

Copyright © 2019

[This article is published under the terms of the Attribution-ShareAlike 4.0 International \(CC](#)

[BY-SA\)](#)



<https://revistas.udistrital.edu.co/ojs/index.php/Tecnura/issue/view/1102>

DOI: [https://doi.org/ 10.14483/22487638.18628](https://doi.org/10.14483/22487638.18628)

Clasificación del artículo: **Investigacion**

Monitoring Algorithm for Electrical Variables

Implementing Blockchain and Python

Algoritmo de monitoreo de variables eléctricas

implementando *blockchain* y Python

Fecha de recepción: 28 de septiembre de 20201

Fecha de aceptación: 20 de octubre de 2021

Cómo citar: Muñoz-Romero., J.A. Giral-Ramírez., D.A. y Ramírez-Vanegas., C.A. (2022).

Monitoring Algorithm for Electrical Variables Implementing Blockchain and Pytho. Tecnura,

26(71). [https://doi.org/ 10.14483/22487638.18628](https://doi.org/10.14483/22487638.18628)

Javier Andres Muñoz-Romero

Electrical Engineering. Electricity Technologist. Project Engineer, Axon Group. Bogotá, Colombia.

Email: javier.munoz@axongroup.com.co

ORCID: <https://orcid.org/0000-0002-7131-7190>

Diego Armando Giral-Ramírez

Master in Electrical Engineering, Electrical Engineer. Assistant professor, Technological Department, Universidad Distrital Francisco José de Caldas. Bogotá D. C., Colombia.

Email: dagiralr@udistrital.edu.co

ORCID: <https://orcid.org/0000-0001-9983-4555>

Carlos Alberto Ramírez-Vanegas

Master in Electrical Engineering, Electrical Engineer. Professor Universidad Tecnológica de Pereira. Pereira, Colombia.

Email: caramirez@utp.edu.co

ORCID: <https://orcid.org/0000-0002-6646-4209>

ABSTRACT

Objective: This article presents a blockchain application in power systems using electrical variable monitoring delivered by a simulated relay.

Methodology: The electrical variables are verified, compared, and uploaded to a blockchain network created by a network of servers. These are responsible for uploading, validating, and exporting information to the algorithm. This study is carried out by means of the Python programming language.

Results: An algorithm capable of integrating these variables from a simulated relay is created, as well as four servers in charge of taking the relay signals every minute, validating that they are unique in the blockchain and uploading them. The first node that completes this task uploads its identification and the information of the relay. Once obtained, the information processing times are approximately 10 seconds.

Conclusions: It is concluded that the algorithm is capable of decentralizing the information collected by the servers in times equal to or greater than one minute, which can be very useful when saving information. In applications such as control by the network operator, it falls short by having times higher than or equal to one minute. It should be taken into account that, for the development of a blockchain with a greater number of equipment, it is advisable to use machines with high levels of processing and much greater RAMs. These characteristics would allow the algorithm to run smoothly and in the shortest time possible.

Financing: Universidad Distrital Francisco José de Caldas

Keywords: blockchain, GetChain, data mining, monitoring algorithm, electrical substation

RESUMEN

Objetivo: Este artículo presenta una aplicación *blockchain* en sistemas de energía que emplea el monitoreo de variables eléctricas entregado por un relé simulado.

Metodología: Las variables eléctricas se verifican, comparan y cargan en una red *blockchain* creada por una red de servidores; estos son responsables de cargar, validar y exportar información al sistema. Este estudio se realiza a través del lenguaje de programación Python.

Resultados: Se crea un algoritmo capaz de integrar estas variables desde un relé simulado, así como 4 servidores que se encargan de tomar las señales del relé cada minuto, validando que son únicas en la *blockchain* y subiéndolas. El primer nodo que completa esta tarea carga su identificación y la información del relé. Una vez obtenida, los tiempos de procesamiento de la información son aproximadamente 10 segundos.

Conclusiones: Se concluye que el algoritmo es capaz de descentralizar la información recolectada por los servidores en tiempos iguales o mayores a un minuto, lo que puede ser de gran utilidad a la hora de guardar información. En aplicaciones como el control por parte del operador de red, se queda corto al tener tiempos mayores o iguales a un minuto. Se debe tener en cuenta que, para el desarrollo de una cadena de bloques con una mayor cantidad de equipos, es recomendable utilizar máquinas con altos niveles de procesamiento y memorias RAM mucho más grandes. Estas características permitirían que el algoritmo funcionara sin problemas y en el menor tiempo posible.

Financiamiento: Universidad Distrital Francisco José de Caldas

Palabras clave: blockchain, GetChain, minería de datos, sistema de monitoreo, subestación eléctrica

INTRODUCTION

Currently, blockchain algorithms are used for the development of smart contracts and controlling digital currencies such as Bitcoin, among other applications, as the benefits of blockchain become more visible. In recent years, there was an exponential growth in the use of systems based on this technology (Pandey *et al.*, 2019). This type of algorithm allows maintaining greater security in the fulfillment of the so-called transactions. This algorithm

also avoids information manipulation through the encryption offered by algorithms such as SHA256 and others, given that information is handled in a decentralized manner through a network of servers connected between them. These servers also control the same information and validate a new distribution made and added to the chain. A blockchain can handle data at a low cost, releasing it from intelligent devices to create economic value (Yu *et al.*, 2019).

General context

Over time, the demand for electricity supply grows more and more, so power systems need to have constant monitoring and also guarantee the users' energy quality. Ongoing monitoring, control, and event recording systems are centralized and monopolized (Alam *et al.*, 2019). Their structure includes a control center, which is responsible for monitoring and controlling the different variables in an electrical substation.

Over the past few years, prosumers (producer and consumer) have emerged, so microgrids have become popular with their bidirectional power supply configuration. However, current systems cannot manage, trade, and ensure the security of these systems. Alam *et al.* (2019) propose a blockchain-based approach to address these challenges, where the algorithms attempt to implement smart contracts, mode coalitions, and negotiate electricity trade bidirectionally.

In the electronic power trading sector, the use of blockchain has several benefits, such as data integrity and the detection of any unauthorized intervention, elimination of monopoly at any stage of the market, transparency in the trading system, security and identity protection, and the facilitation of secure energy and credit transactions within the system (Alam *et al.*, 2019; N. Wang *et al.*, 2019).

Nowadays, blockchain technology has gained increased attention (Pava *et al.*, 2021). However, it still poses several technical challenges, such as scalability and security. Over time, these challenges have been tested to strengthen the barriers against threats and system attacks (Leka *et al.*, 2019). Thus, the public-key infrastructure (PKI) that manages the blockchain is the basis and the core of building network security. Blockchain has many technical features, such as decentralization and the impossibility of being manipulated and forged (Foti *et al.*, 2021). These features increase the reliability in information, security, traceability, and other aspects of traditional technology (R. Wang *et al.*, 2019).

A blockchain contains a series of structures or data blocks. Here, the information is stored in an encrypted blockchain. Likewise, the data is distributed in a network by all the nodes (Pava *et al.*, 2021; Yang *et al.*, 2021). The data records are synchronized, which means they keep the same data throughout the network for each server. This means that data cannot be modified by one server without the authorization and validation of all other network administrators, thus making blockchain an immutable data management system. Moreover, any addition of information is verified (Alam *et al.*, 2019).

Therefore, end-users must supervise power systems according to their needs in order to guarantee the quality of the electric service and the authenticity of the information supplied by the grid operator.

Contributions and scope

The objective of the development of this algorithm is to show that energy service customers can perform system verifications, substation-occurring event validation, and constant monitoring of the algorithm. In addition, customers become watchdogs for the reports issued

by the grid operator on power quality. The data are unlikely to be altered by third parties due to user participation in the network.

Literature review

There are many applications for blockchain networks in power systems. Below are three recently developed blockchain applications.

Pipattanasomporn *et al.* (2018) created a blockchain-based platform that handles solar power exchange. The platform's implementation is, as the authors call it, "lab-scale". Their paper addresses a blockchain network application for solar electricity exchange between participants using Hyperledger, an open-source platform created by Linux. This application lists the participants, assets, and transactions needed to establish the blockchain-based network to track solar PV sales along with the smart contract, use cases, and implementation. According to the authors, there are several blockchain pilots in the energy sector. These pilots focus on commercial, legal, and financial aspects. However, there is no detailed knowledge on how to implement a blockchain-based trading platform.

Ai *et al.* (2020) analyze the challenges of centralized energy charging in the traditional context, which do not apply to using the settlement of distributed energy transactions. The authors do this to address the inconsistency issues with information in the microgrid, the difficulty in establishing a trust system, energy losses, and costs due to pre-sale. Likewise, this study proposes a blockchain-based asynchronous settlement system for microgrid energy transactions. Their experiments showed that this system solves the aforementioned problems well and meets the requirements of practical applications.

Hussain *et al.* (2019) states that the use of blockchain technology for energy trading can eliminate the role of a third-party intermediary entrusted with energy billing. The authors also analyze the feasibility and benefits of using blockchain for smart grids. They developed an Ethereum-based application of blockchain technology for energy trading. They show how blockchain technology can be an excellent alternative to conventional cryptocurrency standards based on third-party trust. The results show that said technology can be successfully implemented to maintain a distributed database for transactions between customers and traders.

METHODOLOGY

This work attempts to integrate the most important measurement data and events delivered in an electrical substation by a relay to a blockchain. This substation is responsible for verifying data authenticity, mining the information, and validating the chain through the computers that make up the network. All the development work is conducted in the Python programming language. Figure 1 presents the flow diagram of the implemented methodology. Each of the stages of the methodology is described below.

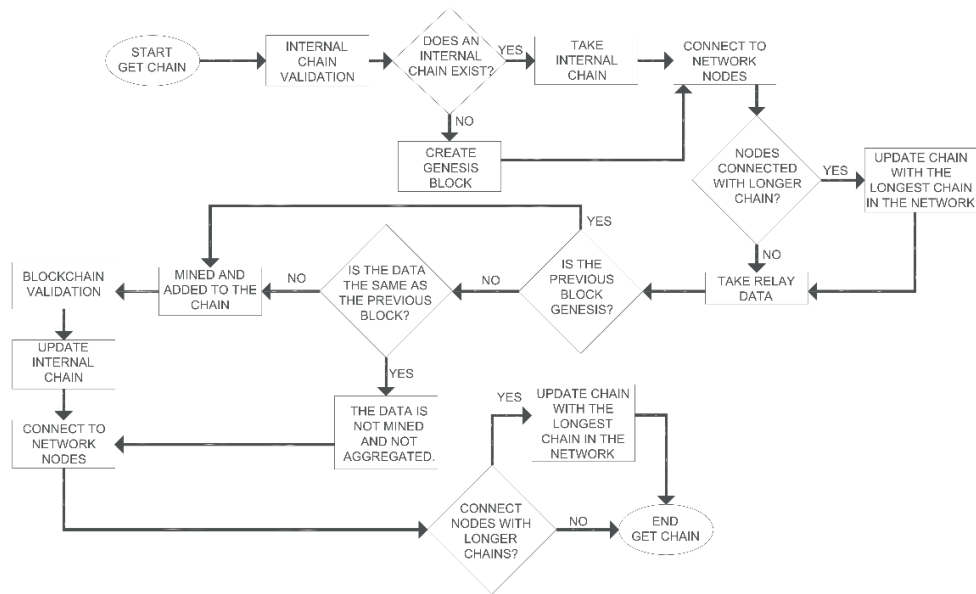


Figure 1. Algorithm development methodology

Source: Authors

Simulated relay data

First, a Python file is created which is responsible for simulating the average and event data commonly obtained from a relay. The Python file also displays the collected data date, the device name, the substation, the events recorded, and measurements such as phase voltage, line currents, frequency, power factor, and active and reactive power, all with their corresponding timestamp.

These events are captured with 10-second intervals between them. They are also stored in a plain text file every minute in a folder on a server. Figure 2 shows the results of the data. The algorithm implemented in the simulation is presented in Algorithm 1.

02 07 2021 18 11
Relay 1
Bogotá Electrical Substation - U.D.

Openings caused by faults: 22.0

Time Stamp;28 09 2021 07 01 18;Vab;115.314;Vbc;116.048;Vca;116.293;Ia;26.082;Ib;30.831;Ic;45.186;F;60.158;FP;0.979;P;236.887;Q;57.743
Time Stamp;28 09 2021 07 01 28;Vab;110.69;Vbc;117.149;Vca;112.374;Ia;41.868;Ib;41.825;Ic;26.057;F;59.877;FP;0.715;P;125.543;Q;615.162
Time Stamp;28 09 2021 07 01 38;Vab;114.175;Vbc;117.479;Vca;111.914;Ia;17.894;Ib;32.324;Ic;49.737;F;59.342;FP;0.513;P;595.397;Q;507.482
Time Stamp;28 09 2021 07 01 48;Vab;116.332;Vbc;113.905;Vca;115.683;Ia;44.307;Ib;0.065;Ic;29.083;F;59.067;FP;0.81;P;122.33;Q;486.381
Time Stamp;28 09 2021 07 01 58;Vab;112.672;Vbc;115.69;Vca;109.112;Ia;19.199;Ib;22.847;Ic;1.307;F;59.477;FP;0.998;P;155.695;Q;50.381
Time Stamp;28 09 2021 07 02 08;Vab;110.577;Vbc;112.259;Vca;112.543;Ia;37.693;Ib;12.535;Ic;29.003;F;60.857;FP;0.306;P;175.923;Q;392.21

Figure 2. Plain text data obtained by simulation of a relay simulation of a relay

Source: Authors

Algorithm 1. Structure of the code implemented for data simulation

Algorithm 1. Data simulation

#First Cicle

While K is greater than 1:

T_general = (current day, month, year, hour and minute)

We create a file called CAPTURING.txt

Header = (Relay name and Electrical Substation name)

Faults = (We indicate how many faults were generated in the last minute indicated by a random number between 0 and 28)

Finally, we write the value of the variable "**Header**" and "**Faults**" in the .TXT file.

#Second Cicle

While T_general is equal to T_date:

T_date = (current day, month, year, hour and minute)

#This guarantees that, when there is a time greater than one minute, the program stops.

Vab = Random value between 109 and 118 taking 3 decimal places

Vbc = Random value between 109 and 118 taking 3 decimal places

Vca = Random value between 109 and 118 taking 3 decimal places

Voltage = **Vab**, **Vbc** and **Vca**

Ia = Random value between 0 and 50 taking 3 decimal places

Ib = Random value between 0 and 50 taking 3 decimal places

Ic = Random value between 0 and 50 taking 3 decimal places

Current = **Ia**, **Ib** and **Ic**

Freq = Random value between 59 and 61 taking 3 decimal places

FP = Random value between 0 and 1 taking 3 decimal places

Pt = Random value between 0 and 700 taking 3 decimal places

Qt = Random value between 0 and 700 taking 3 decimal places

We open the created text document and paste the variables **Voltage**, **Current**, **Freq**, **FP**, **Pt** and **Qt**.

We wait 10 seconds

We look for the path of the text file and modify its name to **T_general**.

When the time increases by one minute, the second cycle is completed, and the new time is plus one minute. This generates a new text file. The first cycle never ends, since the variable K is never modified and is always 1.

Source: Authors

Internal chain update

A Python file is created for each server that connects to the network that will host the code and is responsible for the nodes' connection with the blockchain. This file will establish the data to be added along with the corresponding validation. As a first step, there will be a verification and internal update of the node blockchain cable. Then, programmed functions will update the blockchain data, such as its length. Finally, the algorithm will store the data.

Connection to blockchain nodes

After the internal update of the chain, validation is performed on the nodes, which are previously registered in the network, to verify the connection of the servers to the network. There, the IP address of the nodes and their corresponding work port must be indicated. The algorithm is in charge of verifying, server by server, whether the server is connected or not. If there is no communication with one of the servers, the algorithm stops and does not perform more actions until all the nodes are connected.

Likewise, in this blockchain, four nodes were simulated in the same physical device, where each one has the same IP address but a different port. Table 1 shows the addressing of the servers.

Table 1. Addressing the blockchain nodes

Server	IP Address	Local work port
1	127.0.0.1	5004
2	127.0.0.1	5005
3	127.0.0.1	5006
4	127.0.0.1	5007

Source: Authors

Longest system chain update

Once all the network nodes are connected to the current server, the algorithm validates the longest chain for each node. This process is carried out by asking each linked server its chain and its length. If any of the node's chains are longer than the one in the algorithm, the network's longest chain replaces it. If the chain in the algorithm-system is the longest, the next node is asked for a longer chain until the list of connected servers is completed.

In addition, this method allows knowing if the algorithm has the longest chain; if the algorithm does not have the longest chain, it is necessary to update the algorithm-data internally by changing the current chain by the longest blockchain chain.

When the verification is done, the algorithm prints a message indicating if it has the longest chain or if the chain has been replaced. One of the following three cases may take place.

- *Case 1 - New chain:* No chain has been created by neither the other nodes nor the algorithm, so a genesis block is established.
- *Case 2 - Existing internal chain:* This case starts from the previously developed chain. The genesis block is the first chain link; the other links correspond to the data mining of the relay either by the nodes created by the algorithm or by the other system nodes.
- *Case 3 - Existing external chain:* This case starts from the previous chain developed by another node. The genesis block is the first chain link; the other links correspond to the data mining of the relay either by the algorithm-nodes or by the other system nodes.

In either case, the algorithm prints the value of the current chain with its links and total length.

Capturing relay data

Once the chain update is finished, the algorithm connects to the relay, defining that the relay will have an IP address and a working port, as shown in Table 2. If the device is not connected, the algorithm will not continue its process.

Table 2. Relay addressing

Device	IP Address	Local work port
Relay 1	127.0.0.1	5010

Source: Authors

Once connected, the values from the file are taken, which are delivered by the relay, and verified with the data obtained from the previous transaction of the chain, where one of the following cases will occur:

- *Case 1 - First data in the chain:* It is possible that the blockchain cable still has no information, so the first link created is the genesis block, which contains the number of the section in which is the timestamp of its creation (section is 1 as it is the first link). PROF represents the consensus by which the blockchain transaction was validated and included. The PREVIOUS_HASH defines the key of the previous link (which, in the case of the genesis block, is 0 since it does not have any previous transaction). Finally, the value of the transaction is represented as a "Genesis Block". After validating the information from the previous transaction, the algorithm-system adds the first blockchain link. This link contains relay data. Then, the algorithm continues with the data mining and validation of the chain.
- *Case 2 - Adding new data to an existing chain:* It is probable that the chain has more than two created sections (genesis and another block), so, the previous block is

validated through the relay time data. Then, the data is compared with the relay time data to be added to the chain; if the information is not repeated when compared, the new relay record is added to a new link.

- *Case 3 - Captured data = Data from the previous block:* Based on Case 2 of this section, where there is a chain with two or more blocks created, the relay time data from the previous segment is considered and compared with the current capture time, but the algorithm observes that the new data is the same as the previous one. If this is true, the algorithm cannot add the 'new data', since the server would be adding repeated values in the time delivered by the relay to the blockchain.

This comparison guarantees that any server that wants to add a new piece of data to the blockchain compares whether it is repeated data or not. If the data is repeated, it will not be added, and the network chain will be updated.

RESULTS

The results obtained for each of the stages of the previously described methodology are shown below.

Section transaction and mining

Once validated, by relaying non-repetition data to add, the algorithm proceeds with the new transaction and the mining of the new block. Figure 3 created blockchain transactions and show how a new transaction is added to one section. In this case, Figure 3 displays the second one ("1_index":2), where the added equipment ("A. Equipment") is observed, which node adds that new transaction ("B. Server"), and what information the server brings from the relay ("C. Information").

```

{
  "1_length": 5,
  "2_chain": [
    {
      "1_Index": 1,
      "2_Estampa_de_tiempo": "2021-09-28 06:58:11.088496",
      "3_proof": 1,
      "4_previous_hash": "0",
      "5_transactions": [
        "Bloque Génesis"
      ]
    },
    {
      "1_Index": 2,
      "2_Estampa_de_tiempo": "2021-09-28 06:59:30.842493",
      "3_proof": 533,
      "4_previous_hash": "81c5919c6a13b571179148a20f58a82e82f3b3a5a8fa2d75236d67907cc0075b",
      "5_transactions": [
        {
          "A.
          Equipo": "Rele_1",
          "B.
          Servidor": "Red_BKC_NODO_5005",
          "C.
          Informacion": "28 09 2021 06 58\nRelay 1\nBogotá Electrical Substation - U.D.\n\n0penings caused by faults: 15.0\n\nTi

```

Figure 3. Created blockchain transactions

Source: Authors

When adding the transaction, the block employs the previous block key or "Previous_hash" and the consensus protocol or PROF. Then, the algorithm performs the proof of work or "Proof_of_of_work." This proof confirms the transaction and produces a new block in the chain through the SHA256 encryption and validation algorithm, an algorithm that uses the "Previous_hash" to encrypt the recent key of the block to be added.

Publication of the chain on the blockchain

When completing the block validation and mining, the internal chain of the node is updated with the new block and reconnected to the other nodes, thus leaving a precedent that there is a recent blockchain publication.

Multiple connections between network nodes

According to Figure 4, nodes can connect between each other in the local network. Likewise, nodes update the information that each one has according to the longest chain available in the network.

Information decentralization

The relay-obtained data is similar to the one obtained from a substation or even a better power system. These results prove that not only does one server have the information in a centralized way, as commonly handled, but each server in the network may handle the same data and update according to the executed tests.

Greater control of information

Everyone can interrogate the relay and define whether the information delivered is already in the chain or constitutes new data. Likewise, this algorithm prevents the addition of altered information from the devices to be monitored.

Performance of the device used as a server

When executing the steps described in the methodology and the longer the chain is, the equipment used to run the algorithm presents delays in its update; the algorithm's processor increases the temperature and turns on the cooling fans.

Execution times for network update

The update of each server with the blockchain chain is delayed by about 10 seconds, considering that the only processes running on the machine are the algorithm and Postman (the graphical interface that allows visualizing the network values and communicating with the algorithm).

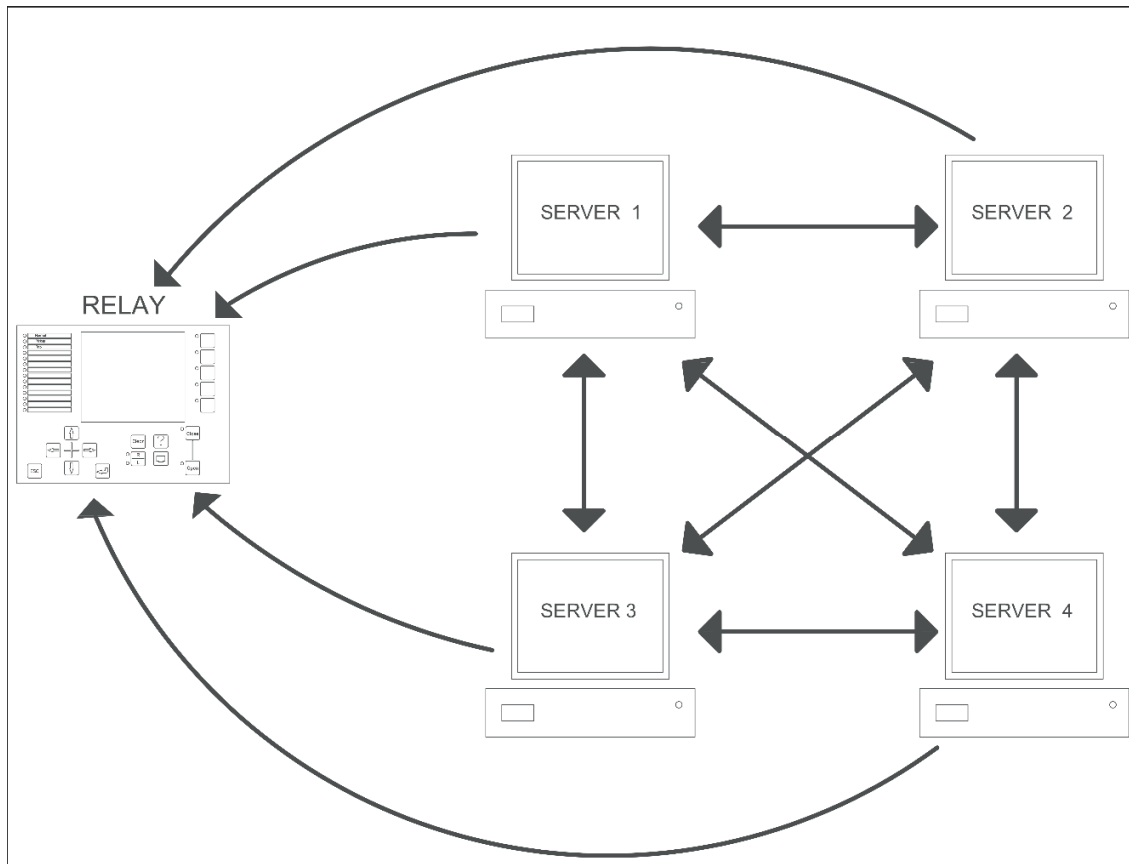


Figure 4. Connecting equipment on the blockchain

Source: Authors

CONCLUSION

This work developed a blockchain algorithm capable of connecting multiple nodes to the network and decentralizing the information. In this algorithm, additional equipment such as a relay can be contacted. Likewise, the servers connected to the network interrogate the device, collecting information on measurements and events that have occurred during the last minute, thus achieving the capture of such information and adding it to the chain without repetition.

In addition, it is relevant to highlight the decentralization of the information compared to conventional event recording methods. This allows greater control of the information and much faster and more reliable access by network users to the stored data. Moreover, to

develop a blockchain with a high amount of equipment, it is recommended to use machines with high processing capabilities, higher RAMs, and modules that intervene in the information linking process to the equipment with the greater capacity, both in storage and performance. These characteristics allow the algorithm to run smoothly and in the shortest possible time.

In short, using this type of technology in an environment where information must be transported from the substation to a control center and give orders to the equipment is not recommended, since the execution times of the commands can delay the proper operation of the algorithm. However, it is advisable to manage the algorithm as an extensive decentralized database with few cybersecurity vulnerabilities.

FUTURE WORK

Over time, more data could be integrated, which would depend on the protocols handled by the substation. For instance, IEC61850 could carry data on measurements, positions, and other electrical and mechanical cell variables on time by dealing with more precise standards for information handling.

In the same way, the possible creation of a graphical interface allows a better user experience when reading data and performing validation.

Finally, the expansion of this algorithm into a higher network could be conducted as future research, with the integration of a minimum of 15 remotely connected computers through the Internet, thus allowing participation at an international level.

REFERENCES

Ai, S., Hu, D., Zhang, T., Jiang, Y., Rong, C., & Cao, J. (2020, May 25-28). *Blockchain based Power Transaction Asynchronous Settlement System* [Conference presentation]. IEEE Vehicular Technology Conference, Antwerp, Belgium.
<https://doi.org/10.1109/VTC2020-Spring48590.2020.9129593>

Alam, A., Islam, M. T., & Ferdous, A. (2019, February 7-9). *Towards Blockchain-based Electricity Trading System and Cyber Resilient Microgrids* [Conference presentation]. 2nd International Conference on Electrical, Computer and Communication Engineering, Cox'sBazar, Bangladesh. <https://doi.org/10.1109/ECACE.2019.8679442>

Foti, M., Mavromatis, C., & Vavalis, M. (2021). Decentralized blockchain-based consensus for Optimal Power Flow solutions. *Applied Energy*, 283, 116100.
<https://doi.org/10.1016/j.apenergy.2020.116100>

Hussain, S. M. S., Farooq, S. M., & Ustun, T. S. (2019, March 22-23). *Implementation of Blockchain technology for Energy Trading with Smart Meters* [Conference presentation]. 2019 Innovations in Power and Advanced Computing Technologies, Vellore, India. <https://doi.org/10.1109/i-PACT44901.2019.8960243>

Leka, E., Selimi, B., & Lamani, L. (2019, September 19-20). *Systematic Literature Review of Blockchain Applications: Smart Contracts* [Conference presentation]. 2019 International Conference on Information Technologies, Varna, Bulgaria.
<https://doi.org/10.1109/InfoTech.2019.8860872>

Pandey, S., Ojha, G., Shrestha, B., & Kumar, R. (2019, May 14-17). *Blocksim: A practical simulation tool for optimal network design, stability and planning* [Conference presentation]. 2019 IEEE International Conference on Blockchain and Cryptocurrency, Seoul, South Korea. <https://doi.org/10.1109/BLOC.2019.8751320>

Pava, R., Pérez-Castillo, J. N., & Niño-Vásquez, L. F. (2021). Perspectiva para el uso del modelo P6 de atención en salud bajo un escenario soportado en IoT y blockchain. *Tecnura*, 25(67 SE-Revisión), 112-130. <https://doi.org/10.14483/22487638.16159>

Pipattanasomporn, M., Kuzlu, M., & Rahman, S. (2018, October 24-26). *A Blockchain-based Platform for Exchange of Solar Energy: Laboratory-scale Implementation* [Conference presentation]. 2018 International Conference and Utility Exhibition on Green Energy for Sustainable Development, Phuket, Thailand. <https://doi.org/10.23919/ICUE-GESD.2018.8635679>

Wang, N., Zhou, X., Lu, X., Guan, Z., Wu, L., Du, X., & Guizani, M. (2019). When energy trading meets blockchain in electrical power system: The state of the art. *Applied Sciences*, 9(8), 1561. <https://doi.org/10.3390/app9081561>

Wang, R., He, J., Liu, C., Li, Q., Tsai, W. T., & Deng, E. (2019, November 23-25). *A Privacy-Aware PKI System Based on Permissioned Blockchains* [Conference presentation]. IEEE International Conference on Software Engineering and Service Sciences, Beijing, China. <https://doi.org/10.1109/ICSESS.2018.8663738>

Yang, Q., Wang, H., Wang, T., Zhang, S., Wu, X., & Wang, H. (2021). Blockchain-based decentralized energy management platform for residential distributed energy resources in a virtual power plant. *Applied Energy*, 294, 117026.

<https://doi.org/10.1016/j.apenergy.2021.117026>

Yu, S., Lv, K., Shao, Z., Guo, Y., Zou, J., & Zhang, B. (2019, August 15-17). *A High Performance Blockchain Platform for Intelligent Devices* [Conference presentation]. 2018 1st IEEE International Conference on Hot Information-Centric Networking, Shenzhen, China. <https://doi.org/10.1109/HOTICN.2018.8606017>