IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 7 November 2022, accepted 6 December 2022, date of publication 21 December 2022, date of current version 5 January 2023.

Digital Object Identifier 10.1109/ACCESS.2022.3231491

RESEARCH ARTICLE

Device Agent Assisted Blockchain Leveraged Framework for Internet of Things

TARIQUE MOHAMMED NASRULLAH[1], MD. MANOWARUL ISLAM[1], MD. ASHRAF UDDIN[1], MD. ANISUZZAMAN KHAN[1], MD. ABU LAYEK[1,2], (Senior Member, IEEE), ANDREW STRANIERI[3], (Member, IEEE), AND EUI-NAM HUH[2], (Senior Member, IEEE)

[1]Department of Computer Science and Engineering, Jagannath University, Dhaka 1100, Bangladesh
[2]Department of Computer Science and Engineering, Kyung Hee University, Global Campus, Yongin 17104, South Korea
[3]Internet Commerce Security Laboratory, Centre for Informatics and Applied Optimisation, Federation University, Ballarat, VIC 3350, Australia

Corresponding author: Eui-Nam Huh (johnhuh@khu.ac.kr)

This work was supported by the Ministry of Science and ICT (MSIT), South Korea, under the Grand Information Technology Research Center Support Program Supervised by the Institute for Information & Communications Technology Planning & Evaluation (IITP) under Grant IITP-2022-2015-0-00742.

ABSTRACT Blockchain (BC) is a burgeoning technology that has emerged as a promising solution to peer-to-peer communication security and privacy challenges. As a revolutionary technology, blockchain has drawn the attention of academics and researchers. Cryptocurrencies have already effectively utilized BC technology. Many researchers have sought to implement this technique in different sectors, including the Internet of Things. To store and manage IoT data, we present in this paper a lightweight BC-based architecture with a modified raft algorithm-based consensus protocol. We designed a Device Agent that executes a novel registration procedure to connect IoT devices to the blockchain. We implemented the framework on Docker using the Go programming language. We have simulated the framework on a Linux environment hosted in the cloud. We have conducted a detailed performance analysis using a variety of measures. The results demonstrate that our suggested solution is suitable for facilitating the management of IoT data with increased security and privacy. In terms of throughput and block generation time, the results indicate that our solution might be 40% to 45% faster than the existing blockchain.

INDEX TERMS Blockchain, cryptocurrency, distributed, IoT, ledger, device agent, device registration, miners, docker.

I. INTRODUCTION

The Internet of Things (IoT) is a network of interrelated computing devices having the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction [1]. IoT connects the physical environment to a cyber-physical system (CPS) to support a wide range of infrastructural, industrial, military, and organizational applications in the fields of medical healthcare, manufacturing, agriculture, energy management, and battlefield [2]. In addition, the IoT aims to minimize the downtime of systems and facilitate continuous monitoring of physical data.

However, the increasing usage of IoT devices brings some additional challenges that are associated with privacy, immutability, reliability, and performance [3]. Traditional IoT service architectures involve the transmission of data captured by IoT devices to cloud entities that third parties manage. Third-party management makes maintaining data privacy and data security difficult [3].

The concern of data transmission and the necessity of secure storage and data immutability increase with the rapid growth of IoT devices. These devices generate a massive volume of sensitive data [4]. Many private organizations aim to automate and monitor their production and supply chains with IoT technology. However, this sensitive data can be accessed by unauthorized parties. Therefore, researchers have been motivated to secure private data by maintaining distributed, immutable logs of transactions and operations.

The associate editor coordinating the review of this manuscript and approving it for publication was Rahim Rahmani.

To ensure the privacy, immutability, and reliability of generated transactions by IoT devices, many researchers [5], [6], [7] have suggested blockchain technology for managing IoT applications. IoT data management on a decentralized platform with higher security and privacy can be a promising solution.

Blockchain technology is built upon a decentralized network that maintains a database of verified and validated transactions by multiple participants, also called nodes. The records on the BC are stored by a relatively large community where no single entity has control over them [8]. One of the most recognized and successful uses of BC is Bitcoin, which is a decentralized digital ledger across a peer-to-peer network and provides a mechanism to store and trade digital currencies called bitcoins [8].

A. CURRENT BLOCKCHAIN's PROBLEM

In BC technology, decentralization refers to transferring control and decision-making from a centralized entity to a distributed controller. BC decentralized networks reduce or eliminate the level of trust that participants must place in one another in the conventional system. Most BC uses asymmetric cryptography to secure data transportation across multiple storage mediums of a large network. Asymmetric cryptography uses public and private key pairs where it is mathematically impossible to guess a user's private key from their public key. This ensures the security and confidentiality of a user's data. This BC structure can ensure higher security and privacy for IoT applications. However, the integration of BC with IoT devices poses a couple of challenges. Several BC challenges while adopting it in the IoT are described below.

1) **High computational cost:** The current blockchain technology used in bitcoin or Ethereum consumes high power for processing transactions and blocks. BC participants are required to compete to solve a mathematical puzzle called "Proof of Work" to mine a block that needs a significant amount of power. Zhou [9] reported that the energy consumed for Bitcoin is more than the entire energy required for many countries. In IoT applications, transactions are produced more frequently. As a result, BC for IoT will require more energy than Bitcoin BC. Since IoT devices are restricted to low computational processing power and memory, they cannot directly accommodate BC technology [10]. Further, according to Sukhwani et al., utilizing the PBFT consensus process on the permissioned blockchain network such as Hyperledger Febric, the time to execute block is unexpectedly long-by an order of magnitude-compared to the time to consensus for blocks with an average size of three. This results in a system performance bottleneck when a high TPS (transaction throughput per second) is needed.
2) **Low throughput:** Bitcoin BC can confirm a block within around 10 minutes, and Ethereum BC can process around 20 transactions per minute, whereas the current VISA or MasterCard can process more than 2000 transactions per second [5]. This low throughput of the BC hinders its adoption in IoT applications. Therefore, there is a need to design a customized blockchain for IoT.
3) **Huge volume of memory and scalability issue:** Depending on the design strategy used in BC, a significant number of participants might download the entire ledger in order to attend the mining process. This can overwhelm the participant's storage. IoT devices generate a massive amount of data. Cisco experts predicted that the amount of IoT data was already 500 zettabytes by 2019 [11]. The global data center's IP traffic would reach around 10.4 zettabytes [12]. Although cloud providers can support on-demand processing power and storage for massive data, there is a trade-off between storage and latency. Thus, storing this data in a distributed system is a challenging task. Since IoT devices are limited to a low capacity of memory, they cannot participate in mining and validating blocks in the BC. Therefore, there need to be modifications to the BC so that low-profile devices can take part in the different BC operations.

To address the above-discussed challenges, we have proposed a lightweight BC-based IoT framework, and the analysis of the performances proves that our architecture can support diverse IoT applications with respect to throughput and power consumption. Furthermore, our contribution minimizes the computational cost and energy and increases throughput by reducing the time required for validating and verifying data blocks.

With the advancement in embedded processors, actuators, sensors, and communication systems, IoT devices are equipped to communicate, compute, and complete automated tasks. Many daily life appliances have been able to connect to the Internet. IoT devices include smart pacemakers, heart rate monitors, smart refrigerators, smart coffee makers, smart television, smart home assistants, and smart door locks. These devices collect and transmit a large amount of privacy-related data to a centralized server, often a cloud server. For example, IoT devices such as smart cameras, smart health monitoring devices such as heart rate monitors, and glucose level monitors can reveal private information about users. Due to the limited processing capabilities of IoT devices, they usually leverage externally controlled third-party service providers to perform additional data processing [13]. By transmitting sensitive user data to third-party service providers, users are forced to trust service providers to enforce data protection and ensure data privacy. However, service providers often violate data privacy policies by using data collected from users for unauthorized purposes [10].

Blockchain systems are designed to ensure the privacy and immutability of data. In the blockchain architecture, data is stored as transactions, which are small chunks of data. Validating and verifying this transaction requires relatively high computational power and time. Frequent transaction generation from IoT systems demands minimizing validation time to ensure service quality. Therefore, a lightweight blockchain architecture needs to be designed for IoT data management.

B. BLOCKCHIAN's ROLE IN IoT

The goal of adopting BC technology for IoT applications is to facilitate decentralized storage, higher privacy, and security-enhanced performance in the quality of service and reshape the production of the industrial supply chain by securing sensitive data. The following properties of BC have motivated researchers to explore BC in the IoT:

1) **Decentralized storage for IoT:** In a conventional system, a centralized database stores IoT data. As a result, the person who maintains the database server can alter data without the knowledge or permission of the data owner. Besides, a single database can cause bottleneck problems while responding to many requests. Therefore, to avoid a single point of failure and ensure tamper-proofing of the IoT data, the ideal solution is to store IoT data in the ledger of BC.
2) **Higher security and privacy:** BC uses a public and private key for making digital signatures and their addresses. Consequently, BC can provide the participants with anonymity, ensuring the users' privacy. Further, BC never suffers from a single point of failure and can withstand significant cyberattacks, including ransomware and denial of service.
3) **Enhanced performances:** Users can access data from multiple points, ensuring quick responses to users' queries. Moreover, BC can perform processing without the requirement of third parties. As a result, transactions are committed within a short period, whereas in the conventional banking system, users need to wait for several days to have their transactions committed when they send currency to a user having an account in a foreign bank.

C. CONTRIBUTION

In this research, we have proposed and implemented a fast, lightweight, and secure BC model to make IoT data secure for private usage. Blockchain provides a storage system that makes data immutable, authentic, and auditable. Consequently, IoT data can be shared without the trust of third parties. Furthermore, the proposed system can eliminate the time needed for a private authority to deploy a distributed system across the globe and scale up immediately. Our significant contributions include the following.

1) First, we have proposed a lightweight blockchain-based decentralized IoT architecture. We have customized our blockchain to overcome the downsides of the existing blockchain. We have selected miners based on the Raft algorithm to minimize energy consumption. Unlike other BC networks, only a few nodes in the network are responsible for validating a block instead of competing. This reduces the overall energy consumption and transaction throughput of the network.
2) Secondly, we have introduced a device agent whose functionalities differ from those of the Patient-Centric Agent in [5], [13], and [14] with respect to functionalities. The device agent connects IoT devices with our designed and customized BC through a secure proposed registration process. In addition, the device agents are responsible for ensuring security and privacy for IoT devices using asymmetric cryptography.
3) Thirdly, we have proposed using multiple chains of ledgers where the device agent manages a separate ledger for every IoT device associated with the device agent. This modification ensures that low-profile devices can participate in the mining process. Further, maintaining an individual ledger for IoT devices rather than using a single ledger for all IoT devices provides better privacy.
4) Our fourth contribution is to implement the Device Agent and blockchain network using the Go language on Docker and Google Cloud to analyze extensive performances in terms of transaction throughput, latency, propagation delay, and consensus time.

The rest of the article is organized as follows. In Section II, Related Works provides an overview of some existing works related to the article. Section III represents the proposed model of the BC-based framework for IoT systems. In section IV, we have discussed the design tools and implementation details, system configuration, and matrices, testing environment setup, and performance analysis. Finally, Section V, Conclusions, provides an overall discussion of the work, concludes the paper, and outlines the future work. Future work suggests the enhancement of this work.

II. RELATED WORKS

This work aims to alleviate the current weakness of a BC, which is a critical component for the design and development of a secure IoT system. The adoption of BC in the IoT poses several challenges, including the lack of IoT-centric transaction validation rules, the absence of an IoT-oriented consensus protocol, and the secure integration of IoT devices with the BC [15]. This section presents an overview of some BC-based IoT data management frameworks before presenting our proposed methodology.

A. BLOCKCHAIN IN E-HEALTH

Blockchain is a decentralized technology that has gained popularity among academicians and researchers since the Bitcoin white paper was published in 2008 [16]. Blockchain technology provides many benefits, such as decentralization, anonymity, and audibility. There is a wide range of BC applications, including cryptocurrency, financial services, property records, healthcare, and IoT for implementing public and social services [17]. Blockchain can be utilized in the healthcare industry for data management. At present, one of the main problems in healthcare is that organizations hold multiple and fragmented health records of patients [18]. Healthcare organizations and healthcare personnel are using different approaches and tools to exchange the patient's health information [19]. For various reasons, patients need to move from one place to another. Therefore, patients usually seek health care services from different providers in different regions. As a result, health-related information can be fragmented and outdated, leading to a poor connection between the provider and the patient's medical information for data exchange. Further, storing sensitive healthcare data in a centralized cloud server is vulnerable to cyberattacks. However, by leveraging the distributed property, blockchain technology can ensure the accountability and integrity of health data [20]. In the electronic sense, blockchain can promote healthcare transactions (i.e., knowledge sharing) through a distributed ledger spread in different locations rather than through a central authority. It may also provide the patient with data control, where they determine which information is to be exchanged with which organization.

Uddin et al. [5], [6] first proposed a personalized patient-centric agent to manage health data on the blockchain. In [5], they proposed an end-to-end PCA-controlled eHealth architecture where the PCA manages storage on the BC, determines the security and privacy level for a patient, and selects a single miner to generate PoW using a secretary optimization algorithm. Uddin et al. [3], [10] suggested a decentralized PCA to replicate it on the three levels, including the Smartphone, Fog, and Cloud level, to coordinate a portion of blockchain on the Fog and other portions of the BC on the Cloud network. The decentralized PCA was employed to withstand major cyber attacks. In [21], the PCA on the three levels assisted in migrating patients' tasks to foreign agents if they were overloaded with various kinds of patient medical data analytics-related tasks. Uddin et al. [14] recommended a PCA-assisted machine learning-based storage recommendation system to address the big data issue of eHealth. However, our Device Agent differs from the PCA in multiple ways. The Device Agent maintains an IoT device-wise ledger and performs a secure registration for IoT devices on the BC network.

Dwivedi et al. [22] discussed the challenges and problems to incorporate blockchain technology in AI-enabled 5G networks. They proposed a customized blockchain to address the current challenges. The authors suggested using a raft algorithm-based consensus algorithm and ZKP (Zero-knowledge) for maintaining privacy. However, they did not develop a model for analyzing performance.

Most authentication methods do not provide legitimate users with privacy in IoT. To address this problem, Dwivedi et al. [23] proposed a Zero Knowledge-based authentication technique. This authenticates network devices without revealing user identification or other data. This privacy-preserving technique can be applied to IoT-based healthcare applications. In addition, This approach can be applied to any generic use cases where privacy-preserving authentication is required. They also designed a ZKNimble-based data encryption technique that can be used for encryption and decryption by legitimate users after Zero Knowledge Proof authentication. However, the performance analysis has not been done.

Jiang et al. [24] examined the application of blockchain in the Internet of Value (IoV). The authors showed the relationship between IoV and blockchain technology and the technical obstacles of implementing blockchain in IoV applications. The authors classified IoV data blocks and set up multiple networks to incorporate IoV nodes with blockchain. The architecture was simulated and assessed for automobile networking systems. Mbraek et al. [26] suggested an IoT platform in which a mobile agent is used to conduct transactions with a hierarchical blockchain system (MBS). The proposal made use of enterprise blockchain, but did not explain how the mobile agent was implemented. There was no security enforcement at the user level. Fortino et al. [27] developed an algorithm that can group agents based on their reputation capital in IoT contexts and established a reputation model that is centered on the capitalization of an agent's reputation. The currently available blockchain technology was employed. The consensus procedure was not worked on.

Sabir et al. [28] covered how blockchain technology and IoT smart contracts can defend mobile agents against harmful attacks. They suggested using transaction behavior to spot nefarious actors. The suggested system has not been put into use. The suggested design aims to make it possible for mobile agents to migrate securely in order to safeguard IoT applications from bad users and preserve security.

B. BLOCKCHAIN IN IoT DATA

Nowadays, several types of embedded systems are being used to ensure the safety and security of IoT data [29]. A large number of IoT devices in a system are interconnected and connected to the internet to form an Internet of Things (IoT). IoT generates a massive amount of data that keeps increasing day by day. Blockchain technology [30] can be applied for managing IoT data to provide reliability, integrity, and availability. Reyna et al. [31] discussed IoT and Blockchain integration and its objectives. The study summarizes and discusses how IoT devices can be used to accommodate blockchain. Most IoT transactions are maintained by Ethereum-based Blockchain [32]. Using Ethereum and smart contracts, Huh et al. [33] proposed a blockchain PKI management system. This solution uses RSA to generate a public key and store that public key in Ethereum, and private keys are saved on individual devices. But the proposed solution has two problems. The first one is the long duration time of one transaction (approximately 12 seconds) and the second obstacle is a large storage requirement for the constrained hardware of IoT clients. Uddin et al. [13] has proposed a lightweight BC leveraged framework for monitoring IoT devices underwater. They optimized the BC using a lightweight Proof of Stake protocol, and a smart Agent coordinates the routing protocol running on the Internet of Under Things. Two blockchains were used in the proposal. The IoUT devices contain a lightweight BC to save their security key, and the cloud holds a BC to store IoUT data. Uddin et al. [1] designed

TABLE 1. The summary of the existing literature.

Authors	Tools	Main Contributions	Limitations/Remarks
Uddin et al. [3]	Java programming was used to implement blockchain and Scyther security tool was used to verify and validate security protocol	The authors proposed a decentralized Patient Centric Agent that bridges the medical sensors and blockchain. Further, they proposed a modified Proof of Stake protocol	Extensive performance analysis has not been done for the security protocol.
Dwivedi et al. [22]	Tools are not described	The authors suggested using a consensus process based on the raft algorithm and Zero-Knowledge Proofs (ZKP) to maintain privacy.	However, they did not develop a model for analyzing performance.
Dwivedi et al. [23]	Mathematical analysis has been shown	They presented a method of authentication based on Zero Knowledge. This authenticates network devices without revealing user identity or other sensitive information. This technique for protecting privacy is applicable to IoT-based healthcare applications.	However, the performance analysis has not been done.
Jiang et al. [24]	MATLAB was used to simulate the vehicular network and theoretical analysis has been performed for measuring the performance of the blockchain	They explored blockchain's application in the Internet of Value (IoV). The authors illustrated the relationship between the Internet of Vehicles and blockchain technology, as well as the technical challenges of integrating blockchain in IoV applications.	The architecture was simulated and assessed for automobile networking systems.
Liang et al. [25]	NSL-KDD dataset was used. Python Keras TensorFlow used to develop the model.	The authors proposed a multi-agent-based intrusion detection system. The multi-agent assisted in collecting data and deep learning was used to detect intrusion.	Although multi-agent concepts of gathering information were mentioned in the framework, however, no test bed was developed to show how multi-agent works in the blockchain-based environment. No performances related to blockchain were analyzed.
Mbraek et al. [26]	Hyperledger Fabric	The authors recommended IoT platform where the mobile agent was utilized to perform transactions with a multilevel blockchain system (MBS).	The proposal used enterprise blockchain, however, did not describe how the mobile agent was implemented. No security was enforced at the user level.
Fortino et al. [27]	The experiment has been conducted on the Ethereum blockchain	The paper introduced a reputation model centered on the capitalization of an agent's reputation and an algorithm that is capable of grouping agents based on their reputation capital in IoT contexts was developed.	The existing blockchain technology has been used. They did not work on the consensus protocol.
Sabir et al. [28]	The proposed system has not been implemented.	The author discussed how mobile agents can be protected from malicious attacks using blockchain and smart contracts in IoT. They proposed to use the behavior of transactions to identify malicious agents.	The proposed system has not been implemented. The suggested architecture intends to enable a secure migration of mobile agents to protect IoT applications from malicious agents and maintain security.

a secure framework by using blockchain for monitoring the smart home. They have used a certificateless sign encryption to ensure privacy for IoT devices. The network manager in the architecture provides the IoT devices with pseudonymous identifiers to communicate with BC.

Blockchain technology is a distributed consensus scheme that allows data to be securely stored and verified without the need for any centralized authority. It is based on a peer-to-peer network, where each node is equal to all others. It increases the overall security and resiliency of the system [34]. In the education sector, blockchain can solve several issues that are linked with securing data in the case of ubiquitous learning environments [35]. Nethaji et al. developed TrueRec based on the Ethereum blockchain system that stores professional and academic credentials [36]. The effective application of blockchain can be in governance for automation, transparency, and audibility. The World Citizen project [37] to identify citizens all over the world is an example of a decentralized passport service. Along with the abovementioned sectors, blockchain can be applied in the energy sector [38], and supply chain management [39] to secure transactions.

Electronic voting has been established to ensure high privacy and verification of ballots in recent times. In particular, cryptography-based blockchain technology is highly open and transparent for individual transactions. Anyone can use the blockchain to access transaction content. However, blockchain decentralization and faster e-voting computation properties may influence the voting mechanism and may not reproduce the actual voting process.

Electronic voting (e-voting), which uses an electronic device to help cast and count votes, has been an important research subject in cryptography for the past few decades. [40], [41]. One of the most challenging problems in the modern world is ensuring the security of information. Secure e-voting can be implemented using multi-party computation (MPC) technology [42]. Liu [43] described the design of an e-voting system based on blockchain technology and a blind signature encryption mechanism. The paper's main contribution is that they added blockchain technology and a blind signature encryption mechanism without relaying trusted third parties.

BC can be a permissionless or permissioned blockchain depending on accessibility. In a permissioned blockchain, a selective group of trusted authorities determines who gets what access to the blockchain and how much. This BC paradigm restricts who can use the BC and how, making it easier to monitor the ledger and pinpoint the culprit behind any unauthorized modifications. This BC validates digital certificates to either add or remove users. The benefits of the blockchain's security and privacy are vivid. Most of the world's largest corporations have started using this approach, and it's gaining in popularity every year. Most applications of permissioned Blockchain will be in business contexts where the security of sensitive information is paramount, such as in supply chain management, contract formation, and payment verification. In contrast, permissionless blockchains are the opposite of Permissioned blockchains. In the permissionless model, often known as a public blockchain, there are no participation limitations and no administrator controls participation. Anyone can validate the facts and participate in the consensus. There are no administrators who allow people to participate or grant them the power and authority to make the modifications. It is a decentralized blockchain platform between unidentified parties. In this paper, we aim to develop a permissioned consortium blockchain.

III. THE PROPOSED BC LEVERAGED IoT ARCHITECTURE

The proposed Blockchain-based IoT architecture consists of three different types of networks: the IoT network, Device Agent network, and blockchain network, as illustrated in Figure 1. Particular numbers of IoT devices are associated with a Device Agent, which is placed in the local network. The IoT devices are connected to the global blockchain network via a Device Agent. IoT devices send their data to their affiliated Device Agent. A Device Agent (DA) is a dedicated node between the IoT device and the blockchain network. DA maintains a portion of the blockchain which was generated by the connected IoT device. DA does not actively participate in the network. Instead, it is capable of discovering the blockchain network. The Device Agent randomly selects BC miners to include the IoT data in the BC ledger.

In the BC network, miner nodes form a peer-to-peer network to support network protocols, including routing. All miners validate and propagate blocks and discover and maintain connections to peers. Miners also maintain a complete and up-to-date copy of the entire ledger. Figure 1 depicts the overall architecture of our proposed system. We discuss every part of the proposed system below.

A. IoT NETWORK

This network comprises various IoT devices, including smartwatches, temperature sensors, ECG sensors, cameras, and smartphones. These devices communicate with the device agent using the ZigBee or Bluetooth protocol. Specific numbers of IoT devices can be affiliated with a Device Agent as they cannot directly communicate with blockchain networks due to their limited processing and network resources.

B. DEVICE AGENT(DA) NETWORK

The second most important component of this IoT architecture is the Device Agent. IoT devices are usually restricted in their processing and memory capacities. So it is inconvenient for a device to communicate with the blockchain network directly. A Device Agent is a dedicated hardware-running software agent. The Device Agent acts as an intermediary between IoT devices and the blockchain network. An IoT device communicates with the DA using its own protocol. DA collects IoT data and sends it to the blockchain network.

To uniquely identify the IoT device on the blockchain network, DA first registers the device on the blockchain through **Device Registration Service(DRS)**. DRS takes user and device information from DA and checks if a user is authorized to register a device. If a user is authorized to use DRS, the DA chooses a miner from the BC network. Next, the miner in the BC network generates a signature token by cryptographically signing the user and device information. After that, DRS returns this signature token to the DA. Finally, the selected miner generates a genesis block for the newly registered device and propagates the block across the network. DA stores the signature token and then directly communicates with the blockchain using this token. The role of DA consists of seven steps, as described in Figure 2.

1) **Device Registration:** The Device Agent registers a device with the BC via its DRS and acquires a signature token from the BC network. Next, the device agent discovers miner nodes from the BC network using a peer discovery algorithm. Next, the DA randomly chooses a miner node to register itself on the BC network.
   After selecting miner nodes, DA sends a gRPC(remote procedure call) request to the nominated miner node. The miner node provides the DA with a token. This token is pre-generated and securely provided to the user who owns the Device Agent. Every device agent obtains a unique token upon registration in the BC network. The BC network recognizes the DA through its token. The selected miner node verifies every token and digitally signs it. One of the miner nodes creates a genesis block for every Device Agent, validates it, and propagates the block to the network. The token is stored in the local database of the Device Agent. The DA needs to include the token for all future requests with the gRPC calls to the blockchain network. The token generation process is presented in Algorithm 1.
   Similarly, the Device Agent calls a gRPC service called Device Registration Service (DRS) to register an IoT device. To call the DRS service, the DA needs to send a request message that includes the username, password, and deviceId (identifier of the DA). The miner nodes then verify the username and password recorded in their ledger. If the user is authorized to

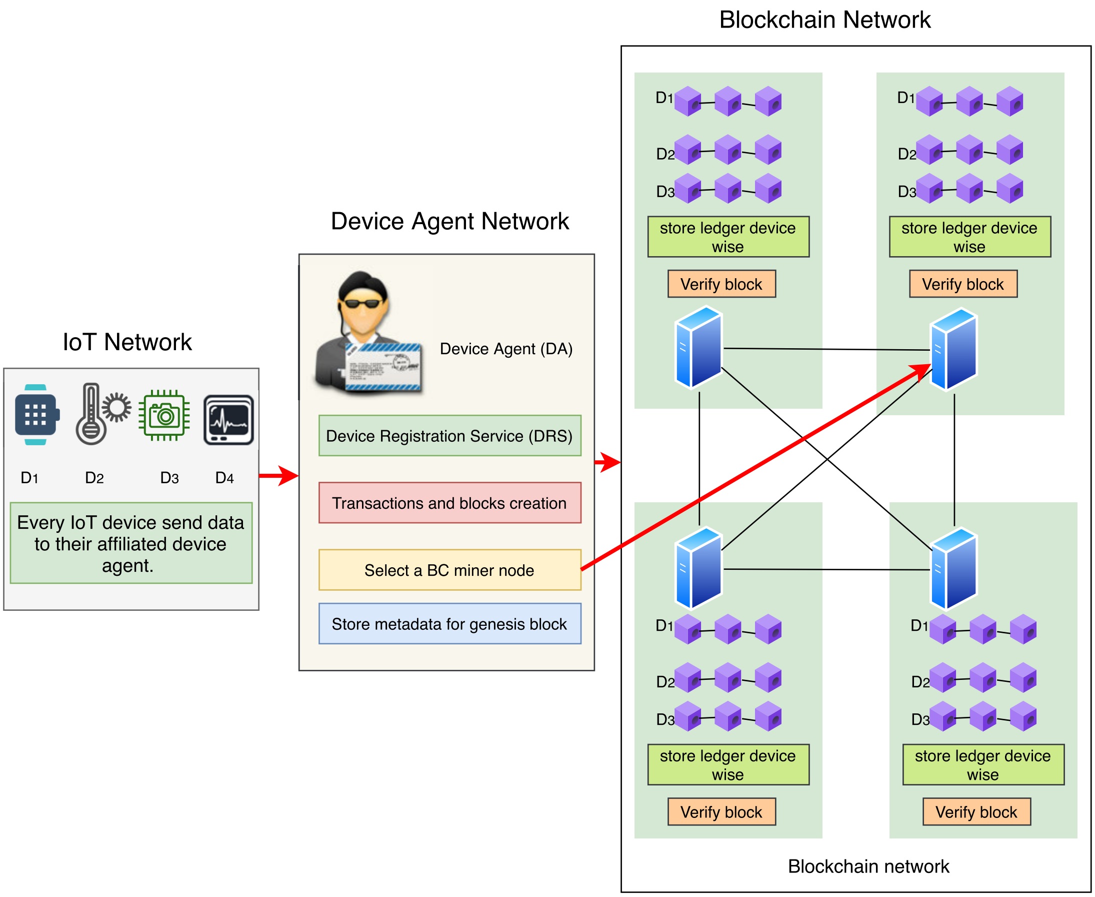
Blockchain Network
Device Agent Network
IoT Network
D1
D2
D3
D4
Every IoT device send data to their affiliated device agent.
Device Agent (DA)
Device Registration Service (DRS)
Transactions and blocks creation
Select a BC miner node
Store metadata for genesis block
store ledger device wise
Verify block
Blockchain network

FIGURE 1. High-level view of the proposed BC-based IoT data management architecture.

register a new device, the DRS then generates a signature token, cryptographically signing information regarding the user and device. Lastly, one of the miners returns this signature token to the DA.

Algorithm 1: Device Registration Signature Token Generation

```
if $username\ and\ password\ matched$ then
    $hash \leftarrow Sha256(username + deviceID)$;
    $r, s \leftarrow ecdsa.Sign(rand.Reader, ecdsa.Privatekey, hash[:])$;
    $token \leftarrow append(r.bytes(), s.bytes() \ldots)$;
    return $token$;
else
    return $Unauthorized$;
end
```

After having the device signature token, the Device Agent stores the token on its local storage. Any further gRPC calls made by this device agent must include this token. After returning the token to the DA, the miner node creates a genesis block, stores the genesis block in its local storage, and propagates the block to other miner nodes in the network. The block generation process will be discussed in later sections.

2) **Synchronization:** After successfully registering with the blockchain network, the DA attempts to synchronize its local chain of blocks with the blockchain network. In the device registration process, a genesis block was created; DA downloads the genesis block and synchronizes itself with the Blockchain Network. This process consists of several steps. First, DA discovers miners from a portion of the BC network. Next, the DA runs a miner node selection algorithm. After that, DA downloads the necessary blocks from the selected miner node and synchronizes its ledger with the BC ledger.
3) **Block Creation:** The Device Agent makes a Block with transactions collected from IoT Device. The DA then encrypts and assigns cryptographic signatures to the block. DA is responsible for communicating with the IoT device. DA collects data from IoT device, creates blocks, encrypt the blocks, and digitally signs them. A block consists of the fields described on the following page.
    a) PrevHash: This refers to a pointer that links its previous block. This contains the cryptographic hash code of a previously committed block.

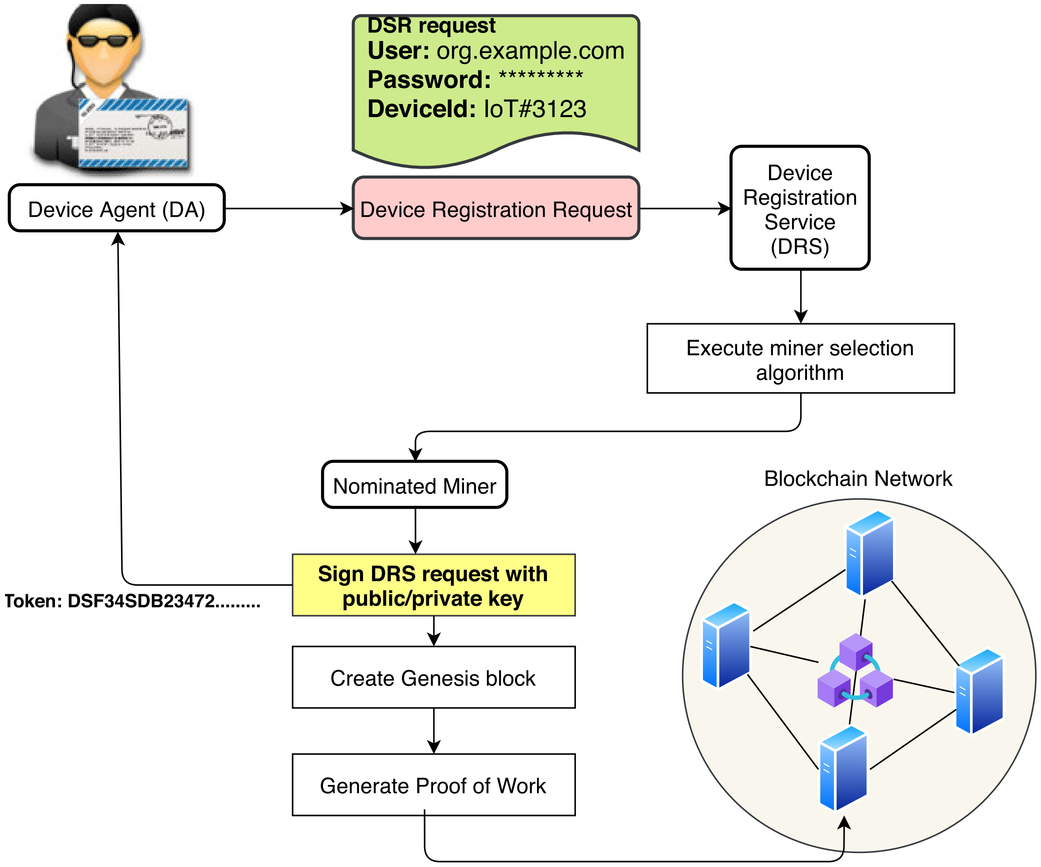
DSR request
User: org.example.com
Password: *********
DeviceId: IoT#3123
Device Agent (DA)
Device Registration Request
Device Registration Service (DRS)
Execute miner selection algorithm
Nominated Miner
Blockchain Network
Sign DRS request with public/private key
Token: DSF34SDB23472.........
Create Genesis block
Generate Proof of Work

FIGURE 2. Process of device registration.

b) Hash: A hash function such as SHA256 produces a fixed size of code by digesting any length of contents.

c) Nonce: A number added to a hashed or encrypted—block in a blockchain that, when rehashed, meets the difficulty level restrictions. The nonce is the number that blockchain miners are solving for.

d) Signature: The owner of the block or transactions electrically signs the contents to ensure the integrity and authenticity of the block or transaction.

e) Token: A unique identifier of an IoT device or Device Agent

f) PublicKey: Cryptographic key used for signing block.

g) Transactions: Indicates a list of transactions.

DA will assign the values for PrevHash, Signature, Token, PublicKey, and Transaction fields on a block.

4) **Generation of Proof of Work:** The nominated Miner updates the block by inserting the hash of the most recent mined block and increasing the nonce into the block. Then, the Miner generates a target hash with a certain number of leading zeroes (called proof of work) by incrementing a variable field called the nonce of the block. The algorithm of Proof of Work is presented in Algorithm 3. The Miner generates the target hash and then broadcasts the block to the Blockchain network. In Bitcoin, Proof of Work in digital cryptocurrencies consumes massive processing power because all of the miners compete to be the first to generate the target

Algorithm 2: Network Discovery Algorithm

```
ConnectedNodes ← []Connection;
queue ← []String;
queue ← initialKnownMinerIPAddress;
while queue is not empty do
    address ← queue.pop();
    addressList ← network.getNeighbours();
    for addr in addressList do
        if addr isnotinConnectedNodes then
            queue.append(addr);
            conn ← network.Connect(addr);
            ConnectedNodes[conn.Target()] ← conn;
        end
    end
end
return ConnectedNodes;
```

hash of the block to prevent tampering with the record. In our architecture, we propose to select a Miner based on a heuristic derived from a Miner's performance.

As only one miner node of the network mines a block and not every node competes with each other, all the nodes should develop a consensus mechanism to validate a block. We use the Proof of Work Algorithm to validate a block in our proposed method. A block is acceptable to all the nodes if it meets the requirement of a certain amount of work done to create this block. Algorithm 3 demonstrates the algorithm used to give proof of work to the block.

Algorithm 3: Proof of Work Algorithm

```
$nonce \leftarrow 0$ ;
$hash \leftarrow []$;
while  $condition$ do
    $data \leftarrow HashOfBlock$;
    $hash \leftarrow Sha256(data)$;
    if $leadingZeroCount(hash) \leq difficulty$ then
        $nonce \leftarrow nonce + 1$;
    else
        Break;
    end
end
return $hash$;
```

5) **Verify Block:** All the nodes in Blockchain verify the block and add the block to the current Blockchain.
6) **Store Block:** Finally the Device Agent retrieves the block from the blockchain and store it to local chain for further processing.

C. BLOCKCHAIN NETWORK

The last part of the architecture is the blockchain network. In the proposed architecture, every miner node maintains connectivity to other miner nodes, thus creating a start topology of the p2p network, which we call the blockchain network. When a new miner wants to join the network, it requires knowing at least one of the IP addresses of a miner node in the network. The new miner node then uses this IP address to discover the whole network and connects to them.

A miner node has multiple services developed with gRPC. gRPC is a modern, high-performance RPC framework that can run in any environment. gRPC technology can efficiently connect services in and across data centers with pluggable support for load balancing, tracing, health checking, and authentication. Each miner node contains a gRPC server and gRPC client. gRPC server is responsible for returning a response to the client request, such as miner IP address, neighbor IP address, chain sync request, etc. On the other hand, the gRPC client is able to request other miners for different necessary resources. More details about these services will be discussed in later sections.

1) MINER NODES IN BC

The DA participates in storing streamed data in a Block. Rather than maintaining a single Blockchain to store all transactions for all devices together, our approach stores the events for a device in multiple Blockchains dedicated solely to an IoT device. Data generated from an IoT device is stored in a ledger that is solely maintained for that IoT device.

Blockchain Miners in cryptocurrencies aggregate transactions from different senders and make a Block that contains a certain amount of transactions, such as 1MB in Bitcoin. All miners consume a large amount of power and computational resources to generate a key of the required complexity. The DA in our architecture selects a single miner rather than having all miners compete to win the privilege of preparing the Block and inserting it in a single chain. By having the Miner selected by a DA, there is less delay and cost than having the Blockchain nodes compete to select a Miner. The DA randomly selects a Miner or applies an algorithm based on CPU resources available and prior track record [5].

a: MINER SELECTION PROCESS

The device agent communicates/send write or read requests to a leader that is elected by the Raft algorithm [44]. Our Raft algorithm-based consensus algorithm is described below. The Raft algorithm involves much traffic with the increasing number of miner nodes. To reduce the network traffic, we select a set of healthy miners using the approach in [13]. In our proposed consensus protocol, a blockchain node might be one of the states: leader, candidate, and follower. In this protocol, the selected group of nodes can become a participant in the leader election. All the other nodes remain as a follower of the leader node.

1) A set of healthy is selected by the system using the TOPSIS methods according to Uddin et al. [13]. This set of nodes is called competitors that participate in running the Raft algorithm.
2) A node from the healthy group can only be the candidate to be a leader. All other nodes that cannot be the candidate/competitors (general nodes) and the candidates that cannot be a leader are the followers, where competitors/non-elected candidates are the follower of the leader node, and the general nodes are the follower of its nearby competitor node.
3) A competitor node is a candidate when it takes part in the leader election. Each candidate node sets a random time at every term—the node whose timer expires requests votes from other candidates. A candidate node rejects a vote request if the sender's term is less than that of the receiver candidate.
4) A node declares itself as the leader node if it receives two-thirds of the votes. Then, the leader node keeps sending a heartbeat message to the competitors. If a competitor does not get a heartbeat within its waiting time, the competitor becomes a candidate and requests a vote from other competitors for being a leader.

5) The DA as a client sends registration requests and transactions to the leader for making a block. After performing verification, the leader sends the block to all the followers to update their ledger. The followers generate the hash code of the block after adding it to the current chain and send the leader the hash code. If the leader receives the same hash code from two-thirds of the followers, the leader inserts the block in its local ledger.

b: TIME COMPLEXITY OF THE MINER SELECTION ALGORITHM

The miner selection algorithm involves a TOPSIS method to choose a certain number of healthy node in terms of their performances in the network. The time complexity of the TOPSIS algorithm is $O(n^2)$. The selected numbers of node participate in a voting process to select their leader which requires around $O(n)$ time. The elected leader leader needs to send a heartbeat message to its every followers. To propagate the message, the required time is around $O(n)$. The overall time complexity of modified consensus protocol is presented in Table 2. Further, to generate token, we use SHA-256 algorithm to hash DA's identity, and SHA-256 algorithm complexity is O(n); where n is the length of data.

TABLE 2. **Complexity analysis of miner selection process.**

TOPSIS analysis complexity	O(n^2)
Voting with n number of nodes	O(n)
Leader declaration	O(1)
Leader registration prop. time	O(n)
Overall Complexity	O(n^2)+O(n)+O(1)+O(n)=O(n^2)

IV. IMPLEMENTATION OF THE PROPOSED ARCHITECTURE

To implement the proposed system, we have used different technologies and tools that include protocol buffers, gRPC, BadgerDB, Docker, Golang programming language, and Visual Studio Code IDE. The design technologies for implementing the proposed blockchain system is presented in Figure 4. Protocol buffers refer to a method of serializing structured data. Protocol buffers are used for storing data while network nodes communicate with each others.

A. ENVIRONMENT SETTING

Using gRPC, a client application calls a method on a server application running on a different machine. gRPC defines a service by declaring the remote-callable methods, their parameters, and their return types. The server implements an interface for executing a gRPC server in order to reply to client requests. The client has a stub (in some languages, merely a client) that provides the same interface as the server. Various programming languages such as Java, Go, Python, and Ruby can be used to develop gRPC technologies. BadgerDB is a database written in the Go programming language. BadgerBD is a key-value (KV) database that is embeddable, persistent, and quick. BadgerBD is the database on which Dgraph, a fast, distributed graph database, is built.

Docker is a collection of platform as a service (PaaS) tools that utilize OS-level virtualization to distribute software in containers. Containers are segregated from one another and ship with their own software, libraries, and configuration files; they can communicate over well-defined channels. All containers are managed by a single kernel of the operating system and hence consume fewer resources than virtual machines.

B. IMPLEMENTATION OF A PEER-TO-PEER NETWORK

The Bitcoin (BTC) blockchain is constructed on a decentralized peer-to-peer (P2P) network, which is used to broadcast vital information such as BTC transactions and blockchain updates. Numerous studies have described the BTC blockchain from multiple angles, including transaction throughput and the availability of public distributed ledgers, due to its significance and popularity (DLT). However, widespread adoption of standardized approaches for designing P2P apps is not yet possible. Therefore, a fully functional peer-to-peer network must be created in order to enable blockchain functionality on top of it.

We plan to develop a blockchain network that is efficient and lightweight. The latency of the REST API is substantially lower than that of data serialization and deserialization in JSON format. gRPC serializes and deserializes data into binary and strongly couples the server-client system, requiring both parties to employ the same data format. This makes gRPC considerably faster than REST. Thus, we created a peer-to-peer network using gRPC and the programming language Goland.

Our implemented peer-to-peer network has a mesh topology in which each node or device is directly connected. Figure 5 depicts a visual depiction of the mesh topology in its entirety. If a new node wishes to join the network, it conducts a node discovery process and creates bidirectional connections with all existing nodes. If a connection fails, both nodes will automatically rebuild it. The algorithm for node finding is detailed in this section.

We implemented BC on top of a p2p network where participant node in network is given some set of functionality. These are

- SendAddress: Returns a set of ip addresses connected to the node as response.
- GetAddress: Get a set of ip addresses from another node.
- FullHeight: Returns total height of its local chain of blocks as response.
- Height: Returns height of a chain for a particular IoT device as response.
- GetFullChain: Downloads chain from other node.
- PropagateBlock: Propagates a verified Block to its neighbour nodes.
- Token: Used by Device Agent to register an IoT device and get a token as response.

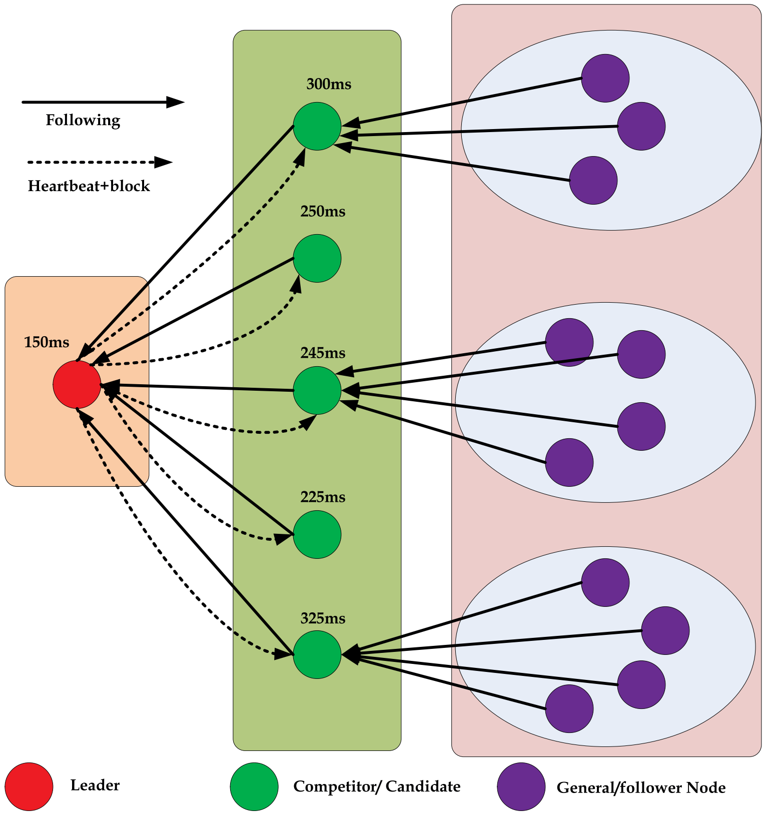

FIGURE 3. The miner selection and consensus algorithm.

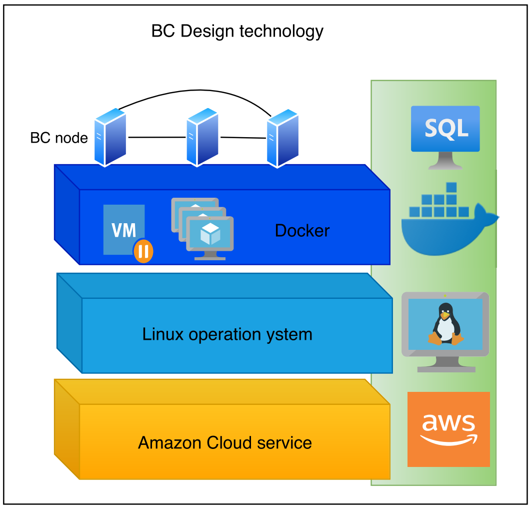

FIGURE 4. The blockchain implementation technology.

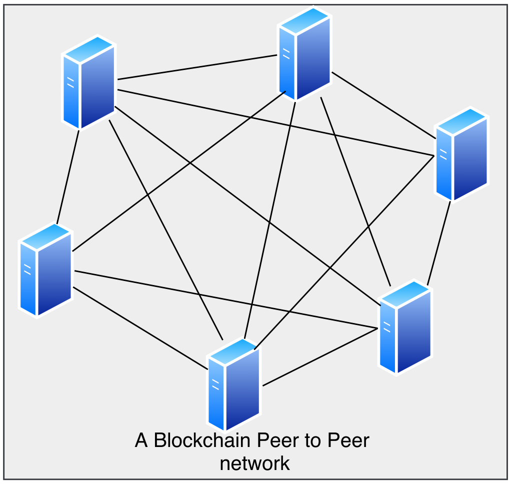

FIGURE 5. Fully connected mesh topology.

- Mine: Used by Device Agent to send a unverified block to the BC network

C. SYSTEM CONFIGURATION AND PERFORMANCE ANALYSIS

A sandbox is an isolated testing environment that allows users to execute programs or data without affecting the application, system, or platform on which they are running. Among the various existing sandboxing tools, Docker is a collection of platforms that use OS-level virtualization to distribute software in packages known as containers. Containerization of software or services enables the creation of predictable environments that are isolated from other programs, and it is simple to ship software or services without having to worry about the underlying operating system.

To test our developed framework, we containerized the entire application with Docker and ran multiple instances of the application on Google Cloud. Globally, the Google Cloud delivers on-demand CPU services. Multiple miner nodes were deployed on various CPUs in multiple locations of the world, and a BC network of multiple miner nodes was established. This testing methodology provides the ability to evaluate the performance of the proposed BC for IoT in real-world circumstances.

To create test results and evaluate the performance of the proposed architecture, we must configure our designed BC system. The subsequent sections discuss the configuration of the custom blockchain. Then, we examined the system's performance based on a variety of parameters, such as block production time, transaction throughput, consensus time, etc. Multiple instances of a miner node have been spawned on a Linux computer that operates in numerous Virtual Sandboxes. After the blockchain has been initialized, we must construct an instance of the Device Agent and register an IoT device with the blockChain. The Device Agent must synchronize its local ledger with the BC network's global ledger.

D. PERFORMANCE ANALYSIS

To simulate our implementation, we employ a 1.4 GHz quad-core 8th generation Intel Core i5 CPU with Turbo Boost up to 3.9 GHz and a block size range of 1 KB, 10 KB, 20 KB, and 100 KB. We conducted the simulation for one thousand blocks of each block size and calculated the average. To compute throughput and propagation latency, we altered the number of miner nodes on a blockchain network. As our codebase can be packaged in a Docker container, it can be executed on any Docker-compatible machine. For instance, Google Cloud offers a service known as Cloud Run. We ran up to 15 Docker containers in Google Cloud Run, computed the propagation delay of blocks from the node, and got the average. Table 3 displays the simulation parameters of our customized blockchain.

TABLE 3. Simulation parameters.

Simulation parameters	Values
IoT device CPU capacity	10000 MIPS
Miners CPu capacity	82,300 MIPS
IoT device RAM capacity	512 MB
Miners RAM capacity	8 Gb
Iot device Bandwidth	150Mbps
Miner Bandwidth	300 Mbps
IoT device Power consumption rate	10W
Miner device Power Consumption Rate(per Hour)	400W
Transaction Size	100-200 bytes
Block Size	1000-2000 bytes

1) BLOCK GENERATION TIME

Average Block Validation Time is defined as the average time taken to validate a single block in a miner node. This time depends directly on the block size and computational power of the node. The average time required to generate a block in our architecture is presented in Figure 6. The blockchain time is increasing with the increase of block size. According to graph presented in Figure 6, the generation time is shortest when the block size is 1KB, and it is longest when the block size is 100KB.

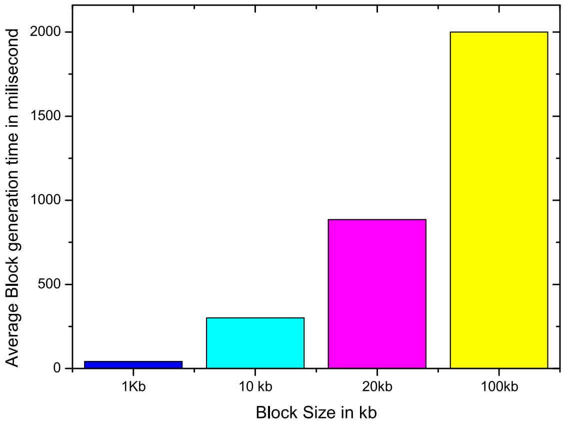

FIGURE 6. Average block generation time.

2) TRANSACTION THROUGHPUT

Transaction throughput is the number of committed transactions per second. This performance analysis was obtained by varying the number of miner nodes in the blockchain network. Our codebase can be wrapped in a Docker container, allowing it to run on any Docker-compatible machine. Google Cloud offers us a service known as Cloud Run. We ran up to 15 Docker containers in Google Cloud Run, and for each instance of the node, we counted the number of transactions committed per second and got an average. The transaction throughput of the proposed BC is illustrated in Figure 7. The graph indicated that the throughput of the proposed blockchain grows dramatically as the number of nodes increases, whereas the throughput of the existing blockchain drops as more nodes compete to mine a block.

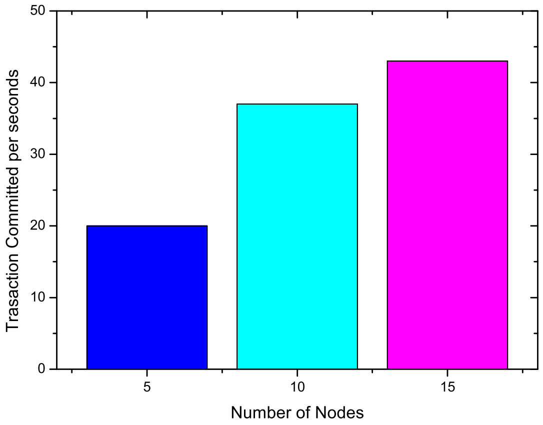

FIGURE 7. Transaction throughput.

3) CONSENSUS COST TIME

Consensus Cost Time is the average time taken to generate proof of work for a single block. To calculate the performance analysis, we varied the number of miner nodes in the blockchain network. The consensus time of the proposed time

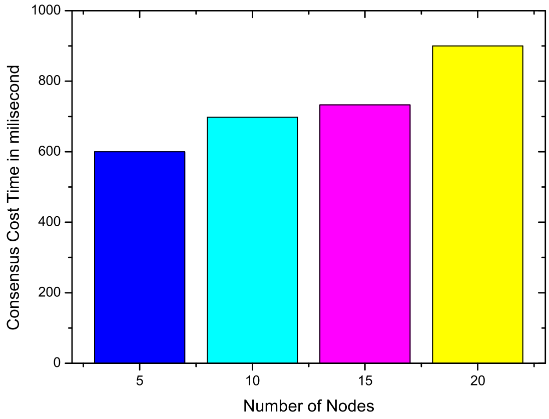

FIGURE 8. Consensus cost time.

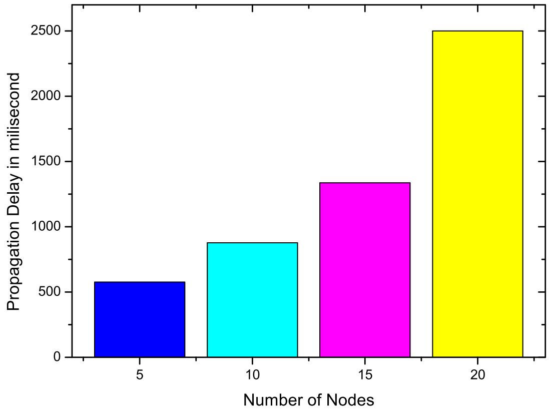

FIGURE 9. Propagation delay.

is presented in Figure 8. As the number of nodes participating in voting and validating a block increases, the consensus time also increases. The graph demonstrates that the consensus time does not vary considerably between 10 and 15 nodes. To determine the best number of nodes, the experiment must be done with a larger number of nodes.

4) PROPAGATION DELAY

Propagation delay is the average time taken for each block to be propagated to the miner node. The propagation delay of the BC is depicted in Figure 9. As fewer nodes participate in the system's committing block, the propagation delay will reduce. The graph demonstrates that as the number of nodes increases, the propagation delay increases dramatically due to the selection of followers in the consensus algorithm.

5) PEER DISCOVERY TIME

Peer discovery time indicates the time required for a miner to discover its peer miners in the BC peer-to-peer network. Figure 10 illustrates the peer discovery delay. If there are more nodes in a network, it takes longer to discover peers.

The overall performance analysis shows the feasibility of using our architecture to manage IoT data on the BC. We compared the performance of our framework with popular blockchain technology with respect to average block

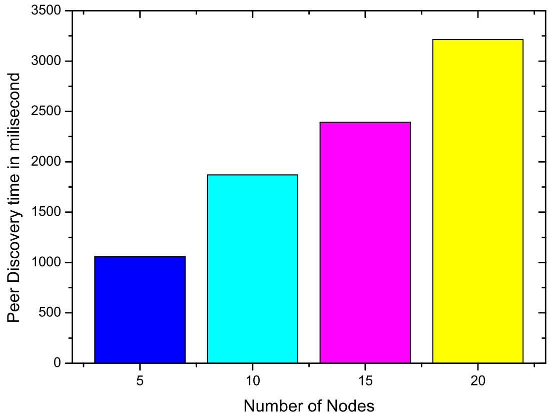

FIGURE 10. Peer discovery delay.

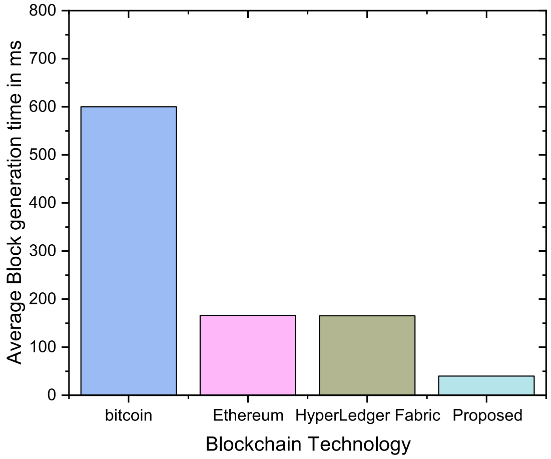

FIGURE 11. Comparison of block generation time.

generation time in our experimental setting where the number of BC nodes is 10. Figure 11 shows the comparison of block generation time. Our framework showed less block generation time than other blockchain approaches because we employed a lightweight consensus protocol where not many nodes are responsible to validate and verify a block. The comparison between our proposed BC and popular existing BCs with respect to consensus time is presented in Table 4.

TABLE 4. The comparison of consensus time.

Blockchain Technology	Consensus Protocol	Time(s)
Ethereum	Proof of stake	300
Bitcoin	Proof of work	2400
Hyperledger fabric	PBFT	0.3314
Proposed BC	RAFT	50.5

Every device that is registered to the Blockchain Network receives a special device identification that allows access to the data that is generated by that device. Data on one device cannot be accessed by another. Currently, when saving raw data in the miner node, we don't encrypt it, which is something that might be done to improve the proposed framework. As opposed to this, edge IoT devices can pre-encrypt data before transmitting it to the blockchain network; miner nodes will simply store the data in its current state, and the edge device will be in charge of decrypting it. The proposed architecture employs the Grpc protocol, which can be encrypted, for all forms of communication between edge devices and nodes, ensuring the security of data transmission.

V. CONCLUSION

Instead of keeping a single database on a central server, BC establishes a distributed ledger of transactions that are shared across all network nodes. Participants who are enrolled in the BC system are able to record and review transactions. As each member in the BC maintains a local copy of the whole ledger, the BC's ledger is tamper-proof, meaning that no one can alter anything entered on the BC. By eliminating the requirement for a centralized authority, the BC can decrease both installation and operation expenses. BC has been implemented in a wide range of applications, such as cryptocurrency, the Internet of Things (IoT), government, and economics. Due to their high computational cost and low throughput, however, the existing BCs deployed in cryptocurrencies are not suitable for storing and maintaining IoT transactions. In this paper, we have proposed a lightweight BC-based framework with a modified raft algorithm-based consensus protocol to store and manage IoT data. We design an IoT Device Agent (DA) to manage IoT on behalf of IoT devices for inserting IoT data into BC. We have designed an IoT framework that can accommodate multiple chains of the ledger with the aid of the device Agent. Finally, we have implemented the proposed BC leveraged IoT framework using Golang on the Google Cloud to analyze its performance.

Although we simulated our framework and analyzed performance extensively, the system has not been validated using real IoT devices. In this architecture, a Device Agent manages the blockchain on behalf of several IoT devices. As a result, the DA is susceptible to several cyberattacks, including denial of service, ransomware attacks, and single points of failure. Further, the security strength of the architecture is not validated.

There are several scopes to improve the proposed system in the future. Our first future work will test the system by incorporating IoT devices with the Device Agent and running the BC on several computers. Our second future work is to devise a decentralized Device Agent so that the architecture can withstand major cyber attacks. Finally, the security strength of the proposed architecture will be analyzed in the future.

REFERENCES

[1] M. A. Uddin, A. Stranieri, I. Gondal, and V. Balasubramanian, "An efficient selective miner consensus protocol in blockchain oriented IoT smart monitoring," in *Proc. ICIT*, Feb. 2019, pp. 1135–1142.

[2] H.-N. Dai, Z. Zheng, and Y. Zhang, "Blockchain for Internet of Things: A survey," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8076–8094, Oct. 2019.

[3] M. A. Uddin, A. Stranieri, I. Gondal, and V. Balasubramanian, "Blockchain leveraged decentralized IoT eHealth framework," *Internet Things*, vol. 9, Mar. 2020, Art. no. 100159.

[4] J. Ni, K. Zhang, X. Lin, and X. S. Shen, "Securing fog computing for Internet of Things applications: Challenges and solutions," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 1, pp. 601–628, 1st Quart., 2018.

[5] M. A. Uddin, A. Stranieri, I. Gondal, and V. Balasubramanian, "Continuous patient monitoring with a patient centric agent: A block architecture," *IEEE Access*, vol. 6, pp. 32700–32726, 2018.

[6] M. A. Uddin, A. Stranier, I. Gondal, and V. Balasubramanian, "A patient agent to manage blockchains for remote patient monitoring," *Stud. Health Technol. Inform.*, vol. 254, pp. 105–115, Apr. 2018.

[7] M. Ashraf Uddin, A. Stranieri, I. Gondal, and V. Balasubramanian, "Dynamically recommending repositories for health data: A machine learning model," in *Proc. Australas. Comput. Sci. Week Multiconf.*, Feb. 2020, pp. 1–10.

[8] S. S. Sarmah, "Understanding blockchain technology," *Comput. Sci. Eng.*, vol. 8, no. 2, pp. 23–29, 2018.

[9] Q. Zhou, H. Huang, Z. Zheng, and J. Bian, "Solutions to scalability of blockchain: A survey," *IEEE Access*, vol. 8, pp. 16440–16455, 2020.

[10] M. A. Uddin, A. Stranieri, I. Gondal, and V. Balasubramanian, "A decentralized patient agent controlled blockchain for remote patient monitoring," in *Proc. Int. Conf. Wireless Mobile Comput., Netw. Commun. (WiMob)*, Oct. 2019, pp. 1–8.

[11] N. Tariq, M. Asim, F. Al-Obeidat, M. F. Farooqi, T. Baker, M. Hammoudeh, and I. Ghafir, "The security of big data in fog-enabled IoT applications including blockchain: A survey," *Sensors*, vol. 19, no. 8, p. 1788, 2019.

[12] *Cisco Global Cloud Index: Forecast and Methodology, 2015–2020*, Cisco Vis. Netw., Cisco Public, San Jose, CA, USA, 2016.

[13] M. A. Uddin, A. Stranieri, I. Gondal, and V. Balasurbramanian, "A lightweight blockchain based framework for underwater IoT," *Electronics*, vol. 8, no. 12, p. 1552, 2019.

[14] M. A. Uddin, A. Stranieri, I. Gondal, and V. Balasubramanian, "Rapid health data repository allocation using predictive machine learning," *Health Informat. J.*, vol. 26, no. 4, pp. 3009–3036, Dec. 2020.

[15] I. Makhdoom, M. Abolhasan, and W. Ni, "Blockchain for IoT: The challenges and a way forward," in *Proc. 15th Int. Joint Conf. e-Bus. Telecommun. (ICETE)*, 2018, pp. 1–12.

[16] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," Decentralized Bus. Rev., 2008, p. 21260.

[17] M. S. Ali, M. Vecchio, M. Pincheira, K. Dolui, F. Antonelli, and M. H. Rehmani, "Applications of blockchains in the Internet of Things: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 2, pp. 1676–1717, 2nd Quart., 2019.

[18] P. C. Tang, J. S. Ash, D. W. Bates, J. M. Overhage, and D. Z. Sands, "Personal health records: Definitions, benefits, and strategies for overcoming barriers to adoption," *J. Amer. Med. Informat. Assoc.*, vol. 13, no. 2, pp. 121–126, 2006.

[19] J. Archenaa and E. A. M. Anita, "Interactive big data management in healthcare using spark," in *Proc. 3rd Int. Symp. Big Data Cloud Comput. Challenges (ISBCC)*, in Smart Innovation, Systems and Technologies, vol. 49, V. Vijayakumar and V. Neelanarayanan, Eds. Cham, Switzerland: Springer, 2016. [Online]. Available: https://doi-org.jnu.idm.oclc.org/10.1007/978-3-319-30348-2_21

[20] A. A. Alomar, M. Z. A. Bhuiyan, A. Basu, S. Kiyomoto, and M. S. Rahman, "Privacy-friendly platform for healthcare data in cloud based on blockchain environment," *Future Gener. Comput. Syst.*, vol. 95, pp. 511–521, Jun. 2019.

[21] M. A. Uddin, A. Stranieri, I. Gondal, and V. Balasubramanian, "Blockchain leveraged task migration in body area sensor networks," in *Proc. 25th Asia–Pacific Conf. Commun. (APCC)*, Nov. 2019, pp. 177–184.

[22] A. D. Dwivedi, R. Singh, K. Kaushik, R. R. Mukkamala, and W. S. Alnumay, "Blockchain and artificial intelligence for 5G-enabled Internet of Things: Challenges, opportunities, and solutions," *Trans. Emerg. Telecommun. Technol.*, p. e4329. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/ett.4329, doi: 10.1002/ett.4329.

[23] A. D. Dwivedi, R. Singh, U. Ghosh, R. R. Mukkamala, A. Tolba, and O. Said, "Privacy preserving authentication system based on non-interactive zero knowledge proof suitable for Internet of Things," *J. Ambient Intell. Hum. Comput.*, vol. 13, no. 10, pp. 4639–4649, 2022.

[24] T. G. Jiang, H. Fang, and H. Wang, "Blockchain-based Internet of Vehicles: Distributed network architecture and performance analysis," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4640–4649, Jun. 2019.

[25] C. Liang, B. Shanmugam, S. Azam, A. Karim, A. Islam, M. Zamani, S. Kavianpour, and N. B. Idris, "Intrusion detection system for the Internet of Things based on blockchain and multi-agent systems," *Electronics*, vol. 9, no. 7, p. 1120, Jul. 2020.

[26] B. Mbarek, N. Jabeur, T. Pitner, and A.-U.-H. Yasar, "MBS: Multilevel blockchain system for IoT," *Pers. Ubiquitous Comput.*, vol. 25, no. 1, pp. 247–254, Feb. 2021.

[27] G. Fortino, F. Messina, D. Rosaci, and G. M. L. Sarné, "Using blockchain in a reputation-based model for grouping agents in the Internet of Things," *IEEE Trans. Eng. Manag.*, vol. 67, no. 4, pp. 1231–1243, Nov. 2020.

[28] B. E. Sabir, M. Youssfi, O. Bouattane, and H. Allali, "Towards a new model to secure IoT-based smart home mobile agents using blockchain technology," *Eng., Technol. Appl. Sci. Res.*, vol. 10, no. 2, pp. 5441–5447, Apr. 2020.

[29] N. Cam-Winget, A.-R. Sadeghi, and Y. Jin, "Invited: Can IoT be secured: Emerging challenges in connecting the unconnected," in *Proc. 53rd ACM/EDAC/IEEE Annu. Design Autom. Conf.*, Jun. 2016, pp. 1–6.

[30] H. Shafagh, L. Burkhalter, A. Hithnawi, and S. Duquennoy, "Towards blockchain-based auditable storage and sharing of IoT data," in *Proc. Cloud Comput. Secur. Workshop*, Nov. 2017, pp. 45–50.

[31] A. Reyna, C. Martín, J. Chen, E. Soler, and M. Díaz, "On blockchain and its integration with IoT. Challenges and opportunities," *Future Gener. Comput. Syst.*, vol. 88, pp. 173–190, Nov. 2018.

[32] W. Chen, Z. Zheng, J. Cui, E. Ngai, P. Zheng, and Y. Zhou, "Detecting Ponzi schemes on ethereum: Towards healthier blockchain technology," in *Proc. World Wide Web Conf. World Wide Web*, 2018, pp. 1409–1418.

[33] S. Huh, S. Cho, and S. Kim, "Managing IoT devices using blockchain platform," in *Proc. 19th Int. Conf. Adv. Commun. Technol. (ICACT)*, 2017, pp. 464–467.

[34] M. Mylrea and S. N. G. Gourisetti, "Blockchain for smart grid resilience: Exchanging distributed energy at speed, scale and security," in *Proc. Resilience Week (RWS)*, Sep. 2017, pp. 18–23.

[35] R. Bdiwi, C. de Runz, S. Faiz, and A. A. Cherif, "Towards a new ubiquitous learning environment based on blockchain technology," in *Proc. IEEE 17th Int. Conf. Adv. Learn. Technol. (ICALT)*, Jul. 2017, pp. 101–102.

[36] B. Boeser, "Meet TrueRec by SAP: Trusted digital credentials powered by blockchain| SAP news center," Tech. Rep., 2017. [Online]. Available: https://news.sap.com/2017/07/meet-truerec-by-sap-trusted-digital-credentials-powered-by-blockchain/

[37] R. McMillan, "Hacker dreams up crypto passport using the tech behind Bitcoin," WIRED, San Francisco, CA, USA, Tech. Rep., 2014.

[38] K. Bilal, S. U. R. Malik, O. Khalid, A. Hameed, E. Alvarez, V. Wijaysekara, R. Irfan, S. Shrestha, D. Dwivedy, M. Ali, U. S. Khan, A. Abbas, N. Jalil, and S. U. Khan, "A taxonomy and survey on Green data center networks," *Futur. Gener. Comput. Syst.*, vol. 36, pp. 189–208, Jul. 2014.

[39] S. Apte and N. Petrovsky, "Will blockchain technology revolutionize excipient supply chain management?" *J. Excipients Food Chem.*, vol. 7, no. 3, p. 910, 2016.

[40] L. Fouard, M. Duclos, and P. Lafourcade, "Survey on electronic voting schemes," Supported ANR Project AVOTÉ, Tech. Rep., 2007.

[41] D. A. Gritzalis, *Secure Electronic Voting*, vol. 7. New York, NY, USA: Springer, 2012.

[42] A. C. Yao, "Protocols for secure computations," in *Proc. 23rd Annu. Symp. Found. Comput. Sci. (SFCS)*, Nov. 1982, pp. 160–164.

[43] Y. Liu and Q. Wang, "An E-voting protocol based on blockchain," *IACR Cryptol. ePrint Arch.*, vol. 2017, no. 1043, p. 1043, 2017.

[44] D. Huang, X. Ma, and S. Zhang, "Performance analysis of the raft consensus algorithm for private blockchains," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 50, no. 1, pp. 172–181, Jan. 2020.

TARIQUE MOHAMMED NASRULLAH received the B.Sc. degree in computer science and engineering from Jagannath University, Dhaka, Bangladesh, where he is currently pursuing the M.Sc. degree in computer science and engineering. His research interests include blockchain and machine learning.

MD. MANOWARUL ISLAM received the B.Sc. and M.S. degrees in computer science and engineering from the University of Dhaka, Bangladesh, and the Ph.D. degree from the Department of Electrical and Communication Engineering, Okayama University, Japan. Currently, he is working as an Assistant Professor with the Department of Computer Science and Engineering, Jagannath University, Dhaka, Bangladesh. His research interests include machine learning, artificial intelligence, bioinformatics, and computer networking.

MD. ASHRAF UDDIN received the B.S. and M.S. degrees in computer science and engineering from the University of Dhaka, and the Ph.D. degree from Federation University, Australia. Currently, he is working as an Associate Professor with the Department of Computer Science and Engineering, Jagannath University, Dhaka, Bangladesh. His research interests include Blockchain, privacy and security in remote patient monitoring, machine learning, modeling, analysis, and optimization of protocols, and architectures for underwater sensor networks. He has published more than 40 papers at different international journals and conferences.

MD. ANISUZZAMAN KHAN received the B.Sc. degree in computer science and engineering from Jagannath University, Dhaka, Bangladesh, where he is currently pursuing the M.Sc. degree in computer science and engineering. His research interests include blockchain and machine learning.

MD. ABU LAYEK (Senior Member, IEEE) received the B.Sc. and M.Sc. degrees from the Department of ICT, Islamic University, Bangladesh, in 2004 and 2006, respectively, and the Ph.D. degree from Kyung Hee University, Republic of Korea. He is currently an Associate Professor with the Department of Computer Science and Engineering, Jagannath University, Dhaka, Bangladesh. His current research interests include cloud computing, the Internet of Things, machine learning, screen contents coding, and image quality assessment.

ANDREW STRANIERI (Member, IEEE) is currently a Researcher with the Centre for Informatics and Applied Optimisation, Federation University, Australia. His research in health informatics spans data mining in health, complementary and alternative medicine informatics, telemedicine, and intelligent decision support systems. He has authored over 250 peer-reviewed journals and conference papers and has published two books. He has been awarded seven Australian Research Council grants and supervised 15 Ph.D. students to completion.

EUI-NAM HUH (Senior Member, IEEE) received the B.S. degree from Busan National University, South Korea, the master's degree in computer science from The University of Texas at Arlington, TX, USA, in 1995, and the Ph.D. degree from Ohio University, Athens, OH, USA, in 2002. He is currently a Professor with the Department of Computer Science and Engineering, Kyung Hee University, South Korea. His research interests include cloud computing, the Internet of Things, future internet, distributed real-time systems, mobile computing, big data, and security. He is a Review Board of the National Research Foundation of Korea. He has also served many community services for ICCSA, WPDRTS/IPDPS, APAN Sensor Network Group, ICUIMC, ICONI, APIC-IST, ICUFN, and SoICT as various types of chairs. He is the Vice-Chairperson of the Cloud/Bigdata Special Technical Group of TTA, and an Editor of ITU-T SG13 Q17.