Permissioned Blockchain-Driven Internet of Things Gateway Using Bluetooth Low Energy

Marc Jayson Baucas and Petros Spachos
School of Engineering, University of Guelph, Guelph, ON, Canada

***Abstract*—An Internet of Things (IoT) network can have different components such as servers, gateways, and the end devices. An important source of performance constraint in such an IoT network is found in the limitations of its gateway. The capability of a gateway can dictate the effectiveness of a network and its services. The capacity, power consumption, and security of an IoT gateway are revealed as sources of network bottlenecks and service constraints. Blockchain technology can create a decentralized structure that can offload these strains. To unify these nodes as gateways under the same network, we need an effective means of communication. This paper proposes a setup that makes use of the decentralized capabilities of private blockchain technology partnered with the low-powered and secure connection of Bluetooth Low Energy (BLE). This provides a more secure means of wireless communication and prevents the nodes from being concentrated within an area. The architecture was compared against a standard WiFi network (2.4GHz) to prove its feasibility in effectively carrying out its functionality. In an experiment that used 4 gateway nodes, BLE proved to be more feasible than WiFi by yielding a better verification packet rate of 14 per minute compared to its counterpart that measured 4 per minute. Also, it showed to be more efficient in terms of power consumption with an average of 1095.40 mW, while the WiFi setup was measured to be 1191.83 mW. These results show promise in using BLE paired with blockchain technology to solve the capacity, power and security issues in IoT networks.**

I. Introduction

Internet of Things (IoT) network allows devices to connect to a server and access information world-wide [1]. In this way, the connected device is given the ability to request and receive services or data from any provider connected to the same network. A common structure of an IoT network divides it into three sections; the online storage or server, the devices or clients, and the hub or gateway [2]. The server is the central data storage of the IoT network. Its structure ranges from a simple database table to a large-scoped cloud storage. The clients of a network are the devices that connect to the server and can contribute or benefit from the information stored in the database. The gateway serves as a bridge between the server and the client. Being the only bridge for a device to obtain information from the server, the gateway becomes a potential bottleneck to any IoT service. Internet service providers advertise their services based on the speed of their modems or routers which is an alternate term for their gateways. This is due to the gateway being the one that dictates the quality of the data exchange between the server and the client. Therefore, in terms of hardware architecture, the limits of a gateway are defined by its design.

An issue that limits a network through the gateway is its data throughput/ capacity. The download and upload speeds of a network's service is often due to the capabilities of its modem. A network needs a gateway design that can manage multiple devices demanding services simultaneously. A second issue is power consumption. Wireless transmissions require large amounts of power to be carried out by a modem. A setup that requires multiple modems will result in significant power requirements. A third and important issue is security. Some setups that use multiple modems to cover an area run into interference due to the centralized nature of the network. Due to the distance, a wireless signal needs to cover, a network is more prone to attacks.

This paper proposes a setup that makes use of the decentralized capabilities of private blockchain technology partnered with the low-powered and secure connection of Bluetooth Low Energy (BLE). With these two technologies, we create a decentralized network configuration that addresses these three issues and improves the overall performance of IoT networks.

The rest of this paper is organized as follows: A discussion of the two main technologies used in formulating the solution is in Section II. Section III describes the design and methodology of the solution followed by Section IV where the feasibility of the design in supporting a gateway is evaluated. Finally, conclusions are given in Section V.

II. Background

It is important to discuss the different concepts used in the design, before presenting the implementation. The major technologies used in this work are blockchain and BLE.

A. Blockchain

Blockchain technology is a type of linked list data structure that allows the storage of transactions and other forms of information in a manner that is decentralized, transparent, immutable and automated [3]. In this work, the main focus is on the decentralized, immutable, and automated aspects of the technology. Decentralization is a type of network architecture that has node within the network given equal authorization. This architecture adapts a specific type of network diplomacy where no node is given higher control or authority over the other nodes in the system [4]. Most IoT network architectures implement a centralized gateway when managing their devices. However, having a centralized hub creates a potential bottleneck for any IoT network architecture. By using a decentralized design, the load that puts a strain on

978-1-7281-5089-5/20/$31.00 ©2020 IEEE

any hub can be distributed to other nodes that have the same capabilities in terms of provided services [5].

A centralized gateway shows an obvious focus for network attacks. By distributing the load between nodes, the security strains of a single entity are reduced. Blockchain technology has proven to be an excellent architecture for decentralized implementations. With multiple instances of a blockchain being exposed in a network, server tampering appears to be a bigger risk. However, with the immutability of blockchains, data within it can be protected from any unwanted changes or modifications. Blocks that have been previously placed onto the blockchain will be difficult to replace due to each link being protected by a one-way hashing algorithm [3]. With this, the blockchain becomes the main storage of vital information, which is the foundation of the network.

By building on the decentralization of the network using blockchain technology, each node is given an exact copy of the blockchain. This allows the establishment of a synchronized consensus system. A consensus system allows the network to decide on the action through a democratic votation. Any node within the network that has a copy of the most common version of the blockchain is allowed to vote for the network's verdict over the proposed action [6]. There are two types of blockchains; public and permissioned [7]. Public blockchains allow any node that has access to a public key to take part in the voting process as long as its hardware is capable of providing its proof work. Permissioned blockchains are different because the nodes that are allowed to vote in this type are already pre-authorized thus not requiring any proof of work.

Most blockchain implementations for IoT networks use the public approach. However, based on the intentions of this paper, only pre-defined nodes should be allowed to vote on which blocks to append to the blockchain. This means that the paper will be using a permissioned or private approach to blockchain technology. With this, security within the design is more inclined to authorization than processing power. Most public blockchains rely on proof of work to maintain security. In permissioned blockchains, authorization through identification and verification is used. This improves the overall speed of the design by removing the additional checks that are usually done in public blockchains that require massive amounts of processing power.

B. Bluetooth Low Energy (BLE)

A standard Bluetooth protocol runs on a 2.4 GHz radio band. This allows it to operate at a range of around 20 m depending on the type of Bluetooth device. Bluetooth signals can also penetrate solid objects, therefore, it is not constrained by the Line of Sight (LoS) [8]. Succeeding this protocol is BLE which serves as an extension to the Bluetooth classic [9].

Also known as Bluetooth smart, BLE is a protocol that focuses on conserving power when transmitting data over a wireless medium [10]. Due to this, BLE protocols are most commonly used for low-power wireless sensors. Large data is rarely transmitted using this protocol. This is due to the low-power constraints of the devices that use the protocol. Smaller groups of data such as messages are formed into packets that are shared across paired devices.

In terms of how BLE devices pair, BLE devices implement an authorization protocol that allows communication between these devices to remain secure. Before two or more devices can share data between each other via BLE, the devices should first bond using a selected pairing scheme. There are four different pairing schemes [11]; Numeric comparison, Passkey entry, Out of band, and Just work. A numeric comparison is when the pairing devices generate a shared six digit number and have it displayed on the device's screens. In turn, the owners of the devices can authenticate and check if the right devices are connected based on the displayed number. Passkey entry is when one device sets a password. If this password is provided by the other device, the two devices are paired. Out of band takes the generated key and shares it with the other device using a different method apart from BLE. Then, the other device can verify the pairing accordingly once the key is received. Lastly, Just work takes both devices and have them exchange information to generate a shared temporary key to enable pairing. Each method allows the device to verify any request for a connection from another device. This adds a layer of security between each device that is actively communicating with each other.

BLE-based devices such as beacons and Bluetooth smartphones have been widely used in IoT networks [12], [13]. Some common applications of BLE in IoT networks are on sensing networks that use low-powered devices to collect data [14]. The data that is collected is then sent to a central hub for processing. Most of these network designs are star topologies due to its central unit surrounded by worker units. However, the intended architecture in this paper plans to use a decentralized network that is closer to mesh topology.

Multiple implementations propose a mesh topology implementation of BLE technology for general data communications [15]–[17]. Their implementation inspired the BLE framework implemented within this work. Though our proposal has a similar blockchain concept, we use BLE differently. We use BLE to connect the blockchain nodes due to being low-powered and secure due to its authorization and verification protocols. Having multiple nodes that are continuously communicating with each other can be strenuous to the network. However, by taking advantage of the low-powered aspect of BLE, the design can run its consensus system without using too much power. In addition, this design can benefit from the security layer that BLE communication provides in terms of data entry authorization and device verification.

Communication between the gateway nodes is what drives the design proposed in this paper. We chose BLE because it provides a pairing mechanism that makes this possible while maintaining a secure point-to-point medium for wireless communication.

III. Methodology

This section provides an overview of the design as well as a discussion of how the proposed configuration was set up.

A. *Design overview*

We propose an IoT gateway design that addresses issues of network gateways in terms of throughput, security, and power consumption. Most gateway designs are constricted by the number of devices that they can handle at a given time. The data rate of the gateway is oftentimes affected by the number of users currently connected to the network. Improvements in terms of hardware have their limits. In this case, accommodating more devices can be too demanding in terms of the resources it needs. The delegation of work can prove to be a useful method when it comes to dealing with multiple users. By having more than one station that can provide the same service, a gateway's performance is increased. This proposal incorporates the design by creating multiple instances of the gateway and chaining them together to increase its overall capacity. By having multiple units of the gateway that provide the same service, the throughput issues created by a centralized network can be addressed.

This decentralized approach was implemented by using permissioned blockchain technology. With the decentralized structure, the network can achieve a level of parallelism that can alleviate the constraints on a standard centralized gateway. This creates a system that can automate itself in a way similar to the consensus protocol that blockchain technology is capable of implementing. This protocol allows a systematic way of checking if each node is still fully functional and secure. In addition, by having the same private ledger shared among each gateway, it will be easier to keep track of the history of each registered node. This allows us to make sure that the placement of each node caters to the demand of the service. The dynamic architecture is then created through behaviour and history analysis that blockchains enable. As a result, the network will be able to cater to more devices by having more secure entry points via the multiple gateway nodes. However, to make sure that these nodes are capable of working together in unison, we need a means of communication that is secure, low-powered, and efficient.

This design requirement led us to BLE. We chose this technology to minimize the power drawn by each node while making more secure communications. Continuous communication is key to this architecture. The pairing mechanism used by BLE can make this communication easier to manage. By pairing all the nodes to each other, a decentralized topology is established where one node can equally communicate with another. One of the advantages of BLE is that communication cannot be established unless pairing is initiated and accepted. This creates a secure point-to-point medium that is unique to each node pair. Therefore, each node can be securely chained to one another through a BLE protocol. BLE communication can be tracked for each device through a special Bluetooth Media Access Control (MAC) address so any external device that attempts to eavesdrop can easily be filtered out. This creates a secure avenue for gateways to effectively communicate with one another to maintain a decentralized control over the network. With BLE, the scope of the network is also increased. By taking advantage of the range provided by BLE, each node can be strategically placed apart from each other while maintaining stable communication.

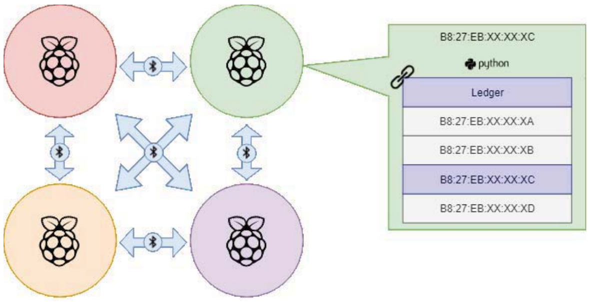

Fig. 1: Experimental gateway setup.

B. *Implementation*

The proposed design has the following implementation.

1) Physical gateway setup: The setup was created using Raspberry Pi 3 that will function as each individual gateway node. This decision was made due to the device's built-in Bluetooth 4.0 compatibility and services which allows the establishment of BLE communication. Each device is identified using its Bluetooth MAC address. Then, with the use of a ledger created through blockchain technology, other nodes that have been initialized are paired via Bluetooth. For this implementation, the blockchain will be used as a ledger that holds all the Bluetooth MAC addresses of the nodes that later becomes the gateway. The gateway will be composed of a chain of these nodes that communicate with each other.

Our experimental setup is shown in Fig. 1. Each Raspberry Pi runs on Raspbian and contains a script programmed in Python 3.6. A Bluetooth library called PyBluez was used to set up communication between the nodes. Each script is set up by using a programmed node class that is initialized at its execution. The functions used by the node class to interact with the different classes included in the setup are shown in Listing 1.

```
class Node:
    def __init__(self):
        self.ledger
        self.cipher
        self.key
    def get_local_bdaddr(self):
        ... return bluetooth_address
    def get_ledger(self):
        ... return hashed_ledger
    def verify_ledger(self, hashed_ledger):
        ... return hashed_ledger == hashed_base_ledger
    def get_consensus_packet(self):
        ... return packet
    def read_packet(self, packet):
        ... return decrypted_packet
```

Listing 1: Node class code abstraction.

Upon initialization, the node class defines the Bluetooth MAC address of the device. It then loads in the ledger class that will be used by the gateway. This ledger class contains functions that manage the blockchain within the code. The ledger class is built using two classes; a block class, and a chain class. The block class contains the functions and variables necessary for each block within the blockchain, while the chain class contains the outside functions and variables that manage the blockchain as a whole. This completes the ledger and an abstraction of the block class and the chain class. Listings 2 and 3 show an abstraction of the functions used by the block class and chain class respectively to build the blockchain module and interact with the rest of the node.

```
class Block:
   def __init__(self, index, timestamp,
       transactions, previous_hash, public_key):
       self.public_key
       self.index
       self.timestamp
       self.transactions
       self.previous_hash
       self.hash
   def gen_hashed_block(self):
       ... return hashed_block
   def validate_private_key(self,private_key):
       ... return hashed_key == self.public_key
   def display_block_info(self):
```

Listing 2: Blockchain block class code abstraction.

```
class Chain
   def __init__(self)
       self.chain
   def gen_next_block(self, private_key,
       transactions):
       ... self.chain.append(Block(...))
   def display_contents(self):
   def gen_genesis_block():
```

Listing 3: Blockchain class code abstraction.

Once the ledger has been loaded, the class initializes the cipher class. The cipher class was programmed to use a python library called PyCrypto. This cipher specifically uses AES (Advanced Encryption Standard) encryption and decryption functions from the library. Listing 4 presents an abstraction of the cipher class and that function that it uses to interact with the blockchain module.

```
class AESCipher
   def __init__(self):
       self.base
   def encrypt(self, secret, key):
       ... return encrypted_message
   def decrypt(self,encrypted, key):
       ... return decrypted_message
```

Listing 4: AES cipher class code abstraction.

After all of the necessary classes have been initialized, a Bluetooth socket is opened using the device's MAC address provided by the node class.

2) Blockchain setup: To keep the collection of nodes that compose the gateway exclusive, the addresses must be stored in an immutable structure to prevent any real-time tampering. By using blockchain technology as the main storage, the addresses are protected from being easily overwritten giving it a certain level of security. This results in a read-only ledger that holds all of the addresses that identify the member of the gateway. After setting up the blockchain, identical copies of it are given to each node. This sets up the decentralized system by having each node equal privileges and capabilities in the network. By taking advantage of this decentralized system, the nodes are able to apply common functions from other blockchain implementations via automated protocols.

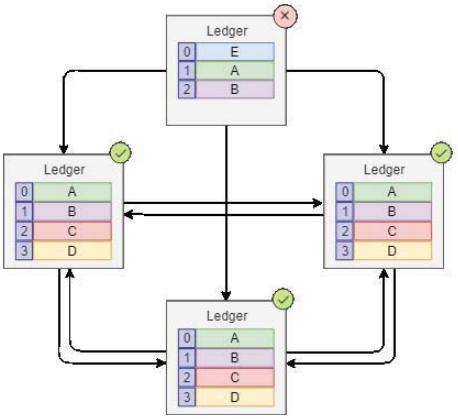

Fig. 2: Consensus system detecting unauthorized node and denying communication.

The main protocol integrated into this design is the consensus protocol. This protocol is implemented by programming the nodes to communicate with each other through the consensus system that holds each node accountable for keeping an up to date copy of the blockchain. This is done by sending a hashed copy of the blockchain placed inside a packet to all the other nodes in the gateway. Each node then verifies if the received blockchain hash matches with theirs. If the verification fails, then the source is reported and denied future communications. A diagram that shows a situation where the consensus system detecting the unauthorized node and denying its communication to the gateway is shown in Fig. 2.

3) Node communication setup: As mentioned, BLE is used to connect the devices. When sending a packet, the node is programmed to pair with the desired device through its Bluetooth MAC address. To keep the connection safe and secure, the created connection socket is closed after successfully sending the packet. This protects the device from any device that attempts to eavesdrop from the connection. In addition, this creates a socket connection protocol that allows the usage of Bluetooth connectivity within the device to be controlled.

The created connection protocol also has a function that drops any packet in case the device fails to establish a connection with the destination. No matter if the packet is sent or not, the device then goes into an idle state listening for an incoming packet from any of the other nodes that is part of the gateway. If a packet is received, the node then authenticates and decrypts the packet through a combination of an AES cipher and a hashing library. Once the packet has been successfully authenticated and decrypted, the contents of the packet are read and processed based on its type. To minimize any collisions and stalling in packet sharing, aside from actively dropping failed packets, the script uses round robin scheduling to send the packets. Since every device has an exact copy of the blockchain, each device has the same sequence of addresses built in the chain. This means that by taking the next address in the sequence, the node will keep trying to send to the next device down the list. With each

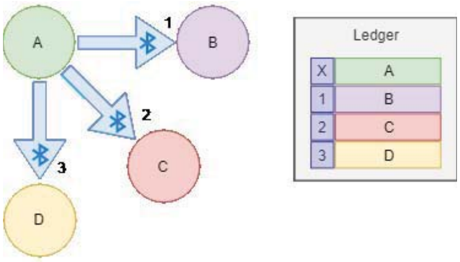

Fig. 3: Round robin scheduling example.

device starting from the next address after its own, it works its way around the sequence. This reduces the number of sources of collisions.

In addition, this scheduling method provides a simple scheme that prevents the device from getting stuck trying to connect to a node that keeps failing. This is helpful for any nodes that go offline for some reason. The scheduling still goes through each address but the active packet dropping will allow the node to bypass any offline nodes and at the same time accommodate any incoming nodes come back online. This creates a dynamic protocol of internal gateway communication where nodes can be actively swapped in and out of the gateway without disturbing its overall functionality as long as the node is registered within its ledger. A diagram of a round robin scheduling example of a four node setup where node A is sending packets to nodes B, C, and D in an order dictated by the blockchain ledger, is shown in Fig. 3.

IV. Testing and Evaluation

The feasibility of the proposed design was examined through testing. By obtaining the number successfully sent and verified packets within the gateway chain, we get an idea of the throughput of the proposed design. A better throughput can result in better communication between gateways. The test was set up by having 4 gateway nodes attempt to communicate with each other by sending verification packets. These packets contain the identification of the sender that is verified by the recipient via its blockchain. The script was modified to record all attempts and specify which packets failed and succeeded. In addition, each recorded entry is timestamped. The recorded data were grouped in time intervals of 1 minute. After grouping the data accordingly, the information was plotted as the relationship between the time intervals and the number of successful packets sent. The test was carried out for 1 hour.

In terms of the communication medium between gateway nodes, we have two configurations. The first configuration uses BLE, while the second uses WiFi via server-client sockets. The WiFi configuration will function as the control for this experiment. Also, it will be using the same structure as the BLE method in verifying messages. However, instead of using the Bluetooth MAC addresses to identify each node, it will be using the Internet protocol (IP) address of each device.

The first test will gauge how well can the configuration transmit packets within the network. The number of packets that were successfully transmitted and verified within the network is shown in Fig. 4. This data is aggregated within minute intervals. As it can be seen, the BLE configuration is better on systematically transmitting data within the chain of devices than its WiFi counterpart. The BLE test yielded a verification packet rate of 14 per minute, while the WiFi test was 4 per minute. These results show that in a decentralized setup, BLE is more advantageous than WiFi. This observation can be attributed to the pairing capabilities of BLE technology. BLE can pair with multiple devices and provide service in parallel which results in a better exchange of data [15]. On this other hand, WiFi focuses on a single data stream which needs to be regulated in order to manage a more complex network. Also, the pairing between two devices ensures a higher success rate in transmitting data [10].

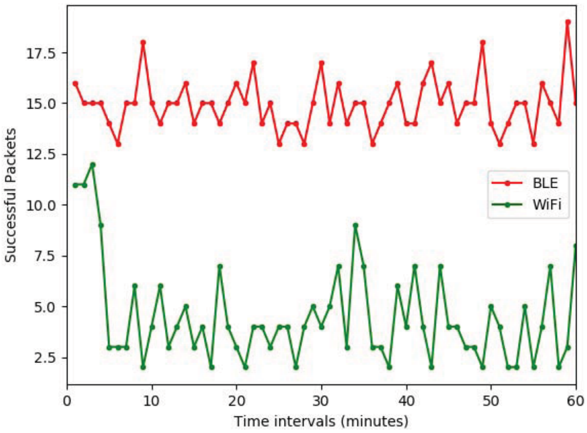

Fig. 4: Packet throughput test results.

The next test takes the two configurations and measures their average power consumption. We arbitrarily select 1 of the 4 nodes in the setup and use it as the device that will be measured. To obtain the energy consumed by each configuration, we used a power monitor. The power monitor provides power to the selected device and at the same time measures the amount of power it consumes while running our network setup. A comparison between the two results in intervals of seconds is shown in Fig. 5. Our proposed configuration does a better job at conserving energy than the control. We calculated the averages of each iteration. The BLE test yielded 1095.40 mW while the WiFi test was 1191.83 mW. The two tests resulted in a difference of 96.43 mW in favour of the BLE configuration.

According to the two experiments, BLE is a better fit in terms of decentralized networks. In a topology that uses a mesh structure, BLE technology can give better results in terms of power consumption. In addition, gateway nodes have a better means of communicating due to the paired structure of BLE. As a result, the data exchange between gateways will have a better flow.

To further investigate the feasibility of BLE as our medium of data transmission, we tested to see if varying the number of nodes would drastically change its performance. Hence, we set up experiments using 2, 3, 4, 5, and 6 gateway nodes. The

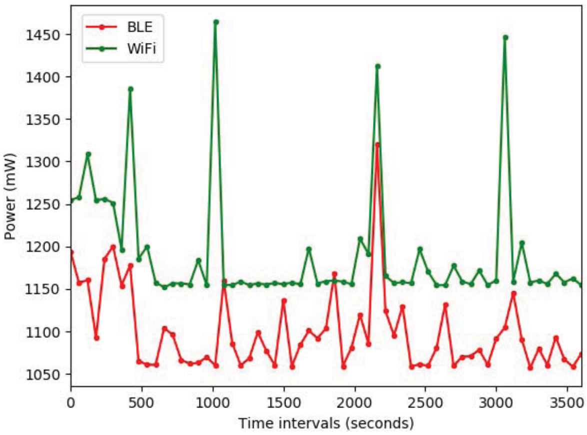

Fig. 5: Power consumption test results.

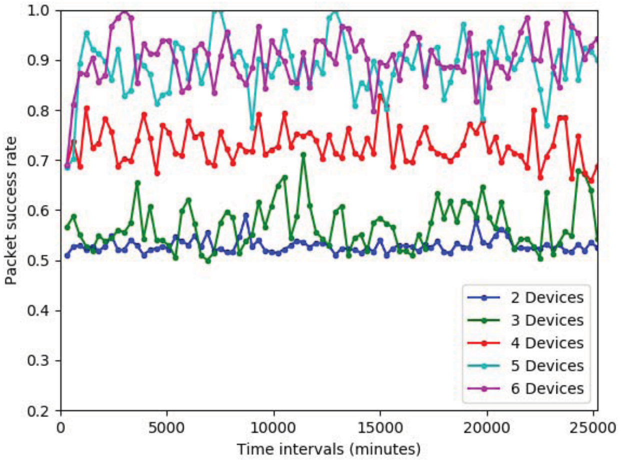

Fig. 6: Packet success rate vs. Run time (min) over variable gateway node counts.

success rate of the packets as the number of nodes increased in the gateway chain is shown in Fig. 6. According to the results, 2 and 3 nodes yielded the worst success rate, while 5 and 6 nodes had the best. We took these results as an indicator of potential scalability capabilities for the design. For future iterations of this design, we aim to investigate further the ability of the design to be scaled up and if there is a ceiling to it. Also, we aim to investigate more on the concept of decentralization and how to maximize its features to create a better network setup.

V. Conclusion

In this paper, we created a network setup for an IoT gateway. The proposed design is composed of decentralized nodes that are connected through BLE technology and synchronized using blockchains. According to experimental results, the data exchange between the gateway nodes were better for the BLE test than the WiFi test. The metric used was the number of successfully transmitted packets within the network. The BLE test resulted in an average of 14 verification packets per minute while the WiFi test was 4. Also, the BLE configuration yielded a better average power consumption of 1095.40 mW, while the WiFi alternative was measured to be 1191.83 mW.

A potential constraint to the design is in the testbed. The number of devices that can be added to the experiment is limited. For future iterations, we aim to create a more adaptive testbed that can properly model a growing network that can accommodate a larger number of devices.

Overall, blockchain technology gives the network the ability to become decentralized while remaining secure and synchronized. Paired with BLE, the gateway nodes that compose the network have a more secure, faster, and energy efficient means of communicating. The result is a design that could potentially be more effective in maintaining an IoT network in terms of capacity, power, and security.

References

[1] K. Patel and S. Patel, "Internet of things-iot: Definition, characteristics, architecture, enabling technologies, application & future challenges," *International Journal of Engineering Science and Computing (IJESC)*, vol. 6, no. 5, pp. 5:6122–5:6131, May 2016.

[2] W. Lee and A. Sharma, "Smart sensing for iot applications," in *2016 13th IEEE International Conference on Solid-State and Integrated Circuit Technology (ICSICT)*, Oct 2016, pp. 362–364.

[3] T. Aste, P. Tasca, and T. Di Matteo, "Blockchain technologies: The foreseeable impact on society and industry," *Computer*, vol. 50, no. 9, pp. 18–28, 2017.

[4] O. Novo, "Blockchain meets iot: An architecture for scalable access management in iot," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 1184–1195, April 2018.

[5] M. Singh, A. Singh, and S. Kim, "Blockchain: A game changer for securing iot data," in *2018 IEEE 4th World Forum on Internet of Things (WF-IoT)*, Feb 2018, pp. 51–55.

[6] Y. Yu, Y. Li, J. Tian, and J. Liu, "Blockchain-based solutions to security and privacy issues in the internet of things," *IEEE Wireless Communications*, vol. 25, no. 6, pp. 12–18, December 2018.

[7] M. E. Peck, "Blockchain world - do you need a blockchain? this chart will tell you if the technology can solve your problem," *IEEE Spectrum*, vol. 54, no. 10, pp. 38–60, October 2017.

[8] W. Ting, H. Jun-da, G. Chenjie, Z. Jia, and Y. Wei, "Wireless monitoring system based on bluetooth smart phones," in *2010 International Conference on Networking and Digital Society*, May 2010, vol. 2, pp. 648–651.

[9] J. DeCuir, "Introducing bluetooth smart: Part 1: A look at both classic and new technologies.," *IEEE Consumer Electronics Magazine*, vol. 3, no. 1, pp. 12–18, Jan 2014.

[10] K. Chang, "Bluetooth: a viable solution for iot? [industry perspectives]," *IEEE Wireless Communications*, vol. 21, no. 6, pp. 6–7, December 2014.

[11] S. Cha, M. Chuang, K. Yeh, Z. Huang, and C. Su, "A user-friendly privacy framework for users to achieve consents with nearby ble devices," *IEEE Access*, vol. 6, pp. 20779–20787, 2018.

[12] K. E. Jeon, J. She, P. Soonsawad, and P. C. Ng, "Ble beacons for internet of things applications: Survey, challenges, and opportunities," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 811–828, April 2018.

[13] W. Ting, H. Jun-da, G. Chenjie, Z. Jia, and Y. Wei, "Wireless monitoring system based on bluetooth smart phones," in *2010 International Conference on Networking and Digital Society*, May 2010, vol. 2, pp. 648–651.

[14] S. Alletto, R. Cucchiara, G. Del Fiore, L. Mainetti, V. Mighali, L. Patrono, and G. Serra, "An indoor location-aware system for an iot-based smart museum," *IEEE Internet of Things Journal*, vol. 3, no. 2, pp. 244–253, April 2016.

[15] S. M. Darroudi and C. Gomez, "Modeling the connectivity of data-channel-based bluetooth low energy mesh networks," *IEEE Communications Letters*, vol. 22, no. 10, pp. 2124–2127, Oct 2018.

[16] C. Garrido-Hidalgo, D. Hortelano, L. Roda-Sanchez, T. Olivares, M. C. Ruiz, and V. Lopez, "Iot heterogeneous mesh network deployment for human-in-the-loop challenges towards a social and sustainable industry 4.0," *IEEE Access*, vol. 6, pp. 28417–28437, 2018.

[17] S. Cha, J. Chen, C. Su, and K. Yeh, "A blockchain connected gateway for ble-based devices in the internet of things," *IEEE Access*, vol. 6, pp. 24639–24649, 2018.