Analysis and Evaluation of Functionality of RAFT Consensus Algorithm in Internet of Things Domain

Bartosz Czapski - D10123621
TU060 - MSc in Computer Science (Advanced Software Development) - Technological University Dublin
Starting date: 01/02/2022 All exams passed
1st attempt (Revised)
Data set for this research will be gathered during the experiments

1 Background, Context and Scope

Consensus is regarded as the primary problem which, once solved, will enable implementation of fault-tolerant distributed system (Lamport, 2005). Distributed systems merge with Internet of Things (IoT) creating decentralized system of collaborating Smart Objects(SOs) which sense, store, and interpret information from surrounding environment and act on their own, cooperate with each other exchange information with other kinds of IoT devices, or client nodes (Shi, Cao, Zhang, Li, & Xu, 2016).

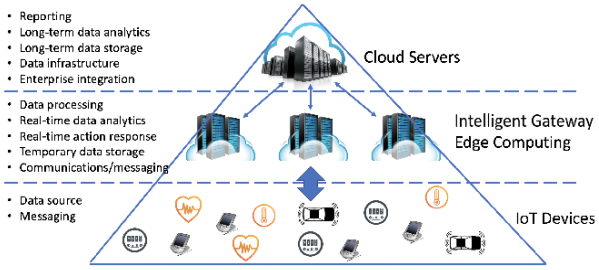


Figure 1: *Layer architecture of edge computing-based IoT* (Yu et al., 2018)

Consensus algorithms enable collection of machines to cooperate as a coherent group, able to survive the failures of some of its members. This feature warrants consensus algorithms to play a key role in building reliable large-scale distributed systems (Ongaro & Ousterhout, 2014).

Multiple authors recently focused on the idea of distributed consensus algorithms, their application in IoT domain (Bof, Carli, & Schenato, 2017; Raghav, Andola, Venkatesan, & Verma, 2020; Whittaker et al., 2020; Zhang, Wu, Han, Feng, & Shu, 2020; Fortino, Fotia, Messina, Rosaci, & Sarné, 2020), and implementation of self-organizing real-time systems architecture (Tošić, Vičič, & Mrissa, 2019; Guerrero, Lera, & Juiz, 2019; Yi, Hao, Qin, & Li, 2015).

Distributed consensus algorithms, when adopted for IoT, provide mechanism for balanced decision making on the edge nodes, and avoiding losing data from IoT devices in the presence of a number of malfunctioning devices by improving the robustness and reliability of the decision process (Li, Oikonomou, Tryfonas, Chen, & Xu, 2014).

2 Problem Description

A consensus algorithm is used to attain, in distributed and multi-agent systems, overall system reliability in the presence of a number of faulty processes, or involving multiple unreliable nodes. When discussed in the context of IoT, consensus algorithm is required to achieve fast event ordering, predictable delivery time and minimal packet loss rate in edge computing networks.

Current approach to consensus between the nodes in IoT domain includes modifying existing consensus algorithms: blockchain (Wang et al., 2020; Tošić et al., 2019), cooperative game model (Gulati & Kaur, 2020), proportional-integral-derivative (PID) (Shi & Yang, 2018), developing new algorithm (Castellano, Esposito, & Risso, 2019), or using revised Paxos consensus algorithm (Poirot, Al Nahas, & Landsiedel, 2019; Cachin, 2010).

Raft algorithm achieves consensus via an elected leader. Raft cluster is built of leaders and followers, and a node in a cluster can be either leader, or fol-

lower (in case of unavailability of the leader, follower becomes candidate and undergoes election process). Raft uses heartbeat messages to implement leader – followers' relationship, where each of the followers expects heartbeat from leader within the timeout frame. Election process start when message is not received within time limit, otherwise timeout is reset (Ongaro & Ousterhout, 2014).

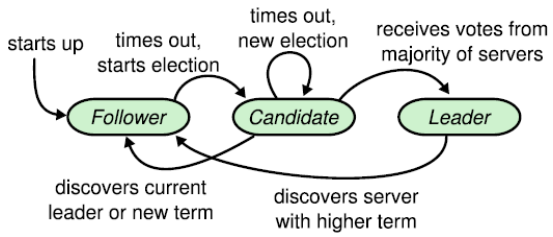


Figure 2: *RAFT protocol*
(Ongaro & Ousterhout, 2014)

Raft has an advantage over the other consensus algorithms, noticeably Paxos, that its understandability is greatly improved (Howard, 2014). As of date of writing this paper, author is unaware of Raft algorithm being used in distributed IoT architecture for achieving consensus among the edge nodes.

2.1 Approaches to solve the problem

Recent works discuss various consensus methods that are applicable to IoT networks. Below is the selection of most interesting propositions of solving consensus problem within IoT domain.

Tošić et al. (2019) proposed decentralized self-balancing architecture that implements consensus algorithm with an aim to improve usage of all IoT devices on the network, and resilience to disconnection while providing increased privacy and security. In their work, authors focused on small number of devices and missed the opportunity to test their architecture with different consensus algorithms against multiple use cases.

Li et al. (2014) designed a distributed consensus algorithm with the goal of improving decision making at the edge nodes of IoT, with a focus on global consensus of multiple services. As per authors' suggestion, more focus should be given to research on collaborative methods for the realization of information exchange and resource allocation in IoT environment.

Similar attempt to develop Distributed Resource AssiGnment and OrchestratioN (DRAGON) algorithm is presented in the paper authored by Castellano et al. (2019). Authors tested proposed algorithm within the scope of two use cases focusing on its convergence and performance properties.

Hao et al. (2018) proposed a novel consensus protocol, EdgeCons, for achieving fast event ordering in edge distributed systems, and although primary results are very promising authors have to design proper monitoring of the system. Additionally, timeframe of recovery of EdgeCons "leadership share" can take interminable amount of time.

Byzantine Fault (BFT) consensus algorithm is adapted by Gramoli (2020) for blockchain technology, and distributed systems. This algorithm, practical BFT (pBFT), achieves clear consensus in each round of operation and cannot be manipulated.

Gulati and Kaur (2020) presented different approach with implementing Weighted Voting Game (WVG) model for conflict resolution in argumentation enabled social IoT networks. Authors simulated and evaluated proposed model's performance within one use case scenario. Additionally, authors notice that implementing model in multi-agent environment to enable agreement among conflicting agents is very promising future work.

Meritocratic mechanism is the base for Fortino et al. (2020) the Friendship and Group Formation (FGF) algorithm designed with organization and improvement of IoT objects collaboration in mind. Authors of the paper could extend the research by implementing additional use cases with multiple agents cooperating.

Oróstica and Núñez (2019) discussed distributed multi-cast algorithm for robust average consensus over internet of things environments. Proposed algorithm was tested and compared with Push-Sum-Based algorithm introduced by Bof et al. (2017) that implemented asynchronous communication between the nodes on the network. Oróstica and Núñez mentioned that their algorithm outperformed Bof et al. algorithm, but missed the opportunity to compare it with additional consensus algorithms.

Bahreini and Grosu (2017) discussed the problem of distributed placement algorithm for ad-hoc optimization in Mobile Edge Computing. They developed algorithm for solving the online version of the placement problem, based on an iterative matching process, which achieved good performance during experiments. As results presented in the paper are very promising, authors did not attempt to implement proposed algorithm in 'real-life' multiple scenarios, and focused on online single use case simulation only.

It is worth mentioning work of Cui et al. (2020) on

positive edge consensus of nodal networks, where authors presented consensus algorithm addressing the issue of consensus in complex networks. It is not directly related to IoT environments but has a potential as "computationally cheaper especially for large-scale systems."

Rocha and Brandão (2019) proposed successful model of scalable architecture for discovering and managing group of IoT devices. Whereas their model is successful, it does not take into account impact of devices' energy consumption, or increase in number of nodes connected.

Salimitari et al. (2020) compared multiple blockchain based consensus methods that can be applied to IoT networks. They did not achieve implementation satisfying all challenges listed in the paper.

Qin et al. (2017) presented a survey advances in consensus of Multi-Agent Systems (MAS), but as they pointed out "survey is far from an exhaustive literature review and there may still be some important results missing in the review due to space limitation."

Similarly, Zhang et al. (2015) and Yang et al. (2016) proposed distributed algorithms that are utilising MAS properties (consensus and invariant summation of state variables) for Economic Dispatch Problem (EDP) which could have interesting applications for IoT architecture.

Al-Doghman et al. (2018) introduced consensus techniques that provide a complete information about network status by implementing one case study using the consensus aggregation within Fog environments.

Abdelwahab et al. (2015) developed algorithms for sensing resource discovery in IoT with their main focus on gossip policy. Paper did not discuss implementation of other communication protocols (e.g., asymmetric, or coordinated broadcast).

Mondal and Tsourdos (2020) discussed the optimal topology problem and way of solving it using Genetic Algorithm (GA) technique to achieve optimal adjacency matrix. Authors came to conclusion that their research can be extended by including more agents in the experimental network by changing the dimension of adjacency matrix.

Raghav et al. (2020) proposed proof of elapsed work and luck (PoEWAL) consensus algorithm for satisfying security requirements (e.g., integrity, authentication, and availability) on IoT devices. Although PoEWAL was compared with other consensus algorithms in terms of consensus time, energy consumption, and network latency authors missed the opportunity to make it more complete by not including scalability, computing requirements and decentralization features.

Similarly, Whittaker et al. (2020) compared multiple consensus protocols (e.g., Paxos, Matchmaker Paxos and Raft), and applied them to IoT universe, focusing on applying proactive reconfiguration in elastic systems.

2.2 Gaps in Research

Although there is extensive body of research in the field of consensus algorithms, and their application for IoT architecture author of this paper noticed that not all avenues are properly explored. One common gap that emerges is the research is the lack of the 'real-life' scenarios (Gulati & Kaur, 2020; Bahreini & Grosu, 2017) and inadequate number of use cases tested (Tošić et al., 2019; Castellano et al., 2019; Fortino et al., 2020; Mondal & Tsourdos, 2020). Another important issue is the absence of proper monitoring (Hao et al., 2018) , and missed opportunity to discus consensus with nodes scalability in mind (Li et al., 2014; Rocha & Brandão, 2019; Raghav et al., 2020). While some of papers discussed provided good comparison of multiple algorithms, most fall short from the comprehensiveness perspective (Oróstica & Núñez, 2019; Salimitari et al., 2020; Abdelwahab et al., 2015). One avenue, that is of particular interest, is the implementation of Raft consensus algorithm for the purpose of minimising latency and packet loss on IoT network in the face of multiple node failures, and contrasting the results with those of Paxos and pBFT algorithms (Whittaker et al., 2020; Gramoli, 2020).

3 Research Question

Can implementation of Raft consensus algorithm in IoT network decrease latency on the network in the presence of malfunctioning edge nodes when compared to Paxos consensus algorithm?

4 Hypothesis

The objective of this research is to implement, analyse and evaluate performance of Raft consensus algorithm on IoT network in the presence of multiple node failures. As current research on the topic of consensus problem in IoT domain does not discuss, nor benchmark Raft, author will attempt to provide

performance measurements (latency, faulty nodes discovery and packet loss rate) in the presence of number of malfunctioning devices. Additionally, results of the above experiment will be compared against the results of Paxos algorithm.

Author of this research will conduct experiment where original (non-existent) data will be collected, and different consensus algorithms will be compared on IoT network to confirm H1 and H4. Quantitative (numerical) data from the experiment will be collected and analysed, and provide definitive result. Exhaustive answers will be provided by comparing quantitative data, author will test and confirm an existing theory.

Null hypothesis H_0: Implementation of the Raft consensus algorithm will not improve latency of the IoT network in the presence of faulty edge nodes.

Alternative hypothesis H_1: Implementation of the Raft consensus algorithm will significantly improve latency of the IoT network in the presence of faulty edge nodes.

Alternative hypothesis H_2: Implementation of the Raft consensus algorithm will increase discovery of the faulty edge nodes in the IoT network.

Alternative hypothesis H_3: Implementation of the Raft consensus algorithm will decrease packet loss rate in in the IoT network.

Alternative hypothesis H_4: Implementation of the Raft consensus algorithm will outperform Paxos consensus algorithm.

5 Design and Implementation

Below table represents planned activities for this this research.

Literature Review	Review of research papers
Design phase	Create use cases: Test Use Cases: • Network A: three edge nodes with sensors attached, • Network B: five edge nodes with sensors attached, • Network C: seven edge nodes with sensors attached, Use Cases for each of the proposed algorithms (Raft and Paxos): • One node failure on each network (A, B, C), • Two nodes failure on networks B and C, • Three nodes failure on networks B and C, • Four nodes failure on network C, Design Dataset
Create an experiment	Build network of nodes, as per use cases, with matching number of identical sensors attached to the nodes.
Conduct control experiments	• Raft Implement Raft algorithm. Measure network latency on A, B, C networks, as per Test Use Cases. Record results. Measure packet loss rate on A, B, C networks, as per Test Use Cases. Record results. • Paxos Implement Paxos algorithm. Measure network latency on A, B, C networks, as per Test Use Cases. Record results. Measure packet loss rate on A, B, C networks, as per Test Use Cases. Record results.
Conduct experiment	• Raft Implement Raft algorithm. Measure network latency on A, B, C networks, as per Use Cases. Record results. Measure packet loss rate on A, B, C networks, as per Use Cases. Record results. • Paxos Implement Paxos algorithm. Measure network latency on A, B, C networks, as per Use Cases. Record results. Measure packet loss rate on A, B, C networks, as per Use Cases. Record results.

Figure 3: *Tasks and Objectives to test hypothesis*

During the Literature review phase additional information regarding the research problem is hoped to be revealed. Design phase will establish use cases that will be followed during the experiments. As this research problem include comparison of efficiency of different algorithms, use cases will be divided in two categories: test use cases and use cases.

Test use cases are planned to be implemented during the control experiment where three networks are created: A, B and C. Network A consists of three IoT nodes, Network B of five IoT nodes, and Network C of seven IoT nodes.

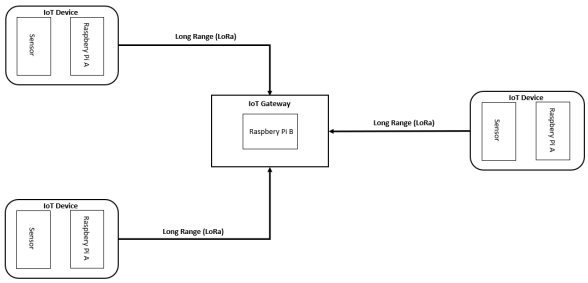


Figure 4: *Example of three node network with gateway*

Third-generation Raspberry Pi (RPP) 3 Model A+ single board computer (SBC) will serve as an IoT edge node. It features a Broadcom BCM2837B0, Cortex-A53 (ARMv8) 64-bit SoC @ 1.4GHz quad-core processor System on a Chip (SoC), 512MB LPDDR2 SDRAM, 2.4GHz and 5GHz IEEE 802.11.b/g/n/ac wireless LAN and Bluetooth 4.2/BLE, Extended 40-pin GPIO header, Single USB

2.0 ports and Micro SD port. Raspberry Pi 3 Model A+ operates on 5V/2.5A DC power input. Sensor attached to each IoT node is DockerPi Sensor Hub Development Board (EP-0106). It integrates various environmental sensors: temperature sensors, humidity sensors, air pressure sensors, lighting, and thermal imaging sensors. Both, RPP and DockerPi sensor, will be connected using MB102 Breadboard with 3.3V 5V MB102 Breadboard Power Supply Module. Environmental data between sensors and nodes will be transmitted continuously. Raspbery Pi 3 B will be used as IoT gateway. Lenovo Ideapad 530s with 16GB of RAM and 8th gen Core i5 processor laptop will be used for debugging. Connectivity between edge nodes and gateway will be established with TTGO Lora32 868/915mhz Sx1276 Esp32 Wifi Lora module.

Gateway	IoT Node/Device	Sensor	Breadboard	Wifi Lora module
Raspberry Pi 3 Model B	Raspberry Pi 3 Model A+	DockerPi Sensor Hub Development Board (EP-0106)	MB102 Breadboard with 3.3V 5V MB102 Breadboard Power Supply module	TTGO Lora32 868/915mhz Sx1276 Esp32

Figure 5: *Components*

Author of this paper chose Python programming language to implement Raft and Paxos algorithms. Python is widely used, and well documented programming language. Raspberry Pi SBC supports is natively. Additionally, Python libraries provide implementation of all two consensus algorithms.

During the control experiment Packet Error Rate (PER) and Round-Trip Time (RTT) between the client and edge note, and packet rate will be measured in networks A, B and C. Control experiment will be repeated for each algorithm. The Raspberry Pi 3 B will act as an IoT gateway (receiver), receiving the environmental sensor data from the IoT devices, the cluster of Raspberry Pi 3 A with DockerPi Sensor Hub (transmitter) attached. The Raspberry Pi B will run a Python script, responsible for receiving, decrypting, and parsing the data payload from the sensor. Received data in JSON format will contain external temperature data, onboard temperature data, humidity data, light sensitivity data and pressure data. The dataset for this project will be constructed from transactions sent between the receiver and transmitters. Author will measure PER by counting the number of packets received by the IoT gateway out of a series of consecutive packets transmitted by the sensors. Additionally, author will attempt to measure RTT which can be defined as the amount of time it takes a packet to get from the transmitter to the receiver and back.

Malfunctioning devices, as per use, cases will be simulated by manually disconnecting Raspberry Pi SBC during the next Experiment stage. Planned use cases include:

- Failure of one node on networks A, B and C.
- Failure two nodes on networks B and C.
- Failure three nodes on networks B and C.
- Failure four nodes on network C.

	Use Case 1	Use Case 2	Use Case 3	Use Case 4
3 nodes network	1 node failure	X	X	X
5 nodes network	1 node failure	2 node failure	3 node failure	X
7 nodes network	1 node failure	2 node failure	3 node failure	4 node failure

Figure 6: *Use cases*

Data collected (RTT in milliseconds, and number of packet send between the nodes) during both phases, Control experiment, and Experiment will create dataset used for comparison of the efficiency of evaluated consensus algorithms (Hao et al., 2018; Borella, Swider, Uludag, & Brewster, 1998). The workload will be capped to 1,000 events (sensor outputs) that will be transmitted on the path sensor-node-client, with their intervals evenly distributed to avoid any inconsistencies.

6 Evaluation

Evaluation process will start with reviewing of data collected during the experiment.

Results Review	• Review data collected during the experiment. • Use paired-samples t-test to statistically compare performance during experiment with different conditions.
Confirm hypothesis	Confirm or reject hypothesis based on the data collected

Figure 7: *Evaluation process*

T-test with paired-samples will be used to statistically compare performance of consensus algorithms during both experiments with different conditions. All RTT data collected during control experiment will be statistically compared to algorithms performances during research experiments.

Proposed hypothesis H_1, H_2, H_3 and H_4 will be accepted independently based on the results achieved. Confirming of hypothesis H_1 and H_4 will confirm the research question.

7 Bibliography

Abdelwahab, S., Hamdaoui, B., Guizani, M., & Znati, T. (2015). Cloud of things for sensing as a service: Sensing resource discovery

and virtualization. In *2015 ieee global communications conference (globecom)* (p. 1-7). doi: 10.1109/GLOCOM.2015.7417252

Al-Doghman, F., Chaczko, Z., & Brookes, W. (2018). Adaptive consensus-based aggregation for edge computing. In *2018 26th international conference on systems engineering (icseng)* (p. 1-8). doi: 10.1109/ICSENG.2018.8638200

Bahreini, T., & Grosu, D. (2017). Efficient placement of multi-component applications in edge computing systems. In *Proceedings of the second acm/ieee symposium on edge computing.* New York, NY, USA: Association for Computing Machinery. doi: 10.1145/3132211.3134454

Bof, N., Carli, R., & Schenato, L. (2017). Average consensus with asynchronous updates and unreliable communication. *IFAC-PapersOnLine*, *50*(1), 601 - 606. (20th IFAC World Congress) doi: https://doi.org/10.1016/j.ifacol.2017.08.093

Borella, M. S., Swider, D., Uludag, S., & Brewster, G. B. (1998). Internet packet loss: measurement and implications for end-to-end qos. In *Proceedings of the 1998 icpp workshop on architectural and os support for multimedia applications flexible communication systems. wireless networks and mobile computing (cat. no.98ex206)* (p. 3-12). doi: 10.1109/ICPPW.1998.721868

Cachin, C. (2010). Yet another visit to paxos..

Castellano, G., Esposito, F., & Risso, F. (2019). A distributed orchestration algorithm for edge computing resources with guarantees. In *Ieee infocom 2019 - ieee conference on computer communications* (p. 2548-2556). doi: 10.1109/INFOCOM.2019.8737532

Cui, Y., Yang, N., & Liu, J. J. (2020). A novel approach for positive edge consensus of nodal networks. *Journal of the Franklin Institute*, *357*(7), 4349 - 4362. doi: https://doi.org/10.1016/j.jfranklin.2020.02.054

Fortino, G., Fotia, L., Messina, F., Rosaci, D., & Sarné, G. (2020). A meritocratic trust-based group formation in an iot environment for smart cities. *Future Generation Computer Systems*, *108*, 34 - 45. doi: https://doi.org/10.1016/j.future.2020.02.035

Gramoli, V. (2020). From blockchain consensus back to byzantine consensus. *Future Generation Computer Systems*, *107*, 760 - 769. doi: https://doi.org/10.1016/j.future.2017.09.023

Guerrero, C., Lera, I., & Juiz, C. (2019). Evaluation and efficiency comparison of evolutionary algorithms for service placement optimization in fog architectures. *Future Generation Computer Systems*, *97*, 131 - 144. doi: https://doi.org/10.1016/j.future.2019.02.056

Gulati, N., & Kaur, P. D. (2020). A game theoretic approach for conflict resolution in argumentation enabled social iot networks. *Ad Hoc Networks*, *107*, 102222. doi: https://doi.org/10.1016/j.adhoc.2020.102222

Hao, Z., Yi, S., & Li, Q. (2018). Edgecons: Achieving efficient consensus in edge computing networks. In *Hotedge.*

Howard, H. (2014, July). *ARC: Analysis of Raft Consensus* (Tech. Rep. No. UCAM-CL-TR-857). University of Cambridge, Computer Laboratory. Retrieved from `https://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-857.pdf`

Lamport, L. (2005). Generalized consensus and paxos..

Li, S., Oikonomou, G., Tryfonas, T., Chen, T. M., & Xu, L. D. (2014). A distributed consensus algorithm for decision making in service-oriented internet of things. *IEEE Transactions on Industrial Informatics*, *10*(2), 1461-1468. doi: 10.1109/TII.2014.2306331

Mondal, S., & Tsourdos, A. (2020). Optimal topology for consensus using genetic algorithm. *Neurocomputing*, *404*, 41 - 49. doi: https://doi.org/10.1016/j.neucom.2020.04.107

Ongaro, D., & Ousterhout, J. (2014). In search of an understandable consensus algorithm. In *Proceedings of the 2014 usenix conference on usenix annual technical conference* (p. 305–320). USA: USENIX Association.

Oróstica, B., & Núñez, F. (2019). A multi-cast algorithm for robust average consensus over internet of things environments. *Computer Communications*, *140-141*, 15 - 22. doi: https://doi.org/10.1016/j.comcom.2019.04.007

Poirot, V., Al Nahas, B., & Landsiedel, O. (2019). Paxos made wireless: Consensus in the air. In *Proceedings of the 2019 international conference on embedded wireless systems and networks* (p. 1–12). USA: Junction Publishing.

Qin, J., Ma, Q., Shi, Y., & Wang, L. (2017). Recent advances in consensus of multi-agent systems: A brief survey. *IEEE Transactions on Industrial Electronics*, *64*(6), 4972-4983. doi: 10.1109/TIE.2016.2636810

Raghav, Andola, N., Venkatesan, S., & Verma, S. (2020). Poewal: A lightweight consensus mechanism for blockchain in iot. *Pervasive and Mobile Computing*, *69*, 101291. doi:

https://doi.org/10.1016/j.pmcj.2020.101291

Rocha, V., & Brandão, A. A. F. (2019). A scalable multiagent architecture for monitoring iot devices. *Journal of Network and Computer Applications*, *139*, 1 - 14. doi: https://doi.org/10.1016/j.jnca.2019.04.017

Salimitari, M., Chatterjee, M., & Fallah, Y. P. (2020). A survey on consensus methods in blockchain for resource-constrained iot networks. *Internet of Things*, *11*, 100212. doi: https://doi.org/10.1016/j.iot.2020.100212

Shi, C.-X., & Yang, G.-H. (2018). Robust consensus control for a class of multi-agent systems via distributed pid algorithm and weighted edge dynamics. *Applied Mathematics and Computation*, *316*, 73 - 88. doi: https://doi.org/10.1016/j.amc.2017.07.069

Shi, W., Cao, J., Zhang, Q., Li, Y., & Xu, L. (2016). Edge computing: Vision and challenges. *IEEE Internet of Things Journal*, *3*(5), 637-646. doi: 10.1109/JIOT.2016.2579198

Tošić, A., Vičič, J., & Mrissa, M. (2019). A blockchain-based decentralized self-balancing architecture for the web of things. In T. Welzer et al. (Eds.), *New trends in databases and information systems* (pp. 325–336). Cham: Springer International Publishing.

Wang, E. K., Sun, R., Chen, C.-M., Liang, Z., Kumari, S., & Khurram Khan, M. (2020). Proof of x-repute blockchain consensus protocol for iot systems. *Computers 'I&' Security*, *95*, 101871. doi: https://doi.org/10.1016/j.cose.2020.101871

Whittaker, M., Giridharan, N., Szekeres, A., Hellerstein, J. M., Howard, H., Nawab, F., & Stoica, I. (2020). *Matchmaker paxos: A reconfigurable consensus protocol [technical report].* doi: https://arxiv.org/abs/2007.09468

Yang, T., Wu, D., Sun, Y., & Lian, J. (2016). Minimum-time consensus-based approach for power system applications. *IEEE Transactions on Industrial Electronics*, *63*(2), 1318-1328. doi: 10.1109/TIE.2015.2504050

Yi, S., Hao, Z., Qin, Z., & Li, Q. (2015). Fog computing: Platform and applications. In *2015 third ieee workshop on hot topics in web systems and technologies (hotweb)* (p. 73-78). doi: 10.1109/HotWeb.2015.22

Yu, W., Liang, F., He, X., Hatcher, W. G., Lu, C., Lin, J., & Yang, X. (2018). A survey on the edge computing for the internet of things. *IEEE Access*, *6*, 6900-6919. doi: 10.1109/ACCESS.2017.2778504

Zhang, W., Liu, W., Wang, X., Liu, L., & Ferrese, F. (2015). Online optimal generation control based on constrained distributed gradient algorithm. *IEEE Transactions on Power Systems*, *30*(1), 35-45. doi: 10.1109/TPWRS.2014.2319315

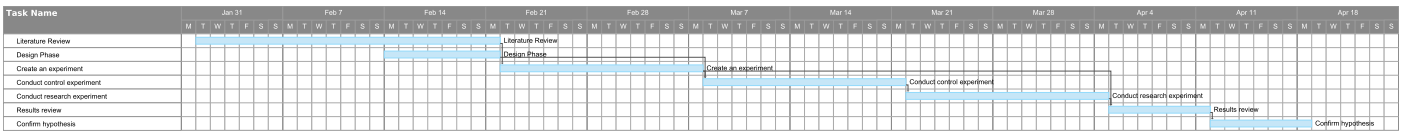
Zhang, W., Wu, Z., Han, G., Feng, Y., & Shu, L. (2020). Ldc: A lightweight dada consensus algorithm based on the blockchain for the industrial internet of things for smart city applications. *Future Generation Computer Systems*, *108*, 574 - 582. doi: https://doi.org/10.1016/j.future.2020.03.009

8 Activities

Below activities planned for this research are spread over the period of 13 weeks.

- O1: Literature Review - 3 weeks
- O2: Design phase - 1 week
- O3: Create an experiment - 2 weeks
- O4: Conduct control experiment - 2 weeks
- O5: Conduct experiment - 3 weeks
- O6: Results Review - 1 week
- O7: Confirm hypothesis - 1 week

Research Proposal Gant Chart



9 Appendix

Research question, hypothesis and preliminary design

Bartosz Czapski, TU060, ASD, D10123621

Domain and scope of research - ACM 2012

A: Social and professional topics → Computing / technology policy → Privacy policies
(Senarath & Arachchilage, 2019; Biega et al., 2020; Senarath & Arachchilage, 2018)

B: Social and professional topics → Computing / technology policy → Government technology policy → Governmental regulations
(Pernot-Leplay, 2020; Sullivan, 2019)

C: Security and privacy → Human and societal aspects of security and privacy → Social aspects of security and privacy
(Lee, 2020; Balapour et al., 2020; Li et al., 2019)

D: Security and privacy → Human and societal aspects of security and privacy → Privacy protections
(Amato et al., 2020; Mousavi et al., 2020; Mazel et al., 2019)

E: Security and privacy → Software and application security → Domain-specific security and privacy architectures
(Chen, 2020; Barbosa et al., 2019; Antignac, Le Métayer, 2014)

Domain: Data Governance and Privacy Implementation

Scope: Research on design and implementation of Data Minimization methodology during systems design process.

Gaps in the literature and research question

Data Minimization is a principle of collecting, storing and managing minimal amount of data required by the system to fulfil its designed purpose. Senarath and Arachchilage (2019) attempted to design Privacy Engineering Methodology (PEM) by conducting three studies: Study I with 24 participants, Study II - 9 participants, and Study III - 149 participants. Authors missed opportunity of involving multinational organizations where PEM could be tested among larger, and more organized group of participants. Using multiple, different scenarios instead of health application used by Senarath and Arachchilage (2018) could potentially provide broader, more complete data. Study group size was addressed in Balapour, et al. (2020) research where authors used 1544 participants divided into two groups, tasked with collection of more and less sensitive data. Authors used five-point Likert scale (from "strongly disagree" to "strongly agree") which might be considered insufficient. Additionally, all participants of Balapour, et al. (2020) research were located in U.S.A. which considering how Data Privacy is regulated across different jurisdictions, might have affected their responses (Pernot-Leplay, 2020). Implementation of Vulnerability-Privacy Concern-Resistance (VPR) framework in Lee's (2020) research focused on Internet of Things (IoT) where author's goal was to determine factors that can inhibit acceptance of the technology among participants. Lee (2020) examined response from 265 participants, where 66% had previous IoT home experience. Increase in number of participants without IoT experience, and more balanced group could greatly improve proposed model. Barbosa, et al. (2019) proposed Privacy by Evidence (PbE) methodology for implementing privacy guidelines into systems development process. Case studies used in this research are very generic, and missing details on conditions and research concluded. Additionally, use of weights and calculation of the final metrics is unclear and oversimplified. As mentioned by Barbosa, et al. (2019) ", a quantitative experiment to compare results between team of developers using PbE and team of developers not following methodology would be a great addition to the results of experiment."

Research Question: Can early adoption of Data Minimization methodology during system design improve data privacy?

Hypothesis

Null hypothesis H_0:

Early implementation of data minimisation methodology has nugatory effect on data collection during systems design.

Alternative hypothesis H_1:

Early implementation of data minimization methodology significantly improves data collection process and data privacy during systems design.

Feasibility of the study

TASK	DESCRIPTION	TIMEFRAME
Corporate Engagement	Involvement of the large corporations into the research of implementation of Data Minimization methodology (DMM). By involving tech companies into research broader response can be achieved, with more reliable results.	8 - 10 weeks
Literature Review	Review of research papers.	3 -4 weeks
Design phase	Design of DMM used in research with adequate data set, questionnaire, and workshops design. Design of the system development scenario where data governance and DMM can be used.	10 - 12 weeks
Control Study	Workshop with developers tasked with system design without DMM in place to test unconstrained approach to data collection.	1 week
Methodology Review	Review of the control study results, and incorporating findings into methodology design.	2 weeks
Research Study	Workshop with developers tasked with system design using DMM to test its functionality and usability.	1 week
Methodology Review	Review of the research study results, and incorporating findings into methodology design.	2 weeks
Final Study	Workshop with developers tasked with application design using DMM (with assumption that additional changes to methodology improved process of data minimization).	1 week
Results Review	Review and address findings.	4 weeks

Bibliography

Amato F., Coppolino, L., D'Antonio, S., Mazzocca, N., Moscato, F., Sgaglione, L. (2020). An abstract reasoning architecture for privacy policies monitoring. Future Generation Computer Systems 106, 393-400. https://doi.org/10.1016/j.future.2020.01.019.

Antignac T., Le Métayer D. (2014) Privacy Architectures: Reasoning about Data Minimisation and Integrity. In: Mauw S., Jensen C.D. (eds) Security and Trust Management. STM 2014. Lecture Notes in Computer Science, vol 8743, (17-32) . Springer.

Balapour, A., Nikkhah, H., R., Sabherwal, R. (2020). Mobile application security: Role of perceived privacy as the predictor of security perceptions. International Journal of Information Management 52, 102063. https://doi.org/10.1016/j.ijinfomgt.2019.102063.

Barbosa, P., Brito, A., Almeida, H. (2019). Privacy by Evidence: A Methodology to develop privacy-friendly software applications. Information Sciences. Available online 25 September 2019. https://doi.org/10.1016/j.ins.2019.09.040.

Biega, A., J., Potash, P., Daumé III, H., Diaz, F., Finck. M. (2020). Operationalizing the Legal Principle of Data Minimization for Personalization. SIGIR '20: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information, 399–408. https://doi.org/10.1145/3397271.3401034.

Chen, X. (2020). A security integration model for private data of intelligent mobile communication based on edge computing. Computer Communications 162, 204-211. https://doi.org/10.1016/j.comcom.2020.08.026.

General Data Protection Regulation (EU), (2016). European Union data protection. https://ec.europa.eu/info/law/law-topic/data-protection/data-protection-eu_en. Accessed on: 2020-03-31.

Lee, H. (2020). Home IoT resistance: Extended privacy and vulnerability perspective. Telematics and Informatics 49, 101377. https://doi.org/10.1016/j.tele.2020.101377.

Li, P., Cho, H., Goh, Z., H. (2019). Unpacking the process of privacy management and self-disclosure from the perspectives of regulatory focus and privacy calculus. Telematics and Informatics 41, 114–125. https://doi.org/10.1016/j.tele.2019.04.006.

Bibliography

Mazel, J., Garnier, R., Fukuda, K. (2019). A comparison of web privacy protection techniques. Computer Communications 144, 162-174. https://doi.org/10.1016/j.comcom.2019.04.005.

Mousavi, R., Chen, R., Kim, D., J., Chen, K. (2020). Effectiveness of privacy assurance mechanisms in users' privacy protection on social networking sites from the perspective of protection motivation theory. Decision Support Systems 135, 113323. https://doi.org/10.1016/j.dss.2020.113323.

Pernot-Leplay, E., (2020). China's Approach on Data Privacy Law: A Third Way Between the U.S. and the EU? Penn State Journal of Law & International Affairs, 8. Available at: https://ssrn.com/abstract=3542820. Accessed on: 13.10.2020.

Sedayao, J. , Bhardwaj, R. , Gorade, N. (2014). Making big data, privacy, and anonymization work together in the enterprise: experiences and issues. 2014 3rd International Congress on Big Data (Big Data Congress), (601–607). https://doi.org/10.1109/BigData.Congress.2014.92.

Senarath, A., Arachchilage, N., A., G. (2018). Understanding user privacy expectations: A software developer's perspective. Telematics and Informatics 35, 1845-1862. https://doi.org/10.1016/j.tele.2018.05.012.

Senarath, A., Arachchilage, N., A., G. (2019). A data minimization model for embedding privacy into software systems. Computers & Security 87, 101605. https://doi.org/10.1016/j.cose.2019.101605.

Sullivan, C. (2019). EU GDPR or APEC CBPR? A comparative analysis of the approach of the EU and APEC to cross border data transfers and protection of personal data in the IoT era. Computer Law & Security Review 35, 380-397. https://doi.org/10.1016/j.clsr.2019.05.004.

Dataset

Dataset for this study is based on GDPR (2016), and all data will be collected during workshops where developers will be presented with system design scenario. Every data field relevance will be described using seven-point Likert scale, ranging from “Strongly Disagree” to “Strongly Agree”.

Personal Data	Type	Description
name	String	Name of the individual
surname	String	Surname of the individual
address	String	Address of the individual
email	String	Email of the individual
mobile_phone	String	Phone number of the individual
age	Int	Age of the individual

Sensitive Personal Data	Type	Description
race	String	Ethnic origin of the individual
religion	String	Religious beliefs of the individual
mental_issues	String	Mental health of the individual
physical_issues	String	Physical health of the individual
sexuality	String	Sexuality of the individual
offence	String	Offence committed or alleged to have been committed

Likert Scale
Strongly Disagree
Disagree
Somewhat Disagree
Neither Agree nor Disagree
Somewhat Agree
Agree
Strongly Agree

Research question, hypothesis and refined design

Bartosz Czapski, TU060, ASD, D10123621

Domain and scope of research - ACM 2012

A:**Computer systems organization → Real-time systems → Real-time system architecture**
(Tošić, Vičič & Mrissa, 2019; Castellano, Esposito & Risso, 2019; Rocha & Brandão, 2019; Guerrero, Lera & Juiz, 2019)

B:**Theory of computation → Design and analysis of algorithms → Distributed algorithms → Self-organization**
(Li et al., 2014; Castellano, Salimitari, Chatterjee & Fallah, 2020; Oróstica & Núñez, 2019; Qin et al., 2017; Gulati & Kaur, 2020; Mondal & Tsourdos, 2020; Shi & Yang, 2018)

C:**Computer systems organization → Architectures → Distributed architectures**
(Tapas et al. 2020; Abdelwahab et al., 2015)

D:**Computing methodologies → Distributed computing methodologies → Distributed algorithms → Self-organization**
(Bof, Carli & Schenato, 2017; Raghav et al., 2020; Whittaker et al., 2020; Zhang et al., 2020; Fortino, et al., 2020)

E:**Information systems → Information systems applications → Computing platforms**
(Fortino et al. 2020; Al-Doghman, Brookes & Chaczko, 2018)

Domain:

Internet of Things architecture

Scope:

Distributed consensus algorithms

Gaps in the literature and research question

Consensus algorithms, when adopted for Internet of Things (IoT), provide mechanism for balanced decision making on the edge nodes, and avoiding loosing data from IoT devices in the presence of a number of malfunctioning processes (Li et al. 2014). Tošić et al. (2019) proposed decentralized self-balancing architecture that implements consensus algorithm with an aim to improve usage of all IoT devices on the network, and resilience to disconnection while providing increased privacy and security. In their work, authors focused on small number of devices and missed the opportunity to test their architecture with different consensus algorithms against multiple use cases. Li et al. (2014) designed a distributed consensus algorithm with the goal of improving decision making at the edge nodes of IoT, with a focus on global consensus of multiple services. As per authors' suggestion, more focus should be given to research on collaborative methods for the realization of information exchange and resource allocation in IoT environment. Similar attempt to develop Distributed Resource AssiGnment and OrchestratioN (DRAGON) algorithm is presented in the paper authored by Castellano, et al. (2019). Authors tested proposed algorithm within the scope of two use cases focusing on its convergence and performance properties. Gulati & Kaur (2020) present different approach with implementing Weighted Voting Game (WVG) model for conflict resolution in argumentation enabled social IoT networks. Authors simulated and evaluated proposed model's performance within one use case scenario. Additionally, authors notice that implementing model in multi-agent environment to enable agreement among conflicting agents is very promising future work. Meritocratic mechanism is the base for Fortino, et al. (2020) the Friendship and Group Formation (FGF) algorithm designed with organization and improvement of IoT objects collaboration in mind. Authors of the paper could extend the research by implementing additional use cases with multiple agents cooperating. Oróstica & Núñez (2019) discussed distributed multi-cast algorithm for robust average consensus over internet of things environments. Proposed algorithm was tested and compared with Push-Sum-Based algorithm introduced by Bof et al. (2017) that implemented asynchronous communication between the nodes on the network. Oróstica & Núñez mentioned that their algorithm outperformed Bof et al. algorithm, but missed the opportunity to compare it with additional consensus algorithms. It worth mentioning work of Cui, Yang & Liu (2020) on positive edge consensus of nodal networks, where authors presented consensus algorithm addressing the issue of consensus in complex networks. It is not directly related to IoT environments but has a potential as "computationally cheaper especially for large-scale systems." Rocha & Brandão (2019) proposed scalable architecture for discovering and managing group of IoT devices. Whereas their model is successful, it does not take into account impact of devices' energy consumption, or increase in number of nodes connected. Salimitari, Chatterjee & Fallah (2020) compared multiple blockchain based consensus methods that can be applied to IoT networks. They did not achieve implementation satisfying all challenges listed in the paper. Qin, et al. (2017) present a survey advances in consensus of Multi-Agent Systems (MAS), but as they pointed out "survey is far from an exhaustive literature review and there may still be some important results missing in the review due to space limitation." Al-Doghman, Brookes & Chaczko (2018) introduced consensus techniques that provide a complete information about network status by implementing one case study using the consensus aggregation within Fog environments. Abdelwahab et al. (2015) developed algorithms for sensing resource discovery in IoT with their main focus on gossip policy. Paper did not discuss implementation of other communication protocols (e.g., asymmetric, or coordinated broadcast). Mondal & Tsourdos (2020) discussed the optimal topology problem and way of solving it using Genetic Algorithm (GA) technique to achieve optimal adjacency matrix. Authors came to conclusion that their research can be extended by including more agents in the experimental network (by changing the dimension of adjacency matrix). Raghav et al. (2020) proposed proof of elapsed work and luck (PoEWAL) consensus algorithm for satisfying security requirements (e.g., integrity, authentication, and availability) on IoT devices. Although PoEWAL was compared with other consensus algorithms in terms of consensus time, energy consumption, and network latency authors missed the opportunity to make it more complete by not including scalability, computing requirements and decentralization features. Similarly, Whittaker et al. (2020) compared multiple consensus protocols (e.g., Paxos, Matchmaker Paxos and Raft), and applied them to IoT universe, focusing on applying proactive reconfiguration in elastic systems.

Research Question:

Can implementation of Raft consensus algorithm in IoT network decrease latency on the network in the presence of malfunctioning edge nodes when compared to other leading consensus algorithms?

Hypothesis + research methods

Null hypothesis H_0: Implementation of the Raft consensus algorithm will not improve latency of the IoT network in the presence of faulty edge nodes.

Alternative hypothesis H_1: Implementation of the Raft consensus algorithm will significantly improve latency of the IoT network in the presence of faulty edge nodes.

Alternative hypothesis H_2: Implementation of the Raft consensus algorithm will increase discovery of the faulty edge nodes in the IoT network.

Research methods:

- **By type: Primary research**
  Author of this research will conduct experiment where original (non-existent) data will be collected. Different consensus algorithms will be compared on IoT network to confirm H_1 and H_2.
- **By objective: Quantitative research**
  Quantitative (numerical) data from the experiment will be collected and analysed, and provide definitive result.
- **By form: Empirical research**
  The answers will be provided by comparing quantitative data.
- **By reasoning: Deductive research**
  Author will test and confirm an existing theory.

Tasks/Objectives to test hypothesis using statistical tools

- **O1** **Literature Review** — Review of research papers — **2 -3 weeks**

- **O2** **Design phase** — Create use cases (network of three, five and seven edge nodes) and design experiment and data set — **1 week**

	Use Case 1	Use Case 2	Use Case 3	Use Case 4
3 nodes network	1 node failure	X	X	X
5 nodes network	1 node failure	2 node failure	3 node failure	X
7 nodes network	1 node failure	2 node failure	3 node failure	4 node failure

- **O3** **Create an experiment** — Build network of nodes, as per use cases, with matching number of identical sensors attached to the nodes. — **2 weeks**

Client	IoT Node	Sensor	Transceiver Module
Lenoco Ideapad 530s	Raspberry Pi 3 Model B+	Arduino Nano 33 BLE Sense	REYAX RYLR896 LoRa

- **O4** **Conduct control experiment** — **2 weeks**
  - Implement Raft algorithm and measure network latency on three, five and seven nodes IoT network.
  - Implement Paxos algorithm and measure network latency on three, five and seven nodes IoT network.
  - Implement DRAGON algorithm and measure network latency on three, five and seven nodes IoT network.
  - Implement PoEWAL algorithm and measure network latency on three, five and seven nodes IoT network.

- **O5** **Conduct experiment** — Test algorithms performance (network latency) from control experiment in below experiments — **3 weeks**

	Experiment 1	Experiment 2	Experiment 3	Experiment 4
3 nodes network	1 node failure	X	X	X
5 nodes network	1 node failure	2 node failure	3 node failure	X
7 nodes network	1 node failure	2 node failure	3 node failure	4 node failure

- **O6** **Results Review** — Review data collected during the experiment. Use paired-samples t-test to statistically compare performance during experiment with different conditions. — **1 week**

- **O7** **Confirm hypothesis** — Confirm or reject hypothesis based on the data collected — **1 week**

Bibliography

Abdelwahab, S., Hamdaoui, B., Guizani, M., Znati, T. (2015). Cloud of Things for Sensing as a Service: Sensing Resource Discovery and Virtualization. 2015 IEEE Global Communications Conference (GLOBECOM) 1-7. https://doi.org/10.1109/GLOCOM.2015.7417252.

Al-Doghman, F., Brookes, W., Chaczko, Z. (2018). Adaptive Consensus-based Aggregation for Edge Computing. 26th International Conference on Systems Engineering (ICSEng), 1-8. https://doi.org/10.1109/ICSENG.2018.8638200.

Bof, N., Carli, R., Schenato, L. (2017). Average Consensus with Asynchronous Updates and Unreliable Communication. IFAC-PapersOnLine 50(1), 601-606. https://doi.org/10.1016/j.ifacol.2017.08.093.

Casado-Vara, R., Chamoso, P., De la Prieta, F., Prieto, J., Corchado, J., M. (2019). Non-linear adaptive closed-loop control system for improved efficiency in IoT- blockchain management. Information Fusion 49, 227–239. https://doi.org/10.1016/j.inffus.2018.12.007.

Castellano, G., Esposito F., Risso, F. (2019). A Distributed Orchestration Algorithm for Edge Computing Resources with Guarantees. IEEE INFOCOM 2019 - IEEE Conference on Computer Communications, 2548-2556. https://doi.org/10.1109/INFOCOM.2019.8737532.

Cui, Y., Yang, N., Liu, J. J. R. (2020). A novel approach for positive edge consensus of nodal networks. Journal of the Franklin Institute 357(7), 4349-4362. https://doi.org/10.1016/j.jfranklin.2020.02.054.

Forti, S., Gaglianese, M., Brogi, A. (2020). Lightweight self-organising distributed monitoring of Fog infrastructures. Future Generation Computer Systems 114, 605–618. https://doi.org/10.1016/j.future.2020.08.011.

Fortino, G., Fotia, L., Messina, F., Rosaci, D., Sarné, G,. M., L. (2020). A meritocratic trust-based group formation in an IoT environment for smart cities. Future Generation Computer Systems 108, 34–45. https://doi.org/10.1016/j.future.2020.02.035.

Guerrero, C., Lera, I., Juiz, C. (2019). Evaluation and efficiency comparison of evolutionary algorithms for service placement optimization in fog architectures. Future Generation Computer Systems 97, 131–144. https://doi.org/10.1016/j.future.2019.02.056.

Gulati, N., Kaur, P., D. (2020). A game theoretic approach for conflict resolution in argumentation enabled social IoT networks. Ad Hoc Networks 107, 102222 https://doi.org/10.1016/j.adhoc.2020.102222.

Li, S. C., Oikonomou, G., Tryfonas, T., Chen, T. M., Xu, L. D. (2014). A distributed consensus algorithm for decision making in service-oriented internet of things. IEEE Transactions on Industrial Informatics, 10(2), 1461-1468. https://doi.org/10.1109/tii.2014.2306331.

Bibliography

Mondal, S., Tsourdos, A. (2020). Optimal topology for consensus using genetic algorithm. Neurocomputing 404, 41–49. https://doi.org/10.1016/j.neucom.2020.04.107.

Oróstica, B., Núñez, F. (2019). A multi-cast algorithm for robust average consensus over internet of things environments. Computer Communications 140–141, 15–22. https://doi.org/10.1016/j.comcom.2019.04.007.

Qin, J., Ma, Q., Shi, Y., Wang, L. (2017). Recent Advances in Consensus of Multi-Agent Systems: A Brief Survey. IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, 64(6), 4972-4983. https://doi.org/10.1109/TIE.2016.2636810.

Raghav, Andola, N., Venkatesan, S., Verma, S. (2020). PoEWAL: A lightweight consensus mechanism for blockchain in IoT. Pervasive and Mobile Computing 69, 101291. https://doi.org/10.1016/j.pmcj.2020.101291.

Rocha, V., Brandão. A., A., F. (2019). A scalable multiagent architecture for monitoring IoT devices. Journal of Network and Computer Applications 139, 1–14. https://doi.org/10.1016/j.jnca.2019.04.017.

Salimitari, M., Chatterjee, M., Fallah, Y. P. (2020). A survey on consensus methods in blockchain for resource-constrained IoT networks. Internet of Things 11, 100212. https://doi.org/10.1016/j.iot.2020.100212.

Shi, C.-X., Yang, G.-H. (2018). Robust consensus control for a class of multi-agent systems via distributed PID algorithm and weighted edge dynamics. Applied Mathematics and Computation 316, 73–88. http://dx.doi.org/10.1016/j.amc.2017.07.069.

Tapas, N., Longo, F., Merlino, G., Puliafito, A. (2020). Experimenting with smart contracts for access control and delegation in IoT. Future Generation Computer Systems 111, 324–338. https://doi.org/10.1016/j.future.2020.04.020.

Tošić A., Vičič J., Mrissa M. (2019) A Blockchain-Based Decentralized Self-balancing Architecture for the Web of Things. In: Welzer T. et al. (eds) New Trends in Databases and Information Systems. ADBIS 2019. Communications in Computer and Information Science, vol 1064. Springer, Cham. https://doi.org/10.1007/978-3-030-30278-8_34.

Whittaker, M., Giridharan, N., Szekeres, A., Hellerstein, J.M., Howard, H., Nawab, F., & Stoica, I. (2020). Matchmaker Paxos: A Reconfigurable Consensus Protocol [Technical Report]. ArXiv, abs/2007.09468. Available at: https://arxiv.org/abs/2007.09468. Retrieved on: 15/11/2020.

Zhang, W., Wu, Z., Han, G., Feng, Y., Shu, L. (2020). LDC: A lightweight dada consensus algorithm based on the blockchain for the industrial Internet of Things for smart city applications. Future Generation Computer Systems 108, 574–582. https://doi.org/10.1016/j.future.2020.03.009.

Dataset

Data set for this research will be gathered during the experiments. It will be the collection of message round-trip time (nanoseconds) between the edge node - Raspberry Pi 3 Model B+, and the client – Lenovo Idepad 530s. As edge nodes will be constantly fed the output of the sensors, they will update client within an identical interval.

All round-trip time data collected during control experiment will be statistically compared to algorithms performances during research experiments.