One of the aspects that seem to surprise many people that only casually follow the progress of Ethereum Serenity is the quadratic leak that is imposed upon validators for being offline and missing a slot. For those unwilling/unable to read the Casper FFG paper, I wrote up a quick explanation of how one could arrive at this solution using fairly conventional wisdom from distributed systems in computer science.

I am curious what others think of this explanation. Any and all notes/insights/feedback are welcome.

Liveness

One of Serenity's main goals is to guarantee liveness (i.e. continue to finalize blocks) in the event of a major internet partition (e.g. World War 3). This liveness guarantee comes at a steep cost which makes it important to understand the predicament and possible tradeoffs.

The CAP Theorem

The CAP Theorem for distributed systems, tells us that:

You can't simultaneously guarantee more than two of the following:

- Consistency: Every read receives the most recent write or an error
- Availability: Every request receives a (non-error) response – without the guarantee that it contains the most recent write
- Partition Tolerance: The system continues to operate despite an arbitrary number of messages being dropped (or delayed) by the network

The Assumption

By viewing the argument through the lens of the CAP Theorem

, we can deduce the rationale for the inactivity leak

by accepting the following assumption:

No network can guarantee message delivery because the network itself is not safe from failures (e.g. client disconnects, data center outages, connection loss).

Partition Tolerance

Since message delivery cannot be guaranteed, the logical thing to do is to tolerate prolonged message loss. This is equivalent to Partition Tolerance

.

Sidenote: Think of the World War 3 scenario as a dysphemism for prolonged message loss between groups of validators.

With Partition Tolerance

as a hard requirement, we are now limited to tradeoffs between Consistency

and Availability

.

World War 3

In the World War 3

scenario, where the network is severed, the validators are split into two partitions. FromCasper FFG, we know that in order for both partitions to continue finalizing blocks, we need two-thirds majority of validators to be online in both partitions. This is obviously not possible; however, we can prevent the chain from stalling forever if we are willing explore a compromise between our Availability

and Consistency

guarantees.

The Compromise

This is accomplished by introducing an inactivity leak

that drains the deposit of unresponsive validators each time a slot is missed until the remaining validators in each partition become the supermajority.

At this point, blocks in both network partitions can begin to finalize; however, if the network partition is healed we are left with two valid and separate networks.