PoS forging algorithms: multi-strategy forging and related security issues

andruiman, andruiman@gmail.com*

February 1, 2015

Abstract

Different properties of the multibranch forging algorithm in the Proof-of-Stake cryptocurrencies are investigated. A comparison with a single-branch forging strategy is presented (treating a multibranch ability as an attack) along with the inner multibranch attacks like Nothing-At-Stake problem. The latter is formally defined and an algorithm to model it is given. Most of the results of the study reveal advantages of the multibranch approach over the classical one.

Keywords: PoS crypto-currencies, forging, blocktree, strategy, simulation

1 Pre-introduction

We continue the series of papers on various aspects of Proof-of-Stake forging algorithms. We act as the Consensus Research Group `http://consensusresearch.org` and publish our updates both on the website and the Nxt forum `https://nxtforum.org/consensus-research/`. We plan to concentrate on one result in each paper and update our status with at least one new paper or software upload every two weeks describing what we will have achieved. For the papers that need corrections or important changes we will update in the usual way, issuing the next versions.

*To support this work please use our asset in AE: 5841059555983208287

2 Introduction

Recently we had published the paper `https://www.scribd.com/doc/248208963/Multibranch-forging` concerning the multibranch possibilities in the Proof-of-Stake crypto-currencies networks and showed its statistical properties which can be retrieved using our simulation tool `https://github.com/ConsensusResearch/MultiBranch` written in Haskell with some Coq proofs about.

In this paper we continue presenting the simulation results and concentrate on the aspects of

1. Comparison of the different types of multibranch/single-branch forging strategies running by different accounts.
2. Hidden multibranch attacks possibilities.
3. Possible (dis)advantages of distributing the stake between several multibranch accounts in the competition with the single-branch account.
3. Nothing-at-Stake problem formulation and approach to its simulation.

All the necessary definitions and the simulation model one can also find in the previous paper mentioned above. Here we present mostly the results of simulations and comments to them.

3 Measures for a chain

In the previous paper we used the simplest type of the measure function which is calculated along a chain to choose between them in the blocktree. It was set to the constant function $m_0(b) \equiv 1$. This function allows to choose the longest chain when maximizing the cumulative value of a chain $CD(C) = \sum\limits_{b \in C} m_0(b) =$ length C. However the disadvantage of the constant function is the arising positive effect of `baseTarget` growth. So bigger `baseTarget` leads in average to shorter time to the next block and the hidden attacker (or isolated subnet) can potentially create the blocks with the same length but with less cumulative stake due to the retargeting procedure. In any isolated network the retargeting procedure tries to stabilize block generation to the same rate. After reconnecting to the main network the isolated generated blocks will be literally the same measured as normally generated comparing them by the chain length. So the good idea is to measure the block with some inversely proportional functions like $m_1(b) = 1/(\texttt{baseTarget b})$. In this paper we do not investigate the exact type of the measure function so we choose it equal to $m_1(\cdot)$. In the real Nxt network it is chosen the same way. In this case the isolated network will tend to have the retargeted `baseTarget` as bigger as their cumulative stake is lower than the main subnet stake to have the similar block rate and as the result the isolated chain loses.

However it is still open question about the influence of the measure function to consensus properties of the network and its attack invulnerability. We will dedicate the next paper to investigating the properties of different measure functions in terms of the network convergence.

4 Open individual multibranch vs collective single-branch

The model: we investigate a competition between the individual multibranch account and the single-branch network. We suggest that the single-branch (classical) account can forge only to the last block of the best chain in the tree, but the multibranch account forges to all possible blocks and all generated blocks are immediately shared between all the nodes. So we take the following parameters: multibranch stake $s_1 = 10\%, 20\%, 30\%, 40\%, 50\%$, the numbers of uniform single-branch accounts are set to $N - 1 = 1, 2, 4, 8$, $\tau = 10$, steps to simulate 1500. The depth of the multibranch forging is set ∞. The parameter we are interesting in is the portion of blocks generated by both sides, so it shows which strategy is more effective.

Table 1: Portion of blocks for the multibranch account, $T = 1500$, $\texttt{tfdepth} = \infty$

1st stake \ $N-1$	1	2	4	8
10%	$7/106 \approx 0.066$	$8/108 \approx 0.074$	$7/75 \approx 0.093$	$17/97 \approx 0.18$
20%	$24/105 \approx 0.23$	$23/106 \approx 0.22$	$32/76 \approx 0.42$	$46/83 \approx 0.42$
30%	$31/107 \approx 0.29$	$50/99 \approx 0.51$	$47/92 \approx 0.51$	$51/92 \approx 0.55$
40%	$43/93 \approx 0.46$	$57/98 \approx 0.58$	$66/92 \approx 0.72$	$54/92 \approx 0.59$
50%	$71/83 \approx 0.86$	$83/107 \approx 0.78$	$83/93 \approx 0.89$	$77/96 \approx 0.80$

For the (8s+1m) system we also simulate the different number of steps to observe a stable proportion.

Table 2: Portion of blocks for the multibranch account, $N = 8s + 1m$

1st stake \ T	1000	1500	2000
10%	$10/67 \approx 0.15$	$17/97 \approx 0.18$	$28/126 \approx 0.22$
30%	$31/63 \approx 0.49$	$51/92 \approx 0.55$	$62/116 \approx 0.53$
50%	$60/72 \approx 0.83$	$77/96 \approx 0.80$	$107/131 \approx 0.82$

To be more inspired from properties of the multibranch ability we investigate the same "battle" but with finite `tfdepth` parameter.

Table 3: Portion of blocks for the multibranch account, $T = 2000$, `tfdepth` $=$ 20

1st stake \ $N-1$	1	2	4	8
10%	$9/151 \approx 0.06$	$10/144 \approx 0.07$	$7/95 \approx 0.074$	$28/126 \approx 0.22$
20%	$23/146 \approx 0.16$	$25/142 \approx 0.18$	$49/105 \approx 0.47$	$55/115 \approx 0.48$
30%	$29/119 \approx 0.24$	$54/127 \approx 0.43$	$59/127 \approx 0.46$	$63/128 \approx 0.49$
40%	$56/91 \approx 0.62$	$81/131 \approx 0.62$	$93/128 \approx 0.73$	$77/122 \approx 0.63$
50%	$107/137 \approx 0.78$	$101/140 \approx 0.72$	$112/129 \approx 0.87$	$107/131 \approx 0.82$

The results show that the multibranch strategy is more preferable than the single-branch and gives the possibility to find the best chain with more blocks generated by the multibranch account. However the depth of the multibranch forging is not very important and could be set to some not big number as 20 which leads to economy of computational power. Actually for long run networks the power needed to compute the full multibranch forging is practically goes to infinity, so it doesn't lead to any significant benefits in terms of the generated blocks (and fee rewarded). So we conclude that the multibranch forging matters but its depth can be set to a reasonably small parameter to save the power having almost the same efficiency. We do not expect therefore the *arms race* as we observe in the PoW networks.

5 Open collective multibranch vs individual single-branch

In this section we investigate the opposite case, when several multibranch accounts "struggle" against the individual single-branch one. So we asked ourselves the question about (dis)advantage of dividing the stake between several accounts in a hope to get more blocks generated. So the model is: $N-1 = 1, 2, 4, 8$, `tfdepth` is set to 20, $T = 2000, 3000$, single-branch account' stake is set to a value from 10-50% interval.

Table 4: Portion of blocks for the single-branch account, $T = 2000$

1st stake \ $N-1$	1	2	4	8
10%	$4/141 \approx 0.028$	$3/130 \approx 0.023$	$4/134 \approx 0.03$	$1/131 \approx 0.0076$
20%	$6/141 \approx 0.043$	$5/148 \approx 0.034$	$8/123 \approx 0.07$	$9/148 \approx 0.06$
30%	$8/129 \approx 0.062$	$18/129 \approx 0.14$	$8/134 \approx 0.06$	$20/143 \approx 0.14$
40%	$32/136 \approx 0.24$	$29/138 \approx 0.21$	$24/132 \approx 0.18$	$36/135 \approx 0.27$
50%	$22/116 \approx 0.19$	$22/126 \approx 0.17$	$26/121 \approx 0.21$	$48/139 \approx 0.35$

Table 5: Portion of blocks for the single-branch account, $T = 3000$

1st stake \ $N-1$	1	2	4	8
10%	$13/215 \approx 0.06$	$4/195 \approx 0.02$	$4/205 \approx 0.02$	$2/196 \approx 0.01$
20%	$12/224 \approx 0.053$	$6/212 \approx 0.03$	$9/195 \approx 0.046$	$12/163 \approx 0.074$
30%	$14/207 \approx 0.07$	$22/192 \approx 0.11$	$12/209 \approx 0.057$	$21/149 \approx 0.14$
40%	$36/196 \approx 0.18$	$42/197 \approx 0.21$	$32/211 \approx 0.15$	$43/169 \approx 0.25$
50%	$40/181 \approx 0.22$	$40/188 \approx 0.21$	$41/187 \approx 0.22$	$68/179 \approx 0.38$

The results show again the advantage of the multibranch forging at all. However the effect of the dividing the stake is negligible or even negative as one can see from the $N-1 = 8$ column. Some possible advantage can arise when the dividable stake is very big (e.g. for 10% row) but this cannot be observable IRL. So the less multibranch stake we have the more we lose when dividing it. So the sybil attack is quite unlikely in the multibranch environment.

6 Hidden multibranch attack

Next we consider the following experiment: given a certain number of single-branch accounts we wonder whether it could be possible to generate a better chain not sharing the blocks with multibranch ability. That is - suppose some multibranch account hides its generated blocks from the network, however it can receive blocks from the outside. The question is: can it produce a better chain to rewrite the history unpredictably and what stake it should have to perform such attack.

It is often referred as the "long range attack" `https://github.com/ethereum/wiki/wiki/Problems`: *Another problem to keep in mind is the issue of so-called "long-range attacks" - attacks where the miner attempts to start a fork not five or ten blocks behind the head of the main chain, as happens normally, but hundreds of thousands of blocks back. If an algorithm is designed incorrectly, it may be possible for an attacker to start from that far back, and then mine billions of blocks into the future (since no proof of work is required), and new users would not be able to tell that the blockchain with billions of blocks more is illegitimate. This can generally be solved with timestamping, but special corner cases do tend to appear in overcomplicated designs.*

So the experiment is quite similar to the given in the section 4, but the multibranch account is set to be hidden until some moment (at the end of simulation) when it reveals its best blockchain. The `tfdepth` parameter for this account is set to ∞.

Table 6: Portion of blocks for the hidden multibranch account, $N = 1m + 8s$

attacker stake \ T	500	1000	1500	2000
10%	3/20	0	2/80	0
20%	3/33	0	0	9/112
25%	2/31	0	23/91	3/110
30%	28/30	0	6/97	6/125

Interesting thing we observe here is that actually this attack can be renamed to the "short range attack" that is the attack opportunity arises occasionally but can disappear later. So the attacker strategy can be the following: hide the generated blocks until some moment when it can be considered as local maximum and then reveal the best chain and do this repeatedly. So local attack arises locally and then iterated. One can see that even with 30% stake the attacker got a chance to make 28/30 portion of blocks for the 500 steps run, but later within 1000 steps interval the attack opportunity disappears. So after the local attack the best blockchain ends with a sequence of blocks generated by the attacker and this can be easily detected.

As shown in the sections above the *long* range attack is quite unlikely due to retargeting procedure and chain measuring. However the possibility of the *short* range attack from the multibranch account still exists. However it is obvious that for the whole multibranch environment it is impossible. Another important thing is that for the most cases the long run of the hidden forging is much worse than the open behaviour as we can see for the row 10% in comparison with the same row in the table 1.

7 Nothing-at-Stake formulation and simulation

The Nothing-at-stake problem is well known and formulated for example here `https://github.com/ethereum/wiki/wiki/Problems`: *However, this algorithm has one important flaw: there is "nothing at stake". In the event of a fork, whether the fork is accidental or a malicious attempt to rewrite history and reverse a transaction, the optimal strategy for any miner is to mine on every chain, so that the miner gets their reward no matter which fork wins. Thus, assuming a large number of economically interested miners, an attacker may be able to send a transaction in exchange for some digital good (usually another cryptocurrency), receive the good, then start a fork of the blockchain from one block behind the transaction and send the money to themselves instead, and even with 1% of the total stake the attacker's fork would win because everyone else is mining on both.*

So as we have developed the multibranch forging environment we could model it and understand its realization. To recall the problem and give some

IRL analogue let's consider the following story.

The story. There are multi-storey houses and builders. Usual builder just builds some house. In real life at the same moment a builder cannot build more than one house with the same resource. But our builders could be called quantum. A quantum builder can build every house he observes at the same moment and with the same resources. Builders get their salary just for building but only if the house is completely built and accepted by the committee. In the transparent environment it is obvious that to get more salary builders should build every house they can observe. So they choose the *equal* concurrent strategy. Their salary will depends only on their effort. Well, let's add to the community some special builder which we can call the insider. The insider besides his builder ability is also an apartment owner in some of the built houses. He knows these houses but anybody else - do not. So the fact that some builder has an additional interest cannot influence on the strategy of every builder of concern. They still choose to build every house they observe. But our insider can choose another strategy - the cost of his apartment lost is much more valuable for him than some insignificant lost in salary. So he chooses to build only those houses where he has an apartment. Everybody can detect this but that is no significant arguments for them to suspect something serious. Maybe he just doesn't want to work much. They get their salary anyway and even more - if he has a much power they prefer to help him as their possibility to get the salary sooner increases. So that is the example of attack on the labor democracy: the insider can manipulate the result as long as other parties remain indifferent. *End.*

This story reflects what we potentially could observe in the PoS cryptocurrencies where the cost of the forging resources is insignificant and multi-branch forging is allowed (it is very unlikely that it should be prohibited). As opposite in the PoW network the builders are not quantum as they cannot spend their resources at the same moment to different houses. Strictly speaking in the PoS they also cannot but the resource demanded is very small in comparison with the block generation time. So the Bitcoin-like networks we can call classical and Nxt-like - quantum. One could refer the current implementation of the Nxt as a semi-classical approximation.

We consider the Nothing-at-Stake attack in the following definition: what are the conditions when an account with arbitrary small stake can manipulate the resulted best chain by the selective forging. We slightly modify the initial formulation given at the beginning of the section to propose the exact (strict) Nothing-at-Stake model we take into consideration:

1. The attacker with a certain balance B immediately sends two transactions of B_1, B_2 coins to different connected peers (subnets) such as $B_1 + B_2 > B$. The reason is to send contradicting transactions which cannot be included to one blockchain by the honest node. That is no need to wait the first transaction to be confirmed and no need to start fork by himself. Actually it is quite unlikely that an attacker with small stake could start the correct fork at any given moment.

2. The transactions are propagated by peers and included in different branches as they cannot be included together. One of the transaction at some moment is included to the confirmed blockchain. We consider this case as the argument for a receiver to do something IRL (like fiat exchange action) or send crypto-currency coins stored within another blockchain. The attacker could also help the first transaction to be confirmed by his forging power directed to the correspondent chains. However he could remain indifferent as at some moment one of the transactions will have become confirmed.

3. Immediately after this the attacker directs his forging power to the branches where the second transaction is included and stops any other forgings.

4. If at some moment the second transaction is included to the best chain and the network confirms this, we suggest that the attacker wins because he has double spent.

It is obvious that if all the accounts are indifferent and don't try to detect the attack and resist it, the attacker always wins at some moment, with arbitrary small forging balance. However such the attack can be detected because the nodes accepted the first transaction will at some moment receive the second one and calculating the attacker balance get a negative number. So it is at least the reason not to include the second transaction to the blockchains, but this information can be used also to detect a potentially malicious action and try to prevent it somehow.

As we didn't yet included the transaction processing to the multibranch forging model so we need to simulate it with some tricks. However the trick is very simple: the stake-half accounts can mark the generated block as "black", other accounts - as "white". The option to mark a block arises after a certain number of steps and only once for both accounts subset and for every branch. The attacker's account can mark the generated block with any color. Then all descendants of the "black" blocks become "black" and the same inheritance for the "white" blocks. So at some moment all the new generated blocks become colored. The attacker firstly supports the "white" chains until there will be a certain number of blocks after some "white" block in the best chain (confirmed). As all the child blocks have the same color all the tail sequence become "white" colored. After this moment the attacker immediately changes the tactics and begins to support the "black" chains. If at some moment the "black"-blocked tail become longer than the confirmation length in the best chain we think that the attacker wins.

As we have all the algorithms developed to simulate N@S attack we present result in the separate paper along with possible ways to resist it. Giving some results now we present not the full picture of the problem. Following this section it is reasonable to get the impression that this problem actually matters and we concentrate to possible solutions at the moment.

8 Conclusion and future work

In this paper we investigate different properties of the multibranch forging algorithm in the PoS networks. Some attacks are also taken into considerations. We showed that in comparison with single-branch strategy the multibranch one is more efficient in terms of the number of generated block (and therefore the fee rewarded). It is also resistant to the long range attack, which we rename to the *short* range as it only occur in the simulations over short ranges and not over long ranges. The formal approach to the Nothing-At-Stake problem is given along with the simulation algorithm and the exact attacker' strategy.

The open question for the future work are: (1) the PoS consensus dependence on the measure function (2) the ways to avoid N@S attack if any (3) the optimal confirmation length investigation (4) the optimal multibranch depth investigation.