A Time-Segmented Consortium Blockchain for Robotic Event Registration

Vasco Lopes

NOVA LINCS, Universidade da Beira Interior, vasco.lopes@ubi.pt

Miguel Fernandes

INSTITUTO ITALIANO DE TECNOLOGIA, Italy, killazfern@gmail.com

Nuno Pereira

NOVA LINCS, Universidade da Beira Interior, nuno.pereira@ubi.pt

Luís A. Alexandre

NOVA LINCS, Universidade da Beira Interior, luis.alexandre@ubi.pt

A blockchain, during its lifetime, records large amounts of data. In a robotics environment, the old information is useful for human evaluation, or to perform analysis, but it is not useful for robots that require only current information to continue their work. This causes a storage problem in Blockchain nodes as in the case of nodes attached to robots that are usually built around embedded solutions. This paper presents a time-segmentation solution for devices with limited storage capacity, integrated into a particular robot-directed Blockchain called RobotChain. The experiments conducted show that the goal of restricting each node's capacity is reached without compromising all the benefits that arise from the use of Blockchains in these contexts and it allows for cheap nodes to use this Blockchain, reducing storage costs and allowing faster deployment of new nodes.

CCS CONCEPTS • Blockchain • Time-Segmented • RobotChain

ACM Reference Format:

First Author's Name, Initials, and Last Name, Second Author's Name, Initials, and Last Name, and Third Author's Name, Initials, and Last Name. 2018. The Title of the Paper: ACM Conference Proceedings Manuscript Submission Template: This is the subtitle of the paper, this document both explains and embodies the submission format for authors using Word. In Woodstock '18: ACM Symposium on Neural Gaze Detection, June 03–05, 2018, Woodstock, NY. ACM, New York, NY, USA, 10 pages. NOTE: This block will be automatically generated when manuscripts are processed after acceptance.

1 INTRODUCTION

Blockchain has gained immense visibility and growth in recent years, due to its capability of enabling digital transactions to take place without the limitations of fiat currency. Blockchain allows transactions to take place without the need for a central authority, with total transparency, allowing transactions to be audited if the Blockchain is public and gives anonymity to the people involved in the transaction. But, the properties that Blockchain has, don't only give power to the users involved, but also provide decentralization and immutability of the data, and enforce that every transaction is non-repudiable by imposing that all parties must sign their data before sending it to the Blockchain. The advent of Blockchain is not only due to the aforementioned properties. The first major contribution of this technology was a way to trust a network that may have unknown and untrustworthy nodes - the consensus algorithm. The consensus algorithm first introduced by Bitcoin was proof-of-work, where all the nodes of the network must solve a cryptographic puzzle in order to validate a transaction and earn a reward for validating it. The second major contribution was later presented with further development, which are the smart-contracts. Smart-contracts are pieces of software that run autonomously inside the Blockchain and as transactions, they are immutable, meaning that they will always perform the same way and that they can't be altered or tampered with. The innovation and utility of Blockchain made it a technology that is disrupting multiple sectors ranging from financial, to healthcare and robotics [1], [2].

In short, Blockchain technology enables the creation of an immutable electronic ledger of information in a decentralized way, where every transaction is replicated throughout the network. But, even though this technology presents useful properties, the amount of information stored inside it presents a challenge, since it continues to grow in time and is never deleted or updated meaning that systems that rely on low-capacity devices, such as many robot applications, can't use Blockchain on a long term, since the information to store will eventually surpass the capability of such devices and negatively impact the overall performance and scalability of the network [3]. Many methods have been proposed to solve the problem of eventually reaching a unfeasible Blockchain size by either pruning the blocks or by implementing off-chain protocols to reduce the amount of information that is inserted in the main Blockchain by only sending it important information [4]. Those approaches are good short to mid-term solutions, as they can reduce the amount of information inserted into the Blockchain and give the capability for small devices to be part of a Blockchain network. However, these approaches don't solve the long-term problem of Blockchain's growing size.

In this paper, we propose a novel way to reduce Blockchain sizes - a time-segmented Blockchain approach that can be built over a normal Blockchain or over the aforementioned approaches to solve the presented issues. The proposed method was designed and evaluated in RobotChain [5], which is a Consortium Blockchain designed for Robots, based in the public Blockchain Tezos. The proposed method provides a way to segment a Blockchain into segments over time, where each segment of the network is connected to the previous segment by a cryptographic hash [6]. This way, the Blockchain maintains its integrity through its life span. In the new time-segmented Blockchain, a node can be configured to be either a compute device node, or a cold storage node, where a compute device node will only have the current segment, and the cold storage node will work with all segments belonging to the Blockchain. This idea allows small devices to participate in the Blockchain and also improves the performance, lowers storage costs, and supports faster new node deployment.

2 RELATED WORK

Although only recently Blockchain has started to be integrated with Robotics, due to its associated problems such as latency in validation and continuous growing size, there are promising proposals that conduct this integration. Ferrer [7] presents how Blockchain technology can be used as a mean to improve robot swarms and how it can solve occurring problems in such networks, such as data confidentiality, distributed decision making and dynamic environment working capacity without master control program modification.

In [8] it is shown a conceptualization of how it is possible to share critical information between robots using a Blockchain. In this, the idea is that robots insert information about their events and actions in the Blockchain about Human-Robot interactions and with that, other Robots improve their models. However, byzantine agents in the form of robots can use robotic Blockchains in order to propagate bad information or lead others to erroneous actions. Strobel et al. [9] propose a method to solve this problem in swarm robotics that use smart-contracts that forces robots to vote in order achieve consensus. Work has been conducted to solve the problem of byzantine robots by using a reputation system [10]. With this approach, consensus can be achieved, and byzantine robots mitigated. Blockchain has also been successfully used to create coalitions of robots by sharing information between them [11].

The Blockchain in which we based our work - RobotChain [5], has also been the basis for multiple proposals that use it in conjunction with robots to enhance their capabilities. In [12], the authors show how it is possible to use Blockchain to safely store robot logs and then use smart-contracts to autonomously detect robot anomalies. In [13], is shown how RobotChain can be used to control robots and use smart-contracts to allow external parties to communicate with it in order to provide analytics. Last, in [14], a novel way of monitoring robot workspaces is proposed, that uses RobotChain to store information about the robots and sensors and then uses external parties to conduct image analysis and smart-contracts to adjust the robot's behaviours depending on the identity and location of the people surrounding the robots.

Regarding the Blockchain scalability problem, off-chain methods have been extensively proposed. The most effective ones are based on storing information outside the main Blockchain, either by having traditional data-bases or secondary chains [15], or by improving the consensus algorithm to reduce the space it requires in terms of security protocols and validation time [16]. But by far, the most prominent one, as it is being implemented in the biggest public Blockchains, is Lightning Network [17]. The Lightning Network is a payment protocol that operates on top of a Blockchain-based cryptocurrency, such as Bitcoin or Ethereum. The idea of this protocol is to enable fast transactions between participating nodes. However, this proposals differs from the one we present in this paper because: 1) we do not change the consensus algorithm; 2) we do not require off-chain protocols and 3) the proposed method can be used on different Blockchains, as it is not dependent on a Blockchain architecture or protocols but rather on configuring them to allow segmentation.

3 THE UNDERLYING BLOCKCHAIN

3.1 Tezos Blockchain

Tezos is a self-amending crypto-ledger implemented in *Ocaml* [6]. Instead of using a traditional genesis block, this Blockchain starts with a genesis protocol, which contains a genesis block, but this block contains functions that allow the amending of the protocol such that it can evolve. This is the base for the main feature of this blockchain, which is the fact that it implements a protocol that can adapt itself without the need for a hard fork. These amendments work over cycles, and are suggested by submitting proposals to the chain, where stakeholders vote on these amendments. These amendments are considered an extremely positive point due to the fact that this allows the community to enact changes in the Blockchain, in order to improve it, preventing Blockchain hard forks, meaning that the inner features, such as the consensus algorithm, can be changed without the need to create a new and separate chain.

3.2 RobotChain

RobotChain contemplates the use of Blockchain technology in order to solve the problem of keeping accurate immutable records of robotic actions in a factory environment. A public Blockchain is not desired, since factory environments are private and, as such, management does not allow outside access to its internal information. So RobotChain is designed as a Consortium Blockchain: it has many of the advantages of a private Blockchain, but instead of a single entity being the leader, it operates under the leadership of a group, to allow for trust to be developed among the factory owners and the equipment providers. We need that all the robot manufacturers and the factory management, trust the event records, in the case that there is any kind of accident, there is no way to tamper the registers. The event records, that are stored in the Blockchain, can be used for further goals such as understanding and improving manufacturing productivity.

Figure 1 presents RobotChain in a schematic way, where each robot is connected to a computation module. This is a two-way connection in order to allow robots to send their events and logs to the Blockchain and to allow smart-contracts to change robot's behaviours. The use of computational modules serves two purposes: 1) to ensure a uniform input into the Blockchain, 2) to ensure that robots are not negatively affected with additional software running, which could cause degraded performance or other unforeseen consequences. In addition, there can be query nodes connected to the Blockchain in order to query it for information or, if allowed, insert analytics into smart-contracts. This architecture makes it possible to detect production line bottlenecks, to improve management understanding of the factory without directly interfacing with the robotic units, but more important, is the fact that RobotChain does not impact in any form the performance of the robots.

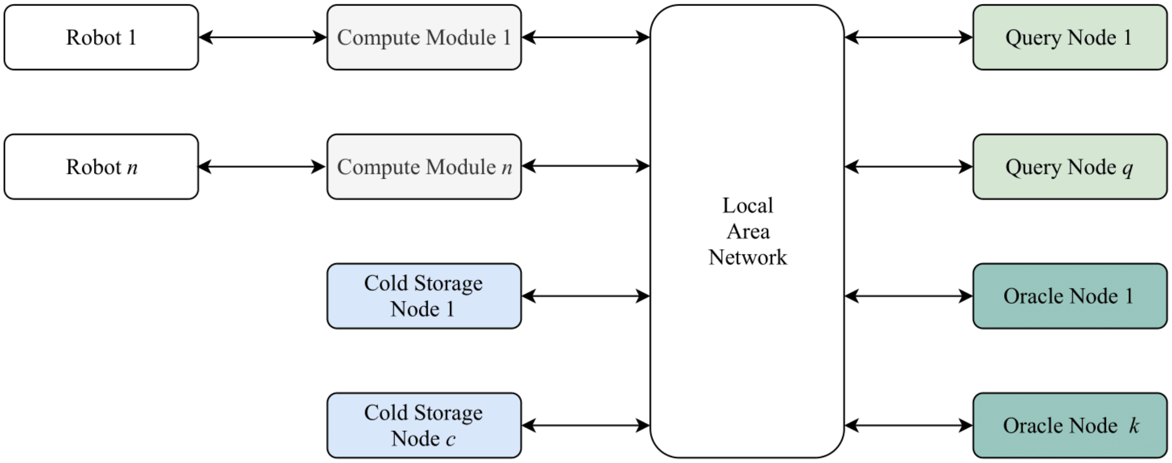

Figure 1: RobotChain overview. The compute modules devices serve as interface between robots and RobotChain. Cold storage nodes save all the segments of the Blockchain. Query nodes allow queries to the Blockchain and Oracles are external entities that interact with the Blockchain through smart-contracts.

Due to the fact that our proposal uses compute devices instead of running its code directly on the robot, the embedded compute devices are limited regarding data storage. As such, it is unfeasible to maintain a copy of the entirety of the Blockchain on each device in the long term. This paper deals with the fact that, although these records are important for the managers of the factories, they are not important for the day-to-day processing of the robot, since the logs of what a robot did months ago are not important to its current functioning. So, this paper improves upon the original proposal of RobotChain [5], with the introduction of the time-segmentation proposal.

3.3 Tezos History Mode

As of February 2019, Tezos launched a new feature [18] - the history mode, that contains similarities with the proposed solution in this paper. This history mode changes how a node keeps its past data, with three different modes. These modes rely on the checkpoint feature present in Tezos, where the checkpoints act as a regular interval anchor of consensus. The three presented modes are: the archive mode, where the node keeps every record, which corresponds to the current Tezos default working mode; the full mode, where the node stores all data from the beginning of the chain, but drops information from previous checkpoints but keeping the headers and operations from the checkpoints; and the rolling mode, where nodes only keep the latest checkpoint, effectively deleting old information. These modes are a configuration parameter of a node in the Tezos Blockchain.

Initialization of the nodes, regardless of the activated mode, is still based on the regular Blockchain synchronization method via peer-to-peer or with the new snapshot feature which consists on a file import/export.

4 TIME-SEGMENTATION IMPLEMENTATION

4.1 Overview

Our proposal of time-segmentation of a Blockchain consists in creating linked sub-Blockchains, referred to as segments throughout the paper, allowing compute devices with low storage capacity, the possibility to keep only the latest segment instead of the entire Blockchain, while maintaining the non-modification of the chain itself. Non-modification is ensured via the first block on the new segment, in our case, the RobotChain protocol activation, containing the segment identifier (an integer) and the hash of the last block of the previous segment. A second block is also created at the same time, in order to re-insert the smart-contracts present on the previous segment.

With this approach, three types of nodes are now introduced: the genesis node, the cold storage nodes and the compute device nodes. Genesis nodes are meant to serve as a bootstrap point, protocol activation and smart-contract initialization and cold storage of the previous segments. The cold storage type is meant to only store all the segments, to retrieve older segments when needed and to aid the bootstrap process. The compute devices nodes are the interface between the Blockchain and the robots. This solution is proposed to solve the storage limitation of the compute devices and the fact that the older blocks are not entirely relevant to the continuous processing of the robot in a factory. This allows compute devices with possibly small storage capacity to support a Blockchain solution for an arbitrarily long time period. Note that the number of cold storage nodes is typically much smaller than the number of compute device nodes and that this proposal maintains a chained link to the genesis block, which continues to allow amendment processes and that the cryptographic hash to the predecessor segment re-forces the security and the immutability of the Blockchain.

In the new developed Blockchain, the modifications are enacted on the Tezos Blockchain version with the history mode and all the three types of nodes (cold storage node, genesis node and compute device node) are run in the archive mode that was presented earlier. But, as the modifications made are protocol agnostic, the Blockchain can be updated to future newer versions of Tezos or even downgraded and still work as defined.

4.2 Segment Creation Process

The first segment, segment 1, starts the network as a regular Blockchain, with the activation block receiving as parameters the segment ID, and the original genesis block hash. Then, on its $n - th$ block, the Blockchain increments the segment ID on the node's configuration file and shuts down the validation and database part of the Blockchain but leaving the peer-to-peer interface online. Such that there is no need to rediscover peers.

The state and validation are then reactivated with the updated configuration file, creating a new segment from scratch. The genesis node then activates the protocol, receiving as parameter the current segment ID, the predecessor segment hash and the hash from the last block of the previous segment, with the other nodes receiving this activation and resuming normal operations. The first block is then used to initialize smart-contracts present on the previous segment and other features needed. Figure 2 presents this process in a visual way.

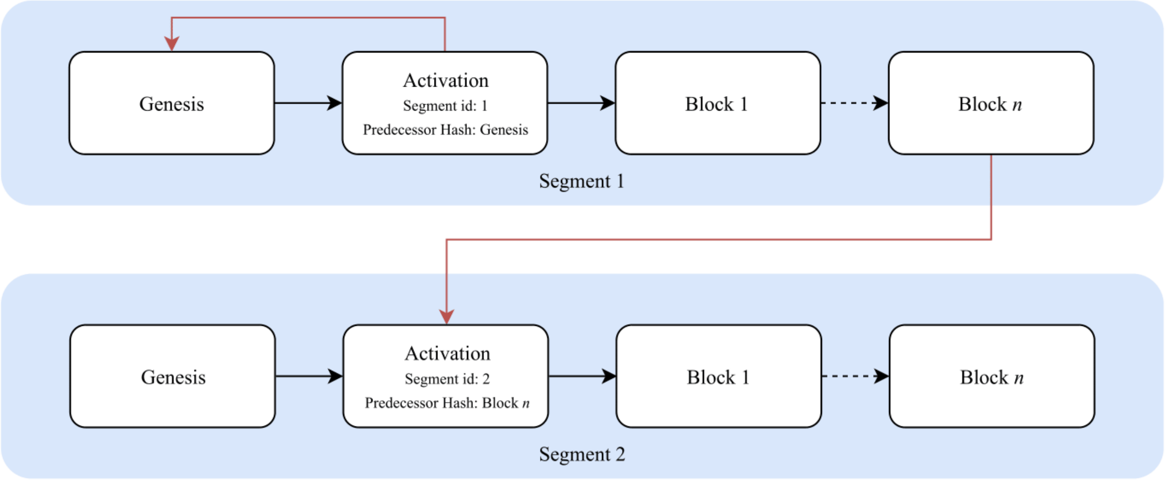

Figure 2: Visual representation of the time-segmentation process's flow. The red arrows represent the use of hashes to allow inter-segment connectivity. Each activation block has the current segment id and the hash of the last block of the previous segment. In the first segment, the predecessor hash is the genesis block hash, making segment 1 the "genesis segment".

A new computing device node that joins the Blockchain, it will only synchronize the latest segment, the one that is currently running on the network. In the case that the segmentation process happens before the bootstrap finishes, the node receives a reboot signal sent by the genesis node that instructs the node to create the new segment as described previously. In the case that the new node is a cold storage node, the node will synchronize previous segments and keep the previous segments stored instead of deleting them.

The presented approach has some similarities to the new Tezos history mode, where the compute device nodes would correspond to Tezos nodes running in the rolling mode and the cold storage nodes would work as the archive mode. But, there are several differences between our proposal and the Tezos history mode, such as node initialization, where the node has to synchronize from the genesis block up to the current information, or use the snapshot feature introduced. In the case of a factory fast pace environment, creating snapshots and waiting for new elements to catch-up in order to start operations is feasible but not practical, which is what Tezos history mode requires.

Our solution has the advantage of only needing to synchronize the latest segment, without the need to request older segment information, with the sole exception of the activation node's previous segment hash that is needed for protocol activation only once per segment. This advantage counterpoints the need of synchronizing the entirety of the Blockchain data from a snapshot generated from a full/archive node and results on a long-term approach to mitigate the growing size of Blockchains in small devices, as opposed to traditional approaches that conduct off-chain operations or compression algorithms, which can also be integrated in our approach.

5 EXPERIMENTAL RESULTS

5.1 Compute Device Node Storage

Experiments were conducted to evaluate the storage requirements of our proposal, comparing the unsegmented Blockchain, running the three node versions (archive, full, rolling) and the proposed approach configured as full, to build the segmented Blockchain, segmenting Blockchain every 10 blocks.

The tests were ran varying the total number of blocks. Random transactions were injected into the network up to a maximum of 32 transaction clients running simultaneously. These clients inject a transaction between random accounts, with value 1 and transaction description with minimum 1000 random characters.

The storage test results are shown in Figure 3. In this, is shown that the proposed method uses an average size of 2282 Kilobytes per segment, having a definite hard limit for the maximum storage occupied by each segment, using 10 as the number of blocks per segment, which is a small number in order to force segments to be created often. The fact that our proposal fixes a maximum size per segment allows the Blockchain to increase its longevity arbitrarily. It also aids with bootstrap due to the fact that a new node won't need to obtain every segment from the start, needing only the latest one to work. Moreover, the size of the rolling mode is not only higher in each node, but the overall Blockchain size continues to heavily grow over time.

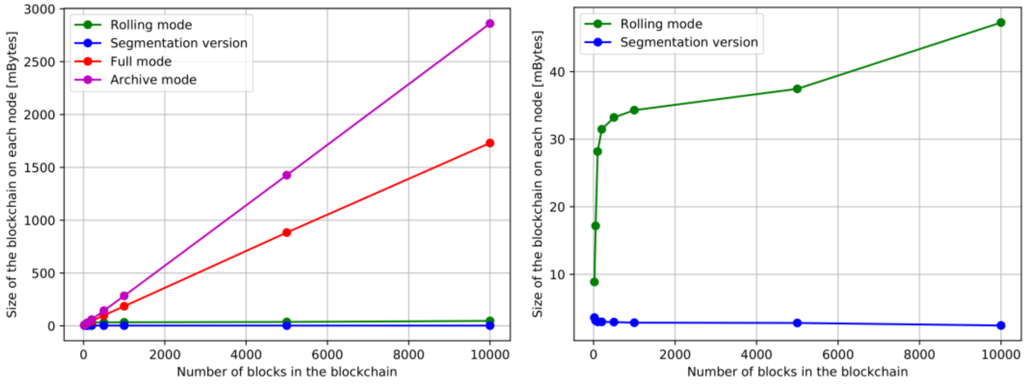

Figure 3: Storage size on each compute device node (not cold storage), for four versions of the Blockchain, as a function of the number of blocks. The segmented Blockchain running mode is archive. The left figure presents the results for the four tests. The right figure presents the results for the rolling mode and the segmentation version, for improved comparison.

With the ability of creating segments limited with respect to the necessary storage space, RAM disk execution on the compute devices is a possibility which improves the Blockchain speed.

5.2 Cold Storage Node

Additional experiments were made to understand how the segmentation affected the storage capabilities of both the compute device nodes and the cold storage nodes, and how the network would grow with the various segments, considering the repeated addition of the genesis and activation blocks to each segment. The tests were ran for 20, 50, 100, 250 blocks per segment with a total of 1000 blocks per experiment. As with the previous experiment, random transactions were injected into the network up to a maximum of 32 transaction clients running simultaneously. The rule for these clients is to inject a transaction between random accounts, with a value of 1, and with a transaction description with a minimum 1000 random characters. The storage results are presented in Figure 4.

The segmentation approach has a smaller running storage footprint, with cold storage space occupied similar to the archive node and depending on the value for the number of segments per block, the resulting space occupied can even be inferior to the rolling mode. As referred, our approach as the benefit of providing a definite hard limit for the segments, with a cold storage space occupied similar to a node running archive mode, considering the increased number of blocks with the added genesis and activation block. In addition, defining an appropriate segment value like 100, can also translate into a smaller size per segment when compared to other modes. This approach has also the advantage of not requiring bootstrapping the entire network and just needing the latest segment.

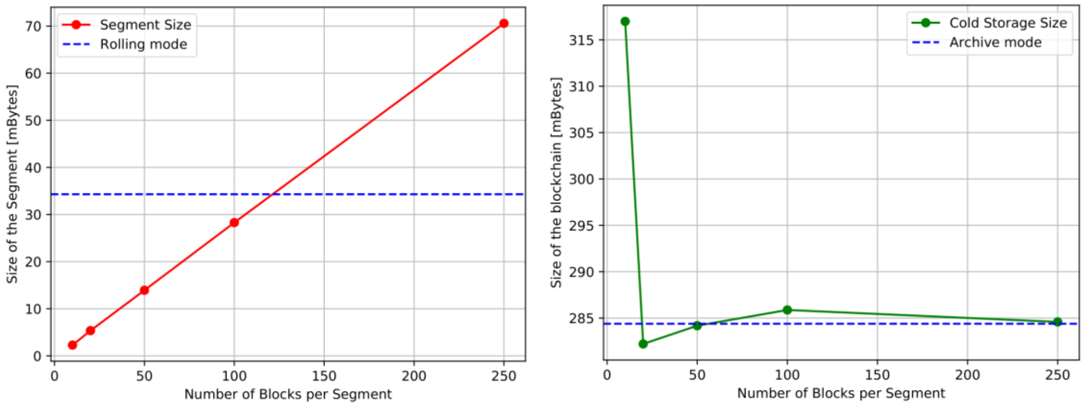

Figure 4: The left figure presents the compute device storage size as a function of the number of blocks per segment for a total of 1000 blocks, compared with the corresponding rolling mode storage size of the Tezos Blockchain. The right figure presents the cold storage size as a number of blocks per segment for a total of 1000 blocks, compared with the archive mode of the Tezos Blockchain. Results for full mode are not presented since they are not comparable to any of the node types on our approach.

Finally it is important to state that both the cold storage node and the compute device node are both running in archive mode and that Tezos rolling mode does not guarantee a fixed maximum node memory size, and the memory requirements slowly grow as can be seen in Figure 3-right. This invalidates the Tezos rolling mode as a solution to the limited capacity of the said nodes, since the rolling mode would eventually exhaust the available memory and the network would stop working.

6 CONCLUSION

This paper proposes a method that improves upon the original proposal of RobotChain [5], a robotic event storage solution, that enables robot monitoring, control, and cooperation, with the introduction of the time-segmentation proposal to solve problems related to small storage capacity of compute modules. This allows the use of cheap compute modules for the majority of network nodes (all but the cold storage ones) and makes the processing and connection of new nodes faster both by allowing the use of faster memory for storing the segment and also because only the current segment is needed for syncing the new node with the network. The solution presented allows the creation of a time-segmented Blockchain that has a definite hard limit on the segments storage capacity, that is independent of how long the Blockchain has run for, which can increase arbitrarily the longevity of the Blockchain.

As future work, other features related to this time-segmentation solution can be implemented, such as RPC interfaces for block retrieval of previous segments, and the possibility of defining a different number of blocks per segment on a node-to-node basis, that could be useful for accommodating nodes with different capabilities.

ACKNOWLEDGMENTS

This work was partially supported by the Tezos Fundation through a grant for project Robotchain and partially supported by project 026653 (POCI-01-0247-FEDER-026653) INDTECH 4.0 – New technologies for smart manufacturing, cofinanced by the Portugal 2020 Program (PT 2020), Compete 2020 Program and the European Union through the European Regional Development Fund (ERDF).

REFERENCES

[1] I. Afanasyev, A. Kolotov, R. Rezin, K. Danilov, A. Kashevnik, and V. Jotsov, “Blockchain solutions for multiagent robotic systems: Related work and open questions,” 2019.

[2] V. Lopes and L. A. Alexandre, “An Overview of Blockchain Integration with Robotics and Artificial Intelligence,” Ledger, vol. 4, Apr 2019.

[3] K. Salah, M. H. U. Rehman, N. Nizamuddin, and A. Al-Fuqaha, “Blockchain for ai: review and open research challenges”, IEEE Access, vol. 7, pp. 10 127–10 149,2019

[4] J. Eberhardt and S. Tai, “On or off the blockchain? insights on off-chaining computation and data,” in European Conference on Service-Oriented and Cloud Computing. Springer, 2017, pp. 3–15.

[5] M. Fernandes and L. A. Alexandre, “RobotChain: Using Tezos Technology for Robot Event Management,” 2018.

[6] L. Goodman, “White paper: Tezos - a self-amending crypto-ledger,” 2014.

[7] E. C. Ferrer, “The blockchain: a new framework for robotic swarm systems,” in Proceedings of the Future Technologies Conference Springer, 2018, pp. 1037–1058

[8] E. C. Ferrer, O. Rudovic, T. Hardjono, and A. Pentland, “RoboChain: A Secure Data-Sharing Framework for Human-Robot Interaction,”

feb 2018.

[9] V. Strobel, E. Castello Ferrer, and M. Dorigo, “Managing Byzantine Robots via Blockchain Technology in a Swarm Robotics Collective Decision-Making Scenario,” Tech. Rep., 2018.

[10] V. Strobel and M. Dorigo, “Blockchain technology for robot swarms: A shared knowledge and reputation management system for collective estimation,” in Swarm Intelligence: 11th International Conference, ANTS, 2018.

[11] N. Teslya and A. Smirnov, “Blockchain-based frame-work for ontology-oriented robots’ coalition formation in cyberphysical systems,” in MATEC Web of Conferences, vol. 161. EDP Sciences, 2018, p. 03018.

[12] V. Lopes and L. A. Alexandre, “Detecting Robotic Anomalies Using RobotChain,” in 2019 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC), April 2019.

[13] V. Lopes, L. A. Alexandre, and N. Pereira, “Controlling Robots using Artificial Intelligence and a Consortium Blockchain,” p. arXiv:1903.00660, 2019.

[14] V. Lopes, N. Pereira, and L. A. Alexandre, “Robot Workspace monitoring using a Blockchain-based 3D Vision Approach,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2019.

[15] J. Eberhardt and J. Heiss, “Off-chaining models and approaches to off-chain computations,” in Proceedings of the 2nd Workshop on Scalable and Resilient Infrastructures for Distributed Ledgers. ACM, 2018, pp. 7–12.

[16] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, “An overview of blockchain technology: Architecture, consensus, and future trends,” in 2017 IEEE International Congress on Big Data (BigData Congress). IEEE,2017, pp. 557–564.

[17] J. Poon and T. Dryja, “The bitcoin lightning network: Scalable off-chain instant payments”, 2016.

[18] Nomadic Labs. (2019) Introducing snapshots and his-tory modes for the tezos node.