4-2019

Blockchain for Power Grids

Christian Banks
United States Military Academy, Christian.Banks@westpoint.edu

Samuel Kim
United States Military Academy, Samuel.Kim@westpoiint.edu

Michael Neposchlan
United States Military Academy, Michael.Neposchlan@westpoint.edu

Nicholas Velez
United States Military Academy, Nicholas.Velez@westpoint.edu

K.J Duncan
United States Military Academy, katherine.duncan@westpoint.edu

See next page for additional authors

Authors
Christian Banks, Samuel Kim, Michael Neposchlan, Nicholas Velez, K.J Duncan, John James, Aaron St. Leger, and Daniel Hawthorne

Blockchain for Power Grids

Christian Banks, Samuel Kim, Michael Neposchlan, Nicholas Velez, K.J. Duncan, J. James, A. St. Leger, D. Hawthorne
Department of Electrical Engineering and Computer Science
United States Military Academy
West Point, NY
Christian.Banks@westpoint.edu

***Abstract*— Sharing information is an important part of regulating and maintaining efficient and safe power grids. This project's goal is to develop a way of using blockchain technology to share transaction information among different power grids in a secure, controlled, monitored, and efficient manner. The biggest concern regarding the data is integrity. By leveraging blockchain technology, the data will be reliable and resilient to attacks, such as man-in-the-middle and data spoofing attacks. The Hyperledger Fabric implementation provides a permissioned network in which power grids will act as nodes that maintain ledger information. By using a distributed ledger to validate transactions through the process of consensus, the system can share information in a manner that is more secure and transparent than traditional information sharing systems in which data is less secure and takes longer to validate. The additional layers of security and speed that Hyperledger technology provides help to prevent issues, such as power grid failures, that could stem from the latency or integrity issues involved with traditional methods of validating, processing, and reacting to shared data.**

***Keywords*— Blockchain, Hyperledger fabric**

I. INTRODUCTION

Sharing information across large networks poses many issues and potential risks involving security and usability. For the purpose of sharing information between microgrids, our team has posited that, while availability, integrity and confidentiality of data are important, the application of blockchain technology to also achieve non-repudiation of shared data will substantially improve resilience of microgrids to man-in-the-middle and data spoofing attacks. Blockchain architectures provide a variety of solutions to sharing information. Of the many different blockchain solutions currently available, Hyperledger provides superior scalability, modularity, and security [1]. For this project, a Hyperledger Fabric network is being implemented to facilitate the sharing of information among various microgrids in a way that will potentially be scalable to much larger industry grids.

II. BACKGROUND

A. *Blockchain*

Blockchain is a group of transactions that are linked to their previous modification on a specific channel [2]. The chain is a log that contains the transactions of all previous 'blocks' for that particular chain. When a new block is appended unto the chain, the transaction from that previous block is also appended [2].

A large feature of a blockchain is the use of a distributed ledger [2]. Blockchain ledgers are often decentralized because each person on the network is working with their own replication of the block [2]. Utilizing a decentralized ledger helps add security to the network as all information is not funneling into one node. The ledger contains two characteristics, a world state and a blockchain [3].

A blockchain was explained in the previous paragraph, however, a world state is a database that keeps the values of a ledger state at the current time [3]. This allows for the pulling of the current state at any amount of time without having to sweep a log [3].

Blockchain is being applied to a variety of energy sector uses but not in the area of control of power grid dynamics. That is, existing approaches for sharing data do not exploit the use of permissioned blockchains to create irrefutable ledgers of state estimation and control data being shared among participating institutions. Such irrefutability is needed for use in generating control solution to maintain stability of the power generation, transmission, and distribution processes. One review article, [4], does indicate that there is considerable interest in a variety of start-up companies applying blockchain technology to activities such as:

- improving existing markets for trading electricity or even create new ones,

- tracking the production of clean energy,

- making it easier to pay for charging electric vehicles, and

- managing customer appliances.

Indeed, one widely-reported application is in improving accountability of energy sector financial transactions. The Brooklyn Microgrid, [5], is developing a local community microgrid in which participants can engage in a sustainable energy network and choose their preferred energy sources, locally. Brooklyn Microgrid uses the ledger accountability

This work was supported by the Office of Naval Research. The views expressed herein are those of the authors and do not purport to reflect the position of the United States Military Academy, the Department of the Army, or the Department of Defense.

effort to support the financial aspects of power generation, transmission, and distribution to meet demand response energy balancing of microgrids. Lo3 Energy, [6] , is providing the blockchain technology to enable Brooklyn Microgrid to solve the financial aspects of energy production and distribution.

Unlike Brooklyn Microgrid, our project is interested in the sharing of state estimation and control data to maintain the stability of the dynamics of the generation, transmission, and distribution process in order to meet demand response energy balancing of microgrids. We believe that no other effort has yet tried to use blockchain to resolve the issue of non-refutability of shared data for use in wide area control. That is, to our knowledge, our effort is the first to experiment with use of blockchain to share power grid state estimation and control data in order to improve reliability and resiliency of power grid control solutions.

B. Hyperledger fabric

Hyperledger fabric is a type of blockchain that involves a collaborative approach to community sharing, property rights and the development of standards over time [7]. Like other Blockchain implementations, Hyperledger has a ledger and utilizes smart contracts [7].

III. DESIGN

A. Powergrid Network

Our planning and design for our project relied on a previously completed Wide Area Monitoring and Control system (WAMC) [8]. In this project, an existing smart grid testbed was reported [8], the specific foundations of the testbed are explained. Seven Phasor Measurement Units (PMUs) provide phasor measurements for each power system bus [8]. All the PMU data is consolidated in a Microsoft SQL database by OpenPDC [8]. How the data is being accessed from the SQL database will be explained later in the paper.

The purpose of this project is to provide a foundation to solve the problem of the data distribution that would be a major driving force in implementing a safe and efficient WAMC in the real world. In the future, WAMC can help to prevent issues and errors that stem from anomalous activity in a power grid. Traditional Supervisory Control and Data Acquisition (SCADA) systems are limited in their capability to detect anomalies in power grids. SCADA provides asynchronous data. Additionally, there is a significant delay, ranging in length from seconds to minutes [9]. Using blockchain technology for power grids could potentially be a solution to these issues seen in SCADA systems.

Currently, our team has completed a two node architecture and were able to get the two nodes to communicate. Referencing the diagram, the two nodes that we have successfully completed reside in the USMA domain. Figure 1 is an example of the final network design in which all six peer nodes will reside.

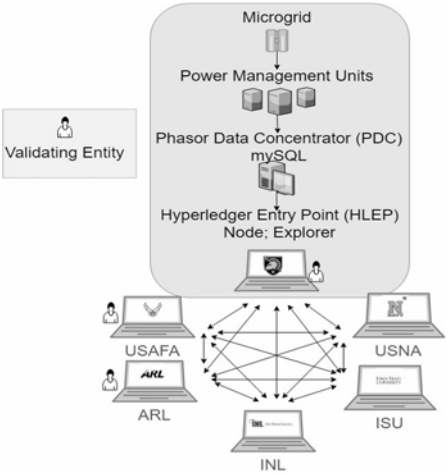

Fig. 1. Final Network Design

B. Internal Network

The process of data retrieval from the node network is explained in our internal network diagram found on Figure 2 below.

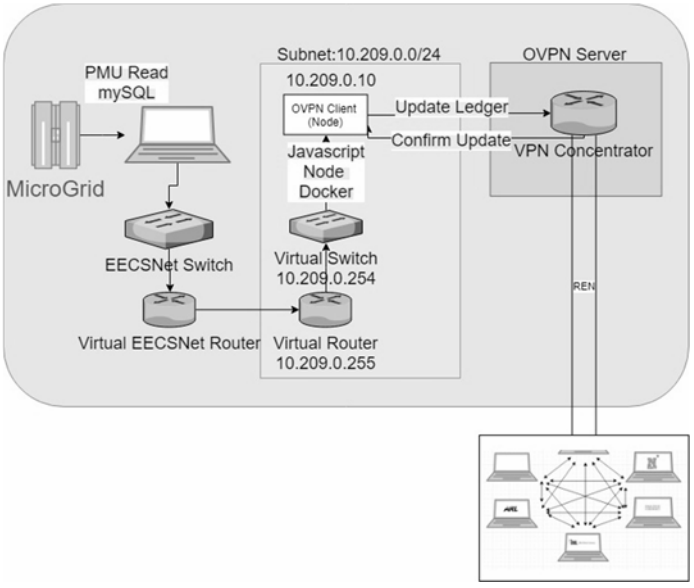

Fig. 2. Internal Network Design

MySQL data from the USMA testbed is read by one of the hosts located at USMA. The data from that PC is read by the database OPVN client (Node). Data is obtained by running a database query algorithm. This algorithm which is explained in the next paragraph. Then from the OPVN client, data will be sent out the VPN concentrator with an update request to the rest of the nodes on the network, and the nodes will confirm the request to update the ledger. The other nodes are then able to connect to the VPN concentrator in order to send confirmation requests back to our client. Once the request is sent back to our

client node validating the update, the ledger state is updated with the appropriate information. Each node on the network will run a similar instance of OPVN client node as seen on Figure 2.

C. Database Query Algorithm

The modules that we used were PyMySQL and JSON. The PyMySQL module enables us to interface with the SQL database [10]. The JSON module acts like and encoder and decoder to help us convert our data into a useable JSON format [11].

```
#!/usr/bin/python3

#PyMySQL
import pymysql.cursors
#JSON for exporting
import json
```

Fig. 3. Python Code: Modules

The process of data retrieval from the node network is explained in our internal network diagram found on Figure 2. The python script first establishes a connection to the testbed SQL database using proper authentication. After successfully connecting, the script obtains information related to PMU data from a text file.

```
try:
    with connection.cursor() as cursor:

        #Opens necessary files
        file = open('someFile.txt','a')
        constraints = open('constraints.txt','r')
        lines = constraints.read().split('\n')
```

Fig. 4. Python Code: Establishing a connection and Script Reading

This file contains unique the identifiers for all PMU measurements and applicable variables necessary for queries. The program then loops through each PMU and queries the SQL database for the latest readings.

```
        #Our loop that runs through the PMUs listed in the constraints file, expandable
        for x in range(16):
            #Creates the applicable variables for our queries
            items = lines[x].split(',')
            pmu = items[0]
            pmuNum = items[1]
            dataType = items[2]
            dataMin = items[3]
            dataMax = items[4]

        #Base SQL Query
        cursor.execute("SELECT Value,tsmID FROM timeseriesmeasurement WHERE SignalID = %s AND Value <= %s AND Value >= %s AND tsmID >=
        result = cursor.fetchone()
```

Fig. 5. Python Code: Looping and SQL Query

The queries are written to a file in readable format. Additionally, each query is printed to the console (for verification the script is working).

```
            #Prints to console
            print(result)

        #Closes all files
        constraints.close()
        file.write("\n")
        file.close()


finally:
    connection.close()
```

Fig. 6. Python Code: Printing the Result, Closing Files, Ending Connection

After completing the above actions, the script closes all files and connections. In order to more easily create historic data for our network, the script is being updated to accommodate JSON formatting.

D. Scalability

Once a successful proof of concept can be run, our framework can be adapted on a larger scale to allow peer nodes to communicate over large distances. Latency would be affected by the changes in the environment. However, as the algorithm improves and is optimized, these effects can be mitigated. Figure 7 shows the six testbeds that will share information from the blockchain.

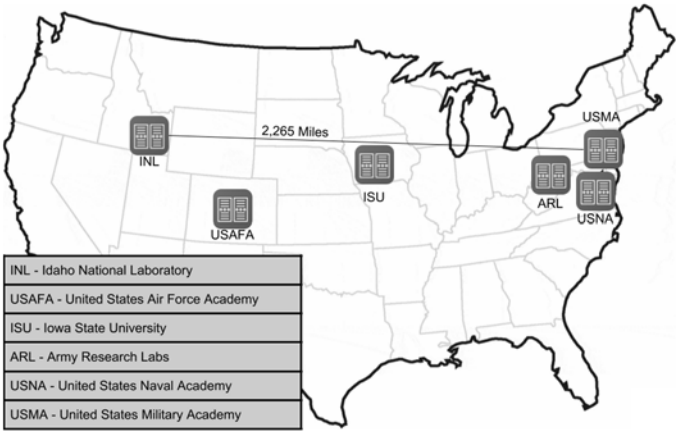

Fig. 7. Testbed Diagram

As observed on Figure 7, latency issues arise due to the long distance between the microgrids in Idaho, Colorado, and Iowa and the microgrids on the east coast. With larger amounts of latency, the potential for the control system to become unstable rises. Our experiment in sharing data will enable recording and analyzing data sharing latencies to determine stability margins of the connected set of microgrids for application of alternative WAMC algorithms. Currently, our team has successfully created a database query algorithm with python with a successful update to accommodate JSON formatting. Furthermore, we were successful in pulling data from the USMA micro grid.

IV. Implementation

A. Database interface

The data that is being shared in this project is collected from PMUs connected to the local power grid. These units collect voltage and current measurements from different parts of the testbed. Through python coding, we are able to successfully pull data from the testbed at West Point. The data includes: current phasors, voltage phasors, frequency, rate of change of frequency and time stamps. Figure 8 is an example of a successful pull from the West Point testbed.

```
PMU 1 voltage Reading: {'Value': 26735.5, 'tsmID': 1745805026}
PMU 2 voltage Reading: {'Value': 26732.3, 'tsmID': 1745805126}
PMU 3 voltage Reading: {'Value': 26735.5, 'tsmID': 1745805034}
PMU 4 voltage Reading: {'Value': 26722.4, 'tsmID': 1745805018}
PMU 5 voltage Reading: {'Value': 26719.0, 'tsmID': 1745805081}
```

Fig. 8. Example of Successful Data Pull from WestPoint MicroGrid

Our code is modified to display what PMU we are accessing, the value of the voltage and the time stamp ID (seen as tsmID in Fig. 8). The voltage reading that displays none, means that the PMU has not been recording or has no data for that timestamp. Our next goal is to use Javascript code to inject our script into the Hyperledger network that we created.

B. Peer Node

For the peer node network, we are currently implementing Hyperledger architecture in a virtual environment. Our team adapted a tutorial in order to create a two node Hyperledger architecture utilizing virtual machines to successful conduct a hello world test. In order to verify the communication amongst the nodes, a CouchDB server was utilized to record all relevant information - which was encrypted through hashing. The Couchdb server also allows for the database to be viewed through a web interface. On the client node, there is an orderer terminal, a command line interface, and a log file being updated in the background. On the manager node, there is a certificate authority server, an orderer, and a Couchdb server.

V. Future Work

The end state for this project is a network with connectivity across six peer nodes including: the United States Military Academy (USMA), the United States Naval Academy (USNA), the United States Air Force Academy (USAFA), Army Research Labs (ARL), Iowa State University (ISU), and Idaho national labs (INL). Each of these nodes represent microgrids that securely communicate with each other and share data using Hyperledger. The validating entities inside of the network are USMA, USAFA and ISU. The validating entities validate transactions and maintain the ledger while ensuring consensus has been achieved on all transactions. Additionally, the latency will be optimized to ensure that data is being transferred and received at the fastest speed possible without bearing a large load on the system.

Future iterations of this project will seek to improve upon the local inefficiency and vulnerability of the SQL database by moving the Hyperledger Nodes closer to the micro grids. By directly interfacing between the Hyperledger nodes and the PDCs, the performance of the network will be further improved and assumed initial integrity of the data will be better protected.

VI. Conclusion

The implications of this technology apply to a global level of communication amongst power grids that rely on one another to supplement power when needed. A Hyperledger architecture supports a secure method of communication in a highly scalable network. This process expedites the sharing of information which results in the prevention of potential blackouts relevant to power grids.

This paper begins with the basics of blockchain and Hyperledger Fabric to help provide context the project: Blockchain for powergrid. First, the design of the powergrid network, and internal network, database query algorithm and scalability is discussed. The implementation of our system and the current results were also discussed. Lastly, recommendations for future work were given, such as upscaling our project past communication between schools and movement to communication between larger scale power grids. By expanding upon our design we can create a system that allows microgrids to communicate relevant data across many peer nodes. We have established a virtual private network (VPN) connection with Idaho National Laboratory (INL) and are ready to begin sharing microgrid data between the two sites as well as measuring latencies associated with the data sharing process. We will be able to demonstrate that the application of blockchain technology will achieve non-repudiation of shared data which will substantially decrease the likelihood of man-in-the-middle and data spoofing attacks on data shared using blockchain technology.

References

[1] https://www.hyperledger.org/projects/fabric
[2] https://hyperledger-fabric.readthedocs.io/en/release-1.2/glossary.html
[3] https://hyperledger-fabric.readthedocs.io/en/release-1.2/ledger/ledger.html
[4] https://www.cfr.org/report/applying-blockchain-technology-electric-power-systems (accessed on 15 Feb, 2019)
[5] https://www.brooklyn.energy/ (accessed on 15 Feb, 2019)
[6] https://lo3energy.com/ (accessed on 15 Feb, 2019)
[7] https://hyperledger-fabric.readthedocs.io/en/release-1.2/blockchain.html
[8] A. St. Leger, J. Spruce, T. Banwell, and M. Collins, “Smart grid testbed

for Wide-Area Monitoring and Control systems," *2016 IEEE/PES Transmission and Distribution Conference and Exposition (T&D)*, pp. 1-5, 2016.

[9] S. J. Matthews and A. St. Leger, "Leveraging single board computers for anomaly detection in the smart grid," *2017 IEEE 8th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON)*, pp 1-7, 2017.

[10] "User Guide," *Examples - PyMySQL 0.7.2 documentation*. [Online]. Available: https://pymysql.readthedocs.io/en/latest/user/index.html. [Accessed: 05-Feb-2019].

[11] "18.2. json - JSON encoder and decoder," *16.2. threading - Higher-level threading interface - Python 2.7.15 documentation*. [Online]. Available: https://docs.python.org/2/library/json.html. [Accessed: 05-Feb-2019].