Optimizing Blockchain Monetary Velocity

David Naccache[1]

DIÉNS, ÉNS, CNRS, PSL University, Paris, France
45 rue d'Ulm, 75230, Paris CEDEX 05, France
{given_name.family_name}@ens.fr

1 Introduction

In economy *the velocity of money* measures the average number of times that a currency unit is used to purchase goods and services within a given time period. The concept relates the global economic activity to the economy's money supply, called the *monetary base*.

A commonly heard criticism of blockchains is their *latency* (also called *block time*). Latency is the time required to generate the next block of transactions in the chain. i.e. it reflects the delay that a user must wait after sending a transaction to see his transaction appear in the blockchain. By opposition to fiat, the velocity of cryptocurrencies as well as other economical metrics[1] are not determined only by pure economical factors but also by artificial technological limitations due to mining.

As in the case of transportation, system designers can resort to four throughput optimization strategies:

increase the number of trains	$\cong$ more miners
increase the speed of trains	$\cong$ more powerful hardware
reduce the number of passengers	$\cong$ process less transactions
schedule commuting more intelligently	$\cong$ this paper

This paper investigates the fourth option, i.e. better schedule transactions to optimize monetary velocity.

2 The Intuition

Consider a blockchain processing k transactions (a "block") per time unit and a set of $n = \tau \times k$ transactions waiting to be scheduled. It is granted that after τ time units all transactions will be processed. Note that by "processed" we indifferently mean either approved or rejected. Hence the final state of the system (balances of all users at time τ) depends on the feasibility of transactions that stems from their partition into τ blocks $B_0, \ldots, B_{\tau-1}$. In particular, because transactions potentially condition each other their scheduling is very important and different schedulings may result indifferent final system states.

[1] e.g. the blockchain's monetary supply or monetary base.

A transaction μ_i, for $0 \leq i \leq n-1$ is a triple $\mu_i = \{a_i, s_i, r_i\} \in \mathbb{N} \times U^2$ where a_i, s_i, r_i respectively denote the amount, the sender and the receiver who belong to a set of users U. We assume that at time $t = 0$ each user j has an initial balance $b_{0,j} \in \mathbb{N}$. After a transaction μ_i is executed the balances b_{t,s_i} and b_{t,r_i} are respectively updated to $b_{t+1,s_i} = b_{t,s_i} - a_i$ and $b_{t+1,r_i} = b_{t,r_i} + a_i$ provided that $b_{t,s_i} \geq a_i$, i.e. that the sender s_i had enough funds to honor the transaction.

A *velocity optimization* seeks to optimize σ, the sum of successful transactions corresponding to the scheduling $B_0, \ldots, B_{\tau-1}$ when the timer reaches $\tau - 1$.

$$\sigma = \frac{1}{2} \sum_{t=0}^{\tau-1} \sum_{i \in U} |b_{t+1,i} - b_{t,i}|$$

3 The Proposed Challenge

Develop algorithmic strategies to optimize σ.