Step-by-Step Process of Decentralized Inference

The following outlines the decentralized inference process, taking into account the techniques and protocols established within the previous sections:

Step 1: User Request Submission

Users submit inference requests to the blockchain. Each request includes the user’s input data for the AI model and the fee paid by the user, which determines the priority of the request in the system.

Step 2: Priority Queue Management

Requests are registered in a smart contract that acts as a priority queue manager. The smart contract organizes the requests based on the respective fees, ensuring those who paid higher fees receive higher priority in the execution queue.

Step 3: Inference Committee Selection

At the beginning of each slot, a committee of nodes is selected to perform the inference task. The selection is based on the output of a VRF that each node executes using its private key. Nodes generating qualifying random numbers are chosen for the committee.

Step 4: Inference Execution

The selected committee of nodes independently performs the requested AI infer- ence task. Each node computes the result using the input data provided in the user’s request.

Step 5: Commitment to Inference Results

To prevent free-riding, each node in the committee commits to its computed infer- ence result using a cryptographic commitment scheme. This involves sending a hash of the result and a nonce to the blockchain within a specific timeframe.

Step 6: Reveal of Inference Results

Within a subsequent timeframe, each node reveals its committed inference result by submitting both the result and the nonce to the blockchain. If a node fails to reveal its result, submits after the deadline, or reveals a result that does not match its commitment, it faces a penalty.

Step 7: Result Aggregation and Majority Vote

Submitted inference results are collected by a smart contract that performs a majority vote to determine the consensus result. The result most frequently reported by the committee is considered the correct outcome.

Step 8: Penalty Enforcement and Reward Distribution

The smart contract identifies and penalizes nodes that submitted incorrect results (not aligning with the majority). Penalties may include slashing of stakes or reduction in reputation. Nodes that contributed to the majority result receive rewards, as specified by the system's incentive structure, for their accurate and honest work.

Step 9: Finalization and User Notification

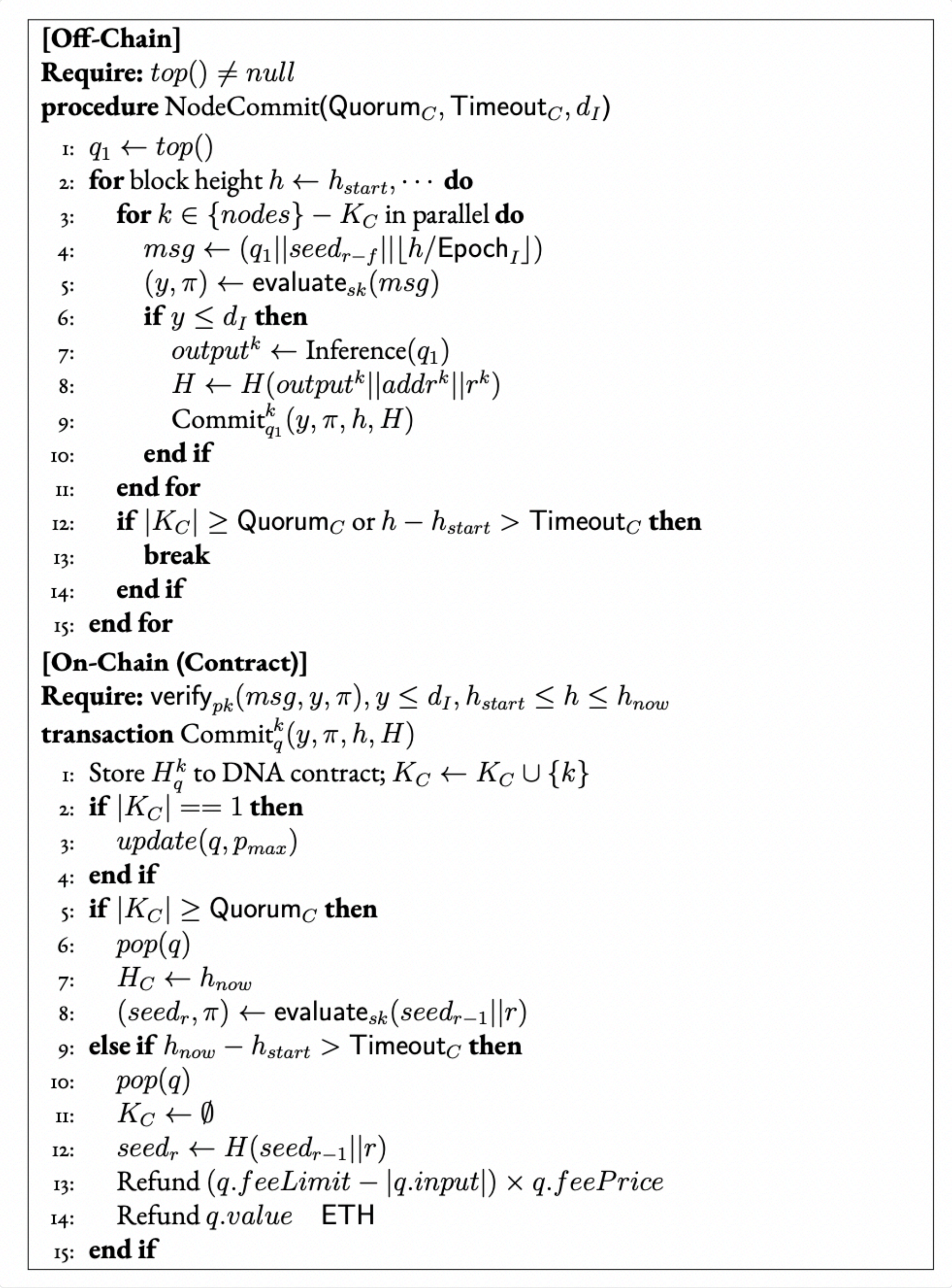
The smart contract finalizes the consensus result of the inference task and records it on the blockchain. The user is notified of the result, which concludes the inference process.

This step-by-step summary encapsulates the core sequence of operations within Nesa's decentralized inference system. It details the journey from request submission to the final delivery of the consensus inference result, highlighting the roles of smart contracts, priority queues, committee selection, result commitment and reveal, and aggregation via majority vote, all without any specific privacy-preserving or cryptographic enhancements from Chapter 4.

The system ensures a transparent, fair, and secure process, motivating nodes to produce accurate results and maintaining reliability across the network.

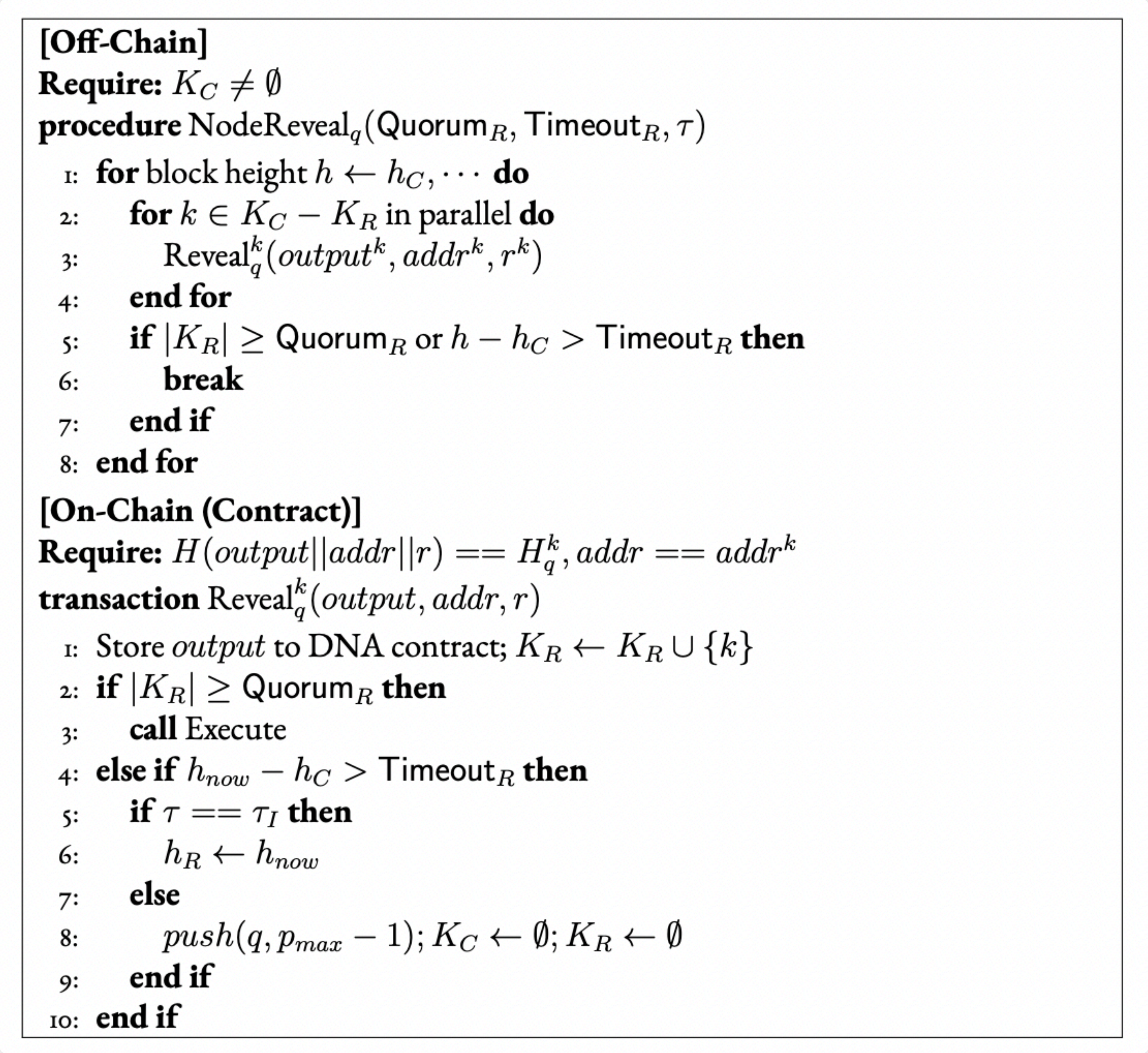
Nesa's default aggregation system by smart contract epitomizes an equilibrium between democratic principles and technological enforcement, allowing us to mitigate risks associated with node aberrance or malicious intents. By provision of penalties and reputation diminution for deviant or deceitful submissions, we are engendering a self-regulating ecosystem that encourages computational precision and integrity.

The process maintains detailed control over results set retrieval, an array of sophisticated statistical methods for conclusive consensus, and reward allocation commensurate with the predefined incentives ensuring adversarial resilience and motivation for continued honest engagement by Nesa miners.



```
[Off-Chain]
Require: $top() \neq null$
procedure NodeCommit($\mathsf{Quorum}_C, \mathsf{Timeout}_C, d_I$)
  $q_1 \leftarrow top()$
  for block height $h \leftarrow h_{start}, \cdots$ do
    for $k \in \{nodes\} - K_C$ in parallel do
      $msg \leftarrow (q_1 || seed_{r-f} || \lfloor h/\mathsf{Epoch}_I \rfloor)$
      $(y, \pi) \leftarrow \mathsf{evaluate}_{sk}(msg)$
      if $y \leq d_I$ then
        $output^k \leftarrow$ Inference$(q_1)$
        $H \leftarrow H(output^k || addr^k || r^k)$
        $\mathrm{Commit}^k_{q_1}(y, \pi, h, H)$
      end if
    end for
    if $|K_C| \geq \mathsf{Quorum}_C$ or $h - h_{start} > \mathsf{Timeout}_C$ then
      break
    end if
  end for
[On-Chain (Contract)]
Require: $\mathsf{verify}_{pk}(msg, y, \pi), y \leq d_I, h_{start} \leq h \leq h_{now}$
transaction $\mathrm{Commit}^k_q(y, \pi, h, H)$
  Store $H^k_q$ to DNA contract; $K_C \leftarrow K_C \cup \{k\}$
  if $|K_C| == 1$ then
    $update(q, p_{max})$
  end if
  if $|K_C| \geq \mathsf{Quorum}_C$ then
    $pop(q)$
    $H_C \leftarrow h_{now}$
    $(seed_r, \pi) \leftarrow \mathsf{evaluate}_{sk}(seed_{r-1} || r)$
  else if $h_{now} - h_{start} > \mathsf{Timeout}_C$ then
    $pop(q)$
    $K_C \leftarrow \emptyset$
    $seed_r \leftarrow H(seed_{r-1} || r)$
    Refund $(q.feeLimit - |q.input|) \times q.feePrice$
    Refund $q.value$ $\mathsf{ETH}$
  end if
```

Off-chain and on-chain algorithms for the Commit phase. The Commit phase is stage one of the two-part subphase for Nesa's commit-reveal paradigm within the system's decentralized inference protocol.



```
[Off-Chain]
Require: $K_C \neq \emptyset$
procedure $\text{NodeReveal}_q(\mathsf{Quorum}_R, \mathsf{Timeout}_R, \tau)$
    for block height $h \leftarrow h_C, \cdots$ do
        for $k \in K_C - K_R$ in parallel do
            $\text{Reveal}_q^k(output^k, addr^k, r^k)$
        end for
        if $|K_R| \geq \mathsf{Quorum}_R$ or $h - h_C > \mathsf{Timeout}_R$ then
            break
        end if
    end for
[On-Chain (Contract)]
Require: $H(output||addr||r) == H_q^k, addr == addr^k$
transaction $\text{Reveal}_q^k(output, addr, r)$
    Store $output$ to DNA contract; $K_R \leftarrow K_R \cup \{k\}$
    if $|K_R| \geq \mathsf{Quorum}_R$ then
        call Execute
    else if $h_{now} - h_C > \mathsf{Timeout}_R$ then
        if $\tau == \tau_I$ then
            $h_R \leftarrow h_{now}$
        else
            $push(q, p_{max} - 1); K_C \leftarrow \emptyset; K_R \leftarrow \emptyset$
        end if
    end if
```

Off-chain and on-chain algorithms for the Reveal phase. The Reveal phase is stage two of the two-part subphase for Nesa's commit-reveal paradigm within the system's decentralized inference protocol. The Reveal phase allows nodes to disclose their previously committed results. Each node must reveal both the result m and the nonce r used in the commitment, allowing others to verify the hash against the commitment made in the previous phase. This phase also has a timeout limit, and failure to reveal on time, or revealing a different result from what was committed, will result in punitive measures against the offending node. After the reveal, submitted inference results are collected by a smart contract that performs a majority vote to determine the consensus result. The result most frequently reported by the committee is considered the correct outcome.

Previous
Aggregation of Inference Results

Next
Hybrid Design for Enhanced Privacy

Last updated 1 month ago