Internet Computer Consensus Protocol

PYD 2022/2/16

MIXLABS

What Is Consensus Protocol

Distributed Ledger （we talk about）

Replica Statement Machine (what is used in IC implement)

MIXLABS

Definition Of Security Of Consensus Protocol

Safety. All messages after consensus will form an ordered message sequence N, and the message sequence of the local view of all replicas is prefix of $N[:k] \preceq N$, where $k \leq |N|$

Liveness. The length of the consensus message sequence will continue to grow

MIXLABS

Communication Module

Synchronous Model. There exists some known finite time bound δ, such that for any message sent, it will be delivered in less than time

Partial Synchrony Model. For each subnet, communication among replicas in that subnet is periodically synchronous for short intervals of time; moreover, the synchrony bound δ does not need to be known in advance.

Asynchronous Model. In the asynchronous model, for any message sent, the adversary can delay its delivery by any finite amount of time, so that there is no bound on the time to deliver a message.

MIXLABS

Some Definition And Assumption

Fault Models:

Crash faults and Byzantine faults

Security Assumption:

$n > 3f + 1$ where n is the amount of replica in protocol and f is the amount of fault nodes

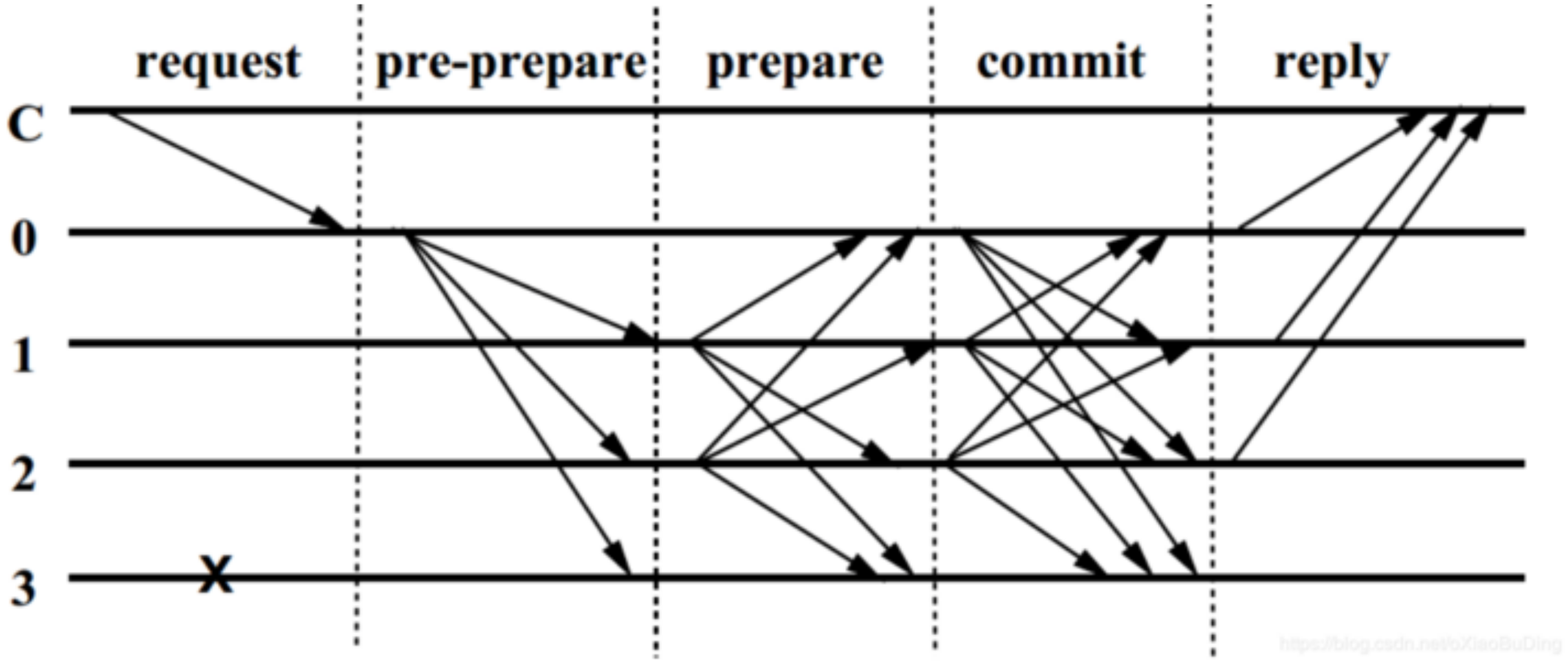
Community Assumption:

Asynchronous model for safety

Synchronous model for liveness

MIXLABS

Start From PBFT——Normal Case



MIXLABS

Start From PBFT——View Change

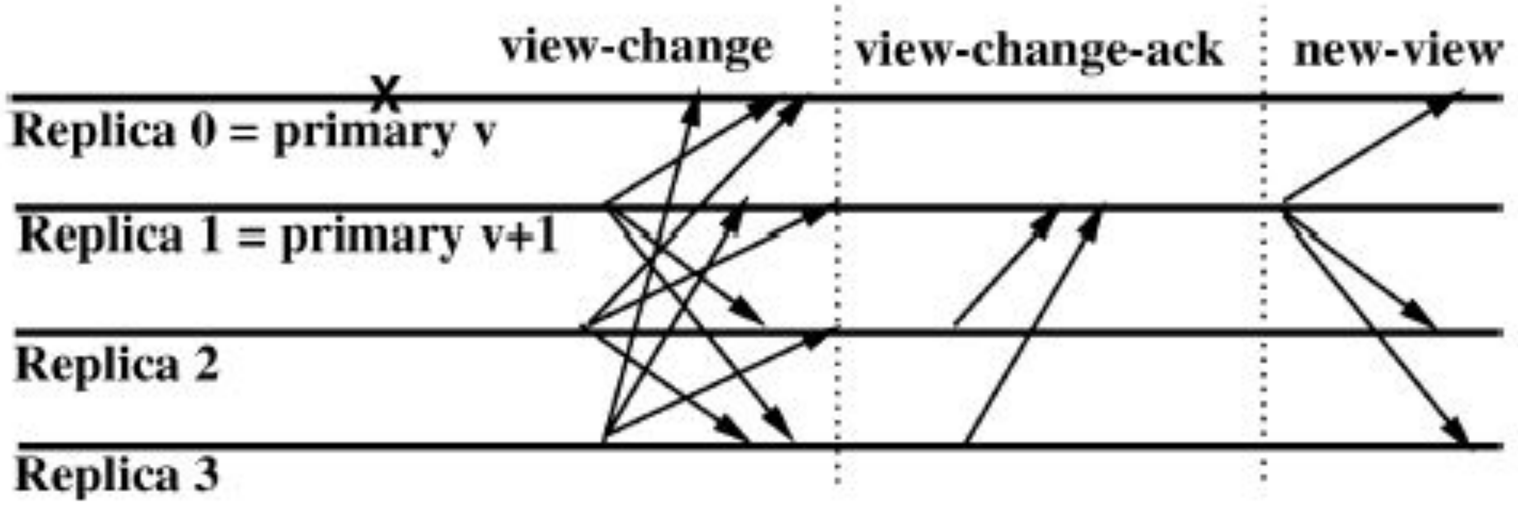


Fig. 2. View-change protocol: the primary for view v (replica 0) fails causing a view change to view $v+1$.

MIXLABS

ICC ——Cryptography Prime

Paring: e: $e(g^a, (g')^b) = e(g, g')^{ab}$ where $g \in \mathbb{G}$ and $g' \in \mathbb{G}$ are are elliptic curves of a special type

BLS Signature:
Private key : $x \in \mathbb{G}$ where $\mathbb{G}$ is a group with prime order q and the generator g
Public key : $V := g^x \in \mathbb{G}$
Hash function: $H_{\mathbb{G}'}$ which maps its input to an element of $\mathbb{G}'$, where $\mathbb{G}'$ is a group with prime order q and the generator g', oracle model
Sign function : $Sign(m) := H_{\mathbb{G}}(m)^x = (h')^x \in \mathbb{G}' = \delta$ where $h' \in \mathbb{G}'$
Verify function: test if $e(\delta, g) = e(h', V) \rightarrow e((h')^x, g) \rightarrow e(h', g^x) = e((h')^x, g)$

Aggregated Signature:

$$\text{Aggregated signature} : \delta \leftarrow \prod_j \delta_j = \prod_j (h')^{x_j} = (h')^{\sum_j x_j},\ V = \prod_j V_j = \prod_j g^{x_j} = g^{\sum x_j}$$

(t-out-of-n) Threshold Signature：
In the t-out-of-n threshold setting, we have n replicas, any t of which may be used to generate a signature on a message. In somewhat more detail, replica P_j holds a share $x_j \in \mathbb{Z}_q$ of the secret signing key $x \in \mathbb{Z}_q$, which is privately held by P_j, while the group element $V_j := g^{x_j}$ is publicly available. The shares $(x_1, \ldots, x_n)$ are a t-out-of-n secret-sharing of x,signature share $\sigma_j := (h)^{x_j} \in G$, signature share $\sigma_j := (h)^{x_j} \in G$, $\delta_j := (h')^{x_j} \in \mathbb{G}'$:

$$\text{Aggregated signature} : \delta \leftarrow \prod_j \delta_j^{\lambda_j} = \prod_j (h')^{x_j \lambda_j} = h'^{\sum_j x_j \lambda_j}$$

MIXLABS

ICC ——Cryptography Prime

Usage In PBFT: CBFT communication complexity reduction：

Normal case： $O(n^2) \rightarrow O(n)$

View Change: $O(n^3) \rightarrow O(n^2)$

Usage In ICC

Let n be the number of replicas in a subnet and let f be a bound on the number of corrupt replicas.

- The Consensus Layer makes use of an (f + 1)-out-of-n threshold signature to realize a random beacon
- The Execution Layer makes use of an (f + 1)-out-of-n threshold signature to realize a random tape, which is used to provide unpredictable pseudorandom numbers to canisters
- The Execution Layer makes use of an (n − f)-out-of-n threshold signature to certify the replicated state. This is used both to authenticate the outputs of a subnet and to implement the fast-forwarding feature of the IC’s chain-evolution technology

MIXLABS

ICC ——Block Making

block B

- the payload,
- the hash of B',
- the rank of the block maker,
- the height h of the block.

block B proposal

the block B,

- the block maker's identity, and
- the block maker's signature on B.

MIXLABS

ICC ——Notarization

block B notarization share

the hash of B
the height h of B
the identity of the supporting replica
the supporting replica's signature δ

block B notarization

the hash of B,
the height h of B
the set of identities of the n – f supporting replicas,
an aggregation of the n – f signatures

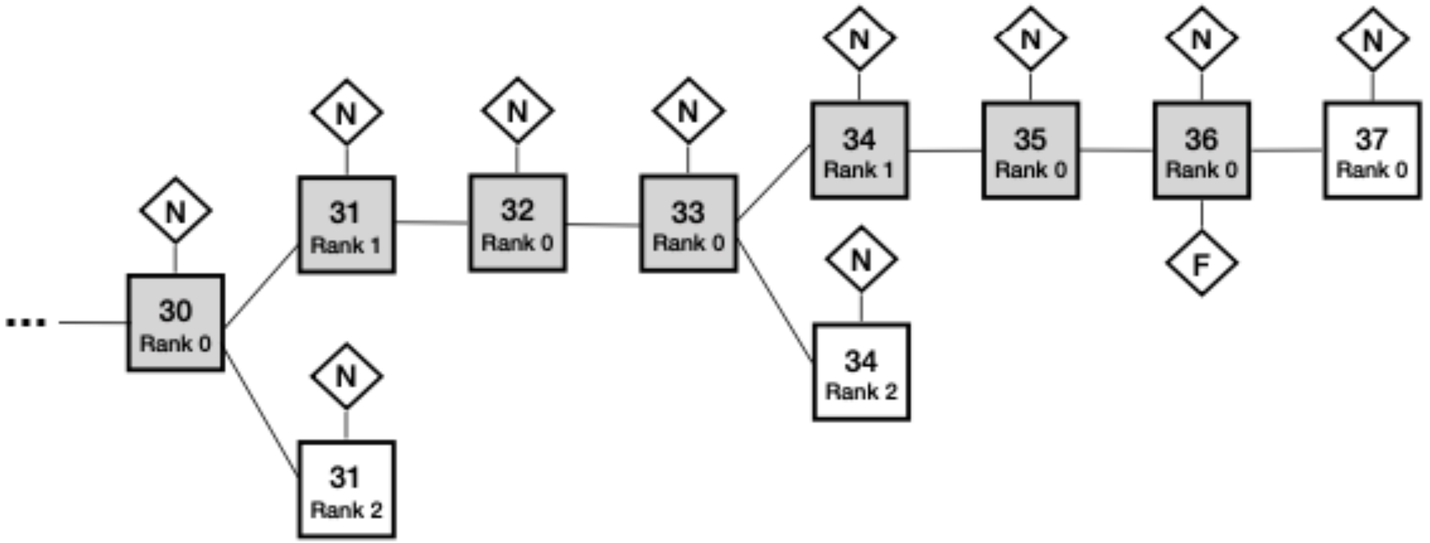
When a block is notarized: $n - f$ distinct replicas must support its notarization.

As soon as a replica obtains a notarized block of height h, it will finish round h, and will subsequently not support the notarization of any other blocks at height h. At this point in time, such a replica will also relay this notarization to all other replicas. Note that this replica may have obtained the notarization either by (1) receiving it from another replica, or (2) aggregating n – f notarization shares that it has received.

ICC ——Notarization

Block Tree

A block is effectively added to the tree of blocks when it becomes notarized



Replica will check if it supported the notarization of any block at height h other than block B (it may or may not have supported the notarization of B itself). If not, the replica will support the finalization of B by broadcasting a finalization share for B.

ICC —— Finalization

MIXLABS

block B finalization share

the hash of B

the height h of B

the identity of the supporting replica

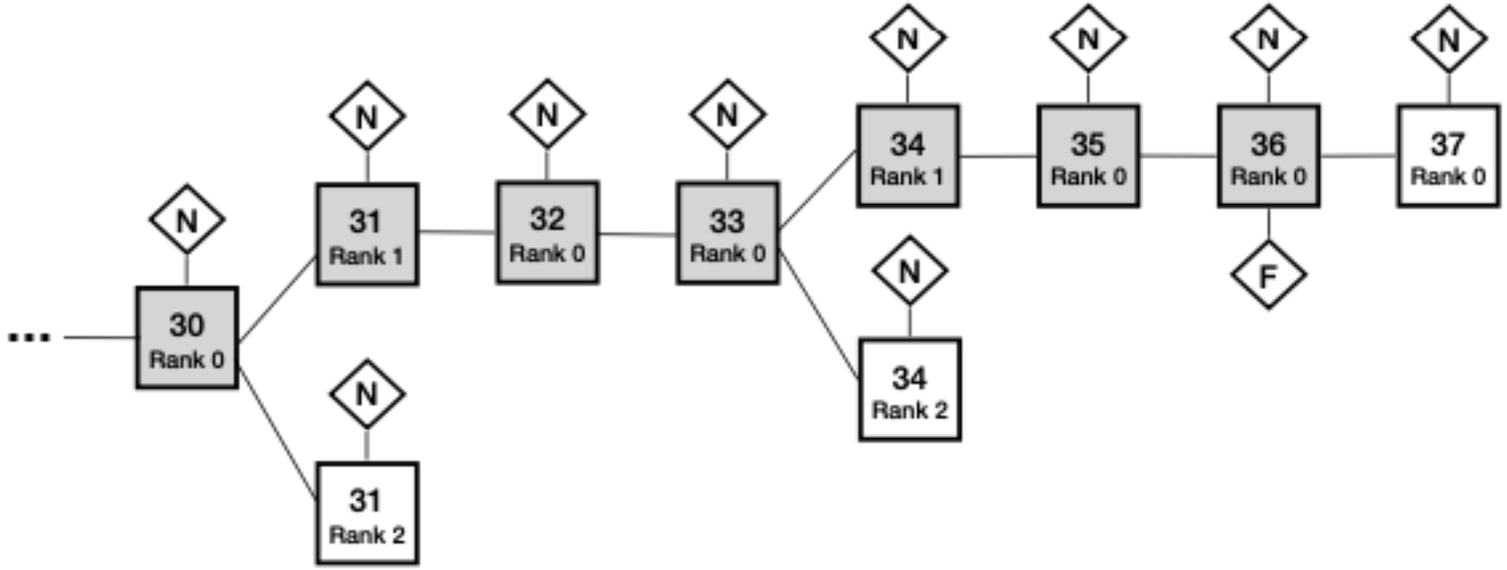
the supporting replica's signature δ

block B finalization

the hash of B,
the height h of B
the set of identities of the n – f supporting replicas,
an aggregation of the n – f signatures

Replica will check if it supported the notarization of any block at height h other than block B (it may or may not have supported the notarization of B itself). If not, the replica will support the finalization of B by broadcasting a finalization share for B.

ICC —— Reverse Finalization



When a block B is finalized, all block from root block to block B become finalized.

ICC ——Delay function

The protocol makes use of two delay functions, Δ_m and Δ_n, which control the timing of block making and notarization activity.

Recommended Definition: $\Delta_m(r) = 2\delta r$ & $\Delta_n(r) = 2\delta r + \varepsilon$

The Usage Of Delay Function:

Δ_m: change leader

Δ_n: adjust network delay

With these definitions, liveness will be ensured in those rounds in which (1) the leader is honest, and (2) messages really are delivered between honest replicas within time δ. Indeed, if (1) and (2) both hold in a given round, then the block proposed by the leader in that round will be finalized.

ICC ——Block Maker Selection

MIXLABS

Random Beacon:

As soon as a replica has received the random beacon for round h, or enough shares to contruct the random beacon for round h, it will relay the random beacom for round h to all other replicas. As soon as a replica enters round h, it will generate and broadcast its share of the random beacon at round h + 1.

Who To Select Block Maker:

In each round, a pseudo-random process is used to assign each replica a unique rank, which is an integer in the range $(0,...,n-1)$.The pseudo-random process is implemented using a random beacon. The replica of lowest rank is the leader of that round.

When Block Maker P To Propose A Block:

- At least $\Delta_m(r_P)$time units have passed since the beginning of the round
- there is no valid lower ranked block currently seen by P.

Improvement of ICC to PBFT

MIXLABS

More Fair： Select a new lead in each round(remove view change): Tx can be involved in block chain eventually unless as long as an honest replica become leader:

Faster: Lower communication complexity: from $O(n^2)$ to $O(n)$ and Pipeline process

Larger Network Scale

Stable Throughput: Although lateness maybe increase if the leader is corrupted or the network become asynchronous, but the throughput would not be impacted.