An analysis of turbine propagation with merkle shreds

1 Definitions

Assume a block comprised of only 1 FEC set containing 32 data and 32 coding shreds. In order to recover the block, a node needs to receive at least 32 shreds.

Let layer 0 be the turbine root for each tree.

For the purpose of this analysis assume that malicious nodes receive the block through a side channel and are incentivized to propagate the block to as many participants as possible. If a node is online, it will always broadcast any received or recovered shreds. If a node is malicious, it will always broadcast all shreds.

- p the probability that a given node is online
- m the probability that a given online node is malicious.
- $C_i(l)$ the probability that a node in layer l has i-th coding shred
- $D_i(l)$ the probability that a node in layer l has i-th data shred
- $B(l)$ the probability that a node with data shred in max layer l has a playable block
- $R(l)$ the probability that a node with a data shred in layer l is missing that data shred, and is able to recover the block.

2 Determining a lower bound

Given the data fanout in mainnet is 200 we can analyze a network with 40,000 nodes and determine the lower bound by assessing a node that is part of layer 2 for all shreds.

2.1 Coding shreds

Under construction

2.2 Data shreds

Data shreds will be present if either the turbine parent is online and has the shred, or if we are able to recover the shred through the erasure coding.

Under construction

2.3 Recovery

Note that turbine does not employ partial reed solomen, so in order to recover the i-th data shred, we must have recovered the entire block b of which it is a part of. Thus we must receive at least 32 shreds total.

Under construction

Preprint. Under review.

2.4 Propagation

Finally, we have the upper bound on probability of a node in receiving a playable block if only p part of the network is online, the probability that we receive all 32 data shreds and the probability that we recover while missing a shred:

Under construction

3 Results

We show the lower bound along with some simulations. Equal stake weight simulation [1] uses a 10,000 node network with equal stake and shred recovery. Mainnet stake weight simulation [2] mimics the exact node count and stake distribution of mainnet and does not perform shred recovery.

Using the mainnet stake distribution we see a vast increase in recovered stake, although node counts are still extremely low. This can be explained by the current stake distribution being highly favorable for the L1 nodes - the top 75 staked nodes hold close to 50% of network stake.

Table 1: Median stake recovered with no malicious, 10K trials

Percentage online	Equal stake	Mainnet stake
33%	0%	0%
50%	0%	14.06%
60%	0.05%	35.89%
61%	0.13%	37.67%
62%	0.33%	39.14%
63%	0.85%	41.28%
64%	1.79%	43.27%
65%	4.2%	45.38%
66%	10.29%	64.17%
67%	28.84%	50.51%
68%	51.87%	53.79%
70%	65.80%	60.94%
75%	74.84%	74.57%

Table 2: Median stake recovered with 33% malicious, 10K trials

Percentage online	Equal stake	Mainnet stake
33%	33%	33%
40%	33%	33%
45%	33.3%	33.09%
46%	33.4%	33.46%
47%	33.54%	33.58%
48%	33.71%	34.78%
49%	33.97%	36.21%
50%	34.28%	39.93%
51%	34.70%	42.13%
52%	35.09%	43.42%
53%	35.85%	45.23%
54%	36.88%	46.42%
55%	37.96%	47.95%
60%	48.95%	55.51%
66%	64.05%	64.08%
75%	74.98%	74.59%

[1] https://github.com/AshwinSekar/turbine-simulation/blob/master/src/main.rs

[2] https://github.com/AshwinSekar/solana/commits/turbine-simulation

4 An upper bound on versions of a block

Given the following rules for replay, we can upper bound the number of versions that will reach a vote threshold.

- If you receive 2 shreds for the same FEC set with different merkle roots, do not further vote on the block. Additionally send the duplicate witness proof to turbine descendants and through gossip.
- If a witness is passed from a turbine parent or through gossip, do not attempt to vote on the block.

In the worst case assume 33% of the network is malicious and will vote on all versions of the block a duplicate leader broadcasts. We see that the vote threshold must be greater than this number, to maximize liveness we chose 34%.

Assume for sake of contradiction that there can be 6 versions that reach the 34%. This requires at least 6% of honest stake to vote, along with the 33% malicious that will vote on every version.

Given the requirements for voting we outlined above, it is easy to see that the 1% of honest stake that votes on each version must be in separate network partitions.

If we assume that 2 validators are in the same network partition we would require that every turbine parent for the last FEC set be malicious, otherwise an honest node would have been able to either pass a duplicate witness, or send an alternate version of the block.

We can model the probability by considering for each shred we are either:

- The turbine root (leader can choose which version to send us)
- The child of a malicious turbine parent
- Our parent is honest, and has also only been sent this version and cannot produce a duplicate witness proof.

Additionally there must be a gossip failure, or no node must be able to gossip a duplicate witness proof.

Given that the 1% stake must be in a network partition, we can look at the propagation rates from Table 2 to estimate the size of the partition. Assuming the 33% malicious are able to operate in any partition, we need at least 15% honest nodes in the partition in order to have a 1% propagation rate (not including the malicious stake).

However this leaves us with

$$6 * 15\%(\text{honest}) + 33\%(\text{malicious}) > 100\%$$

stake in the network a contradiction. Thus we can conclude that there are less than 6 versions that can reach the 34% vote threshold. □