Solana Transaction Fee Mechanisms

August 9, 2023

1 Model

The following is a model for designing TFMs on Solana

1.1 Agents

These are the different players and objects in our model

- Every transaction t is associated with a tuple (t, A_t, c_t, v_t, s_t) corresponding to its access list, CU resources used, and value for inclusion, and whether or not it is spam respectively
- Each user submits a tuple (t, A'_t, c'_t, b_t) to the mechanism corresponding to the transaction, account access list, requested CU amount, and bid
  - We only allow the user to submit an access list $A'_t \supseteq A_t$ and a requested CU amount $c'_t \geq c_t$. That is the user can only over-represent the accounts their transaction conflicts with or how many resources their transaction requires.
  - We can also consider simpler cases where $A'_t = A_t$ and $c'_t = c_t$
- $s_t = 0$ if transaction t is spam and $s_t = 1$ otherwise
  - s_t is not submitted by the user and is not observable by the mechanism
- Every block has a total size limit C of total CU that can be included
- Every account has a limit of C^* of total CU that can be used per block
  - If t is included, all of the accounts in A'_t have c'_t contributed to their limit
- There is a block producer that wants to maximize their revenue
- A block B consists of some set of submitted transactions with the welfare of a chosen block B being $W(B) = \sum_{t \in B} s_t v_t - s(1 - s_t)$
  - Currently leaving out negative penalty for leaving unused space in blocks. Will revisit this later to see what the best functional form for that is.
- The blockset $\mathcal{B}$ is the set of blocks that are valid under the size limits
- The optimal welfare $OPT = \max_{B \in \mathcal{B}} W(B)$

1.2 TFM Definition

These are the rules that characterize

- Payment Rule: Function p that takes as input all the tuple bids by users and specifies a payment for each user for each potential block in the block set
- Burning Rule: Function q that takes as input all the tuple bids by users and specifies a burn the block producer has to pay for every potential block they can producer
- Allocation Rule: Function x that takes as input all the tuple bids by users and specifies which block the block producer should chose

1.3 Agent Utility Functions

- Users: Have value $v_t - p_t(b_t, \cdot)$ if their transaction is included, otherwise they have utility 0
- Block Producer: Their value for picking a block is the net cashflow they get from picking that block as defined by the payment and burning rules, namely, $\sum_t p_t(\cdot) - q(\cdot)$

2 Goals

- DSIC: Every user has a dominant strategy to submit their bid with $A'_t = A_t, c'_t = c_t$, and $b_t = v_t$
- BPIC: Given the payment and burning rules, the block producer is incentivized to follow the allocation rule. The block producer doesn't have incentives to submit dummy transactions or censor transactions to profit.
- OCA: No incentive for collusion between users and the block producer. Will leave this out for now because it's hard enough to achieve the other properties first. Will revisit later.
- Welfare Maximizing: The welfare of the block chosen by the allocation rule is close to the optimal possible welfare
- Griefing Resistant: This is a subset of DSIC that users don't have an incentive to over-represent their access lists
- Non-Contested Inclusion: Blocks shouldn't have empty capacity if there are valid transactions that could fill them. Can make this part of the analysis by giving a negative penalty to the welfare of blocks that aren't full. Could also give some boost to including payment transactions in the welfare by some factor.
- Vote Inclusion: Leaving this out of the model for now, but pretty easy to include. Similar to how every transaction has a spam attribute, could give every transaction a vote attribute for whether or not it is a vote. Could make this parameter public to the mechanism and have some negative welfare to not including enough votes in a block.

By defining DSIC + Welfare properly, we should be able to make the only goals DSIC, BPIC, and Welfare Maximization to capture the other informal properties we want as well.