

The following is a scheme that allows users to only store ETH for paying transaction fees on one shard, while still being able to send transactions on any shard and pay the required fees for them. This can be incorporated into the protocol intrinsically, or built entirely as a layer-2 system.

On some particular shard, there is a “fee settlement contract” (FSC). Anyone wishing to send transactions must deposit some ETH into the FSC, from where it can be withdrawn with a 7 day delay. On every shard, there is a “fee forwarding contract” (FFC). If someone wishes to pay a fee along with a transaction that they want to send, the following happens:

1. The transaction execution itself, in lieu of a direct ETH balance transfer, makes a call to the FFC in the same shard, with the fee amount as an argument.
2. The FFC simply records (i) the fee amount, (ii) the current block producer, which is the presumed recipient of the fees, and (iii) the fee payer.
3. At the end of every (block | hour | other timespan), it sends a single cross-shard call containing all of the fee records to the FSC. Note that for fees within one block, the recipient of the fees only needs to be specified once; additionally, one can save space by informing the FFC of the index of any specific fee payer address, allowing the FFC to substitute the address (20 bytes) with the fee payer (possibly ~4 bytes). All in all, optimal overhead for the fee is ~8 bytes per transaction.
4. The FSC processes all of the records, and appropriately increments and decrements the accounts. Note that if an account has a balance  $D$ , and there are  $N$  shards, it restricts the fees allowed to be paid from each shard to  $D/N$ .

A block proposer can verify fees as follows:

1. Check that the transaction, when executed, actually sends the message to the FFC
2. Check that the sender has an account on the FSC
3. Check that the sender’s account balance is high enough to make the fee payment in the FSC. For full safety, one can do the following calculation. Suppose that an asynchronous call will take less than one FFC call period (this can always be done by simply making the period appropriately long). Suppose the sender’s balance is  $D$ , and there are  $N$  shards. Suppose  $P$  fees have already been registered in the FFC since the last period, and the current transaction’s fee is  $F$ . Accept the transaction if  $D / N - P \geq F$ .

If there are too many transactions for one FSC to be able to handle them, then one can place multiple FSCs in multiple shards. Users would have accounts in one specific FSC, and the transaction paying the fee would specify to the FFC which FSC it goes to.