

Managing Forks

The ledger is permitted to fork at slot boundaries. The resulting data structure forms a tree called a blockstore. When the validator interprets the blockstore, it must maintain state for each fork in the chain. It is the responsibility of a validator to weigh those forks, such that it may eventually select a fork. Details for selection and voting on these forks can be found in [Tower Bft](#)

Forks

A fork is as a sequence of slots originating from some root. For example:

2 - 4 - 6 - 8 / 0 - 1 12 - 13 \ / 3 - 5 \ 7 - 9 - 10 - 11 The following sequences are forks:

- {0, 1, 2, 4, 6, 8}
- {0, 1, 3, 5, 12, 13}
- {0, 1, 3, 5, 7, 9, 10, 11}

Pruning and Squashing

As the chain grows, storing the local forks view becomes detrimental to performance. Fortunately we can take advantage of the properties of tower bft roots to prune this data structure. Recall a root is a slot that has reached the max lockout depth. The assumption is that this slot has accrued enough lockout that it would be impossible to roll this slot back.

Thus, the validator prunes forks that do not originate from its local root, and then takes the opportunity to minimize its memory usage by squashing any nodes it can into the root. Although not necessary for consensus, to enable some RPC use cases the validator chooses to keep ancestors of its local root up until the last slot rooted by the super majority of the cluster. We call this the super majority root (SMR).

Starting from the above example imagine a max lockout depth of 3. Our validator votes on slots 0, 1, 3, 5, 7, 9. Upon the final vote at 9, our local root is 3. Assume the latest super majority root is 0. After pruning this is our local fork view.

SMR 0 - 1 12 - 13 \ / 3 - 5 ROOT \ 7 - 9 - 10 - 11 Now imagine we vote on 10, which roots 5. At the same time the cluster catches up and the latest super majority root is now 3. After pruning this is our local fork view.

12 - 13 / 3 - 5 ROOT SMR \ 7 - 9 - 10 - 11 Finally a vote on 11 will root 7, pruning the final fork

3 - 5 - 7 - 9 - 10 - 11 SMR ROOT [Previous Leader Rotation Next Stake Delegation and Rewards](#)