Internet Computer Consensus

PODC, July 2022

Jan Camenisch, Manu Drijvers, Timo Hanke, Yvonne-Anne Pignolet, Victor Shoup, Dominic Williams

DFINITY Foundation, Switzerland

We are hiring!
www.dfinity.org/careers
Outline

• What is the Internet Computer?
• Internet Computer Consensus
• Measurements
What is the mission of the Internet Computer?
What is the mission of the Internet Computer?

Platform to run **any computation**, using blockchain technology for decentralisation and security.
Nodes in Independent Data Centers

https://dashboard.internetcomputer.org/
Canister Smart Contracts: Combination of Data and Code

- Data: Memory pages
- Code: WebAssembly bytecode

Canister smart contract
Developers and users interact directly with Canisters on the IC.
Developers and users interact directly with Canisters on the IC

End user

UX

Internet Computer

DEPLOY

Developer
More than 60,000 canisters deployed

https://internetcomputer.org/showcase/
Scalability: Nodes and Subnets

Nodes are partitioned into subnets

Canister smart contracts are assigned to different subnets
Scalability: Nodes and Subnets

Nodes are partitioned into subnets

Canister smart contracts are assigned to different subnets

One subnet is special: it host the **Network Nervous System (NNS)** canisters which govern the IC

ICP token holders vote on
- Creation of new subnets
- Upgrades to new protocol version
- Replacement of nodes
- …
Each Subnet is a Replicated State Machine

State:
• canisters and their queues

Inputs:
• new canisters to be installed,
• messages from users and other canisters

Outputs:
• responses to users and other canisters

Transition function:
• message routing and scheduling
• canister code
Consensus on the Internet Computer
Requirements

High Throughput
• Several thousands of messages per second

Low Latency
• ~1 second (+ user network latency) for state changes

Robustness
• Tolerate bad communication links between nodes as well as Byzantine node behaviour
  • Safety under asynchrony
  • Liveness under short intervals of synchrony
• Graceful degradation
  • “slow path” is a simple variation on “fast path”

Simplicity
• Facilitate fast implementation and debugging
Consensus Properties

Messages are placed in **blocks**. We reach agreement using a blockchain.

The following properties must hold even if up to $f < n/3$ nodes misbehave:

- **Safety**: For any $i$, if two honest nodes think that the $i$-th block is agreed upon, they must have the same block.
- **Liveness**: For any $i$, at some point every honest node will consider the $i$-th block is agreed upon.

We use $n = 4, f = 1$ in examples.
Block Making

- Message (user → canister)
- Message (canister → canister)

Node selects available messages and combines them into a block together with reference to predecessor and meta-data and broadcasts it.
Block Making

- Message (user → canister)
- Message (canister → canister)

Node selects available messages and combines them into a block together with reference to predecessor and meta-data and broadcasts it.

```
...<24, 25, 26, 27, 28, 29>. 30
```

```
30’ → 30 → 30”
```
Notarization

The notarization process ensures that a valid block is known for every round.

**Step 1**
Node 1 receives a block proposal for height 30, building on some notarized height 29 block.

**Step 2**
Node 1 sees that the block is valid, signs it, and broadcasts it together with its notarization share.

**Step 3**
Node 1 sees that nodes 3 and 4 also published their notarization shares on the block.

**Step 4**
3 notarization shares are sufficient approval: the shares are aggregated into a single full notarization. Block 30 is now notarized, and nodes wait for height 31 blocks.
Nodes may notary-sign multiple blocks to ensure that at least one block becomes fully notarized.

**Step 1**
Node 1 receives a block proposal for height 30, building on some notarized height 29 block.

**Step 2**
Node 1 sees that the block is valid, signs it, and broadcasts it together with its notarization share.

**Step 3**
Nodes 1 sees another height 30 block, which is also valid, and it broadcasts it together with another notarization share.

**Step 4**
Both height 30 blocks get enough support to become notarized.
Multiple notarized blocks may exist at the same height, at least one per height
Random Beacon

At every height, there is an unpredictable random value shared by the nodes

BLS-Threshold Signatures

- Pseudo random (not predictable, no last actor bias)
- Non-interactive distributed key generation
- Non-interactive independent signature share creation
- Unique: for every message $m$ there exist one signature, regardless of the threshold group
Random Beacon

At every height, there is an unpredictable random value shared by the nodes.

**Step 1**
Node 1 has Random Beacon 29 and wants to help constructing Random Beacon 30.

**Step 2**
Node 1 signs RB29 using a threshold signature scheme, yielding a share of random beacon 30.

**Step 3**
Nodes 1 sees that node 2 also published a share of Random Beacon 30.

**Step 4**
2 random beacon shares are sufficient to reconstruct a full threshold signature, which is Random Beacon 30.
Block Maker Ranking

The Random Beacon is used to rank block makers

<table>
<thead>
<tr>
<th>Rank 0</th>
<th>Rank 1</th>
<th>Rank 2</th>
<th>Rank 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Node 1</td>
<td>Node 4</td>
<td>Node 2</td>
<td>Node 3</td>
</tr>
<tr>
<td>Node 3</td>
<td>Node 1</td>
<td>Node 4</td>
<td>Node 4</td>
</tr>
<tr>
<td>Node 4</td>
<td>Node 3</td>
<td>Node 4</td>
<td>Node 1</td>
</tr>
<tr>
<td>Node 3</td>
<td>Node 1</td>
<td>Node 3</td>
<td>Node 3</td>
</tr>
<tr>
<td>Node 2</td>
<td>Node 2</td>
<td>Node 1</td>
<td>Node 3</td>
</tr>
</tbody>
</table>

High Priority
Low Priority
Notarization with Block Maker Ranking

Rounds are divided into time slots defining when block maker proposals are considered.
Notarization with Block Maker Ranking

The block ranks can reduce the number of notarized blocks

**Step 1**
Node 1 receives a rank-1 block proposal for height 30, building on some notarized height 29 block

**Step 2**
Node 1 is still in time slot 0, not willing to notary-sign a rank-1 block yet

**Step 3**
Nodes 1 sees a valid rank-0 height 30 block, and it broadcasts it together with a notarization share

**Step 4**
Eventually, only the rank 0 block becomes notarized
Notarization with Block Maker Ranking

One notarized block $b$ at a height $h = \text{Agreement up to } h$

How can we detect this…?
Notarization with Block Maker Ranking

Synchronous communication → Forks can be removed
Notarization with Block Maker Ranking

Asynchronous communication → Forks cannot be removed!
Finalization

Nodes create finalization shares if they did not notary-sign any other block at that height

**Step 1**
Node 1 notary-signs block \( b \) at height 30

**Step 2**
Node 1 observes that block \( b \) is fully notarized and will no longer notary-sign blocks at height \( \leq 30 \)

**Step 3**
Since node 1 did not notary-sign any other block than block \( b \), it creates a finalization-share on \( b \)

**Step 4**
Nodes 2 and 4 also cast finalization shares on block \( b \)

**Step 5**
3 finalization-shares are sufficient approval: the shares are aggregated into a single full finalization

Node 1 did not notary-sign any height 30 block other than \( b \)
Finalization

Finalization on block $b$ at height $h = $ Proof that no other block is notarized at height $h$

The chain up to this block is final
Algorithm Summary

1. Block tree building with notarization threshold signatures
   => at least one block per round

2. Random beacon from BLS threshold signature chain to rank block makers
   => reduce message and bit complexity

3. Recursive tree pruning with finalization threshold signatures
   => exactly one block per round
If block $b$ at height $h$ is finalized, then there is no finalized block $b' \neq b$ at height $h$.

**Proof sketch:**

1. A full finalization on $b$ requires $n-f$ nodes to finality-sign (by construction)
2. At least $n-2f$ of the $n-f$ nodes that finality-signed $b$ must be honest
   (by assumption that $\leq f$ nodes are corrupt)
3. An honest node that finality-signed $b$ did not notary-sign any other block at height $h$ (by construction)
4. At least $n-2f$ nodes did not notary-sign any height $h$ block other than $b$ (by 2. & 3.)
5. A full notarization requires $n-f$ notarization-shares (by construction)
6. The $n-(n-2f) < n-f$ remaining nodes that may have notary-signed a block $b'$ are not sufficient to reach the notarization threshold of $n-f$ (by 4. & 5.)
Liveness

The communication network is $\delta$-synchronous at time $T$ if all messages sent by honest nodes by time $T$ are delivered by $T+\delta$

Assume that:
(i) $k > 1$, the first honest node $P$ to enter round $k$ does so at time $T$
(ii) Node $Q$ with rank 0 in round $k$ is honest;
(iii) the communication network is $\delta$-synchronous at times $T$ and $T+\delta$;
(iv) slot 0 lasts at least $2\delta$.

Then when all round-$k$ messages from honest nodes have been delivered to all honest nodes, each honest node will have $Q$’s round-$k$ proposed block as a finalized block.
Measurements

<table>
<thead>
<tr>
<th></th>
<th>without load</th>
<th>with load</th>
<th>with load and node failures</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>13 node subnet</strong></td>
<td>1.09 blocks/s</td>
<td>1.10 blocks/s</td>
<td>0.45 blocks/s</td>
</tr>
<tr>
<td></td>
<td>1.64 Mb/s</td>
<td>4.72 Mb/s</td>
<td>4.39 Mb/s</td>
</tr>
<tr>
<td><strong>40 node subnet</strong></td>
<td>0.41 blocks/s</td>
<td>0.41 blocks/s</td>
<td>0.16 blocks/s</td>
</tr>
<tr>
<td></td>
<td>4.63 Mb/s</td>
<td>7.32 Mb/s</td>
<td>5.06 Mb/s</td>
</tr>
</tbody>
</table>

Average block rate and sent traffic.
Wanna know more?

- Full version with proofs [link](#)
  - Includes protocol variants + analysis for message complexity, latency, ...

- Internet Computer Wiki [link](#)
- Technical Library: [here](#) (videos of talks) and [here](#) (blogposts)

Friday 10:30 - 11:15

DARE 2022: 2nd Workshop on Distributed Algorithms on Realistic Network Models

All editions →

July 29, 2022

Yvonne-Anne Pignolet
Catching up on the Internet Computer
Abstract →
Bio →
We are hiring!

www.dfinity.org/careers

yvonneanne@dfinity.org