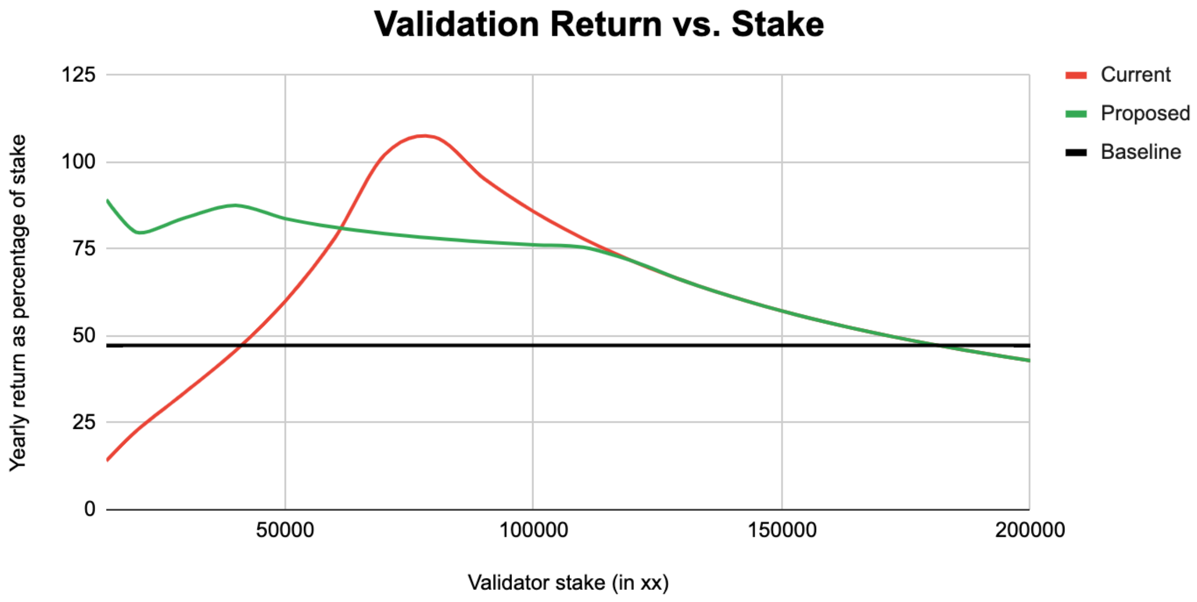
Team Multiplier Economic Analysis

Bernardo Cardoso

This document analyzes the impact of team multiplier in the current economic environment of the xx network. Furthermore, it explores the economic impact of Ben's proposal that would see the team multiplier take its share of rewards. The end result of the analysis backs the implementation of the proposal, with further optimizations also being explored.

Introduction

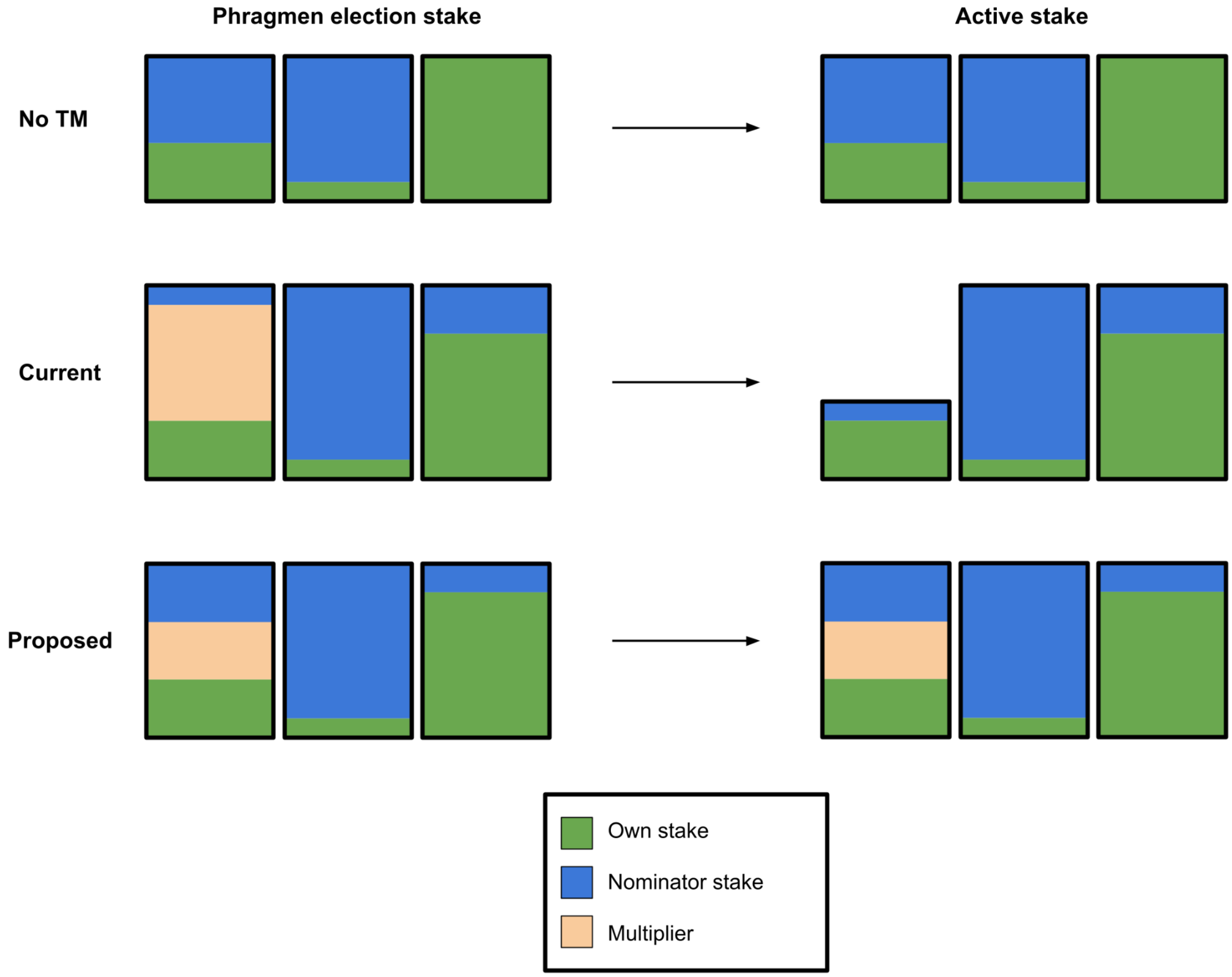
Back on February 8th, the team released an analysis and proposal for economic adjustments to the team multiplier. This proposal was focused on solving inequalities between nodes within the MainNet Transition Program (referred to as TM nodes for the remainder of this document). The following graph is taken from that proposal, and it displays the original disparity of rewards and the revised curve.



However, the team never compared the validation return between TM and non TM nodes. This document starts by doing exactly that, using information from the current state of the network.

Problem Statement

The following diagram shows a visual representation of the interaction of the team multiplier with the Phragmen election algorithm. It compares three situations: the regular operation of the algorithm when no team multiplier exists; the current implementation with TM; the implementation of Ben's proposal, where TM is counted in active stake.

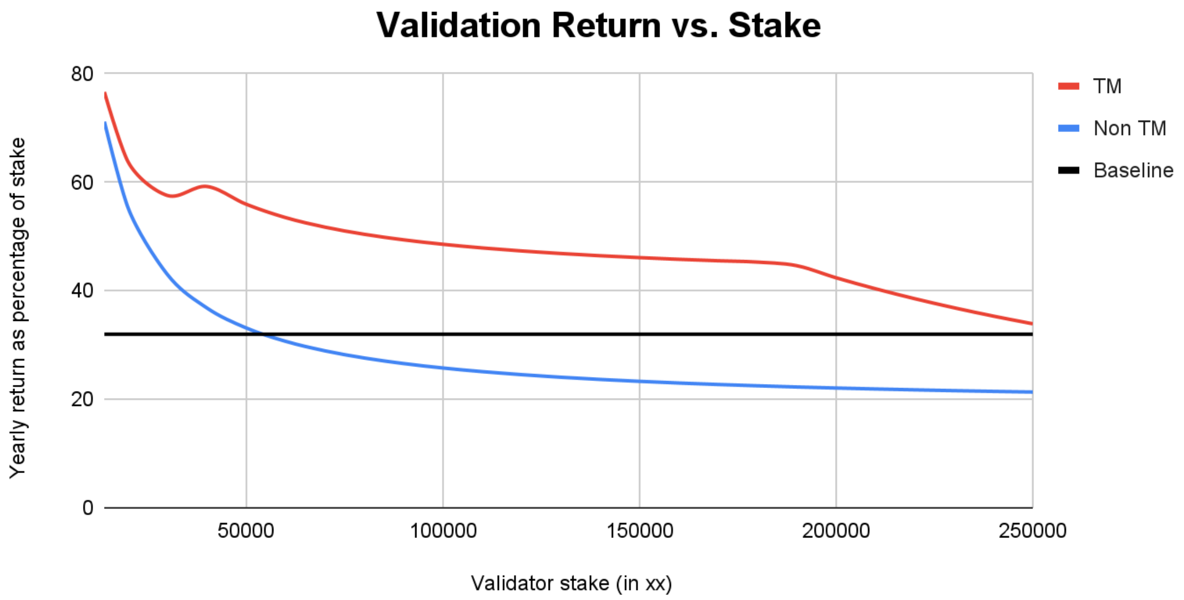


Phragmen always tries to optimize the deviation of election stake (first column), i.e., it tries to have all nodes with total stake as close as possible. As can be seen, the issue with the current implementation is that the TM stake is not counted in the active stake. This leads to TM nodes always having lower active stake than what Phragmen tried to optimize, and thus much higher returns to nominator stake.

As done before back in February, to better understand the issue, let's use the same Phragmen model for the average node. For simplicity of analysis, assume that the daily average validator reward is 262 xx (as seen currently in MainNet). Also, abstract away the issue of commission, by assuming all nodes have set it to 18%. Finally, fix a baseline for the average staking returns of nominating, which is 32%.

Let's look at the log for the multiplier script run of July 5th and get current network metrics: stakeable supply is ~149M, amount staked in the network is ~108.9M, total value of multipliers is ~63.5M, total TM nodes own stake is ~31.7M, with 291 TM nodes active. This leads to a multiplier cap of ~233k. Also, there are 8 bootnodes run by the xx foundation which have ~20M staked on them. The average election stake can be calculated by adding the total stake and total multipliers, removing the foundation stake, and dividing by the size of the validator set without bootnodes (370-8=362), which leads to ~421k.

The following graph compares the validation return between TM, non TM nodes and the baseline.



The graph clearly shows the disparity between nodes that was previously missed by the team. Furthermore, it also shows that validating without TM is less profitable than simply nominating, for stakes higher than 50k. Both of these are big issues: the TM should not interfere with the fairness of an open staking market, and validating should always be more profitable than nominating.

Multiplier Economic Analysis

Let's now analyze the solution proposed by Ben, which consists of the following 3 components:

"

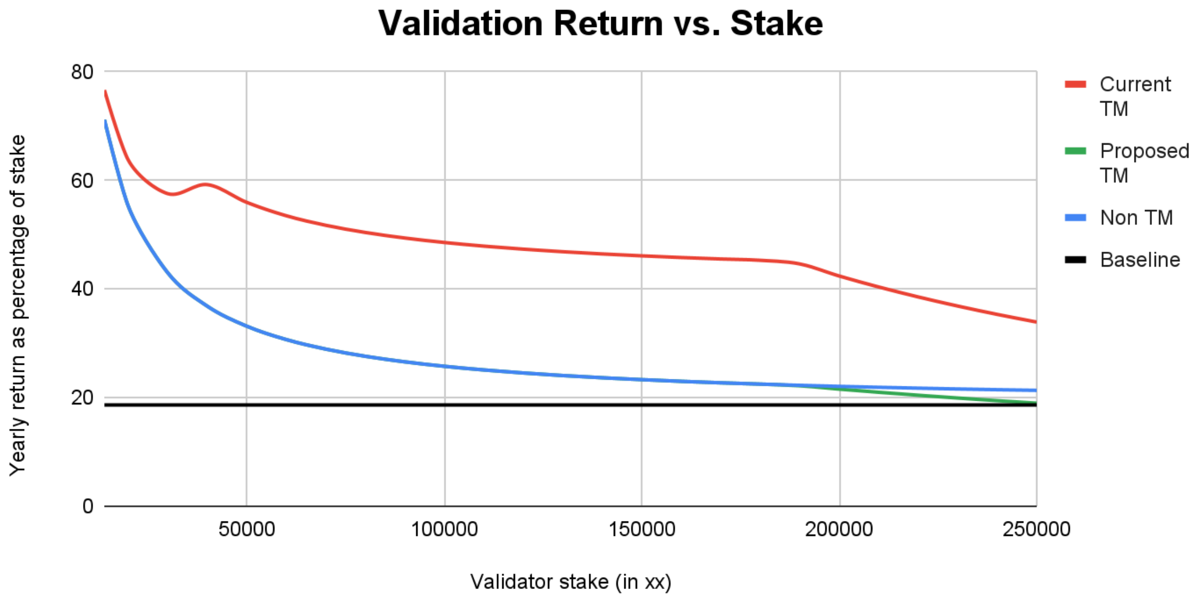
- *Make Team Multiplier take its share of rewards and return them back to the rewards pool*
- *Make it possible to adjust the amount of TM you receive (up to your maximum) through the Team Multiplier applet*
- *Increase minimum requirements for TM (Set realtime and precomp requirements)*

"

The third bullet point is irrelevant in economic terms, so won't be discussed in this document.

Solution 1. Multiplier takes share of rewards

Let's start by evaluating the impact of the team multiplier taking its share of rewards. In order to do this, we can reuse the model from the last section, but recompute all the returns taking the multiplier values into account for TM nodes. This leads to the following graph.



The green line (noticeable past 200k stake) shows the new return curve for TM nodes, which is exactly equal to Non TM nodes, except for very large stakes, which is not really an issue, since the large stake might belong to the node operator but be in a nominator wallet, which means it is split by Phragmen between more nodes.

The graph shows that having the TM take its share of the reward solves the disparity between TM and non TM nodes. Essentially, it brings TM nodes down to the same returns as non TM nodes. However, notice that the new average staking rewards baseline is down to 18.64%, a 41.74% reduction from the previous value of 32%. This means that validating is now always more profitable than nominating, while overall staking returns are reduced network wide.

This is a result of the rewards that are allocated to the multiplier being taken from the total nominator rewards (after commission is paid to validators). Furthermore, the daily total reward value remains the same, since the multiplier stake still doesn't exist economically (multiplier comes from team coins under custody).

It is then important to minimize the impact of the team multiplier in the reduction of staking rewards. This can be done by reducing the multiplier values applied to each node.

Solution 2. Adjustable multiplier

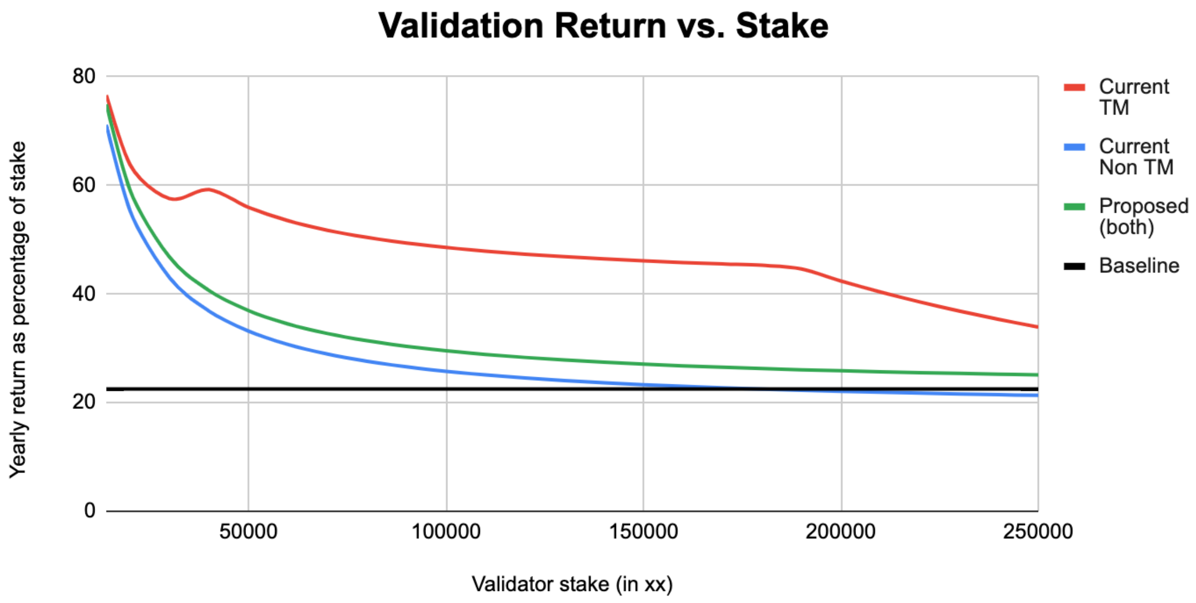
This is where making the multiplier adjustable by each node operator comes into play. In order to maximize the network's staking returns, each node operator is incentivized to take the lowest amount of multiplier it needs.

On average, each TM node operator needs a multiplier amount that complements its own stake up to the network's average (non foundation) stake. This value can be computed as (108.9M - 20M) / (370-8) = ~245k.

In order to analyze this solution, the multiplier script was modified to include this new calculation, which selects the multiplier for each node using the following formula: multiplier = min(node_cap, 245k-own_stake). This means that each node that requires the full amount of their multiplier will be able to use it, but nodes that have enough own stake will select lower multiplier values. The overall multiplier cap of ~233k is still maintained (more on this later).

The result of this tweaked multiplier script is that the total multiplier value is reduced from ~63.5M to ~37.5M, a decrease of around 40%. This means that now, the percentage of rewards taken by multipliers decreases from 41.74% to 29.64%, which results in the baseline staking rewards improving from 18.64% to 22.5%.

The results are displayed in the following graph, where it can be seen that rewards improve for everyone: TM and non TM nodes are equal, and better than current non TM nodes, and the baseline staking rewards are also raised.



Note that this analysis is always for the average situation: by selecting a multiplier that puts its node at the network average stake, the TM node operator is ensuring that it is absolutely safe from being dropped from the active set.

However, in reality, it is expected that node operators could select even lower multiplier values, which would place them with enough stake to not be, for example, in the bottom 10 staked nodes. This would lead to even lower total multiplier value, which increases staking returns for everyone, at the cost of a small increase in risk.

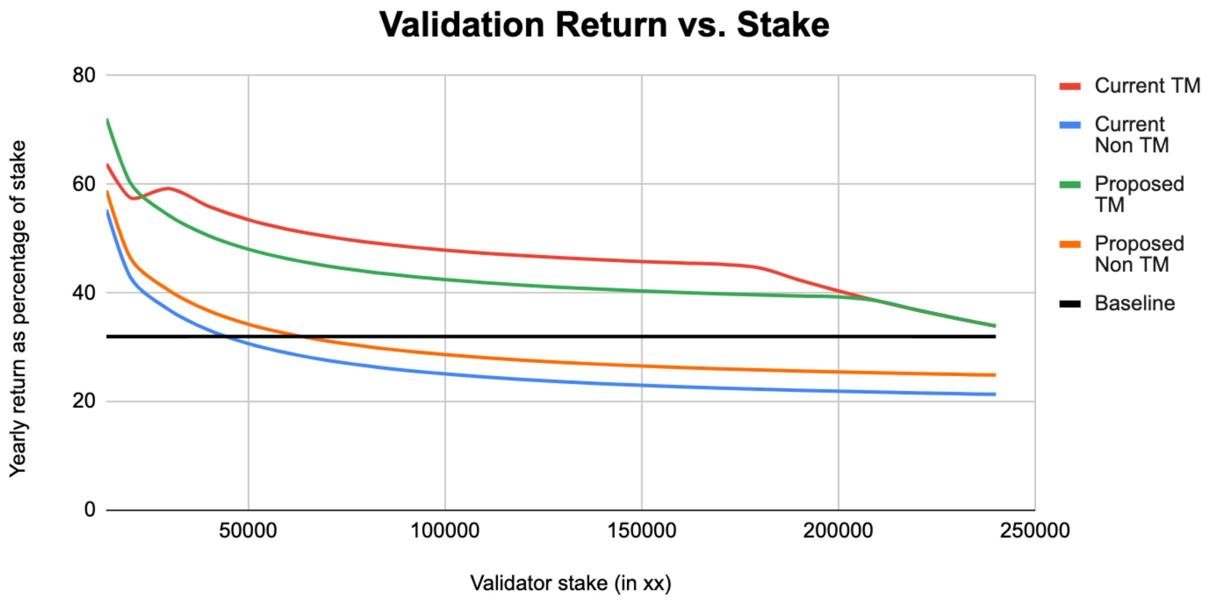
Multiplier Cap

Looking at the analysis above, it is clear that the current overall multiplier cap is too high, i.e., most TM nodes have a multiplier value that is unnecessarily large. This is because the multiplier cap is computed based on the ideal average stake of the network. In my personal opinion, the computation should be changed to use the actual average non foundation stake of the network instead. This would lead to the TM cap being reduced from ~233k to (245k - 31.7M/291) ~ 136k.

Let's analyze the impact of this new multiplier cap, first with the current system, then combined with the team multiplier taking its share of rewards, and finally with per node multiplier selection.

Solution 3. Lower cap

The result of the multiplier cap being lowered is that the total multiplier value is reduced from ~63.5M to ~38.8M, a decrease of around 39%. This means that the average election stake is reduced from ~421k to ~353k, a decrease of 16%. This reduction allows some nomination stake that is now in non TM nodes to flow back into the majority of TM nodes, providing a good first step to balance the system. The results are shown in the following graph.



Note that, since the multiplier is not taking its share of rewards in this analysis, there is still a gap between TM and non TM nodes. However, the gap does get smaller, and validating as a non TM node becomes profitable up to slightly higher stakes.

Solution 4. Lower cap + Multiplier takes share of rewards

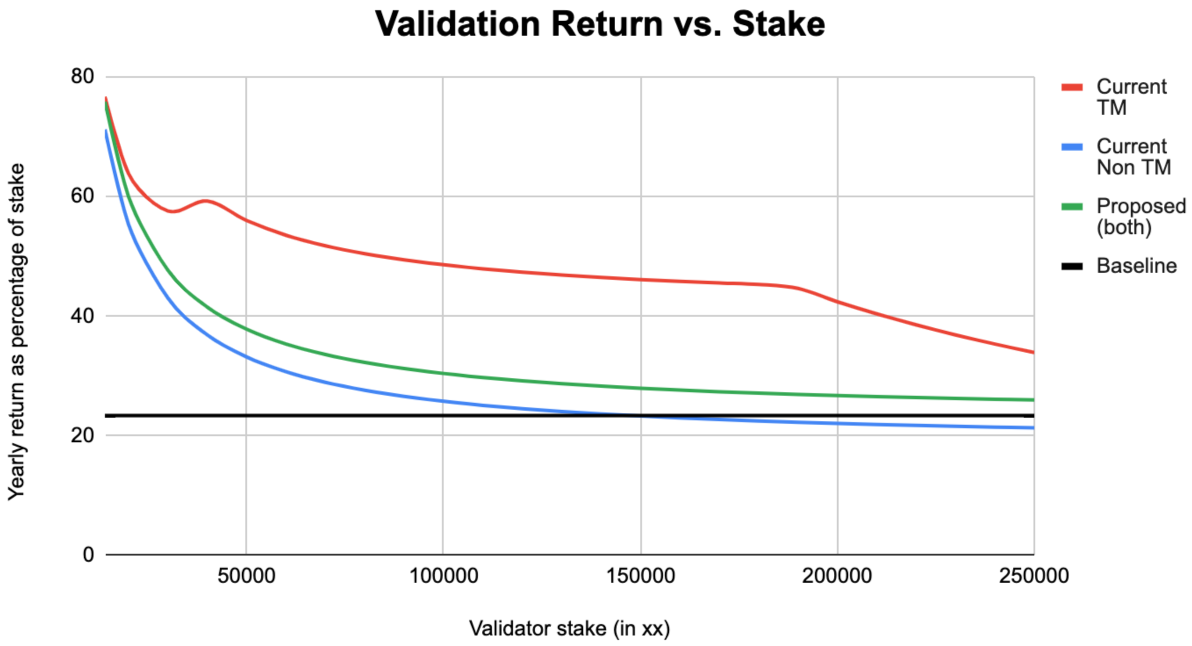
Now we define that the multiplier is taking its share of rewards, while maintaining the lower overall cap. Comparing the numbers to **Solution 2**, the percentage of rewards taken by multipliers has a slight increase from 29.64% to 30.35%, which results in the baseline staking rewards slightly reducing from 22.5% to 22.3%.

This solution is very effective, as it is very close to what is achieved by **Solution 2**, but much easier to implement.

Solution 5. Lower cap + Adjustable multiplier

Finally, the ultimate solution is to combine the lower cap, with the ability of nodes to select their own multiplier value. This results in the total multiplier value being reduced from ~63.5M to ~32.8M, a decrease of 48.2%. This also means that the percentage of rewards taken by multipliers decreases to a minimum of 26.9%, which results in the baseline staking rewards improving to 23.4%.

The results of this final solution are shown in the following graph. It is quite similar to the graph for **Solution 2**, with the difference that staking rewards improve for all stakers: TM validators, non TM validators and nominators.



NOTE:
It is important to stress that the portion of rewards that is allocated to the multiplier is **NOT claimable**. This means that while staking rewards decrease for all stakers with the inclusion of multiplier stake, the actual coins are kept in the staking rewards pool. This simply means that the pool will last longer, as only real stakers can keep collecting rewards.

Proposal

My recommendation is that the team should take a three-phased approach in order to fix the multiplier system.

1. The first phase would modify, as proposed in this document, the multiplier cap computation so that it uses real average stake instead of the ideal one. This change is a trivial modification to the multiplier script. The quicker this is done, the sooner the rewards gap between TM and non TM nodes can be reduced. This corresponds to **Solution 3**.

2. The second phase would implement, as proposed by Ben, multiplier stake being counted when computing staking rewards. This phase requires changes to the xx network blockchain's runtime (proof of concept started, 2 more days of work required to complete implementation and testing of pallet migration), and subsequent on-chain voting to perform the upgrade. This corresponds to **Solution 4**.

3. The third and final phase would implement the remainder of Ben's proposal, which involves multiplier script and backend changes, estimated at around 1 week of work. This corresponds to **Solution 5**.

After phase 2 is implemented, as analyzed previously, the overall staking rewards of the network drop by around 30%. In order to curb this reduction, I am making a proposal to the xx foundation to increase its stake in the network.

Currently, there are about ~16M coins belonging to the xx foundation that are already vested, but still kept under custody. My expectation is that the xx foundation will remove those coins from custody and stake them on the 8 bootnodes that it runs. This would increase the overall stakeable supply and the staking ratio, without affecting staking rewards, since the xx foundation stake would continue to be concentrated in its own (very over-staked) nodes only. This proposal would raise the average per-validator reward from ~262 to ~300 coins, an increase of 14.5%. The end result is that the 30% reward reduction is now applied to 114.5%, which results in a final number of ~80%, i.e. only a 20% reduction from current rewards.