In standard chain-based consensus algos, proposal

of transaction packages and confirmation

of proposals are the same process; a block/collation is simultaneously a proposal for a new transaction package to be added to the history and a vote in the process of confirming previous blocks.

As discussed in the previous post on consensus/execution separation, even if consensus on transaction ordering and state calculation are separated, the nodes participating in the transaction ordering process still need to have access to the state, and still need to perform state executions, because they need to know whether or not the transactions they accept will pay for gas.

However, we could conceivably go one step further, and separate the sharded chain into three

processes:

1. Collation proposal
2. Collation consensus
3. State execution

(2) must

have the property that a participant is not local to any single shard, but rather bounces between shards very rapidly, so that one cannot attack the system by just corrupting a few validators in one particular shard. However, as I already mentioned in the previous post, (3) does not

need this property, because state execution is not a consensus game; it can be designed in a Truebit-style interactive protocol which leads to correct answers even if more than 90% of the participants are malicious.

Here, I will go further and posit that participants in (1) can also be local to one specific shard. This means that participants in (2) would not need to deal with stateless client mechanisms, witnesses and other related mechanics at all, and would simply need to adjudicate availability of regular blocks.

One possible mechanism is simple: anyone can make and sign proposals (ie. packages of transactions) for any shard, and each proposal must contain a “pass-through fee”. A collation is made in the usual way, except instead of including transactions directly, a collation simply includes exactly one proposal. The pass-through fee of the proposal is deducted from the proposer and given to the collator on the VMC level. In this way, proposers do

need to specialize in making proposals for particular shards, but validators do not need to concern themselves with this at all, as the fee that they receive is guaranteed as part of the proposal header.

Note that one could replicate something similar extra-protocol by having validators outsource the job of making proposals with cryptoeconomic proofs. Validators could solicit collation proposals, where each proposal would come with a signed message signed by a user with a bonded security deposit, saying “if you make a collation in this period on top of this parent, using these transactions, then either your balance will increase by at least XXX ETH due to fees, or I will pay the difference out of my bond”.

Either approach allows validators who need to frequently bounce between shards to not need to worry about the state of any specific shard, allowing specialized nodes to perform the (state-demanding) task of actually creating collations for each shard.