **biases** in the existing systems and how our system can **provide fairness at an acceptable overhead**

01

What is fairness?

Fairness in accepting papers

Accepting papers based on the quality of work => *Fair*

- Quality of work is a **relevant feature** for acceptance

Accepting papers based on the authors' identity => *Unfair*

- Authors' identity is an **irrelevant feature** for acceptance

Biases: Irrelevant features affect the output of a system

- If the same paper submitted by different authors would get different treatments, then the system is **biased**

Bias in ordering transactions

Pompē orders transactions by their medium timestamps from 2f+1 servers

- timestamp = invocation time (**relevant**) + network latency (**irrelevant**)

This creates **a bias favoring low latency connections** to 2f+1 servers

- Transactions issued physically close to 2f+1 servers and/or delivered to them through a faster network will be ordered earlier
- Exploitable by front-runners!

Ideal Fairness: Impartiality

Impartiality: Transactions with the same relevant features should be ordered uniformly randomly across all possible permutations

Example: Two transactions issued simultaneously => **50% chance** of either order

However, Byzantine behaviors makes perfect impartiality infeasible under the assumption that the adversary can always distinguish transactions

ε-Ordering Equality

Let **U** be the set of all transactions.

A system satisfies ε-ordering equality iff:

For any **S**, a subset of **U** containing **n** transactions with the same relevant features

for any total order **>** over **S**,

let **Pr[>]** be the probability of **>** being the output order of transactions in **S**,

$$\left|Pr[>] - \frac{1}{n!}\right| \leq \varepsilon(n)$$

Realty — Ideal — Function over n

Δ-Ordering Separation

For any two transactions $\mathbf{i_1}$ and $\mathbf{i_2} \in U$,

if $\mathbf{i_1}$ is invoked at least Δ time earlier than $\mathbf{i_2}$,

then $\mathbf{i_1}$ is guaranteed to be ordered before $\mathbf{i_2}$ in the output order.

An upper bound on the time a transaction has to wait until finalized!

02

How to achieve fairness?

Mitigate biases with randomness

score = relevant feature + **biased** irrelevant feature + **fair** random noise

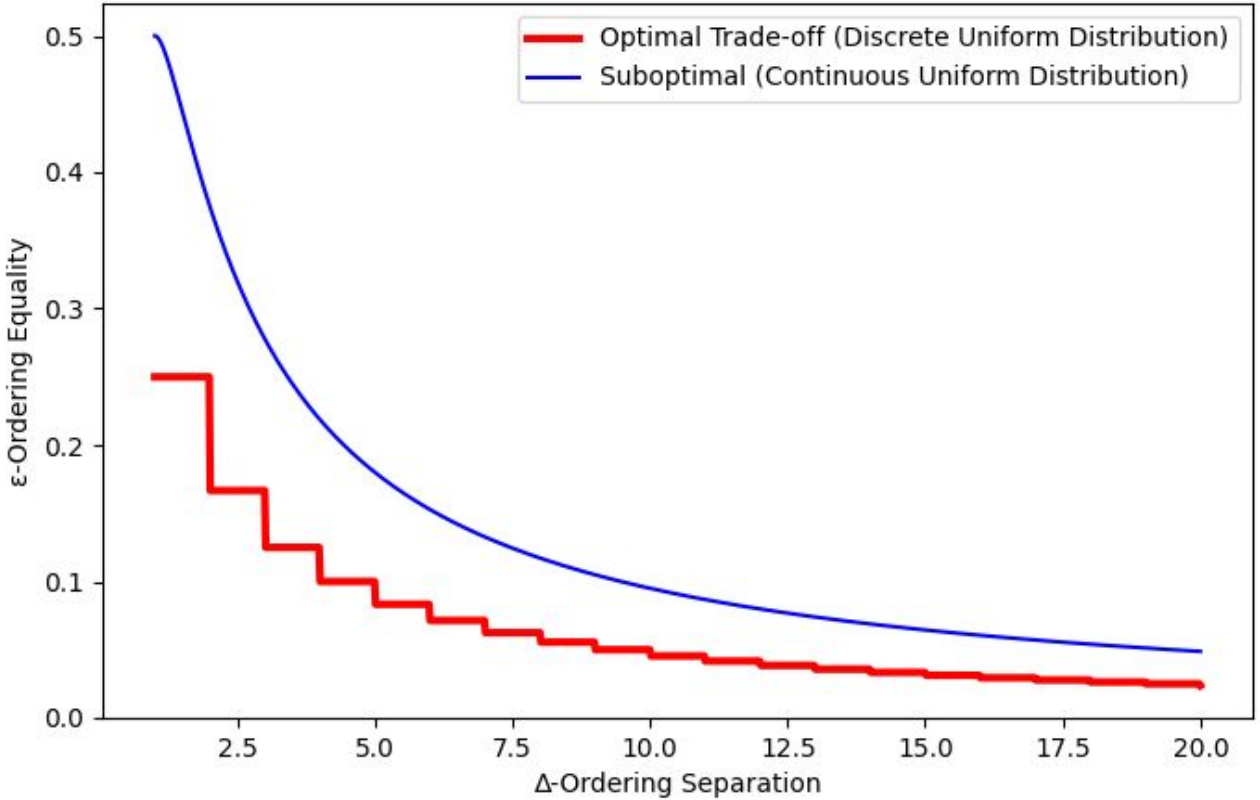
Observation

The sweet spot:

relevant features >> random noise >> irrelevant features

The distribution of the random noise matters!

The Trade-off

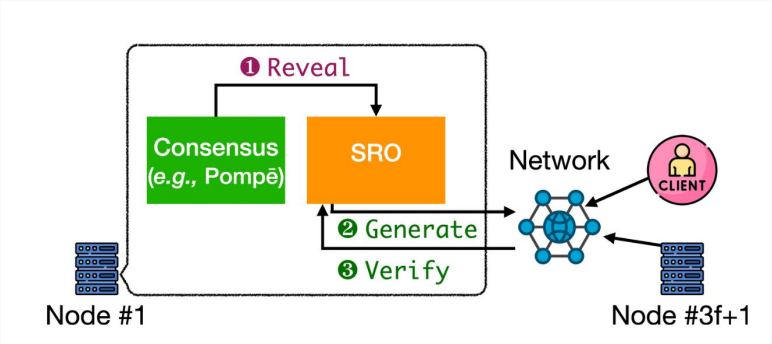


Challenge:

BFT Secret

Randomness

Secret Random Oracle (SRO)



Confidentiality: can only be computed after a block has been finalized by requiring a quorum showing the decision has been made

Two implementations:

trusted hardware and **threshold verifiable random function**

03

Did we achieve fairness?
What is the performance overhead?

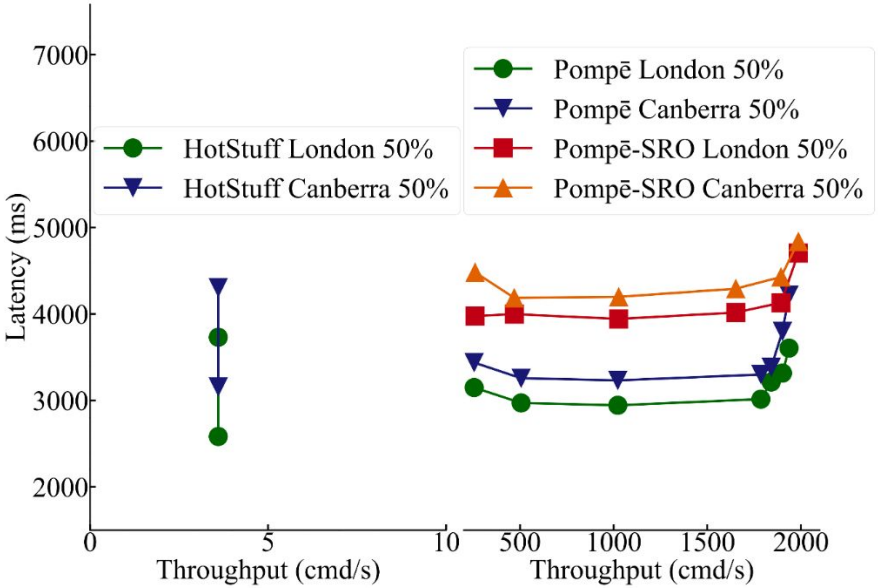
Evaluation: Fairness

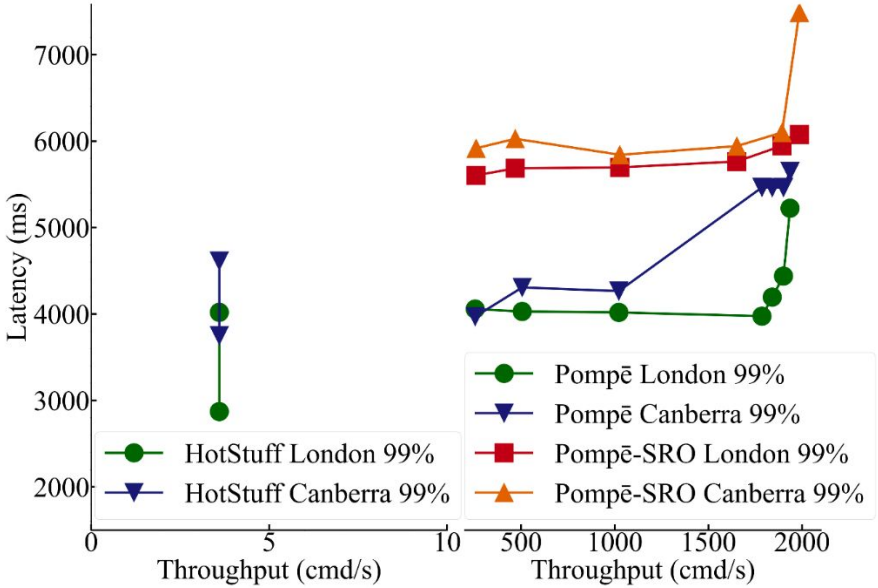
Simulated Ethereum server geo-distribution:

- ~40% US (West/Mid/East), ~40% Europe (Major cities), ~20% Asia + Oceania

	Pompē	Themis	HotStuff (Rotating)	Pompē-SRO Δnet	Pompē-SRO 5Δnet
London vs. Washington	100% L	100% W	**53% W**	62% L	**53% L**
Washington vs. Tokyo	100% W	100% W	92% W	65% W	**54% W**
London vs. Munich	100% L	100% L	74% L	**54% L**	**50% L**
Munich vs. Tokyo	100% M	100% M	74% M	73% M	**55% M**

Evaluation: Overhead





Takeaways

Inherent **biases** exist in consensus protocols.

Biases can be **mitigated** by introducing randomness ***carefully***.

Reusable insight: conceptual framework of **fairness**

More **equal opportunities** for future research!