The Internet Computer and its networks

Yotam, Yvonne-Anne, Rüdiger, April, 2022

We are hiring: dfinity.org/careers
Agenda

1) What is the IC?

2) What are its networking patterns and requirements?

3) Show me the numbers!

4) Q&A
What is the Internet Computer?
What is the Internet Computer?

Platform to run any computation, using blockchain technology for decentralisation and security.
Internet Computer Protocol (ICP)

Coordination of nodes in independent data centers, jointly performing any computation for anyone

- Create Internet Computer blockchains
- Guarantee safety and liveness of smart contract execution despite Byzantine participants
Canister Smart Contracts

- Data: Memory pages
- Code: WebAssembly bytecode
Developers and users interact directly with Canisters
Developers and users interact directly with Canisters
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
- …
State Machine Replication

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
ICP Layers

- Execution Environment
- Message Routing
- Consensus
- Networking

Deterministic computation

Message acquisition and ordering
Comparison with other Blockchain Systems

### Layer-1 Performance Comparison

<table>
<thead>
<tr>
<th></th>
<th>Ethereum</th>
<th>Cardano</th>
<th>Solana</th>
<th>Avalanche</th>
<th>Algorand</th>
<th>Internet Computer</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Transaction Speed</strong></td>
<td>15-20 TPS</td>
<td>2 TPS</td>
<td>2,000-3,000 TPS</td>
<td>4,500 TPS</td>
<td>20 TPS</td>
<td>11,500 TPS</td>
</tr>
<tr>
<td><strong>Transaction Finality</strong></td>
<td>14 minutes</td>
<td>10-60 minutes</td>
<td>21-46 seconds</td>
<td>2-3 seconds</td>
<td>4-5 seconds</td>
<td>1 second</td>
</tr>
<tr>
<td><strong>Scalability</strong></td>
<td>Not very scalable</td>
<td>Not very scalable</td>
<td>Not very scalable</td>
<td>Not very scalable</td>
<td>More scalability</td>
<td>Indefinite scalability</td>
</tr>
<tr>
<td><strong>Node Count</strong></td>
<td>6,000 nodes</td>
<td>3,173 nodes</td>
<td>1,603 nodes</td>
<td>1,243 nodes</td>
<td>1,997 nodes</td>
<td>443 nodes</td>
</tr>
<tr>
<td><strong>Storage Costs</strong></td>
<td>$73,000,000 / GB</td>
<td>Inadequate data storage</td>
<td>$1,000,000 / GB</td>
<td>$988,000 / GB</td>
<td>IPFS off-chain storage</td>
<td>$5 / GB</td>
</tr>
<tr>
<td><strong>Cloud Service Dependency</strong></td>
<td>70% of nodes run on AWS</td>
<td>Unclear how many are cloud</td>
<td>Most nodes run on cloud</td>
<td>Unclear how many are cloud</td>
<td>Most nodes run on cloud</td>
<td>Independent data centers</td>
</tr>
</tbody>
</table>

Fast Growing Ecosystem
Following a canister call
Following a canister call

Intra-Subnet P2P
Following a canister call

Xnet between Subnets
Requirements 1/2

• Bounded-time/eventual delivery despite Byzantine faults

Up to a certain maximum volume of valid artifacts that are not dropped by any honest node reaches all honest nodes in bounded time/eventually despite attacks (under certain network assumptions).

• Reserved resources for different components/peers

Memory/bandwidth/CPU guarantees for different components and peers

• Prioritization for different artifacts

Not all artifacts are equal, different priorities depending on attributes (e.g., type, size, round,...). Priorities change over time.
Requirements 2/2

- High efficiency
  High throughput is more important than low latency
  Avoid duplicates: don’t waste bandwidth downloading same artifact “too many times”

- DOS/SPAM resilience
  Bad participants cannot prevent progress.

- Low accessibility requirements for users
  Support browser and IPv4 access
Networking of the IC

- Geographically distributed: datacenters all over the world
Networking of the IC

- **Geographically distributed**: datacenters all over the world
- **Decentralized**: a subnet is composed of nodes in different datacenters
  - Some nodes in the same subnet may be very far apart
  - Independent node providers with different skills and DC contracts
  - Communication over public internet
    - High latencies possible
    - Many transient network failures
- **Secure**: a subnet should make progress even if up to \( \frac{1}{3} \) of the nodes are faulty / malicious
  - We can’t trust specific nodes (e.g., geographically close by)
  - Even nodes in the same subnet should not trust each other
Intra-Subnet P2P Networking

- Peer-to-peer network of nodes
  - Gossip protocol for artifact distribution
    - Advert - Request - Response
  - Eventual / bounded time delivery with priorities (~reliable broadcast optimized for Consensus)

- Untrusted communication
  - TLS / TCP to all nodes in the subnet, certificates in NNS
  - Authenticity and integrity of artifacts can be verified by higher layers
  - Nodes can still do evil
Xnet Inter-Subnet Networking

- Canisters on one subnet can send messages to canisters on other subnets, called “cross-net communication” (or Xnet)

- Currently this is done quite naively, where any node on one subnet can fetch messages from any other node on the other subnet with a HTTPS request

- We can probably improve this on several aspects:
  - Scalability: decide which nodes connect to which, and when
  - Performance: leverage the fact that some nodes in both subnets are close to each other (content is signed by the subnet, so we do not need to trust a specific node up to some extent)
Numbers...
The IC in Current Numbers

Application Layer:

- 60K+ canisters (smart contracts/dapps)
- > 2 Mio registered identities
- ~1TB total state (and counting...)

https://dashboard.internetcomputer.org/
The IC in Current Numbers

Application Layer:
- 60K+ canisters (smart contracts/dapps)
- > 2 Mio registered identities
- ~1TB total state (and counting…)

Governance:
- So far:
  - 56K+ NNS proposals
  - 3.4M+ ICP transactions

https://dashboard.internetcomputer.org/
The IC in Current Numbers

Consensus

- 758M+ blocks created
- ~34 blocks per second
- ~2800 transactions per second

https://dashboard.internetcomputer.org/
The IC in Current Numbers

Network Layer:

- 477 nodes
  - From 54 node providers
- 33 subnets
  - 40 nodes in NNS subnet
  - 13 nodes in App subnets
- Avg ⅔ dissemination latency:
  - NNS avg=1.39s, 95%=3.3s
  - App avg=0.57s, 95%=1.1s

https://dashboard.internetcomputer.org/
Example Subnet Dashboard
Research on the IC
Open Research Problems

- Intra-subnet communications scalability (growing size of subnets)
- Inter-subnet communications scalability (growing number of subnets)
- Ongoing firewall rule management
- Resilience against malicious activity
- Monitoring of node and network behavior
- Dynamic load balancing
- Caching
- Canister addressing
Testnets

DFINITY-internal infrastructure

- Deploy complete IC instances in our 5 data centers (2 more in May)
  - Chicago, San Francisco, Des Moines, Frankfurt, Zurich, ..
- Variable size and VM capabilities
- Can be used for experiments, metrics, correctness and performance tests
Metrics

We can collect two general types of metrics:

- **Code metrics**
  - On mainnet
  - On testnets
- **Infrastructure metrics**
Logging

- Events can be logged in the code
- Log can be fetched from testnet machines
- Policy monitoring with MonPoly from Prof. Basin’s group
Case Study: “Idle” vs. Workload Traffic

31 nodes deployment
- 13 in NNS
- 18 in Subnet 1

Workload generation
- only in Subnet 1
- 100 requests per sec
- 1 kb each

Conclusion
- ICP produces 0.1-0.2MBytes/s for the protocol to make progress.
Case Study: “Intra DC” == Internet

iperf between testnet hosts
- Chicago to San Francisco
- 60s in total

Conclusion
- Packet loss has a significant impact on the achieved throughput.
Case Study: RTT and Packet Loss

ping between testnet hosts
- Chicago to San Francisco
- every 1s for 10mins

Conclusion
- Extremely high packet loss (3.5%).
- Path change towards the end?
More information

- Infographic: here

- Technical Library: here (videos of talks) and here (blogposts)

- 200,000,000 CHF Developer Grant Program here

- DFINITY SDK: here