11-2018

Towards Optimal Implementation of Decentralized Currencies: How to Best Select Probabilities in an Ethereum-Type Proof-of-Stake Protocol

Thach N. Nguyen
Banking University of Ho Chi Minh City, ajeb@buh.edu.vn

Christian Servin
El Paso Community College, cservin@gmail.com

Vladik Kreinovich
The University of Texas at El Paso, vladik@utep.edu

Comments:
Technical Report: UTEP-CS-18-79

Towards Optimal Implementation of Decentralized Currencies: How to Best Select Probabilities in an Ethereum-Type Proof-of-Stake Protocol

Thach Ngoc Nguyen[1], Christian Servin[2], and Vladik Kreinovich[3]
[1]Banking University of Ho Chi Minh City
56 Hoang Dieu 2, Quan Thu Duc, Thu Duc
Ho Chi Minh City, Vietnam, Thachnn@buh.edu.vn
[2]Computer Science and Information Technology
Systems Department
El Paso Community College, 919 Hunter
El Paso, TX 79915, USA, cservin@gmail.com
[3]Department of Computer Science
University of Texas at El Paso
El Paso, Texas 79968, USA, vladik@utep.edu

Abstract

Nowadays, most financial transactions are based on a centralized system, when all the transaction records are stored in a central location. This centralization makes the financial system vulnerable to cyber-attacks. A natural way to make the financial system more robust and less vulnerable is to switch to decentralized currencies. Such a transition will also make financial system more transparent. Historically first currency of this type – bitcoin – use a large amount of electric energy to mine new coins and is, thus, not scalable to the level of financial system as a whole. A more realistic and less energy-consuming scheme is provided by proof-of-stake currencies, where the right to mint a new coin is assigned to a randomly selected user, with probability depending of the user's stake (e.g., his/her number of coins). What probabilities should we choose? In this paper, we find the probability selection that provides the optimal result – optimal in the sense that it is the least inductive to cheating.

1 Formulation of the Problem

Need for decentralized currencies. The existing currencies are based on the centralized record keeping and centralized control. This centralization make the

system vulnerable to attacks: indeed, an attack on the central depositary is sufficient. To make the financial system more robust, a natural idea is to decentralize the record keeping, to keep all the records of all the financial transactions in as many locations as possible – this will also help make financial transactions more transparent. With distributed currency, transactions are also easier and often faster – since there is no need to contact the central bank or other centralized authority.

This decentralization is the main idea behind cryptocurrencies.

Bitcoin – the world's first decentralized currency: idea and its limitations. Historically the first decentralized currency was the Bitcoin. One of the main question that needs to be resolved when launching a new decentralized currency is how new coins are generated ("mined"). Bitcoin uses what is called a *proof-of-work* protocol: to generate a new coin, it is necessary to solve a complex time-consuming computational problem. This problem is made very complex to purpose, to limit the number of bitcoins and thus, to avoid inflation.

The need of proof-of-work, however, leads to the main drawback of Bitcoin in particular and proof-of-work scheme in general: mining of bitcoins requires a lot of computing power and thus, a lot of electric energy. Already now, with bitcoins constituting a very small part of the financial system, their mining takes on (and thus wastes) a dis-proportionally huge amount of electric energy. As a result, the proof-of-work protocol is not very scalable – it is not realistic to expect a significantly larger amount of financial transactions to distributed currency.

Proof-of-stake protocol: a solution to bitcoin limitations. To make distributed currencies more energy efficient, it is reasonable to base the right to mine new coins not on a new problem-to-solve, but on the current states of different users. For example, we can base this right on the amount of currency (stake) that each user has. To be more precise, the probability p_i of allowing user i to mine the next coin depends on i's stake s_i. Namely, the probability p_i is proportional to $f(s_i)$, for some function $f(z)$: $p_i = c \cdot f(s_i)$, where $c > 0$ is some constant.

These probabilities should add up to 1, so we must have

$$1 = \sum p_j = c \cdot \sum_j f(s_j),$$

hence $c = \dfrac{1}{\sum\limits_j f(s_j)}$ and

$$p_i = \frac{f(s_i)}{\sum\limits_j f(s_j)}. \tag{1}$$

We want to make sure that the use assigned to mint a new coin will not disrupt the whole system. Clearly, users with a larger stake in the system are more interested in preserving it, so it makes sense to have the probability increasing when the stake increases, i.e., to use an increasing function $f(z)$.

This scheme is known as a *proof-of-stake* scheme. It is used, e.g., in Ethereum – another popular cryptocurrency; see, e.g., [2, 3, 4, 5].

Remaining problem: how to select probabilities? The remaining question is: what is the best way to select probabilities? In other words, what is the best choice of the function $f(z)$?

Ethereum uses the simplest possible increasing function $f(z) = z$. In this case:

$$p_i = \frac{s_i}{\sum\limits_j s_j}. \tag{2}$$

But is this selection optimal? Maybe there are more efficient functions?

What we do in this paper. In this paper, we show that the Ethereum's choice $f(z) = z$ is indeed optimal – in the sense that this scheme is the most robust against cheating.

2 Analysis of the Problem and the Main Result

What we want. One of the main ideas behind distributed currencies is to provide transparency to all financial transactions, to minimize possible cheating.

We therefore want to select probabilities in such a way so as to minimize the incentives for cheating.

Main idea behind our analysis. The main idea behind our analysis of the situation is that every user wants to maximize his/her participation in coin mining. This is a natural desire – we never fully trust others, so if we do something ourselves, we have the largest possible confidence that the financial system is not damaged by inadequate coin minting actions.

What we do in this section. In this section, we show that the above idea leads to $f(z) = c \cdot z$ for some $c > 0$ – i.e., to Ethereum's probabilities (2).

The above idea leads to $f(a) + f(b) \le f(a + b)$. Let us first show that the above idea leads to the super-additivity inequality

$$f(a) + f(b) \le f(a + b) \tag{3}$$

for all $a > 0$ and $b > 0$.

We will prove this by reduction to a contradiction. Indeed, suppose that for some a and b, we have $f(a) + f(b) > f(a + b)$. In this case, a user who has $m \stackrel{\text{def}}{=} a + b$ coins, can increase his probability of minting a new coin if he fictitiously splits him/herself into two "users", with stakes a and b, accordingly.

Before the split, the probability of this user minting a coin was proportional to $f(m) = f(a + b)$. After the split, the probability is proportional to the sum $f(a) + f(b)$ and is, thus, higher.

So, the need to avoid incentives for cheating indeed leads to the inequality (3).

The above idea leads to $f(a+b) \le f(a) + f(b)$. Let us now show that the above idea also leads to the sub-additivity inequality

$$f(a+b) \le f(a) + f(b) \tag{4}$$

for all $a > 0$ and $b > 0$.

Indeed, suppose that for some a and b, we have $f(a+b) > f(a) + f(b)$. In this case, two users with stakes a and b can agree to pretend that they are actually one user. Before this pretense, the probability that one of them will be selected to mint the next coin is proportional to $f(a) + f(b)$. Once they pretend to be a single user, this probability increases to $f(a+b) > f(a) + f(b)$. The user can then decide between themselves who actually gets to mint the coin – e.g., by selecting the first user with probability

$$\frac{f(a)}{f(a) + f(b)}$$

and selecting the second user with the remaining probability

$$\frac{f(b)}{f(a) + f(b)}.$$

This way, for each user, the probability of being selecting increases.

Indeed, for the first user, the new probability of being selected is proportional to

$$f(a+b) \cdot \frac{f(a)}{f(a) + f(b)}.$$

Since $f(a+b) > f(a) + f(b)$, we have

$$f(a+b) \cdot \frac{f(a)}{f(a) + f(b)} > (f(a) + f(b)) \cdot \frac{f(a)}{f(a) + f(b)} = f(a),$$

i.e., the probability indeed increases.

Similarly, for the second user, the new probability of being selected is proportional to

$$f(a+b) \cdot \frac{f(b)}{f(a) + f(b)}.$$

Since $f(a+b) > f(a) + f(b)$, we have

$$f(a+b) \cdot \frac{f(b)}{f(a) + f(b)} > (f(a) + f(b)) \cdot \frac{f(b)}{f(a) + f(b)} = f(b),$$

i.e., the probability indeed increases.

Thus, the only way to eliminate incentives for cheating is to select a function $f(z)$ that always satisfies the inequality (4).

Conclusion: the Ethereum-style selection $f(z) = c \cdot z$ is optimal. In the previous two subsections, we showed that the only way to avoid incentives for

cheating is to have inequalities (3) and (4) always satisfied. Combining these two inequalities, we conclude that

$$f(a+b) = f(a) + f(b) \tag{5}$$

for all $a > 0$ and $b > 0$.

We assumed that the function $f(z)$ is increasing. It is known (see, e.g., [1]) that the only increasing functions that satisfy the additivity property (5) are functions $f(z) = c \cdot z$, for some $c > 0$. Thus, the Ethereum-style selection $f(z) = c \cdot z$ is indeed the only one optimal in our sense – i.e., the only one that minimizes the incentives for cheating.

One can check that with this selection of probabilities, there is indeed no incentive for cheating: a user can fictitiously split into two or more pieces, two or more user can fictitiously claim that they are a single user – none of this will change the probability of each user being selected to mint a new coin.

Acknowledgments

This work was supported in part by the US National Science Foundation via grant HRD-1242122 (Cyber-ShARE Center of Excellence).

The authors are greatly thankful to Omar Badreddin for valuable suggestions.

References

[1] J. Aczél and J. Dhombres, *Functional Equations in Several Variables*, Cambridge University Press, 2008.

[2] A. M. Antonopoulos and G. Wood, *Mastering Ethereum: Building Smart Contracts and DApps*, O'Reilly Media, Sebastopol, California, 2018.

[3] C. Dannen, *Introducing Ethereum and Solidity: Foundations of Cryptocurrency and Blockchain Programming for Beginners*, Apress, New York, 2017.

[4] H. Diedrich, *Ethereum: Blockchains, Digital Assets, Smart Contracts, Decentralized Autonomous Organizations*, CreateSpace Independent Publishing Platform, 2016.

[5] I. Takashima, *Ethereum: The Ultimate Guide to the World of Ethereum, Ethereum Mining, Ethereum Investing, Smart Contracts, Dapps and DAOs, Ether, Blockchain Technology*, CreateSpace Independent Publishing Platform, 2017.