OPTIMIZING EXIT QUEUES FOR PROOF-OF-STAKE BLOCKCHAINS

Michael Neuder
Mallesh Pai
Max Resnick

AFT 2024

SPECIAL MECHANISMS GROUP ethereum foundation

ACCOUNTABLE SAFETY

- **Accountable Safety** is extremely desirable for blockchain designers and an active area of research:
  - Casper FFG (Buterin and Griffith 2017)
  - Accountable safety implies finality (Neu, Tas, Tse 2023)
  - The economic limits of permissionless consensus (Budish, Lewis-Pye, Roughgarden 2024)

Definition (Neu, Tas, and Tse 2023)

Accountable safety means that in any case of inconsistency, a certain fraction of validators can be identified to have provably violated the protocol.

STATIC VS DYNAMIC VALIDATOR SET

- The formal guarantees assume static validator sets.
- Real-world blockchains need to allow people to enter and **exit**.

Remark

When you allow a validator to exit, they may cause a safety violation before exiting the protocol – their stake cannot be held accountable.

OUR PAPER

- We study the trade-off between accountable safety and withdrawal speed.
- More formally, we derive a mechanism to minimize the disutility of validators waiting to exit without violating accountable safety.

MODEL

- We model the Accountable Safety requirements as a collection of validator set consistency constraints:

$$\mathcal{C} = \{(\delta_1, T_1), ..., (\delta_k, T_k)\}$$

- "No more than δ_1 of the stake can exit within T_1 days" and "no more than δ_2 of the stake can exit within T_2 hours."
- e.g., "No more than 10% of the stake can exit within 7 days" and "no more than 2% of the stake can exit within 12 hours."

MODEL

- Time is discrete and represents when withdrawals can be processed (e.g., epoch boundaries in Ethereum).
- Each agent has a cost of waiting, $c_i < 0$. Thus, their overall disutility is their cost times the amount of time they spend in the queue:

$$U_i = c_i \Delta_i.$$

MODEL

- The social planner chooses a mechanism $M(\mathcal{H})$ which takes as input a history $\mathcal{H}$ of arrivals and withdrawals and outputs the withdrawals in that period.
- We want to minimize the expected disutility of the validators under some arrival process for new withdrawal requests.

Social planner chooses

$$\arg\max_{M \in \mathcal{M}} = \sum_{i=1}^{n} U_i(M)$$

s.t. $\nexists R(\cdot)$ that causes a consistency constraint in $\mathcal{C}$ to be violated.

MAIN RESULT UNDER COMMON VALUES

Theorem

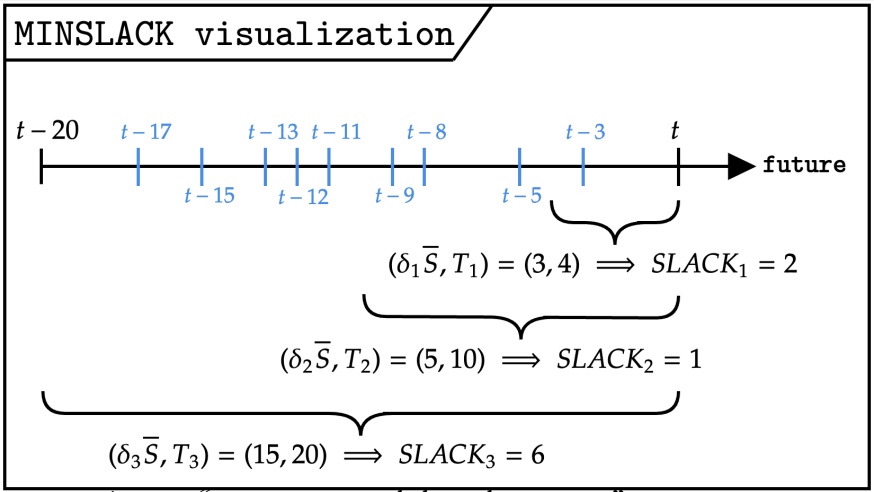
Given any sequence of withdrawal requests $R(\cdot)$, let $P(\cdot)$ be the processed withdrawal requests and $\overline{P}(\cdot)$ be the resulting total amount withdrawn in each period by `MINSLACK`*. Then:*

1. ***Feasibility:*** *$P(\cdot)$ is feasible with respect to the protocol constraints.*
2. ***Optimality:*** *For any other feasible withdrawal decisions with total withdrawn in each period given by $\overline{P}'(\cdot)$, it must be the case that:*

$$\forall t \geq 1 : \sum_{\tau=1}^{t} \overline{P}'(\tau) \leq \sum_{\tau=1}^{t} \overline{P}(\tau). \tag{1}$$

MINSLACK

- "At each period, greedily process as many withdrawals as possible given the constraints."
- Simple example: $\mathcal{C} \implies \{(3,4),(5,10),(15,20)\}$



- $\text{MINSLACK} = 1 \implies$ "process one withdrawal in time t."

MINSLACK

Input: Constraints $\mathcal{C} = \{(\delta_1, T_1), \ldots, (\delta_k, T_k)\}$.
Input: Initial staking $S(\cdot, 0)$.
$\overline{S}(0) \leftarrow \sum_v S(v, 0)$.
Initialize: $H(0), W(0), P(0) \leftarrow$ `NULL`.
Initialize: $\overline{P}(0) = 0$.
for each period $t \geq 1$ **do**
    $W(t) \leftarrow W(t-1) \setminus P(t-1) \cup R(t)$.
    for each constraint $i \leq k$ **do**
        $\texttt{SLACK}_i \leftarrow \delta_i \overline{S}(t - T_i) - \sum_{\tau=t-T_i+1}^{t-1} \overline{P}(\tau)$.
    end for
    $\texttt{MINSLACK} \leftarrow \min\{\texttt{SLACK}_i : 1 \leq i \leq k\}$.
    $P(t) \leftarrow$ Largest prefix of W(t) such that total withdrawn $\leq$ `MINSLACK`
    $\overline{P}(t) \leftarrow$ Total withdrawn in $P(t)$
    $H(t+1) \leftarrow H(t) \cup P(t)$
    Update: $S(v, t)$ based on $P(t)$.
end for

INTRODUCING HETEROGENEITY

- `MINSLACK` is optimal under common values...
- **Intuition**: People might have **extremely** different time preferences (e.g., you need to top up an AAVE borrow position).
- With heterogeneous values, `MINSLACK` is no longer optimal (with respect to the welfare loss given each validator "cost", c_i) because it doesn't reserve any capacity.

OPTIMAL UNDER HETEROGENEITY & PRIO-MINSLACK

- Given the (i) history, (ii) pending withdrawals, and (iii) distribution of future arrivals, we need to determine how much capacity to consume.
- $\uparrow$ is solvable with an MDP but super messy – what if we just did a priority queue instead of FCFS: PRIO-MINSLACK.

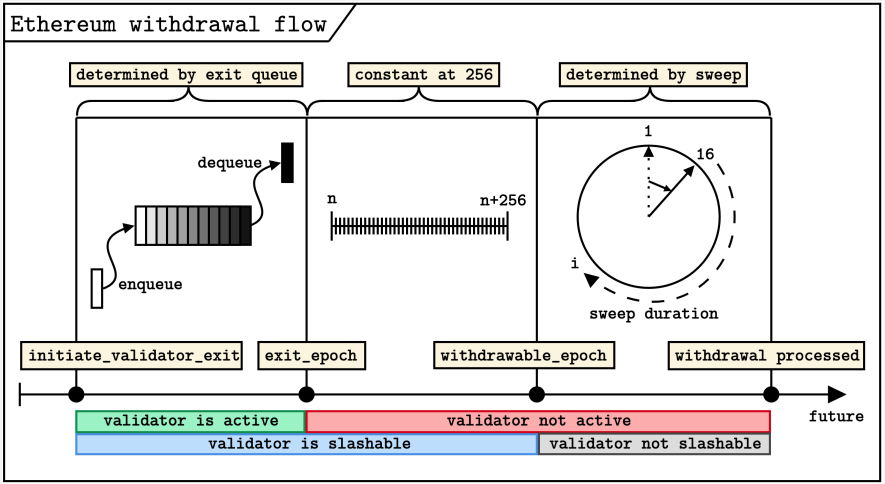
A FEW NUMERICAL RESULTS

Algorithm	Arrival dist.	Value dist.	Discount	Performance
OPTIMAL	$Y \sim [0,1,5]$ *w.p.* $[0.5,0.4,0.1]$	$X \sim [1,5]$ *w.p.* $[0.9,0.1]$	0.9	−2.428
PRIO-MINSLACK				−2.422
OPTIMAL		$X \sim [1,10]$ *w.p.* $[0.9,0.1]$		−2.959
PRIO-MINSLACK				−3.005
OPTIMAL		$X \sim [1,20]$ *w.p.* $[0.9,0.1]$		−3.902
PRIO-MINSLACK				−4.151
OPTIMAL	$Y \sim [0,1,2]$ *w.p.* $[0.4,0.4,0.2]$	$X \sim [1,10]$ *w.p.* $[0.9,0.1]$	0.9	−1.637
PRIO-MINSLACK				−1.638
OPTIMAL	$Y \sim [0,1,5]$ *w.p.* $[0.5,0.4,0.1]$			−2.925
PRIO-MINSLACK				−2.969
OPTIMAL	$Y \sim [0,1,10]$ *w.p.* $[0.6,0.35,0.05]$			−3.610
PRIO-MINSLACK				−3.620

- 6% vs. 0.06% improvements $\Longrightarrow$ very dependent on your arrival and value distributions!

Neuder, Pai, Resnick (2024): Optimizing Exit Queues for PoS

ETHEREUM CASE STUDY



SPECIAL MECHANISMS GROUP

ethereum foundation

Neuder, Pai, Resnick (2024): Optimizing Exit Queues for PoS

FURTHER EXAMPLES

Protocol	Staking purpose	Withdrawal mechanism	One-line analysis
Ethereum [4]	Consensus safety	Rate-limited FCFS queue with minimum duration.	Aims to be fast in the average case, but partial withdrawals induce high-variance delay.
Cosmos [19]	Consensus safety	Fixed 21-day unbonding period.	Simple but inefficient.
Solana [39]	Sybil resistance	All deactivations happen at epoch boundaries. A maximum of 25% of stake can deactivate at any given epoch boundary.	With no slashing, stake does not provide accountable safety to the protocol. Limiting withdrawals ensures the entire stake cannot exit in a single epoch.
EigenLayer [22]	Economic security guarantees	Fixed 7-day escrow period for all `ETH`-denominated withdrawals. Staked `EIGEN` has a fixed 24-day escrow period.	Withdrawals need to be limited because EigenLayer introduces new slashing conditions. Native restaked `ETH` may be withdrawn from the beacon chain during the EigenLayer escrow period.

Neuder, Pai, Resnick (2024): Optimizing Exit Queues for PoS

THANKS :-)

https://arxiv.org/pdf/2406.05124

SPECIAL MECHANISMS GROUP

ethereum foundation