Modelling of the Internet Computer Protocol Architecture: the Next Generation Blockchain [⋆]

AoXuan Li[1], Luca Serena[2], Mirko Zichichi[3], Su-Kit Tang[1], Gabriele D'Angelo[2], and Stefano Ferretti[4]

[1] Faculty of Applied Sciences, Macao Polytechnic University, Macao SAR, China
aoxuan.li@mpu.edu.mo, sktang@ipm.edu.mo
[2] Department of Computer Science and Engineering, University of Bologna, Italy
luca.serena2@unibo.it, g.dangelo@unibo.it
[3] Ontology Engineering Group, Universidad Politécnica de Madrid, Spain
mirko.zichichi@upm.es
[4] Department of Pure and Applied Sciences, University of Urbino "Carlo Bo", Italy
stefano.ferretti@uniurb.it

Abstract. The Internet Computer Protocol is described as a third-generation blockchain system that aims to provide secure and scalable distributed systems through blockchains and smart contracts. In this position paper, this innovative architecture is introduced and then discussed in view of its modeling and simulation aspects. In fact, a properly defined digital twin of the Internet Computer Protocol could help its design, development, and evaluation in terms of performance and resilience to specific security attacks. To this extent, we propose a multi-level simulation model that follows an agent-based paradigm. The main issues of the modeling and simulation, and the main expected outcomes, are described and discussed.

Keywords: Internet Computer · Distributed Ledger Technology · Modelling and Simulation · Blockchain.

1 Introduction

Cloud computing has undoubtedly been the fastest growing and most successful in delivering technical and economic benefits for application and system development in recent years [30, 26]. Starting from startups up to large companies, everyone is adopting cloud computing to get rid of the risk of capital investment, cutting the cost of hardware and software infrastructure, and availing themselves of services according to their demand. This is why paradigms such as 'Infrastructure-as-a-Service (IaaS)', 'Platform-as-a-Service (PaaS)', and 'Software-as-a-Service (SaaS)' have emerged. In general, however, cloud service providers maintain their customers with an opaque knowledge about the location

[⋆] This work has received funding from the EU H2020 research and innovation programme under the MSCA ITN grant agreement No 814177 LAST-JD-RIoE; and the research grant (No.: RP/ESCA-04/2020) offered by Macao Polytechnic University.

and storage of data, the privacy offered to users, and the type of hardware infrastructure used. This leads firstly to a problem of trust by users [19]. Secondly, security and privacy are undermined by the centrality of these solutions, which more easily attracts cyber-attacks, i.e. single points of failure [30]. In addition, it should not be forgotten that centralized solutions will not be able to support the huge amount of data generated globally by users and Internet-of-Things devices for much longer [26]. Finally, it is commonly difficult to assess if Quality of Service (QoS) guarantees are met and Service Level Agreements (SLA) negotiated between users and the cloud provider are satisfied, due to the absence of trusted logs [5]. All this motivates the transition towards a completely decentralized approach. The benefits of this solution are many. In fact, the decentralization of the system removes the presence of a single point of failure, allows for inherently increasing scalability, curbs illicit activities of malicious nodes, and can also provide for accountability guarantees. Clearly, in order to realize a similar kind of system, it becomes necessary to encourage node participation that can be somehow rewarded through incentive mechanisms [33].

The Internet Computer Protocol (ICP) architecture[5] aims to establish a network of networks by defining a protocol for combining the resources of several decentralized computers into the reading, replication, modification, and procurement of an application state. A network of nodes runs the protocol through independently-operated data centers to provide general-purpose (largely) transparent computations for end-users. On the other hand, the development of applications on top of the ICP is facilitated by reliable message delivery, transparent accountability, and resilience. The typical use-case would involve users interacting with a decentralized application as is on a public or private cloud. This is enabled by the use of Canisters, i.e. tamper-proof and autonomous smart contracts hosted on-chain, that can be run concurrently and interact with each other. With respect to other smart contract implementations, such as Ethereum's ones, the Canisters enable applications, systems, and services to be created and accessed by users without incorporating websites running on centralized cloud hosting, e.g. a canister can directly serve HTTP requests created by end-users through their browser. All of this paves the way for the creation of decentralized services where the user is constantly at the center of the process.

However, the design of the ICP requires a complete understanding of the technologies involved and the interactions among these building blocks. There is a need for viable modeling and simulation strategies that allow for what-if analyses and manageable evaluation studies. In this paper, we describe the rationale behind the design of an ICP digital twin that could serve this purpose. Due to the complexity of the system, high levels of detail should be kept only when needed, while coarse simulations should be exploited when dealing with a high number of involved nodes. This leads to a multi-level simulator design [13].

This paper is structured as follows. Section 2 provides the necessary background about the technologies used in the ICP and a discussion of the related

[5] Authors are not sponsored or affiliated in any way with the DFINITY Foundation which is the not-for-profit organization that develops the Internet Computer.

work. Section 3 presents a specific introduction of the ICP; while in Section 4, the main modelling and simulation issues of the ICP are discussed. Finally, Section 5 provides the concluding remarks.

2 Background and Related Work

2.1 Related Technologies

In this section, we briefly describe the background technologies and methodologies that are necessary for understanding the ICP architecture and evaluating the main problems that are related to its modelling and simulation.

Blockchains Informally, a blockchain is a public ledger that may hold any data, e.g., transactions between different parties, email records, or even daily grocery records. The ledger is distributed among all network participants, and it is immutable once written down. As the name suggests, a blockchain is a chain of blocks while each block contains a set of records. Moreover, a block also contains a timestamp and the hash value of the previous block. If any adversary user tries to change intermediate blocks, he/she has to change all following blocks. However, this is impossible since the ledger is decentralized. For any new block, it will not automatically join the chain until the majority of parties agree so. Blockchains have made impacts on various areas [20].

Consensus Algorithms Consensus algorithms allow (the majority of) nodes to agree on the status of the ledger. That is, they agree on the validity of transactions in a block, the validity of the block itself, and if there is more than one proposed block, on which block is appended to the chain. There are different types of consensus algorithms. Among them, two are worthy of mention here, i.e. proof-based algorithm and vote-based algorithm [24]. In a proof-based algorithm, parties need to solve a cryptography puzzle, and the first successful one gets the right to append the block. In a vote-based algorithm, if a party wants to append a block, there must be more than T parties appending the same block where T is a threshold number.

Smart contracts Smart contracts are a set of instructions (or the source code from which such instructions were compiled from) stored in the blockchain and automatically triggered once the default condition is met [2]. This execution is triggered via a transaction and will produce a change in the blockchain state. Each node executing the instructions receives the same inputs and produces the same outputs, thanks to a shared protocol. Smart contracts enable the execution of a service without a trusted human third party validator to check the terms of an agreement, however the smart contract issuer must be sure that the behaviour implemented is correct [2]. For instance, the creation of smart contract-based services may enable users to interact with devices/vehicles or favor interoperability in smart cities [14, 32].

2.2 Related Work

While there are no specific simulations of the ICP architecture, some simulators have been proposed for modelling blockchains and distributed ledgers technologies. The main difference between the simulators is the level of detail (and the corresponding simulation methodology) that has been chosen for modelling the system to be represented. For example, in [27], some of the authors of this paper have proposed an agent-based simulation to investigate some well-known network attacks on blockchains and distributed ledgers. In [1], the authors present BlockSim which is a discrete-event simulator of blockchain systems (implemented in Python) that specifically considers the modeling and simulation of block creation through the proof-of-work consensus algorithm. A totally different approach is introduced in [22], in which the queuing theory is used for modelling blockchain systems. In [29], the authors propose VIBES which is another blockchain simulator but specifically designed and implemented for large-scale peer-to-peer networks and able to simulate blockchain systems beyond Bitcoin and support large-scale simulations with thousands of nodes. Finally, in [25], the authors propose an approach that is based on stochastic blockchain models (i.e. Monte Carlo simulations).

3 The Internet Computer Protocol (ICP)

The ICP is defined as the third generation of the blockchain systems [31], where the first generation is Bitcoin [23], and the second generation is Ethereum [4]. The ICP provides an infinite blockchain where we may hold everything. Unlike previous blockchain systems, it aims to be scalable and to run at web speed. The main technical components of the Internet Computer are the Canister [6] and the Network Nervous System (NNS) [7]. The canister is a special type of smart contract. Users may interact with a canister directly as long as they know the identity of the canister. In the ICP, communication between the different nodes is demanded to the Network Nervous System (NNS, see Section 3.1).

The ICP has a four-layer structure. From bottom to top, there are data centers, nodes, subnets, and canisters. Data centers are hardware devices for holding nodes, and each node is a physical computer providing computational power . Each data center may have many nodes, and nodes from different data centers could build up a subnet. Each subnet hosts many canisters, which is the application program on the ICP. Figure 1 reports a high-level representation of this design structure. Each subnet handles the trust and immutability of the Canister with a blockchain. The blockchain grows in rounds, and, in each round, a randomly selected node proposes a block containing the canister inputs and the hash of the previous block. If the majority of nodes agree on the subnet's state and the validity of the new block, this new block is appended to the blockchain.

The ICP design guarantees the availability of canisters in subnets. In fact, by implementing a replication approach, the canisters do not suddenly stop running in case of localised failures. As long as more than two-thirds of replicas are online, the canister is available. A critical requirement for this approach is that

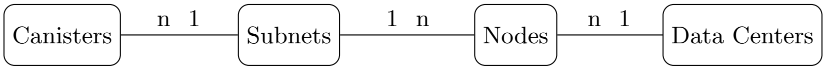

Fig. 1. The ICP high-level design architecture.

all replicas must catch up with the latest state. In previous blockchains, like Bitcoin and Ethereum, this would require downloading the whole blockchain. The ICP provides a CatchUp Package (CUP) [8] so that a node only needs to download a limited amount of data to catch up with the current state of the blockchain. The CUP contains an intermediate replica state and a subsequent of the blockchain. With the replica state, the node can compute the next state on itself, and with the sub-blockchain, it can verify the replica states.

The ICP community claims that their blockchain could scale out to billions of users [9]. Since each canister can only support up to 4 GB of memory (i.e. due to the limitations of WebAssembly) then the Internet Computer uses a multi-canister architecture. For example, for a video-sharing application, it would be possible to split the user-uploaded content into multiple chunks and store them into multiple canisters. When a user wants to retrieve a video, the user makes a query call to the front-end canister, which in turn will make cross-canisters requests to multiple storage canisters. It is worth noticing that all these operations are transparent to users. Table 1 summarizes the main terms used to describe the ICP architecture.

3.1 Network Nervous System

To obtain a scalable and highly efficient system, the ICP must be able to host any number of canisters and to run them concurrently. The ICP introduces a novel Decentralized Autonomous Organization (DAO) that is called Network Nervous System (NNS). The NNS is designed for managing all the base nodes of the system through a Proof-of-Stake consensus protocol.

More specifically, NNS is a set of initial canister programs that oversee the whole network. For example, a data center may apply to the NNS to join the network. NNS also manages how the subnets are formed and how the replication of the nodes is managed. Moreover, the NNS is in charge of upgrading the ICP. For example, the users are enabled to submit proposals for changing the ICP

Term	Definition
Canister	A special type of smart contract.
Catch Up Packages (CUP)	A schema for state synchronization.
Data Center	The decentralized hardware of the ICP architecture.
Network Nervous System (NNS)	A special canister serving as the governance body.
Node	The peer computer in data centers.
Subnet	The blockchain for providing computing resources.

Table 1. Main terms used in the ICP architecture.

design and implementation. NNS will host the proposal and then allow users to vote on the proposal. Finally, the NNS will implement and deploy the proposal, if the majority of the users have approved it.

3.2 Chain Key Cryptography

Most likely, the main scientific breakthrough provided by the ICP is the Chain Key Cryptography [16, 10]. In the ICP, a canister is replicated through a subnet, and those nodes in the subnet have to agree on the computation results. The high-level process is described below:

1. each node holds a secret key share;
2. if enough nodes agree on the result then they can jointly sign the message with their respective key share;
3. the user may verify the received message with a single public-key.

If some nodes have failed or crashed then the NNS will add new nodes to the subnet, and the remaining active nodes will reshare the secret key while keeping the same public key. In the ICP, all subnets have a public key and corresponding secret key shares, and all those public keys could be verified with a single 48-byte public key. Even if the Internet Computer had millions of nodes, the network would only need one public key to verify all messages. This technology is called Chain Key Cryptography[10]. The used protocol builds on Shamir secret sharing [28] and BLS signature [3], and moreover, it facilitates the secret sharing keys creation and refreshing.

4 Modelling and Simulation of the ICP architecture

The aim of the Modelling and Simulation (M&S) techniques is to reproduce the behaviour of the system under investigation, in order to (i) study the dynamics of interaction among the various components, (ii) evaluate the resilience of the system under specific conditions (e.g. cyberattacks or failures) and (iii) assess the impact of possible future extensions or features to the system before its implementation or even to support their design. Specifically for the ICP architecture, different aspects are of interest under a M&S viewpoint. First of all, it is important to model the consensus protocol, in order to analyze how the block creation flow is working.

While state-of-the art works regarding the simulation of Distributed Ledger Technology (DLT) focus on Bitcoin-like protocols, with only one blockchain collecting the incoming data [27], we are interested in modeling a DLT where multiple partial blockchains (i.e. subnets) work asynchronously in parallel, exchanging information when necessary. Furthermore, we think that, what is needed to model, are also the aspects specifically related to the DAO (i.e. the Decentralized Autonomous Organization [18]), that is in charge of managing the policies and the future developments of the DLT. This is because the level of security, scalability, decentralization and also the economical sustainability of the whole architecture strongly depends on the DAO's decisions.

A common problem in M&S is the level of detail to be chosen when abstracting the real system to a simulation model. In fact, a fine grained model that considers the very detailed aspects (such as the transmission of every network packet in the distributed system of interest) would permit a high-level model accuracy at the cost of a relevant complexity in the management of the obtained model and in the execution overhead of the simulator. On the other hand, a high-level model in which, for example, the communication aspects in the distributed system are neglected, would permit the building of a very simple (and fast) simulator with poor accuracy. Choosing the most appropriate level of detail to be used in the M&S is not easy and it is strongly linked to the desired outcomes of the simulator. For example, the modelling of some specific cyber-security attacks in a distributed system often requires a very specific level of abstraction in the communications. For these reasons, we think that in a scenario such as the ICP architecture in which we are interested in many different aspects of different abstraction layers, a solution based on a fixed level of abstraction would not be optimal. In fact, we plan to employ an approach that is based on multi-level modelling and simulation [17]. This approach is not new but it is still not very common in the simulation of complex systems. More in detail, we plan to build an ICP model in which the different components are represented by two (or more) simulated models that will be alternatively used depending on the specific analysis that we are interested in. For example, when the specific aspects related to the DAO will be investigated, some low-level details of the model will not be required and therefore the "high-level" (i.e. coarse-grained) version of some components will be used. On the other hand, when the security of the consensus protocol will be investigated then the "fine-grained" models will be required.

4.1 Design and implementation of an ICP simulator

In order to model and simulate the ICP architecture, we decided to employ an agent-based approach. Agent-Based Simulation (ABS) is a widely diffused technique that in the years gained a lot of popularity in many different fields such as engineering, economics, and computational social sciences [21]. In ABS, the most relevant system components and modules are represented by means of agents. Every agent is then characterized by a specific behavior and interacts with other agents using interactions (that are often implemented as messages). In other words, the system evolution is represented through changes in the local state of the agents (and of the environment) in which they are located.

Referring more specifically to the modeling of the ICP architecture, two types of agents populate our simulation scenario: firstly, there are the clients of the system, which can carry out transactions and requests to the system. Secondly, there are the nodes of the ICP, each one localized in a specific data center, and operating in a specific subnet. Figure 2 shows a possible modelling of the ICP nodes. All the nodes are located in a certain datacenter, belong to a specific

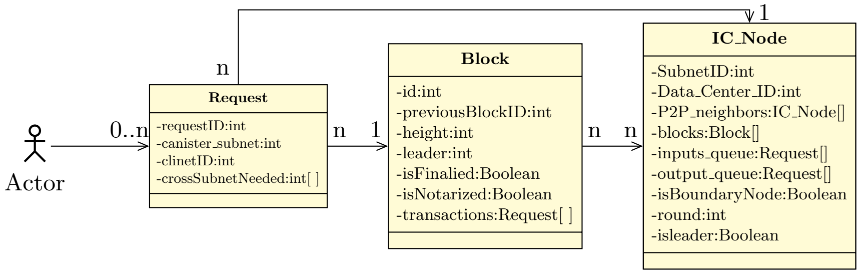

Fig. 2. Class diagram of the main components of the ICP architecture.

subnet and maintain a set of blocks as well as a set of transactions still to validate.

The current implementation of the ICP architecture relies on a very low number of nodes, since 32 subnets exist, each one with 13 nodes contributing to store the transactions (except the NNS, which is dealt with as a special subnet, composed of 40 nodes) [11]. Thus, for the modelling and simulation of the current setup of the ICP architecture, the simulator's scalability is not a big concern. However, it is expected and already planned that the future developments of the ICP will lead to a considerable growth of the network size, with many more nodes and subnets involved in the validation of transactions. We plan to use the developed simulation tools to be able to investigate and properly assess how such a network growth should be managed. For example, right now it is easy for the nodes to be directly in contact with all the other peers belonging to the same subnet, but with many more nodes managing a single subnet, a gossip algorithm might be adopted to efficiently disseminate blocks and transactions inside each subnet [12]. Moreover, from a simulation point of view, more simulated entities entail a larger amount of computing resources employed and a greater execution time. Thus, Parallel And Distributed Simulation (PADS) [15] approaches might be necessary to efficiently carry out the tests.

5 Conclusions

The Internet Computer Protocol (ICP) architecture is a third generation blockchain system that is being designed, implemented, and deployed to provide a secure and a scalable way for creating very large-scale distributed systems. In this position paper, we have introduced the ICP architecture and its main problems in terms of modelling and simulation. In fact, the usage of proper simulation techniques would permit us to investigate some very relevant aspects of the ICP architecture and support its design. The main issues related to the modelling and simulation of the ICP concern the specific level of detail used for abstracting the system in a model that can be then evaluated using a simulation. In the following of this paper, we described our current effort in the creation of an agent-based simulator of the ICP that is able to both provide the desired level of detail and the needed scalability. The creation of the ICP simulator is an ongoing activity that requires a relevant effort in many different phases (e.g., design, implementation, and validation) that will likely permit us to release a preliminary version of the simulator in the next months.

References

1. Alharby, M., van Moorsel, A.: Blocksim: A simulation framework for blockchain systems. SIGMETRICS Perform. Eval. Rev. **46**(3), 135–138 (Jan 2019)
2. Becker, M., Bodó, B.: Trust in blockchain-based systems. Internet Policy Review **10**(2) (2021)
3. Boneh, D., Lynn, B., Shacham, H.: Short signatures from the weil pairing. In: Int. conf. on the theory and application of cryptology and information security. pp. 514–532. Springer (2001)
4. Buterin, V., et al.: A next-generation smart contract and decentralized application platform. white paper **3**(37) (2014)
5. D'Angelo, G., Ferretti, S., Marzolla, M.: A blockchain-based flight data recorder for cloud accountability. In: Proc. of the 1st Workshop on Cryptocurrencies and Blockchains for Distributed Systems (CryBlock) (2018)
6. Dfinity: A closer look at software canisters, an evolution of smart contracts (Sep 2021), https://medium.com/dfinity/software-canisters-an-evolution-of-smart-contracts-internet-computer-f1f92f1bfffb
7. Dfinity: The network nervous system: Governing the internet computer (Sep 2021), https://medium.com/dfinity/the-network-nervous-system-governing-the-internet-computer-1d176605d66a
8. Dfinity: Resumption: How internet computer nodes quickly catch up to the blockchain's latest state (Oct 2021), https://medium.com/dfinity/resumption-how-internet-computer-nodes-quickly-catch-up-to-the-blockchains-latest-state-5af6e53e2a7
9. Dfinity: A technical overview of the internet computer (Sep 2021), https://medium.com/dfinity/a-technical-overview-of-the-internet-computer-f57c62abc20f
10. Dfinity: Chain key cryptography: The scientific breakthrough behind the internet computer (Jan 2022), https://medium.com/dfinity/chain-key-technology-one-public-key-for-the-internet-computer-6a3644901e28
11. Dfinity: Internet computer network status (June 2022), https://dashboard.internetcomputer.org
12. D'Angelo, G., Ferretti, S.: Highly intensive data dissemination in complex networks. Journal of Parallel and Distributed Computing **99**, 28–50 (2017)
13. D'Angelo, G., Ferretti, S., Ghini, V.: Multi-level simulation of internet of things on smart territories. Simulation Modelling Practice and Theory **73**, 3–21 (2017), smart Cities and Internet of Things
14. Ferraro, P., King, C., Shorten, R.: Distributed ledger technology for smart cities, the sharing economy, and social compliance. IEEE Access **6**, 62728–62746 (2018)
15. Fujimoto, R.M.: Parallel and Distribution Simulation Systems. John Wiley and Sons, Inc., USA, 1st edn. (1999)
16. Groth, J.: Non-interactive distributed key generation and key resharing. Cryptology ePrint Archive, Report 2021/339 (2021), https://ia.cr/2021/339

17. Huraux, T., Sabouret, N., Haradji, Y.: A multi-level model for multi-agent based simulation. In: ICAART (2). pp. 139–146 (2014)
18. Jentzsch, C.: Decentralized autonomous organization to automate governance. White paper, November (2016)
19. Khan, K.M., Malluhi, Q.: Establishing trust in cloud computing. IT professional **12**(5), 20–27 (2010)
20. Lei, I.S., Tang, S.K., Tse, R.: Integrating consortium blockchain into edge server to defense against ransomware attack. Procedia Computer Science **177**, 120–127 (2020)
21. Macal, C., North, M.: Introductory tutorial: Agent-based modeling and simulation. In: Proceedings of the Winter Simulation Conference 2014. pp. 6–20 (2014)
22. Memon, R.A., Li, J.P., Ahmed, J.: Simulation model for blockchain systems using queuing theory. Electronics **8**(2) (2019)
23. Nakamoto, S.: Bitcoin: A peer-to-peer electronic cash system. Decentralized Business Review p. 21260 (2008)
24. Nguyen, G.T., Kim, K.: A survey about consensus algorithms used in blockchain. Journal of Information processing systems **14**(1), 101–128 (2018)
25. Piriou, P.Y., Dumas, J.F.: Simulation of stochastic blockchain models. In: 2018 14th European Dependable Computing Conference (EDCC). pp. 150–157 (2018)
26. Sadeeq, M.M., Abdulkareem, N.M., Zeebaree, S.R., Ahmed, D.M., Sami, A.S., Zebari, R.R.: Iot and cloud computing issues, challenges and opportunities: A review. Qubahan Academic Journal **1**(2), 1–7 (2021)
27. Serena, L., D'Angelo, G., Ferretti, S.: Security analysis of distributed ledgers and blockchains through agent-based simulation. Simulation Modelling Practice and Theory **114**, 102413 (2022)
28. Shamir, A.: How to share a secret. Communications of the ACM **22**(11), 612–613 (1979)
29. Stoykov, L., Zhang, K., Jacobsen, H.A.: Vibes: Fast blockchain simulations for large-scale peer-to-peer networks: Demo. In: Proc. of the 18th Middleware Conference: Posters and Demos. p. 19–20. Association for Computing Machinery (2017)
30. Subramanian, N., Jeyaraj, A.: Recent security challenges in cloud computing. Computers & Electrical Engineering **71**, 28–42 (2018)
31. Team, D., et al.: The internet computer for geeks. Cryptology ePrint Archive (2022)
32. Zichichi, M., Ferretti, S., D'Angelo, G.: A framework based on distributed ledger technologies for data management and services in intelligent transportation systems. IEEE Access pp. 100384–100402 (2020)
33. Zichichi, M., Ferretti, S., D'angelo, G.: On the efficiency of decentralized file storage for personal information management systems. 2020 IEEE Symposium on Computers and Communications (ISCC) pp. 1–6 (2020)