A FAIR & DECENTRALIZED CLOCK NETWORK FOR TRANSACTION ORDERING

Andrei Constantinescu, Diana Ghinea, Lioba Heimbach, Zilin Wang, Roger Wattenhofer
ETH Zürich
TRANSACTIONS & FRONT-RUNNING
TRANSACTIONS & FRONT-RUNNING

Well... Alice found a good deal.
Well... Alice found a good deal. I'll bribe the validators to place my transaction first!
ADDITIONAL LAYER: CLOCKS

Alice found a good deal, but I’m a bit late....
ADDITIONAL LAYER: CLOCKS

We consider $n$ nodes equipped with clocks in an asynchronous network. Out of these nodes, $f < n/3$ may be byzantine.
ADDITIONAL LAYER: CLOCKS

1. Agree on a timestamp of receipt
2. Reconstruct $tx$ from the shares
WHAT IS A FAIR TIMESTAMP?

• If \texttt{tx1} is received before \texttt{tx2},
  then \texttt{tx1} ‘s timestamp should not be later than \texttt{tx2}’s timestamp.

... but what does that mean for a decentralized network?
WHAT IS A FAIR TIMESTAMP?

• If the median time when honest nodes receive tx1 is before the median time when honest nodes receive tx2, then tx1’s timestamp should not be later than tx2’s timestamp.

...Byzantine Agreement with Median Validity?
BYZANTINE AGREEMENT

Assume $n$ nodes hold input values.

Even when $f$ out of these nodes are byzantine:

(Termination) All honest nodes output some value.

(Agreement) All honest nodes output the same value.

(All-Same Validity) If all honest nodes hold the same input value, that’s the value they output.
BYZANTINE AGREEMENT + MEDIAN VALIDITY

Assume $n$ nodes hold input values.

Even when $f$ out of these nodes are byzantine:

(Termination) All honest nodes output some value.

(Agreement) All honest nodes output the same value.

(Median Validity) Honest nodes’ outputs are close to the honest inputs’ median.
$$\delta \text{-MEDIAN VALIDITY}$$

$$\tau_1 \leq \tau_2 \leq \cdots \leq \tau_{\text{median} - \delta} \leq \cdots \leq \tau_{\text{median}} \leq \cdots \leq \tau_{\text{median} + \delta} \leq \cdots \leq \tau_{n-f}$$

Any value in $[\tau_{\text{median} - \delta}, \tau_{\text{median} + \delta}]$ is valid.
\(\delta\) - MEDIAN VALIDITY

**Synchronous model:** \(\delta \geq f/2\) sufficient and necessary.

[OPODIS:StoWat15, SRDS:MelWat18]

**Asynchronous model:** ? ? ?
**δ - MEDIAN VALIDITY**

**Synchronous model:** $\delta \geq f/2$ sufficient and necessary.

[OPODIS:StoWat15, SRDS:MelWat18]

**Asynchronous model:** $\delta \geq f$ necessary!!!

This is quite weak when $f < n/3$...

...Compromise?
TIMESTAMP AGREEMENT

Assume \( n \) nodes hold (integer) timestamps as input values.

Even when \( f \) out of these nodes are byzantine:

(Termination) All honest nodes output some value.

(Agreement) All honest nodes output the same value.

(\( \delta \)-Median Validity) Honest outputs are in \([\tau_{\text{median}} - \delta, \tau_{\text{median}} + \delta]\).

For \( \delta = f/2 \) if the network is synchronous and \( \delta = f \) otherwise.
ACHIEVING TIMESTAMP AGREEMENT

1. Every node sends its initial timestamp to everyone.

   When sufficient time to allow for one synchronous round has passed and \( n - f + k \) values were received (with \( 0 \leq k \leq f \)):

   \[ \tau_{\text{median}} := \text{the} \left( \lfloor (n - f)/2 \rfloor + \lfloor k/2 \rfloor \right)-\text{th lowest value received.} \]
ACHIEVING TIMESTAMP AGREEMENT

1. Every node sends its initial timestamp to everyone. 

   When sufficient time to allow for one synchronous round has passed and \( n - f + k \) values were received (with \( 0 \leq k \leq f \)):

   \[ \tau_{\text{median}} := \text{the } [(n - f)/2] + [k/2]-\text{th lowest value received.} \]

=> \( \tau_{\text{median}} \) satisfies \( \delta \)-Median Validity for:

   synchronous network => \( \delta = f/2 \).  \( \delta \)-Median Validity ✓

   asynchronous network => \( \delta = f \).  Agreement ❌
ACHIEVING TIMESTAMP AGREEMENT

2. Join \texttt{ApproximateAgreement} with $\tau_{\text{median}}$ as input. Obtain output $\tau_{\text{AA}}$.

$\Rightarrow$ honest nodes obtain $\varepsilon$-close outputs $\tau_{\text{AA}}$ ($0 < \varepsilon < 0.5$) within the range of their inputs.

$\delta$-Median Validity $\checkmark$

Agreement Up to an error $< 0.5$
ACHIEVING TIMESTAMP AGREEMENT

3. Nodes need to decide whether to round their values $\tau_{AA}$ up or down, such that they end up with the same value.
3. Nodes need to decide whether to round their values \( \tau_{AA} \) up or down, such that they end up with the same value.

Run \text{ByzantineAgreement} with input \( b = 0 \) if the closest integer is even and \( b = 1 \) if the closest integer is odd.

Output the closest even integer if \text{output bit} = 0, otherwise the closest odd integer.
Thanks You & Summary

- Transactions are *fairly* ordered based on the ~median timestamp of receipt.
- The timestamp of receipt is decided by a network of nodes equipped with clocks.
- **Timestamp Agreement** protocol:
  - Asynchronous Byzantine Agreement with Median Validity.
  - Optimal resilience guarantees.
  - Optimal Median Validity guarantees for the actual network conditions.